CIS 635 - Knowledge Discovery & Data Mining

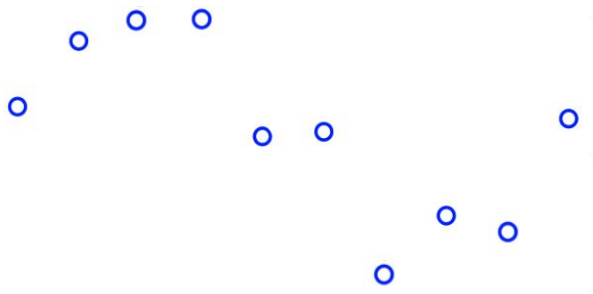
- Linear to Polynomial Regression
- Model Regularization

Plan

- LR to Polynomial Regression
- Regularization
  - Theory
  - Practical - Notebook presentation

Non linear data/function

- Does this data points seem familiar matching a known function?

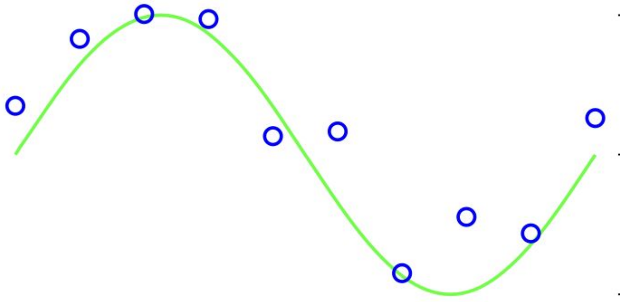


Non linear data/function

- Does this data points seem familiar matching a known function?

    - A Sinusoidal function

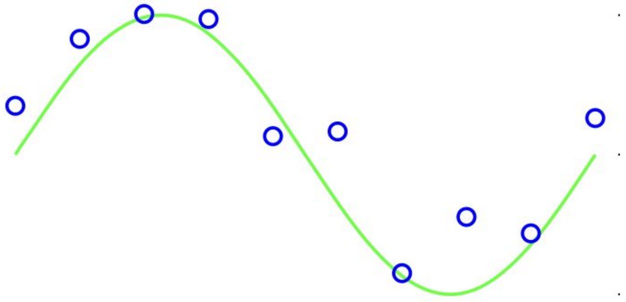
$$y(t) = A\sin(\omega t + \varphi) = A\sin(2\pi ft + \varphi)$$



Non linear data/function

- Does this data points seem familiar matching a known function?
    - A Sinusoidal function

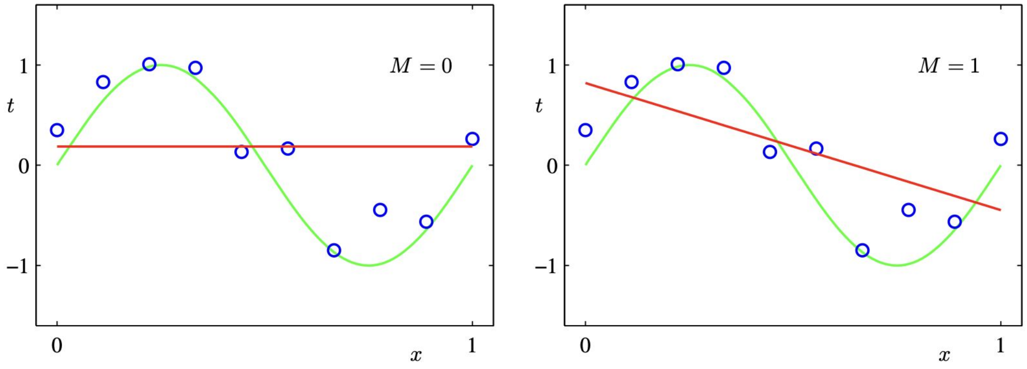
$$y(t) = A\sin(\omega t + \varphi) = A\sin(2\pi f t + \varphi)$$



Clearly this is not a linear function; right?

Non linear data/function

- Does this data points seem familiar matching a known function?
- Can we approximate this function using LR ?



$$\hat{y} = \beta_0 + \beta_1 x$$

LR will not work; right?

What no-linear functions we are aware of?

- Can you recall any nonlinear function you learned at your high school/colleges?

