A Fresh Perspective: Learning to Sparsify for Detection in Massive Noisy Sensor Networks

Matthew Faulkner
Computer Science
Caltech
mfaulk@caltech.edu

Annie H. Liu
Computer Science
Caltech
aliu@cms.caltech.edu

Andreas Krause
Computer Science
ETH Zurich
krausea@ethz.ch

ABSTRACT

Can one trade sensor quality for quantity? While larger networks with greater sensor density promise to allow us to use noisier sensors yet measure subtler phenomena, aggregating data and designing decision rules is challenging. Motivated by dense, participatory seismic networks, we seek efficient aggregation methods for event detection. We propose to perform aggregation by *sparsification*: roughly, a sparsifying basis is a linear transformation that aggregates measurements from groups of sensors that tend to co-activate, and each event is observed by only a few groups of sensors. We show how a simple class of sparsifying bases provably improves detection with noisy binary sensors, even when only qualitative information about the network is available. We then describe how detection can be further improved by learning a better sparsifying basis from network observations or simulations. Learning can be done offline, and makes use of powerful off-the-shelf optimization packages. Our approach outperforms state of the art detectors on real measurements from seismic networks with hundreds of sensors, and on simulated epidemics in the Gnutella P2P communication network.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; G.3 [**Probability and Statistics**]: Experimental Design; I.2.6 [**AI**]: Learning

General Terms

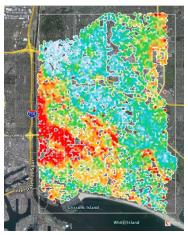
Algorithms, Experimentation, Theory

Keywords

Sparsifying transformation, basis learning, sensor networks, community sensing, event detection, ICA, SLSA

IPSN'13, April 8–11, 2013, Philadelphia, Pennsylvania, USA.

(a) CSN Participants



(b)

Figure 1: (a) CSN sensors; (b) Peak amplitude of Compton M3.4 quake measured by 4089 sensors. Note the complex spatial correlation.

1. INTRODUCTION

In recent years, millions of accelerometers have appeared across cities around the world. Most of these sensors are in privately owned, Internet-enabled devices like smartphones and laptops. Several participatory sensing projects, including the Community Seismic Network (CSN)[1], the Quake Catcher Network (QCN)[2], and iShake[3] are working to unify these numerous but noisy devices to measure and detect strong earthquakes.

Quake detection in community networks requires finding a complex spatio-temporal pattern in a large set of noisy sensor measurements. The start of a quake may only affect a small fraction of the network, so the event can easily be concealed in both single-sensor measurements and network-wide statistics. Data from recent high-density seismic studies, Fig. 1(b), show that localized variations in ground structure significantly impact the magnitude of shaking at locations only a few kilometers apart. Consequently, effective quake detection requires algorithms that can learn subtle dependencies among sensor data, and detect changes within groups of dependent sensors. In this sense, quake detection is prototypical of many challenging real-time detection problems, including detecting epidemic outbreaks [22], intrusions in networks [27], and sudden changes in traffic patterns [9].

Particularly challenging in massive networks is dealing with the flood of data. By utilizing even a small fraction of the millions of existing internet-enabled consumer sensor devices, community sensor networks can reach scales where regularly transmitting even summary statistics would

[1] `http://csn.caltech.edu`
[2] `http://qcn.stanford.edu/`
[3] `http://ishakeberkeley.appspot.com/`

be prohibitive. Instead, we adopt a decentralized approach, where sensors individually detect events and only transmit "pick" messages indicating a detection. These individual detections typically will be very noisy, including many false alarms and missed detections. This reduces the server-side problem to one of detecting event signals in a (noisy) binary activation pattern. Event detection may be the primary task, or serve as a precursor for additional data collection and processing.

Standard approaches in decentralized detection [24] assume that the sensors provide i.i.d. measurements conditioned on the occurrence or non-occurrence of an event. In this case, the fusion center would declare a detection if a sufficiently large number of sensors report picks. However, in many practical applications, the particular spatial configuration of the sensors matters, and the i.i.d. assumption is violated. Here, the natural question arises of how (qualitative) knowledge about the nature of the event can be exploited in order to improve detection performance. In this work, we propose to use *sparsification* to optimize detection. In particular, we linearly represent the network-wide noisy, binary activation patterns in a suitable basis, which is carefully chosen so that "typical" activations (associated with the events of interest) are sparsely represented in the basis. This effectively concentrates the signal energy along a small number of basis coordinates. Natural questions, addressed in this work, are thus: When can we expect sparse representations to aid detection? And, which bases are appropriate for this purpose?

As our first major contribution, we consider a wavelet basis that emerges naturally when sensors are clustered hierarchically. We prove theoretically that when the wavelet basis sparsifies the received picks, decentralized detection becomes possible in a noise regime that cannot be handled by a simple network-wide average. We derive strong bounds on the detection rate when events are drawn from a recently proposed *latent tree model* that produce strong localized dependencies and weaker long-range dependencies.

One of the strengths of the wavelet basis is that it can be constructed using as little information as a matrix of pairwise similarity between sensors. However, additional information such as event simulations or measurements of events in the network are often available. Incorporating this information should improve detection. As our second major contribution, we show how modern results from dictionary learning can be used to directly learn sparsifying bases from simulated or measured training data.

As third main contribution, we perform extensive empirical studies of detection using measurements of 1795 earthquakes following the Japanese Tohuku M9.0 quake, quake measurements from the Signal Hill dense seismic study, from the Community Seismic Network as well as simulated virus outbreaks in the Gnutella P2P network.

In summary, our main contributions are:

- New theoretical guarantees about decentralized detection of sparsifiable events,
- A framework for learning sparsifying bases from simulated or measured data, and
- Extensive experiments on real data from three seismic networks, and simulated epidemics in P2P networks.

2. PROBLEM STATEMENT

We are interested in the problem of detecting whether or not some phenomenon (say an earthquake with magnitude above some threshold, or an epidemic) is present at any locations monitored by a massive network of noisy sensors. We model the presence of the phenomenon at locations $1, \ldots, p$ as a binary vector $\mathbf{x} = [x_1, \ldots, x_p] \in \{0,1\}^p$ that is observed by noisy sensors. The Gaussian noise model is a natural choice for sensor observations, where sensor i observes

$$y_i = x_i + \epsilon_i,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. In our seismic detection application, the variables y_i may refer to accelerometer readings of a sensor deployed at location i. This continuous noise model captures how a subset of sensors in areas experiencing shaking observe a shift in the mean of their accelerometer measurements, while the rest of the network observes i.i.d. noise.

The decentralized setting. In many domains, collecting the raw sensor measurements of all sensors would require prohibitive bandwidth to transmit (e.g. the accelerometers in one million smartphones produce ≈ 30 Terabytes of data each day). A natural way to circumvent this bottleneck is to use decentralized detection [24] where sensors individually test their measurements and report the occurrence of a possible event. As an example, the CSN system employs a *hierarchical anomaly detection* approach [8] that allows each sensor to transmit only the results of a local anomaly detection computation (known as a *picking algorithm*) to the fusion center. We can model the resulting picks using a *binary symmetric channel* noise model, where

$$y_i = \begin{cases} x_i & \text{with prob. } 1-\pi \\ 1-x_i & \text{with prob. } \pi, \end{cases}$$

for some error rate $0 < \pi \leq \frac{1}{2}$. The goal of the detection problem is to distinguish the null hypothesis $\mathcal{H}_0$, $x_i = 0$ for all i (i.e., no earthquake present) from the alternate hypothesis $\mathcal{H}_1$, where $x_i = 1$ for one or more i (i.e., the earth is shaking at least at one location i).

Decentralized linear detection. While (decentralized) hypothesis testing in general has been studied extensively, here we focus on the particularly challenging, and not well understood, setting where the patterns $\mathbf{x}$ are *sparse* and have strong noise. This is exactly the case for our motivating example of community seismic networks, where we wish to detect the event as early as possible (i.e., few sensors have been reached yet), and each sensor is very noisy. Formally, we quantify the sparsity of a vector $\mathbf{x}$ as the number of non-zero elements $x_i \neq 0$, denoted by the ℓ_0-norm $||\mathbf{x}||_0$. Generally, we will be interested in quantifying the detection performance as the network grows. We say $\mathbf{x}$ is *sparse* if $||\mathbf{x}||_0$ grows as $p^{1-\alpha}$ for some $1/2 < \alpha < 1$, where a larger α means a sparser signal. Thus, as the number p of sensors grows, the ratio of sensors reached by the event $||\mathbf{x}||_0/p = p^{-\alpha}$ vanishes as $p \to \infty$.

