SENSING CAPACITY FOR DISCRETE SENSOR NETWORK APPLICATIONS

Yaron Rachlin, Rohit Negi, and Pradeep Khosla

ECE, Carnegie Mellon University, {rachlin, negi, pkk}@ece.cmu.edu

ABSTRACT

We bound the number of sensors required to achieve a desired level of sensing accuracy in a discrete sensor network application (e.g. distributed detection). We model the state of nature being sensed as a discrete vector, and the sensor network as an encoder. Our model assumes that each sensor observes only a subset of the state of nature, that sensor observations are localized, and that sensor network output across different states of nature is neither identical nor independently distributed. Using a random coding argument we prove a lower bound on the 'sensing capacity' of a sensor network, which characterizes the ability of a sensor network to distinguish among all states of nature. We compute this lower bound for sensors of varying range, noise models, and sensing functions. We compare this lower bound to the empirical performance of a belief propagation based sensor network decoder for a simple seismic sensor network scenario. The key contribution of this paper is to introduce the idea of a sharp cut-off function in the number of required sensors, to the sensor network community.

1. INTRODUCTION

How many sensors are required to sense an environment to within a desired accuracy? In this paper, we explore this question in the context of discrete sensor network applications such as distributed detection and classification. The number of sensors required to achieve a desired performance level depends on a large number of characteristics such as the noise, range, and sensing function of the constituent sensors, as well as resource constraints such as the power, computation, and communications available at each sensing node. Resource constraints such as communications and power are important to consider in the design of sensor networks due to the limitations they impose on, among other things, network lifetime and sampling rate. However, even if these resource constraints were eliminated, many basic questions about the theoretical design limitations of sensor networks are not yet adequately addressed. The sensing capabilities of the sensors and the required accuracy of the sensing task imposes sharp limitations on the number of sensors required to achieve a desired performance level. In this paper we seek to elucidate this purely sensing based limitation by demonstrating a lower bound on the minimum number of sensors required to achieve a desired sensing performance, given the sensing capabilities of the sensors.

In our discrete sensor network application, we model the state of nature as a discrete vector and the sensor network as a "channel encoder." For ease of discussion, we assume that the discrete state of nature represents a spatial configuration of targets. Our model assumes that each sensor observes only a subset of target positions, and that sensor observations are localized (i.e. a sensor observes adjacent target positions). Viewing the sensor network as a channel encoder allows us to use ideas from Shannon coding theory. However, as we will show, the "codebook" obtained has codewords which are neither independent nor identical, thus requiring a novel analysis and a novel concept of 'sensing capacity' $C(D)$. $C(D)$ characterizes the ability of the sensor network to distinguish among all spatial target configurations to within a given distortion D. D is the maximum tolerable fraction of spatial positions which may be erroneously sensed. For a given D, $C(D)$ represents the maximum ratio of the total number of target positions under observation to the number of sensors, such that below this ratio, there exist sensor networks whose maximal probability of error goes to zero as the number of possible target positions and sensors goes to infinity. In previous work [1], we introduced the concept of a sensing capacity and provided a lower bound on this quantity for a rather restricted family of sensor networks. This previous model assumed that sensors can sense all targets with uniform probability, and that the sensors output a noise corrupted sum of the targets which they observe. Such a model is not well suited to many applications of interest, such as seismic sensor networks and networks of cameras. Therefore, in this paper we relax both of these assumptions, and demonstrate a lower bound for sensing capacity for a sensor network model with localized sensor observations and arbitrary sensing functions.

Research on the theoretical performance limits of sensor networks typically considers how system performance scales with the number of sensors. The first set of results can be broadly categorized as the constraints that resources such as communication, computation, and power impose on the sensor network when the number of sensors increases. [2] extends the results in [3] to account for the different traffic models that arise in a sensor network. [4] studies network transport capacity for the case of regular sensor networks. [5] studies the impact of computational constraints on the communication efficiency of sensor networks. Another set of results considers the effect of the number of sensors on accomplishing a sensing task, given resource constraints. [6] studies the effect of transport capacity on approximating a set of continuous random processes. [7] considers the estimation of parameters of a set of underlying random processes. [8] considers a decentralized binary decision problem with noisy communication links to obtain error exponents.

In contrast to the aforementioned results, we explore a notion of a 'sensing capacity' inherent purely to the sensing task of distinguishing among discrete states of nature to within a desired distortion. Section 2 introduces and motivates our sensor network model. Section 3 states a lower bound on sensing capacity for the model. Section 4 extends the result to heterogeneous sensors and non-binary target vectors. Illustrative calculations of the sensing capacity are presented in Section 5. We apply our model to a seismic sensor network scenario, and compare empirical detection performance to our bound in Section 6. Section 7 concludes the paper.

2. SENSOR NETWORK MODEL

We denote random variables by upper-case letters and instantiations or constants by lower-case letters. Bold-font denotes vectors, and bold-font upper-case letters denote random vectors. $\log(\cdot)$ has base-2.

We considered discrete sensor applications with spatially localized sensing in formulating our model. Examples include a target counting protocol using a seismic sensor network implemented by [9]. A multi-camera network was designed to count the number of people in a crowd [10] and to localize moving objects in a grid [11]. [12] performs distributed vehicle classification using acoustic and seismic sensor data. [13] formulated distributed robot exploration as a discrete sensing task, using belief propagation to fuse robot observations. In all these applications, each sensor views a contiguous region of space (i.e. spatially localized observations).

[14] proposed an abstract sensor network model for detecting discrete target locations. This work introduces the idea of viewing sensor networks as encoders, and uses algebraic coding theory to design highly structured sensor networks, but no notion of capacity is discussed. There exists a large body of work in distributed detection [15], but we are not aware of the existence of any 'sensing capacity' results. [16] studies algorithms for distributed classification, but does not explore a notion of capacity.

Our sensor network model is motivated by the following specific discrete sensing scenarios. Before we present the details of our model, we review these scenarios and discuss how to model them as a discrete sensing tasks. In a seismic sensor network, sensors detect the intensity of target induced vibrations. We can model the environment as a grid world where each block represents the presence or absence of a target. A sensor is affected by targets in a localized region, whose extent is defined by random variations in soil composition and the limits of the sensor's range. The intensity of vibration is dependent on the target's distance from the sensor, and therefore the sensor observes the weighted sum of target vibrations. In a camera-based motion mapping scenario, the area under observation can be viewed as a grid. Each grid block contains a one or a zero, corresponding to motion or lack of motion in the grid block. If we assume that the cameras are calibrated, each camera observes a known subset of grid squares in its field of view. Due to the geometry of the scenario the observations are localized, and the sensing function of each camera produces an estimated motion map in the subsection of the grid under observation. Such a system combines multiple localized overlapping camera observations to obtain a single motion map of the environment. One can model distributed robotic mine detection [17] as a discrete classification task where the environment is modeled as a non-binary grid such that each block contains either nothing, a landmine, or metallic clutter. Each robot samples a localized subset of the grid at a time, and produces a noisy estimate of the grid contents under observation. The robots can cooperatively map the contents of the grid.

