Catching Elephants with Mice: Sparse Sampling for Monitoring Sensor Networks

Sorabh Gandhi*
Department of Computer Science
UC Santa Barbara
sorabh@cs.ucsb.edu

Subhash Suri†
Department of Computer Science
UC Santa Barbara
suri@cs.ucsb.edu

Emo Welzl
Department of Computer Science
ETH Zurich
emo@inf.ethz.ch

Categories and Subject Descriptors

F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Nonnumerical Algorithms and Problems*

General Terms

Algorithms

Keywords

Sensor networks, monitoring, VC-dimension, ε-nets

Abstract

We propose a scalably efficient scheme for detecting large-scale physically-correlated events in sensor networks. Specifically, we show that in a network of n sensors arbitrarily distributed in the plane, a sample of $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$ sensor nodes (*mice*) is sufficient to catch any, and *only those*, events that affect $\Omega(\varepsilon n)$ nodes (*elephants*), for any $0 < \varepsilon < 1$, as long as the *geometry* of the event has a bounded Vapnik-Chervonenkis (VC) dimension. In fact, the scheme is provably able to estimate the size of an event within the approximation error of $\pm\varepsilon n/4$, which can be improved further at the expense of more mice. The detection algorithm itself requires knowledge of the event geometry (e.g. circle, ellipse, or rectangle) for the sake of computational efficiency, but the combinatorial bound on the sample size (set of mice) depends only on the VC dimension of the event class and not the precise shape geometry.

While nearly optimal in theory, due to implicit constant factors, these "scale-free" bounds still prove too large in practice if applied blindly. We, therefore, propose heuristic improvements and perform empirical parameter tuning to counter the pessimism inherent in these theoretical estimates. Using a variety of data distributions and event geometries, we show through simulations that the final scheme is eminently scalable and practical for large-scale network, say, with $n \geq 1000$. The overall simplicity and generality of our technique suggests that it may be well-suited for a wide class of sensornet applications, including monitoring of physical environments, network anomalies, network security, or any abstract binary event that affects a significant number of nodes in the network.

*This research was funded by the National Science Foundation under grants CNS-0626954 and CCF-0514738.

† This research was funded in part by the National Science Foundation under grants CNS-0626954 and CCF-0514738. The author also wishes to acknowledge the hospitality of the Institute of Theoretical Computer Science, ETH, Zurich, during his sabbatical when this research was performed.

SenSys'07, November 6–9, 2007, Sydney, Australia.

1 Introduction

Sensor networks are enablers of what has been called sensory ubiquity or omnipresence: tiny, inexpensive, untethered sensor devices can measure and observe various environmental parameters, often in hazardous or humanly inaccessible places, thereby allowing real-time and fine-grained monitoring of physical spaces around us. Many potential applications of this technology relate to surveillance [1] or environmental monitoring [6, 25, 29], necessitating large-scale networks spanning wide-spread geographical areas. In many of these applications, the phenomena of interest are *global* in the sense that they are not discernible at the level of individual nodes, and require corroborative input from many sensors—that is, events only become significant if sensed data of many nodes support them. Indeed, this issue of making global inferences from local data is characteristic of many distributed systems, but it takes on a uniquely geometric form due to the physical embedding of sensor networks. Let us begin by considering a few motivating examples to frame the context.

- Imagine a sensor network in an *environmental monitoring* application, e.g., to detect wild fires in a forest or to track pollution levels in a habitat [16, 19]. An abnormal or above average sensor reading at a single node, or even several but widely scattered nodes, is hardly

a cause for concern, as local and temporal variations are routine in nature. On the other hand, abnormal measurements by a large number of nodes concentrated in a *geographical neighborhood* suggest a significant event that may require immediate action.

- Due to the conditions of their physical deployment (outdoors, often hazardous and hostile environments), sensor networks are especially vulnerable to natural faults and adversarial attacks [30, 32]. Thus, maintaining a visibility into the *operational health* of the system is crucial. Given the physical conditions, isolated node failures or poor quality of wireless medium in some location is both expected and routine. On the other hand, a systemic failure that affects or partitions a large segment of the network is a significant event.

One can imagine many other examples of this kind: sudden *energy* drop among sensor nodes in a neighborhood, abnormal *congestion* in a region, correlated changes in the sensed data at nodes in a neighborhood, and so on. Since a centralized data collection approach, where a central processor continuously collects signals from all the nodes and learns the state of the network, does not scale well with the size and the complexity of large scale sensor networks, we seek more efficient solutions. In a nutshell, our approach will be to monitor only a small subset (*sparse sample*) of the nodes in the network, and *infer* the presence or absence of a significant event just from the signals received from this subset.

Our vision is that these sparse samples will act as a trigger system*, allowing most of the network to exist in a state of low alertness, to be "woken up" only when an event of interest is reliably detected.* In many settings, highly detailed sensor measurements are desirable only *after* a significant event occurs. Yet sensors may be forced to collect and communicate enormous quantities of data all the time, draining precious system resources, since events are often unpredictable. With our scheme, most of the network can essentially "sleep" or gather data at a very conservative rate, and only switch to the high-alert mode when our sparse samples sound an alarm. (We note that our work differs from the existing work on "sensor coverage" that attempts to choose a subset of sensors that cover the field of interest. Indeed, our sample nodes are too sparse to cover the field—they are only guaranteed to catch large-events.) Before we formally define our problem and its main parameters, let us continue with an informal description of the main technical issues.

1.1 Of Elephants and Mice

Two important features characterize the kinds of phenomena we aim to address in this paper: *relative scale* and *non-locality*. By relative scale, we mean that our focus is on phenomena that affect a given (user-specified) *fraction* of the network, rather than an absolute number of nodes. From the application side, this is meaningful in all the scenarios we mention: the event is considered significant only if a non-trivial fraction of the network is partitioned, or has a consensus value for the fire alarm, etc. Just as significantly, the relative threshold is also *necessary* mathematically to achieve a scalable efficient detection scheme: by focusing on ε fractions of the network, we will show a detection scheme whose efficiency depends only on ε, independent of the network size! On the other hand, it is easy to see that schemes that must detect all events affecting an *absolute* number of nodes require detection effort that grows *linearly* with the network size in the worst-case.

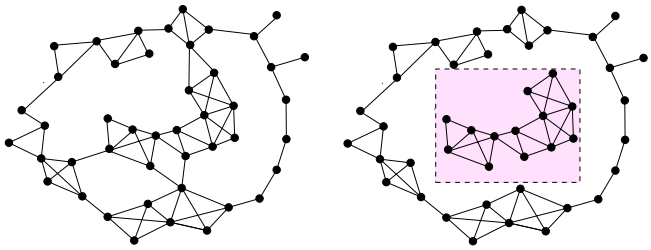


Figure 1. The figure shows that the global connectivity can change substantially without any significant change in the local connectivity of the nodes: the loss of just a few isolated edges can cause major network partitions.

The second key feature is the non-locality of the phenomena: events such as a network partition, a potential wild fire, a jamming attack etc. are best thought of as a signal that is *smeared* over a geographical neighborhood, not always discernible to individual nodes. For instance, consider the problem of network partition: individual nodes in the two components can be well-connected to their local neighbors, yet the global connectivity is lost. Figure 1 further illustrates the non-locality of the partition problem by showing two examples where the local picture looks essentially the same, yet one is disconnected while the other is not. This non-locality is the main technical challenge facing any detection scheme, and this is where we exploit the *physical embedding* of the sensor networks: because sensors are embedded in a geometric space (typically, a two-dimensional plane), geographical neighborhoods can be captured using simple geometric shapes. In particular, we use the theory of *VC-dimension* to formalize the shape complexity of spatial phenomena and derive mathematical bounds on the number of sensor nodes that suffice for our distributed monitoring.

The title *Catching Elephants with Mice* offers a visual metaphor for our approach: elephants are the large events smeared across the network, and mice are the sparse samples of sensors that allow us to catch these events. Our primary focus is on the feasibility of such sparse samples (mice) and characterization of events (elephants) that can be scalably detected. We, therefore, abstract away the low level networking issues, such as routing, and assume that a reliable communication protocol exists to send data from individual nodes to a central processor (the base station), where all the computation occurs. The communication overhead is quite minimal in our scheme anyway, because each of the monitoring nodes (mice) only sends a *binary* value to the base station. Thus, the scalability of our scheme can be measured roughly by the number of mice. With this background, let us now formalize the problem and summarize our results.

