Underwater Minefield Detection in Clutter Data Using Spatial Point-Process Models

Darshan Bryner, Fred Huffer, Anuj Srivastava, *Senior Member, IEEE*, and J. Derek Tucker, *Member, IEEE*

***Abstract*—In this paper, we study the problem of detection of underwater minefields amidst dense clutter as that of statistical inference under a spatial point-process model. Specifically, we model the locations (mine and clutter) as samples of a Thomas point process with parent locations representing mines and children representing clutter. Accordingly, the parents are distributed according to a homogeneous Poisson process and, given the parent locations, the children are distributed as independent Poisson processes with intensity functions that are Gaussian densities centered at the parents. This provides a likelihood function for parent locations given the observed clutter (children). Under this model, we develop a framework for penalized maximum-likelihood (ML) estimation of model parameters and parent locations. The optimization is performed using a combination of analytical and Monte Carlo methods; the Monte Carlo part relies on a birth–death–move procedure for adding/removing points in the parent set. This framework is illustrated using both simulated and real data sets, the latter obtained courtesy of Naval Surface Warfare Center Panama City Division (NSWC-PCD), Panama City, FL, USA. The results, evaluated both qualitatively and quantitatively, underscore success in estimating parent locations and other parameters, at a reasonable computation cost.**

***Index Terms*—Maximum-likelihood estimation, simulated annealing, spatial point process, synthetic aperture sonar, Thomas process.**

I. Introduction

UNDERWATER mines are unspent, explosive devices that have been left unattended on the ocean floor from past wars and other related efforts. They can be potentially catastrophic for passenger and merchant ships, and other marine vehicles, that have to navigate through mined terrains. The problem of detecting and neutralizing underwater mines is very important from many perspectives, including transport, commerce, nature preservation, and national defense. The main sensor for detecting underwater mines is sidescan sonar, a (sound-based) imaging device that scans the ocean floor and generates image maps of the observed terrain. A sonar device emits sound waves in different audio frequencies and measures the waves reflected/scattered by different objects in an observation space; the strengths of these returns form pixel values in resulting images. Scientists and engineers study patterns of pixels in these images to detect the presence of mines amidst a tremendous amount of natural and artificial debris that is littered across the ocean floor. The approaches/algorithms for detecting mine occurrences using image pixels are generically termed as automated target recognition (ATR). A large number of ATR procedures have been proposed over the years, relying on a variety of ideas ranging from signal processing, machine learning, and statistical modeling (see [1] for description of the state of the art). The general goal of ATR algorithms is to detect, recognize, and help neutralize mines using sonar images. However, despite significant progress, the ATR performance remains mediocre in general conditions and one requires additional ideas to achieve further improvements in mine detection performance.

The biggest challenge in mine detection comes from mine-like objects that are present in imaged areas but are not mines. These include artificial debris (bottles, boxes, fish traps, etc.) and natural objects (fish, rock, coral, etc.) that have appearances (pixel values, object size, and pixel patterns) similar to mines in sonar images. Since ATR algorithms rely on pixel patterns to perform mine detection and classification, this often leads to an ATR algorithm generating a large number of false detections over the search space, and it becomes difficult to distinguish mines from these false detections, also termed "clutter." Since we are interested in detecting minefields, and not just isolated mines, we can exploit patterns formed by locations of mines in a minefield. Our goal is to model the occurrences of mines and clutter using spatial point processes and use these larger spatial patterns in the data to help separate mines from clutter. This approach is further justified by the fact that the spatial extent of most mines in sonar images is typically a few pixels—ranging from tens to hundreds—but not more, due to the size of mines relative to the bandwidths used in synthetic aperture sonar (SAS) imaging. Thus, the limited spatial extent of mines allows us to treat them as individual points in the observed spatial domain, and the focus shifts from appearance-based pixel patterns to location-based spatial patterns. We can also assume the availability of an ATR score, a real-number associated with each point, that quantifies the confidence an ATR algorithm has about presence of a mine at that point. In this paper, however, we threshold the observed points according to their ATR scores and ignore the ATR score of the selected points

Manuscript received September 19, 2014; revised June 24, 2015; accepted October 09, 2015. This work was supported by the U.S. Office of Naval Research (ONR). This work is approved for public release; distribution is unlimited.

Associate Editor: T. Cobb.

D. Bryner is with the Naval Surface Warfare Center Panama City Division (NSWC-PCD), Panama City, FL 32407-7001 USA (e-mail: darshan.bryner@navy.mil).

F. Huffer and A. Srivastava are with the Department of Statistics, Florida State University, Tallahassee, FL 32306 USA (e-mail: huffer@stat.fsu.edu; anuj@stat.fsu.edu).

J. D. Tucker is with Sandia National Laboratories, Albuquerque, NM 87015 USA (e-mail: jdtuck@sandia.gov).

Digital Object Identifier 10.1109/JOE.2015.2493598

while focusing only on modeling their positions. This thresholding can be treated as an initial screening to reduce clutter in the data which is then analyzed using statistical techniques. We remark that this setting is quite different from the problem of detecting isolated, individual mines where the spatial patterns are not available for help in mine detection.

The next question is the choice of point process model for this problem. While there are a variety of choices available for modeling mine location patterns, the use of cluster processes is natural here. This is motivated in part by the observation that clutter points are often attracted toward mines, and these points form clusters rather than being homogeneously scattered over the spatial domain. Also, it is observed that the individual mines themselves do not cluster together but are generally positioned away from each other. This can be reasoned from the perspective of an enemy minelaying process where the mines are uniformly spread along the minelayer trajectory, rather than being concentrated in one small area, for maximal coverage of the shipping channel.

Clusters arise in the data for two reasons. First, since the ATR system scans the seafloor in straight line paths of a uniform spacing that is less than the range of the onboard sonar, it obtains multiple looks at the same location. Each time the sonar images a detectable target, the ATR algorithm will assign a contact to that target regardless of any previous duplicate detections. Due to slight global positioning system (GPS) location errors, a multiply detected target will be represented by several points clustered around the same location with some given variance. A second source of clustering in the data arises from curious marine life investigating the targets in the test field. Oftentimes the ATR algorithm fails to distinguish the acoustic returns of fish, rays, sharks, etc., from those of actual targets hence contributing to the clustering of data points about the true target location.

Based on these considerations, we choose a specific cluster process—the Thomas process—as a model for the mines and clutter. In the Thomas process, the mine locations are viewed as observations of a parent process and the clutter locations (conditional on the mine locations) as observations of a child process. Here we do not emphasize the statistical model underlying the parent process, and regard the mine locations simply as parameters to be estimated; we work mainly with the conditional distribution of the clutter given the mine locations, which forms our likelihood function. There are a number of sources that cover mathematical formulation and certain statistical analyses associated with a Thomas process: [2, Ch. 5], [3]–[5], [6, Ch. 5], [7, Ch. 3], [8], and [9], and we refer readers to those sources for background material.

A number of papers in recent years have treated the problem of mine detection, albeit in the context of landmines imaged aerially, using tools from spatial point processes. For example, Walsh and Raftery [10] study the problem of ascertaining whether mines are present in a certain point-location data. They model the mine locations as a Strauss process and the background clutter as a Poisson process and develop a hypothesis test based on the ratio of Bayes factors. A similar problem is studied in [11] using hierarchical models. Lake [12] develops several sample statistics, such as the VC statistics (named after Viktorova and Chistyakov; see references within [12]), for performing mine detection. In contrast, we focus on the problem where the mines are known to be present and our goal is to detect mine locations in the presence of clutter. Our model is similar to [10] except we choose to work with the Thomas process since it allows the clutter (children) to be clustered. Priebe *et al.* [13] use the scan statistic, and its computation using importance sampling, to test the hypothesis of homogeneity versus inhomogeneity for a certain region. In this context, inhomogeneity implies presence of mines. Byers and Raftery [14] take a different approach and address the problem of clutter by removing it. They use the K-nearest neighbor distance as a statistic to identify clutter points and remove them from the data. Walsh and Raftery [15] take yet another approach where they assume that mines are laid out in parallel tracks, amidst uniform clutter, such that the distances between mines are Gaussian and the angles made by segments connecting mines are nearly equal. A similar notion of collinearity is also used in [16] in detecting mines. Agarwal *et al.* [17] use tools from clustering, in the form of marked point processes, to decide between mines and clutter. Trang *et al.* [18], [19] advocate the use of shape and spectral features, as scores for computing "mineness" of detected points, in the form of marked point processes for mine detection.

Once we have chosen a model, we pose the overall inference problem—given observed points (treated as children), estimate the location of parents—using an appropriate tool. While it is possible that insights into the minelaying process might provide a suitable prior distribution for mine locations and, thus, model parameters, we pursue a frequentist approach in this paper and seek a maximum-likelihood (ML) solution. The basic ML formulation by itself is not sufficient as it always favors adding new parent locations to the current estimates. To balance this tendency we add a penalty term to the log likelihood that simply penalizes the number of parent points in a linear fashion. A balance between increasing likelihood and complexity penalty provides the final solution. For solving the estimation problem, we note that the objective function may be easily maximized with respect to some of the model parameters, but is too difficult to maximize analytically for the parent locations. By combining analytic approaches for some parameters with a birth–death algorithm (in the context of Monte Carlo simulated annealing) for the parent locations, we optimize over the full set of unknown parameters.

