Linear Model for Regression

CSCC11 – Topic 01

Computer & Mathematical Sciences
UNIVERSITY OF TORONTO
SCARBOROUGH

Topics

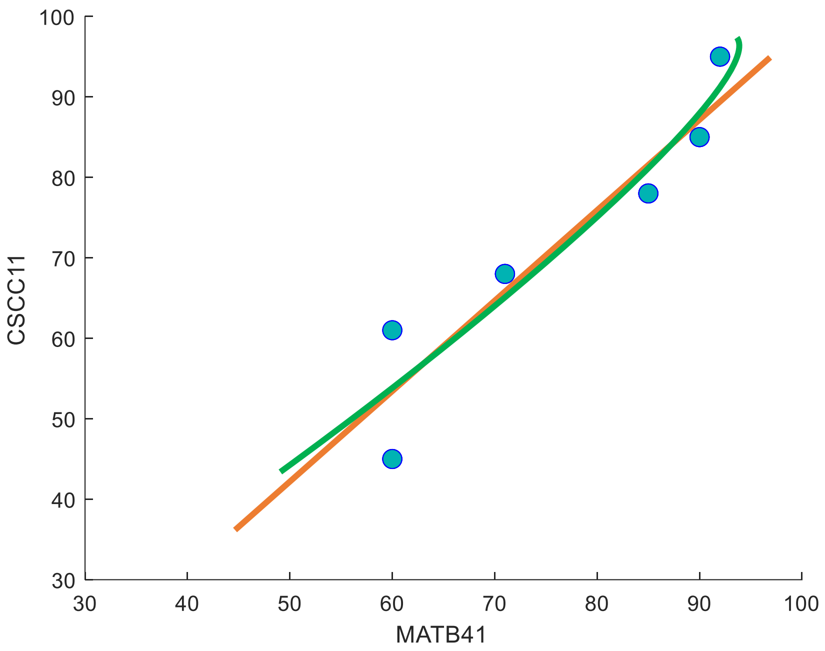
- Linear Regression
  - Univariate Linear Regression
  - Multiple Linear Regression
  - Multivariate Linear Regression
- Basis Function Regression
  - Polynomials
  - Radial Basis Function (RBF)
- Regularization
  - Conditioning of a matrix
  - Overfitting and underfitting
  - Bias and Variance Tradeoff
- KNN Regression

Linear Regression

Problem 1: $f(x)$: $\mathbb{R} \rightarrow \mathbb{R}$

- Predict C11 mark

Example	MATB41 x	C11 y
1	90	85
2	71	68
3	92	95
4	60	61
5	85	78
6	60	45
	65	???



- What function (i.e. hypothesis) fits the data well?
- How to measure the quality of the fit?

Notations and Terminologies

- $x \in \mathbb{R}$: input, feature, independent variable, regressor,
- $y \in \mathbb{R}$: output, target, dependent variable, response
- $\hat{y} \equiv f(x) \in \mathbb{R}$: predicted output given x
- $\{(x_i, y_i)\}_{i=1}^{N}$: training data
  - Training data index: i
  - Number of examples: N
- Input vector $\vec{x} \equiv \mathbf{x} \equiv [x_1, x_2 \dots, x_N]^T$
- Output vector $\vec{y} \equiv \mathbf{y} \equiv [y_1, y_2 \dots, y_N]^T$
- Augmented input record $\tilde{\mathbf{x}} = [1 \;\; x]^{\boldsymbol{T}}$
- Augmented input matrix $\widetilde{\mathbf{X}} = [\mathbf{1} \;\; \mathbf{x}]$

$$\widetilde{\mathbf{X}} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ & \vdots \\ 1 & x_N \end{bmatrix} \qquad \tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ x \end{bmatrix} \qquad \tilde{\mathbf{x}}_{\boldsymbol{i}} = \begin{bmatrix} 1 \\ x_i \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \qquad \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{bmatrix}$$

$f\colon \mathbb{R} \to \mathbb{R}$ Model

Example	MATB41 x	C11 y
1	90	85
2	71	68
…	…	…
N	60	45

$$\hat{y} \equiv f(x) = wx + b$$
$$= b + wx$$

$$\hat{y} = f(\tilde{\mathbf{x}}) = [b, w] \begin{bmatrix} 1 \\ x \end{bmatrix} = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}$$

$$\hat{y} = f(\tilde{\mathbf{x}}) = [1, x] \begin{bmatrix} b \\ w \end{bmatrix} = \tilde{\mathbf{x}}^T \tilde{\mathbf{w}}$$

$$\hat{y}_i = f(\tilde{\mathbf{x}}_i) = \tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}}$$

- Each row of the matrix $\tilde{\mathbf{X}}$ is an augmented input record

$$\tilde{\mathbf{w}} = \begin{bmatrix} b \\ w \end{bmatrix} \qquad \tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

Per sample

$$\hat{y} = f(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}^T \tilde{\mathbf{w}}$$

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix} = \begin{bmatrix} \text{---} & \tilde{\mathbf{x}}_1^T & \text{---} \\ \text{---} & \tilde{\mathbf{x}}_2^T & \text{---} \\ \vdots & \vdots & \vdots \\ \text{---} & \tilde{\mathbf{x}}_N^T & \text{---} \end{bmatrix}$$

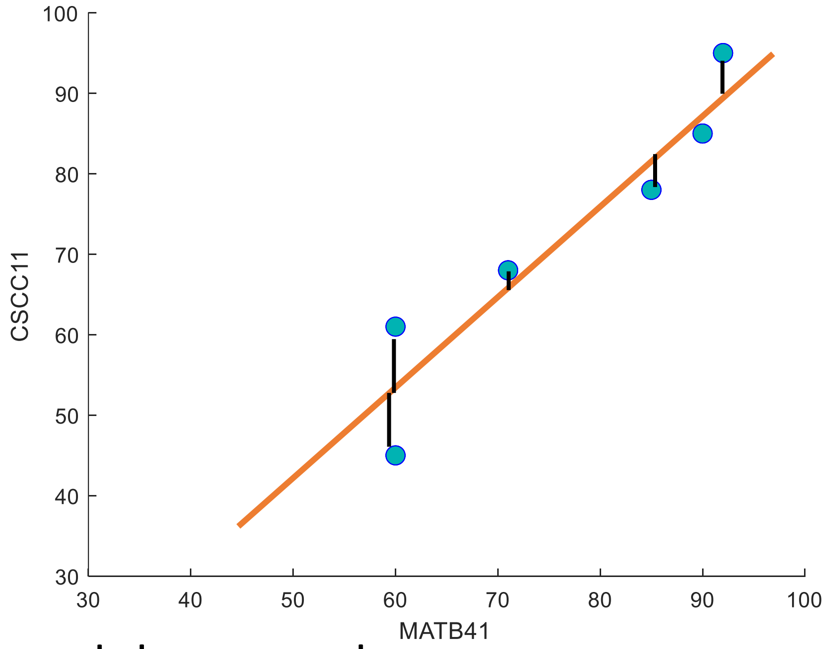
$$\hat{\mathbf{y}} = f(\tilde{\mathbf{X}}) = \tilde{\mathbf{X}} \tilde{\mathbf{w}}$$

