Towards a Discipline of Geospatial Distributed Event Based Systems

Annie Liu
Computer Science, Caltech
1200 E California Blvd
Pasadena, CA 91125, USA
aliu@cms.caltech.edu

Michael Olson
Computer Science, Caltech
1200 E California Blvd
Pasadena, CA 91125, USA
molson@cms.caltech.edu

Julian Bunn
CACR, Caltech
1200 E California Blvd
Pasadena, CA 91125, USA
julian.bunn@caltech.edu

K. Mani Chandy
Computer Science, Caltech
1200 E California Blvd
Pasadena, CA 91125, USA
mani@cms.caltech.edu

ABSTRACT

A geospatial system is one in which the state space includes one, two or three-dimensional space and time. A geospatial event is one in which an event impacts points in space over time. Examples of geospatial events include floods, tsunamis, earthquakes, and emission of toxic plumes. This paper discusses aspects of the theory of geospatial distributed event based systems (GDEBS). The paper describes algorithms for rapid detection of geospatial events which can be used on Cloud computing architectures, in which many servers collaborate to detect events by analyzing data streams from large numbers of sensors. Sensor noise and timing errors may result in false detection or missed detection as well as incorrect identification of event attributes such as the location of the event source. The paper presents mathematical analyses and simulations dealing with rapid event detection for geospatial events of varying speeds in the presence of substantial sensor noise and timing error. The paper also describes some of the algorithmic and machine-learning techniques for improving event detection in the Cloud with large numbers of noisy sensors. Experience with GDEBS using a seismic network is described.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; G.3 [**Probability and Statistics**]: Experimental Design

General Terms

Algorithms, Design, Experimentation

DEBS '12, July 16–20, 2012, Berlin, Germany.
Copyright 2012 ACM 978-1-4503-1315-5 ...$10.00.

Keywords

event detection, sensor networks, geospatial data, Bayesian estimation, seismology

1. INTRODUCTION

Geospatial distributed event based systems (GDEBS) are used for three main goals [1]:

1. Providing early warning so that people or machine components can react before a disaster hits; for example alerting people about impending intensive shaking from earthquakes.
2. Providing continuing situational awareness as a disaster unfolds; for instance, giving first responders information about which areas have been most badly damaged.
3. Providing data that is useful for scientific analysis such as data about background radiation measured in an area over time.

This paper describes event detection algorithms and theories, and data from a working system implemented to satisfy all three goals of GDEBS.

Accuracy is crucial in achieveing all three goals. For example, the onset of earthquakes must be detected in a few seconds so that people can take action before they experience intensive shaking. People ignore systems that produce too many false alerts. If a significant event to which people must respond occurs once every five years, then they may pay inadequate attention to a system that generates false alerts at a rate of once every three months. Therefore GDEBS must detect events accurately.

A focus of this paper is on community based systems in which members of different communities — ordinary residents, power and water utilities, and government agencies — participate by hosting sensors, actuators, and computation engines. Multi-community systems have to deal with extremes of heterogeneity, uncertainty, distribution, poor deployment, unreliable operation, and high load variability. Therefore, these systems are good vehicles for studying some of the limits of event detection. Fig. 1 shows the sensor distribution in an existing community-based system.

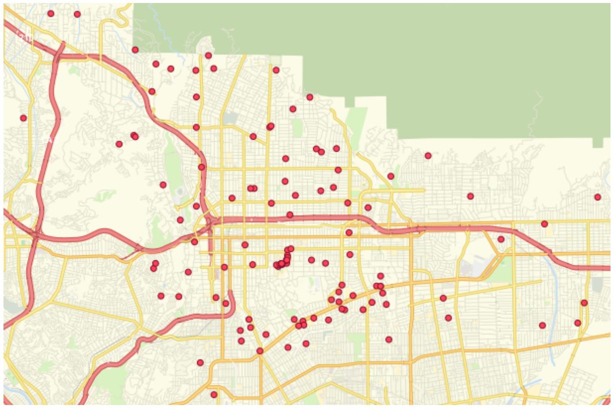


Figure 1: CSN[2] sensors in the Pasadena, CA area.

Geospatial event detection is based on models of the propagation of activity across space and time. Streams of data from sensors are used to estimate the model that best fits the data. If a specific model fits the data with high confidence, then that model is used to determine the appropriate responses; otherwise the system waits for additional data. A difficulty with building general-purpose GDEBS event processing notations is the wide difference in models for the propagation of different types of geospatial activity, such as earthquakes and floods.

In this paper, we do not attempt to address the differences in models for all geospatial events. Instead, we introduce a general high-level event propagation and detection model (Sec. 2) and apply them to a seismology application. The model is supplied with a distributed, efficient Bayesian detection algorithm (Sec. 3) whose theoretical properties are analyzed (Sec. 4) and validated (Sec. 6).

2. MODELS OF GEOSPATIAL EVENT DETECTION

2.1 Models of Geospatial Events

We describe a model of geospatial activity and then describe its limitations for different applications. This model is adapted in the rest of the paper.

A geospatial event is initiated at a point in space-time. For 3-D space, the event initiation point is specified by the space-time coordinates (x_0, y_0, z_0, t_0). For example, the model assumes that an earthquake starts at a point in 3-D space (the hypocenter) at an instant in time. Likewise, the model assumes that a forest fire starts with a spark at a specific location and time.

Associated with an event initiation are parameters that describe the event. For example, the parameters that describe an earthquake include its magnitude; the parameters for a toxic plume include the concentration and type of toxic material. Let M be the set of parameters that describe the event. Then, an event initiation is completely described by the 5-tuple (M, x_0, y_0, z_0, t_0). For example, a simple model of earthquake initiations has M as the magnitude of the quake.

The manifestation of a geospatial event at any space-time point (x, y, z, t) is given by a vector $H(x, y, z, t)$ of environmental factors such as temperature, concentration of pollutants, and acceleration at point (x, y, z) in space and time t. The model of propagation of geospatial activity is specified by a function f that gives the manifestation H of the event at each point in space time, given an initiation (M, x_0, y_0, z_0, t_0) of a geospatial event:

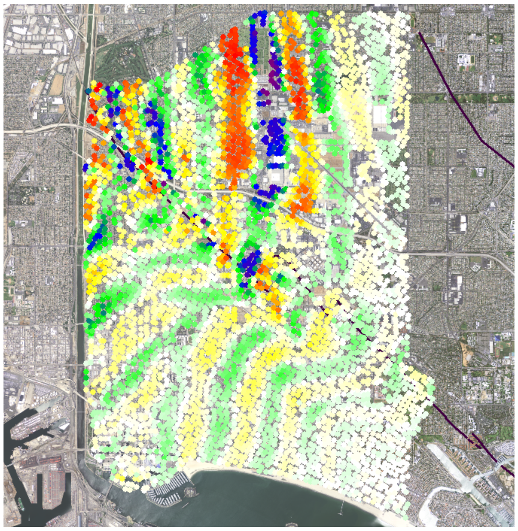


Figure 2: Snapshot of $H(x, y, t)$ of a seismic event with an event initiation in the upward left direction

$$H(x, y, z, t) = f(M, x_0, y_0, z_0, t_0, x, y, z, t)$$

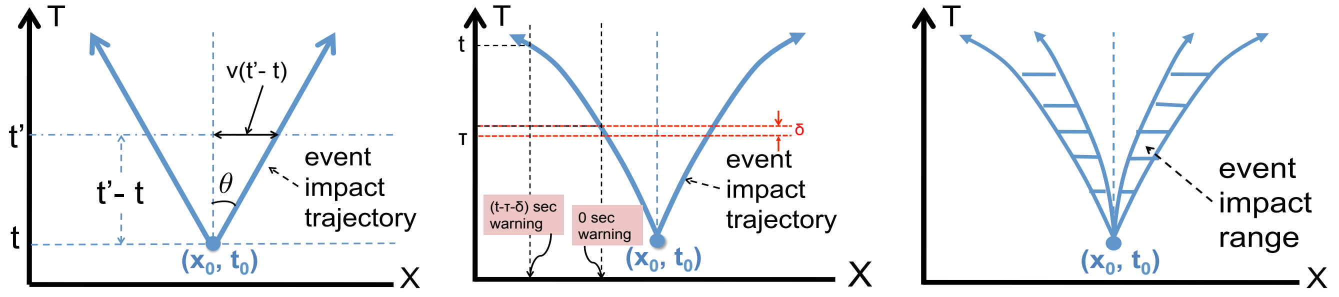
In this paper, we will often consider functions that are horizontally isotropic (invariant under horizontal translation) and that are independent of vertical distance z.

$$H(x, y, t) = f'(M, d(x - x_0, y - y_0), t - t_0)$$

where $d(u, v)$ is the length of the 2-D vector (u, v). Fig. 2 is a snapshot of H visualized during a seismic event.

