EE2211 Pre-Tutorial 6

Dr Feng LIN
feng_lin@nus.edu.sg

Agenda

- Recap
- Self-learning
- Tutorial 6

Recap

- Linear Classification
  - Binary classification
  - Multi-category classification
- Ridge regression
  - Penalty term
  - Primal and dual forms
- Polynomial Regression
  - Nonlinear decision boundary

Linear Regression

Collect Data → Model and Loss Function → Learning/Training: compute w → Prediction /Testing: Compute y_{new}

$$Xw = y$$

- X: Samples
- y: Target values

$$\frac{1}{m}\sum_{i=1}^{m}(f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2$$

- Linear or Affine function
- Squared error loss function

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

- Check the invertibility
- Least square approximation (left-inverse)

$$\hat{f}_{\mathbf{w}}(\mathbf{X}_{new}) = \mathbf{X}_{new}\hat{\mathbf{w}}$$

- Prediction for new inputs
- Testing: Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Review: Linear Regression

Learning of Scalar Function (Single Output)

For one sample: a linear model $f_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$ scalar function

For m samples: $\boldsymbol{f}_{\mathbf{w}}(\mathbf{X}) = \mathbf{X}\mathbf{w} = \mathbf{y}$

$$\mathbf{y} = \begin{bmatrix} \mathbf{x}_1^T\mathbf{w} \\ \vdots \\ \mathbf{x}_m^T\mathbf{w} \end{bmatrix} \quad \text{where} \quad \mathbf{x}_i^T = [1, x_{i,1}, \dots, x_{i,d}]$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m,1} & \dots & x_{m,d} \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

Objective: $\sum_{i=1}^{m}(f_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2 = \mathbf{e}^T\mathbf{e} = (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y})$

Learning/training when $\mathbf{X}^T\mathbf{X}$ is invertible

Least square solution: $\widehat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

Prediction/testing: $\boldsymbol{y}_{new} = \hat{\boldsymbol{f}}_{\mathbf{w}}(\mathbf{X}_{new}) = \mathbf{X}_{new}\widehat{\mathbf{w}}$

Linear Regression

Learning of Scalar Function (Single Output)

Objective: $\sum_{i=1}^{m}(f_w(x_i) - y_i)^2 = \mathbf{e}^T\mathbf{e} = (\mathbf{Xw} - \mathbf{y})^T(\mathbf{Xw} - \mathbf{y})$

$$\sum_{i=1}^{m}(f_w(x_i) - y_i)^2 = \sum_{i=1}^{m}\left(x_i^T w - y_i\right)^2 = (x_1^T w - y_1)^2 + (x_2^T w - y_2)^2 + \cdots + (x_m^T w - y_m)^2$$

$$= [x_1^T w - y_1 \quad x_2^T w - y_2 \quad \cdots \quad x_m^T w - y_m]\begin{bmatrix} x_1^T w - y_1 \\ x_2^T w - y_2 \\ \vdots \\ x_m^T w - y_m \end{bmatrix}$$

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

Review: Linear Regression

Learning of Vectored Function (Multiple Outputs)

$$\mathbf{F_w}(\mathbf{X}) = \mathbf{X}\mathbf{W} = \mathbf{Y}$$

$$\begin{array}{l} \text{Sample } 1 \dashrightarrow \\ \vdots \\ \text{Sample } m \dashrightarrow \end{array} = \begin{bmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_m^T \end{bmatrix} \mathbf{W} = \begin{bmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{m,1} & \dots & x_{m,d} \end{bmatrix} \begin{bmatrix} w_{0,1} & \dots & w_{0,h} \\ w_{1,1} & \dots & w_{1,h} \\ \vdots & \ddots & \vdots \\ w_{d,1} & \dots & w_{d,h} \end{bmatrix}$$

$$\begin{array}{l} \text{Sample 1's output} \dashrightarrow \\ \vdots \\ \text{Sample } m\text{'s output} \dashrightarrow \end{array} = \begin{bmatrix} y_{1,1} & \dots & y_{1,h} \\ & \vdots & \\ y_{m,1} & \dots & y_{m,h} \end{bmatrix}$$

Least Squares Regression

If $\mathbf{X}^T\mathbf{X}$ is invertible, then

Learning/training: $\widehat{\mathbf{W}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$