1.2 Problem Formulation and Our Contributions

Suppose a set S of n sensor nodes is distributed in the two-dimensional plane. We will model each sensor as a point, with (x_i, y_i) as it coordinates. We make no assumption about the distribution of these points (sensors), so our results hold for all, even adversarial, sets—what matters is the combinatorial nature of the distribution, such as which points are inside some shape and which are outside, and not its scale or metric properties. In addition, there is one distinguished node u that acts as the central processing node, or the *base station*, and knows the locations of all the sensors in the network. Throughout, we assume *idealized* sensing, meaning that each sensor inside the event detects it perfectly and no sensor outside the event detects it. In closing remarks, we briefly discuss some of the ramifications of these assumptions along with possible directions for future research.

A simple and abstract way to model an *event* is as a binary function over the xy-plane. We imagine that the function has value 1 over some geographical neighborhood (denoting the extent of the event) and 0 elsewhere. This model is sufficiently general to include all the applications we mentioned earlier. For instance, in the wild fire monitoring application, each node can locally decide its function value, setting it to 1 if it is above some preset temperature threshold, and 0 otherwise. The same holds also for energy level, or packet drop rates due to congestion etc. In other cases, such as network partition, a node can learn whether it is connected to the base station by communicating a test bit. In our scheme, each of the monitoring nodes (mice) will periodically send a binary bit to the base station. From these messages, the base station can also infer which nodes are able to reach it (have value 0), and which one cannot (have value 1). We assume that the chosen nodes will send their bits to the base station at some user-defined frequency, which determines the frequency of epochs at which event-detection algorithm runs. We assume that there is at most one event in the network at any time.

Let $E \subseteq S$ be the set of nodes with boolean value 1. We call E a *large event* or, metaphorically, an *elephant* if $|E| \geq \varepsilon|S|$, for a predefined user parameter ε, where $0 < \varepsilon < 1$. That is, if at least ε fraction of the nodes are affected by the event, then we call it an elephant. (Thus, the size of an event is measured by the number of sensor nodes that fall in it, and not by the physical size of the area covered by it.) Our goal is to choose a subset $M \subseteq S$ of sensors as monitors, or *mice*, and design an algorithm $\mathcal{A}$ such that, *given only the boolean values for the nodes of M*, the algorithm $\mathcal{A}$ can always decide whether an event is an elephant or not. (In fact, our scheme will achieve something stronger: for any event E, it can estimate the size of $|E|$ within the approximation error $O(\varepsilon|S|)$; the scheme also offers a natural and linear tradeoff between the estimation error and the size of the monitoring set.) Such a solution will be useful and scalable if the size of the set M is significantly smaller than the network size $|S|$.

A moment's reflection, however, makes it clear that unless the elephants are somehow "nicely shaped," one cannot hope to catch them all by a sparse, or *even linear* size, sample. Indeed, even if we choose *half* of all the sensors as mice, the adversary can create an event E that includes all the remaining nodes, and the algorithm has no way to detect this elephantine event. Put another way, if the geometric shape of the class of events E is allowed to be arbitrarily complex, then one would need arbitrarily large fraction of the nodes as mice to catch the events. Our result is a positive result in the *other direction*: for class of events whose shapes can be described in "nice" ways, *extremely sparse mice sets suffice.* We formalize the nice-ness of event shapes using the concept of *Vapnik-Chervonenkis (VC) dimension* from computational geometry and statistics [10, 26]. Most familiar geometric shapes, such as halfplanes, circles (disks), axis-parallel rectangles, arbitrarily oriented rectangles, have small VC-dimension (respectively, 3, 3, 4, and 5).

We show that if the class of events has VC-dimension at most d, then we can choose a set M of

$$O\left(\frac{d\log d}{\varepsilon}\log\frac{d\log d}{\varepsilon}\right)$$

sensors (mice) that can estimate the size of any event in our class within the (additive) approximation error of $O(\varepsilon|S|)$. A significant aspect of this result is that the number of mice needed to catch any elephant is *independent* of the network size, making this a scale-free property. In particular, for the most natural types of events that can be approximated by simple geometric shapes of constant VC dimension, such as circles, ellipses, rectangles and triangles, the size of the sample is $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$.

The theoretical bound presented above is scalable (independent of network size, and (almost) linear in $1/\varepsilon$ and d), computationally efficient, and also the best possible in a general setting. Yet, like many worst-case upper bounds, it appears to be highly pessimistic and unattractive for realistic-size networks, if applied blindly. We, therefore, propose several heuristic improvements to reduce the size of the monitoring set, and also perform extensive empirical testing to measure this "pessimism factor" inherent in the worst-case analysis. We conclude that the size of the final monitoring set can be reduced by at least a *factor of 30*, by the combination of our redundancy-aware sampling and empirical tuning.

Using a variety of synthetic and real-world data distributions (random, clustered, contoured, geographic, among others) and event geometries (circles, ellipses, rectangles), we provide a comprehensive simulation study of our proposed approach. Our simulation results validate the scalability and practical applicability of our approach. As an example, 50 mice suffice for $\varepsilon = 0.10$, and at most 150 mice are needed for $\varepsilon = 0.05$, for detecting and estimating circular, elliptical, or rectangular events (irrespective of the network size).

1.3 Organization

Our paper is organized in six sections. In Section 2, we briefly review the theory of VC dimension, and the related concept of ε-nets. In Section 3, we describe our main combinatorial result, namely, the bounds on the size of a sparse sample in terms of the VC dimension of the elephants. In Section 4, we discuss algorithmic aspects of our scheme, describe heuristic improvements for the size of monitoring

sets, and present redundancy-aware sampling. In Section 5, we present our simulation results, in Section 6, we survey related work, and in Section 7, we present our conclusions.

2 VC-Dimension and Geometric Complexity

In this section, we review the key concept of Vapnik-Chervonenkis (VC) dimension to express the shape complexity of geometric phenomena (elephants) that we wish to detect—an interested reader may refer to the excellent book by Matousek [17] for an in-depth coverage of these concepts. VC dimension, which arose in statistics [26] and developed further in computation geometry and learning theory [10, 2], gives a framework to formalize our informal intuition that a circle is a simpler shape than an axis-parallel rectangle, which in turn is simpler than a triangle. The basic idea is best explained using the abstract language of set systems.

A pair $(\mathcal{X},\mathcal{R})$ is called a *range space* if $\mathcal{X}$ is a ground set and $\mathcal{R}$ is a family of subsets of $\mathcal{X}$. The elements of the set $\mathcal{R}$ are often called the *ranges*. In our setting, $\mathcal{X}$ is the set of sensor nodes S, and $\mathcal{R}$ is a collection of subsets that arise from the events that we wish to detect. For instance, consider the class of events where, by placing a jamming device, an attacker can disable all nodes within a circular disk of radius 100. Then, $\mathcal{R}$ is the collection of all *combinatorially distinct* subsets of S that can be formed by disks of radius 100 whose centers can be placed anywhere in the plane. (In this example, even though there are infinitely many disks, one can show that the number of distinct subsets possible is at most $O(|S|^2)$.)[1]

Given a subset $A \subseteq \mathcal{X}$, we say that the set A is *shattered* by the set of ranges $\mathcal{R}$ if all $2^{|A|}$ subsets of A can be obtained by intersecting A with an appropriate member of $\mathcal{R}$. That is, for every subset $B \subseteq A$, there exists a range $R \in \mathcal{R}$, such that $B = A \cap R$. The *VC dimension* of the range space $(\mathcal{X},\mathcal{R})$ is the cardinality of the *largest* set $A \subseteq \mathcal{X}$ that is shattered by $\mathcal{R}$. (Not all sets of size d need to be shattered for the VC dimension to be d; one shattered set is enough.)

It may help to illustrate this concept with a few simple examples.

