Extended Linear Order Statistic (ELOS)  
Aggregation and Regression

Siva K. Kakula\textsuperscript{a}, Anthony J. Pinar\textsuperscript{b}, Timothy C. Havens\textsuperscript{a,\textsuperscript{b}}  
\textsuperscript{a}College of Computing  
\textsuperscript{b}Department of Electrical and Computer Engineering  
Michigan Technological University  
Houghton, MI 49931, USA  
e-mail: \{skakula, ahpinar, thavens\}@mtu.edu

Derek T. Anderson  
Department of Electrical Engineering and Computer Science  
University of Missouri  
Columbia, MO, 65211, USA  
e-mail: andersondt@missouri.edu

Abstract—The ordered weighted average (OWA) operator is a well-known aggregation tool that is primarily used for decision-level fusion. However, the OWA is a convex sum, i.e., its learned coefficients are constrained to sum to one, and thus the output is restricted to lie between the maximum and minimum values of the inputs. Relaxing this constraint on the sum of weights transforms the OWA into a linear order statistic (LOS), which allows the aggregation operation to map the input to any value on the set of reals, thus behaving more like a regression operator. The LOS parameterizes the regression operation of \( d \)-features using just \( d \) parameters, which helps with the model’s interpretability. However, learning just \( d \) parameters limits the amount of non-linear space explored for an optimal solution, and thus reduces the expressibility of the LOS algorithm. We propose a novel aggregation method called the extended linear order statistic (ELOS), where for each position in the sorted input vector we have \( d \) parameters, one for each input feature, thus learning a total of \( d^2 \) weights for the aggregation of \( d \) features. The increased number of parameters helps the algorithm improve its expressibility while maintaining interpretability. In our experiments on real-world benchmark data sets, ELOS has outperformed both linear regression and LOS in 8 out of 10 experiments.

Keywords—ordered weighted average, linear order statistic, linear regression, machine learning, explainable AI

I. INTRODUCTION

The ordered weighted average (OWA) aggregation operator was introduced by Yager in 1988 \cite{Yager1988}. It was primarily designed to aggregate the outputs from multiple decision makers to produce an overall fused decision function. An OWA operator on \( d \) dimensional data is a mapping \( F : \mathbb{R}^d \rightarrow \mathbb{R} \). Given an input vector \( \mathbf{x} = (x_1, x_2, \ldots, x_d) \) and the corresponding weight vector \( \mathbf{v} \), the OWA function is given by

\[
\text{OWA}(\mathbf{x}, \mathbf{v}) = \sum_{i=1}^{d} v_i x_{(i)}, \tag{1}
\]

where \( x_{(1)} \geq x_{(2)} \geq \cdots \geq x_{(d)}, v_i \geq 0, \) and \( \sum_{i=1}^{d} v_i = 1 \).

The OWA induces non-linearity in the solution by sorting the input vector prior to the aggregation operation. It also limits the outputs of aggregation between the minimum and the maximum values of the input sample \( \mathbf{x} \), and thus is best suited for decision-level fusion. In Yager’s later work \cite{Yager1992}, he extended the application of OWA to regression problems. This work introduced an OWA-based approach to evaluate the fitness of a solution to the data, where, the weighting vector of the OWA operator controls the penalties for each data point, based on the magnitude of the error measure (e.g. squared-error). Yager et al. demonstrated that OWA-based regression provides a generic formulation of the regression problem in which existing classical methods like least squares (LS) regression, least absolute deviation (LAD) regression, and maximum likelihood (ML) estimators are special cases. Also, the OWA-based regression solutions were found to be less sensitive to outliers as compared to the traditional methods like LS, LAD, and ML-estimators.

The OWA function at (1) can be modified to

\[
\text{OWA}_g(\mathbf{x}, \mathbf{v}) = \frac{\sum_{i=1}^{d} v_i x_{(i)}}{\sum_{i=1}^{d} v_i}, \tag{2}
\]

where \( x_{(1)} \geq x_{(2)} \geq \cdots \geq x_{(d)}, \) and \( v_i \geq 0 \). While this is equivalent to (1), as it implicitly encodes the constraint on the weights on \( \mathbf{x} \) to sum to 1, it does help with certain learning problems. This form can be relaxed to the linear order statistic (LOS), which has the form

\[
\text{LOS}(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{d} w_i x_{(i)}, \tag{3}
\]

where the weights \( \mathbf{w} \) are no longer constrained to sum to 1, and can also take negative values. This enables the aggregation operation to behave more like a regression operator that can map the input to any value on the set of reals.

