HDR TRIPODS: From detection to reaction - computation in resource constrained sensor networks

April 28, 2019

1 Summary

Any agent, be it a human, an animal or a robot, has to react to it's environment to take advantage of opportunities and to avoid dangers. The transformation of events to reaction can be partitioned into three steps: **(1) physical events** are transformed by sensors into **raw data**, **(2)** Computation transforms the **raw data** into a **knowledge** (representation of the environment), and **(3)** an **action** is chosen based on the **knowledge**.

The design of the sensors is dominated by considerations of sensitivity and resolution (temporal and spatial). The goal is to detect the smallest, faintest and most transient signals, by exploiting priors on the physical model of signal acquisition, and the geometry of signal representation. Computation is used to reduce raw data into an internal representation and then into actions.

These days the leading architecture of reactive systems is wireless sensor networks. Sensor networks consist of large numbers of small independent units, each with sensors, computation and wireless communication. Such systems are constrained by power and communication bandwidth.

One important consequence of these constraints is *pushing computation to the edge*. Instead of communicating the raw information from each sensor to a central computer, each sensor unit locally computes summaries, or sketches, which are shorter and therefore cheaper to communicate. This also reduces the computation load on the units that receive the information.

2 Framework

This proposal combines several related lines of work. To facilitate the exposition, we start by introducing some terminology and notation that will be used throughout. **Figure 1** describes a simple sensor network that tracks a car using the sound waves the car is emitting.

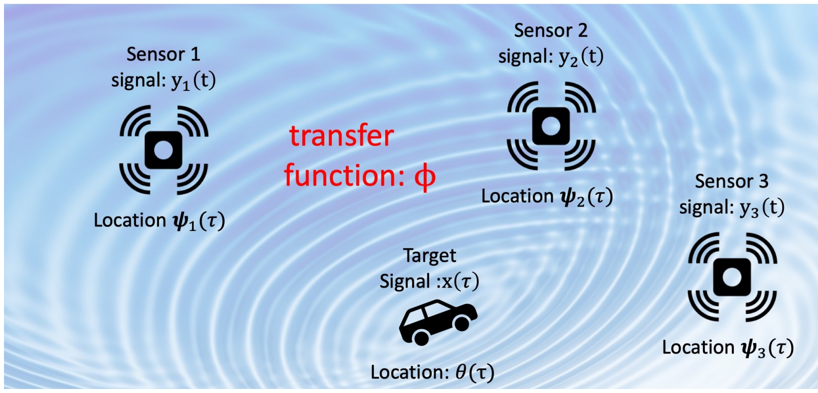

Figure 1: An example of a sensor network. The goal of the network is to track a car. The location of the car as a function of time is $\theta(\tau)$, and the goal of the network to produce $\hat{\theta}(\tau)$. The car emits a sound wave, which we denote by $x(\tau)$. The sound wave travels through the physical environment and arrives at each sensor i, where it is digitized and made into a time sequence $y_i(t)$. The transformation of the signal $x(\tau)$ into $y_i(t)$ is represented by a transfer function ϕ. In this car tracking example $y_i(t) = \phi(x(\tau), \theta(\tau), \psi_i(\tau))$. We are seeking an inverse transformation that would map the measurements vector $y_1(t), y_2(t), y_3(t)$ to an estimate of the car location $\hat{\theta}(t)$

We now expand this simple example into a more general framework. We assume that the network consists of a n sensors and m targets. Sensor i's state at time τ is denoted $\psi_i(\tau)$. Similarly, the state of target j at time τ is denoted $\theta_j(\tau)$. Here and in the rest of this section, we don't specify the spaces in which ψ_i or θ_j are members. These allows for a general introduction which will be made more specific in later sections. When appropriate, we will denote the combined state of all targets by $\Theta(\tau)$ and the combined state of all sensor by $\Psi(\tau)$. The third state component is the state of physical environment in which the targets and sensors reside. We denote the state of the environment by $\mathbf{E}(\tau)$

The targets generate signals, which we call the *raw* signals. We denote the raw signal generated by target i as $\mathbf{x}_i(\tau)$. We denote the collection of all m signals by $\mathbf{X}(\tau)$. On the receiving end, each sensor i captures a digital signal $\mathbf{y}_i$. These digital signals are the inputs to the computations we will discuss. As the signals arrive at physically separated sensors, the computation is inherently distributed. The main goal of this proposal is to develop algorithms that achieve desired tasks with minimal communication between the sensors.

The transfer function $\mathbf{\Phi}$ defines the way by which the raw signals $\mathbf{X}$ are transformed into the digitized signals $\mathbf{Y}$. This function represents both the point transfer function of the physical environment, the analog-to-digital transformation of the physical signal into a discrete time physical signal, and the noise that is added through this process. The transformation is defined by:[1]

$$\mathbf{Y} = \mathbf{\Phi}(\mathbf{X}, \Theta, \Psi, \mathbf{E})$$

The problems we plan to tackle in this proposal are inverse computation problems. We assume that some aspects of the physical space are known, i.e. we know a subset of $\mathbf{X}, \Theta, \Psi, \mathbf{E}$. Given the digitized signals $\mathbf{Y}$, our task is to estimate the unknown parts of the physica space. Reliable methods for computing such estimates exist. However, they typically require high communication bandwidth. The goal of this proposal is to find distributed estimation algorithms that achieve good performance while using significantly less communication.

Some specific tasks

We give a few specific examples of tasks. We will elaborate on some of these tasks below

1. **Target Localization:** Figure 1 depicts an archetypal target localization task. In this case the locations of the sensors Ψ and the state of the environment $\mathbf{E}$ are assumed to be fixed. A typical additional assumption, which is represented in the transfer function $\mathbf{\Phi}$, is that strongest signal corresponds to the straight line of transmission between the target and the sensors. A common approach to target localization is to estimate the delay between the arrival of the signal at different sensors by using some type of cross correlation []. This calculation is performed It is well known that placing sensors far from each other provides the most accurate localization. However, achieving this accuracy with bounded communication between the sensors remains a challenge.

2. **Signal reconstruction** Voice based systems such as speakerphones, and voice activated computers, need to reconstruct the speech signal. Microphone arrays are sensor networks where the sensor is a microphone. Accurately reproducing the speech signal when there is more than one speaker is an open challenge. In this problem the location of the sensors is known, the goal is to reproduce the raw signal $\mathbf{X}$ where the environment $\mathbf{E}$ is unknown and the transfer function $\mathbf{\Phi}$ is complex (reverberating environment).

3. **Optimizing sensor placement:** The accuracy of target localization depends on the location of the target and of the sensors. While the location of the target is not under our control, the location of the sensors is. Methods for optimizing the locations of the sensors will be described in Section sec:sensor-placement.