In the presence of multiple events, the net manifestation is assumed to be the sum of the manifestations due to each of the events. This is a simplistic model because it assumes that the net effect of multiple events is additive. For example, the model assumes that the acceleration at a point due to two separate concurrent earthquakes is the sum of the accelerations due to each quake. This simplistic model gives useful results in situations for which the probability of concurrent events is low — i.e., the effect of one quake at a point dies down before the effect of the next quake is felt at that point. The model is also relevant for the case where the impact of a single geospatial event is small at each point in space-time, so that a linear model gives accurate results.

The behavior of the entire geospatial system is captured by the function f. In the case of dispersion of a toxic plume, conditions such a wind patterns, humidity, and precipitation are captured by f. The function f for any realization of a propagating geospatial event is drawn from a distribution F. For example, the speed of seismic waves depends on the geological structure of the material through which the waves propagate, and the structure may not be known. We assume that there is some speed with which waves travel in a given earthquake, but the speed is unknown to the designers of the GDEBS application, though designers may know the distribution from which f is drawn.



(a) Constant speed model (b) Space-time dependent speed model (c) Event Impact region

Figure 3: Geospatial event model illustrated in space-time plots. (x_0, t_0) **is the event origin in space-time.**

2.2 Models of Isotropic Events

We introduce models for geospatial events by first considering isotropic events in which the impact of an event propagates equally in all directions. In reality, the impact propagates faster and with greater intensity in some directions, and we will deal with this effect later.

Without loss of generality, first consider a 1-D model with the 1-D space dimension on the x-axis and time t on the y-axis. An event initiated at a point (x, t) in space time will propagate along a cone with apex (x, t) where the boundary rays of the cone have an angle θ where $tan\theta = v$ (Fig. 3(a)). Generally, the speed of propagation is not constant, and variable speeds give rise to non-conic shapes as in Fig. 3(b).

In most cases, the impact of a geospatial event at a point in space continues for some time. For example, shaking from an earthquake may continue for many seconds and even minutes. Likewise, dangerous levels of radioactive material may remain at a point in space for days. Thus the region of impact of a geospatial event is represented by the points between two cones if the velocities of arrival and departure of the impact are constant; more generally the region of impact is represented by the space-time points between two upwardly increasing shapes (Fig. 3(c)).

The intensity of the impact of an event also varies with space and time. If waves of intensity emanate in all directions from a point source, like ripples in a pond, then the points of maximum intensity correspond to collections of cones with the same apex. Generally, intensities vary in complex and random ways.

2.3 Metrics for Evaluating GDEBS

Fig. 3(b) shows an event detection at a time τ. Alerts about the event are sent electronically to the impacted region, and as a first approximation, we assume that alerts reach all points in the region after a delay of δ seconds. The warning time for a point x in space is the delay between the instant $(\tau + \delta)$ and the instant at which dangerous intensities are experienced at point x. As shown in Fig. 3(b), some locations may have adequate warning while other locations do not. One of the metrics for evaluating the effectiveness of GDEBS is the amount of area, or more appropriately, the fraction of the population in the region, that gets adequate warning time.

A system may generate false warnings. Warnings generated a short interval after the initiation of an event are based on data gathered over short times, and hence are more likely to be erroneous. Some GDEBS systems associate probabilities with warnings, generally giving higher probability warnings as time progresses. Another metric for evaluating the effectiveness of GDEBS systems is the rate of false positives.

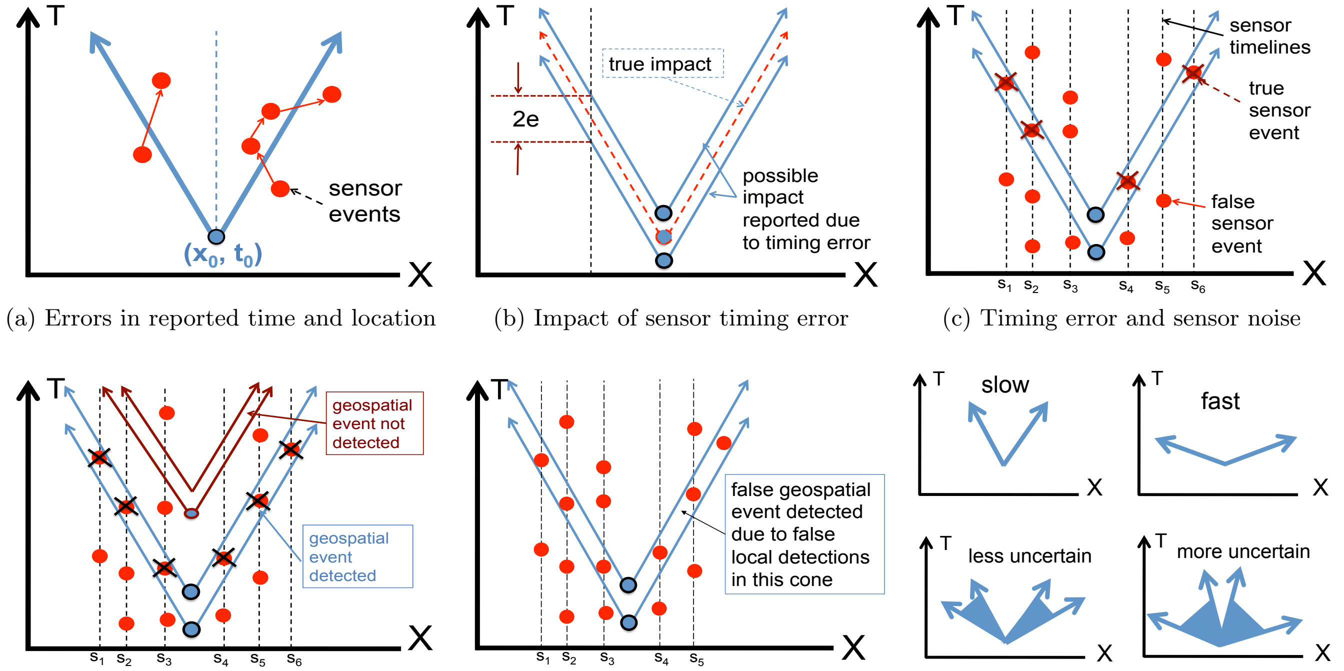
A warning sent to a location can be a single bit — "You are about to feel the impact of a geospatial event," or it can have an associated probability and it can have an estimate of the magnitude of the impact of the event at that location. For example, Southern California has had six earthquakes reported by the U.S. Geological Service in the last six months, but all of them were too small to require any sort of reaction. Warnings about such events would be considered to be false by most people who only want to be alerted when they have to respond.

The appropriate metrics for the evaluation of GDEBS depend on which of the three goals the application is used for — (1) warning, (2) ongoing situational awareness, or (3) science. The metrics for warning applications include the timeliness of the warning, the rate of generation of false positives, and the accuracy of the estimation of the magnitude of the impact at each location and time. The metrics for ongoing situational awareness are similar, with the goal of detecting changes to an unfolding situation. Usually, first responders react in minutes whereas an elevator can be slowed down or a gas valve can be shut in seconds. Generally, constraints on the timeliness for situational awareness are less acute than for early warning. Science applications use data streams for data mining rather than event detection. For example, geologists are interested in tiny earthquakes that aren't felt by anybody; the detection and characterization of such quakes can be carried out days after the event.

2.4 Models of Sensors

A sensor of type s that experiences a manifestation H at time t measures a manifestation $a_s(H)$. The functions a_s, for example, maps true acceleration to measured acceleration for an expensive 20-bit accelerometer will be more accurate and more precise (i.e., closer to the identity function) than the function for a 10-bit accelerometer in a phone. Similiarly, sensors may report incorrect timestamps or they may not be located where designers think they are. Errors in location, timing, and measurement are captured by a_s that impact the efficacy of GDEBS applications.

