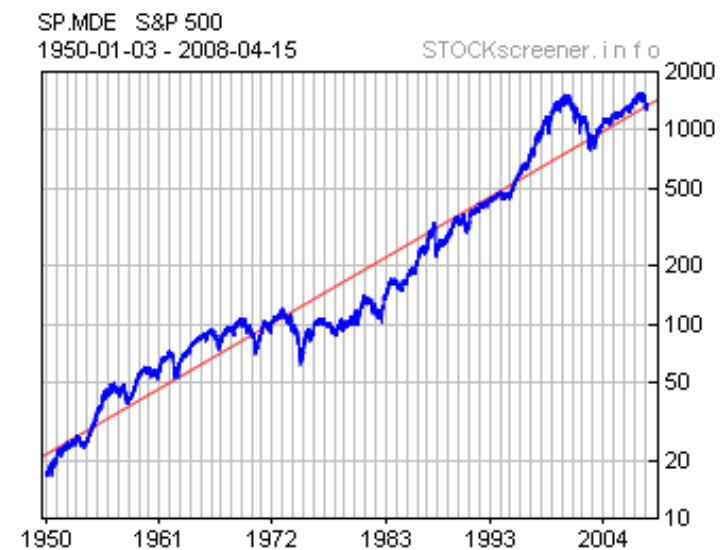


430.457

Introduction to Intelligent Systems

Prof. Songhwai Oh
ECE, SNU

LINEAR REGRESSION



Univariate Linear Regression

- Univariate linear function: $y = w_1x + w_0$, where $w_0, w_1 \in \mathbb{R}$.
- Let $\mathbf{w} = [w_0, w_1]^T$ and define

$$h_{\mathbf{w}}(x) = w_1x + w_0.$$

- **Linear regression:** Given a training set $\{(x_i, y_i) : 1 \leq i \leq N\}$, find $h_{\mathbf{w}}$ that best fits the training set.

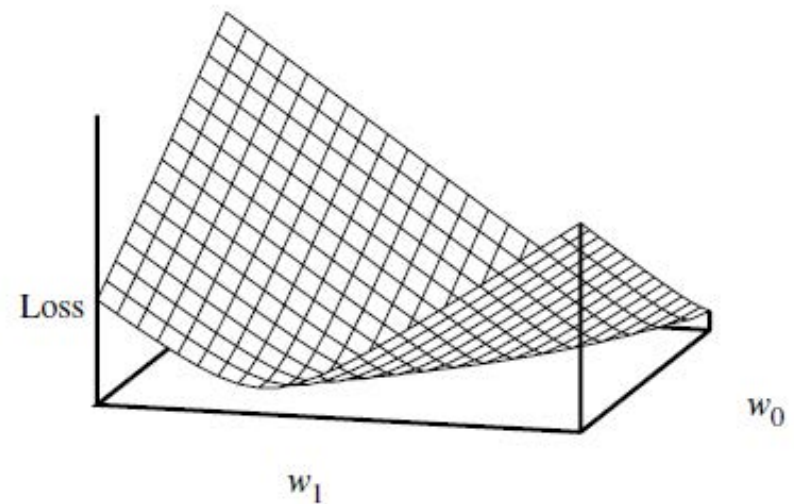
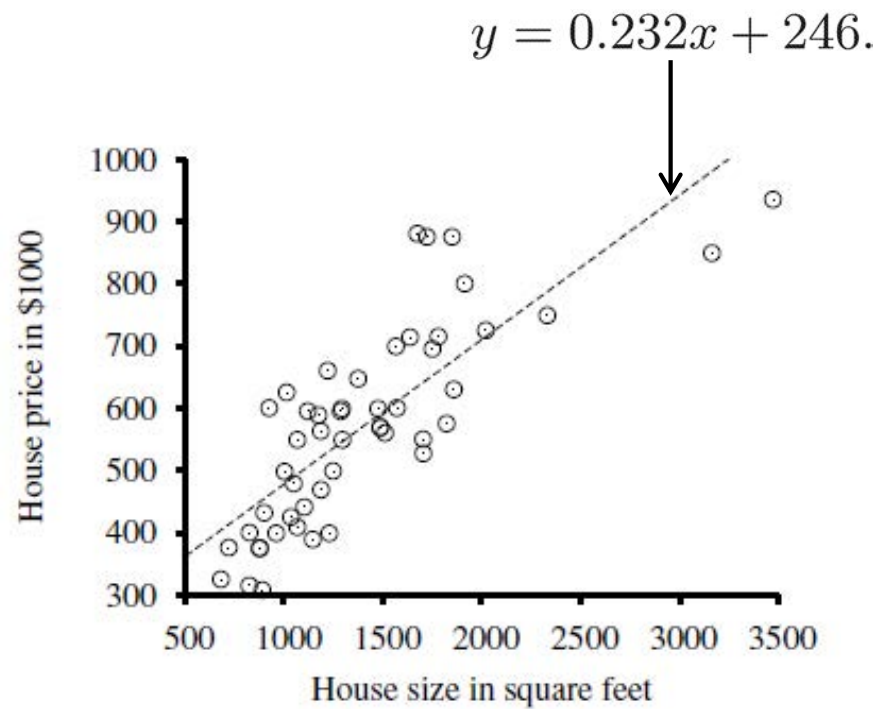
$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^N L_2(y_j, h_{\mathbf{w}}(x_j)) = \sum_{j=1}^N (y_j - h_{\mathbf{w}}(x_j))^2 = \sum_{j=1}^N (y_j - (w_1x_j + w_0))^2.$$