Prediction/testing: $\hat{\mathbf{F}}_\mathbf{w}(\mathbf{X}_{new}) = \mathbf{X}_{new}\widehat{\mathbf{W}}$

$$\mathbf{X} \in \mathcal{R}^{m\times(d+1)},\ \mathbf{W} \in \mathcal{R}^{(d+1)\times h},\ \mathbf{Y} \in \mathcal{R}^{m\times h}$$

Review: Linear Regression

Learning of Vectored Function (Multiple Outputs)

Each $\mathbf{w}$ is associated with one output

$$J(\mathbf{W}) = \mathrm{trace}(\mathbf{E}^T\mathbf{E})$$

$$= \mathrm{trace}\left(\begin{bmatrix} \mathbf{e}_1^T \\ \vdots \\ \mathbf{e}_h^T \end{bmatrix} [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \dots \quad \mathbf{e}_h]\right)$$

$$= \mathrm{trace}\left(\begin{bmatrix} \mathbf{e}_1^T\mathbf{e}_1 & \mathbf{e}_1^T\mathbf{e}_2 & \dots & \mathbf{e}_1^T\mathbf{e}_h \\ \mathbf{e}_2^T\mathbf{e}_1 & \mathbf{e}_2^T\mathbf{e}_2 & \dots & \mathbf{e}_2^T\mathbf{e}_h \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{e}_h^T\mathbf{e}_1 & \mathbf{e}_h^T\mathbf{e}_2 & \dots & \mathbf{e}_h^T\mathbf{e}_h \end{bmatrix}\right) = \sum_{k=1}^{h} \mathbf{e}_k^T\mathbf{e}_k$$

$$\mathbf{E} = \mathbf{X}\mathbf{W} - \mathbf{Y} = \mathbf{X}[\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_h] - [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_h]$$

$$= [\mathbf{X}\mathbf{w}_1 - \mathbf{y}_1, \mathbf{X}\mathbf{w}_2 - \mathbf{y}_2, \cdots, \mathbf{X}\mathbf{w}_h - \mathbf{y}_h]$$

$$= [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_h]$$

Linear Classification

Linear Methods for Classification

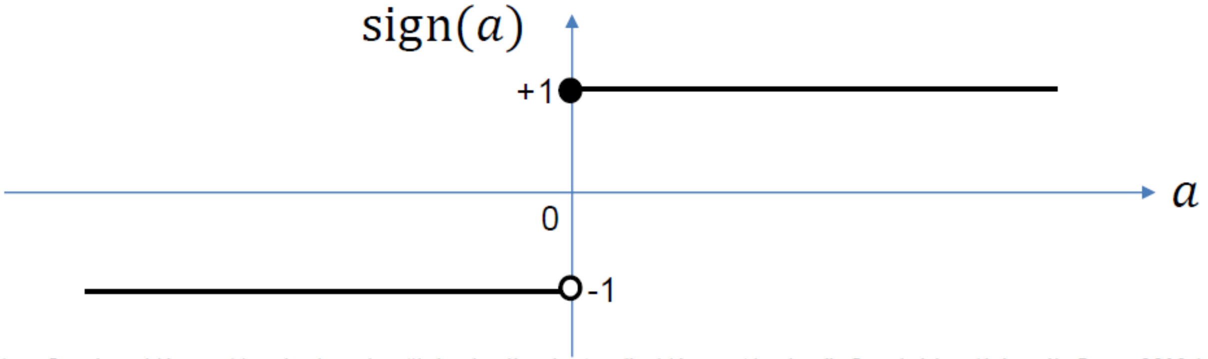
Binary Classification:

If $\mathbf{X}^T\mathbf{X}$ is invertible, then

Learning: $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, $y_i \in \{-1, +1\}, i = 1, \dots, m$

Prediction: $\hat{f}^c_{\mathbf{w}}(\mathbf{x}_{new}) = \text{sign}(\mathbf{x}^T_{new}\hat{\mathbf{w}})$ for each row $\mathbf{x}^T_{new}$ of $\mathbf{X}_{new}$

$$\textbf{sign}(a) = +1 \text{ for } a \geq 0 \text{ and } -1 \text{ for } a < 0$$



Ref: [Book4] Stephen Boyd and Lieven Vandenberghe, "Introduction to Applied Linear Algebra", Cambridge University Press, 2018 (chp.14)

Linear Classification

Linear Methods for Classification

Multi-Category Classification:

If $\mathbf{X}^T\mathbf{X}$ is invertible, then

Learning: $\widehat{\mathbf{W}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}, \quad \mathbf{Y} \in \mathbf{R}^{m \times C}$