Entire Data Set

$f: \mathbb{R} \to \mathbb{R}$ Loss Function

$$\hat{y} = f(x): \mathbb{R} \to \mathbb{R}$$

$$\hat{y} = wx + b$$



- **Residual**
  measures the difference between the predicted value and the true value

$$e_i = y_i - (wx_i + b) = y_i - \hat{y}_i$$

- **Loss Function**
  measures the distance between the predicted value and the true value

$$\mathcal{L}\big(y_i, f(x_i)\big) = e_i^2 = (y_i - (wx_i + b))^2 = (y_i - \hat{y}_i)^2$$

$f\colon \mathbb{R} \to \mathbb{R}$ Cost Function

$\|\mathbf{v}\|_2^2 = \mathbf{v}^T\mathbf{v} = \sum_i v_i^2$

Model

- predicts the output
- x is the unknown
- w, b are given

$$\hat{y} = f(x) = wx + b = \tilde{\mathbf{x}}^T\tilde{\mathbf{w}}$$

$$\hat{\mathbf{y}} = f(\tilde{\mathbf{X}}) = \tilde{\mathbf{X}}\tilde{\mathbf{w}}$$

Cost Function (Objective Function)

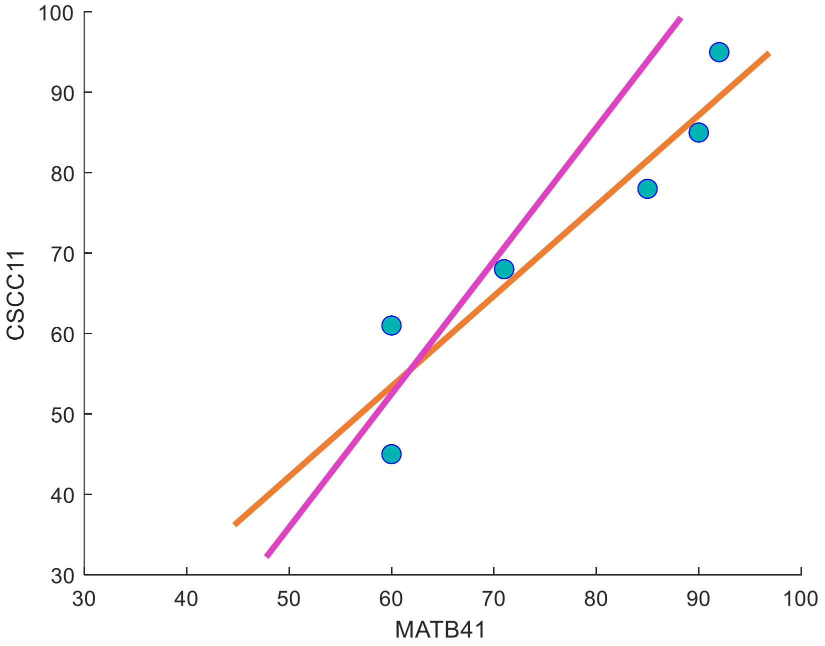
- measures errors over all training data
- w, b are the unknowns
- $\{(x_i, y_i)\}_{i=1}^N$ are given

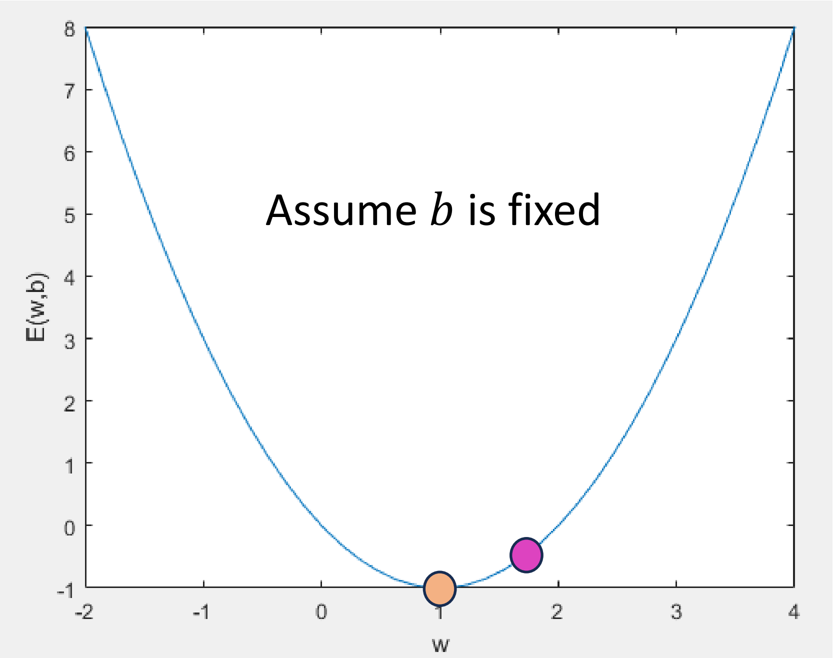
$$\begin{aligned} E(\tilde{\mathbf{w}}) &= E(w, b) \\ &= \sum_{i=1}^N e_i^2 = \sum_{i=1}^N (y_i - (wx_i + b))^2 \\ &= (\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}}) \\ &= \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 \\ &= \left\|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\right\|_2^2 \end{aligned}$$

