H13-33
SOURCE TERM ESTIMATION FOR RAPID HAZARD ASSESSMENT

Gareth Brown and Peter Robins[1]

[1]The Defence Science and Technology Laboratory (Dstl), Salisbury, England

Abstract: A methodology is developed for making inference about parameters associated with a possible chemical or biological atmospheric release from sensor readings. The key difficulty in performing this inference is that the results must be obtained in a very short timescale in order to make use of the inference for protection. The methodology developed uses some of the components in a sequential Monte Carlo algorithm. This technique employs Bayesian probability reasoning over a dynamic sample-set of typically thousands of hypothesized releases. For each release, a Gaussian puff model is run and its output is used for posterior probability density calculation from event data through sensor and observer likelihood models.

Key words: *Bayesian data fusion, source-term estimation, inverse dispersion modelling.*

INTRODUCTION

In the event of a Chemical or Biological (CB) release, effective incident response requires a robust knowledge management system that makes optimal and timely use of all available sensory data. Prompt and accurate assessment of the hazard can initiate appropriate response procedures minimizing human exposure. This is achieved through hazard predictions obtained from operational dispersion models. However, in order to run a dispersion model we must infer the characteristics describing the source-term, and its local meteorological environment, from available sensor data. It is also necessary to determine whether such a release has actually occurred. Assuming a release has occurred, the true characteristics of the source are generally uncertain. Furthermore, for hazard assessment within the first five minutes of release, there is likely to be little data available and limited computational resources to process it. Thus we are interested in rapidly estimating a source-term's location, in order to successfully define an appropriate hazard area.

BAYESIAN INFERENCE

The problem described is highly complex due to the probabilistic nature of atmospheric dispersion and sensor outputs. Inference is further complicated because it is assumed that while the parameters to be inferred remain fixed, the sensor outputs are obtained sequentially in time. The solution, described herein as the Monte Carlo Bayesian Data Fusion Algorithm (MCBDF), incorporates a real-time Bayesian posterior probability density sampling algorithm. This algorithm uses Bayesian probability reasoning over a sample set of many hypothesised source-terms (we use the term source-term to include meteorological variables) and allows the set to be updated when new information is received. The system also enables disparate data of varying quality to be combined in a computationally expedient way. The object of interest is the posterior distribution; this describes the probability distribution of the source-term. Assuming a distribution with a single peak, the location of the peak in the parameter space identifies the best source-term estimate and the width of the n dimensional peak describes the uncertainty in that best estimate. The posterior distribution is calculated using Bayes' rule and it is given as:

$$\underbrace{p(\theta|\mathbf{D})}_{posterior} \propto \underbrace{p(\theta)}_{prior}\underbrace{p(\mathbf{D}|\theta)}_{likelihood}, \qquad (1)$$

where θ is the source-term and $\mathbf{D}$ is the data. The prior distribution, $p(\theta)$, is defined as the assumed probability of the source-term parameters before any data have been received. The likelihood distribution, $p(\mathbf{D}|\theta)$, is a measure of how likely a particular data set is given a particular source term. The posterior distribution, $p(\theta|\mathbf{D})$, indicates how likely the source term parameters are, given the data. The symbol θ represents an $n=9$ dimensional vector encompassing the parameters: location (x,y), time of release, t, mass m, agent type a, surface wind-vector (u,v), Monin Obukhov length L and surface roughness z_0, that is, $\theta=(x,y,t,m,a,u,v,L,z_0)$. It is worth noting that these parameters are all highly correlated. For example, a more massive release further back in time and further away from the sensors may produce similar sensor readings to a small release closer to the sensors at a later time.