[1] Note that the transfer function $\mathbf{\Phi}$ operates on the whole sequences, not just on the sequence at a single time τ. That is because signal propagation takes time, so $\mathbf{Y}(t)$ depends on $\mathbf{X}$ at multiple time points.

4. **Mapping the environment** Sometime the goal of the system is to estimate the environment $\mathbf{E}$. One example is to use Radar, Sonar or Lidar to create a 2D or 3D representation of the environment for a smart car. Another example, coming from seismology is to use controlled vibration sources and many acceleration sensors to map the subterranean earth. In these settings the locations of the sensors Ψ and the targets Θ (called transmitters in this context) is fixed and known, as is the raw (transmitted) signal $\mathbf{X}$. The goal is to deduce $\mathbf{E}$ from the collected signal $\mathbf{Y}$. (Peter, does this make sense to you, can you use this formulation in your sections about tomography and dictionaries?)

5. **Monitoring** In many situations the goal of the sensor network is to track the environment, identify trends and detect anomalies. Examples include: Security systems, systems for monitoring patients or the elderly, highway monitoring and factory floor monitoring. Many of these environments are too complex to estimate a fully detailed representation. Instead, we suggest building a statistical model which captures the major degrees of freedom of the environment and the way they relate to major clocks such as time of day and day of week. The challange here is to learn such a model in an unsupervised or weakly supervised way. Kernels, noise-shaping quantization, and sketching methods can be used here. (Rayan, does this make sense? Can you make it more coherent?)

2.1 Target Localization using minimal number of sensors

Yoav : Piya, can you use the notation I defined in the framework section? Also I would like to merge this subsection and the following one, which describes sensing geometry and the goal of minimizing the number of sensors. A sensor network consisting of M sensing units aims to capture information of interest (often described in terms of parameters) regarding the physical environment by acquiring measurements in space (dictated by sensor locations) and in time (dictated by the sampling technique employed at each sensor). In many applications (especially those concerning high-resolution/super-resolution imaging), the goal is to detect certain parameters $\theta_i \in \mathbb{C}^P, i = 1, \ldots, K$ from K targets of interest in the environment by acquiring signals emitted by them.

Yoav : This describes a more general framework, using $\mathbb{C}^P$ (does that mean each coordinate is complex?). What is gained from this generality? maybe drop the general notation? Also how does high resolution/super-resolution fit here? If you have worked on such problem, I suggest you devote a paragraph and cite, rather than just mentioning in passing.

As an example, consider a network consisting of active radar units (for example, those mounted on autonomous vehicles) attempting to create a map of the environment. In this case, K can denote the total number of pedestrians, bicyclist's and other cars and $\theta_i \in \mathbb{R}^3$ for the ith target will consist of its location $\mathbf{x}_i = [x_i, y_i]^T$ and velocity (v_i) parameters, i.e.

$$\theta_i = [x_i, y_i, vx_i, vy_i]^T, \quad 1 \leq i \leq K \tag{1}$$

Yoav : Shouldn't K be estimated? Mathematically the space-time measurements collected at the mth sensing element can be described as

$$y_m(t) = \sum_{i=1}^{K} \phi(\mathbf{d}_m, \theta_i, t) + w_m(t), \quad 1 \leq m \leq M \tag{2}$$

where $w_m(t)$ is the additive noise. Here $\mathbf{d}_m \in \mathbb{R}^3$ denotes the location of the mth sensor and the function $\phi(.)$ characterizes the measurement model (often referred to as the point-spread function in the context of imaging) that depends on the physical laws governing wave propagation, and properties of the medium. Depending on the application and model assumptions, the function $\phi(.)$ can be linear, non-linear, and potentially, even non-convex. However, it can be *partially designed* by choice of senor locations $\mathbf{d}_m$. This will be a key enabler towards obtaining compressed sketches of measurements (or reducing the number of sensing units) while preserving the ability to reliably infer the parameter $\theta_i, 1 \leq i \leq K$.

The basic model assumes targets as point sources, but in many situations, they are distributed. **Piya :** Perhaps Peter can help characterize this model, since SONAR deals with such targets.

The main objective is to obtain estimates $\hat{\theta}_i, 1 \leq i \leq K$ of the parameters of interest (θ_i) using *minimal number of measurements/minimizing the number of sensing elements*. These estimates essentially are some appropriate functions of the spatio-temporal measurements $Y_T = \{y_m(t), 1 \leq m \leq M, 1 \leq t \leq T\}$, i.e.,

$$\hat{\theta}_i(T) = g_i(Y_T) \tag{3}$$

In many scenarios, the parameters of interest can be reliably inferred from the *correlation of the measurements*. In other words, the correlation of the measurements act as a sufficient statistic for the parameters to be inferred. Depending on the application, the correlation matrix can be spatial (when the source signals are stationary), or spatio-temporal (when the temporal dynamics need to be tracked, such as for change-point detection). In these cases, we can effectively summarize the large amount of raw sensor measurements by only retaining and communicating their correlation. **Yoav :** The way I was thinking about it, each sensor has only one signal. In a one scenario, the quantity of interest is the "time delay of arrival" or the time shift of one signal relative to another that would maximize the correlation. Is there anything known about computing this time delay without communicating the whole time series?

Spatial Correlation and Localization: Suppose we compute the spatial correlation between $y_m(t)$ and $y_n(t)$ by averaging over T time samples (the signals are assumed to be stationary over this interval) [2]

$$r_{m,n}(T) = \frac{1}{T}\sum_{t=1}^{T} y_m(t) y_n^*(t) \tag{4}$$

We can summarize the self and cross correlation between M time-series measurements (collected at M sensors) using these M^2 correlation values (collected in the form of a correlation matrix R_T). Owing to the geometry of the measurements, these correlation values directly depend on the sensor locations $\mathbf{d}_m$ (via the mapping $\phi(.)$). Hence, it is natural to ask

1. Can we exploit the geometry of the measurement model to further compress the correlation matrix R_T? What is the role of sensor geometry in this case? We should still be able reliably infer $\theta_i, i = 1, 2, \ldots, K$ from such a compressed sketch.

2. How large should M be (in comparison to K) ?

3 Correlation-Aware Sensing (Piya)

Yoav : How is this problem different from the problem of finding the best placement for the sensors in order to maximize the accuracy of localization, ignoring communication bendwidth?

With the aim of obtaining a compressive sketch of the correlation matrix (also termed as compressive covariance sensing), we will optimize the design of sensor array (i.e. choice of $\mathbf{d}_m, 1 \leq m \leq M$) by understanding how the array geometry controls the algebraic structure of R_T. One of the main objectives will be to understand how much communication is needed (and between which subset of sensors) to achieve a certain level of accuracy. To illustrate this, we briefly discuss Co-PI Pal's recent work in structured sampler design (e.g., nested, coprime and generalized nested samplers) which utilize the idea of difference sets.