The model we present attempts to abstractly characterize various discrete sensor network applications with localized sensing, as motivated by the above scenarios. Figure 1 shows an example of our sensor network model. There are k discrete spatial positions that need to be sensed. Each position may represent an actual region in space. In our initial exposition, each discrete position may contain no target or one target, though extensions to non-binary targets is straightforward as shown in Section 4. A k-bit 'target vector' $\boldsymbol{v}$ represents the target configuration in these k positions. Our figure contains $\boldsymbol{v} = (0,0,1,0,1,1,0)$, indicating 3 targets among the 7 positions. The possible target vectors are denoted $\boldsymbol{v_i}$, $i \in \{1,\ldots,2^k\}$. We say that 'a certain $\boldsymbol{v}$ has occurred' if that vector represents the true target configuration in the spatial positions. The sensor network has n identical sensors. Sensor ℓ is connected to (i.e., senses) exactly c contiguous positions out of the k spatial positions. In contrast, our original model [1] did not account for localized sensor observations since each sensor could sense any c (not necessarily contiguous) spatial positions. Each sensor senses a value $x \in \mathcal{X}$ that is an arbitrary function of the targets bits to which it is connected, $x = \Psi(v_t,\ldots,v_{t+c-1})$. For example, a linear sensing function, such as a seismic sensor, would sense the weighted sum of the target bits which the sensor observes, $x = \sum_{u=0}^{c-1} w_u v_{t+u}$. In our previous model [1], the sensing function was restricted to be an *un-weighted sum* of the observed spatial positions. Our figure illustrates this sensing function for a specific sensor network, set of weights, and target vector. Thus, the 'ideal output vector' of the sensor network $\boldsymbol{x}$ depends on the sensor connections, sensing function, and on the target vector $\boldsymbol{v}$ that occurs. However, we assume that each sensor output $y \in \mathcal{Y}$ is corrupted by noise, so that the conditional p.m.f. $P_{Y|X}(y|x)$ determines the observed output. Since the sensors are identical, $P_{Y|X}$ is the same for all the sensors (we extend our result to heterogeneous sensors in Section 4). Further, we assume that the noise is independent in the sensors, so that the 'sensor output vector' $\boldsymbol{y}$ relates to the ideal output $\boldsymbol{x}$ as $P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x}) = \prod_{\ell=1}^{n} P_{Y|X}(y_\ell|x_\ell)$. Observing the output $\boldsymbol{y}$, a decoder (described in detail below) must determine which of the 2^k target vectors $\boldsymbol{v_i}$ have actually occurred.

We define the sensor network $S(k, n, c)$ as a bipartite graph, as shown in Figure 1. The figure shows the connections between the sensors and the k spatial positions, for sensors whose sensing function outputs a weighted sum of the observed targets. We assume a simple model for randomly constructing such sensor networks, where each sensor randomly chooses c contiguous spatial positions with equal probability among the set of possible contiguous blocks of length c. This would occur, for example, if sensors were randomly dropped on a field, or robots moved randomly over a region. This model represents an improvement over our previous model for the discrete sensor network applications described above because it accounts for the fact that sensor observations are localized, and allows for arbitrary sensing functions.

3. SENSOR NETWORK CAPACITY THEOREM

For a sensor network, randomly generated as explained earlier, the ideal output $\boldsymbol{x}$ is a function of the sensor network instantiation $s(k, n, c)$, the sensing function Ψ, and the occurring target vector $\boldsymbol{v}$. Denote $\boldsymbol{X_i}$ as the random vector which occurs when $\boldsymbol{v_i}$ is the target vector (where $\boldsymbol{X_i}$ is random because of the random generation of the sensor network $S(k, n, c)$). Since each sensor independently forms connections to a subset of targets, $P_{\boldsymbol{X_i}}(\boldsymbol{x_i}) = \prod_{\ell=1}^{n} P_{X_i}(x_{i\ell})$. However, it is important to note that the random vectors $\boldsymbol{X_i}$ and $\boldsymbol{X_j}$, associated with a *pair of target vectors* $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ respectively, are *not independent*, since the sensor network configuration produces a dependency between them. i.e. similar target vectors are likely to produce a similar sensor network output. Thus, the 'codewords' $\{\boldsymbol{X_i}, i = 1, 2, \ldots, 2^k\}$ of the sensor network (one corresponding to each $\boldsymbol{v_i}$) are non-identical and dependent on each other, unlike channel codes in classical information theory.

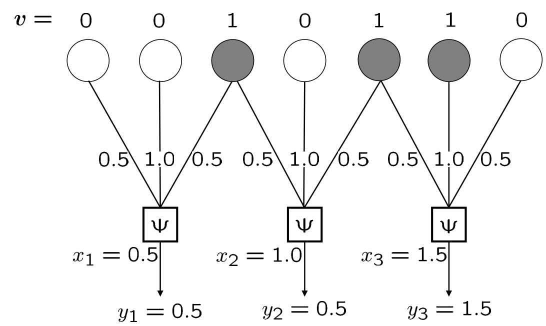

Fig. 1. Sensor network model with $k = 7, n = 3, c = 3$, spatially dependent connections, and a sensing function corresponding to the weighted sum of the observed targets.

Given the noise corrupted output $\boldsymbol{y}$ of the sensor network, we estimate the target vector $\boldsymbol{v}$ which generated this noisy output by using a decoder $g(\boldsymbol{y})$. We allow the decoder a distortion of $D \in [0, 1]$. i.e., if $d_{\text{H}}(\boldsymbol{v_i}, \boldsymbol{v_j})$ is the Hamming distance between two target vectors and if we define the tolerable distortion region of $\boldsymbol{v_i}$ as $\mathcal{D}_i = \{j : \frac{1}{k} d_{\text{H}}(\boldsymbol{v_i}, \boldsymbol{v_j}) < D\}$, then given that $\boldsymbol{v_i}$ occurred, the probability of error is

The statement of the main result requires an explanation of *c-order types* and *c-order joint types* [19]. We define the c-order type of a sequence of binary symbols as a 2^c dimensional vector, $\boldsymbol{\gamma}$, where each entry in the vector corresponds to the frequency of occurrence of one of the possible subsequences of length c. The total number of subsequences of length c that can occur in a sequence of length k is $k-c+1$. For example, for a binary target vector and $c=2$, $\boldsymbol{\gamma}=(\gamma_{00},\gamma_{01},\gamma_{10},\gamma_{11})$.

