Regression Analysis II

Jin Young Choi
Seoul National University

Interim Summary

- linear regression
  - simple linear regression
  - multiple linear regression
- nonlinear regression
  - logistic regression
  - high-order regression
  - basis-function regression
- matrix form for regression
  - recursive least squares
- partial least squares
  - over-fitting and underfitting
  - bias/variance
  - principle component regression
  - partial least squares algorithm
  - ridge regression
  - lasso, elastic regression
- Gaussian process regression
- Kalman filtering

Parameter Estimation in Matrix form

- Least Squares Estimation

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|\boldsymbol{\epsilon}\|^2 = \|\boldsymbol{y} - \Phi\theta\|^2 \cong S(\theta)$$

Solution:

$$\nabla_\theta S(\theta) = 0 \text{ at } \hat{\theta}$$

$$\nabla_\theta (\boldsymbol{y} - \Phi\theta)^T (\boldsymbol{y} - \Phi\theta) = 0 \text{ at } \hat{\theta}$$

$$2\Phi^T \left(\boldsymbol{y} - \Phi\hat{\theta}\right) = 0$$

$$\Phi^T \boldsymbol{y} - \Phi^T\Phi\hat{\theta} = 0$$

$$\hat{\theta} = (\Phi^T\Phi)^{-1} \Phi^T \boldsymbol{y}$$

Parameter Estimation in Matrix form

- Least Squares Estimation

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\,\Phi^T\,\boldsymbol{y} \leftarrow \boldsymbol{y} = \Phi\theta + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I})$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_\mathrm{k} \end{bmatrix} = \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_k^T \end{bmatrix} \theta + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_\mathrm{k} \end{bmatrix}, \; \Phi_k = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1p} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{k1} & \phi_{k2} & \cdots & \phi_{kp} \end{bmatrix}$$

$$y_i = \phi_i^T\theta + \epsilon_i$$

$$y_i = \theta_0 + \theta_1\phi_{i1} + \theta_2\phi_{i2} + \cdots + \theta_p\phi_{i(p-1)} + \epsilon_i,$$

$$i = 1, \cdots, k, \cdots, n, \cdots$$

- Observation Matrix

$$\Phi_k = [\phi_1\ \phi_2\ \ \cdots\ \phi_k]^T \rightarrow \Phi_k^T\,\Phi_k = [\phi_1\ \phi_2\ \ \cdots\ \phi_k] \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_k^T \end{bmatrix} = \sum_{i=1}^{k} \phi_i\,\phi_i^T$$

$$\boldsymbol{y}_k = [y_1\ \ y_2\ \ \cdots\ y_k]^T$$

- Recursive Least Squares

$$\hat{\theta}_k = (\Phi_k^T\Phi_k)^{-1}\Phi_k^T\boldsymbol{y}_k \rightarrow \hat{\theta}_{k+1} = (\Phi_k^T\Phi_k + \phi_{k+1}\phi_{k+1}^T)^{-1}\Phi_{k+1}^T\boldsymbol{y}_{k+1}$$

Parameter Estimation in Matrix form

- Matrix Inversion Lemma

$$(A + BDC)^{-1} = A^{-1} - A^{-1}B(D^{-1} + CA^{-1}B)^{-1}C\,A^{-1}$$

Sherman-Morrison formula: $(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}$

- Recursive Least Squares

$$\hat{\theta}_{k+1} = (\Phi_k^T\Phi_k + \phi_{k+1}\phi_{k+1}^T)^{-1}\Phi_{k+1}^T\boldsymbol{y}_{k+1}$$

define $P_k \cong (\Phi_k^T\Phi_k)^{-1}$,

$$\hat{\theta}_{k+1} = (P_k^{-1} + \phi_{k+1}\phi_{k+1}^T)^{-1}\Phi_{k+1}^T\boldsymbol{y}_{k+1}$$

$$= \left(P_k - \frac{P_k\phi_{k+1}\phi_{k+1}^TP_k}{1+\phi_{k+1}^TP_k\phi_{k+1}}\right)\Phi_{k+1}^T\boldsymbol{y}_{k+1},$$ (no inverse)

define $G_k \cong \frac{P_k\phi_{k+1}}{1+\phi_{k+1}^TP_k\phi_{k+1}} \quad \Longrightarrow \quad P_{k+1} = P_k - \frac{P_k\phi_{k+1}\phi_{k+1}^TP_k}{1+\phi_{k+1}^TP_k\phi_{k+1}} = P_k - G_k\phi_{k+1}^TP_k$

Parameter Estimation in Matrix form

- Recursive Least Squares (cont.)

$$\hat{\theta}_{k+1} = (P_k - G_k\phi_{k+1}^T P_k)[\Phi_k^T \quad \phi_{k+1}]\begin{bmatrix}\mathbf{y}_k \\ y_{k+1}\end{bmatrix}$$

$$= (P_k - G_k\phi_{k+1}^T P_k)(\Phi_k^T \mathbf{y}_k + \phi_{k+1} y_{k+1})$$

$$= (I - G_k\phi_{k+1}^T)(P_k\Phi_k^T \mathbf{y}_k + P_k\phi_{k+1} y_{k+1})$$

$$= (I - G_k\phi_{k+1}^T)(\hat{\theta}_k + P_k\phi_{k+1} y_{k+1})$$

$$= \hat{\theta}_k - G_k\phi_{k+1}^T\hat{\theta}_k + P_k\phi_{k+1}y_{k+1} - G_k\phi_{k+1}^T P_k\phi_{k+1}y_{k+1}$$

$$= \hat{\theta}_k - G_k\phi_{k+1}^T\hat{\theta}_k + G_k y_{k+1} + G_k\phi_{k+1}^T P_k\phi_{k+1}y_{k+1} - G_k\phi_{k+1}^T P_k\phi_{k+1}y_{k+1}$$

$$\Phi_k = [\phi_1 \ \phi_2 \ \cdots \ \phi_k]^T$$
$$\mathbf{y}_k = [y_1 \ y_2 \ \cdots \ y_k]^T$$

$$P_k \cong (\Phi_k^T\Phi_k)^{-1}$$

$$G_k \cong \frac{P_k\phi_{k+1}}{1+\phi_{k+1}^T P_k\phi_{k+1}}$$

$$P_{k+1} = P_k - G_k\phi_{k+1}^T P_k$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + G_k(y_{k+1} - \phi_{k+1}^T\hat{\theta}_k),\ P_0 = \alpha\mathbf{I},\ \alpha \gg 0.$$

$$\hat{\theta} = \hat{\theta}_n \qquad G_k \cong \frac{P_k\phi_{k+1}}{1+\phi_{k+1}^T P_k\phi_{k+1}} \qquad P_{k+1} = P_k - G_k\phi_{k+1}^T P_k$$

Parameter Estimation in Matrix form

- Weighted Recursive Least Squares

$$\hat{\theta}_{k+1} = (\lambda \Phi_k^T \Phi_k + \phi_{k+1}\phi_{k+1}^T)^{-1} \Phi_{k+1}^T \boldsymbol{y}_{k+1}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + G_k(y_{k+1} - \phi_{k+1}^T \hat{\theta}_k),\ P_0 = \alpha \mathbf{I},\ \alpha \gg 0$$

$$\hat{\theta} = \hat{\theta}_n$$

$$G_k \cong \frac{\lambda^{-1} P_k \phi_{k+1}}{1 + \lambda^{-1} \phi_{k+1}^T P_k \phi_{k+1}}$$

$$P_{k+1} = \lambda^{-1} P_k - \lambda^{-1} G_k \phi_{k+1}^T P_k$$

$$\Phi_k = [\phi_1\ \phi_2\ \cdots\ \phi_k]^T$$
$$\boldsymbol{y}_k = [y_1\ y_2\ \cdots\ y_k]^T$$

$$P_k \cong (\Phi_k^T \Phi_k)^{-1}$$

Quality of Fit in Matrix form

- Regression model in matrix form

$$\boldsymbol{y} = \Phi\theta + \boldsymbol{\epsilon}$$

- Estimated parameter

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\,\Phi^T\,\boldsymbol{y} = \theta + (\Phi^T\Phi)^{-1}\,\Phi^T\boldsymbol{\epsilon} \text{ (unbiased estimate)}$$

- Confidence Interval

$$E\left(\hat{\theta}\right) = \theta,\; E\left((\theta-\hat{\theta})^T(\theta-\hat{\theta})\right) = E\boldsymbol{\epsilon}^T\Phi(\Phi^T\Phi)^{-1}(\Phi^T\Phi)^{-1}\,\Phi^T\boldsymbol{\epsilon}$$

$$= E\boldsymbol{\epsilon}^T\Phi\Phi^{-1}\Phi^{-T}\Phi^{-1}\Phi^{-T}\Phi^T\boldsymbol{\epsilon} = E\boldsymbol{\epsilon}^T\Phi^{-T}\Phi^{-1}\boldsymbol{\epsilon}$$

$$= Tr\left((\Phi^T\Phi)^{-1}\right)\sigma^2 \rightarrow \hat{\theta} = \theta \pm \alpha\sigma$$

- Prediction

$$\widehat{\boldsymbol{y}} = \Phi\hat{\theta} = \Phi\theta + \Phi(\Phi^T\Phi)^{-1}\,\Phi^T\boldsymbol{\epsilon} = \Phi\theta + \mathbb{H}\boldsymbol{\epsilon},$$

where $\mathbb{H}$ is symmetric and idempotent ($\mathbb{H}^2 = \mathbb{H}$), $\mathbb{H}\Phi = \Phi$.

$$\mathbb{H}\widehat{\boldsymbol{y}} = \mathbb{H}\Phi\theta + \mathbb{H}\boldsymbol{\epsilon} = \Phi\theta + \mathbb{H}\boldsymbol{\epsilon} = \widehat{\boldsymbol{y}}$$

- Residual vector : $\boldsymbol{e} = \boldsymbol{y} - \widehat{\boldsymbol{y}} = (\mathbf{I} - \mathbb{H})\boldsymbol{\epsilon}$

Quality of Fit in Matrix form

- Residual vector

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}} = (\mathbf{I} - \mathbb{H})\boldsymbol{\epsilon}$$

$$E(\boldsymbol{e}^T\boldsymbol{e}) = E(\boldsymbol{\epsilon}^T(\mathbf{I} - \mathbb{H})(\mathbf{I} - \mathbb{H})\boldsymbol{\epsilon}) = E(\boldsymbol{\epsilon}^T(\mathbf{I} - \mathbb{H})\boldsymbol{\epsilon})$$

$$= Tr(\mathbf{I} - \mathbb{H})E(\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}) = Tr(\mathbf{I} - \mathbb{H})\sigma^2$$

here

$$Tr(\mathbf{I} - \mathbb{H}) = Tr(\mathbf{I}) - Tr(\mathbb{H}) = n - Tr(\Phi(\Phi^T\Phi)^{-1}\,\Phi^T)$$

$$= n - Tr\left((\Phi^T\Phi)^{-1}\,\Phi^T\Phi\right) = n - (p+1), p+1: \#\ of\ parameters$$

hence

$$E(\boldsymbol{e}^T\boldsymbol{e}/(n-p-1)) = \sigma^2 \rightarrow \frac{\boldsymbol{e}^T\boldsymbol{e}}{n-p-1}$$: unbiased estimate of σ^2