- **Difference set-inspired Designs:** I will review some results in the context of array processing and DOA estimation...(to be filled in).

- **Proposed Research:** Motivated by these results, our goal will be to develop a rigorous framework for further developing the key idea of correlation-aware sensing to a distributed scenario and make it applicable for imaging problems beyond point target localization.

  1. **Distributed Sensing:** The idea of difference set inspired sampler design can be actually generalized beyond that of antenna arrays, to acquire *compressive sketches* of the correlation between signals

[2] Reasonable to do so when the source signals are stationary and emit independent signals. This is the common practice in source localization using antenna arrays. We can also use more sophisticated regularized estimation of correlation.

acquired between pairs of sensors. In general, given N sensors, it is natural to think that one needs to compute the correlation between all $\binom{N}{2}$ time series (from all possible sensor-pairs) to construct the overall $N \times N$ correlation matrix R_T. However, using the idea of difference-set sampling, one can only compute cross-correlation values between a much smaller subset of size $\Theta(\sqrt{N})$ of *suitably selected sensor-pairs* and recreate the entire $N \times N$ correlation matrix R_T. In the context of distributed sensing, this automatically means that only these sensors need to communicate and exchange information.

Exploiting Distance-based Redundancies: The key idea behind achieving such reduction is to exploit the redundancies present in the correlation values that naturally result from the physical spatial signal model. A widely used example of such a redundancy is that the correlation $r_{m,n} = E\Big(y_m(t)y_n^*(t)\Big)$ between mth and nth sensors is of the following form

$$r_{m,n} \approx f(\mathbf{d}_m - \mathbf{d_n}) \tag{5}$$

In other words, the correlation is spatially only a function of the *inter-sensor distance*, and this is a direct consequence of the functional form of $\phi(.)$ **Piya :** Can give specific examples if needed. This is also referred to as spatial stationarity and it is (exactly or approximately) true for many applications as narrowband and wideband radar [3], super-resolution optical imaging [], mmWave wireless channels [] and so forth. Hence, depending on the inter-sensor distances, many of these $\binom{N}{2}$ correlation values are actually repeated/redundant. Based on this observation, we propose to use a new sketching technique developed by co-PI Pal, called **Generalized Nested Sampling (GNS) to reduce the amount of inter-sensor communication**. Suppose the sensors are located on a uniform grid. In one dimension, (5) implies that the correlation matrix R_T has Toeplitz structure and GNS provides an optimal way to select sensors to sketch such a matrix.

Definition 1 **Piya :** *Definition of GNS goes here..*

Hence, GNS dictates how to select a subset $\mathcal{S}_{\text{GNS}}$ of $M == \Theta(\sqrt{N})$ sensors out of N available sensors. Let $R_{\mathcal{S}} \in \mathbb{C}^{M \times M}$ be the correlation matrix computed by aggregating the signals from these sensors. Then GNS ensures that $R_{\mathcal{S}}$ is a *lossless* sketch of the high-dimensional correlation matrix R_T. **Piya :** To Add (i) Finite sample performance guarantees (ii) two and three-dimensional extension (iii) low-rank extension and (iv) Time-varying model.

2. *Beyond Point Target Localization: Using Priors and Sparsity* In many applications such as camera networks, the quantities of interest are

[3] In the latter case, this holds at individual frequency bands after splitting the wideband signal into narrow frequency bins using a filter bank

not the low-level measurements acquired at the CCD sensors, but the processed images I_t. In such cases, we need to obtain a compressive sketch of the image $A(I_t)$ via the sketching operator $A(.)$ using low dimensional representation (over unions of subspaces or manifolds). In addition to conventional sparsity and low-rank priors, one can also utilize (partial) knowledge of the prior distribution of the images $I_t \sim \mathcal{D}$. Utilizing these priors can lead to more effective compression for a given level of sparsity. [To be written..]

Piya : These tasks can be further integrated with the binary embedding based sketching ideas proposed by Rayan and Alex. **Rayan :** Agree!

4 Localization of weak sources (Peter)

Yoav : Can the description of SCM be folded into Piya's introduction?

The focus here is detecting weak sources within a sensor network without a fusion center. To observe weak sources, as much information as possible should be used. Thus, at first there is no attempt to reduce the information in the data by sketching or special sensor arrangements. The network could consist of sensors with know location, partially unknown or unknown positions.

The propagation path from a given source location would here represent multiple propagation paths in a non-uniform media. The frequency domain transfer function from a source location to N receivers $\mathbf{a}$. Assuming K uncorrelated sources of complex amplitude $\mathbf{s}$ at spatial location $\mathbf{x}_k$, the received signal $\mathbf{y} \in \mathcal{R}^N$ on N receivers is

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{n}, \tag{6}$$

where $\mathbf{A} = [\mathbf{a}_1, \ldots \mathbf{a}_K]$ and $\mathbf{n}$ is uncorrelated noise. The sources might be located in the near field and composed of many propagation paths. Examples of many propagation paths from a single source could be waves from 1) a source in a house propagating though the air and though the wall. 2) a cell phone signal with a direct path, a reflected path or refracted path. 3) a car radiating noise though the air and though the ground. Further, the sensors are not placed in a regular order, but where practical and maybe with unknown location. Thus the elements in $\mathbf{a}_k$ are unknown.

Yoav : What is the relationship between a_k and x_k?

To make observations of weak sources we observe L snapshots assuming stationarity $\mathbf{Y} = [\mathbf{y}_1 \ldots \mathbf{y}_L]$. We can here form the sample covariance matrix (SCM)

$$\mathbf{S} = \mathbf{Y}\mathbf{Y}^H / L \tag{7}$$

and form the the normalized SCM $\hat{\mathbf{S}}$ or coherence with elements

$$S_{ji} = \frac{S_{ji}}{\sqrt{S_{ii} S_{jj}}} \tag{8}$$

Yoav : I am confused about the definition of coherence, should it not be the maximum correlation when one signal to be shifted relative to the other?

Forming the ensemble mean over multiple snapshots give the cross spectral density matrix$\mathbf{C} \in \mathcal{R}^N \times N$

$$\mathbf{C} = \mathcal{E}[\mathbf{y}\mathbf{y}^H] = \mathbf{A}\mathbf{s}\mathbf{s}^H\mathbf{A}^H + \mathbf{N}, \tag{9}$$

The array signal processing literature is ample with processing of this type, especially with the structure of the $\mathbf{A}$ matrix partially known. In this work we will focus on pushing the computations to the sensor nodes and thus only observing part of SCM.

4.1 Graph signal processing approach without a fusion center

Yoav : I think this section can be combined with Piya's sections. Choosing which pairs should communicate is clearly related to their geometric layout.

