Linear Regression

" $y = wx + c$ " (affine model), or " $X\_aug \cdot \hat{w} = y$ " (matrix eqn)

$y = \underbrace{(1 \;\; x)}_{x_{aug}} \underbrace{\begin{pmatrix} c \\ w \end{pmatrix}}_{w_{aug}}$

e.g. $f(x_1, x_2) = w_0 \cdot 1 + w_1 x_1 + w_2 x_2$ where $f(\cdot)$ predicts y using x_1 & x_2.

Available data: X_train, y_train, X_test

what to do?

① Augment X: $X\_a \leftarrow \begin{pmatrix} 1 & X\_train \\ \vdots & \\ 1 & \end{pmatrix}$

② Find $\hat{w}$.

(2A) determine shape of X_a: m = no. of rows (samples), d = no. of coln. (features)

(2B) Apply Left / Right Inverse to get linear weights for the linear model

$m > d$ (primal): $\hat{w} \leftarrow (X_a^T X_a)^{-1} X_a^T \cdot y\_train$ (Left inverse)

$m < d$ (dual): $\hat{w} \leftarrow X_a^T (X_a X_a^T)^{-1} \cdot y\_train$ (Right inverse)

use $X := X\_aug$

Note: $\hat{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{pmatrix}$ constant coeff (w_0); w_i is the weight for x_i for $X = (x_1, \dots, x_n)$

③ Apply linear model to X_test to predict y_pred

! need to augment X_test first!

ⓐ $X\_test\_aug \leftarrow \begin{pmatrix} 1 & X\_test \\ \vdots & \\ 1 & \end{pmatrix}$

ⓑ y_pred ← $X\_test\_aug \cdot \hat{w}$

(opt: 4) compare y_pred with a hidden y_test: compute MSE (y_pred, y_test) } test error.

Ridge Regression

What happens if $X^T X$ (resp. XX^T) is not invertible?

→ use $X^T X + \lambda I$ (resp. $XX^T + \lambda I$) instead with $\lambda > 0$ small. e.g. $\lambda = 0.0001$

→ this is called regularisation.

→ two purposes: ① makes $X^T X + \lambda I$ invertible ② biases the cost fn, to prefer $\hat{w}$ that is smaller

more precisely, s.t. $\|\hat{w}\|^2 = \hat{w}^T w$ is small

cost fn in primal form

original cost: $\min_w \|Xw - y\|^2 = \min_w (Xw - y)^T (Xw - y)$

regularised cost: $\min_w \|Xw - y\|^2 + \lambda \|w\|^2 = \min_w (Xw - y)^T (Xw - y) + \lambda w^T w$

How to get w that minimises the cost?: differentiate to get soln → Left Inverse. is now $\boxed{(X^T X + \lambda I)^{-1} X^T}$

cost fn in dual form (extra)

original cost: $\min_w \|Xw - y\|^2 = \min_w \sum_{i=1}^{m} (y_i - x^{(i)} w)^2$ where $X = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(m)} \end{pmatrix}$, $y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$

regularised cost: $\min_w \|Xw - y\|^2 + \lambda \|w\|^2 = \min_w \sum_{i=1}^{m} (y_i - x^{(i)} w)^2 + \lambda \sum_{j=1}^{d} w_j^2$, where $w = \begin{pmatrix} w_1 \\ \vdots \\ w_d \end{pmatrix}$

How to get w that minimises the cost? (extra): need to apply Lagrangian duality. See KKT theory. This is why this is extra.

rough idea: get dual problem. solve dual problem.

Ridge Regression

(what to do? SAME as Linear regression but change the Left / Right Inverse)

① Augment X : $X\_aug \leftarrow \left(1 \,\vdots\, X\_train \right)$

② determine shape of X_aug : m = no. of rows, d = no. of coln.