We denote the set of all c-order types over the alphabet $\{0,1\}^c$ for target vectors of length k as $\mathcal{T}_k(\{0,1\}^c)$. Since each sensor independently chooses a block of c contiguous spatial positions, the distribution of its ideal output X_i (which is sensed when the i^{th} target vector $\boldsymbol{v_i}$ occurs) depends only on the c-order type $\boldsymbol{\gamma}$ of $\boldsymbol{v_i}$. i.e., for a sensing function Ψ and a target vector $\boldsymbol{v_i}$ of type $\boldsymbol{\gamma}$,

$$P_{X_i}(X_i=x)=\sum_{\substack{\{a_1\ldots a_c\}\in\{0,1\}^c\\ \Psi(a_1\ldots a_c)=x}}\gamma_{a_1\ldots a_c}\ \doteq\ P^{\boldsymbol{\gamma}}(x)$$

Thus, $P_{\boldsymbol{X_i}}(\boldsymbol{x_i})=P^{\boldsymbol{\gamma},n}(\boldsymbol{x_i})=\prod_{\ell=1}^{n}P^{\boldsymbol{\gamma}}(x_{i\ell})$ for all $\boldsymbol{v_i}$ of type $\boldsymbol{\gamma}$.

Next, we note that the conditional probability $P_{\boldsymbol{X_j}|\boldsymbol{X_i}}$ depends on the *c-order joint type* $\boldsymbol{\lambda}$ of the i^{th} and j^{th} target vectors $\boldsymbol{v_i},\boldsymbol{v_j}$. $\boldsymbol{\lambda}$ is the vector of $\lambda_{(a_1\ldots a_c)(b_1\ldots b_c)}$, the fraction of positions in $\boldsymbol{v_i},\boldsymbol{v_j}$ where $\boldsymbol{v_i}$ has a bit subsequence $a_1\ldots a_c$ while $\boldsymbol{v_j}$ has a bit subsequence $b_1\ldots b_c$. For example, when $c=2$, $\boldsymbol{\lambda}=(\lambda_{(00)(00)},\ldots,\lambda_{(11)(11)})$. We denote the set of all c-order joint types over the alphabet $\{0,1\}^{2c}$ for target vectors of length k as $\mathcal{L}_k(\{0,1\}^{2c})$. We denote $\lambda_{(a_1\cdots)(b_1\cdots)}=\sum_{\{a_2\ldots a_c\}\in\{0,1\}^{c-1}}\sum_{\{b_2\ldots b_c\}\in\{0,1\}^{c-1}}\lambda_{(a_1\ldots a_c)(b_1\ldots b_c)}$. Since each sensor depends only on the c contiguous targets bits which it senses, $P_{\boldsymbol{X_j}|\boldsymbol{X_i}}$ depends only on the joint type $\boldsymbol{\lambda}$. i.e. for target vectors $\boldsymbol{v_i}$,$\boldsymbol{v_j}$ of c-order joint type $\boldsymbol{\lambda}$,

$$\begin{aligned}P_{X_iX_j}(X_i=x_i,X_j=x_j)&=\sum_{\substack{\{a_1\ldots a_c\}\in\{0,1\}^c\\ \{b_1\ldots b_c\}\in\{0,1\}^c\\ \Psi(a_1\ldots a_c)=x_i\\ \Psi(b_1\ldots b_c)=x_j}}\lambda_{(a_1\ldots a_c)(b_1\ldots b_c)}\\ &\doteq P^{\boldsymbol{\lambda}}(x_i,x_j)=P^{\boldsymbol{\lambda}}(x_j|x_i)P^{\boldsymbol{\gamma}}(x_i)\end{aligned}$$

Thus, $P_{\boldsymbol{X_j}|\boldsymbol{X_i}}(\boldsymbol{x_j}|\boldsymbol{x_i})=P^{\boldsymbol{\lambda},n}(\boldsymbol{x_j}|\boldsymbol{x_i})=\prod_{\ell=1}^{n}P^{\boldsymbol{\lambda}}(x_{j\ell}|x_{i\ell})$ for all i,j of the same joint type $\boldsymbol{\lambda}$.

For example, for binary target vectors and $c=2$, vectors $00000000, 01000111, 11111111$ have $\boldsymbol{\gamma}=(1,0,0,0)$, $(2/7,2/7,1/7,2/7)$, $(0,0,0,1)$ respectively. Table 1 contains the 2-order joint type of two target vectors. Consider a sensor network where each sensor is randomly connected to $c=2$ contiguous spatial positions. We assume that Ψ outputs the sum of the target bits which the sensor observes. Thus, each sensor has an ideal output alphabet $\mathcal{X}=\{0,1,2\}$. For target vectors of type $\boldsymbol{\gamma}$, $P(X_i=0)=\gamma_{00}$, $P(X_i=1)=\gamma_{01}+\gamma_{10}$, $P(X_i=2)=\gamma_{11}$ respectively. Given two target vectors $\boldsymbol{v_i},\boldsymbol{v_j}$ of joint type $\boldsymbol{\lambda}$, a sensor will output '0' for both target vectors only if both of its connections see a '0' bit in both target vectors. This happens with probability $\lambda_{(00)(00)}$. Table 2 lists the joint p.m.f. $P_{X_iX_j}(x_i,x_j)=P^{\boldsymbol{\gamma}}(x_i)P^{\boldsymbol{\lambda}}(x_j|x_i)$ for all output pairs x_i,x_j corresponding to joint type $\boldsymbol{\lambda}$. The table shows that X_i,X_j are not independent, in general.

$\lambda_{(ab)(cd)}$	$cd=00$	$cd=01$	$cd=10$	$cd=11$
$ab=00$	0	0	0	2/7
$ab=01$	1/7	1/7	0	0
$ab=10$	1/7	1/7	0	0
$ab=11$	0	0	1/7	0

Table 1. $\boldsymbol{\lambda}$ with $c=2$ for $\boldsymbol{v_i}=01101000$ and $\boldsymbol{v_j}=01000111$.

