exercise-01

October 25, 2018

Name	Matriknr.	Studiengang
Zhang.Guangde	4200165	Msc. Informatik
Lan.Songnian	4055640	Msc. Informatik
Yi.Zixin	4126223	Msc. Informatik

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import csv
        import pandas as pd
        %matplotlib inline
```

1 Refresher on Linear Algebra and Derivatives

- (a) Let A be a 3×4 matrix and B a 3×2 matrix, what is the size of $A^T B$.

Answer:
As we known A is a 3×4 matrix, so the transpose of A is a 4×3. Then the size of $A^T B$ is a 4×2 matrix.

$$\because A = \begin{Bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{Bmatrix} \text{ and } B = \begin{Bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{Bmatrix}$$

$$\therefore A^T = \begin{Bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \\ a_{14} & a_{24} & a_{34} \end{Bmatrix}$$

$$A^T B = \begin{Bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \\ a_{14} & a_{24} & a_{34} \end{Bmatrix} \begin{Bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{Bmatrix}$$

$$= \begin{Bmatrix} \sum_{i=1}^{3} \sum_{j=1}^{3} a_{i1} b_{j1} & \sum_{i=1}^{3} \sum_{j=1}^{3} a_{i1} b_{j2} \\ \sum_{i=2}^{3} \sum_{j=1}^{3} a_{i1} b_{j1} & \sum_{i=2}^{3} \sum_{j=1}^{3} a_{i1} b_{j2} \\ \sum_{i=3}^{3} \sum_{j=1}^{3} a_{i1} b_{j1} & \sum_{i=3}^{3} \sum_{j=1}^{3} a_{i1} b_{j2} \\ \sum_{i=4}^{3} \sum_{j=1}^{3} a_{i1} b_{j1} & \sum_{i=4}^{3} \sum_{j=1}^{3} a_{i1} b_{j2} \end{Bmatrix}$$

```
In [2]: A = np.ones(shape=(3,4))
        B = np.ones(shape=(3,2))
```

```
C = A.T.dot(B)
print("A.T:{} x B:{}=C:{}".format(A.T.shape, B.shape, C.shape))
```

```
A.T:(4, 3) x B:(3, 2)=C:(4, 2)
```

- (b) Let $x \in R^n$ be a column vector (vectors are always columns for us) and A a mn matrix. What is the size of Ax.

Answer:
From above we know that , x is a column vector ,also $n1$ matrix. A is a mn matrix. So the size of Ax is $m1$ matrix.

assume $x = \left\{x_1 \dots x_n\right\}^T$ and $A = \begin{Bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} & A_{m,2} & \dots & A_{m,n} \end{Bmatrix}$

$$Ax = \begin{Bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} & A_{m,2} & \dots & A_{m,n} \end{Bmatrix} \begin{Bmatrix} x_1 \\ \vdots \\ x_n \end{Bmatrix}$$
$$= \begin{Bmatrix} \sum_{i=1}^n A_{1,i} x_i \\ \vdots \\ \sum_{i=1}^n A_{m,i} x_i \end{Bmatrix}$$

```
In [3]: x = np.ones(shape=(4,1))
        A = np.ones(shape=(5,4))
        y = A.dot(x)
        print("A:{} * x:{} = y:{}".format(A.shape, x.shape, y.shape))
```

