Extended Linear Order Statistic (ELOS) Aggregation and Regression

Siva K. Kakula[a], Anthony J. Pinar[b], Timothy C. Havens[a,b]
[a]College of Computing
[b]Department of Electrical and Computer Engineering
Michigan Technological University
Houghton, MI 49931, USA
e-mail: {skakula, ajpinar, 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 [1]. 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 $F : \mathcal{R}^d \rightarrow \mathcal{R}$. 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 [2], 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, $\mathbf{x}_\pi = (x_{(1)}, x_{(2)}, \ldots, x_{(d)})$. 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

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

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 ℓ_1- and ℓ_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 OWA to regression problems and demonstrated that OWA-based 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 $(\mathbf{y}, X)$, where $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, $\mathbf{x}_i \subset \mathbb{R}^d$ (a set of feature vectors) and $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$ (a vector of outputs)[1], 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

$$\mathbf{y} \approx f(\mathbf{x}, \mathbf{w}),$$

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

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} \left(f(\mathbf{x}_i, \mathbf{w}) - \mathbf{y}_i\right)^2. \tag{4}$$

Consider the prepended input $\mathbf{x}_i = (x_{i,0}, x_{i,1}, x_{i,2}, \ldots x_{i,d})^T$, 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 $f(\mathbf{x}_i, \mathbf{w})$ takes the form

$$f(\mathbf{x}_i, \mathbf{w}) = \sum_{j=0}^{d} x_{i,j} w_j = \mathbf{w}^T \mathbf{x}_i, \tag{5}$$

where w_o 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 (4),

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

where

$$X^T = \begin{bmatrix} 1 & \mathbf{x}_1^T \\ 1 & \mathbf{x}_2^T \\ \vdots & \vdots \\ 1 & \mathbf{x}_n^T \end{bmatrix} \tag{7}$$

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 $\mathbf{y}$ 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 $\mathbf{x}_i$ are first sorted in descending order,

$$f_{LOS}(\mathbf{x}_i, \mathbf{w}) = \sum_{j=0}^{d} (\mathbf{x}_i)_{\pi_i(j)} w_j = \mathbf{w}^T (\mathbf{x}_i)_{\pi_i}, \tag{8}$$

where π is a sorting function, such that $(\mathbf{x}_i)_{\pi_i(1)} \geq (\mathbf{x}_i)_{\pi_i(2)} \geq \cdots \geq (\mathbf{x}_i)_{\pi_i(d)}$; $(\mathbf{x}_i)_{\pi_i(0)} = 1$ 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,

$$X_\pi^T = \begin{bmatrix} 1 & (\mathbf{x}_1)_{\pi_1}^T \\ 1 & (\mathbf{x}_2)_{\pi_2}^T \\ \vdots & \vdots \\ 1 & (\mathbf{x}_n)_{\pi_n}^T \end{bmatrix}, \tag{9}$$

where $(\mathbf{x}_i)_{\pi_i}$ is simply the sorted version of the ith input vector. Finally, the LOS weight vector $\mathbf{w}$ that minimizes (4) can be calculated by

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

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

[1]Note that the output $\mathbf{y}$ could be extended to multiple outputs for each input vector without loss of generality.

TABLE I: ELOS Weight Matrix for 5-dimensional Data

Input	Sort Order, $\pi(i)$				
$x_{i,1}$	$w_{1,1}$	$w_{1,2}$	$\boldsymbol{w_{1,3}}$	$w_{1,4}$	$w_{1,5}$
$x_{i,2}$	$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$\boldsymbol{w_{2,4}}$	$w_{2,5}$
$x_{i,3}$	$w_{3,1}$	$w_{3,2}$	$w_{3,3}$	$w_{3,4}$	$\boldsymbol{w_{3,5}}$
$x_{i,4}$	$\boldsymbol{w_{4,1}}$	$w_{4,2}$	$w_{4,3}$	$w_{4,4}$	$w_{4,5}$
$x_{i,5}$	$w_{5,1}$	$\boldsymbol{w_{5,2}}$	$w_{5,3}$	$w_{5,4}$	$w_{5,5}$

5×5 weight matrix W for ELOS regression of 5-dimensional data. The weights selected for the aggregation of the example input vector $\mathbf{x} = (1.3, 0.7, -0.2, 2.1, 1.6)^T$ are marked in bold font. Note that we will learn an additional bias weight β, for a total of $5^2 + 1 = 26$ parameters.

corresponds to an individual variable, plus a bias parameter; i.e., the regression solution comprises $d^2 + 1$ weights.