II. Background: Thomas Process

We provide a basic summary of a Thomas process, a special case of a doubly stochastic or a Cox process, on a bounded subset $W \subset \mathbb{R}^2$. A Cox process is a natural extension of a Poisson process in that its intensity function is sampled from a random field rather than having a fixed function definition. A realization of the intensity function is defined from a so-called "parent" point set C, which is sampled from a Poisson process with intensity $\kappa(u)$. In a Thomas process, the realization of the intensity function, say $\lambda(u)$, is given as a mixture of Gaussians centered at each of the points in C. In this case, each Gaussian has the same amplitude α and isotropic variance $\omega^2 I$. The conditional point process $X|C$ is Poisson with intensity $\lambda(u)$, and a sample point set from this process forms Gaussian-

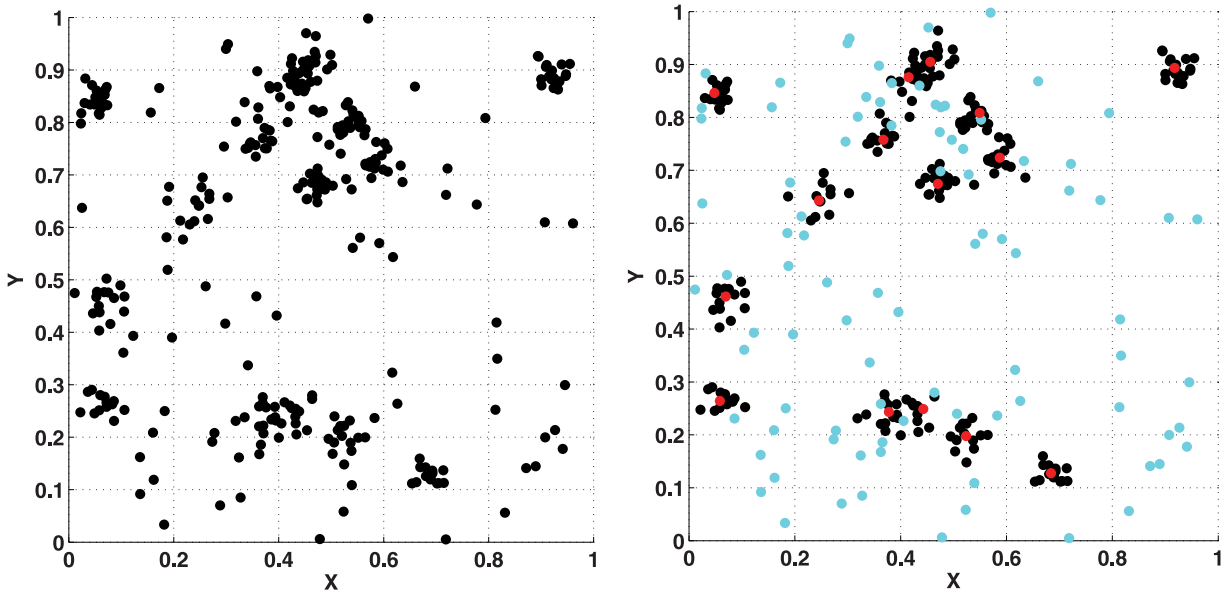

Fig. 1. Simulated data with homogeneous background clutter and homogeneous parent location. (Left) The data X. (Right) Color-coded data—children (black), background clutter (cyan), and unobserved parents (red).

TABLE I
INTENSITY FUNCTIONS USED IN VARIOUS THOMAS PROCESS MODELS

Parent Model	Observed Data Model
Homogeneous Poisson with intensity $\kappa(u) \equiv \kappa$	Without background clutter (Model 1): Poisson with intensity $\lambda(u) = \alpha\varphi_C(u;\omega)$
Inhomogeneous Poisson with intensity $\kappa(u) = \exp(v^T z(u))$	With background clutter (Model 2): Poisson with intensity $\lambda(u) = \alpha\varphi_C(u;\omega) + \eta$

distributed clusters centered at the unobserved parent locations. The Thomas process is therefore defined by the following hierarchy. First, C is sampled from the Poisson process with intensity $\kappa(u)$. Then, X is sampled from the Poisson process with intensity $\lambda(u)$, which depends on the locations C. The following equation shows the hierarchical model that defines a Thomas process:

$$\begin{aligned} C &\sim \text{Poisson}(\kappa(u), W) \\ X|C &\sim \text{Poisson}(\lambda(u), W). \end{aligned} \tag{1}$$

Due to this hierarchy we formulate two models, a parent model and an observed data model, and investigate variations on each of the respective models.

We consider two different parent models: one with homogeneously distributed parent locations and one with an inhomogeneous distribution. In the homogeneous case, the intensity function $\kappa(u)$ is defined simply to equal the constant scalar $\kappa > 0$. In the more complex inhomogeneous case, the intensity function is defined as a function of spatial covariates via $\kappa(u) = \exp(v^T z(u))$, where $z(u) = (1, z_1(u), \ldots, z_K(u))^T$ is the vector of covariate functions evaluated at the point u, and $v = (v_0, v_1, \ldots, v_K)^T$ is the vector of regression coefficients. This choice of $\kappa(u)$ is suggested by the form of a Poisson regression model. These covariate functions are selected beforehand with knowledge of the real life mechanisms by which parents arise and are distributed. For example, in the case of mine locations, these covariate functions could be given by water depth, bottom type, bottom gradient, or formed from an *a priori* estimate of the enemy's most likely mine laying path.

We formulate two models for the observed data. Each model consists of Gaussian distributed clusters centered at the parent locations, the points of which are called "children," however one model additionally incorporates an independent, homogeneous background clutter process. In the simpler case, henceforth referred to as model 1, the intensity function for the observed data $X|C$ is given as $\lambda(u) = \alpha \sum_{j=1}^{n(C)} k(u - c_j; \omega)$, where $\alpha, \omega > 0$, and $k(\cdot - c_j; \omega)$ represents the Gaussian centered at c_j with covariance $\omega^2 I$. This is a special case of a mixture of Gaussians model. In the case of the presence of homogeneous background clutter, henceforth referred to as model 2, the observed data are the union of two independent processes: the Gaussian mixture process and the background clutter process. Let $\eta > 0$ be the homogeneous rate of background clutter. Since independent Poisson processes are additive, the intensity function for this case is given as $\lambda(u) = \alpha \sum_{j=1}^{n(C)} k(u - c_j; \omega) + \eta$. From now on we let $\varphi_C(u;\omega) = \sum_{j=1}^{n(C)} k(u - c_j; \omega)$. Table I summarizes the intensity function definitions that arise from the above scenarios.

Fig. 1 shows a simulation of the case of a homogeneous parent model combined with an observed data model with background clutter. The parameter values for the example in Fig. 1 are given as $\kappa = \alpha = 15$, $\omega = 0.02$, and $\eta = 90$ on $W = [0,1] \times [0,1]$. In this particular realization the number of cluster centers $n(C)$ equals 15 by coincidence. The left plot shows the observed data as black points. The right plot shows the structure of the data by leaving the children as black points and changing the background clutter to cyan points while additionally showing the unobserved parents as red points.

III. MAXIMUM-LIKELIHOOD ESTIMATION

There are a number of approaches to inference for Thomas processes and related cluster processes. Quick estimates of the

parameters (κ, α, ω) may be obtained by minimum contrast estimation (see [2, Ch. 10] and [6, Ch. 6]), which may be used in conjunction with estimating equations for inhomogeneous Thomas processes (see [4] and [5]). Computationally intensive Monte Carlo Markov chain (MCMC) approaches exist for obtaining ML estimates or Bayes estimates, with the Bayesian techniques also leading to natural estimates of the cluster centers C via the posterior distribution for C (see [20]–[22]). Once given estimates of (κ, α, ω), estimates of the cluster centers may always be obtained by using general Metropolis–Hastings schemes to simulate from the conditional distribution of C given X (see [2, Ch. 7]). In our work, we separate the Thomas process parameter estimation into two steps by first estimating the parameters involved in the conditional process $X|C$, i.e., the observed data model, and then estimating those that belong to the parent process.