We focus on hypothesis tests of *linear functions of the observations*, i.e. for some matrix $\mathbf{B}$ with columns $\mathbf{b}_1, \ldots, \mathbf{b}_n$, we consider hypothesis tests of the form

$$\max_i \mathbf{b}_i^T \mathbf{y} \lessgtr \tau$$

for some threshold τ. As we will show in this paper, proper

choice of the basis $\mathbf{B}$ can lead to dramatically improved detection performance, i.e., with the same false positive rate much sparser signals (or much higher noise) can be tolerated.

3. DETECTING SPARSIFIABLE EVENTS

Detecting sparse signals in the decentralized setting is fundamentally challenging. Suppose the expected number of errors in the network is p^γ for some $0 < \gamma < 1$, and the per-sensor error rate $\pi = p^\gamma/p$. Could we use the observed number of picks $||\mathbf{y}||_0$ to detect a pattern with $||\mathbf{x}||_0 = p^{1-\alpha} < p^{0.5}$ non-zero entries?

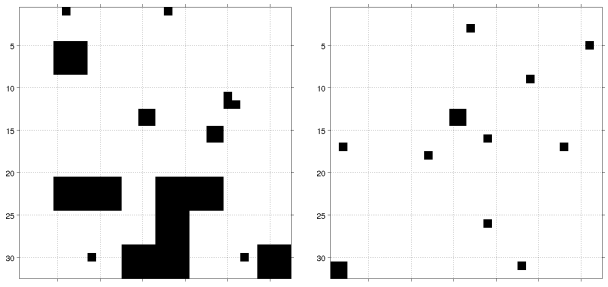
Under both $\mathcal{H}_0$ and $\mathcal{H}_1$, the variance of $||\mathbf{y}||_0$ grows as p^γ. Consider the variable $||\mathbf{y}||_0/\sqrt{p^\gamma}$: it has variance converging to 1 under both $\mathcal{H}_0$ and $\mathcal{H}_1$. Under $\mathcal{H}_0$, its mean is $p^{0.5\gamma}$, and under $\mathcal{H}_1$ its mean is $p^{1-\alpha-0.5\gamma}(1-2\pi) + p^{0.5\gamma}$. For $\gamma > 2(1-\alpha)$, the distributions of $||\mathbf{y}||_0$ under $\mathcal{H}_0$ and $\mathcal{H}_1$ converge, while for $\gamma < 2(1-\alpha) < 1$ the distributions are asymptotically separable. The statistic $||\mathbf{y}||_0$ (classically used in decentralized detection) can only provide reliable detection if the *per-sensor* error rate *decreases* ($\pi = p^\gamma/p \to 0$) as the network size p grows. That is, as the network grows, the sensors must have vanishing error rate for $\mathcal{H}_0$ and $\mathcal{H}_1$ to be separable in the case of sparse signals.

Fortunately, data is rarely unstructured. Even when the network-wide activation pattern is sparse, the activation pattern *within some groups* may be dense and thus more easily detectable. Hierarchical clustering is useful for finding meaningful clusters at a range of scales, and is compatible with efficient data aggregation systems [19] for sensor networks. Recently, hierarchical clustering has been used to define wavelets bases for trees, graphs, and high-dimensional data [11, 23]. For example, a Haar wavelet basis is defined by a hierarchical clustering: whenever two clusters c_l and c_r are merged into a cluster of coarser scale, a unit vector is created,

$$\mathbf{b}_i \propto \left(\frac{1}{|c_l|} \mathbb{1}_{c_l} - \frac{1}{|c_r|} \mathbb{1}_{c_r} \right) \tag{1}$$

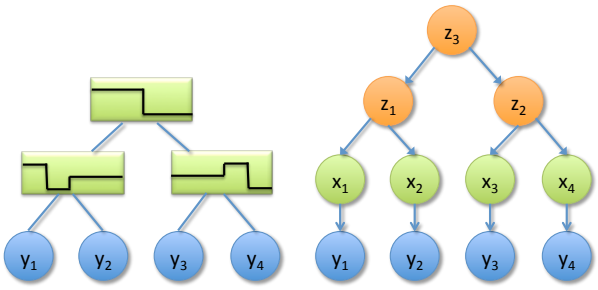
where $\mathbb{1}_c$ indicates the support of cluster c. The clustering algorithm performs $p-1$ merges; the $p-1$ vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{p-1}$ along with the constant vector $\frac{1}{\sqrt{p}}\mathbb{1}_p$ form the columns of an orthonormal matrix $\mathbf{B}$. Multiplying the network observations $\mathbf{y}$ by $\mathbf{B}$ is a projection onto a new basis, where each coordinate $\mathbf{b}_i$ corresponds to the difference between the relative number of activations in a pair of merged clusters. Fig. 3(a) illustrates the basis functions of the transformation. The transform $\mathbf{B}$ has the property that each element $\mathbf{b}_i$ corresponds to *local* averages over sets of related nodes c_l and c_r. Under the assumption that many sets usually activate (or do not activate) jointly, events may be clearly apparent as strong signal along a small number of basis elements. More formally, patterns in $\mathbf{x}$ supported on the clusters used to define $\mathbf{B}$ will tend to be concentrated in a few elements $\mathbf{b}_i$, and so $||\mathbf{B}^T\mathbf{x}||_0 \ll ||\mathbf{x}||_0$. Fig. 2 shows that sparsifiable data is inherently structured.

A basis for detection. Just as a Fourier transform maps an acoustic signal into a coordinate frame that yields insight about the frequency content of the signal, multiplying network activations $\mathbf{x}$ by the basis $\mathbf{B}$ maps the sensor data onto a new coordinate system defined by the hierarchical clustering, and can expose correlated activations. In the following, we prove that with the Haar wavelet basis, the "sparsification ratio" $\frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0}$ plays a central role in achievable error rates. In particular, the following new theorem (with proof outline in the appendix) shows the power of a sparsifying transform to concentrate the signal along at least one new coordinate without concentrating the random noise.



(a) $||\mathbf{x}||_0 = p^{.9}, ||\mathbf{B}^T\mathbf{x}||_0 = p^{.25}$ (b) $||\mathbf{x}||_0 = p^{.7}, ||\mathbf{B}^T\mathbf{x}||_0 = p^{.25}$

Figure 2: Sparsification $||\mathbf{B}^T\mathbf{x}||_0 \ll ||\mathbf{x}||_0$ exploits spatially coherent activation patterns, while small $||\mathbf{x}||_0$ produces fewer activations. The data are drawn from a quad-tree of height 5.



(a) Transform (b) Latent Tree Model

Figure 3: Illustration of the transform (a), constant $\mathbf{b}_0$ not shown. (b) Latent Tree Model for $d = 2$ and $p = 4$. The sensors $\mathbf{y}$ measure the pattern $\mathbf{x} = [x_1, \ldots, x_4]$ up to some noise. The pattern is structured hierarchically; variables z_i represent the variables of the latent tree.

THEOREM 1. *Let $\mathbf{B}$ be a Haar basis that sparsifies a signal $\mathbf{x}$, i.e. $||\mathbf{B}^T\mathbf{x}||_0 = p^{1-\beta}$, $||\mathbf{x}||_0 = p^{1-\alpha}$, $0 < \alpha < \beta < 1$. Let $\mathbf{y}$ be the signal observed through a binary symmetric channel, with error rate π bounded away from $1/2$ (i.e. for some $\epsilon > 0$, $\pi < 1/2 - \epsilon$). Then applying the test $|\mathbf{b}^T\mathbf{y}| > \frac{(1-2\pi)}{2}\sqrt{\frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0}}$ to each of the $p-1$ non-constant basis elements $\mathbf{b} \in \mathbf{B}$ gives false negative rate (FNR) and false positive rate (FPR) bounded as*

$$FNR \leq 2p \exp\left(-\frac{(1-2\pi)^2}{2} \frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0} \right) \to 0 \text{ as } p \to \infty,$$

$$FPR \leq 2p \exp\left(-\frac{(1-2\pi)^2}{2} \frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0} \right) \to 0 \text{ as } p \to \infty.$$

This theorem states that *for any* constant error rate π, as the network size p grows, the probability of a missed detection (FNR) and the probability of false alarm (FPR) are driven to 0 by the decision rule that declares "event"

when $|\mathbf{b}^T\mathbf{y}|$ exceeds the specified threshold, for any of the $p-1$ non-constant basis elements $\mathbf{b}$. For comparison, recall that reliable detection using the network-wide pick count $||\mathbf{y}||_0$ (the standard statistic for decentralized detection) requires the error rate π to rapidly decay to zero as p grows.