- The range space $(\mathcal{X},\mathcal{R})$ where $\mathcal{X}$ is the real line (or, a finite set of points on it) and $\mathcal{R}$ is the set of all intervals of that line, has VC dimension 2. No set of three points $\{a,b,c\}$ can be shattered since there is no interval whose intersection with the set gives the subset containing just the two extreme points (excluding the middle). On the other hand, for any two points a,b, we can form all four subsets $\{\emptyset,\{a\},\{b\},\{a,b\}\}$ by intersecting with an appropriate interval.
- The range space $(\mathcal{X},\mathcal{R})$ where $\mathcal{X}$ is a set of points in the two-dimensional plane, not all on a common line, and $\mathcal{R}$ is the set of all possible disks (with arbitrary radii), has VC dimension 3. It is easy to see that any subset of $\mathcal{X}$ with 3 non-collinear points can be shattered—one can obtain all 2^3 subsets by intersecting the set with an appropriate disk. On the other hand, no set of 4 points can be shattered because if the convex hull of the four points is a triangle, then it is impossible to obtain the subset containing just the three convex hull vertices without the middle fourth point; otherwise, all four points are on the convex hull, say, in the cyclic order a,b,c,d; then we cannot form both the subsets $\{a,c\}$ and $\{b,d\}$. Thus, the range space of circular disks has VC dimension 3.

Similar arguments show that the VC dimension of range spaces using other familiar geometric shapes are also small constants. For instance, axis-aligned rectangles have dimension 4, arbitrarily oriented ellipses have dimension 5, arbitrary triangles have dimension 7, etc. Lest one should think that the VC dimension is always finite, it is worth pointing that the VC dimension of the range space $(\mathcal{X},\mathcal{R})$, with $\mathcal{X}$ being the plane and $\mathcal{R}$ being all convex polygons is *infinite*. Indeed, by considering a set of points on a circle, we notice that *any* subset of it can be obtained by using exactly those points as the vertices of a convex polygon.

Let us now turn to the main problem at hand: what role does VC dimension play in our problem of catching elephants? The main connection is through the concept of ε-nets, which are defined as follows; we again refer an interested reader to [17] for further details.

Given a range space $(\mathcal{X},\mathcal{R})$, a subset $B \subseteq \mathcal{X}$ is called an ε-*net* for $\mathcal{X}$ if, for any range in $\mathbf{r} \in \mathcal{R}$, whenever $|\mathbf{r} \cap \mathcal{X}| \geq \varepsilon|\mathcal{X}|$, we also have that $\mathbf{r} \cap B \neq \emptyset$. In other words, a subset B is an ε-net if it has a non-empty intersection with any range that is an *elephant*. The ε-net theorem of Hausler-Welzl [10] shows that if we draw m independent random draws from $\mathcal{X}$, where

$$m \;\geq\; \max\left(\frac{8d}{\varepsilon}\log\frac{8d}{\varepsilon},\; \frac{4}{\varepsilon}\log\frac{2}{\delta}\right) \qquad (1)$$

then this random sample is an ε-net with probability at least $(1-\delta)$. There are deterministic constructions of ε-net as well [4] of size $O(\frac{d}{\varepsilon}\log\frac{d}{\varepsilon})$, however, those algorithms are quite expensive in running time complexity and complicated to implement, which makes them ill-suited for the lightweight computing model of sensor networks. (For instance, the deterministic scheme in [4] runs in time $O(|\mathcal{X}|d^{3d}(\frac{1}{\varepsilon})^{2d}\log^d(\frac{d}{\varepsilon}))$.)

At first glance, the ε-nets seem to be just the right tool for our problem, using the following simple scheme:

If we are interested in elephants of VC dimension at most d, we compute an ε-net for VC dimension d, and simply monitor the $O(\frac{d}{\varepsilon}\log\frac{d}{\varepsilon})$ nodes (mice) of this net. As soon as *any* node s in our ε-net reports signal value $\sigma(s) = 1$, we declare that a large event has occurred.

While clearly simple and scalable, this scheme suffers from the following two major drawbacks, which make it ill-suited for our problem.

- **False Alarms:** *The scheme has no guarantee that the events caught are elephants.* This is because the ε-nets

[1] This is seen most easily through a geometric duality: consider all disks of radii 100 with centers at the points of S. These n disks partition the plane into $O(n^2)$ distinct cells, where each such cell has the *equivalence* property that no matter where a disk is centered in it, it covers the same subset of S.

offer only a *one-sided guarantee*: they only tell us that whenever a large event occurs, at least one node of the ε-net will also detect it. But they lack the equivalent guarantee in the other direction: events that intersect the ε-net need not be large (indeed, in the worst-case, the event may include just one node of the network). A good alarm-raising mechanism, on the other hand, should come with an "if and only if" guarantee: large events should, of course, be caught, but any event reported to be an elephant must also have a guarantee of being sufficiently large. This is especially critical in applications like remote environmental monitoring and surveillance, where investigating a false alarm can be both costly and inconvenient.

- **No Size Estimation:** The ε-nets attempt only to catch the ranges whose size is above the εn threshold—*they are not designed to estimate the size of the range*. Thus, even when they catch an elephant, ε-nets offer no guarantee on the actual size of the event.

There is a classical strengthening of the ε-net, called ε-*sample* (also known as ε-approximation), which remedies both these problems, but unfortunately requires much larger sample size [10, 17, 26]. Specifically, if we are willing to maintain a sample of size

$$\Theta\left(\frac{d}{\varepsilon^2}\log\frac{d}{\varepsilon}+\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$$

then one can estimate the size of any range within additive error $O(\varepsilon n)$. However, since the size of this ε-sample is $\Theta(\frac{1}{\varepsilon})$ times larger than the ε-net, it does not scale nearly as well.

In the following section, we show that by using the ε-nets not on the original range space, but instead on their *symmetric differences*, we can remedy both problems mentioned above *without* increasing the asymptotic size of the sample set. We first describe the combinatorial result, then discuss the algorithmic aspects of the scheme.

3 Catching Elephants with Guarantees

In order to ensure a two-sided guarantee on the size of the events detected using sparse samples, it turns out that the key idea is to work with the *symmetric differences* of the ranges. For ease of presentation, we will frame our discussion in terms of *circular* ranges, but it will be self-evident that the approach is completely general.

3.1 ε-nets with Symmetric Difference

Consider two circular disks D_1 and D_2 in the plane. Their *symmetric difference*, denoted $D_1 \oplus D_2$, is the set $(D_1 \cup D_2) \setminus (D_1 \cap D_2)$, namely, the set of points that are in one but not both the disks. We will be mainly interested in situations when the two disks have a non-empty intersection. Clearly, the notion of symmetric difference holds for any two sets, not just disks, and does not require the two sets to be of the same type (see Figure 2 for an illustration). An important property of the symmetric difference is that it does not increase the VC dimension of the underlying range spaces too much. In particular, we have the following.

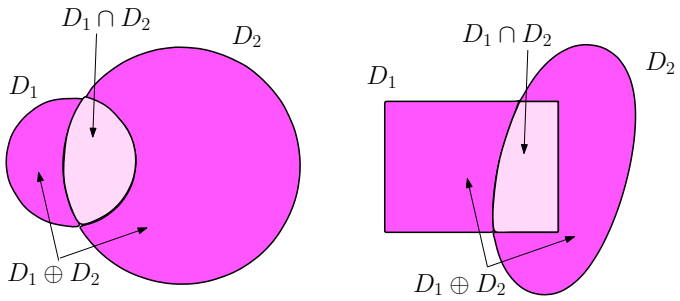


Figure 2. The symmetric difference of two disks (on the left) and a rectangle and an arbitrary convex set (on the right).

LEMMA 1. *Let $(\mathcal{X},\mathcal{R})$ be a range space of VC dimension d. Then, the range space $(\mathcal{X},\mathcal{R}')$, where $\mathcal{R}'$ is the set of ranges defined by the symmetric differences of pairs of ranges $\mathbf{r}_1,\mathbf{r}_2 \in R$, has VC dimension $O(d\log d)$.*

The proof of this fact follows from two observations: (1) the VC dimension of a range space equals the VC dimension of its complement range space, where each range $\mathbf{r}$ in the original space has a member $\mathbf{r}' = \mathcal{X} \setminus \mathbf{r}$ in the complement; and (2) the range space obtained by taking the unions or the intersections of two range spaces of VC dimension d have VC dimension $O(d\log d)$. An interested reader may consult Matousek [17] for more details.