- Coefficient of Determination

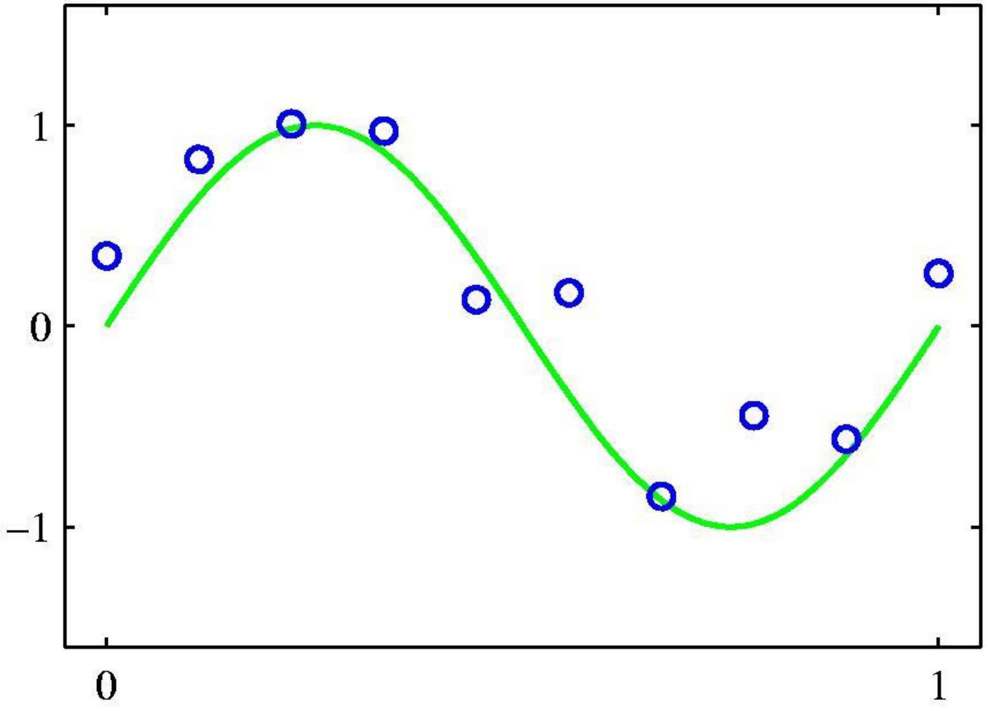
$$R^2 = 1 - \frac{\boldsymbol{e}^T\boldsymbol{e}}{(\boldsymbol{y} - \bar{y}\mathbf{1})^T(\boldsymbol{y} - \bar{y}\mathbf{1})}, R_a^2 = 1 - \frac{\boldsymbol{e}^T\boldsymbol{e}/(n-p-1)}{(\boldsymbol{y} - \bar{y}\mathbf{1})^T(\boldsymbol{y} - \bar{y}\mathbf{1})/(n-1)}$$