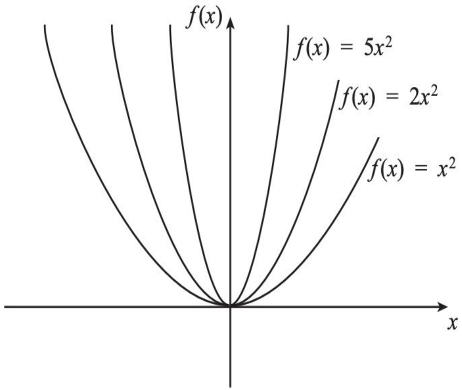
ref

What no-linear functions we are aware of?

- Can you recall any nonlinear function you learned at your high school/colleges?
- **Quadratic (x^2)**

$$f(x) = x^2, \qquad f(x) = 2x^2, \qquad f(x) = 5x^2.$$

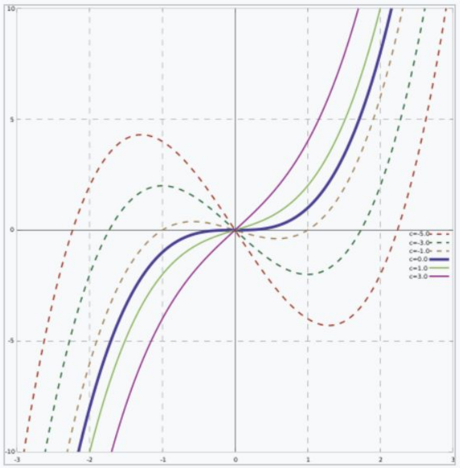
What is the impact of changing the coefficient of x^2 as we have done in these examples? One way to find out is to sketch the graphs of the functions.



ref

What no-linear functions we are aware of?

- Can you recall any nonlinear function you learned at your high school/colleges?
- **Cubic (x^3)**



Cubic functions of the form

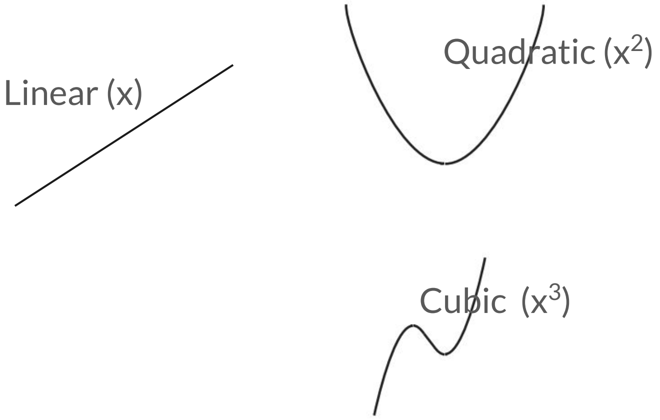
$y = x^3 + cx.$

The graph of any cubic function is similar to such a curve.

What no-linear functions we are aware of?

- Can you recall any nonlinear function you learned at your high school/colleges?
- Quadratic (x^2)
- Cubic (x^3)
-

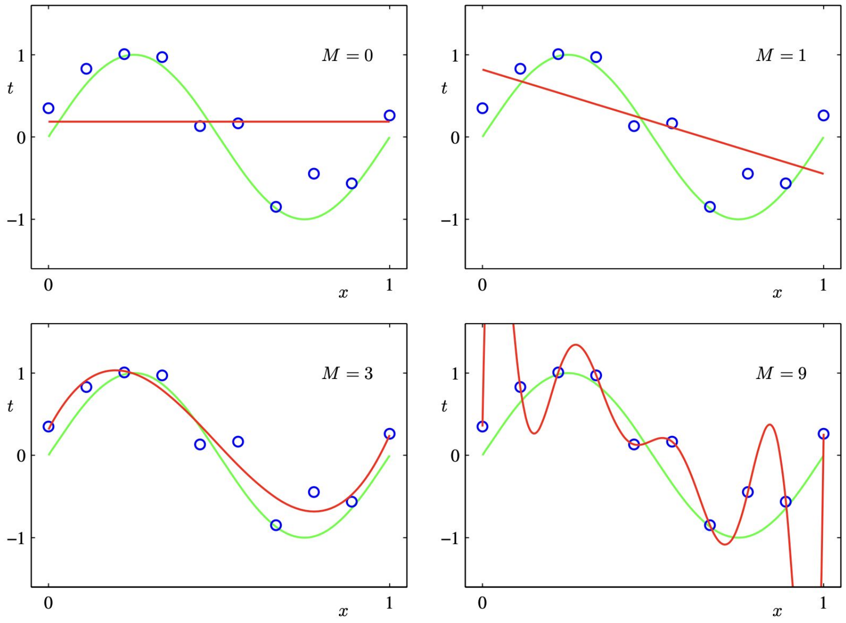
$$\hat{y} = \beta_0 + \beta_1 x$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3 + \ldots$$



