

Non-monotone Averaging Aggregation

Gleb Beliakov, Shui Yu
School of Information Technology
Deakin University
VIC 3125, Australia
Email: {gleb, syu}@deakin.edu.au

Daniel Paternain
Departamento de Automática y Computación
Universidad Pública de Navarra
31006 Pamplona, Spain.
Email: daniel.paternain@unavarra.es

Abstract—We advance the theory of aggregation operators and introduce non-monotone aggregation methods based on minimization of a penalty for inputs disagreements. The application in mind is processing data sets which may contain noisy values. Our aim is to filter out noise while at the same time preserve signs of unusual values. We review various methods of robust estimators of location, and then introduce a new estimator based on penalty minimisation.

Index Terms—Aggregation operator, penalty function, estimators of location, outliers.

I. INTRODUCTION

The need to aggregate several inputs into a single representative output frequently arises in fuzzy systems. Aggregation of membership values is typically performed in fuzzy rule-based systems by using triangular norms and conorms (t-norms and t-conorms). Averaging aggregation functions, including weighted arithmetic means, ordered weighted averaging (OWA) functions and fuzzy integrals (Choquet and Sugeno integrals in particular) are also frequently used in multicriteria decision making, image analysis and machine learning, see [1]–[3].

By their definition, aggregation functions are monotone in all variables. The role of averaging functions is to provide a representative value of the inputs. Some of the existing averaging functions, however, are not monotone. A particular example is the mode, illustrated in Section II. One can argue that the mode does provide a representative value, as it returns the most frequently observed input. But when the inputs are not repeated, the mode is not defined uniquely: any input is the mode in that case. In this article we investigate other non-monotone averaging functions which play the same role as the mode, but can be used when all inputs are different. This is important when the number of inputs is small, but they can take many values, and specifically any real value from $[0, 1]$, as it happens in fuzzy logic.

Our motivation is the following. Suppose we have a flow of noisy inputs, such as computer network traffic, or digital images. We want to combine the inputs into representative values, for example to reduce the size of the data in image reduction. On the other hand, we want to preserve some interesting aspects of the data, such as narrow spikes in pixel intensity or network traffic. When we aggregate the data using standard averaging functions, these interesting features may be lost. For example a small detail in an image will disappear in

the background. However, we wish to reduce the noise in the data at the same time.

We think of a datum as noise when its value is distinct from some group of similar values of the neighboring data. In the image processing context, too large or too small a value of an individual pixel compared to the neighboring pixels is considered noise. But if two or more neighboring pixels take similar values distinct from the background, they possibly represent a fine detail of an object. Such values should be preserved when averaging across the neighboring pixels. A similar reasoning applies to a flow of network packets in network traffic analysis. A narrow spike in the intensity may be an early indication of a cyberattack.

Our objective is to study averaging operators which have the capability to filter out noise, yet preserve interesting but not dominant features of the inputs.

The structure of the paper is as follows. In Section II we provide preliminary definitions. In Section III we review various definitions of existing non-monotone means. In Section IV we introduce a new type of mode-like aggregation. Conclusions are presented in Section V.

II. PRELIMINARIES

A. Aggregation functions

The research effort concerning aggregation functions, their behavior and properties, has been disseminated throughout various fields including decision making, knowledge based systems, artificial intelligence and image processing. Recent books providing a comprehensive overview include [1]–[3].

Definition 1: A function $f : [a, b]^n \rightarrow [a, b]$ is called an aggregation function if it is monotone non-decreasing in each variable and satisfies $f(\mathbf{a}) = a$, $f(\mathbf{b}) = b$, with $\mathbf{a} = (a, a, \dots, a)$, $\mathbf{b} = (b, b, \dots, b)$.

Definition 2: An aggregation function f is called *averaging* if it is bounded by the minimum and maximum of its arguments

$$\begin{aligned} \min(\mathbf{x}) := \min(x_1, \dots, x_n) &\leq f(x_1, \dots, x_n) \\ &\leq \max(x_1, \dots, x_n) := \max(\mathbf{x}). \end{aligned}$$

It is immediate that averaging aggregation functions are idempotent (i.e., $\forall t \in [a, b] : f(t, t, \dots, t) = t$) and (because of monotonicity) vice versa. Then clearly the boundary conditions $f(\mathbf{a}) = a$, $f(\mathbf{b}) = b$ are satisfied.

