Scoring Practices for Remote Sensing

Technical report prepared for CoVar Applied Technologies as part of the 32$^{\text{nd}}$ Workshop on Mathematical Problems in Industry, held June 13–17, 2016 at Duke University, Durham, NC

CoVar Representatives: Miles Crosskey, William LeFew, Kenneth Morton

Adel Al Weshah (Univ. of Akron), Aaron Bardall (NCSU), Valeria Barra (NJIT), Dean Duffy (NASA ret.), Hamza Ghadgali (Duke Univ.), Sarafa Iyaniwura (UBC), Hangjie Ji (Duke Univ.), Qingxia Li (Fisk Univ.), Richard Moore (NJIT), Ryan Pellico (Trinity Coll.), Christina Selby (RHIT), Razvan Stefanescu (NCSU), Melissa Strait (NCSU), Praneeth Vepakomma (Motorola Solutions), Zhe Wang (Duke Univ.), Andrés Vargas (RPI)

A summary of these results was presented to CoVar on June 17, 2016 by Andrès Vargas

Edited by Richard O. Moore (rmoore@njit.edu)

Contents

Abstract

This technical report examines the process by which sensor data from a region suspected to contain landmines is used to determine a set of alarm sites, including the manner in which the alarm set provided is scored against competing algorithms. The work and recommendations based on it were developed during the Mathematical Problems in Industry Workshop, held at Duke University June 13-17, 2016.

Chapter 1

Introduction

The United Nations (UN) estimates that fifteen to 20 thousand people die by landmine each year, most of them children, women and the elderly. Many of these deaths occur in areas where conflict ended long ago. Despite the 1997 passage of a UN Convention banning their production, use and export, landmines are still used today, guaranteeing that these unfortunate deaths will continue to occur for years to come.

The effort to address this global problem through landmine detection has been helped considerably by advances in tools for imaging and data processing. The availability of electromagnetic and acoustic sensor data across a broad region of the spectrum and the ubiquity of fast computers with ample storage are providing increasingly accurate predictions of where landmines lie within a region of interest (see Fig. 1.1 for an example scenario).

This report discusses algorithmic ways to improve the processing of sensor data to produce an alarm set of predicted target sites that can be provided to personnel in the field. The process by which remote sensing data, such as the radar scan in Fig. 1.2 of a defined region suspected to contain land mines, is used to obtain an alarm set such as that depicted in Fig. 1.3 involves multiple stages. Briefly summarized, these include:

1. processing of the raw data to remove artifacts, smooth over physically irrelevant spatial scales, etc.;

2. conversion of the processed data into a confidence map;

3. conversion of the confidence map into a discrete set of points and confidence values (the alarm set); and

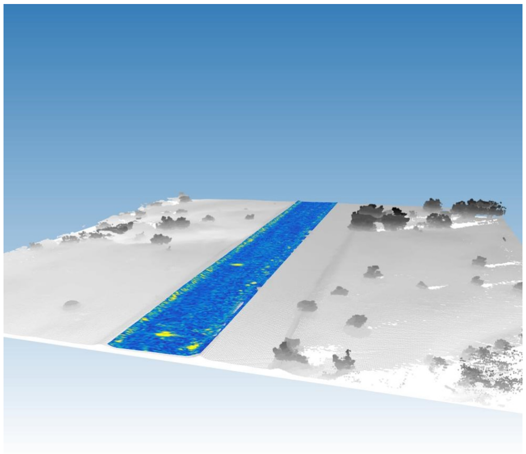

Figure 1.1: Example scenario of radar data obtained from a region of interest in the field.

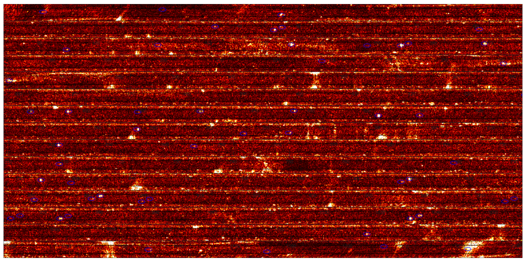

Figure 1.2: Stacked (false-color) image of radar data from detection experiment similar to that depicted in Fig. 1.1. Landmines identified by faint blue ellipses.

4. establishment of a threshold that flags a subset of the alarm sites as predicted target sites.

Each of these stages offers an interesting set of mathematical questions related to optimizing detection rates. This optimization is often framed by the client (e.g., the Department of Defense) in terms of obtaining the minimum possible false detection rate for a given positive detection rate, but more complex scoring mechanisms such as the receiver operating characteristic (ROC), depicted in Fig. 1.4 and explained in Appendix A, exist to quantify the skill of a particular algorithm.

In addition to comparing different alarm set generation algorithms against

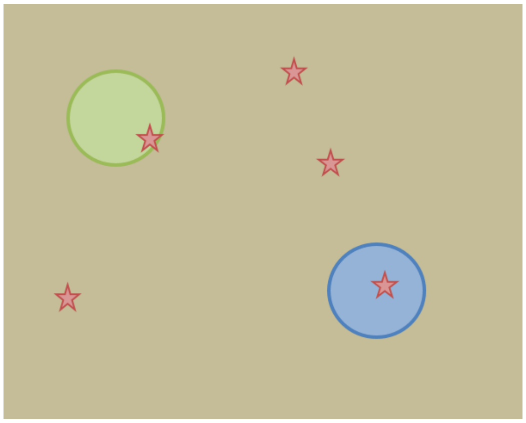

Figure 1.3: Cartoon of alarm set generated by algorithm applied to sensor data.

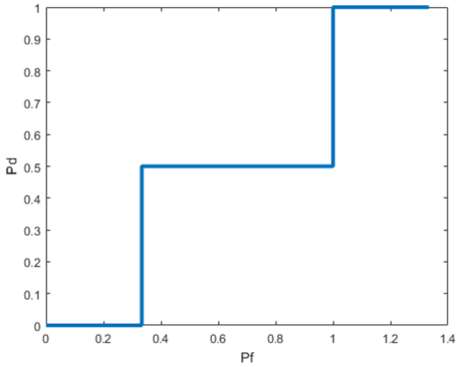

Figure 1.4: Idealized ROC curve plotting true positive detections P_d against false positives P_f.

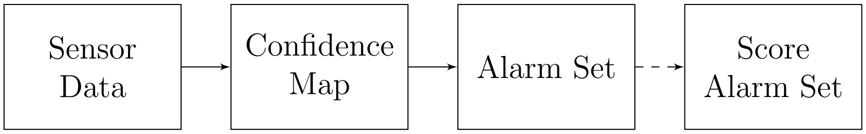

Figure 1.5: Schematic of process from provision of sensor data to scoring of alarm set generation algorithm.

each other, it is of interest to compare them with the best possible algorithm that does not use any sensor data. This algorithm provides an alternative alarm set that is generated based only on the geometry of the search region and assumptions regarding the distribution of targets. In the remainder of the report, references to a "diligent digger" denote algorithms that have no access to data but have otherwise been optimized with respect to *a priori* variables such as geometry and native landmine distributions to do as well as possible relative to a given scoring metric. A "diligent digger" algorithm contrasts with a "naive digger" algorithm, which has no access to sensor data and does not take into account geometry and other factors.

Figure 1 broadly depicts the process by which sensor data is transformed into an alarm set, that is then scored relative to uninformed algorithms to assess the value added by the sensor data. The remaining chapters in this report focus on the three arrows in this schematic, followed by conclusions and recommendations to CoVar Applied Technologies for further study.

Chapter 2

Building Confidence Maps from Sensor Data

The confidence level $C(x, y)$ is intended to qualitatively reflect the likelihood of discovering a landmine within a fixed distance from the point (x, y). In principle C could be made into a probability density; however, this would require detailed knowledge of uncertainty in the sensor data and the landmine distribution. Figure 1.2 clearly shows artifacts and noise in the sensor data that affect the background level so as to artificially increase the probability of an alarm placed in certain regions.

This effect of background noise variation can be mitigated by defining the noise background level through

$$n(x, y) = \frac{1}{m(D(x, y))} \int_{D(x,y)} C_0(x', y')\, dx'\, dy', \tag{2.1}$$

where C_0 is the original sensor data amplitude and $m(D)$ is the area of a rectangle (assumed here to be fixed in size) centered at point (x, y). The confidence map is then defined as

$$C(x, y) = C_0(x, y)/n(x, y). \tag{2.2}$$

Applying this denoising technique produces Fig. 2.2 from 2.1, with a noticeable improved in the ROC-FAR curve, as seen in Fig. 2.3. It is expected that more sophisticated image processing and filtering techniques that incorporate more information from the problem at hand would provide even better results.

