Homework 1

2020 Spring CSCI 5525: Machine Learning

Due on Fabruary 16th 11:59pm

Homework Policy. (1) You are encouraged to collaborate with your classmates on homework problems, but each person must write up the final solutions individually. You need to fill in above to specify which problems were a collaborative effort and with whom. (2) Regarding online resources, you should **not**:

- Google around for solutions to homework problems,
- Ask for help on online.
- Look up things/post on sites like Quora, StackExchange, etc.

Submission. Submit a PDF using this LaTeX template for written assignment part and submit Python jupyter or Colab python notebooks (.ipynb) for all programming part. You should upload all the files on Canvas.

Written Assignment

Instruction. For each problem, you are required to write down a full mathematical proof to establish the claim.

Problem 1. Two helpful matrices.

Let us first recall the notations in linear regression. The design matrix and the response vector are are defined as:

$$A = \begin{bmatrix} \leftarrow x_1^\intercal \rightarrow \\ \vdots \\ \leftarrow x_n^\intercal \rightarrow \end{bmatrix} \qquad \mathbf{b} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

For this problem, we will assume the covariance matrix $A^\intercal A$ is invertible, and so $(A^\intercal A)^{-1}$ is well-defined (**Clearly mention the properties of matrix operations used while solving**).

Problem 1.1. The residual matrix. For any weight vector $\mathbf{w}$, let us define the vector of least squares residuals as

$$e = \mathbf{b} - A\mathbf{w}$$

Now if $\mathbf{w}$ is the least square solution given by $\mathbf{w} = (A^\intercal A)^{-1}A^\intercal \mathbf{b}$, we can rewrite e as

$$e = \mathbf{b} - A(A^\intercal A)^{-1}A^\intercal \mathbf{b} = \left(I - A(A^\intercal A)^{-1}A^\intercal\right)\mathbf{b}$$

Now let $M = \left(I - A(A^\intercal A)^{-1}A^\intercal\right)$. Show that

- M is symmetric (i.e. $M = M^\intercal$). (**2 points**)
- M is idempotent (i.e. $M^2 = M$). (**2 points**)
- $MA = 0$. (**1 point**)

Your answer.

For the first question:

$$\begin{aligned} M^\intercal &= \left(I - A(A^\intercal A)^{-1}A^\intercal\right)^\intercal \\ &= I^\intercal - \left[A(A^\intercal A)^{-1}A^\intercal\right]^\intercal \\ &= I - A\left[A(A^\intercal A)^{-1}\right]^\intercal \\ &= I - A\left[(A^\intercal A)^{-1}\right]^\intercal A^\intercal \end{aligned}$$

$A^\intercal A$ is a diagonal matrix, so is $(A^\intercal A)^{-1}$, thus $[(A^\intercal A)^{-1}]^\intercal = (A^\intercal A)^{-1}$. We have

$$M^\intercal = I - A\left[(A^\intercal A)^{-1}\right]^\intercal A^\intercal = I - A(A^\intercal A)^{-1}A^\intercal = M$$

For the second question:

$$\begin{aligned} M^2 &= \left(I - A(A^\intercal A)^{-1}A^\intercal\right)^2 \\ &= \left(I - A(A^\intercal A)^{-1}A^\intercal\right) * \left(I - A(A^\intercal A)^{-1}A^\intercal\right) \\ &= I - 2A(A^\intercal A)^{-1}A^\intercal + A(A^\intercal A)^{-1}A^\intercal A(A^\intercal A)^{-1}A^\intercal \\ &= I - 2A(A^\intercal A)^{-1}A^\intercal + A(A^\intercal A)^{-1}A^\intercal \\ &= I - A(A^\intercal A)^{-1}A^\intercal = M \end{aligned}$$

For the third question:

$$\begin{aligned} MA &= \left(I - A(A^\intercal A)^{-1}A^\intercal\right) A \\ &= A - A(A^\intercal A)^{-1}A^\intercal A \\ &= A - A = 0 \end{aligned}$$

Problem 1.2. The hat matrix. Using the residual maker, we can derive another matrix, the hat matrix or projection matrix $P = I - M = A(A^\intercal A)^{-1}A^\intercal$. Note that the predicted value by the least squares solution is given by $P\mathbf{b}$. Show that

- P is symmetric. (**1 point**)
- P is idempotent. (**1 point**)

Your answer.

