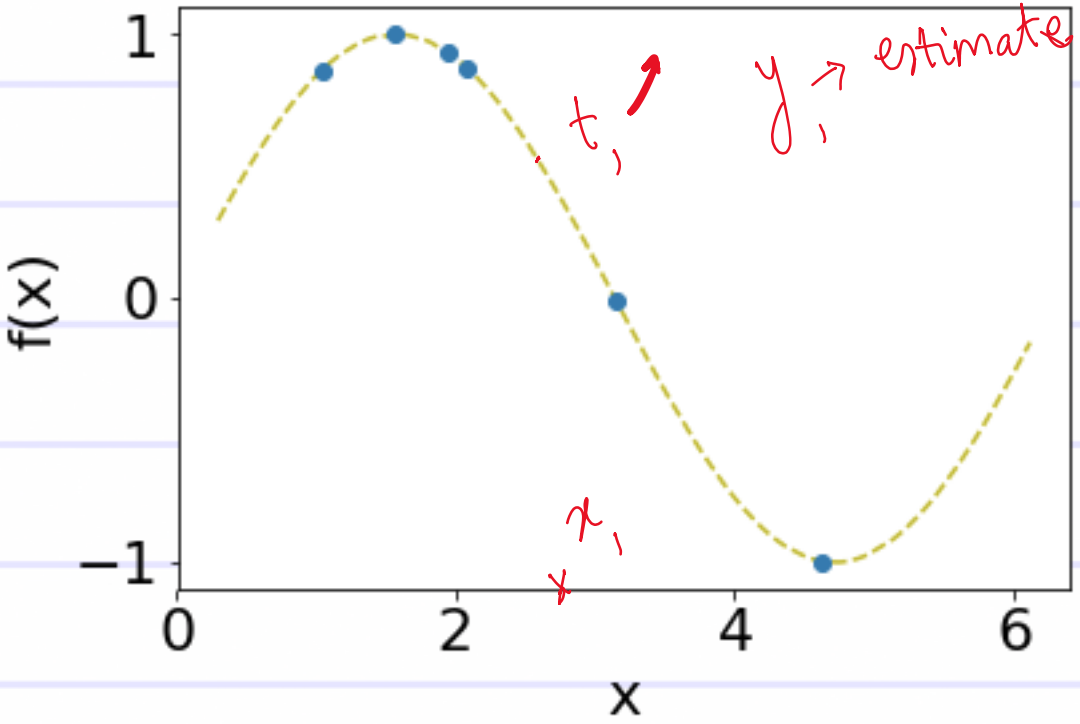
Linear Regression

EE698V - Machine Learning for Signal Processing

Vipul Arora



$x \rightarrow [F] \rightarrow y$

training data

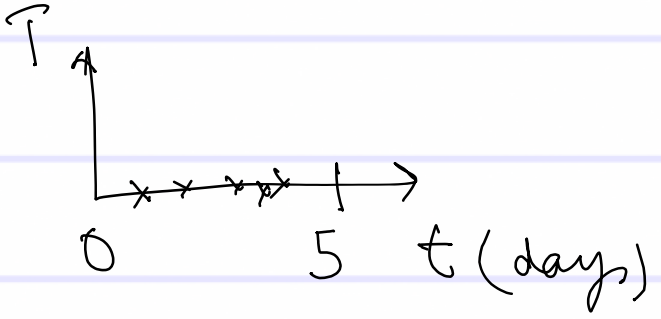
x_1 ~~y_1~~ t_1

x_2 ~~y_2~~ t_2

⋮

x_5 ~~y_5~~ t_5

Given the blue points, how to estimate f(x) for any x



Actual value = t

Let us take a linear model

estimate

$$y = w_0 + w_1 x + w_2 x^2 + \ldots + w_D x^D$$

$x \rightarrow [F] \rightarrow y = \begin{bmatrix} 1 & x & x^2 & \cdots & x^D \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix}$

$D+1$ unknowns $\rightarrow w_0, \ldots, w_D$

5 eqⁿ s

$$\begin{bmatrix} x_1 \\ \vdots \\ x_5 \end{bmatrix} \qquad \begin{bmatrix} y_1 \\ \vdots \\ y_5 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^D \\ & \vdots & & & \\ 1 & x_5 & x_5^2 & \cdots & x_5^D \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{bmatrix}$$

$\vec{y} = \Phi \vec{w}$

$y = \Phi w$

if $D+1 = S$, then

$w = \Phi^{-1} y$ unique.