PARTIAL LEAST SQUARES REGRESSION

JIN YOUNG CHOI

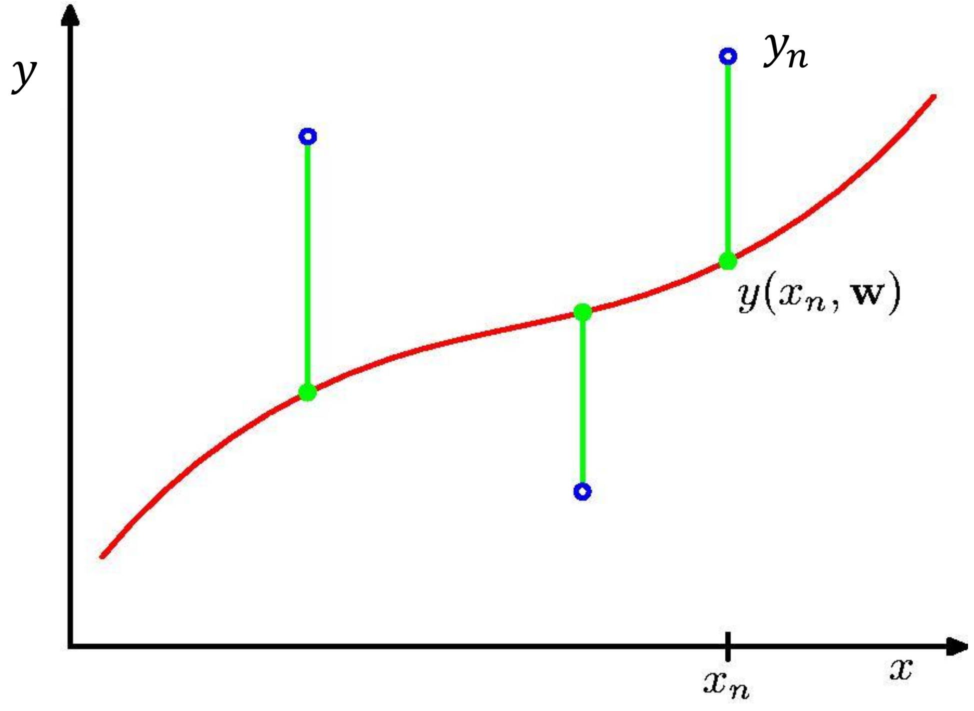
ECE, SEOUL NATIONAL UNIVERSITY

Overfitting and Underfitting

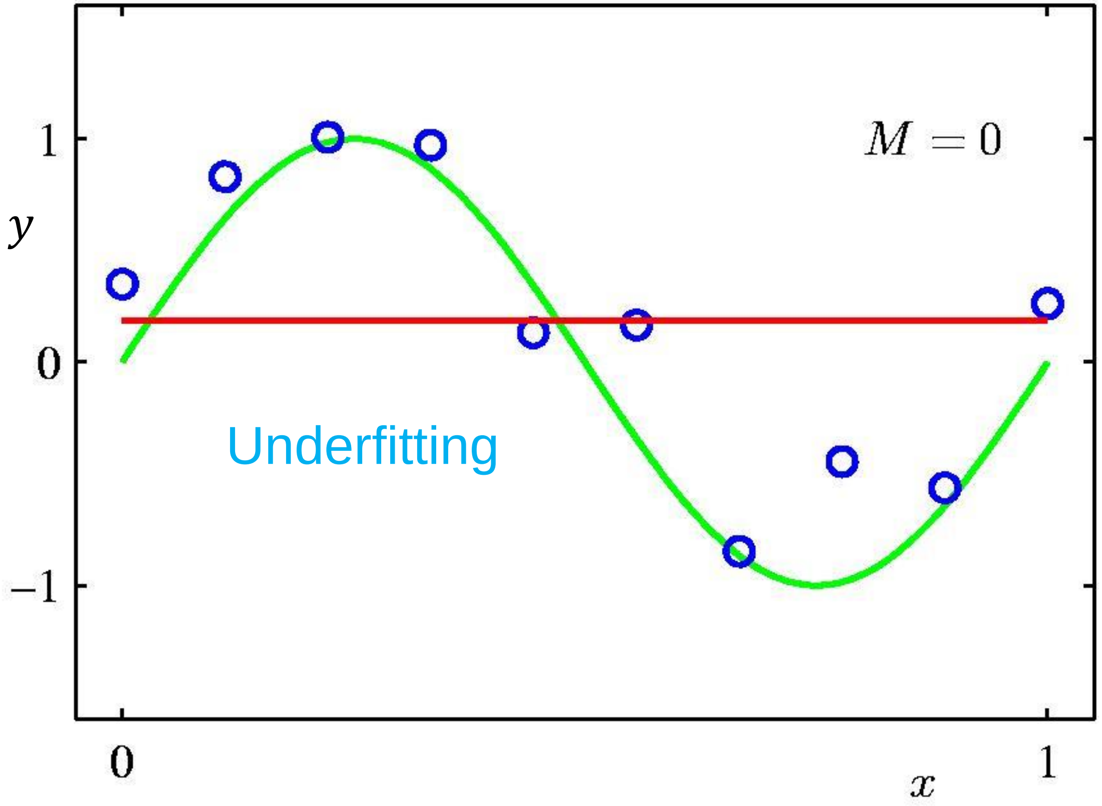


$$Y = \theta_0 + \theta_1 X + \theta_2 X^2 + \cdots + \theta_M X^M + \epsilon$$

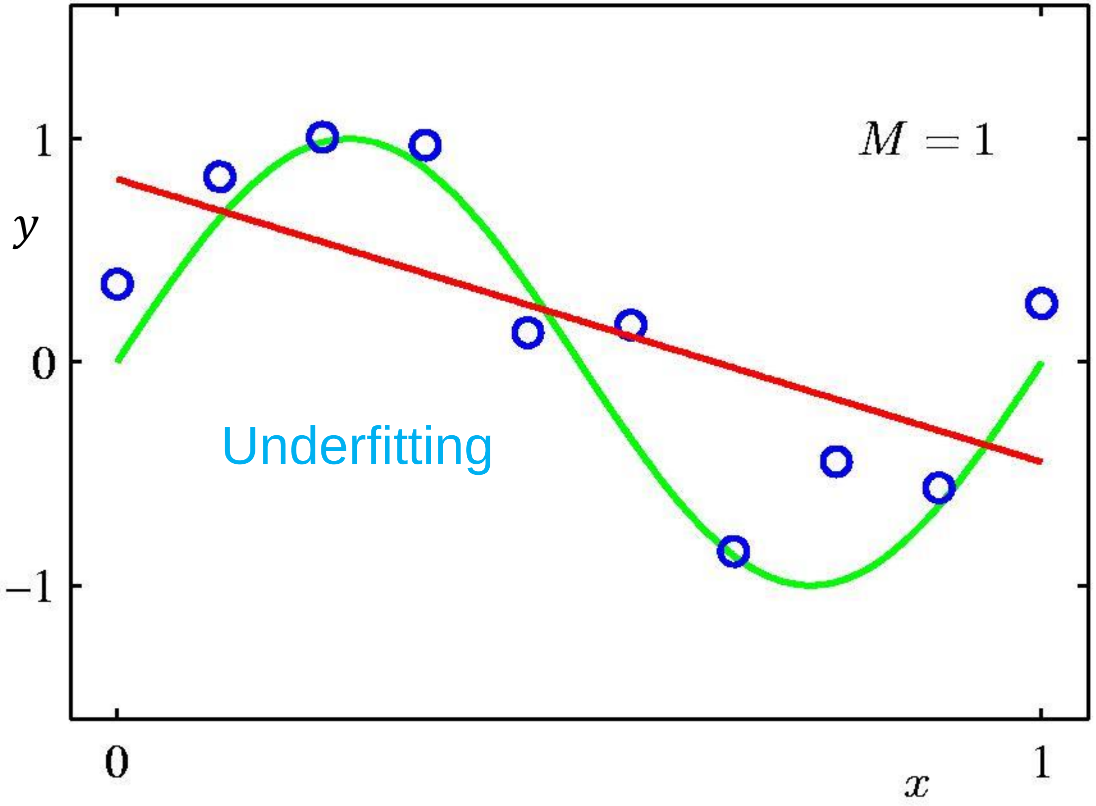
Sum-of-Squares Error Function



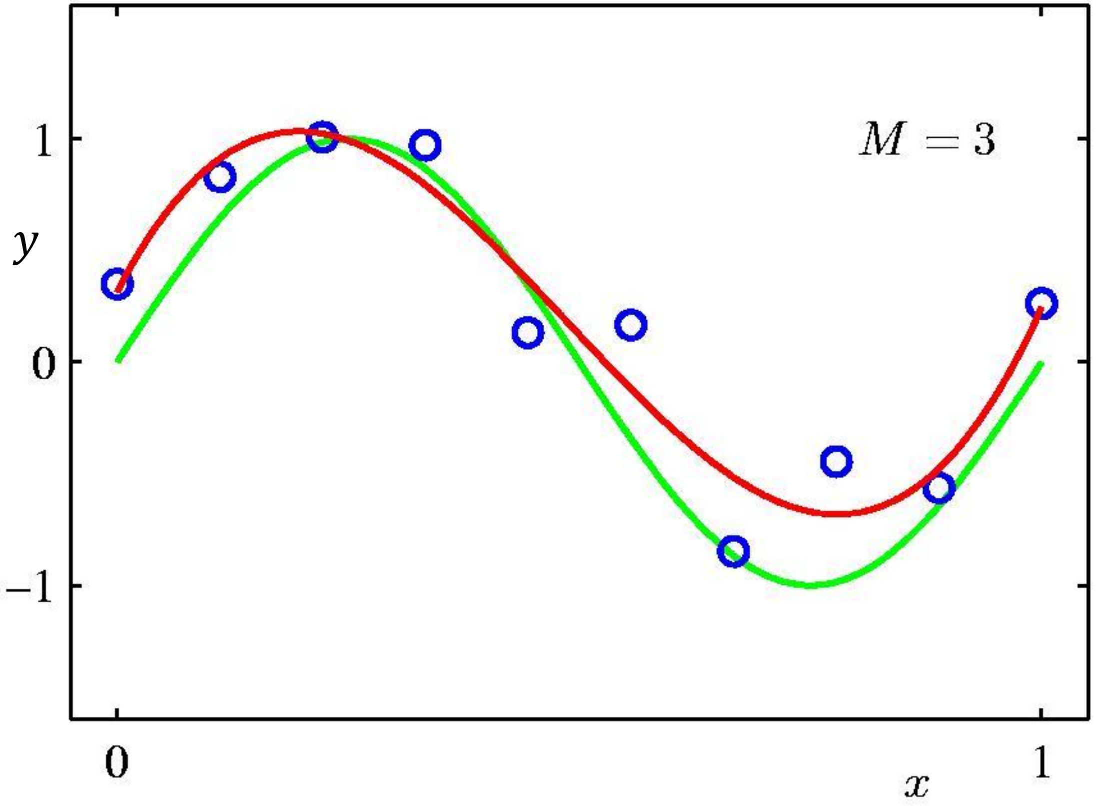
0^{th} Order Polynomial



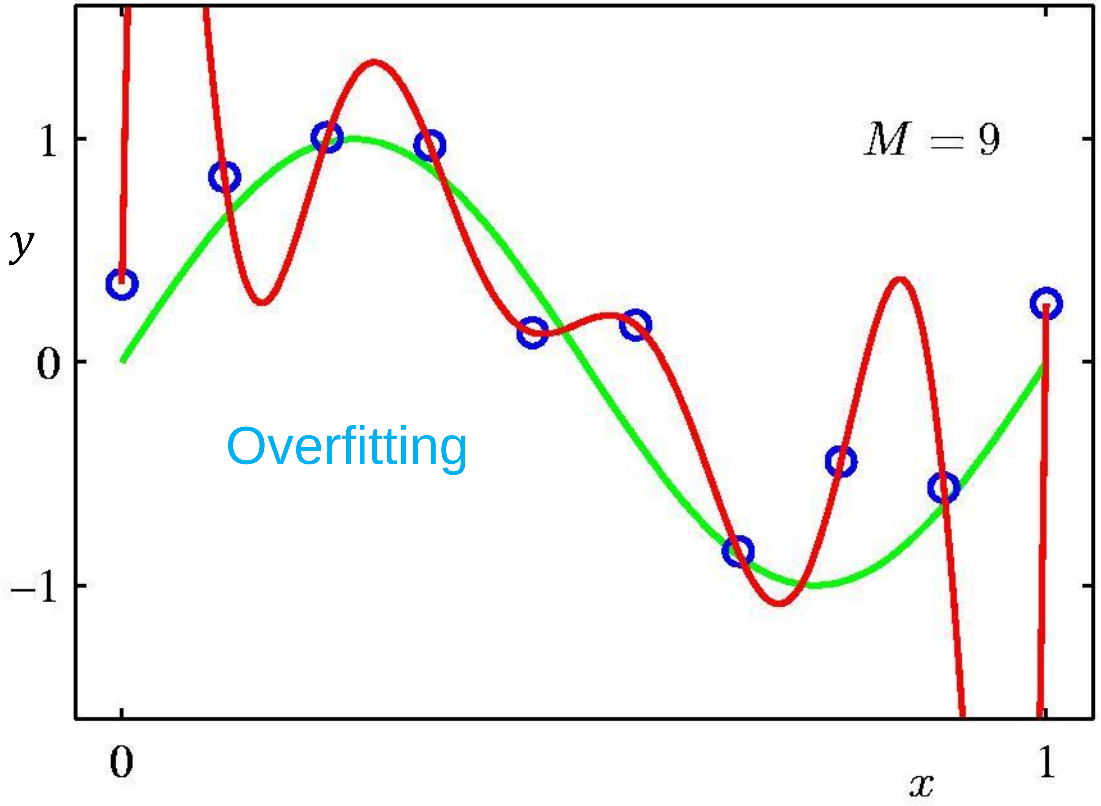
1^{st} Order Polynomial



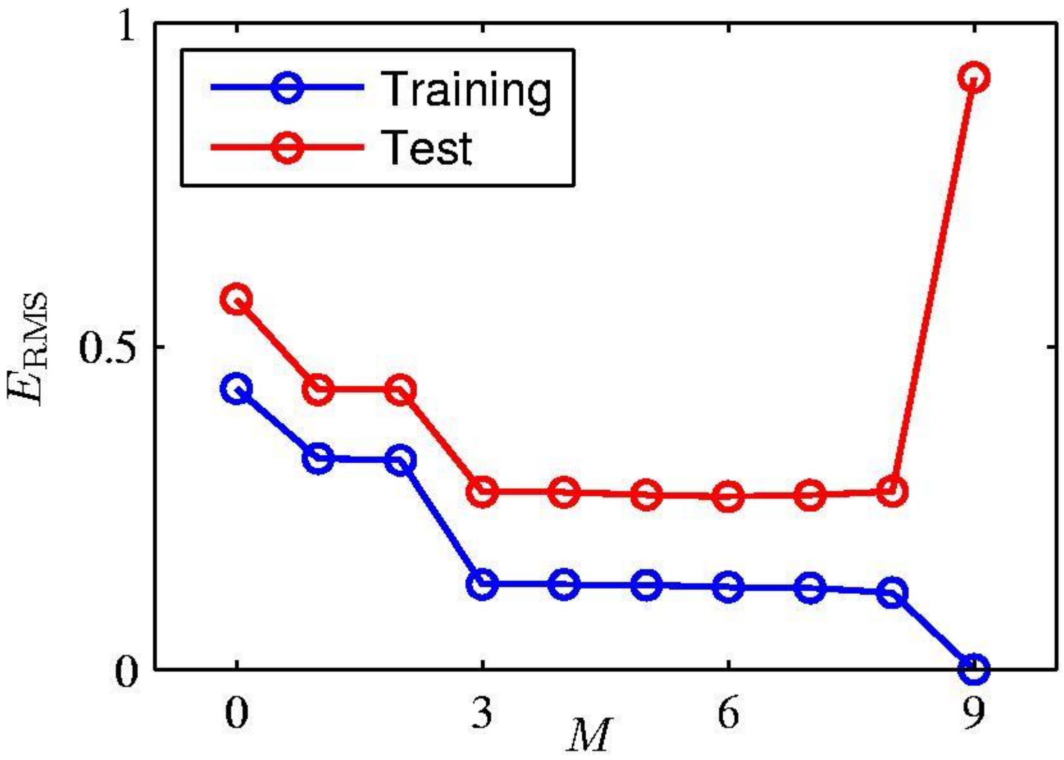
3^{rd} Order Polynomial



9^{th} Order Polynomial



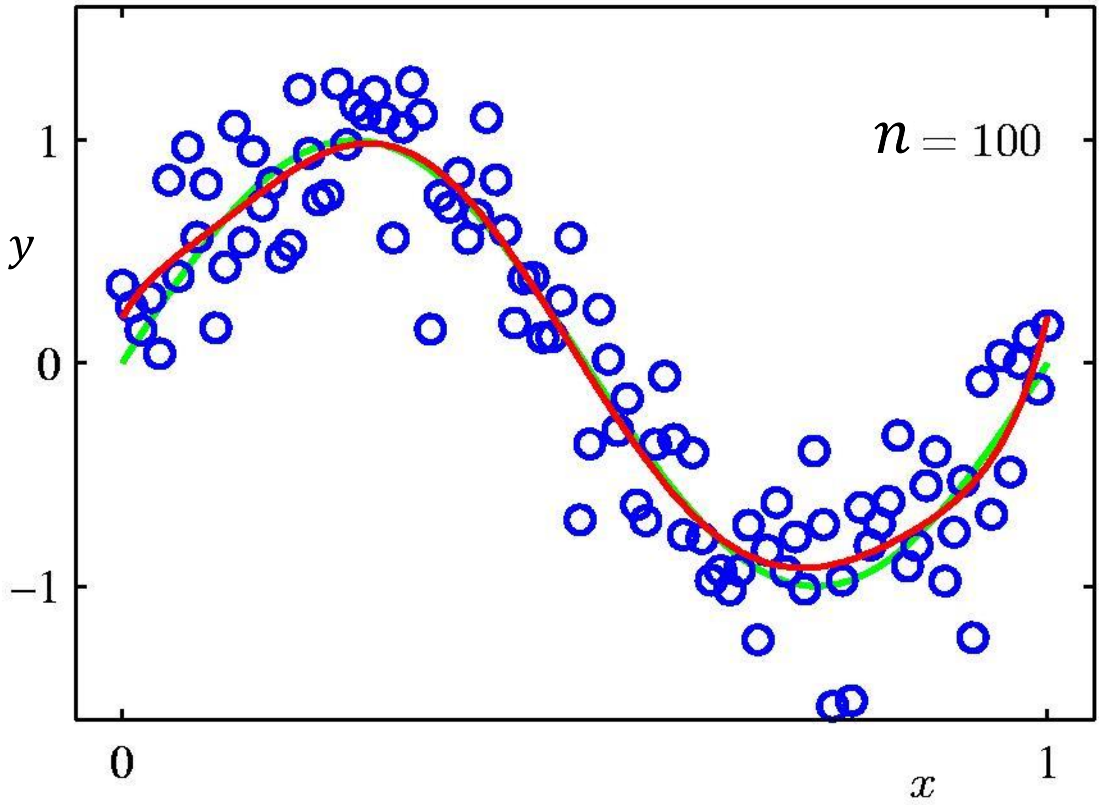
Over-fitting



Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{E(\theta^\star)/n}$

Data Set Size: $n = 100$

9th Order Polynomial



Bias and Variance in Parameter Estimation

- Mean Squared Error(MSE) decomposition

$$MSE(\hat{\theta}) = E\left((\hat{\theta}-\theta)^2\right)$$

$$= E\left(\left(\hat{\theta}-E(\hat{\theta})+E(\hat{\theta})-\theta\right)^2\right)$$

$$= E\left(\left(\hat{\theta}-E(\hat{\theta})\right)^2 + 2\left(\hat{\theta}-E(\hat{\theta})\right)\left(E(\hat{\theta})-\theta\right) + \left(E(\hat{\theta})-\theta\right)^2\right)$$

$$= E\left(\hat{\theta}-E(\hat{\theta})\right)^2 + 2\overbrace{E\left(\hat{\theta}-E(\hat{\theta})\right)}^{E(\hat{\theta})-E(\hat{\theta})=0}\left(E(\hat{\theta})-\theta\right) + \left(E(\hat{\theta})-\theta\right)^2$$

$$= E\left(\hat{\theta}-E(\hat{\theta})\right)^2 + \left(E(\hat{\theta})-\theta\right)^2$$