For the first question:

$$\begin{aligned} P^\intercal &= \left(A(A^\intercal A)^{-1}A^\intercal\right)^\intercal \\ &= A\left[A(A^\intercal A)^{-1}\right]^\intercal \\ &= A\left[(A^\intercal A)^{-1}\right]^\intercal A^\intercal \end{aligned}$$

$A^\intercal A$ is a diagonal matrix, so is $(A^\intercal A)^{-1}$, thus $[(A^\intercal A)^{-1}]^\intercal = (A^\intercal A)^{-1}$. We have

$$\begin{aligned} P^\intercal &= A\left[(A^\intercal A)^{-1}\right]^\intercal A^\intercal \\ &= A(A^\intercal A)^{-1}A^\intercal = P \end{aligned}$$

For the second question:

$$\begin{aligned} P^2 &= A(A^\intercal A)^{-1}A^\intercal A(A^\intercal A)^{-1}A^\intercal \\ &= A(A^\intercal A)^{-1}A^\intercal = P \end{aligned}$$

Problem 2. Gradient of conditional log-likelihood.

For any $a \in \mathbb{R}$, let $\sigma(a) = \frac{1}{1+\exp(-a)}$. For each example $(x_i, y_i) \in \mathbb{R}^d \times \{0, 1\}$, the conditional log-likelihood of logistic regression is

$$\ell(y_i \mid x_i, \mathbf{w}) = y_i \ln(\sigma(\mathbf{w}^\intercal x_i)) + (1 - y_i)\ln(\sigma(-\mathbf{w}^\intercal x_i))$$

Derive the gradient of $\ell(y_i \mid x_i, \mathbf{w})$ with respect to w_j (i.e. the j-th coordinate of $\mathbf{w}$), i.e. $\frac{\partial}{\partial w_j}\ell(y_i \mid x_i, \mathbf{w})$ (**Clearly mention the properties of derivatives used while solving**). (**6 points**)

Your answer.

The derivation for $\sigma(a)$ is:

$$\frac{\partial \sigma}{\partial a} = \frac{\exp(-a)}{(1+\exp(-a))^2} = \sigma(1-\sigma)$$

Let $a = \mathbf{w}^\intercal x_i$, with the chain rule of derivation, we have:

$$\begin{aligned} \frac{\partial \ell}{\partial w_j} &= \frac{\partial \ell}{\partial \sigma}\frac{\partial \sigma}{\partial a}\frac{\partial a}{\partial w_j} \\ &= \frac{y_i}{\sigma(a)}\sigma(a)(1-\sigma(a))x_{ij} + \frac{1-y_i}{\sigma(-a)}\sigma(-a)(1-\sigma(-a))(-x_{ij}) \\ &= y_i(1-\sigma(a))x_{ij} - (1-y_i)(1-\sigma(-a))x_{ij} \\ &= y_i(1-\sigma(a))x_{ij} - (1-y_i)\sigma(a)x_{ij} \\ &= (y_i - \sigma(a))x_{ij} \end{aligned}$$

Problem 3. Derivation of Ridge Regression Solution.

Recall that in class we claim that the solution to ridge regression ERM:

$$\min_{\mathbf{w}} \; (\|A\mathbf{w}-\mathbf{b}\|_2^2 + \lambda\|\mathbf{w}\|_2^2)$$

is $\mathbf{w}^* = (A^\intercal A + \lambda I)^{-1}A^\intercal\mathbf{b}$. Now provide a proof. (**6 points**)
(Hint: recall that $\nabla F(\mathbf{w}) = \mathbf{0}$ is a sufficient condition for $\mathbf{w}$ to be a minimizer of any convex function F. To get a full credit, you should be able to show why $A^\intercal A + \lambda I$ is invertible.) (**Clearly mention the properties of matrix calculus used while solving**)

Your answer.