N

if $D+1 < S$, then

minimize $\|t - y\|$

$$y_1 = \begin{bmatrix} y_1 \\ y_5 \end{bmatrix}$$

$$y = \Phi w$$

least squares solution

$$\min \quad E = (t - \Phi w)^T (t - \Phi w) = \| t - y \|^2$$

max

min

$$\frac{\partial E}{\partial w} = \left(\frac{\partial}{\partial w} (t - \Phi w) \right) (t - \Phi w) \times 2$$

$$0 = -\Phi^T (t - \Phi w) \times 2$$

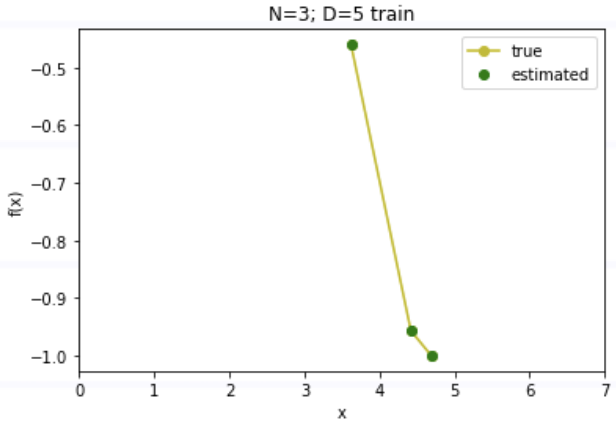
$$0 = -\Phi^T t + \Phi^T \Phi w$$

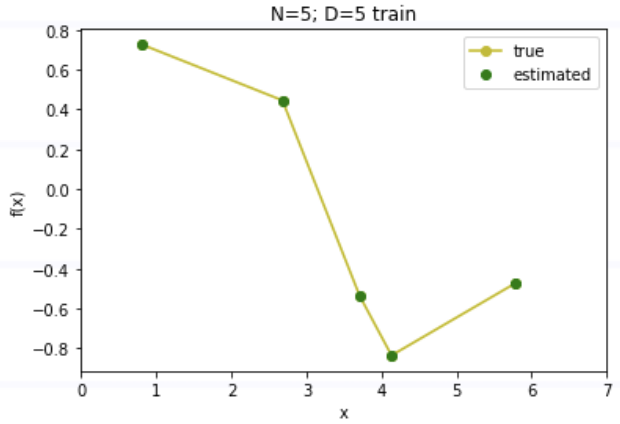
$$w = (\Phi^T \Phi)^{-1} \Phi^T t$$

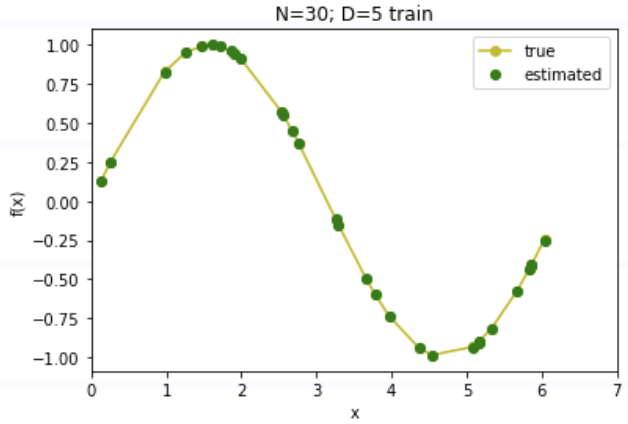
$$\frac{\partial (u^T v)}{\partial w} = \frac{\partial u}{\partial w} v + \frac{\partial v}{\partial w} u$$

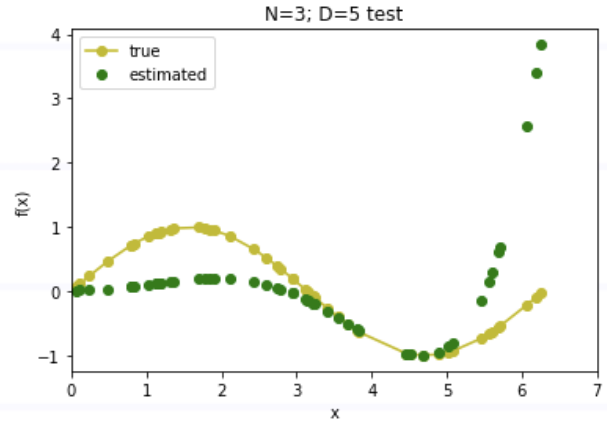
$$\frac{\partial}{\partial w} \Phi u = \frac{\partial u}{\partial w} \Phi^T$$