Prediction: $\hat{f}^c_{\mathbf{w}}(\mathbf{x}_{new}) = \arg\max_{k=1,\dots,C}\left(\mathbf{x}^T_{new}\widehat{\mathbf{W}}(:,k)\right)$ for each $\mathbf{x}^T_{new}$ of $\mathbf{X}_{new}$

Each row (of $i = 1, \dots, m$) in $\mathbf{Y}$ has an **one-hot** encoding/assignment:

e.g., target for class-1 is labelled as $\mathbf{y}_i^T = [1, 0, 0, \dots, 0]$ for the ith sample,

target for class-2 is labelled as $\mathbf{y}_j^T = [0, 1, 0, \dots, 0]$ for the jth sample,

target for class-C is labelled as $\mathbf{y}_m^T = \underbrace{[0, 0, \dots, 0, 1]}_{C}$ for the mth sample.

Ref: Hastie, Tibshirani, Friedman, "The Elements of Statistical Learning", (2nd ed., 12th printing) 2017 (chp.4)

Ridge Regression

Recall Linear regression

Objective: $\widehat{\mathbf{w}} = \text{argmin} \sum_{i=1}^{m} (f_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2 = (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y})$

The learning computation: $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

We cannot guarantee that the matrix $\mathbf{X}^T\mathbf{X}$ is invertible

Ridge regression: shrinks the regression coefficients *w* by imposing a penalty on their size

Objective: $\widehat{\mathbf{w}} = \text{argmin} \sum_{i=1}^{m} (f_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2 + \lambda \sum_{j=1}^{d} w_j^2$

$= \text{argmin}\ (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^T\mathbf{w}$

Here $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of λ, the greater the amount of shrinkage.

Note: *m* samples & *d* parameters

Ridge Regression

Using a linear model:

$$\min_{\mathbf{w}} (\mathbf{Xw} - \mathbf{y})^T (\mathbf{Xw} - \mathbf{y}) + \lambda \mathbf{w}^T \mathbf{w}$$

Solution:

$$\frac{\partial}{\partial \mathbf{w}} \left((\mathbf{Xw} - \mathbf{y})^T (\mathbf{Xw} - \mathbf{y}) + \lambda \mathbf{w}^T \mathbf{w} \right) = \mathbf{0}$$

$$\Rightarrow 2\mathbf{X}^T\mathbf{Xw} - 2\mathbf{X}^T\mathbf{y} + 2\lambda\mathbf{w} = \mathbf{0}$$

$$\Rightarrow \mathbf{X}^T\mathbf{Xw} + \lambda\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

$$\Rightarrow (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

where **I** is the *dxd* identity matrix

Page 10, Lecture 5

$$\frac{d\mathbf{Ax}}{d\mathbf{x}} = \mathbf{A}$$

$$\frac{d(\boldsymbol{b}^T\mathbf{x})}{d\mathbf{x}} = \boldsymbol{b} \qquad \frac{d(\mathbf{y}^T\mathbf{Ax})}{d\mathbf{x}} = \mathbf{A}^T\mathbf{y}$$

$$\frac{d(\mathbf{x}^T\mathbf{Ax})}{d\mathbf{x}} = (\mathbf{A} + \mathbf{A}^T)\mathbf{x}$$

Here on, we shall focus on single column of output $\mathbf{y}$ in derivations in the sequel

Learning: $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$

Ref: Hastie, Tibshirani, Friedman, "The Elements of Statistical Learning", (2nd ed., 12th printing) 2017 (chp.3)

Ridge Regression

The learning computation:

$$\hat{\boldsymbol{w}} = (X^T X)^{-1} X^T y$$

$$(X^T X)^{-1} = \frac{1}{|X^T X|}(X^T X)^* \rightarrow \frac{1}{0}(X^T X)^* \Rightarrow \hat{\boldsymbol{w}} \rightarrow \infty$$

If $X^T X$ is not invertible, that means its determination is 0. This causes the denominator of $(X^T X)^{-1}$ to approach 0, which in turn causes w to approach infinity, making it impossible to fit the data well.