A sensor may send continuous measurements periodically to event processing engines (which are servers in the Cloud in our implementation). Communication bandwidth is re-


T
X
sensor events
(x_0, t_0)
T
X
2e
true impact
possible impact reported due to timing error
T
X
sensor timelines
true sensor event
false sensor event
s_1 s_2 s_3 s_4 s_5 s_6
T
X
geospatial event not detected
geospatial event detected
s_1 s_2 s_3 s_4 s_5 s_6
T
X
false geospatial event detected due to false local detections in this cone
s_1 s_2 s_3 s_4 s_5
T
X
slow
fast
less uncertain
more uncertain

(a) Errors in reported time and location

(b) Impact of sensor timing error

(c) Timing error and sensor noise

(d) True positive and false negative

(e) False detection

(f) Impact of event speed

Figure 4: Sensor detection models illustrated in space-time plots

duced by having sensors send event messages to processing engines only when sensors detect events — local anomalies. An event message sent by the sensor has bits that identify the anomaly, the timestamp of the anomaly, and possibly measurements made during a short interval around the anomaly. The size and the frequency of event messages play a crucial role in the cost of the system and on its efficacy.

Fig. 4(a) shows the difference between the true impact of an event shown by the straight lines of a cone and the measured impact shown by the dots identifying anomalies detected by sensors. Some sensors may not detect anomalies and the number of sensors near the source of the event may be small, and as a consequence the problem of identifying the geospatial event from the sensor readings (i.e. identifying the lines of the cone from the dots) can be difficult.

Sensors may detect anomalies that are not associated with geospatial events. For example, an accelerometer may detect an anomalous situation when a heavy truck passes nearby. Our experiments show that sensors also generate erroneous signals due to electronic noise. The detection of an anomaly by a sensor is called a sensor event. Thus a sensor event may be due to a true geospatial event or due to local activity or due to noise. We generally assume that sensors generate noisy sensor events in a random Poisson manner (though measurements show that a few sensors generate noise events in more predictable ways).

By choosing sensor parameters, sensors can be sensitive to small signals, or relatively insensitive and detects only large signals. Setting parameters so that sensors are more sensitive results in sensors generating more noisy sensor events. The optimum settings of sensor parameters will depend on the network constraints and detection algorithm limitation.

2.5 Accuracy of Geospatial Event Detection

Next, we discuss issues that impact the accuracy and speed of geospatial event detection in terms of simple geometrical concepts.

Consider a simple model, that assumes that we know the precise speed of propagation of an event, i.e., the angle of the cone. Errors in timestamps are represented by a time interval $[-e, +e]$, where the probability of errors outside this interval are low. The timing error is represented by a band around the cone with vertical thickness $2e$ (Fig. 4(b)).

Consider a geospatial event that is initiated at time t at location x. The measured impact of this event is represented by a cone of thickness $2e$ with apex (x, t). A sensor event generated because a sensor detects the local manifestation of a geospatial event is a true (local) positive, while a sensor event generated for other reasons (such as local activity or electronic noise) is a false positive. In Fig. 4(c) sensor events that are true positives are shown as crosses while false positive are shown as dots.

A simple algorithm to detect geospatial events is as follows: detect a geospatial event when the number of sensors reporting events within a cone exceeds some multiple of the number of sensors that don't report events in that cone. We don't know the position of the apex of the cone, and therefore we consider all possible values of (x, t) for the location of the apex. (We use faster methods in our implementation). Fig. 4(d) shows two cones where the apex of each cone represents a true geospatial event. All the sensors generate sensor events in one of the cones, and no sensors generate events in the other cone. A geospatial event will be detected for the first cone — a true positive — but not for the second — which would be a false negative. Fig. 4(e) shows a false positive detection of a geospatial event because sensors generated detections due to local noise in a cone.

The rate of false positives increases with: uncertainty about the propagation of geospatial events, constraints on the speed of detection, and noise in sensors. Uncertainty about propagation speed is represented by cones with greater thickness in Fig. 4(f). Constraints on the speed of detection are represented by cones of shorter height, since the height of the cone represents the time from event initiation to detection; the number of sensors that can produce signals within a cone decreases with the height of the cone, and so detecting collections of true sensor events from false sensor events gets more difficult.

3. EVENT DETECTION AND ESTIMATION

Sensor activity, without true geospatial events, may vary over time. For example, accelerometers in homes and offices generally show more acceleration during the day when people and equipment are moving about than they do at night. Therefore, sensors detect local anomalies based on analysis of activity in a time window from the current time to a time in the past. In the Community Seismic Network[2], sensors detect a local anomaly if a measured value is more than K standard deviations away from the mean (we call such an event a “pick”), where the standard deviation and mean are estimated for the time window. The value of K is set to control the rate at which sensors send event messages to the event processing engine.

Sensors may be distributed across space in different ways. Ideally, sensors would be placed to optimize detection given prior knowledge of the kinds of geospatial events and the locations of their initiation points. In the case of community networks, members of the community deploy sensors in their homes and offices, and in this case sensor distribution depends on population distribution. We begin by studying two distributions: (a) sensors distributed in clusters in larger cities and towns and (b) sensors distributed in a uniform grid. Comparison between the two extreme distributions provides insight into the impact of sensor distributions on GDEBS applications.

3.1 Detection with Clustered Sensors

We first consider the case of sensors placed in a single cluster. The current community seismic network has sensors placed around a city with about 100 sensors distributed across a 10 km^2 area. We present a simple model for event detection that ignores spatial placement of sensors and treats all the sensors in a cluster as being at the same location, i.e. all sensors in the cluster detects equal amount of signal from an event but with different local noise profile.

Let T be the length of the interval during which the sensor determines whether it should generate an event. In the case of seismic applications, T is the time for a quake to travel across all the sensors in the cluster plus additional time to account for errors in sensor event timestamps. For example, when the cluster lies within a 10 km^2 region, the value of T is approximately 2 seconds.

Sensor Noise and True Events. Let λ_j be the rate at which sensor j generates noise events (i.e., the rate at which sensor events occur when there is no ongoing geospatial event). Assuming that sensor noise events are random, the probability q_j that sensor j will generate a noise event in an interval of duration T is:

$$q_j = 1 - e^{-\lambda_j T}$$

Let a be the manifestation of a geospatial event and let $p_j(a)$ be the probability that a sensor will generate an event in an interval T due to experiencing a passing seismic wave of magnitude M. In general $p(a)$ is monotone increasing with a. Then q_j is the probability of a noise event, and $p_j(a)$ is the probability of a true event given a, for sensor j.

Probability of a geospatial event from cluster data. Let V be a vector of sensor events in a certain duration of time; $V[j] = 1$ if sensor j generated an event; $V[j] = 0$ otherwise. Let α be the prior probability of the occurrence of an event in the given amount of time. Denote random variable E as an event. Assuming sensor measurements are independent, the probability $P(E = 1|V)$ of the presence of a geospatial event given V, and the probability $P(E = 0|V)$ of the absence of an event given V are:

$$\mathbb{P}[E = 1|V] = C\,\alpha \prod_{\{j|V[j]=1\}} p[j] \prod_{\{j|V[j]=0\}} (1 - p[j])$$

$$\mathbb{P}[E = 0|V] = C\,(1 - \alpha) \prod_{\{j|V[j]=1\}} q[j] \prod_{\{j|V[j]=0\}} (1 - q[j])$$

where C is a constant of proportionality so that

$$\mathbb{P}[E = 1|V] + \mathbb{P}[E = 0|V] = 1$$

Therefore,

$$C = \alpha \prod_{\{j|V[j]=1\}} p[j] \prod_{\{j|V[j]=0\}} (1 - p[j]) + (1 - \alpha) \prod_{\{j|V[j]=1\}} q[j] \prod_{\{j|V[j]=0\}} (1 - q[j])$$

Let

$$\beta = \frac{1-\alpha}{\alpha},\ r[j] = \frac{q[j]}{p[j]},\ s[j] = \frac{(1 - q[j])}{(1 - p[j])}$$

then

$$\mathbb{P}[E = 1|V] = \left(1 + \beta \prod_{\{j|V[j]=1\}} r[j] \prod_{\{j|V[j]=0\}} s[j]\right)^{-1}$$