- Our goal is to find $\mathbf{w}^* = \arg \min_{\mathbf{w}} Loss(h_{\mathbf{w}})$.

$$\frac{\partial}{\partial w_0} \sum_{j=1}^N (y_j - (w_1x_j + w_0))^2 = 0 \text{ and } \frac{\partial}{\partial w_1} \sum_{j=1}^N (y_j - (w_1x_j + w_0))^2 = 0$$

$$w_1 = \frac{N(\sum x_j y_j) - (\sum x_j)(\sum y_j)}{N(\sum x_j^2) - (\sum x_j)^2}; \quad w_0 = (\sum y_j - w_1(\sum x_j))/N.$$

Example



$$\mathbf{w}^* = [w_0^*, w_1^*]^T = [246, 0.232]$$

$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^N (y_j - (w_1 x_j + w_0))^2.$$

Gradient Descent

```
w ← any point in the parameter space
loop until convergence do
  for each  $w_i$  in w do
     $w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} \text{Loss}(\mathbf{w})$ 
```

α : step size or learning rate.

$$\begin{aligned}\frac{\partial}{\partial w_i} \text{Loss}(\mathbf{w}) &= \frac{\partial}{\partial w_i} (y - h_{\mathbf{w}}(x))^2 \\ &= 2(y - h_{\mathbf{w}}(x)) \times \frac{\partial}{\partial w_i} (y - h_{\mathbf{w}}(x)) \\ &= 2(y - h_{\mathbf{w}}(x)) \times \frac{\partial}{\partial w_i} (y - (w_1 x + w_0))\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial w_0} \text{Loss}(\mathbf{w}) &= -2(y - h_{\mathbf{w}}(x)) \\ \frac{\partial}{\partial w_1} \text{Loss}(\mathbf{w}) &= -2(y - h_{\mathbf{w}}(x)) \times x\end{aligned}$$

$$w_0 \leftarrow w_0 + \alpha (y - h_{\mathbf{w}}(x)); \quad w_1 \leftarrow w_1 + \alpha (y - h_{\mathbf{w}}(x)) \times x$$

Batch gradient descent: $w_0 \leftarrow w_0 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j)); \quad w_1 \leftarrow w_1 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j)) \times x_j$
(Steepest descent)

Stochastic gradient descent: processes one data point at a time

Multivariate Linear Regression

- When $\mathbf{x}_j \in \mathbb{R}^n$, the hypothesis space for linear regression is spanned by functions of the following form.

$$h_{\mathbf{w}}(\mathbf{x}_j) = w_0 + \sum_{i=1}^n w_i x_{j,i}.$$

- Now, redefine $\mathbf{x}_j = [1, x_{j,1}, x_{j,2}, \dots, x_{j,n}]^T \in \mathbb{R}^{n+1}$. Then

$$h_{\mathbf{w}}(\mathbf{x}_j) = \sum_{i=0}^n w_i x_{j,i} = \mathbf{w}^T \mathbf{x}_j.$$

- Solution to linear regression:

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_j L_2(y_j, \mathbf{w}^T \mathbf{x}_j).$$

- Gradient descent update equation: $w_i = w_i + \alpha \sum_j x_{j,i}(y_j - h_{\mathbf{w}}(\mathbf{x}_j))$.
- Closed form solution: $\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$.

Closed-Form Solution to Linear Regression

- Let $\mathbf{y} = [y_1, \dots, y_N]^T$ and $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}$ be the data matrix (or design matrix).