Here the processing is done locally at each node. A graph signal processing approach was used in Ref [1] for a 5000 element seismic array by processing the whole normalized SCM at once, i.e., using a fusion center. When the coherence $\hat{\mathbf{S}}$ is above a certain threshold at element ij it is likely that a signal is observed and has propagated between nodes i and j, essentially forming an edge between nodes i and j in a graph. When a sufficient set of connected edges are detected in a region of the network a source is likely in that region. Part of the extracted SCM can then be used localize the source more precisely.

To extract very weak signals with a well estimated and robust SCM is needed. Thus we pass the full time series between local nodes i and j, not the whole array and develop robust signal processing methods[2]. This will represent a lot of communication demand and thus we will only pass signal between neighboring stations. Once a graph edge is formed it could either be communicated further to a wider set of nodes.

Robust signal processing methods[2] would entail making the processing insensitive to outliers. Qualitative robustness can be investigated via the influence function. A qualitatively robust estimator is characterized by an IF that is continuous and bounded. Continuity implies that small changes in the observed sample cause only small changes in the estimate. The boundedness implies that a small amount of contamination cannot lead to an unbounded error the estimate.

4.2 Tomography

Yoav : I believe you are talking here about tomography, or reconstructing the environment. Can you write a paragraph of introduction, what is the problem? What is the desired solution? Sparse modeling assumes that signals can be reconstructed using a few (sparse) vectors, called atoms, from a potentially large set of atoms, called a dictionary. Recent ocean acoustics works utilizing sparse modeling is beamforming[3], matched field processing [4], and geoacoustic inversion [5]. One challenge in sparse modeling is finding the best dictionary for sparsely representing specific signals. Such dictionaries can be composed of wavelets, or the discrete cosine transform (DCT). These predefined dictionaries perform well for many signals. However, using a form of unsupervised machine learning, called dictionary learning, optimal dictionaries can be learned directly from specific data[6]. It has been shown that learned dictionaries outperform generic dictionaries when sufficient signal examples are available. Machine learning, and specifically dictionary learning, have recently obtained compelling results in ocean acoustics citeBianco2017 and seismology[7].

Yoav : The simplest type of Dictionary learning is learning a codebook for Vector Quantization. Dictionaries are different than VQ if they combine (add) multiple vectors to represent a singe location in the signal. Is there a chance that VQ methods might work?

Yoav : I don't understand the following three paragraphs, can you give some technical details? Formulas? In current work, we have developed a machine learning-based travel time tomography method called locally sparse travel time tomography (LST)[8]. In LST, small scale local features contained in small rectangular groups of pixels, called patches, in an overall slowness (inverse speed) map are constrained using a sparse model. Further, the sparsifying dictionary is adapted to the specific slowness data using dictionary learning. Larger scale, or global features spanning the map, are constrained with least-squares regularization. Unlike conventional tomography, in which model features are forced to be exclusively smooth or discontinuous, the LST approach permits smooth and discontinuous local features via dictionary learning.

Whereas many machine learning techniques in geoscience[7], are reliant on large amounts of training data, LST requires none. In LST we adopt the adaptive dictionary learning paradigm from image denoising [9] and medical imaging[10], in which dictionaries are learned directly from patches of the corrupted image. In LST, slowness dictionaries are learned from patches of a least squares regularized inversion, and are then used to reconstruct a sparsity-constrained slowness image. Assuming sufficiently dense ray sampling, the dictionary is initially unknown and is learned in parallel with the inversion. LST obtains high resolution by assuming that small patches of discrete slowness maps are repetitions of few elemental patterns from a dictionary of patterns. These patterns, which are described by the atoms in the dictionaries, are extracted from the data by dictionary learning. The increase in performance for synthetic slownesses relative to competing methods, are demonstrated for ambient noise tomography[8].

Assuming that the travel paths between sensors has been estimated[11, 12] We here propose the future development of machine learning-based tomography methods in ocean acoustics. Such methods will help to more fully-exploit both existing hydrophone and environmental data, as well as very dense sampling from future arrays with many sensors. Such large scale, mobile, and deformable arrays, will use ambient noise processing [11], to obtain very dense and rich data sets. We propose for future work to: (1) further develop a dictionary learning-based travel time tomography [8], accounting for uncertainty in the measurements and physics; (2) formulate the dictionary learning-based approach as CNN via CSC; and (3) apply this CSC tomography framework to data assimilation, to obtain higher-resolution estimates of water column parameters over conventional methods. We further propose to develop (5) acoustic event detection methods that leverage recent advances in machine learning.

5 Fast binary embeddings... (Saab, Cloninger,...)

Yoav : Rayan and Alex, please link your sections to one or more of the specific tasks above. Feel free to add other tasks, but these should be specific real world applications, not just mathematical frameworks. **Rayan : Hi All: Please don't edit this section until after our meeting. Alex and Rayan are working on this. Comments welcome of course.**

Often, distributed systems have severe bandwidth and energy constraints that necessitate extremely coarse, e.g., binary, quantization of their measurements (e.g, [13, 14]). Simple binary representations of data can be also quite appealing in hardware implementations, particularly if they are computationally inexpensive and consequently promote speed in hardware devices ([15, 16]). On the other hand, a major concern in using very coarsely sketched data is possible loss of accuracy when performing various tasks of interest, ranging from signal reconstruction to clustering and statistical hypothesis testing, among others.

5.1 Background and Prior Work

Indeed a growing body of work, which co-PI Saab has contributed significantly to (e.g., [17, 18, 19, 20, 21]), has focused on signal reconstruction from coarsely quantized compressive measurements. One important theme that emerges from this line of work is that if one collects more coarsely quantized measurements than a critical minimal number, and uses sophisticated quantization schemes, then the extra measurements can be efficiently used in quantization-aware algorithms to rapidly drive the reconstruction error down. Co-PI Saab has recently extended this observation beyond signal reconstruction, to the context of (Euclidean) distance-preserving binary embeddings [22]. To be precise, we now briefly describe this contribution as it is pertinent to our ensuing discussion. In [22] $A : \mathbb{R}^n \to \mathbb{R}^m$ is a *random* Johnson-Lindenstrauss (i.e., distance preserving) map and $\mathcal{T} \subset \mathbb{R}^n$ is a set of finite or infinite cardinality. Moreover, $Q : \mathbb{R}^m \to \{\pm 1\}^m$ is a noise-shaping quantizer (e.g., a $\Sigma\Delta$ [**?**] or β [**?**] quantizer). In their most commonly used form, noise-shaping quantizers act sequentially on the measurements, say y_i. For example, the simplest noise-shaping quantization scheme, the so-called greedy 1st order $\Sigma\Delta$ scheme, is given, for $i = 1, ..., m$, by

$$q_i = \text{sign}(y_i + u_{i-1}) \quad (10)$$

$$u_i = u_{i-1} + y_i - q_i, \quad (11)$$

where the state-variable sequence u_i is initialized via, e.g., $u_0 = 0$. In matrix-vector notation, this yields the relationship

$$y = Du + q \tag{12}$$

between the measurement vector, the state variables, and the resulting quantization, with D being the $m \times m$ first-order difference matrix. Crucially both for the analysis and for practical implementation, this scheme is *stable*, that is,

$$\|y\|_\infty \leq c_1 \implies \|u\|_\infty \leq c_2, \tag{13}$$

for dimension independent constants c_1, c_2.

