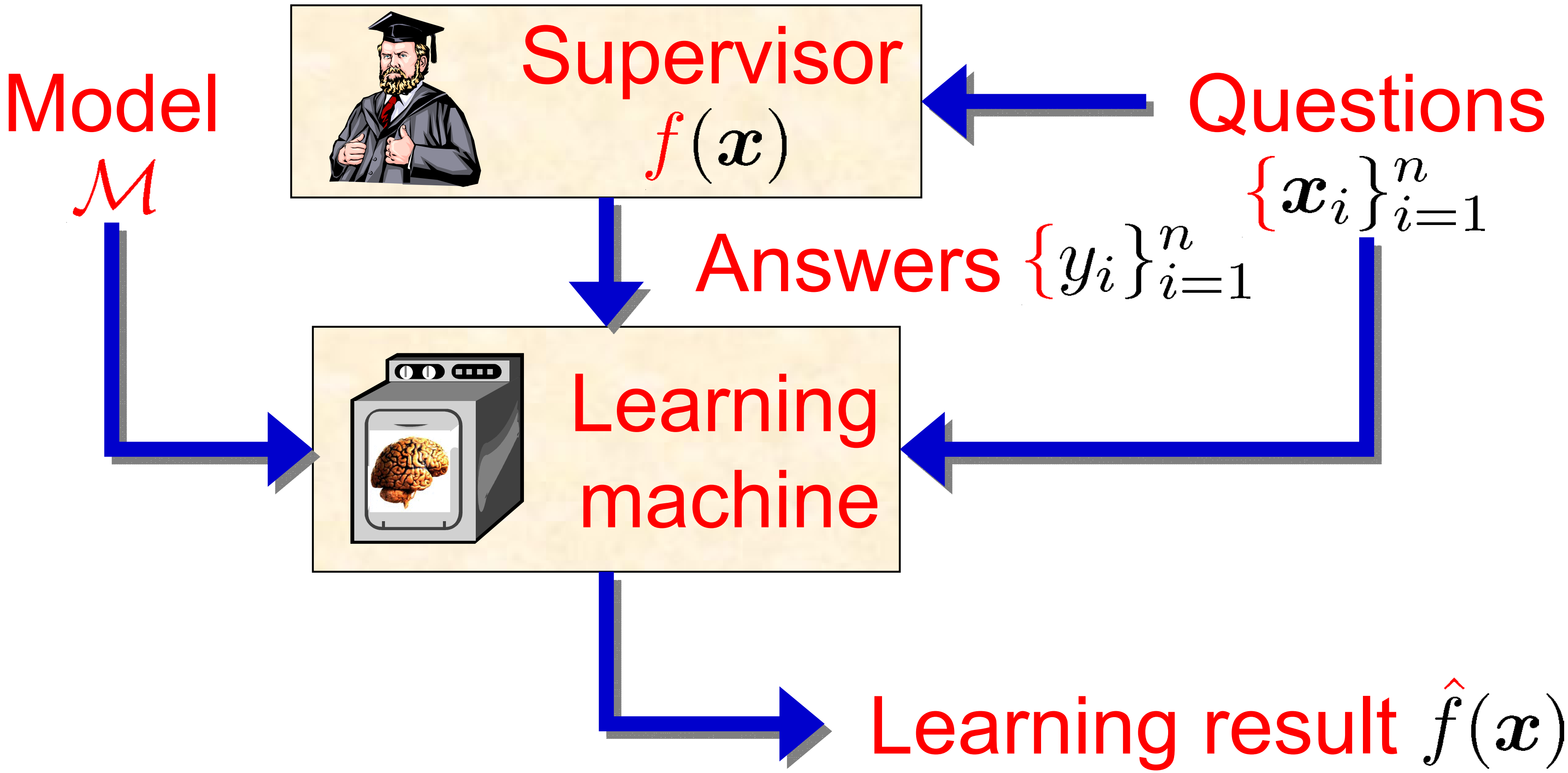
Pattern Information Processing
パターン情報処理

Masashi Sugiyama
(Department of Computer Science)
杉山　将（計算工学専攻）

Contact:　W8E-505
sugi@cs.titech.ac.jp
http://sugiyama-www.cs.titech.ac.jp/~sugi/

Diagram of Supervised Learning



Model is a set of functions from which $\hat{f}(\boldsymbol{x})$ is searched.