Again, consider an input vector $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,d})^T$, 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

$$f_{ELOS}(\mathbf{x}_i, \mathbf{w}) = \sum_{j=1}^{d} (\mathbf{x}_i)_{\pi_i(j)} w_{j,i} + \beta, \tag{11}$$

where π is again a sorting function, such that $(\mathbf{x}_i)_{\pi_i(1)} \geq (\mathbf{x}_i)_{\pi_i(2)} \geq \cdots \geq (\mathbf{x}_i)_{\pi_i(d)}$, and here β 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 β.

A graphical representation of the associated weights for each input is shown in Table I for a 5-dimensional input vector, $\mathbf{x} = (1.3, 0.7, -0.2, 2.1, 1.6)^T$. 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 β. Hence, we reform the input vectors $\mathbf{x}$ 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 ith input vector $\mathbf{x}_i$,

$$\mathbf{x}_i^e = (1, (\mathbf{x}_i)_{\pi_i}^e)^T, \tag{12}$$

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

$$(\mathbf{x}_i)_{\pi_i}^e = ([(x_{i,1})_{\pi_i}^e]^T, [(x_{i,2})_{\pi_i}^e]^T, \ldots, [(x_{i,d})_{\pi_i}^e]^T)^T. \tag{13}$$

Each of $[(x_{i,j})_{\pi_i}^e]$ is simply a vector of zeroes, with each element of $\mathbf{x}_i$ sorted into the corresponding spot in the sort. Figure 1 illustrates the construction of $(\mathbf{x})_{\pi}^e$ for the example input vector $\mathbf{x} = (0.4, -0.1, 0.7)^T$, with sort $\pi(1) = 3, \pi(2) = 1, \pi(3) = 2$. The first element of $(\mathbf{x})_{\pi}^e$ is the bias multiplier.

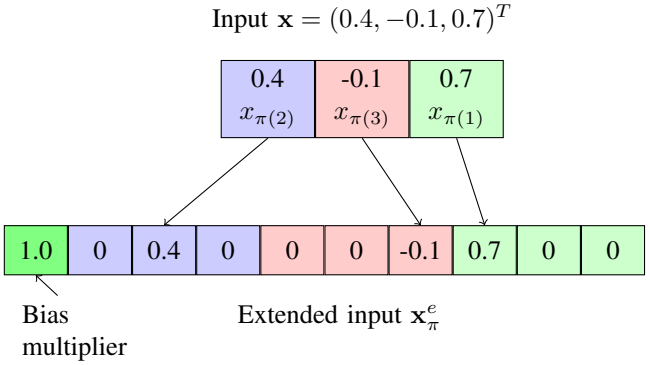

Fig. 1: Transformation of an example 3-dimensional input vector $\mathbf{x}$ to its extended form.

The blue chunk is $(x_1)_{\pi}^e$, where x_1 has been sorted into the second spot. The red chunk is $(x_2)_{\pi}^e$, where x_2 has been sorted into the third spot. And similarity for the green chunk, where x_3 has been sorted into the first spot. While this may seem complicated notation, it significantly simplifies the data-driven learning process. Since each input vector $\mathbf{x}_i$ is extended to $(\mathbf{x}_i)_{\pi_i}^e$, the weight matrix W must be correspondingly extended.

Let the extended form of W be

$$\mathbf{w}^e = (\beta, w_{1,1}, w_{1,2}, \ldots, w_{1,d}, w_{2,1}, \ldots, w_{d,d})^T, \tag{14}$$

where all we have done is take each row of W sequentially to form a long vector and prepended the bias term as the first element of $\mathbf{w}^e$.

We can now rewrite (11) as

$$f_{ELOS}(\mathbf{x}_i, \mathbf{w}) = (\mathbf{w}^e)^T \mathbf{x}_i^e, \tag{15}$$

which you can see is most pleasing—we have essentially written ELOS regression as a linear regression equation.