③ Apply Left / Right Inverse to get linear weights for the linear model

$m > d$ (primal) : $\hat{w} \leftarrow (X^T X + \lambda I)^{-1} X^T \cdot y\_train$ (Left inverse)

$m < d$ (dual) : $\hat{w} \leftarrow X^T (X X^T + \lambda I)^{-1} \cdot y\_train$ (Right inverse)

use $X := X\_aug$

Note: constant coeff

$$\hat{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{pmatrix}$$

w_i is the weight for x_i

for $X = (x_1, \ldots, x_n)$

④ Apply linear model to X_test to predict y_pred

! need to augment X_test first!

ⓐ $X\_test\_aug \leftarrow \left(1 \,\vdots\, X\_test \right)$

ⓑ $y\_pred \leftarrow X\_test\_aug \cdot \hat{w}$

Polynomial Regression (examples)

→ non-linear model

→ e.g. $f(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_k x^k$. This is a k^{th} order (deg) polynomial in one variable x.

given x_train, we want to estimate weights $w_0, w_1, \ldots, w_k$, which are the "linear weights for each monomial" $x^i \sim w_i$.

→ e.g. $f(x_1, x_2) = w_{0,0} + w_{1,0} x_1 + w_{0,1} x_2 + w_{2,0} x_1^2 + w_{1,1} x_1 x_2 + w_{0,2} x_2^2$ } This is a 2nd-order polynomial model in two variables, x_1 & x_2.

X_train = (x_1, x_2)

observe that $f(x_1, x_2) = \begin{pmatrix} 1 & x_1 & x_2 & x_1^2 & x_1 x_2 & x_2^2 \end{pmatrix} \begin{pmatrix} w_{0,0} \\ w_{1,0} \\ w_{0,1} \\ w_{2,0} \\ w_{1,1} \\ w_{0,2} \end{pmatrix} = P\_train \cdot \hat{w}$

so we can express our polynomial model LINEARLY with respect to monomials

Model : $P\_train \, \hat{w} = y$,

where for each sample of X_train, e.g. $(x_1, x_2) = (2, 5)$

we replace with $P\_train \leftarrow (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)$
$= (1, 2, 5, 4, 10, 25)$

! Notice that P_train is already augmented. the "1" correspond to constant coeff $w_{0,0}$.
& $x_1^i x_2^j$ correspond to $w_{i,j}$.

Polynomial Regression (steps)

Data: X_train, y_train, X_test

what to do?

① : transform X_train to P_train row by row.

e.g. $$X\_train = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ x_{3,1} & x_{3,2} \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & x_{1,1} & x_{2,1} & x_{1,1}^2 & x_{1,1}x_{1,2} & x_{1,2}^2 \\ 1 & x_{2,1} & x_{2,1} & x_{2,1}^2 & x_{2,1}x_{2,2} & x_{2,2}^2 \\ 1 & x_{3,1} & x_{3,1} & x_{3,1}^2 & x_{3,1}x_{3,2} & x_{3,2}^2 \end{pmatrix} = P\_train$$

$$\begin{array}{cccccc} \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ \text{constant} & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \end{array}$$

(2A) Determine shape of P_train

m = ? same as X.

d = how many monomials ?

M1 : count one-by-one

M2 : use formula $\binom{d+k}{k}$, where we have d variables $\{x_1, x_2, \dots, x_d\}$ & k^{th} order polynomial model

---

(2B) Apply Left / Right Inverse to get linear weights for the polynomial model

$m > d$ (primal) : $\hat{w} \longleftarrow (P^TP+\lambda I)^{-1}P^T \cdot y\_train$ (Left inverse)

$m < d$ (dual) : $\hat{w} \longleftarrow P^T(PP^T+\lambda I)^{-1} \cdot y\_train$ (Right inverse)

use $P = P\_train$. since P_train is alr augmented, do not augment again

$\hat{w}$ looks like $\hat{w} = \begin{pmatrix} w_{0,0} \\ w_{1,0} \\ \vdots \\ w_{1,1} \end{pmatrix}$

④ Apply polynomial model to X_test to predict y_pred

! need to transform X_test to P_test first

ⓐ P_test gotten similarly to step ①

ⓑ y_pred ⟵ P_test · $\hat{w}$! This is same as y_pred = f(x_test)

Binary Classification "class = sgn(Xw) = sgn(y)"

Data: X_train , class_train , X_test , where class_train = $\begin{pmatrix} \pm 1 \\ \pm 1 \\ \vdots \end{pmatrix} \in \{\pm 1\}^m$

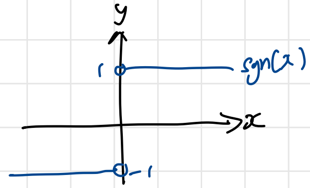
what to do?

