Experiment 3
Predicting Energy Efficiency for Residential Buildings
(Revised June 20, 2019)

1. Introduction and Objective

In this experiment, we build a multi-output linear model to predict energy efficiency of residential buildings in terms of heating and cooling loads. The database, from which the model in question learns, was created by A. Tsanas and A. Xifara in 2012 [R1] and has since been popular for performance evaluation of various prediction as well as classification techniques [R2].

In [R1], an energy analysis using 12 different building shapes was carried out, where the buildings differ with respect to a total of eight features including relative compactness, surface area, wall area, glazing area, and glazing area distribution, and so on. The data set contains 768 samples, each sample is characterized by a vector $\boldsymbol{x}$ with 8 components which are numerical values of the eight features mentioned above. Also associated with each sample is a 2-component output vector $\boldsymbol{y}$ representing heating and cooling loads of the building. Here we take the term "output vector" to mean a functional mapping from a set of eight features of a building as seen in a vector $\boldsymbol{x}$ to the building's heating and cooling loads.

Clearly, we are dealing with a dataset of the form $\{(\boldsymbol{x}_n, \boldsymbol{y}_n),\ n = 1, 2, \ldots, N\}$ with $\boldsymbol{x}_n \in R^{8\times 1}$, $\boldsymbol{y}_n \in R^{2\times 1}$ and $N = 768$. The objective of the experiment is to develop a 2-output linear model that predicts heating and cooling loads for an "unseen" residential building characterized by a new feature vector $\boldsymbol{x}$.

In this experiment, the above data set was divided into two sets – one for training and the other for testing. The train data include 640 samples and the associated outputs while the test data include 128 samples and their outputs, and the division was done at random.

References

[R1] A. Tsanas and A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools", *Energy and Buildings*, vol. 49, pp. 560-567, 2012.

[R2] UCI Machine Learning, http://archive.ics.uci.edu/ml, University of California Irvine, School of Information and Computer Science.

2. Background

2.1 Multi-output linear model for prediction

Multi-output linear model for prediction is studied in Sec. 1.5, see pages 22-24 of the course notes. Below we summarize the method where the quantities involved are explicitly specified in accordance with the above dataset. The linear model of interest is given by

$$\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x} + \boldsymbol{b} \qquad \text{(E3.1)}$$

where $\boldsymbol{y} \in R^{2\times 1}$, $\boldsymbol{W} \in R^{8\times 2}$ and $\boldsymbol{b} \in R^{2\times 1}$. With train data $\{(\boldsymbol{x}_n, \boldsymbol{y}_n), n = 1, 2, ..., 640\}$ (see Sec. 2.2 below) , we construct matrices

$$\hat{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{x}_1^T & 1 \\ \boldsymbol{x}_2^T & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_{640}^T & 1 \end{bmatrix} \tag{E3.2}$$

and follow Eq. (1.30) from the course notes to compute the optimal parameters $\boldsymbol{W}^* = \begin{bmatrix} \boldsymbol{w}_1^* & \boldsymbol{w}_2^* \end{bmatrix}$ and $\boldsymbol{b}^* = \begin{bmatrix} b_1^* \\ b_2^* \end{bmatrix}$ as

$$\begin{bmatrix} \boldsymbol{w}_1^* & \boldsymbol{w}_2^* \\ b_1^* & b_2^* \end{bmatrix} = \left(\hat{\boldsymbol{X}}^T \hat{\boldsymbol{X}} + \varepsilon \boldsymbol{I}_9 \right)^{-1} \hat{\boldsymbol{X}}^T \begin{bmatrix} \boldsymbol{y}_1^T \\ \boldsymbol{y}_2^T \\ \vdots \\ \boldsymbol{y}_{640}^T \end{bmatrix} \tag{E3.3}$$

where we have include a term $\varepsilon \boldsymbol{I}_9$ to assure inverse of matrix $\hat{\boldsymbol{X}}^T \hat{\boldsymbol{X}} + \varepsilon \boldsymbol{I}_9$ does exist. Here scalar $\varepsilon > 0$ shall be small and we recommend that $\varepsilon = 0.01$ be used for this experiment.