In the first step of estimation, we view the observed data model as being essentially a Gaussian mixture plus background clutter process parametrized by $(C, \alpha, \omega, \eta)$. This differs from other methods in that we perform inference on this conditional process and treat C as a model parameter rather than a random variable. In other words, with regards to C, we assume no model in this step, and we are only concerned with estimating the point locations. Since C does not have a fixed dimension, we use a penalized ML approach to prevent overfitting in our parameter estimation. In Gaussian mixture models, parameters are often estimated by performing a series of expectation–maximization (EM)-type algorithms, varying the number of mixture components, and then using a complexity penalty to choose among the different solutions (e.g., in [23]). Other approaches are Bayesian in nature and assume some sort of prior distribution in their models. The simulated annealing algorithm we propose supplies an alternative approach that simultaneously estimates the number of clusters along with their locations (and the other parameters in our model) without the use of a prior. Our strategy differs further from these other approaches in that we do not cluster points—that is, we do not assign clutter points to specific clusters—but rather we focus only on estimating the cluster centers along with the other model parameters. An additional feature that we include beyond these other techniques is the ability to estimate the rate of background clutter not belonging to any particular cluster. This simulated annealing approach obtains estimates of $(C, \alpha, \omega, \eta)$ using the penalized ML estimation (MLE) described in Section III-A with the detailed algorithm given in Section IV.

In the second step of estimation, we gain at least a rough idea of the values of the parent model parameters by treating the estimate of C from the previous step as exact and using these data to compute ML estimates of the parameters. In the homogeneous case, we need only to estimate the scalar parameter κ, and in the inhomogeneous case, we estimate the regression parameter vector v. Here, we use a relatively straightforward MLE technique, and Section III-C details our approach. This approach is an alternative to the much more computationally intensive fully Bayesian approach, which computes a posterior distribution of the coefficient vector v.

A. Penalized ML for Observed Data Model

To set up the MLE, we first consider the density function of the conditional process $X|C$, which is given by

$$f(X|C,\theta) = \exp\left\{|W| - \int_W \lambda(u)du\right\} \prod_{i=1}^{n(X)} \lambda(x_i) \qquad (2)$$

where θ is the vector of model parameters and $X = (x_1, x_2, \ldots, x_{n(X)})$. Let g be a function obtained by dividing the likelihood function f [see (2)] by the constant term $e^{|W|}$. A maximization of the function

$$\log g(X|C,\theta) = \sum_{i=1}^{n(X)} \log(\lambda(x_i)) - \int_W \lambda(u)du \qquad (3)$$

is thus equivalent to a maximization of f. Note that the ML fit of (3) is given by $n(X)$ delta functions centered at each point x_i of the data X. That is, if $C = X$, $\alpha = 1$, $\eta = 0$, and $\omega \to 0$, then $\log g(X|C,\theta) \to \infty$. To prevent this overfitting, we introduce a penalty term which regulates the number of cluster centers (parents); we instead maximize the penalized log-likelihood function given by

$$h_\beta(X|C,\theta) = \log g(X|C,\theta) - \beta\left(n(C) + \frac{\dim(\theta)}{2}\right). \qquad (4)$$

That is, we solve

$$(\hat{C}, \hat{\theta}) = \underset{C\in\mathbf{C},\theta\in\Theta}{\operatorname{argmax}} \{h_\beta(X|C,\theta)\}$$

where $\beta \geq 0$ is a user-specified penalty, $\mathbf{C}$ is the set of all finite point sets on W, and Θ is the appropriate model parameter space. Noting that each additional cluster center introduces two more parameters, we see that the value $\beta = 2$ corresponds to the well-known Akaike information criterion (AIC) penalty [24], and $\beta = \log(n(X))$ corresponds to the Schwarz Bayesian criterion (SBC) or Bayesian information criterion (BIC) penalty [25]. Setting $\beta = 0$ represents no penalty. Using AIC or SBC penalty terms is a common approach to restricting the number of parameters in a model.

B. Optimization Details

To reduce the computational complexity of the numerical optimization, we reparametrize the intensity function and then find a closed-form expression for the maximizer of one of the parameters (fixing the others). This technique reduces the dimensionality by 1 and thus speeds up the computation time. Additionally, in this reparameterization, we eliminate the positivity constraint on the components of θ so that we can perform an unconstrained optimization, thus further reducing the complexity. Define $\alpha = e^\phi$, $\omega = e^\gamma$, $\eta = e^\rho$, and $\psi = \eta/\alpha = e^{\rho-\phi}$. The reparameterized intensity function is given as $\lambda(u) = e^\phi(\varphi_C(u; e^\gamma) + \psi)$. Now

$$\log g(X|C,\theta) = n(X)\phi + \sum_{i=1}^{n(X)} \log(\varphi_C(x_i; e^\gamma) + \psi) - e^\phi \int_W (\varphi_C(u; e^\gamma) + \psi)du.$$

Setting ψ, γ, and C constant and maximizing over ϕ, we come to the expression

$$\hat{\phi} = \log\left(\frac{n(X)}{\int_W(\varphi_C(u;e^{\gamma})+\psi)du}\right).$$

After substituting this expression back into the expression for $\log g$ and setting $\theta = (\gamma, \psi)$, we can rewrite $\log g$ as

$$\begin{aligned}\log g(X|C,\theta) = &\sum_{i=1}^{n(X)} \log(\varphi_C(x_i;e^{\gamma})+\psi) \\ &- n(X)\log\left(\int_W(\varphi_C(u;e^{\gamma})+\psi)du\right) \\ &+ n(X)(\log(n(X))-1). \end{aligned} \tag{5}$$

Note that throughout, we refer to θ as a general set of parameters to be estimated, the precise definition of which can change. Here, we have shifted to the assignment $\theta = (\gamma, \psi)$, whereas previously, the assignment was $\theta = (\alpha, \omega, \eta)$. Now we perform the optimization over $\theta \in \mathbb{R}^2$, i.e., we solve for

$$(\hat{C}, \hat{\theta}) = \underset{C\in\mathbf{C},\theta\in\mathbb{R}^2}{\operatorname{argmax}} \{h_\beta(X|C,\theta)\}$$

where h_β is defined from the version of $\log g$ given in (5) but with the constant term $n(X)(\log(n(X))-1)$ removed. To complete the optimization, we obtain the estimated values of our original parameters by computing $\hat{\alpha} = n(X)/\int_W(\varphi_C(u;e^{\hat{\gamma}})+\hat{\psi})du$, $\hat{\omega} = e^{\hat{\gamma}}$, and $\hat{\eta} = \hat{\alpha}\hat{\psi}$.

C. ML for Parent Model

The first estimation step, given in Section III-B, does not perform any inference on the model from which the parents arise; rather, it only provides an estimate of the point locations. This section focuses on the second estimation step, which provides a parametric estimation of the parent process intensity function by considering $\hat{C}$ from the first estimation step as exact, both in number and in location. The homogeneous case is straightforward, and the MLE is given as $\hat{\kappa} = n(\hat{C})/|W|$. The estimation for the inhomogeneous case is as follows. Conditional on $n(C)$, the true centers $C = \{c_1, c_2, \ldots, c_{n(C)}\}$ are independent identically distributed (i.i.d.) from the intensity $\kappa(u)$ normalized to be a density via

$$f(u|v) = \frac{\kappa(u)}{\int_W \kappa(s)ds} = \frac{e^{v^T z(u)}}{\int_W e^{v^T z(s)}ds}.$$

Thus, the conditional log-likelihood function is given by

$$\begin{aligned}\log f(C|v) &= \log\left(\prod_{j=1}^{n(C)} f(c_j|v)\right) \\ &= \sum_{j=1}^{n(C)} v^T z(c_j) - n(C)\log\left(\int_W e^{v^T z(s)}ds\right).\end{aligned}$$

Again, by considering $\hat{C}$ to be exact, we compute the MLE of the regression parameters as

$$\hat{v} = \operatorname{argmax}_{v\in\mathbb{R}^{K+1}}\left\{\log f(\hat{C}|v)\right\}.$$

Unlike the first estimation step, the parameters here have fixed dimension, and thus we compute this optimization using standard numerical tools available in MATLAB (the *fminsearch* or *fminunc* function).

IV. Optimization Using Simulated Annealing

Simulated annealing is a procedure designed to find an approximation of the global optimum of a given function. The process is inspired by that of annealing in metallurgy, whereby a metal would be heated to its liquid state, then cooled slowly enough so that its individual particles could arrange themselves in a ground state with optimally low energy as it reformed back to a solid. If the temperature were not lowered slowly enough, then the metal would be frozen in a metastable state rather than into the ideal ground state. Simulated annealing for function optimization is an analogous process that implements a series of Metropolis algorithms evaluated at slowly decreasing values of the control parameter—usually referred to as the temperature. The procedure is typically designed to converge to the energy minimizing state, or the argmin of the objective function (see [26] for more information on simulated annealing). Here, we set the energy function to be $E = -h_\beta$ from (4), the negative penalized log-likelihood function, and thus, minimizing the energy is equivalent to maximizing the penalized likelihood.

One iteration of simulated annealing, i.e., the Metropolis algorithm, is as follows. Given the current state s with energy $E(s)$, compute the energy $E(s')$ of a perturbed state s'. Accept the perturbed state with probability 1 if $E(s') - E(s) \leq 0$ and with probability $e^{-(E(s')-E(s))/T}$ otherwise, where T is the current temperature. Initially, when the temperature is large, the procedure is likely to accept more states, irrespective of whether they increase or decrease energy, and as the temperature is decreased, the procedure becomes more selective toward states that decrease energy. This design allows for a broad search area and, with an appropriate cooling schedule, is guaranteed to converge to the global minimum given infinite time. This is in contrast to, say, a gradient descent method, which can only lower the energy at each iteration and consequently is only guaranteed to converge to a local minimum.