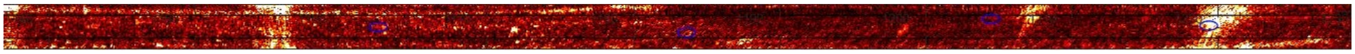

Figure 2.1: Original sensor data amplitude.

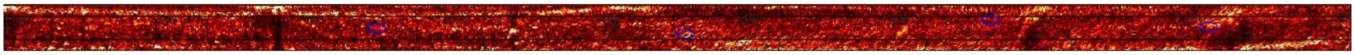

Figure 2.2: Confidence map obtained from sensor data after removing noise background.

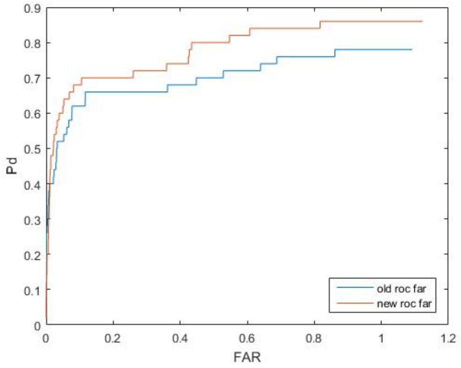

Figure 2.3: ROC-FAR curves for alarm sets generated from sensor data (blue) and confidence map removing noise background from sensor data (orange).

Chapter 3

Building Alarm Sets from Confidence Maps

3.1 Topological algorithm

Figure 3.5 shows a typical alarm set generated by the algorithm currently in use at CoVar. It is immediately apparent that there are several alerts that are placed very close to each other, or in other words, that are clustered. We want to reduce these clusters by retaining only the most prominent peak for each cluster. We do so by looking at the peaks that topologically belong to the same "mountain", or region of clustered peaks. We developed an algorithm based on topological data analysis: We gradually lower the threshold to which determine the peaks on the confidence map $C(x, y)$ (as showed in Fig. 3.1, where for simplicity sake we have considered a one-dimensional version of the confidence map $C(x)$); We find the regions that are topologically connected; For each region sufficiently large, we keep the prominent peak and then lower the remaining peaks to the threshold value.

On a two-dimensional confidence map $C(x, y)$ the result of our topological algorithm is illustrated in Fig. 3.2. A single peak has been retained over a region of several peaks. This topological approach can equivalently be regarded as a smoothing algorithm applied iteratively to produce a new confidence map from sensor data, in order to clean up large clusters of alerts. The results on the data are plotted in a close-up version of Fig. 1.2 alongside, for comparison sake, in Fig. 3.3.

To evaluate the performance, we plot the ROC curves obtained using just

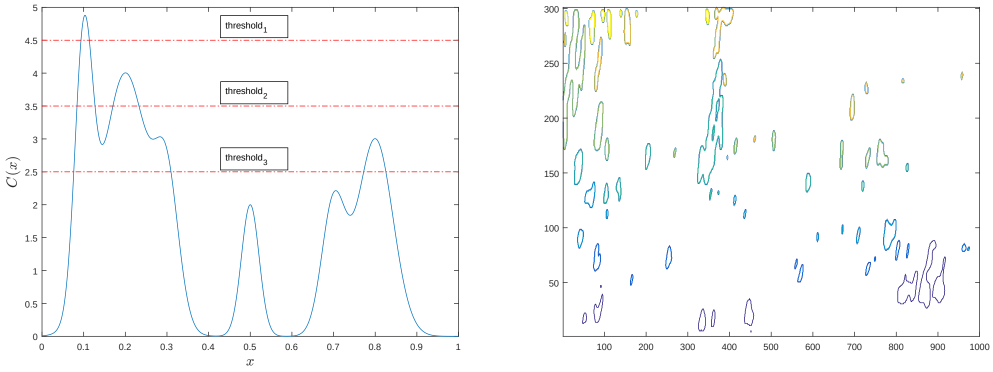

Figure 3.1: Left: A 1-d version of the confidence map $C(x)$; as we lower the threshold, we find larger regions of constant confidence that are topologically connected. Right: Contour plots of constant confidence.

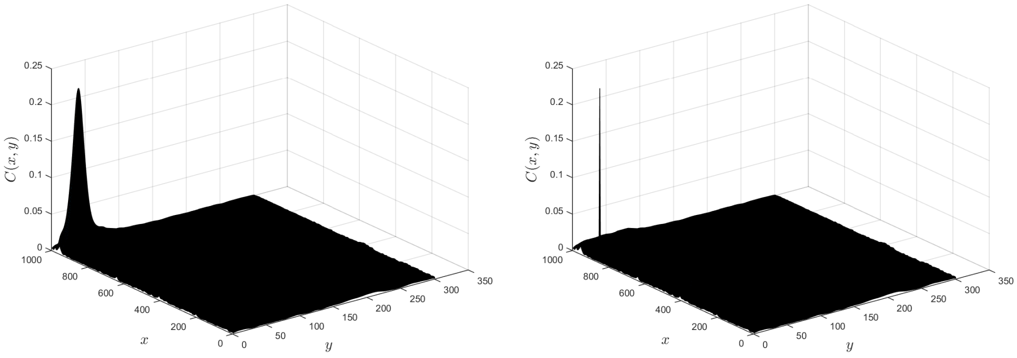

Figure 3.2: Left: Confidence map $C(x, y)$ given by CoVar, where alert points would be placed on every relative maximum. Right: The same confidence map with only one alert point retained at the maximum peak for the same cluster of alert points.

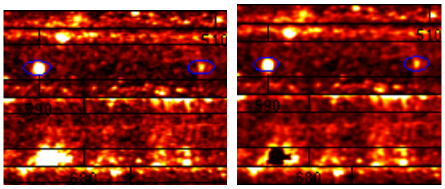

Figure 3.3: Left: Sensor data before the application of our algorithm. Right: Confidence map resulting from the clustering algorithm.

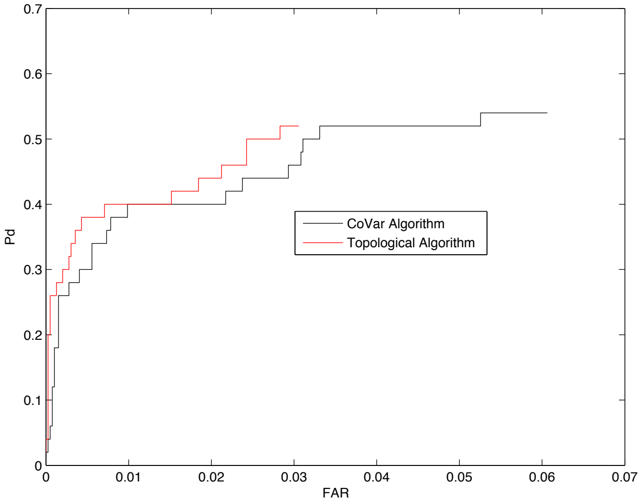

Figure 3.4: The ROC curves based on the given algorithm by CoVar and by the topological algorithm we developed.

the sensor data, and the one obtained with our topological algorithm. We can see in Fig. 3.4 that our algorithm has a ROC curve (red curve) that shows improved performance.

3.2 Alarm aggregation and hierarchical clustering

An alternative approach to address the alarm set clustering apparent in Fig. 3.5 uses aggregation based on hierarchical clustering with the goal of

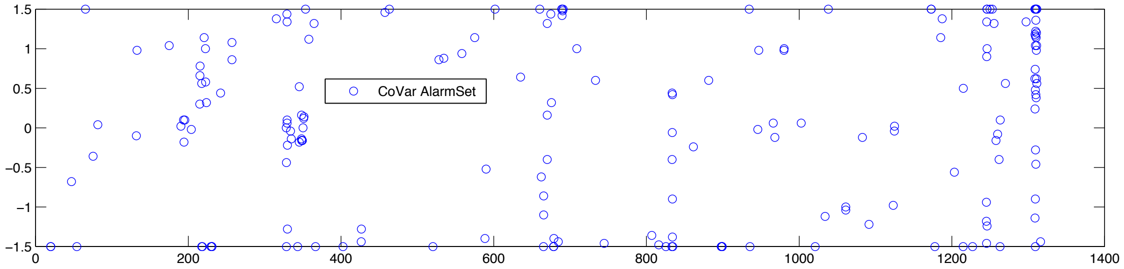

Figure 3.5: Alarm set predicted by CoVar algorithm.

reducing the spatial density of alarms and increasing the score of the algorithm as based on the ROC curve.