The sparsifying basis $\mathbf{B}$ thus enables reliable detection in a broad noise regime that cannot be detected by the network-wide average. This insight shows that indeed quality of sensors can be traded against quantity. Of course, this strong result assumes that the event signal $\mathbf{x}$ is sufficiently sparsifiable by the basis $\mathbf{B}$. In this paper, we show that this assumption holds both in a natural theoretical model, as well as on real sensor data.

Modeling sparsifiable events. When is sensor data sparsifiable? Let us consider again the natural hierarchical basis $\mathbf{B}$, defined according to Eq. 1 as introduced at the beginning of this section. Singh [23] shows that for this particular basis, the assumption $||\mathbf{B}^T\mathbf{x}||_0 \ll ||\mathbf{x}||_0 \leq \sqrt{p}$ is fulfilled when the pattern $\mathbf{x}$ is drawn from an intuitive class of generative models. For completeness, the model is presented here. In this model, dependencies among sensors are modeled via a *tree* of regular degree d: the leaves correspond to the event occurrence x_i at each sensor, and internal nodes correspond to the occurrence or non-occurrence of an event at a particular region and scale. Let $\ell = 0, 1, \ldots L$ denote the level in the tree, where the activations $\{x_i\}$, $i = 1, \ldots, p$ are leaves at level $L = \log_d p$, and the root is at $\ell = 0$. The internal (non-leaf) nodes in the tree capture multi-scale dependencies among the leaves. Let $\mathbf{z}$ denote all nodes in the tree. The joint distribution of $\mathbf{z}$ factorizes as

$$p(\mathbf{z}) = p(z_0) \prod_{\ell=1}^{L} \prod_{i \in V_\ell} p(z_i | z_{\text{parent}(i)}) \tag{2}$$

where V_l denotes the vertices at layer l. The probability that a node equals its parent is specified by $\gamma_\ell = \ell\beta \log d$. This coupling is weaker near the root and stronger near the leaves, producing multi-scale dependencies. Sufficiently weak dependencies are considered negligible, and so the latent variables $z_i \in \ell_0$ at some initial level ℓ_0 are drawn independently from their parents: $p(z_i = 1 | z_{\text{parent}(i)}) = p(z_i = 1) \propto e^{\gamma_{\ell_0}}$. This approximates distant regions of the network as independent. The conditional probability of a node z_i at $\ell > \ell_o$ is

$$p(z_i | z_{\text{parent}(i)}) \propto \begin{cases} e^{\gamma_\ell} & \text{if } z_i = z_{\text{parent}(i)}, \\ 1 & \text{if } z_i \neq z_{\text{parent}(i)}. \end{cases}$$

Patterns drawn from this model are localized and multi-scale, as illustrated with a quad-tree in Fig. 2.

Bounds for finite networks. When the event $\mathbf{x}$ is drawn via Eq. (2), Singh [23] showed that as the number of sensors goes to infinity, the assumption $||\mathbf{B}^T\mathbf{x}||_0 \ll ||\mathbf{x}||_0 \leq \sqrt{p}$ holds with high probability. However, these results do not clearly indicate whether the bounds are effective for large (e.g. hundreds to tens of thousands of sensors) but finite networks. Next, we provide a stronger bound on the sparsification ratio obtained by the wavelet transform, and explain how Theorem 1 can be strengthened to provide bounds on FNR and FPR for fixed network size.

THEOREM 2. *Let $\mathbf{x}$ be a pattern drawn at random from the latent tree model with uniform degree d and depth $L = \log_d p$. Let $\ell_0 = \frac{\alpha}{\beta}$ and $\gamma_\ell = \ell\beta \log d$ for $\ell \geq \ell_0$, where $0 \leq \alpha \leq \beta \leq 1$. Then for $0 < \epsilon < 1$,*

$$\mathbb{P}\left[\frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0} > \frac{\kappa(\epsilon)}{\log_d p} \cdot \frac{p^{1-\alpha}}{p^{1-\beta}}\right] \geq 1 - 2\exp\left(-\frac{c\epsilon^2}{2} p^{\alpha(\frac{1}{\beta}-1)}\right)$$

where $c = \left(\frac{1}{4}\right)^{(\frac{1}{\alpha}-\frac{1}{\beta}+0.5)}$ and $\kappa(\epsilon) = \frac{(1-\epsilon)}{(1+\epsilon)}\frac{c}{d^2}$ are constant with respect to p.

This result shows that the crucial sparsification ratio $\frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0}$ in Theorem 1 grows at (within a log factor of) the desired rate $p^{1-\alpha}/p^{1-\beta}$, with probability that increases exponentially with network size p. This theorem can be used to derive bounds on FNR and FPR for a specified network size p and model parameters α, β, degree d: the bound is substituted for $\frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0}$ in Theorem 1, and the probability that the above bound does not hold can be added to the resulting FNR and FPR.

4. SPARSIFYING BASIS LEARNING

Sec. 3 shows that if an event is "sparsifiable" we can better separate $\mathcal{H}_0$ and $\mathcal{H}_1$ by projecting (multiplying by a basis $\mathbf{B}$) the observations $\mathbf{y}$ onto a different coordinate system where the signal is concentrated into fewer components (a "sparser representation" of the signal). The Haar wavelet basis is an example of a basis that improves detection of signals with certain structured (hierarchical) dependencies. In general, can we construct or learn a sparsifying basis without assuming such dependencies?

Let $\mathbf{B}$ be an orthonormal matrix and $\mathbf{x}$ a vector of uncorrupted binary activations, Theorem 1 states that the sparsification ratio $\frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0}$ directly impacts the amount of separation between $\mathcal{H}_0$ and $\mathcal{H}_1$. In fact, given that $||\mathbf{x}||_0$ is fixed, the two hypotheses are maximally separated when $||\mathbf{B}^T\mathbf{x}||_0$ is minimized. In other words, we can construct the optimal basis by solving the following optimization problem:

$$\arg\min_{\mathbf{B}} ||\mathbf{B}^T\mathbf{X}||_0, \text{ subject to } \mathbf{B}\mathbf{B}^T = \mathbf{I} \tag{3}$$

where $\mathbf{X}$ is a matrix that contains binary observations as its columns and $||\cdot||_0$ is the sum of non-zero elements in the matrix. The constraint $\mathbf{B}\mathbf{B}^T = \mathbf{I}$ ensures that $\mathbf{B}$ remains orthonormal.

However, direct minimization of $||\mathbf{B}^T\mathbf{X}||_0$ is NP-hard in general [7]. In practice, the ℓ_0-norm is often replaced by the convex and "sparsity-promoting" ℓ_1-norm [4]. This suggests the following relaxation heuristic for (3):

$$\arg\min_{\mathbf{B}} ||\mathbf{B}^T\mathbf{X}||_1, \text{ subject to } \mathbf{B}\mathbf{B}^T = \mathbf{I}, \tag{4}$$

where $||\cdot||_1$ is the maximum absolute column sum of the matrix.

Direct approximation. For large problems, we are interested in efficiently computable heuristics for Eq. (4). *Independent Component Analysis* (`ICA`) is one such approximation, and solves the following optimization problem:

$$\arg\min_{\mathbf{B}} G(\mathbf{B}^T\mathbf{X}), \text{ subject to } \mathbf{B}\mathbf{B}^T = \mathbf{I}, \tag{5}$$

where G is a nonlinear convex smooth approximation to the ℓ_1 penalty function, e.g. $\log\cosh(x)$, $-\exp(-x^2/2)$, and x^4 [13]. Fig. 4 illustrates these functions in relation to the linear penalty function.

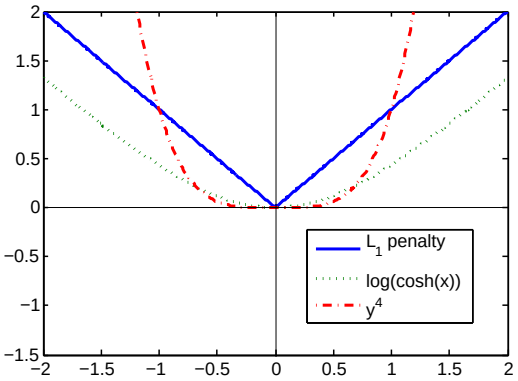


Figure 4: Smooth ℓ_1 approximation functions used in ICA with the linear ℓ_1 penalty function plotted in blue solid line.

Eq. (5) can be solved with stochastic gradient algorithm by taking the derivative of G. However this approach is often slow and requires fine tuning; this leads to the development of "FastICA", an efficient fixed-point algorithm. Implementation details of FastICA and in-depth analysis can be found in [13].

Let $g = G'$, the one unit algorithm for FastICA is given below for completeness.