Before we present the simulated annealing algorithm to estimate the parameters of the Thomas process with background clutter, we must first define a mechanism to obtain a perturbation of the current state to act as a candidate state. Here, we use a birth–death–move algorithm to perturb the point set C. A birth is defined as $C \mapsto C \cup \xi$, where $\xi \sim q_b$, a given density function on W. A death is defined as $C \mapsto C\backslash c_j$, where $j \sim q_d$ is a given discrete density on the index set of C. A move is defined as $C \mapsto (C\backslash c_j) \cup \zeta$, where $j \sim q_d$ and $\zeta \sim q_m(\cdot|c_j)$, a given density function on W that depends on the selection of the move point c_j. A new point set C' is generated from C via the birth–death–move subroutine presented in Algorithm 1. It is

important to note that one can go from any point configuration to any other using finite combinations of these perturbations.

Algorithm 1: Birth–Death–Move

1: Given C, q_b, q_d, and q_m
2: Generate $u \sim$ Uniform$[0,1]$.
3: **if** $0 \leq u < 1/3$ **then**
4: Perform a birth, i.e., sample $\xi \sim q_b$ and let $C' = C \cup \xi$.
5: **else if** $1/3 \leq u < 2/3$ **then**
6: Perform a death, i.e., sample $j \sim q_d$ and let $C' = C \backslash c_j$.
7: **else**
8: Perform a move, i.e., sample $j \sim q_d$, $\zeta \sim q_m(\cdot|c_j)$, and let $C' = (C \backslash c_j) \cup \zeta$.
9: **end if**

In our implementation, we use the following densities: $q_b = \hat{\lambda} / \int_W \hat{\lambda}$, where $\hat{\lambda}$ is a nonparametric estimate of the intensity function of the conditional process $X|C$ (see [2, Sec. 4.3.1]); q_d is equal to the discrete uniform density on $\{1, \ldots, n(C)\}$; and $q_m(\cdot|c_j)$ is set to equal $\hat{\lambda}$ within $B(r, c_j) \cap W$, equal to zero elsewhere in W, and then normalized. We use the notation $B(r, c_j)$ to refer to the ball of radius r centered at c_j. Here, r is a parameter that dictates the extent to which a point c_j can be moved in the birth–death–move algorithm. The full simulated annealing algorithm is presented in Algorithm 2.

Algorithm 2: Simulated Annealing Algorithm

1: Given the data X on a domain W, a penalty $\beta \geq 0$, a constant temperature decrease rate $0 < \rho < 1$, and the initial values $C^{(0)}$, $\theta^{(0)}$, and $T^{(0)}$,
2: Compute the energy $E^{(0)} = -h_\beta(X|C^{(0)}, \theta^{(0)})$ [see (4)].
3: **for** $m = 1 : M$ **do**
4: Generate the candidate C' from the birth–death–move subroutine (Algorithm 1).
5: Compute the candidate $\theta' = \operatorname{argmin}_{\theta \in \mathbb{R}^2}\{-h_\beta(X|C', \theta)\}$ via standard numerical techniques (the *fminunc* function in MATLAB).
6: Set $E' = -h_\beta(X|C', \theta')$.
7: Set the temperature $T^{(m)} = \rho T^{(m-1)}$.
8: Generate $u \sim$ Uniform$[0,1]$.
9: **if** $u < \min\{\exp(-(E' - E^{(m-1)})/T^{(m)}), 1\}$ **then**
10: Accept proposals and set $C^{(m)} = C'$, $\theta^{(m)} = \theta'$, and $E^{(m)} = E'$.
11: **else**
12: Reject proposals and set $C^{(m)} = C^{(m-1)}$, $\theta^{(m)} = \theta^{(m-1)}$, and $E^{(m)} = E^{(m-1)}$.
13: **end if**
14: **end for**

V. Simulation Experiments

In this section, we present results obtained from Algorithm 2 on several data sets, including simulated and real mine-detection data sets. Although the algorithm returns the value $\hat{\theta} = (\hat{\gamma}, \hat{\psi})$, we show the results as the estimated values of our original parameter definitions $\hat{\alpha}$, $\hat{\omega}$, and $\hat{\eta}$ (when applicable) to provide a direct comparison to the meaningful parameters used in the simulation. For comparison, we supply parameter estimates obtained by fixing $\hat{C}$ at $\hat{C} = C$ (the true centers) and then numerically maximizing the penalized log-likelihood function over θ. These estimates, which do not require the simulated annealing routine to compute, are labeled as "opt" (for "optimum") and are expected to be close to the global maximizers of the penalized log-likelihood function. By comparing the estimates from our simulated annealing algorithm to these "opt" estimates, we can ensure that the algorithm is achieving something close to a global maximum.

For each data set, we run Algorithm 2 under four scenarios: a) model 1 with AIC penalty; b) model 2 with AIC penalty; c) model 1 with SBC penalty; and d) model 2 with SBC penalty. Recall that model 1 is the observed data model without homogeneous background clutter and that model 2 is the observed data model with homogeneous background clutter (see Table I). Since θ has different dimensionality across models, the inclusion of $\dim(\theta)/2$ in the penalty term within the formula for h_β [see (4)] allows us to directly compare values of h_β obtained from different models under the same penalty value. Therefore, the maximum values of h_β obtained from Algorithm 2 under scenarios a) and b) are comparable, as well as those of c) and d), and the scenario that produces the higher h_β value of the two represents the better model fit.

To compare results across penalty values, we use a more informal inspection of the estimated number of cluster centers versus the true number of centers. Since our problem is that of minefield detection, the number of cluster centers, or mines, detected is the most important output of our algorithm. Typically, in model selection, restricting the number of parameters according to the AIC penalty is more likely to overfit the data with more model parameters; whereas, restricting according to the stronger SBC penalty is less likely to overfit the data. That is, under the AIC penalty Algorithm 2 should typically yield $n(\hat{C})$ greater than $n(C)$, and under the SBC penalty it should typically yield $n(\hat{C})$ approximately equal to or slightly less than $n(C)$. In minefield detection, it is generally more desirable to overestimate rather than to underestimate the number of mines because the consequences of a false alarm are less grave than that of a missed detection. Therefore, the AIC penalty on average should be the more desirable one, provided that it does not overfit the data too much, yielding too many false alarms.

In all of the following computations, we use the annealing temperature schedule determined by the parameters $T_0 = 15$, $\rho = 0.996$, and $M = 2000$. Also, the domain for each case has been set to $W = [0,1] \times [0,1]$, and the move parameter r used in the density q_m within Algorithm 1 has been set to $r = 0.1$. Computation time, of course, varies with $n(X)$, $n(C)$, and the temperature schedule, but for the following experiments the average computation time for Algorithm 2 under model 1 is about 30 min, and the average computation time for Algorithm 2 under model 2 is about 1 h. We ran our program in MATLAB on a laptop with 2.53 GHz.

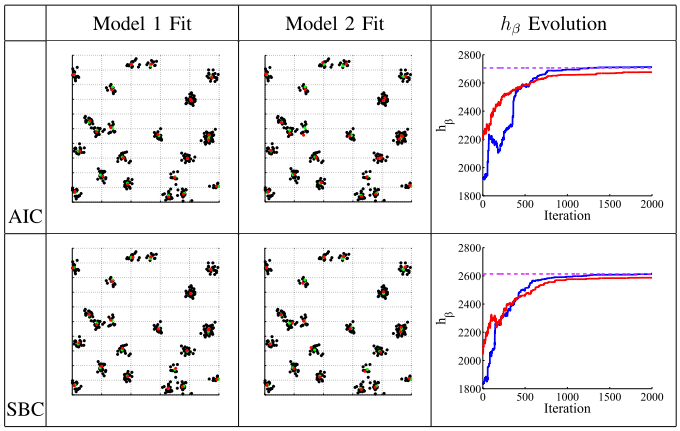

Fig. 2. Results of Algorithm 2 on data set 1 simulated from case 1: homogeneous parents, and no background clutter (model 1).

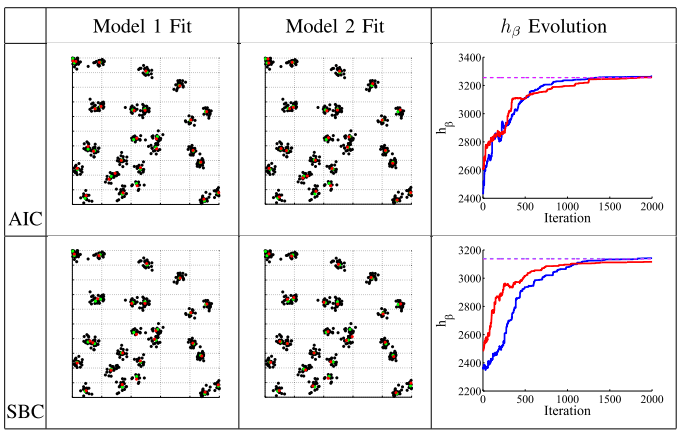

Fig. 3. Results of Algorithm 2 on data set 2 simulated from case 1: homogeneous parents, and no background clutter (model 1).