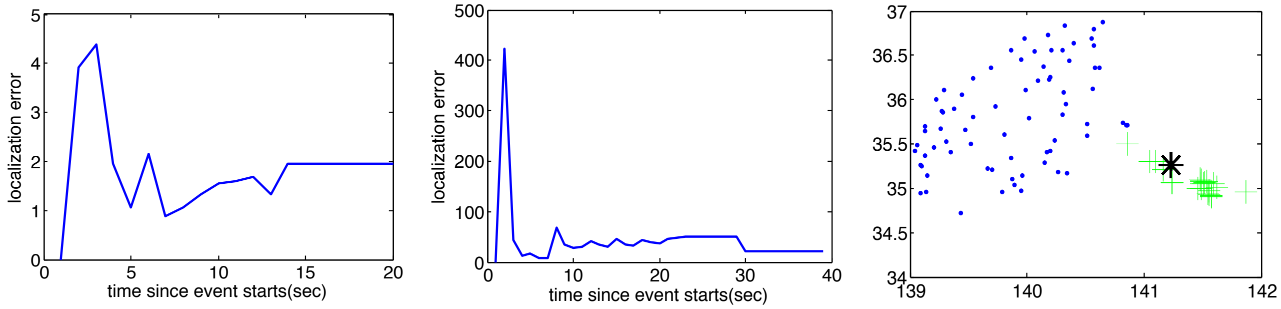
which is calculated rapidly in the cloud while maintaining adequate numerical accuracy. Grouping some terms into γ, then

$$\mathbb{P}[E = 1|V] = \frac{1}{1+\gamma}$$

γ can be calculated rapidly after $\log(r[j])$ and $\log(s[j])$ have been computed by summing the logs.

$$\log(\gamma) = \log(\beta) + \sum_{\{j|V[j]=1\}} \log(r[j]) + \sum_{\{j|V[j]=0\}} \log(s[j])$$

Speeding up the computation. Sensors can be put into categories based on their values of $p[j]$ and $q[j]$ which enables precomputations of $\log(r[j])$ and $\log(s[j])$ for each category. Let M be a vector where $M[i]$ is the number of sensors in category k for $k = 1, \ldots, K$, and let the values of p and q for category k be $p[k]$ and $q[k]$. Let W be a vector



(a) Inland Event

(b) Offshore Event

(c) Offshore epicenter estimates

Figure 5: Simple arrival time algorithm for event localization with earthquake data from Japan. (a) inland epicentre, (b) offshore epicentre, (c) blue dots: sensors, black cross: epicenter identified by JMA, green crosses: epicenter determined by the simple algorithm in each time step.

Table 1: Seismic events of $M \geq 3$ in southern California since September 2011.

LOCATION	MAGNITUDE	DIST(Km)	IMPACT
Newhall	4.2	37.7	80
Yucaipa	4.1	100	15
Irvine	3.5	65	7
Saugas	3.3	46.9	7
Ontario	3.5	52.6	9

Table 2: Detection probability jumps to one as wave comes in.

TIME (sec)	703	707	711	715	719
$\mathbb{P}[E=1\vert V]$	10^{-40}	10^{-37}	**1.0**	10^{-22}	10^{-37}

where $W[k]$ is the count of the number of sensors in category k that "picked" in an interval of duration T. Then

$$\log(\gamma) = \log(\beta) + \sum_{k} \left[W[k]\log(r[k]) + (M[k]-W[k])\log(s[k])\right]$$

Statistics. Tab. 1 shows earthquakes of magnitude 3 or higher within 100 km of the cluster since September 2011. During this period the network had at least 30 sensors. The impact is the manifestation of the event measured in terms of relative acceleration according to the Kanamori model[3].

All these events were detected by the system, in seconds after the wave reached the cluster. No false alerts were detected except for a strong clap of thunder on December 15, 2011 for which the system reported a geospatial event with probability 0.69. Tab. 2 shows the probability of a geospatial event reported by the system immediately before and after an event. The posterior probability quickly jumps from almost 0 to 1 as the wave arrives.

3.2 Characterization with Sparse Sensors

Next, we describe a simple arrival time based algorithm to estimate the location of the origin of a geospatial event. The model ignores the vertical coordinate of the event initiation location and it assumes that the event propagation velocity is fixed. These assumptions are necessary for a generalized discussion but simplistic. For example, the depth at which strain in the earth is first released impacts how waves propagate through the earth, and velocity of the wave changes quite significantly with distance.

Let (x_0, y_0, t_0) be the event initiation point in the space-time dimension. The time t_j at which the event first reaches a sensor j at location (x_j, y_j) is $t_j = \frac{d_j}{v} + t_0$, where d_j is the distance of sensor j from the event origin and v is the velocity of the propagation. Therefore:

$$t_j = \frac{\sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}}{v} + t_0$$

If sensor j generates a pick corresponding to observation of the event, let the time at which it generates the pick be $\hat{t_j}$. The error e_j in the time that sensor picks, according to this model is: $e_j = \hat{t_j} - t_j$ Let z be the sum of error squared, $z = \sum_j e_j^2$. Our goal of this analysis is to compute (x, y, t) to minimize z given the set $(x_j, y_j, t_j), j = 1, \ldots, n$ of sensor measurements while discarding outlier picks.

A local minimum of z is the solution to the nonlinear equations:

$$\sum_j \frac{(t_j - \frac{d_j}{v} - t)(x_j - x_0)}{d_j} = 0$$

$$\sum_j \frac{(t_j - \frac{d_j}{v} - t_0)(y_j - y_0)}{d_j} = 0$$

where $t = t_j - \frac{d_j}{v}$.

Case Study. We analyzed two independent events from publicly available seismic data from Japan, one with an inland epicenter (Fig. 5(a)), the other with an offshore epicenter (Fig. 5(b)). These high quality sensors are roughly 20 km apart. Fig. 5 shows the error in the estimate of the epicenter in kilometers based on this simple and relatively fast calculation compared with the epicenter calculated by geological services based on extensive data analysis and more sophisticated models [4, 5].

4. THEORETICAL ANALYSIS

Sensors in community networks are subject to higher levels of noise in comparison to professionally deployed networks. In addition to ambient noise and electronic noise,

noise also arises from errors in timing, reported sensor location, and measured signal strengths, to name a few. Given a network of noisy sensors, what can we say about the quality of the estimates of a geospatial event with that network? To answer this question we study how event and network parameters such as event propagation speed and magnitude, sensor numbers and quality, and timing errors affect detection and location estimates.

4.1 Preliminaries

The theoretical results in this section are based on the following assumptions.

- the event can be modeled as a point source with origin location x_0 and start time t_0
- the event propagates through the network at a constant speed v,
- the sensor timing error can be modeled as a Gaussian random variable $\mathcal{N}(0; \sigma)$,
- the prior probability distribution is uniform, i.e. the event can start anywhere in space-time,
- the n sensors in the network are semi-randomly distributed (e.g. they don't all lie on a single point or single line)

4.2 Detection Performance

Detection performance of geospatial events is measured by the tradeoff between true detection rate (TPR = $\mathbb{P}\left[\mathit{detect} = 1|E = 1\right]$) and false detection rate (FPR = $\mathbb{P}\left[\mathit{detect} = 1|E = 0\right]$). For detection of geospatial events, sensors in the network can be spread out over a large area. It is unreasonable to calculate TPR and FPR for the whole network. Instead, we calculate those values over some unit area.

4.2.1 Sensor Model

In this analysis, we consider a sensor model $a_s(H)$ that closely models two types of real sensors. Suppose the sensors are randomly distributed over the infinite plane such that on average there are d sensors per km^2. We model the sensor detection probability $d_i = 1$ as a function of distance to the source r_i.

$$\mathbb{P}\left[d_i = 1|E = 1\right] = \alpha\, e^{-\beta r_i^2} \qquad (1)$$

where α and β are functions of sensor quality and event magnitude and are specific to the type of geospatial event. Fig. 6(a) compares this model to experimental results for earthquake detection with two types of sensors of different quality — *Phidget* and *Android*.

4.2.2 TPR Bound

Let the detection metric be such that at least m out of n sensors detect within a certain amount of time t in an area of size $\pi R^2, R = vt$, where v is speed the event propagates at. Since the sensors are randomly distributed in an infinite field with density d, let this area be centered at the event origin location. The probability that exactly k sensors detect the event in t is

$$\mathbb{P}\left[K = k|E = 1\right] = \sum_{A\in F_k} \prod_{i\in A} p_i \prod_{j\in A^c} (1 - p_j)$$

where F_k is all the subset of size k of the set $\{1, 2, \cdots, n\}$, $n = \pi R^2 d$. A^c is the complement of set A. $p = \mathbb{P}\left[d = 1|E = 1\right]$. Since p is different for every sensor i depending on its distance r_i to the event, this is a Poisson Binomial Distribution (PBD) with expectation

$$\mathbb{E}\left[K\right] = \sum_i p_i = \int_{r=0}^{R} (2\pi r d) \left(\alpha e^{-\beta r^2}\right) \mathrm{d}r = \frac{\pi\alpha d}{b}\left(1 - e^{-\beta R^2}\right)$$

The network's true detection rate is then for $m < \mathbb{E}\left[K\right]$

$$\begin{aligned}\mathbb{P}\left[K \geq m|E = 1\right] &= 1 - \mathbb{P}\left[K < m|E = 1\right]\\ &\geq 1 - \exp\left(-2(\mathbb{E}\left[K\right] - m)^2/\pi R^2 d\right) \qquad (2)\end{aligned}$$

The inequality comes directly from the special case of the Chernoff/Hoeffding bounds for the PBD tail [6].