$\mathbf{b} \leftarrow$ random unit vector
while $\mathbf{b}$ *not converged* **do**
  $\mathbf{b} \leftarrow \mathbb{E}\left[\mathbf{x}g(\mathbf{b}^T\mathbf{x})\right] - \mathbb{E}\left[g'(\mathbf{b}^T\mathbf{x})\right]\mathbf{x}$;
  $\mathbf{b} \leftarrow \mathbf{b}/\|\mathbf{b}\|$;

Algorithm 1: `ICA` one-unit solution

Noise-tolerant relaxed approximation. Ideally we want to learn from noise-free observations $\mathbf{X}$. However, training data constructed from real-world measurements will contain noise or outliers, and instead we are forced to train with $\mathbf{Y}$, which is the observation matrix $\mathbf{X}$ corrupted with noise.

Consequently, we may not be able to obtain the "best" basis by optimizing $\mathbf{B}^T\mathbf{Y}$ as in `ICA`. Instead, we may wish to find a basis that sparsely represents "most of" the observations. More formally, we introduce a latent matrix $\mathbf{Z}$, which can be thought of as the "cause", in the transform domain, of the noise-free signal $\mathbf{X}$. In other words $\mathbf{X} = \mathbf{BZ}$. We desire $\mathbf{Z}$ to be sparse, and $\mathbf{BZ}$ to be close to the observed signal $\mathbf{Y}$. This motivates the next optimization:

$$\arg\min_{\mathbf{B},\mathbf{Z}} \|\mathbf{Y} - \mathbf{BZ}\|_F^2 + \lambda\|\mathbf{Z}\|_1, \text{ subject to } \mathbf{BB}^T = \mathbf{I} \quad (6)$$

where $\|\cdot\|_F$ is the matrix Frobenius norm, and $\lambda > 0$ is a free parameter. Eq. (6) essentially balances the difference between $\mathbf{Y}$ and $\mathbf{X}$ with the sparsity of $\mathbf{Z}$: increasing λ more strongly penalizes choices of $\mathbf{Z}$ that are not sparse.

Although Eq. (6) is non-convex, fixing either $\mathbf{B}$ or $\mathbf{Z}$ makes the objective function with respect to the other convex. The objective can then be solved in an iterative two-step convex optimization process — *Orthogonal Procrustes* [12] and *LASSO with orthonormal design* [3]. The two-step procedure is given below.

Step 1: Orthogonal Procrustes
Fix $\mathbf{Z}$, solve $\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{BZ}\|_F^2, : \mathbf{BB}^T = \mathbf{I}$
  $M \leftarrow \mathbf{YZ}^T$;
  $M = U\Sigma V^T$;
  $\mathbf{B} \leftarrow UV$;
Step 2: LASSO with orthonormal design
Fix $\mathbf{B}$, solve $\min_{\mathbf{Z}} \|\mathbf{Y} - \mathbf{BZ}\|_F^2 + \lambda\|Z\|_1$
  $K \leftarrow \mathbf{Z}^T\mathbf{Y}$;
  $\mathbf{Z} \leftarrow \text{sign}(K) \times \max(|K| - \lambda)$;

Algorithm 2: SLSA two-step convex optimization procedure

The formulation of Eq. (6) and solution in Alg. 2 is equivalent to *Sparse Latent Semantic Analysis* (`SLSA`) [5], which was introduced for applications involving topic models for text data. Here we adopt the name for consistency.

We note that both Eq. (5) and Eq. (6) should be viewed as efficiently computable heuristics for Eq. (3), which is a non-convex optimization over the Stiefel manifold of all size-p orthonormal matrices. As such, they are practical expedients towards the goal of obtaining a sparsifying basis.

5. IMPLEMENTATION IN WSN

In this section, we describe practical issues necessary for using a sparsifying basis for event detection in real-world sensor networks. We highlight how the previous problem formulation can be separated into two computational steps:

- *Offline training* of basis learning and detection threshold selection;
- *Online detection* via decentralized detection or in-network data aggregation.

5.1 Offline Training

The three sparsifying bases (`haar` wavelet, `ICA`, `SLSA`) considered in this paper can be easily implemented and are available in many off-the-shelf optimization packages. Basis learning in small networks ($p < 100$) is very fast in general (within seconds) and can be done online. For larger networks ($p > 500$), offline training may be more suitable.

Basis learning. Learning a basis for p sensors requires at least p measurements of the network. If this is not available (e.g. a seismic network with 1000 sensors may not yet have observed 1000 earthquakes), then simulations provide a practical way to supplement real data. One advantage of using simulations in this way is that while simulations may be slow and compute-intensive, the learned basis produces a fast and efficient detection rule. In Sec. 6, we empirically assess the amount of data required to train a good basis and present two case studies using only data generated from simulations.

Selecting the detection threshold. An event is reported whenever $|\mathbf{b}_i^T\mathbf{y}| \geq \tau$ for any non-constant $\mathbf{b}_i \in \mathbf{B}$. The threshold τ is typically chosen as a value that satisfies constraints on the false positive rate during cross validation with historical data of event observations. This approach does not rely on positive training examples, and so a threshold τ can be learned using only the noise profile of each sensor. Suppose sensors $i = 1, 2, \ldots, p$ have binary error rates

$\pi_1, \ldots, \pi_p$, we have $\mathbb{E}\left[|\mathbf{b}^T\mathbf{y}|\right] = \sum_i^p b_i\pi_i$. Given that the basis is orthonormal, under $\mathcal{H}_0$, Hoeffding's Inequality states that

$$\mathbb{P}\left[|\mathbf{b}^T\mathbf{y}| > \tau\right] \leq \exp\left(-2\left(\tau - \mathbb{E}\left[|\mathbf{b}^T\mathbf{y}|\right]\right)^2\right).$$

By setting the right hand side to a false positive rate constraint, we can easily derive a threshold that satisfies the system requirement. In particular, in order to ensure that $|\mathbf{b}^T\mathbf{y}| \leq \tau$ for all $\mathbf{b} \in \mathbf{B}$ (i.e., no false alarm happens) with probability at least $1 - \delta$, it suffices to choose

$$\tau = \max_{\mathbf{b}\in\mathbf{B}} \mathbb{E}\left[|\mathbf{b}^T\mathbf{y}|\right] + \sqrt{\frac{1}{2}\log\frac{p}{\delta}}.$$

This approach is similar in flavor to the threshold selection method in [8].

5.2 Online Detection

At runtime, the fusion center collects information from the sensors and applies the threshold τ to the statistics $|\mathbf{B}^T\mathbf{y}|$. Depending on the network structure, this aggregation can be done in-network.

Decentralized Detection. The proposed sparsifying bases are suitable for both measurements from binary or other real-valued sensors. However, in large sensor networks such as the ones mentioned in Sec. 1, it is infeasible to constantly stream raw measurements to the fusion center. Instead, it may be desirable to offload the computation from the fusion center to each sensor locally so that only a small amount of information (e.g. a single message) is communicated infrequently when a significant signal is detected. For example, sensors in the Community Seismic Network perform *local anomaly detection* and communicate "abnormal" accelerations (using hypothesis testing) as a binary signal [8].

In-network Aggregation. If the learned basis exhibits hierarchical structure such as the `haar` wavelets inherently do, then it may be possible to adopt in-network aggregation to reduce transmission cost. This takes advantages of the resemblance of network communication topology and basis hierarchy. For example, by using the number of hops needed to communicate between a pair of nodes as a measure of the dissimilarity of two sensors, hierarchical clustering produces the transforms $\mathbf{B}$ supported over groups of communication-efficient clusters. These clusters may compute the transform $\mathbf{B}^T\mathbf{y}$ in a bottom-up fashion while simultaneously testing for detection.

For bases that lack obvious spatial hierarchy, it is possible to adaptively build a routing tree to minimize the communication distance between groups of sensors that tend to co-activate in a sparse sensor setting [10].

6. EXPERIMENTS

We empirically evaluate the detection performance of the three sparsifying bases: `SLSA`, `ICA`, and hierarchical wavelets (`haar`) trained and tested on both simulated and real measurements in different domains. The experimental setup is summarized here.

Baseline algorithms. In keeping with our focus of very large community sensor networks, we compare against baselines that could potentially be computed for real-time detection on tens of thousands of sensors, and that are naturally suited to the client-server communication model of internet-enabled sensors.

- `avg`: network-wide average, $1/p\sum_i^p y_i$;
- `max`: single sensor maximum, $\max_i y_i$;
- `SS`-k: scan statistics that aggregates the k-nearest neighbors for each sensor [20];
- `SS`-r: scan statistics that aggregates all sensors within a radius r for each sensor [20].

