Ellipsoids for Anomaly Detection in Remote Sensing Imagery

Guen Grosklos and James Theiler

Los Alamos National Laboratory, Los Alamos, NM 87545

ABSTRACT

For many target and anomaly detection algorithms, a key step is the estimation of a centroid (relatively easy) and a covariance matrix (somewhat harder) that characterize the background clutter. For a background that can be modeled as a multivariate Gaussian, the centroid and covariance lead to an explicit probability density function that can be used in likelihood ratio tests for optimal detection statistics. But ellipsoidal contours can characterize a much larger class of multivariate density function, and the ellipsoids that characterize the outer periphery of the distribution are most appropriate for detection in the low false alarm rate regime. Traditionally the sample mean and sample covariance are used to estimate ellipsoid location and shape, but these quantities are confounded both by large lever-arm outliers and non-Gaussian distributions within the ellipsoid of interest.

This paper compares a variety of centroid and covariance estimation schemes with the aim of characterizing the periphery of the background distribution. In particular, we will consider a robust variant of the Khachiyan algorithm for minimum-volume enclosing ellipsoid. The performance of these different approaches is evaluated on multispectral and hyperspectral remote sensing imagery using coverage plots of ellipsoid volume versus false alarm rate.

Keywords: Anomaly Detection, Multispectral Imagery, Hyperspectral Imagery, Background Estimation, Low False Alarm Rate

1. INTRODUCTION

Characterization of the background is a key consideration in the detection of targets and anomalies in multispectral and hyperspectral imagery.[1] The Gaussian distribution is often the first choice for that characterization, and although conceptually simple, it can be surprisingly effective.[2]

For background data that can be modeled with a d-dimensional multivariate Gaussian, the distribution is specified by the mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $C \in \mathbb{R}^{d \times d}$, and the natural choice for anomaly detection is the Mahalanobis distance:[3]

$$\mathcal{A}(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) \tag{1}$$

This is sometimes called the Global RX detector, as it is a special cases of a local approach to anomaly detection developed by Reed, Yu, and Stocker.[4, 5]

One way in which this choice of anomaly detector is optimal is that it has the smallest volume for a given false alarm rate. The contours of the Gaussian are ellipsoidal, and in particular the ellipsoid specified by $(\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) = r$ encloses a volume given by

$$V = \frac{\pi^{d/2}}{\Gamma(1 + d/2)} |C|^{1/2} r^{d/2}. \tag{2}$$

A simple generalization of the Gaussian is given by elliptically-contoured (EC) distributions, which preserves many of its properties, but provides a more realistic model for the tail of the distribution. The effectiveness of EC distributions for hyperspectral data has been demonstrated in a variety of situations.[6–8] Two very important properties of the Gaussian that are inherited by more general EC distributions, are: that Eq. (1) is an optimal choice of anomaly detector, and Eq. (2) expresses the volume enclosed by contours.

One can certainly consider more general distributions (*e.g.,* multimodal Gaussians or kernel-based models), but the focus here is on how to estimate from data the ellipsoids that characterize elliptically-contoured distributions, and on how that affects anomaly detection performance. In particular, we will consider two aspects of the estimation problem: one is robustness against overfitting and the other is the importance of estimating the distribution on its periphery,[9] where one finds the boundary between normal and anomalous. To this end, we will compare a robust scheme (MCD) and a periphery-characterizing scheme (MVEE), and will develop a new algorithm (MVEE-h) that enables us to add "a little bit of robustness" to MVEE. Evaluating these schemes on several multispectral and hyperspectral datasets suggests that (at least for the global ellipsoidal characterization that we consider here), characterizing the periphery is more important than achieving robustness.

Although we do not consider more general target detection problems here, we remark that the boundary that separates target from clutter is also on the periphery of the clutter distribution.

2. SAMPLE COVARIANCE MATRIX

Define $\mathbf{x}_n = [x_n(1), x_n(2), \ldots, x_n(d)]^T$ as input spectral signals consisting of d spectral bands and let X be a $d \times N$ matrix of the N background pixels. Each observed spectral pixel is represented as a column in the sample matrix X

$$X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \tag{3}$$

From these data pixels, we define the sample mean $\boldsymbol{\mu} = (1/N)\sum_{i=1}^{N} \mathbf{x}_i$, and the mean-subtracted sample covariance matrix $C = (1/N)\sum_{i \in N} (\mathbf{x}_i - \boldsymbol{\mu})^T(\mathbf{x}_i - \boldsymbol{\mu})$.

Although the sample covariance matrix is the best estimator (in the sense of maximum likelihood) for the covariance, *per se*, we observe that we really need to estimate the inverse covariance in order to compute Mahalanobis distance with Eq. (1). The best estimator of the inverse covariance matrix is not necessarily the inverse of the best estimator of the covariance; in particular, an over-fit estimate leads to under-estimates of the smallest eigenvalues of the covariance, and matrix inversion magnifies these small values. To reduce the effects of over-fitting, a wide variety of regularization schemes have been proposed.[10–23] For the discussion here, however, we will presume that we have enough data ($N \gg d$) to avoid overfitting, and instead worry about the effects of non-Gaussian distributions, heavy tails, and outliers.

3. MINIMUM COVARIANCE DETERMINANT (MCD)

Campbell[24] makes the case for "robust" covariance estimation, arguing that traditional statistics (such as the sample covariance matrix) give too much leverage to points far from the central core of the data. Such points might be outliers (not truly belonging to the distribution whose covariance is being estimated), but even if they *are* sampled from the distribution of interest, their distance from the centroid gives them undue influence in estimating that distribution. Baesner[25] also argues that removing outliers in the estimation of background model leads to a better anomaly detector.

Rousseeuw[26–28] developed the minimum covariance determinant (MCD) algorithm as a specific robust covariance estimator that seeks the sample covariance for a subset of points (h out of N) that constitute the central core data. Since there are, *a priori*, a combinatorically large number of such subsets (N choose h), the algorithm employs a heuristic strategy involving multiple restarts to identify good choices for the core data points, and is based on the premise that small volumes are best, although it does not seek to optimize volume directly.

As the name suggests, what MCD attempts to optimize is the determinant of the sample covariance matrix. Strictly speaking, this minimization is an NP-hard problem, but the MCD algorithm does a good job of finding a good (even if not the ultimately optimal) solution.

The pseudocode in Algorithm 1 describes MCD in more detail, though the "fast MCD" described in Ref. [28] employs some further speed-ups that are not described here. A key component of the MCD algorithm is the CSTEP, which iterates from one subset of h points to another subset. A theorem is proved that a CSTEP iteration can never increase the determinant; a sequence of CSTEP iterations inevitably leads to a minimum, albeit a local minimum.

