Proceedings of IAM, V.14, N.2, 2025, pp.222-233
https://doi.org/10.30546/2225-0530.14.2.2025.2017

ROBUST OSLEM IN THE PRESENCE OF OUTLIER

Reza Habibi[1] and Hamed Habibi[2]

[1]Iran Banking Institute, Central Bank of Iran, Tehran, Iran
[2]Affliation Department of Mechanical Engineering, Curtin University, Australia

e-mail: habibi1356@yahoo.com, hamed.habibi@postgrad.curtin.edu.au

Abstract. Extreme Learning Machine (ELM) is a widely-used structure as the topology of neural networks. However, since it is expressed in terms of the least squares method, it is sensitive to outliers. One of the alternatives of least square method, which is robust to outlier data, is the M-estimation method. In this paper, using the M-estimation method, the recursive relations for tuning the parameters in the ELM are presented.

Keywords: ELM, M-estimation, Outliers, Robust methods.

AMS Subject Classification: 41A45.

1. Introduction

Feed-forward neural networks have shown salient capabilities for universal approximation on compact input sets with a finite set of training samples. For instance, approximation of unknown nonlinear functions and identifying the behavior of dynamic systems are two common applications, both of which are challenging using classical parametric approaches [3]. Conventionally, the network parameters, i.e., weights and biases, are tuned in the training process iteratively using, for example, gradient descent-based methods. However, it is been theoretically and practically proven that the learning speed of these methods is slow. Also, the learning algorithm might converge to local minima and, consequently, additional steps are required to resolve this. This further slows down the learning process and, in turn, causes major issues for real-world applications [1].

The Online Sequential Extreme Learning Machine (OSELM) is an extension of the Extreme Learning Machine (ELM) designed specifically for online learning scenarios. Here are some advanced aspects of OSELM in analyzing data:

1. Online Learning Capability

OSELM processes data in a sequential manner, allowing it to update its model incrementally as new data arrives. This is particularly beneficial for dynamic environments where the data distribution may change over time.

2. Efficiency and Speed

OSELM is computationally efficient because it requires only a single pass through the data. The training phase involves a simple matrix inversion operation, making it suitable for large datasets and real-time applications.

3. Generalization Performance

OSELM has been shown to achieve good generalization performance, often comparable to or better than traditional machine learning algorithms. Its ability to handle high-dimensional feature spaces effectively is one of its strong suits.

4. Robustness to Noise

The inherent structure of OSELM can make it more robust to noise and outliers in the data compared to some other algorithms. This robustness is crucial for real-world applications where data quality may be variable.

5. Adaptability to Non-Stationary Environments

OSELM can adapt to changes in the underlying data distribution over time, making it suitable for applications like stock price prediction, sensor data analysis, or any situation where data evolves continually.

6. Multi-class Classification and Regression

OSELM can handle multi-class classification problems and regression tasks, making it a versatile choice for a wide range of data analysis problems.

7. Integration with Other Techniques

OSELM can be integrated with other machine learning techniques, such as ensemble methods or deep learning architectures, to enhance its performance further.

8. Parameter Tuning

While OSELM is generally less sensitive to hyper-parameters than other models, there are still parameters (like the number of hidden nodes) that can be tuned for improved performance based on the specific dataset.

9. Implementation in Real-Time Systems

OSELM’s efficiency makes it suitable for real-time systems where timely data processing and decision-making is crucial, such as applications, online recommendation systems, and automated trading systems.

10. Visualization and Interpretability

Advanced implementations of OSELM might include visualization tools and techniques to interpret the model's decisions, making it easier for users to trust and understand the model’s predictions.

Considerations such as the nature of the data, the specific analytical goals, and computational resources should guide the decision to use OSELM or explore other methodologies. If you have a specific use case in mind, feel free to ask for more tailored insights.

As a resolution, unlike conventional approaches, ELM was proposed in [2], using a single-hidden layer structure with randomly selected parameters. More importantly, the output weights are determined analytically. This eliminates the iterative learning process and, in turn, increases the convergence speed [2]. More importantly, it has been proven that an ELM with N hidden nodes and randomly selected parameters can exactly learn N distinct samples if activation functions are infinitely differentiable [1]. Mathematically, an ELM can be seen as linear regression and the output weights are determined, analytically, by the least square method with the generalized inverse operation. Theoretically, ELM has shown considerable generalization performance for online fast prototyping (smallest

training error) with fast learning speed. This provides the ability in terms of online approximation and identification.

The so-called batch ELM, proposed in [1], assumes that all the training data is available for training. However, practically, training data may be received gradually during the operation, i.e., chunk-by-chunk or one-by-one. Additionally, it might be required to retrain the network online in some situations. Therefore, the Online Sequential ELM (OSELM) was proposed in [2] using data blocks, arriving sequentially, by solving an analytic recursive equation. Moreover, the data blocks for which the training has been done, are discarded. This improved the generalization, by reducing the chance of over-fitting. Unlike the other sequential approaches with many parameters, the number of hidden nodes is the only hyper-parameters to be tuned in OSELM.