We firstly approve that $A^\intercal A + \lambda I$ is invertible. Let $S = A^\intercal A + \lambda I$, apparently, S is a square matrix. Let $V \in \mathbb{R}^d$ where d is the dimension of S, for any V not zero vector, we have:

$$V^\intercal S V = V^T(A^\intercal A + \lambda I)V = V^\intercal A^\intercal A V + \lambda V^\intercal V = \|AV\|_2^2 + \lambda\|V\|_2^2 > 0$$

S is positive definite matrix, so S is invertible.

Then, Let $E = \|A\mathbf{w}-\mathbf{b}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$, we take derivation of E to $\mathbf{w}$, and let the derivation equals 0:

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}}(A\mathbf{w}-b)^\intercal(A\mathbf{w}-b) + 2\lambda\mathbf{w} \\ &= \frac{\partial \mathbf{w}^\intercal A^\intercal A\mathbf{w}}{\partial \mathbf{w}} - \frac{\partial \mathbf{w}^\intercal A^\intercal b}{\partial \mathbf{w}} - \frac{\partial b^\intercal A\mathbf{w}}{\partial \mathbf{w}} + 2\lambda\mathbf{w} \\ &= 2A^\intercal A\mathbf{w} - 2A^\intercal b + 2\lambda\mathbf{w} = 0\end{aligned}$$

Thus, we get:

$$\begin{aligned}A^\intercal A\mathbf{w} + \lambda\mathbf{w} &= A^\intercal b \\ w &= (A^\intercal A + \lambda I)^{-1}A^\intercal b\end{aligned}$$

Problem 4. Minimizing a Squared Norm Plus an Affine Function.

A generalization of the least squares problem adds an affine function to the least squares objective,

$$\min_{\mathbf{w}} \; \|A\mathbf{w}-\mathbf{b}\|_2^2 + \mathbf{c}^\intercal\mathbf{w} + d$$

where $A \in \mathbb{R}^{m\times n}, \mathbf{w} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m, \mathbf{c} \in \mathbb{R}^n, d \in \mathbb{R}$. Assume the column of A are linearly independent. This generalized problem can be solved by reducing it to a standard least squares problem, using a trick called *completing the square.*

Show that the objective of the problem above can be expressed in the form

$$\|A\mathbf{w}-\mathbf{b}\|_2^2 + \mathbf{c}^\intercal\mathbf{w} + d = \|A\mathbf{w}-\mathbf{b}+\mathbf{f}\|_2^2 + g$$

where $\mathbf{f} \in \mathbb{R}^m, g \in \mathbb{R}$.It follows that we can solve the generalized least squares problem by minimizing $\|A\mathbf{w} - (\mathbf{b}-\mathbf{f})\|_2^2$

(Hint: Express the norm squared term on the right-hand side as $\|(A\mathbf{w}-\mathbf{b})+\mathbf{f})\|_2^2$ and expand it. Then argue that the equality above holds provided $2A^\intercal\mathbf{f} = \mathbf{c}$. One possible choice is $f = \frac{1}{2}(A^\dagger)^\intercal\mathbf{c}$.) (You must justify these statements.) (**6 point**)

Your answer.

$$\begin{aligned}\|A\mathbf{w}-\mathbf{b}+\mathbf{f}\|_2^2+g &= (A\mathbf{w}-b+f)^\intercal(A\mathbf{w}-b+f)+g\\ &= (A\mathbf{w}-b)^\intercal(A\mathbf{w}-b)+(A\mathbf{w}-b)^\intercal f+f^\intercal(A\mathbf{w}-b)+f^\intercal f+g\\ &= \|A\mathbf{w}-b\|_2^2+\mathbf{w}^\intercal A^\intercal f-b^\intercal f+f^\intercal A\mathbf{w}-f^\intercal b+f^\intercal f+g\end{aligned}$$