First we compute the Euclidean distance between each pairs of alarm locations, which is used to get a hierarchical cluster tree by the `linkage` function in MATLAB. Then we use the `cluster` function to construct clusters from the cluster tree: When the inconsistent value between a node and all of its subnodes are below the criterion 0.2, we group the leaves at the node and its corresponding subnodes into a cluster. Within each cluster the alarm location with the highest confidence is then set to be the "center" of the cluster and we only keep this alarm with highest confidence in the cluster.

To study the effectiveness of this clustering method, we plot the reduced alarm sets with red squares in Figure 3.6 from the full alarm set generated by CoVar algorithm in Figure (3.5). It is worth mentioning that there is a blue circle beneath each red square in Figure 3.6, while the uncovered blue circles correspond to the alarms that are reduced by the clustering algorithm. The comparison in terms of ROC curve in Figure (3.7) shows that the reduced alarm set generated by the clustering algorithm performs slightly better than the original algorithm when FAR is small. Better performance of this algorithm may be achievable if a more appropriate criterion for the inconsistent value is chosen.

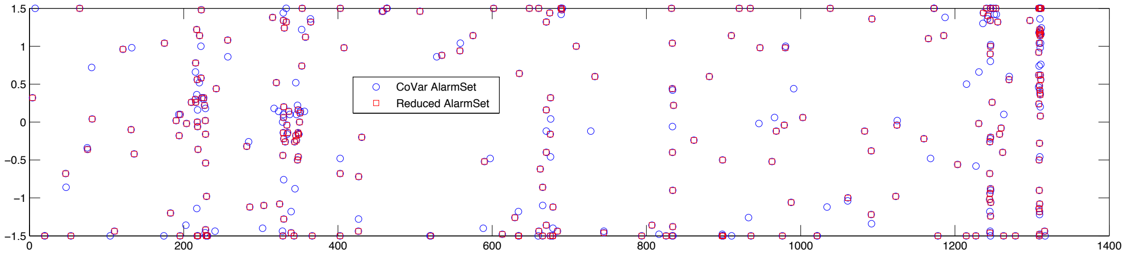

Figure 3.6: A comparison of alarm set before (blue circles) and after (red squares) hierarchical clustering. The alarm set after clustering has fewer alarms.

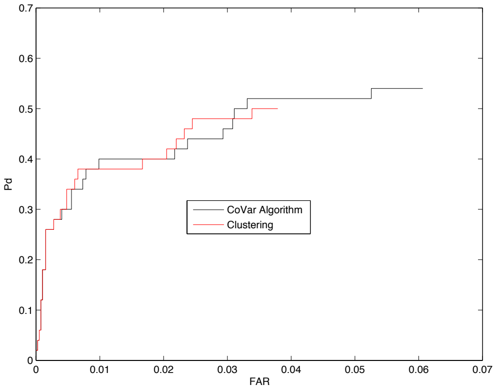

Figure 3.7: A comparison of the ROC curves for alarm sets generated by the CoVar algorithm (black) and after applying the clustering algorithm (red).

3.3 Iterative centroid scheme for alarm set generation

Covar currently determines the location of alarms by determining the location of local maxima in the confidence map. In practice, a user would dig a circle of radius r around the location of the alarm in search for an explosive device. This may not be the appropriate action if the confidence map is not symmetric about the alarm location. An alternate approach for placing alarms utilizes the location of the centroid of the region over the disk of radius r, initialized at the location of the local maximum. The density function used

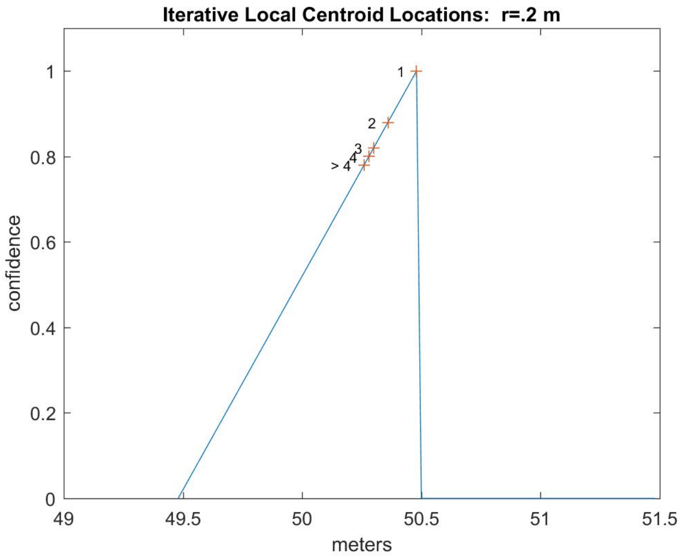

Figure 3.8: One-dimensional example of results of centroid iterative algorithm for alarm generation. Numbers label result of iteration number.

in the centroid calculation is the provided confidence map. The centroid is iteratively recomputed over the support of the circular disk until it converges. The determined centroid location is a new alarm location.

This technique may also provide information about how to dig about an alarm location in a more informed manner. Instead of digging in a circle about a given local maximum, the customer may dig in a path determined by the locations of the centroids determined in this iterative process.

3.4 Integrating confidence map over eligible configurations

Alarm sets can also be generated using a combinatorics approach over a discretized domain, taking all compliant target configurations as equally likely. Let $n_{\Delta x}$ be the number of eligible configurations given a discretization Δx of the domain, and let $I_j(x)$ be an indicator function for the domain coverage associated with configuration j for $j = 1, \ldots, n_{\Delta x}$. The confidence score associated to each eligible configuration on domain $[0, L]$ is then

$$C_j = \int_0^L C(x) I_j(x)\, dx. \tag{3.1}$$

These confidence scores can be rank-ordered to determine the most likely landmine locations and orientations from among the eligible configurations.

Figure 3.9 illustrates this detection approach using a noisy sensor dataset generated by applying a Gaussian random noise field to an indicator function with amplitude μ defined on uniform partition $\{x_i\} = \{0, \Delta x, 2\Delta x, \ldots, L\}$, i.e.,

$$s(x_i) = \mu I(x_i) + \xi(x_i) \tag{3.2}$$

where $E[(\xi(x_i))\xi(x_j)] = \chi\delta_{ij}$. Whereas an algorithm basing its alarm set on the confidence peaks misidentifies the single target in the domain, a method integrating the confidence $C(x)$ over the support of the target configuration correctly identifies its location.

Figure 3.10 further demonstrates that, for sensor datasets generated using Eqn. 3.2 with varying values of signal strength μ, integrating the confidence map over the support of the landmine configuration consistently outperforms the peak detection method. Both methods naturally agree in the extremes of vanishingly small signal strength (i.e., random detection) and very large signal strength (perfect detection). This would not be the case with multiple landmines, as illustrated in Sec. 3.4.1.

3.4.1 2-d problem

The combinatorics approach above can be adapted in a straightforward way to a rectangular domain in two dimensions if the targets are also assumed rectangular, yielding configurations similar to those depicted in Fig. 3.11

Figure 3.4.1 provides an example sensor dataset and the 3-alarm set generated by the integrated and peak detection methods. The improvement in predictive skill is significantly increased in two dimensions relative to the one-dimensional case, as illustrated by Fig. 3.4.1. In this case, the peak detection method does not asymptote to perfect detection in the limit of large signal strength due to the nonvanishing probability of two peaks occurring over a single landmine.