$$= Var(\hat{\theta}) + Bias(\hat{\theta},\theta)^2$$

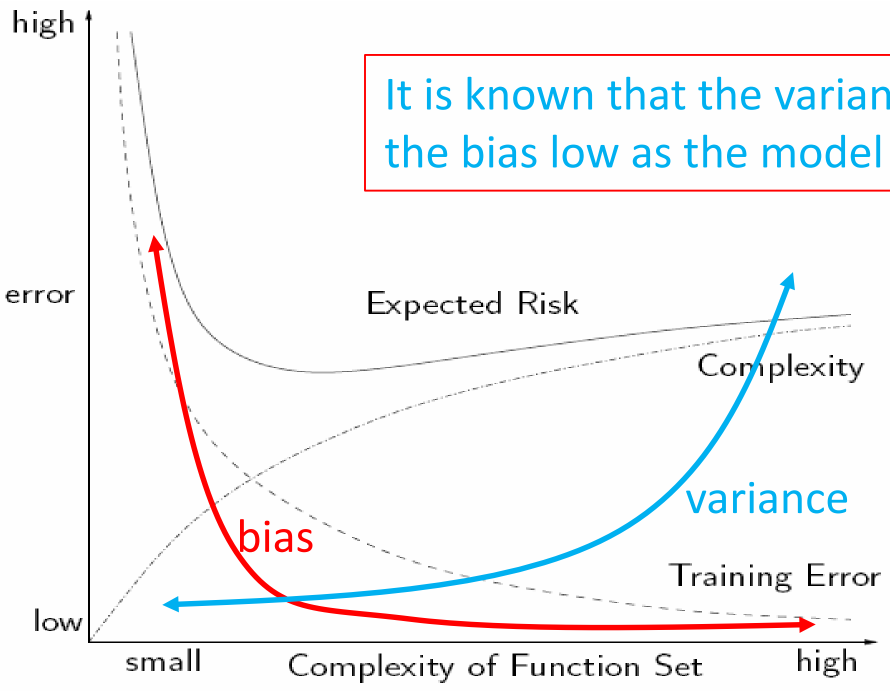
overfitting underfitting

Bias and Variance in Parameter Estimation

- Mean Squared Error(MSE) decomposition

$$MSE(\hat{\theta}) = E\left((\hat{\theta} - \theta)^2\right)$$

$$= Var(\hat{\theta}) + Bias(\hat{\theta}, \theta)^2$$



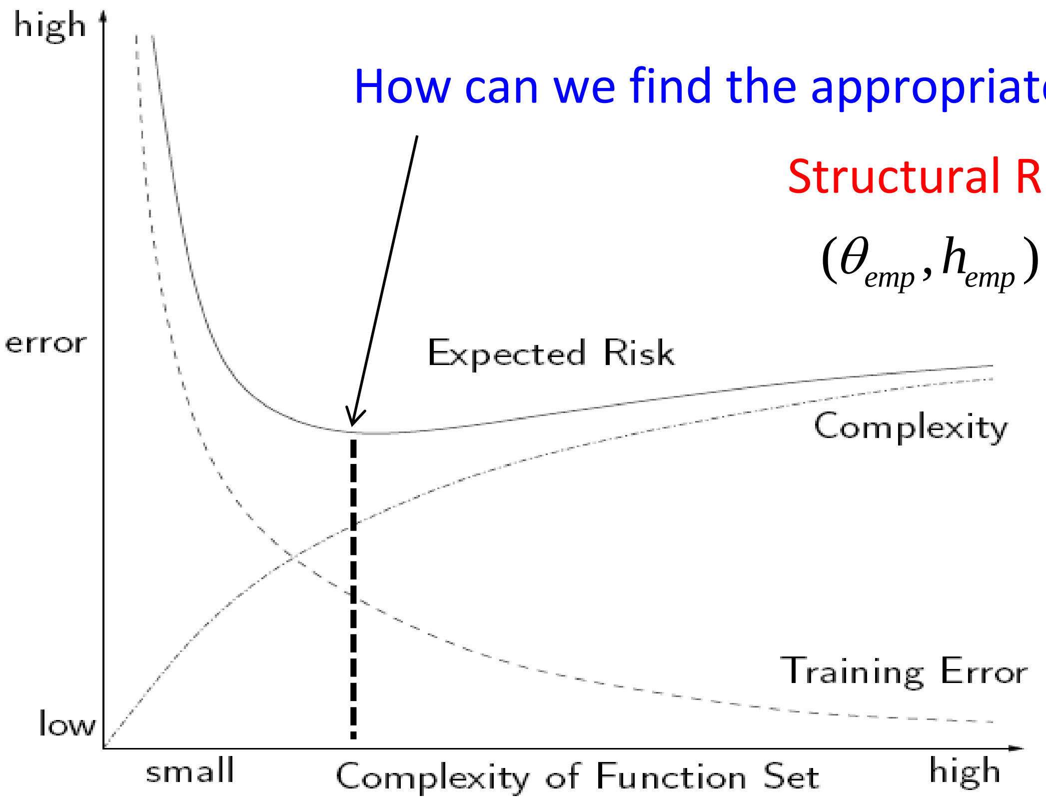
It is known that the variance becomes high and the bias low as the model complexity increases†.

† https://datacadamia.com/data_mining/bias_trade-off

Structural Risk Minimization

- For fixed training samples n

How can we find the appropriate complexity ?



Structural Risk Minimization

$$(\theta_{emp}, h_{emp}) = \arg\min_{\theta,h} R_{emp}(\theta, h)$$

$\Downarrow$

Pruning (F-test, PCA)

Regularization

- ridge regression
- Lasso regression

SVM

Partial Least Squares

- Matrix-vector form for General Regression (Revisit)

  Let $\theta = [\theta_0 \;\; \theta_1 \;\; \cdots \; \theta_M]^T, \quad \phi_i = [1 \;\; \phi_{i1} \cdots \phi_{iM}]^T$

  $\boldsymbol{y} = [y_1 \;\; y_2 \;\; \cdots \;\; y_n]^T, \quad \boldsymbol{\epsilon} = [\epsilon_1 \;\; \epsilon_2 \;\; \cdots \;\; \epsilon_n]^T$

  Then $y_i = \phi_i^T \theta + \epsilon_i, \;\; i = 1, \; \cdots, \; n.$

  $\boldsymbol{y} = \Phi\theta + \boldsymbol{\epsilon}, \quad \Phi = [\phi_1 \, \phi_2 \;\; \cdots \;\; \phi_n]^T$

- Matrix-vector form for Multivariate Regression with no-intercept

  $y_i = \mathbf{x}_i^T \theta + \epsilon_i, \;\; i = 1, \; \cdots, \; n$

  $\boldsymbol{y} = \mathbf{X}\theta + \boldsymbol{\epsilon}, \;\; \mathbf{X} = [\mathbf{x}_1 \, \mathbf{x}_2 \;\; \cdots \;\; \mathbf{x}_n]^T$ ⬅

  $\mathbf{x}_i = [x_{i1} \cdots x_{ip}]^T, \, \theta = [\theta_1 \;\; \cdots \; \theta_p]^T$

  $\mathbf{x}_i = \mathbf{x}_i^o - \mu, \;\; \mu = 1/n \sum_i \mathbf{x}_i^o$

- Goal: reduce the input & parameter dimension: $p > q$

  $\mathbf{x}_i = [x_{i1} \cdots x_{ip}]^T, \, \theta = [\theta_1 \;\; \cdots \; \theta_p]^T \;\; \longrightarrow \;\; \mathbf{x}_i = [x_{i1} \cdots x_{iq}]^T, \, \theta = [\theta_1 \;\; \cdots \; \theta_q]^T$

Principal Component Regression

$\mathbf{a}_k = E^T(\mathbf{x}_k - \mathbf{m})$

- Principal Component Analysis for $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_n]$

$$\mathbf{S} = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T = \mathbf{X}\mathbf{X}^T,\ \mathbf{S}\mathbf{u}_k = \lambda_k \mathbf{u}_k, \lambda_1 > \lambda_2 \cdots > \lambda_p$$

$$\mathbf{cov}(\mathbf{X}, \mathbf{X}) = \frac{1}{n-1}\mathbf{X}\,\mathbf{X}^T$$

- Reduced dim. vector ($q < p$ dim.)

$$\boldsymbol{z}_i = \bar{\mathbf{U}}^T \mathbf{x}_i,\ \ \bar{\mathbf{U}} = [\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_q]$$

$\mathrm{U}^T = \mathrm{U}^{-1}$ for orthonormal eigenvectors

$q \times 1$ $\quad$ $p \times q$

$$\mathbf{Z} = \bar{\mathbf{U}}^T \mathbf{X} \rightarrow \mathbf{Z}^T = \mathbf{X}^T \bar{\mathbf{U}},$$

$$\mathbf{Z} = [\mathbf{z}_1\ \mathbf{z}_2\ \cdots\ \mathbf{z}_n] = \bar{\mathbf{U}}^T\ [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_n]$$

- Applying LS algorithm to $\boldsymbol{y} = \mathbf{Z}^T \theta + \boldsymbol{\epsilon}$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|\boldsymbol{\epsilon}\|^2 = \|\boldsymbol{y} - \mathbf{Z}^T \theta\|^2 \rightarrow \hat{\theta} = (\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\boldsymbol{y} \rightarrow \hat{y} = \boldsymbol{z}^T \hat{\theta},\ \boldsymbol{z} = \bar{\mathbf{U}}^T \mathbf{x}$$

Partial Least Squares

- Nonlinear Iterative Partial Least Squares (NIPALS) algorithm

$$\mathbf{X}\mathbf{X}^T\mathbf{u} = \lambda\mathbf{u}$$

Let $\mathbf{t} = \mathbf{X}^T\mathbf{u}$

$$\mathbf{u} = \frac{1}{\lambda}\mathbf{X}\mathbf{t}$$

Since $\|\mathbf{u}\| := 1 = \frac{1}{\lambda}\|\mathbf{X}\mathbf{t}\|$

$$\lambda = \|\mathbf{X}\mathbf{t}\|$$

$$\bar{\mathbf{U}} = [\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_q],\ \boldsymbol{z}_i = \bar{\mathbf{U}}^T\mathbf{x}_i$$

$$\mathbf{Z} = \bar{\mathbf{U}}^T\mathbf{X} \to \mathbf{Z}^T = \mathbf{X}^T\bar{\mathbf{U}},\quad \mathbf{Z} = [\mathbf{z}_1\ \mathbf{z}_2\ \cdots\ \mathbf{z}_n]$$

$\mathbf{t} := \mathbf{x}_j$ for some j

Loop

$$\mathbf{u} = \mathbf{X}\mathbf{t}/\|\mathbf{X}\mathbf{t}\|$$

$$\mathbf{t} = \mathbf{X}^T\mathbf{u}$$

Until $\mathbf{t}$ stop changing

$$\mathbf{X}^T := \mathbf{X}^T - \mathbf{t}\mathbf{u}^T = \mathbf{X}(\mathrm{I} - \mathbf{u}\mathbf{u}^T)$$

Repeat the Loop up to a small $\|\mathbf{X}\mathbf{t}\|$

- Applying LS algorithm to $\boldsymbol{y} = \mathbf{Z}^T\theta + \boldsymbol{\epsilon}$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}}\|\boldsymbol{\epsilon}\|^2 = \|\boldsymbol{y} - \mathbf{Z}^T\theta\|^2 \to \hat{\theta} = (\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\boldsymbol{y} \to \hat{y} = \boldsymbol{z}^T\hat{\theta},\ \boldsymbol{z} = \bar{\mathbf{U}}^T\mathbf{x}$$

Ridge Regression for Regularization

- l_2 regularization term is added

$$\hat{\theta} = \underset{\theta}{\text{argmin}} \|\boldsymbol{y} - \Phi\theta\|^2 + \gamma\|\theta\|_2^2 \ (= S(\theta))$$

- solution:

$$\nabla_\theta \left((\boldsymbol{y} - \Phi\theta)^T (\boldsymbol{y} - \Phi\theta) + \gamma\theta^T\theta \right) = 0 \text{ at } \hat{\theta}$$

$$2\Phi^T \left(\boldsymbol{y} - \Phi\hat{\theta}\right) + 2\gamma\hat{\theta} = 0$$

$$\hat{\theta} = (\Phi^T\Phi - \gamma\mathbf{I})^{-1} \Phi^T \boldsymbol{y}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + G_k(y_{k+1} - \phi_{k+1}^T\hat{\theta}_k),$$

$$G_k \cong \frac{\lambda^{-1}P_k\phi_{k+1}}{1+\lambda^{-1}\phi_{k+1}^T P_k \phi_{k+1}}$$

$$P_{k+1} = \lambda^{-1}P_k - \lambda^{-1}G_k\phi_{k+1}^T P_k, \ P_0 = -\gamma\mathbf{I}$$

Lasso Regression for Regularization

- LASSO(Least Absolute Shrinkage Selector Operator)
- l_1 regularization term is added

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|\boldsymbol{y} - \Phi\theta\|^2 + \gamma\|\theta\|_1$$