A. *Simulated Data*

To demonstrate the effectiveness of our algorithm in estimating model parameters and selecting an observed data model, we run Algorithm 2 under the four scenarios outlined above on a total of six data sets simulated from various model assumptions, or cases. In the first case, we generate two data sets from the homogeneous parent model and the observed data model without background clutter. In the second case, we generate two data sets from the homogeneous parent model and the observed data model with background clutter. In the third case, we generate two data sets from the inhomogeneous parent model with covariates of our selection and the observed data model with background clutter. Therefore, in the first case, the observed data is simulated from model 1, and in the second and third cases, it is simulated from model 2. Only in the third case do we proceed to carry out the second stage of estimation, that of the inhomogeneous parent intensity parameters v. The covariate functions we use in this case are simply $z_1(u) = x$ and $z_2(u) = y$, and we choose to not include the constant scalar term in the model, setting $v_0 = 0$ in our simulations and estimating only v_1 and v_2. The number of points $n(X)$ for each data set ranges between 218 and 475.

For each simulated data set, we carry out the following experiment. For each scenario a)–d), run Algorithm 2 a total of ten times, each with different random initializations of $n(C_0) = 20$ cluster center locations. Compute the mean and standard deviation of each set of ten final estimates of α, ω, η (when applicable), $n(C)$, and v (when applicable), as well as the set of ten final values of h_β. Select the single run out of ten that yielded the highest final value of h_β, the so-called "best" run, and report the final values of α, ω, η (when applicable), $n(C)$, v (when applicable), and h_β for this run. Also compute and report the "opt" and "true" values of each of these terms, where the "true" value of h_β is equal to the function evaluated at the true parameter values and the true cluster centers.

Tables II–VII and Figs. 2–7 show results from this experiment on each of the six respective simulated data sets for scenarios a)–d). Each figure consists of the following six plots: four

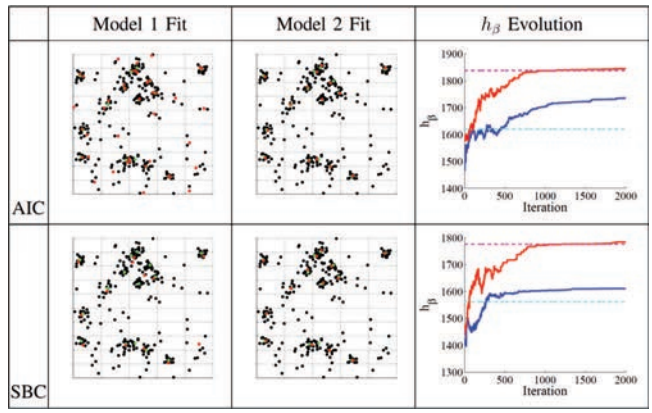

Fig. 4. Results of Algorithm 2 on data set 1 simulated from case 2: homogeneous parents, and with background clutter (model 2).

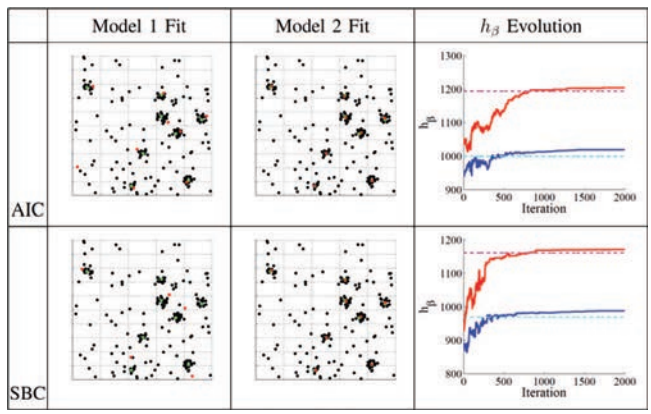

Fig. 5. Results of Algorithm 2 on data set 2 simulated from case 2: homogeneous parents, and with background clutter (model 2).

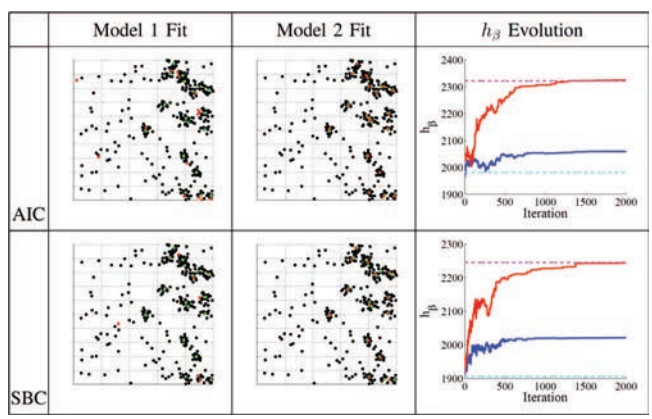

Fig. 6. Results of Algorithm 2 on data set 1 simulated from case 3: inhomogeneous parents, and with background clutter (model 2).

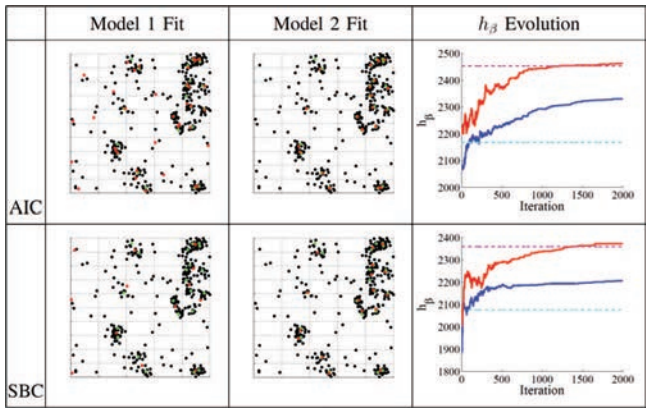

Fig. 7. Results of Algorithm 2 on data set 2 simulated from case 3: inhomogeneous parents, and with background clutter (model 2).

TABLE II
RESULTS OF ALGORITHM 2 ON DATA SET 1 SIMULATED FROM CASE 1: HOMOGENEOUS PARENTS, AND NO BACKGROUND CLUTTER (MODEL 1)

		(a) AIC, model 1	(b) AIC, model 2	(c) SBC, model 1	(d) SBC, model 2
$\hat{\alpha}$	Mean	16.22	16.62	18.54	18.57
	Std	1.898	1.360	0.3580	0.1312
	Best	17.74	15.83	18.71	18.51
	Opt	18.68	18.68	18.68	18.68
	True	20.00	20.00	20.00	20.00
$\hat{\omega}$	Mean	0.01969	0.02045	0.02036	0.02054
	Std	5.998×10^{-4}	5.931×10^{-4}	2.605×10^{-4}	6.701×10^{-4}
	Best	0.01980	0.02034	0.02028	0.02041
	Opt	0.02018	0.02018	0.02018	0.02018
	True	0.02000	0.02000	0.02000	0.02000
$\hat{\eta}$	Mean		6.606		5.721
	Std		6.552		5.822
	Best	N/A	2.638	N/A	2.513
	Opt		6.094×10^{-5}		6.094×10^{-5}
	True		0		0
$n(\hat{C})$	Mean	25.60	24.60	22.10	21.70
	Std	3.340	2.675	0.3162	0.4831
	Best	23	26	22	22
	Opt	22	22	22	22
	True	22	22	22	22
h_β	Mean	2704.5	2661.5	2607.3	2574.2
	Std	4.6312	7.9750	6.4461	6.9802
	Best	2712.4	2676.0	2615.3	2586.5
	Opt	2706.8	2705.8	2615.4	2612.4
	True	2705.8	2704.8	2614.4	2611.4

plots corresponding to each scenario a)–d) of the estimated centers $\hat{C}$ (red) overlayed on the data X (black) and the true centers C (green), one plot showing the evolution of h_β in Algorithm 2 using the AIC penalty [scenarios a) and b)], and one plot showing the evolution of h_β in Algorithm 2 using the SBC penalty [scenarios c) and d)]. In each of these h_β evolution plots, the result obtained from model 1 is given by the solid blue curve, and the result from model 2 is given by the solid red curve. The dashed cyan line represents the "opt" estimate of h_β under model 1, and the dashed magenta line represents the "opt" estimate of h_β under model 2. Again, for each penalty value, the solid curve with the higher value of h_β after $M = 2000$ iterations represents the better model fit.