Notation

- $f(\boldsymbol{x})$:Learning target function
- $\mathcal{D} \subset \mathbb{R}^d$:Domain of $f(\boldsymbol{x})$
- $\boldsymbol{x}_i$:Training input point $\boldsymbol{x}_i \overset{i.i.d.}{\sim} p(\boldsymbol{x})$
- $y_i = f(\boldsymbol{x}_i) + \epsilon_i$:Training output value
- ϵ_i :zero-mean noise $\mathbb{E}_{\boldsymbol{\epsilon}} \epsilon_i = 0$
- $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$:Training examples
- $\hat{f}(\boldsymbol{x})$:Learned function
- $\mathcal{M}$:Model

3 Important Problems

$$J = \int_{\mathcal{D}} \left(\hat{f}(\boldsymbol{x}_{test}) - f(\boldsymbol{x}_{test}) \right)^2 p(\boldsymbol{x}_{test}) d\boldsymbol{x}$$

- Active learning: $\min_{\{\boldsymbol{x}_i\}_{i=1}^{n}} J$
- Model selection: $\min_{\mathcal{M}} J$
- Learning method: $\min_{\hat{f} \in \mathcal{M}} J$

Today's Plan

- Linear models / Kernel models
- Least-squares learning
  - Justification in realizable cases
  - Justification in unrealizable cases

Linear/Non-Linear Models

- Model is a set of functions from which learning result functions are searched.
- We use a family of functions $\hat{f}(\boldsymbol{x})$ parameterized by

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_p)^\top$$

- Linear model: $\hat{f}(\boldsymbol{x})$ is linear w.r.t. $\boldsymbol{\alpha}$
- Non-linear model: Otherwise

Linear Models

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{p} \alpha_i \varphi_i(\boldsymbol{x})$$

- $\{\varphi_i(\boldsymbol{x})\}_{i=1}^{p}$:Linearly independent functions
- For example, when $d = 1$
  - Polynomial

$$1, x, x^2, \ldots, x^{p-1}$$

  - Trigonometric polynomial

$$1, \sin x, \cos x, \ldots, \sin kx, \cos kx$$

$$p = 2k + 1$$

Multi-Dimensional Linear Models

- For multidimensional input $d > 1$, tensor product could be used.

$$\hat{f}(\boldsymbol{x}) = \sum_{i_1=1}^{p'} \sum_{i_2=1}^{p'} \cdots \sum_{i_d=1}^{p'} \alpha_{i_1,i_2,\ldots,i_d} \varphi_{i_1}(x^{(1)}) \varphi_{i_2}(x^{(2)}) \cdots \varphi_{i_d}(x^{(d)})$$

$$\boldsymbol{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(d)})^\top$$

- The number of parameters is $p = (p')^d$, which increases exponentially w.r.t. d.
- Infeasible for large d !

Additive Models

- For large d, we have to reduce the number of parameters.
- Additive model:

$$\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{d} \sum_{i=1}^{p'} \alpha_{i,j} \varphi_i(x^{(j)})$$

- The number of parameters is only $p = dp'$.
- However, this is too simple so its representation capability may not be rich enough in some application.

Kernel Models

- Linear model:

  $\{\varphi_i(\boldsymbol{x})\}_{i=1}^{p}$ do not depend on $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$

- Kernel model:

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i)$$

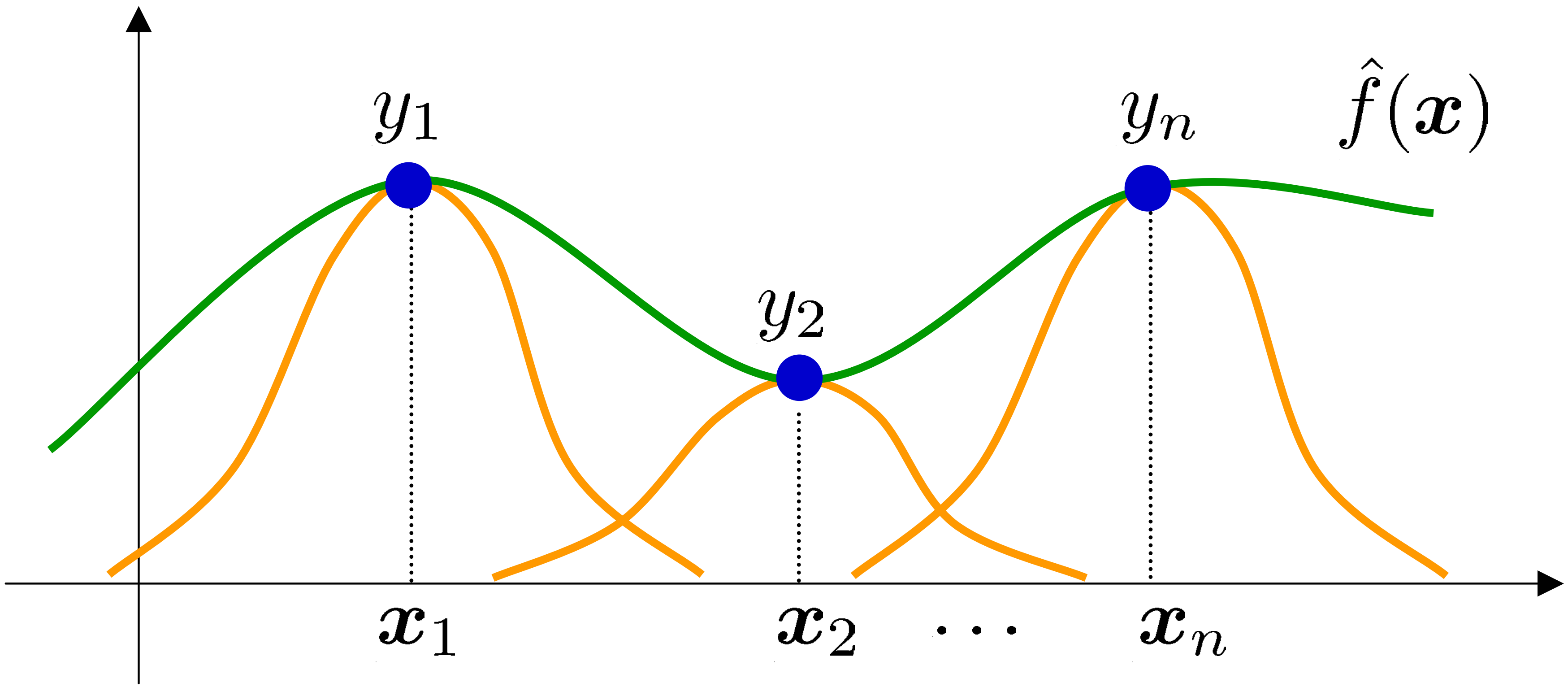
- $K(\boldsymbol{x}, \boldsymbol{x}')$:Kernel function

  e.g., Gaussian kernel

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2c^2}\right)$$

Kernel Models (cont.)

- Put kernel functions at training input points.



Kernel Models (cont.)

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i)$$

- The number of parameters is n , which is independent of the input dimensionality d .
- Although kernel model is linear, the number of parameters depends on the number of parameters.
- For this reason, mathematical treatment could be different from ordinary linear models (e.g., called non-parametric models in statistics).

Summary of Linear Models

- Tensor product

  High flexibility, high complexity
- Additive model

  Low flexibility, low complexity
- Kernel model

  Middle flexibility, middle complexity

Learning Methods

- Linear learning methods:

  Parameter vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_p)^\top$ is estimated linearly w.r.t.