The primary difficulty with this approach is the computational cost associated with determining all eligible configurations. This difficulty is mitigated

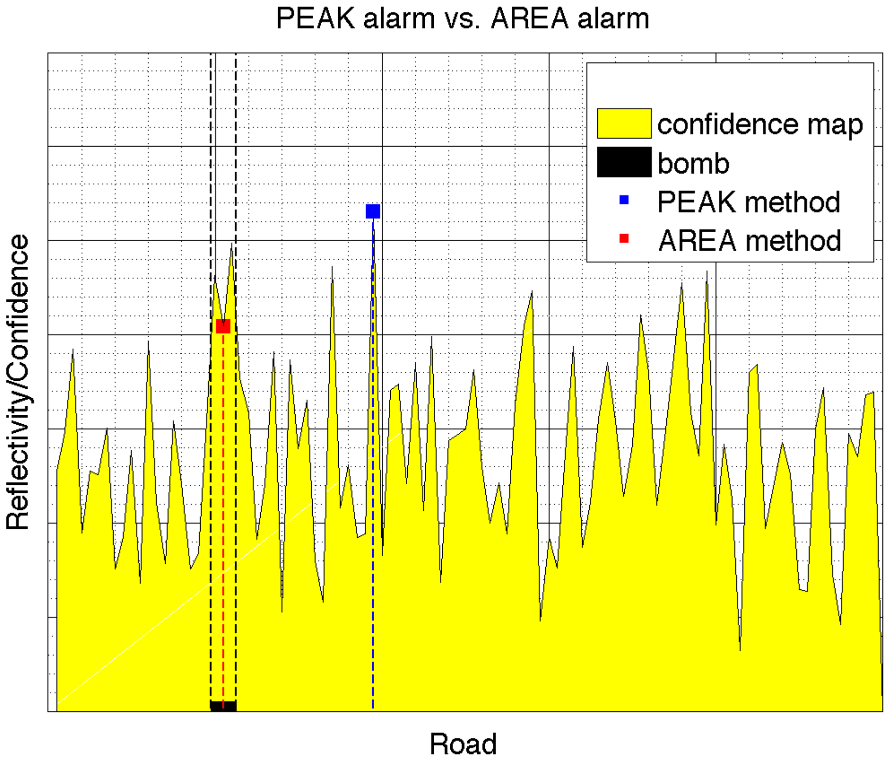

Figure 3.9: Area method integrates confidence map over eligible configurations to successfully identify target. Covar algorithm based on peak detection misidentifies target. Here $\mu = 1.5$ and $\chi = 1$.

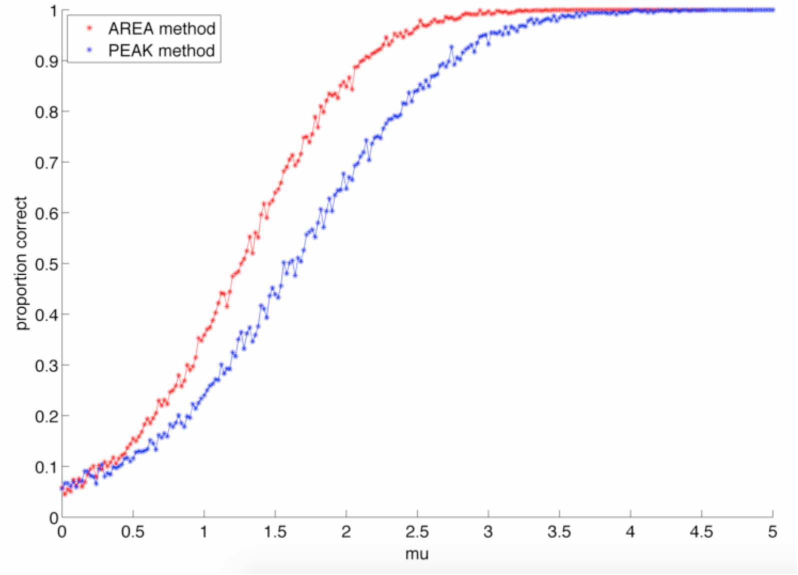

Figure 3.10: Comparison of integrated and peak detection methods with varying signal strength μ and fixed unit noise variance. Performance is averaged over 1000 simulations.

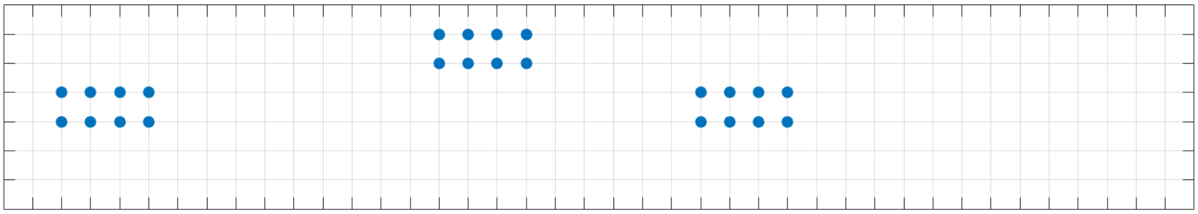

Figure 3.11: Depiction of three eligible landmine configurations on 2-d domain.

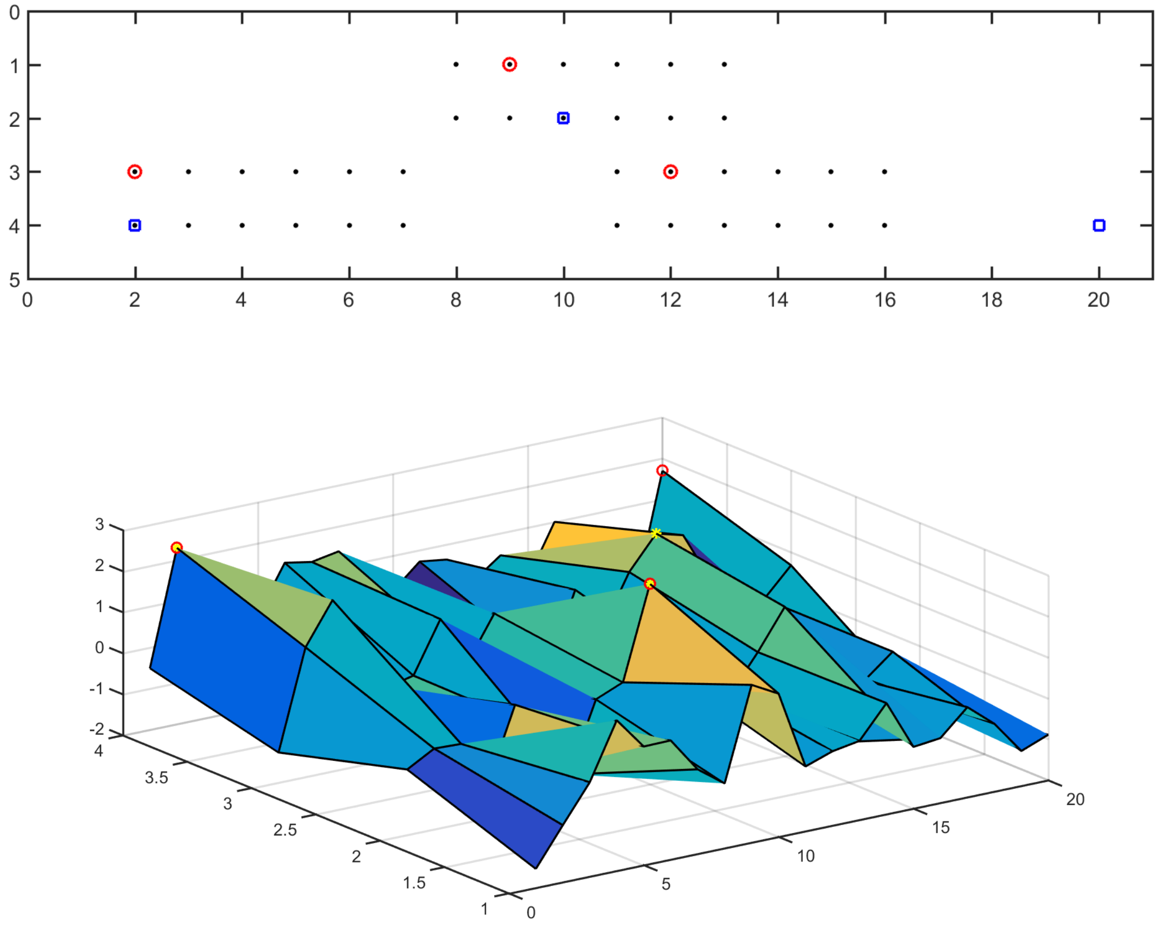

Figure 3.12: Alarm sets generated by the integrated (red circles) and peak (blue squares) detection methods. The peak detection method misidentifies one of the targets.