THE MONTE CARLO BAYESIAN DATA FUSION ALGORITHM

Standard Bayesian analysis normally relies on Markov Chain Monte Carlo (MCMC) algorithms to perform thousands of likelihood calculations. In situations where likelihood calculations are computationally expensive and there are time or processor constraints, standard approaches may prove infeasible. There are several new Monte Carlo techniques which allow Bayesian computation when there are computational constraints including Sequential Monte Carlo (SMC) (Doucetet *et al.*, 2000 and Doucetet *et al.*, 2001), approximate Bayesian computation (ABC) (Beaumont *et al.*, 2002) and SMC Samplers (Del Moral *et al.*, 2006). The MCBDF methodology incorporates the combination of MCMC with aspects of an SMC algorithm. It combines them with some innovative, problem specific techniques, to make inference about a highly complex multimodal posterior distribution where likelihood calculations are computationally expensive and sequential information about an event in the past is received in real-time (Robins, 2009). This approach is designed to minimize the computational burden of evaluating a time-dependent posterior and minimize the likelihood of becoming 'stuck' in a local mode.

Rather than performing the computationally expensive task of calculating the complete posterior distribution for the source-term parameters, MCBDF employs a sampling approach to approximate the posterior (Ristic *et al.*, 2004). This sampling is performed as follows: at every time instant k define a large collection of N weighted random samples $\{\theta_k^{(i)}, w_k^{(i)}\}$ for $i = 1,...,N$, such that $w_k^{(i)} > 0$ for all i and all k and $\Sigma_i^N w_k^{(i)} = 1$. A hypothesis at time k is denoted $\theta_k^{(i)}$ and $w_k^{(i)}$ is the associated weight, which reflects the importance of that hypothesis relative to the complete set of hypotheses. MCMC and Particle Filter algorithms are designed such that as the number of hypotheses increases, the empirical distribution converges asymptotically to the target posterior distribution of the parameters θ. Hence, the approximation of the true posterior distribution is achieved through the combination of the clustering of distinct hypotheses around areas of high density in the posterior and also the associated weight of that hypothesis. The reason for adopting a weighted sample approach in particular is that the source-term estimation problem requires the data to be processed on-line, as it arrives. This constraint is due to the cost of storing dispersion data and the rapidly changing state of knowledge about the source-term. This approach also ensures that whenever the user requests an updated hazard prediction, the current source-term posterior probability density distribution is the most accurate one possible.

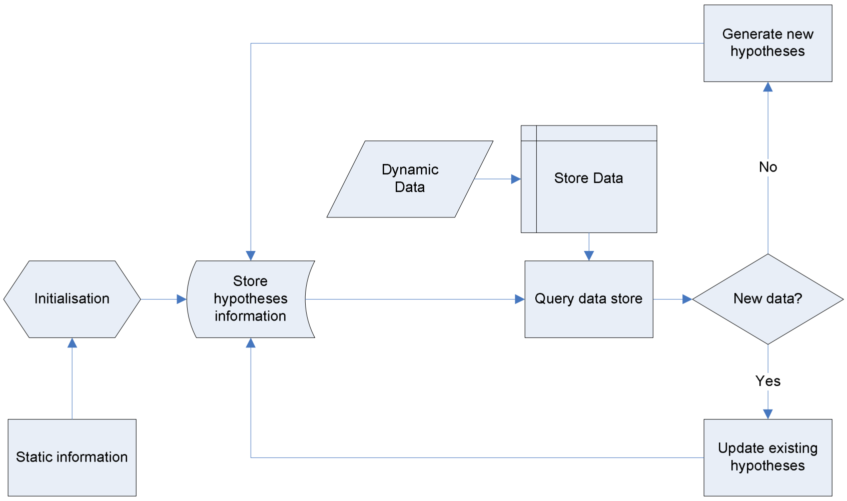

Figure 1 Flow chart of the MCBDF algorithm. Initialisation data starts the program, which then enters a continuous loop. MCBDF queries its data store for new data. If new data exists the weights of current hypotheses are updated based on the related sensor data likelihood calculation. If no new sensor data is available then more source-term hypotheses are generated until the data store is again checked.