Generally, in learning processes of neural networks, the selection of training data affects the performance of the network. For instance, if some outlier data exist for training, the generalization of the network is significantly compromised. As a result, there might be some outliers in the data, due to errors stemming from measurement, extraction from heterogeneous databases, data processing and sampling, see [8] and references therein. In this regard, in the OSELM method, the stream of training data, received online, cannot be controlled. There, the learning process is based on recursive relations of unknown parameters of network, derived in [2] using the least square method. Beside this, in [2], there is no result about the standard deviation and accuracy of recursive estimations. Therefore, statistically, in the presence of outliers, the least square method leads to a large error with high variance or bias, slow convergence rates, and inconsistent estimation, see [4]. Consequently, the training process fails to estimate the weight accurately and the trained network is no longer valid for real-time application.

Jumps, change points, structural breaks and outliers among the data stream are common phenomena that are frequently seen in high dimensional financial time series (see [3]). There are some solutions to overcome these difficulties such as weighted LS (for solving the problem of jumps), rolling estimates (to consider the change points), the use of forgetting factor (o insist on new-comer observations) or exponentially weighted moving average approaches (to overcome the problem of structural breaks) (see [4]). However, to consider the outliers which affect substantially the result of the learning process robust analysis such as M-estimation is required (see, [5]). In M-estimation a different loss function is used instead of quadratic function used in least square method is used.

Therefore, it is desirable to have an OSELM algorithm modified to be robust against the outliers, and, more importantly, to obtain the accuracy of the estimation, both of which are not considered in the recursive OSELM method [2]. The accuracy, then, can be presented as a level of confidence which, in turn, enables the user to decide on the (re)training schedule. Motivated by the aforementioned issues, in this paper, a modification of OSELM is presented, considering the existence of outliers in the data, using a robust M-estimate method.

In summary, the main contributions of this paper are as follows.

(i) Presenting recursive equation as regression point of view

(ii) Considering the outliers in data stream
(iii) Proposing robust inference M-estimation to overcome the difficulty of outliers
In the next section, problem formulation from the regression point of view is presented. Then, the M-estimates of parameters are proposed in section 3. Recursive relations are proposed in section 4.

2. Regression formulation

Following the notation of [2], and assuming $\widetilde{N}$ hidden nodes for the network, the OSELM is presented as least square (LS) system of equations

$$\mathrm{H}\beta = \mathrm{T},$$

where $H_{N\times\widetilde{N}}$ is the matrix of hidden layer with components h_{ij} computed by activation function $g(.)$. The $T_{N\times m}$ and $\beta_{\widetilde{N}\times m}$ are the matrices of target values and unknown weights, respectively. In [2], authors assumed that training input $x_i \in \mathbb{R}^n$ and the corresponding target vector $t_i \in \mathbb{R}^m$ are received chunk by chunk sequentially as (with length of N_k of k^{th} chunk with $N_{-1} = 0$)

$$\mathcal{N}_k = \{(x_i, t_i) | i = N_{k-1} + 1, \dots, N_k \}, k \geq 0,$$

and by minimization the recursive norm LS for $\bigcup_{j=0}^{k+1} \mathcal{N}_j$ chunk, they estimated β recursively as

$$\beta^{(k+1)} = \beta^{(k)} + P_{k+1} H_{k+1}^T (T_{k+1} - H_{k+1}\beta^{(k)}),$$

where P_k's are defined in [2]. The above recursive is the heart of their learning process, however, the main fault regarding this equation is its accuracy. That is, there is no guaranty that estimates $\beta^{(k)}$ are close to the actual value β. Indeed, there is no issue that standard deviations of $\beta^{(k)}$ are small.
From regression outline, equation $\mathrm{H}\beta = \mathrm{T}$ can be interpreted as the normal equation of the regression model

$$\mathrm{T} = \mathrm{H}\beta + \varepsilon,$$

where, ε is the error term. In traditional regression models, it is assumed that the ε is a sequence of variables with finite variance property which justifies the use of the LS estimation method. By this assumption, happening outliers is a rare event, however, in many cases, such as financial time series data, many outliers are observed in the data stream, frequently. Economic environment changes and probably financial shocks, crisis, structural breaks and abrupt changes occur which doubts the use of the LS method as well as its accuracy and robustness. In practice, the simplest actions facing the outliers are deleting them from the data set. The following remark shows that it is not a suitable action. Before, going ahead, notice that the estimation of T is

$$\mathrm{H}\beta^{(k)} = \mathrm{RT}$$

where R is the projection matrix of T on span H defined by

$$\mathrm{R} = \mathrm{H}\left(\mathrm{H}^{\mathrm{T}}\mathrm{H}\right)^{-1}\mathrm{H}.$$