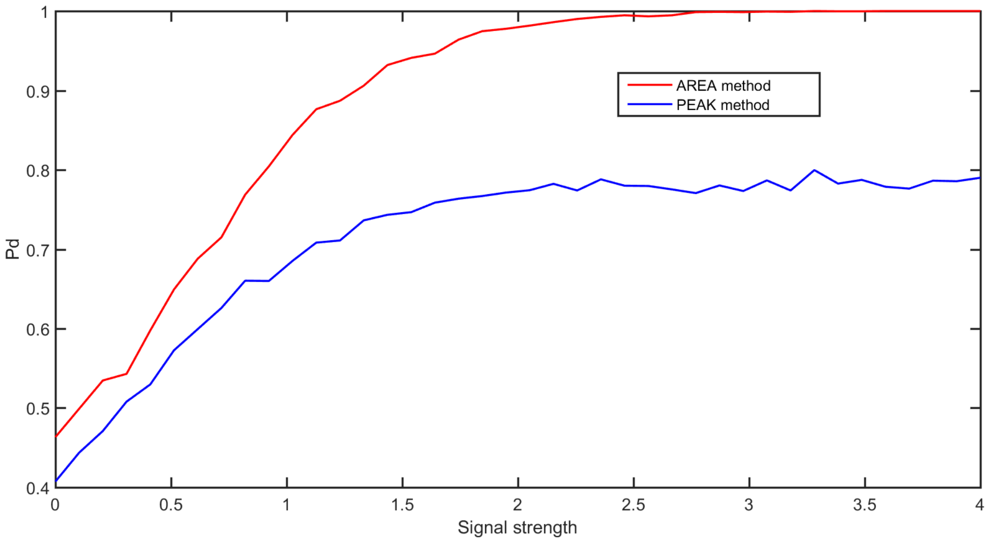

Figure 3.13: Probability of detection vs. signal strength μ, using three 1×3 bombs in a 2×10 domain.

by computing a configuration library in advance to apply simple screens for the purpose of integrating confidence scores.

Chapter 4

Diligent Diggers

Taken as a whole, an alarm set generated from sensor data should reasonably be expected to outperform one that does not incorporate the data. If the process of generating an alarm set is considered sequentially as targets are identified in reverse order of confidence, however, then at some point the confidence peaks upon which alarm sites are based become practically indistinguishable from noise. At this point, a data-uninformed but otherwise optimized "diligent digger" algorithm should perform equally as well as the data-informed algorithm, but with less associated computational cost. The development of diligent digger algorithms that apply under different geometries, different known mine distributions, and different scoring methods is therefore important both for providing a baseline against which the data-informed algorithms can be measured, and also for providing a fast and cheap alternative to the data-informed algorithm once the high-confidence targets have been identified.

The performance of a diligent digger algorithm will depend on the scoring procedure used and will obviously vary depending on the ground truth but should be optimal in the sense of expectations against the target distribution. Lower bounds for the performance of this "diligent digger" are provided by the presumably equivalent (in an averaged sense) examples of locating the alarm set sites at vertices of a regular mesh and drawing the sites from a distribution that is uniform in the x- and y-coordinates. It is not clear a priori how to choose the diligent digger's confidence values or how to choose the size (cardinality) of the diligent digger's alarm set.

The sensor-based confidence map is very effective at detecting targets for locations with high confidence values. However, the alarms generated with low confidence values from the confidence map do not perform well at detecting the remaining targets. From the slope of the ROC-FAR curve at these points, we expect that at some confidence threshold the sensor-based algorithm does no better than alarms that are randomly placed. To test this, we have constructed several uninformed algorithms that place alarms without the use of sensor data. These uninformed algorithms are then improved by using information about the length scale of the landmines and detection radius of the alarms to place alarms sufficiently far from one another maximize the detection area.

4.1 Random digger

In distinction to the optimized diligent digger discussed above, an uninformed 'naive digger' algorithm assigns random confidence values to alarm locations that are either chosen randomly or arranged in a grid format. As expected, uninformed 'naive digger' algorithm alone perform substantially worse than the data-informed algorithm. The ROC-FAR curves for the sensor-based algorithm and the two naive digger algorithms are shown in Fig. 4.1. In this figure two realizations are shown of the random alarm placement algorithm and the grid placement algorithm in addition to the average curve from 1000 simulations for these two methods. The sensor-based algorithm identifies targets well using alarms with high confidence values, which is associated with the steep part of the ROC curve for small FAR values. The performance of the sensor-based algorithm decreases as more targets are identified.

These uninformed methods were then augmented with the informed algorithm to attempt to improve later target detection performance. Setting a confidence threshold for which the confidence map would no longer be trusted, the lower confidence alarms were then replaced with the naive digger algorithms discussed above. Though there was better performance in placing alarm locations in a uniform grid than the random alarm location when used alone, the performance of these two augmented algorithms is virtually the same. Results for these augmented methods are shown in Fig. 4.2.

The average performance of the augmented random method can be compared to the expected slopes of these lines based on the remaining target area $A_{\text{remaining targets}}$ and total area A_{tot} of the region of interest. The expected slope for a fixed FAR value can be calculated using the relationship between number of true alarms t and false alarms f. Assuming random alarm

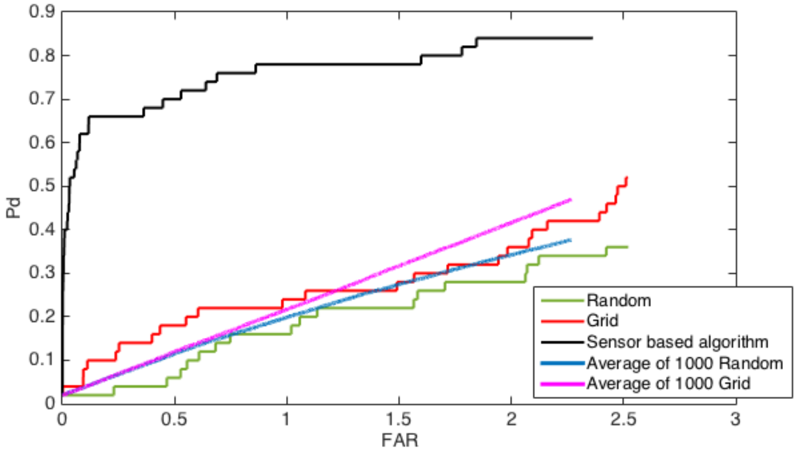

Figure 4.1: ROC-FAR curves for the sensor based algorithm and uninformed algorithms.

placement we expect that this ratio is equal to the ratio of $A_{\text{remaining targets}}$ to the area remaining after target removal. This leads to the following equation for the expected value of the slope for the performance of the augmented random method

$$E[\text{slope}] = \frac{A_{\text{remaining targets}} \cdot A_{\text{tot}}}{(A_{\text{tot}} - A_{\text{remaining targets}})N}, \tag{4.1}$$

where N is the number of total targets. These expected slopes are shown with the quivers in Fig. 4.3, while the average of the random simulations are shown with the colored curves in Fig. 4.3. From the figure we see that the slopes from the simulations are very similar to the expected slopes given by the quivers. For example, at a 0 FAR value, the expected slope is 0.1968, and the slope of the 1000 averaged simulations is 0.1828. It is also clear from this figure that switching to a random algorithm at some confidence threshold performs just as well as the sensor based algorithm.

4.2 Shape-optimized diligent digger

These methods can be improved by using information about the expected landmine size. To construct our 'diligent' digger with this information, we build on the idea of a uniform grid but avoid placing alarms at locations within a landmine radius distance to an alarm already placed using sensor data. This guarantees that the remaining area is not double counted. This systematically performs better than the removed low-confidence targets. The ROC-FAR plot for these methods is plotted in Fig. 4.4.

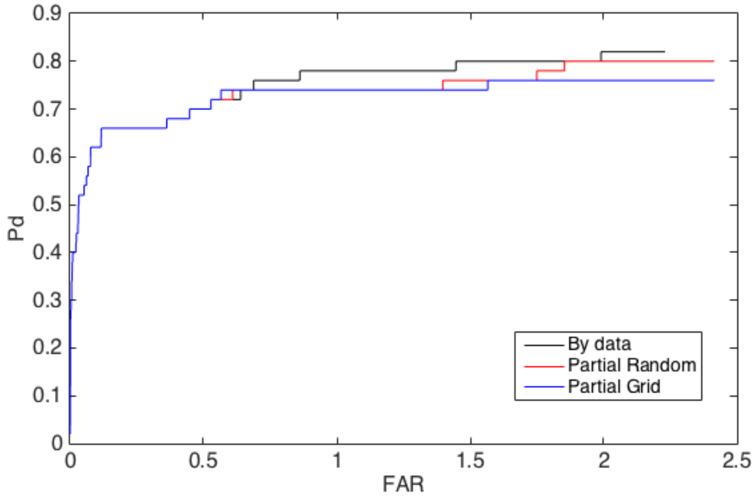

Figure 4.2: ROC-FAR curves for the sensor based algorithm, and augmented sensor based algorithms for alarms with high confidence and uninformed algorithms for alarms with low confidence.

Examples of the alarm placements for the augmented algorithms are shown in Fig. 4.6.