Finally, it is easy to see that ELOS can be solved much the same as linear and LOS regression were solved,

$$(\mathbf{w}^e)^* = ([X^e]^T X^e)^{-1} [X^e]^T \mathbf{y}, \tag{16}$$

where

$$[X^e]^T = \begin{bmatrix} (\mathbf{x}_1^e)^T \\ (\mathbf{x}_2^e)^T \\ \vdots \\ (\mathbf{x}_n^e)^T \end{bmatrix}, \tag{17}$$

Once the $(d^2 + 1)$-ELOS weight vector $(\mathbf{w}^e)^*$ is learned using a training data set, the ELOS regression output for a new input $\mathbf{x}$ can be calculated using (15). Example 1 demonstrates the ELOS regression calculation in more detail.

Example 1. Consider the problem of learning an ELOS regression model on a 5-dimensional training data set $(\mathbf{y}, X)$, where $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, $\mathbf{x}_i \subset \mathbb{R}^5$ and $\mathbf{y} = (y_1, y_2, ..., y_n)^T$. Consider the input vector $\mathbf{x} = (1.3, 0.7, -0.2, 2.1, 1.6)^T$. The sort order of the variables in $\mathbf{x}$ are $\pi(1) = 4, \pi(2) = 5, \pi(3) = 1, \pi(4) = 2, \pi(5) = 3$. Thus,

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

$$y = \beta + w_{4,1}x_4 + w_{5,2}x_5 + w_{1,3}x_1 + w_{2,4}x_2 + w_{3,5}x_3. \tag{18}$$

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 $w_{i,1} = w_{i,2} = \ldots = w_{i,d}$, $\forall i$.

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 ℓ_1- and ℓ_2-regularization.

A. ℓ_2-regularization: Ridge regression

The sum of squared-error (SSE) function at (4) can be modified to include the ℓ_2-regularization penalty to make the ℓ_2-penalized-SSE function

$$SSE_{\ell_2} = \sum_{i=1}^{n} \left(f(\mathbf{x}_i, \mathbf{w}) - y_i\right)^2 + \lambda \sum_{j=1}^{d} w_j^2,\ \lambda \geq 0, \tag{19}$$

where λ 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 $\mathbf{w}^T\mathbf{x}$; hence, (19) can be rewritten as

$$SSE_{\ell_2} = \sum_{i=1}^{n} \left(\mathbf{w}^T\mathbf{x}_i - y_i\right)^2 + \lambda \sum_{j=1}^{d} w_j^2. \tag{20}$$

Expanding (20) gives

$$SSE_{\ell_2} = \left(\mathbf{w}^T X - \mathbf{y}\right)^T \left(\mathbf{w}^T X - \mathbf{y}\right) + \lambda \|\mathbf{w}\|_2^2. \tag{21}$$

By taking the derivative of (21) and setting to zero, it can be shown that SSE_{ℓ_2} is minimized when

$$\mathbf{w} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}, \tag{22}$$

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_π or X^e and the appropriate form of the weight vector $\mathbf{w}$. For more extensive detail on ℓ_2-regularization, in general, we suggest [16]. We used the Matlab's `fitrlinear` function to apply ℓ_2-regularization, which accounts for numerical issues that can occur with the closed-form solution at (22).

B. ℓ_1-regularization: Lasso regression

The SSE function at (4) can be modified to include the ℓ_1-regularization penalty as

$$SSE_{\ell_1} = \sum_{i=1}^{n} \left(f(\mathbf{x}_i, \mathbf{w}) - \mathbf{y}_i\right)^2 + \lambda \sum_{j=1}^{d} |\mathbf{w}_j|, \tag{23}$$

where λ is again the regularization parameter. Unlike ridge regression, ℓ_1-regularization does not have a closed-form solution. We used Matlab's `fitrlinear` function to apply ℓ_1-regularization. Matlab implements the *Alternating Direction Method of Multipliers* (ADMM) algorithm [17] to solve for the optimal weight vector $\mathbf{w}$ subject to ℓ_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 ℓ_1- and ℓ_2-regularization on each of these methods through a grid search over a set of values for the regularization parameter λ, ranging on a logarithmic scale between 0.0001 and 1000. We reported the results with the best λ. 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

[2]See Table III in Appendix A for details on the UCI regression data sets used in the experiments.