Evaluation data sets. The data sets include

- **Synthetic** data from latent tree model, 1296 nodes;
- **Gnutella P2P network**: 1769 nodes;
- **Japan seismic network**: 721 nodes;
- **CSN seismic network**: 128 nodes;
- **Long Beach** seismic network: 1,000 nodes.

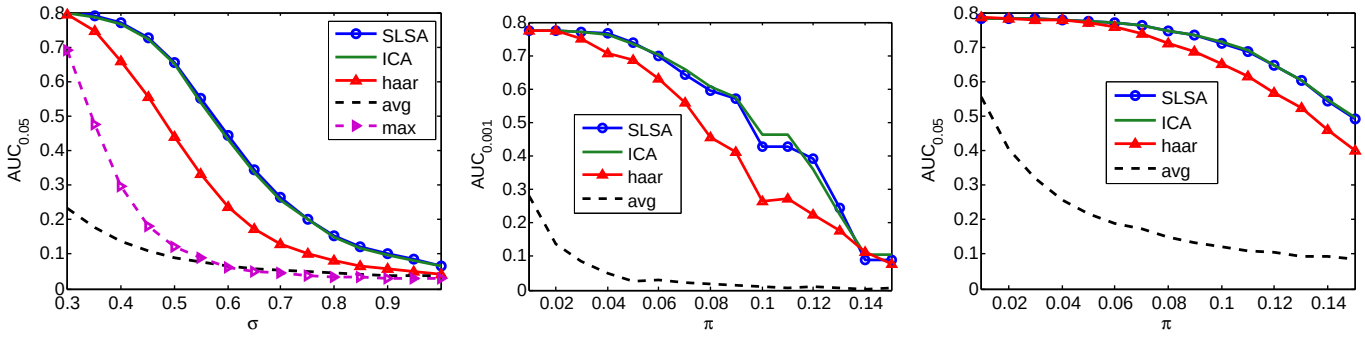
Evaluation metrics and goals. We adopt two metrics in the evaluation of detection performance:

- AUC_f: measures the area-under-curve (AUC) in the Receiver Operating Characteristic (ROC) curve only for false positive rates between 0 and $f, f \leq 1$. The integral AUC_f takes values in $[0, f]$ and is normalized to 1 for simplicity. E.g. $AUC_{0.05} = 0.8$ indicates that the detection performance reaches 80% of the optimal performance under the false positive constraint of 5 false alarms every 100 tests.
- *Detection time*: the time it takes for the test statistics to exceed a threshold that is selected to satisfy a certain system false positive requirement. Rapid and reliable detection is a key requirement for many time sensitive applications. For example, in earthquake response (sub-)seconds improvement in detection time can allow utility companies to shut down large transformers that are responsible for long and costly recovering period after occurrence of a major earthquake.

6.1 Synthetic Data

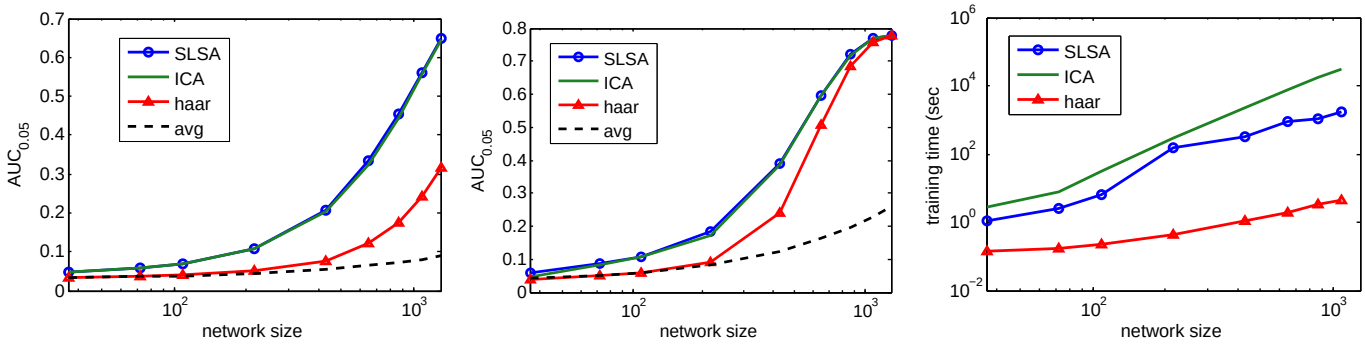
We generate samples from the latent tree model for network activation as described in Sec. 3. The tree contains $p = 1296$ leaf nodes with degree $d = 6$ and depth $L = 4$. We choose the sparsifying parameters $\alpha = 0.5$ and $\beta = 0.95$ so that that the expected number of total activations $||\mathbf{x}||_0 < \sqrt{p}$ is sparse. Of the three bases, `haar` is constructed from the known tree model whereas `ICA` and `SLSA` are trained with 20,000 samples drawn from the model. The bases are tested on 20,000 separate samples corrupted with Gaussian or binary channel noise. For the Gaussian noise case, the range of σ is chosen to satisfy the weak signal constraint, i.e. $\sigma > \frac{1}{\sqrt{2\log p}} = 0.2641$.

Fig. 5 shows that all three bases outperform the naive baselines under both Gaussian and binary noise. Note that, perhaps surprisingly, both the learned `ICA` and `SLSA` outperform `haar` even though the latter is constructed from the known latent tree model.



(a) Gaussian noise (b) Binary noise, $AUC_{0.001}$ (c) Binary noise, $AUC_{0.05}$

Figure 5: Comparing the three bases — `SLSA`, `ICA`, `haar` to baselines — global average (and single max in (a)) on a synthetic data set generated from the latent tree model. Figures (b) and (c) evaluate two different false positive constraints. The learned bases significantly outperform the baselines under strong noise.



(a) Gaussian noise, $\sigma = 0.5$ (b) Binary noise, $\pi = 0.04$ (c) Training time

Figure 6: Detection performance as a function of network size $p = [36, 72, 108, 216, 432, 648, 864, 1080, 1296]$ using all 20,000 training samples. The learned bases show more than 5x performance improvement compared to the baselines in (a) and (b).

Next we study how the network size and the number of training samples affect the quality of learned basis and detection performance.

Increasing network size. We perform basis learning with subsets of the network, using $p = [36, 72, 108, 216, 432, 648, 864, 1080, 1296]$ sensors and $n = 20,000$ training samples. Fig. 6 shows that the detection performance of the learned bases grows more than 5x faster than the baseline. Note that `haar` is now learned from data; this accounts for the slight inferior performance compared to that in Fig. 5.

Increasing number of training samples. With the network size fixed, we evaluate bases learned from increasing numbers of training samples $n = [20, 100, 200, 1000, 2000, 4000, 10000, 15000]$. Fig. 7 shows that `haar` outperforms at smaller training size since it assumes a simple hierarchical structure. It also shows that it takes only 2,000 samples for `ICA` and `SLSA` to achieve the same detection performance as using all 20,000 samples.

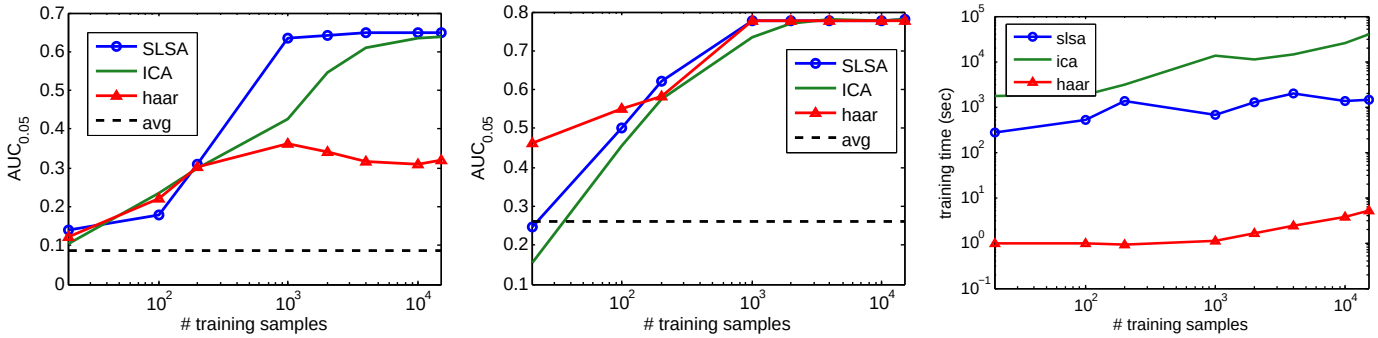
6.2 Gnutella P2P network data

Our next set of experiments simulate virus outbreaks on a peer-to-peer network. We obtain a snapshot of the Gnutella P2P file sharing network[4] through the Stanford Network Analysis Project (SNAP). 1,769 nodes of the highest degree of connectivity were selected from this network for the experiment. Fig. 8(a) visualizes part of this sub network. We simulate 40,000 outbreak events – "*cascades*" – that mimic virus outbreaks on this directed network. We adopt the independent cascade model, where a starting node is picked at random, and whenever a node r is infected, a connected node w is infected with decreasing probability as a function of distance to r.