4.2.1 Distribution of configurations

The alarms placed in the bottom right axes of Fig. 4.6 simply exclude grid-points based on a minimum distance criterion. A more accurate but also more computationally intensive alternative is to compute all eligible configurations of landmines with a given size and configuration, as discussed above in Sec. 3.4. In the following sections we elaborate on this approach using a one-dimensional example, commenting that adding dimensions is straightforward.

Consider the placement of n targets of fixed length L in an interval of length 1. To devise an optimal strategy for the diligent digger, we need to

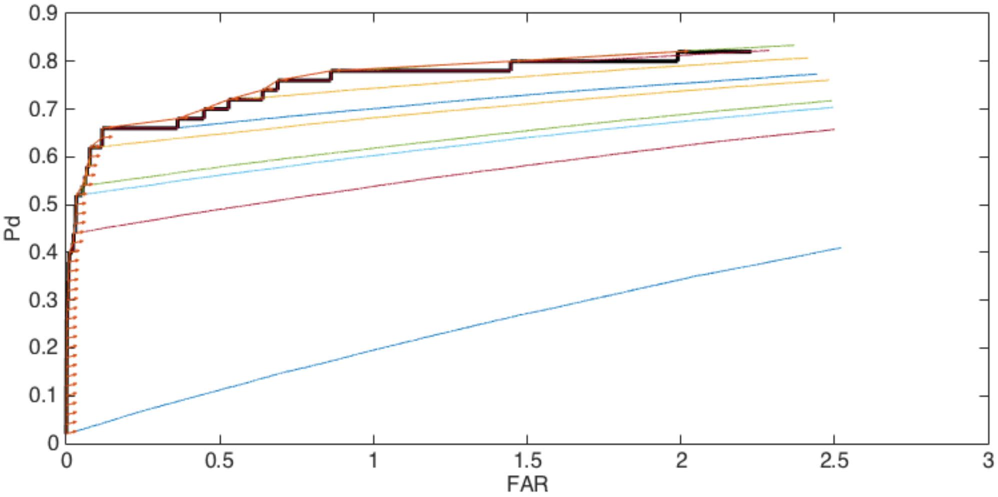

Figure 4.3: ROC-FAR curves taking the average of 1000 simulations using the sensor based and random augmented algorithm. The colored curves correspond to switching to difference confidence thresholds of placing alarms using a random scheme instead of using the sensor data. The quivers correspond to the expected slopes of the curves based on the remaining target area and lane area.

understand what constitutes a uniform distribution over the set of allowable target configurations. If we choose x_i, $i = 1, \ldots, n$ to represent the left end points of the indistinguishable targets, we may assume without loss of generality that $x_1 \geq 0$, $x_{i+1} - x_i \geq L$ for $i = 1, \ldots, n-1$, and $x_n \leq 1 - L$. A uniform distribution in $\mathbf{x} \in \mathbb{R}^n$ that satisfies these hyperplane constraints can be expressed by defining $y_0 = x_1$, $y_i = x_{i+1} - x_i - L$, and $y_n = 1 - L - x_n$. Then $\mathbf{y}$ is assumed uniformly distributed with all components positive and confined to hyperplane

$$\sum_{i=0}^{n} y_i = 1 - nL. \tag{4.2}$$

A concise algorithm for generating $\mathbf{y}$ draws n values ξ_i from $U[0, 1 - nL]$, indexed in increasing order, and sets $y_0 = \xi_1$, $y_i = \xi_{i+1} - \xi_i$ for $i = 1, \ldots, n-1$, and $y_n = 1 - \xi_n$. Thus, $x_1 = \xi_1$ and $x_i = \xi_i + iL$ for $i = 2, \ldots, n$.

This distribution in combination with a scoring method provides a systematic basis to optimize alarm placement, with objective function $E[S]$

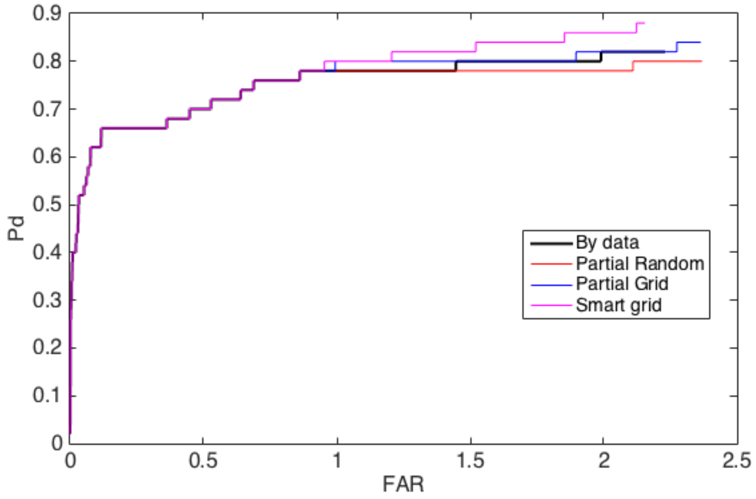

Figure 4.4: ROC-FAR curves for the sensor based algorithm and augmented algorithms for alarms with low confidence ("smart grid" refers to grid placement with minimum distance criterion).

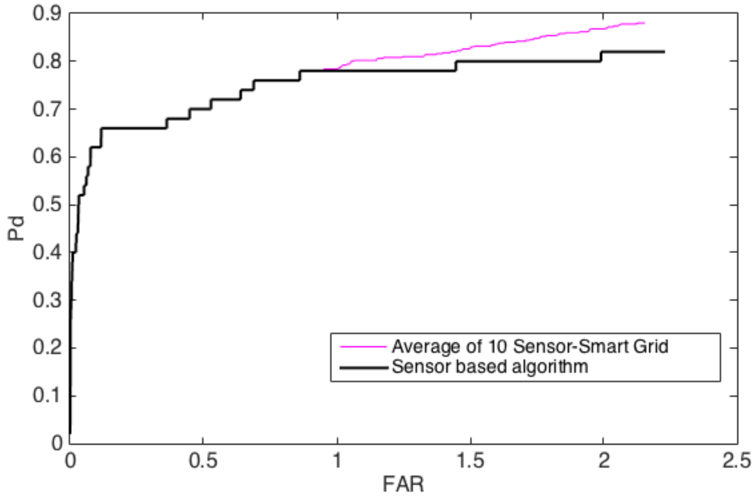

Figure 4.5: ROC-FAR curves for the sensor based algorithm and the average of 10 trials using the augmented grid algorithm with minimum distance criterion.

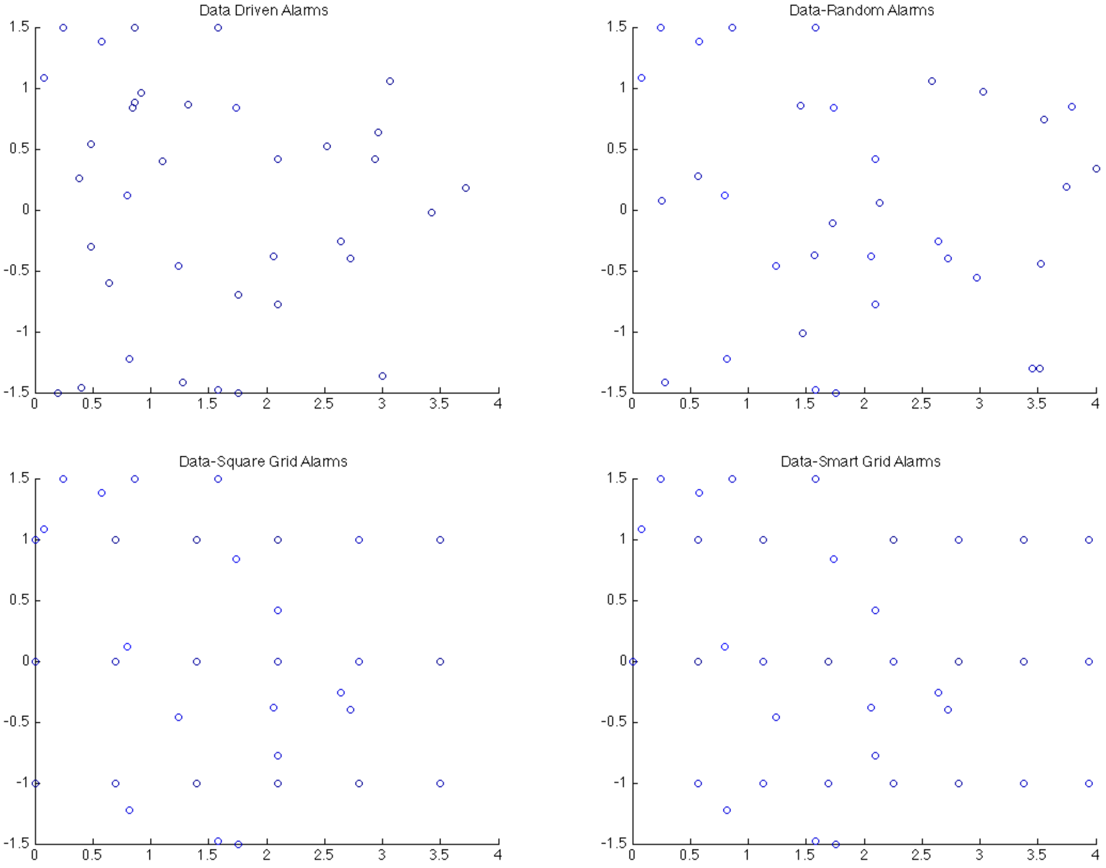

Figure 4.6: Example alarm placements using the sensor based algorithm (top left), augmented sensor based algorithm for highly confident locations and random placement for low confidence alarms (top right), augmented algorithm using grid placement for low confidence alarms (bottom left), augmented algorithm using grid placement with minimum distance (bottom right).