Consider a dataset $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of d-variate observations. Let $H \subset \{1, \ldots, N\}$ with $h < N$ elements. Put $\boldsymbol{\mu} = (1/h)\sum_{i \in H} \mathbf{x}_i$ and $C = (1/h)\sum_{i \in H} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$. If $|C| \neq 0$, define the relative distances

$$r_i := (\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \qquad \text{for } i = 1, \ldots, N \tag{4}$$

Now take H' to be the set of indices of the h smallest distances; thus $\{r_i \ : \ i \in H'\} = \{r_{\pi[1]}, \ldots, r_{\pi[h]}\}$, where π indicates the permutation of indices that sorts the distances: $r_{\pi[1]} \leq r_{\pi[2]} \leq \cdots \leq r_{\pi[N]}$. Re-compute $\boldsymbol{\mu}'$ and C' based on the points in H' instead of H. Then the theorem asserts that the determinant of the new covariance will not increase; that is:

$$|C'| \leq |C|, \tag{5}$$

with equality if and only if $\boldsymbol{\mu}' = \boldsymbol{\mu}$ and $C' = C$. The theorem assumes $|C| \neq 0$. If $|C| = 0$, the minimum volume will have already been achieved, and there is no need for further iterations.[28]

Consecutive iterations of CSTEP produce a nonnegative, monotonically decreasing sequence of determinants. Because there are finitely many h-subsets, this sequence of determinants must converge in a bounded number of steps. The stopping criteria for CSTEP occurs when the resulting determinant reaches zero or is equal to the previous iteration's determinant. This does not guarantee a global minimum, so the CSTEP iteration is performed multiple times, each time using a different randomly drawn subset as an initial condition. To more effectively explore the diversity of possible solutions, the initial subset is very small, with just enough points (typically $d+1$ points) to span the d dimensional space, so that the initial determinant is nonzero. CSTEP is applied to each subset until convergence, and only the solution with the lowest determinant is kept.

Algorithm 1 Minimum Covariance Determinant (MCD)

```
Require: Data set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with $\mathbf{x}_i \in \mathbb{R}^d$
Require: $T$,$h$                                   ▷ $T$ is number of trials; $h \leq N$ is size of subset
for $t = 1, \ldots, T$ trials do
    $J \leftarrow$ random draw of $d+1$ points from $X$
    Compute $\boldsymbol{\mu} \leftarrow \text{ave}(J)$ and $C \leftarrow \text{cov}(J)$
    while $|C| = 0$ do                               ▷ in case points in $J$ don't span space
        $J \leftarrow J \cup \{$random draw of one sample from $X\}$   ▷ add new points until they do
        Recompute $\boldsymbol{\mu} \leftarrow \text{ave}(J)$ and $C \leftarrow \text{cov}(J)$
    end while
    repeat
        $\boldsymbol{\mu}, C \leftarrow \text{CSTEP}(X, \boldsymbol{\mu}, C, h)$   ▷ $|C|$ gets smaller with each iteration
    until convergence                               ▷ Convergence when $|C|$ reaches zero or no longer decreases
    Set $\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}$ and $C_t \leftarrow C$   ▷ Save $\boldsymbol{\mu}, C$ for each trial
end for
$t_* = \text{argmin}_t \ |C_t|$                    ▷ Identify trial whose covariance matrix $C_t$ has minimum determinant
Output: $\boldsymbol{\mu}_{t_*}$, $C_{t_*}$

procedure CSTEP$(X, \boldsymbol{\mu}, C, h)$
    for $i = 1, ..., N$ do
        Compute distance $r_i \leftarrow (\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_i - \boldsymbol{\mu})$   ▷ Mahalanobis distance to centroid
    end for
    $\pi \leftarrow \text{argsort}(r_1, \ldots, r_N)$   ▷ Ensures $r_{\pi[1]} \leq \cdots \leq r_{\pi[N]}$
    Create set: $H \leftarrow \{\mathbf{x}_{\pi[1]}, ..., \mathbf{x}_{\pi[h]}\}$   ▷ $H$ is "core" subset of $X$ that excludes outlying points
    Compute $\boldsymbol{\mu}' \leftarrow \text{ave}(H)$ and $C' \leftarrow \text{cov}(H)$
    return $\boldsymbol{\mu}', C'$                   ▷ A theorem ensures $|C'| \leq |C|$
end procedure
```

4. MINIMUM VOLUME ENCLOSING ELLIPSOID (MVEE)

The MCD algorithm finds a subset of points such that the volume of the ellipsoid defined by the sample covariance of those points is minimized, whereas the MVEE algorithm more directly seeks the ellipsoid of minimum volume that encloses the points.

