ON THE USE OF THE HUBER ESTIMATOR IN NONLINEAR IMAGE PROCESSING

Charles G. Boncelet Jr.

University of Delaware
Newark DE 19711
boncelet@udel.edu

1. ABSTRACT

Despite considerable appeal in the statistics literature, the Huber estimate is little used in engineering. We believe this is due primarily to two factors: difficulty computing the estimate and the need for a corresponding scale estimate. We present a variation of the Huber estimate which we call the "trimmed-Huber" estimate that addresses both of these concerns. A fixed fraction of the data points will be "trimmed off", but unlike the trimmed mean, these data points do not have to by symmetrically trimmed.

2. INTRODUCTION

The use of robust statistics to combat non-Gaussian (generally heavy tailed) noises has a long history, dating back thousands of years. Starting in earnest in the 1960's and 70's, considerable scientific study of robust estimators was completed. Two excellent volumes were written summarizing this work, one by Huber [5] and one by Hampel [3].

Briefly, three classes of robust estimators were considered: L, M, and R estimates. The "L" estimates are those derived from linear combinations of order statistics. "M" estimates are generalizations of maximum likelihood estimates in that they minimize a loss function. Finally, the "R" estimates are based on rank tests. Since R-estimates are not germane to the rest of this paper, they will be ignored hereon.

The engineering community is interested in robust estimates for two reasons: Firstly, in many applications, heavy tailed noise is present. For example, transmission errors often introduce "salt and pepper" noise—black and white dots—into images. Generally speaking, salt and pepper noise is easily filtered out with simple nonlinear filters, e.g., [4].

The first estimators considered in the engineering literature to eliminate these noises were the median filters [2]. The median can output a finite value as long as less than half the data points are finite.

Over the years numerous generalizations of the median have been proposed. These include recursive medians, linear combinations of order statistics and its simpler variant, the trimmed mean, Ll-filters, LUM filters, permutation filters, morphological filters, stack filters, and others.

Secondly, the engineering community has considered robust statistics because the signals in many applications are non-Gaussian in nature. Nowhere is this more apparent than in image processing. Edges and other features are not well modeled by Gaussian statistics. Here, again, the median filter offers advantages. It can pass a simple edge without distortion.

From the statistics point of view, all or almost all these filters are L-estimates (or are based on L-estimates). Statisticians are more likely to consider M-estimates for their robust procedures. One reason for this is that M-estimates generalize more easily to multivariate situations. A likely second reason is that statisticians are less concerned with issues of computational time required and implementations; engineers often want to be able to filter images in "real time."

Interestingly, the image filtering problem is usually implemented in a sliding window fashion, with each instance of the filter producing one estimate of one pixel. In this manner, each estimation is scalar. However, the whole problem is multivariate, not scalar.

3. M-ESTIMATES

M-estimates are the result of minimizing a loss function. For the simple location problem in independent and identically distributed noise (iid), the estimate becomes

$$\hat{x} = \arg\min \sum_{i=1}^{n} \rho(x_i - \hat{x}) \tag{1}$$

Prepared through collaborative participation in the Advanced Telecommunications/Information Distribution Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-002.

After taking a derivative with respect to $\hat{x}$, one obtains "normal equations":

$$0 = \sum_{i=1}^{n} \Psi(x_i - \hat{x}) \tag{2}$$

where $\Psi(r) = d\rho(r)/dr$.

For the least squares estimate, $\rho(r) = r^2/2$ and $\Psi(r) = r$; for the least absolute residuals estimate, $\rho(r) = |r|$ and $\Psi(r) = \text{sign}(r)$.

Huber introduced the concept of "least favorable" distributions and the corresponding minimax loss function:

$$\hat{x}_H = \arg\min \sum_{i=1}^{n} \rho_H(x_i - \hat{x}_H) \tag{3}$$

or, equivalently,

$$0 = \sum_{i=1}^{n} \Psi_H(x_i - \hat{x}_H) \tag{4}$$

where

$$\rho_H(r) = \begin{cases} r^2/2 & |r| \le k \\ k|r| - k^2/2 & |r| \ge k \end{cases} \tag{5}$$

and

$$\Psi_H(r) = \max(\min(k, r), -k) \tag{6}$$

The Huber location estimate has a number of desirable properties:

- As $k \to +\infty$, the estimate reduces to the sample mean; as $k \to 0$, the estimate reduces to the sample median. Thus, k can be considered as a robust tuning parameter. Small k's yield robust estimates, while large k's result in greater averaging.
- Heuristically at least, the Huber loss function makes sense. Small errors, those most likely to be Gaussian in origin, are weighted quadratically; large errors, those more likely to be outliers, are given less weight than ordinary squared error does.
- The Huber loss function is convex. Other, more robust, loss functions are not. Convexity implies that the estimate can be computed by procedures that search for local optima.