TABLE III
RESULTS OF ALGORITHM 2 ON DATA SET 2 SIMULATED FROM CASE 1: HOMOGENEOUS PARENTS, AND NO BACKGROUND CLUTTER (MODEL 1)

		(a) AIC, model 1	(b) AIC, model 2	(c) SBC, model 1	(d) SBC, model 2
$\hat{\alpha}$	Mean	17.85	17.72	19.45	19.95
	Std	0.5805	0.9223	0.3356	0.6089
	Best	17.83	17.76	19.69	18.80
	Opt	18.70	18.65	18.70	18.65
	True	20.00	20.00	20.00	20.00
$\hat{\omega}$	Mean	0.02027	0.02098	0.02109	0.02200
	Std	3.937×10^{-4}	7.361×10^{-4}	3.703×10^{-4}	0.001618
	Best	0.01994	0.02015	0.02099	0.02079
	Opt	0.02057	0.02038	0.02057	0.02038
	True	0.02000	0.02000	0.02000	0.02000
$\hat{\eta}$	Mean		1.252		4.874
	Std		0.8352		6.023
	Best	N/A	1.098	N/A	1.505
	Opt		1.192		1.192
	True		0		0
$n(\hat{C})$	Mean	27.90	28.20	25.40	24.50
	Std	0.8756	1.549	0.5164	0.7071
	Best	28	28	25	26
	Opt	27	27	27	27
	True	27	27	27	27
h_β	Mean	3252.3	3223.7	3129.7	3088.1
	Std	8.0193	21.035	8.4190	38.505
	Best	3265.2	3257.1	3142.0	3116.2
	Opt	3256.2	3255.4	3139.6	3136.7
	True	3254.4	3253.4	3137.8	3134.7

The general trends of the results shown in these tables and figures are as follows. For all data sets in all cases, the value of h_β output from Algorithm 2, under either penalty, is shown to be higher when fitting the observed data model from which the data were simulated. That is, our algorithm selects the correct observed data model (model 1 or model 2) for every data set regardless of the selected penalty value. Under the correct model, for both penalty values, the parameter estimates are, on average, quite close to the "opt" estimates, and as expected, results with the AIC penalty tend to overestimate the number of centers slightly while the SBC results tend to underestimate slightly. In the context of minefield detection, this result favors the use of the AIC penalty; however, these favorable AIC results do come with an important caveat. Under the AIC penalty, Algorithm 2 yields less consistent parameter estimates than that of the SBC penalty, as seen by the higher standard errors. The reason for this is because, with the weaker AIC penalty, it is more likely to see outliers yielding an unusually high number of cluster center estimates, while under the stronger SBC penalty we do not see such outliers. Moreover, under the SBC penalty it is less likely to see outliers yielding an unusually low number of cluster center estimates since the penalty is not too strong. Therefore, the tradeoff is consistency for desirable mean behavior. Since there is no definitive answer on which penalty to

TABLE IV
RESULTS OF ALGORITHM 2 ON DATA SET 1 SIMULATED FROM CASE 2: HOMOGENEOUS PARENTS, AND WITH BACKGROUND CLUTTER (MODEL 2)

		(a) AIC, model 1	(b) AIC, model 2	(c) SBC, model 1	(d) SBC, model 2
$\hat{\alpha}$	Mean	15.34	15.75	31.58	16.83
	Std	3.553	0.4271	6.090	0.5888
	Best	7.565	15.92	25.83	16.93
	Opt	22.30	15.46	22.30	15.46
	True	15.00	15.00	15.00	15.00
$\hat{\omega}$	Mean	0.04513	0.02231	0.07198	0.02283
	Std	0.007949	3.264×10^{-4}	0.007537	4.499×10^{-4}
	Best	0.03135	0.02192	0.0652	0.02257
	Opt	0.07207	0.02135	0.07207	0.02135
	True	0.02000	0.02000	0.02000	0.02000
$\hat{\eta}$	Mean		73.04		75.29
	Std		1.300		1.855
	Best	N/A	73.24	N/A	75.30
	Opt		80.30		80.30
	True		90		90
$n(\hat{C})$	Mean	22.80	15.20	11.40	14.10
	Std	7.627	0.4216	1.713	0.5677
	Best	43	15	13	14
	Opt	15	15	15	15
	True	15	15	15	15
h_β	Mean	1693.5	1840.4	1606.1	1778.3
	Std	18.890	2.4373	3.7598	3.0737
	Best	1736.4	1844.3	1610.5	1783.6
	Opt	1620.9	1837.6	1561.0	1775.8
	True	-34.270	1836.5	-94.158	1774.7

TABLE V
RESULTS OF ALGORITHM 2 ON DATA SET 2 SIMULATED FROM CASE 2: HOMOGENEOUS PARENTS, AND WITH BACKGROUND CLUTTER (MODEL 2)

		(a) AIC, model 1	(b) AIC, model 2	(c) SBC, model 1	(d) SBC, model 2
$\hat{\alpha}$	Mean	31.01	15.95	70.68	16.12
	Std	5.794	0.5562	23.63	0.05687
	Best	27.96	16.17	56.36	16.08
	Opt	31.11	16.22	31.11	16.22
	True	15.00	15.00	15.00	15.00
$\hat{\omega}$	Mean	0.09689	0.01917	0.1663	0.01923
	Std	0.009824	1.719×10^{-4}	0.02724	1.863×10^{-4}
	Best	0.09062	0.01905	0.1437	0.01899
	Opt	0.1148	0.02010	0.1148	0.02010
	True	0.02000	0.02000	0.02000	0.02000
$\hat{\eta}$	Mean		88.95		89.04
	Std		0.3154		0.4548
	Best	N/A	88.64	N/A	89.35
	Opt		88.22		88.22
	True		90		90
$n(\hat{C})$	Mean	8.600	8.100	4.500	8.000
	Std	1.506	0.3162	0.7071	0
	Best	9	8	5	8
	Opt	8	8	8	8
	True	8	8	8	8
h_β	Mean	1015.9	1201.6	984.53	1169.0
	Std	2.7070	1.8538	3.8459	2.1840
	Best	1018.9	1203.8	987.84	1170.9
	Opt	999.61	1193.3	969.15	1161.2
	True	-2365.7	1193.0	-2396.2	1160.8

select for all situations, we examine the outcome of our estimation on real data sets for which we have ground truth target locations.

B. Real Data

We test our estimation procedure on two data sets of contacts obtained from an automatic target recognition (ATR) system developed by the Naval Surface Warfare Center Panama City Division (NSWC-PCD), Panama City, FL, USA. Acoustic data were collected from an autonomous underwater vehicle (AUV) equipped with a high-resolution, high-frequency, synthetic aperture sidescan sonar. The vehicle traveled in a uniform spaced search pattern to cover the entire test field, which was approximately one nautical square mile and contained 19 targets of interest laying on the seafloor. The raw sonar data were then postprocessed to form a complex-valued image via a k-space or wave number beamformer (see [27, Ch. 6]). The sonar imagery was then fed to an onboard ATR algorithm that detected and output all potential target locations, or contacts. Associated with each contact, the algorithm also output a classification score from 0 to 1 indicating the likelihood of it being a target of interest. We direct the reader to [28] and [29] for more information on the detection and classification procedures within the ATR algorithm. The data in black seen in Fig. 8 represent the contact locations from these two data sets that have a score greater than 0.5, i.e., those contacts with a positive target classification, and the data in green represent the ground truth target locations.

Data sets 1 and 2 differ in the following ways: they were collected from different locations, the target fields were different, and a different sonar system was used in each case. All of these factors contribute to a greater occurrence of background clutter in data set 2 compared to that of data set 1, thus making for an interesting comparison of results from our algorithm. Since we have no meaningful covariate functions z_i available that are attributed to these data sets, and since by inspection the cluster centers appear to have a homogeneous distribution, we only use the homogeneous parent model in our analysis. We compare results for each data set under the four scenarios a)–d) as in our simulation experiments. However, since we do not have true parameter values available to us other than the cluster center locations, we cannot display any true parameter values in Tables VIII and IX in this section. Moreover, we only run Algorithm 2 once for each scenario instead of ten times as in our simulation experiments, and thus, these tables also do not include entries for the mean and standard deviation of parameter estimates. Figs. 9 and 10 are analogous to the figures associated with the simulation experiments presented in Section V-A.