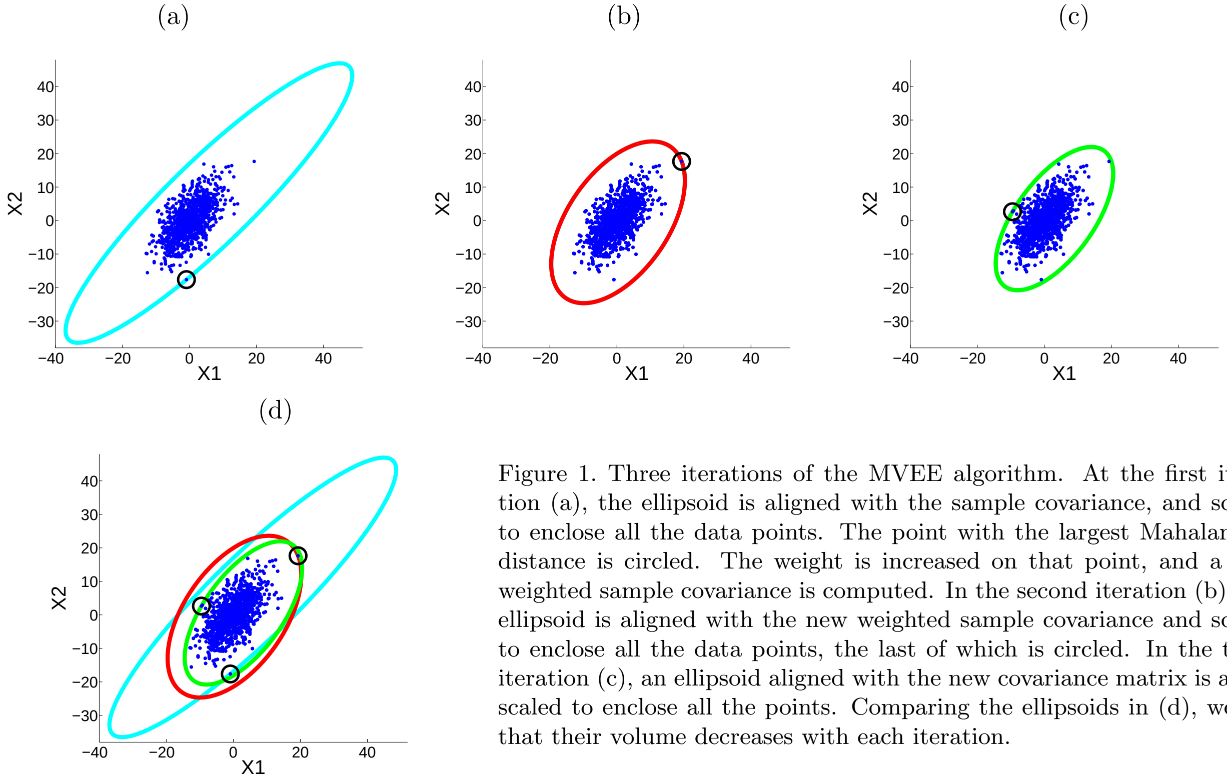


Figure 1. Three iterations of the MVEE algorithm. At the first iteration (a), the ellipsoid is aligned with the sample covariance, and scaled to enclose all the data points. The point with the largest Mahalanobis distance is circled. The weight is increased on that point, and a new weighted sample covariance is computed. In the second iteration (b), the ellipsoid is aligned with the new weighted sample covariance and scaled to enclose all the data points, the last of which is circled. In the third iteration (c), an ellipsoid aligned with the new covariance matrix is again scaled to enclose all the points. Comparing the ellipsoids in (d), we see that their volume decreases with each iteration.

Given a set of points: $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, with $\mathbf{x}_i \in \mathbb{R}^d$, we seek a centroid $\boldsymbol{\mu}$, and a symmetric positive-definite matrix C such that all points satisfy $(\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \leq 1$. We observe that the set $E = \{\mathbf{x} \; : \; (\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq 1\}$ is a d-dimensional ellipsoid whose volume is given by Eq. (2) with $r = 1$.

Thus, to minimize the volume of E, we need to minimize the determinant of the covariance.

$$\min_{\mu, C} |C| \text{ subject to constraints:} (\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \leq 1 \text{ for all } i \tag{6}$$

This problem, the minimum volume enclosing ellipsoid (MVEE), is a convex optimization problem [29, p. 401], and one of the classic solutions is given by Khachiyan's algorithm.[30]

4.1 Khachiyan Algorithm

In Khachiyan's algorithm,[30] weights are assigned to all the points in the data, and $\boldsymbol{\mu}$ and C are given by the weighted sample mean and covariance. The covariance produced in Khachiyan's algorithm is scaled by d and the weights are re-adjusted until $(\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \leq 1$ for all i. This is illustrated in Figure 1, and described with pseudocode in Algorithm 2.

In Appendix A, we derive the fact that the average Mahalanobis distance is d. Since the average r_i is d, it follows that the largest r_i (which we denote r_j) is at least that large: that is, $r_j = \max(r_i) \geq \text{average}(r_i) = d$, with equality holding only if $r_i = d$ for all i for which $u_i > 0$. The Khachiyan Algorithm exploits this property by iteratively seeking the sample point with the largest Mahalanobis distance and increasing the weight u_j on that sample. This increase is by an amount β, whose value is derived in Appendix B. The increase is just enough so that at the next iteration, $r_j = d$.

At each iteration, the weights are adjusted so that the largest Mahalanobis distance becomes equal to the (weighted) average Mahalanobis distance. This will generally cause some other Mahalanobis distances to exceed the average, and the largest one of those samples is targeted for the next iteration. After many iterations, the largest Mahalanobis distance will be only slightly larger than d. Furthermore, as MVEE approaches its limit, most of the u_i's will approach zero. The points with nonzero values are the "support vectors," or points on the periphery of the data or very close to the surface of the enclosing ellipsoid.

Although the basic Khachiyan Algorithm, as described in Algorithm 2, is adequate to the task of identifying the MVEE for hyperspectral data, a number of speed-ups have been suggested,[31–33] and some of these are included in the more detailed pseudocode shown in Algorithm 3. For instance, Algorithm 3 appends a row of ones to the original data; this is a technique that enables the computation of $\boldsymbol{\mu}$ and C with a single matrix operation.

5. ROBUST MINIMUM VOLUME ENCLOSING ELLIPSOID (MVEE-H)

We remark that the sensitivity of the sample covariance estimator to outliers that concerned Campbell[24] and Rousseeuw[26–28] is even more pronounced in the MVEE algorithm, which depends only on a small number, typically $O(d)$, of the most outlying points. In fact, the standard MVEE can be described as an "anti-robust" estimator.[9] In this section, we propose a variant of the MVEE algorithm which is more robust to outliers than MVEE.

Table 1. Four covariance estimators and how they are related

	All data samples	Subset of h samples
Sample covariance matrix	Mahalanobis/RX	MCD
Minimum volume matrix	MVEE	MVEE-h

When fitting hyperspectral imaging data, especially those with anomalous points, EC distributions based on centroid and covariance estimations of the whole data may not work so well. Lever-arm outliers and non-Gaussian distributions are poorly represented through ellipsoidal descriptions. Rather, if the estimations are calculated using a subset of the data that excludes a few of the most outlying points, the algorithm can better fit around the Gaussian-like center.

The robust minimum volume enclosing ellipsoid (MVEE-h), described in pseudocode in Algorithm 4, utilizes the same algorithm as MVEE while incorporating MCD's $\boldsymbol{\mu}$ and C subset method by excluding the $N-h$ most outlying points from the next computation of $\boldsymbol{\mu}$ and C. Much like MCD's relation to RX, MVEE-h aims to provide a more robust method than MVEE of finding a well-fitting model. A key difference between the relation of RX and MCD to MVEE and MVEE-h is the multiple trials T of small subsets J that MCD uses, whereas our implementation of MVEE-h employs a single trial, and begins with the full dataset. Table 1 shows the relation between the different covariance estimators. The updated $\boldsymbol{\mu}$ and C are applied to all of the data, and the algorithm once again reduces the data by $N-h$ points. This process continues until the tolerance is met.