Model vs Cost Function

$$f(x) = wx + b$$

$$E(w,b) = \sum_{i=1}^{N} (y_i - (wx_i + b))^2$$

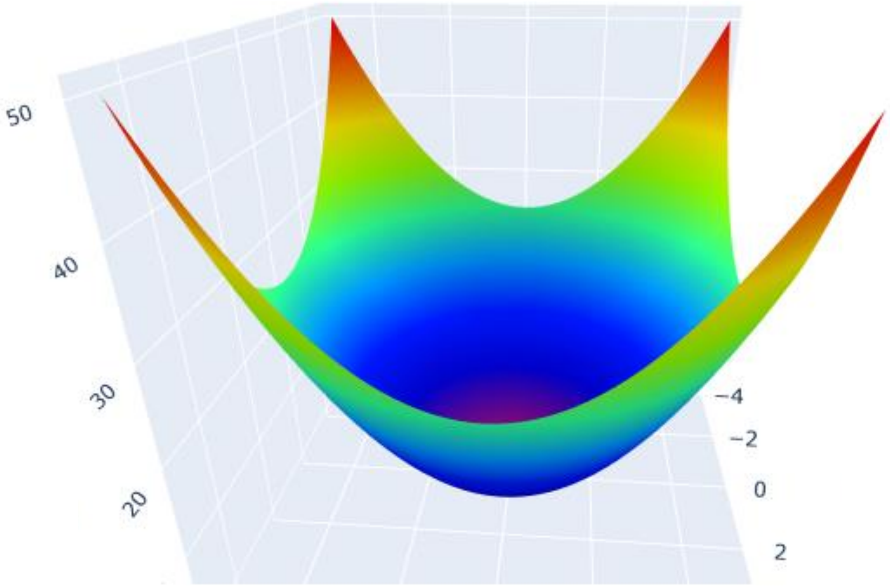


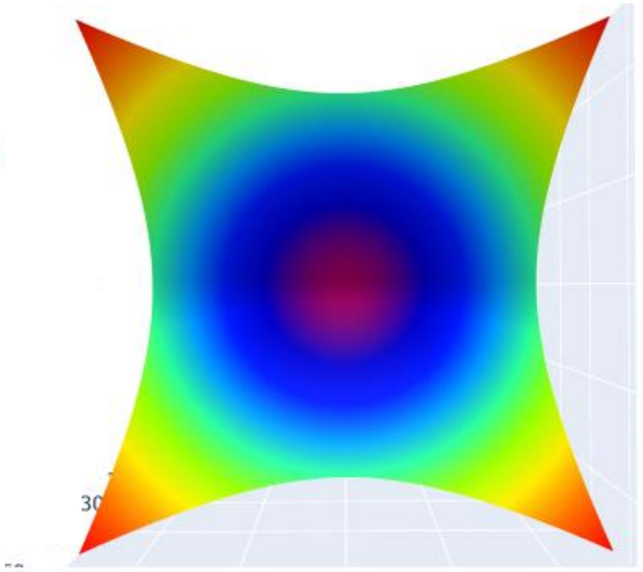


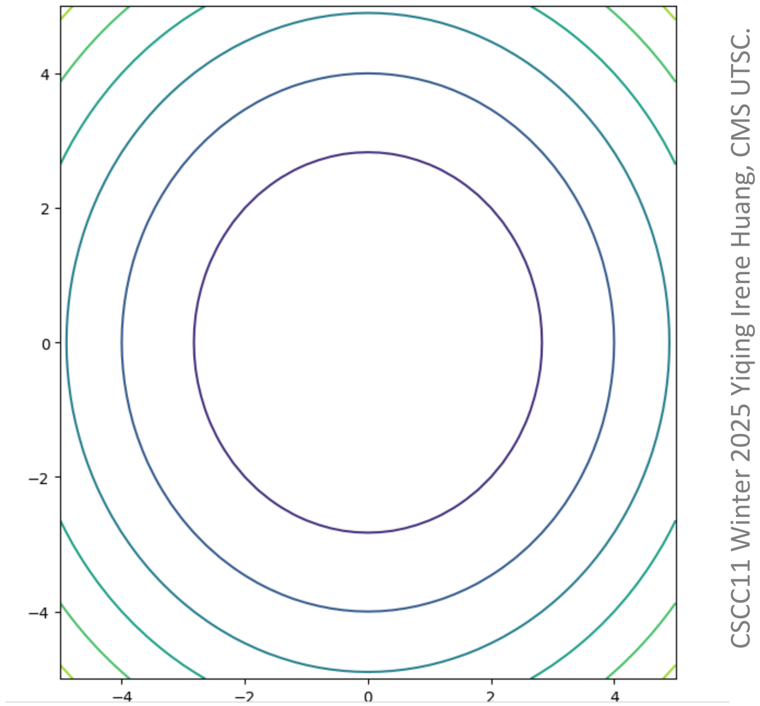
$$f(\tilde{\mathbf{X}}) = \tilde{\mathbf{X}}\tilde{\mathbf{w}}$$

$$E(\tilde{\mathbf{w}}) = \left\|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\right\|_2^2$$

Cost Function in 2-D







Note: Check the python_output.ipynb for the 3D plot of a two-dimensional quadratic function.

Solution of Weights

- Let $\frac{\partial E}{\partial b} = 0$, we get

$$b^* = \bar{y} - w\bar{x}, \quad \text{where } \bar{y} = \frac{\sum_i y_i}{N}, \quad \bar{x} = \frac{\sum_i x_i}{N}$$

- Let $\frac{\partial E}{\partial w} = 0$ with b set to b^*, we obtain

$$w^* = \frac{\sum_i (y_{i-}\bar{y})(x_i - \bar{x})}{\sum_i (x_i - \bar{x})^2}$$

Note: The detailed algebraic derivation will be done using the board

Vectorized Solution of optimal weights

Pseudoinverse of $\mathbf{A}$

$$\mathbf{A}^{+} = (\mathbf{A}^{\mathbf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathbf{T}}$$

- Find the weights to minimize the cost function

$$\tilde{\mathbf{w}}^{*} = \underset{\tilde{\mathbf{w}}}{\operatorname{argmin}} \left\|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\right\|_2^2$$

$$E(\tilde{\mathbf{w}}) = \left\|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\right\|_2^2$$

$$= (\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}})^T(\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}})$$

$$= \underbrace{\mathbf{y}^T\mathbf{y}}_{\text{constant}} - \underbrace{2\tilde{\mathbf{w}}^T\tilde{\mathbf{X}}^T\mathbf{y}}_{\text{Linear Term}} + \underbrace{\mathbf{w}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\tilde{\mathbf{w}}}_{\text{Quadratic Term}}$$