Then

$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^N (y_j - \mathbf{w}^T \mathbf{x}_j)^2 = (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}).$$

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} Loss(h_{\mathbf{w}}) &= \frac{\partial}{\partial \mathbf{w}} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) \\ &= \frac{\partial}{\partial \mathbf{w}} (\mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{X}\mathbf{w} - \mathbf{w}^T \mathbf{X}^T \mathbf{y} + \mathbf{w}^T \mathbf{X}^T \mathbf{X} \mathbf{w}) \\ &= -2\mathbf{X}^T \mathbf{y} + 2\mathbf{X}^T \mathbf{X} \mathbf{w} \end{aligned}$$

- Setting it to zero, we have $\mathbf{X}^T \mathbf{X} \mathbf{w} = \mathbf{X}^T \mathbf{y}$, which is known as a **normal equation**. If $\mathbf{X}^T \mathbf{X}$ is invertible, we have

$$\mathbf{w}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

General Linear Model

$$y = \sum_{i=1}^M w_i h_i(x)$$

$h_i[n]$: nonlinear function of n

$$h_i(x) = x^{i-1}$$

$$h_i(x) = \exp\left(-\frac{1}{2s^2}(x - \mu_i)^2\right)$$

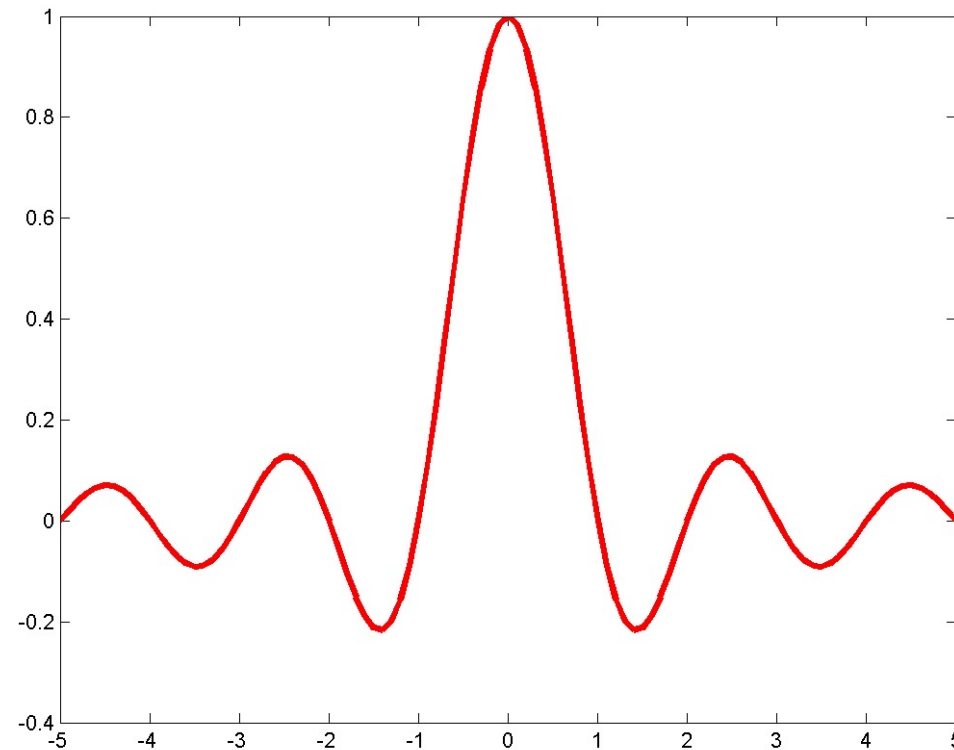
s and μ_i are fixed parameters.

$$\mathbf{y} = \mathbf{H}\mathbf{w}$$

$$\mathbf{H} = \begin{bmatrix} h_1(x_0) & h_2(x_0) & \cdots & h_M(x_0) \\ h_1(x_1) & h_2(x_1) & \cdots & h_M(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_{N-1}) & h_2(x_{N-1}) & \cdots & h_M(x_{N-1}) \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix}$$