- solution: l_1 norm is not differentiable→ constrained convex form

  by adding new optimization variables,

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|\boldsymbol{y} - \Phi\theta\|^2 + \gamma \mathbf{1}^T \boldsymbol{s}$$

$$\text{subject to} \quad |\theta_i| \leq s_i, \quad i = 1, \cdots, n$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|\boldsymbol{y} - \Phi\theta\|^2 + \gamma \mathbf{1}^T \boldsymbol{s}$$

$$\text{subject to} \quad -s_i \leq \theta_i \leq s_i, \quad i = 1, \cdots, n$$

Elastic Regression for Regularization

- Ridge + LASSO

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|\boldsymbol{y} - \Phi\theta\|^2 + \gamma_1 \|\theta\|_2^2 + \gamma_2 \|\theta\|_1$$

- solution: l_1 norm is not differentiable→ constrained convex form by adding new optimization variables,

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|\boldsymbol{y} - \Phi\theta\|^2 + \gamma_1 \|\theta\|_2^2 + \gamma_2 \mathbf{1}^T \boldsymbol{s}$$

$$\text{subject to} \quad |\theta_i| \le s_i, \quad i = 1, \cdots, n$$

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \|\boldsymbol{y} - \Phi\theta\|^2 + \gamma_1 \|\theta\|_2^2 + \gamma_2 \mathbf{1}^T \boldsymbol{s}$$

$$\text{subject to} \quad -s_i \le \theta_i \le s_i, \quad i = 1, \cdots, n$$

Interim Summary

- linear regression
  - simple linear regression
  - multiple linear regression
- nonlinear regression
  - logistic regression
  - high-order regression
  - basis-function regression
- matrix form for regression
  - recursive least squares
- partial least squares
  - over-fitting and underfitting
  - bias/variance
  - principle component regression
  - partial least squares algorithm
  - ridge regression
  - lasso, elastic regression
- Gaussian process regression
- Kalman filtering

Regression Analysis III

Jin Young Choi
Seoul National University

Outline

- linear regression
  - simple linear regression
  - multiple linear regression
- nonlinear regression
  - logistic regression
  - high-order regression
  - basis-function regression
- matrix form for regression
  - recursive least squares
- partial least squares
  - over-fitting and underfitting
  - bias/variance
  - principle component regression
  - partial least squares algorithm
  - ridge regression
  - lasso, elastic regression
- Gaussian process regression
- Kalman filtering

GAUSSIAN PROCESS REGRESSION

JIN YOUNG CHOI

ECE, SEOUL NATIONAL UNIVERSITY

https://arxiv.org/pdf/2009.10862.pdf
https://github.com/jwangjie/Gaussian-Processes-Regression-Tutorial
http://mlg.eng.cam.ac.uk/tutorials/06/es.pdf
https://www.sciencedirect.com/science/article/abs/pii/S0022249617302158
http://www.gaussianprocess.org/gpml/chapters/RW.pdf

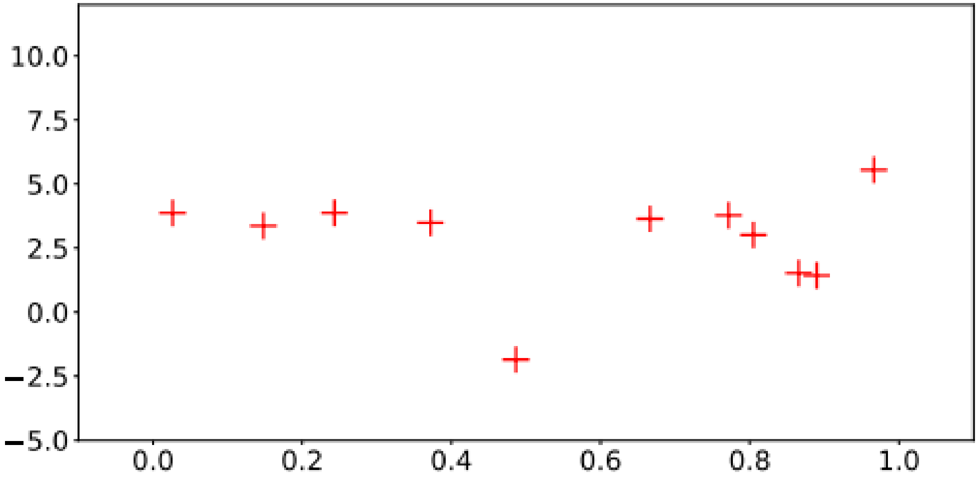
Gaussian Process Regression

- General regression model (single variable)

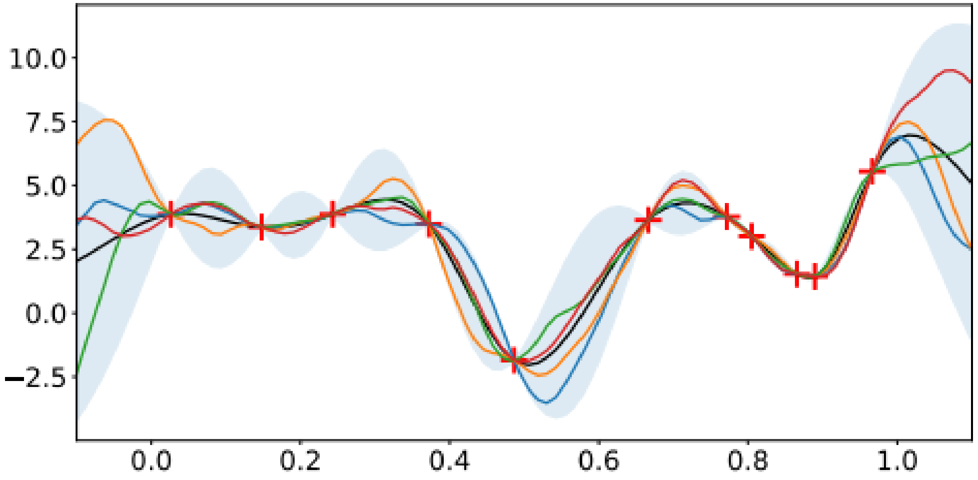
$$y = f(x) + \epsilon,$$

where $\epsilon \sim N(0, \sigma^2)$ and so x, y are Gaussian random variables.

- Goal : to estimate $f(x)$ with uncertainty from observation data $D = \{(x_i, y_i) | i = 1, \cdots, n\}$
- x_i, y_i are treated as Gaussian random variables.



(a) Data point observations



(b) Five possible regression functions by GPR

Gaussian Process Regression

- General regression model (single variable)

$$y = f(x) + \epsilon,$$

where $\epsilon \sim N(0, \sigma^2)$ and so x, y are Gaussian random variables.

- Define

$$\mathbf{x}^{\mathrm{T}} = [x_1 \quad \cdots \quad x_n], \qquad \mathbf{y}^{\mathrm{T}} = [y_1 \quad \cdots \quad y_n], \qquad \mathbf{f} := \mathbf{f}(\mathbf{x}) = [f(x_1) \quad \cdots \quad f(x_n)].$$

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} - \mu\right)^T \Sigma^{-1}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} - \mu\right)\right] := \boldsymbol{\mathcal{N}}(\mu, \Sigma)$$

- conditional probability

$$f_{X|Y}(x|y) = \frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{\det\Sigma_{X|y}}} \exp\left(-\frac{1}{2}(x - \mu_{X|y})^t \Sigma_{X|y}^{-1}(x - \mu_{X|y})\right),$$

where

$\mu_{X|y} = A(y - \mu_Y) + \mu_X$ and

$\Sigma_{X|y} = \Sigma_X - AC_{YX}$, where $A\Sigma_Y = \Sigma_{XY}$.

Gaussian Process Regression

- General regression model (single variable)

$$y = f(x) + \epsilon,$$

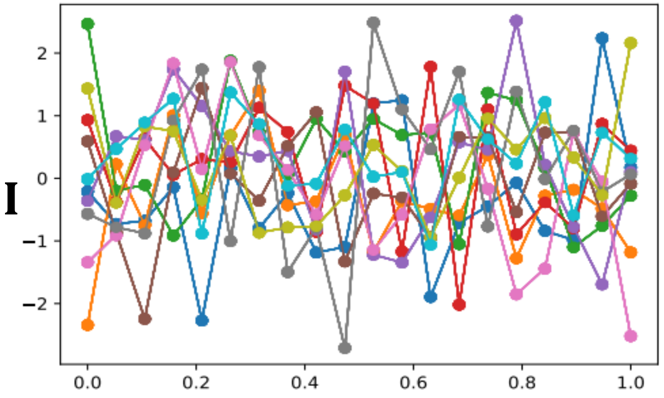
where $\epsilon \sim N(0, \sigma^2)$ and so x, y are Gaussian random variables.

- Define

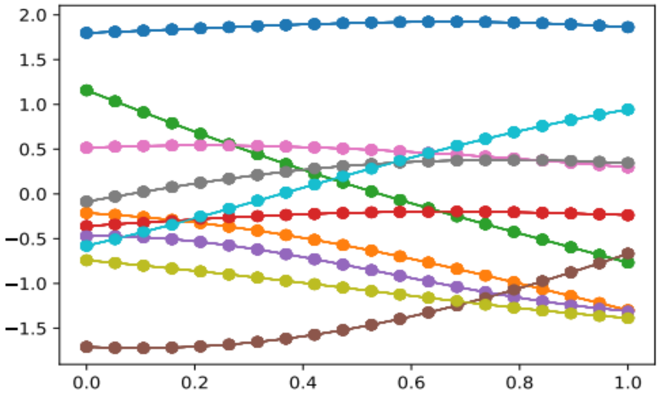
$$\mathbf{x} = [x_1 \quad \cdots \quad x_n], \qquad \mathbf{y} = [y_1 \quad \cdots \quad y_n], \qquad \mathbf{f} := \mathbf{f}(\mathbf{x}) = [f(x_1) \quad \cdots \quad f(x_n)].$$

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{(2\pi)^{d/2}|\Sigma_{\mathbf{x}|\mathbf{y}}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_{\mathbf{x}|\mathbf{y}})^T \Sigma_{\mathbf{x}|\mathbf{y}}{}^{-1}(\mathbf{x} - \mu_{\mathbf{x}|\mathbf{y}})\right] := \mathcal{N}(\mu_{\mathbf{x}|\mathbf{y}}, \Sigma_{\mathbf{x}|\mathbf{y}})$$

$$\Sigma_{\mathbf{x}|\mathbf{y}} = \sigma^2\mathbf{I}$$



Independent Gaussian



Jointly Gaussian

$$x = f^{-1}(y)$$

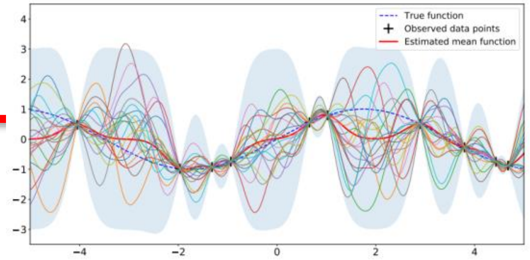
$$\Sigma_{\mathbf{x}|\mathbf{y}} \neq \sigma^2\mathbf{I}$$

$$\Sigma_{\mathbf{x}|\mathbf{y}_{ij}} = cov(x_i, x_j)$$

$$:= exp\left(-\frac{(x_i - x_j)^2}{2}\right)$$

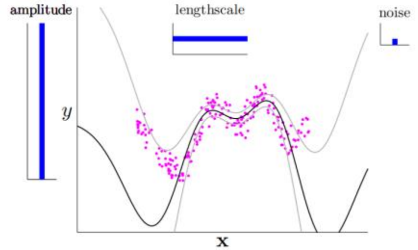
a RBF kernel

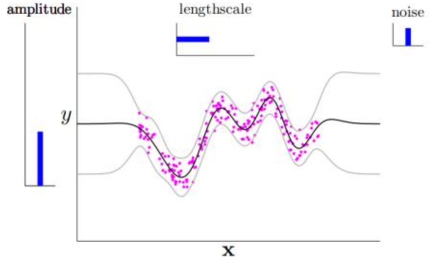
Gaussian Process Regression



- Gaussian Processes ($\mathcal{GP}$) for multivariate regression
$$y = f(\mathbf{x}) + \epsilon.$$

- define $\mu_f(\mathbf{x}) := \mathbb{E}(f(\mathbf{x}))$, then we assume $f(\mathbf{x})$ is distributed as a Gaussian process
$$f(x) \sim \mathcal{GP}\left(\mu_f(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$$
where $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}\left[\left(f(\mathbf{x}) - \mu_f(\mathbf{x})\right)\left(f(\mathbf{x}') - \mu_f(\mathbf{x}')\right)\right]$ called the kernel of $\mathcal{GP}$.