Note: $\tilde{\mathbf{w}}^T\tilde{\mathbf{X}}^T\mathbf{y} = (\tilde{\mathbf{w}}^T\tilde{\mathbf{X}}^T\mathbf{y})^T = \mathbf{y}^T\tilde{\mathbf{X}}\tilde{\mathbf{w}}$

$$\frac{\partial E}{\partial \tilde{\mathbf{w}}} = -2\tilde{\mathbf{X}}^T\mathbf{y} + 2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\tilde{\mathbf{w}} = \mathbf{0}$$

$$\Downarrow$$

$$\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\tilde{\mathbf{w}} = \tilde{\mathbf{X}}^T\mathbf{y}$$

$$\Downarrow$$

$$\tilde{\mathbf{w}}^{*} = (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^T\mathbf{y} = \tilde{\mathbf{X}}^{+}\mathbf{y}$$

- Assume $(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})$ is non-singular

Problem 2: $f(x)$: $\mathbb{R}^D \rightarrow \mathbb{R}$

- Predict C11 mark

$$\overbrace{\hspace{2em}}^{D\ =\ 4}$$

Example	MATB41 x_1	MATB24 x_2	STAB52 x_3	CGPA x_4	C11 y
1	90	87	82	3.6	85
2	71	67	85	3.0	68
3	92	96	93	3.8	95
4	60	62	71	2.8	61
5	85	81	74	3.1	78
6	60	61	60	2.7	45

7	65	73	82	3.1	???

$f: \mathbb{R}^D \rightarrow \mathbb{R}$ Model

Example	MATB41 x	…	C11 y
1	90	…	85
2	71	…	68
…	…	…	…
N	60	…	45

$$\hat{y} \equiv f(x) = b + w_1 x_1 + \cdots + w_D x_D$$

$$\hat{y} = f(\tilde{\mathbf{x}}) = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}$$

$$\hat{y} = f(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}^T \tilde{\mathbf{w}}$$

$$\hat{y}_i = f(\tilde{\mathbf{x}}_i) = \tilde{\mathbf{x}}_i^T \tilde{\mathbf{w}}$$

$$\tilde{\mathbf{x}}_{\boldsymbol{i}} = \begin{bmatrix} 1 \\ (x_i)_2 \\ \vdots \\ (x_i)_D \end{bmatrix} = \begin{bmatrix} 1 \\ x_{2i} \\ \vdots \\ x_{Di} \end{bmatrix}$$

$$\tilde{\mathbf{w}} = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_D \end{bmatrix} \quad \tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_D \end{bmatrix}$$

Per sample

$$\hat{y} = f(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}^T \tilde{\mathbf{w}}$$

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1D} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{N1} & \cdots & x_{ND} \end{bmatrix}$$

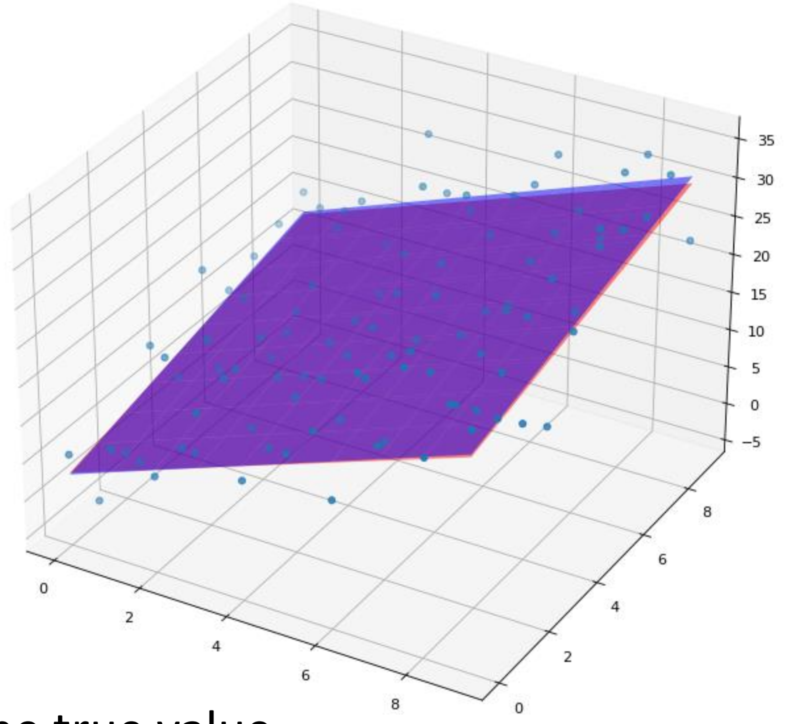
$$\hat{\mathbf{y}} = f(\tilde{\mathbf{X}}) = \tilde{\mathbf{X}}\tilde{\mathbf{w}}$$

Entire Data Set

$f: \mathbb{R}^D \to \mathbb{R}$ Loss Function

$$\hat{y} = f(x): \mathbb{R}^D \to \mathbb{R}$$

$$\hat{y} = b + w_1 x_1 + \cdots + w_D x_D$$



- Residual
  measures the difference between the predicted value and the true value

$$e_i = y_i - (b + w_1 x_{i1} + \cdots + w_D x_{iD}) = y_i - \hat{y}_i$$

- Loss Function
  measures the distance between the predicted value and the true value

$$\mathcal{L}\big(y_i, f(\mathbf{x}_i)\big) = e_i^2 = (y_i - (b + w_1 x_{i1} + \cdots + w_D x_{iD}))^2 = (y_i - \hat{y}_i)^2$$

$f: \mathbb{R}^D \rightarrow \mathbb{R}$ Cost Function

$\|\mathbf{v}\|_2^2 = \mathbf{v}^T\mathbf{v} = \sum_i v_i^2$

Model

- predicts the output
- x is the unknown
- w_j, b are given

Cost Function (Objective Function)

- measures errors over all training data
- w_j, b are the unknowns
- $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ are given

$$\hat{y} = f(\mathbf{x}) = b + \sum_{j=1}^D w_j x_{ij}$$

$$= \tilde{\mathbf{x}}^T \tilde{\mathbf{w}}$$

$$\hat{\mathbf{y}} = f(\tilde{\mathbf{X}}) = \tilde{\mathbf{X}}\tilde{\mathbf{w}}$$