$$\mathbf{w}^* = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}$$

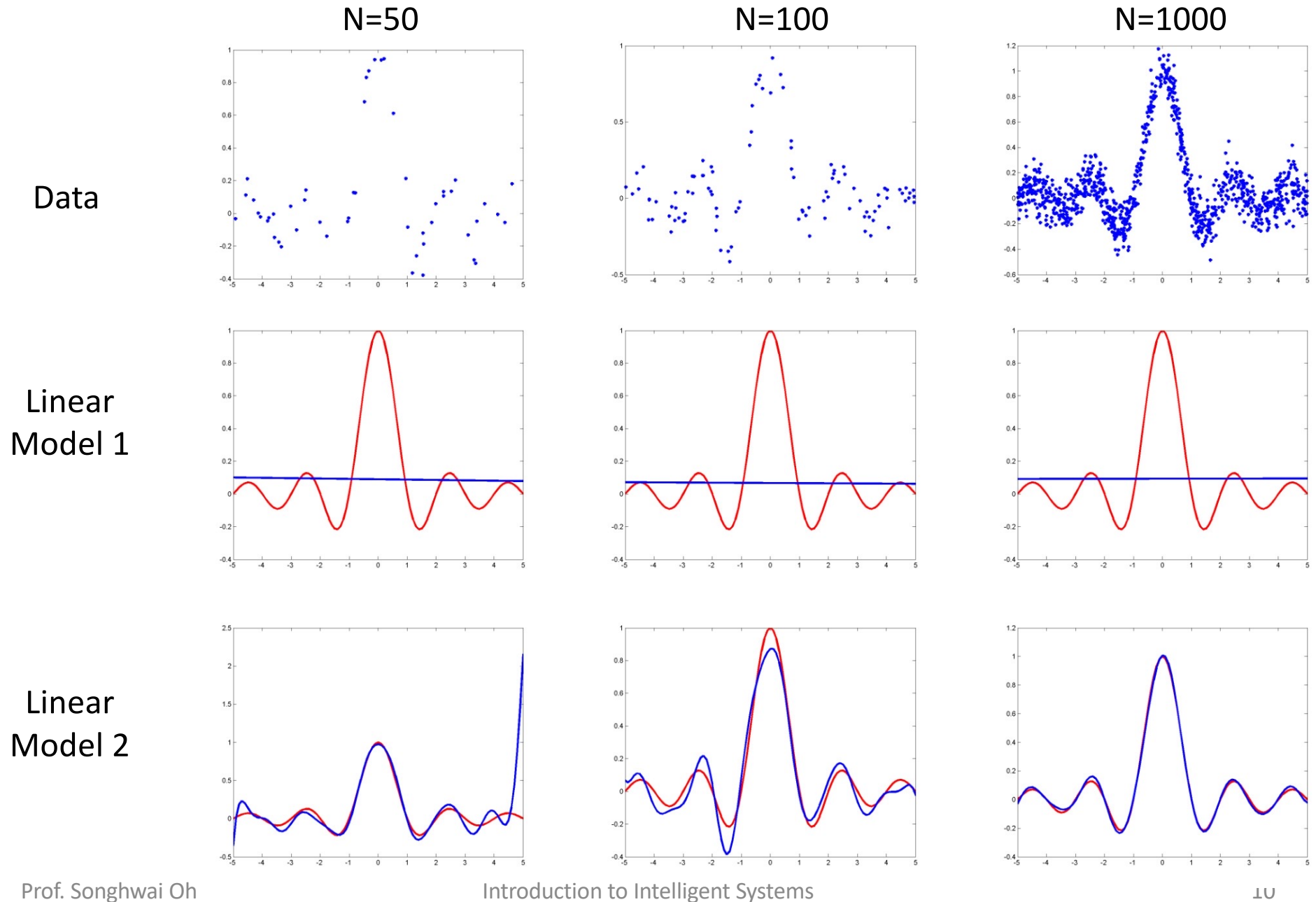
Example: Linear modeling of the SINC function



Model 1: $y = w_0 + w_1x$

Model 2: $y = \sum_{i=1}^M w_i h_i(x)$ $h_i(x) = \exp\left(-\frac{1}{2s^2}(x - \mu_i)^2\right)$

Example: Linear modeling of the SINC function



Regularization

- To avoid overfitting, we may use regularization

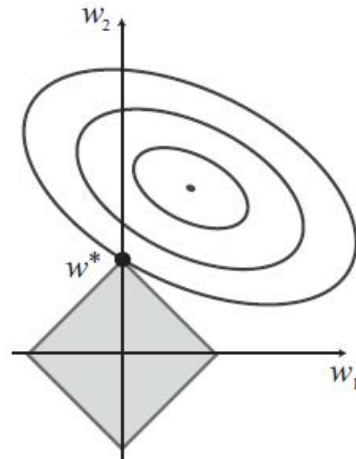
$$Cost(h) = EmpLoss(h) + \lambda Complexity(h),$$

where we can consider

$$Complexity(h_{\mathbf{w}}) = L_q(\mathbf{w}) = \sum_i |w_i|^q.$$

- L_1 regularization if $q = 1$; L_2 regularization if $q = 2$.
- L_1 regularization tends to produce a sparse model, i.e., it sets many weights to zero. A sparse model is less likely to overfit.

L_1 Regularization



L_2 Regularization