Algorithm 2 Minimum Volume Enclosing Ellipsoid (MVEE)

```
Require: Data set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with $\mathbf{x}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N$
Initialize weights: $u_i \leftarrow \frac{1}{N}$ for all $i$    ▷ Begin with equal weights; note $\sum_i u_i = 1$
repeat
    $\boldsymbol{\mu} \leftarrow \sum_i u_i \mathbf{x}_i$    ▷ weighted sample mean
    $C \leftarrow \sum_i u_i (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$    ▷ weighted sample covariance
    $r_i \leftarrow (\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_i - \boldsymbol{\mu})$ for all $i$    ▷ Mahalanobis distances
    $j \leftarrow \mathrm{argmax}_i\, r_i$    ▷ Identify most outlying point (largest Mahalanobis distance)
    Compute $\beta \leftarrow (r_j - d)/((d+1) r_j)$    ▷ $\beta > 0$ implies $r_j > 0$ implies $(\mathbf{x}_j - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_j - \boldsymbol{\mu}) > d$
                                                        ▷ which implies that $\mathbf{x}_j$ is not enclosed by ellipsoid $(\boldsymbol{\mu}, C)$
    Update weights: $u_i \leftarrow (1-\beta) u_i + \beta \delta_{ij}$ for all $i$    ▷ All weights $u$ are reduced, except $u_j$ is increased
until convergence    ▷ Convergence when $\beta$ is small enough
$\boldsymbol{\mu} \leftarrow \sum_i u_i \mathbf{x}_i$
$C \leftarrow d \times \sum_i u_i (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$    ▷ Scale by $d$ so that $(\mathbf{x} - \boldsymbol{\mu})^T C^{-1} (\mathbf{x} - \boldsymbol{\mu}) \leq 1$
Output: $\boldsymbol{\mu}$, $C$
```

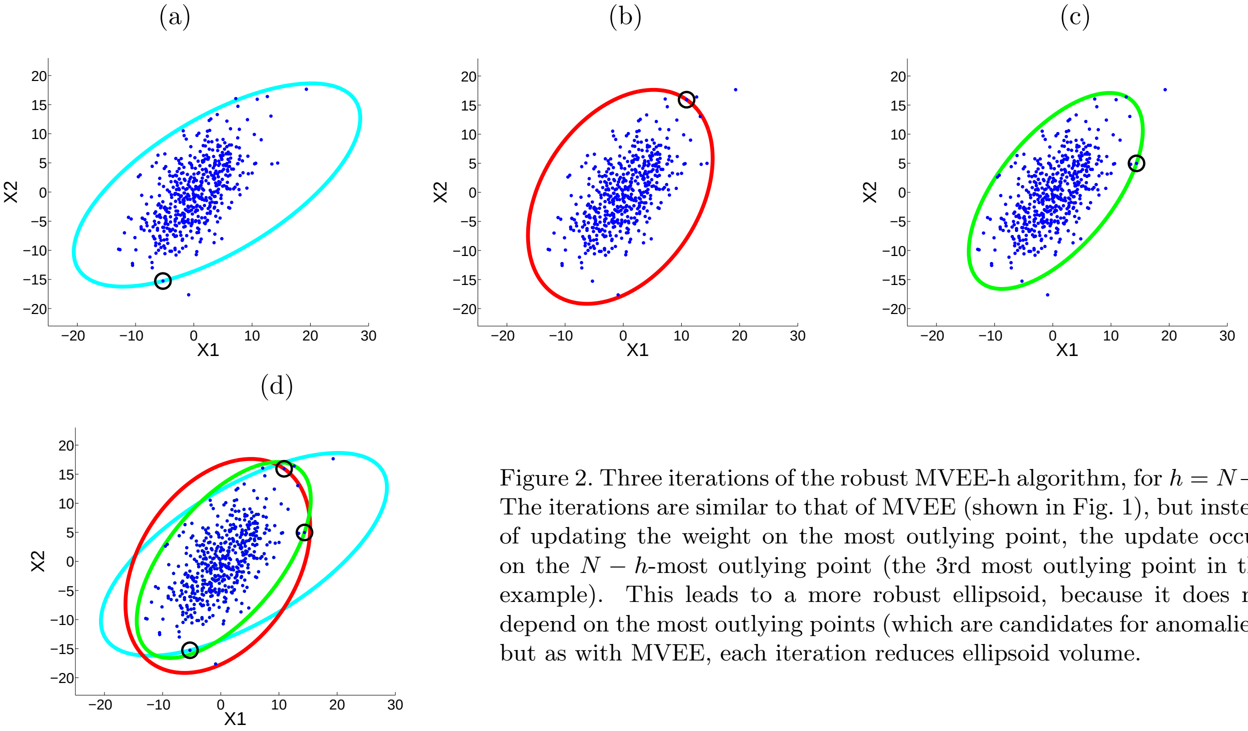


Figure 2. Three iterations of the robust MVEE-h algorithm, for $h = N-2$. The iterations are similar to that of MVEE (shown in Fig. 1), but instead of updating the weight on the most outlying point, the update occurs on the $N-h$-most outlying point (the 3rd most outlying point in this example). This leads to a more robust ellipsoid, because it does not depend on the most outlying points (which are candidates for anomalies), but as with MVEE, each iteration reduces ellipsoid volume.

Algorithm 3 Minimum Volume Enclosing Ellipsoid (MVEE) – vectorized implementation

```
Require: Data set $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ with $\mathbf{x}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N$    ▷ $X \in \mathbb{R}^{d\times N}$ is a $d \times N$ matrix
Initialize weights $\mathbf{u} \leftarrow \text{ones}(1, N)/N$    ▷ $\mathbf{u} \in \mathbb{R}^N$ is a vector
Append a row of ones to data, $X_a \leftarrow [X; \text{ones}(1, N)] \in \mathbb{R}^{(d+1)\times N}$    ▷ Simplifies linear algebra
                                                                  ▷ Allows $\boldsymbol{\mu}, C$ to be computed together
repeat
    $C_a \leftarrow X_a \,\text{diag}(\mathbf{u})\, X_a^T$    ▷ $C_a \in \mathbb{R}^{(d+1)\times(d+1)}$ combines covariance and mean
    $\mathbf{r}_+ = \text{diag}(X_a^T C_a^{-1} X_a)$    ▷ $\mathbf{r}_+ \in \mathbb{R}^N$ is a vector of Mahalanobis-like distances
                                                                  ▷ In fact, $\mathbf{r}_+ = \mathbf{r} + 1$, where $r$ is vector of actual Mahalanobis distances
    $j \leftarrow \text{argmax}(\mathbf{r}_+)$    ▷ Index of maximum distance
    $\kappa \leftarrow r_+(j)$    ▷ $\kappa - 1$ is maximum Mahalanobis distance
    Compute $\beta \leftarrow (\kappa - 1 - d)/((d+1)(\kappa - 1))$
    Update weights: $\mathbf{u} \leftarrow (1-\beta)\mathbf{u} + \beta\mathbf{e}_j$    ▷ Vector $\mathbf{e}_j$ has components $e_j(i) = \delta_{ij}$
until convergence
$\boldsymbol{\mu} \leftarrow X\,\mathbf{u}$    ▷ weighted sample mean
$C \leftarrow d \times (X\,\text{diag}(\mathbf{u})\,X^T - \boldsymbol{\mu}\boldsymbol{\mu}^T)$    ▷ weighted sample covariance, scaled by $d$
Output: $\boldsymbol{\mu}$, $C$
```

