PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Classification based on fast and robust approximations to order statistics

Palm, Hans Christian

Hans Christian Palm, "Classification based on fast and robust approximations to order statistics," Proc. SPIE 11400, Pattern Recognition and Tracking XXXI, 114000I (22 April 2020); doi: 10.1117/12.2558502

Event: SPIE Defense + Commercial Sensing, 2020, Online Only, California, United States

Classification based on fast and robust approximations to order statistics

Hans Christian Palm*[a]
[a]Forsvarets forskningsinstitutt, P. O. Box 25, N-2027 Kjeller, Norway

ABSTRACT

A test system with four cameras in the infrared and visual spectra is under development at FFI (The Norwegian Defence Research Establishment). The system may be mounted on a jet aircraft or may be used in a land-based version. It can be used for image acquisition or for testing of automatic target recognition (ATR) algorithms. The sensors on board generate large amounts of data, and the scene may be rather cluttered or include anomalies (e.g. sun glare). This means we need algorithms which are robust, fast, able to handle complex scenes, and data from up to four sensors simultaneously. Typically, estimates of mean and covariance are needed for the processing. However, the common maximum likelihood (ML) estimates are in general too sensitive towards outliers. Algorithms based on order statistics are known to be robust and reliable. However, they are computationally very heavy. But approximations to order statistics do exist. Median of medians is one example. This is a technique where an approximation of the median of a sequence is found by first dividing the sequence in subsequences, and then calculating median (of medians) recursively. This technique can be applied for estimating the mean as well as the standard deviation. In this paper we extend this method for estimating the covariance matrix and the mean vector, and discuss the strategy with respect to robustness and computational efficiency. Applications for use in image processing and pattern recognition are given.

Keywords: Order statistics, classification, image processing, segmentation

1. INTRODUCTION

The origin of this work is a test system with four cameras in the infrared and visual spectra which is under development at FFI (The Norwegian Defence Research Establishment) [1]. This system may be mounted on a high speed jet aircraft, in a land-based version, or used in the lab. Applications are image acquisition as well as development and test of automatic target recognition algorithms (ATR). We need fast and robust algorithms for both image processing and visualization. Such algorithms may involve use of order statistics.

Order statistics for estimating position and deviation, like median and quartile difference, is known to be robust [2]. They are little affected by anomalies in a dataset. Thus, they are very useful in applications with data containing outliers. In our applications, sun glare in infrared images is an example of such outliers. However, the drawback with such methods is that they in general are very time-consuming. This is because they often involve sorting of the dataset's elements.

If the dataset consists of integer elements, efficient histogram-based algorithms are available for determining various quantiles. This is the situation if we have 8 bit or 16 bit input from an image sensor. However, this strategy does obviously not work if we have a floating point input. This is the case when the input is a feature image, e.g. a texture image.

We haven't found very much related work reported in the literature. Pratt et al presented the pseudomedian in [3]. This strategy is based on concepts of mathematical morphology, and it can be computed very efficiently. However, it is not robust against outliers. Yang et al has proposed a median filter algorithm without sorting [4]. We haven't studied it further, mainly because it uses integer input. Other approaches which apply medians, (ex. [5]) seem to be very focused on preprocessing of images, especially for removing salt and pepper noise.

*hans-chr.palm@ffi.no

Pattern Recognition and Tracking XXXI, edited by Mohammad S. Alam,
Proc. of SPIE Vol. 11400, 114000I · © 2020 SPIE · CCC
code: 0277-786X/20/$21 · doi: 10.1117/12.2558502

An approximation for the median based on the median of medians has existed for a long time [6], mainly applied for preprocessing in sorting algorithms. This is based on first dividing a sequence into subsequences, next calculating the median on small subsequences and then calculating median (of medians) recursively. In [7] we analyzed the strategy and presented examples of image-analysis applications. We found that it performed quite well. However, it was all on univariate data. Our test system has up to four cameras, and thus it is desirable to extend the algorithm to handle multivariate data.

In the next section we will give a brief description of the median of medians algorithm, next present how it can be extended to handle multivariate data, and then study its characteristics. In section 3 we will present some experiments. Finally, we will summarize and conclude.

2. ESTIMATES BASED ON MEDIAN OF MEDIANS

2.1 Calculating the median of medians

Median of medians is far from a new technique. It was first presented by Blum et al back in 1973 [6]. It seems that it has mainly been used as a preprocessing of sorting algorithms. But there is no reason why it can't be used "as is".

A median of medians will of course be an approximation to a median, but our experience is that it has been a sufficiently good approximation in our applications.

It is easily shown that the median of a sequence with length 3 can be found as

$$\mathrm{med}(\mathrm{x}_1, x_2, x_3) \quad = \quad \max(\min(x_1, x_2), \min(x_1, x_3), \min(x_2, x_3)) \tag{1}$$

This procedure may be extended to larger sequences, but for sequences larger than five, this median-computing procedure becomes impractical.

Let us for a short while restrict ourselves to sequences of length 3^n, $n = 1,2,3,\ldots$. Let's assume we have a sequence of length nine. Then an approximation to the median could be

$$m = \mathrm{med}(\mathrm{med}(x_1, x_2, x_3), \mathrm{med}(x_4, x_5, x_6), \mathrm{med}(x_7, x_8, x_9)) \tag{3}$$

Instead of calculating the median of a sequence of length nine, we can approximate it with calculating four medians, each with length three; three medians of parts of the sequence, and one median of the medians. This strategy is obviously very easy to extend to larger sequence lengths.