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_n)^\top$$

- Non-linear learning methods: Otherwise

Linear Learning for Linear and Kernel Models

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{p} \alpha_i \varphi_i(\boldsymbol{x})$$

- In linear learning methods, a learned parameter vector is given by

$$\hat{\boldsymbol{\alpha}} = \boldsymbol{L}\boldsymbol{y}$$

$\boldsymbol{L}$:Learning matrix

- $\boldsymbol{X}_{i,j} = \varphi_j(\boldsymbol{x}_i)$:Design matrix
- Suppose $\mathrm{rank}(\boldsymbol{X}) = p$

Least-Squares Learning

- Try to make the output $\hat{f}(\boldsymbol{x}_i)$ as close to y_i as possible:

$$\hat{\boldsymbol{\alpha}}_{LS} = \underset{\boldsymbol{\alpha}}{\mathrm{argmin}}\, J_{LS}(\boldsymbol{\alpha})$$

$$J_{LS}(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \left(\hat{f}(\boldsymbol{x}_i) - y_i \right)^2$$

- Using the design matrix,

$$J_{LS}(\boldsymbol{\alpha}) = \|\boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{y}\|^2$$

How to Obtain Solutions

- Saddle-point equation:

$$\nabla J_{LS}(\hat{\boldsymbol{\alpha}}_{LS}) = 2\boldsymbol{X}^\top(\boldsymbol{X}\hat{\boldsymbol{\alpha}}_{LS} - \boldsymbol{y}) = 0$$

$$\hat{\boldsymbol{\alpha}}_{LS} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$$

- Therefore, LS is linear learning.

$$\hat{\boldsymbol{\alpha}}_{LS} = \boldsymbol{L}_{LS}\boldsymbol{y}$$

$$\boldsymbol{L}_{LS} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top$$

Justification of LS (Realizable Cases)

- Realizable: $f(\boldsymbol{x})$ is included in the model.

$$f(\boldsymbol{x}) = \sum_{i=1}^{p} \alpha_i^* \varphi_i(\boldsymbol{x})$$

- Generalization error:

$$J = \int_{\mathcal{D}} \left(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}) \right)^2 p(\boldsymbol{x}) d\boldsymbol{x}$$

$$= \|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|_{\boldsymbol{U}}^2$$

$$U_{i,j} = \int_{\mathcal{D}} \varphi_i(\boldsymbol{x}) \varphi_j(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}$$

Bias/Variance Decomposition

- Expected generalization error:

$$\mathbb{E}_\epsilon J = \mathbb{E}_\epsilon \|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|_U^2$$
$$= \underbrace{\mathbb{E}_\epsilon \|\boldsymbol{\alpha} - \mathbb{E}_\epsilon \boldsymbol{\alpha}\|_U^2}_{\text{Variance}} + \underbrace{\|\mathbb{E}_\epsilon \boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|_U^2}_{\text{Bias}}$$

$\mathbb{E}_\epsilon$:Expectation over noise

Unbiasedness and BLUE

- Unbiased estimator:

$$\mathbb{E}_{\epsilon}\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}^*$$

- Best linear unbiased estimator (BLUE): A linear estimator which has the smallest variance among all linear unbiased estimators.

$$\mathbb{E}_{\epsilon}\|\hat{\boldsymbol{\alpha}}_{BLUE} - \mathbb{E}_{\epsilon}\hat{\boldsymbol{\alpha}}_{BLUE}\|^2 \leq \mathbb{E}_{\epsilon}\|\hat{\boldsymbol{\alpha}}_{LU} - \mathbb{E}_{\epsilon}\hat{\boldsymbol{\alpha}}_{LU}\|^2$$

$$\text{for any linear unbiased estimator } \hat{\boldsymbol{\alpha}}_{LU}$$

- When $f(\boldsymbol{x})$ is realizable, $\hat{\boldsymbol{\alpha}}_{LS}$ is unbiased.
- When realizable and iid noise, it is BLUE.