Yoav : How does one create an estimate of y_i from q_i?**Rayan :** see the blue text

For example, in the context of compressed sensing [], where the signals x are sparse and the measurements are of the form $y = Ax$, one may recover an estimate $\hat{x}$ of x from q by solving an optimization problem that encourages, say, sparsity, with the constraint that $\|D^{-1}(A\hat{x} - q)\|_\infty \leq c_2$. The constraint is critical; it ensures that the solution $\hat{x}$ respects the quantizer by satisfying (12) and (13). More generally, other *stable* noise-shaping quantizers act sequentially on the measurements y_i, and yield relations of the form

$$y = Hu + q,$$

where H is a lower-triangular matrix associated with the scheme. Quantizers of interest to us include stable rth-order (with $r \geq 1$) $\Sigma\Delta$ schemes [], with the relationship $y = D^r u + q$, and distributed-β encoding schemes [] where H is a block diagonal matrix constituted of identical lower-triangular blocks, say G, given by $G_{i,i} = 1, G_{i+1,i} = -\beta$, and $G_{i,j} = 0$ otherwise.

With these quantization schemes playing a prominent role, in [22] co-PI Saab constructed approximately isometric (i.e., distance preserving) embeddings between the metric space $(\mathcal{T}, \|\cdot\|_2)$ and the binary cube $\{-1, +1\}^m$ endowed with the pseudometric

$$d_V(\tilde{q}, q) := \|V(\tilde{q} - q)\|_2$$

where V is a carefully constructed matrix. For a matrix $A \in \mathbb{R}^{m\times n}$, and a noise-shaping quantizer Q, as above, the algorithm for computing these embeddings is simply given by

$$\begin{aligned} g : \mathcal{T} &\to \{\pm 1\}^m \\ x &\mapsto q = Q(Ax). \end{aligned}$$

In particular, when A is a fast Johnson-Lindenstrauss matrix (e.g., []), the constructed embeddings support fast computation, and despite their highly quantized non-linear nature, they perform as well as linear Johnson-Lindenstrauss methods up to an additive error that decays exponentially (for β quantizers) or polynomially (for $\Sigma\Delta$ quantizers) in m. Indeed, when $\mathcal{T}$ is finite [22] shows that with high probability and for prescribed distortion α

$$m \gtrsim \frac{\log(|\mathcal{T}|)\log^4 n}{\alpha^2} \quad \Longrightarrow \quad \left|d_{\widetilde{V}}(g(x), g(\tilde{x})) - \|x-\tilde{x}\|_2\right| \le \alpha\|x-\tilde{x}\|_2 + c\eta(m),$$
$$\text{where } \eta(m) \xrightarrow[m\to\infty]{} 0.$$

Above, $\eta(m)$ decays polynomially fast in m (when $\mathcal{Q}$ is a $\Sigma\Delta$ quantizer), or exponentially fast (when $\mathcal{Q}$ is a distributed noise shaping quantizer). Additionally, when $\mathcal{T}$ is arbitrary (with possibly infinite cardinality, e.g., a compact manifold) [22] show that with high probability and for prescribed distortion α

$$m \gtrsim \frac{\log^4 n}{\alpha^2}\cdot\frac{\omega(\mathcal{T})^2}{\mathcal{R}(\mathcal{T})^2} \quad \Longrightarrow \quad \left|d_{\widetilde{V}}(g(x), g(\tilde{x})) - \|x-\tilde{x}\|_2\right| \le \max(\sqrt{\alpha}, \alpha)\mathcal{R}(\mathcal{T}) + c\eta(m)$$

where $\eta(m)$ is as before and where $\mathcal{R}(\mathcal{T})$ and $\omega(\mathcal{T})$ denote the Euclidean radius of $\mathcal{T}$ and its Gaussian width (which roughly scales with the average radius of $\mathcal{T}$, so that intrinsically low-dimensional sets have a small Gaussian width). In short, with *very few measurements* compared to the ambient dimension of the signals, one can very efficiently (roughly at the cost of a fast Fourier transform) obtain low-dimensional binary sketches of the data. These sketches approximately preserve all pairwise distances in the original set and the distances in the embedded space can be computed efficiently. **Rayan :** may have to edit this depending on how the themes in the proposal evolve

While signal reconstruction is not a major focus of our work, the promising results obtained in that context, and particularly in the context of binary embeddings, lead us to believe that the above techniques can be generalized to other tasks. Our goal is to present a complete, theoretically rigorous, framework for performing various statistical, signal processing, and learning tasks from highly quantized data representations. We focus on the case of 1-bit representations as a theoretical extreme case, but emphasize that our methods should apply to more finely quantized data. We will propose, and analyze (a) algorithms for producing quantized sketches of data as well as (b) associated algorithms for performing the afore-mentioned tasks. Our strategy will be to develop these algorithms in tandem; that is, we will propose task-based quantization algorithms and quantization-aware algorithms for performing the tasks. We will strive for methods that support fast computation, and that lend themselves to distributed computing on, e.g., a sensor network.

5.2 Models

Suppose signals of interest are modeled as $Y \in \mathcal{X} \subset \mathbb{R}^d$, where $\mathcal{X}$ could be a finite set, i.e., a point cloud, or an infinite set (e.g., a compact manifold, or the set of all sparse vectors). **Yoav :** In the discussion so far, the time t has a special role. Is t one of the d coordinates or is it an extra coordinate. In other words, should we use $Y(t)$?