The number of medians of length 3 which has to be computed for a sequence of length 3^n is

$$N \quad = \quad \frac{1}{2}(3^n - 1) \tag{4}$$

2.2 Univariate mean and deviation estimates

<u>Mean estimate:</u>

The median of medians is used as a mean estimate.

<u>Standard deviation estimate:</u>

A robust deviation estimate, which we have applied, is based on the median of absolute deviation (MAD) [8]. It is defined as

$$\hat{\sigma} \quad = \quad 1.4826 \cdot \mathrm{MAD}(\mathbf{x}) \quad = \quad 1.4826 \cdot \mathrm{med}(|\mathbf{x} - \mathrm{med}(\mathbf{x})|) \tag{5}$$

where $\mathbf{x}$ is a vector containing the samples. 1.4826 is a scale factor so that estimates based on samples drawn from Gaussian distributions will be unbiased.

2.3 Length not equal to 3^n

In the previous section we assumed the sequence length to be a power of 3. In general, this is not the case. So what do we do when the assumption is not fulfilled? As far as we see, there are two strategies. One is simply to pick 3^n samples of the N samples in the sequence, where n is the largest integer such that $3^n < N$. This implies that $N - 3^n$ samples are discarded, which of course is not desirable. The other strategy is to extend the sequence with $3^{n+1} - N$ randomly picked samples. The serious drawback with this approach is that several samples are used more than once. Hence the samples are definitely not independent any longer.

Both these strategies were tested and evaluated in [7]. We found that the latter one (extending the sequence) performed best. We will therefore use the same strategy here.

2.4 Extension to multivariate data

For multivariate data, mean vector and covariance matrix have to be estimated.

Estimating the mean vector is straightforward from the univariate case. The median is simply determined for each dimension separately.

Concerning covariance matrix, Pasman and Shevlyakov [9] have suggested estimating the correlation coefficient using the correlation median estimator;

$$r_{COMED} = \frac{\text{med}\left(\left(x - \text{med}(x)\right)\left(y - \text{med}(y)\right)\right)}{\text{MAD}(x)\text{MAD}(y)} \tag{6}$$

where x and y are two sequences to be correlated. Hence, an estimate for the covariance is simply to skip the denominator, i.e. the element (i, j) in the covariance matrix is estimated to

$$\mathbf{\Sigma}\left(i, j\right) = \gamma \cdot \text{med}\left(\left(\mathbf{x}_i - \text{med}(\mathbf{x}_i)\right)\left(\mathbf{x}_j - \text{med}(\mathbf{x}_j)\right)\right) \tag{7}$$

where γ is a normalization constant.

2.5 Characteristics

We will carry out three experiments to investigate how well the parameter values are estimated. This will be done by use of Monte Carlo simulations.

In the first experiment, we have drawn samples from a bivariate Gaussian distribution with zero mean and covariance matrix

$$\mathbf{\Sigma} = \begin{pmatrix} 3.0 & -3.5 \\ -3.5 & 7.0 \end{pmatrix}$$

Of course, many other covariance matrices could have been chosen. Our rationale was just to pick a non-diagonal matrix. The mean vector and covariance matrix are estimated using the median of medians. In addition we have also estimated the mean vector and covariance matrix using "full" median, the standard/classical maximum likelihood estimate, and as well as two robust reference methods. One of the robust reference methods is based on an M-estimate [10], and the second one is the minimum covariance determinant estimator (MCD) [11]. We have implemented the first of these ourselves. The LIBRA package (LIBRA – library for robust analysis) [12] was used for the latter one. As evaluation criteria we have used the norm of the mean estimate (which should be "small"), and the trace and sum of absolute differences between the elements of the estimated covariance matrix and the covariance matrix of the distribution the samples are drawn from. The first one should be close to 10, and the second one close to zero. The results are shown in Figure 1.

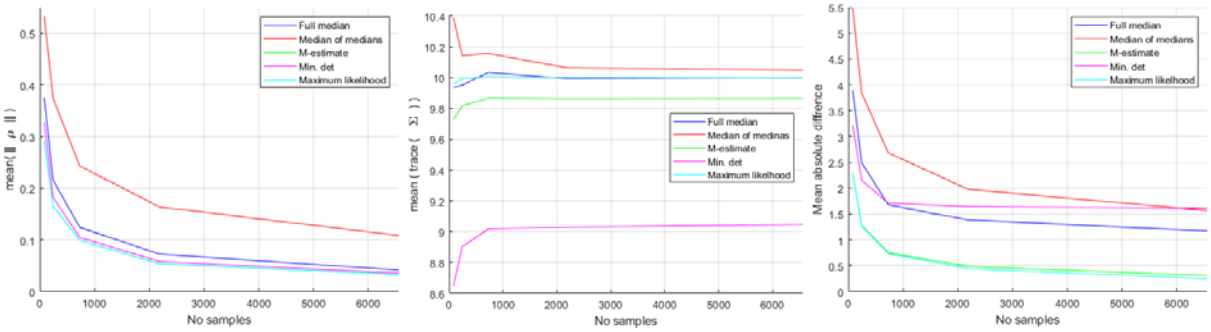

Figure 1 Results from estimation of mean vector and covariance matrix using different estimators. Samples drawn from a Gaussian distribution are applied. Evaluation criteria are norm of the mean vector (to the left), trace of the covariance matrix (in the middle), and the mean absolute difference between estimated covariance matrix and the "true" covariance matrix. See text for details. 1000 replications are used.

