@ operator needs py3.5 [what's new in py3.5?]

140

HW Assignment 1 (Due by 10:30am on Sep 26)

Bhishan Poudel

1 Theory (40 points)

1. [**Polynomial Curve Fitting**, 20 points]
Consider the problem of fitting a dataset of N points with a polynomial of degree M, by minimizing the sum-of-squares error:

$$J(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}(h_{\mathbf{w}}(\mathbf{x}_n) - t_n)^2 \tag{1}$$

where $h_{\mathbf{w}}(\mathbf{x}) = \sum_{j=0}^{M} w_j x^j$. We have shown in class that the solution to minimizing $J(\mathbf{w})$ satisfies the following set of linear equations:

$$\sum_{j=0}^{M} A_{ij} w_j = T_i \tag{2}$$

$$\text{where } A_{ij} = \sum_{n=1}^{N} x_n^{i+j} \text{ and } T_i = \sum_{n=1}^{N} x_n^i t_n \tag{3}$$

Derive the solution for the regularized version of polynomial curve fitting, which minimizes the objective function below:

$$J(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}(h_{\mathbf{w}}(\mathbf{x}_n) - t_n)^2 + \frac{\lambda}{2}||\mathbf{w}||^2 \tag{4}$$

2. [**Probability Theory**, 20 points] *Exercise 1.3, page 58 in PRML.*
Suppose that we have three coloured boxes r (red), b (blue), and g (green). Box r contains 3 apples, 4 oranges, and 3 limes, box b contains 1 apple, 1 orange, and 0 limes, and box g contains 3 apples, 3 oranges, and 4 limes. If a box is chosen at random with probabilities $p(r) = 0.2$, $p(b) = 0.2$, $p(g) = 0.6$, and a piece of fruit is removed from the box (with equal probability of selecting any of the items in the box), then (a) what is the probability of selecting an apple? If we observe that the selected fruit is in fact an orange, (b) what is the probability that it came from the green box?

3. [**Ridge Regression (*)**, 20 points]
Consider the regularized linear regression objective shown below:

$$J(\mathbf{w}) = \frac{1}{2N}\sum_{n=1}^{N}(h(x_n, \mathbf{w}) - t_n)^2 + \frac{\lambda}{2}||\mathbf{w}||^2$$

(a) Minimizing the L2 norm of $\mathbf{w}$ drives all parameters, including w_0, towards 0. Are there situations in which we do not want to constrain w_0 to be small? If yes, give an example, if not show why it is useful to constrain all the weights to be small, including w_0.

(b) We have seen in class how to compute the weights $\mathbf{w}$ that minimize $J(\mathbf{w})$. Assume now that we replace $||\mathbf{w}||$ in $J(\mathbf{w})$ with $||\mathbf{w}_{[1:]}||$, where $\mathbf{w}_{[1:]} = [w_1, w_2, ..., w_M]$. Derive the solution for $\mathbf{w}$ that minimizes this new objective function.

2 Implementation (80 points)

In this exercise, you are asked to run an experimental evaluation of linear regression on the Athens houses dataset, and on an artificial dataset with and without L2 regularization. The input data is available at `http://ace.cs.ohio.edu/~razvan/courses/ml4900/hw01.zip`. Make sure that you organize your code in folders as shown in the table below. Write code only in the Python files indicated in bold.

$w = (X^T X)^{-1} X^T t = [w_0\ w_1\ w_2\ w_3]$ (row 2d array)

$RMSE = \sqrt{\frac{\sum_{n=1}^{N}(h_n - t_n)^2}{N}}$

$= \frac{np.sqrt(np.sum(h-t)^2)}{N}$

$= np.sqrt((h-t)^2.mean())$

ml4900/
hw01/
code/
univariate.py
multivariate.py
polyfit.py
train_test_line.png
data/
univariate/
train.txt, test.txt
multivariate/
train.txt, test.txt
polyfit/
train.txt, test.txt, devel.txt

1. [**Univariate Regression**, 20 points]
   Train a univariate linear regression model to predict house prices as a function of their floor size, based on the solution to the system with 2 linear equations discussed in class. Use the dataset from the folder `hw01/data/univariate`. Python3 skeleton code is provided in `univariate.py`. After training print the parameters and report the RMSE and the objective function values on the training and test data. Plot the training using the default blue circles and test examples using lime green triangles. On the same graph also plot the linear approximation.

2. [**Multivariate Regression**, 20 points]
   Train a univariate linear regression model to predict house prices as a function of their floor size, number of bedrooms, and year. Use the normal equations discussed in class, and evaluate on the dataset from the folder `hw01/data/multivariate`. Python3 skeleton code is provided in `multivariate.py`. After training print the parameters and report the RMSE and the objective function values on the training and test data. Compare the test RMSE with the one from the univariate case above.

3. [**Polynomial Curve Fitting**, 40 points]
   In this exercise, you are asked to run an experimental evaluation of a linear regression model, with and without regularization. Use the normal equations discussed in class, and evaluate on the dataset from the folder `hw01/data/polyfit`.

