AN ADAPTIVE THRESHOLD METHOD FOR HYPERSPECTRAL TARGET DETECTION

Joshua Broadwater, Rama Chellappa

University of Maryland
Center for Automation Research
College Park, MD USA
jbroadwa@glue.umd.edu, rama@cfar.umd.edu

ABSTRACT

In this paper, we present a new approach to automatically determine a detector threshold. This research problem is especially important in hyperspectral target detection as targets are typically very similar to the background. While a number of methods exist to determine the threshold, these methods require either large amounts of data or make simplifying assumptions about the background distribution. We use a method called inverse blind importance sampling which requires few samples and makes no a-priori assumptions about the background statistics. Results show the promise of this algorithm to determine thresholds for fixed false alarm densities in hyperspectral detectors.

1. INTRODUCTION

In hyperspectral (HSI) detection, a threshold has to be determined to separate the targets from the background for a specified probability of false alarm α_0. A number of methods have been developed to identify this threshold.

For a class of detectors called Constant False Alarm Rate (CFAR) detectors, the threshold can be calculated theoretically from the closed-form solution of their detection statistic. The problem with this method is that the detectors are usually based on an assumed underlying probability distribution (e.g. the normal distribution). Unfortunately, HSI data is known to be rarely normally distributed [1] and researchers have shown that theoretically derived detection statistics do not match real-world HSI data [2].

This realization has led to the use of elliptically-contoured distributions to model the output detection statistic. The detector data is used to estimate parameters which in turn provide a distribution from which a detection threshold can be theoretically calculated. The usefulness of this method is still being investigated, but its applications have been limited to CFAR detectors developed using the normality assumption [3].

A standard non-parametric way of determining the desired detector threshold is to use a rank-ordered test. The detector output is sorted in descending order to create an ordered list. The length of the output N is then multiplied by the desired α_0 and rounded to the nearest integer. This integer is used to identify the position in the ordered list that will be used as the detection threshold. The strength of this approach is that any detector output can be used – not just those that are CFAR. Unfortunately, this solution requires that the product $N\alpha_0$ be greater than one. If not, the index will be zero and no threshold can be selected. This can be problematic when extremely low false alarm rates are required.

This paper introduces a new method of determining the detector threshold based on importance sampling (IS). Importance sampling is a forced Monte Carlo method that is used to simulate rare events [4]. A branch of IS research called inverse importance sampling (IIS) was developed that can calculate a detection threshold given a α_0 and probability distribution. In 1998, Bucklew published an algorithm called blind importance sampling (BIS) that removed the need to know a-priori knowledge of the underlying distribution [5]. This allows BIS algorithms to measure the performance of very complex systems and provide thresholds for a fixed α_0 even when the underlying distribution changes.

For this paper, we apply the inverse blind importance sampling (IBIS) technique to the problem of detection threshold selection for HSI detectors. Section 2 presents an overview of IBIS techniques. Section 3 demonstrates the IBIS algorithm's abilities in a series of experiments. Section 4 summarizes our results and presents future research directions.

2. ALGORITHM DERIVATION

Blind importance sampling uses experimental data to estimate rare events. In the context of detectors, this can be used to estimate a desired tail distribution or in the inverse method, a threshold designed to provide a fixed α_0. The following sub-sections detail the derivation of this technique following that found in [4].

2.1. Importance Sampling

Consider there exists a set of K data points X_i taken from some known distribution f. At some threshold t, the number of samples greater than this threshold can be modeled by a Binomial distribution

$$P(k_t = k) = \binom{K}{k} p_t^k \left(1 - p_t\right)^{K-k} \tag{1}$$

where k represents the number of samples above t with a probability of success p_t. Using (1), a maximum likelihood estimate of p_t can be found such that

$$\hat{p}_t = \frac{1}{K}\sum_{i=1}^{K} 1(X_i \geq t) \tag{2}$$

where $1(x)$ is an indicator function. This function can also be seen as the estimated tail probability or probability of rare event if t is sufficiently large. For a Monte Carlo simulation, this requires a very long simulation to collect enough samples to properly estimate p_t.

IS reduces the length of the Monte Carlo simulation by biasing the estimate with a weighting function $W(x)$ (also called an importance function) such that

$$\hat{p}_t = \frac{1}{K}\sum_{i=1}^{K} 1(X_i \geq t) W(X_i), \quad X_i \sim f_* \tag{3}$$

where f_* is a biasing density and

$$W(x) \equiv \frac{f(x)}{f_*(x)}. \tag{4}$$

Most of IS is concerned with the proper design of $W(x)$ such that the estimate of p_t can be done with fewer points and improved precision.