Let $\mathrm{Q} = \mathrm{I} - \mathrm{R}$. It is easy to see that, R is an idempotent matrix, $\mathrm{R}^2 = \mathrm{R}$ also R, Q are orthogonal matrices, that is,

$$RQ = 0 \text{ and } QR = 0.$$

Remark 1. Let $H = [H_1 \vdots H_2]$ and $\beta^{\mathrm{T}} = \left[\beta_1^{\mathrm{T}} \vdots \beta_2^{\mathrm{T}}\right]$ be partitioned (in column), with appropriate dimensions. Thus, the regression is given by

$$T = H_1\beta_1 + H_2\beta_2 + \varepsilon.$$

Let $Q_2 = \mathrm{I} - R_2$ and $R_2 = H_2(H_2^T H_2)^{-1}H_2^T$. One can see that

$$Q_2 T = Q_2 H_1\beta_1 + Q_2 H_2\beta_2 + Q_2\varepsilon = Q_2 H_1\beta_1 + Q_2\varepsilon.$$

Therefore, the LS estimate of β_1 and β_2 are given by

$$(H_1^T Q_2 H_1)^{-1}H_1^T Q_2 T \text{ and } (H_2^T Q_1 H_2)^{-1}H_1^T Q_1 T,$$

respectively. This remark shows that existing outliers in H_1 or H_2 or T has bad effects on both LS estimate of β_1 and β_2. Therefore, even we delete outliers exist in H_1, they affect the estimation accuracy of β_2 via Q_1, indirectly. Therefore, removing the outliers is not a proper solution and robust estimation of parameters using M-estimation method is felt.

3. Robust OSELM

Here, we study the robust OSELM in the presence of outliers. To obtain robust estimate of parameter matrix β against the outliers, in the M-estimation method, the loss function ρ is used instead of squared function. Following [6] and [7] two types of loss functions, instead of squared functions, are considered.

(i) A general form of scalar function ρ which is the convex function with two continuous derivatives which satisfies the following assumptions.

(1) $\rho_1 := \Psi = \frac{d\rho(x)}{dx}$ and $\Psi_1 := \frac{d\Psi(x)}{dx}$ are Lipschitz continuous,

(2) $E\big(\Psi(\varepsilon)\big) = 0$ and $\Psi(\varepsilon)$ have finite second moments,

(3) $\Psi_1(\varepsilon)$ have finite moments.

(ii) The absolute value function.

One of the features of scalar LS norm is its definition in term of inner product of vectors and matrix multiplication, which enables one to extend the method for the case of unknown matrix of parameters as used in [2,3]. However, for M-estimation method, it is not easy to define ρ in the multivariate cases. An exception is the Manhattan multivariate norm which is defined by $\rho(x) = |x|$, see [7,8]. In order to solve this difficulty, notice that the model $\mathrm{T} = \mathrm{H}\beta + \varepsilon$ is the multivariate multiple regression as presented in [9] page 387. Following them, consider the partitioned (in columns) matrices

$$T = \left[\!\left[T_{(1)} \vdots T_{(2)} \vdots \cdots \vdots T_{(m)}\right]\!\right]$$
$$\beta = \left[\!\left[\beta_{(1)} \vdots \beta_{(2)} \vdots \cdots \vdots \beta_{(m)}\right]\!\right]$$
$$\varepsilon = \left[\!\left[\varepsilon_{(1)} \vdots \varepsilon_{(2)} \vdots \cdots \vdots \varepsilon_{(m)}\right]\!\right]$$

Then, the above multivariable regression model can be considered as m regression equations

$$T_{(j)} = \mathrm{H}\beta_{(j)} + \varepsilon_{(j)}\ j = 1, \ldots, m.$$

By this point of view and following [9], m vectors of parameters are estimated, separately. Hereafter, subscript i is removed and vectors $T_{(j)}$, $\beta_{(j)}$ and $\varepsilon_{(j)}$ are denoted bolded. Therefore, the j-th typical regression model is

$$\boldsymbol{T} = H\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

Let $\boldsymbol{H}_i'$, $i = 1,2,\dots,\mathrm{N}$ be the i-th row of matrix H. Notation ' stands for transpose of vector. Then, the above regression model may be represented as

$$\begin{pmatrix} T_1 \\ \vdots \\ T_{\mathrm{N}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{H}_1'\boldsymbol{\beta} \\ \vdots \\ \boldsymbol{H}_{\widetilde{N}}'\boldsymbol{\beta} \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_{\mathrm{N}} \end{pmatrix}.$$

In this form, the M-estimation method is applicable with scalar function ρ. To make sure that all errors terms ε_i, $i = 1,2,\dots,\mathrm{N}$ are under the control, it is enough to minimize M-estimation criterion (call it M-criterion)

$$\sum_{i=1}^{\mathrm{N}} \rho(T_i - \boldsymbol{H}_i'\boldsymbol{\beta}),$$

with respect to vector $\boldsymbol{\beta}$. Here, the M-criterion is

$$\sum_{i=1}^{\mathrm{N}} \rho(T_i - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{N}}),$$

and $\sum_{i=1}^{\mathrm{N}} \Psi(T_i - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{N}}) = \boldsymbol{0}$. For the similarity of the recursive relations obtained here with the return relations in control engineering, two symbol N is replaced with n.

4. Recursive relations

Hereafter, three types of recursive formulas for $\widehat{\boldsymbol{\beta}}_{\mathrm{n}}$ are derived. A main advantage of recursive relation of $\widehat{\boldsymbol{\beta}}_{\mathrm{n}}$, in spite of [2], is that the concept of convergence $\widehat{\boldsymbol{\beta}}_{\mathrm{n}}$ to $\boldsymbol{\beta}$ is considered in recursive relations.