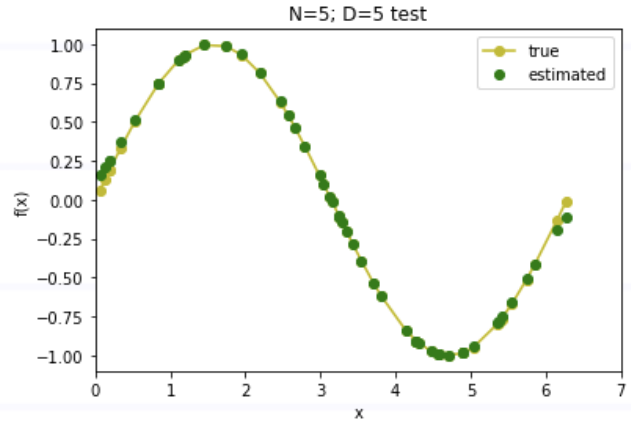
Effect of increasing data points (fixed D)

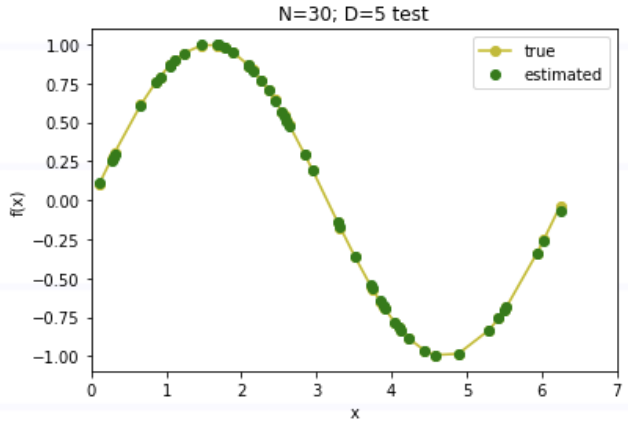




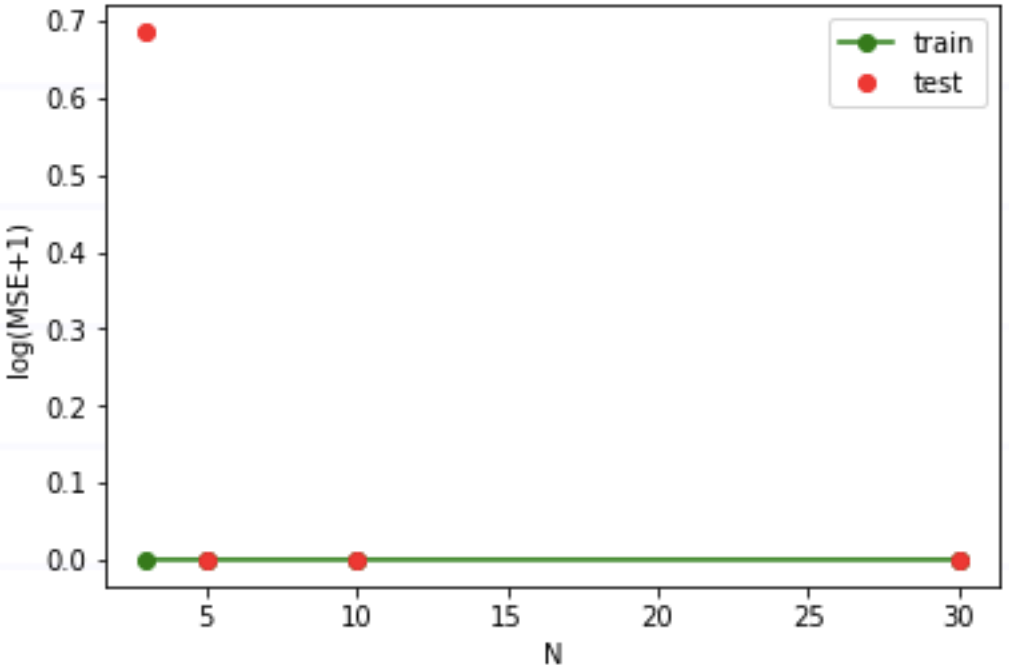




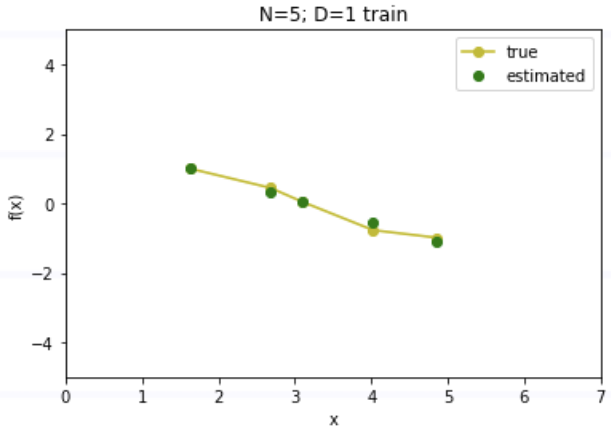