Using the idea of symmetric differences, we now propose the following scheme for catching elephants, where S is the set of sensors, d is the maximum VC dimension of the events we want to catch, and ε is the fraction that defines an elephant event.

CATCHELEPHANTS (S,d,ε)

1. Let $d' = O(d\log d)$ be the dimension of the symmetric difference of range spaces derived from dimension d ranges.
2. Construct an $\frac{\varepsilon}{4}$-net for S with respect to the symmetric difference ranges of dimension d'. By the result of [10], such a net can be constructed with high probability by taking $O(\frac{d'}{\varepsilon}\log\frac{d'}{\varepsilon})$ random samples from S.
3. Let M be the set of nodes chosen as the $\frac{\varepsilon}{4}$-net, namely, our monitors or mice. Let $T \subseteq M$ be the subset of monitors with boolean value 1—this is the intersection of the event with our monitoring set.
4. Compute a disk D containing the locations of all the nodes in T and excluding the locations of all the nodes in $M \setminus T$. (Any arbitrary choice for such a disk will do, but we could also compute the smallest such disk.) Since the base station knows the locations of all the sensors, such a disk can be computed locally at the base station.
5. Compute the size $K = |S \cap D|$, the number of sensors in the network that lie inside the disk D.

6. If $K \geq 3\varepsilon n/4$, then report the event as an elephant, with K as its predicted size. Otherwise, the event is not an elephant.

The following theorem establishes the correctness of this algorithm, and proves that the algorithm catches *all* events of size at least εn (the elephants) and *never* reports an event of size less than $\frac{1}{2}\varepsilon n$ (two-sided guarantee). For events whose size lies in the range $(\varepsilon n/2, \varepsilon n)$, the algorithm is free to go either way—this is the approximation gray zone. In our proof, we continue to focus on circular disks, but the result is completely general, as we will argue following the proof of the theorem.

THEOREM 1. *Let E be an event in the sensor network, and let K be its size estimated by the algorithm* CATCHELEPHANTS. *Then,*

$$(K - \frac{\varepsilon n}{4}) \leq |E| \leq (K + \frac{\varepsilon n}{4}).$$

PROOF. Suppose the true event is induced by the disk D^*, while our algorithm estimates it with disk D. We first observe the following fact:

$$T \subseteq D \cap D^*, \tag{2}$$

where recall that $T \subseteq M$ is the set of monitors with value 1. This is true because all sensor locations with value 1 must clearly be inside D^*, and they are included in D by construction.

Let $K^* = |E|$ be the true size of the event E. Then, we note that our estimation error $|K^* - K|$ is bounded by the number of sensor nodes that lie in the symmetric difference $D \oplus D^*$: this follows because $D^* \setminus D$ contains all those sensors that are in D^* but not included in D (the undercount), and $D \setminus D^*$ contains all those sensors that our construction puts in D but are not in the true event (the overcount).

The key observation is that there is no monitor node in the symmetric difference, that is, $M \cap (D \oplus D^*) = \emptyset$. This holds because all monitors with values 1 are in $D \cap D^*$, there are no monitors with values 0 in D (by construction), and there are no monitors with values 0 in D^* by the property of the event. By the ε-net property, however, this implies that the number of nodes in the set $D \oplus D^*$ is at most $\frac{\varepsilon n}{4}$. Thus, in the worst-case, K can undercount or overcount $|E|$ by at most $\frac{\varepsilon n}{4}$, which proves the claim. □

One can see that this theorem immediately implies the guarantee associated with CATCHELEPHANTS: *every* elephant event is caught because its estimated size is never less than $3\varepsilon n/4$, while no *non-elephant* event (of size less than $\varepsilon n/2$) can be erroneously caught because its estimated size is always less than $3\varepsilon n/4$. For events of size in the range $(\varepsilon n/2, \varepsilon n)$, the algorithm can decide either way, depending on the disk D found in Step (4). But all these decisions are consistent with our approximation guarantees.

Remark: Technically speaking, the preceding theorem and its consequences should be written as probabilistic statements because we are using random sampling to compute an ε-net. For sake of simplicity, we have chosen to omit this qualifier. All our results, however, remain true even in the worst-case because *deterministic* constructions of ε-net are also possible; we have just opted to use the randomized version because in practice this is the construction of choice.

Remark: One step in the above described approach, which allows us to use the smaller size ε-nets instead of ε-approximations, is to classify the ranges with respect to how they intersect the ε-net—in this way we have approximately classified the ranges with respect to their size. A similar idea of putting the ranges in equivalence classes with respect to intersections with an ε-net was also used in [18] in their proof of Theorem 1.3.

3.2 Universal Guarantees

It is easy to see that the scheme CATCHELEPHANTS works for *any* range space of VC dimension d, and not just disks. The proof of Theorem 1 is purely combinatorial, relying on the set-theoretical properties of symmetric difference and ε-nets, and not the specific geometry of disks. In particular, the only place where we use disk in any significant way is the Step (4) of CATCHELEPHANTS, where we compute a disk D separating all monitors with value 1 from those with value 0. In the general scheme, D can be replaced by *any* geometric shape of VC dimension at most d—it need not be of the same shape as the event! This works because we only require that the symmetric difference $D \oplus D^*$ has VC dimension $O(d \log d)$, which holds *as long as each of D and D^* have VC dimensions at most d*; we are abusing the notation D and D^* slightly to denote generic shapes of dimension d. However, *algorithmically* we need more specific knowledge of the event geometry to find a feasible separating shape. We will discuss these algorithmic aspects in the next section, and conclude our discussion by stating our combinatorial theorem.

THEOREM 2. *Given a set S of n sensors in the plane, a user-specified parameter ε, where $0 < \varepsilon < 1$, we can choose a subset $M \subset S$ of size $O\left(\frac{d \log d}{\varepsilon} \log \frac{d \log d}{\varepsilon}\right)$ that serves as a universal sample for catching every ε-event (elephant) of VC dimension at most d, with an approximation error of $\pm \frac{\varepsilon n}{4}$.*

In the next section, we address some of the algorithmic issues behind CATCHELEPHANTS, and discuss several heuristic improvements that we implemented.

4 Algorithmic Issues, Heuristic Improvements, and Optimizations

There are two main algorithmic issues that arise in the implementation of CATCHELEPHANTS: computing the ε-nets (Step 2) and determining an appropriate range D separating the subset of nodes with boolean value 1, namely, T from the subset with boolean value 0, namely, $M \setminus T$ (Step 4). We begin with the algorithmic details of Step 4, and then describe our heuristic improvements and optimizations that lead to a dramatic reduction in the sample size.

4.1 Computing Geometric Separators

Step 4 of CATCHELEPHANTS requires specialized algorithms for computing a member of the event class separating the locations of T from those in $M \setminus T$. For some classes, such as axis-parallel rectangles, the algorithm is trivially straightforward and efficient (running in time $O(|T|)$). The key observation is that if *any* rectangle separates T from $M \setminus T$, then the *minimal (smallest-area)* rectangle containing T also separates them. The minimal rectangle containing T is determined by the (upto) four points that have the minimum and the maximum x and y coordinates in T, which can be found in linear time by a single scan of T.

For other shapes, such as circles and ellipses, more sophisticated algorithms are called for. In general, the minimality property exploited above for axis-parallel rectangles also is false for shapes such as circles, ellipses, or triangles. In other words, one can easily show examples where the minimal (by area, perimeter, or any other natural measure) shape of a class containing T may also contain some members of $M \setminus T$, while there exists another non-minimal shape separating T from $M \setminus T$. As a result, various well-known algorithms from computational geometry for finding minimum area enclosing shapes (circle, ellipse, triangle etc.) [9, 20, 28] are not useful in our setting—we require *separating* geometric objects, not enclosing ones. However, basic ideas from those schemes can still be used to frame the problem as a quadratic or semi-definite programming problem. Below we briefly describe details of these formulations for separation by circles and ellipses.