```
A:(5, 4) * x:(4, 1) = y:(5, 1)
```

- (c) What is the derivative of $f(x) = (2x+y)^2$ w.r.t. x:$\frac{\partial}{\partial x} f(x)$

$$\begin{aligned} \frac{\partial}{\partial x} f(x) &= \frac{\partial}{\partial x}(2x+y)^2 \\ &= 2(2x+y).2 \\ &= 8x+4y \end{aligned}$$

- (d) Given $f(x) = g(x^2)$ where $g(x) = (x+y)^2$, what is $\frac{\partial}{\partial x} f(x)$

$$\begin{aligned} \frac{\partial}{\partial x} f(x) &= \frac{\partial}{\partial x} g(x^2) \\ &= \left(\frac{\partial}{\partial x}(x^2+y)^2\right) * \left(\frac{\partial}{\partial x}(x^2)\right) \\ &= (2(x+y)) * (2x) \\ &= 4x^3 + 4xy \end{aligned}$$

2 Multivariable Calculus

Recall that a matrix $A \in R^{n\times n}$ is symmetric if $A^T = A$, that is, $A_{ij} = A_{ji}$ for all i, j. Also recall the gradient $\nabla f(x)$ of a function $f : R^n R$ is the $n$vector of partial derivatives

$$\nabla f(x) = \begin{Bmatrix} \frac{\partial}{\partial x_1} f(x) \\ ... \\ \frac{\partial}{\partial x_1} f(x) \end{Bmatrix}$$

where

$$x = \begin{Bmatrix} x_1 \\ ... \\ x_n \end{Bmatrix}$$

The hessian $\nabla^2 f(x)$ is the $n \times n$ symmetric matrix of twice partial derivatives,

$$\nabla^2 f(x) = \begin{Bmatrix} \frac{\partial^2}{\partial x_1^2} f(x) & \frac{\partial^2}{\partial x_1 \partial x_2} f(x) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_n} \\ \frac{\partial^2}{\partial x_2 \partial x_1} f(x) & \frac{\partial^2}{\partial x_2^2} f(x) & \cdots & \frac{\partial^2}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_n \partial x_1} f(x) & \frac{\partial^2}{\partial x_n \partial x_2} f(x) & \cdots & \frac{\partial^2}{\partial x_n^2} \end{Bmatrix}$$

- a) Let $f(x) = \frac{1}{2} x^T A x + b^T x$, where a is a sysmmetric matrix and $b \in R^n$ is a vector. What is $\nabla f(x)$?

$\because A \in R^{n\times n}$, $A_{ij} = A_{ji}$ and $f(x) = \frac{1}{2} x^T A x + b^T x$
$\therefore$

$$\begin{aligned}
\frac{\partial}{\partial x_i} f(x) &= \frac{\partial}{\partial x_i} (\frac{1}{2} x^T A x + b^T x) \\
&= \frac{\partial}{\partial x_i} (\frac{1}{2} \begin{Bmatrix} x_1 \cdots x_n \end{Bmatrix} \begin{Bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \cdots & A_{n,n} \end{Bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} + \begin{Bmatrix} b_1 b_2 ... b_n \end{Bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}) \\
&= \frac{\partial}{\partial x_i} (\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i x_j + \sum_{j=1}^{n} b_j x_i) \\
&= \frac{\partial}{\partial x_i} (\frac{1}{2} (\sum_{i=1}^{n} A_{ii} x_i^2 + \sum_{i=1, i\neq j}^{n} \sum_{j=1, i\neq j}^{n} A_{ij} x_i x_j) + \sum_{j=1}^{n} b_j x_i) \\
&= A_{ii} x_i + \sum_{j=1, i\neq j}^{n} A_{ij} x_i + \sum_{j=1}^{n} b_j \\
&= \sum_{j=1}^{n} A_{ij} x_i + \sum_{j=1}^{n} b_j
\end{aligned}$$

$\therefore$

$$
\begin{aligned}
\nabla f(x) &= \begin{Bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{Bmatrix} \\
&= \begin{Bmatrix} \sum_{j=1}^{n} A_{1j} x_1 + \sum_{j=1}^{n} b_j \\ \vdots \\ \sum_{j=1}^{n} A_{nj} x_n + \sum_{j=1}^{n} b_j \end{Bmatrix} \\
&= Ax + b
\end{aligned}
$$

- b) Let $f(x) = g(h(x))$, where $g : R \to R$ is differentiable and $h : R^n \to R$ is differentiable. What is $\nabla f(x)$?

$$
\begin{aligned}
\nabla f(x) &= \nabla g(h(x)) \\
&= \frac{\partial g(h(x))}{\partial h(x)} \cdot \frac{\partial h(x)}{\partial x} \\
&= \frac{\partial g(h(x))}{\partial h(x)} \cdot \begin{Bmatrix} \frac{\partial h(x)}{\partial x_1} \\ \vdots \\ \frac{\partial h(x)}{\partial x_i} \\ \vdots \\ \frac{\partial h(x)}{\partial x_n} \end{Bmatrix}
\end{aligned}
$$

- c) Let $f(x) = \frac{1}{2} x^T x + b^T x$ as in a. What is $\nabla^2 f(x)$?

From (a) knows that $\nabla f(x) = Ax + b$

$\therefore$

$$
\begin{aligned}
\nabla^2 f(x) &= \nabla(\nabla f(x)) \\
&= \nabla(Ax + b) \\
&= \begin{Bmatrix} \frac{\partial (Ax+b)}{\partial x_1} \\ \vdots \\ \frac{\partial (Ax+b)}{\partial x_n} \end{Bmatrix} \\
&= A
\end{aligned}
$$

- d) Let $f(x) = g(a^T x)$, where $g : R \to R$ is continuously differentiable and $a \in R^n$ is a vector. What are $\nabla f(x)$ and $\nabla^2 f(x)$?

assume that $h(x) = a^T x$

$$
\begin{aligned}
\frac{\partial h(x)}{\partial x} &= \frac{\partial a^T x}{\partial x} \\
&= a
\end{aligned}
$$

∴

$$
\begin{aligned}
\nabla f(x) &= \nabla g(h(x)) \\
&= \frac{\partial g(h)}{\partial h} \cdot \frac{\partial h(x)}{\partial x} \\
&= \frac{\partial g(h)}{\partial h} \cdot \frac{\partial a^T x}{\partial x} \\
&= \frac{\partial g(h)}{\partial h} \cdot a \\
\nabla^2 f(x) &= \nabla(\nabla f(x)) \\
&= \nabla(\frac{\partial g(h)}{\partial h} \cdot a) \\
&= 0
\end{aligned}
$$

3 Hands On