We specify two probability distributions which we will utilize in the main theorem. The first is the joint distribution of the ideal output $\boldsymbol{x_i}$ when $\boldsymbol{v_i}$ occurs and the noise corrupted version $\boldsymbol{y}$ of $\boldsymbol{x_i}$. i.e., $P_{\boldsymbol{X_i}\boldsymbol{Y}}(\boldsymbol{x_i},\boldsymbol{y})=\prod_{\ell=1}^{n}P_{X_iY}(x_{i\ell},y_\ell)=\prod_{\ell=1}^{n}P_{X_i}(x_{i\ell})P_{Y|X}(y_\ell|x_{i\ell})$. The second distribution is the joint distribution of the ideal output $\boldsymbol{x_i}$ corresponding to $\boldsymbol{v_i}$ and the noise corrupted output $\boldsymbol{y}$ generated by the occurrence of a *different* target vector $\boldsymbol{v_j}$. We can write this joint distribution as $Q^{(j)}_{\boldsymbol{X_i}\boldsymbol{Y}}(\boldsymbol{x_i},\boldsymbol{y})=\prod_{\ell=1}^{n}Q^{(j)}_{X_iY}(x_{i\ell},y_\ell)=\prod_{\ell=1}^{n}\sum_{a\in\mathcal{X}}P_{X_i}(x_{i\ell})P_{X_j|X_i}(x_j=a|x_{i\ell})P_{Y|X}(y_\ell|x_j=a)$. Note that $\boldsymbol{X_i},\boldsymbol{Y}$ are dependent here, although $\boldsymbol{Y}$ was produced by $\boldsymbol{X_j}$ because of the dependence of $\boldsymbol{X_i},\boldsymbol{X_j}$. This is unlike Shannon codes, where the codewords are independent.

Since each sensor in the sensor network depends only on the c contiguous targets which it observes, $P_{\boldsymbol{X_i}\boldsymbol{Y}}(\boldsymbol{x_i},\boldsymbol{y})$ depends only on the type $\boldsymbol{\gamma}$ of $\boldsymbol{v_i}$. Thus, we write $P_{\boldsymbol{X_i}\boldsymbol{Y}}(\boldsymbol{x_i},\boldsymbol{y})=\prod_{\ell=1}^{n}P^{\boldsymbol{\gamma}}_{X_iY}(x_{i\ell},y_\ell)$ where $P^{\boldsymbol{\gamma}}_{X_iY}(x_i,y)=P^{\boldsymbol{\gamma}}(x_i)P_{Y|X}(y|x_i)$. Similarly, $Q^{(j)}_{\boldsymbol{X_i}\boldsymbol{Y}}(\boldsymbol{x_i},\boldsymbol{y})$ depends only on the joint type $\boldsymbol{\lambda}$ of $\boldsymbol{v_i},\boldsymbol{v_j}$ and can be written as $\prod_{\ell=1}^{n}Q^{\boldsymbol{\lambda}}_{X_iY}(x_{i\ell},y_\ell)$ where $Q^{\boldsymbol{\lambda}}_{X_iY}(x_i,y)=\sum_{a\in\mathcal{X}}P^{\boldsymbol{\gamma}}(x_i)\cdot P^{\boldsymbol{\lambda}}(x_j=a|x_i)P_{Y|X}(y|x_j=a)$. We are now ready to state the main theorem of this paper.

Denoting $D(P||Q)$ as Kullback-Leibler distance and $H(P)$ as entropy, the sensing capacity at distortion D satisfies,

Sensing Capacity Theorem.

$$C(D)\geq C_{LB}(D)=\min_{\boldsymbol{\gamma}}\ \min_{\substack{\boldsymbol{\lambda}\\ \lambda_{(0\cdots)(1\cdots)}+\\ \lambda_{(1\cdots)(0\cdots)}>D}}\frac{D\left(P^{\boldsymbol{\gamma}}_{X_iY}||Q^{\boldsymbol{\lambda}}_{X_iY}\right)}{H(\boldsymbol{\lambda})-H(\boldsymbol{\gamma})}\tag{1}$$

where $\boldsymbol{\gamma}\in\mathcal{T}(\{0,1\}^c)$ and $\boldsymbol{\lambda}\in\mathcal{L}(\{0,1\}^{2c})$ are in the set of c-order types and c-order joint types respectively.

From the definition of $Q^{\boldsymbol{\lambda}}_{X_iY}$, we notice that if the 'codewords' $\boldsymbol{X_i}$ were independent, the Kullback-Leibler distance in (1) would reduce to the mutual information between $\boldsymbol{X_i}$

$P_{X_iX_j}$	$X_j = 0$	$X_j = 1$	$X_j = 2$
$X_i = 0$	$\lambda_{(00)(00)}$	$\lambda_{(00)(01)} + \lambda_{(00)(10)}$	$\lambda_{(00)(11)}$
$X_i = 1$	$\lambda_{(10)(00)} + \lambda_{(01)(00)}$	$\lambda_{(01)(01)} + \lambda_{(01)(10)} + \lambda_{(10)(01)} + \lambda_{(10)(10)}$	$\lambda_{(10)(11)} + \lambda_{(01)(11)}$
$X_i = 2$	$\lambda_{(11)(00)}$	$\lambda_{(11)(01)} + \lambda_{(11)(10)}$	$\lambda_{(11)(11)}$

Table 2. Joint distribution of X_j and X_i in terms of the joint type $\boldsymbol{\lambda}$ of $\boldsymbol{v_j}$ and $\boldsymbol{v_i}$, with $c = 2$.

and its noisy version $\boldsymbol{Y}$. Further, the denominator in (1) accounts for the non-identical distribution of the codewords. The minimization over the joint type appears, because the closest pair of codewords dominates the error probability. Thus, the 'sensing capacity' is similar to classical channel capacity, with differences arising due to non-identical, dependent codeword distribution. If we specialize this result to the case of $c = 1$ and restrict the sensing function to be a simple sum, this theorem provides a bound that coincides with our original result [1] for the case of $c = 1$.