For the norm of the mean vector estimates, we see that it is a little higher for the median-of-median based estimates than for all other ones, which have practically the same behavior. This does not surprise as long as the median of medians is an approximation to the "full" median. Moreover, based on our experiments in [7], this was also expected. There, we found that the expectation of the median-of medians based estimate was good, but that its standard deviation was somewhat higher than the estimate based on the "full" median, which in turn was a little higher than for classical maximum likelihood estimator. For the trace of the covariance matrix, all estimators but MCD give similar results. It's interesting to register the small difference between the median-based estimators and the maximum likelihood. The MCD seems to underestimate the trace somewhat. Concerning the mean absolute difference, the M-estimate shows the same performance as the maximum likelihood estimator. As expected the median based estimators have a little higher mean absolute difference. Surprisingly the MCD doesn't perform better than the median based estimators. We haven't, however, tried to study the MCD in depth in order to try to find why this happened.

In the second evaluation we have tested the estimators in a situation where 5% of the samples are contaminated with "salt" samples. Salt samples are defined as samples with values [-3.16, 15.5], i.e. 5σ sigma along each principal direction. The results are shown in Figure 2.

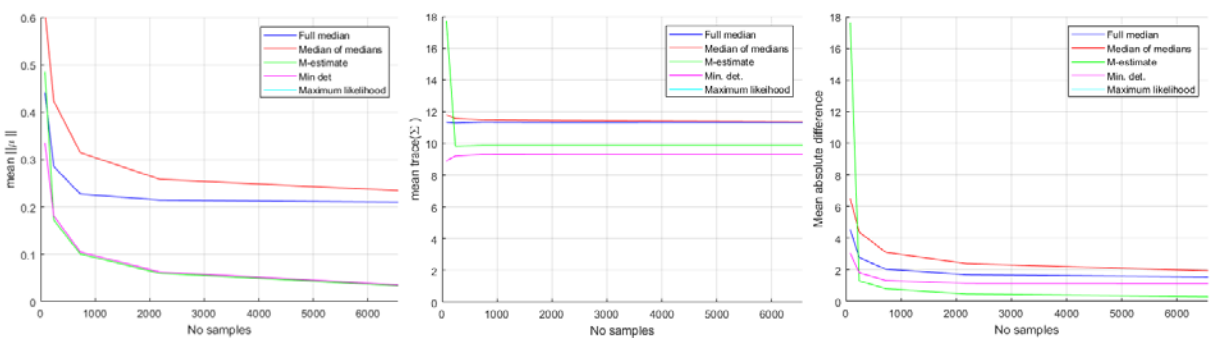

Figure 2 Results from estimation of mean vector and covariance matrix using different estimators. Samples drawn from a contaminated Gaussian distribution are applied. The contamination consists of 5% "salt" samples. Evaluation criteria are norm of the mean vector (to the left), trace of the covariance matrix (in the middle), and the mean absolute difference between estimated covariance matrix and the "true" covariance matrix. See text for details. 1000 replications are used.

In this example, the classical (maximum likelihood) estimator has broken down. The values of the figure of merits were too high to be included in the plots (they were around 2.3, 112 and 200 for the plots (in reading order)). Moreover, the first thing to notice is that using median based estimates norm of the mean vector is higher than in the previous experiment. This is due to a non-symmetrical contamination. In this particular case with 5% contamination, the median is in fact the 0.525-quantile. For the same reason, the median-based methods' estimates of the trace are also a little too high. The M-estimate and MCF both perform very well.

Finally, in the third evaluation we investigate the breakdownpoint, i.e. how large can the contamination be before an estimator breaks down. Samples from the Gaussian distribution defined above are contaminated with "salt" samples as defined above. The figure of merit is computed for various degree of salt contamination, from 0% to 70%. N=729 samples are used. The results are shown in Figure 3.

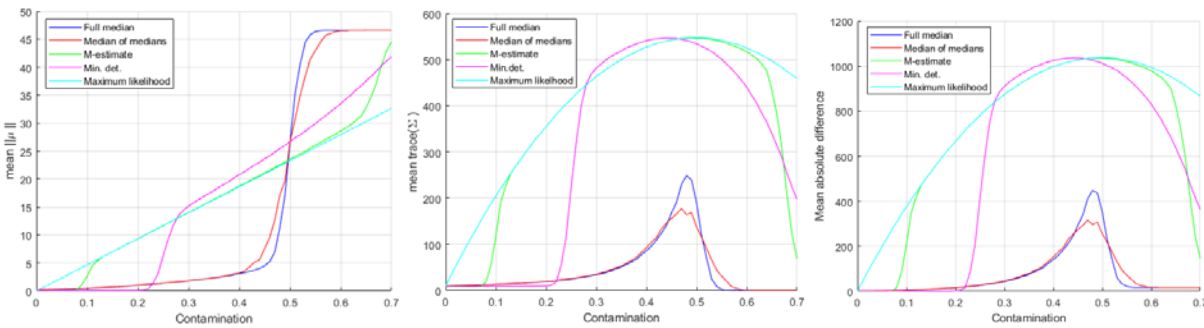

Figure 3 The performance for the different estimators as a function of contamination. The contamination consists of "salt" samples, and varies from 0% to 70%. 729 samples are used in the simulations. Evaluation criteria are norm of the mean vector (to the left), trace of the covariance matrix (in the middle), and the mean absolute difference between estimated covariance matrix and the "true" covariance matrix. See text for details. 1000 replications are used.

The M-estimate and MCD perform very well until the break down. However, surprisingly, the M-estimate breaks down even for a relatively small amount of contamination. For a contamination of slightly less than 10% or more, the estimator seems to be virtually worthless. The same is the situation for MCD if the contamination is a little less than 25% or more. The two median-based estimators have almost the same behavior. They have an almost linear decrease in performance until they break down.