Note that $\mathbf{x}_i$ is multi-dimensional

$$E(w_j, b) = \sum_{i=1}^N e_i^2 = \sum_{i=1}^N \left(y_i - (b + \sum_{j=1}^D w_j x_{ij})\right)^2$$

$$= (\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}})$$

$$= \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$$

$$= \left\|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\right\|_2^2$$

Solution of optimal weights

Pseudoinverse of $\mathbf{A}$

$$\mathbf{A}^{+} = (\mathbf{A}^{\mathbf{T}}\mathbf{A})^{-\mathbf{1}}\mathbf{A}^{\mathbf{T}}$$

- Find the weights to minimize the cost function

$$\tilde{\mathbf{w}}^{*} = \underset{\tilde{\mathbf{w}}}{\operatorname{argmin}} \left\|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\right\|_2^2$$

$$E(\tilde{\mathbf{w}}) = \left\|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\right\|_2^2$$

$$= (\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}})^T(\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}})$$

$$= \underbrace{\mathbf{y}^T\mathbf{y}}_{\text{constant}} - \underbrace{2\tilde{\mathbf{w}}^T\tilde{\mathbf{X}}^T\mathbf{y}}_{\text{Linear Term}} + \underbrace{\mathbf{w}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\tilde{\mathbf{w}}}_{\text{Quadratic Term}}$$

Note: $\tilde{\mathbf{w}}^T\tilde{\mathbf{X}}^T\mathbf{y} = (\tilde{\mathbf{w}}^T\tilde{\mathbf{X}}^T\mathbf{y})^T = \mathbf{y}^T\tilde{\mathbf{X}}\tilde{\mathbf{w}}$

$$\frac{\partial E}{\partial \tilde{\mathbf{w}}} = -\mathbf{2}\tilde{\mathbf{X}}^T\mathbf{y} + 2\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\tilde{\mathbf{w}} = \mathbf{0}$$

$$\Downarrow$$

$$\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\tilde{\mathbf{w}} = \tilde{\mathbf{X}}^T\mathbf{y}$$

$$\Downarrow$$

$$\tilde{\mathbf{w}}^{*} = (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^T\mathbf{y} = \tilde{\mathbf{X}}^{+}\mathbf{y}$$

- Assume $(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})$ is non-singular

Multiple Regression Summary

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} \quad \mathbf{x}_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{Di} \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_N \end{bmatrix}$$

$$\tilde{\mathbf{w}} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \quad \tilde{\mathbf{x}} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \quad \tilde{\mathbf{x}}_i = \begin{bmatrix} 1 \\ \mathbf{x}_i \end{bmatrix} \quad \tilde{\mathbf{X}} = \begin{bmatrix} 1 & \mathbf{x}_1^T \\ 1 & \vdots \\ 1 & \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1D} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{N1} & \cdots & x_{ND} \end{bmatrix}$$

$$\hat{y} = f(\tilde{\mathbf{x}}) = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}} = \tilde{\mathbf{x}}^T \tilde{\mathbf{w}}$$

$$\hat{\mathbf{y}} = f(\tilde{\mathbf{X}}) = \tilde{\mathbf{X}} \tilde{\mathbf{w}}$$

$$E(\tilde{\mathbf{w}}) = \left\| \mathbf{y} - \tilde{\mathbf{X}} \tilde{\mathbf{w}} \right\|_2^2$$

$$\tilde{\mathbf{w}}^* = \underset{\tilde{\mathbf{w}}}{\operatorname{argmin}} \left\| \mathbf{y} - \tilde{\mathbf{X}} \tilde{\mathbf{w}} \right\|_2^2 = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{y} = \tilde{\mathbf{X}}^+ \mathbf{y}$$

Problem 3: $f(x)$: $\mathbb{R}^D \to \mathbb{R}^K$

- Predict C11 marks

$D = 4$ $K = 2$

Example	MATB41 x_1	MATB24 x_2	STAB52 x_3	CGPA x_4	C11 y_1	Final y_2
1	90	87	82	3.6	85	80
2	71	67	85	3.0	68	70
3	92	96	93	3.8	95	92
4	60	62	71	2.8	61	56
5	85	81	74	3.1	78	72
6	60	61	60	2.7	45	40
7	65	73	82	3.1	???	???

Matrix Multiplication

$$AB = A \begin{bmatrix} | & \cdots & | \\ b_1 & \cdots & b_K \\ | & \cdots & | \end{bmatrix} = \begin{bmatrix} | & \cdots & | \\ Ab_1 & \cdots & Ab_K \\ | & \cdots & | \end{bmatrix}$$

$A \in \mathbb{R}^{N \times p}$ $\qquad$ $B \in \mathbb{R}^{p \times K}$

Multivariate Regression Matrix Form

$$\tilde{\mathbf{w}}_j = \begin{bmatrix} b_j \\ w_{1j} \\ \vdots \\ w_{Dj} \end{bmatrix}$$

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1D} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{N1} & \cdots & x_{ND} \end{bmatrix}_{N\times(D+1)} \qquad \tilde{\mathbf{W}} = \begin{bmatrix} b_1 & \cdots & b_K \\ w_{11} & \cdots & w_{1K} \\ \vdots & \vdots & \vdots \\ w_{D1} & \cdots & w_{DK} \end{bmatrix}_{(D+1)\times K} \qquad \mathbf{y}'_j = \begin{bmatrix} y_{1j} \\ y_{2j} \\ \vdots \\ y_{Nj} \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} y_{11} & \cdots & y_{1K} \\ \vdots & \cdots & \vdots \\ y_{N1} & \cdots & y_{NK} \end{bmatrix}_{N\times K} = \begin{bmatrix} | & \cdots & | \\ \mathbf{y}'_1 & \cdots & \mathbf{y}'_K \\ | & \cdots & | \end{bmatrix}_{N\times K}$$

$$\hat{y}_{ij} = f_j(\tilde{\mathbf{x}}_i) = \tilde{\mathbf{w}}_j{}^T \tilde{\mathbf{x}}_i = \tilde{\mathbf{x}}_i{}^T \tilde{\mathbf{w}}_j \,, \quad \hat{\mathbf{y}}'_j = f_j(\tilde{\mathbf{X}}) = \tilde{\mathbf{X}}\tilde{\mathbf{w}}_j$$

$$\hat{\mathbf{Y}} = f(\tilde{\mathbf{X}}) = \tilde{\mathbf{X}}\tilde{\mathbf{W}}$$

$$E(\tilde{\mathbf{W}}) = \sum_{j=1}^{K} \left\|\mathbf{y}'_j - \tilde{\mathbf{X}}\tilde{\mathbf{w}}_j\right\|_2^2$$

$$E(\tilde{\mathbf{W}}) = \left\|\mathbf{Y} - \tilde{\mathbf{X}}\tilde{\mathbf{W}}\right\|_F^2$$