Alternatively, we may model signals as random vectors drawn according to some distribution, i.e., $Y \sim \mathcal{D}$ ofer R^d **Yoav :** assuming that the signal for a single time point is generated by and IID draw from a fixed distribution seem unnatural in the context of a sensor network, where the state of the environment $\theta(t)$ changes according to some dynamics. where $\mathcal{D}$ accounts for any structure in the signals. Further, assume that the measurement operator, accounting for all the measurements at all the sensors, is given by $\mathbf{\Phi} : \mathbb{R}^d \to \mathbb{R}^m$, **Yoav :** I am assuming that A is equal to the transfer function ϕ as described in the framework section and by Piya? **Rayan :** possibly, if the transfer function includes design elements of the sensor. In the case of distributed sensing systems, we can assume each sensor collects/computes a portion, $\mathbf{\Phi}_i Y \in \mathbb{R}^{m_i}$, of $\mathbf{\Phi} Y \in \mathbb{R}^m$ (so that $\sum_i m_i = m$) and the corresponding portion of $Q(Ax)$ where the quantization map $Q : \mathbb{R}^m \to \{\pm 1\}^m$ maps the measurements to bits. The sensors then must collaborate to estimate a quantity of interest, such as the state of the target, $\theta(x)$ using and estimation function $f(Q(Ax)) := f\left(Q(A_1 x), ..., Q(A_k x)\right) \approx g(x)$.

$$x \mapsto Q(Ax) = \begin{bmatrix} Q(A_1 x) \\ \vdots \\ Q(A_k x) \end{bmatrix} \mapsto f(Q(Ax)) \approx g(x) \tag{14}$$

5.3 Statistics on Binarized Sketching of Complex Measurements

The type of information that can be derived from $f(Q(Ax))$ depends on the choice of measurement operator A_i at each sensor, as well as the quantization scheme. For several of the tasks below, we will design f, A, and Q so that $\langle f(Q(Ax)), f(Q(Ay))\rangle$ approximates the Gaussian kernel $\langle g(x), g(y)\rangle = e^{-\|x-y\|_2^2/\sigma^2}$. As a first step, we will show that this can be achieved by appropriately quantizing $cos(2\pi\langle w, \cdot\rangle)$ for random choices of w [23]. To achieve this in practice, each physical sensor could collect one or more linear measurements $\langle \omega, \cdot\rangle$, and then the (e.g., cosine) non-linearity could be implemented as part of the quantizer. Alternatively, the sensor could measure the non-linear function directly prior to quantization.

Background and Motivation

Statistical distances, or accurately measuring distances between distributions, arise in a large number of applications. With sensors, one example where these distances would be important is in testing the hypothesis of whether two sensors have measured the same underlying distribution. An example of this would be for acoustic sensors, where each sensor collects the local power spectral density of a signal. Each sensor i now has a set of high dimensional data $X_i \mathbb{R}^d$, and the question is whether X_1 and X_2 are distributionally the same, up to a time shift. After constructing a kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+^d$ to define similarity between any two points, and ignoring communication constraints, a simple statistic to construct for a point $z \in X_1$ is a difference of kernel means over the two data sets $\frac{1}{|X_1|}\sum_{x\in X_1} K(z,x) - \frac{1}{|X_2|}\sum_{x\in X_2} K(z,x)$. If X_1 and X_2 came from the same distribution, then this statistic would be unbiased at $z \in X_i$, otherwise there would be a bias for particular z. Thus, computing the mean square error over all $z \in X_1 \cup X_2$ yields a statistic that would be close to 0 if $X_1, X_2 \sim p$, and would be biased if $X_1 \sim p$ and $X_2 \sim q$ for $p \neq q$. This is a well studied statistic known as kernel Maximum Mean Discrepancy []. In what follows, we will describe the mathematical framework and guarantees established, and propose methods for dealing with communication and computation constraints in this framework through randomized sketching and binary embeddings. We will also detail a larger set of sensor problems that can be addressed in this framework.

Generally, the approach described above is measuring the distance between the distributions' *mean embeddings* [24]. A mean embedding of a distribution $\mu_p : \mathbb{R}^d \to \mathcal{H}$ of a probability distribution p is computed as

$$\mu_p(z) := \mathbb{E}_{x\sim p}\big[K(z,x)\big].$$

Effectively, mean embedding is mapping the distribution p to a point in the Reproducing Kernel Hilbert Space $\mathcal{H}$ that is induced by the kerrnel K.

When K is a *universal* kernel (e.g. Gaussian, linear correlation) [25], then the mean embedding satisfies a key property that $\|\mu_p - \mu_q\|_{\mathcal{H}}$ is bi-Lipschitz with respect to $\|p-q\|_{L^\infty}$ for absolutely continuous distributions p and q. This effectively means that the mean embedding transform maintains the same information as working in $\mathbb{R}^d$, with the benefit that mean embeddings also satisfy nice statistical convergence properties. One key property is that, if we are only given n finite samples $X \sim p$ and $Y \sim q$ to compute the empirical mean embeddings $\widehat{\mu}_X$ and $\widehat{\mu}_Y$, and we compute the mean embedding at all $z \in X \cup Y$, then $\|\widehat{\mu}_X - \widehat{\mu}_Y\|_2 \to \|\mu_p - \mu_q\|$ at a rate $O\left(\frac{1}{\sqrt{n}}\right)$. A statistical interpretation of the mean embedding distance is that it computes a mean shift alternative test on the eigenfunctions of K rather than in the original space $\mathbb{R}^d$, and that two distributions having matching means in the eigenfunction space is equivalent to the distributions having all moments matching in $\mathbb{R}^d$. A large benefit of the mean embeddings is that this calculation can be done without explicitly computing the eigendecomposition of K.

However, statistics of this type suffer from a number of issues under computation and communication bottlenecks, as they requires storing and communicating all points in $X \cup Y$. In particular, computing $\hat{\mu}(z)$ at any one point z requires evaluating the kernel at all points in $X \cup Y$, which can be prohibitively expensive. This has motivated computational speed ups presented in [26] of undersampling $z \in S \subset X \cup Y$ under the condition that a kernel matrix K can be decomposed as $K \approx RR^T$ for R that can be efficiently accessed, as well as notions of kernel compression via randomized sketching [27]. However, even these approaches still require communication of a number of points between sensors, or passing double precision complex valued summary statistics.

Goals

Our overarching goal is to reduce the communication constraints by utilizing both sketching and binarization through use of $g(x) = f(Q(A(x)))$. The idea is that, rather than communicate and compute with $x \in \mathbb{R}^d$ where d may be large, one only needs to compute and communicate $Q(Ax) \in \{\pm 1\}^m$, with $m \ll d$, or $f(Q(Ax))$. In what follows, $\mathbb{E}_{x \in X} g(x)$ serves as a proxy for the empirical mean embedding of the data set X. For sensor networks, we can either compare between sensors by embedding each sensors' data through $g(X_i) = f(Q(A(X_i)))$ for a common A at each sensor, or have the sensors work collaboratively across a common set of points by building a concatenated sketching from data across all sensors $g(X) = \begin{bmatrix} f(Q(A_1(X))), & \ldots & f(Q(A_M(X))) \end{bmatrix}$. We aim to prove that this low complexity vector still converges to a type of mean embedding, and to provide a rigorous analysis of the statistical power, convergence rates, and minimal detectable separation criteria between the distributions. Below we highlight the benefit of this approach in a number of different sensor problems.