- The kernel is based on assumptions such as smoothness, that is, similar $\mathbf{x}, \mathbf{x}'$ yields similar $f(\mathbf{x})$ and $f(\mathbf{x}')$. Thus a popular kernel is
$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \mathbf{exp}\left(-\frac{1}{2\lambda}(\mathbf{x} - \mathbf{x}')^T(\mathbf{x} - \mathbf{x}')\right),$$
where hyperparameters λ and σ_f^2 represents the length-scale and signal (f) variance to control relation between $\mathbf{x}$ and $f(\mathbf{x})$.

Gaussian Process Regression

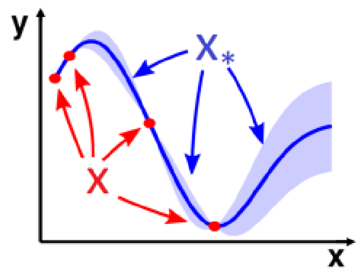
$$k(\mathbf{x},\mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\lambda}(\mathbf{x}-\mathbf{x}')^T(\mathbf{x}-\mathbf{x}')\right)$$

Modeling of prior sampling function of $\mathcal{GP}$

- Denote $\mathbf{X} = [\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_n]$, $\mathbf{y}^T = [y_1 \quad \cdots \quad y_n]$, $\mathbf{f}^T := [f(\mathbf{x}_1) \quad \cdots \quad f(\mathbf{x}_n)]$.

  Let $\mathbf{X}_*$ be a matrix containing a new input points $\mathbf{x}_i^*, i = 1, \cdots, n$. Then define the kernel matrix as

$$\mathbf{K}(\mathbf{X}_*, \mathbf{X}_*) = \begin{bmatrix} k(\mathbf{x}_1^*, \mathbf{x}_1^*) & k(\mathbf{x}_1^*, \mathbf{x}_2^*) & \cdots & k(\mathbf{x}_1^*, \mathbf{x}_n^*) \\ k(\mathbf{x}_2^*, \mathbf{x}_1^*) & k(\mathbf{x}_2^*, \mathbf{x}_2^*) & \cdots & k(\mathbf{x}_2^*, \mathbf{x}_n^*) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n^*, \mathbf{x}_1^*) & k(\mathbf{x}_n^*, \mathbf{x}_2^*) & \cdots & k(\mathbf{x}_n^*, \mathbf{x}_n^*) \end{bmatrix}$$



- Choosing the prior mean function $\mu_f(\mathbf{x}) = 0$, we can sample values of f at inputs $\mathbf{X}_*$ from $\mathcal{GP}$ as

$$\mathbf{f}_* \sim \mathcal{N}\big(0, \mathbf{K}(\mathbf{X}_*, \mathbf{X}_*)\big)$$

  which is the prior distribution model without observation data $D = \{(x_i, y_i) | i = 1, \cdots, n\}$.

Gaussian Process Regression

Posterior predictions from a $\mathcal{GP}$

- Observations are $D = \{(\mathbf{x}_i, y_i) | i = 1, \cdots, n\} = \{\mathbf{X}, \mathbf{y}\}$, $\mathbf{X} = [\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_n]$, $\mathbf{y}^T = [y_1 \quad \cdots \quad y_n]$.
- The predictions for new inputs $\mathbf{X}_*$ by drawing $\mathbf{f}_*$ from the posterior distribution $p(f \,|D)$.

  A joint Gaussian distribution of $\mathbf{y}$ and $\mathbf{f}_*$ Let $\mathbf{X}_*$ follows

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2 \mathbf{I} & \mathbf{K}(\mathbf{X},\mathbf{X}_*) \\ \mathbf{K}(\mathbf{X}_*,\mathbf{X}) & \mathbf{K}(\mathbf{X}_*,\mathbf{X}_*) \end{bmatrix}\right),$$

  where σ_ϵ^2 is the assumed noise level of the observations.

- The conditional distribution $p(\mathbf{f}_* | \mathbf{X}, \mathbf{y}, \mathbf{X}_*)$ can be derived to a multivariate normal distribution with mean

$$\mathbf{K}(\mathbf{X}_*,\mathbf{X})[\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}]^{-1}\mathbf{y}$$

  and variance

$$\mathbf{K}(\mathbf{X}_*,\mathbf{X}_*) - \mathbf{K}(\mathbf{X}_*,\mathbf{X})[\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}]^{-1}\mathbf{K}(\mathbf{X},\mathbf{X}_*)$$

Gaussian Process Regression

Posterior predictions from a $\mathcal{GP}$

- The mean function of the $\mathcal{GP}$ can be given as

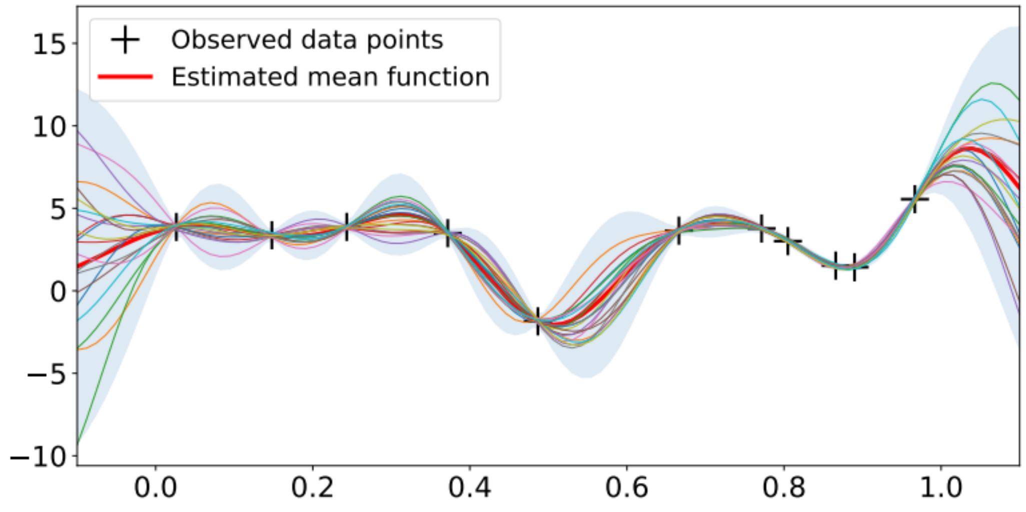
$$\mu_f(\mathbf{x}) = \mathbf{K}(\mathbf{x},\mathbf{X})[\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2\mathbf{I}]^{-1}\mathbf{y}$$

and covariance function as

$$cov(\mathbf{x},\mathbf{x}') = k(\mathbf{x},\mathbf{x}') - \mathbf{K}(\mathbf{x},\mathbf{X})[\mathbf{K}(\mathbf{X},\mathbf{X}) + \sigma_\epsilon^2\mathbf{I}]^{-1}\mathbf{K}(\mathbf{X},\mathbf{x}')$$

$$\mathbf{K}(\mathbf{x},\mathbf{X}) = [k(\mathbf{x},\mathbf{x}_1) \quad k(\mathbf{x},\mathbf{x}_2) \quad \cdots \quad k(\mathbf{x},\mathbf{x}_n)]$$

$$\mathbf{K}(\mathbf{X},\mathbf{x}) = \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}) \\ k(\mathbf{x}_2,\mathbf{x}) \\ \\ k(\mathbf{x}_n,\mathbf{x}) \end{bmatrix}$$

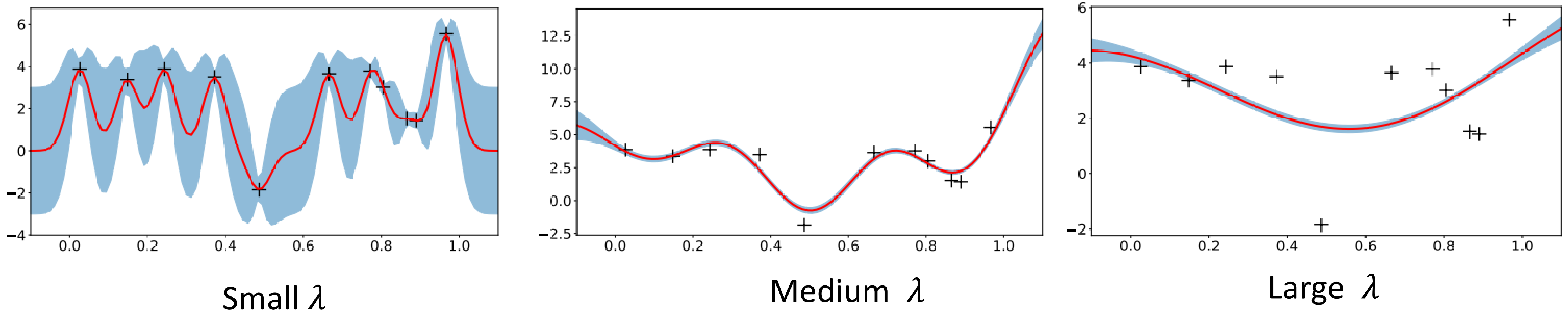


Gaussian Process Regression

- The effect of the hyperparameters λ and σ_f^2 of the kernel

$$k(\mathbf{x},\mathbf{x}') = \sigma_f^2 \mathbf{exp}\left(-\frac{1}{2\lambda}(\mathbf{x}-\mathbf{x}')^T(\mathbf{x}-\mathbf{x}')\right) \approx \mathbb{E}\left[\left(f(\mathbf{x})-\mu_f(\mathbf{x})\right)\left(f(\mathbf{x}')-\mu_f(\mathbf{x}')\right)\right],$$

λ : length-scale, σ_f^2 : signal (f) variance to control relation between $\mathbf{x}$ and $f(\mathbf{x})$.