4.1.1 Separating Circles

In order to find a circle that separates the points in T from those in $M \setminus T$, we formulate the problem as a special kind of a *quadratic program*. The objective function of the formulation is convex quadratic function, the constraints are linear, and the program involves a constant number of variables a, b, c:

$$\begin{array}{ll} \min \ (a^2/4 + b^2/4 + c) & \text{s.t.} \\ ax + by + (x^2 + y^2) \leq c, & (x,y) \in T \\ ax + by + (x^2 + y^2) \geq c, & (x,y) \in M \setminus T \end{array}$$

Intuitively, a circle is determined by 3 unknowns, the coordinates of its center and its radius. In the formulation above, we solve for the three unknowns a, b, c such that the circle with center $(a/2, b/2)$ and radius $(a^2/4 + b^2/4 + c)^{1/2}$ contains all the points of T inside it, and all other points outside it; this particular form of the expression helps simplify the algebraic manipulation. Our objective function therefore seeks to find such a circle with the smallest radius. (The reader familiar with geometric duality will recognize that this is an instance of paraboloid transform: circular separability in two dimensions ($\mathbb{R}^2$) is mapped to planar separability in three dimensions ($\mathbb{R}^3$). Specifically, points of the (x,y)-plane are mapped onto the paraboloid $z = x^2 + y^2$. Thus, the point (u,v) lifts to the point (u,v,u^2+v^2) in 3-space. The key observation is that the intersection of any non-vertical plane with the paraboloid projects to a circle on the x–y plane, and so a plane that has all transformed points of T below and all transformed points of $M \setminus T$ above projects to a circle separating the two sets. As a result, the problem of finding the minimum radius circle separating T and $M \setminus T$ can be written as linear separability in three variables, as shown above.)

The key point is that such a constant-dimensional mathematical programming problem can be solved in time linear in the number of constraints using the technique of Megiddo [20].

4.1.2 Separating Ellipses

The problem of finding an ellipse separating two point sets can also be formulated and solved using linear programming. The following mathematical program describes the formulation of the ellipse separability:

$$\begin{array}{rll} \min \ 0 & \text{s.t.} & \\ p'Ap + B'p + C \geq -1, & & p \in T \\ p'Ap + B'p + C \leq -1, & & p \in M \setminus T \\ A \leq -I, & & \end{array}$$

where p denotes a point in vector form, and the 2×2 matrix A, the 1×2 matrix B, and the scalar C encode the unknown coefficients representing the ellipse in standard form. Z' denotes the transpose of the matrix Z, and I is the identity matrix.

In addition to circles and ellipses, many other simple separation problems can be expressed using these techniques as a linear program. In fact, any shape that has a fixed-degree polynomial representation can be solved this way.

4.2 Heuristic Improvements to Monitoring Sets

A straightforward application of Theorem 2 yields highly pessimistic bounds for the size of monitoring sets. Like many worst-case theoretical bounds, this is an artifact both of the difficulty of exact analysis as well as the inherent nature of worst-case analysis, which tends to focus on rare pathological cases. In this section, we propose two ideas that dramatically reduce the size of the monitoring sets in practice. The first idea is simply to use empirical data to estimate the "degree of pessimism" built into the theoretical bound. Through extensive simulations using a variety of data and event geometries, we find that the size of the monitoring set can be reduced in practice by a *factor of fifteen* without ever sacrificing its error guarantees. The second optimization is to use a *redundancy-aware sampling* to overcome some of the ill-effects (clustering) of blind sampling. This further reduces the size of the monitoring sets by half. The two heuristics together, therefore, lead to a factor of 30 reduction in the sample size without compromising the error guarantees of the scheme. We discuss these two heuristics in some detail below.

4.2.1 *Estimating Theoretical Pessimism*

In order to empirically estimate the factor by which Theorem 2 may overestimate the size of the monitoring sets in practice, we carried out extensive simulations using several different data sets of sizes varying from 2000 to 5000 points in the two-dimensional plane. (These data sets are described in more detail in the Simulations section, but briefly they vary from uniform random, to clustered, to contour, to geographical features in a census data.) For each data set, we experimented with different event geometries (circles, ellipses, and rectangles).

For a given data set, a given event geometry, and a given value of ε, our experiment started with a monitoring set matching the theoretical bound of Theorem 2, and then we tried sets of successively smaller sizes, until the *maximum* error of event-size estimation exceeded the theoretical guarantee of εn. A total of 30,000 event geometries were tested for a given value of ε, as for each of the 15 dataset-event geometry pairs, we generated 2000 events randomly in the size range $[0.1n, 0.3n]$. Thus, a lower value of the monitoring set was chosen only if *all* 30,000 random events are estimated within the guaranteed approximation bound.

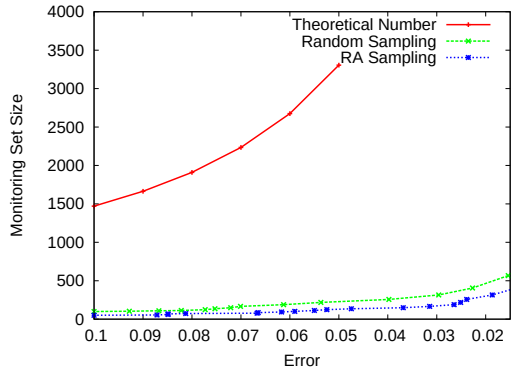
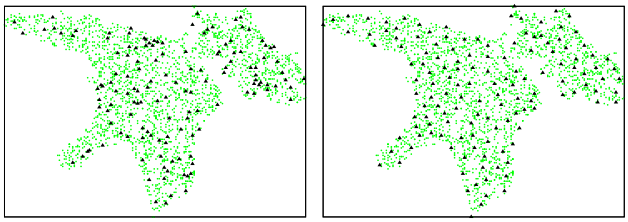


Figure 3. The size of the monitoring set for various values of the parameter ε. The top curve is the theoretical bound; the second curve is computed by empirical data. The third curve shows the improvement given by Redundancy-Aware sampling, which is described in the next subsection.

In Figure 3, we show a plot of the monitoring set size versus the *maximum* error observed for any event in the simulation. In this experiment, we used only circle events, but the results are identical for the other two event geometries (ellipses and rectangles) as well. There are three curves—the topmost curve is the size required by Theorem 2; the second curve (labeled Random Sampling) is the conservative size observed empirically, which ensures the same error bound; the third curve (labeled RA Sampling) is the outcome of our redundancy-aware sampling, which we discuss in the next subsection. The important conclusion is that the theoretical estimate is at least 15 times larger than the conservative size needed in practice. With the addition of Redundancy-Aware sampling, which we discuss next, the overestimation in size is by a factor of at least 30!

4.2.2 *Redundancy-aware Sampling*

Random sampling has the virtue of simplicity, but it also tends to lead to larger-than-necessary sample sizes. The well-known *coupon collector's problem* is a particularly simple setting to observe this phenomenon. Suppose we have a large bin containing multiple (equal number of) copies of m colored balls, and we want to randomly draw samples from the bin until we get at least one of each color. Then, $\Theta(m \log m)$ random draws on average are necessary (and sufficient), while an "oracle" can clearly produce the outcome with just m draws, never drawing two balls of the same color. Similarly, in our application, the simple sampling can result in redundant points near each other, which leads to more monitors overall than necessary.



Lake (2500 nodes)

Figure 4. Regular random sampling (left), and REDUNDANCY-AWARE sampling (right).

In order to further improve the size of the monitoring set, we therefore employ a simple heuristic optimization: we discard any sample that is "close" to a previously chosen sample, and resample. The notion of closeness, however, is chosen carefully to reflect the combinatorial nature of the problem, and is not based on Euclidean distance. (Indeed, we do want monitors to mimic the distribution of the underlying sensor set, and therefore expect them to be packed densely in clustered areas and sparsely in other areas. Thus, a uniform distance-based resampling strategy will not work.) Our measure of "combinatorial distance" of two points u and v is the number of points (sensor locations) that lie inside the circle defined by uv as diameter.