where S is the scoring metric (e.g., AUC or another metric derived from the ROC) and the expectation is taken over the above distribution. If we simply use probability of detection as a proxy for S, we can compute the spatial density associated with the random distribution described above. Figure 4.2.1 illustrates four sample densities for different choices of n targets of fixed length L. In each case, as $L \to 1$ the density acquires structure that informs the diligent digger where to place the alarms. As $L \to 0$, the density approaches uniformity, suggesting that all alarm placements with separation of at least L between each other and from the boundaries have equal utility relative to the objective of expected probability of detection.

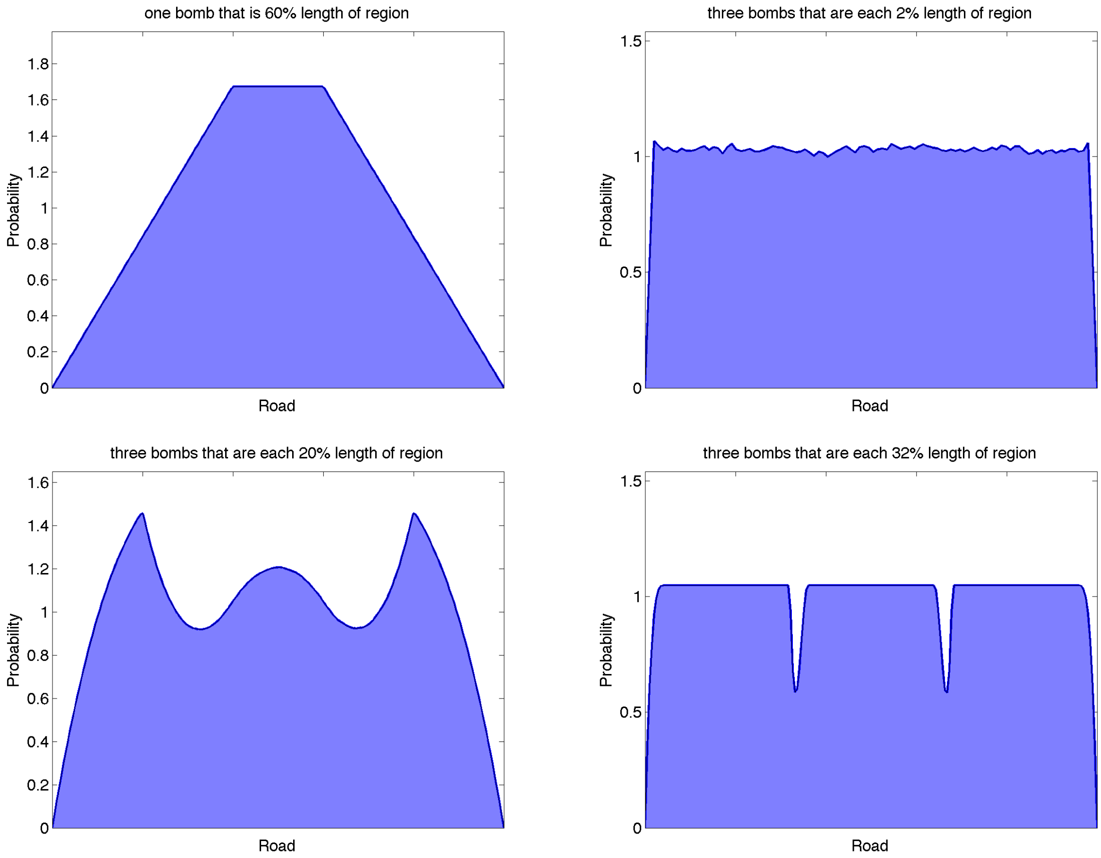

Figure 4.7: Four sample densities provided by eligible configurations in 1-d domain. Examples are 1 bomb with length 0.6L (top left), 3 bombs with length 0.02L (top right), 3 bombs with length 0.2L, and 3 bombs with length 0.32L.

Chapter 5

Alternative Scoring Methods

Any algorithm used to generate alarm sets from sensor data must be tested against alternative algorithms that may preferable due to their computational cost or ease of deployment, starting with the diligent digger algorithms discussed in Chapter 4. This comparison is based on a scoring metric which may be supplied by the customer, e.g., minimum false detection rate for fixed positive detection rate, or may be a standard metric discussed in the statistical literature, such as the AUC described in App. A. A third alternative proposed here treats the generated alarm set and the target locations as two sets of data generated from distributions that may or may not be statistically independent.

In this section, we develop another alternative to algorithm evaluation based on a hypothesis testing framework. The alarm-detection algorithm can be thought of as a bivariate random variable X^a whose realizations are alarm location predictions $(i, j)^a$. Similarly, the true locations of the alarms can be thought of as realizations $(i, j)^t$ of a bivariate random variable X^t. Denote the probability distributions of X^a and X^t as D^a and D^t, respectively. Each time the algorithm is run, we obtain a sample $x^a = (x^a_1, x^a_2, ..., x^a_{n_a})$ from D^a, and each set of true alarm locations $x^t = (x^t_1, x^t_2, ..., x^t_{n_t})$ is a sample from D^t. Note that the elements of these samples are pairs of indices (i, j) that correspond to either bomb location predictions (x^a) or true bomb locations (x^t). To test the performance of the algorithm, we can determine if x^a comes from the same distribution as x^t via a hypothesis test that we define as follows:

$$H_0 : D^a = D^t$$

$$H_1 : D^a \neq D^t$$

Here H_0 denotes the null hypothesis and H_1 the alternative. The design of the crossmatch statistical test means that, for bomb detection algorithm evaluation, we hope that the null hypothesis will not be rejected because this will mean that the algorithm performed well.

5.1 Hypothesis testing with the crossmatch statistics

To construct the crossmatch test statistic A_1, first group the samples into one vector $x = (x^a, x^t)^T$. Next, find the optimal non-bipartite pairing of the elements of x. In words, the optimal non-partite pairing of x is a paring of the elements of x such that the the sum of the distances between the elements of the pairs is minimized. This optimal pairing is a tractable optimization problem that can be performed in $O(n^3)$ operations, where $n = n_a + n_t = dim(x)$. For more details on the pairing, see [3].

Let x_o be the optimal non-bipartite pairing of x and define A_1 to be the number of pairs in x_o that have one member of the pair belonging to x^a and the other member belonging to x^t. Intuitively, a small value of A_1 indicates that the sum of the pairwise distances was minimized by forming pairs where both elements either belonged to x^a or x^t so that, in a sense, the samples x^a and x^t form distinct clusters. On the other hand, a large value of A_1 indicates that there was no advantage (with respect to the minimization problem) to be gained by keeping points belonging to x^a and x^t separate, and thus we can conclude that there is a lot of overlap between x^a and x^t. Hence, we reject the null hypothesis if A_1 is small. Formally, the distribution of A_1 under the null-hypothesis is

$$\mathbb{P}(A_1 = a_1) = \frac{2^{a_1}\ (n/2)!}{\binom{n}{n_a}\ a_0!\ a_1!\ a_2!}\ ,$$

where $a_0 = \frac{n_t - a_1}{2}$ and $a_2 = \frac{n_a - a_1}{2}$. a_0 is the number of pairs that have no members belonging to x^a. a_2 is the number of pairs that have both members belonging to x^a. This distribution is exact, i.e. we need not make any assumptions about the distribution of A_1. [3] describes what to do when n is odd and also performs the full derivation of $\mathbb{P}(A_1 = a_1)$. We reject the null hypothesis if $\mathbb{P}(A_1 \leq a_1) < \alpha$, where α is the desired significance level.

