Week 1 Note

Linear regression

- **Linear regression**: find linear function with small discrepancy(差异)

Problem setup

- **Dataset** D: n input/output pairs(Experience(E))

$$D = \{(x^1, y^1), (x^2, y^2), \dots, (x^n, y^n)\}$$

- $x^i \in \mathbb{R}^d$ is the "**input**" for the i^{th} data point as a feature vector with d elements
- $y^i \in \mathbb{R}$ is the "**output**" for the i^{th} data point
- **Regression task**(T): **find a model** such that the predicted output $f(x)$ is close to the true output y
- **Linear Model**: a linear regression model has the form

$$f(x) = w_0 + w_1x_1 + w_2x_2 + \dots + w_dx_d = (w_0 + w_1 + \dots + w_d)\begin{pmatrix}1\\ \vec{x}\end{pmatrix} = \vec{w}^T\bar{x}$$

- **bias**(intercept): w_0
- **weight parameters**: $w_1, w_2, \dots, w_d$
- **feature**: x_i is the i^{th} component of $x \in \mathbb{R}^d$
- **Cost function**:

$$C(\vec{w}) = \frac{1}{2n}\sum_{i=1}^{n}(y^i - \vec{x}^{i^T}\vec{w}) = \frac{1}{2n}(\vec{w}^TX^TX\vec{w} - 2\vec{w}^TX^Ty + \vec{y}^T\vec{y})$$

> $X = \begin{pmatrix} \vec{x}^{1^T} \\ ... \\ \vec{x}^{n^T} \end{pmatrix} \in \mathbb{R}^{n \times d}$

> $\vec{y} = \begin{pmatrix} \vec{y}^1 \\ ... \\ \vec{y}^n \end{pmatrix} \in \mathbb{R}^n$

> note: $(X\vec{w})^T = \vec{w}^T X^T$

- **Optimal** w^*:

$$\vec{w}^* = \frac{\sum_{i=1}^n y^i x^i}{\sum_{i=1}^n x^{i^2}} = (X^T X)^{-1} X^T \vec{y}$$

Summary: Linear Regression

- Linear regression(or least square regression)
- model linear relationship between input and output(**task T**)
- Example points(**experience E**)
- mean square error as loss function(**performance P**)
- closed-form solution(or exact solution)

Polynomial regression

- Polynomial regression model:

$$f(x) = w_0 + w_1 x + w_2 (x)^2 + ... + w_M (x)^M = \vec{w}^T \phi(x) = \phi(x)^T \vec{w} = \bar{X}\vec{w}$$

> where $(x)^i$ denotes i^{th} power of x

> Define the **feature map**: $\phi(x) = \begin{pmatrix} 1 \\ x \\ (x)^2 \\ ... \\ (x)^M \end{pmatrix}$

> $$X = \begin{pmatrix} \vec{x}^{1^T} \\ ... \\ \vec{x}^{n^T} \end{pmatrix} \mapsto \begin{pmatrix} \phi(x^1)^T \\ \phi(x^2)^T \\ ... \\ \phi(x^n)^T \end{pmatrix} = \begin{pmatrix} 1 & x^1 & (x^1)^2 & ... & (x^1)^M \\ 1 & x^2 & (x^2)^2 & ... & (x^2)^M \\ ... & ... & ... & ... & ... \\ 1 & x^n & (x^n)^2 & ... & (x^n)^M \end{pmatrix} = \bar{X}$$

- **Cost function**:

$$C(\vec{w}) == \underbrace{\frac{1}{2n}(\vec{w}^T X^T X \vec{w} - 2\vec{w}^T X^T y + \vec{y}^T \vec{y})}_{\text{fitting to data}} + \underbrace{\frac{\lambda}{2}||w||_2^2}_{\text{regulariser}}$$

- The optimal weights can be found as:

$$\vec{w}^* = (\bar{X}^T \bar{X})^{-1} \bar{X}^T \vec{y} = (\frac{1}{n}(X^T X) + \lambda I)^-1(\frac{1}{n} X^T \vec{y})$$

> where $I \in \mathbb{R}^{n \times n}$ is the identity matrix

- If $\lambda = 0$, then this becomes the solution of the least squares regression problem.
- If $\lambda = \infty$, we get $w^* = 0$, which is a trivial solution. We need to choose an appropriate λ

Summary: Polynomial Regression

- Polynomial regression
- Polynomial fitting
- Feature mapping
- Underfitting
- Overfitting
- Regularisation