2.1. Blind Importance Sampling

For BIS, a few changes are necessary as we do not know the underlying distribution $f(x)$. Now we assume there exists a set of K experimental data samples X_i taken from some unknown distribution f. Using an acceptance-rejection method proposed by Bucklew [5], we create a new set of samples Y_i such that

$$\{Y_j\}_1^{K_r} = \{X_i \mid U_i \leq h(X_i)\}_1^K \tag{5}$$

where U_i are random samples drawn from a uniform distribution and $h(X)$ is a function bounded between 0 and 1. Using this construct the Y_j are distributed according to

$$h_*(x) = \frac{1}{a_h} h(x) f(x) \tag{6}$$

where a_h is the new probability of success such that

$$P(K_r = j) = \binom{K}{j} a_h^j (1 - a_h)^{K-j}. \tag{7}$$

The blind importance function becomes

$$W(x) \equiv \frac{a_h}{h(x)}. \tag{8}$$

Combining (3), (7), and (8), we obtain the blind tail probability estimate

$$\hat{\alpha}_t = \frac{1}{K_r}\sum_{i=1}^{K_r} 1(X_i - t \geq 0) \frac{a_h}{h(X_i)}, \quad X_i \sim h_*. \tag{9}$$

In partially blind importance sampling, the probability of success a_h is known a-priori. In our blind case, we must estimate this value. Knowing that

$$a_h = P(U_i \leq h(X_i)) = E\{h(X_i)\}, \tag{10}$$

we can estimate the mean of $h(X)$ such that

$$a_h = \frac{1}{K}\sum_{i=1}^{K} h(X_i). \tag{11}$$

Note that equations (9) through (11) depend on the selection of a proper $h(X)$ called the h-function. An h-function suggested by Srinivasan [4] is

$$h(x) = e^{s(x-c)} 1(x \leq c) + 1(x > c) \tag{12}$$

where c is a constant called the truncation parameter that guarantees $h(x) \leq 1$. Additionally, the truncation parameter should be greater than the detection threshold t to minimize the variance of the estimated p_t. While Srinivasan has an approach to find a good choice for c, it depends on knowing the detection threshold t. Clearly for a problem where we are interested in obtaining the t from the data, this remains an active area for research.

The other parameter s in the h-function does have an automated way to find the optimal value which minimizes the variance of our tail estimate. From [4], the variance of the tail estimate is bounded above by the estimate

$$\hat{I}_b = \hat{a}_h e^{-2st} \left(\frac{1}{K}\sum_{i=1}^{K} \frac{e^{2sx_i}}{h(x_i)} \right). \tag{13}$$

Note that (13) can be estimated entirely from the original data samples X_i. To find the optimal s, an iterative approach using a gradient descent method can be used such that

$$s_{m+1} = s_m - \delta \frac{\hat{I}_b'}{\hat{I}_b''} \tag{14}$$

where m is the iteration, δ is a value controlling the rate of descent, $\hat{I}_b'$ is the first derivative of the variance with respect to s, and $\hat{I}_b''$ is the second derivative of the variance with respect to s. The calculation of these derivatives is fairly straight-forward but rather involved and is left out of this paper for space considerations.

2.2. Inverse Blind Importance Sampling

The earlier constructs provided a way to calculate the estimated tail probabilities given a threshold. We are interested in finding a threshold for a given probability. This is called inverse blind importance sampling. An iterative approach using another gradient descent method is used to identify the appropriate threshold t that minimizes the error between the desired tail probability and estimated tail probability. The iterative equation is

$$t_{m+1} = t_m + \delta \frac{\alpha_0 - \hat{\alpha}_t}{\hat{\alpha}_t'} \tag{15}$$

where m is the iteration, δ is a value controlling the rate of descent, α_0 is the desired tail probability, $\hat{\alpha}_t$ is the current estimated tail probability, and $\hat{\alpha}_t'$ is derivative of the current estimated tail probability with respect to t.

To solve (15), we need to take the derivative of (9) with respect to t. The identity function makes this difficult; so, we replace the identity function with

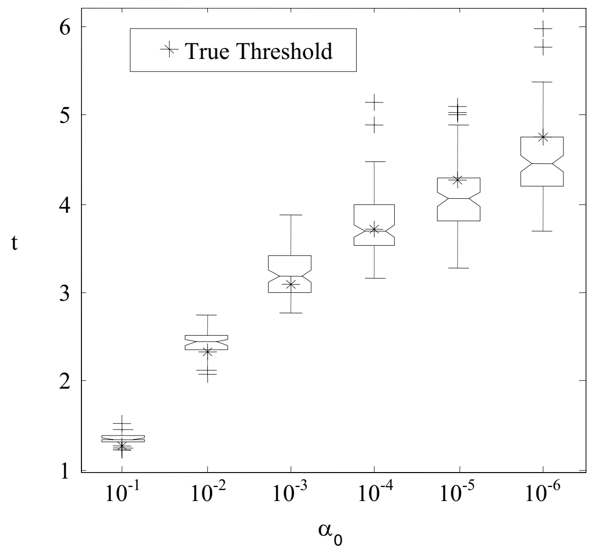

Fig. 1. Comparison of Ideal Threshold with Estimated IBIS Thresholds

$$1(\varepsilon) \approx \frac{1}{1+e^{a\varepsilon}} \qquad (16)$$

where $\varepsilon = X_i - t$, and a is a parameter used to control how close the function approximates a true identify function. The larger the value of a, the better the approximation is to the identity function.