- Two Sample Testing: In the context of sensors, the two sample problem can be summarized as follows: each sensor collects a data set $X_i \sim p_i$, and the goal is to determine whether the X_i were distributed similarly. To address this, we will analyze the binarized statistic

  $$\left\| \frac{1}{n} \sum_{x \in X_1} g(x) - \frac{1}{m} \sum_{y \in X_2} g(y) \right\|,$$

  under appropriate norm, and seek to characterize the minimal conditions under which a deviation between p_1 and p_2 can be detected. The approach requires characterizing the types of deviations $p_1 - p_2$ that can be detected, namely those for which

  $$\left\| \int e^{-\|x-y\|^2/\sigma^2} (p_1(y) - p_2(y)) dy \right\| > \epsilon,$$

  as well as the rates at which these deviations can be detected. The communication benefit of such a statistic is that the sensors need only transmit the mean of $g(x)$, rather than all the individual points.

  Alex : Could expand on this, or reference other comments in proposal and a few papers about this for JL embeddings

  Rayan : let's refine this discussion a little – I have some ideas here

- Change Point Detection: A variant of the two sample testing problem is change point detection, in which the data is streaming $X(t)$ according to some underlying stochastic process. At some time t^*, the distribution of the stochastic process changes from one distribution to another, and the issue is how quickly after t^* this change can be detected. Unlike the two sample context in which we were testing whether the sensors detected the same distribution, here we can use the sensors collaboratively by constructing the concatenated sketching matrix $g(X) =$

$\left[f(Q(A_1(X))), \quad ... \quad f(Q(A_M(X)))\right]$. There exists a mean embedding approach to change point analysis [?] which uses the kernel Fisher discriminant ratio and mean embedding to measure the homogeneity between time segments of the process. However, this once again requires storage of all points over the length of the detection window and communication of those points across sensors for both computing the window mean and variance. It also suffers from an inability to begin computing the change point statistic until all points in the window have been collected. We aim to introduce the binarized sketching framework to produce an efficient computation to the kernel mean as in two sample testing, as well as a fast construction of a low rank approximation to the kernel covariance matrix. We will derive it's new limiting distribution under the null model of no change, as well as the consistency under the alternative distribution when a change does occur.

- Multiple Sensor Common Factor Identification: A common issue is aggregating multiple sensors to identify and magnify the signal detected by both sensors. Under a linear model, algorithms such as Canonical Correlation Analysis [] act on multiple streams of simultaneously collected data $X_i(t)$ to filter noise and recover highly correlated linear projections from two the data sets. Recently, a kernel CCA technique called alternating diffusion [?] has be used to identify common nonlinear effects by building a kernel K_i from each sensing modality and analyzing the product kernel $(K_1K_2)^t$. Concretely, assume that there exists a hidden manifold $\mathcal{M}$ and two nuisance manifolds $\mathcal{N}_1$ and $\mathcal{N}_2$, and samples $s((x_i, y_i^{(1)}, y_i^{(2)}))$ for $(x_i, y_i^{(1)}, y_i^{(2)}) \in \mathcal{M} \times \mathcal{N}_1 \times \mathcal{N}_2$. Due to sensor location or modality, sensor 1 collects data points $S_1 = s((x_i, y_i^{(1)}, \xi))$ and sensor 2 collecting data points $S_2 = s((x_i, \zeta, y_i^{(2)}))$. Then roughly speaking, for some assumptions on s and for $K_1 : S_1 \times S_1 \to \mathbb{R}$ and $K_2 : S_2 \times S_2 \to \mathbb{R}$, Talmon and Wu [?] proved $(K_1K_2)^t$ is the diffusion kernel on $\mathcal{M}$ only. This is a very beneficial feature, as it means that one can compute kernel statistics for distributions defined on $\mathcal{M}$ only, independent of the nuisance features that may differ between sensors due only to modality or location.

  However, this requires communication of all n points across sensors to compute the kernel product. We propose to analyze such approaches under minimal communication constraints by utilizing the low rank binarized decomposition $K_i \approx g(S_i)g(S_i)^*$. As the key feature of alternating diffusion is computing the inner product matrix $g(S_1)^*g(S_2)$, which has dimension $d < n$ and is also low rank. This implies it is possible to sketch $g(S_1)$ (resp. $g(S_2)$) with a small set of vectors v_j (resp. w_k) and only communicate vectors $g(S_1)^*v_j$ (resp. $w_k^*g(S_2)$) such that $\mathbb{E}_{j,k}[g(S_1)^*v_jw_k^*g(S_2)] \approx g(S_1)^*g(S_2)$. The amplification of the common factors observed by both sensors can serve to boost the power of the two sample and change point statistics on the shared observable manifold $\mathcal{M}$, as well as other compressed statistics that can be computed [?].

- **Rayan :** Put in classification
  - Idea 1: Training reduced to computing averages μ_j, over the data of say $Q(\cos(Ax+b))$ for each class $x \sim \mathcal{X}_j$. Classification reduced to computing $\mu_y = Q(\cos(Ay+b))$ and finding the closest μ_j to μ_y. In a sense, this is like 2-sample testing, but with a point mass at y. Advantages: Each sensor only keeps the averages relevant to its own portion of $Q(\cos(Ax))$ for x in training set, and later for y to be classified. Each sensor can compute its own piece of the inner product $\langle \mu_j, \mu_y \rangle$, you are ultimately averaging bits, so memory needed at each sensor grows like $m_i \times \log(P)$ where P is the total number of points and m_i is the size of the sketch at each sensor... The idea is that $m_i \ll d$ and $\log(P) \ll P$ so you save a ton on storage/communication.
  - Idea 2: Exploit the hierarchical/distributed nature of the bits produced by binary embeddings. Requires more thought...

  and clustering: Analogous to the above...

6 Tell me something new (Yoav)

We propose a general framework for designing and analyzing sensor networks. Suppose we have two sensors two sensors whose task is to track state of target whose trajectory is defined by $\theta(t)$. Suppose the sensors use a dynamic model to predict future states. For the sake of concreteness, suppose a Newtonian model of motion is used which predicts constant velocity of the target unless a disruptive force acts on it.

Let $\hat{\theta}_1(t), \hat{\theta}_2(t)$ be the estimates of the target location for each sensor. *In addition, each sensor maintains an estimate of the estimate of the other sensor.* $\hat{\theta}_{1,2}(t), \hat{\theta}_{2,1}$. Each sensor updates it's estimate of the velocity of the location of the target according to the signal it measures, but it does not update it's estimate of the other's estimate. If the two estimate are close to each other $\hat{\theta}_i(t) \approx \hat{\theta}_{i,j}(t)$ then sensor i sends no information to sensor j. On the other hand, if $\hat{\theta}_i(t)$ is far from $\hat{\theta}_{i,j}(t)$, then $\hat{\theta}_i$ is transmitted from sensor i to sensor j. Thus if the target is moving in constant speed, uninterrupted, there is not communication between the sensors.