(I) General ρ cases. First consider the general form loss function of (i).

(I_1) Residual-based recursive. Here, it is interested to study the closeness of $\widehat{\boldsymbol{\beta}}_{\mathrm{n}}$ to $\boldsymbol{\beta}$, as n gets large. To this end, let $\boldsymbol{v}_n \coloneqq \widehat{\boldsymbol{\beta}}_{\mathrm{n}} - \boldsymbol{\beta}$. The M-criterion is given by

$$\sum_{i=1}^{\mathrm{n}} \rho(T_i - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{n}}) = \sum_{i=1}^{\mathrm{n}} \rho(T_i - \boldsymbol{H}_i'(\boldsymbol{v}_n + \boldsymbol{\beta})),$$

$$= \sum_{i=1}^{\mathrm{n}} \rho(\varepsilon_i - \boldsymbol{H}_i'\boldsymbol{v}_n).$$

An equivalent form for the M-criterion is the following criterion (modified version)

$$\sum_{i=1}^{\mathrm{n}} \{\rho(\varepsilon_i - \boldsymbol{H}_i'\boldsymbol{v}_n) - \rho(\varepsilon_i)\}.$$

Then, using the Taylor method, it is seen that the modified version is given by

$$\boldsymbol{L}(\boldsymbol{v}_n) \coloneqq -\boldsymbol{v}_n'\boldsymbol{\gamma}_n + \frac{1}{2}\boldsymbol{v}_n'\Gamma_n\boldsymbol{v}_n,$$

where $\boldsymbol{\gamma}_n = \sum_{j=1}^{\mathrm{n}} \boldsymbol{H}_j\Psi(\varepsilon_j)$ and $\Gamma_n = \sum_{i=1}^{\mathrm{n}} \boldsymbol{H}_j\boldsymbol{H}_j'\Psi_1(\varepsilon_i)$. By maximizing $L(\boldsymbol{v}_n)$ with respect to $\boldsymbol{v}_n$ by letting $\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{v}_n} = \boldsymbol{0}$, implies that

$$-\boldsymbol{\gamma}_n + \Gamma_n\boldsymbol{v}_n = \boldsymbol{0}.$$

Thus, $\boldsymbol{v}_n = \Gamma_n^{-1}\boldsymbol{\gamma}_n$. To find recursive relation, notice that

$$\begin{cases} \boldsymbol{\gamma}_n = \boldsymbol{\gamma}_{n-1} + \boldsymbol{H}_n\Psi(\varepsilon_n), \\ \Gamma_n = \Gamma_{n-1} + \boldsymbol{H}_n\boldsymbol{H}_n'\Psi_1(\varepsilon_n). \end{cases}$$

Hence,

$$\Gamma_n^{-1} = (\Gamma_{n-1} + \boldsymbol{H}_n\boldsymbol{H}_n'\Psi_1(\varepsilon_n))^{-1} =$$

$$(I + \Gamma_{n-1}^{-1}\boldsymbol{H}_n\boldsymbol{H}_n'\Psi_1(\varepsilon_n))^{-1}\Gamma_{n-1}^{-1} = \Delta_n^{-1}\Gamma_{n-1}^{-1}$$

at which $I = I_{(n-1)\times(n-1)}$ is identity matrix and

$$\Delta_n \coloneqq I + \Gamma_{n-1}^{-1}\boldsymbol{H}_n\boldsymbol{H}_n'\Psi_1(\varepsilon_n).$$

Using the above equations, we got that

$$\boldsymbol{v}_n = \Delta_n^{-1}\boldsymbol{v}_{n-1} + \boldsymbol{\delta}_n.$$

Here, $\boldsymbol{\delta}_n = \Delta_n^{-1}\Gamma_{n-1}^{-1}\boldsymbol{H}_n\Psi(\varepsilon_n)$. This recursive related to residual terms. To overcome this difficulty, assuming $\widehat{\boldsymbol{\beta}}_{\mathrm{n}}$ is close to $\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}$, ε_n is replaced with $T_i - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}$.

Remark 1. In the case of scalar dimension regression, $T = H\beta + \varepsilon$, then

$$v_n = \frac{\sum_{i=1}^{n}\Psi(\varepsilon_i)H_i}{\sum_{i=1}^{n}\Psi_1(\varepsilon_i)H_i^2}.$$

Here, a stochastic approximation is derived for recursive relation of v_n, as follows

$$v_n = (1-\sigma_n)v_{n-1} + \sigma_n\pi_n,$$

where $\sigma_n = \frac{\Psi_1(\varepsilon_n)H_n^2}{\sum_{i=1}^{n}\Psi_1(\varepsilon_i)H_i^2} \in (0,1)$ is forgetting factor and $\pi_n = \frac{\Psi(\varepsilon_n)}{\Psi_1(\varepsilon_n)H_n}$.