An LOS for the aggregation of \( d \) sources is parameterized by \( d \) values, each representing the weight corresponding to each position in the sorted input vector, \( \mathbf{x}_v = (x_{(1)}, x_{(2)}, \ldots, x_{(d)}) \). While having just \( d \) parameters makes the solution more explainable, since we have only a single parameter for each sorted position, the LOS algorithm is quite limited in terms of the amount of non-linear space it explores for an optimal solution—i.e., its “expressibility” is limited. In this paper, we propose a novel aggregation method called the Extended Linear Order Statistic (ELOS), where the aggregation of \( d \) sources is parameterized by \( d^2 \) weights. For each position in the sorted input vector we again have \( d \) weights, one for each source. The increased number of parameters helps the
algorithm improve its expressibility, but it still maintains its interpretability.

The remainder of this paper is organized as follows. Section II presents the background on OWA operators and OWA-based regression, then Section III discusses the problem formulation and training process of LOS. In Section IV we introduce ELOS and describe the training process. Section V discusses the \( \ell_1 \) - and \( \ell_2 \)-regularization. We then compare the performance of ELOS with linear regression and LOS in Section VI. Section VII summarizes this work and discusses possible future work.

II. BACKGROUND AND PRIOR WORK

The OWA has been used in many fields, such as decision making [3–6], risk analysis [7, 8], environment assessment [9, 10], and sports performance analysis [11, 12]. Given the wide range of applications, several OWA-based aggregation operators were proposed. Induced ordered weighted average (IOWA) by Yager et al. [13] introduced a modified ordering approach where the ordering is induced by a variable called the order inducing variable. Chiclana et al. [14] introduced the ordered weighted geometric (OWG) aggregation operator, a geometric mean-based OWA operator. Yager et al. [15] introduced continuous OWA (C-OWA) to aggregate continuous interval values. While most of these developments were oriented towards OWA-based aggregation tools, in 2009, Yager et al. [2] extended the application of OW A to regression particularly outperforms traditional least-squares and least-absolute-deviation methods when the data contains a significant portion of outliers. The ELOS regression approach we propose builds on these prior works and parameterizes the aggregation of d-dimensional inputs using \( d^2 \) parameters, wherein for each of the \( d \) sorted positions in the input we again have \( d \) weights, each corresponding to individual variables in the input vector—more details in Section IV.

III. PROBLEM FORMULATION

Given a set of training data \((y, X)\), where \( X = \{x_1, x_2, \ldots, x_n\}\), \( x_i \subseteq \mathbb{R}^d \) (a set of feature vectors) and \( y = (y_1, y_2, \ldots, y_n)^T \) (a vector of outputs) the classic regression problem involves learning a function that maps the input data \( X \) to the output. Such function is a parameterized model such that

\[
y \approx f(x, w),
\]

where \( w \) is the set of learned parameters of the regression function \( f \). During training, the regression parameters \( w \) are optimized with respect to an error function, usually squared-error,

\[
w^* = \arg \min_w \sum_{i=1}^n (f(x_i, w) - y_i)^2. \tag{4}
\]

Consider the prepended input \( x_i = (x_{i,0}, x_{i,1}, x_{i,2}, \ldots, x_{i,d})^T \), where \( x_{i,0} \) is defined as the constant bias multiplier 1, and \((x_{i,1}, x_{i,2}, \ldots, x_{i,d})^T \) are the \( d \)-features of the input, the function \( f(x_i, w) \) takes the form

\[
f(x_i, w) = \sum_{j=0}^d x_{i,j}w_j = w^T x_i, \tag{5}
\]

where \( w_0 \) is the bias term and each weight in \((w_1, w_2, \ldots, w_d)^T \) is the coefficient of the corresponding variable in the input vector \((x_{i,1}, x_{i,2}, \ldots, x_{i,d})\). This is the well-known least-squares problem with a closed-form solution for \( 1 \).

\[
w^* = (X^T X)^{-1} X^T y, \tag{6}
\]

where

\[
X^T = \begin{bmatrix}
1 & x_1^T \\
1 & x_2^T \\
\vdots & \vdots \\
1 & x_n^T 
\end{bmatrix} \tag{7}
\]

is the \( n \times (d + 1) \) input matrix in which each row is an input vector (with the prepended bias multiplier 1 in the first position), and \( y \) is the vector of outputs in the training set. For more extensive details on regression, in general, we suggest [16].