The Huber estimate is not without criticism:

- In some situations, more robustness is needed. The loss function should increase less rapidly than $|r|$, or even decrease. Note, these loss functions are nonconvex which makes computation problematic.
- Generally, there is little guidance in how to choose k, especially in time-varying situations (which make it difficult to measure a local estimate of scale).
- Too much computation may be required, even though the loss function is convex.

In the next section, we address the computation and scale issues and propose solutions.

4. THE TRIMMED-HUBER FILTER

We suggest choosing k so that a fixed fraction of the data points are "trimmed off". We call this estimate the "trimmed-Huber" estimate (or *filter*, depending on the application). The trimmed-Huber estimate is similar to the trimmed mean, but with one very important difference: the points can be trimmed off asymmetrically. Exactly which points get trimmed off are determined by the data. Below we illustrate a relatively simple algorithm that computes the trimmed-Huber estimate. This algorithm was originally presented in [1] in the context of multivariate robust regression.

The idea behind the algorithm is to consider k as a parameter that can be changed or adjusted. Initially, $k = +\infty$ and the Huber estimate coincides with the sample mean, which is trivial to compute. Then k is reduced and the optimal estimate is continually adjusted until the desired number of data points are trimmed off. Note, if $k \to +0$, then the Huber estimate reduces to the sample median.

For the moment, consider k fixed, and let $\hat{x}_H(k)$ denote the optimal Huber estimate as a function of k. Define the following three sets, $A = \{i : x_i - \hat{x}_H(k) \le -k\}$, $B = \{i : -k \le x_i - \hat{x}_H(k) \le k\}$, and $C = \{i : k \le x_i - \hat{x}_H(k)\}$. If some point, say j, has $x_j - \hat{x}_H(k) = \pm k$ then we say that point is at a *corner*. It can be arbitrarily assigned to either of the two possible sets.

The normal equations reduce to

$$0 = \sum_{i \in A}(-k) + \sum_{i \in B}(x_i - \hat{x}_H(k)) + \sum_{i \in C} k \tag{7}$$

Letting n_A equal the number of elements in A, n_B in B, and n_C in C, we can easily solve for $\hat{x}_H(k)$

$$\begin{aligned} \hat{x}_H(k) &= \frac{\sum_{i \in B} x_i}{n_B} + k\frac{n_C - n_A}{n_B} && (8) \\ &= \hat{x}_B + k\frac{n_C - n_A}{n_B} && (9) \end{aligned}$$

where $\hat{x}_B$ is the least squares estimate (sample mean) based only on those points in B. If the partition yields a consistent $\hat{x}_H(k)$, then we say the partition is *valid*.

As a comment, this suggests a conceptually simple algorithm for computing $\hat{x}_H(k)$: guess a partition, compute $\hat{x}_H(k)$, and check to see if the partition is valid. If so, then we are done; if not, guess a new one and repeat. While this may be acceptable in some situations, without guidance as to how the guessing should be done, the number of partitions checked may be unacceptably large. We will not consider this approach any further in this paper.

The initial valid partition is $n_A = 0, n_B = n, n_C = 0$, and $k = +\infty$, corresponding to the sample mean. Now assume that we have a valid partition for some $k > 0$. Compute $\hat{x}_H(k)$ as above. Now reduce k until some point is at a corner. Move it from set B to set A or C (whichever is appropriate).

Assume the point is moving from set B to set A. Then,

$$-k \leq x_i - \hat{x}_B - k\frac{n_C - n_A}{n_B} \tag{10}$$

Simple rearrangement yields

$$k \geq \frac{\hat{x}_B - x_i}{1 - \frac{n_C - n_A}{n_B}} \tag{11}$$

Similarly, if the point is moving from B to C,

$$k \geq \frac{x_i - \hat{x}_B}{1 + \frac{n_C - n_A}{n_B}} \tag{12}$$

The new partition is valid for a new range of k.

It is straightforward to show that $|\frac{n_C - n_A}{n_B}| \leq 1$. This guarantees that the directions of the inequalities above are correct and also that no point ever moves from A to B or from C to B.

The only candidates to move out of B an any step are the smallest (to A) and the largest (to C). If the data is presorted, then only these two points need to be checked and the overall complexity will be $O(n)$ (except for the sorting which will require $O(n \log n)$ in general.)

Thus the algorithm is as follows:

1. Presort the data. Set $n_B = n$, $n_A = 0$, and $n_C = 0$ and compute $\hat{x}_B = \sum_{i=1}^{n} x_i$ and $k = \max |x_i - \hat{x}_B|$.
2. Do until finished,
   (a) Check the smallest and largest elements in B to see which leaves.
   (b) Move that point out of B. Decrement n_B, and increment n_A or n_C.
   (c) Compute the new estimate.
3. Compute $\hat{x}_H(k) = \hat{x}_B + k(n_C - n_A)/n_B$.

Each time through the loop requires $O(1)$ operations and there are at most $n - 1$ times through the loop. Thus, on presorted data, $O(n)$ operations are required.