(I_2) Difference-based recursive. Here, instead of investigating the convergence of $\widehat{\boldsymbol{\beta}}_{\mathrm{n}}$ to $\boldsymbol{\beta}$, the closeness of $\widehat{\boldsymbol{\beta}}_{\mathrm{n}}$ to $\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}$ is studied which is another interpretation of convergence. Therefore, let $\boldsymbol{d}_n = \widehat{\boldsymbol{\beta}}_{\mathrm{n}} - \widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}$. Then the M-criterion may be represented as

$$\sum_{i=1}^{\mathrm{n}}\rho(T_i - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1} - \boldsymbol{H}_i'\boldsymbol{d}_n).$$

Using the Taylor expansion with respect to $\boldsymbol{d}_n$ about $\mathbf{0}$, the M-criterion is approximated as

$$\boldsymbol{L}(\boldsymbol{d}_n) \coloneqq \sum_{i=1}^{\mathrm{n}-1}\rho\big(T_i - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}\big) - \sum_{i=1}^{\mathrm{n}}\Psi\big(T_i - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}\big)\boldsymbol{H}_i'\boldsymbol{d}_n + \frac{1}{2}\boldsymbol{d}_n'\Lambda_n\boldsymbol{d}_n,$$

where $\Lambda_n \coloneqq \sum_{i=1}^{\mathrm{n}}\Psi\big(T_i - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}\big)\boldsymbol{H}_i\boldsymbol{H}_i'$. Notice that

$$\sum_{i=1}^{\mathrm{n}}\Psi\big(T_i - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}\big)\boldsymbol{H}_i' = \Psi\big(T_n - \boldsymbol{H}_n'\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}\big)\boldsymbol{H}_n' \coloneqq \boldsymbol{\lambda_n}.$$

By maximizing $L(\boldsymbol{d}_n)$ with respect to $\boldsymbol{d}_n$ by letting $\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{d}_n} = \mathbf{0}$, implies that

$$-\boldsymbol{\lambda_n} + \Lambda_n\boldsymbol{d}_n = \mathbf{0}.$$

Therefore, $\boldsymbol{d}_n = \Lambda_n^{-1}\boldsymbol{\lambda_n}$. Thus, assuming Λ_n is invertible, we have $\boldsymbol{d}_n = \Lambda_n^{-1}\boldsymbol{\lambda_n}$, equivalently, then

$$\widehat{\boldsymbol{\beta}}_{\mathrm{n}} = \widehat{\boldsymbol{\beta}}_{\mathrm{n}-1} + \Lambda_n^{-1}\boldsymbol{\lambda_n}.$$

Notice that although $\boldsymbol{\lambda_n}$ depends on $\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}$, however, Λ_n depends on $\widehat{\boldsymbol{\beta}}_{\mathrm{i}}, i = 1,2,\dots,n-1$. This property causes that this recursive relation becomes a time consuming process. To overcome this difficulty, the weighted least square is given: consider the weighted M-criterion as

$$\sum_{i=1}^{\mathrm{n}} w_j\rho(T_i - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1} - \boldsymbol{H}_i'\boldsymbol{d}_n),$$

for some weights w_i, $i = 1,2,\dots,n$. Then, Λ_n and $\boldsymbol{\lambda_n}$ are changed to Λ_n^* and $\boldsymbol{\lambda}_n^*$ as follows, respectively

$$\Lambda_n^* \coloneqq \sum_{i=1}^{\mathrm{n}} w_i\Psi\big(T_j - \boldsymbol{H}_i'\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}\big)\boldsymbol{H}_i\boldsymbol{H}_i',$$

$$\boldsymbol{\lambda}_n^* \coloneqq w_n\Psi\big(T_n - \boldsymbol{H}_n'\widehat{\boldsymbol{\beta}}_{\mathrm{n}-1}\big)\boldsymbol{H}_n'.$$

Then, $\widehat{\boldsymbol{\beta}}_{\mathrm{n}} = \widehat{\boldsymbol{\beta}}_{\mathrm{n}-1} + \Lambda_n^{*-1}\boldsymbol{\lambda}_n^*$. Using weighting scheme such as exponentially weighted moving average (EWMA) method, it is possible to reduce the effect of initial data and fastening computations.

(I_3) Dynamic programming based recursive. In previous section, the M-estimate $\widehat{\boldsymbol{\beta}}_n$ of $\boldsymbol{\beta}$ was defined as a vector which minimizes the M-criterion

$$u_n(\widehat{\boldsymbol{\beta}}_n) := \sum_{i=1}^{n} \rho(T_i - \boldsymbol{H}_i' \widehat{\boldsymbol{\beta}}_n);$$

for each fixed n. However, from the dynamic programming point of view, the suitable estimate is a sequence of $\boldsymbol{\beta}_n$ which minimizes discounted utility

$$\sum_{t=1}^{\infty} \gamma^t \, u_t(\boldsymbol{\beta}_t),$$

for some discount factor $\gamma \in (0,1)$. The advantage of this method is that all $u_n(\boldsymbol{\beta}_n), n \geq 1$, are minimized, simultaneously. Let $\boldsymbol{e}_n = \boldsymbol{\beta}_n - \boldsymbol{\beta}_{n-1}$. Then,

$$\boldsymbol{\beta}_n = \boldsymbol{\beta}_0 + \sum_{\mathrm{i}=1}^{\mathrm{n}} \boldsymbol{e}_{\mathrm{i}}.$$