For both real data sets the value of h_β output from Algorithm 2, under either penalty, is shown to be higher when fitting model 2 than when fitting model 1. That is, our algorithm selects the

TABLE VI
RESULTS OF ALGORITHM 2 ON DATA SET 1 SIMULATED FROM CASE 3: INHOMOGENEOUS PARENTS, AND WITH BACKGROUND CLUTTER (MODEL 2)

		(a) AIC, model 1	(b) AIC, model 2	(c) SBC, model 1	(d) SBC, model 2
$\hat{\alpha}$	Mean	63.14	16.13	239.6	19.12
	Std	12.58	1.037	142.9	1.464
	Best	35.53	15.52	95.40	21.15
	Opt	33.92	16.36	33.92	16.36
	True	15.00	15.00	15.00	15.00
$\hat{\omega}$	Mean	0.1526	0.02133	0.2438	0.02339
	Std	0.01671	7.025×10^{-4}	0.07873	9.158×10^{-4}
	Best	0.1144	0.02065	0.1581	0.02435
	Opt	0.2075	0.02123	0.2075	0.02123
	True	0.02000	0.02000	0.02000	0.02000
$\hat{\eta}$	Mean		94.58		94.85
	Std		1.167		1.897
	Best	N/A	95.84	N/A	93.08
	Opt		96.85		96.85
	True		90.00		90.00
$n(\hat{C})$	Mean	10.00	18.50	4.500	15.60
	Std	1.944	1.080	1.581	1.265
	Best	15	19	6	14
	Opt	18	18	18	18
	True	18	18	18	18
$\hat{v}_1$	Mean	4.209	5.177	12.97	5.004
	Std	0.7385	0.1601	10.15	0.1555
	Best	3.058	5.182	3.978	4.797
	Opt	5.243	5.243	5.243	5.243
	True	4.000	4.000	4.000	4.000
$\hat{v}_2$	Mean	0.01239	0.7777	0.1810	0.6834
	Std	0.1199	0.1568	0.5990	0.2719
	Best	-0.08427	0.8970	-0.3189	0.6781
	Opt	0.7142	0.7142	0.7142	0.7142
	True	1.000	1.000	1.000	1.000
h_β	Mean	2055.2	2318.1	2014.8	2237.5
	Std	1.4369	3.6245	5.1244	3.6115
	Best	2059.2	2324.5	2021.2	2243.7
	Opt	1980.9	2321.9	1906.0	2244.9
	True	-6128.9	2319.5	-6203.9	2242.5

TABLE VII
RESULTS OF ALGORITHM 2 ON DATA SET 2 SIMULATED FROM CASE 3: INHOMOGENEOUS PARENTS, AND WITH BACKGROUND CLUTTER (MODEL 2)

		(a) AIC, model 1	(b) AIC, model 2	(c) SBC, model 1	(d) SBC, model 2
$\hat{\alpha}$	Mean	15.19	15.11	42.54	19.78
	Std	5.873	0.7073	3.587	2.410
	Best	10.75	15.15	34.85	18.14
	Opt	21.71	14.68	21.71	14.68
	True	15.00	15.00	15.00	15.00
$\hat{\omega}$	Mean	0.04404	0.02281	0.06919	0.02620
	Std	0.007977	8.067×10^{-4}	0.003399	0.002531
	Best	0.03656	0.02174	0.06103	0.02411
	Opt	0.08180	0.02167	0.08180	0.02167
	True	0.02000	0.02000	0.02000	0.02000
$\hat{\eta}$	Mean		78.09		75.38
	Std		2.762		2.923
	Best	N/A	83.09	N/A	75.94
	Opt		78.69		78.69
	True		90.00		90.00
$n(\hat{C})$	Mean	31.70	21.60	11.60	17.00
	Std	7.072	1.075	1.075	1.633
	Best	40	21	14	18
	Opt	22	22	22	22
	True	22	22	22	22
$\hat{v}_1$	Mean	1.850	3.172	1.346	3.174
	Std	0.2912	0.2138	0.3062	0.1792
	Best	1.564	3.356	0.7244	2.950
	Opt	3.120	3.120	3.120	3.120
	True	4.000	4.000	4.000	4.000
$\hat{v}_2$	Mean	0.6403	0.8870	0.4287	0.8561
	Std	0.1809	0.1421	0.1884	0.1011
	Best	0.5208	0.7933	0.7974	0.8146
	Opt	1.039	1.039	1.039	1.039
	True	1.000	1.000	1.000	1.000
h_β	Mean	2305.1	2453.6	2202.4	2359.3
	Std	17.064	5.796	2.7949	9.6393
	Best	2323.0	2461.9	2206.3	2372.6
	Opt	2167.5	2452.9	2075.8	2359.2
	True	-176.65	2450.4	-268.33	2356.7

observed data model that includes homogeneous background clutter for both data sets, which is sensible for the true nature of the data. Under the AIC penalty, for both models and for both data sets, the simulated annealing procedure yields an estimate for the number of cluster centers that is more than two times the true number of centers; hence, the other remaining model parameter estimates are not close to the "opt" estimates. However, the results from scenario d)—SBC penalty and model 2—are quite accurate, with an estimation very close to the "opt" parameter values in both data sets. In data set 1, the number of estimated centers is only one more than the true number of centers, and in data set 2, the number of estimated centers is exactly the same as the true number. In the context of minefield detection, this result favors the use of the SBC penalty since it does not underfit the data and since the AIC penalty allows for too many false alarms.

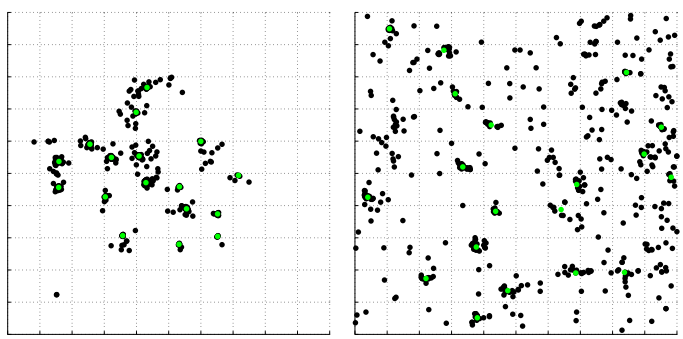

Fig. 8. Real data sets 1 and 2 with ground truth target locations (green).

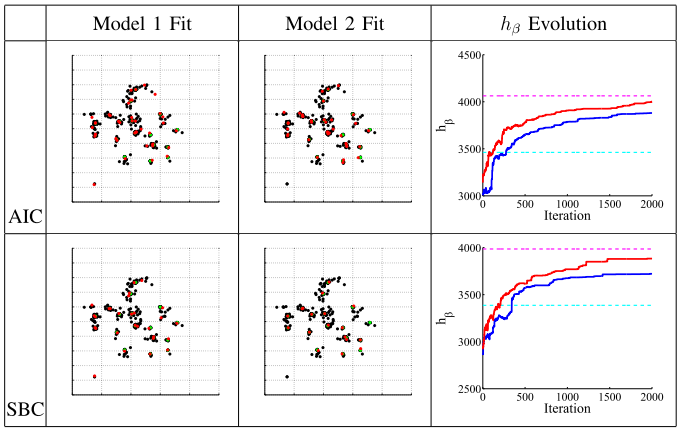

Fig. 9. Results of Algorithm 2 on real data set 1.

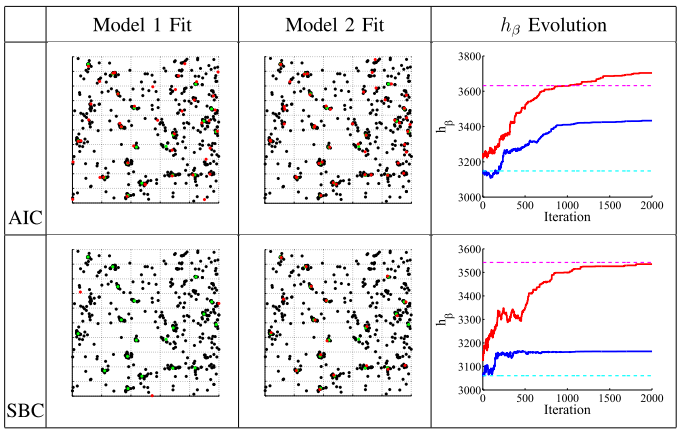

Fig. 10. Results of Algorithm 2 on real data set 2.

TABLE VIII
Results From Algorithm 2 on Real Data Set 1

		(a) AIC, model 1	(b) AIC, model 2	(c) SBC, model 1	(d) SBC, model 2
$\hat{\alpha}$	Best	10.84	11.54	17.04	21.21
	Opt	28.05	21.50	28.06	21.50
$\hat{\omega}$	Best	0.009607	0.006408	0.01230	0.005991
	Opt	0.02012	0.004533	0.02012	0.004533
$\hat{\eta}$	Best	N/A	49.92	N/A	95.26
	Opt		111.6		111.6
$n(\hat{C})$	Best	44	37	28	18
	Opt	17	17	17	17
h_β	Best	3946.7	4002.3	3721.7	3883.2
	Opt	3461.7	4062.7	3386.7	3985.6

VI. Conclusion

We have presented a statistical framework for mine detection in a spatial region. In this framework, the mine locations are modeled using a spatial point process, a Thomas process, with a certain cluster structure that is hierarchical—the locations of mines are modeled as parents while the marine life and other false detections are treated as children and clutter, respectively. Given the parents, the children locations are modeled as isotropic normal around the parents, and the remaining clutter follows an independent Poisson process. We have introduced a simulated annealing procedure for finding ML estimates of the model parameters. These parameters include parent locations that provide an estimate of mine locations despite being immersed in heavy clutter. The results obtained using both simulated and real SAS-based detections support the effectiveness of this framework in situations where observed contacts are clustered around mine locations.