Obviously, the dimensions of $f^\intercal f, b^\intercal f, \mathbf{w}^\intercal A^\intercal f$ are $1*1$, so $\mathbf{w}^\intercal A^\intercal f=(\mathbf{w}^\intercal A^\intercal f)^\intercal=f^\intercal A\mathbf{w}$. Thus, we get the equation:

$$\begin{aligned}\|A\mathbf{w}-\mathbf{b}+\mathbf{f}\|_2^2+g &= \|A\mathbf{w}-b\|_2^2+\mathbf{w}^\intercal A^\intercal f-b^\intercal f+f^\intercal A\mathbf{w}-f^\intercal b+f^\intercal f+g\\ &= \|A\mathbf{w}-b\|_2^2+2f^\intercal A\mathbf{w}+f^\intercal f-b^\intercal f-f^\intercal b+g\end{aligned}$$

Let $c=2A^\intercal f, d=f^\intercal f-b^\intercal f-f^\intercal b+g$, we get $\|A\mathbf{w}-\mathbf{b}\|_2^2+\mathbf{c}^\intercal\mathbf{w}+d=\|A\mathbf{w}-\mathbf{b}+\mathbf{f}\|_2^2+g$

Problem 5. Iterative Method for Least Squares Problem.

In this exercise we explore an iterative method, due to the mathematician Lewis Richardson, that can be used to compute $\hat{\mathbf{w}}=A^\dagger\mathbf{b}$, we define $\mathbf{w}^{(1)}=0$ and for $k=1,2,3,...,$

$$\mathbf{w}^{(k+1)}=\mathbf{w}^{(k)}-\mu A^\intercal(A\mathbf{w}^{(k)}-\mathbf{b})$$

where μ is a positive parameter, and the superscripts denote the iteration number.This defines a sequence of vectors that converge to $\hat{\mathbf{w}}$ provided μ is not too large; The iteration is terminated when $A^\intercal(A\mathbf{w}^{(k)}-\mathbf{b})$ is small enough, which means the least squares optimality conditions are almost satisfied. To implement the method we only need to multiply vectors by A and by $A^\intercal$. If we have efficient methods for carrying out these two matrix-vector multiplications, this iterative method can be faster. Iterative methods are often used for very large scale least squares problems.

(a) Show that if $\mathbf{w}^{(k+1)}=\mathbf{w}^{(k)}$, we have $\mathbf{w}^{(k)}=\hat{\mathbf{w}}$. (**5 points**)

(b) Express the vector sequence $x^{(k)}$ as a linear dynamical system with constant dynamics matrix and offset, i.e., in the form $\mathbf{w}^{(k+1)}=F\mathbf{w}^{(k)}+g$. (**3 points**)

Your answer.

(a) if $\mathbf{w}^{(k+1)}=\mathbf{w}^{(k)}$, $\mu A^\intercal(A\mathbf{w}^{(k)}-\mathbf{b})=0$, i.e.

$$A^\intercal A\mathbf{w}^{(k)}=A^\intercal\mathbf{b}$$

With the SVD, we have $A=USV^\intercal$, S is a full rank diagonal matrix, V and U are orthogonal matrices, i.e. $V^\intercal V=I$ and $U^\intercal U=I$, so the equation above is equivalent to:

$$\begin{aligned}&VS^\intercal U^\intercal USV^\intercal\mathbf{w}^{(k)}=VS^\intercal U^\intercal\mathbf{b}\\ \Longleftrightarrow\ &VS^\intercal SV^\intercal\mathbf{w}^{(k)}=VS^\intercal U^\intercal\mathbf{b}\end{aligned}$$