ref

LR to Polynomial Regression

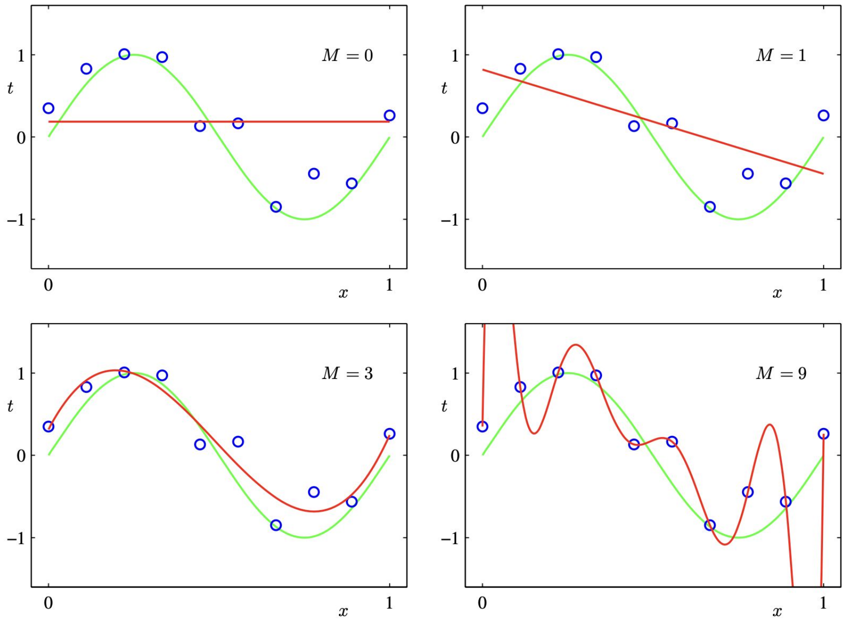
- Polynomial function
  - M is the order ..



$$\hat{y} = \beta_0 + \beta_1 x$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3 + \ldots$$

LR to Polynomial Regression

- Polynomial function
  - M is the order ..
  - **Where to stop? What is the best M?**



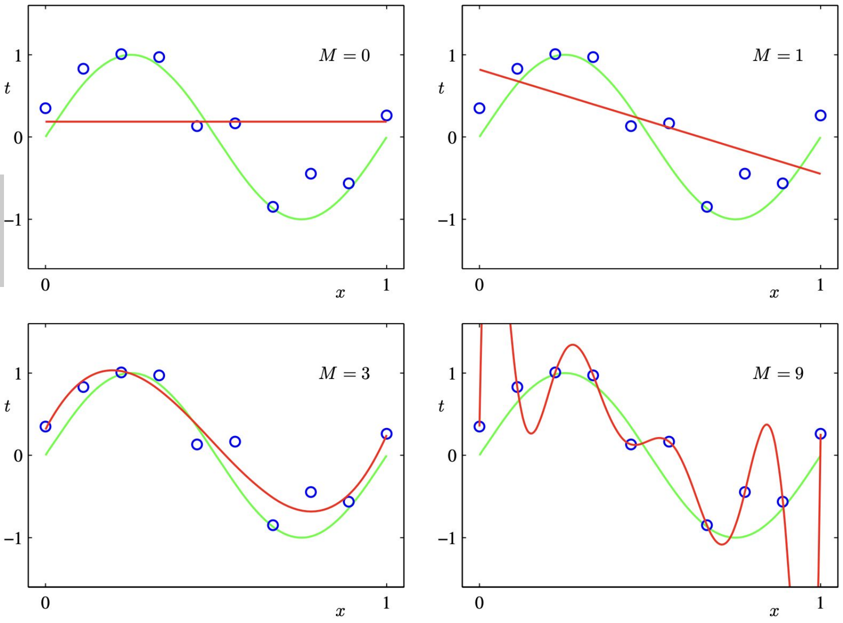
$$\hat{y} = \beta_0 + \beta_1 x$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3 + \ldots$$