We haven't done any comparison of the methods with respect to processing time. One of the reasons is that we haven't made any efficient implementation in a high level language of the M-estimator and MCD. This is because we know they are both quite complex, e.g. the M-estimator is iterative involving a lot of matrix computation in each iteration, and thus not feasible for real time computation. The other reason is that the processing time of the median and median of medians is compared in [7].

3. EXPERIMENTS

We will in this section present three experiments utilizing the median of medians approximation. They are all about discrimination. Our focus is not primarily on how well can the median estimate parameters in an underlying distribution, but how well will a resulting classifier discriminate between groups. That's what matters for us.

The first experiment is a classification example. It consists of two classes with four dimensional contaminated Gaussian distributed samples. The training samples are contaminated with samples from an outlier distribution. The second experiment is classification of acoustic signals. The third experiment is an image segmentation experiment of two-plane images (long and medium IR).

3.1 Classification

We have designed a two-class four-dimensional experiment. The samples of both classes are drawn from Gaussian distributions with some heavy contamination.

The non-contaminated part of the samples from the first class, ω_1, have zero mean, and covariance matrix

$$\mathbf{\Sigma}_1 = \text{diag}(9, 1, 1, 1)$$

The non-contaminated part of the samples from the second class, ω_2, have mean vector and covariance matrix

$$\boldsymbol{\mu}_2 = [t, 0, 0, 0]$$

$$\mathbf{\Sigma}_2 = \text{diag}(1, 1, 1, 9)$$

We have used $2 \le t \le 6$ in this experiment.

We will present experiments with two different contaminations. In the first one we let *p*% of the samples be drawn from a Gaussian distribution with 100 times larger covariance matrix (i.e. 10 times larger standard deviation for each dimension). In the second one, the contamination consists of *p*% "salt" samples, defined as $\mathbf{x} = [3.35, 1.12, 1.12, 3.35]$; the value in each dimension is proportional with maximum deviation in that direction, and $\|\mathbf{x}\| = 5$.

Equal prior probability has been used.

Monte Carlo simulation is used for the evaluation. For a given set of design-set samples, 1000 test samples (without any contamination) are drawn and applied. We have used 1000 replications.

In addition to median based estimates ("full" median and median of medians), we have also applied M-estimates [10], the minimum covariance determinant estimator (MCD) [11], and the classical (maximum likelihood) estimator. All these classifiers are compared to the optimal one; the classifier where the correct parameters are applied.

Figure 4 shows the error rate and standard deviation of the error rate estimate for *N* = 729 samples in the case of no contamination.

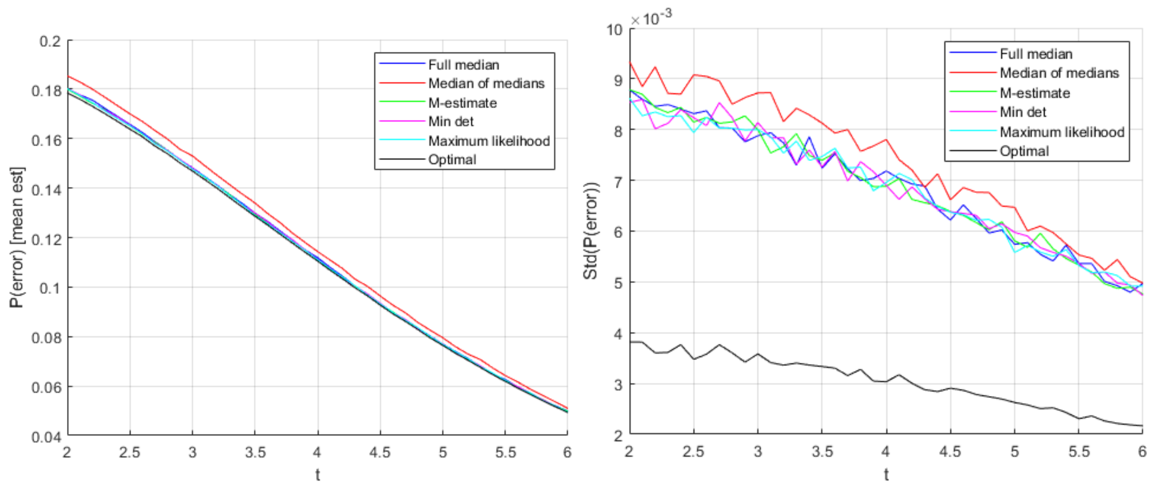

Figure 4 Error rate (left) and standard deviation of the error rate (right) for various classifiers. 729 samples from Gaussian distributions as defined in the text are used. The estimates are based on 1000 replications.

Four of the classifiers show a more or less identical performance, and they are very close to the optimal one. It is interesting to notice that the (full) median results are almost the same as the maximum likelihood classifier, which in this example is the best one because the sample distributions and the classifier assumptions are the same. The median of medians had slightly poorer performance. This was expected since it is an approximation of the median.

Figure 5 shows a situation of contaminated samples. 95% of the samples are drawn from the same distribution as used in the previous example. The 5% rest are drawn from a distribution with 10 times higher standard deviation. Thus we have a symmetrical contamination distribution.