Algorithm 4 Robust Minimum Volume Enclosing Ellipsoid (MVEE-h)

```
Require: Data set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ with $\mathbf{x}_i \in \mathbb{R}^d$ for $i = 1, \ldots, N$
Require: $h$                                          ▷ $h \le N$ is size of subset
Initialize weights: $u_i \leftarrow \frac{1}{N}$ for all $i$      ▷ Begin with equal weights; note $\sum_i u_i = 1$
repeat
    $\boldsymbol{\mu} \leftarrow \sum_i u_i \mathbf{x}_i$                      ▷ weighted sample mean
    $C \leftarrow \sum_i u_i (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$
    $r_i \leftarrow (\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_i - \boldsymbol{\mu})$ for all $i$      ▷ Mahalanobis distances
    $\pi \leftarrow \operatorname{argsort}(r_1, \ldots, r_N)$                 ▷ Ensures $r_{\pi[1]} \le \cdots \le r_{\pi[N]}$
    $j = \pi[h]$                ▷ Most outlying point, except for the last few $N - h$ outliers
    Compute $\beta \leftarrow (r_j - d)/((d+1) r_j)$
    Update weights: $u_i \leftarrow (1 - \beta) u_i + \beta \delta_{ij}$ for all $i$
until convergence
$\boldsymbol{\mu} \leftarrow \sum_i u_i \mathbf{x}_i$                          ▷ weighted sample mean
$C \leftarrow d \times \sum_i u_i (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$     ▷ weighted sample covariance, scaled by d
Output: $\boldsymbol{\mu}$, $C$
```

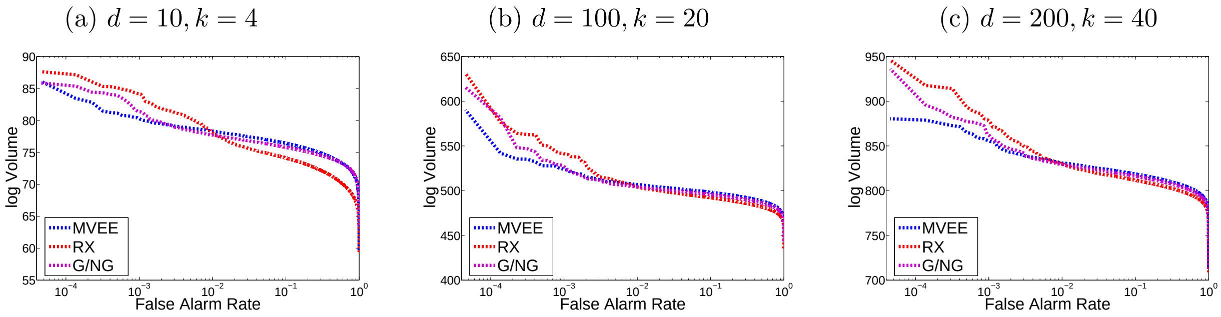
6. GAUSSIAN, NON-GAUSSIAN (G/NG)

While hyperspectral data is not typically Gaussian, there is a sense that it can be more Gaussian in “some directions,” particularly the directions with lower variance.[34–36] This suggests that the distribution can be modeled as a hybrid of Gaussian in the “more Gaussian” dimensions, and non-Gaussian in the (typically larger-variance) directions. Suggested non-Gaussian models in this context include a simplex-based distribution[37] or a histogram-based distribution.[38]

Piggybacking this idea, the Gaussian, non-Gaussian (G/NG) method utilizes the superior estimation abilities of MVEE in the high-variance directions with the efficiency of the RX algorithm in the low-variance directions, to create a hybrid method that is fast and fits the data well. In high dimensions, MVEE's iteration process can become unwieldy. Rather than performing MVEE on all of the data's dimensions, G/NG proposes to use MVEE on the k-dimensions with highest variance, and RX on the rest. This technique does as good a job as MVEE but with a significant decrease in run time (See Fig. 3).

7. NUMERICAL EXPERIMENTS

The efficacy of each algorithm is determined by relating the false alarm rate (FAR) with its corresponding ellipsoid volume. FAR is the fraction of data points outside of the defined ellipsoid and is considered a “false


(a) $d = 10, k = 4$

(b) $d = 100, k = 20$

(c) $d = 200, k = 40$

Figure 3. Coverage plots (volume vs. false alarm rate) for three different ellipsoid generation algorithms: MVEE, RX, and G/NG. The models are taken from the 200-channel “Indian Pines” data set. At all dimensions d, MVEE produces better models, however with higher dimensions and larger data sets, MVEE can become unwieldy. G/NG offers a solution by performing MVEE on the k most variant dimensions, and RX on the remainder. Doing so yields results competitive with MVEE, while reducing overall runtime.

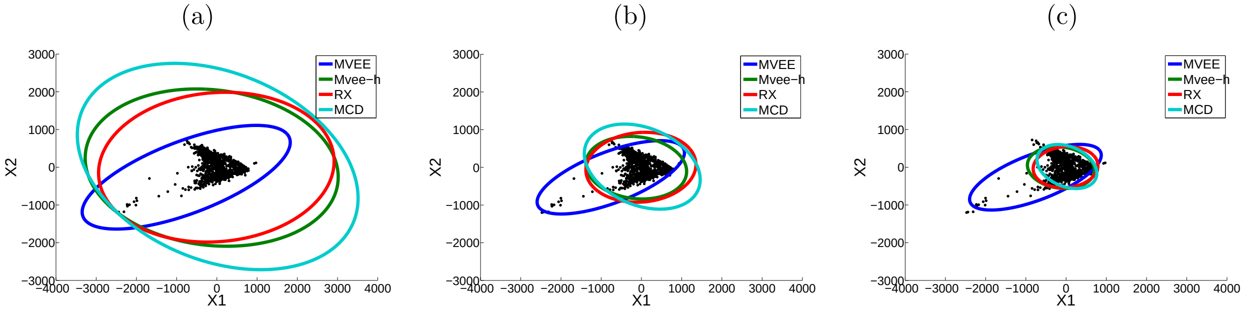


Figure 4. 2D representation, taken from the Florida data test set, of the different ellipsoids and how they behave at different FARs. These plots show that the choice of outliers is important for finding the smallest ellipsoid. (a) Ellipses enclose all of the data. MVEE does particularly well to fit the data as it does not have excessive white space. (b) One percent of the data is omitted. The designated h was at 99% of the data, so it is intuitive that MVEE-h seems to perform really well compared to the other algorithms. (c) At 5 percent data omission, MVEE maintains a fairly steady volume, while RX, MCD, and G/NG get noticeably smaller since they are centered around the main cluster. Recall that the goal is to find an estimator that fits the data well at low false alarm rates, so even though MCD, RX, and MVEE-h do well at 95% enclosure, MVEE outperforms the others at very low data omissions.

alarm" since data conformity is assumed and objects outside of the ellipsoid are treated as anomalous. An ellipsoid with smaller volumes at low FARs is the desired result.