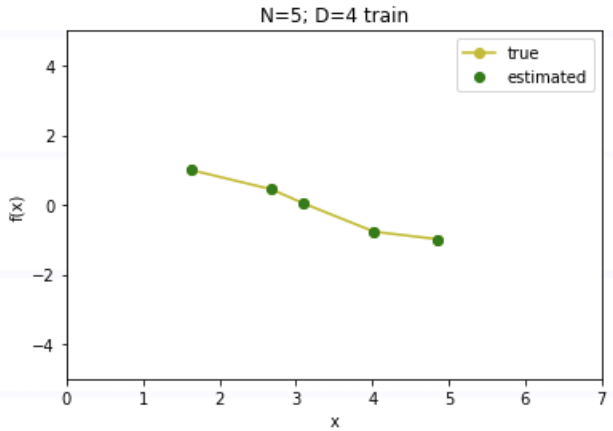


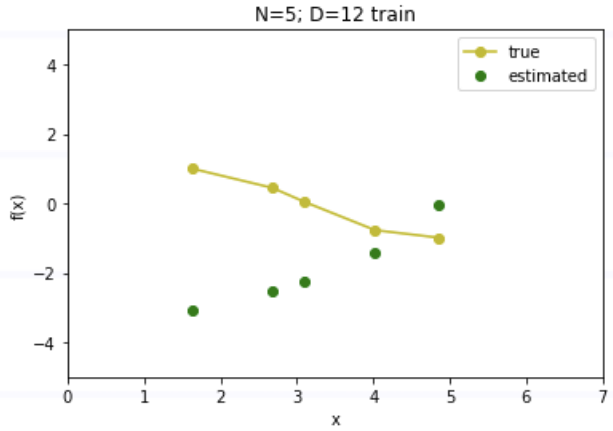
Effect of increasing data (fixed model size)

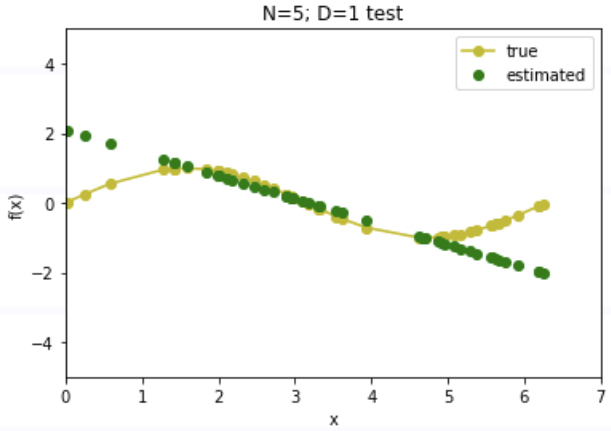


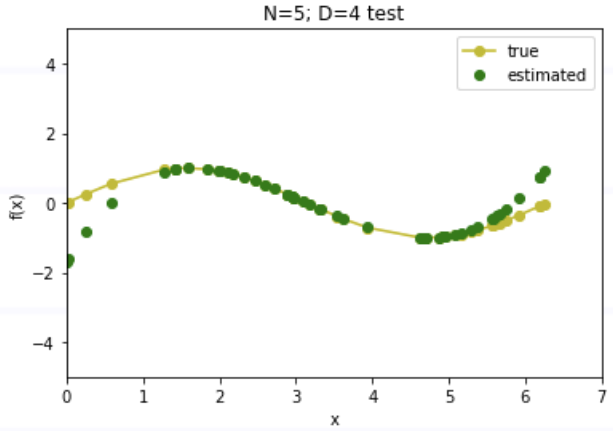
Effect of changing model size (same data)