The basic idea here is that a sensor sends out information only if that information cannot be predicted by the receiver. Similar ideas have been used in arithmetic coding and **Yoav :** I think in $\Sigma\Delta$ encoding.

Recently, PI Freund [**?**] proposed an asynchronous computation model called "Tell Me Something New" in which each agent broadcasts a message only when the estimate it computes differs from the existing estimates in a statistically significant way.

One important application of sensor networks is to monitor activity and identify anomalies. Examples include: building security systems, factory floors, highway monitoring, health monitoring for the sick or elderly and many others.

On its face, this might seem like an under-constrained impossible problem. However, note that for all of the environments listed above there is a highly repetitive pattern from day to day and from week to week. Add to that the sensors are stationary, and one would expect that most sensors observe highly regular and highly predictable patterns.

The approach we propose in this case is that each sensor creates a model of the characteristics of the signals that it observes during normal operations. It alerts neighboring sensors if it observes something that is abnormal, i.e. a signal that has very low probability according to the model. When several sensors send an alert with a short time window, and when the alerts are consistent with each other, a global alert is sent to the human operators.

6.1 Spatio-temporal fields

6.2 Use of a sensor network

- Track an EM source in a home based on an ad hoc network.
- Estimating foot traffic (as in match point)
- Localize a speaker in a room
- Localizing a weak source in a floating network where the sensor locations are unknown. In e.g., The ocean
- Structural monitoring using
- Localizing an acoustic source anywhere in a home using a sensor few sensors in a room. A human can do it.

7 results from previous grants

References

[1] Nima Riahi and Peter Gerstoft. Using graph clustering to locate sources within a dense sensor array. *Signal Processing*, 132:110–120, 2017.

[2] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma. *Robust Statistics for Signal Processing*. Cambridge University Press, 2018.

[3] A. Xenaki, P. Gerstoft, and K. Mosegaard. Compressive beamforming. *J. Acoust. Soc. Am.*, **136**(1):260–271, 2014.

[4] K. L Gemba, S. Nannuru, P. Gerstoft, and W.S. Hodgkiss. Multi-frequency sparse Bayesian learning for robust matched field processing. *J. Acoust. Soc. Am.*, 141(5):3411–3420, 2017.

[5] Peter Gerstoft, Christoph F Mecklenbräuker, Woojae Seong, and Michael Bianco. Introduction to compressive sensing in acoustics, 2018.

[6] Stéphane Mallat. *A wavelet tour of signal processing.* Elsevier, 1999.

[7] Qingkai Kong, Daniel T Trugman, Zachary E Ross, Michael J Bianco, Brendan J Meade, and Peter Gerstoft. Machine learning in seismology: Turning data into insights. *Seismological Research Letters*, 90(1):3–14, 2018.

[8] Michael J Bianco and Peter Gerstoft. Travel time tomography with adaptive dictionaries. *IEEE Transactions on Computational Imaging*, 4(4):499–511, 2018.

[9] Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing.* Springer Science & Business Media, 2010.

[10] Saiprasad Ravishankar and Yoram Bresler. Mr image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE transactions on medical imaging*, 30(5):1028–1041, 2011.

[11] Karim G Sabra, Peter Gerstoft, Philippe Roux, WA Kuperman, and Michael C Fehler. Surface wave tomography from microseisms in southern california. *Geophysical Research Letters*, 32(14), 2005.

[12] Peter Gerstoft, Karim G Sabra, Philippe Roux, WA Kuperman, and Michael C Fehler. Green's functions extraction and surface-wave tomography from microseisms in southern california. *Geophysics*, 71(4):SI23–SI31, 2006.

[13] Jun Fang, Yanning Shen, Hongbin Li, and Zhi Ren. Sparse signal recovery from one-bit quantized data: An iterative reweighted algorithm. *Signal Processing*, 102:201–206, 2014.

[14] Petros T Boufounos and Richard G Baraniuk. 1-bit compressive sensing. In *2008 42nd Annual Conference on Information Sciences and Systems*, pages 16–21. IEEE, 2008.

[15] Laurent Jacques, Jason N Laska, Petros T Boufounos, and Richard G Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Transactions on Information Theory*, 59(4):2082–2102, 2013.

[16] Bin Le, Thomas W Rondeau, Jeffrey H Reed, and Charles W Bostian. Analog-to-digital converters. *IEEE Signal Processing Magazine*, 22(6):69–77, 2005.

[17] R. Saab, R. Wang, and Ö. Yılmaz. From compressed sensing to compressed bit-streams: Practical encoders, tractable decoders. *IEEE Transactions on Information Theory*, 64(9):6098–6114, Sep. 2018.

[18] Karin Knudson, Rayan Saab, and Rachel Ward. One-bit compressive sensing with norm estimation. *IEEE Transactions on Information Theory*, 62(5):2748–2758, 2016.

[19] Rayan Saab, Rongrong Wang, and Özgür Yılmaz. Quantization of compressive samples with stable and robust recovery. *Applied and Computational Harmonic Analysis*, 44(1):123–143, 2018.

[20] Rayan Saab and Eric Lybrand. Quantization for low-rank matrix recovery. *Information and Inference: A Journal of the IMA*, 8(1):161–180, 05 2018.

[21] Mark Iwen, Eric Lybrand, Aaron Nelson, and Rayan Saab. New algorithms and improved guarantees for one-bit compressed sensing on manifolds. *arXiv preprint arXiv:1902.03726*, 2019.

[22] Thang Huynh and Rayan Saab. Fast binary embeddings, and quantized compressed sensing with structured matrices. *arXiv preprint arXiv:1801.08639*, 2018.

[23] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[24] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

[25] Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667, 2006.

[26] Xiuyuan Cheng, Alexander Cloninger, and Ronald R Coifman. Two-sample statistics based on anisotropic kernels. *arXiv preprint arXiv:1709.05006*, 2017.

[27] Rémi Gribonval, Gilles Blanchard, Nicolas Keriven, and Yann Traonmilin. Compressive statistical learning with random feature moments. *arXiv preprint arXiv:1706.07180*, 2017.

[28] M. Bianco and P. Gerstoft. Dictionary learning of sound speed profiles. *J. Acoust. Soc. Am.*, 141(3):1749–1758, 2017.

[29] Yotam Abramson and Yoav Freund. Active learning for visual object recognition. Technical report, UCSD, 2004.

[30] Evan Ettinger and Yoav Freund. Camera pointing with coordinate-free localization and tracking. In Pawel Strumillo, editor, *Advances in Sound Localization*. Intech, 2011. http://www.intechopen.com/articles/show/title/camera-pointing-with-coordinate-free-localization-and-tracking.

[31] Sanjoy Dasgupta and Yoav Freund. Random projection trees for vector quantization. *IEEE Transactions on Information Theory*, 55(7), July 2009.