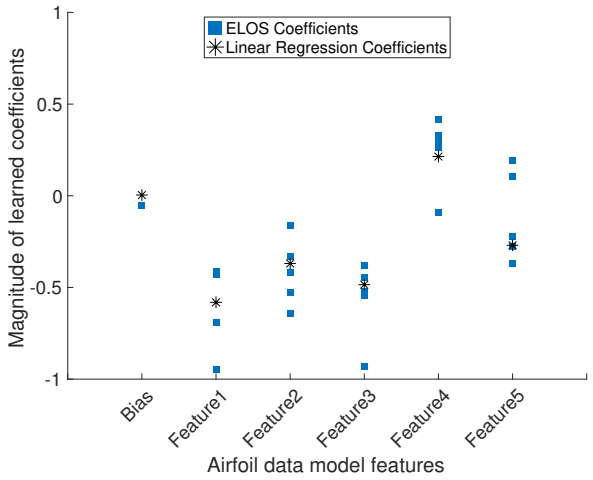

Fig. 2: Comparison of learned parameters of ELOS and linear regression on Airfoil data set. For each feature, ELOS has learned 5 weights, each corresponding to sort position of that feature, whereas linear regression learns only one weight per feature. ELOS was able to capture non-linearity in the input-output relation, which is represented by the variation in the learned weights for each feature.

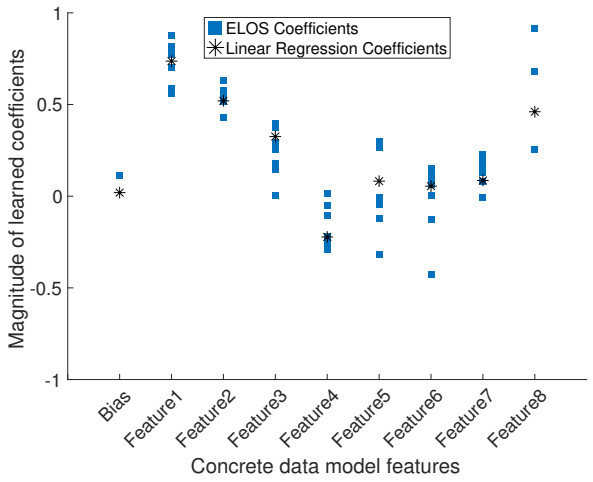

Fig. 3: Comparison of learned parameters of ELOS and linear regression on Concrete data set. For each feature, ELOS has learned 8 weights, each corresponding to sort position of that feature.

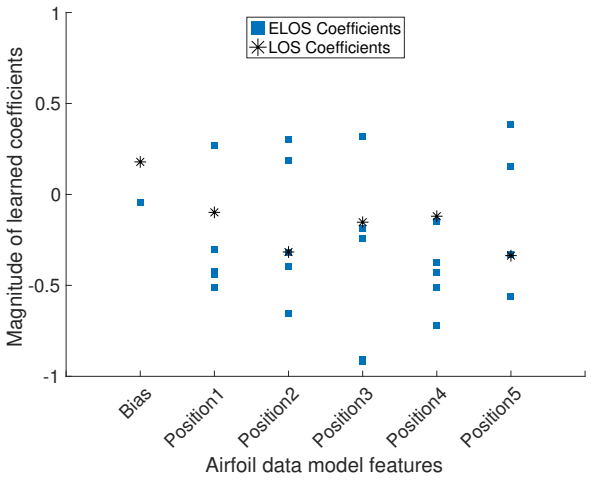

Fig. 4: Comparison of learned parameters of ELOS and LOS on Airfoil data set. For each sort position, ELOS has learned 5 weights, one for each feature, where as the LOS has learned only one weight per sort position.

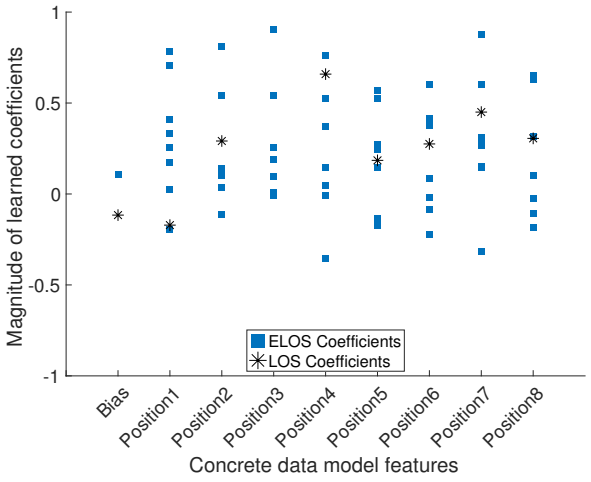

Fig. 5: Comparison of learned parameters of ELOS and LOS on Concrete data set. For each sort position, ELOS has learned 8 weights, one for each feature, where as the LOS has learned only one weight per sort position