Efficiency

- The Cramer-Rao lower bound: Lower bound of the variance of all (possibly non-linear) unbiased estimators.
- Efficient estimator: An unbiased estimator whose variance attains Cramer-Rao bound.
- For the linear regression model, Cramer-Rao bound is

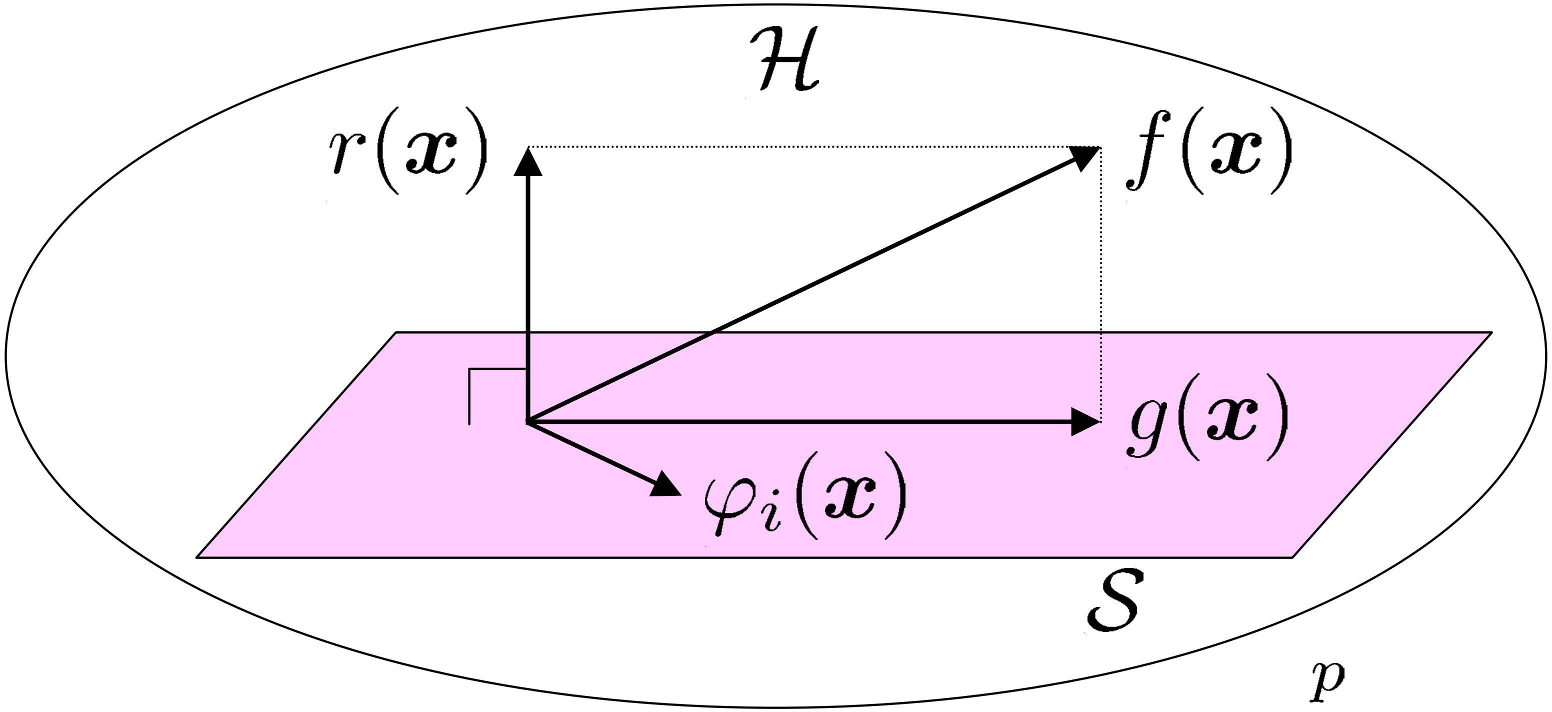
$$\sigma^2 \mathrm{tr}((\boldsymbol{X}^\top \boldsymbol{X})^{-1})_{\boldsymbol{U}}$$

- When $\epsilon_i \overset{i.i.d.}{\sim} N(0, \sigma^2)$, LS is efficient.