① Apply Linear or Polynomial Regression to X_train, class_train, X_test to get y_pred = X_test · $\hat{w}$ (or y_pred = P_test · $\hat{w}$)

② Apply sign-function to y_pred

class_pred = sgn(y_pred).

set y = 0 as threshold.



Multi-Category Classification "class = $\underset{i}{\text{argmax}}\ Xw$"

Data: X_train, class_train, X_test, where class_train = $\begin{pmatrix} 1 \\ 2 \\ 1 \\ 3 \\ 1 \end{pmatrix} \in \{1,2,\dots,c\}^m$

! class_train has 'c' amount of classes.

what to do?

Step ①: Apply one-hot-encoding y_train = OHE(class_train).

idea: sample by sample, convert each output in class to a standard basis vector in $\mathbb{R}^c$

e.g. $c=3$

$$\text{class\_train} = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 3 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \text{y\_train}$$

! observe that y_train is a (m,c) array

Step ②: Perform Linear Regression OR Polynomial Regression on X_train, y_train, X_test.

→ get y_pred.

so in linear reg. "$Xw = y$"
w is a (d,c)-array

e.g. $$\text{y\_pred} = \begin{pmatrix} 0.5 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.4 \\ 0.3 & 0.2 & 0.7 \\ 0.8 & 0.2 & 0.1 \end{pmatrix}$$

Step ③: Get the class that is 'most probable'

e.g. $$\text{y\_pred} = \begin{pmatrix} 0.5 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.4 \\ 0.3 & 0.2 & 0.7 \\ 0.8 & 0.2 & 0.1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 3 \\ 3 \\ 1 \end{pmatrix} = \text{class\_pred}$$

EE2211 Tutorial 6

How many monomials?

to count monomials up to deg K from $\{x_1, \dots, x_d\}$, e.g. to count $\{1, x_1, \dots, x_d, x_1^2, x_1x_2, \dots, \dots x_1x_n, x_2^2, \dots, \dots, x_d^K\}$

$\Leftrightarrow$ counting $\{x_0^K, x_0^{K-1}x_1, \dots, x_0^{K-1}x_d, x_0^{K-2}x_1^2, \dots\}$ and sub $x_0 = 1$.

$\Leftrightarrow$ choosing K items from a list of $d+1$ items with replacement $\{x_0, x_1, \dots, x_d\}$

Formula: $\binom{(d+1)+K-1}{K}$ (no. of item; no. chosen) $= \binom{d+K}{K}$

(Ridge Regression in Dual Form)

Question 1:

Derive the solution for linear ridge regression in dual form (see Lecture 6 notes page 16).

$*$ P

$x_1 = -10$, $x_1 = -8$ $\begin{pmatrix} 1 & -10 & (-10)^2 & (-10)^3 \\ 1 & -8 & (-8)^2 & (-8)^3 \\ \vdots \end{pmatrix}$ $x = -10.$

linear model $f(x) = w_1 x + w_0.$ $x \rightarrow \boxed{} \rightarrow y$

(Polynomial Regression, 1D data)

Question 2:

Given the following data pairs for training

$$\{x = -10\} \rightarrow \{y = 5\}$$
$$\{x = -8\} \rightarrow \{y = 5\}$$
$$\{x = -3\} \rightarrow \{y = 4\}$$
$$\{x = -1\} \rightarrow \{y = 3\}$$
$$\{x = 2\} \rightarrow \{y = 2\}$$
$$\{x = 8\} \rightarrow \{y = 2\}$$

3°

por ejemplo.