To evaluate the different centroid and covariance estimation schemes, we used various multispectral and hyperspectral remote sensing imagery and performed the different algorithms on each set. The data ranged from 8 to 224 spectral channels. For each experiment, we randomly pulled 10,000 points for our training set and used that as our basis to create an ellipsoidal description. We then test the model on the remaining data, creating a series of multi-dimensional enclosing ellipsoids. Figure 4 illustrates the nature of each algorithm in a two-dimensional setting. Then in Figure 5, we have numerous FAR vs. log volume curves where we can actively see which methods do well in the low FAR region.

The best in-sample performance (lowest volume at low FAR) is observed to be the MVEE algorithm. This is expected; by definition, MVEE creates *the* minimum volume ellipsoid that covers all of the data. Somewhat surprisingly, however, MVEE also outperforms the other methods for the majority of the out-of-sample sets, suggesting that it mostly avoids overfitting. In general, these results indicate that overfitting is less of an issue than periphery characterization. In fact, it is only in Figure 5 (c) where MVEE-h performs better than MVEE in the out-of-sample set.

The fact that MCD does so poorly, often worse than RX, furthers the indication that robustness was not the main issue. Indeed, as also observed in Ref. [9], it is the "anti-robust" estimators that do a better job of characterizing the periphery. What the new MVEE-h estimator offers is a way of introducing "a little robustness" without giving up the emphasis on periphery.

APPENDIX A. AVERAGE MAHALANOBIS DISTANCE

In this section, we derive the result that for data in $\mathbb{R}^d$, the average Mahalanobis distance is d. We will derive this in the case of weighted data samples, but it applies to the unweighted case as well. Recall the definition of Mahalanobis distance:

$$r_i = (\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \tag{7}$$

For weighted data with weights u_i with $\sum_i u_i = 1$, we have the sample mean and sample covariance given by

$$\boldsymbol{\mu} = \sum_i u_i \mathbf{x}_i, \tag{8}$$

$$C = \sum_i u_i (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T. \tag{9}$$

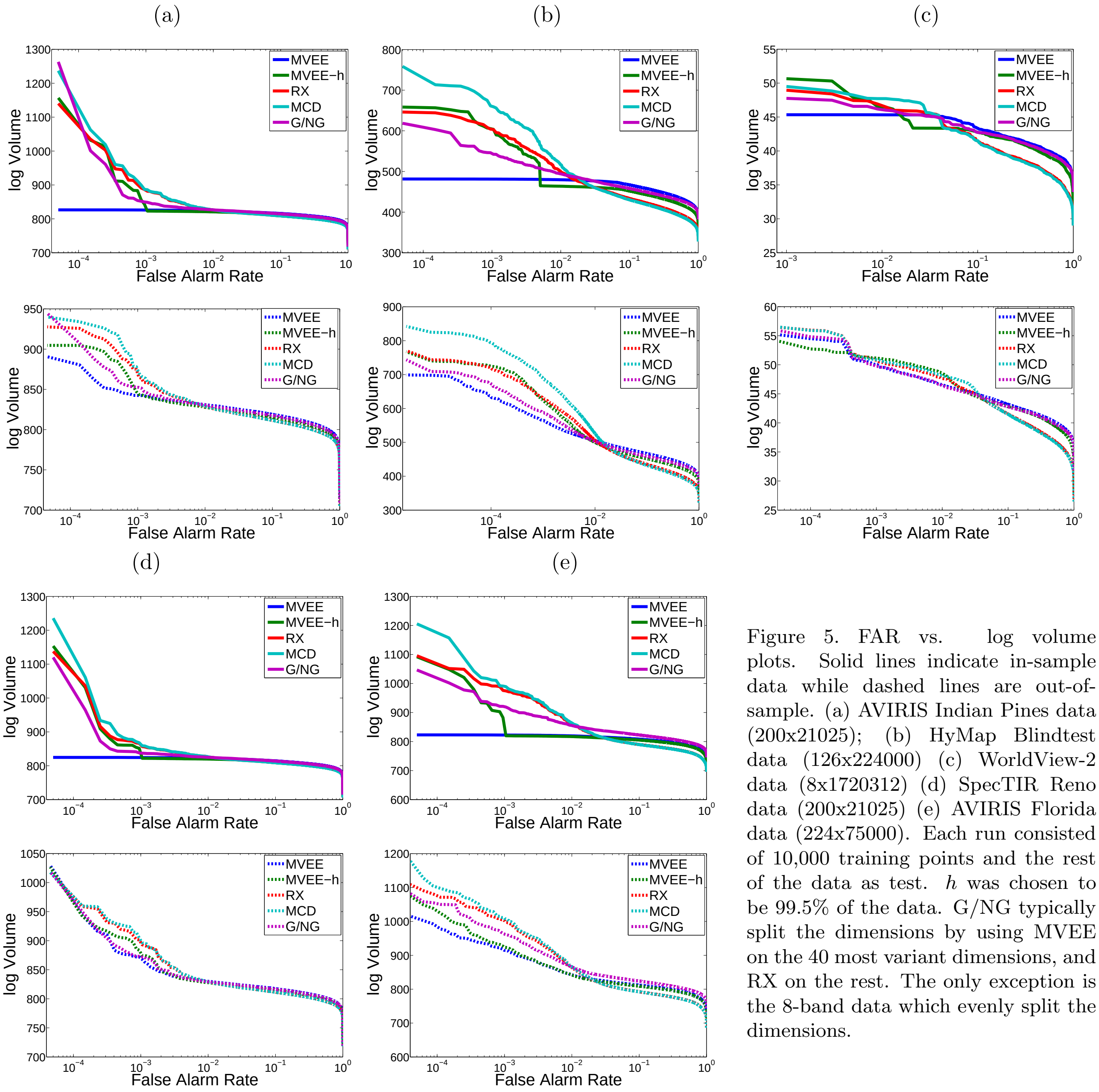


Figure 5. FAR vs. log volume plots. Solid lines indicate in-sample data while dashed lines are out-of-sample. (a) AVIRIS Indian Pines data (200x21025); (b) HyMap Blindtest data (126x224000) (c) WorldView-2 data (8x1720312) (d) SpecTIR Reno data (200x21025) (e) AVIRIS Florida data (224x75000). Each run consisted of 10,000 training points and the rest of the data as test. *h* was chosen to be 99.5% of the data. G/NG typically split the dimensions by using MVEE on the 40 most variant dimensions, and RX on the rest. The only exception is the 8-band data which evenly split the dimensions.

