2004

Sensing Capacity for Target Detection

Yaron Rachlin
Carnegie Mellon University

Rohit Negi
Carnegie Mellon University

Pradeep Khosla
Carnegie Mellon University

Sensing Capacity for Target Detection

Yaron Rachlin, Rohit Negi, and Pradeep Khosla
Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213
{rachlin, negi, pkk}@ece.cmu.edu

Abstract — **We define a notion of 'sensing capacity' that characterizes the ability of a sensor network to successfully distinguish among a discrete set of targets. Sensing capacity is defined as the maximum ratio of target positions to sensors for which inference of targets within a certain distortion is achievable. We demonstrate a lower bound on this capacity. Unlike previous work on 'sensor network capacity', our notion of sensing capacity is defined by the sensing task itself, as opposed to external resource constraints such as power, communications, and processing.**

I. Introduction

Sensor networks consist of a set of sensors that cooperatively sense an environment. Previous research on limits of performance of such networks concentrated on network channel capacity, under various resource constraints. In this paper, we consider a target detection problem and prove the existence of a 'sensing capacity' $C(D)$, such that, for a given tolerable distortion D, if the ratio of target positions to sensors is smaller than $C(D)$, the maximal average error probability converges to zero as the number of target positions (and sensors) goes to infinity.

Previous work on 'sensor network capacity' seeks to characterize the capacity by exploring the constraints imposed by power, communications, and computation. [2], [3] extend the results in [1] to a sensor network scenario and show communication-based limits on the amount of data that a sensor network can transport. [4] has considered the interaction between transmission rates and power constraints to obtain statements of capacity. Interestingly, [5] combines the notion of transport capacity with knowledge about the nature of the sensing task, which in his case is to sense an underlying continuous random process to within a given distortion. [6] extends this work by accounting for compression at each node.

In contrast, we provide a statement of *sensing capacity* inherent to the sensing task of detecting discrete targets to within a given distortion, rather than using external resource constraints. Section II introduces the sensor network model. Section III states the main theorem. Illustrative calculations of the sensing capacity are presented in Section IV, and empirical evaluations of capacity using a belief propagation algorithm are presented in Section V. Section VI extends the result to more general cases and Section VII concludes the paper.

II. Sensor Network Model

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

We consider the problem of detecting discrete targets. An example of such work includes a target counting protocol for a sensor network consisting of seismic sensors implemented by [7]. In another example, a sensor network consisting of multiple cameras was designed to count the number of people in a crowd [8]. Chemical sensor arrays, consisting of an array of semi-selective chemical sensors, can distinguish among a discrete set of substances and represent another application [9]. [10] proposed an abstract sensor network model for discrete target location. A coding-based approach was demonstrated to bound the minimum number of sensors required for discrimination, but no notion of sensing capacity was considered.

Our sensor network model is motivated by the following specific scenarios. In a seismic sensor network, each sensor can count the number of targets it senses (based on the intensity of vibration). Each sensor is affected by targets in several locations within a region, randomly distributed due to variations in soil composition. In a camera-based people-counting scenario, the view is broken into a target-sized grid, where each grid square may contain at most one person. In this scenario, each camera is affected by several grid squares randomly, due to random occlusions in its view. In a chemical sensor array scenario, a complex substance can be modeled as a discrete vector where each bit represents a constituent compound. Each sensor in the array responds to a random subset of compounds, and the output of the chemical sensor can be modeled as linear under certain conditions. The output of such an array can distinguish among different substances. In large scale image processing to detect sparsely distributed targets, instead of searching over the entire image, one can break up the image into a grid and process random combinations of the grid squares to save computation. Element analysis provides another scenario of sensor cooperation. Rather than separately analyzing several substances for a constituent element, one can view the set of substances as a bit vector, where '1' indicates the presence of the element. Combinations of substances can be analyzed (each analysis corresponds to one sensor) to detect the element.

The model we present here is a first-cut attempt to abstractly characterize the essence of various discrete target detection applications for sensor networks, as motivated by the above scenarios. Figure 1 shows our sensor network model. There are k spatial positions that need to be sensed. Each position may represent an actual region in space, or may have other interpretations, such as a substance, in the elemental analysis example. Each position may contain no target or one target. A k-bit 'target vector' $\boldsymbol{v}$ represents the target configuration in these k positions. The figure shows $\boldsymbol{v} = (0, 0, 1, 0, 1, 1, 1)$ indicating 4 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.