Here, `haar` is constructed as spanning tree wavelet basis, using the known network structure [14] and Wilson's uniform spanning tree (UST) sampling method on a directed graph via random walk [26]. We also apply the subset scan baseline `SS-k` [20] for reference. The parameter k is "optimally" selected based on the prior knowledge that on average between 10 and 30 nodes are activated in each event in the cascade model.

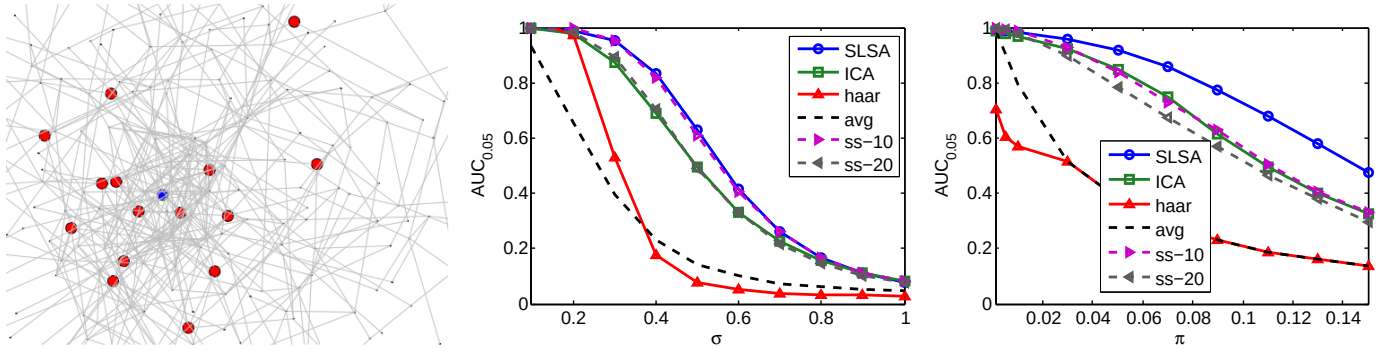
Fig. 8(b) and Fig. 8(c) compare the detection performance evaluated on 40,000 testing samples. Both `SLSA` and `ICA` demonstrate superior detection performance compared to the state of the art algorithms that use additional prior knowledge of the network.

[4] `http://snap.stanford.edu/data/p2p-Gnutella05.html`



(a) Gaussian noise, $\sigma = 0.5$ (b) Binary noise, $\pi = 0.04$ (c) Training time

Figure 7: Detection performance as a function of of training data size. (a)(b) shows it only takes approximately 2,000 samples for both ICA and SLSA to achieve the same performance as using all 20,000 samples. SLSA is 10 times faster to train than ICA as shown in (c).



(a) Sample network activation (b) Gaussian noise (c) Binary noise

Figure 8: Experiment with Gnutella-P2P network. (a) visualizes $\sim 1/10$ of the total network with a sample activation pattern colored. Blue: first infected node, Red: nodes subsequently infected through the cascade. (b)(c) shows that the learned bases achieve and exceed the state of the art algorithms that use additional prior knowledge of the network.

6.3 Japan seismic network data

Next we turn to perhaps one of the most robust and long-running sensor networks in the world – the Japan seismic network. We obtain 48-hour, 150 GB of recordings from 721 *Hi-net* NIED seismometers for the dates March 18 and 19, 2011, just one week after the Tohoku M9.0 earthquake on March 11, 2011. On both days, 1,000+ events ranging from M1.0 - M6.0 were recorded in the the Japan Meteorology Agency catalog[5]. Many events triggered clustered activations as observed in Fig. 9(b).

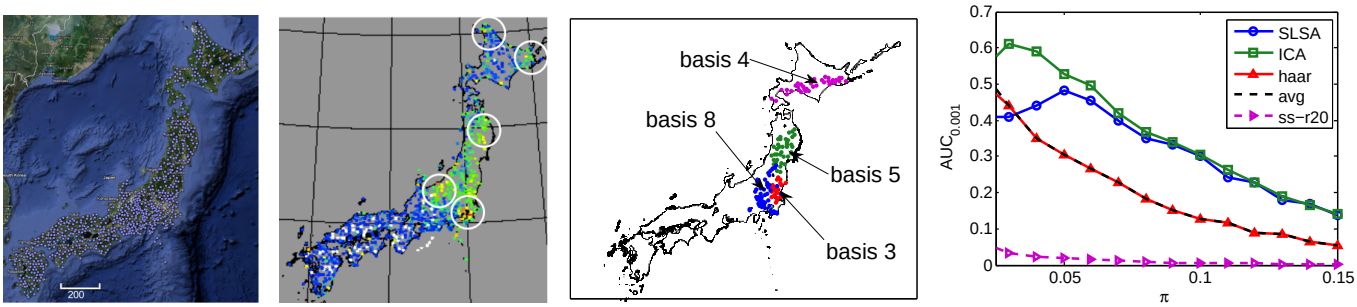
For all 1795 events recorded on March 18, 2011, 10 snapshots of network activations at a two-second period were taken after the first detection at each event to construct the training data set of [p x n] = [721 x 17950]. The learned bases are tested on the first one-second data of the 1324 events recorded on March 19, 2011. We added binary noise of different error rate to control the problem complexity.

For the comparison with the SS-r baseline, the aggregation distance r is selected to be 20km which is roughly the distance covered by the seismic waves in a 2-second period. Fig. 9(d) presents the performance in detecting within two seconds of event arrival under a very small false positive constraint of 0.001. Of the three learned bases, both ICA and SLSA show significant gain in detection power, whereas haar has no improvement over the avg baseline. Perhaps surprisingly SS-r20 performs very poorly in comparison. An explanation is that most of the events during this period originated from the ocean and affects an array of stations along the coast. However, this pattern is not captured by the fixed radius subset scan construction. This explanation is supported by the plot of four prominent basis elements from ICA in Fig. 9(c). This example demonstrates the limited detection capability of subset scan for unknown patterns and the power of learning-based detection algorithms such as ICA and SLSA.

[5] http://www.hinet.bosai.go.jp/REGS/JMA/?LANG=en

6.4 Dense and participatory seismic networks

Lastly, we return to the dense participatory sensor networks that served as a motivating example in Sec. 1. We consider two dense, real-world seismic networks in Southern California. We show that good bases can be learned without historical sensor data: instead, we simply use basic earthquake simulators to generate the binary activation patterns for training, as discussed in Sec. 5.1. In the shortage of testing data – only a small number of events have



(a) (b) (c) 4 prominent ICA basis (d) Detection performance

Figure 9: Japan's seismic network. The 721 Hi-net stations in (a) frequently exhibit localized activation patterns as circled in (b), which plots raw accelerations (red: large shaking, blue: no shaking). The learned bases are able to capture these nonuniform patterns with basis elements such as the ones in (c) and show 2x better detection performance compared to the baselines (d), while algorithms with hard-coded patterns such as SS-r20 fail to perform well in this scenario.

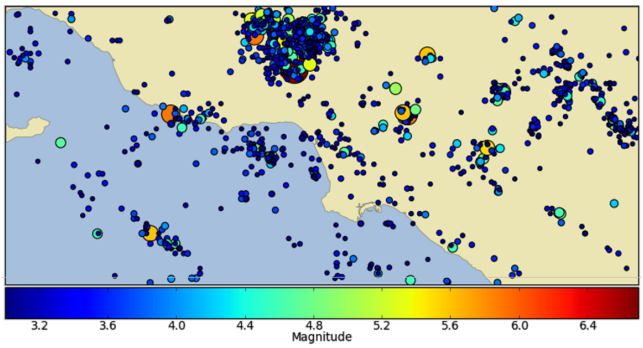


Figure 10: Southern California quakes since 1973

been recorded by these networks, not enough to reliably compute AUC scores – the detection performance is evaluated in terms of detection time with detection thresholds computed as described in Sec. 5.1. This measure of time is critical in many applications; seconds or sub-second savings may enable automated responses that prevent huge loss of capital and lives.

Generating training data. We generate training data using a basic earthquake simulator in the following two steps. First we randomly sample an earthquake from a prior distribution of seismic events in Southern California that is constructed from a list of historic earthquakes (Fig. 10) available in the USGS database[6].