Ridge regression: shrinks the regression coefficients w by impose penalty on their size

$$\textbf{Objective: } \widehat{\mathbf{w}} = \operatorname{argmin} \sum_{i=1}^{m} (f_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2 + \lambda \sum_{j=1}^{d} w_j^2$$

$$= \operatorname{argmin} (\mathbf{Xw} - \mathbf{y})^T (\mathbf{Xw} - \mathbf{y}) + \lambda \mathbf{w}^T \mathbf{w}$$

$\lambda \|\boldsymbol{w}\|^2$

Regularization or penalty term or ridge term

Ridge Regression

Ridge Regression in Primal Form (when m > d)

$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})$ is invertible for $\lambda > 0$,

Learning: $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$

Prediction: $\hat{f}_{\mathbf{w}}(\mathbf{X}_{new}) = \mathbf{X}_{new}\hat{\mathbf{w}}$

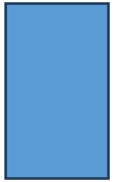
Ridge Regression in Dual Form (when m < d)

$(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})$ is invertible for $\lambda > 0$,

Learning: $\hat{\mathbf{w}} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{y}$

Prediction: $\hat{f}_{\mathbf{w}}(\mathbf{X}_{new}) = \mathbf{X}_{new}\hat{\mathbf{w}}$

Linear and Ridge Regression

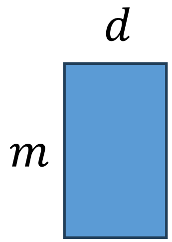


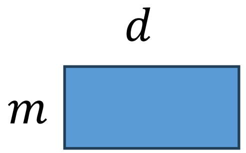
	Linear Regression	Ridge Regression
Over-determined system $(m > d)$	Left inverse $\hat{\boldsymbol{w}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$	Primal Form $\hat{\boldsymbol{w}} = \left(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$
Under-determined system $(m < d)$	Right inverse $\hat{\boldsymbol{w}} = \boldsymbol{X}^T\left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1}\boldsymbol{y}$	Dual Form $\hat{\boldsymbol{w}} = \boldsymbol{X}^T\left(\boldsymbol{X}\boldsymbol{X}^T + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{y}$

Noted: 1) The primal form can be used to solve under-determined system, but it is better suited for over-determined system. 2) The dual form of ridge regression is often more computationally efficient in under-determined system than the primal form.

When to Use Primal vs. Dual Form?

Ridge Regression	Matrix Inversion Size	Best for
Primal Form $\hat{\boldsymbol{w}} = \left(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$	$d \times d$	Over-determined system $(m > d)$
Dual Form $\hat{\boldsymbol{w}} = \boldsymbol{X}^T\left(\boldsymbol{X}\boldsymbol{X}^T + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{y}$	$m \times m$	Under-determined system $(m < d)$





Polynomial Regression

Motivation: nonlinear decision surface

- Based on the sum of products of the variables
- E.g. when the input dimension is *d=2,*

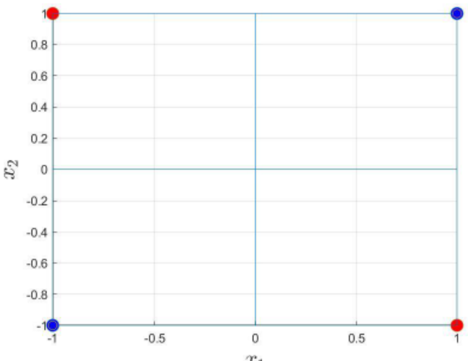
a polynomial function of degree = 2 is:

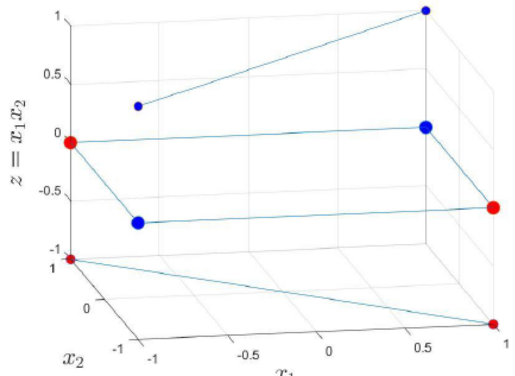
$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_{12}\, x_1 x_2 + w_{11}\, x_1^2 + w_{22}\, x_2^2 .$$