Here, beside $\boldsymbol{\beta}_t$'s, the errors $\boldsymbol{e}_t$'s should be controlled. Therefore, mimicking the cake eating problem, see Ljungqvist and Sargent (2004), the following minimization problem is considered:

$$min_{\{\boldsymbol{\beta}_t\},\{\boldsymbol{e}_t\}} \sum_{t=1}^{\infty} \gamma^t \, u_t(\boldsymbol{\beta}_t),$$

subject to $\boldsymbol{\beta}_t = \boldsymbol{\beta}_0 + \sum_{\mathrm{i}=1}^{\mathrm{t}} \boldsymbol{e}_{\mathrm{i}}$. Let $V_t(\boldsymbol{e}_t)$ be the minimal discounted utility with respect to $\boldsymbol{\beta}_t$, then, the Bellman equation implies that

$$V_t(\boldsymbol{e}_t) = min_{\{\boldsymbol{\beta}_t\}}\{u_t(\boldsymbol{\beta}_t) + \gamma V_{t-1}(\boldsymbol{e}_{t+1})\}.$$

The Lagrange multiplier method considers the following function

$$\int = \sum_{t=1}^{\infty} \gamma^t \, u_t(\boldsymbol{\beta}_t) - \boldsymbol{\lambda}'\big(\boldsymbol{\beta}_t - \boldsymbol{\beta}_0 - \sum_{\mathrm{i}=1}^{\mathrm{t}} \boldsymbol{e}_{\mathrm{i}}\big).$$

From $\frac{\partial \int}{\partial \boldsymbol{\beta}_t} = 0$, it is concluded that $\gamma^t u_{1t}(\boldsymbol{\beta}_t) = \boldsymbol{\lambda}'$, where $u_{1t}(x) := \frac{du_t(x)}{dx}$. Therefore,

$$\gamma u_{1t}(\boldsymbol{\beta}_t) = u_{1(t-1)}(\boldsymbol{\beta}_{t-1}),$$

which is a type of Euler's equation.

(II) Absolute value loss function. Next, let $\rho(x) = |x|$. Here, ρ is not differentiable at zero, however, identities as follows are useful

$$|\tau - \theta| = \theta + \int_0^{\tau} (2I(\theta \leq t) - 1)\, dt =$$
$$\theta - \tau + 2\int_0^{\tau} I(\theta \leq t) dt =$$
$$\theta - \tau + 2\int_0^{1} \tau I(\theta \leq \tau u) du,$$

for every real numbers τ and θ. Here, $I(\theta \leq t)$ is one if $\theta \leq t$ and zero otherwise. The scaled M-criterion (dividend by n) is

$$M_n = \frac{1}{n}\sum_{i=1}^{n} \big|T_i - \boldsymbol{H}_i' \widehat{\boldsymbol{\beta}}_n\big|.$$

(II_1) I_1Continued. Notice that $M_n = \frac{1}{n}\sum_{i=1}^{n} |\varepsilon_i - \boldsymbol{H}_i' \boldsymbol{e}_n|$, where $\boldsymbol{e}_n = \widehat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}$**.** Let $\theta = \varepsilon_j$ and $\tau = \boldsymbol{H}_i' \boldsymbol{e}_n$. Thus,

$$M_n = \frac{1}{n}\sum_{i=1}^{n} \{\varepsilon_i - \boldsymbol{H}_i' \boldsymbol{e}_n + 2\int_0^{1} \boldsymbol{e}_n' \boldsymbol{H}_i I(\varepsilon_i \leq \boldsymbol{e}_n' \boldsymbol{H}_i u) du =$$
$$\bar{\varepsilon}_n - \bar{\boldsymbol{H}}' \boldsymbol{e}_n + \frac{2}{n}\int_0^{1} \sum_{i=1}^{n} \boldsymbol{e}_n' \boldsymbol{H}_i I(\varepsilon_i \leq \boldsymbol{e}_n' \boldsymbol{H}_i u)\, du,$$

where $\bar{\varepsilon}_n = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i$, $\bar{\boldsymbol{H}}_n = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{H}_i$, and in the first term, we used the identity $\boldsymbol{H}_i' \boldsymbol{e}_n = \boldsymbol{e}_n' \boldsymbol{H}_i$. Now, for fixed real number x, consider the summation

$$I = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{e}_n' \boldsymbol{H}_i I(\varepsilon_i \leq x) =$$
$$\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{e}_n' \boldsymbol{H}_i \{I(\varepsilon_i \leq x) - F(x)\} + \boldsymbol{e}_n' \bar{\boldsymbol{H}}_n F(x),$$

where $F = F_{\varepsilon_j}$ is the distribution function of $\varepsilon_i, i \geq 1$. For $\gamma \in (0,1]$, consider the stochastic process

$$K_n(\gamma, x) := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{e}_n' \boldsymbol{H}_i \{I(\varepsilon_i \leq x) - F(x)\}.$$