The trimmed-Huber estimate has the following advantages:

- The amount of computation needed to compute the estimate is completely predictable.
- When the samples are trimmed off symmetrically (so that $n_A = n_C$), the estimate coincides with the trimmed mean.

Step	n_A	n_B	n_C	$\hat{x}_B$	k	$\hat{x}_H(k)$
0	0	6	0	63/6	129/6	63/6
1	0	5	1	31/5	49/6	235/30
2	0	4	2	15/4	11/2	26/4
3	1	3	2	14/3	4	6
4	2	2	2	6	2	6

- The estimate can trim off the data points asymmetrically. This is entirely reasonable, especially in the usual image processing situation of a fairly small sliding window.

An edge region will contain points that are dissimilar from one another. The Huber estimate can trim off those points asymmetrically. Consider a one-dimensional signal with a perfect edge. Let the first l points be 0 and the next $n - l$ be 1. Then as long as the number of points being trimmed off is at least $n/2$, the trimmed-Huber estimate will pass this edge perfectly. The only trimmed mean that can pass the edge perfectly is the limiting case of the median.

5. NUMERICAL RESULTS

As an example of how the computations go, consider the following (contrived) data set: $X = \{1, 2, 4, 8, 16, 32\}$. There are 6 data points. The results of computing the median are listed in the table below. (We are not suggesting that this is an efficient algorithm for computing the median, merely that the median is the end result if the computation is pursued that far.)

This example illustrates some interesting points. Note that the first two samples trimmed off are both on the same side. As the estimates approach the median, $n_A \approx n_C$. (Recall, $|n_C - n_A| \leq n_B$.) Also, the contrived nature of the data results in a one-directional convergence to the median; this is not true in general.

We also consider present the results of a simple image filtering experiment. Two images, lena and aerial, both 512 × 512 with 8 bits per pixel, were filtered both with and without Gaussian noise by four sliding window filters: the mean, median, trimmed mean, and trimmed-Huber. In all cases, the window was 3 × 3. The trimmed mean and trimmed Huber both trimmed off 4 of the 9 points. Lena is typical of a fairly smooth image; aerial is a more detailed image.

From the tables, we can see several things: The trimmed-Huber estimate generally performs best, especially when the noise is small or the image has sharp features. (Not shown here, but the trimmed-Huber estimate produces the best looking images.) The mean generally does the worst.[1]

[1] A fact which should surprise no one at this meeting!

	$\sigma = 0$		$\sigma = 5$		$\sigma = 20$	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Mean	5.26	3.27	5.23	3.63	8.52	11.12
Trim Mean	4.50	2.81	4.94	3.35	8.75	6.81
Trim-Huber	4.35	2.67	4.84	3.30	8.89	6.93
Median	4.25	2.25	4.93	3.37	9.65	7.54

Table 1: Measured RMSE and MAE for the Lena image filtered by four different filters.

	$\sigma = 0$		$\sigma = 5$		$\sigma = 20$	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Mean	11.12	6.75	11.25	7.02	12.98	9.23
Trim Mean	9.73	5.49	9.99	6.03	12.60	9.08
Trim-Huber	9.43	5.10	9.73	5.78	12.64	9.12
Median	9.54	4.51	9.91	5.69	13.36	9.71

Table 2: Measured RMSE and MAE for the Aerial image filtered by four different filters.

6. CONCLUSIONS

This paper is merely a beginning at exploring the trimmed-Huber estimator. Future work will consider down-weighting samples further from the center and the search for optimal trimming fractions. Nevertheless, we believe the results presented here are encouraging: The trimmed-Huber estimator combines much of the averaging capability of the mean and the edge passing ability of the median. It will outperform the corresponding trimmed mean in salt and pepper noise since it can trim off points asymmetrically.

The trimmed Huber estimate can be extended to more general multivariate situations and can (relatively easily) accept equality and inequality constraints.

7. DISCLAIMER

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

8. REFERENCES

[1] C. G. Boncelet Jr. and B. W. Dickinson. A variant of huber robust regression. *SIAM J. Sci. Stat. Comput.*, 5(3):720–734, September 1984.

[2] N. C. Gallagher, Jr. and G. L. Wise. A theoretical analysis of the properties of median filters. *IEEE Trans. on Acoust. Speech, Signal Proc.*, ASSP-29(6):1136–1141, December 1981.

[3] F. R. Hampel. *Robust Statistics: The Approach based on Influence Functions.* Probability and Mathematical Statistics Ser. Wiley, New York, 1985.

[4] R. C. Hardie and C. G. Boncelet Jr. LUM filters: a class of order statistic based filters for smoothing and sharpening. *IEEE Trans. on Signal Processing*, 41(3):1061–1076, March 1993.

[5] P. J. Huber. *Robust Statistics.* J. Wiley & Sons, New York, NY, 1981.