TABLE IX
Results From Algorithm 2 on Real Data Set 2

		(a) AIC, model 1	(b) AIC, model 2	(c) SBC, model 1	(d) SBC, model 2
$\hat{\alpha}$	Best	14.72	8.155	415.9	14.06
	Opt	35.14	13.54	35.14	13.54
$\hat{\omega}$	Best	0.03377	0.01176	0.2713	0.01186
	Opt	0.09316	0.01096	0.09316	0.01096
$\hat{\eta}$	Best	N/A	216.2	N/A	321.8
	Opt		332.4		332.4
$n(\hat{C})$	Best	43	46	3	19
	Opt	19	19	19	19
h_β	Best	3433.4	3703.6	3164.1	3535.1
	Opt	3148.2	3632.0	3060.6	3542.2

As future work, we plan to extend this framework to situations where mine locations can be both clustered and scattered. Another possibility is to include ATR scores as a real-valued "mark" in addition to the location information in the inference process. One can also focus on scenarios where the quantity of clutter (false detections) compared to the number of mines in a scene is an order of magnitude higher than in the data sets given here.

References

[1] J. Stack, "Automation for underwater mine recognition: Current trends and future strategy," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 8017, pp. 80170K–80170K-21, 2011.

[2] J. Møller and R. Waagepetersen, *Statistical Inference and Simulation for Spatial Point Processes*. London, U.K.: Chapman&Hall/CRC, 2004.

[3] J. Møller and R. Waagepetersen, "Modern statistics for spatial point processes," *Scand. J. Stat.*, vol. 34, no. 4, pp. 643–684, 2007.

[4] R. Waagepetersen, "An estimating function approach to inference for inhomogeneous Neyman-Scott processes," *Biometrics*, vol. 63, no. 1, pp. 252–258, 2007.

[5] R. Waagepetersen and Y. Guan, "Two-step estimation for inhomogeneous spatial point processes," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 71, no. 3, pp. 685–702, 2009.

[6] P. Diggle, *Statistical Analysis of Spatial Point Patterns*, 2nd ed. Oxford, U.K.: Oxford Univ. Press, 2003.

[7] J. Illian, A. Penttinan, H. Stoyan, and D. Stoyan, *Statistical Analysis and Modelling of Spatial Point Processes*. New York, NY, USA: Wiley, 2008.

[8] P. Guttorp and T. Thorarinsdottir, "Bayesian inference for non-Markovian point processes," in *Advances and Challenges in Space-Time Modelling of Natural Events*, ser. Lecture Notes in Statistics, E. Porcu, J. Montero, and M. Schlather, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 79–102.

[9] T. Mrkvička, M. Muška, and J. Kubečka, "Two step estimation for Neyman-Scott point process with inhomogeneous cluster centers," *Stat. Comput.*, vol. 24, no. 1, pp. 91–100, 2014.

[10] D. C. Walsh and A. E. Raftery, "Classification of mixtures of spatial point processes via partial Bayes factors," *J. Comput. Graph. Stat.*, vol. 14, no. 1, pp. 139–154, 2005.

[11] N. Cressie and A. B. Lawson, "Bayesian hierarchical analysis of minefield data," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 3392, pp. 930–940, 1998.

[12] D. E. Lake, "Families of statistics for detecting minefields," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 3392, pp. 963–974, 1998.

[13] C. E. Priebe, D. Q. Naiman, and L. M. Cope, "Important sampling for spatial scan analysis: Computing scan statistic p-values for marked point process," *Comput. Stat. Data Anal.*, vol. 35, pp. 475–485, 2001.
[14] S. Byers and A. E. Raftery, "Nearest-neighbor clutter removal for estimating features in spatial point processes," *J. Amer. Stat. Assoc.*, vol. 93, no. 442, pp. 577–584, 1998.
[15] D. C. Walsh and A. E. Raftery, "Detecting mines in minefields with linear characteristics," *Technometrics*, vol. 44, no. 1, pp. 34–44, 2002.
[16] D. E. Lake, B. Sadler, and S. Casey, "Detecting regularity in minefields using collinearity and a modified Euclidean algorithm," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 3079, pp. 500–507, 1997.
[17] S. Agarwal, H. Ramachandran, S. Kummamuru, and O. R. Mitchell, "Algorithms and architecture for airborne minefield detection," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 4742, pp. 96–107, 2002.
[18] A. Trang, S. Agarwal, J. T. Broach, and T. Smith, "Exploiting "mineness" for scatterable minefield detection," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 6953, pp. 1–12, 2008.
[19] A. Trang, S. Agarwal, J. T. Broach, and T. Smith, "Validating spectral spatial detection based on MMPP formulation," *Proc. SPIE—Int. Soc. Opt. Eng.*, vol. 8017, pp. 1–12, 2011.
[20] M. N. M. v. Lieshout and A. J. Baddeley, "Extrapolating and interpolating spatial patterns," in *Spatial Cluster Modelling*, A. B. Lawson and D. Denison, Eds. Boca Raton, FL, USA: Chapman&Hall/CRC Press, 2002, pp. 61–86.
[21] I. W. McKeague and M. Loizeaux, "Perfect sampling for point process cluster modelling," in *Spatial Cluster Modelling*, A. B. Lawson and D. Denison, Eds. Boca Raton, FL, USA: Chapman & Hall/CRC Press, 2002, pp. 87–107.
[22] R. Waagepetersen and T. Schweder, "Likelihood-based inference for clustered line transect data," *J. Agricultural Biol. Environ. Stat.*, vol. 11, no. 3, pp. 264–279, 2006.
[23] C. Fraley, A. E. Raftery, T. B. Murphy, and L. Scrucca, "Mclust version 4 for R: Normal mixture modeling for model-based clustering, classification, and density estimation," Dept. Stat., Univ. Washington, Seattle, WA, USA, Tech. Rep. 597, 2012.
[24] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Nov. 1974.
[25] G. E. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.
[26] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Chichester, U.K.: Wiley, 1989.
[27] M. Soumekh, *Synthetic Aperture Radar Signal Processing*. New York, NY, USA: Wiley, 1999.
[28] J. Tucker and M. Azimi-Sadjadi, "Coherence-based underwater target detection from multiple sonar platforms," *IEEE J. Ocean. Eng.*, vol. 36, no. 1, pp. 37–51, Jan. 2011.
[29] J. Isaacs and J. Tucker, "Diffusion features for target specific recognition with synthetic aperture sonar raw signals and acoustic color," in *Proc. IEEE Int. Conf. Pattern Recognit. Comput. Vis. Workshops*, Jun. 2011, pp. 27–32.

Darshan Bryner received the B.S. and M.S. degrees in mathematics from Georgia Institute of Technology, Atlanta, GA, USA, in 2007 and 2008, respectively, and the Ph.D. degree in statistics from Florida State University, Tallahassee, FL, USA, in 2013, under the co-advisement of Dr. A. Srivastava and Dr. E. Klassen.

In 2008, he joined the Naval Surface Warfare Center Panama City Division, Panama City, FL, USA, as a Mathematician. He continues to work at the Naval Surface Warfare Center in Panama City where his research focuses on statistical shape analysis, spatial point processes, and computational biology.

Fred Huffer received the B.S. degree in mathematics from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1978 and the Ph.D. degree in statistics from Stanford University, Stanford, CA, USA, in 1982.

Since 1982, he has been a member of the faculty in the Department of Statistics, Florida State University, Tallahassee, FL, USA.

Anuj Srivastava (S'87–M'96–SM'09) received the Ph.D. degree from Washington University in St. Louis, St. Louis, MO, USA, in 1996.

He is a Professor of Statistics and a Distinguished Research Professor at the Florida State University, Tallahassee, FL, USA. He was a Visiting Research Associate at Brown University, Providence, RI, USA, during 1996–1997. He joined the Department of Statistics, Florida State University, in 1997 as an Assistant Professor, and became a full Professor in 2007. His areas of research include statistics on nonlinear manifolds, functional data analysis, and statistical shape theory. He has published more than 200 papers in journals and conference proceedings.

Dr. Srivastava is/has been an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and the IEEE TRANSACTIONS ON IMAGE PROCESSING.

J. Derek Tucker (S'05–M'08) received the B.S. degree in electrical engineering (*cum laude*) and the M.S. degree in electrical engineering from Colorado State University, Fort Collins, CO, USA, in 2007 and 2009, respectively, and the Ph.D. degree in statistics from Florida State University, Tallahassee, FL, USA, in 2014, under the co-advisement of Dr. Anuj Srivastava and Dr. Wei Wu.

Upon completion of these degrees, he began working as a Research Scientist at the Naval Surface Warfare Center Panama City Division (NSWC-PCD), Panama City, FL, USA. Recently, he has joined the Radar and Signal Analysis Department at Sandia National Laboratories, Albuquerque, NM, USA. His research is focused on pattern theoretic approaches to problems in image analysis, computer vision, signal processing, and functional data analysis.