XOR problem

$$\mathbf{x}_1 = \begin{bmatrix}+1 & +1\end{bmatrix}^\top \quad y_1 = +1$$
$$\mathbf{x}_2 = \begin{bmatrix}-1 & +1\end{bmatrix}^\top \quad y_2 = -1$$
$$\mathbf{x}_3 = \begin{bmatrix}+1 & -1\end{bmatrix}^\top \quad y_3 = -1$$
$$\mathbf{x}_4 = \begin{bmatrix}-1 & -1\end{bmatrix}^\top \quad y_4 = +1$$

$$f_{\mathbf{w}}(\mathbf{x}) = x_1 x_2$$





Polynomial Regression

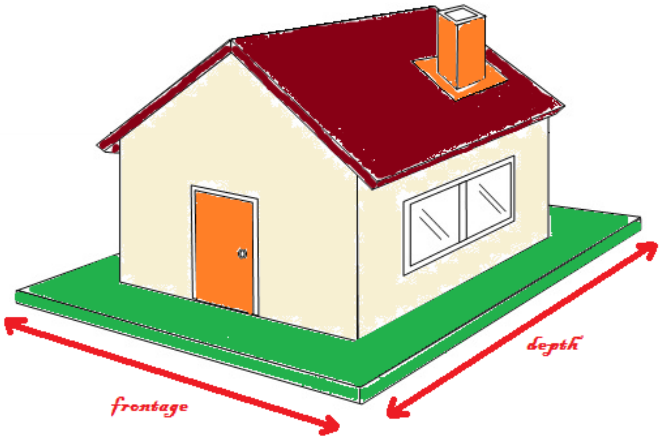
Motivation: Nonlinear Prediction

E.g. predicting the price of the house. Suppose you have two features:

- x_1: the frontage of house (the width of the property)
- x_2: the depth of the house.

We might build a linear regression model like this

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$



If we want to predict house prices, we might focus on the house or land area as key factors and create a new feature accordingly.

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_{12} \underbrace{x_1 x_2}_{\text{Aera}}$$

Machine Learning — Andrew

Polynomial Regression

Polynomial Expansion

- The linear model $f_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$ can be written as

$$f_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^T\mathbf{w}$$
$$= \sum_{i=0}^{d} x_i w_i \,, \qquad x_0 = 1$$
$$= w_0 + \sum_{i=1}^{d} x_i w_i \,.$$

- By including additional terms involving the products of pairs of components of $\mathbf{x}$, we obtain a quadratic model:

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij} x_i x_j \,.$$

2nd order: $f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_{12} x_1 x_2 + w_{11} x_1^2 + w_{22} x_2^2$

3rd order: $f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_{12} x_1 x_2 + w_{11} x_1^2 + w_{22} x_2^2 +$
$\sum_{i=1}^{d} \sum_{j=1}^{d} \sum_{k=1}^{d} w_{ijk} x_i x_j x_k, \;\; d = 2$

Ref: Duda, Hart, and Stork, "Pattern Classification", 2001 (Chp.5)

Polynomial Regression Terms Table

$$C(n,r) = \frac{(n+r-1)!}{r!\,(n-1)!}$$

n: input features +1
r: polynomial degree

For example, 2 input features and 4 degree, then $n = 2 + 1 = 3,\ r = 4$

$$C(3,4) = \frac{(3+4-1)!}{4!\,(3-1)!} = 15$$

	Degree 1	Degree 2	Degree 3	Degree 4	Degree 5
1 feature(s)	2	3	4	5	6
2 feature(s)	3	6	10	15	21
3 feature(s)	4	10	20	35	56
4 feature(s)	5	15	35	70	126
5 feature(s)	6	21	56	126	252
6 feature(s)	7	28	84	210	462
7 feature(s)	8	36	120	330	792

Polynomial Regression

Generalized Linear Discriminant Function

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij} x_i x_j + \sum_{i=1}^{d}\sum_{j=1}^{d}\sum_{k=1}^{d} w_{ijk} x_i x_j x_k + \cdots$$

$$\boldsymbol{f}_{\mathbf{w}}(\mathbf{x}) = \mathbf{P}\mathbf{w}$$

(Note: $\mathbf{P} \triangleq \mathbf{P}(\mathbf{X})$ for symbol simplicity)

$$= \begin{bmatrix} \boldsymbol{p}_1^T \mathbf{w} \\ \vdots \\ \boldsymbol{p}_m^T \mathbf{w} \end{bmatrix}$$

where $\boldsymbol{p}_l^T \mathbf{w} = [1, x_{l,1}, \ldots, x_{l,d}, \ldots, x_{l,i}x_{l,j}, \ldots, x_{l,i}x_{l,j}x_{l,k}, \ldots] \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \\ \vdots \\ w_{ij} \\ \vdots \\ w_{ijk} \\ \vdots \end{bmatrix}$