Similarly, the probability that exactly k sensors detect within the same amount of time from noise is

$$\mathbb{P}\left[K = k|E = 0\right] = \sum_{A\in F_k} \prod_{i\in A} \lambda_i t \prod_{j\in A^c} (1 - \lambda_j t)$$

$\lambda = \mathbb{P}\left[d = 1|E = 0\right]$. Assume that all sensors have the same noise rate λ. The false detection rate is for $m > \mathbb{E}\left[K\right]$

$$\begin{aligned}\mathbb{P}\left[K \geq m|E = 0\right] &\leq \exp\left(-2(\mathbb{E}\left[K\right] - m)^2/\pi R^2 d\right)\\ &= \exp\left(-2\left(\frac{\lambda d\pi R^3}{v} - m\right)^2/\pi R^2 d\right) \qquad (3)\end{aligned}$$

To ensure the maximum tolerable false positive rate $FPR = g \leq 1$ is not exceeded, set $\mathbb{P}\left[K \geq m|E = 0\right] \leq g$ and solve for m

$$m \geq \frac{\lambda d\pi R^3}{v} + R\sqrt{\frac{\ln(g)d\pi}{2}} \qquad (4)$$

Plugging (4) into Eq. (2), we acquire the lower bound on TPR. Setting $g = 1 \times 10^{-7}$ which is roughly equivalent to 1 false alarm a month, Fig. 6(b) shows how the TPR changes as the event speed increases in a fixed size region.

Optimal integration window size. Fig. 6(c) suggests that while keeping the speed constant, there is an optimal integration window size, or in other words, an optimal number of sensors considered in the calculation. This result may be counter intuitive since having more information should always improve the detection performance. Indeed, this result is an artifact from ignoring magnitude parameter in the Bayesian computation.

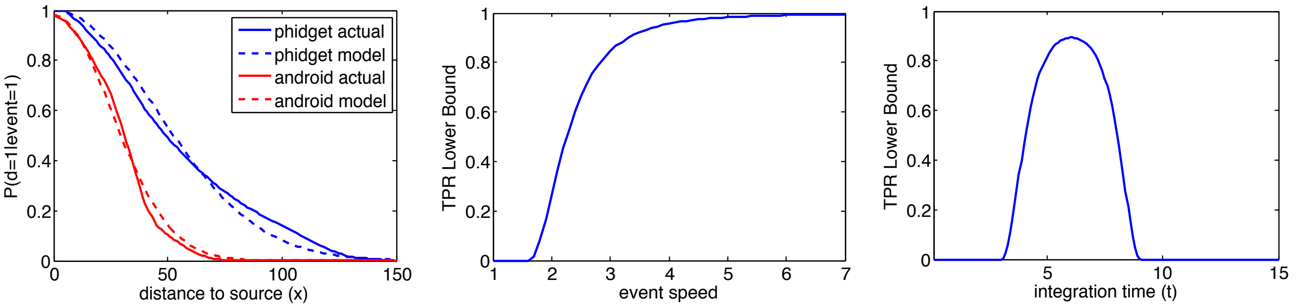
The probability $p_j(a)$ that a sensor j detects an event is determined by the manifestation of the event a and noise q_j.

$$\mathbb{P}\left[\mathit{detect} = 1|E = 1\right] = p_j(a) + q_j - p_j(a)q$$

$p_j(a)$ depends on the magnitude of the event. For a large event, the probability of an event detection is higher for larger window because the impact propagates over a larger region. Likewise the probability is higher for shorter window when the magnitude is low because of the impact falls off and the difference between the impact and noise cannot be differentiated. An optimal integration window size can be derived by optimizing (2) for a specific sensor layout and magnitude. Since magnitude information can't be assumed a priorly, a good choice of window size is the one that optimizes (2) for the minimum magnitude one wishes to detect.

4.3 Parameter Estimation Accuracy

Event parameters such as origin location and time can be elegantly estimated with Bayes estimation theory. The



(a) Attenuation model

(b) Increasing speed

(c) Increasing integration time

Figure 6: (a) Sensor detection probability - comparison between the model and experimental results from synthetic data on detection of a large event. (b) TPR lower bound for increasing event velocity $R = 50$, $d = 0.006$, $\lambda = 0.01$. (b) TPR lower bound for increasing integration time window $v = 5.5$, $d = 0.014 \approx 100$ sensors, $\lambda = 0.01$.

distribution of the posterior probability measures the quality of the estimates. The "sharper" the distribution is, or equivalently, the lower the variance of the distribution, the more confidence we can have on the estimates. The results presented here give a lower bound on the variance for Bayes estimates in terms of the number of sensors, timing error, and the speed of the event propagation.

4.3.1 *Bayesian Estimation Algorithm*

Given that an event has been detected, the posterior probability of event parameters can be computed with a list of sensor detection data around the detection time. The Bayes posterior distribution of the event origin (x) and origin time (t) is:

$$\mathbb{P}\left[x, t|\cdot\right] \propto \prod_{i=1}^{n} \left[\mathbf{1}_i p_i + (1 - \mathbf{1}_i)(1 - p_i)\right]$$

$\mathbf{1}_i$ is the indicator function. $\mathbf{1}_i = 1$ when sensor i at location s_i has at least one detection for the event, propagating at speed v, in the time interval

$$\left[t + \frac{\|s_i - x\|}{v} - k,\ t + \frac{\|s_i - x\|}{v} + k\right]$$

k is the window width. p_i is the probability of detection for a sensor at distance r_i from the event. $p_i = p(a_s(H))$.

4.3.2 *Estimate Variance Bounds*

Assuming that the magnitude of the event is large enough such that for all sensors, $p_i = 1, \forall i = 1, \ldots, n$ and the detection threshold is high enough that there are no detections due to noise, we have the following results for the variance of location and time estimates.

Theorem 1. *Under the aforementioned assumptions, the variance of the Bayes posterior probability for event location estimate x and event origin time estimate t is bounded below in terms of sensor timing error (σ), number of sensors (n), and the speed the event travel at (v).*

$$Var\left[x\right] \geq \sqrt{\frac{2\pi}{n}}\ \frac{(4\sigma^2 + nt^2)v^2}{2^{n+3}\sigma} \tag{5}$$

$$Var\left[t\right] \geq \sqrt{\frac{2\pi}{n}}\ \frac{4\sigma^2 v^2 + nx^2}{2^{n+3}\sigma v^3} \tag{6}$$

These results show that the quality of the location estimate is influenced by timing error ($Var\left[x\right] \propto \sigma$) and how fast the event travels ($Var\left[x\right] \propto v^2$). The same factors affect the quality of the time estimate ($Var\left[t\right] \propto \sigma$), but the error in timing actually improves for fast events $Var\left[t\right] \propto 1/v$.

To prove these results, we first state the following lemma with its proof included in the appendix.

Lemma 1. *Let $f(x)$ be the probability density function of a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Let $G(x)$ be the area under curve of $f(x)$ in the interval $[x - m\sigma,\ x + m\sigma]$, $m \geq 0$ then*

$$G(x) = \int_{x-m\sigma}^{x+m\sigma} f(x)\,\mathrm{d}x = \int_{x-m\sigma}^{x+m\sigma} \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

$G(x)$ is bounded below and above

$$C\ e^{-(x-\mu)^2/2\sigma^2} \leq G(x)\ \leq\ 2\ C\ e^{-(x-\mu)^2/4\sigma^2} \tag{7}$$

where $C = m\sqrt{\frac{2}{\pi e^m}}$.