A function is called *internal* if its output coincides with one of the inputs. A function f is called *shift-invariant*, if $f(\mathbf{x} + \mathbf{a}) = f(\mathbf{x}) + a$, where $\mathbf{a} = (a, a, \dots, a)$. A function f is *homogeneous* (of degree one) if $f(\lambda \mathbf{x}) = \lambda f(\mathbf{x})$.

Well known examples of averaging functions are the arithmetic mean and the median. It is known that the arithmetic means and the median are solutions to simple optimization problems, in which a measure of disagreement between the inputs is minimized (see [2], [4]–[6]). The reasoning is as follows. Let $\mathbf{x}$ be the inputs and y the output. If all the inputs coincide $x = x_1 = \dots = x_n$, then the output is $y = x$, and we have a unanimous vote. If some input $x_i \neq y$, then we impose a “penalty” for this disagreement. The larger the disagreement, and the more inputs disagree with the output, the larger (in general) is the penalty. We look for an aggregated value which minimizes the penalty.

Let us define a suitable measure of disagreement, or dissimilarity.

Definition 3: Let $P : [a, b]^{n+1} \rightarrow \mathbb{R}$ be a penalty function with the properties

- i) $P(\mathbf{x}, y) \geq 0$ for all $\mathbf{x}, y$;
- ii) $P(\mathbf{x}, y) = 0$ if all $x_i = y$;
- iii) $P(\mathbf{x}, y)$ is quasiconvex in y for any $\mathbf{x}$.

The penalty based function is

$$f(\mathbf{x}) = \arg \min_y P(\mathbf{x}, y),$$

if y is the unique minimizer. If the set of minimizers is the interval $[a, b]$ then $y = \frac{a+b}{2}$.

Remark 1: $f(y)$ is quasiconvex if $f(ay_1 + (1-a)y_2) \leq \max\{f(y_1), f(y_2)\}$ for all $a \in [0, 1]$ and all y_1, y_2 within its domain.

In [4] it was shown that any averaging aggregation function can be represented as a penalty based function. Further, the classical means, such as the arithmetic mean and the median are represented via the following penalty functions. The arithmetic mean is the solution to

$$\text{minimize}_y \sum_{i=1}^n (x_i - y)^2$$

whereas the median is a solution to

$$\text{minimize}_y \sum_{i=1}^n |x_i - y|.$$

Not all penalty-based functions are monotone. Let $P(\mathbf{x}, y) = \sum_{i=1}^n p(x_i, y)$, with $p(t, s) = 1, t \neq s$ and $p(t, t) = 0$. Then the minimizer of P is the mode. Note that the mode is not monotone non-decreasing, for example

$$\text{Mode}(5, 5, 5, 3, 3, 2, 2) = 5 > 3 = \text{Mode}(5, 5, 5, 3, 3, 3, 3).$$

Hence mode is not an aggregation function.

III. NON-MONOTONE AVERAGING FUNCTIONS

As we mentioned, the mode has a disadvantage in that it is not well defined when the inputs are distinct. The usual practice to discretize the inputs by assigning frequency values to intervals of equal distance is too sensitive to the choice of the intervals. Here we look at other alternatives which provide a mode-type average.

Our motivation is the following. We think of y as a representative value of the inputs x_i if at least some of the inputs are grouped together around y , i.e. form a compact cluster around y . Other inputs could be far from y , in which case we do not take them into account to avoid any bias toward them. Consider the following inputs $\mathbf{x} = (1, 2, 3, 6, 6, 1)$. It appears that the two largest values form a cluster, whereas the first three inputs are dispersed. Hence we may want to choose y near 6 as a representative of the inputs. If x_5 were 6, then the mode would have given us $y = 6$.

The median is one aggregation function robust with respect to outliers in the data. In our example the median is 3, and it does not represent a cluster around 6. Let us now recall other robust estimators of location, which appeared in statistical literature, see e.g., [7].

Definition 4: The *shorth* of a sample is the arithmetic mean of the shortest contiguous subsample with the half of the observations.

To calculate the shorth, we sort the inputs in ascending order and then find the smallest of the differences $x_{(h)} - x_{(1)}, x_{(h+1)} - x_{(2)}, \dots, x_{(n)} - x_{(n-h+1)}$, with $h = \lfloor \frac{n}{2} \rfloor + 1$. Let the smallest difference be $x_{(h+j-1)} - x_{(j)}$. Here and in the rest of this paper $x_{(i)}$ denotes the i -th smallest input. Then the inputs falling into the interval $[x_{(h+j-1)}, x_{(j)}]$ are averaged.