Justification of LS (Unrealizable Cases)

- Unrealizable: $f(\boldsymbol{x})$ is not included in the model.

$$f(\boldsymbol{x}) = g(\boldsymbol{x}) + r(\boldsymbol{x})$$



$$\int_{\mathcal{D}} \varphi_i(\boldsymbol{x}) r(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} = 0 \qquad g(\boldsymbol{x}) = \sum_{i=1}^{p} \alpha_i^* \varphi_i(\boldsymbol{x})$$

Asymptotic Unbiasedness and Efficiency

- Asymptotically unbiased estimator:

$$\mathbb{E}_{\boldsymbol{\epsilon}} \hat{\boldsymbol{\alpha}} \to \boldsymbol{\alpha}^* \text{ as } n \to \infty$$

- Asymptotically efficient estimator: An unbiased estimator whose variance asymptotically attains Cramer-Rao's lower bound.
- LS estimator is asymptotically unbiased.
- When $\epsilon_i \overset{i.i.d.}{\sim} N(0, \sigma^2)$, LS estimator is asymptotically efficient.

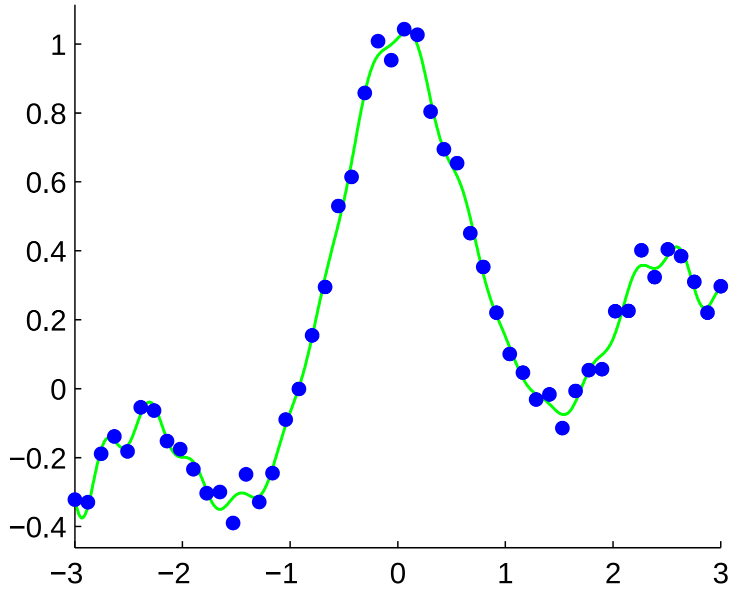
Example of LS

$$\hat{f}(\boldsymbol{x}) = \sum_{i=1}^{p} \alpha_i \varphi_i(\boldsymbol{x})$$

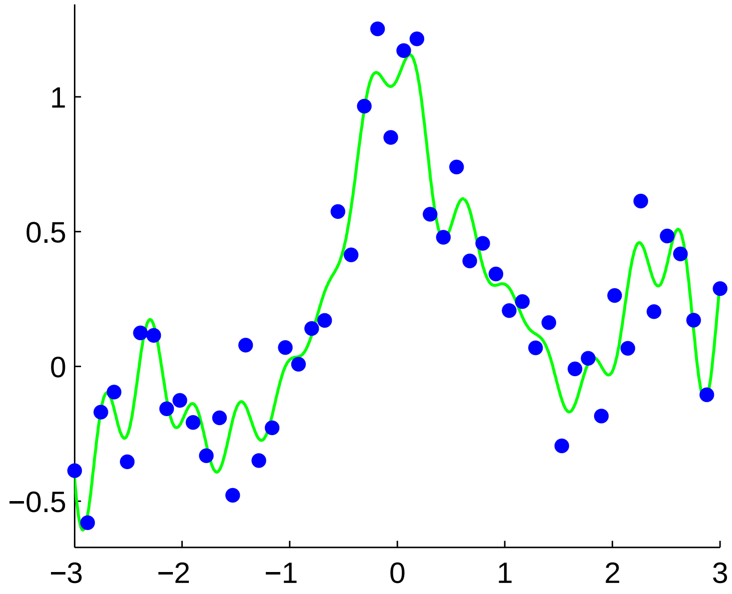
- Trigonometric polynomial model

$$1, \sin x, \cos x, \ldots, \sin 15x, \cos 15x \quad (p = 31)$$

$n = 50$



Small noise



Large noise