Proof of Theorem 1. If t^i is the actual time when a sensor i first detects the event and $\hat{t^i}$ is the timestamp it reports, then the timing error $e = \hat{t^i} - t^i \sim \mathcal{N}(0; \sigma)$, assuming the error can be modeled as Gaussian. Let $k(\cdot)$ be the time difference between $t^i_{x_0,t_0}$ (expected sensor detection time for an event at (x_0, t_0)) and $t^i_{x,t}$ (expected sensor detection time for an event at (x, t))

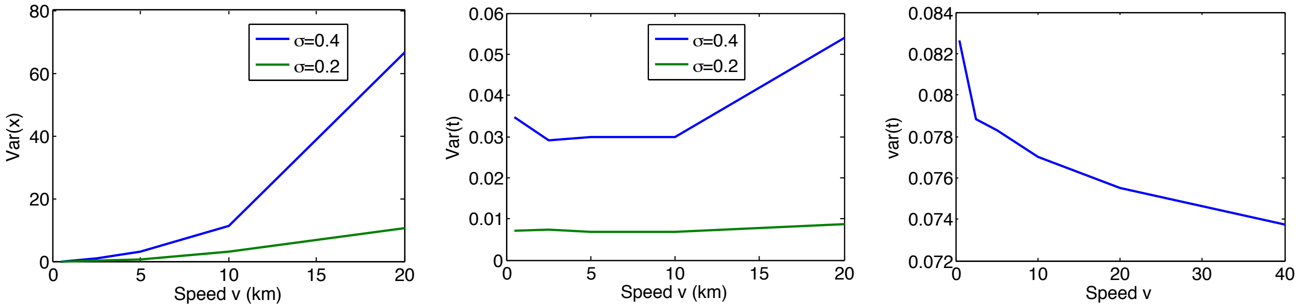
$$k(s_i, x, t) = t^i_{x,t} - t^i_{x_0,t_0} = \frac{\|s_i - x\| - \|s_i - x_0\|}{v} + (t - t_0)$$

where s_i is the sensor location, and

$$0 \leq |k| \leq \frac{\|x - x_0\|}{v} + |t - t_0| \tag{8}$$

The probability that p is the detection time for an event (x, t) is the probability that q falls between the interval $[k - m\sigma, k + m\sigma]$ which is a window of arbitrary width $2m\sigma$ centered at $k(\cdot)$. Let's call this probability $G(k)$. Assuming a uniform prior, the posterior probability is then

$$\mathbb{P}\left[x, t|\cdot\right] = \frac{\mathbb{P}\left[\cdot|x, t\right]\mathbb{P}_0\left[x, t\right]}{\int \mathbb{P}\left[\cdot|x, t\right]\mathbb{P}_0\left[x, t\right]\mathrm{d}(x, t)} = \frac{1}{S}\mathbb{P}\left[\cdot|x, t\right] = \frac{1}{S}\prod_{i=1}^{n} G(k_i)$$



(a) (b) (c)

Figure 7: Simulation results from 10,000 runs with 16 sensors (a) variance in Bayes location estimates (b) variance in Bayes time estimates (c) variance in Bayes time estimates with fixed x

Without loss of generality, let $x_0 = (0,0,0)$, $t_0 = 0$, $m = 1$. Assuming the sensors are semi-randomly placed, from Eq. (8), $G(k_i)$ can be approximated with $G(k)$, $k = \frac{1}{2}\left(\frac{|x|}{v} + |t|\right) \geq 0$.

$$\mathbb{P}\left[x,t|\cdot\right] \approx \frac{1}{S}\left[G(k)\right]^n \geq \frac{1}{S}\left(\frac{2}{\pi e}\right)^{\frac{n}{2}} e^{-\frac{nk^2}{2\sigma^2}}$$

The inequality comes directly from the lower bound in Lemma 1. To simplify the notation, denote $x = |x|$ and $t = |t|$. S is the normalizing factor and can be computed by integrating over all $x \geq 0$ and $t \geq 0$. Using the upper bound in Lemma 1 and substituting k,

$$\begin{aligned} S &= \int_0^\infty \int_0^\infty \left[G(k)\right]^n \, \mathrm{d}x \, \mathrm{d}t \\ &\leq \int_0^\infty \int_0^\infty 2\left(\frac{2}{\pi e}\right)^{\frac{n}{2}} e^{-\frac{nk^2}{4\sigma^2}} \, \mathrm{d}x \, \mathrm{d}t = \left(\frac{2}{\pi e}\right)^{\frac{n}{2}} \frac{2^{(n+3)}\sigma^2 v}{n} \end{aligned}$$

Substituting S and k into (9), we get

$$\mathbb{P}\left[x,t|\cdot\right] \geq \frac{n}{2^{(n+3)}\sigma^2 v} \exp\left(-\frac{n}{8\sigma^2}\left(\frac{x}{v}+t\right)^2\right)$$

With a simple calculation, we know $\mathbb{E}\left[\frac{x}{v}+t\right] = 0$. Since $x \geq 0$ and $t \geq 0$, we have $\mathbb{E}\left[x\right] = 0$ and $\mathbb{E}\left[t\right] = 0$ as expected. The variance of x and t can be computed as

$$\begin{aligned} Var\left[x\right] = \mathbb{E}\left[x^2\right] - \mathbb{E}\left[x\right]^2 &= \int_0^\infty x^2 \, \mathbb{P}\left[x,t|\cdot\right] \, \mathrm{d}x - 0 \\ &\geq \frac{n}{2^{(n+3)}\sigma^2 v} \int_0^\infty x^2 \exp\left(-\frac{n}{8\sigma^2}\left(\frac{x}{v}+t\right)^2\right) \, \mathrm{d}x \\ &\geq \frac{n}{2^{(n+3)}\sigma^2 v}\left[\frac{1}{2}\int_{-\infty}^\infty x^2 \exp\left(-\frac{n}{8\sigma^2}\left(\frac{x}{v}+t\right)^2\right) \, \mathrm{d}x\right] \\ &= \sqrt{\frac{2\pi}{n}} \frac{(4\sigma^2 + nt^2)v^2}{2^{n+3}\sigma} \end{aligned} \qquad (9)$$

The second inequality comes from the observation that the exponential function is symmetrical and centered at a negative value. Therefore the integral from 0 to ∞ is smaller than the integral from $-\infty$ to 0. Similarly

$$Var\left[t\right] \geq \sqrt{\frac{2\pi}{n}} \frac{4\sigma^2 v^2 + nx^2}{2^{n+3}\sigma v^3} \qquad (10)$$

□

5. EVENT DETECTION WITH MULTIPLE SERVERS IN THE CLOUD

The ability to easily utilize multiple servers is one of the benefits of working with Cloud platforms, but it poses obstacles for conventional event detection strategies. Most event detection algorithms, including those proposed in this work, rely on aggregated state. This does not necessitate centralized processing, but does require that information be shared between disparate nodes in a processing network. That is, computing the result of these algorithms cannot rely on information private to an individual processing node. Algorithms which rely on incomplete information about the network will be explored in a future work.

Our solution to the creation, retrieval, and maintenance of shared information relies on the use of Geocell objects[7] to separate the state needed for a given region into disparate objects. This technique is analogous to counter sharding[8]; contention over shared state is managed by splitting that state into related objects that can be easily aggregated. This makes the performance for access to shared state manageable even under the real time processing constraints our target applications require.

A further obstacle with distributed servers is the issue of resolving the time at which events occur. As time is a critical component for event detection, any algorithm that operates on the Cloud must determine how it is going to achieve an accurate estimate of event times. In the CSN network, we achieve this by operating a time server that all clients must coordinate with. We then accept client times, rather than server times, as the true time. This does not imply that client times are always accurate; rather, it is a reflection of the fact that we do not control the system time on Cloud servers, and, as servers are constantly moving in and out of usage, we cannot accurately estimate the time drift of a single server. This is one reason why our algorithms allow for time inconsistency. A forthcoming publication will specifically address the amount of time variation we experience and tactics for managing that error.

6. EXPERIMENTS

We carried out Monte Carlo simulation experiments for two types of sensor distributions: (1) idealized uniform with event initiation in the center of the network, and (2) distribution according to real population distribution in South-

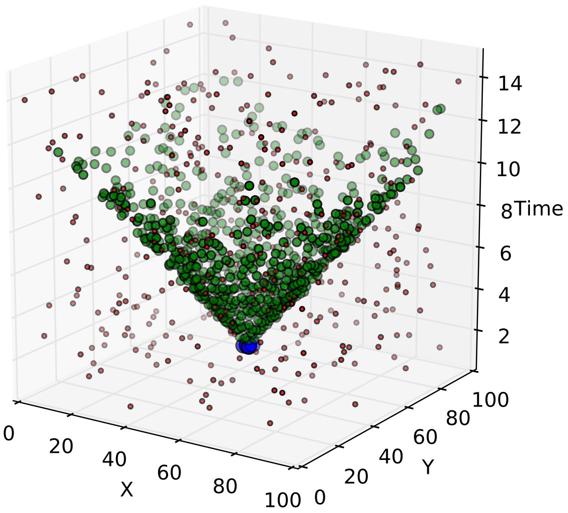


Figure 8: An entire simulation with a uniform sensor distribution. Green dots mark sensor detection due to the event; red dots mark detection due to noise.

ern California. In both sets, we assumed an isotropic event model as discussed in Sec. 2.2 and a sensor model as in Sec. 2.4 with a measurement-based signal attenuation model. The detection and estimation algorithm were described in Sec. 3.2.