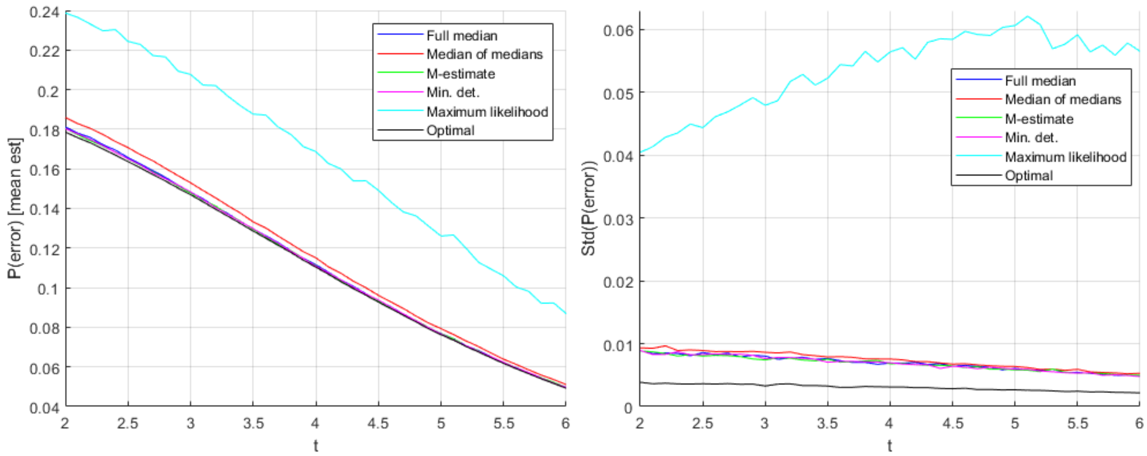

Figure 5 Error rate (left) and standard deviation of the error rate (right) for various classifiers. 729 samples are drawn in each replication; 95% from Gaussian distributions as defined in the text, and 5% with 10 times larger deviation. The estimates are based on 1000 replications.

Obviously, the classifier based on maximum likelihood estimates performs poorly. This was expected as long as the estimates don't handle outliers. The difference between the four others is small, and compared with the optimal classifier, they all handle the outliers. The classifier based on median of medians estimates performs as expected slightly poorer than the other ones.

Figure 6 shows another situation with contaminated samples. In this example, 15% of the samples are "salt" samples, as defined previously.

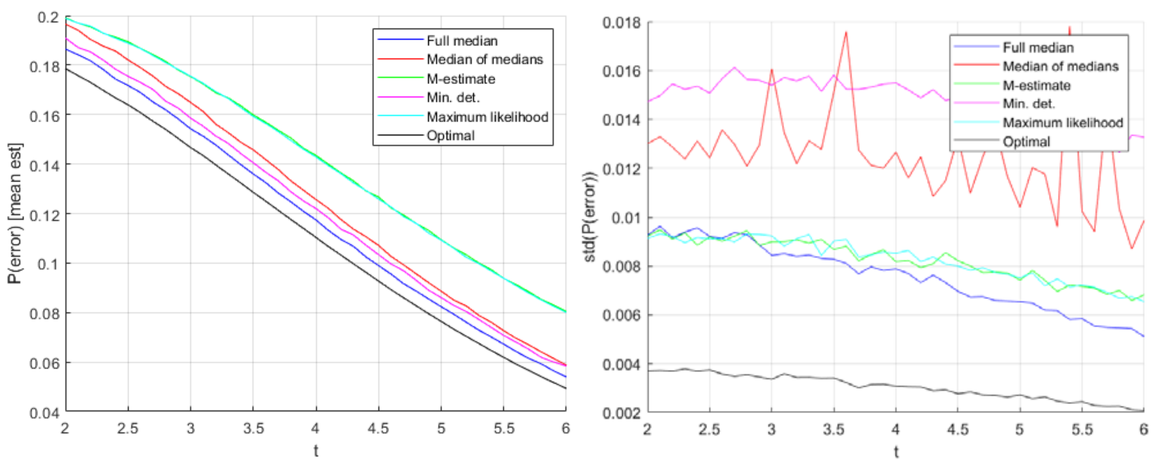

Figure 6 Error rate (left) and standard deviation of the error rate (right) for various classifiers. 729 samples are drawn in each replication; 85% from Gaussian distributions as defined in the text, and 15% are "salt" samples. The estimates are based on 1000 replications.

The first thing to notice is that the M-estimator does not handle the "salt" contamination. It performs just as poor as the classifier based on maximum likelihood estimates. This doesn't surprise, it corresponds with our finding in Figure 3. Based on this figure, we would have expected that the classifier based on MCD had had a better performance. Even though it handles the contamination, it is more affected than the classifier based on "full" median estimates. The classifier based on median of medians estimates does also handle the contamination, and it performs only slightly worse than the MCD.

3.2 Classification of acoustic signals

This experiment consists of samples derived from acoustic signals, which was kindly provided by Idar Dyrdal, FFI. The signals are recordings of either a car engine or a drone. 270 samples from car engines are available, and 426 samples from a drone. Five features were derived. All 31 combinations were evaluated, and a combination of two features was found to be the best. Figure 7 shows a scatterplot of the samples in the dataset.

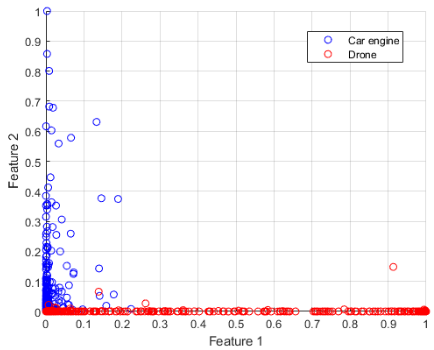

Figure 7 Scatterplot of the samples in the car engine and drone experiment.

In addition to the classifiers used in the previous experiment, we have included the nearest neighbor rule (NNR) [13] and a linear classifier based on the mean squared error (MSE) [13]. The classifiers are evaluated using cross validation [13]. The results are shown in Figure 8.