Multiply both sides of the equation by $VS^{-1}S^{-1}V^\intercal$, we have $\mathbf{w}^{(k)} = \hat{\mathbf{w}}$, v_i and u_i refer to the column vector of V and U respectively, s_i is the diagonal element of S:

$$\begin{aligned}
VS^{-1}S^{-1}V^\intercal VS^\intercal SV^\intercal \mathbf{w}^{(k)} &= \mathbf{w}^{(k)} \\
&= VS^{-1}S^{-1}V^\intercal VS^\intercal U^\intercal \mathbf{b} \\
&= VS^{-1}U^\intercal \mathbf{b} \\
&= \sum_{i=1}^{r} \frac{1}{s_i} v_i u_i^\intercal \mathbf{b} \\
&= A^\dagger \mathbf{b} = \hat{\mathbf{w}}
\end{aligned}$$

(b)

$$\begin{aligned}
\mathbf{w}^{(k+1)} &= \mathbf{w}^{(k)} - \mu A^\intercal (A\mathbf{w}^{(k)} - \mathbf{b}) \\
&= \mathbf{w}^{(k)} - \mu A^\intercal A \mathbf{w}^{(k)} + \mu A^\intercal \mathbf{b} \\
&= (I - \mu A^\intercal A)\mathbf{w}^\intercal + \mu A^\intercal \mathbf{b}
\end{aligned}$$

So we have:

$$\begin{aligned}
F &= I - \mu A^\intercal A \\
g &= \mu A^\intercal \mathbf{b}
\end{aligned}$$

Programming Assignment

Instruction. For each problem, you are required to report descriptions and results in the PDF and submit code as python file (.py) (as per the question).

- **Python** version: Python 3.
- Please follow PEP 8 style of writing your Python code for better readability in case you are wondering how to name functions & variables, put comments and indent your code
- **Packages allowed**: numpy, pandas, matplotlib
- Please PROPERLY COMMENT your code in order to have utmost readability
- Please provide the required functions for each problem.
- There shall be NO runtime errors involved.
- There would be PENALTY if any of the above is not followed
- **Submission**: For programming parts, **ONLY THE PYTHON 3 NOTEBOOKS WILL BE ACCEPTED**

Problem 6. Iterative Method for Least Squares Problem (Cont'd).

For this problem, you will implement Richardson algorithm introduced in Problem 5 and report the required graphs. Please submit a Python script file (name hw1_lsq_iter.py)

- Generate a random 20 × 10 matrix A and 20-vector b, and compute $\hat{\mathbf{w}} = A^{\dagger}\mathbf{b}$. Run the Richardson algorithm with $\mu = \frac{1}{\|A\|^2}$ for 500 iterations, and plot $\|\mathbf{w}^{(k)} - \hat{\mathbf{w}}\|$ to verify that $x^{(k)}$ appears to be converging to $\hat{x}$. Please properly analyze and describe your plot. (**12 points**)
- Note: function np.linalg.lstsq is not allowed.

  To get a full credit, the main script should output the required plots with explicitly named x-y axis and the following function should be provided:

  1. $\mathbf{w}$ = lsq(A, $\mathbf{b}$) where $\mathbf{w} = \arg\min_{\mathbf{w}} \ \|A\mathbf{w} - \mathbf{b}\|_2^2$ solved by closed form.
  2. $\mathbf{w}$ = lsq_iter(A, $\mathbf{b}$) where $\mathbf{w} = \arg\min_{\mathbf{w}} \ \|A\mathbf{w} - \mathbf{b}\|_2^2$ solved by Richardson algorithm.

Your answer.

In my code, I use SVD to find the pseudoinverse. I use np.linalg.svd function and the U, V, σ obtained to calculate $\hat{\mathbf{w}}$. The plot for $\|\mathbf{w}^{(k)} - \hat{\mathbf{w}}\|$ shows in figure 1, with more iteration times, the $\|\mathbf{w}^{(k)} - \hat{\mathbf{w}}\|$ gradually goes to 0.