   (a) Select 30 values for $x \in [0, 1]$ uniformly spaced, and generate corresponding t values according to $t(x) = sin(2\pi x) + x(x + 1)/4 + \epsilon$, where $\epsilon = N(0, 0.005)$ is a zero mean Gaussian with variance 0.005. Save and plot all the values. Done in `dataset.txt`.

(b) Split the 30 samples (x_n, t_n) in three sets: 10 samples for training, 10 samples for validation, and 10 samples for testing. Save and plot the 3 datasets separately. Done in `train.txt, test.txt, devel.txt.`

(c) Consider a linear regression model with polynomial basis functions, trained with the objective shown below:

$$J(\mathbf{w}) = \underbrace{\frac{1}{2N}\sum_{n=1}^{N}(h(x_n, \mathbf{w}) - t_n)^2}_{\text{half MSE}} + \underbrace{\frac{\lambda}{2}||\mathbf{w}||^2}_{\text{L2 Regularizer}}$$

$\sum_{j=1}^{M} w_j^2$ generally we do not regularize w_0.

Show the closed form solution (vectorized) for the weights $\mathbf{w}$ that minimize $J(\mathbf{w})$.

(d) Train and evaluate the linear regression model in the following scenarios:

1. Without regularization: Use the training data to infer the parameters $\mathbf{w}$ for all values of $M \in [0, 9]$. For each order M, compute the RMSE separately for the training and test data, and plot all the values on the same graph, as shown in class.
2. With regularization: Fixing $M = 9$, use the training data to infer the parameters $\mathbf{w}$, one parameter vector for each value of $\ln \lambda \in [-50, 0]$ in steps of 5. For each parameter vector (lambda value), compute the RMSE separately for the training and validation data, and plot all the values on the same graph, as shown in class. Select the regularization parameter that leads to the parameter vector that obtains the lowest RMSE on the validation data, and use it to evaluate the model on the test data. Report and compare the test RMSE with the one obtained without regularization.

3 Submission

Turn in a hard copy of your homework report at the beginning of class on the due date. Electronically submit on Blackboard a hw01.zip file that contains the hw01 folder in which your code is in the 3 required files.

On a Linux system, creating the archive can be done using the command:

```
> zip -r hw01.zip hw01.
```

Please observe the following when handing in homework:

1. Structure, indent, and format your code well.
2. Use adequate comments, both block and in-line to document your code.
3. On the theory assignment, **clear and complete explanations and proofs of your results are as important as getting the right answer.**
4. Make sure your code runs correctly when used in the directory structure shown above.

HW 1

Due: Sep 19 TU

Bhishan Poudel

<u>QN 1.1</u> Polynomial curve fitting

cost function without the regularization is

$$\boxed{J(w) = \frac{1}{2N} \sum_{n=1}^{N} \left(h_w(x_n) - t_n\right)^2} \quad \text{——— (1)}$$

where, predicted values

$$h_w(x) = \underbrace{w_0 x_0}_{=w_0} + w_1 x_1 + w_2 x_2 + \cdots + w^M x^M$$

$w_0, w_1, \ldots, w_M$ = model parameters
w_0 = bias term or intercept term

$$\boxed{h_w(x) = \sum_{j=0}^{M} w_j x^j}$$

Given

minimizing the cost function w/o regularization,

$\frac{\partial J}{\partial w} = 0$ gives

$$\boxed{\sum_{j=0}^{M} A_{ij} w_j = T_i} \quad \text{——— (2)}$$

where

$$\boxed{\begin{aligned} A_{ij} &= \sum_{n=1}^{N} x_n^{i+j} \\ T_i &= \sum_{n=1}^{N} x_n^{i} t_n \end{aligned}} \quad \text{——— (3)}$$

problem: minimize the function, λ is shrinkage

penalized residual sum of squares (PRSS)

$$J(W) = \underbrace{\frac{1}{2N} \sum_{n=1}^{N} \left(h_w(x_n) - t_n\right)^2}_{\frac{1}{2}\cdot SSE} + \underbrace{\frac{\lambda}{2} \|w\|^2}_{\text{L2 regularizer or shrinkage penalty}}$$