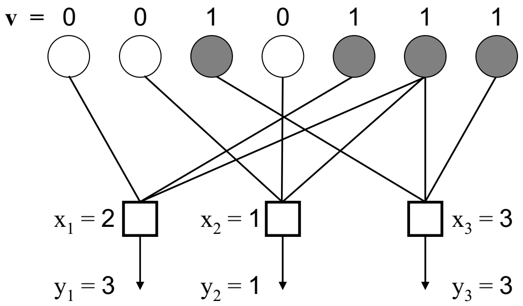

Figure 1: Sensor network model with $k = 7, n = 3, c = 3$.

The sensor network has n identical sensors. Sensor ℓ is connected to (i.e., senses) exactly c out of the k spatial positions. Its function is to indicate the *number of positions* (among the c positions it senses) $x \in \mathcal{X} = \{0, 1, \ldots, c\}$ that contain a target. For example, a seismic sensor can sense the intensity of vibration to detect the number of targets. Thus, the 'ideal output vector' of the sensor network $\boldsymbol{x}$ depends on the sensor connections, 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 output. Since the sensors are identical, $P_{Y|X}$ is the same for all the sensors. 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 $S(k, n)$ as the bipartite graph showing the connections of the sensors to the k spatial positions. We assume a simple sensor network model, where each of the c connections of each sensor is independently made to a spatial position, chosen equi-probable among the k positions. Although this model is a simplification of general sensor networks, it accurately describes the specific sensing scenarios described above. Furthermore, its analysis will, hopefully, motivate the analysis of more complicated models in the future.

III. Sensor Network Capacity Theorem

For such a randomly generated sensor network, the ideal output $\boldsymbol{x}$ is a function of the sensor network instantiation $s(k, n)$ and the occurring target vector $\boldsymbol{v}$. Considering $\boldsymbol{X_i}$ as the random vector which occurs when $\boldsymbol{v_i}$ is the target vector (i.e., $\boldsymbol{X_i}$ is random because of the random generation of $S(k, n)$), we can obtain the p.m.f. of $\boldsymbol{X_i}$ very simply. Since each sensor counts the number of targets it senses, and its connections are formed independently, $P_{\boldsymbol{X_i}}(\boldsymbol{x_i}) = \prod_{\ell=1}^{n} P_{X_i}(x_{i\ell})$. 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 connections produce a dependency between them. However, the sensors are independent, given the target vector, so that $P_{\boldsymbol{X_i}\boldsymbol{X_j}}(\boldsymbol{x_i}, \boldsymbol{x_j}) = \prod_{\ell=1}^{n} P_{X_i X_j}(x_{i\ell}, x_{j\ell})$. Thus, the 'codewords' $\{\boldsymbol{X_i}, i = 1, 2, \ldots, 2^k\}$ of the sensor network are non-identical and dependent on each other, unlike channel codes in classical information theory.

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 $d_{\mathrm{H}}(\boldsymbol{v_i}, \boldsymbol{v_j})$ is the Hamming distance between two target vectors and if we define the tolerable distortion region of $\boldsymbol{v_i}$ as $\mathcal{D}_i = \{j : \frac{1}{k} d_{\mathrm{H}}(\boldsymbol{v_i}, \boldsymbol{v_j}) < D\}$, then given that $\boldsymbol{v_i}$ occurred, the probability of error is $P_{e,i,s} = \Pr[\text{error}|i, s, \boldsymbol{x_i}, \boldsymbol{y}] = \Pr[g(\boldsymbol{y}) \notin \mathcal{D}_i | i, s, \boldsymbol{x_i}, \boldsymbol{y}]$. Averaging this probability over all sensor networks, we write the average error probability, given that $\boldsymbol{v_i}$ occurred, as $P_{e,i} = E[P_{e,i,s}]$. We use the maximal average error probability $P_{e,\max} = \max_i \, P_{e,i}$ as our error metric.

We define the 'rate' of the sensor network as the ratio of target positions to sensors, $R = \frac{k}{n}$. Given a tolerable distortion D, we call R achievable if the sequence of sensors networks $S(\lceil nR \rceil, n)$ satisfies $P_{e,\max} \to 0$ as $n \to \infty$. The *sensing capacity* of the sensor network is defined as $C(D) = \max R$ over achievable R.

The main result of this paper is to show that the sensing capacity $C(D)$ of the sensor network is non-zero, and to characterize it as a function of noise $P_{Y|X}$ and sensor connections c. The proof of this result broadly follows the proof of channel capacity provided by Gallager [11], by analyzing a union bound of pair-wise error probabilities, averaged over randomly generated sensor networks. However, it differs from [11] in several important ways. In our sensor network model, the distribution of the 'encoder' (i.e., sensor network generation) is fixed. Given the encoder (sensor network), the codewords are dependent on each other. Further, the 'codebook' $\{\boldsymbol{x_i}\}$ obtained is non-linear, so that techniques used to analyze linear random codes [12], which use the parity check matrix for analysis, are not applicable. However, since each sensor in our network counts the number of targets, our model is symmetric with respect to permutations of the target vector $\boldsymbol{v}$. This allows us to use the method of types to group the exponential number of pair-wise error probability terms into a polynomial number of (joint) types in order to prove convergence of error probability.

