Module 3

Regression

Spring 2026

What is regression?

- We observe data: $(x_n, t_n), \quad n = 1, \ldots, N$
- Assume a model: $t_n = y(x_n, \mathbf{w}) + \epsilon_n$
- Where: $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$
- Goal: infer parameters $\mathbf{w}$

Not "draw a curve"
→ infer a *model that explains noisy measurements*

From model to probability

- ❑ Noise assumption gives a likelihood: $p(t_n \mid \mathbf{w}) = \mathcal{N}\big(t_n \mid y(x_n, \mathbf{w}), \sigma^2\big)$

From model to probability

- Noise assumption gives a likelihood: $p(t_n \mid \mathbf{w}) = \mathcal{N}(t_n \mid y(x_n, \mathbf{w}), \sigma^2)$

- For independent datapoints:

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid y(x_n, \mathbf{w}), \sigma^2)$$

From model to probability

- Noise assumption gives a likelihood: $p(t_n \mid \mathbf{w}) = \mathcal{N}(t_n \mid y(x_n, \mathbf{w}), \sigma^2)$

- For independent datapoints:

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid y(x_n, \mathbf{w}), \sigma^2)$$

- Take log:

$$\log p(\mathbf{t}|\mathbf{w}) = -\frac{1}{2\sigma^2} \sum_n \big(t_n - y(x_n, \mathbf{w})\big)^2 + \text{const}$$

Least squares emerges

- Maximize likelihood ⇔ minimize:

$$\chi^2(\mathbf{w}) = \sum_n \big(t_n - y(x_n, \mathbf{w})\big)^2$$

Remember from last module!

- Least squares = MLE for Gaussian noise

- Important: fitting depends on noise assumptions

Linear-in-parameters model

- ‘Linear Regression model’ -> model ***parameters*** are linear

$$y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j \phi_j(x)$$

Linear-in-parameters model

- ‘Linear Regression model’ -> model ***parameters*** are linear

$$y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j \phi_j(x)$$

Basis functions are not necessarily linear – e.g. polynomial, Fourier

Linear-in-parameters model

- ‘Linear Regression model’ -> model ***parameters*** are linear

$$y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j \phi_j(x)$$

Basis functions are not necessarily linear – e.g. polynomial, Fourier

- Define design matrix

$$\Phi_{nj} = \phi_j(x_n)$$

Linear-in-parameters model

- 'Linear Regression model' -> model ***parameters*** are linear

$$y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j \phi_j(x)$$

Basis functions are not necessarily linear – e.g. polynomial, Fourier

- Define design matrix

$$\Phi_{nj} = \phi_j(x_n)$$

- Model becomes:

$$t_n = y(x_n, \mathbf{w}) + \epsilon_n \longrightarrow \mathbf{t} = \Phi\mathbf{w} + \boldsymbol{\epsilon}$$

Deriving the normal equations

- Goal - Minimize: $E(\mathbf{w}) = ||\mathbf{t} - \Phi\mathbf{w}||^2$

Deriving the normal equations

- Goal - Minimize: $$E(\mathbf{w}) = ||\mathbf{t} - \Phi\mathbf{w}||^2$$
- Differentiate and solve $$\frac{\partial E}{\partial \mathbf{w}} = -2\Phi^T(\mathbf{t} - \Phi\mathbf{w}) = 0$$

Deriving the normal equations

- ❑Goal - Minimize: $$E(\mathbf{w}) = ||\mathbf{t} - \Phi\mathbf{w}||^2$$

- ❑Differentiate and solve $$\frac{\partial E}{\partial \mathbf{w}} = -2\Phi^T(\mathbf{t} - \Phi\mathbf{w}) = 0$$