$(h_n - t_n)^2$

solution: The hypothesis is

$$h_w(x_n) = \sum_{j=0}^{M} w_j x_n^j$$

$n = 0, \dots, N \rightarrow N$ samples $x_1, x_2, \dots, x_N$

$j = 0, \dots, M \rightarrow$ M features $w_0, w_1, \dots w_M$

bias w_0, tumor-size w_1, age w_2

$t =$ target (t is y) (price)

we can write cost function as,

$$\boxed{J(W) = \frac{1}{2N} \sum_{n=1}^{N} \left(\sum_{j=0}^{M} w_j x_n^j - t_n \right)^2 + \frac{\lambda}{2} \sum_{j=0}^{M} w_j^2}$$

we want to minimize the cost function $J(w)$ w.r.t weights w_i,

$$\nabla_{w_i} J(w) = 0$$

$$\frac{\partial}{\partial w_i} J(w_i) = 0$$

$$0 \cdot N = N \frac{1}{2N} 2 \sum_{n=1}^{N} \left(\sum_{j=0}^{M} w_j x_n^j - t_n \right) \cdot x_n^i + \frac{\lambda}{2} \cdot 2 w_i \cdot N$$

$$0 = \sum_{n=1}^{N} \sum_{j=0}^{M} w_j x_n^{i+j} - \sum_{n=1}^{N} t_n x_n^i + \lambda N w_i$$

or, $\sum_{j=0}^{M} \left(\sum_{n=1}^{N} x_n^{i+j} \right) w_i + \lambda N w_i = \sum_{n=1}^{N} t_n x_n^{i}$

$$\Rightarrow \boxed{\sum_{j=0}^{M} A_{ij} w_j + \lambda N w_i = T_i}$$ # Answer

where,

$$A_{ij} = \sum_{n=1}^{N} x_n^{i+j}$$

$$T_i = \sum_{n=1}^{N} t_n x_n^{i}$$

2

QN1.2 probability Theory

Ex 1.3, page 58 in PRML > By Bishop

Red R: 3a 4o 3l

blue B : 1a 1o 0l

Green G : 3a 3o 4l

$p(r) = 0.2$ $p(b) = 0.2$ $p(g) = 0.6$

(a) If a piece of fruit is removed from the box, then what is the prob of selecting an apple?

(b) If we observe that the selected fruit is in fact an orange, what is the prob that it came from the green box?

Solution:

(a)

Red	Blue	Green
3a	1a	3a
4o	1o	3o
3l	0l	4l
$N_R = 10$	$N_B = 2$	$N_G = 10$

$N = N_R + N_B + N_G$
$= 10 + 2 + 10$
$= 22$

$P(a|R) = \frac{3}{10} = 0.3$ $P(a|B) = \frac{1}{2} = 0.5$ $P(a|G) = \frac{3}{10} = 0.3$

$P(o|R) = \frac{4}{10} = 0.4$ $P(o|B) = \frac{1}{2} = 0.5$ $P(o|G) = \frac{3}{10} = 0.3$

from figure

$P(R) = 0.2$ $P(B) = 0.2$ $P(G) = 0.6$ given

Implementation

QN 3.C

Given objective function for linear regression with polynomial basis functions is,

cost function
$$\boxed{J(\vec{w}) = \frac{1}{2N} \sum_{n=1}^{N} [h(x_n, \vec{w}) - t_n]^2 + \frac{\lambda}{2} \|\vec{w}\|^2} \quad \text{—— } (*)$$

Now, we have to find the vectorized solution of w which minimizes cost function J.

For this,

we first write hypothesis in terms of design matrix X and weight vector W.

$$h = XW \quad \text{—— } (1)$$

Then, eqn (*) can be written as,

$$\boxed{J = \frac{1}{2N} (XW - t)^T (XW - t) + \frac{\lambda}{2} W^T W}$$

we minimize this equation w.r.t. W,

$$0 = \frac{\partial J}{\partial W}$$

$$= \frac{1}{2N} \cdot 2X^T (XW - t) + \frac{\lambda}{2} \cdot 2W$$

Here we have used matrix derivative formula.

$$\frac{d}{dx}(x^T x) = 2x$$

$$\frac{d}{dx}(Ax+b) = 2A^T(Ax+b)$$

A, x, b are matrices

FORMULA

Then,

$$0 = X^T X W - X^T t + \lambda N M$$

$$X^T t = (X^T X + \lambda N) W$$

$$\boxed{W = (\lambda N I + X^T X)^{-1} (X^T t)}$$

Answer

Note that $(X^T X)^{-1} X^T$ is called Moore-penrose pseudoinverse of any shape matrix X.

~~eg~~

$$X^T X = (4,50)(50,4) = (4,4) = I_{4,4}$$

$$X^T t = (4,50)(50,1) = (4,1)$$

$$w = (4,4)(4,1) = (4,1) = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_M \end{bmatrix}$$

$$w^T = [w_0 \; w_1 \; w_2 \; w_M]_{1 \times (M+1)}$$

$$I_{4,4} = \begin{bmatrix} 1 & & & 0 \\ & 1 & & \\ & & 1 & \\ 0 & & & 1 \end{bmatrix}$$

set $I[0][0] = 0$

$$I = \begin{bmatrix} 0 & & & 0 \\ & 1 & & \\ & & 1 & \\ 0 & & & 1 \end{bmatrix}$$

we donot regularize bias term.

if we regularize bias term, it will shift the mean of target vector. Then adding const value to all the target DOES NOT hypothesize similarly shifted hypothesis.

I-1 20

I-2 20

I-3 40