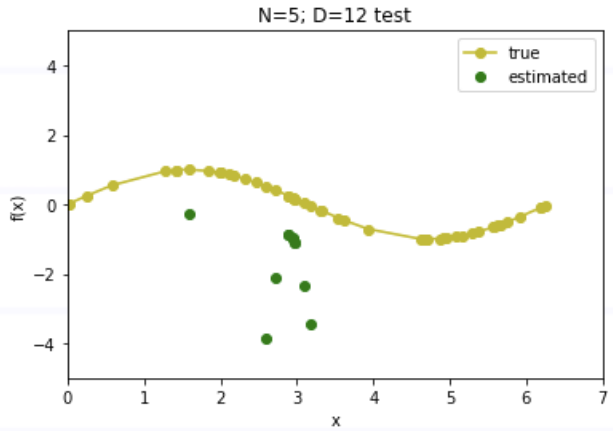


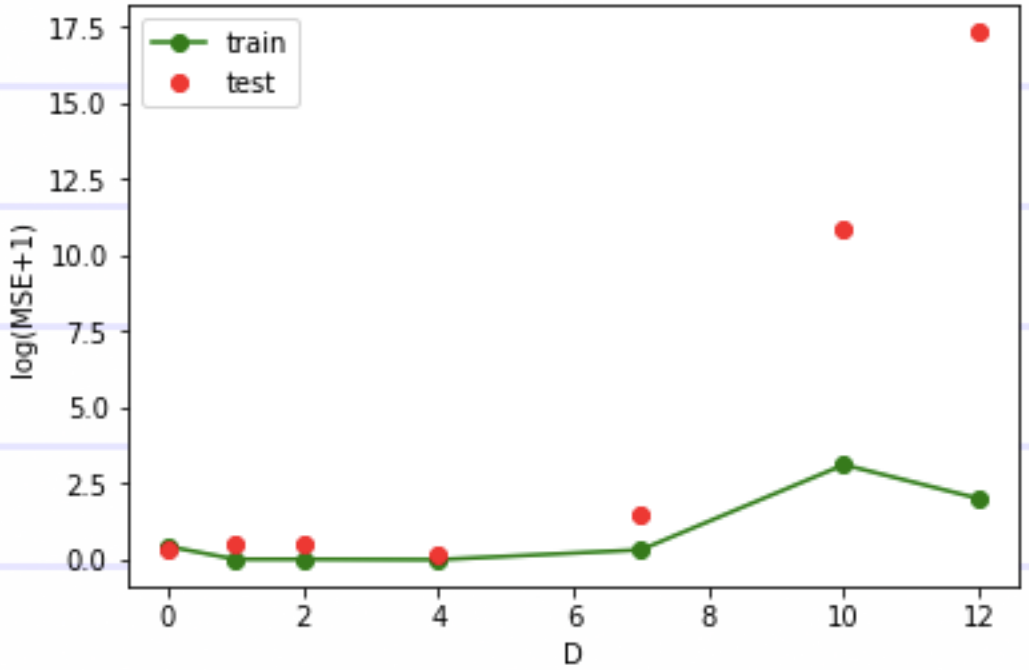











train
test
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
log(MSE+1)
0
2
4
6
8
10
12
D

L2 Regularization or Ridge Regression

$$y = \Phi w$$

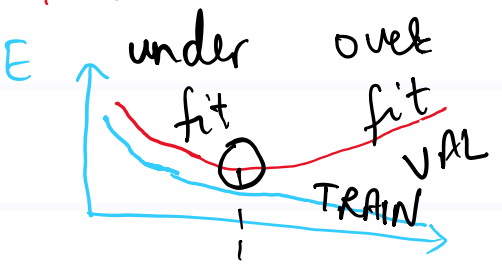
$$E = (t - \Phi w)^T (t - \Phi w) + \lambda w^T w$$

h.p.

$$\frac{\partial E}{\partial w} = \left(-\Phi^T t + \Phi^T \Phi w + \lambda w\right) \times 2 = 0$$

$$\Rightarrow \quad w = \left(\lambda I + \Phi^T \Phi\right)^{-1} \Phi^T t$$

RIDGE REGRESSION



model capacity

Validation

Another L2 normalization: Min-norm

$D+1 > N$

$$E = |t - \Phi w| + \lambda w^T w$$

$$\frac{\partial E}{\partial w} = -\Phi^T \operatorname{sgn}(t - \Phi w) + 2\lambda w$$

$$= 0$$

$$w = \frac{1}{2\lambda} \Phi^T \operatorname{sgn}(t - \Phi w) \quad \text{--- (i)}$$

Consider, $t = \Phi w = \frac{\Phi \Phi^T}{2\lambda} \operatorname{sgn}(t - \Phi w)$

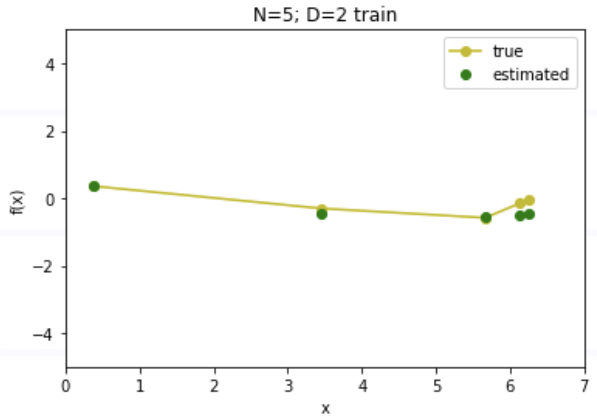
$$\Rightarrow \frac{1}{2\lambda} \operatorname{sgn}(t - \Phi w) = (\Phi \Phi^T)^{-1} t$$