Our sampling scheme, which we call REDUNDANCY-AWARE sampling, is implemented in rounds, and in each round a distance *threshold* is used to decide whether to add the next node to our monitoring set or not. The threshold is initially set to $c|M|/(2k)$, where c is a small constant (set to 10 in our simulations), $|M|$ is the target size of the monitoring set, and k is the round number. In each round, $c|M|/2$ nodes are picked uniformly at random, and for each node we perform the following simple test: if the *minimum* combinatorial distance of the new node from the current set of monitors exceeds the threshold, then we add it to the monitoring set; otherwise, we skip it. The process is continued until we reach the target count for the monitoring set. Figure 4 shows an example of REDUNDANCY-AWARE sampling on the Lake data set: one can readily see that the samples chosen by REDUNDANCY-AWARE sampling (right) are much more uniformly distributed than the blind random sampling

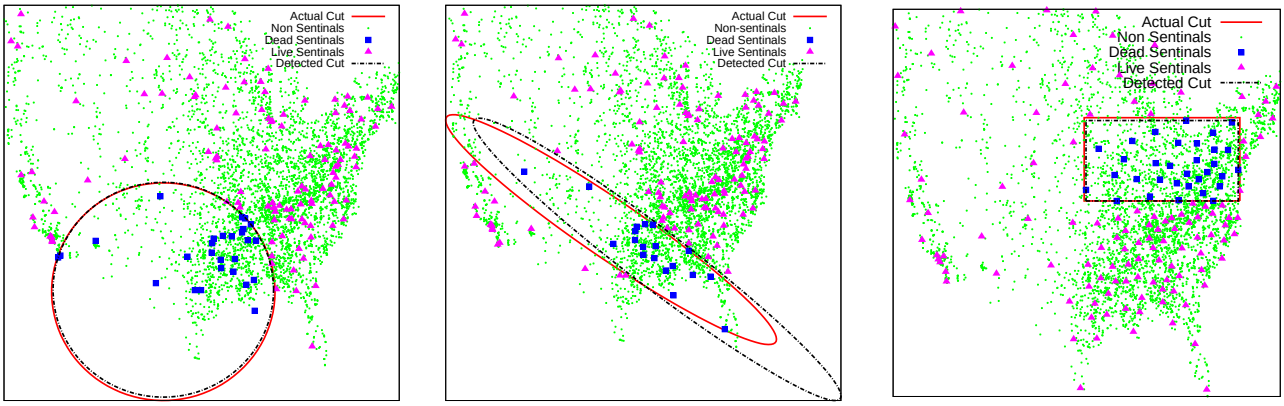


Figure 5. An illustration of our scheme on the $n = 5000$ point Census dataset, with $\varepsilon = 0.1$. The monitors outside the event are shown as solid triangles, while those inside the event (dead mice) are shown as solid squares. The dotted lines show the geometry of the true event, while solid lines show the geometry of the event shape found by CATCHELEPHANTS. The maximum estimation error in these examples is less than $0.01n$, well within the target approximation guarantee of $0.1n$. The maximum number of monitors in these examples is 140.

(left), which has the predicted clustering effect in several places.

5 Simulation Results

In this section, we discuss the empirical performance of our proposed scheme on a variety of synthetic and real data using simulation. We begin by describing the general methodology and parameters of our simulation study.

5.1 Methodology and Parameters

We use a number of different data distributions to capture aspects of different geographical settings in which large-scale sensor networks may be deployed. We envision our scheme to be useful for moderately large-scale sensor networks, so all our data sets are in the range $n = 1000$ to $n = 45000$. The different data sets have the following characteristics:

- **Census** is a portion of GIS data containing locations of 5000 geographical features in the United States—this set is suggestive of a large scale deployment around physical sites in the world.
- **Clustered** is a synthetically generated set of 5000 points, 90% of which form 15 Gaussian distributed clusters with random centers, and the remaining 10% are distributed uniformly—this set is suggestive of a mixed distribution where most sensors are part of a few dense clusters, intermixed with a sparse background uniform distribution.
- **Lake** data contains 2500 distributed uniformly at random inside the boundary of Lake Huron—this set is suggestive of a deployment inside a geographical area with a complex boundary.
- **Contour** data is a portion of a weather isocontour, with 2000 points placed randomly on the boundary. Such a set is potentially useful in settings when the sensors are placed along the boundary of some region.
- **Random** is a synthetically generated set of points distributed uniformly at random inside a rectangle. We used this data for various scalability experiments, with n upto $100,000$.

In all our experiments, the monitoring sets are constructed by using REDUNDANCY-AWARE sampling, and their size is chosen conservatively according to the empirical plot of Figure 3.

We use three natural event geometries: circles, ellipses, and axis-aligned rectangles. Each event was associated with a target size, which was typically chosen randomly in the size $[0.1n,\ 0.3n]$. To generate a circle event, we choose its center uniformly at random in the plane, and then grow the circle until the number of points inside it reaches the target count. To generate a rectangle event, we also choose a center at random, and then grow the rectangle, keeping its aspect ratio (width to height ratio fixed, which we set to two in our experiments) until the number of points inside the rectangle reaches the target. Finally, ellipses are generated by choosing five random points, which define a unique ellipse. If number of points inside this ellipse is not in the desired target range, we try again.

Thus, there are 15 dataset-event geometry pairs, and for each such pair we generated 2000 random events. Thus, each of our experiment reports the cumulative outcome of these 30,000 tests. We now discuss the results of our simulations.

5.2 Qualitative Impression

We begin with some qualitative (visual) results to convey the overall effectiveness of the scheme in dealing with different event geometries and data sets. (We will discuss the quantitative results, estimation errors, and scaling effects

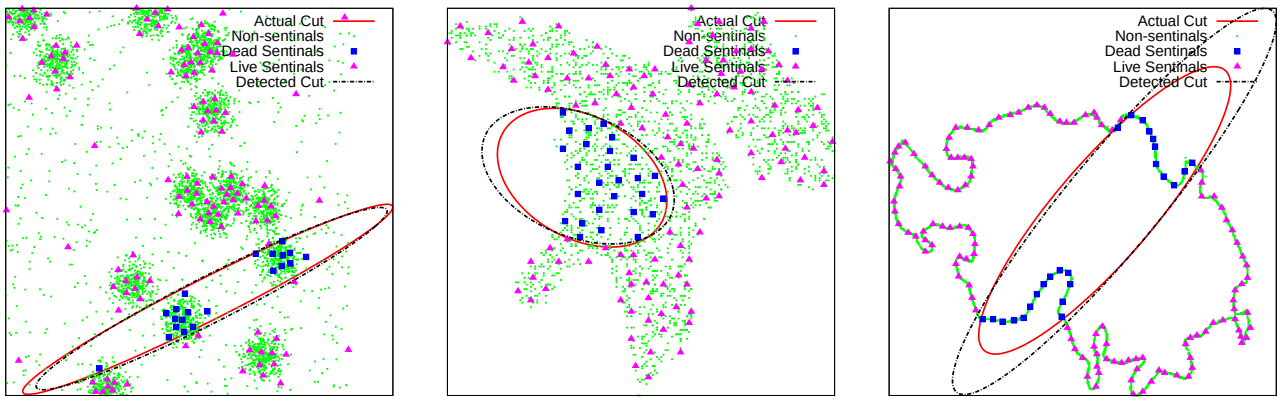


Figure 6. The figure shows detection of ellipse-shaped events on three different data sets: clustered, Lake, and Temperature. The maximum estimation error in these examples is again less than $0.01n$, while the target approximation guarantee is $0.1n$.

later, but these figures are meant to convey a general visual impression of how well the event shapes computed by our algorithm matches the true events.) Figure 5 shows the results of three different event geometries (circle, ellipse, and axis-parallel rectangle) on the 5000 point Census data set. Figure 6 is similar in nature, except it shows events of a single shape family (ellipse) on three widely different data sets: clustered, lake, and temperature. In both sets of figures, one can see that our scheme performs quite well, and returns a plausible event boundary that is very close to the truth.

Our next two experiments are designed to evaluate the scalability of the scheme: how does the monitoring set scale in practice as a function of ε, and the network size n.