Then time sequences of sensor activations are generated from an earthquake model that computes the expected wave arrival time with the encoded speed of seismic waves and distance to the hypocenter. This model is simplistic compared to many state-of-the-art earthquake simulators, yet captures qualitative spatio-temporal dependencies. An activation probability similar to that in [18] is used to simulate signal attenuation for unreliable noisy sensors.

Community Seismic Network. We simulate 1,000 network activation snapshots for 128 Community Seismic Network [6] sensors as described above. After training, each algorithm is then evaluated on its ability to detect four recent events using real measurements recorded by the network. Fig. 11(a) shows the spatial layout of the network and the hypocenters of the four events. Fig. 11(b) summarizes detection performance: the bases learned from simple simulations in general achieve faster detection than other algorithms, e.g. 8 seconds faster in detecting the Beverly Hills event. Note that `ICA` performs better than `SLSA`, as simulations are noise-free.

Long Beach Array. The Long Beach network consists of approximately 5,000 sensors covering an area of 5 x 7 km. The network was deployed for 6 months during the first half of 2011 to provide detailed images of the Signal Hill Oil Field in Long Beach, California. During the deployment period, a total number of 5 detectable earthquakes were recorded by the network (Fig. 12(a)). Fig. 1 is a visualization of one of the events.

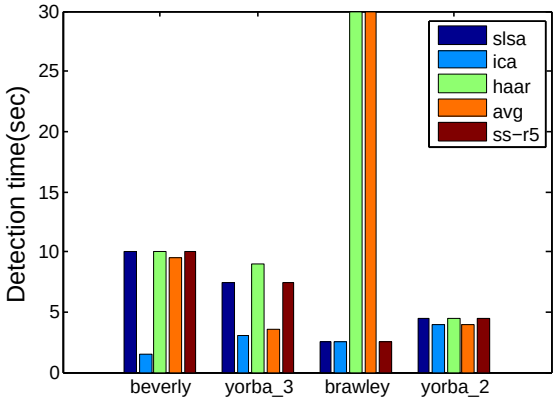
We take a subset of 1,000 sensors and train the sparsifying bases with 2,000 simulated events. The results in Fig. 12(b) show that the learned bases detect on average 0.1 second faster, especially for the more difficult events that are smaller and further away. This improvement in detection time is significant considering that it only takes about one second for the quake to travel through the entire network.

7. RELATED WORK

Sparse detection. Detecting a sparse signal in the presence of strong noise is challenging without placing some assumptions on the class of signals. [23], [11], and [17] propose multi-scale bases for signals with tree structure. [14] further extends the analysis to graph structured network defined over spanning trees. The work of Singh et. al [23] is particularly relevant, as they identify the asymptotic limits of detectability for the orthonormal basis and generative models that we consider here in the centralized setting under Gaussian noise. In contrast, we focus on the decentralized case with binary channel noise, and provide theoretical guarantees that hold even in the non-asymptotic regime. [2] describes detecting sparse binary patterns with a variety of combinatoric structures under Gaussian noise. Lower bounds on minimax detection rates are given, and it is shown

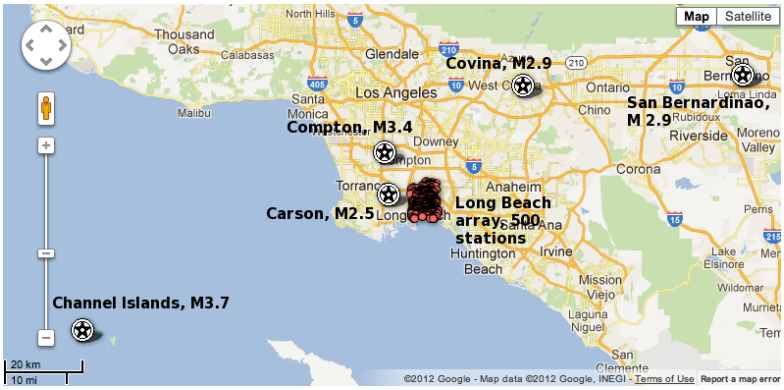
[6] `http://earthquake.usgs.gov/earthquakes/eqarchives/epic/`

(a) Sensors (red) and events (starred)

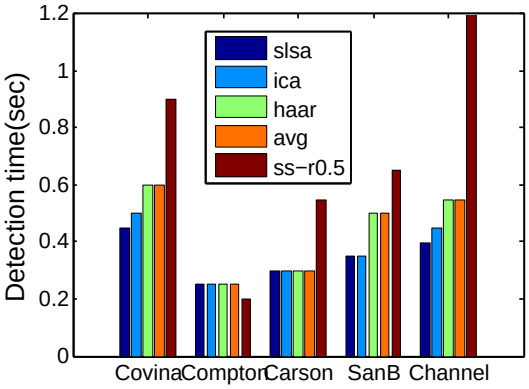


(b) Detection time comparison

Figure 11: CSN network. (a) plots the layout of 128 sensors and epicenter of 4 recorded events. (b) The learned bases detect on average several seconds faster than the baselines under the constraint of at most one false alarm a year.



(a) Sensors and events layout



(b) Detection time comparison

Figure 12: Long Beach array. (a) shows the layout of 1,000 stations and 5 recorded events. (b) Under the constraint of at most one false alarm a year, the learned bases detect on average 0.1 seconds faster than the baselines, which is significant considering it only takes 1 second for the seismic wave to travel through the network and only 0.5 seconds for the network to be saturated with signals.

that forms of the scan statistic achieve within a log factor of these rates. However, these results are asymptotic and appear computationally intractable for large sensor networks.

Scan statistics. Spatial and space-time scan statistics were first developed to monitor health data and disease outbreaks [15]. The main idea is to evaluate all subsets of the data for possible events. Of course, enumerating all possible $O\left(2^n\right)$ subsets is infeasible for even moderately sized problems, so later refinements test only certain subsets of distinct sizes and shapes [21, 20]. This approach reduces the complexity to $O\left(n^2\right)$ (and to $O\left(n\right)$ in [20]) but may also impair the detection performance if important signals are not well captured by tested subsets.

Basis Learning. Learning a sparsifying basis is intimately related to dictionary learning and topic models. Dictionary learning [1] attempts to find an overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{n,K}$, $K > n$ that can sparsely encode signals in $\mathbb{R}^n$. Similarly, topic models [25] represent text data as a linear combination of a "topics", e.g. vectors of word frequencies. Topic models seek topics that sparsely approximate the documents, though the number of topics is significantly less than the number of words (i.e. the topic matrix is undercomplete).

ICA is a transformation method developed to recover nongaussian, statistically independent components $\mathbf{z}$ from their linear combination $\mathbf{x}$ [13], assuming that the linear transformation matrix $\mathbf{B}$ is orthonormal and $\mathbf{x} = \mathbf{Bz}$. The nongaussianity is required because the orthogonal transformation of any number of Gaussian distributions is inseparable. Strongly nongaussian data (i.e., having a very different distribution from Gaussian) is often sparse, and so nongaussianity is yet another measure of sparsity. ICA has enjoyed most success in signal separation and unsupervised feature learning. Recent work has extended it for overcomplete dictionary learning [16].

8. CONCLUSIONS

Motivated by quake detection in large community seismic networks, we proposed learning a *sparsifying basis* to en-

able detection of sparse event patterns in the decentralized setting. We obtain theoretical bounds on the power of sparsification using a `haar` wavelet basis and obtained strong bounds on error rates for events produced by the latent tree model that can be evaluated for any network size. These results strengthen and complement previous work on the limits of detectability of sparse patterns in Gaussian noise.

We then extended the intuition for the wavelet transform's success - its ability to concentrate signals with a small number of basis elements - and obtained a general framework to learn sparsifying bases for detection. We considered two optimizations, `ICA` and `SLSA`, for learning a basis that approximately maximizes sparsification, and explain how it can be implemented in sensor networks using real or simulated data in the absence of sufficient training data.

Finally, we thoroughly evaluate the detection performance of the sparsifying bases on several problem domains: simulated virus outbreaks on the Gnutella P2P network; detecting quakes following the Tohuku M9.0 event in the Japan seismic network with bases learned from network measurements; and detecting quakes recorded by the dense Long Beach and Community Seismic Network sensors using simulated measurements for training. In all domains, learned bases outperform previous state-of-the-art algorithms. We believe that our insights are an important step towards solving challenging detection problems using large-scale, noisy, participatory sensor networks.

