A Multi-ELM Model for Incomplete Data

Baichuan Chi, Faculty Mentor: Amaury Lendasse, Renjie Hu, Edward Ratner
ILT Department, College of Technology

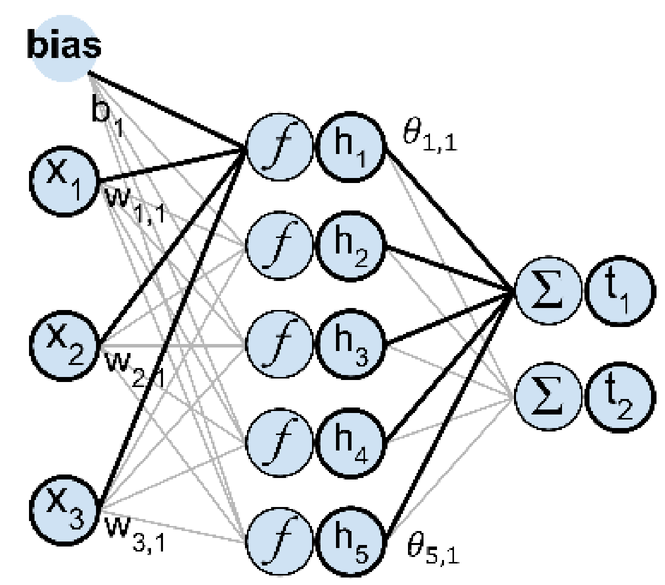
UNIVERSITY of HOUSTON

Introduction

Dataset with missing values (incomplete data) is observed frequently in real-world machine learning tasks. The most used strategies currently includes, 1) discard incomplete data and utilize only complete data and 2) create imputations of incomplete data. Discarding incomplete data is impractical when the amount of missing data is vast as data collection could be costly and information contained in incomplete data is discarded. Imputation approach also has drawback as noise is introduced and predictions become less accurate.
Mean imputation (MI) and K Nearest Neighbors (KNN) imputation are two very popular imputation methods. But they may be effective only when a small amount of value is missing, and they make prediction biased as estimated values are included.
Besides discarding the data and imputation, the third approach would be to apply machine learning models directly on the incomplete data to avoid errors introduced by imputations. But not many machine learning models can handle incomplete data directly. Our research proposes a novel machine learning model based on the Extreme Learning Machine (ELM) which could be applied directly on incomplete data. The proposed method is capable of handling a variety of missing patterns in the data and has been proved to be more efficient than current methods when the percentage of missing values is high in data.

Method

The Extreme Learning Machines [9, 10] are Single-Layer Feed-forward Networks (SLFNs) [11]. According to Huang et al. in [9], ELM has universal approximation capability



$\boldsymbol{w}$ are random weights

$$\mathbf{H} = \begin{pmatrix} h_1(x_1) & \dots & h_L(x_1) \\ \dots & \ddots & \dots \\ h_1(x_N) & \dots & h_L(x_N) \end{pmatrix}$$

The optimal $\boldsymbol{\beta}^*$ is solved by $\boldsymbol{\beta}^* = \operatorname{argmin} \| \mathbf{H}\boldsymbol{\beta} - \mathbf{T} \|_2^2$

$$\boldsymbol{\beta}^* = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{T}$$

Figure 1 Extreme Learning Machine

If original data are incomplete, $\mathbf{H}$ can be written as $\mathbf{H} = \mathbf{H}^c + \mathbf{H}^m$. $\mathbf{H}^c$ is the calculated by using the complete variables of $\boldsymbol{x}$, $\mathbf{H}^m$ is missing and can not be compute directly.

$$\mathbf{H}^c = \begin{pmatrix} h_1^c(\boldsymbol{x}_1^c) & \dots & h_L^c(\boldsymbol{x}_1^m) \\ \dots & \ddots & \dots \\ h_1^c(\boldsymbol{x}_N^c) & \dots & h_L^c(\boldsymbol{x}_N^m) \end{pmatrix},$$

$$\mathbf{H}^m = \begin{pmatrix} h_1^m(\boldsymbol{x}_1^c) & \dots & h_L^m(\boldsymbol{x}_1^m) \\ \dots & \ddots & \dots \\ h_1^m(\boldsymbol{x}_N^c) & \dots & h_L^m(\boldsymbol{x}_N^m) \end{pmatrix}$$