5.3 Scaling of the monitoring set with ε

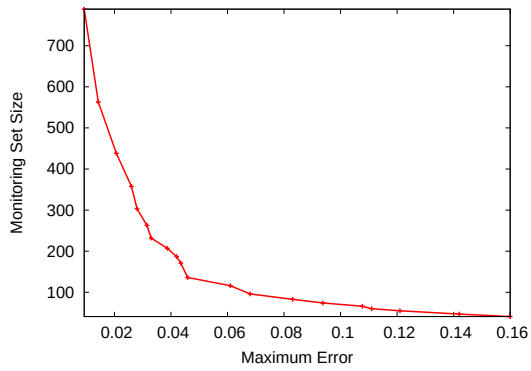


Figure 7. The maximum error over all datasets versus varying monitoring set sizes.

We vary the ε parameter from 0.16 down to 0.01, and for each value of ε, we find the smallest monitoring set that correctly estimates the size of all 30,000 randomly generated events (2000 events each of the three types, and for each of the 5 data sets) within the approximation bound. The smallest monitoring set is found using the method outlined in Section 4.2.1, using the REDUNDANCY-AWARE sampling. Figure 7 shows the resulting plot of the monitoring set size as a function of ε. As expected, the size of the monitoring set grows inversely proportional to ε, but it is reasonably small (less than 150) for all values of ε greater than 0.05.

Our next experiment verifies the scale-free property of our scheme, namely, that the size of the monitoring set does not grow with the network size.

5.4 Size of the monitoring set vs. Network size

In order to empirically demonstrate that our monitoring set is scale-free, we used the synthetic random and clustered data sets so we can test over a large range of network size. For both data sets, we started with $n = 1000$, and we grew the network size in increments of 5000, ending with $n = 46,000$. Throughout this experiment, we use only two representative values of the ε parameter, namely, $\varepsilon = 0.05$ and $\varepsilon = 0.1$. (The results were nearly identical for other values of ε as well, but we present only these two to keep the figure uncluttered.)

For each value of ε and each network size, we compute the monitoring set and then test it against the randomly generated 2000 events each of the 3 types (circles, ellipse, and rectangles) to ensure that it estimates the size of all events within the approximation bound.

Figure 8 shows the plot of the monitoring set size versus the network size. The figure clearly shows that the monitoring set size is essentially constant over the whole network size range.

Based on plots of Figure 7 and Figure 8, one can conclude that while for different data sets and network sizes, the minimal size of monitoring set may vary a little, one can conservatively choose a size close to that predicted by Figure 7 as a safe estimate in practice.

5.5 Universality and Error Performance

In this experiment, we evaluate the error performance and the universality of our scheme. For each data set, we fix a

Dataset	Shape	Maximum Error			Average Error		
		0.125	0.050	0.025	0.125	0.050	0.025
Random	Circle	0.0945	0.0471	0.017	0.0528	0.0218	0.0081
	Rectangle	0.0835	0.0481	0.022	0.0392	0.0171	0.0073
	Ellipse	0.0754	0.0310	0.0102	0.0145	0.0067	0.0025
Census	Circle	0.0926	0.0356	0.0154	0.0376	0.0157	0.0060
	Rectangle	0.1071	0.0463	0.0184	0.0355	0.0154	0.0066
	Ellipse	0.0536	0.0254	0.0100	0.0117	0.0056	0.0022
Clustered	Circle	0.0981	0.0359	0.0138	0.0438	0.0174	0.0066
	Rectangle	0.1024	0.0421	0.0165	0.0351	0.0160	0.0066
	Ellipse	0.0482	0.0216	0.0110	0.0120	0.0059	0.0024
Contour	Circle	0.0595	0.0315	0.0107	0.0324	0.0126	0.0045
	Rectangle	0.0794	0.0357	0.0180	0.0320	0.0159	0.0056
	Ellipse	0.0455	0.0155	0.0071	0.0090	0.0040	0.0015
Lake	Circle	0.0791	0.0367	0.0140	0.0390	0.0163	0.0058
	Rectangle	0.0942	0.0486	0.0240	0.0315	0.0142	0.0054
	Ellipse	0.0548	0.0280	0.0092	0.0128	0.0058	0.0023

Figure 9. Comparison between the error produced by using different event shapes on different datasets.

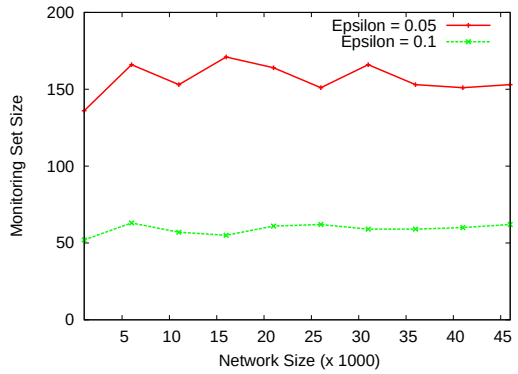


Figure 8. The monitoring set sizes for ε = 0.05 and 0.10 for different network sizes.

value of the parameter ε, and compute the universal sample for that ε (assuming a VC dimension commensurate with our three event geometries). We then use this monitoring set to approximate the events of all three geometries (circles, ellipses, and rectangles)—as before, we use 2000 random events of each class. We measure both the *maximum* error and the *average* error of approximation. This experiments reveals some more details of the behavior of different data sets and different geometries that had been aggregated in the earlier experiments.

We also use three different values of ε, namely, $0.125, 0.050$, and 0.025. The results are shown in Figure 9. In all cases, both the maximum and the average error are well within the target approximation factor ε. We also observe that, in general, the error is larger for circles and rectangles than ellipses. This is largely due to the fact that our algorithm computes a *minimal* separator for circles and rectangles, which tend to bias the results towards the lower bound in the size estimation. This effect disappears if we also use any feasible separator for circles and rectangles, instead of a minimal one; those results are not reported here.

5.6 Estimation without Geometry: Proportionate Scaling

Our scheme CATCHELEPHANTS requires some knowledge of the event shape class to compute its hypothesis event D (Step 4). While the scheme is quite robust in the sense that *any* shape of the VC-dimension d will serve, still it entails a certain computational overhead. It may, therefore, be tempting to try a simpler heuristic: estimate the event to have size $n|T|/|M|$. (Since $|T|/|M|$ fraction of the monitors are in the event, we estimate the same fraction of the whole network to be in the event.) This *scaling-based* scheme has *no provable theoretical guarantee*—indeed, the theory of ε-sample shows that such a guarantee can only be assured with at least $\Omega(\frac{d}{\varepsilon^2}\log\frac{d}{\varepsilon})$ samples—yet it is instructive to see how it performs in practice.

The influence of the extra $1/\varepsilon$ term in the sample size is prominent only when ε becomes small. Therefore, we focused on the values of ε between 0.01 and 0.001, and ran experiments on the synthetic **random** and **clustered** data sets of $n = 100,000$ points. Again, we use all three event geometries, generating 2000 random events for every dataset-event geometry pair in the range $[0.1n, 0.3n]$. Figure 10 summarizes the results of this experiment, and shows that in general the estimation error of the scaling-based well exceeds the target approximation error, approaching *9 times* the target error when $\varepsilon = 0.001$. However, when ε is relatively large (say, 0.1), even this simplistic method gives acceptable performance, and may be a useful alternative to our geometry-based counting.

6 Related Work

Detection of events in sensor networks is a very broad topic and, indeed, a major application domain. It is beyond the scope of this paper to even attempt a comprehensive summary of all the research threads related to this rich topic. Instead, we briefly touch upon some of the work that is related

Target ε	Error of the Scaling Scheme	Ratio
0.01	0.021	2.10
0.008	0.0198	2.47
0.006	0.0185	3.08
0.004	0.0158	3.95
0.002	0.0120	6.0
0.001	0.089	8.9

Figure 10. Relative estimation errors for the scaling-based counting scheme. Thus, for $\varepsilon = 0.001$, the estimate of the scaling scheme has roughly *9 times* the target error.

to our narrow focus in either by the nature of technical problems considered or by the technical approach used.

One important thread concerns the extraction of "boundary" in sensor networks, which can abstractly be thought of as determining the extent of an event embedded in the sensor field [5, 15, 21, 23, 27]. There are also schemes for representing the boundary of an event or signal landscape of the sensor network compactly using in-network aggregation [3, 8, 11, 24]. Then there are contour-based methods for deciding the type of an event [31]. Our focus is quite different from this thread of research, as we are not interested in a detailed boundary description of the event, but only in detecting its onset and, as a result, our methods are also fundamentally different.