Proof. We assume a maximum-likelihood decoder $g_{\text{ML}}(\boldsymbol{y}) = \arg\max_j P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_j})$. For this decoder, we consider $P_{e,\max} = \max_i P_{e,i}$, where $P_{e,i}$ is averaged over the random sensor network. We seek to bound $P_{e,i}$, which we write out below.

$$P_{e,i} = \sum_{\boldsymbol{x_i}\in\mathcal{X}^n} \sum_{\boldsymbol{y}\in\mathcal{Y}^n} P_{\boldsymbol{X}_i}(\boldsymbol{x_i}) P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_i}) \Pr[\text{error}|i, \boldsymbol{x_i}, \boldsymbol{y}] \tag{2}$$

We bound $\Pr[\text{error}|i, \boldsymbol{x_i}, \boldsymbol{y}]$ by defining events $A_{ij} = \{\boldsymbol{x_j} : P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_j}) \geq P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_i}) \mid i, \boldsymbol{x_i}, \boldsymbol{y}\}$ and using the union bound. Since decoding to a $j \notin \mathcal{D}_i$ results in error,

$$\Pr[\text{error}|i, \boldsymbol{x_i}, \boldsymbol{y}] \leq P(\cup_{j\notin\mathcal{D}_i} A_{ij}) \leq \sum_{j\notin\mathcal{D}_i} P(A_{ij}) \tag{3}$$

We proceed to bound $P(A_{ij})$. For any $s_{ij} \geq 0$:

$$\begin{aligned} P(A_{ij}) &= \sum_{\boldsymbol{x_j}\in A_{ij}} P_{\boldsymbol{X_j}|\boldsymbol{X_i}}(\boldsymbol{x_j}|\boldsymbol{x_i}) \\ &\leq \sum_{\boldsymbol{x_j}\in\mathcal{X}^n} P_{\boldsymbol{X_j}|\boldsymbol{X_i}}(\boldsymbol{x_j}|\boldsymbol{x_i}) \frac{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_j})^{s_{ij}}}{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_i})^{s_{ij}}} \end{aligned} \tag{4}$$

Using (3) and (4) in (2),

$$\begin{aligned} P_{e,i} \leq &\sum_{\boldsymbol{x_i}\in\mathcal{X}^n} \sum_{\boldsymbol{y}\in\mathcal{Y}^n} P_{\boldsymbol{X}_i}(\boldsymbol{x_i}) P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_i}) \cdot \\ &\sum_{j\notin\mathcal{D}_i} \sum_{\boldsymbol{x_j}\in\mathcal{X}^n} P_{\boldsymbol{X_j}|\boldsymbol{X_i}}(\boldsymbol{x_j}|\boldsymbol{x_i}) \frac{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_j})^{s_{ij}}}{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_i})^{s_{ij}}} \end{aligned} \tag{5}$$

The bound (5) has an exponential number of terms. However, it was argued earlier that in our sensor network, $P_{\boldsymbol{X}_i}(\boldsymbol{x_i}) = P^{\boldsymbol{\gamma},n}(\boldsymbol{x})$ depends only on the c-order type $\boldsymbol{\gamma}$ of the i^{th} target vector, while $P_{\boldsymbol{X_j}|\boldsymbol{X_i}}(\boldsymbol{x_j}|\boldsymbol{x_i}) = P^{\boldsymbol{\lambda},n}(\boldsymbol{x_j}|\boldsymbol{x_i})$ depends on the *c order joint type* of the i^{th} and j^{th} target vectors. Thus, we can rewrite (5) by grouping terms according to their c-order joint type $\boldsymbol{\lambda}$.

$$\begin{aligned} &\sum_{j\notin\mathcal{D}_i} \sum_{\boldsymbol{x_j}\in\mathcal{X}^n} P_{\boldsymbol{X_j}|\boldsymbol{X_i}}(\boldsymbol{x_j}|\boldsymbol{x_i}) \frac{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_j})^{s_{ij}}}{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_i})^{s_{ij}}} \leq \\ &\sum_{\boldsymbol{\lambda}\in S_i(D)} \beta(i,\boldsymbol{\lambda};k) \sum_{\boldsymbol{x_j}\in\mathcal{X}^n} P^{\boldsymbol{\lambda},n}(\boldsymbol{x_j}|\boldsymbol{x_i}) \frac{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_j})^{s_{\boldsymbol{\lambda}}}}{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_i})^{s_{\boldsymbol{\lambda}}}} \end{aligned} \tag{6}$$

where $S_i(D)$ is the set of c-order joint types that result in an error.[1] i.e.,

$$\begin{aligned} S_i(D) = \{\boldsymbol{\lambda} : \boldsymbol{\lambda} \in \mathcal{L}_k(\{0,1\}^{2c}), \\ \lambda_{(0\cdots)(1\cdots)} + \lambda_{(1\cdots)(0\cdots)} > D\} \end{aligned} \tag{7}$$

and where we choose $s_{ij} = s_{\boldsymbol{\lambda}}$ for all $\{i,j\}$ of c-order joint type $\boldsymbol{\lambda}$. Here $\beta(i,\boldsymbol{\lambda};k)$ is the number of vectors $\boldsymbol{v_j}$ that have a joint type $\boldsymbol{\lambda}$ with respect to $\boldsymbol{v_i}$. To obtain (6), we used the fact that $\lambda_{(0\cdots)(1\cdots)} + \lambda_{(1\cdots)(0\cdots)} \leq d_H(\boldsymbol{v_i},\boldsymbol{v_j}) \leq \lambda_{(0\cdots)(1\cdots)} + \lambda_{(1\cdots)(0\cdots)} + \frac{c-1}{k}$. For large k, equality is achieved in (6). $\beta(i,\boldsymbol{\lambda};k)$ is bounded as,

$$\beta(i,\boldsymbol{\lambda};k) \leq 2^{k(H(\boldsymbol{\lambda})-H(\boldsymbol{\gamma}))} \tag{8}$$

Combining equations (5),(6), and (8),

$$\begin{aligned} P_{e,i} \leq &\sum_{\boldsymbol{x_i}\in\mathcal{X}^n} \sum_{\boldsymbol{y}\in\mathcal{Y}^n} P^{\boldsymbol{\gamma},n}(\boldsymbol{x_i}) P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_i}) \cdot \\ &\sum_{\boldsymbol{\lambda}\in S_i(D)} 2^{k(H(\boldsymbol{\lambda})-H(\boldsymbol{\gamma}))} \sum_{\boldsymbol{x_j}\in\mathcal{X}^n} P^{\boldsymbol{\lambda},n}(\boldsymbol{x_j}|\boldsymbol{x_i}) \frac{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_j})^{s_{\boldsymbol{\lambda}}}}{P_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y}|\boldsymbol{x_i})^{s_{\boldsymbol{\lambda}}}} \end{aligned}$$

We now use the independence of the sensor outputs. Further, since we are bounding a probability, the following bound holds for $\rho_{\boldsymbol{\lambda}} \in [0,1]$ and $s_{\boldsymbol{\lambda}} = \frac{1}{1+\rho_{\boldsymbol{\lambda}}}$.

$$\begin{aligned} P_{e,i} \leq &\sum_{\boldsymbol{\lambda}\in S_i(D)} 2^{\rho_{\boldsymbol{\lambda}} k(H(\boldsymbol{\lambda})-H(\boldsymbol{\gamma}))} \cdot \Bigg(\sum_{a_i\in\mathcal{X}} \sum_{b\in\mathcal{Y}} P_{Y|X}(b|a_i)^{\frac{1}{1+\rho_{\boldsymbol{\lambda}}}} \\ &\cdot P^{\boldsymbol{\gamma}}(a_i) \left(\sum_{a_j\in\mathcal{X}} P^{\boldsymbol{\lambda}}(a_j|a_i) P_{Y|X}(b|a_j)^{\frac{1}{1+\rho_{\boldsymbol{\lambda}}}} \right)^{\rho_{\boldsymbol{\lambda}}} \Bigg)^n \end{aligned} \tag{9}$$