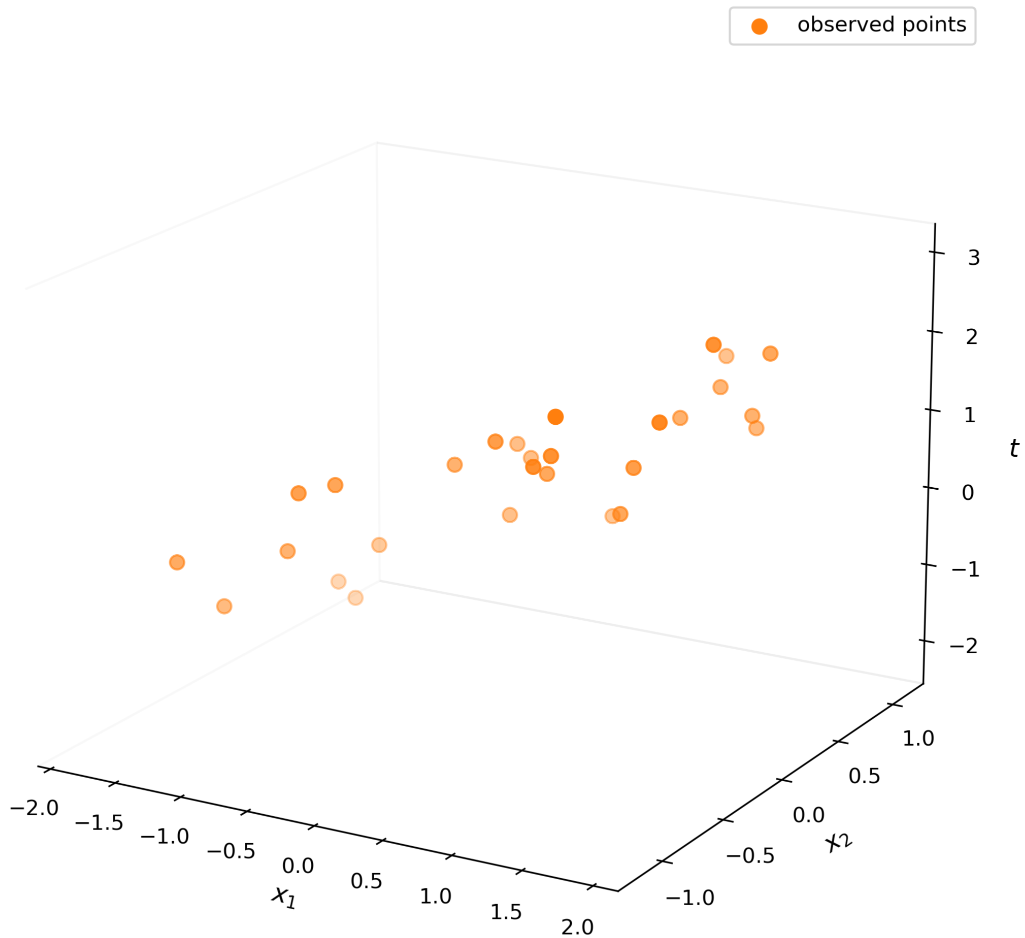
$$\Phi^T\Phi\mathbf{w} = \Phi^T\mathbf{t}$$

$$\boxed{\mathbf{w}_{ML} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{t}}$$

Note: use linear solver for last step – inverting the matrix will get computationally expensive

Geometric meaning

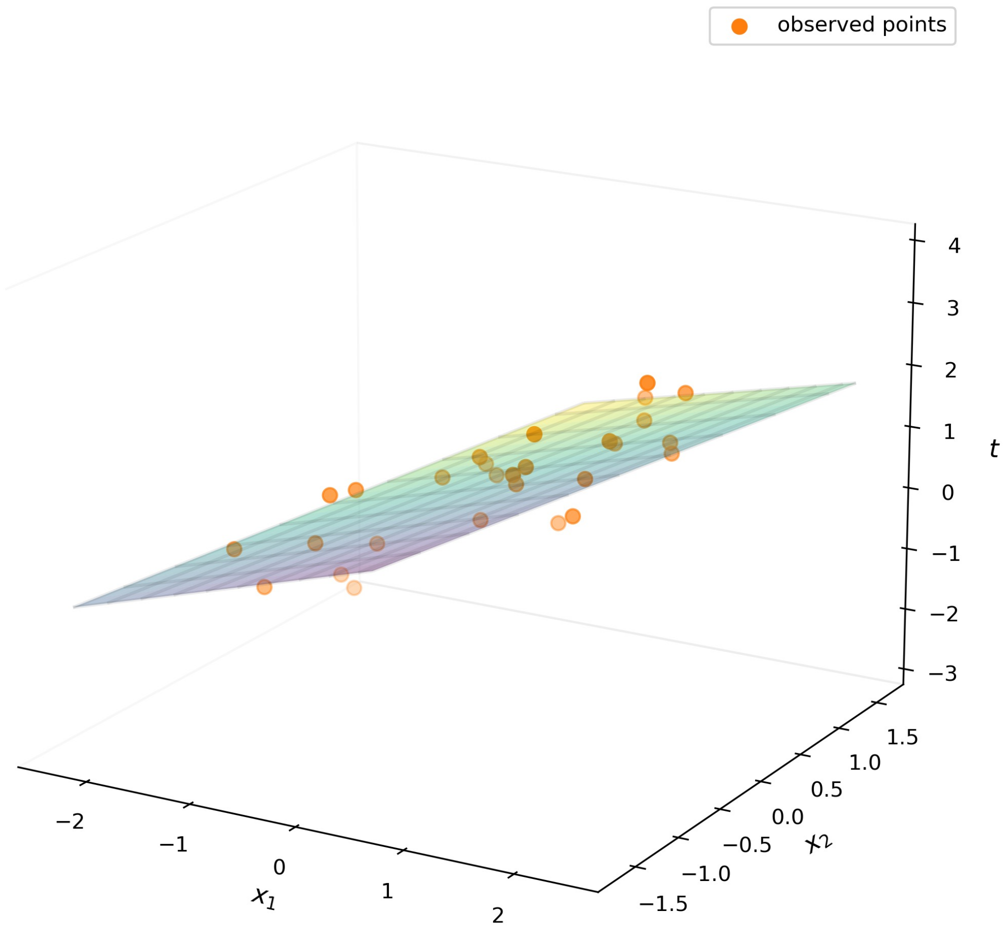
- Measurements of data do not lie in perfect plane due to noise



Geometric meaning

- ❑ Measurements of data do not lie in perfect plane due to noise

- ❑ Hyperplane represents best prediction of output given our input data



Geometric meaning

- Measurements of data do not lie in perfect plane due to noise
- Hyperplane represents best prediction of output given our input data
- Does this by minimizing distance of projections onto hyperplane