6.1 Effect of Event Propagation Speed

We ran simulations with $n = 16$ uniform sensors in a region of size of 100x100 km^2. In each simulation, the Bayesian posterior distribution for x and t was computed. The average of variances across the simulations is shown in Fig. 7 for increasing event speed v. From Fig. 7(a) and Fig. 7(b), it is clear that the estimates are negatively affected by increased sensor timing error from $\sigma = 0.2$ to $\sigma = 0.4$ as Theorem 1 suggests. Moreover, $Var[x]$ grows as v^2 (Fig. 7(a)) whereas $Var[t]$ decreases as v for a fixed location (Fig. 7(c)). However, the overall variance in time estimates still grows with v since the location estimate gets increasingly less accurate more quickly as v gets larger.

6.2 Uniform Distribution

Again with uniformly distributed sensors, simulations were run with a variety of parameters to test the effects of different variables. All of the simulation results discussed operated with a region size of 100x100 km^2, a potential timing error of standard deviation $\sigma = 1$ second, sensor noise rate at 1 per 100 seconds ($\lambda = 0.01$), and an unknown randomly chosen seismic wave speed between $v = 5 - 6$ km/sec. The simulation set utilizes a uniformly distributed set of sensors with an event initiation in the center of the region. We varied the number and distribution of sensors, the strength of the earthquake, and the size of the search space (integration window) to find changes in the true positive rate (TPR) and location error (LocErr, expressed in kilometers). Since it requires a large number of simulation rums to acquire a reliable false alarm rate (FPR), we derive them mathematically. Fig. 8 is a space-time plot that helps visualize the sensor detections during an event.

Tab. 3 shows a subset of results from these runs. Aside from the first row — 100 sensors with 2-second integration window — that has a hight FPR of 1.0×10^{-5}, all other configurations have FPR close to 0. These results show that while the location estimates improve as the integration window size increases, the TPR decreases, but a small window size has poor FPR performance. This was well predicted in Sec. 4. Note that the TPR values in column 4 (highlighted column) shows the same trend as in Fig. 6(c), as explained in Sec. 4.2.2.

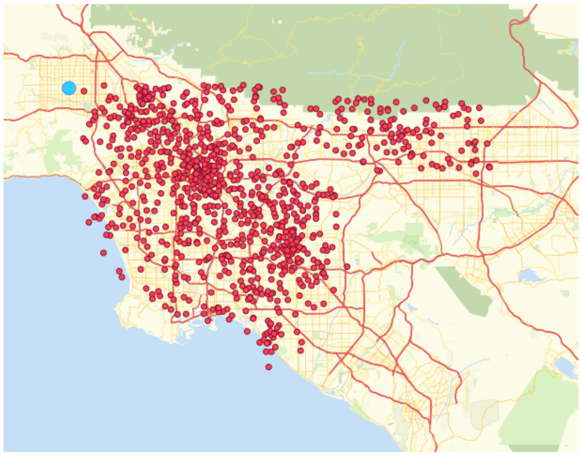


Figure 9: 1,000 sensors drawn from a basic model based on population density in southern California.

Table 3: Uniform distribution

# Sensors	M	Integration Window	**TPR**	**LocErr**
10x10	2	2	0.90	16.0
10x10	2	4	1.00	5.0
10x10	2	6	0.97	3.0
10x10	2	8	0.62	3.5
10x10	2	10	0.09	3.3
25x25	2	2	1.00	2.0
25x25	2	4	1.00	1.4
25x25	2	6	1.00	1.2
25x25	2	8	0.97	3.5
25x25	2	10	0.02	3.6

6.3 Population-Based Distribution

In this simulation set, we used a sensor distribution model based on population density in Southern California. We then placed the earthquake origin at the epicenter of the historic 1994 Northridge earthquake, just outside the network (illustrated in Fig. 9).

These simulations utilized 300 sensors in the same region size and timing error as before, but increased the sensor noise rate ($\lambda = 0.1$). FPR estimates were also derived for this distribution and noise rate. The value is approximated to be at most 1.0×10^-7 for all runs. TPR results in Tab. 4 are comparatively lower than in Tab. 3 because of the out-of-network effect. However, the algorithm is still capable to detect a medium size event 80% of the time with a small number of sensors ($N = 300$). It should be noted that the location errors in Tab. 4 are computed differently from those in Tab. 3 and should not be compared together.

Table 4: Population density distribution

# Sensors	M	Integration Window	**TPR**	**LocErr**
300	4	2	0.30	0.3
300	4	4	**0.81**	0.8
300	4	6	0.73	0.7
300	4	8	0.21	0.2
300	4	10	0.06	1.6

7. RELATED WORK

Geospatial Event Detection. Detection of radiation sources with sensor networks has been widely studied in terms of theory and algorithms [9, 10, 11, 12]. While this is a simple form of geospatial event (speed = ∞), it affords insight into methods of detection and estimation formulation for geospatial events. For other classes of geospatial events, in earthquake detection, [13] looks into the development of a self-organizing wireless mesh information network made up of low cost sensors, for the purpose of providing earthquake early warning in Europe. [14] studies the feasibility of a fast early warning system inspired by recent dense accelerometer sensor network deployments. While the goals of fast and accurate detection are similar, these works do not address the problem with the same fidelity for a highly heterogeneous network managed by volunteers. Similar work in geospatial event detection includes wildfire detection using mesh networks [15], tsunami detection with radar arrays [16], and flood detection and prediction [17], most of which do not generalize to other geospatial problems.

Community and Participatory Sensing. There are a growing number of projects that aim to take advantage of sensors owned and operated by citizen scientists to aid in research — this is facilitated by advances in, and availability of, inexpensive sensing technology, e.g.[18, 19]. These applications stand to be benefit from greater sensor densities, but their general aim is to monitor ongoing phenomena rather than to detect rare events with a low false positive rate requirement. [20] and [7] took the initiative by studying such problems using more simplistic models for event and sensor behavior that do not give sufficient accuracy or intuition for realistic network constraints such as timing error and event speed.

8. CONCLUSION

We describe a general model for detection of geospatial events with realistic networks of sensors. Based on this model, we develop robust Bayesian algorithms that can be computed efficiently on distributed cloud servers. The algorithms are supplied with theoretical analysis that bounds the detection and parameter estimation performance in terms of true positive rate, false alarm rate, and variance in event initiation location and time estimates. The theoretical results are verified by simulation experiments, using seismic events as an example, both in an idealized setting with uniform sensor distribution and a more realistic setting with sensors distributed according to population in Southern California. The results show that with only 300 sensors we can, with good confidence, detect an event smaller than the Northridge earthquake in 4 seconds. This would provide ample time to warn the populace outside of the immediate area of the epicenter.

9. ACKNOWLEDGMENT

The authors would like to thank the following Caltech collaborators who are all working to make the Community Seismic Network a success: Prof. Robert Clayton and Prof. Jean-Paul Ampuero from the Seismo Lab; Prof. Tom Heaton, Dr. Monica Kohler, and Ming-Hei Cheng from Earthquake Engineering; Prof. Andreas Krause and Matthew Faulkner from Computer Science; Michael Aivazis, and Leif Strand from the Center for Advanced Computing Research. This research is supported in part by the National Science Foundation Cyber-Physical Systems program.

10. REFERENCES

[1] S. Das, K. Kant, and N. Zhang, *Handbook on Securing Cyber-Physical Critical Infrastructure*. Elsevier Science & Technology, 2012. [Online]. Available: http://books.google.com/books?id=yj15qJtTtf4C

[2] R. W. Clayton, T. Heaton, M. Chandy, A. Krause, M. Kohler, J. Bunn, R. Guy, M. Olson, M. Faulkner, M. Cheng, L. Strand, R. Chandy, D. Obenshain, A. Liu, and M. Aivazis, "Community Seismic Network," *Annals of Geophysics*, vol. 54, no. 6, Jan. 2012.

[3] T. Hanks and H. Kanamori, "A moment magnitude scale," *Journal of Geophysical Research*, 1979.

[4] R. M. Allen and H. Kanamori, "Rapid Determination of Event Source Parameters in Southern California for earthquake early warning," *AGU Fall Meeting Abstracts*, vol. -1, p. 0556, Dec. 2001.