Alternatively, pseudo-inverse of $\hat{\boldsymbol{X}}$, denoted by $\hat{\boldsymbol{X}}^\dagger$, may be used to replace $\left(\hat{\boldsymbol{X}}^T \hat{\boldsymbol{X}} + \varepsilon \boldsymbol{I}_9 \right)^{-1} \hat{\boldsymbol{X}}^T$ so that the formula in (E3.3) becomes

$$\begin{bmatrix} \boldsymbol{w}_1^* & \boldsymbol{w}_2^* \\ b_1^* & b_2^* \end{bmatrix} = \hat{\boldsymbol{X}}^\dagger \begin{bmatrix} \boldsymbol{y}_1^T \\ \boldsymbol{y}_2^T \\ \vdots \\ \boldsymbol{y}_{640}^T \end{bmatrix} \tag{E3.4}$$

Remark: MATLAB provides a function named **`pinv`** to calculate pseudo-inverse.

2.2 The dataset

Listed below are eight features collected in the dataset created by Tsanas and Xifara:

- Relative compactness as component x_1
- Surface area as component x_2
- Wall area as component x_3
- Roof area as component x_4
- Overall height as component x_5
- Orientation as component x_6
- Glazing area as component x_7
- Glazing area distribution as component x_8

The measurements of the two outputs corresponding to a given set of these building features are:

· Heating load as component y_1

· Cooling load as component y_2

There are two data sets are available from the course website:
• matrix **D_build_tr** of size 10×640 for training. The first 8 rows of **D_build_tr** constitute data matrix **Xtr** and the last 2 rows form the 2-component output matrix **Ytr** which contains measurements of heating and cooling loads for the corresponding 640 buildings.
• matrix **D_build_te** of size 10×128 for testing. The first 8 rows of **D_build_te** constitute data matrix **Xte** and the last 2 rows form the 2-component output matrix **Yte** which contains measurements of heating and cooling loads for the corresponding 128 buildings.

Once **D_build_tr** and **D_build_te** are down-loaded, data matrix **Xtr** and output matrix **Ytr** for training can be obtained as

```
Xtr = D_build_tr(1:8,:);
Ytr = D_build_tr(9:10,:);
```

and data matrix **Xte** and output matrix **Yte** for test purposes can be obtained as

```
Xte = D_build_te(1:8,:);
Yte = D_build_te(9:10,:);
```

If you did the above correctly, **Xtr** shall contain 640 8-dimensional train samples and **Ytr** contains the corresponding outputs; and **Xte** shall include 128 8-dimensional test samples, and **Yte** contains the corresponding outputs.

3. Procedure

3.1 From the course website download **D_build_tr.mat** and **D_build_te.mat**

3.2 Use Eq. (E3.2) to prepare matrix $\hat{\boldsymbol{X}}$ from the train data **Xtr**, see Sec. 2.2 above.

3.3 Use Eq. (E3.3) or (E3.4) to prepare MATLAB code to compute optimal parameters $\boldsymbol{W}^*$ and $\boldsymbol{b}^*$ for the model in (E3.1).

3.4 Apply the optimized model

$$\boldsymbol{y} = \boldsymbol{W}^{*T}\boldsymbol{x} + \boldsymbol{b}^*$$

to the test data (i.e. **Xte**, see Sec. 2.2 above). This yields 128 predicted output vectors which we denote by $\{\boldsymbol{y}_n^{(p)}, n = 1,2,...,128\}$. Evaluate the prediction performance as follows.

(i) Use $\{\boldsymbol{y}_n^{(p)}, n = 1,2,...,128\}$ to form matrix

$$\boldsymbol{Y}^{(p)} = \left[\begin{array}{cccc} \boldsymbol{y}_1^{(p)} & \boldsymbol{y}_2^{(p)} & \cdots & \boldsymbol{y}_{128}^{(p)} \end{array}\right]$$

and compute the overall relative prediction error as

$$e_p = \frac{\| \mathbf{Y_{te}} - \boldsymbol{Y}^{(p)} \|_F}{\| \mathbf{Y_{te}} \|_F}$$

where matrix `Yte` contains 128 ture output vectors (see Sec. 2.2 above), $\| \cdot \|_F$ denotes Frobenius norm which is defined by $\| \boldsymbol{A} \|_F = \left(\sum_i \sum_j a_{i.j}^2 \right)^{1/2}$.

(ii) For comparison, plot the first row of `Yte` and first row of $\boldsymbol{Y}^{(p)}$ in the same figure, highlighted the two curves with different color. In another figure, plot the second row of `Yte` and second row of $\boldsymbol{Y}^{(p)}$, highlighted these curses with different color. Comment on your visual inspection of the two figures.

Include your MATLAB code in the lab report.