A. Linear Order Statistic (LOS) Regression

The regression function for LOS takes the same form as \( 5 \) except that the input vectors \( x_i \) are first sorted in descending order,

\[
f_{LOS}(x_i, w) = \sum_{j=0}^d (x_{i,j})_{\pi(i)} w_j = w^T (x_{i,\pi(i)}), \tag{8}
\]

where \( \pi \) is a sorting function, such that \((x_{i,1})_{\pi(i)} \geq (x_{i,2})_{\pi(i)} \geq \cdots \geq (x_{i,d})_{\pi(i)}; (x_{i,1})_{\pi(0)} = 1 \) is defined so that \( w_0 \) represents the bias in the regression. Thus, \( w_1 \) corresponds to the weight on the input variable with the highest magnitude, \( w_2 \) corresponds to the weight on the next highest variable, and so on. The closed-form solution at \( 6 \) also applies to LOS-regression by simply forming the following sorted input data matrix,

\[
X^T_{\pi} = \begin{bmatrix}
1 & (x_{1,\pi_1})_{\pi_1}^T \\
1 & (x_{2,\pi_2})_{\pi_2}^T \\
\vdots & \vdots \\
1 & (x_{n,\pi_n})_{\pi_n}^T 
\end{bmatrix}, \tag{9}
\]

where \((x_{i,\pi_i})_{\pi_i}\) is simply the sorted version of the \( i \)th input vector. Finally, the LOS weight vector \( w \) that minimizes \( 4 \) can be calculated by

\[
w^* = (X_{\pi}^T X_{\pi})^{-1} X_{\pi}^T y. \tag{10}
\]

IV. EXTENDED LINEAR ORDER STATISTIC

While the LOS-regression solution of a \( d \)-dimensional input comprises one weight each for each position in the sorted input and an additional bias parameter, ELOS trains \( d \) weights for each position in the sorted input, where each weight...
corresponds to an individual variable, plus a bias parameter; i.e., the regression solution comprises \( d^2 + 1 \) weights.

Again, consider an input vector \( x_i = (x_{i,1}, \ldots, x_{i,d})^T \), where \( (x_{i,1}, x_{i,2}, \ldots, x_{i,d}) \) are the \( d \)-features of the input. The regression function for ELOS takes the form

\[
 f_{ELOS}(x_i; w) = \sum_{j=1}^{d} (x_i)_{\pi(j)} w_{j,i} + \beta, \tag{11} 
\]

where \( \pi \) is again a sorting function, such that \( (x_i)_{\pi(1)} \geq (x_i)_{\pi(2)} \geq \cdots \geq (x_i)_{\pi(d)} \), and here \( \beta \) is the bias term. The regression weights are now a \( d^2 \)-matrix, as shown in Table I. We first write the ELOS regression in this way for ease of understanding, but later we will extend this formulation for ease of data-driven learning of \( W \) and \( \beta \).

A graphical representation of the associated weights for each input is shown in Table I for a 5-dimensional input vector, \( x = (1.3, 0.7, -0.2, 2.1, 1.6)^T \). The sorting function on this vector would be \( \pi = (4, 5, 1, 2, 3) \); hence, the bold weights in the shown matrix would be the weights applied to this input vector. Essentially, the ELOS combines the power of linear regression with that of the LOS regression: each row of \( W \) is associated with each element of the input vector (like linear regression), and each column of \( W \) corresponds to the sort of the input elements (like LOS regression).

The ELOS formulation at (11) is good for illustrating how ELOS works, but this is problematic for data-driven learning of \( W \) and \( \beta \). Hence, we reform the input vectors \( x \) and the weight matrix \( W \) as follows. It may help to examine Fig. 1 as you read along with the following mathematical explanation. First, consider the extension of the \( i \)th input vector \( x_i \),

\[
 x_i^e = (1, (x_i)_{\pi(i)})^T, \tag{12} 
\]

where the first element of 1 is included so that the bias can be implicitly included in \( w^e \) (which we describe later). The vector \( (x_i)_{\pi(i)}^e \) is a \( d^2 \)-length vector with only \( d \) non-zero terms; this vector will enforce the sort, as indicated by \( \pi \). The vector \( (x_i)_{\pi(i)}^e \) has the form

\[
 (x_i)_{\pi(i)}^e = \left( [x_{i,1}]_{\pi(i)}^e, [x_{i,2}]_{\pi(i)}^e, \ldots, [x_{i,d}]_{\pi(i)}^e \right)^T. \tag{13} 
\]