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

TABLE II: MSE on Benchmark Data Sets

Method	Concrete	Real Estate	Fish Toxicity	Aquatic Toxicity	Red Wine	White Wine	ENB-2	Yacht	Airfoil	ISE	# of best instances
n	1,030	414	908	546	1599	4898	768	308	1503	536	-
d	8	5	6	8	11	11	8	6	5	7	-
ELOS	**0.345 (0.003)**	0.379 (0.006)	**0.403 (0.004)**	0.558 (0.012)	**0.67 (0.003)**	**0.757 (0.001)**	**0.057 (0.001)**	**0.332 (0.011)**	**0.392 (0.002)**	0.506 (0.008)	7
ELOS-ℓ_1	0.359 (0.004) λ = 0.01	**0.377 (0.005)** λ = 0.0001	0.409 (0.004) λ = 0.001	0.561 (0.007) λ = 0.01	0.681 (0.003) λ = 0.001	0.772 (0.001) λ = 0.001	0.062 (0.001) λ = 0.0001	0.408 (0.01) λ = 0.0001	0.4 (0.002) λ = 0.001	0.496 (0.009) λ = 0.001	1
ELOS-ℓ_2	0.362 (0.004) λ = 0.01	**0.374 (0.005)** λ = 0.01	0.408 (0.004) λ = 0.01	0.567 (0.01) λ = 0.01	0.678 (0.004) λ = 0.01	0.773 (0.001) λ = 0.001	0.059 (0.001) λ = 0.001	0.398 (0.011) λ = 0.0001	0.401 (0.002) λ = 0.001	0.497 (0.009) λ = 0.01	1
Linear	0.424 (0.002)	0.445 (0.002)	0.437 (0.002)	0.553 (0.003)	0.661 (0.001)	0.729 (0.001)	0.087 (0.000)	0.520 (0.003)	0.505 (0.001)	**0.437 (0.003)**	1
Linear-ℓ_1	0.418 (0.003) λ = 0.01	0.445 (0.002) λ = 0.01	0.435 (0.002) λ = 0.01	0.550 (0.005) λ = 0.01	0.660 (0.001) λ = 0.01	0.728 (0.001) λ = 0.01	0.087 (0.001) λ = 0.0001	0.519 (0.004) λ = 0.0001	0.501 (0.001) λ = 0.01	**0.437 (0.003)** λ = 0.001	1
Linear-ℓ_2	0.411 (0.002) λ = 0.1	0.445 (0.002) λ = 0.01	0.434 (0.002) λ = 0.1	0.552 (0.004) λ = 0.1	0.658 (0.001) λ = 0.1	0.728 (0.001) λ = 0.1	0.087 (0.001) λ = 0.0001	0.519 (0.004) λ = 0.0001	0.493 (0.001) λ = 0.1	**0.437 (0.003)** λ = 0.001	1
LOS	0.708 (0.002)	0.727 (0.005)	0.718 (0.003)	0.940 (0.007)	0.996 (0.002)	0.977 (0.001)	0.758 (0.003)	1.086 (0.009)	0.797 (0.001)	0.508 (0.008)	0
LOS-ℓ_1	0.707 (0.002) λ = 0.01	0.727 (0.007) λ = 0.001	0.709 (0.002) λ = 0.01	0.934 (0.007) λ = 0.01	0.987 (0.011) λ = 0.0001	0.977 (0.001) λ = 0.0001	0.754 (0.003) λ = 0.01	1.088 (0.006) λ = 0.0001	0.797 (0.002) λ = 0.01	0.509 (0.006) λ = 0.1	0
LOS-ℓ_2	0.704 (0.002) λ = 0.01	0.723 (0.006) λ = 0.01	0.716 (0.003) λ = 0.01	0.935 (0.005) λ = 0.1	0.985 (0.012) λ = 0.0001	0.977 (0.001) λ = 0.0001	0.748 (0.004) λ = 0.01	1.086 (0.007) λ = 0.0001	0.790 (0.001) λ = 0.1	0.505 (0.004) λ = 0.01	0