A related quantity is the least median of squares estimator [7], [8], defined by

$$LMS(\mathbf{x}) = \arg \min_y \text{Med}(x_i - y)^2.$$

Here we also take the shortest contiguous half of the sample, but instead of taking the mean, we take the midpoint of the shortest half. If several shortest halves exist, one could take the average of their midpoints. LMS is based on minimizing a special penalty function, cf. Definition 3, which is not quasiconvex in y (and hence several optima may exist).

Another interesting example is the least trimmed squares estimator, see [7], [8],

$$LTS(\mathbf{x}) = \arg \min_y \sum_{i=1}^h r_i^2.$$

with the residuals $r_i = x_i - y$ and $r_{(i)}$ denoting the i -th smallest (by absolute value) residual. Here, in difference to the LMS estimator, the mean of each contiguous half of $\mathbf{x}$ is computed,

$$\bar{x}^{(i)} = \frac{1}{h} \sum_{j=i}^{i+h-1} x_{(j)},$$

and then the corresponding sum of squares is computed

$$SQ^{(i)} = \sum_{j=i}^{i+h-1} (x_{(j)} - \bar{x}^{(i)})^2.$$

The LTS solution is the mean $LTS(\mathbf{x}) = \bar{x}^{(i)}$ which corresponds to the smallest $SQ^{(i)}$. However, it may be different from the shorth.

The least Winsorized squares estimator is

$$LWS(\mathbf{x}) = \arg \min_y \sum_{i=1}^h r_{(i)}^2 + (n-h)r_{(h)}^2.$$

Explicitly, the LWS is the midpoint of the contiguous half which leads to the lowest value of the objective function above.

The averaging functions in this section are symmetric, homogeneous and shift-invariant, but not necessarily monotone or continuous in x_i . Take, for instance, the shorth. We have $Shorth(0, 0.6, 1) = 0.8 > Shorth(0.4, 0.6, 1) = 0.5$.

We note that the functions considered in this section are robust to outliers, the components of $\mathbf{x}$ with aberrantly large or small values. The median is also robust to outliers. The resulting value y remains the same even if half of the components of $\mathbf{x}$ are replaced with arbitrary (possibly infinite) values. However, none of the mentioned functions provides the output close to the desired value 6 in our example.

IV. NON-MONOTONE AVERAGES MINIMIZING A PENALTY

Let us return to Definition 3 and consider other types of penalties.

We are interested in identifying clusters of similar values in the inputs, and taking as the output representative values of such clusters. An input vector may have more than one cluster, and in this case we choose the most compact and large cluster, in some way balancing these two contradictory selection criteria. This can be achieved, in principle, by a clustering algorithm, such as k-means. However, the input vector can also contain only one cluster, which makes clustering methods struggle.

We define an averaging function as follows

$$f(\mathbf{x}) = \arg \min_y P(\mathbf{x}, y) = \arg \min_y \sum_{i=1}^n w_i p(|r_{(i)}|), \quad (1)$$

where p is a penalty term for disagreement with the output. We observe that as in the LTS estimator, the residuals $r_i = x_i - y$ are ordered according to their magnitude. As opposed to LTS, all residuals are taken into account. The function p is chosen as

$$p(t) = \begin{cases} t & \text{if } t \leq C, \\ Cs & \text{otherwise.} \end{cases}$$

C is a threshold which reflects scale, and s is a parameter which reflects the relative weighting of the inputs far from the output y .

The reasoning is as follows. On one hand we want y to represent compact clusters of the inputs, and not be affected by inputs far away from y , which can be seen as noise. On

the other hand, we want to account for the number of inputs far from y : the smaller is such number, i.e., the larger is the cluster around y , the smaller is the penalty. Finally, we want to have a robust estimate of scale, i.e., C should be related to the inputs values rather than fixed a priori.

For a fixed y , we define C as a multiple of the smallest residual which is not zero. If all the inputs are different, and y coincides with one of the inputs, we take $C = M \times |r_{(2)}|$, and we fix M at 3. Then p will account for all the residuals within C units from y . The inputs outside this range will also contribute to the total penalty P , but only in terms of their number, not their distances from y . This way C is determined automatically.

