Polynomial Regression

1. Given $\{(\underline{\underline{x_i}}, t_i)\}_{i=1}^{N}$

2. Polynomial features:

$$\boxed{\phi(x_i) = \begin{bmatrix} x_i^0 \\ x_i^1 \\ \vdots \\ x_i^M \end{bmatrix}}$$

, M = polynomial order

$$\boxed{\begin{aligned} y &= mx + b \\ &= m \cdot x^1 + \underline{\underline{b \cdot x^0}} \\ &= [x^0 \; x^1] \begin{bmatrix} b \\ m \end{bmatrix} \end{aligned}}$$

③ MAPPER:

Linear combination of features:

$$y = f(\phi(x), w) = w_0 x^0 + w_1 x^1 + \dots + w_M x^M$$

$$= [x^0 \;\; x^1 \dots x^M] \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix}$$

↑ want $y = t$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad \text{So,} \quad \boxed{y} = \underbrace{\begin{bmatrix} x_1^0 & x_1^1 & \dots & x_1^M \\ x_2^0 & x_2^1 & \dots & x_2^M \\ \vdots & \vdots & \ddots & \vdots \\ x_N^0 & x_N^1 & \dots & x_N^M \end{bmatrix}}_{} \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix} \boxed{= \mathbb{X} \cdot w}$$