Although $\mathbf{H}^m$ can not be obtained directly, it can be approximated by using additional ELMs.

$$\hat{h}^m\left(\boldsymbol{x}_{\boldsymbol{p}_j}^m\right) = \mathfrak{E}_{\boldsymbol{p}_j}\left(\boldsymbol{x}_{\boldsymbol{p}_j}^c, \boldsymbol{w}_{\boldsymbol{p}_j}^m\right)$$

The additional ELMs $\mathfrak{E}_{\boldsymbol{p}_j}$ used here are called the secondary ELMs.
For every incomplete pattern $\boldsymbol{p}_j$ a secondary ELM is trained. There are multiple secondary ELMs, since the incomplete data have multiple missing patterns. For instance, if an incomplete data point $\boldsymbol{x}_{\boldsymbol{p}_j}$ follows a missing pattern $\boldsymbol{p}_j = (1,1,0,1,0)$, $\boldsymbol{x}_{\boldsymbol{p}_j}^c = (x^3, x^5)$, $\boldsymbol{x}_{\boldsymbol{p}_j}^m = (x^1, x^2, x^4)$. The same logic applies to the hidden neurons' weights $\boldsymbol{w}$ in ELM.

The complete $\mathbf{H}$ matrix in the primary ELM is obtained by adding the approximations from the secondary ELMs on top of the known values: $\mathbf{H}^* = \mathbf{H}^c + \hat{\mathbf{H}}^m$. Hence the $\boldsymbol{\beta}$ in the primary ELM can be solved by plugging in the $\mathbf{H}^*$ as $\boldsymbol{\beta}^* = (\mathbf{H}^{*T}\mathbf{H}^*)^{-1}\mathbf{H}^{*T}\mathbf{T}$. Therefore, the primary ELM can generate approximations for the original target $\mathbf{T}$.

Result

Experiments on the incomplete data has been conducted to examine performance of the proposed method on diverse datasets. The proposed method is compared with other imputation methods, including the mean imputation and the KNN imputation. The mean squared errors (MSE) are adopted as the final measurements of the performances.

Table 1: Experiment Dataset

Name of Dataset	Sample Size	Input Variables	Target Variable
Abalone	2784	7	Age of Abalone
The Boston Housing	506	13	The Boston House Price

Table 2: Incomplete Dataset

Level of Incompleteness	15%		30%	
Name of the dataset	Abalone	Boston	Abalone	Boston
Incomplete Samples	2318	475	2658	503
Complete Samples	466	31	126	3
Total Missing Entrees	2923	986	5846	1973

Table 3: ELM Configurations

Level of Incompleteness	15%		30%	
Name of the dataset	Abalone	Boston	Abalone	Boston
Primary ELM Neurons	27	15	28	24
Number of Secondary ELMs	272	19	136	2

Table 4: MSE Comparison

Level of Incompleteness	15%		30%	
Name of the dataset	Abalone	Boston	Abalone	Boston
Muilt-ELM Model	0.08307	0.3619	1.082	0.8220
Mean Imputation	0.3537	0.5589	1.181	0.7892
KNN Imputation	0.08930	0.4634	0.2426	0.7512

Conclusion

According to the performed experiments on the two datasets, for a reasonable amount of missing data (0% to 20%) the Multi-ELM model is outperforming the other imputation methods. In different data-based concepts, the Multi-ELM model decreases the noises from imputing the missing values directly. Furthermore, it can be concluded that the Multi-ELM model is capable of handling different missing patterns, even if they have never shown up in the training set. It now has the suitability for big data as it can be paralleled easily. In the future, the Multi-ELM model for incomplete data will be extended and tested for its speed and performance on diverse datasets especially large datasets.