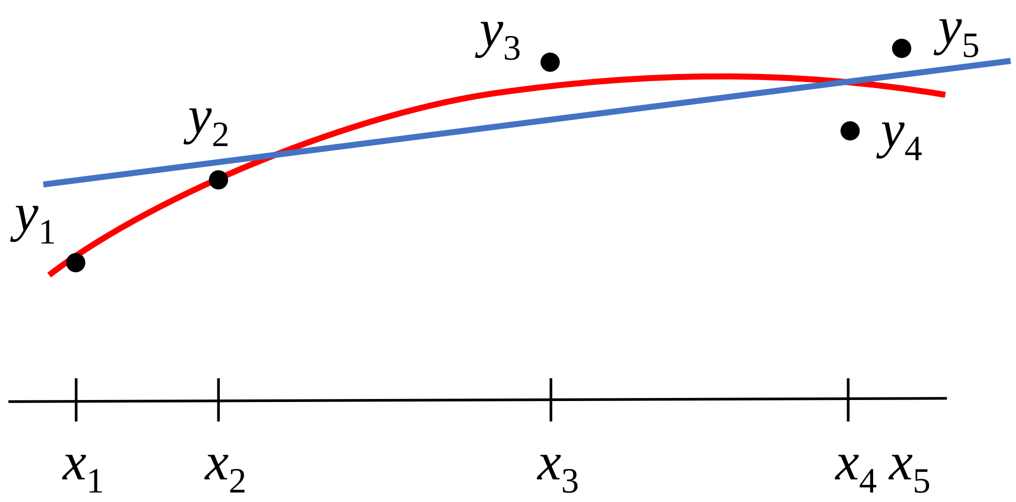
$$\tilde{\mathbf{W}}^* = \underset{\tilde{\mathbf{W}}}{\operatorname{argmin}} \left\|\mathbf{Y} - \tilde{\mathbf{X}}\tilde{\mathbf{W}}\right\|_F^2 = (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^T\mathbf{Y} = \tilde{\mathbf{X}}^+\mathbf{Y}$$

Basis Function Regression

Problem: Non-linear Relationship

- Some training data are poorly represented by a straight line
  - Weight is not a linear function of Height
  - Area of a circle is not a linear function of the radius
- A curve is better suited to fit the data for these cases
- The least squares method can readily be extended to fit the data to different non-linear models

$w_2 x^2 + w_1 x + w_0$



CSCC11 Winter 2025 Yiqing Irene Huang, CMS UTSC.

Basis Function

- The basic idea
  - Do not regress directly from the input $\mathbf{x}$.
  - Transform $\mathbf{x}$ into a set of new features $b_j(\mathbf{x})$, where $b_j(\mathbf{x})$ is non-linear
  - Regress from the new features.
  - The model is nonlinear in the input variable, but linear in the model parameters
- The case when $x \in \mathbb{R}$

$$\hat{y} = f(x) = \sum_{j=1}^{M} w_j b_j(x) + b = \sum_{j=0}^{M} w_j b_j(x), \qquad w_0 = b,\ b_0(x) = 1$$

- The $b_j(\mathbf{x})$ can be learned, in this course, we manually specify them.
- We will focus on learning the weights w_j in this course

Basis Functions Cont'd

- Monomial basis functions for polynomials

$$b_j(x) = x^j, \qquad j = 0, 1, \dots, M$$

$$\hat{y} = f(x) = \sum_{j=0}^{M} w_j x^j = \underbrace{w_0}_{\text{bias}} + \underbrace{w_1 x + w_2 x^2 + \cdots + w_M x^M}_{\text{new features}}$$

- Radial basis functions (RBF)

$$b_j(x) = e^{-\frac{(x-c_j)^2}{2\sigma_j^2}}, \qquad \hat{y} = f(x) = w_0 + \sum_{j=1}^{M} w_j b_j(x) = w_0 + \sum_{j=1}^{M} w_j e^{-\frac{(x-c_j)^2}{2\sigma_j^2}}$$

$$j = 1, \dots, K$$

RBF Properties

- RBFs are unnormalized Gaussian Functions
- The location parameter is c_j
- The width/bandwidth parameter is σ_j
- 2-D RBF $b_j(\mathbf{x})$: $\mathbb{R}^2 \to \mathbb{R}$

$$b_j(\mathbf{x}) = e^{-\frac{(\mathbf{x}-\mathbf{c}_j)^T(\mathbf{x}-\mathbf{c}_j)}{2\sigma_j^2}}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \mathbf{c}_j = \begin{bmatrix} c_{1j} \\ c_{2j} \end{bmatrix}$$

$$= e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|_2^2}{2\sigma_j^2}}$$

Basis Function Selection

- Other basis functions commonly used
  - Sinusoidal functions
  - Sigmoid functions
- Keep the number of basis functions needed small.
- Example
  - If the underlying relation between x and y is sinusoidal, then we just need one sinusoidal function.
  - If the underlying relation between x and y is cubic, then we just need a polynomial of degree 3.

Polynomials vs RBFs

- Polynomials
  - Great for very simple polynomial relationship between x and y.
  - Good for extrapolation
    - Predict at a location which is quite different from the locations where we have made the measurements for
  - Polynomial provides fits that are global.
    - Every weight w_j is influenced by all the training points throughout the entire domain of x.
  - Not good for problems that need local domain support
- RBFs
  - Good for smooth functions where local correlation length are very limited in the local scope
  - Examples: wave forms, speeches and images.