Notice that

$$I = \int_0^1 \boldsymbol{e}_n' \boldsymbol{H}_{[n\gamma]} dK_n(\gamma, x) + \boldsymbol{e}_n' \bar{\boldsymbol{H}}_n F(x).$$

It is well-known that

$$\sup_x |K_n(\gamma, x)| = O_p(n^{-1/2}).$$

Assuming $\boldsymbol{e}_n' \boldsymbol{H}_{[n\gamma]} = O_p(1)$, then I is well-approximated by $o_p(1) + \boldsymbol{e}_n' \bar{\boldsymbol{H}}_n F(\boldsymbol{e}_n' \boldsymbol{H}_i u)$. Also, $\bar{\varepsilon}_n$ is $o(1)$, almost surely, thus, the M_n is approximated by

$$-\boldsymbol{e}_n' \bar{\boldsymbol{H}}_n + \frac{2\boldsymbol{e}_n' \bar{\boldsymbol{H}}_n}{n} \sum_{i=1}^{n} \int_0^1 F(\boldsymbol{e}_n' \boldsymbol{H}_i u) du =$$

$$-\boldsymbol{e}_n' \bar{\boldsymbol{H}}_n \{\frac{2}{n}\sum_{i=1}^{n} \int_0^1 F(\boldsymbol{e}_n' \boldsymbol{H}_i u) du - 1\} =$$

$$-\boldsymbol{e}_n' \bar{\boldsymbol{H}}_n \left\{\frac{2}{n}\sum_{i=1}^{n} \frac{\int_0^{\boldsymbol{e}_n' \boldsymbol{H}_i} F(t) dt}{\boldsymbol{e}_n' \boldsymbol{H}_i} - 1\right\}$$

(*II*$_2$) *I*$_2$ *Continued*.. Rewrite the scaled M-criterion as follows:

$$M_n = \frac{1}{n}\sum_{i=1}^{n} \left| T_i - \boldsymbol{H}_i' \widehat{\boldsymbol{\beta}}_{n-1} - \boldsymbol{H}_i' \boldsymbol{d}_n \right|,$$

where $\boldsymbol{d}_n = \widehat{\boldsymbol{\beta}}_n - \widehat{\boldsymbol{\beta}}_{n-1}$. Assuming $\theta = T_i - \boldsymbol{H}_i' \widehat{\boldsymbol{\beta}}_{n-1}$ and $\tau = \boldsymbol{H}_i' \boldsymbol{d}_n$, and by using the same above arguments, it is seen that M_n is approximated well as follows

$$-\boldsymbol{d}_n' \bar{\boldsymbol{H}}_n \left\{\frac{2}{n}\sum_{i=1}^{n} \frac{\int_0^{\boldsymbol{d}_n' \boldsymbol{H}_i} F(t) dt}{\boldsymbol{d}_n' \boldsymbol{H}_i} - 1\right\},$$

where F is the distribution function of exchangeable sequence $T_i - \boldsymbol{H}_i' \widehat{\boldsymbol{\beta}}_{n-1}$, $i = 1,2,\ldots,n$.

(*II*$_3$) *I*$_3$ *Continued*.. Following part (*c*), the dynamic programming-based recursive estimates of $\boldsymbol{\beta}$ are minimizing $\sum_{t=1}^{\infty} \gamma^t u_t(\boldsymbol{\beta}_t)$, where $\gamma \in (0,1)$ is a pre-determined discount factor and, here, in the case of absolute value loss function, then

$$u_n(\boldsymbol{\beta}_n) = \frac{1}{n}\sum_{j=1}^{n} \left| T_j - \boldsymbol{H}_j' \boldsymbol{\beta}_n \right|.$$

Again, following arguments of part (*c*), the Euler equation is presented as $\gamma u_{1t}(\boldsymbol{\beta}_t) = u_{1(t-1)}(\boldsymbol{\beta}_{t-1})$, at which, in this case

$$u_{1t}(\boldsymbol{\beta}_t) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{H}_i' \text{sign}(T_i - \boldsymbol{H}_i' \boldsymbol{\beta}_n),$$

at which sign is the signature function.

5. Comparisons

To compare a proposed approach, such as extreme learning machine (ELM), with the classical OS-ELM, we can present numerical examples. Let's consider two scenarios: classification and regression tasks, commonly encountered in machine learning.

Example 1: Classification Task. Here, the comparison is done on Iris flowers data set. The dataset comprises some samples of iris flowers, each characterized by four features: sepal length, sepal width, petal length, and petal width. Our mission is to

not only understand the data but also build and optimize a machine learning model for accurate classification.

Dataset: The dataset consists of 150 samples of iris flowers classified into two species based on four features.

Table 1: Comparison in Iris dataset

Methods	*Classical ELM*	*OS-ELM (M-estimation)*
Number of hidden nodes	10	10
Activation function	Sigmoid	Sigmoid
Training time	0.5 seconds	0.3 seconds (improved convergence)
Accuracy	94%	97% (better generalization with adaptive methods)
Average prediction time per sample	0.01 seconds	0.008 seconds (faster predictions)

The proposed approach not only reduces training time but also improves accuracy and prediction speed compared to the classical OS-ELM.