$l = 1, \ldots, m$; d denotes the dimension of input features; m denotes the number of samples

Polynomial Regression

Ridge Regression in Primal Form (m > d)

For $\lambda > 0$,

Learning: $$\hat{\mathbf{w}} = (\mathbf{P}^T\mathbf{P} + \lambda\mathbf{I})^{-1}\mathbf{P}^T\mathbf{y}$$

Prediction: $$\hat{f}_{\mathbf{w}}(\mathbf{P}(\mathbf{X}_{new})) = \mathbf{P}_{new}\hat{\mathbf{w}}$$

Ridge Regression in Dual Form (m < d)

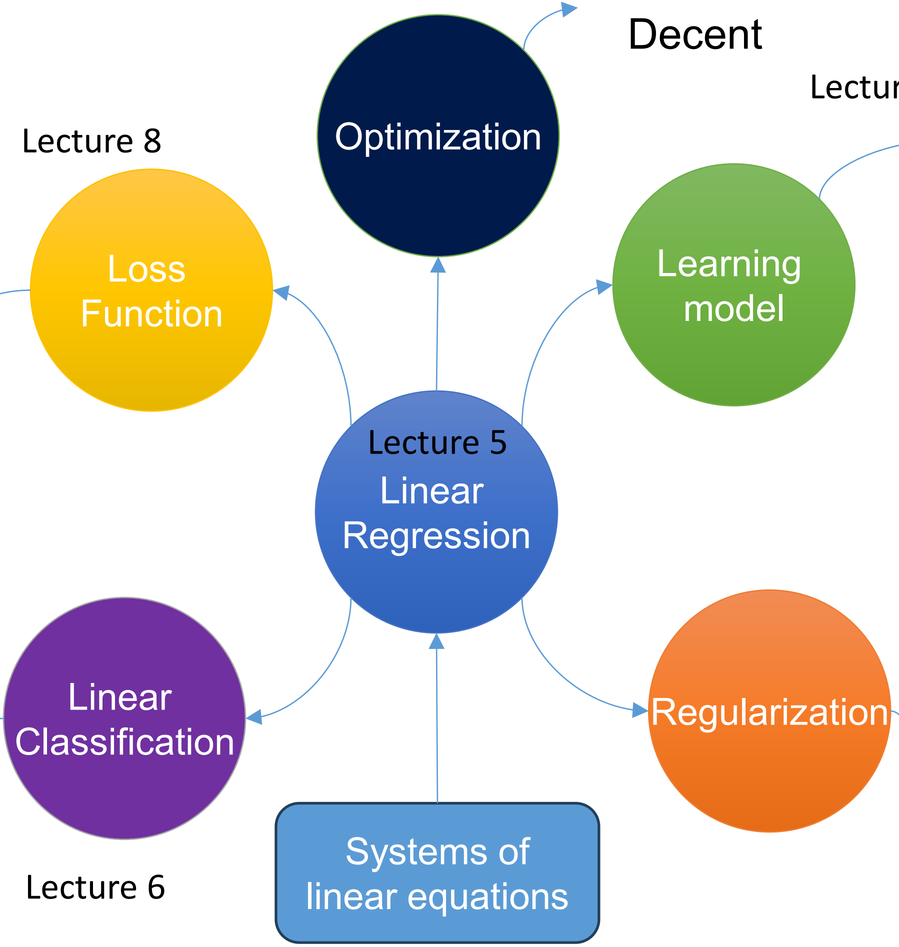
For $\lambda > 0$,

Learning: $$\hat{\mathbf{w}} = \mathbf{P}^T(\mathbf{P}\mathbf{P}^T + \lambda\mathbf{I})^{-1}\mathbf{y}$$

Prediction: $$\hat{f}_{\mathbf{w}}(\mathbf{P}(\mathbf{X}_{new})) = \mathbf{P}_{new}\hat{\mathbf{w}}$$

Note: Change **X** to **P** with reference to slides 15/16; m & d refers to the size of **P** (not **X**)

Recap



Lecture 8

- Gradient Decent

Lecture 6

- Linear model
- Affine model
- Polynomial model
- Sigmoid function
- Rectified linear unit (ReLU)
- Exponential

Lecture 8

- Square error loss
- Binary loss
- Hinge loss
- Exponential loss

Lecture 6&7

- L2 regularization (Ridge)
- L1 regularization (LASSO)
- …

- Binary classification
- Multi-Category Classification

THANK YOU