For unweighted data, $u_i = 1/N$ for all i.

The derivation makes use of the Trace operator; this is defined on square matrices as the sum of the diagonal elements. Thus, the Trace of a scalar is equal to that scalar, and the Trace of the $d \times d$ identity matrix is d. We note that the Trace is a linear operator, and we also use the property that $\mathrm{Trace}(AB) = \mathrm{Trace}(BA)$.

The average Mahalanobis distance will be given by

$$\sum_{i=1}^{N} u_i r_i = \sum_{i=1}^{N} u_i (\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \tag{10}$$

$$= \sum_{i=1}^{N} u_i \mathrm{Trace}\left((\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \right) \tag{11}$$

$$= \sum_{i=1}^{N} u_i \mathrm{Trace}\left((\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T C^{-1} \right) \tag{12}$$

$$= \mathrm{Trace}\left(\left[\sum_{i=1}^{N} u_i (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \right] C^{-1} \right) = \mathrm{Trace}\left(CC^{-1} \right) = \mathrm{Trace}\left(I \right) = d \tag{13}$$

APPENDIX B. DERIVE β IN KHACHIYAN ALGORITHM

The Khachiyan (MVEE) algorithm is described in Algorithm 3, and includes a step in which weights are updated using a parameter β that is computed in Line 8 of the pseudocode. In this Appendix, we derive β.

Let u_i^0 indicate the current weight on the ith data sample. Then the current covariance matrix is given by[*]

$$R_0 = \Sigma_i u_i^0 \mathbf{x}_i \mathbf{x}_i^T, \tag{14}$$

and if j is the index associated with the largest Mahalanobis distance, then the updated weights are given by

$$u_i = (1 - \beta) u_i^0 + \beta \delta_{ij}. \tag{15}$$

Observe the updated weights still sum to one: $\Sigma_i u_i = (1 - \beta)\Sigma_i u_i^0 + \beta = (1 - \beta) \cdot 1 + \beta = 1$. The updated covariance matrix is then given by

$$R = \Sigma_i u_i \mathbf{x}_i \mathbf{x}_i^T = \Sigma_i ((1 - \beta) u_i^0 + \beta \delta_{ij}) \mathbf{x}_i \mathbf{x}_i^T = (1 - \beta)\Sigma_i u_i^0 \mathbf{x}_i \mathbf{x}_i^T) + \beta \mathbf{x}_j \mathbf{x}_j^T = (1 - \beta) R_0 + \beta \mathbf{x}_j \mathbf{x}_j^T \tag{16}$$

Since $\mathbf{x}_j$ is the sample with largest Mahalanobis distance in the current iteration, we have that $\mathbf{x}_j^T R_0^{-1} \mathbf{x}_j > d + 1$, but we want to change weights such that new R will have the property $\mathbf{x}_i^T R^{-1} \mathbf{x}_i = d + 1$. To get this result, we have to utilize the Sherman-Morrison-Woodbury formula defined by

$$(A + \mathbf{a}\mathbf{b}^T)^{-1} = A^{-1} - \frac{A^{-1}\mathbf{a}\mathbf{b}^T A^{-1}}{1 + \mathbf{b}^T A^{-1} \mathbf{a}}. \tag{17}$$

So,

$$r = \mathbf{x}^T R^{-1} \mathbf{x} = \mathbf{x}^T ((1 - \beta) R_0 + \beta \mathbf{x}\mathbf{x}^T)^{-1} \mathbf{x} = (1 - \beta)^{-1} \mathbf{x}^T (R_0 + \frac{\beta}{1 - \beta} \mathbf{x}\mathbf{x}^T)^{-1} \mathbf{x} \tag{18}$$

[*] In this derivation, we are referring to the vectorized implementation of MVEE in Algorithm 3. Thus, $\mathbf{x}_i \in \mathbb{R}^{d+1}$ corresponds to the ith data sample, augmented with a '1' as the last row. And R_0 and R refer to the current and next iteration of the augmented covariance matrix C_a defined in Line 4 of Algorithm 3. One consequence of this is that $\mathbf{x}_i R^{-1} \mathbf{x}_i$ is one plus the Mahalanobis distance; hence its average value is $d + 1$.

Let $\theta = \frac{\beta}{1-\beta}$. Now,

$$\begin{aligned}
\mathbf{x}^T R^{-1}\mathbf{x} &= (1+\theta)\mathbf{x}^T(R_0 + \theta\mathbf{x}\mathbf{x}^T)^{-1}\mathbf{x} \\
&= (1+\theta)\mathbf{x}^T(R_0^{-1} - \frac{R_0^{-1}\theta\mathbf{x}\mathbf{x}^T R_0^{-1}}{1+\mathbf{x}^T R_0^{-1}\theta\mathbf{x}})\mathbf{x}, \qquad (Sherman-Morrison-Woodbury) \\
&= (1+\theta)(r - \theta\frac{r^2}{1+\theta r}) \\
&= (1+\theta)r(\frac{1+\theta r}{1+\theta r} - \frac{\theta r}{1+\theta r}) = \frac{(1+\theta)r}{1+\theta r},
\end{aligned} \tag{19}$$

which is our new $\mathbf{x}^T R^{-1}\mathbf{x}$. Now we would like $d+1 = \mathbf{x}^T R^{-1}\mathbf{x} = \frac{(1+\theta)r}{1+\theta r}$.

$$d+1 = \frac{(1+\theta)r}{1+\theta r} \implies \theta = \frac{r-(d+1)}{rd} \tag{20}$$

Recall $\theta = \frac{\beta}{1-\beta} \implies \beta = \frac{\theta}{1+\theta}$. So,

$$\beta = \frac{\frac{r-(d+1)}{rd}}{1+\frac{r-(d+1)}{rd}} = \frac{r-(d+1)}{rd+r-(d+1)} = \frac{r-1-d}{(d+1)(r-1)}, \tag{21}$$

which is the expression used in Line 8 of Algorithm 3.

ACKNOWLEDGMENTS

This work was supported by the Laboratory Directed Research and Development (LDRD) program at Los Alamos National Laboratory.