N: # training samples

$N \gg M$

$\mathbb{X}$ = feature matrix, $N \times (M+1)$

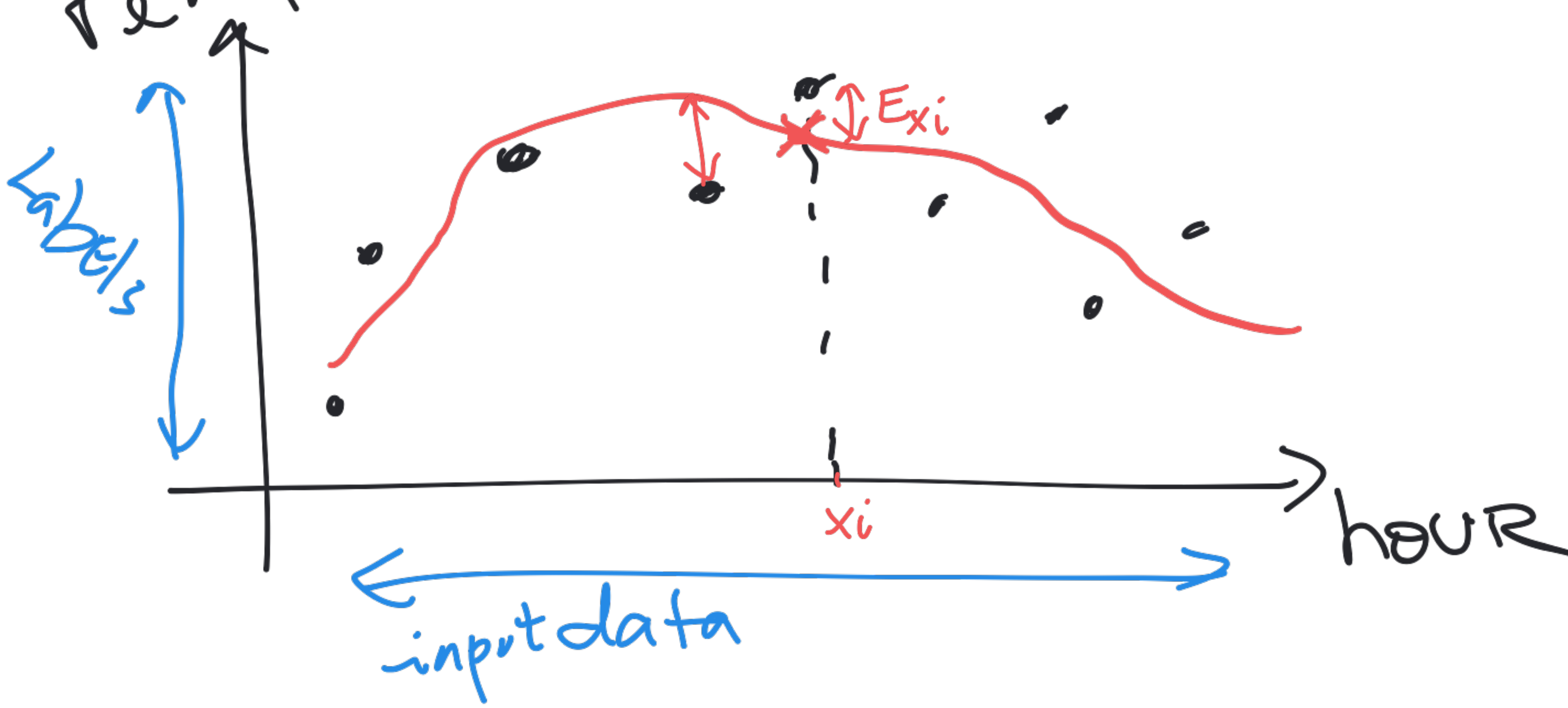
4 Cost function

$$\underline{X} w = t$$ ⟸ this is what we want

$$y = \underline{X} w$$

ERROR = E = t − y

$$= t - \underline{X} w$$



Cost function

$$\boxed{J(w) = \frac{1}{2} E^2}$$

Ridge cost function

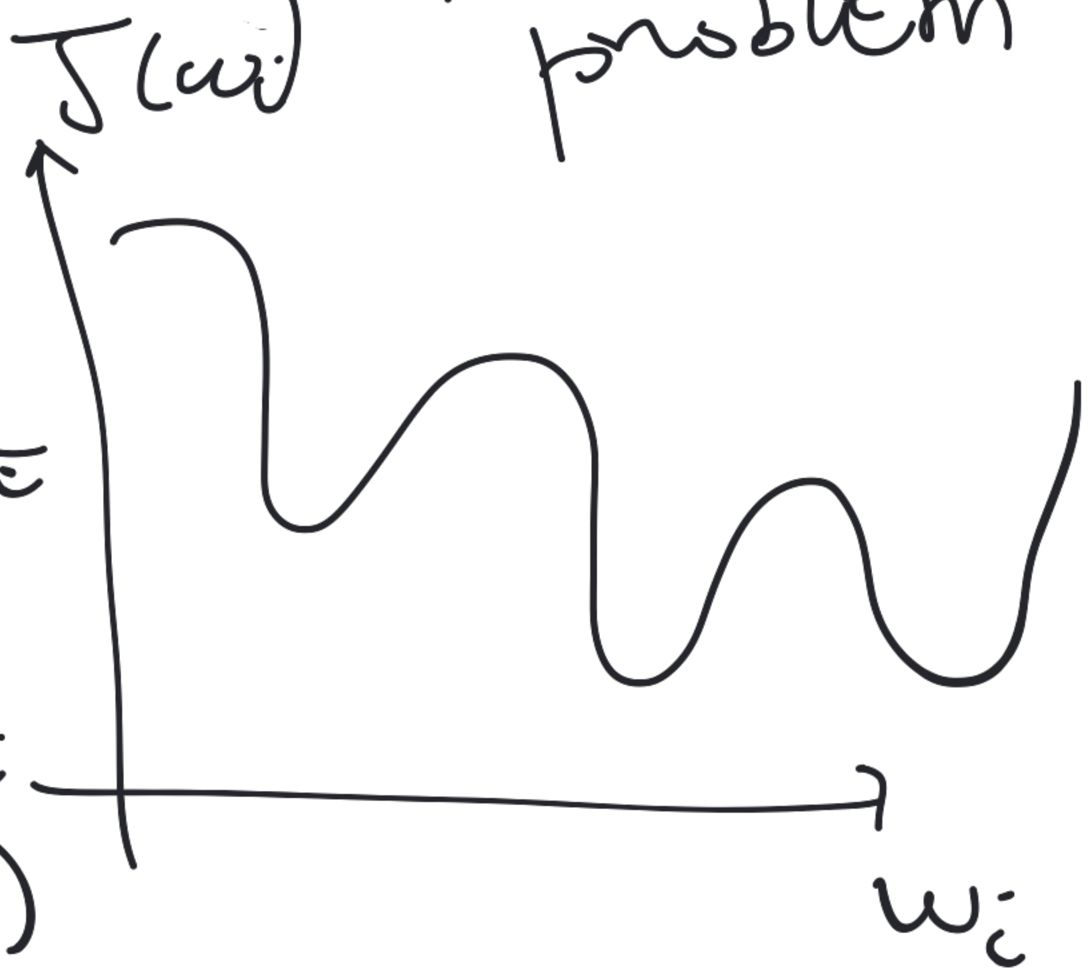
other

$$J(w) = |E|$$

5 Learning Algorithm

$$\arg\min_{w} J(\underline{\underline{w}})$$

optimization problem



$$\frac{\partial J(w)}{\partial w} = 0$$ and solve for w^* (optimal weight vector)