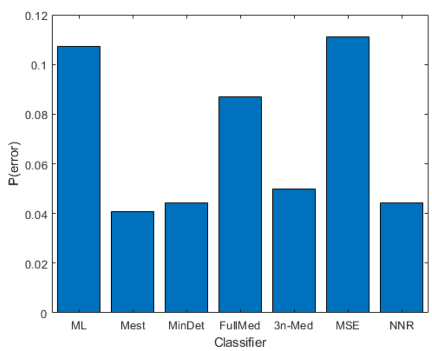

Figure 8 Classification results of car engines and drones. The five first classifiers (in reading order) are all based on Gaussian assumptions and thus quadratic classifiers. The first one (ML) uses the maximum likelihood estimators for determining the parameters, the next one uses M-estimates, then comes the results when the minimum covariance determinant estimator (MCD) is used, "FullMed" uses the median in the estimation, "3n-Med" uses median of medians. MSE is the linear classifier based on mean squared error, and NNR is the nearest neighbor rule.

We see from the scatterplot that the samples are far from Gaussian. Therefore it does not surprise that the classifier assuming Gaussian samples (ML) has a little high error rate. Moreover, we observe that the two classifiers designed for handling outliers are significantly better. They are just as well performing as the non-parametric classifier NNR. This shows that they are able to generate well performing quadratic classifiers also in situation where the features are far from Gaussian. The classifier based on median of medians does also work well. Comparing with the "full"-median based version, it works well, and it surprises that the "full"-median based classifier has a poorer performance than the classifier based on median of medians. The linear classifier based on mean squared error has the poorest performance. This is most likely due to that the sample distribution is not suitable for the mean squared error optimization criteria.

3.3 Image segmentation by anomaly detection

In this experiment we have used two IR cameras (longwave and mediumwave) in our test system. The cameras may be run at a frequency up to 100 Hz. So far, we have used 50 Hz or less. In general each frame is either a 16 bit unsigned integer or a float. Figure 9 shows an image pair of a sequence.

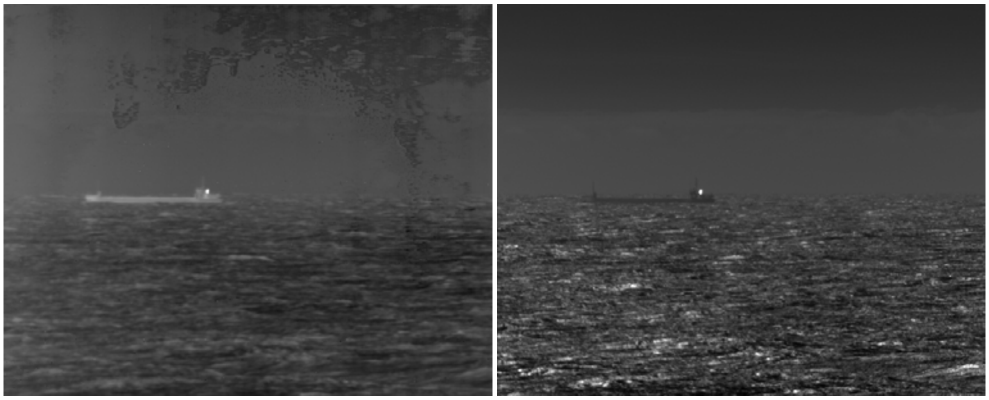

Figure 9 Image pair from an image sequence consisting of a cargo ship. The image to the left is a longwave IR image. The image to the right is a mediumwave IR image.

The idea for this segmentation algorithm is to model the background intensities, and then extract anomalies, i.e. pixels with sufficiently low probability density. The algorithm is a simplified version of a strategy for generating a classifier based on assumption of mixture of Gaussian distributions [14], i.e. the probability density is given as

$$\mathrm{p}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i N_d\left(\boldsymbol{\mu}_\mathbf{i}, \boldsymbol{\Sigma}_\mathbf{i}\right)(\mathbf{x}) \tag{8}$$

where n denotes the number of Gaussian mixtures ("hats") with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. α_i are normalizing constants;

$$\sum_{i=1}^{n} \alpha_i = 1$$

The algorithm initially picks a predefined number of randomly chosen samples (i.e. pixels). Based on these samples, the principle of the algorithm is to cluster the intensities of these samples into a predefined number of groups. We assume that the intensities in each group can be modeled with a Gaussian distribution, and thus estimate the parameters (mean, standard deviation, normalizing constant) for each group. Finally, pixels with sufficiently small probability densities are identified.

Given the samples, and the number of groups, the segmentation procedure consists of the following steps:

1. Estimate the mean vector and covariance matrix for the whole dataset by using the median based estimators. Compute the eigenvector corresponding to the largest eigenvalue and define a hyperplane perpendicular to this vector and passing through the median vector. Divide the dataset into two groups using this hyperplane.
2. Estimate the covariance matrix in each group by using the median based estimators.
3. Identify the group with the largest trace of its covariance matrix. Estimate the mean of the samples in this group (the covariance matrix is estimated already), and divide the samples in two subgroups according to the description in step 1.
4. If the number of groups is less than the predefined number: Goto step 2.

5. For each: group: Estimate the parameters $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and α_i . The median and MAD are used as estimator for the mean and standard deviation respectively.
6. Segment the image, i.e. identify the pixels *i,j* where $p\left(\left[b_{i,j}^{LW}, b_{i,j}^{MW}\right]\right) < \tau$. $\tau > 0$ is an anomaly threshold chosen in advance, and $b_{i,j}^{LW}$ and $b_{i,j}^{MW}$ are the intensities in pixel *i,j* in the longwave and mediumwave band respectively.