```
In [20]: with open("diabetes.txt","r") as file:
             reader = csv.reader(file, delimiter=' ')
             table = np.asarray([row for row in reader], dtype=np.float)

         # table.sort(axis=0)
```

- run the following experiment by taking the first 200 data points as training and the next 200 points as validation set

```
In [164]: training_set = table[:200,:] #taking the first 200 data points as training set
          validation_set = table[200:400,:] # taking the next 200 points as validation set
          # training set
          x_training =training_set[:,:10]
          y_training =training_set[:,10:] #for taining_set choose x_matrix und y_matrix  200ha
          # validation set
          x_validation=validation_set[:,:10]
          y_validation =validation_set[:,10:]  #for validation_set choose x_matrix und y_matri
```

- train a least squares regression model without regularization (using matrix operations)

```
In [165]: # by solve method
          lhs = np.dot(x_training.T, x_training)
          rhs = np.dot(x_training.T, y_training)
          w = np.linalg.solve(lhs, rhs)
          print("solve is: {}.T".format(w.T))
          # by matrix operate
          w = np.mat(x_training.T.dot(x_training)).I.dot(x_training.T).dot(y_training)
          # print("solve is: {}.T".format(w.T))
```

```
solve is: [[ -417.97043881  -239.65634372   573.7328081    164.48588506
  -2438.3248339    1673.911313     688.89188963    98.98921971
   1287.00977365   102.9581032 ]].T
```

- for each regularization parameters $\lambda \in \{2^{-20}, 2^{-19}, ...2^{10}\}$ train a ridge regression model with regularization (using matrix operations)

```
In [181]: # parameters $\lamda$
          h = np.asarray([i for i in range(-20, 10, 1)])
          lamda = np.exp2(h)
          d = 10
          # calculate w
          w = []
          for l in lamda:
              ww = np.linalg.inv(np.sum(x_training.T.dot(x_training))
                                  + np.eye(M=d,N=d).dot(l)).dot(x_training.T).dot(y_training)
              w.append(ww)
          w = np.asarray(w)
```

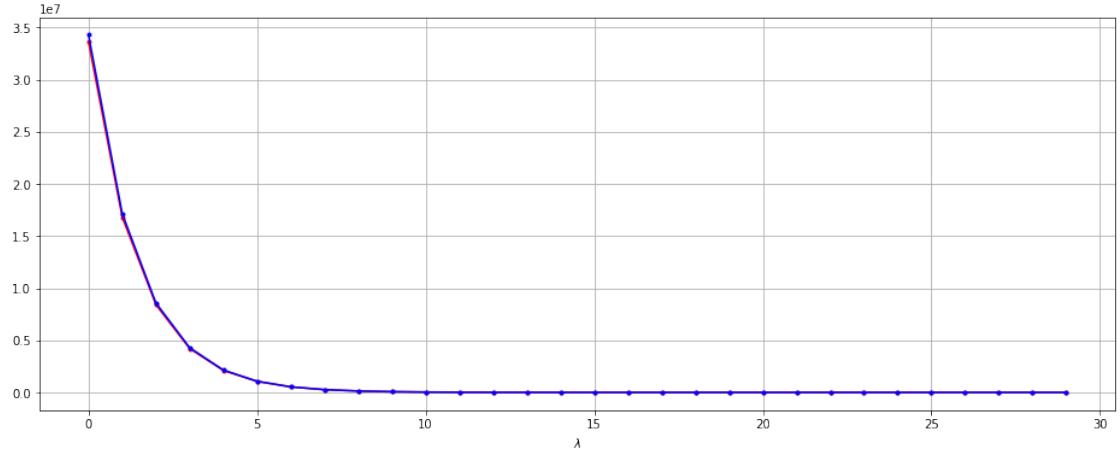
- evaluate the trained models on the validation set