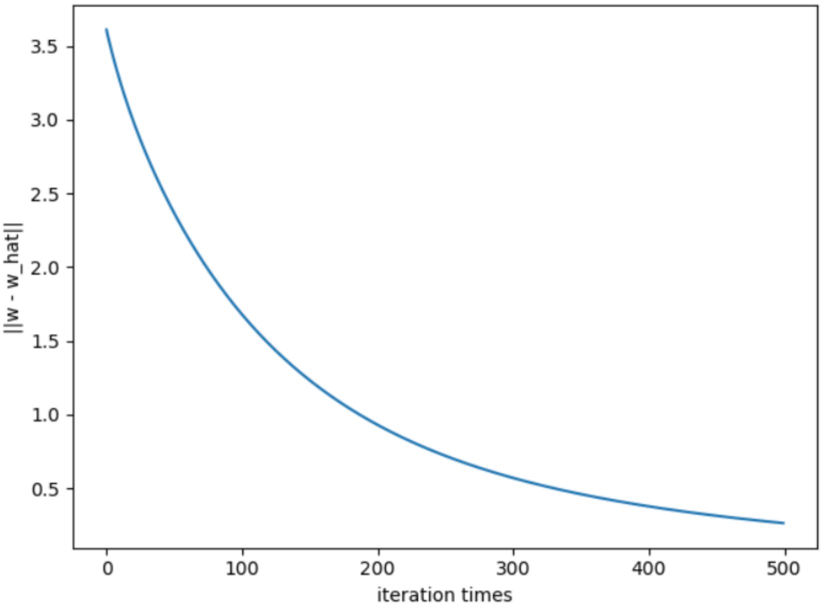


Figure 1: The convergence map

Problem 7. Logistic regression.

For this problem, you will use the IRIS dataset. Features are in the file IRISFeat.csv and labels in the file IRISlabel.csv. The dataset has 150 samples. A python script (name hw1-logistic.py) with all the steps need to be submitted.

a) (**12 Points**) Your goal is to implement logistic regression. You can design or structure your code to fulfil the requirements but to get a full credit, the main script should output the required tables or plots with explicitly named x-y axis and the following function should be provided where X being features and y target.:

   1. Cross validation: Make sure you randomly shuffle the dataset and partition it into almost equal (k=5) folds. Save each of the 5 folds into dictionary X_shuffled and y_shuffled.
      X_train, y_train, X_valid, y_valid = get_next_train_valid(X_shuffled, y_shuffled, itr) where itr is iteration number.
   2. model_weights, model_intercept = train(X_train, y_train)
   3. y_predict_class = predict(X_valid, model_weights, model_intercept)

   USE GRADIENT DESCENT to solve it. You should initialize the weights randomly to begin with.

b) (**3 Points**) At the beginning, briefly describe the approach in one paragraph along with any equations and methods used in your implementation.

c) (**5 Points**) Report the plot of training and validation set error rates (number of misclassified samples/total number of samples) and the confusion matrix for validation set from 5-fold cross validation. Explain your selection of learning rate and How does it affect the performance/training of your model?

(b) I used numpy.random.uniform function to get random $\mathbf{w}$ and $\mathbf{b}$, numpy.random.shuffle function to shuffle the data.
The loss function for the logistic regression is:

$$Loss = \sum_{i=0}^{n} \ln(1 + \exp(-(2y_i - 1)\mathbf{w}^\intercal x_i))$$

The loss derivations to $\mathbf{w}$ and $\mathbf{b}$ are:

$$\frac{\partial L}{\partial \mathbf{w}} = \sum_{i=0}^{n} \frac{-(2y_i - 1)x_i \exp(-(2y_i - 1)\mathbf{w}^\intercal x_i)}{1 + \exp(-(2y_i - 1)\mathbf{w}^\intercal x_i)}$$

$$\frac{\partial L}{\partial \mathbf{b}} = \sum_{i=0}^{n} \frac{-(2y_i - 1) \exp(-(2y_i - 1)\mathbf{w}^\intercal x_i)}{1 + \exp(-(2y_i - 1)\mathbf{w}^\intercal x_i)}$$

So the gradient descent methods are shown below, n is the number of the samples while η is the learning rate:

$$\mathbf{w} = \mathbf{w} - \frac{\eta}{n}\frac{\partial L}{\partial \mathbf{w}}$$

$$\mathbf{b} = \mathbf{b} - \frac{\eta}{n}\frac{\partial L}{\partial \mathbf{b}}$$

(c)The plot of training and validation set error rates shows in figure 2, the confusion matrices show in figure 3. I set 500 iterations for training, and the learning rates for 5 times of training are $[0.01, 0.02, 0.03, 0.04, 0.05]$, respectively.
We can find a trend that when the learning rates get larger, the error rates for training and validation sets get smaller. The reason is that when the learning rate is too small, it needs more iterations to converge.