Example 2: Regression Task. In the regression setting, the Boston housing dataset is considered. This dataset concerns the housing prices in the housing city of Boston. The dataset provided has 506 instances with 13 features.

Dataset: Boston housing dataset, which is used to predict housing prices based on certain features.

Table 2: Comparison in Boston housing dataset

Methods	*Classical ELM*	*OS-ELM (M-estimation)*
Number of hidden nodes	50	50
Activation function	ReLU	ReLU
Training time	1.2 seconds	0.8 seconds (parallel computation reduces time)
Mean Absolute Error (MAE)	3.5	2.8 (reduced error due to better optimization)
Average prediction time per sample	0.02 seconds	0.015 seconds

The enhancements lead to a significant reduction in MAE, indicating better performance in predicting housing prices. Additionally, the training and prediction times have improved with the proposed approach.

Conclusion. From these examples, it can be observed that the proposed approach outperforms the classical OS-ELM in terms of accuracy, training time, and prediction speed. Such improvements could be attributed to techniques like adaptive learning rates, parallel processing, or more efficient optimization methods that enhance the model's capability to learn from data efficiently. This experimental comparison highlights the advantages of the proposed method over classical approaches in both classification and regression tasks.

6. Conclusion

OS-ELM is a variant of Extreme Learning Machine (ELM) that focuses on achieving sparsely in the model while maintaining efficiency and accuracy. Here are some potential applications of OS-ELM in financial forecasting, system identification, and online regression:

1. Financial forecasting

Stock Price Prediction: OS-ELM can be employed to predict future stock prices based on historical data, technical indicators, and macroeconomic factors. The sparsely helps in reducing over-fitting while capturing significant features impacting stock movements.

Credit Risk Assessment: Utilizing OS-ELM for credit scoring models can improve the identification of high-risk borrowers by efficiently modeling nonlinear relationships in borrower attributes and historical repayment behavior.

Option Pricing Models: OS-ELM can be applied to model complex relationships in option pricing, enabling better estimation of prices under varying market conditions and volatilities.

2. System identification

Dynamic System Modeling: OS-ELM can be utilized to identify and model dynamic systems in economic environments, enabling the accurate capture of system behaviors and responses to different stimuli over time.

Time Series Analysis: In econometrics, OS-ELM can provide effective models for time series forecasting, capturing underlying patterns in economic indicators while remaining computationally efficient.

Nonlinear System Identification: OS-ELM's ability to handle nonlinearities makes it suitable for identifying and modeling systems where traditional linear approaches may fall short, such as in behavioral finance models.

3. Online regression

Real-time Market Analytics: OS-ELM can support online regression tasks for real-time analytics, allowing practitioners to update models incrementally with new data, thereby improving responsiveness to market changes.

Algorithmic Trading: In algorithmic trading strategies, OS-ELM can be employed for continuous updates of trading models based on incoming market data, leading to improved decision-making under uncertainty.

Adaptive Risk Management: By utilizing OS-ELM for real-time risk factor modeling, financial institutions can dynamically adjust their risk profiles as new information becomes available, enhancing their ability to manage exposure effectively.

Finally, the utilization of OS-ELM in these domains provides a robust framework for tackling complex financial problems characterized by nonlinearity and large datasets. Its efficiency in handling big data and maintaining model interpretability makes it a promising tool for researchers and practitioners in finance. The potential for real-time adaptability further enhances its appeal in rapidly changing economic landscapes.

References

1. El-Melegy M.T., Essai M.H., Ali A.A. "Robust training of artificial feed forward neural networks". Studies in Computational Intelligence 201, 217 – 242, 2009.
2. Hilbe J. M. and Robinson A. P., Methods of statistical model estimation. CRC Press Boca Raton, 2013.
3. Huang G.-B., Zhu Q.-Y., and Siew C.-K., "Extreme learning machine: theory and applications," Neurocomputing, vol. 70, no. 1-3, pp. 489-501, 2006.
4. Ibragimov M., Ibragimov R., and Walden J., "Heavy-tailed distributions and robustness in economics and finance". Springer, 2015.
5. Johnson, R. A., and Wichern, D. W. (2007). "Applied Multivariate Statistical Analysis". Pearson Prentice Hall, 2007.
6. Knight K., "Estimation in dynamic linear regression models with infinite variance errors," Econometric Theory, vol. 9, no. 4, pp. 570-588, 1993.
7. Liang N.-Y., Huang G.-B., Saratchandran P., and Sundararajan N., "A fast and accurate online sequential learning algorithm for feedforward networks," ITNN, vol. 17, no. 6, pp. 1411-1423, 2006.
8. Lim J.-s., Lee S., and H.-S. Pang, "Low complexity adaptive forgetting factor for online sequential extreme learning machine (OS-ELM) for application to non-stationary system estimations," Neural Computing and Applications, vol. 22, no. 3, pp. 569-576, 2013.
9. Ljungqvist, L and Sargent, T. J. "Recursive macroeconomic theory". Second edition. The MIT Press. Cambridge, Massachusetts. London, England, 2004.