The median of medians is used in the estimation. The number of samples in a group is in general obviously not a power of three. In order to obtain this, we have extended the number of samples in the group as described in section 2.3.

It should be noticed that this algorithm does not give any optimal estimates of $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$, and α_i with respect to maximum likelihood. If that is desirable, an optimization step must be added at the end. In addition, the strategy of extending the number of samples in a group to a power of three should be avoided because it implies that the samples no longer can be assumed to be statistical independent. However, we are not primarily interested in the background distribution; we are merely interested in well performing segmentation. Our experience is that the above procedure gives parameters which reflect the background distribution reasonably well and hence gives good segments.

Figure 10 shows the segmentation of a vessel in thermal IR images using the algorithm described here. A visual inspection shows that the background intensities are grouped around a small number of modes. Therefore we will expect that it is sufficient with small number of Gaussian components in the segmentation. We have used n = 3, and 9 in this experiment. The segmentation is based on the intensities in N = 6561 randomly drawn pixels.

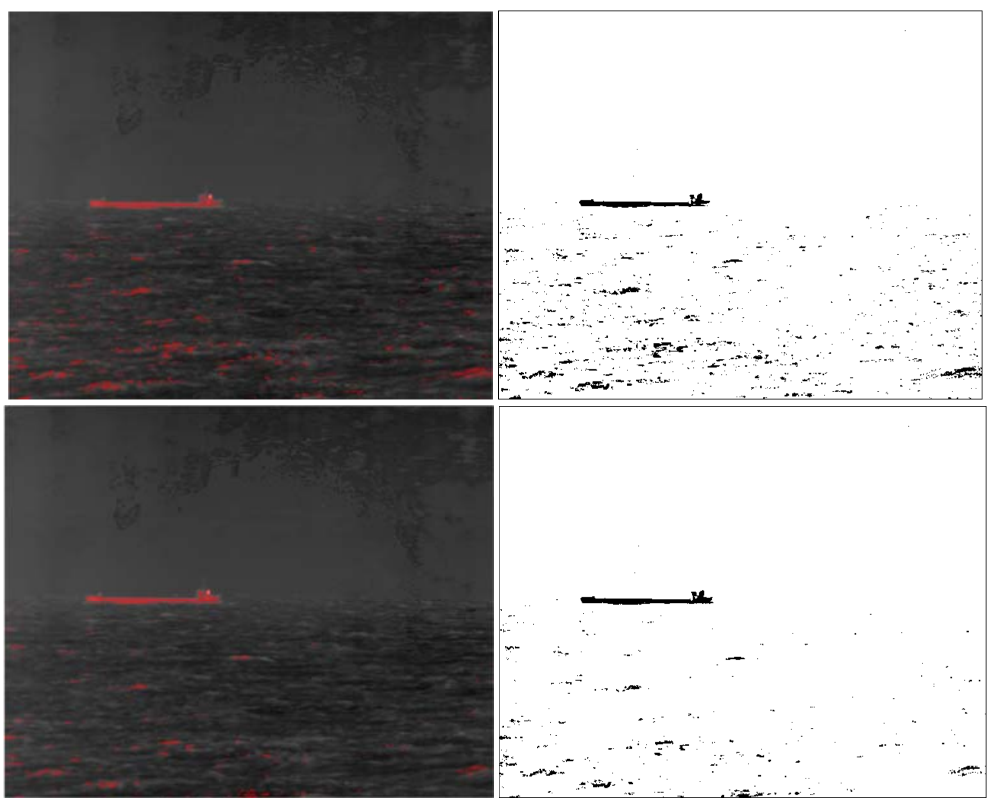

Figure 10 Segmentation with the assumption of background intensities being mixture of Gaussian distributions. Three Gaussian mixtures are used in the upper image, and nine in the lower image. The segmentation is based on N = 6561 samples.

We notice that the segmentation result of the boat in the two images in Figure 10 is more or less the same. There is more clutter using three mixtures than nine mixtures, but the clutter size is mostly small, and can easily be detected and handled by some straightforward postprocessing techniques. This indicates that the number of Gaussian mixtures is not very critical.

Finally, we have added heavy contamination to the image. The intensities in 20% of the pixels are contaminated with "salt" intensities. The contaminated images are shown in Figure 11. These images are used as input to the segmentation. Seven Gaussian mixtures are used. The results are shown in Figure 12.

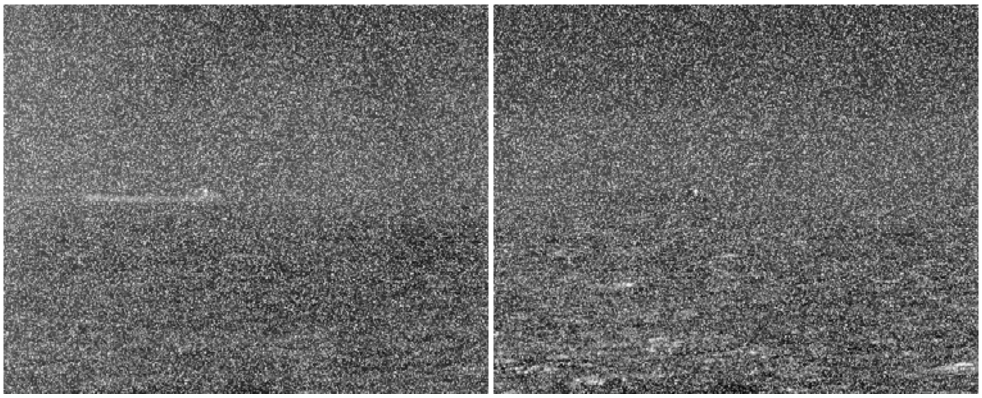

Figure 11 The image pair from Figure 9 contaminated with 20% "salt" pixels.