Acknowledgments. The authors would like to thank their Caltech collaborators working on the Community Seismic Network project: Prof. Robert Clayton and Dr. Richard Guy of Geophysics; Prof. Thomas Heaton, Dr. Monica Kohler, and Ming-Hei Cheng from Earthquake Engineering; Prof. Mani Chandy and Michael Olson from Computer Science; Dr. Julian Bunn, Dr. Michael Aivazis, and Leif Strand from the Center for Advanced Computing Research. Special thanks to Prof. Robert Clayton and NodalSeismic Inc. for the Long Beach array data set and Prof. Masumi Yamada and NIED for the Japan data set. This research is supported in part by a grant from the Betty and Gordon Moore Foundation, by NSF award CNS0932392 and ERC StG 307036.

9. REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: Design of dictionaries for sparse representation. *Proc. of SPARS*, 5:9–12, 2005.

[2] E. Arias-Castro, E. Candes, and A. Durand. Detection of an anomalous cluster in a network. *The Annals of Statistics*, 39(1):278–304, 2011.

[3] P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data*. Methods, Theory and Applications. Springer, June 2011.

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM review*, 2001.

[5] X. Chen, Y. Qi, B. Bai, Q. Lin, and J. G. Carbonell. Sparse latent semantic analysis. *NIPS Workshop*, 2010.

[6] R. Clayton, T. Heaton, et al. Community seismic network. *Annals of Geophysics*, 54(6), 2012.

[7] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98, Mar. 1997.

[8] 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 *Information Processing in Sensor Networks (IPSN)*, 2011.

[9] R. Ganti, I. Mohomed, R. Raghavendra, and A. Ranganathan. Analysis of data from a taxi cab participatory sensor network. *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 197–208, 2012.

[10] J. Gao, L. Guibas, N. Milosavljevic, and J. Hershberger. Sparse data aggregation in sensor networks. In *Information Processing in Sensor Networks (IPSN)*, pages 430–439. ACM, 2007.

[11] M. Gavish, B. Nadler, and R. Coifman. Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning. In *Proc. International Conf. on Machine Learning, Haifa, Israel*, 2010.

[12] J. Gower and G. Dijksterhuis. *Procrustes Problems*. Oxford Statistical Science Series. OUP Oxford, 2004.

[13] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, June 2001.

[14] A. Krishnamurthy, J. Sharpnack, and A. Singh. Detecting Activations over Graphs using Spanning Tree Wavelet Bases. *arXiv.org*, stat.ML, June 2012.

[15] M. Kulldorff. A spatial scan statistic. *Communications in Statistics - Theory and Methods*, 26(6):1481–1496, Jan. 1997.

[16] Q. Le, A. Karpenko, and J. Ngiam. ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning. *Neural Information Processing Systems*, 2011.

[17] A. Lee, B. Nadler, and L. Wasserman. Treelets—an adaptive multi-scale basis for sparse unordered data. *The Annals of Applied Statistics*, 2(2):435–471, 2008.

[18] A. Liu, M. Olson, J. Bunn, and K. M. Chandy. Towards a discipline of geospatial distributed event based systems. In *DEBS '12: Proc. of the 6th ACM International Conf. on Distributed Event-Based Systems*, July 2012.

[19] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. Tinydb: An acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems (TODS)*, 30(1):122–173, 2005.

[20] D. B. Neill. Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2), 2012.

[21] D. B. Neill and A. W. Moore. Rapid detection of significant spatial clusters. In *Proc. of the tenth ACM SIGKDD ...*, 2004.

[22] G. Shmueli and H. Burkom. Statistical challenges facing early outbreak detection in biosurveillance. *Technometrics*, 52(1):39–51, 2010.

[23] A. Singh, R. Nowak, and R. Calderbank. Detecting Weak but Hierarchically-Structured Patterns in Networks. In *The 13th International Conf. on Artificial Intelligence and Statistics (AISTATS)*, Sept. 2010.

[24] J. Tsitsiklis et al. Decentralized detection. *Advances in Statistical Signal Processing*, 2:297–344, 1993.

[25] Q. Wang, J. Xu, H. Li, and N. Craswell. Regularized latent semantic indexing. In *Proc. 34th Internat. ACM SIGIR Conf. on Research and Development in Information, SIGIR*, volume 11, pages 685–694, 2011.

[26] D. B. Wilson. Generating random spanning trees more quickly than the cover time. *Proc. of the twenty-eighth annual ACM symposium on Theory of computing*, pages 296–303, 1996.

[27] C. Zhou, C. Leckie, and S. Karunasekera. A survey of coordinated attacks and collaborative intrusion detection. *Computers & Security*, 29(1):124–140, 2010.

APPENDIX

Proof of Theorem 1. Let k be the size of clusters merged by a non-constant basis element $\mathbf{b}$. Let $\tau = \frac{(1-2\pi)}{2}\sqrt{\frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0}}$. Under $\mathcal{H}_0$, $\mathbb{E}\left[|\mathbf{b}^T\mathbf{y}|\right] = 0$, and is the sum of $2k$ terms (k from each cluster) taking values $\{-\frac{1}{\sqrt{2k}}, 0, \frac{1}{\sqrt{2k}}\}$. Hoeffding's inequality gives $\mathbb{P}\left[|\mathbf{b}^T\mathbf{y}| \geq \tau\right] \leq 2\exp\left(-\tau^2\right) \to 0$. Taking the union bound, FPR $\leq 2p\exp\left(-\tau^2\right) \to 0$. Under $\mathcal{H}_1$, for some $\mathbf{b}$, $\mathbb{E}\left[|\mathbf{b}^T\mathbf{y}|\right] \geq \sqrt{\frac{||\mathbf{x}||_0}{||\mathbf{B}^T\mathbf{x}||_0}}(1-2\pi)$. As under $\mathcal{H}_0$, Hoeffding's inequality bounds the probability of deviation by τ from the mean (conveniently, $\mathbb{E}\left[|\mathbf{b}^T\mathbf{y}|\right] - \tau > \tau$): $\mathbb{P}\left[|\mathbf{b}^{(k)}\mathbf{y}| \leq \tau\right] \leq 2\exp\left(\tau^2\right) \to 0$. Taking the union bound over p basis elements, FNR $\leq 2p\exp\left(-\tau^2\right) \to 0$.

Proof of Theorem 2. Let $X_T^{(i)}$ denote the leaves in the i^{th} subtree rooted at level ℓ_0. In the latent tree model, the nodes at level ℓ_0 are independent, and so the numbers of active leaves in each subtree $\|X_T^{(i)}\|_0$, $i = 1, \ldots, d^{\ell_0}$ are i.i.d.

LEMMA 1. $\mathbb{E}\left[\|X_T^{(i)}\|\right] \geq c \cdot p^{1-\alpha}p^{-\frac{\alpha}{\beta}}$ *where* $c = \left(\frac{1}{4}\right)^{(\frac{1}{\alpha}-\frac{1}{\beta}+0.5)}$ *is constant with respect to* p.

Let $W_T^{(i)} = \frac{\|X_T^{(i)}\|_0}{p^{1-\frac{\alpha}{\beta}}}$, and $W = \sum_i W_T^{(i)}$. There are $p^{1-\frac{\alpha}{\beta}}$ leaves in each $X_T^{(i)}$, so $W_T^{(i)} \in [0,1]$. There are $p^{\frac{\alpha}{\beta}}$ subtrees, so by Lemma 1, $cp^{-\alpha}p^{\frac{\alpha}{\beta}} < \mathbb{E}[W]$. Hoeffding's inequality gives, for $0 < \epsilon < 1$,

$$\mathbb{P}\left[\|X\|_0 < (1-\epsilon)cp^{1-\alpha}\right] \leq \exp\left(\frac{c\epsilon^2}{2}p^{\alpha(\frac{1}{\beta}-1)}\right)$$

Next, we will say an *edge flip* occurs at level ℓ when a node at level ℓ does not equal its parent. The number of non-zero coefficients is bounded as $||\mathbf{B}^T\mathbf{x}||_0 \leq dL \cdot F$, where F is the number of edge flips in the tree. An edge at level ℓ flips with probability $q_\ell = 1/(1+d^{\beta\ell}) < d^{-\beta\ell}$, so we find that

$$\mathbb{E}[F] = \frac{d^{(1-\beta)(L+1)} - d^{1-\beta}}{d^{(1-\beta)} - 1} < d \cdot d^{L(1-\beta)}$$

Let $\bar{\mu} = d \cdot d^{L(1-\beta)}$ For $0 < \epsilon < 1$, the Hoeffding inequality gives

$$\mathbb{P}\left[\left\|\mathbf{B}^T x\right\|_0 > (1+\epsilon)d^2\log_d p \cdot p^{(1-\beta)}\right] \leq \exp\left(-\frac{\epsilon^2}{3}d \cdot p^{(1-\beta)}\right).$$

□