Cost Function for Basis Function Regression

Model: $\hat{y}_i = f(x_i) = \mathbf{b}_{\boldsymbol{j}}(x_i)^T \mathbf{w} = \sum_{j=1}^{M} w_j b_j(x_i)$

Cost Function: $E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f(x_i)\right)^2 = \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{M} w_j b_j(x_i)\right)^2 = \|\mathbf{y} - \mathbf{B}\mathbf{w}\|_2^2$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} \quad \mathbf{B} \equiv \mathbf{B}(\mathbf{x}) = [B_{ij}] = \begin{bmatrix} b_1(x_1) & \cdots & b_M(x_1) \\ b_1(x_2) & \cdots & b_M(x_2) \\ \vdots & \vdots & \vdots \\ b_1(x_N) & \cdots & b_M(x_N) \end{bmatrix}$$

$$B_{ij} \equiv b_j(x_i)$$

The optimal weight $\mathbf{w}^*$

$$E(\mathbf{w}) = \|\mathbf{y} - \mathbf{B}\mathbf{w}\|_2^2$$

$$= (\mathbf{y} - \mathbf{B}\mathbf{w})^T(\mathbf{y} - \mathbf{B}\mathbf{w})$$

$$= \underbrace{\mathbf{y}^T\mathbf{y}}_{\text{constant}} - \underbrace{2\mathbf{w}^T\mathbf{B}^T\mathbf{y}}_{\text{Linear Term}} + \underbrace{\mathbf{w}^T\mathbf{B}^T\mathbf{B}\mathbf{w}}_{\text{Quadratic Term}}$$

$$\frac{\partial E}{\partial \mathbf{w}} = -2\mathbf{B}^T\mathbf{y} + 2\mathbf{B}^T\mathbf{B}\mathbf{w} = \mathbf{0}$$

$$\Downarrow$$

$$\mathbf{B}^T\mathbf{B}\mathbf{w} = \mathbf{B}^T\mathbf{y}$$

$$\Downarrow$$

$$\mathbf{w}^* = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{y}$$

- Assume $(\mathbf{B}^T\mathbf{B})$ is non-singular

Hyperparameters of RBF

- The c_j and σ_j in RBF functions need to be estimated
- It is a much harder problem
- We determine them heuristically
- Method 1
  - Space all the centers c_j uniformly
  - choose all the widths σ_j so that they overlaps with their neighbors
  - Works well in one dimension
  - K^D , the number of basis functions grows exponentially.

Hyperparameters of RBF

- Method 2
  - Put one RBF at each data point
  - Set the width equal to the distance to near neighbors
  - Set width equal to the median distance of all nearest neighbors when we want to use the same width for all basis functions.
- Method 3
  - Cluster data to groups according to some similarity measure. Use clusters to determine the width (later in the course we will talk about clustering)
- Avoid setting widths too small and produce many small RBF bumps.

Overfitting and Regularization

Underfitting and Overfitting

- Underfitting
  - The model fits the training data itself poorly. That is the model is too simple and not expressive enough.
  - Example: fitting a quadrative curve using a line
- Overfitting
  - The model fits the training data too well.
  - The model does the prediction on unseen data poorly
  - The model is extremely sensitive to noise and data
  - Example: fitting a line using a higher order polynomial. The curve oscillates wildly.

Overfitting

- The big challenge in ML is to separate noise from the signal
- Common reasons for this challenge
  - Too many parameters, not enough data
  - Data are too noisy
  - We failed to model uncertainty in the models.
    - When very expressive models are used, there are many models that will fit the data reasonably well.
    - How certain are we that the one we use is the right class of model?

Regularization

- Regularization is a family of methods to control overfitting
  - Smooth model is preferred over models fluctuate widely from point to point.
  - Simple models have smaller number of parameters and do not fit the noise that well.
- Hard Constraint
  - Restrict family of models we fit in the first place
  - Avoid models that fit noise too well
  - Example: only allow small degree polynomials in the polynomial regression.
- Soft Constraint
  - Encourage smoothness with a penalty function during the estimation
  - Penalize the expressiveness of the models
  - Allow more expressive models if the benefits to have them in terms of residual errors outweighs the penalties to have them.

Regularized Least Squares

- AKA Ridge Regression in Statistics, weight decay in deep learning

$$E(\mathbf{w}) = ||\mathbf{y} - \mathbf{B}\mathbf{w}||_2^2 \; + \; \lambda||\mathbf{w}||_2^2$$

Data term

smoothness term

- λ is the regularization parameter. It controls the balance between the smoothness and the data fit
- Small weights (i.e. small $||\mathbf{w}||_2^2$) imply smoothness. Why?

Polynomial Example

$$f(x) = w_0 + w_1 x + w_2 x^2$$

- Smaller weights tend to give smoother functions

- The first derivative describes how fast the function changes

$$\frac{df}{dx} = w_1 + 2w_2 x$$

- The second derivative describes how fast the slope changes

$$\frac{d^2 f}{dx^2} = 2w_2$$

Solution for Regularized LS

$$E(\mathbf{w}) = \|\mathbf{y} - \mathbf{B}\mathbf{w}\|_2^2 \; + \; \lambda\|\mathbf{w}\|_2^2$$

$$= (\mathbf{y} - \mathbf{B}\mathbf{w})^T(\mathbf{y} - \mathbf{B}\mathbf{w}) + \lambda\mathbf{w}^T\mathbf{w}$$

$$= \mathbf{y}^T\mathbf{y} \; - 2\mathbf{w}^T\mathbf{B}^T\mathbf{y} + \mathbf{w}^T\mathbf{B}^T\mathbf{B}\mathbf{w} + \lambda\mathbf{w}^T\mathbf{I}\mathbf{w}$$

$$= \mathbf{y}^T\mathbf{y} \; - 2\mathbf{w}^T\mathbf{B}^T\mathbf{y} + (\mathbf{w}^T(\mathbf{B}^T\mathbf{B})\mathbf{w} + \mathbf{w}^T(\lambda\mathbf{I})\mathbf{w})$$

$$= \mathbf{y}^T\mathbf{y} \; - 2\mathbf{w}^T\mathbf{B}^T\mathbf{y} + \mathbf{w}^T(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{I})\,\mathbf{w}$$

$$\frac{\partial E}{\partial \mathbf{w}} = -2\mathbf{B}^T\mathbf{y} + 2(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{I})\mathbf{w} = \mathbf{0}$$

$$(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{I})\mathbf{w} = \mathbf{B}^T\mathbf{y}^T \qquad \Rightarrow \qquad \mathbf{w}^* = (\mathbf{B}^T\mathbf{B} + \lambda\mathbf{I})^{-1}\mathbf{B}^T\mathbf{y}$$

Conditioning of Linear System

Supplementary Materials

Ill-Conditioned Systems

- Consider the System

$$\mathbf{y} = \begin{pmatrix} 5 \\ 9 \\ 10 \end{pmatrix} \qquad \mathbf{A} = \begin{pmatrix} 4 & -3 & 4 \\ 2 & 4 & 3 \\ 3 & 3 & 4 \end{pmatrix} \qquad \mathbf{w} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