MSE values in the table are the average (and standard deviation) of 100 randomized experimental trails; the MSE of each trial is taken over a 10-fold cross validation. Bold indicates the lowest MSE at a 5% statistical significance based on a two-sample t-test.

values presented in the table are the average values taken over 100 randomized experimental trails, 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 since ℓ_1- and ℓ_2-regularized versions performed better than unregularized-ELOS only on two out of 10 data sets.

VII. Conclusion and future work

In this work we introduced ELOS, an OWA-based regression operator and demonstrated that it is a significant improvement over simple linear and LOS regression. ELOS, by learning a weight vector for each input feature—one weight for each sort position—treats each variable independently and also enables the non-linearity introduced by the sorting process. Thus, it combines the benefits of both linear regression as well as LOS regression. Experiments on real-world benchmark data sets indicated the superior performance of ELOS over linear and LOS regression. Furthermore, ELOS maintains the explainability of learned solutions since we can tease apart the treatment of each feature based on its sort position.

Future work will extend the application of ELOS to decision fusion problems. We will also explore the application of regularization strategies that force the learned weights of 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

[1] R. R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Transactions on systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[2] R. R. Yager and G. Beliakov, "Owa operators in regression problems," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 106–113, 2009.

[3] S. Tesfamariam, R. Sadiq, and H. Najjaran, "Decision making under uncertainty—an example for seismic risk management," *Risk Analysis: An International Journal*, vol. 30, no. 1, pp. 78–94, 2010.

[4] R. Sadiq and S. Tesfamariam, "Probability density functions based weights for ordered weighted averaging (owa) operators: An example of water quality indices," *European Journal of Operational Research*, vol. 182, no. 3, pp. 1350–1368, 2007.

TABLE III: Data sets used in the experimental evaluation

Data set name	n	d	Details	Dependent variable
Concrete [19]	1,030	8	Given the composition of concrete (cement, fly ash, fine and course aggregates, water, age, etc., regression problem is the predict the compressive strength.	Concrete compressive strength
Real Estate [20]	414	5	The market historical data set of real estate valuation are collected from Sindian Dist., New Taipei City, Taiwan.	House price per unit area
Fish Toxicity [21]	908	6	Data set containing values for 6 attributes (molecular descriptors) of 908 chemicals used to predict quantitative acute aquatic toxicity towards the fish Pimephales promelas (fathead minnow)	Aquatic toxicity towards the fish (fathead minnow), measured in mol/L.
Aquatic Toxicity [22]	546	8	Data set containing values for 8 attributes (molecular descriptors) of 546 chemicals used to predict quantitative acute aquatic toxicity towards planktonic crustacean Daphnia Magna.	Aquatic toxicity towards planktonic crustacean Daphnia Magna, measured in mol/L.
Red Wine [23]	1599	11	A dataset of red vinho verde wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests	Quality (score between 0 and 10)
White Wine [23]	4898	11	A dataset of white vinho verde wine samples, from the north of Portugal. The goal is to model wine quality based on physicochemical tests	Quality (score between 0 and 10)
ENB-2 [24]	768	8	Energy analysis using 12 different building shapes. The buildings differ with respect to the glazing area, the glazing area distribution, and the orientation, amongst other parameters. Objective is to predict the Cooling load.	Two variables: Heating load and Cooling load. We randomly chose just the second variable for the regression model.
Yacht [25]	308	6	A data set to predict the hydodynamic performance of sailing yachts. Inputs include the basic hull dimensions and the boat velocity.	Residuary resistance per unit weight of displacement
Airfoil [26]	1503	5	NASA data set, obtained from a series of aerodynamic and acoustic tests of two and three-dimensional airfoil blade sections conducted in an anechoic wind tunnel.	Scaled sound pressure level, in decibels.
ISE [27]	536	7	Data sets includes returns of Istanbul Stock Exchange with seven other international index; SP, DAX, FTSE, NIKKEI, BOVESPA, MSCE_EU, MSCI_EM from Jun 5, 2009 to Feb 22, 2011.	Istanbul stock exchange national 100 index

[5] J. M. Merigó and M. Casanovas, "Induced and uncertain heavy owa operators," *Computers & Industrial Engineering*, vol. 60, no. 1, pp. 106–116, 2011.