REFERENCES

1. S. Matteoli, M. Diani, and J. Theiler, “An overview of background modeling for detection of targets and anomalies in hyperspectral remotely sensed imagery,” *J. Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS)* **7**, pp. 2317–2336, 2014.
2. B. R. Foy, J. Theiler, and A. M. Fraser, “Unreasonable effectiveness of the adaptive matched filter,” *Proc. MSS (Military Sensing Symposia) Passive Sensors Conference* , 2006.
3. P. C. Mahalanobis, “On the generalised distance in statistics,” *Proc. National Institute of Sciences of India* **2**, pp. 49–55, 1936.
4. I. S. Reed and X. Yu, “Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution,” *IEEE Trans. Acoustics, Speech, and Signal Processing* **38**, pp. 1760–1770, 1990.
5. A. D. Stocker, I. S. Reed, and X. Yu, “Multi-dimensional signal processing for electro-optical target detection,” *Proc. SPIE* **1305**, pp. 218–231, 1990.
6. D. Manolakis, D. Marden, J. Kerekes, and G. Shaw, “On the statistics of hyperspectral imaging data,” *Proc. SPIE* **4381**, pp. 308–316, 2001.
7. D. B. Marden and D. Manolakis, “Using elliptically contoured distributions to model hyperspectral imaging data and generate statistically similar synthetic data,” *Proc. SPIE* **5425**, pp. 558–572, 2004.
8. J. Theiler, C. Scovel, B. Wohlberg, and B. R. Foy, “Elliptically-contoured distributions for anomalous change detection in hyperspectral imagery,” *IEEE Geoscience and Remote Sensing Letters* **7**, pp. 271–275, 2010.
9. J. Theiler and D. Hush, “Statistics for characterizing data on the periphery,” *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* , pp. 4764–4767, 2010.
10. J. H. Friedman, “Regularized discriminant analysis,” *J. Am. Statistical Assoc.* **84**, pp. 165–175, 1989.
11. J. P. Hoffbeck and D. A. Landgrebe, “Covariance matrix estimation and classification with limited training data,” *IEEE Trans. Pattern Analysis and Machine Intelligence* **18**, pp. 763–767, 1996.

12. M. J. Daniels and R. E. Kass, "Shrinkage estimators for covariance matrices," *Biometrics* **57**, pp. 1173–1184, 2001.
13. O. Ledoit and M. Wolf, "Improved estimation of the covariance matrix of stock returns with an application to portfolio selection," *J. Empirical Finance* **10**, pp. 603–621, 2003.
14. O. Ledoit and M. Wolf, "Honey, I shrunk the sample covariance matrix," *Journal of Portfolio Management* **30**, pp. 110–119, 2004.
15. J. Schäfer and K. Strimmer, "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics," *Statistical Applications in Genetics and Molecular Biology* **4**(32), 2005.
16. N. M. Nasrabadi, "Regularization for spectral matched filter and RX anomaly detector," *Proc. SPIE* **6966**, p. 696604, 2008.
17. G. Cao and C. A. Bouman, "Covariance estimation for high dimensional data vectors using the sparse matrix transform," in *Advances in Neural Information Processing Systems 21*, pp. 225–232, MIT Press, 2009.
18. C. E. Caefer, J. Silverman, O. Orthal, D. Antonelli, Y. Sharoni, and S. R. Rotman, "Improved covariance matrices for point target detection in hyperspectral data," *Optical Engineering* **7**, p. 076402, 2008.
19. J. Fan, Y. Fan, and J. Lv, "High dimensional covariance matrix estimation using a factor model," *J. Econometrics* **147**, pp. 186–197, 2008.
20. S. Matteoli, M. Diani, and G. Corsini, "Improved estimation of local background covariance matrix for anomaly detection in hyperspectral images," *Optical Engineering* **49**, p. 046201, 2010.
21. A. Ben-David and C. E. Davidson, "Estimation of hyperspectral covariance matrices," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 4324 –4327, 2011.
22. J. Theiler, G. Cao, L. R. Bachega, and C. A. Bouman, "Sparse matrix transform for hyperspectral image processing," *IEEE J. Selected Topics in Signal Processing* **5**, pp. 424–437, 2011.
23. J. Theiler, "The incredible shrinking covariance estimator," *Proc. SPIE* **8391**, p. 83910P, 2012.
24. N. A. Campbell, "Robust procedures in multivariate analysis I: Robust covariance estimation," *Applied Statistics* **29**, pp. 231–237, 1980.
25. W. F. Baesner, "Clutter and anomaly removal for enhanced target detection," *Proc. SPIE* **7695**, p. 769525, 2010.
26. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, Wiley-Interscience, New York, 1987.
27. P. J. Rousseeuw and B. C. van Zomeren, "Unmasking multivariate outliers and leverage points," *Journal of the American Statistical Association* **85**, pp. 633–639, 1990.
28. P. J. Rousseeuw and K. Van Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics* **41**, pp. 212–223, 1999.
29. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
30. L. G. Khachiyan, "Rounding of polytopes in the real number model of computation," *Mathematics of Operations Research* **21**, pp. 307–320, 1996.
31. P. Kumar and E. A. Yildirim, "Minimum-volume enclosing ellipsoids and core sets," *J. Optimization Theory and Applications* **126**, pp. 1–21, 2005.
32. M. J. Kumar and E. A. Yildirim, "On Khachiyan's algorithm for the computation of minimum-volume enclosing ellipsoids," *Discrete Applied Mathematics* **155**, pp. 1731–1744, 2007.
33. W.-J. Cong, H.-W. Liu, F. Ye, and S.-S. Zhou, "Rank-two update algorithms for the minimum volume enclosing ellipsoid problem," *Computational Optimization and Applications* **51**, pp. 241–257, 2012.
34. J. Theiler, B. R. Foy, and A. M. Fraser, "Characterizing non-Gaussian clutter and detecting weak gaseous plumes in hyperspectral imagery," *Proc. SPIE* **5806**, pp. 182–193, 2005.
35. P. Bajorski, "Maximum Gaussianity models for hyperspectral images," *Proc. SPIE* **6966**, p. 69661M, 2008.
36. S. M. Adler-Golden, "Improved hyperspectral anomaly detection in heavy-tailed backgrounds," in *Proc. 1st Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, IEEE, 2009.
37. J. Theiler, "Ellipsoid-simplex hybrid for hyperspectral anomaly detection," in *Proc. 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, IEEE, 2011.
38. G. A. Tidhar and S. R. Rotman, "Target detection in inhomogeneous non-Gaussian hyperspectral data based on nonparametric density estimation," *Proc. SPIE* **8743**, p. 87431A, 2013.