We did not calculate the p-value by hand, but rather with the crossmatch package in R. The results for different numbers of alarms are displayed in Table 5.1.

Table 5.1: Crossmatch p-values for different numbers of alarms placed

# of Alarms Placed	p-value
200	0.1218
1000	0.8714
2100	0.2807
4000	0.7114
5000	0.6915
7000	0.6627

In Table 5.1, high p-values indicate good predictions by the algorithm. We see that the optimal number of alarms to place could be in the vicinity of 1000. Future work could strive to rigorously determine the optimal number of alarms.

5.2 Testing for independence with distance correlation

In situations when the cross-match test rejects the null-hypothesis, we conclude that the algorithm output (which comes from probability distribution D^a) is not an accurate representation of the true locations of the alarms (which come from probability distribution D^t). Even if this happens, all may not be lost: if it turns out that there is any statistical dependence between the D^a and D^t, then this would imply there exists a mapping $F : Val(X^a) \longrightarrow Val(X^t)$, where $Val(Y)$ is the space of values that can be taken by a random variable Y. If we could then find a good approximation $\hat{F}$ to F, then we could use $\hat{F}$ to transform the algorithm output into something that better reflects the true alarm locations. To test for dependence between D^a and D^t, we can use the distance correlation hypothesis test:

$$H_0 : D^a \perp\!\!\!\perp D^t$$

$$H_1 : D^a \not\perp\!\!\!\perp D^t$$

where $\perp\!\!\!\perp$ denotes statistical independence.

The distance correlation test measures both linear and non-linear statistical dependence. Acceptance of the null-hypothesis implies that the alarms and true locations are statistically independent, which would mean the mapping F does not exist. In this case, it would be probably be best to go back to the drawing board and tweak the algorithm. On the other hand, if the null hypothesis is rejected, then we could turn to more sophisticated machine learning methods to try to approximate F.

Chapter 6

Conclusions and Recommendations

The process by which sensor data from a region suspected to contain landmines is used to determine a set of alarm sites is complex and influenced by several conflicting priorities. These include the absolute need for positive detection of targets, the desire for minimization of false detections, and the computational expense of any algorithms that are applied. The procedure discussed in this report focuses on the approach used by CoVar that generates a confidence map from sensor data, then identifies alarm sites from information contained in the confidence map. Each stage, from sensor data to confidence map, from confidence map to alarm set, and from alarm set to a score relative to other competing methods, including "diligent digger" algorithms that use no sensor data, lends itself to a variety of approaches depending on the priorities in the optimization.

The following conclusions have been based on the analysis described in this report:

1. Artifacts oriented transversally to the scan direction (i.e., across the road) can be **smoothed** to mitigate the risk of false detections;

2. **De-clustering data** can identify topologically connected regions in the confidence map based on a level set method and that this allows the identification of the most relevant peaks to target detection;

3. Integrating the confidence map against **library of configurations** can improve target detection rate compared to the peak detection method

currently in use;

4. **Aggregation is not sufficient** to address performance gap between above two methods (integration against configurations and peak detection method);

5. A **library of configurations can also be used to inform "diligent digger"** algorithm, can be computationally intensive but approaches probability density in limit;

6. Simulations confirm that sensor-based algorithm currently in use performs well for easily identifiable targets, but decreases in effectiveness as confidence level drops;

7. Improvements to "uninformed digger" algorithm can have an impact, with variations including **placing remaining alarms on regular grid tiling unoccupied part of domain**, and **randomly placing alarms with weight proportional to partition area**; and

8. Alternative scoring metrics such as those based on **hypothesis testing that target sites are drawn from distribution derived from confidence map** can provide more insight into amount of information contained in confidence map beyond that used to form alarm set.

Based on these findings, we recommend:

1. Implementation of **more sophisticated data smoothing techniques** that account for structure;

2. Further investigation of detection methods that **integrate configurations** or **probability densities** against a confidence map;

3. Use of **greedy algorithms** to select most likely configurations rather than a comprehensive search; and

4. Use of alarm set scoring methods with **statistical tests** that use more information from the confidence map or smoothed data.

Chapter 7

Acknowledgments

The CoVar team from the 2016 MPI Workshop gratefully acknowledges CoVar Applied Technologies for providing this challenging problem, Duke University for hosting the workshop, and the National Science Foundation for financial support.

Appendix A

ROC and ROC-FAR

Scoring is based on the receiver operating characteristic (ROC) curve plotting the probability of correct detection against the probability of false detection, as shown in Fig. A.1. The ROC is parameterized by the threshold value and naturally interpolates between the lower left corner, where an extremely high threshold implies no detections at all, and the upper right corner, where an extremely low threshold implies all detections, both correct and false. Intermediate threshold values should ideally produce a low false positive rate and high true positive rate, i.e., a concave ROC curve. An uninformative (e.g., random) algorithm would produce correct and false positives with equal probability, producing a straight line connecting the bottom left and top right corners. Algorithms that incorporate additional target data are expected to provide curves of increasing concavity, limiting to a Heaviside function that immediately detects all targets before making a single false detection.

Various statistics are used to quantify an algorithm based on its ROC, one of the most common being the area under curve (AUC). This approximately corresponds to the probability that the algorithm will assign a higher value to a true site than to a false one.

A variation on the ROC plots the detection probability against the false detection rate normalized by the area of the region of interest remaining after removal of detected targets, referred to in Fig. A.2 as a receiver operating characteristic - false alarm rate (ROC-FAR). The arrows illustrated in Fig. A.2 point in the direction corresponding to an uninformative (e.g., random) detection algorithm after removal of detected target area. As the threshold is lowered, the ROC slope is also reduced as the algorithm grows increasingly less effective at detecting the remaining targets. When the ROC

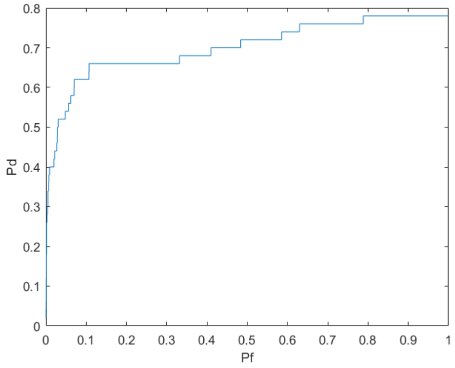

Figure A.1: Receiver operating characteristic (ROC).

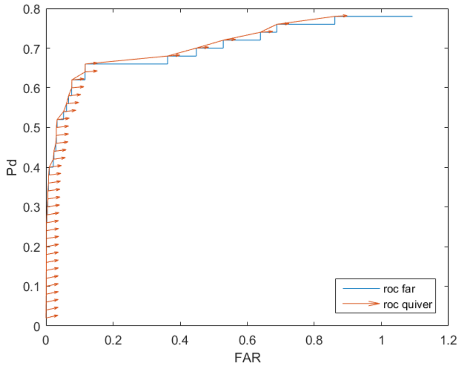

Figure A.2: Receiver operating characteristic with false alarm rate as ordinal axis (ROC-FAR).

slope is comparable to the arrow slope, the algorithm is essentially useless in translating the data into meaningful information regarding target detection. In Fig. A.2, the arrows are produced by a suboptimal uninformed digger with slopes that considerably underestimate those that would be provided by a diligent digger.

Bibliography

[1] Nail K. Bakirov, Maria L. Rizzo, and Gbor J. Szkely. A multivariate nonparametric test of independence. *J. Multivar. Anal.*, 97:1742–1756, 2006.

[2] Noel A. C. Cressie and Christopher K. Wikle. *Statistics for spatio-temporal data*. Wiley series in probability and statistics. Wiley, Hoboken, N.J, 2011.

[3] R. Paul Rosenbaum. An exact distribution-free test comparing two multivariate distributions based on adjacency. *J. R. Statist. Soc. B*, 67(4):515–530, 2005.

[4] Ralph C. Smith. *Uncertainty quantification: theory, implementation, and applications*. Computational science and engineering series. Society for Industrial and Applied Mathematics, Philadelphia, 2013.

[5] Rustam Stolkin and Ionut Florescu. Probability of Detection and Optimal Sensor Placement for Threshold Based Detection Systems. *IEEE Sensors Journal*, 9(1):57–60, January 2009.

[6] J. Gabor Szekely, L. Maria Rizzo, and K. Nail Bakirov. Measuring and testing dependence by correlation of distances. *Ann. Statist.*, 35(6):2769–2794, 2007.