Each of \( [x_{i,j}]_{\pi(i)}^e \) is simply a vector of zeroes, with each element of \( x_i \) sorted into the corresponding spot in the sort. Figure 1 illustrates the construction of \( (x_i)_{\pi(i)}^e \) for the example input vector \( x = (0.4, -0.1, 0.7)^T \), with sort \( \pi(1) = 3, \pi(2) = 1, \pi(3) = 2 \). The first element of \( (x_i)_{\pi(i)}^e \) is the bias multiplier.
the weight applied to the fourth input \(x_4\) would be \(w_{4,1}\), the weight applied to the fifth element \(x_5\) is \(w_{5,2}\), and so on as shown in Table I. Thus the output is calculated as

\[
y = \beta + w_{4,1}x_4 + w_{5,2}x_5 + w_{1,3}x_1 + w_{2,4}x_2 + w_{3,5}x_3.
\]

\[(18)\]

**Remark 1.** It is easy to show that ELOS is equivalent to linear regression or LOS regression when the weight matrix \(W\) takes a certain form. ELOS is equivalent to linear regression if the rows of \(W\), illustrated in Table I, are constant-valued. That is if \(w_{4,1} = w_{4,2} = \ldots = w_{4,d}, \forall i\).

Similarly, ELOS is equivalent to LOS regression if the columns are equal: \(w_{1,j} = w_{2,j} = \ldots, w_{d,j}, \forall j\).

This Remark illustrates that ELOS can do everything both linear and LOS regression are able to do. The only concern is whether ELOS will over-fit to training data. We now turn to describing how we can apply regularization to the regression methods described in this paper.

**V. Regularization**

While increasing the number of learned parameters might improve the expressibility of the algorithm, more parameters may sometimes capture the noise in the training data and thereby result in an over-fit solution. Regularization allows us to restrict the size of the learned parameters and thus discourages the algorithm from learning a solution that is more complex than necessary. In our experiments on ELOS and comparable regression methods in Section VI, we explored the impact of \(\ell_1\)- and \(\ell_2\)-regularization.

**A. \(\ell_2\)-regularization: Ridge regression**

The sum of squared-error (SSE) function at (4) can be modified to include the \(\ell_2\)-regularization penalty to make the \(\ell_2\)-penalized-SSE function

\[
SSE_{\ell_2} = \sum_{i=1}^{n} (f(x_i, w) - y_i)^2 + \lambda \sum_{j=1}^{d} w_j^2, \quad \lambda \geq 0,
\]

\[(19)\]

where \(\lambda\) is the regularization parameter. Each of the regressions (linear, LOS, and ELOS) at (5), (8), and (15) can be written in the form of a simple dot-product \(w^T x\); hence, (19) can be rewritten as

\[
SSE_{\ell_2} = \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \sum_{j=1}^{d} w_j^2.
\]

\[(20)\]

Expanding (20) gives

\[
SSE_{\ell_2} = (w^T X - y)^T (w^T X - y) + \lambda \|w\|^2.
\]

\[(21)\]

By taking the derivative of (21) and setting to zero, it can be shown that \(SSE_{\ell_2}\) is minimized when

\[
w = (X^T X + \lambda I)^{-1} X^T y,
\]

\[(22)\]

which is the well-known ridge-regression solution. While (22) is notated for linear regression, this can be applied to both LOS and ELOS by replacing \(X\) with \(X_{\tau}\) or \(X^c\) and the appropriate form of the weight vector \(w\). For more extensive detail on \(\ell_2\)-regularization, in general, we suggest [16]. We used the Matlab’s fitrlinear function to apply \(\ell_2\)-regularization, which accounts for numerical issues that can occur with the closed-form solution at (22).

**B. \(\ell_1\)-regularization: Lasso regression**

The SSE function at (4) can be modified to include the \(\ell_1\)-regularization penalty as

\[
SSE_{\ell_1} = \sum_{i=1}^{n} (f(x_i, w) - y_i)^2 + \lambda \sum_{j=1}^{d} |w_j|,
\]

\[(23)\]

where \(\lambda\) is again the regularization parameter. Unlike ridge regression, \(\ell_1\)-regularization does not have a closed-form solution. We used Matlab’s fitrlinear function to apply \(\ell_1\)-regularization. Matlab implements the Alternating Direction Method of Multipliers (ADMM) algorithm [17] to solve for the optimal weight vector \(w\) subject to \(\ell_1\) regularization.