How many inputs: 1

shape of X : (6, 1)

polynomial model : $f(x) = w_0 + w_1x_1 + w_2x_1^2 + w_3x_1^3.$

How to get P? : $*$

Is our polynomial regression primal or dual form? $m_p = 6$, $d_p = 4$ LI

$\rightarrow$ how many variables? P is of shape (6, 4)

(a) Perform a 3[rd]-order polynomial regression and sketch the result of line fitting.

(b) Given a test point $\{ x = 9 \}$ predict y using the polynomial model.

(c) Compare this prediction with that of a linear regression.

$\hat{w} = \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{pmatrix}$

(a) X_train $\Rightarrow$ P_train = get_poly_data(X_train, 3).

$\hat{w}$ = LI(P_train, y)

(b) ① X_test = (9) $\rightarrow$ P_test = $(1 \quad 9 \quad 9^2 \quad 9^3)$ ② y = P_test $\cdot \hat{w}$ = f(X_test)

$w_{i,j,k} \rightarrow$ weight attached to $x_1^i x_2^j x_3^k$.

(Polynomial Regression, 3D data, Python)

Question 3:

$y = w_{0,0,0} + w_{1,0,0}x_1 + w_{0,1,0}x_2 + w_{0,0,1}x_3 + w_{2,0,0}x_1^2 + w_{1,1,0}x_1x_2 + w_{1,0,1}x_1x_3 + w_{0,2,0}x_2^2 + \dots + w_{3,0,0}x_1^3 + \dots + w_{0,0,3}x_3^3$

$\binom{3+3}{3} = \binom{6}{3} = \frac{6 \times 5 \times 4}{3 \times 2} = 20.$

(a) Write down the expression for a 3[rd] order polynomial model having a 3-dimensional input.

(b) Write down the **P** matrix for this polynomial given $\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 1 \end{bmatrix}$.

what to check?

$m = 2$, $d_p = 20$

(c) Given $\mathbf{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, can a unique solution be obtained in dual form? If so, proceed to solve it.

(d) Given $\mathbf{y} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, can the primal ridge regression be applied to obtain a unique solution? If so, proceed to solve it.

by right $m < d$ P^TP invertible / not invertible

But: $P^TP + \lambda I$ invertible.

(e)

(Binary Classification, Python)

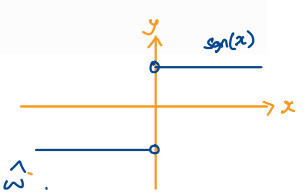
Question 4:

Given the training data:

$$\{x = -1\} \rightarrow \{ y = class1 \}$$
$$\{x = \ \ 0 \} \rightarrow \{ y = class1 \}$$
$$\{x = 0.5\} \rightarrow \{ y = class2 \}$$
$$\{x = 0.3\} \rightarrow \{ y = class1 \}$$
$$\{x = 0.8\} \rightarrow \{ y = class2 \}$$

Predict the class label for $\{x = -0.1\}$ and $\{x = 0.4\}$ using linear regression with signum discrimination.

① $y = \begin{pmatrix} +1 \\ +1 \\ -1 \\ +1 \\ -1 \end{pmatrix}.$



② Do lin. reg to get $\hat{w}$.

y_pred = X_test $\cdot \hat{w} = \begin{pmatrix} 1 & -0.1 \\ 1 & 0.4 \end{pmatrix} \cdot \hat{w}$.

③ class_pred = sgn(y_pred).

normally, $y \sim (m,1)$ array
no. of samples

now (one-hot-encoding), $Y \sim (m,c)$ array where c is no. of classes

$Xw = Y$
(m,d) (m,c)
(d,c)

(Multi-Category Classification, Python)
Question 5:
Given the training data:

$$\{x = -1\} \rightarrow \{\, y = class1 \,\}$$
$$\{x = \;\;0\,\} \rightarrow \{\, y = class1 \,\}$$
$$\{x = 0.5\} \rightarrow \{\, y = class2 \,\}$$
$$\{x = 0.3\} \rightarrow \{\, y = class3 \,\}$$
$$\{x = 0.8\} \rightarrow \{\, y = class2 \,\}$$

Step 1: encoding.
$$Y = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$m=$
$d=$

(a) Predict the class label for $\{x = -0.1\}$ and $\{x = 0.4\}$ based on linear regression towards a one-hot encoded target.