3. EXPERIMENTAL RESULTS

To test the ability of the IBIS algorithm to estimate a detection threshold, we constructed three experiments. The first experiment shows the ability of our BIS algorithm to accurately estimate a detection threshold – even when using very few samples. The second experiment demonstrates the algorithm's ability to automatically adjust a detection threshold to maintain a specified probability of false alarm across changing background distributions. The final experiment applies the algorithm to real-world hyperspectral data which has been processed using a standard anomaly detection algorithm.

3.1. Experiment 1

This experiment demonstrates the ability of IBIS algorithm to accurately estimate a detection threshold for a specified probability of false alarm α_0. We chose six α_0 that ranged from 10^{-1} to 10^{-6}. For each α_0, we processed 100 tests comprised of a thousand iid samples drawn from a standard normal distribution. Since we used a standard normal distribution, we could calculate the ideal threshold for the specified α_0. This ideal threshold was then compared to the IBIS results obtained using a statistical boxplot. The results are shown in Figure 1. Note that we are requiring the algorithm to estimate some α_0 values that are below the number of samples provided.

The IBIS algorithm performs well without any a-priori distribution information. The boxplot shows that the thresholds obtained match the ideal threshold with some degree of accuracy. With larger α_0, the algorithm tends to over-estimate the values while with smaller α_0, the algorithm tends to under-estimate the threshold values. Also, the variance of the detection threshold estimates grows with decreasing α_0.

All of these effects are expected. Without any information of the underlying distribution, the algorithm does track the thresholds to within one standard deviation of the ideal threshold. The variances increase with decreasing α_0 because less and less samples are available to determine the threshold. What is incredible is the ability of the algorithm to estimate fairly accurately detection thresholds for α_0 below 10^{-3} when only one thousand samples were used for each iteration.

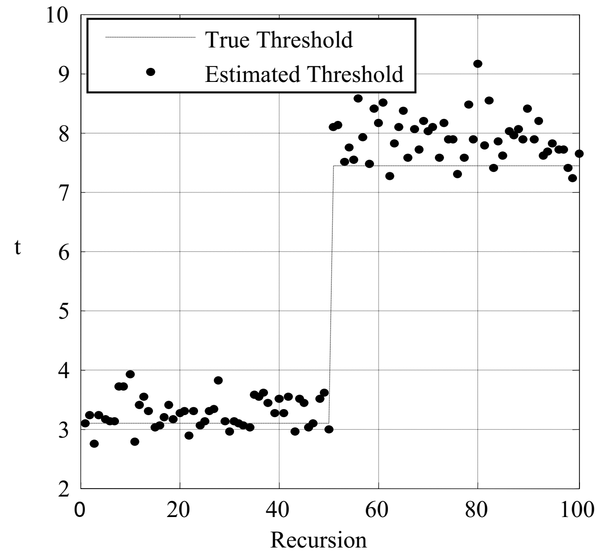

Fig. 2. IBIS Threshold Estimates for Experiment 2

3.1. Experiment 2

To showcase the IBIS algorithm's ability to adjust the detection threshold automatically to maintain a specified α_0, this experiment changes the underlying statistical distribution without informing the algorithm of the change. Initially, one thousands iid samples are drawn from a standard normal distribution. This experiment is repeated fifty times and then the distribution is changed to a Rayleigh distribution and fifty more iterations are used. Throughout this test, the IBIS algorithm is asked to maintain a α_0 of 10^{-3}.

The results of this experiment are shown in Figures 2 and 3. In Figure 2, the IBIS threshold estimates are compared to the known threshold. In Figure 3, the IBIS estimated false alarm rates are compared to the desired false alarm rate of 10^{-3}.

As can be seen in these figures, the IBIS algorithm performs well. It tracks the true detection threshold even when the underlying distribution is changed. This is expected since the algorithm does not use any a-priori distribution information to estimate a threshold. If it did, the algorithm would fail when the distribution changes. It maintains a α_0 that is close to 10^{-3} considering that it is doing this given only 1000 samples per test.