We define the following quantity.

$$\begin{aligned} E(\rho_{\boldsymbol{\lambda}},\boldsymbol{\lambda}) = -\log \Bigg(&\sum_{a_i\in\mathcal{X}} \sum_{b\in\mathcal{Y}} P^{\boldsymbol{\gamma}}(a_i) P_{Y|X}(b|a_i)^{\frac{1}{1+\rho_{\boldsymbol{\lambda}}}} \\ &\cdot \left(\sum_{a_j\in\mathcal{X}} P^{\boldsymbol{\lambda}}(a_j|a_i) P_{Y|X}(b|a_j)^{\frac{1}{1+\rho_{\boldsymbol{\lambda}}}} \right)^{\rho_{\boldsymbol{\lambda}}} \Bigg) \end{aligned} \tag{10}$$

[1] Technically $S_i(D)$ is a bit larger than that set, but the bound still holds.

Since the number of types of $\boldsymbol{\lambda}$ is upper bounded by $(k+1)^{4^c}$, and $k = \lceil nR \rceil$, (9) is bounded as,

$$P_{e,i} \leq \max_{\boldsymbol{\lambda} \in S_i(D)} \min_{0 \leq \rho_{\boldsymbol{\lambda}} \leq 1} 2^{-n\left(\frac{-4^c \log(nR+2)}{n}\right)} \cdot 2^{-n\left(-(1+\frac{1}{nR})\rho_{\boldsymbol{\lambda}} R(H(\boldsymbol{\lambda})-H(\boldsymbol{\gamma}))+E(\rho_{\boldsymbol{\lambda}},\boldsymbol{\lambda})\right)}$$

We seek to bound $\max_i \ P_{e,i}$. However, $P_{e,i}$ only depends on the type $\boldsymbol{\gamma}$ of $\boldsymbol{v_i}$. Thus, we have the bound,

$$P_{e,\max} \leq 2^{-n(-o_1(n)+E_r(R,D))}$$
$$E_r(R,D) = \min_{\boldsymbol{\gamma}} \min_{\boldsymbol{\lambda} \in S(D)} \max_{0 \leq \rho_{\boldsymbol{\lambda}} \leq 1} E(\rho_{\boldsymbol{\lambda}},\boldsymbol{\lambda}) - \rho_{\boldsymbol{\lambda}} R(H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma}))$$

where $\boldsymbol{\gamma} \in \mathcal{T}_k(\{0,1\}^c)$, and $S(D)$ is as in (7), with $\boldsymbol{\gamma}$. Note that $o_1(n) \to 0$ as $n \to \infty$, so we have not included it in the error exponent $E_r(R,D)$. Observing that $E(0,\boldsymbol{\lambda}) = 0 \ \forall \ \boldsymbol{\lambda}$, we let $\rho_{\boldsymbol{\lambda}}$ go to zero, rather than optimizing it, thus resulting in a lower bound on $E_r(R,D)$. In the above expression, this implies that in order for R to be achievable $\frac{E(\rho_{\boldsymbol{\lambda}},\boldsymbol{\lambda})}{\rho_{\boldsymbol{\lambda}}} - R(H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma}))$ must be positive for all $\boldsymbol{\gamma}, \boldsymbol{\lambda}$, even as $\rho_{\boldsymbol{\lambda}} \to 0$. But this implies that the derivative of $E(\rho_{\boldsymbol{\lambda}},\boldsymbol{\lambda})$ with respect to $\rho_{\boldsymbol{\lambda}}$ at $\rho_{\boldsymbol{\lambda}} = 0$ must be greater than $R(H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma}))$. But it can be easily shown that,

$$\left.\frac{\partial E(\rho_{\boldsymbol{\lambda}},\boldsymbol{\lambda})}{\partial \rho_{\boldsymbol{\lambda}}}\right|_{\rho_{\boldsymbol{\lambda}}=0} = D(P^{\boldsymbol{\gamma}}_{X_iY} \| Q^{\boldsymbol{\lambda}}_{X_iY}) \tag{11}$$

Using this derivative in the analysis above, and relaxing the conditions $\boldsymbol{\lambda} \in \mathcal{L}_k(\{0,1\}^{2c})$ by dropping the restriction that target vectors are restricted to length k in the definition (7) of $S(D)$ (thus, weakening the bound), we see that the sensor network can achieve any rate R bounded as below.

$$R \leq \min_{\boldsymbol{\gamma}} \min_{\substack{\boldsymbol{\lambda} \\ \lambda_{(0\cdots)(1\cdots)}+ \\ \lambda_{(1\cdots)(0\cdots)}>D}} \frac{D\left(P^{\boldsymbol{\gamma}}_{X_iY} \| Q^{\boldsymbol{\lambda}}_{X_iY}\right)}{H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma})} \tag{12}$$

Therefore the Right Hand Side is a lower bound on $C(D)$. □

4. SENSOR NETWORK MODEL EXTENSIONS

We consider two straight-forward extensions to our sensor network model. The first extension considers non-binary target vectors. Binary target vectors indicate the presence or absence of targets at the spatial positions. A target vector over a general finite alphabet may indicate, in addition to the presence of targets, the class of a target, or the intensity or concentration of each target. Assuming a target vector over alphabet $\mathcal{A}$, we obtain the capacity bound below.

$$C(D) \geq C_{LB}(D) = \min_{\boldsymbol{\gamma}} \min_{\substack{\boldsymbol{\lambda} \\ \sum_{a \neq b} \lambda_{(a\cdots)(b\cdots)}>D}} \frac{D\left(P^{\boldsymbol{\gamma}}_{X_iY} \| Q^{\boldsymbol{\lambda}}_{X_iY}\right)}{H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma})}$$

where $\boldsymbol{\gamma} \in \mathcal{T}_k(\mathcal{A}^c)$ and $\boldsymbol{\lambda} \in \mathcal{L}_k(\mathcal{A}^{2c})$.