Algorithm structure

The MCBDF system consists of several independent parts controlled by an overarching class which can process live input data in any order. This is necessary to ensure that there is no assumption of consecutive time in the incoming messages, as in a deployed system it is unlikely that messages will be received in time order. The only constraint is that there is a fixed-sized time-window, T_D (typically 30 minutes), in which information will be considered to ensure the computational complexity remains within acceptable bounds. Given the current time, T, this time-window leads to sensor data and hypotheses older than $T - T_D$ being discarded by the algorithm. A flow chart showing the basic structure of the algorithm is displayed in Figure . The MCBDF algorithm is first initialised by static data that defines the problem; this includes for example the domain under consideration, the corresponding size for the time-window and the prior distribution. Based on this information MCBDF creates a single initial hypothesis.

Once the first hypothesis is created, MCBDF enters a loop where it checks for the presence of new data in its data store. If no new sensor data is available then more source-term hypotheses are generated until the data store is again checked, this occurs at a user defined rate (typically every 10 seconds). If new data is present then sensor data and hypotheses older than $T - T_D$ are discarded and the remaining hypotheses' weights, $w_k^{(i)}$, for all N hypotheses in the system, are updated. The weights are updated by calculating the likelihood of the new piece of data, d, for each hypothesis as

$$w_{k+1}^{(i)} = w_k^{(i)} p\left(d \middle| \theta_k^{(i)}\right). \qquad (2)$$

The prior distribution

In order to apply a Bayesian approach, prior distribution functions are required for each of the source term parameters. The prior distributions can incorporate expert judgement and previous experience into the current estimate. Generally, in the absence of any intelligence data to the contrary, Laplace's principle of indifference (O'Hagan *et al.*, 2004) may be used to set the prior distribution as uniform over the whole hypothesis space. Thus, while it is possible to incorporate knowledge about possible release locations into the prior, this knowledge may not always be available. In the general case a 30km square domain is assumed and the prior is set uniform over this space, including exponentially decaying tails outside of the square domain. This allows the mode of the posterior to be outside the specified parameter space if the data suggests.

The primary motivation for carrying out source term estimation is to be able to carry out hazard predictions. It is therefore necessary to determine whether any kind of hazard exists at all. Thus, it is not correct to proceed by making inferences conditional upon the fact that a release has definitely occurred. If the incoming data does not support this hypothesis, then erroneous inferences will be made. As a consequence, and to simplify the sampling, a surrogate mass parameter m^* is used. The prior distribution on the surrogate mass is a double exponential distribution of the form

$$p(m^*) = \frac{1}{2} e^{-\left|m^*\right|} . \tag{3}$$

In an operational system, a great deal of variation is expected in mean release mass between different agents. In order to assist the sampling between agents, the surrogate mass parameter does not represent the true mass, but a mass multiplier. Whenever mass is required for modelling and inference, it is multiplied by the appropriate mean mass, μ_m , for the particular agent being considered, as

$$m = \begin{cases} 0 & m^* \le 0 \\ m^* \mu_m & m^* > 0 \end{cases} . \tag{4}$$

The priors for the meteorological parameters u, v, L, z_0 , are given as follows: the wind speed components prior, $p(u,v)$, is a normal distribution, with no preferred direction, centred at $0 \text{ m} \cdot \text{s}^{-1}$ with variance $100 \text{m}^2 \cdot \text{s}^{-2}$. The Monin-Obukhov length prior, $p(L)$, and the surface roughness prior, $p(z_0)$, are mostly uninformative. The Monin-Obukhov length prior lightly penalizes very unstable and very stable atmospheric conditions.