When comparing the IBIS algorithm to the non-parametric ordered-rank test, the IBIS algorithm provides a 3-fold increase in precision over the rank-ordered test. The variance of the estimate

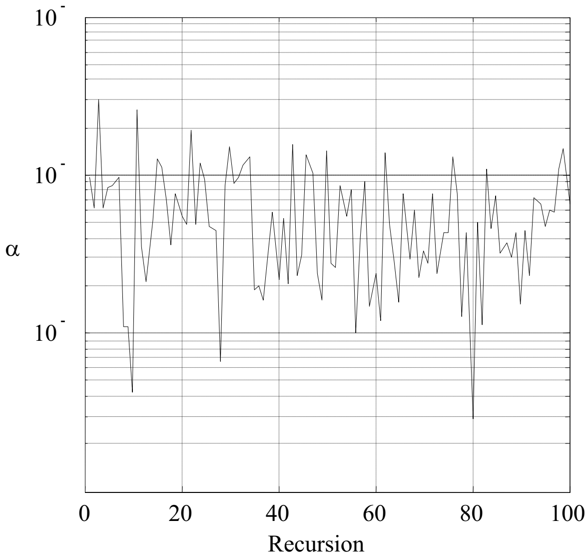

Fig. 3. Estimated α for each iteration in Experiment 2

obtained using the rank-ordered algorithm is $1.06 \cdot 10^{-6}$. The IBIS algorithm variance is $3.61 \cdot 10^{-7}$.

3.2. Experiment 3

The final experiment applies the IBIS algorithm to an anomaly detector operating on real-world HSI data. The HSI data is from the HYDICE sensor (a VIS/NIR/SWIR sensor with 210 spectral bands). The image contains trucks and tarps in open fields and near tree-lines. These targets are identified using the RX algorithm – a standard anomaly detection algorithm for HSI data [6]. The RX output is then the input to our threshold estimation algorithms.

We used three methods to determine the α_0 of 10^{-3} from our data. The first method was a theoretical calculation based on the fact that RX output is a chi-squared distribution with L degrees of freedom. The second method was the rank-ordered method described in the introduction. The third method was the IBIS algorithm. All of these methods were compared in Table 1 to the ideal threshold determined empirically from the data.

Table 1: Comparison of Experiment 3 Results

Estimator	Threshold	Pd	α
Theoretical	279.07	0.15	0.00039
Rank-ordered	269.92	0.16	0.00042
IBIS	176.65	0.23	0.00103
Ideal	180.59	0.22	0.00100

The results show that the IBIS algorithm achieves the closest performance to the ideal false alarm rate of 10^{-3}. The other two methods are considerably lower. The theoretical calculation based on the chi-squared test most likely under estimates due to the fact that the detector output is not truly chi-squared. The rank-ordered method most likely underestimates because it includes the target estimates in its calculations. The IBIS algorithm using the specified h-function with limit c is able to suppress the influence of the target samples and hence provide a better estimate. However, the constant c was derived empirically from the data since no automated method currently exists to find this quantity. Nevertheless, this experiment demonstrates the promise of the IBIS algorithm to automatically determine a threshold given a specified false alarm rate.

4. CONCLUSIONS

A new adaptive threshold method has been demonstrated for use in HSI detection applications. The method is based on inverse blind importance sampling. This method requires no a-priori knowledge of the underlying distribution making it applicable to all detector types – especially those that are too complex to theoretically calculate the detection statistic distribution. Additionally, the IBIS algorithm has shown the ability to automatically adapt to changes in the background distribution.

This work is the beginning of our research. The key component of the IBIS algorithm is the selection of the h-function. The h-function used in this analysis was the standard choice and better performance could be obtained through the use of different functions. Additionally, Srinivasan demonstrated that detectors could be included in the IS architecture. This knowledge of detector architecture could lead to better IBIS estimates. Our current work is focused on incorporating HSI detectors into the IS architecture and selecting more appropriate h-functions.

5. ACKNOWLEDGMENTS

The authors would like to thank Dr. Amit Banerjee of The Johns Hopkins University Applied Physics Laboratory for the HYDICE imagery and ground truth used in our experiments. This work was funded through ARO MURI Grant G41Z93G4 via GTRI.

6. REFERENCES

[1] D.W.J. Stein, S.G. Beaven, L.E. Hoff, E.M. Winter, A.P. Schaum, and A.D. Stocker, "Anomaly detection from hyperspectral imagery," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 58-69, January 2002.

[2] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 29-43, January 2002.

[3] D. Manolakis, "Realistic matched filter performance prediction for hyperspectral target detection," in *Proceedings of the International Geoscience and Remote Sensing Symposium*, IEEE, 2004, vol. 2, pp. 953-955.

[4] R. Srinivasan, *Importance Sampling, Applications in Communications and Detection*, Springer-Verlag, Berlin, Germany, 2002.

[5] J.A. Bucklew, "The blind simulation problem and regenerative processes," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2877-2891, November 1998.

[6] I.S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1760-1770, October 1990.