**CS542 Machine Learning**

**Homework 2**

**Chih Wei Tung (U17550453)**

**3.3**

represented by a diagonal matrix

Use the summing from 1 to N

(i) Data dependent noise variation can be seen as an inverse variance parameter to a data point which modifies the precision matrix

(ii) can be seen as an effective number of replicated observation of a data point

**3.11**

(3.59):

From (3.59)

Set and

: positive definite

: positive semidefinite, nonnegative

**3.14**

,

When ,

Since is new orthonormal basis passing the same space

V, matrix that represents the function that transforms original basis to the new one

(V has an inverse)

First, transform to the new orthonormal basis

Apply to linear transformation and

Use 3.115

**3.21**

Let be real, symmetric matrix

The transformation of can be recreated by multiplying the original vector by

constant

Let be the set of orthonormal vectors

: random optimized value

Recreate matrixes and their inverse with their eigenvalues

If length of is always constant

the derivative of a vector is orthogonal to the vector

Multiplication of orthogonal vectors = 0,

3.117 can be used to derive 3.92

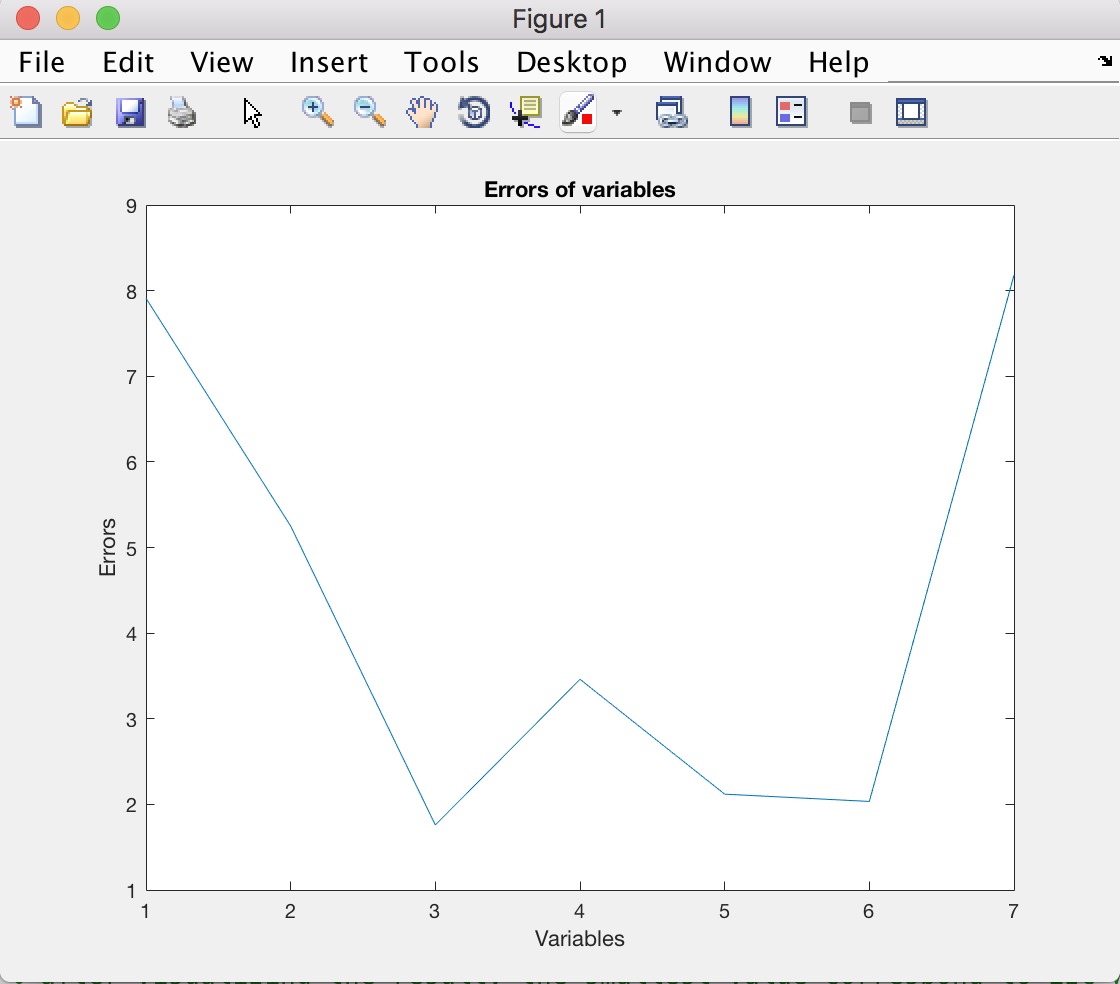
**2(a)**

Before finding the third variable in determining HOM, we must to create a matrix with size of 1\*13 that fill it with 1 and combine it with FTP and WE [1, FTP, WE] (Matrix1)

The way to find the third variable in determining HOM is shown as below:

1. Combine the matrix of targeted variable with Matrix1 [1, FTP, WE, variable] (Matrix2)
2. Get the lowest error for each variable:
3. Get the sum of (Matrix2\*((((Matrix2')\*Matrix2)^(-1))\*(Matrix2')\*HOM))-HOM
4. Lowest error will be “sum/(2\*13)”
5. Find the index of the minimum value from Step 2
6. The minimum of those lowest error is LIC, so **LIC is the third variable in determining HOM**.

The errors of each variables can be seen as below:



**2(b)**

1. The unknown non-number features are replaced by the median value of all the other features. And the unknown number features are replaced by label-conditioned mean (sum of the feature of “+” or “-“ / number of ”+” or “-“).
2. Accuracy Table

|  |  |  |
| --- | --- | --- |
| K value | Lenses | CRX |
| 1 | 5/6 = 83.33% | 114/138 = 82.6% |
| 2 | 4/6 = 66.66% | 113/138 = 81.88% |
| 3 | 5/6 = 83.33% | 113/138 = 81.88% |
| 5 | 3/6 = 50% | 117/138 = 84.78% |
| 10 | 3/6 = 50% | 119/138 = 86.23% |
| 100 | 3/6 = 50% | 108/138 = 78.26% |
| 300 | 3/6 = 50% | 97/138 = 70.28% |
| 500 | 3/6 = 50% | 83/138 = 60.14% |

PS. I couldn’t generate executable file for both of the program of problem 2. But the first one works on MATLAB app, and the second one works with standard commands (python xxx.py ….)