$$X w = t$$

I cannot do this: $w = \underline{\underline{X^{-1}}} \cdot t$ ✗

Instead, we will use pseudo-inverse: $X^{\dagger} = (X^T X)^{-1} \cdot X^T$

$$\boxed{w^* = X^{\dagger} \cdot t}$$

Pseudo-code:

input : training data $\{x_i\}_{i=1}^N$

labels $\{t_i\}_{i=1}^N$

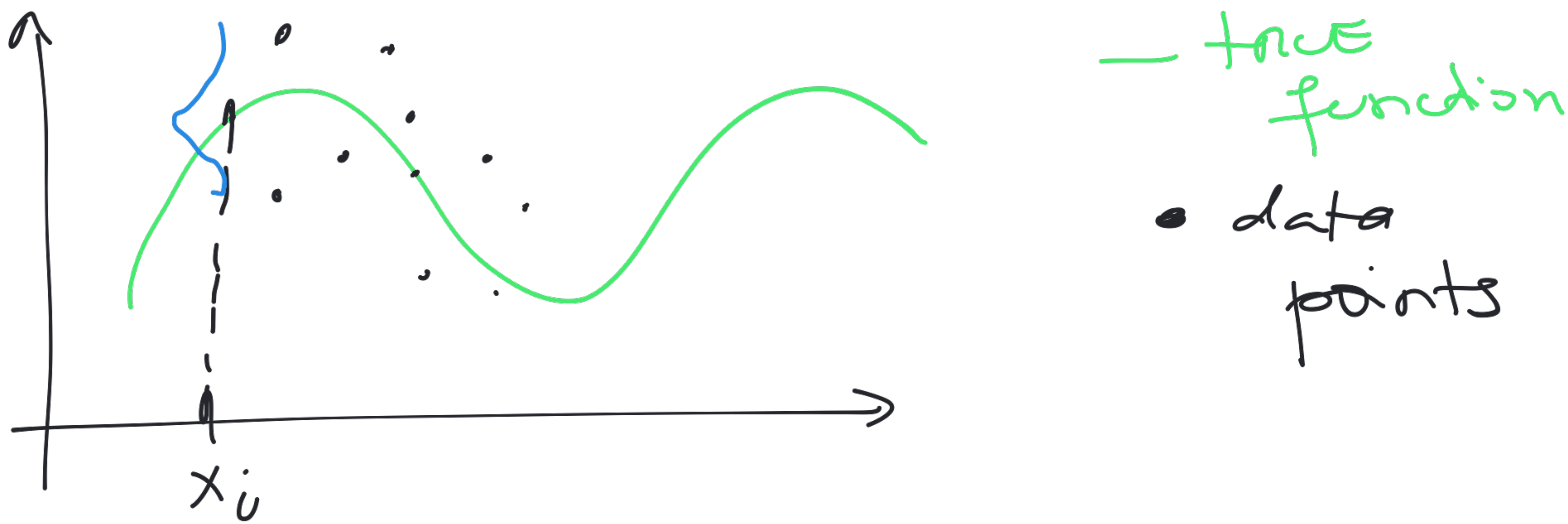
polynomial/model order M

① Compute features and store them in $\underline{X}$

② Compute optimal weights

$$w^* = \underline{X}^\dagger \cdot t$$

output : w^*


x_i
— true function
• data points