```
In [186]: err_tra = []
          err_tst = []
          fig = plt.figure(figsize=(16,6))

          # error of training dataset
          for i, ww in enumerate(w):
              diff = np.subtract(ww.T.dot(x_training.T), y_training)
              err_tra.append(np.std(diff));
          err_tra = np.asarray(err_tra);

          # error of test dataset
          for i, ww in enumerate(w):
              diff = np.subtract(ww.T.dot(x_validation.T), y_validation)
              err_tst.append(np.std(diff));
          err_tst = np.asarray(err_tst);

          plt.plot(err_tra, 'r.-');
          plt.plot(err_tst, 'b.-');
          plt.grid(True);
          plt.xlabel('$\lambda$');
```



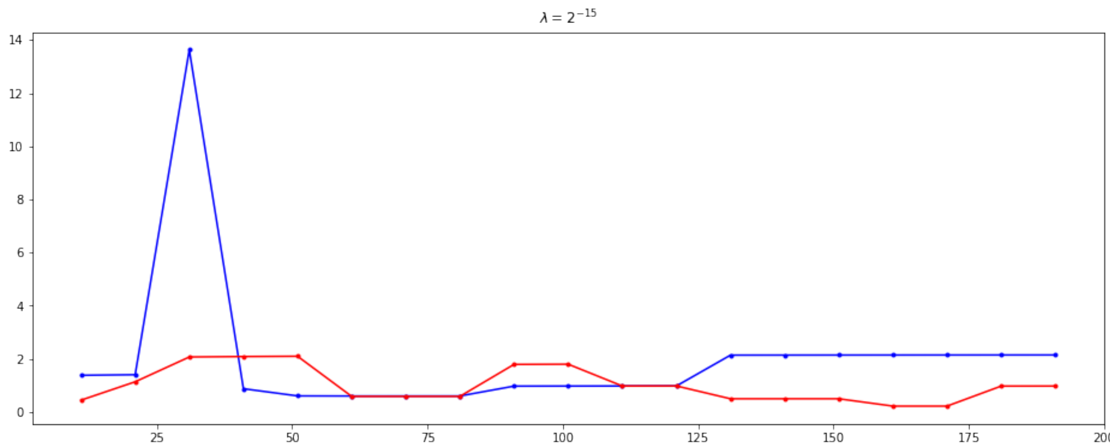
- plot the results in a style that you fnd most appropriate/informative including the selected regularization parameter.

```
In [187]: fig = plt.figure(figsize=(16,6))

          kb = []
          vkb = []
          oplamda = 2**(-15);
          idx = range(1, 201, 10)

          for n in idx:
              x = x_training[:n, :]
              xxt = x.dot(x.T)
              xxtl = x.dot(x.T) + np.dot(np.eye(M=n, N=n), oplamda)
          #         k.append(np.divide(np.argmax(xxt), np.argmin(xxt)))
              kb.append(np.divide(np.argmax(xxtl), np.argmin(xxtl)))
              vx = x_validation[:n, :]
              vxxt = vx.dot(vx.T)
              vxxtl = vx.dot(vx.T) + np.dot(np.eye(M=n, N=n), oplamda);
              vkb.append(np.divide(np.argmax(vxxtl), np.argmin(vxxtl)))
          plt.plot(idx, kb, 'b.-');
          plt.plot(idx, vkb, 'r.-');
          plt.title('$\lambda=2^{-15}$');
```

```
/media/self/develop/branch.git/works/uni/publics/runtime/python3.7/lib/python3.7/site-packages/
  del sys.path[0]
/media/self/develop/branch.git/works/uni/publics/runtime/python3.7/lib/python3.7/site-packages/
```


$\lambda = 2^{-15}$
14
12
10
8
6
4
2
0
25
50
75
100
125
150
175
200