**VI. Experiments**

We tested the ELOS algorithm on real world data sets from the UCI machine learning repository [18]. Using mean squared error (MSE) as the performance measure, we compared ELOS with linear regression and LOS regression on 10 benchmark data sets [2]. We also evaluated the impact of \(\ell_1\)- and \(\ell_2\)-regularization on each of these methods through a grid search over a set of values for the regularization parameter \(\lambda\), ranging on a logarithmic scale between 0.0001 and 1000. We reported the results with the best \(\lambda\). Each experiment consisted of 100 randomized trials, where the result of each trial is the average MSE calculated over a 10-fold cross validation. Table II presents the experimental results, where the MSE reported in each cell is the average MSE of 100 experimental trials; its standard deviation is presented in parentheses. All the experiments are implemented in Matlab.

**A. ELOS versus linear regression**

ELOS, unlike linear regression, learns a weight vector for each feature in the training data—one for each sort position. Figures 2 and 3 compare the weights learned by ELOS and linear regression on Airfoil and Concrete data sets, respectively. In both these figures, we see that the ELOS weights for each feature are spread on either side of the linear regression weights, thus allowing ELOS to treat the features differently depending on their sort order. These figures show that the overall values of the weights of ELOS follow that of the linear regression weights, which is intuitively pleasing.

**B. ELOS vs. LOS**

Figures 4 and 5 compare the learned parameters for ELOS and LOS on the Airfoil and Concrete data sets, respectively. While the LOS has learned one weight for each sorted position, ELOS learns a weight vector for each feature and

\[\text{See Table III in Appendix A for details on the UCI regression data sets used in the experiments.}\]
Fig. 2: Comparison of learned parameters of ELOS and linear regression on Airfoil data set. For each feature, ELOS has learned 5 weights, each corresponding to sort position of that feature, whereas linear regression learns only one weight per feature. ELOS was able to capture non-linearity in the input-output relation, which is represented by the variation in the learned weights for each feature.

Fig. 3: Comparison of learned parameters of ELOS and linear regression on Concrete data set. For each feature, ELOS has learned 8 weights, each corresponding to sort position of that feature.

Fig. 4: Comparison of learned parameters of ELOS and LOS on Airfoil data set. For each sort position, ELOS has learned 5 weights, one for each feature, whereas the LOS has learned only one weight per sort position.

Fig. 5: Comparison of learned parameters of ELOS and LOS on Concrete data set. For each sort position, ELOS has learned 8 weights, one for each feature, whereas the LOS has learned only one weight per sort position.

C. Results on benchmark data sets

Table II shows the performance comparison of ELOS and the other competing methods on real-world data sets. The MSE applies weights according to the sort. In both Figures 4 and 5, the high variance of ELOS weights about the LOS weights for each feature demonstrate the flexibility of ELOS to treat each feature differently based on their sort position. Thus, linear regression and LOS are special cases within ELOS, since ELOS, in addition to learning the weights for individual features, also explores the non-linearity introduced by the sorting of input vector.

C. Results on benchmark data sets

Table II shows the performance comparison of ELOS and the other competing methods on real-world data sets. The MSE
values presented in the table are the average values taken over 100 randomized experimental trials, where the MSE of each trial is the mean MSE over a 10-fold cross validation. The best algorithms on each of these data sets were marked in bold font. We performed a two-sample t-test at a 5% significance level to determine the statistically best results—hence, more than one algorithm can be considered as best. The last column in Table II shows the total number of data sets on which the algorithm produced the best results. Overall, ELOS performed better than Linear regression and LOS in 8 out 10 instances. Regularization did not seem to have a strong impact on ELOS but in some cases, it produced better results. Overall, ELOS performed better than simple linear and LOS regression. ELOS towards predefined structures, which will enable us to identify data sets on which a simpler model like an LOS or a linear regression may be a better fit. We will also explore how ELOS can be instantiated in deep learning architectures, providing explainable layers for deep networks.

Acknowledgement

This work was supported in part by the MTU Institute of Computing and Cybersystems. Superior, a high-performance computing infrastructure at Michigan Technological University, was used in obtaining results presented in this publication.

Appendix A

Benchmark data sets from the UCI machine learning library [18]

Table III contains the information about the data sets used in this paper.

References

A dataset of red vinho verde wine samples, from the north of Portugal. Data set containing values for 8 attributes (molecular descriptors).

Dependent variable

The market historical data set of real estate valuation are collected from Sindian Dist., New Taipei City, Taiwan.

Aquatic toxicity towards the fish (fathead minnow), measured in mol/L.

Aquatic toxicity towards planktonic crustacean Daphnia Magna, measured in mol/L.

Quality (score between 0 and 10)

Quality (score between 0 and 10)

Two variables: Heating load and Cooling load. We randomly chose just the second variable for the regression model.

Residuary resistance per unit weight of displacement

Scaled sound pressure level, in decibels.

Istanbul stock exchange national 100 index