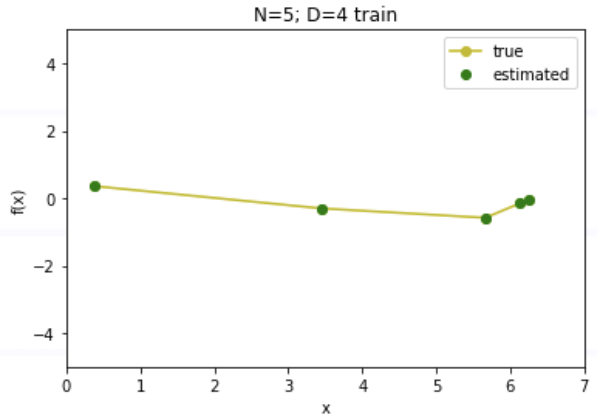
$\therefore$ from (1),

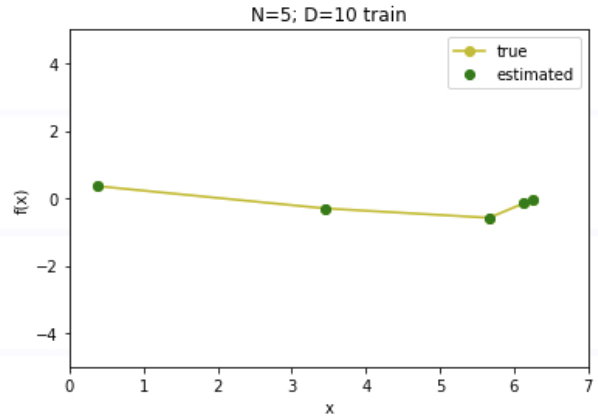
$$w = \Phi^T (\Phi \Phi^T)^{-1} t$$

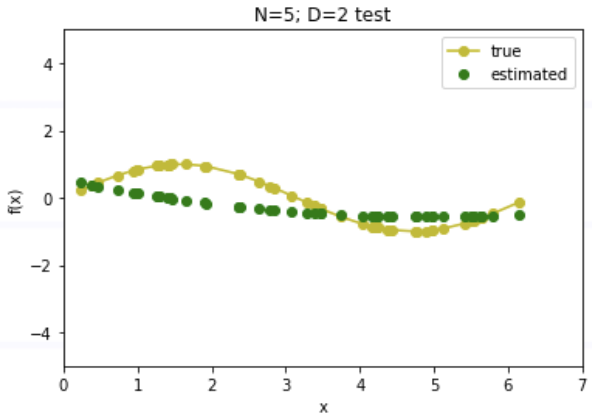
Minimum norm solution

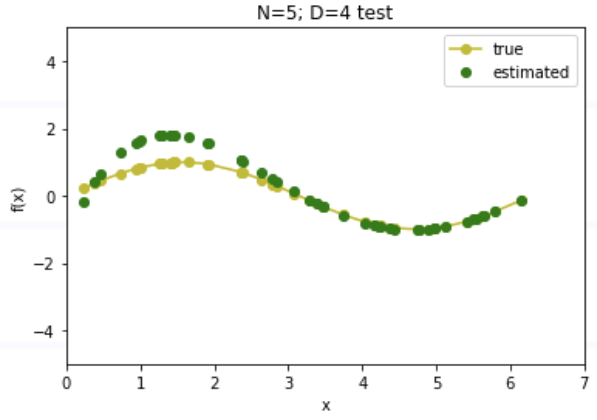
Effect of changing model size (fixed data)