Gaussian Process Regression

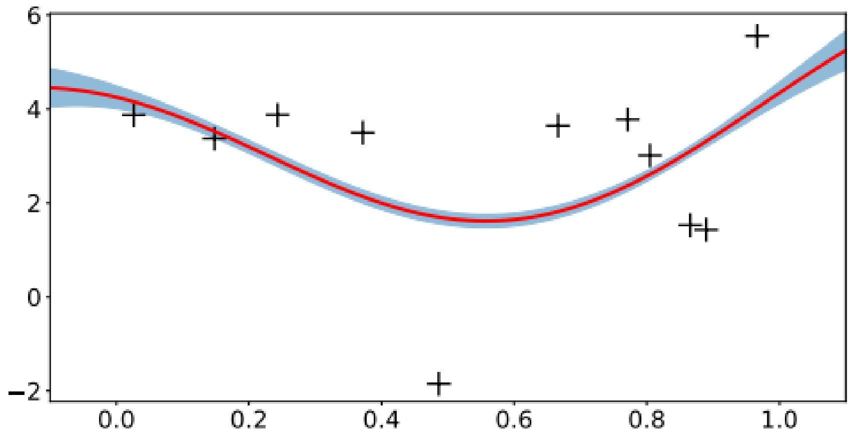
$$\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_*\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K}(\mathbf{X},\mathbf{X})+\sigma_\epsilon^2\mathbf{I} & \mathbf{K}(\mathbf{X},\mathbf{X}_*)\\ \mathbf{K}(\mathbf{X}_*,\mathbf{X}) & \mathbf{K}(\mathbf{X}_*,\mathbf{X}_*)\end{bmatrix}\right)$$

$$k(\mathbf{x},\mathbf{x}') = \sigma_f^2 \mathbf{exp}\left(-\frac{1}{2\lambda}(\mathbf{x}-\mathbf{x}')^T(\mathbf{x}-\mathbf{x}')\right)$$

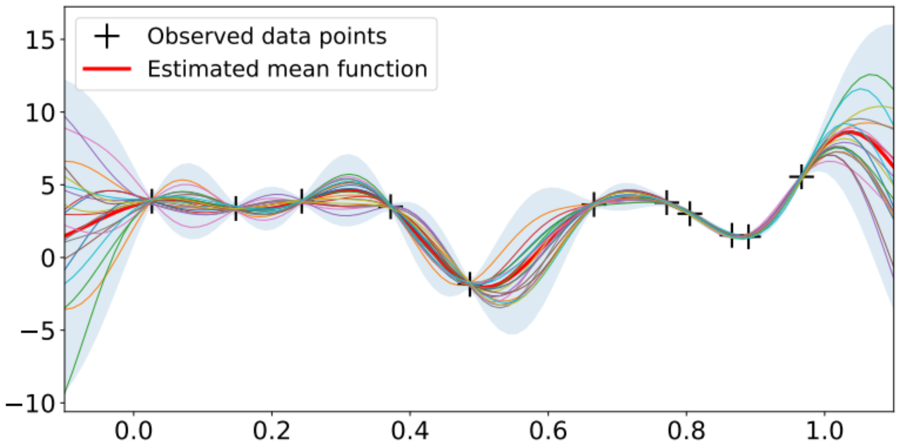
- The optimized hyperparameters λ and σ_f^2

$$\lambda, \sigma_f^2 = \max_{\lambda,\, \sigma_f^2} \log p(\mathbf{y}|\mathbf{X})$$

$$\log p(\mathbf{y}|\mathbf{X}) = -\frac{1}{2}\mathbf{y}^T[\mathbf{K}(\mathbf{X},\mathbf{X})+\sigma_\epsilon^2\mathbf{I}]^{-1}\mathbf{y} - \frac{1}{2}\log\det[\mathbf{K}(\mathbf{X},\mathbf{X})+\sigma_\epsilon^2\mathbf{I}] - \frac{n}{2}\log 2\pi$$



$\sigma_f = 0.0067$
$\lambda = 0.0967$



KALMAN FILTER

JIN YOUNG CHOI

ECE, SEOUL NATIONAL UNIVERSITY

https://www.cse.sc.edu/~terejanu/files/tutorialKF.pdf

https://aircconline.com/ijcses/V8N1/8117ijcses01.pdf

Outline

- Kalman Filter
  - Stochastic time-variant linear system
  - Derivation of Kalman filter
  - Kalman Filtering example
  - Extended Kalman filter

Dynamic process

- Stochastic time-variant linear system

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$
$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$$

control input $\mathbf{u}_k$

initial state $\mathbf{x}_0$

$$\mu_0 = E[\mathbf{x}_0]$$

$$\mathbf{P}_0 = E[(\mathbf{x}_0 - \mu_0)(\mathbf{x}_0 - \mu_0)^T]$$

- Model uncertainty, measurement noise

$$E[\mathbf{w}_k] = 0 \qquad E[\mathbf{w}_k\mathbf{w}_k^T] = \mathbf{Q}_k \qquad E[\mathbf{w}_k\mathbf{w}_j^T] = 0 \text{ for } k \neq j \qquad E[\mathbf{w}_k\mathbf{x}_0^T] = 0 \text{ for all } k$$
$$E[\mathbf{v}_k] = 0 \qquad E[\mathbf{v}_k\mathbf{v}_k^T] = \mathbf{R}_k \qquad E[\mathbf{v}_k\mathbf{v}_j^T] = 0 \text{ for } k \neq j \qquad E[\mathbf{v}_k\mathbf{x}_0^T] = 0 \text{ for all } k$$

$$E[\mathbf{w}_k\mathbf{v}_j^T] = 0 \text{ for all } k \text{ and } j$$

Dynamic process

- Dimension and description of variables:

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$
$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$$

$\mathbf{x}_k$	$n \times 1$	– State vector
$\mathbf{u}_k$	$l \times 1$	– Input/control vector
$\mathbf{w}_k$	$n \times 1$	– Process noise vector
$\mathbf{z}_k$	$m \times 1$	– Observation vector
$\mathbf{v}_k$	$m \times 1$	– Measurement noise vector
$\mathbf{A}_k$	$n \times n$	– State transition matrix
$\mathbf{B}_k$	$n \times l$	– Input/control matrix
$\mathbf{H}_k$	$m \times n$	– Observation matrix
$\mathbf{Q}_k$	$n \times n$	– Process noise covariance matrix
$\mathbf{R}_k$	$m \times m$	– Measurement noise covariance matrix

- Problem:

  How to optimally estimate state $\mathbf{x}_k$ from observations $\{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\}$ and $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\}$?

Kalman Filter

- Initial optimal estimate and error covariance

$$\begin{aligned} \mathbf{x}_0^a &= \mu_0 = E[\mathbf{x}_0] \\ \mathbf{P}_0 &= E[(\mathbf{x}_0 - \mathbf{x}_0^a)(\mathbf{x}_0 - \mathbf{x}_0^a)^T] \end{aligned}$$

- Optimal estimate

$$\mathbf{x}_{k-1}^a \equiv E[\mathbf{x}_{k-1}|\mathbf{Z}_{k-1}]$$

$$\mathbf{P}_{k-1} \equiv E[(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^a)(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^a)^T]$$

- Prediction

$$\begin{aligned} \mathbf{x}_k^f &\equiv E[\mathbf{x}_k|\mathbf{Z}_{k-1}] \\ &= E[\mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}|\mathbf{Z}_{k\text{-}1}] \\ &= \mathbf{A}_{k-1}\mathbf{x}_{k-1}^a + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \end{aligned}$$

Kalman Filter

- Prediction error

$$\begin{aligned} \mathbf{e}_k^f &\equiv \mathbf{x}_k - \mathbf{x}_k^f \\ &= \mathbf{A}_{k-1}(\mathbf{x}_{k-1} - \mathbf{x}_{k-1}^a) + \mathbf{w}_{k-1} \\ &= \mathbf{A}_{k-1}\mathbf{e}_{k-1} + \mathbf{w}_{k-1} \end{aligned}$$

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \end{aligned}$$

$$\mathbf{x}_k^f = \mathbf{A}_{k-1}\mathbf{x}_{k-1}^a + \mathbf{B}_{k-1}\mathbf{u}_{k-1}$$

- Prediction error covariance

$$\begin{aligned} \mathbf{P}_k^f &\equiv E[\mathbf{e}_k^f(\mathbf{e}_k^f)^T] \\ &= E[(\mathbf{A}_{k-1}\mathbf{e}_{k-1} + \mathbf{w}_{k-1})(\mathbf{A}_{k-1}\mathbf{e}_{k-1} + \mathbf{w}_{k-1})^T] \\ &= \mathbf{A}_{k-1}E[\mathbf{e}_{k-1}(\mathbf{e}_{k-1})^T]\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \\ &= \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \end{aligned}$$

Kalman Filter

$$\begin{aligned}\mathbf{x}_k &= \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k\end{aligned}$$

$$\mathbf{P}_k^f = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}$$

- Update of optimal estimate

$$\begin{aligned}\mathbf{x}_k^a &\equiv E[\mathbf{x}_k|Z_k] \\ &= E[\mathbf{x}_k|Z_{k-1}] + E[\mathbf{x}_k|\mathbf{z}_k]\end{aligned}$$

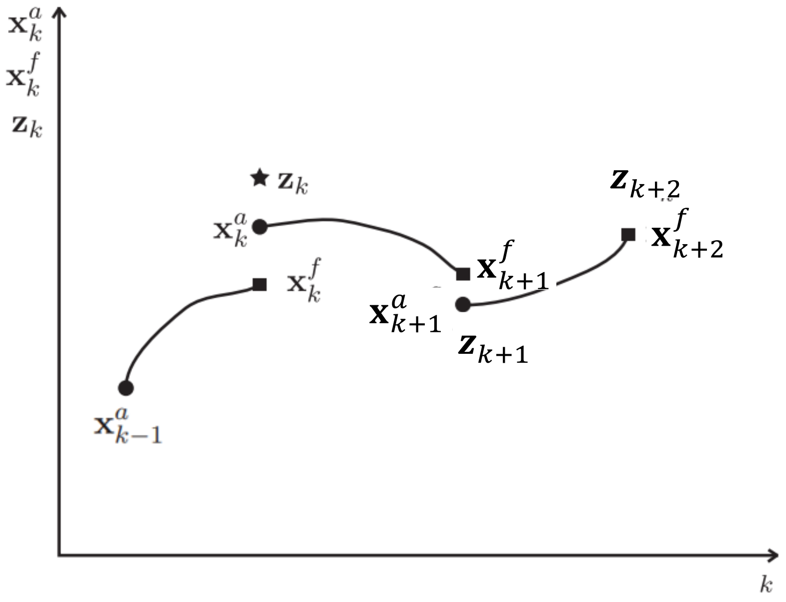
$$(= \mathbf{x}_k^f)$$

- Innovation (new information)

$$\mathbf{z}_k - \mathbf{H}_k x_k^f \quad \Rightarrow \quad E[\mathbf{x}_k|\mathbf{z}_k] = \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_k^f)$$ $\mathbf{K}_k$ is Kalman Gain

- Optimal estimate in k step

$$\begin{aligned}\mathbf{x}_k^a &= \mathbf{x}_k^f + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_k^f) \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{x}_k^f + \mathbf{K}_k\mathbf{z}_k\end{aligned}$$



Kalman Filter

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \end{aligned}$$

- Optimal estimate in k step

$$\begin{aligned} \mathbf{x}_k^a &= \mathbf{x}_k^f + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_k^f) \\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{x}_k^f + \mathbf{K}_k\mathbf{z}_k \end{aligned}$$

- Find Kalman gain to minimize prediction error covariance $tr(\mathbf{P}_k)$ $\mathbf{P}_k = \mathrm{E}[e_k\ e_k^T]$, $e_k = \mathrm{x}_k - \mathrm{x}_k^a$

$$\frac{\partial tr(\mathbf{P}_k)}{\partial \mathbf{K}_k} = 0 \rightarrow \mathbf{K}_k = \mathbf{P}_k^f\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^f\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$

- Kalman Filter

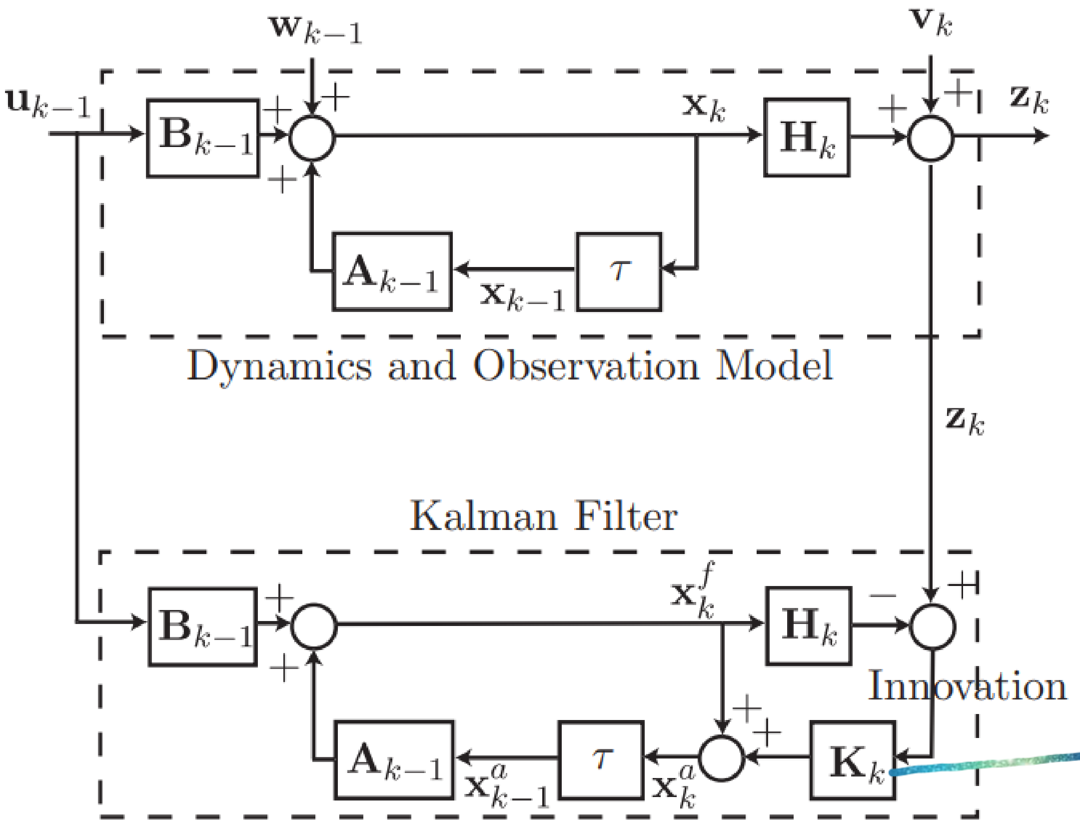
$$\begin{aligned} \mathbf{x}_k^f &= \mathbf{A}_{k-1}\mathbf{x}_{k-1}^a + \mathbf{B}_{k-1}\mathbf{u}_{k-1} \\ \mathbf{P}_k^f &= \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1} \end{aligned}$$

$$\begin{aligned} \mathbf{x}_k^a &= \mathbf{x}_k^f + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_k^f) \\ \mathbf{K}_k &= \mathbf{P}_k^f\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^f\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^f \end{aligned}$$