A further extension considers the case of heterogenous sensors, where each class of sensor has a different sensing function Ψ and noise model $P_{Y|X}$. Let the sensor of class l be used with a relative frequency α_l. Then,

$$C(D) \geq C_{LB}(D) = \min_{\boldsymbol{\gamma}} \min_{\substack{\boldsymbol{\lambda} \\ \sum_{a \neq b} \lambda_{(a\cdots)(b\cdots)}>D}} \frac{\sum_l \alpha_l D\left(P^{\boldsymbol{\gamma},l}_{X_iY} \| Q^{\boldsymbol{\lambda},l}_{X_iY}\right)}{H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma})}$$

where $\boldsymbol{\gamma} \in \mathcal{T}_k(\mathcal{A}^c)$ and $\boldsymbol{\lambda} \in \mathcal{L}_k(\mathcal{A}^{2c})$.

5. CAPACITY BOUND EXAMPLES

We compute the capacity bound $C_{LB}(D)$ for various sensor network models. In Figure 2, we compare $C_{LB}(D)$ for sensor networks with localized and non-localized [1] sensing. We assume that the sensing function Ψ is an un-weighted additive function. The sensor noise model used throughout this section assumes that the probability of error decays exponentially with the error magnitude. In the figures, 'Noise = p' indicates that for a sensor, $P(Y \neq X) = p$, with $\mathcal{Y} = \mathcal{X}$ assumed. Figure 2 demonstrates $C_{LB}(D)$ for localized and non-localized sensing, at two sensor noise levels, and a fixed sensing range $c = 3$. Sensor localization causes a significant reduction in sensing capacity. We conjecture that this effect is similar to the inferior performance of a channel code which has a finite memory, such as convolutional codes, as opposed to LDPC codes which have large memory.

Figure 3 shows $C_{LB}(D = 0.1)$ for a weighted sum sensing function and compares this to an un-weighted sum sensing function. We assume a range $c = 2$ with weights $[0.5, 1]$. The weighted sum sensing function has a higher bound across all sensor noise levels. Intuitively, this occurs because the weighted sum distinguishes between its two connections, resulting in less ambiguity.

Figure 3 also demonstrates that sensor cooperation is more efficient than the commonly used strategy of simple sensor replication. For example, a rate of 0.053 targets/sensor is achievable for sensors with a noise level of 0.2 and the weighted sum sensing function. If instead, each sensor is replicated thrice (thus, requiring three times as many sensors, while also reducing the noise level to $3 \times (0.2)^2 \times 0.8 + (0.2)^3 = 0.1$ due to majority-decoding), then the resulting rate *reduces* to $0.096/3 = 0.032$ targets/sensor.

6. SEISMIC SENSOR NETWORK

We compare our bound to the performance of a practical sensor network decoding algorithm. We consider an idealized seismic sensor network (Figure 1), where each block in a grid contains a target or nothing. Seismic sensors are

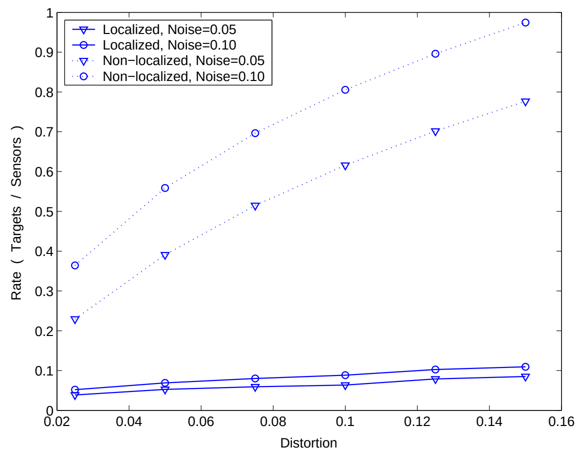

Fig. 2. $C_{LB}(D)$ for localized and non-localized sensors.

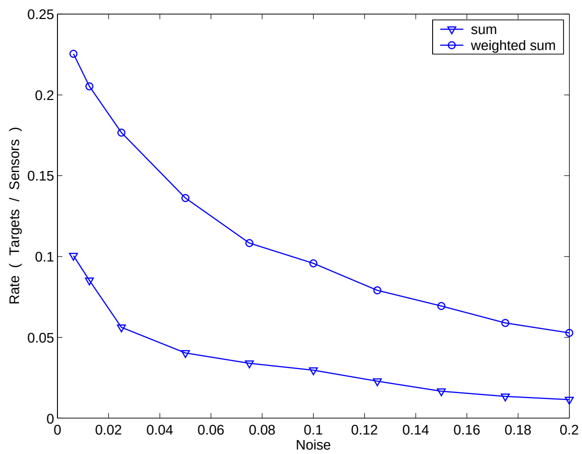

Fig. 3. $C_{LB}(D = 0.1)$ for a weighted sum and un-weighted sum sensing function.

randomly dropped on this grid. Each sensor senses c contiguous blocks, and outputs the weighted sum of vibration amplitudes caused by each target in the sensed blocks.

Inspired by its success in decoding LDPC codes, we used the belief propagation algorithm [20] to fuse the observation of seismic sensors to obtain an estimate of the spatial target configuration in the grid. Borrowing from [20], we introduce the following notation in order to describe the belief propagation algorithm for our sensor network model. We denote the set of targets sensed by sensor ℓ by $\mathcal{M}(\ell)$. Analogously, we define $\mathcal{L}(m)$ as the set of sensors that sense the target m. We denote the set $\mathcal{M}(\ell)$ with target m excluded by $\mathcal{M}(\ell)\backslash m$, and similarly we denote the set $\mathcal{L}(m)$ with sensor ℓ excluded by $\mathcal{L}(m)\backslash\ell$. Let $p^v_m = P(v_m = v)$ be the prior probability of the target bits. The algorithm consists of two parts, where two sets of quantities, $q_{m\ell}$ and $r_{m\ell}$, are iteratively updated. We now proceed to describe the belief propagation algorithm for our sensor network model.