Likelihood calculations for new data

MCBDF's flexibility is encapsulated by its ability to assimilate data of several forms. These data types can be loosely divided into three categories: measurements taken from CB detectors, observations from human observers and measurements from meteorological sensors. The wind measurement likelihood calculation requires access to the meteorology object used by the dispersion model associated to a particular hypothesis $\theta_k^{(i)}$. The inputs to the likelihood calculation are: the wind-vector measurement at the sensor location and height, $\mathbf{u}$, the corresponding hypothesized wind-vector derived from the dispersion model's meteorology, $\boldsymbol{\mu}_u$, and the measurement uncertainty covariance matrix, $\boldsymbol{\Sigma}$. The likelihood, $p(\mathbf{u}|\boldsymbol{\mu}_u, \boldsymbol{\Sigma})$, is calculated as a bivariate normal probability density

$$p(\mathbf{u}|\boldsymbol{\mu}_u, \boldsymbol{\Sigma}) = \phi(\mathbf{u}|\boldsymbol{\mu}_u, \boldsymbol{\Sigma}) . \tag{7}$$

Likelihoods for detector measurements are calculated by first running a dispersion model for each hypothesis $\theta_k^{(i)}$. Dispersion is a stochastic process in nature, therefore the physics models developed to represent dispersion must be ensemble models which account for atmospheric turbulence in a statistical manner. In this application, a Gaussian puff dispersion model is used due to the relatively fast run-time in comparison to other models. This model assumes that the concentration c , for given mean μ and variance σ^2 , is described by a clipped normal distribution $p(c \mid \mu, \sigma^2)$. The likelihood of a given downwind sensor measurement d can be written as:

$$p\left(d \middle| \theta_k^{(i)}\right) = p(d \mid \mu, \sigma^2) = p(d \mid c)\, p(c \mid \mu, \sigma^2) , \tag{8}$$

where $p(d \mid c)$ denotes the probability of the detector measuring the data d conditional on the concentration c . The actual value of the concentration, c , is never directly observed and is therefore a nuisance parameter. A marginal likelihood is obtained by integrating c out of the likelihood model:

$$p(d \mid \mu, \sigma^2) \quad = \quad \int_0^\infty \underbrace{p(d \mid C = c)}_{\substack{\text{measurement} \\ \text{density}}} \underbrace{p(C = c \mid \mu, \sigma^2)}_{\substack{\text{concentration} \\ \text{density}}} \, \text{d}c . \tag{9}$$

The key part of Equation 8 is $p(d \mid c)$, this defines the sensor model and permits, in principle, a likelihood model for any type of detector. A specific example is the concentration sensor. This is modelled as making a direct measurement of the local concentration (with any bias or scaling removed) with a normally distributed measurement error, σ_e . The model includes upper and lower measurement thresholds, $\overline{L}$ and $\underline{L}$ respectively, and the probability of a measurement given the unobserved concentration c is:

$$p(d|c) = \begin{cases} \Phi\left(\underline{L} \middle| c, \sigma_e^2\right) & d = \underline{L} \\ \phi\left(d \middle| c, \sigma_e^2\right) & \underline{L} < d < \overline{L} \\ 1 - \Phi\left(\overline{L} \middle| c, \sigma_e^2\right) & d = \overline{L} \end{cases} . \tag{10}$$

where $\phi(\square|\square,\square)$ is the normal distribution probability density function, $\Phi(\square|\square,\square)$ is the normal cumulative distribution function.

Generating new source-term hypotheses

When the inference engine is not processing incoming data, new samples are generated according to the current posterior distribution. To do this MCBDF uses a scheme known as Differential Evolution Markov Chain (DE-MC) to generate new hypotheses (Braak *et al.*, 2006). This is a variation of traditional Markov Chain Monte Carlo (MCMC) algorithms. Such algorithms aim to generate new hypotheses from existing ones such that the collection of hypotheses generated over time, starting from a single hypothesis, forms a Markov chain. As the chain extends to infinity, the overall set of hypotheses converges to a true representation of the posterior distribution. The DE-MC scheme generates several separate Markov chains of hypotheses, and "breeds" separate "chain-end" hypotheses together to create new hypotheses. This technique allows the set of hypotheses to move to different regions of the hypothesis space, thus ensuring that all possible regions can be covered, while also focusing on those regions where the posterior density is highest.