LR to Polynomial Regression

- Polynomial function
  - M is the order ..
  - **Where to stop? What is the best M?**

Good news is our gradient descent (iterative learning) remains the same!

$$\hat{y} = \beta_0 + \beta_1 x$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3 + \ldots$$

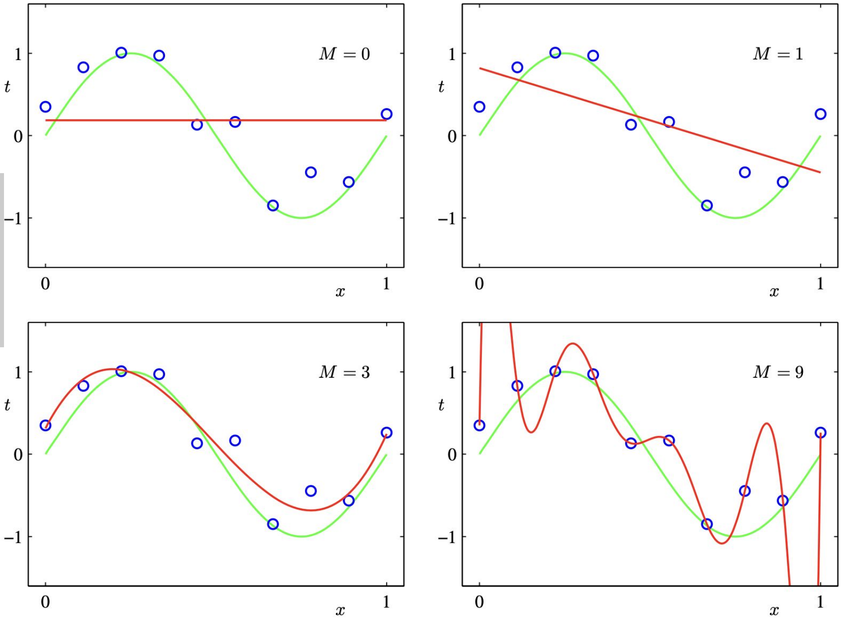


LR to Polynomial Regression

- Polynomial function
    - M is the order ..
    - **Where to stop? What is the best M?**

- Good news is our gradient descent (iterative learning) remains the same!
- You only need to change your objective function (from LR to Polynomial LR)

$$\hat{y} = \beta_0 + \beta_1 x$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3 + \ldots$$



LR to Polynomial Regression

Model

$$\hat{y} = \beta_0 + \beta_1 x$$

$$\Theta = \{\beta_0, \beta_1\}$$

$$\epsilon = |\hat{y} - y|$$

Optimization function

$$E_\Theta = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

$$\Theta^* = \operatorname{argmin}_\Theta E\{(x_i, y_i)\}_{i=1,\cdots,N}$$

$$\hat{y} = \beta_0 + \beta_1 x$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3 + \ldots$$