By construction, the penalty P is aimed to: a) penalize non-compact clusters (through the choice of p and C), b) penalize small clusters (controlled by s), c) penalize y away from the centers of clusters. Among two clusters of inputs of the same size C , P will be smaller for the cluster which contains more data. Among two clusters of the same number of inputs, P will be smaller for the more compact cluster with smaller C . The parameter s controls the relative weighting of two criteria: the size and compactness of clusters. The larger s , the more emphasis is on the size of the cluster. The value s is the only user-defined parameter in our approach.

The weights w_i are associated with the individual inputs, not their distances from y . That is, they can be seen as weights of relative importance of the inputs, and can be omitted if all the inputs have the same importance.

Let us now examine some properties of the function f . First, f is shift-invariant, and symmetric when all the weights are equal. Second, f is also homogeneous. This can be seen from the fact that $p(\lambda x) = \lambda p(x)$, and that the residuals r_i are shift-invariant and homogeneous. The function f is also idempotent, but it is not monotone. The latter can be seen by aggregating the values in the following example: $\mathbf{x}_1 = (0, 1, 2, 3, 3)$ and $\mathbf{x}_2 = (2, 2, 2, 3, 3)$, which results in $f(\mathbf{x}_1) = 3 > 2 = f(\mathbf{x}_2)$.

Minimization of (1) has to be performed numerically, and it requires a global optimization method. For univariate minimization as in (1) we suggest Pijavski-Shubert method [9], see also [10]–[12], which is deterministic and numerically efficient. A significant simplification comes when the function f can be internal. Then the values of $P(\mathbf{x}, y)$ are calculated for all the inputs, and the input with the smallest value of P is returned. The computational complexity in this case is $O(n^2)$, n is the length of the input vector, because a selection algorithm to determine C runs in $O(n)$ time. This is the same complexity as that of a clustering method, when all x_i are tested as cluster centers (we remind that k-means clustering also leads to a global optimization problem, whose objective has complexity $O(n)$, same as in the proposed method).

For the example input vector we considered earlier $\mathbf{x} = (1, 2, 3, 6, 6.1)$, our approach gives $f(\mathbf{x}) = 6$ or 6.1 , as desired. In contrast, $med(\mathbf{x}) = 3$, $Shorth(\mathbf{x}) = LMS(\mathbf{x}) = LTS(\mathbf{x}) = 2$, and mode can be any of the inputs.

Let us now look at the following illustrative example. Consider calculation of the moving average, which is often