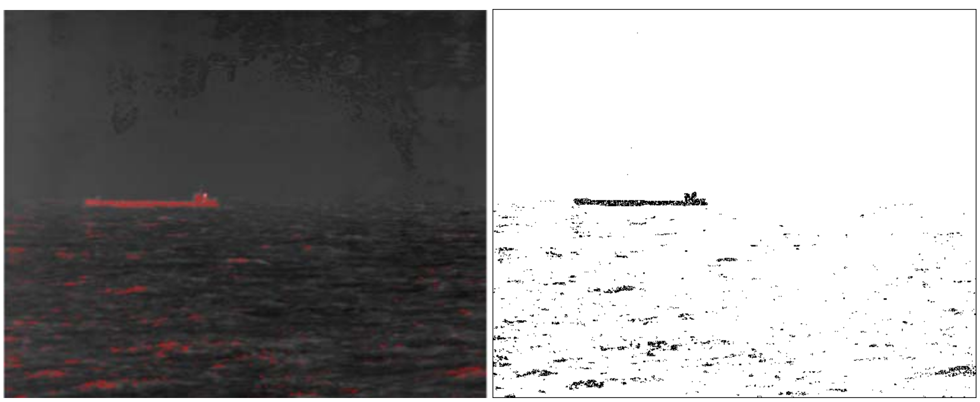

Figure 12 Segmentation with the assumption of background intensities being mixture of Gaussian distributions. The input image is contaminated with 20% "salt" noise. No preprocessing nor postprocessing are applied. Five Gaussian mixtures are used.

The first thing to notice is that the classifier has adapted well to the "salt" pixels. Hence, these are not extracted. Moreover, the results are more or less similar to the results shown in Figure 10. Except for contaminated areas, the ship is extracted well. This demonstrates the algorithm's ability to adapt to the background statistics.

4. SUMMARY AND CONCLUSION

We have in this paper studied whether the median of medians is applicable for estimation of parameters in multivariate data. Since it is based on many median computations of small pieces of a dataset instead of using all samples simultaneously, we were not surprised that the median of medians based estimates show a little lower performance than median based estimates. However, whether we apply the median of medians or "pure" median based estimates, doesn't have much influence on the classification or segmentation results. The main drawback with the median of medians is that number of samples has to be a power of 3. In general, this is not fulfilled. In order to obtain a dataset where the number of elements is a power of three, we have drawn randomly a number of the input samples and included them for the estimation. This is obviously disputable from a statistical point of view. However, from a practical point of view, we haven't seen any negative impact. In addition, it must be emphasized that the median of medians is much faster than the median, which is of importance in real-time applications. Compared with robust estimation techniques like M-estimator or the MCD, median based estimation has been demonstrated to perform very well. Although not demonstrated here, it should nevertheless be mentioned that these estimators are substantially slower than median based techniques.

We have applied the median of medians for classifying multivariate data. Here, we have presented results from Monte Carlo simulations, classification of acoustic signals, and segmentation. So far we have found that estimators based on median of medians produce fast, robust, and reliable results.

REFERENCES

[1] Breivik, G. M., Løkken, K. H., Brattli, A., Palm, H. C., Haavardsholm, T. V., "An ATR architecture for algorithm development and testing," Proc. SPIE 8744 (2013).

[2] Conover, W. J., "Practical nonparametric statistics," John Wiley and Sons, New York (1980).

[3] Pratt, W. K., Cooper, T. J., Kabir, I., "Pseudomedian filter," Proc. SPIE 534, 34-43 (1985).

[4] Yang, W., Zhang, Z., Lu, X., Li, J., Chen, D., Yang, G., "A novel fast median filter algorithm without sorting," Proc. SPIE 9897 (2016).

[5] Wu, C., C., Chen, D. Y., Zhao, C., Y., "Impulsive noise filter using median- and partition-based operation," Optical Engineering, 47(11) (2008).

[6] Blum, M., Floyd, R., W., Pratt, V., R., Rivest, R., L., Tarjan, R., E., "Time bounds for selection," Journal of Computer and System Sciences, 7(4) (1973).

[7] Palm, H. C., "Fast and robust image processing algorithms based on approximations to order statistics," Proc SPIE 10648 (2018).

[8] Wikipedia, "Median of absolute deviation," https://en.wikipedia.org/wiki/Median_absolute_deviation (8 March 2018).

[9] Pasman, V.R., Shevlyakov, G. L., "Robust methods for estimation of a correlation coefficient", Automation and Remote Control, ???, 332-340.

[10] Cambell, N. A., "Robust Procedures in Multivariate Analysis I: Robust Covariance Estimation", *Applied Statistics*, vol. 29, no. 3, pp. 231-237, 1980.

[11] Rousseeuw, P. J, Van Driessen K., "A fast algorithm for the minimum covariance determinant estimator", Technometrics, 3, 212-223.

[12] LIBRA, "Library for robust analysis," https://wis.kuleuven..be/statdatascience/robust (Feb 2020).

[13] Ripley, B. D., "Pattern recognition and neural networks," Cambridge University Press, Cambridge, (1996).

[14] Palm, H. C., "A new method for generating statistical classifiers assuming linear mixtures of Gaussian densities," Proceeding of 12th International Conference on Pattern Recognition, B, 483-486 (1994).