- If $\mathbf{y}' = \begin{pmatrix} 4.99 \\ 8.99 \\ 10.01 \end{pmatrix}$, the solution to $\mathbf{A}\mathbf{w}' = \mathbf{y}'$ is $\mathbf{w}' = \begin{pmatrix} 0.44 \\ 0.91 \\ 1.49 \end{pmatrix}$

- A relative error of $\dfrac{\|\mathbf{y} - \mathbf{y}'\|_2}{\|\mathbf{y}\|_2} \approx 0.0012$ in the target vector

  result in the relative error of $\dfrac{\|\mathbf{w} - \mathbf{w}'\|_2}{\|\mathbf{w}\|_2} \approx 0.43$ in the solution

Regularization and Conditioning of Matrix

- The condition number is the ratio of the largest singular value and the smallest singular value.
- Large condition number means the matrix significantly magnifies the errors of the vector it acts on.
  - A slight perturbation of the input will change the output drastically
- If $\mathbf{B}^T\mathbf{B}$ is nearly singular, its condition number is very large.
  - In Machine Learning, this is a sign of overfitting
- By adding the regularization diagonal matrix to the $\mathbf{B}^T\mathbf{B}$, it improves the condition of the matrix to be inverted.
- See Tutorial for a detailed numerical example

K-Nearest Neighbors Regression

Non-parametric regression model

KNN Method

- Given $\mathbf{x}$, predict $\mathbf{y}$ by using K similar training samples
- We will need to define a similarity measure
- We will need to define how to combine the K training outputs to make the prediction given $\mathbf{x}$
- Let $N_K(\mathbf{x})$ be the set of indices of K training points closest to $\mathbf{x}$
- The simplest way is to compute the average of the training points

$$\mathbf{y} = \frac{\sum_{i \in N_K(\mathbf{x})} \mathbf{y}}{K}$$

Weighted KNN Method

- Given points closer to $\mathbf{x}$ a higher influence factor
- Compute the weighted average of K training points

$$\mathbf{y} = \frac{\sum_{i \in N_K(\mathbf{x})} w(\mathbf{x}_i)\mathbf{y}}{\sum_{i \in N_K(\mathbf{x})} w(\mathbf{x}_i)} \qquad w(\mathbf{x}_i) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}\|_2^2}{2\sigma^2}}$$

- We need to decide parameters of K and σ.

KNN Methods

- Nearest Neighbor methods is a family of non-parametric models
- It provides a nice baseline for many problems in machine learning.
- It keeps around all training data. The training data themselves are our learning model.
- The representation cost of the model increase as the data size increases
- KNN can solve both classification and regression problems

Quadratics

Vector Calculus

Vector Norm and Inner Product

$$\sum_{i=1}^{N} v_i^2 = [v_1, v_2, \dots, v_N]\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix}$$

$$= \mathbf{v}^T\mathbf{v} = \|\mathbf{v}\|_2^2$$

$$v_i \equiv e_i = y_i - \hat{y}_i = y_i - \widetilde{\mathbf{w}}^T\mathbf{x}_i$$

$$\mathbf{v} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}$$

← Residual Vector

$$E(\widetilde{\mathbf{w}}) = \sum_{i=1}^{N} e_i^2 = \|\mathbf{v}\|_2^2$$

$$= \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$$

$$= \left\|\mathbf{y} - \widetilde{\mathbf{X}}\widetilde{\mathbf{w}}\right\|_2^2$$

Quadratic Vector Form

S is symmetric

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} \qquad \nabla f = \frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_D} \end{bmatrix}$$

$$\frac{\partial \mathbf{x}^T \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} = \left(\mathbf{A} + \mathbf{A}^{\mathbf{T}}\right)\mathbf{x} \qquad \frac{\partial \mathbf{x}^T \mathbf{S} \mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{S}\mathbf{x}$$

$$\frac{\partial \mathbf{b}^T \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^T \mathbf{b}}{\partial \mathbf{x}} = \mathbf{b}$$

$$f(\mathbf{x}) = \mathbf{x}^T \mathbf{S} \mathbf{x} + \mathbf{b}^T \mathbf{x} + c$$

$$= \mathbf{x}^T \mathbf{S} \mathbf{x} + \mathbf{x}^T \mathbf{b} + c$$

$$\nabla f = 2\mathbf{S}\mathbf{x} + \mathbf{b} = \mathbf{0}$$

$$\Downarrow$$

$$\mathbf{x}^* = -\mathbf{S}^{-1}\mathbf{b}$$

Assume **S** is non-singular

Quadratic Matrix Form for LS

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} \quad \tilde{\mathbf{w}} = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_D \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix}$$

$\mathbf{S}$ is symmetric and non-singular.

$$f(\mathbf{x}) = \mathbf{x}^T\mathbf{S}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c = \mathbf{x}^T\mathbf{S}\mathbf{x} + \mathbf{x}^T\mathbf{b} + c = \mathbf{0} \Rightarrow \mathbf{x}^* = -\mathbf{S}^{-1}\mathbf{b}$$

$$E(\tilde{\mathbf{w}}) = \mathbf{y}^T\mathbf{y} - 2\tilde{\mathbf{w}}^T\tilde{\mathbf{X}}^T\mathbf{y} + \mathbf{w}^T\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\tilde{\mathbf{w}}$$

$$E(\mathbf{w}) = \mathbf{y}^T\mathbf{y} - 2\mathbf{w}^T\mathbf{B}^T\mathbf{y} + \mathbf{w}^T\mathbf{B}^T\mathbf{B}\mathbf{w}$$

$\Rightarrow$

$$\mathbf{S} = \tilde{\mathbf{X}}^T\tilde{\mathbf{X}} \qquad \mathbf{b} = -2\tilde{\mathbf{X}}^T\mathbf{y}$$

$$\mathbf{S} = \mathbf{B}^T\mathbf{B} \qquad \mathbf{b} = -2\mathbf{B}^T\mathbf{y}$$

$$\mathbf{c} = \mathbf{y}^T\mathbf{y}$$

Acknowledgement

- Prof. David Fleet developed the course. He made his notes and courseware available to all of us.
- Prof. Francisco (Paco) shared his assignments with fellow colleagues.
- Prof. Rawad A. Assi shared past assignments with me