We initialize the algorithm by letting $q^v_{m\ell} = p^v_m$. In the *sensor* step of the algorithm we compute the $r_{m\ell}$ quantities using the following expressions.

$$r^v_{m\ell} = \sum_{b\in\mathcal{X}} P_{Y|X}(y_\ell|b) \sum_{\substack{\boldsymbol{v}'\in\{v_{m'}:m'\in\mathcal{M}(\ell)\backslash m\}\\ \Psi(\boldsymbol{v}',v)=b}} \prod_{m'\in\mathcal{M}(\ell)\backslash m} q^{v_{m'}}_{m'\ell}$$

The *target* step computes $q_{m\ell}$ values from the computed $r_{m\ell}$ values as below (where $\alpha_{m\ell} = q^0_{m\ell} + q^1_{m\ell}$).

$$q^v_{m\ell} = \alpha^{-1}_{m\ell} p^v_m \prod_{\ell'\in\mathcal{L}(m)\backslash\ell} r^v_{m\ell'}$$

After a fixed number of iterations one can halt the algorithm and compute the probabilities of each target bit as shown below (where $\alpha_m = q^0_m + q^1_m$). These probabilities can be used to decode the target vector.

$$q^v_m = \alpha^{-1}_m p^v_m \prod_{\ell\in\mathcal{L}(m)} r^v_{m\ell}$$

Using this decoding algorithm we empirically examined seismic sensor network performance as a function of rate. We generated sensor networks of various rates by setting the number of targets, and varying the number of sensors. We chose the number of connections to be $c = 2$ (with weights 0.5 and 1.0), the distortion level to be 0.1, and the noise level to be 0.1 (i.e. $P(Y \neq X) = 0.1$, with $\mathcal{Y} = \mathcal{X}$). As in the previous section, we assume that the probability of error decays exponentially with error magnitude. We empirically evaluated (using belief propagation) the maximum error rate averaged over a set of randomly generated sensor networks. We plotted the maximum error rate over all sampled target vectors for each rate value, and for various numbers of targets as shown in Figure 4. The capacity value C_{LB} for the model used in this experiment is 0.097. Since belief propagation is suboptimal for graphs with cycles, and given that the error curves converge to zero at rates above 0.097, it appears that our capacity lower bound is not tight. As the number of targets increases, the decay in error becomes sharper, which indicates an information theoretic capacity effect. Unfortunately, belief propagation worked poorly as a practical decoder for more than two connections. We conjecture that this occurs because sensing is localized in our model, and thus the number of short cycles is quite large and the graph does not appear tree-like. Therefore, the loopy belief propagation approximation performs poorly. In future work, we hope to overcome this difficulty by using generalized belief propagation.

7. CONCLUSIONS

We explored a notion of sensing capacity for sensor networks with localized sensing and arbitrary sensing functions. We proved a lower bound $C_{LB}(D)$ to the sensing capacity and computed it for illustrative examples. Our bound

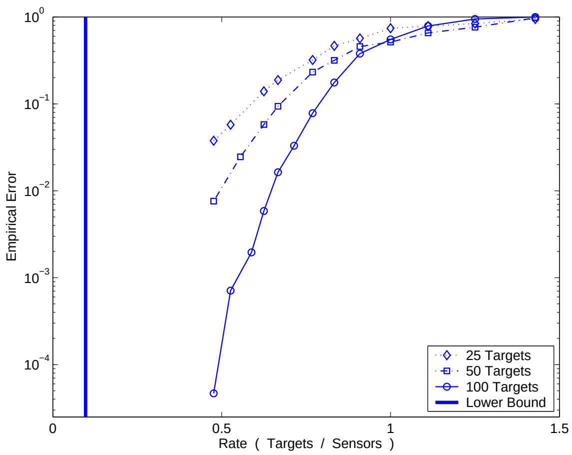

Fig. 4. Maximum (over target vectors) empirical error rate of belief propagation based decoding of seismic sensor networks.

can be extended to account for non-binary target vectors and heterogeneous sensors. We conclude that $C_{LB}(D)$ for sensors with non-localized observations is significantly higher that for sensors with localized observations. We also show that one can significantly vary the sensing capacity by choosing different sensing functions. By examining $C_{LB}(D)$, we concluded that simple sensor replication is inefficient compared to sensor cooperation. We derived a belief propagation algorithm for decoding our sensor network model. Using this algorithm, we empirically evaluated capacity for an idealized seismic sensor network and compared the result to $C_{LB}(D)$.

8. REFERENCES

[1] Y. Rachlin, R. Negi, and P. Khosla, "Sensing capacity for target detection," in *Proc. IEEE Inform. Theory Wksp.*, Oct. 24-29 2004.

[2] E. J. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: organization and capacity," *Comp. Net.: Sp. Issue on Wireless Sensor Net.*, 2003.

[3] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, 2000.

[4] G. Barrenechea, B. Beferull-Lozano, and M. Vetterli, "Lattice sensor networks: Capacity limits, optimal routing and robustness to failures," in *IPSN* 2004.

[5] U. Mitra and A. Sabharwal, "Complexity constrained sensor networks: Achievable rates for two relay networks and generalizations," in *IPSN* 2004.

[6] A. Scaglione and S. D. Servetto, "On the interdependence of routing and data compression in multi-hop sensor networks," in *Proc. 8th ACM Int. Conference on Mobile Computing and Networking*, Sept. 2002.

[7] M. Gastpar and M. Vetterli, "Power-bandwidth-distortion scaling laws for sensor networks," *IPSN* 2004.

[8] J. Chamberland and V. Veeravalli, "Asymptotic results for decentralized detection in power constrained wireless sensor networks," *IEEE JSAC Sp. Issue on Wireless Sensor Networks*, pp. 1007–1015, 2004.

[9] Q. Fang, F. Zhao, and L. Guibas, "Counting targets: Building and managing aggregates in wireless sensor networks," Tech. Rep. P2002-10298, PARC, 2002.

[10] D. Yang, H. Gonzalez-Banos, and L. Guibas, "Counting people in crowds with a real-time network of image sensors," *Proc. Int. Conf. Comp. Vision*, 2003.

[11] A. Hoover and B. Olsen, "A real-time occupancy map from multiple video streams," in *Proc. Int. Conf. on Robotics and Automation*, 1999.

[12] Marco Duarte and Yu-Hen Hu, "Vehicle classification in distributed sensor networks," *Jour. Parallel Dist. Comp.*, pp. 826–838, 2004.

[13] J. Thorpe and R. McEliece, "Data fusion algorithms for collaborative robotic exploration," *IPN Progress Report 42-149*, JPL NASA, 2002.

[14] K. Chakrabarty, S. Iyengar, H. Qi, and E. Cho, "Coding theory framework for target location in distributed sensor networks," *Proc. Int. Conf. Inf. Tech.* , 2001.

[15] P. Varshney, *Distributed Detection and Data Fusion*, Springer-Verlag, 1997.

[16] J. Kotecha, V. Ramachandran, and A. Sayeed, "Distributed multi-target classification in wireless sensor networks," *IEEE JSAC Sp. Self-Organizing Dist. Collab. Sensor Net.*, 2003.

[17] Y. Rachlin, J. Dolan, and P. Khosla, "Multirobot sensor fusion in mine detection," submitted to *Int. Conf. on Robotics and Automation*, 2005.

[18] R.G. Gallager, *Information Theory and Reliable Communications*, Wiley, 1968.

[19] I. Csiszar, "The method of types," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, 1998.

[20] D. Mackay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.