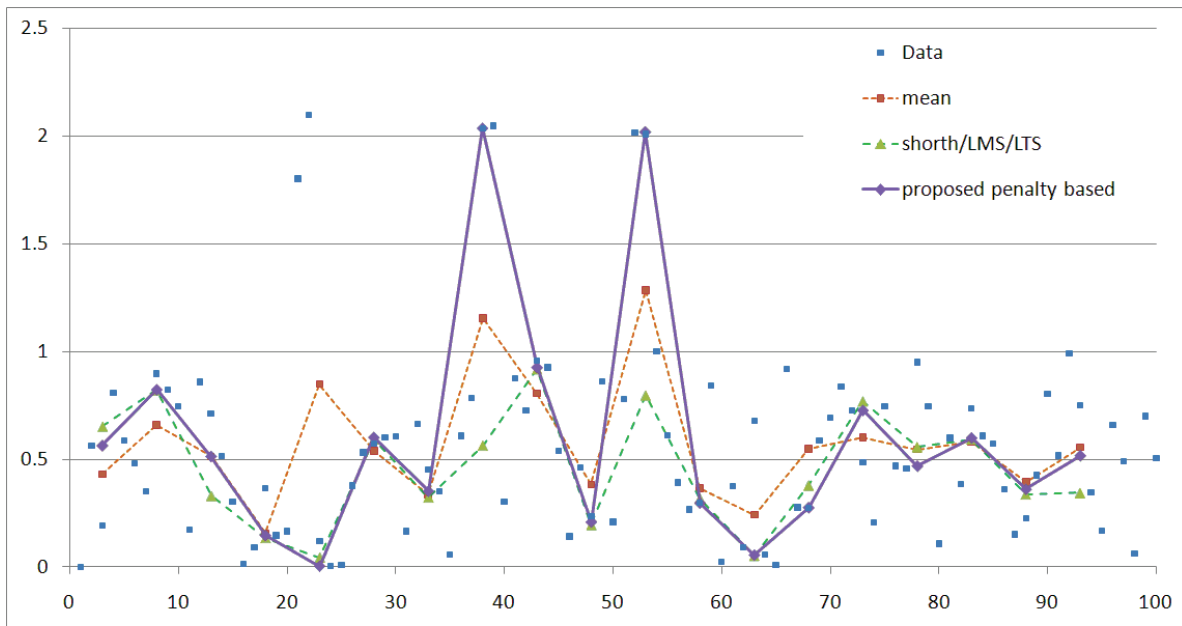


Fig. 1. Illustration of various averages in the moving average example with $m = 5$. The shorth, LMS and LTS estimators give almost the same values in this example and are represented with the dashed line. The arithmetic mean is illustrated with the dotted line. The proposed penalty based average is the solid line. It is the only method which picks two events of interest near 2, and discards abnormal data at positions 21 and 22, because they do not form a compact cluster. Hence these values are considered noise. At other places this method gives the averages similar to shorth/LMS/LTS methods.

used to reduce the size of a time series. We calculate an average of each m consecutive values in the data set, and reduce the length of the data set by the same factor m . Here we randomly generated 100 points with the uniform distribution $U(0, 1)$, and we replaced the data at positions 21, 22, 38, 39, 52 and 53 with (unequal) random values near 2. These spikes in data represent events of interest (e.g., in anomaly detection). The data and various means we considered, along with the simple moving average (for $m = 5$), are illustrated in Figure 1.

We observe that the existing methods filter out the events of interest at the above mentioned positions. The arithmetic mean is affected by the outliers, and results in the values not representative of the corresponding subsamples. The proposed method gives averages close to those of shorth/LMS/LTS methods, but pinpoints the outliers, if they form clusters.

V. CONCLUSIONS

We discussed aggregation operators which are not monotone with respect to the inputs. Our motivation was to calculate the outputs representative of inputs that form clusters. Several robust location estimators have been recalled. We introduced a penalty-based averaging function, which favors the values representative of compact clusters.

Compared with the median, and several existing robust estimators of location, the proposed method delivers the outputs similar to those of the mode function, which are representatives of compact clusters of the input values. Unlike the mode, it

works when all the inputs are distinct. In the subsequent research we aim at applying such averaging to image processing and analysis of network traffic.

REFERENCES

- [1] G. Beliakov, A. Pradera, and T. Calvo, *Aggregation Functions: A Guide for Practitioners*. Heidelberg, Berlin, New York: Springer, 2007.
- [2] Y. Torra and V. Narukawa, *Modeling Decisions. Information Fusion and Aggregation Operators*. Springer, 2007.
- [3] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap, *Aggregation Functions*. Cambridge: Cambridge University Press, 2009.
- [4] T. Calvo and G. Beliakov, "Aggregation functions based on penalties," *Fuzzy Sets and Systems*, vol. 161, pp. 1420–1436, 2010.
- [5] T. Calvo, R. Mesiar, and R. Yager, "Quantitative weights and aggregation," *IEEE Trans. on Fuzzy Systems*, vol. 12, pp. 62–69, 2004.
- [6] R. Yager and A. Rybalov, "Understanding the median as a fusion operator," *Int. J. General Syst.*, vol. 26, pp. 239–263, 1997.
- [7] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. New York: Wiley, 2003.
- [8] P. Rousseeuw, "Least median of squares regression," *Journal of the American Statistical Association*, vol. 79, pp. 871–880, 1984.
- [9] S. Pijavski, "An algorithm for finding the absolute extremum of a function," *USSR Comput. Math. and Math. Phys.*, vol. 2, pp. 57–67, 1972.
- [10] G. Beliakov, "Extended cutting angle method of global optimization," *Pacific J. of Optimization*, vol. 4, pp. 153–176, 2008.
- [11] G. Beliakov and J. Ugon, "Implementation of novel methods of global and non-smooth optimization: GANSO programming library," *Optimization*, vol. 56, pp. 543–546, 2007.
- [12] G. Beliakov, "The cutting angle method - a tool for constrained global optimization," *Optimization Methods and Software*, vol. 19, pp. 137–151, 2004.