[5] G. Cua, M. Fischer, T. Heaton, and S. Wiemer, "Real-time Performance of the Virtual Seismologist Earthquake Early Warning Algorithm in Southern California," *Seismological Research Letters*, vol. 80, no. 5, p. 740, Sep. 2009.

[6] D. Dubhashi and A. Panconesi, "Concentration of Measure for the Analysis of Randomized Algorithms, 1st edition," *Concentration of Measure for the Analysis of Randomized Algorithms, 1st edition*, Jun. 2009.

[7] M. Olson, A. Liu, M. Faulkner, and K. M. Chandy, "Rapid Detection of Rare Geospatial Events: Earthquake Warning Applications," in *DEBS '11: Proceedings of the Fifth ACM International Conference on Distributed Event-Based Systems*. ACM, 2011.

[8] J. Gregorio, "Sharding counters," http://code.google.com/appengine/articles/sharding_counters.html.

[9] K. Chandy, J. Bunn, and A. Liu, "Models and algorithms for radiation detection," in *Modeling and Simulation Workshop for Homeland Security*, Mar 2010, pp. 1–6.

[10] A. H. Liu, J. J. Bunn, and K. M. Chandy, "An analysis of data fusion for radiation detection and localization," in *Proceedings of the 13th International Conference on Information Fusion*, 2010.

[11] A. Liu, J. Bunn, and K. Chandy, "Sensor networks for the detection and tracking of radiation and other threats in cities," in *Proceedings of the10th International Conference on Information Processing in Sensor Networks (IPSN),*, april 2011, pp. 1 –12.

[12] N. Rao, M. Shankar, J. Chin, D. Yau, S. Srivathsan, S. Iyengar, Y. Yang, and J. Hou, "Identification of Low-Level Point Radiation Sources Using a Sensor Network," *Proceedings of the 7th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2008.

[13] J. Fischer, J.-P. Redlich, J. Zschau, C. Milkereit, M. Picozzi, K. Fleming, M. Brumbulli, B. Lichtblau, and I. Eveslage, "A wireless mesh sensing network for early warning," *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 538–547, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2011.07.016

[14] A. Zollo, O. Amoroso, and M. Lancieri, "A threshold-based earthquake early warning using dense accelerometer networks - Zollo - 2010 - Geophysical Journal International - Wiley Online Library," *Geophysical Journal*, 2010.

[15] Z. Chaczko and F. Ahmad, "Wireless sensor network based system for fire endangered areas," in *Third International Conference on Information Technology and Applications, 2005. ICITA 2005.*, vol. 2, july 2005, pp. 203 –207.

[16] M. Galletti, G. Krieger, T. Borner, N. Marquart, and J. Schultz-Stellenfleth, "Concept design of a near-space radar for tsunami detection," in *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, july 2007, pp. 34 –37.

[17] E. A. Basha, S. Ravela, and D. Rus, "Model-Based Monitoring for Early Warning Flood Detection," in *the 6th ACM conference.* New York, New York, USA: ACM Press, 2008, pp. 295–308.

[18] G. Drukier, E. Rubenstein, P. Solomon, M. Wojtowicz, and M. Serio, "Low cost, pervasive detection of radiation threats," in *2011 IEEE International Conference on Technologies for Homeland Security (HST)*, 2011, pp. 365–371.

[19] A. Krause, E. Horvitz, A. Kansal, and F. Zhao, "Toward community sensing," in *Proceedings of the 7th international conference on Information processing in sensor networks.* IEEE Computer Society, 2008, pp. 481–492.

[20] M. Faulkner, M. Olson, R. Chandy, J. Krause, K. M. Chandy, and A. Krause, "The Next Big One: Detecting Earthquakes and Other Rare Events from Community-based Sensors," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks.* ACM, 2011.

APPENDIX

A. PROOF OF LEMMA 1

PROOF. Let $F(x)$ be the cumulative density function of a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Without loss of generality, set $\mu = 0$, then

$$\begin{aligned} G(x) &= \int_{x-m\sigma}^{x+m\sigma} f(x)\,\mathrm{d}x \\ &= F(x+m\sigma) - F(x-m\sigma) \\ &= \frac{1}{2}\left[erf\left(\frac{x+m\sigma}{\sqrt{2\sigma^2}}\right) - erf\left(\frac{x-m\sigma}{\sqrt{2\sigma^2}}\right)\right] \end{aligned}$$

The erf function doesn't have closed form solution, however, we can derive an upper and lower bound that has one. Take the derivative of $G(x)$ with respect to x, we have

$$\begin{aligned} G'(x) &= \frac{-1}{\sigma\sqrt{2\pi e^m}}\, e^{-\frac{x^2}{2\sigma^2}} \left(e^{\frac{mx}{\sigma}} - e^{-\frac{mx}{\sigma}}\right) \qquad (11) \\ &= \frac{-1}{\sigma\sqrt{2\pi e^m}}\, e^{-\frac{x^2}{2\sigma^2}}\, g(x) \qquad (12) \end{aligned}$$

First we prove the lower bound. Observe that $g(x) = \left(e^{\frac{mx}{\sigma}} - e^{-\frac{mx}{\sigma}}\right)$ is smooth, is convex and positive $\forall x \geq 0$ and concave and negative $\forall x \leq 0$, then $g(x)$ and the line tangent to $g(x)$ at $x = 0$ is

$$h(x) = g'(0)\,x = \frac{2m}{\sigma}x$$

Note $sgn(g(x)) = sgn(h(x))$ and $|g(x)| \geq |h(x)|$. With this and substituting $h(x)$ for $g(x)$ in Eq. (12), we get a new function

$$H'(x) = \frac{-1}{\sigma\sqrt{2\pi e^m}}\, e^{-\frac{x^2}{2\sigma^2}} \left(\frac{2m}{\sigma}x\right) = \frac{-m}{\sigma^2}\sqrt{\frac{2}{\pi e^m}}\, x\, e^{-\frac{x^2}{2\sigma^2}}$$

and

$$\left|G'(x)\right| \geq \left|H'(x)\right| \qquad (13)$$

Integrating $H'(x)$, we get

$$H(x) = \int H'(x)\,\mathrm{d}x = m\sqrt{\frac{2}{\pi e^m}}\, e^{-\frac{x^2}{2\sigma^2}}$$

Observe that $sgn(G(x)) = sgn(H(x))$, then it follows from Eq. (13) that

$$G(x) \geq H(x) = m\sqrt{\frac{2}{\pi e^m}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Similarly we prove the upper bound. Rewriting Eq. (11)

$$\begin{aligned} G'(x) &= \frac{-1}{\sigma\sqrt{2\pi e^m}}\, e^{-\frac{x^2}{4\sigma^2}} \left[e^{-\frac{x^2}{4\sigma^2}} \left(e^{mx/\sigma} - e^{-mx/\sigma}\right)\right] \\ &= \frac{-1}{\sigma\sqrt{2\pi e^m}}\, e^{-\frac{x^2}{4\sigma^2}}\, g(x) \qquad (14) \end{aligned}$$

Observe that $g(x)$ is smooth, $g(-x) = -g(x)$, $\lim_{x\to-\infty} g(x) = \lim_{x\to\infty} g(x) = 0$. For $x \geq 0$, $g(x)$ is first concave then convex, then the line tangent to $g(x)$ at $x = 0$ is

$$h(x) = g'(0)\,x = \frac{2m}{\sigma}x$$

Note $sgn(g(x)) = sgn(h(x))$ and $|g(x)| \leq |h(x)|$. With this and substituting $h(x)$ for $g(x)$ in Eq. (14), we get a new function

$$H'(x) = \frac{-1}{\sigma\sqrt{2\pi e^m}}\, e^{-\frac{x^2}{4\sigma^2}} \left(\frac{2m}{\sigma}x\right) = \frac{-m}{\sigma^2}\sqrt{\frac{2}{\pi e^m}}\, x\, e^{-\frac{x^2}{4\sigma^2}}$$

Integrating $H'(x)$, we get

$$H(x) = \int H'(x)\,\mathrm{d}x = 2m\sqrt{\frac{2}{\pi e^m}}\, e^{-\frac{x^2}{4\sigma^2}}$$

Observe that $sgn(G(x)) = sgn(H(x))$ and since $|G'(x)| \leq |H'(x)|$, we have

$$G(x) \leq H(x) = 2m\sqrt{\frac{2}{\pi e^m}}\, e^{-\frac{(x-\mu)^2}{4\sigma^2}}$$

□