Another thread of research involves detecting holes and topological features of a sensor networks [7, 14]. The focus of this work is to use local connectivity to infer global features, but they are interested in static features and require participation of all the nodes in the network. By contrast, we are interested in external events that appear dynamically, and our major focus is the very sparse sampling of the network.

In terms of techniques, our work draws inspiration from Kleinberg et al. [12, 13] and Shrivastava et al. [22]. In [12, 13], the authors consider a network failure problem in a wired network, where the goal is to detect a network partition under the assumption that an adversary can delete upto k links or nodes of the network. They also use the VC dimension framework to show that the dimension of the set-system for this graph-theoretical problem is polynomially bounded in k. Shrivastava et al. [22] considered the network partition problem in the sensor network setting, and focused on geometric cuts defined by *straight lines*—that is, an adversary can kill all communication between two sides of a linear cut.

The VC dimension based techniques are a common thread between our work and these papers, but otherwise there are significant differences. The work in [12, 13] focuses entirely on graph networks, whereas geometry of events plays a central role in our research. Another main technical difference is the two-sided guarantee of approximation—the results of [12, 13] suffer from the same two drawbacks we mentioned in Section 2. The methods of [22] are limited to linear network partition, while our scheme is applicable to a very broad class of event geometries.

7 Closing Remarks

Sensor networks are an ideal technology for enabling (cost-effective) large-scale and fine-grained surveillance and monitoring. A fascinating algorithmic challenge posed by these distributed, resource-starved systems is to stitch together a "global" picture from local sensing of individual nodes. Our work is a contribution in this general quest, where we show the collaborative power of a small number of well-chosen nodes in detecting events that affect a non-trivial portion of the network.

The idea of using a random sample to catch large events is not particularly novel. Our contribution, however, lies in *quantifying* the necessary size of the sample, using a universal measure of geometric complexity, namely, the VC dimension; the VC-dimension bounds are a refinement of the classical Chernoff-style bounds. Our second contribution is to *bridge* the gap between theory and practice: in particular, while our sampling theorem gives a near-ideal theoretical bound, that bound proves to be infeasibly large in practice. Our heuristic improvements and empirical tuning conservatively reduce the sample size by a *factor of 30*, making the approach eminently practical.

In order to derive theoretically rigorous results with worst-case guarantees, we needed to make several simplifying assumptions in the paper. For instance, we assumed that all sensors are identical and the physical size of the event is reflected in the *number* of sensors that lie inside the event. In practice, however, one may assign different level of importance to different sensors (for instance, to reflect different sensor density). In this case, our scheme extends easily: we simply assign weights to sensors, normalize the sum to 1, and look for events that affect ε fraction of the weight. We also assumed that sensors have well-defined circular sensing range, whose radius is small compared to the size of the event being detected, and the detection is *error-free* (no false positives or negatives). These are fairly standard assumptions in sensornet research, consistent with a *threshold-based* approach to sensing. However, it is well-known that the sensing performance is far from this ideal, and modeling the geometry of realistic sensors is a challenging future direction for research.

One possible way to relax the constraints of perfect sensing and tightly-defined event geometries is through the use of a buffer region, as in the quasi-disk model of wireless transmission. Specifically, we may assume that the event boundary is defined by two nested circles (or other simple shapes) such that all sensors inside the inner circle or outside the outer circle detect the event perfectly (the inner ones report 1 and the outer ones report 0), while the sensors in the "annulus" between the two circles have unreliable detection. Our methodology works in this more general setting, except more sophisticated algorithms are needed for the geometric separation.

Acknowledgment

The authors wish to thank the (anonymous) members of the SenSys '07 program committee for their valuable feedback, as well as the guidance provided by our shepherd.

8 References

[1] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, et al. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks*, 46(5):605–634, 2004.

[2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. of the ACM*, 36(4):929–965, 1989.

[3] C. Buragohain, S. Gandhi, J. Hershberger, and S. Suri. Contour approximation in sensor networks. In *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 356–371, 2006.

[4] B. Chazelle and J. Matousek. On linear-time deterministic algorithms for optimization problems in fixed dimension. In *Symposium on Discrete Algorithm (SODA)*, pages 281–290, 1993.

[5] K. Chintalapudi and R. Govindan. Localized edge detection in sensor fields. In *IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, pages 59–70, 2003.

[6] A. Dhariwal, B. Zhang, B. Stauffer, C. Oberg, et al. NAMOS: Networked aquatic microbial observing system. In *ICRA*, 2006.

[7] S. Funke. Topological hole detection in wireless sensor networks and its applications. In *DIALM-POMC*, pages 44–53, 2005.

[8] S. Gandhi, J. Hershberger, and S. Suri. Approximate isocontours and spatial summaries for sensor networks. In *International Conference on Information Processing in Sensor Networks (IPSN)*, 2007.

[9] B. Gartner. Fast and robust smallest enclosing balls. In *ESA*, pages 325–338, 1999.

[10] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete & Computational Geometry*, 2:127–151, 1987.

[11] J. Hellerstein, W. Hong, S. Madden, and K. Stanek. Beyond average: Toward sophisticated sensing with queries. In *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 63–79, 2003.

[12] J. Kleinberg. Detecting a network failure. In *Annual Symposium on Foundations of Computer Science (FOCS)*, page 231, 2000.

[13] J. Kleinberg, M. Sandler, and A. Slivkins. Network failure detection and graph connectivity. In *Symposium on Discrete Algorithms (SODA)*, pages 76–85, 2004.

[14] A. Kroller, S. Fekete, D. Pfisterer, and S. Fischer. Deterministic boundary recognition and topology extraction for large sensor networks. In *Symposium on Discrete Algorithm (SODA)*, pages 1000–1009, 2006.

[15] P. Liao, M. Chang, and C. Kuo. Contour line extraction with wireless sensor networks. In *ICC*, pages 3202–3206, 2005.

[16] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, et al. Wireless sensor networks for habitat monitoring. In *WSNA*, pages 88–97, 2002.

[17] J. Matousek. *Lectures in Discrete Geometry*. Springer-Verlag, 2002.

[18] J. Matoušek, E. Welzl, and L. Wernisch. Discrepancy and approximations for bounded vc-dimension. In *Proc. of the 32nd annual symposium on Foundations of computer science*, pages 424–430. IEEE Computer Society Press, 1991.

[19] K. Mayer, K. Ellis, and K. Taylor. Cattle health monitoring using wireless sensor networks. In *IASTED CCN*, 2004.

[20] N. Megiddo. Linear-time algorithms for linear programming in r^3 and related problems. *SIAM Journal on Computing*, 12(4):759–776, 1983.

[21] R. Nowak and U. Mitra. Boundary estimation in sensor networks: Theory and methods. In *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 80–95, 2003.

[22] N. Shrivastava, S. Suri, and C. Toth. Detecting cuts in sensor networks. In *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 210–217, 2005.

[23] M. Singh, A. Bakshi, and V. Prasanna. Constructing topographic maps in networked sensor systems. In *ASWAN*, 2004.

[24] I. Solis and K. Obraczka. Efficient continuous mapping in sensor networks using isolines. In *MOBIQUITOUS*, pages 325–332, 2005.

[25] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, et al. A macroscope in the redwoods. In *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 51–63, 2005.

[26] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

[27] Y. Wang, J. Gao, and J. Mitchell. Boundary recognition in sensor networks by topological methods. In *MOBICOM*, pages 122–133, 2006.

[28] E. Welzl. Smallest enclosing disks (balls and ellipsoids). *New Results and New Trends in Computer Science, Lecture Notes in Computer Science*, pages 359–370, 1991.

[29] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, et al. Monitoring volcanic eruptions with a wireless sensor network. In *European conference on Wireless Sensor Networks (EWSN)*, 2005.

[30] A. Wood and J. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, 2002.

[31] W. Xue, Q. Luo, L. Chen, and Y. Liu. Contour map matching for event detection in sensor networks. In *In-*

ternational Conference on Management of Data (SIGMOD), pages 145–156, 2006.

[32] Y. Zhao, R. Govindan, and D. Estrin. Residual energy scan for monitoring sensor networks. In *WCNC*, 2002.