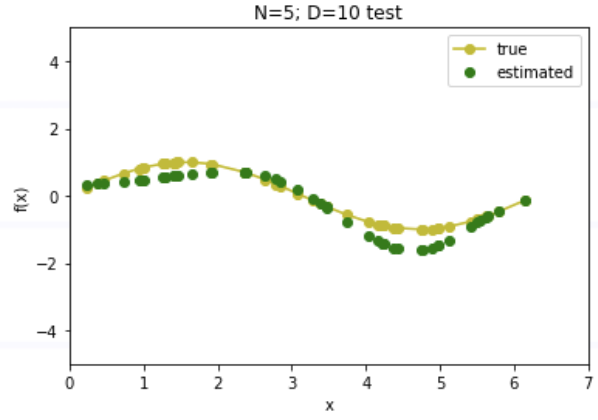


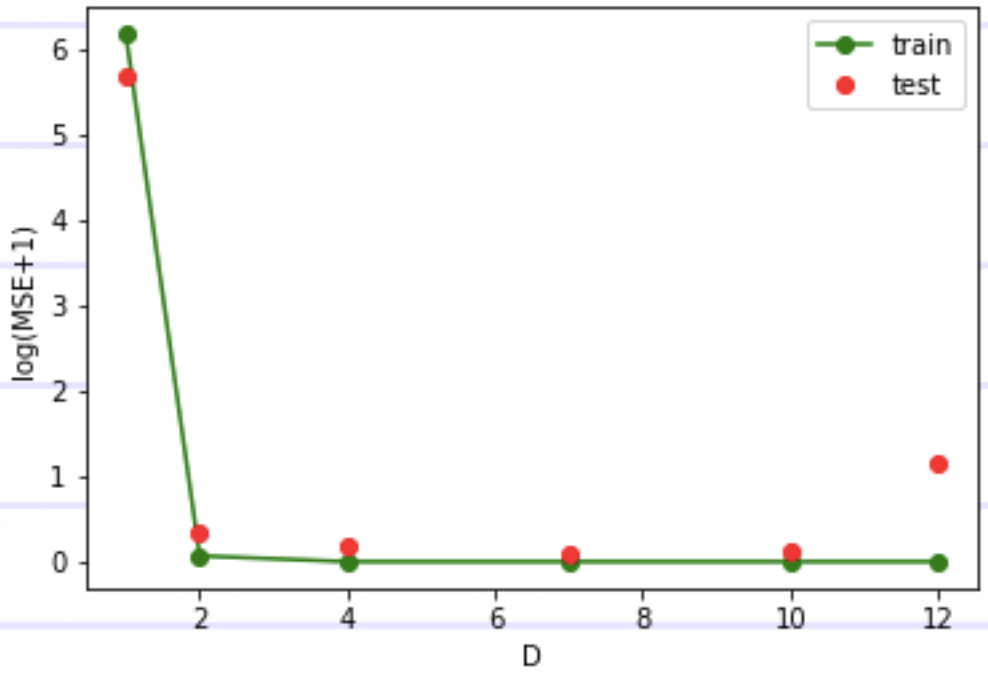






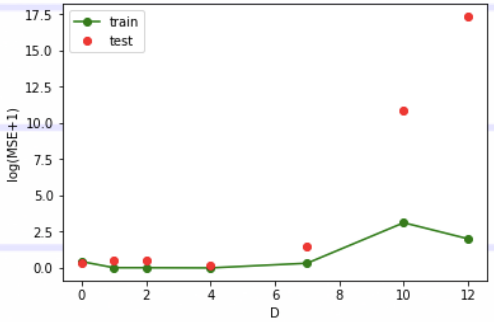






- Does better for large D, but worse for small D

Recall:



- Caution: I have used single run, not averaged over many runs
- A hybrid approach may help