In the hypothesis generation scheme, M of the N existing hypotheses are chosen as the ends of the Markov chains, and used to generate proposals for new hypotheses. These hypotheses, known as the DE-MC population, are labelled θ_r, where $r = 1,\mathrm{K}\ ,M$. A new hypothesis, θ_i^*, is proposed by selecting a member of the population, θ_i, known as the "candidate" hypothesis, and adding to it the weighted difference of two other hypotheses from the population:

$$\theta_i^* = \theta_i + \gamma\left(\theta_j - \theta_k\right) + \varepsilon\,, \tag{11}$$

where ε is normally distributed around the origin of the hypothesis space with a narrow covariance, and γ is a scalar (see Braak, 2006). The candidate hypothesis θ_i is chosen by iterating through the indices $p = 1,\mathrm{K}\ ,M$. The two other hypotheses θ_j and θ_k are chosen randomly such that $i \neq j \neq k$. Proposals are accepted as the next value $\theta_i^{\text{new}} = \theta_i^*$ if

$$U\left(0,1\right) < \frac{p\left(\theta_i^*\right)\prod_{j=1}^{N_d} p\left(d_j \mid \theta_i^*\right)}{p\left(\theta_i\right)\prod_{j=1}^{N_d} p\left(d_j \mid \theta_i\right)}\,, \tag{12}$$

where $U(0,1)$ is a random number uniform over (0,1). The symbol, $p(\theta)$, denotes the total prior, given as the product of individual prior probabilities and $p(d \mid \theta)$ denotes the likelihood of the data, d . The product in Equation 12 is calculated over all available data N_d . If the proposal is not accepted, then the current value of θ_i is retained, $\theta_i^{\text{new}} = \theta_i$.

HAZARD CALCULATION

In an operational system, the prior on a release actually occurring, $P(m^* > 0)$, is likely to be small. Incorporation of this prior into Equation 12 would lead to a proliferation of no release hypotheses which, in turn, could be a large computational and storage overhead despite their individually small storage requirements. The prior on release is therefore ignored for sampling as it is independent of all the other parameters and only applied when inferences are required. The probability of release given the data is calculated as follows:

$$P\left(m^* > 0 \middle| \mathbf{D}\right) = \frac{P\left(m^* > 0\right)\sum_i w_k^{(i)} \mathrm{I}\left(m_i^* > 0\right)}{P\left(m^* > 0\right)\sum_i w_k^{(i)} \mathrm{I}\left(m_i^* > 0\right) + \left(1 - P\left(m^* > 0\right)\right)\sum_i w_k^{(i)} \mathrm{I}\left(m_i^* \leq 0\right)}\,, \tag{13}$$

Where $w_k^{(i)}$ is the weight of the ith hypothesis, $P(m^* > 0)$ is the prior probability of release and $\mathrm{I}(m_i^* > 0)$ is an indicator function that returns one if the hypothesized mass is strictly positive and zero otherwise.

Currently, the user is given output at regular intervals while $P(m^* > 0|\mathbf{D}) > 0.5$. The frequency of this output is a configurable input. The output is currently a 100 member subset of the current hypothesized release parameters contained within the system. The total set would probably be overwhelming for hazard calculations and so a sub-sampling scheme is employed. The scheme starts with the most recently generated hypothesis and moves chronologically down the hypothesis list, adding members to the subset with a probability proportional to each hypothesis' weight; the selected hypothesized release parameters are then used to re-run the dispersion model. For each re-run, the probability of exceeding some user-defined threshold level of effect, at some user-defined time, is calculated over the area of interest and a weighted average of all 100 members taken.