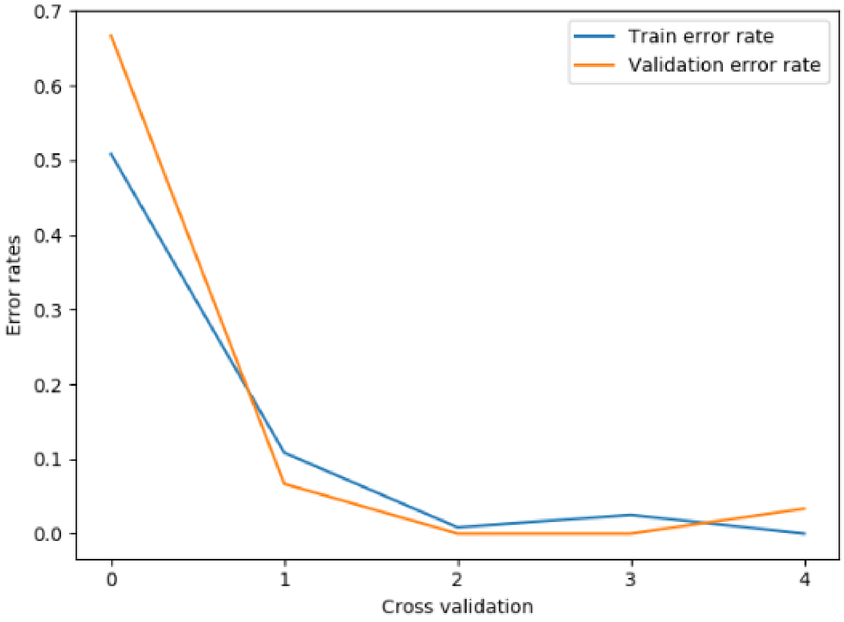
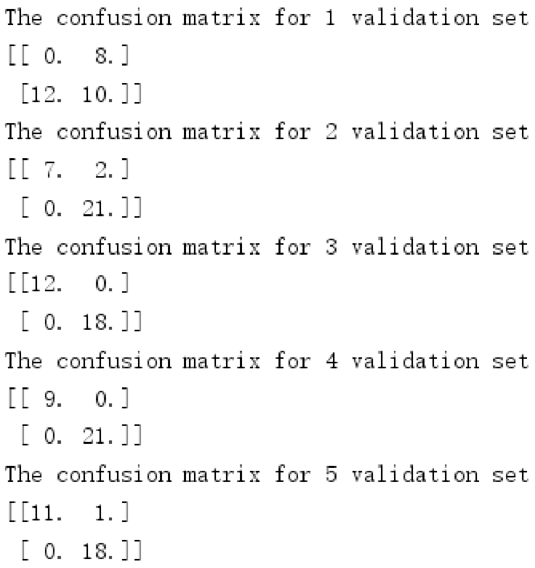


Figure 2: The error rates of training and validation set



```
The confusion matrix for 1 validation set
[[ 0.  8.]
 [12. 10.]]
The confusion matrix for 2 validation set
[[ 7.  2.]
 [ 0. 21.]]
The confusion matrix for 3 validation set
[[12.  0.]
 [ 0. 18.]]
The confusion matrix for 4 validation set
[[ 9.  0.]
 [ 0. 21.]]
The confusion matrix for 5 validation set
[[11.  1.]
 [ 0. 18.]]
```

Figure 3: The confusion matrices