Regularization

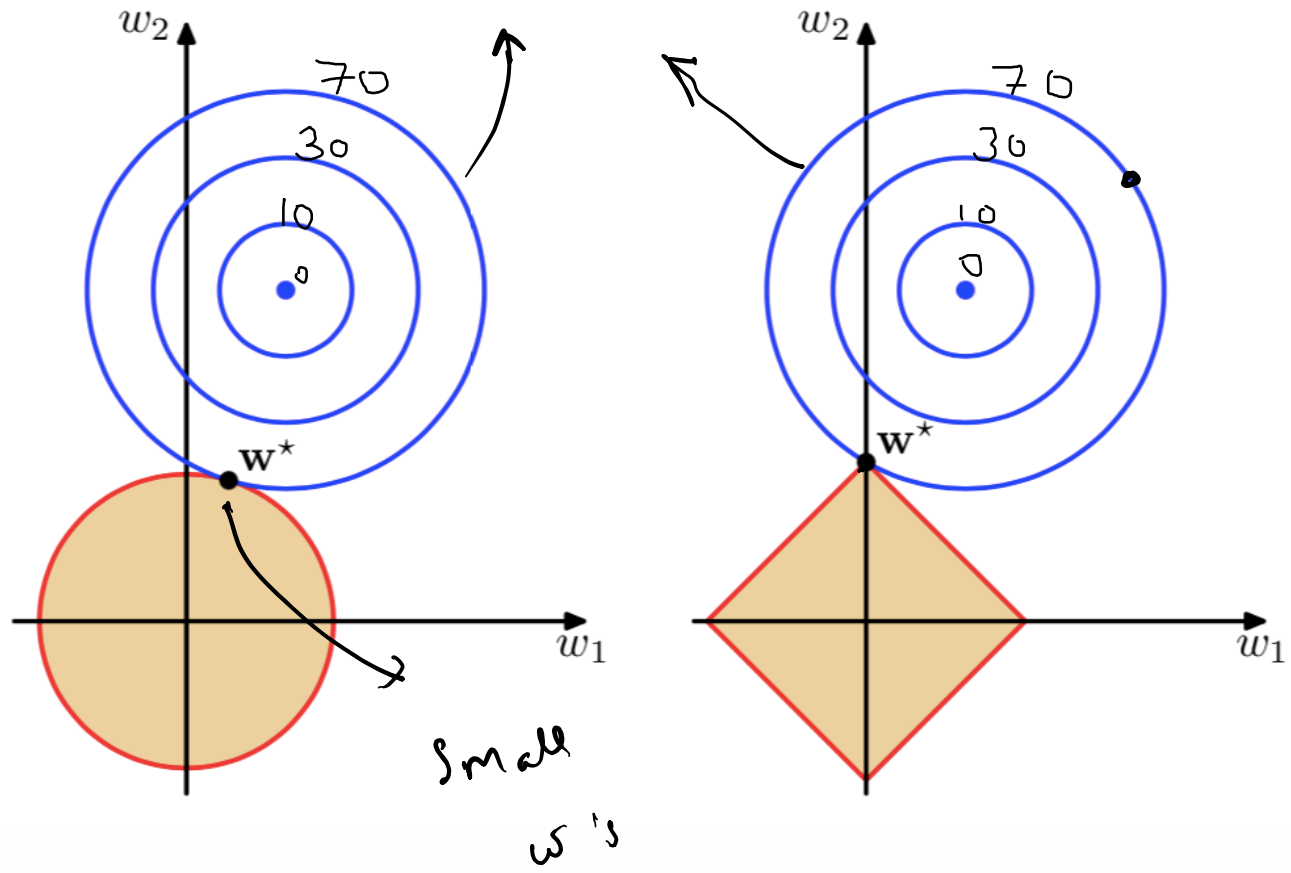
In general, with least squares loss,

$$E = \underbrace{(t - \Phi w)^T (t - \Phi w)}_{\text{least sq. loss}} + \underbrace{\lambda \sum_j |w_j|^q}_{\text{reg. term}}$$

$q = 1$, LASSO

$q = 2$, Ridge Regression

$|t - \Phi w|^2$ ← on train data



$\|w\|^2$

small w's

$\|w\|$

$|w_1| + |w_2| = \text{const}$

$y = \Phi w$

$= [1 \; x \; x^2 \; \dots] \begin{bmatrix} w_0 \\ w_1 \\ \vdots \end{bmatrix}$

PRML: Fig 3.4

Feature selection

Multiple Outputs

Read PRML Sec. 3.1.5

$y_i \rightarrow y_i \quad \forall \quad i = 1, 2, \ldots 5$

y_i is $K \times 1$ vector

$$\begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_5^T \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^D \\ \vdots & & & & \\ 1 & x_5 & x_5^2 & \ldots & x_5^D \end{bmatrix} \begin{bmatrix} w_{11} & \ldots & w_{1K} \\ w_{21} & & w_{2K} \\ \vdots & & \vdots \\ w_{D1} & & w_{DK} \end{bmatrix}$$

Y $5 \times K$ $\quad$ Φ $\quad$ $5 \times D$ $\quad$ W $\quad$ $D \times K$

$$W = \left(\Phi^T \Phi\right)^{-1} \Phi^T T$$

same derivation as seen before

Multiple Inputs

eg. $\Phi = \begin{bmatrix} 1 & e^{(x-1)^2} & e^{(x-5)^2} & \cdots \end{bmatrix}$

- Φ can be a function of x, instead of just x^d (polyn.)
- In general, for single sample

$$y^T = \Phi(x) W$$

i.e.

$$y_j = \sum_i \phi_i(x) w_{ij}$$

$j = 0, 1, 2, \ldots, N_y - 1$

$i = 0, 1, 2, \ldots, D$

For multiple samples

$$Y = \Phi(X)\, W$$

$$y_{j,t} = \sum_i \phi_i(x_t)\, w_{ij}$$

$j = 0, 1, \ldots, N_y - 1$

$i = 0, 1, \ldots, D$

$t = 0, 1, \ldots, T-1$ samples

References

- "Pattern Recognition and Machine Learning", C.M. Bishop, 2nd Edition, Springer, 2011.
  - **Section 1.1 (highly recommended)**
  - **Section 3.1 (highly recommended)**
- Matrix Calculus – wikipedia https://en.wikipedia.org/wiki/Matrix_calculus