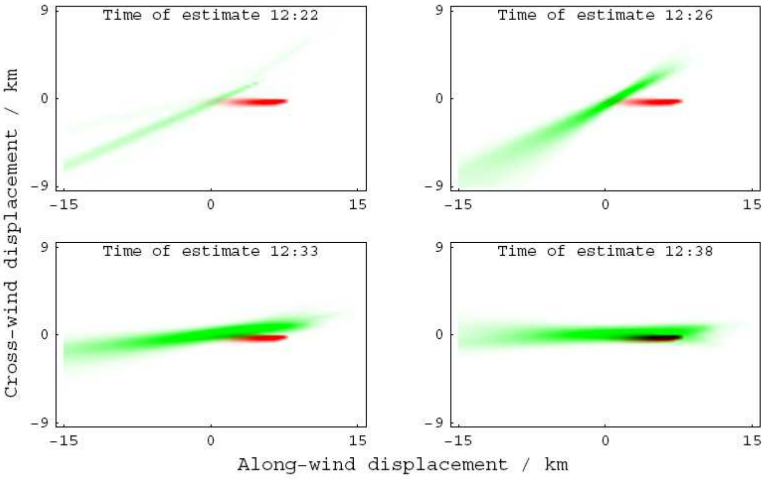

Figure 2 Temporal evolution of the MCBDF hazard reporter with incorrect meteorological data (reported wind direction 15° from the true direction). The challenge, synthetically generated by applying a concentration realisation model to SCIPUFF generated dispersion data, was emitted at 12:00, 8km upwind and 0.7km crosswind from a network of 19 detectors, spaced 500m apart, centred on the origin. As soon as the challenge hits the network an initial estimate of the challenge is provided. As more detector measurements are processed, the erroneous wind direction is disfavoured for wind-directions that are more compatible with the measurements. Red identifies false negative regions, black true positive and green gives false positive regions. Estimates were calculated in real time on a desktop machine.

Example output from the hazard calculator is shown in Figure 2. In the case shown, MCBDF is using the UDM dispersion model to predict the hazard area created by the SCIPUFF (Sykes *et al.*, 2007) dispersion model. This represents a more realistic test of the algorithm since the dispersion model used to make the inference is not the same as the model creating the hazard. Realistic concentration time-series were generated by feeding the SCIPUFF estimates for mean mass-concentration, variance and correlation time-scale into a concentration realisation model. The first image shows MCBDF's initial hazard estimate at 12:22, the time at which $P(m^* > 0|\mathbf{D}) > 0.5$ first occurs. The estimate is poor because there are only a few non-null sensor measurements in the data store and the wind-vector measurement supplied to the algorithm is in error by 15º. Subsequent images show the improvement in MCBDF's assessment as more sensory data is passed to the algorithm. The algorithm ran in real-time on a modern desktop machine.

REFERENCES

Beaumont, M. A., W. Y. Zhang, and D. Balding, 2002: Approximate Bayesian computation in population genetics, *Genetics*, 162 (4), pp. 2025–2035

Braak, C. J., 2006: A Markov chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces. Statistics and Computing 16, pp. 230–249.

Del Moral, P., A. Doucet, and A. Jasra, 2006: Sequential Monte Carlo Samplers. *Journal of the Royal Statistical Society (Series B)*, 68 (3), 411–436.

Doucet, A., S. J. Godsill, and C. Andrieu, 2000: On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing*, 10, pp. 197–208.

Doucet, A., de Freitas N., and Gordon, N., 2001: Sequential Monte Carlo Methods in Practice, *Statistics for Engineering and Information Science*, Springer,

O'Hagan, A. and J. Forster, 2004: Kendall's Advanced Theory of Statistics 2nd Edition Vol. 2B. Bayesian Inference.

Ristic, B., S. Arulampalam, N. Gordon, 2004: Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House Publishers

Robins, P., V.E. Rapley, and N. Green, 2008: Real-time sequential inference of static parameters with expensive likelihood calculations. *Journal of the Royal Statistical Society: Series C.*, **58**, 641-662.

Sykes, R.I., S. Parker, D. Henn, B. Chowdhury, 2007: SCIPUFF Version 2.3 Technical Documentation. L-3 Titan Corp, POB 2229, Princeton, NJ 08543-2229, 336 pp.