$m = 5$ $\quad d = 1+1 = 2.$

$y\_pred = X\_test\_aug \cdot \hat{w} = \begin{pmatrix} 1 & -0.1 \\ 1 & 0.4 \end{pmatrix} \hat{w} = \begin{pmatrix} x & x & x \\ x & x & x \end{pmatrix}.$

(b) Predict the class label for $\{x = -0.1\}$ and $\{x = 0.4\}$ using a polynomial model of 5th order and a one-hot encoded target.

X_train → P_train.
shape =

$f(x) =$

$= (\quad)$

$f(X) = P \cdot w$

(Multi-Category Classification, Python)
Question 6 (continued from Q3 of Tutorial 2):
Get the data set "`from sklearn.datasets import load_iris`". Use Python to perform the following tasks.

(a) Split the database into two sets: 74% of samples for training, and 26% of samples for testing. Hint: you might want to utilize `from sklearn.model_selection import train_test_split` for the splitting.

(b) Construct the target output using one-hot encoding.

(c) Perform a linear regression for classification (without inclusion of ridge, utilizing one-hot encoding for the learning target) and compute the number of test samples that are classified correctly.

(d) Using the same training and test sets as in above, perform a 2nd order polynomial regression for classification (again, without inclusion of ridge, utilizing one-hot encoding for the learning target) and compute the number of test samples that are classified correctly. Hint: you might want to use `from sklearn.preprocessing import PolynomialFeatures` for generation of the polynomial matrix.

Question 7

MCQ: there could be more than one answer. Given three samples of $\underbrace{\text{two-dimensional}}_{d=2}$ data points $\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 3 & 3 \end{bmatrix}$ with corresponding target vector $\mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$. Suppose you want to use a full third-order polynomial model to fit these data.

$k = deg = 3$

Which of the following is/are true?

a) The polynomials model has 10 parameters to learn T
b) The polynomial learning system is an under-determined one T
c) The learning of the polynomial model has infinite number of solutions T.
d) The input matrix **X** has linearly dependent samples T.
e) None of the above

dp=10.
m<dp.

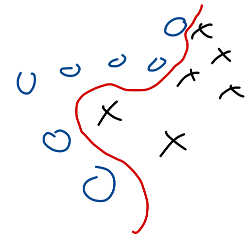
Lin dep?

deg	
0	1
1	x_1, x_2
2	$x_1^2,\ x_1x_2,\ x_2^2$
3	$x_1^3,\ x_1^2x_2,\ x_1x_2^2,\ x_2^3$.

$m > d$: "no soln".

$m < d$: under constraint.
inf. solutions.

no. of parameters $= \binom{d+k}{k}$
$= \binom{3+2}{3} = \binom{5}{3}$
$= \dfrac{5!}{3!(5-3)!} = \dfrac{5 \times 4}{2}$
$= 10.$



Question 8

MCQ: there could be more than one answer. Which of the following is/are true?

a) The polynomial model can be used to solve problems with nonlinear decision boundary. T.

→ "add λI". → classify > 2 classes. ① OvE ② Linear/Poly regression. ③ PKD.

b) The ridge regression cannot be applied to multi-target regression. T.

c) The solution for learning feature $\mathbf{X}$ with target $\mathbf{y}$ based on linear ridge regression can be written as $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$ for $\lambda > 0$. As λ increases, $\underbrace{\hat{\mathbf{w}}^T\hat{\mathbf{w}}}_{= \|w\|^2}$ decreases. T.

cost = MSE(pred, true) + $\lambda \|w\|^2$.

d) If there are four data samples with two input features each, the full second-order polynomial model is an over-determined system. F

$X = 4\begin{pmatrix} & \end{pmatrix}^{2}$

$m = 4$, $d_p = 6$ ( )-determined.

$d = \binom{2+2}{2} = \frac{4 \cdot 3}{2} = 6.$