Our model got a little bigger: 2 params to M param

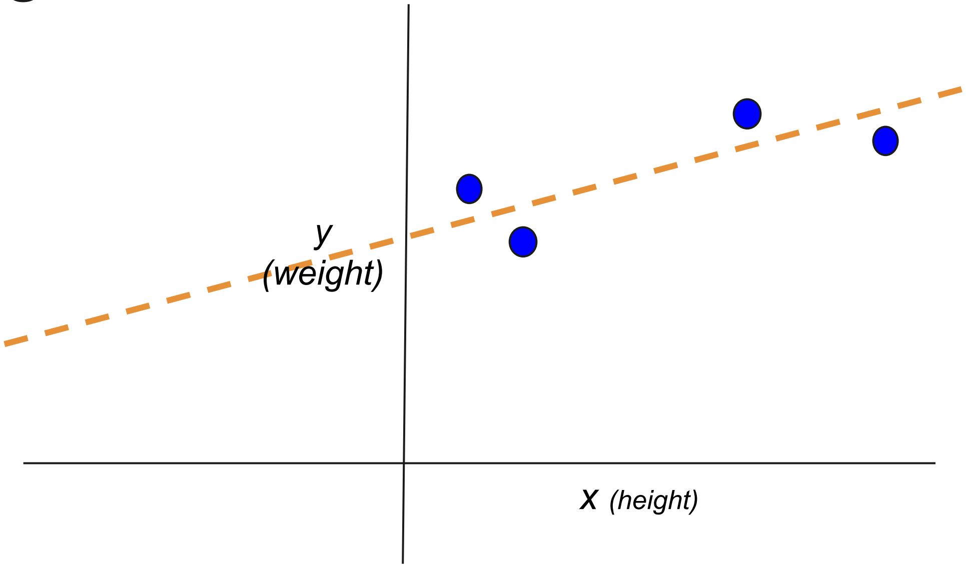
GPT

I know one of your tricks; get you soon!!

Our model today

Regularization

Regularization



So, essentially we are fitting a function; right?

Model

$$\hat{y} = \beta_0 + \beta_1 x$$

$$\Theta = \{\beta_0, \beta_1\}$$

Fitting Error

$$\epsilon = |\hat{y} - y|$$

Optimization function

$$E_\Theta = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

$$\Theta^* = \text{argmin}_\Theta\, E\{(x_i, y_i)\}_{i=1,\cdots,N}$$

Regularization

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2$$

Model

$$\hat{y} = \beta_0 + \beta_1 x$$

$$\Theta = \{\beta_0, \beta_1\}$$

$$\epsilon = |\hat{y} - y|$$

Optimization function

$$E_\Theta = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

$$\Theta^* = \mathrm{argmin}_\Theta\, E\{(x_i, y_i)\}_{i=1,\cdots,N}$$

Regularization

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

Essentially, the same formulation

Generally **ML** vs **Math** conventions

Model

$$\hat{y} = \beta_0 + \beta_1 x$$

$$\Theta = \{\beta_0, \beta_1\}$$

$$\epsilon = |\hat{y} - y|$$

Optimization function

$$E_\Theta = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

$$\Theta^* = \text{argmin}_\Theta\, E\{(x_i, y_i)\}_{i=1,\cdots,N}$$

Regularization

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

$$\boxed{\begin{array}{l} x : \text{ scalar} \\ \boldsymbol{x}, \mathbf{x}: \text{ vector} \\ \mathbf{X}: \text{ Matrix} \end{array}}$$

Model

$$\hat{y} = \beta_0 + \beta_1 x$$

$$\Theta = \{\beta_0, \beta_1\}$$

$$\epsilon = |\hat{y} - y|$$

Optimization function

$$E_\Theta = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

$$\Theta^* = \operatorname{argmin}_\Theta E\{(x_i, y_i)\}_{i=1,\cdots,N}$$

<u>Essentially, the same formulation</u>

Generally ML vs Math conventions

Regularization

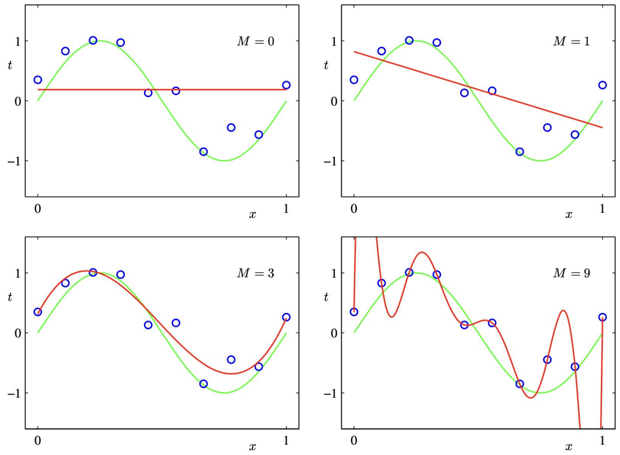


Table 1.1 Table of the coefficients $\mathbf{w}^\star$ for polynomials of various order. Observe how the typical magnitude of the coefficients increases dramatically as the order of the polynomial increases.