Kalman Filter

- The block diagram for Kalman Filter

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$
$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$$



$$\mathbf{x}_k^f = \mathbf{A}_{k-1}\mathbf{x}_{k-1}^a + \mathbf{B}_{k-1}\mathbf{u}_{k-1}$$
$$\mathbf{x}_k^a = \mathbf{x}_k^f + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}_k^f)$$

$$\mathbf{P}_k^f = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}_{k-1}^T + \mathbf{Q}_{k-1}$$
$$\mathbf{K}_k = \mathbf{P}_k^f\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^f\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$$
$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^f$$

Linear Kalman Filtering Example

- Particle dynamics with the object acceleration of gravity

$$\ddot{h}(t) = -g$$

where h is the height of the object in meters, g is gravity ($g = 9.80665\ m/s^2$).

- Discretization

$$\ddot{h}(t) = \frac{\dot{h}(t) - \dot{h}(t-\Delta t)}{\Delta t} = -g$$

$$\dot{h}(t) = \dot{h}(t-\Delta t) - g\Delta t$$

$$h(t) = h(t-\Delta t) + \dot{h}(t-\Delta t) - \frac{1}{2} g (\Delta t)^2$$

Letting $t = k\Delta t$,

$$h(t) = h(k\Delta t) = h_k$$

$$h(t-\Delta t) = h(k\Delta t - \Delta t) = h(\Delta t(k-1)) = h_{k-1}$$

$$\Rightarrow \quad \mathbf{x}_k = \begin{bmatrix} h_k \\ \dot{h}_k \end{bmatrix}$$

$$\Uparrow$$

$$\mathbf{x}_k = \begin{bmatrix} h_{k-1} + \dot{h}_{k-1}\Delta t - \frac{1}{2} g (\Delta t)^2 \\ \dot{h}_{k-1} - g\Delta t \end{bmatrix}$$

$$\Uparrow$$

$$\mathbf{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} -\frac{1}{2} (\Delta t)^2 \\ -\Delta t \end{bmatrix} g$$

Linear Kalman Filtering Example

- Discrete Dynamics

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{u}_{k-1}$$

$$\mathbf{y}_k = \mathbf{H}_k\,\mathbf{x}_k + \mathbf{v}_k$$

↑

$$\mathbf{F}_{k-1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad \mathbf{G}_{k-1} = \begin{bmatrix} -\frac{1}{2}(\Delta t)^2 \\ -\Delta t \end{bmatrix}$$

$$\mathbf{u}_{k-1} = g$$

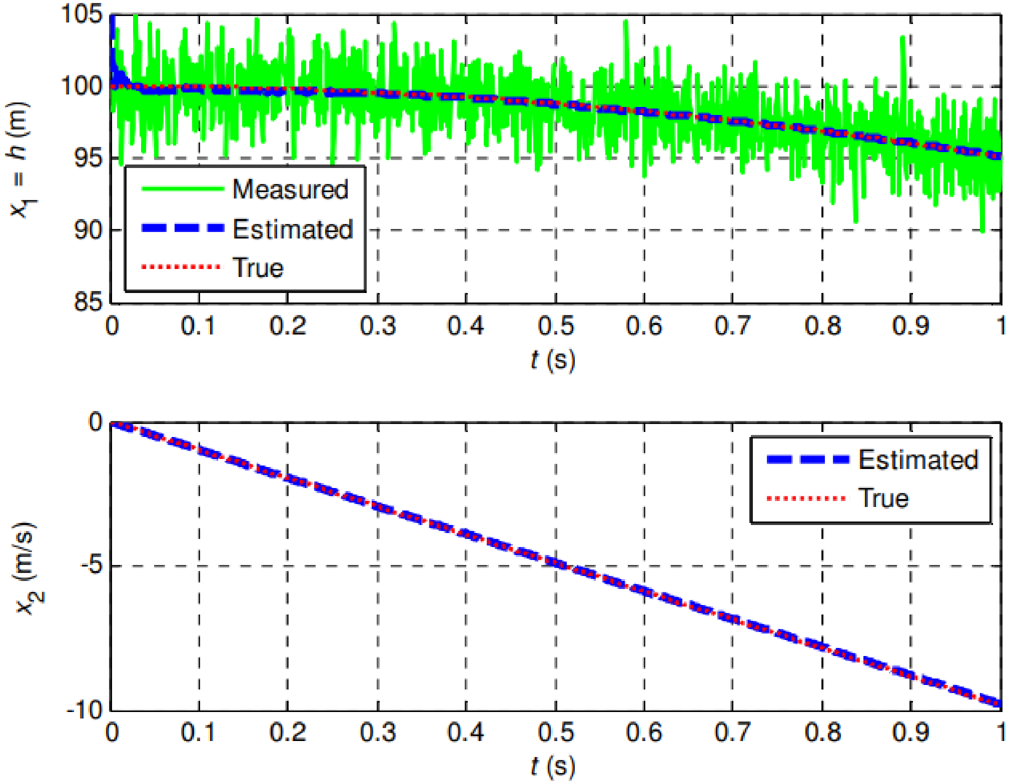
$$\mathbf{y}_k = h_k + \mathbf{v}_k \qquad \mathbf{H}_k = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Process Noise Covariance Matrix	$\mathbf{Q}_{k-1} = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}$
Measurement Noise Covariance Matrix	$\mathbf{R}_k = 4$
True Initial State Vector	$\mathbf{x}_0 = \begin{bmatrix} 100 \\ 0 \end{bmatrix}$
Assumed Initial State Vector	$\hat{\mathbf{x}}_0 = \begin{bmatrix} 105 \\ 0 \end{bmatrix}$
Assumed Initial State Error Covariance Matrix	$\mathbf{P}_0 = \begin{bmatrix} 10 & 0 \\ 0 & 0.01 \end{bmatrix}$
Time Increment	$\Delta t = 0.001$

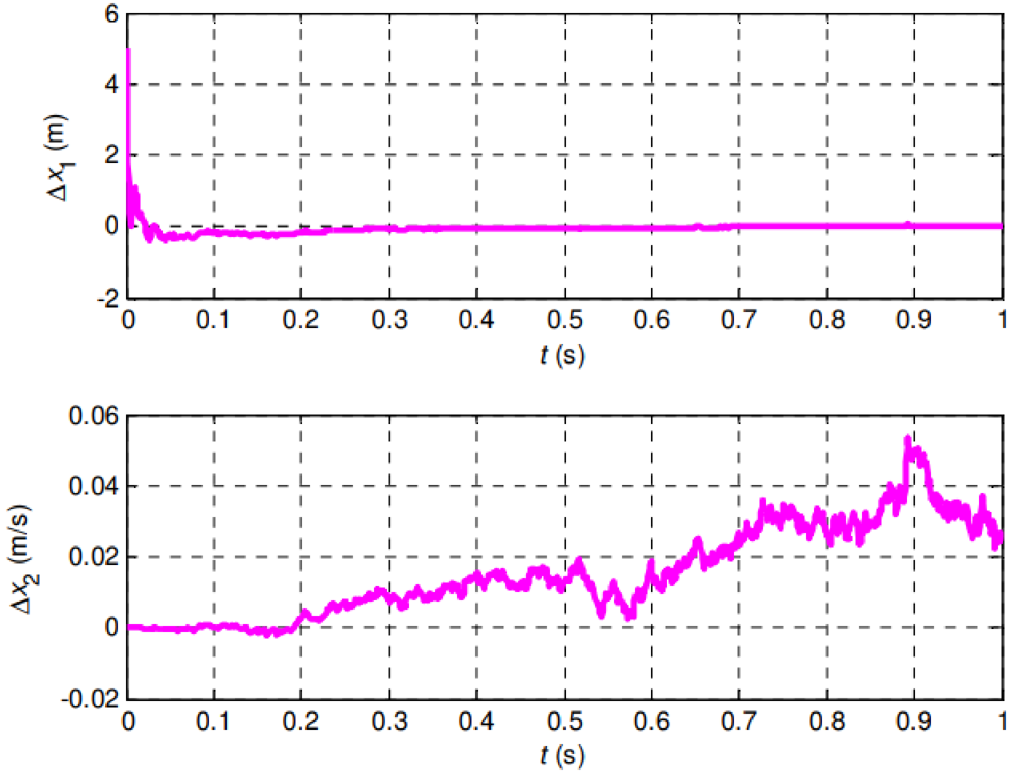
Linear Kalman Filtering Example

- $q_1 = 0, q_2 = 0$

True and estimated states



Errors btw true and estimated states



Extended Kalman Filter

- Nonlinear system

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}$$
$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k$$

⬇

$$\mathbf{A}_{k-1} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}^a_{k-1}} \qquad \mathbf{H}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}^f_k}$$

⬇

$$\mathbf{x}^f_k = \mathbf{f}(\mathbf{x}^a_{k-1}, \mathbf{u}_{k-1})$$
$$\mathbf{P}^f_k = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}^T_{k-1} + \mathbf{Q}_{k-1}$$

$$\mathbf{x}^a_k = \mathbf{x}^f_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}^f_k)$$
$$\mathbf{K}_k = \mathbf{P}^f_k\mathbf{H}^T_k(\mathbf{H}_k\mathbf{P}^f_k\mathbf{H}^T_k + \mathbf{R}_k)^{-1}$$
$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^f_k$$

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$
$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k$$

$$\mathbf{x}^f_k = \mathbf{A}_{k-1}\mathbf{x}^a_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1}$$
$$\mathbf{P}^f_k = \mathbf{A}_{k-1}\mathbf{P}_{k-1}\mathbf{A}^T_{k-1} + \mathbf{Q}_{k-1}$$

$$\mathbf{x}^a_k = \mathbf{x}^f_k + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\mathbf{x}^f_k)$$
$$\mathbf{K}_k = \mathbf{P}^f_k\mathbf{H}^T_k(\mathbf{H}_k\mathbf{P}^f_k\mathbf{H}^T_k + \mathbf{R}_k)^{-1}$$
$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^f_k$$

Summary

- linear regression
  - simple linear regression
  - multiple linear regression
- nonlinear regression
  - logistic regression
  - high-order regression
  - basis-function regression
- matrix form for regression
  - recursive least squares
- partial least squares
  - over-fitting and underfitting
  - bias/variance
  - principle component regression
  - partial least squares algorithm
  - ridge regression
  - lasso, elastic regression
- Gaussian process regression
- Kalman filtering