[6] J. M. Merigo and M. Casanovas, "Decision-making with distance measures and induced aggregation operators," *Computers & Industrial Engineering*, vol. 60, no. 1, pp. 66–76, 2011.

[7] R. Sadiq and S. Tesfamariam, "Developing environmental indices using fuzzy numbers ordered weighted averaging (fn-owa) operators," *Stochastic Environmental Research and Risk Assessment*, vol. 22, no. 4, pp. 495–505, 2008.

[8] N. Feng, X. Yu, R. Dou, and B. Pan, "Managing risk for business processes: A fuzzy based multi-agent system," *Journal of Intelligent & Fuzzy Systems*, vol. 29, no. 6, pp. 2717–2726, 2015.

[9] M. Yeheyis, K. Hewage, M. S. Alam, C. Eskicioglu, and R. Sadiq, "An overview of construction and demolition waste management in canada: a lifecycle analysis approach to sustainability," *Clean Technologies and Environmental Policy*, vol. 15, no. 1, pp. 81–91, 2013.

[10] R. Sadiq, S. A. Haji, G. Cool, and M. J. Rodriguez, "Using penalty functions to evaluate aggregation models for environmental indices," *Journal of environmental management*, vol. 91, no. 3, pp. 706–716, 2010.

[11] S. K. Sharma, R. G. Amin, and S. Gattoufi, "Choosing the best twenty20 cricket batsmen using ordered weighted averaging," *International Journal of Performance Analysis in Sport*, vol. 12, no. 3, pp. 614–628, 2012.

[12] G. R. Amin and S. K. Sharma, "Measuring batting parameters in cricket: A two-stage regression-owa method," *Measurement*, vol. 53, pp. 56–61, 2014.

[13] R. R. Yager and D. P. Filev, "Induced ordered weighted averaging operators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 2, pp. 141–150, 1999.

[14] F. Chiclana, F. Herrera, and E. Herrera-Viedma, "The ordered weighted geometric operator: properties and application in mcdm problems," in *in Proc. 8th Conf. Inform. Processing and Management of Uncertainty in Knowledgebased Systems (IPMU*. Citeseer, 2000.

[15] R. R. Yager, "Owa aggregation over a continuous interval argument with applications to decision making," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 5, pp. 1952–1963, 2004.

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[18] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[19] I.-C. Yeh, "Modeling of strength of high-performance concrete using artificial neural networks," *Cement and Concrete research*, vol. 28, no. 12, pp. 1797–1808, 1998.

[20] I.-C. Yeh and T.-K. Hsu, "Building real estate valuation models with comparative approach through case-based reasoning," *Applied Soft Computing*, vol. 65, pp. 260–271, 2018.

[21] M. Cassotti, D. Ballabio, R. Todeschini, and V. Consonni, "A similarity-based qsar model for predicting acute toxicity towards the fathead minnow (pimephales promelas)," *SAR and QSAR in Environmental Research*, vol. 26, no. 3, pp. 217–243, 2015.

[22] M. Cassotti, D. Ballabio, V. Consonni, A. Mauri, I. V. Tetko, and R. Todeschini, "Prediction of acute aquatic toxicity toward daphnia magna by using the ga-k nn method," *Alternatives to Laboratory Animals*, vol. 42, no. 1, pp. 31–41, 2014.

[23] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems*, vol. 47, no. 4, pp. 547–553, 2009.

[24] 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.

[25] J. Gerritsma, R. Onnink, and A. Versluis, "Geometry, resistance and stability of the delft systematic yacht hull series," *International shipbuilding progress*, vol. 28, no. 328, pp. 276–297, 1981.

[26] T. F. Brooks, D. S. Pope, and M. A. Marcolini, "Airfoil self-noise and prediction," 1989.

[27] O. Akbilgic, H. Bozdogan, and M. E. Balaban, "A novel hybrid rbf neural networks model as a forecaster," *Statistics and Computing*, vol. 24, no. 3, pp. 365–375, 2014.