	$M = 0$	$M = 1$	$M = 6$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

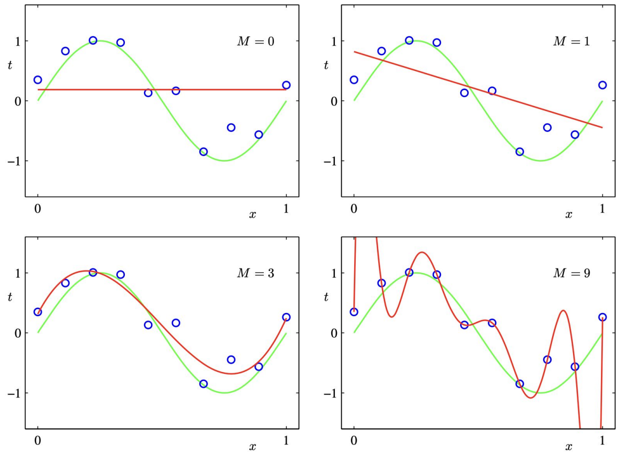
Regularization

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

Regularizer

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

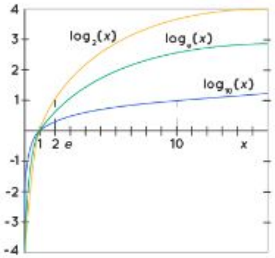
Regularization



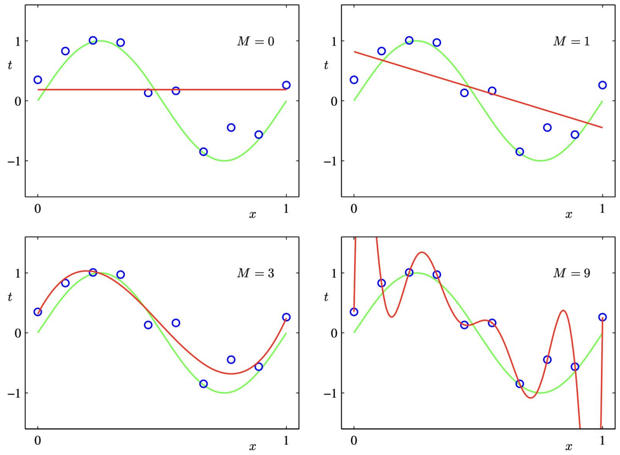
$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Table 1.2 Table of the coefficients $\mathbf{w}^\star$ for $M = 9$ polynomials with various values for the regularization parameter λ. Note that $\ln\lambda = -\infty$ corresponds to a model with no regularization, i.e., to the graph at the bottom right in Figure 1.4. We see that, as the value of λ increases, the typical magnitude of the coefficients gets smaller.

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01



Linear to Polynomial Regression + Regularization

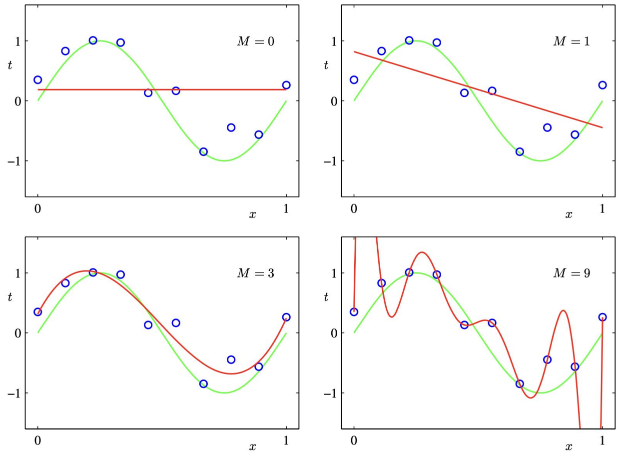


Learned function is **nonlinear**

$$\hat{y} = \beta_0$$
$$\hat{y} = \beta_0 + \beta_1 x$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3 + \ldots$$

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2$$

Linear to Polynomial Regression + Regularization



Learned function is **nonlinear**

$$\hat{y} = \beta_0$$
$$\hat{y} = \beta_0 + \beta_1 x$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3$$
$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_2 x^3 + \ldots$$

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

Model (still) **linear**

Notebook presentation

- Without regularizer
- With regularizer

Predictive modeling: Regression (diabetes)

Predictive modeling: Classification