The statement of the main result requires an explanation of *joint types.* Since each sensor counts the number of targets it observes, and the sensor makes each of its c connections to the spatial positions independently, therefore for each i, the distribution of its ideal output X_i depends only on the type $\boldsymbol{\gamma} = (\gamma_0, \gamma_1)$ of the i^{th} target vector $\boldsymbol{v_i}$. i.e., only on the number of 0's and 1's in $\boldsymbol{v_i}$. Here, γ_0 denotes the fraction of zeros in $\boldsymbol{v_i}$. Due to this permutation symmetry, $P_{\boldsymbol{X_i}}(\boldsymbol{x_i}) = P^{\boldsymbol{\gamma},n}(\boldsymbol{x_i}) = \prod_{\ell=1}^{n} P^{\boldsymbol{\gamma}}(x_{i\ell})$ for all $\boldsymbol{v_i}$ of the same type $\boldsymbol{\gamma}$.

Next, we note that the conditional probability $P_{\boldsymbol{X_j}|\boldsymbol{X_i}}$ depends on the *joint type* of the i^{th} and j^{th} target vectors. i.e., Let λ_{01} be the fraction of positions in $\boldsymbol{v_i}, \boldsymbol{v_j}$ where $\boldsymbol{v_i}$ has bit '0' while $\boldsymbol{v_j}$ has bit '1'. Similarly, define $\lambda_{00}, \lambda_{10}, \lambda_{11}$ and define $\boldsymbol{\lambda} = (\lambda_{00}, \lambda_{01}, \lambda_{10}, \lambda_{11})$. We call $\boldsymbol{\lambda}$ the 'joint type' of $\boldsymbol{v_i}, \boldsymbol{v_j}$. Following the notation introduced in [13], $\boldsymbol{\lambda} \in \mathcal{P}_k(\{0, 1\}^2)$, indicating that $\boldsymbol{\lambda}$ is in the set of joint types of k-bit binary vector pairs. Again, since each sensor exhibits permutation symmetry, $P_{\boldsymbol{X_j}|\boldsymbol{X_i}}$ depends only on the joint type $\boldsymbol{\lambda}$. i.e., $P_{\boldsymbol{X_j}|\boldsymbol{X_i}}(\boldsymbol{x_j}|\boldsymbol{x_i}) = P^{\boldsymbol{\lambda},n}(\boldsymbol{x_j}|\boldsymbol{x_i}) = \prod_{\ell=1}^{n} P^{\boldsymbol{\lambda}}(x_{j\ell}|x_{i\ell})$ for all i, j of the same joint type $\boldsymbol{\lambda}$. Since the joint type $\boldsymbol{\lambda}$ also defines the type $\boldsymbol{\gamma}$ of $\boldsymbol{v_i}$, we must have $\lambda_{00} + \lambda_{01} = \gamma_0$, $\lambda_{10} + \lambda_{11} = \gamma_1$.

To illustrate, Table 1 lists the joint type of 4 vectors $\boldsymbol{v_j}$ with $i = 1$ (Thus, $\gamma = (5/8, 3/8)$ here.) As an example, consider a sensor network where each sensor is connected randomly to

$\boldsymbol{v_1}$ 01101000	(5/8,0,0,3/8)
$\boldsymbol{v_2}$ 00011110	(2/8,3/8,2/8,1/8)
$\boldsymbol{v_3}$ 01000111	(2/8,3/8,2/8,1/8)
$\boldsymbol{v_4}$ 00000000	(5/8,0,3/8,0)

Table 1: $\boldsymbol{\lambda}$ for 4 target vectors $\boldsymbol{v_j}$, with $i = 1$.

$c = 2$ spatial positions. Thus, each sensor has an ideal output alphabet $\mathcal{X} = \{0, 1, 2\}$. 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 its connections are connected to spatial positions that have a '0' bit in both these target vectors. This happens with probability $(\lambda_{00})^2$. Table 2 lists the joint p.m.f. $P_{X_iX_j}(x_i, x_j) = P^{\gamma}(x_i)P^{\boldsymbol{\lambda}}(x_j|x_i)$ for all output pairs x_i, x_j corresponding to joint type $\boldsymbol{\lambda}$.

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 output $\boldsymbol{y}$ caused by it. i.e., $P_{\boldsymbol{X_iY}}(\boldsymbol{x_i}, \boldsymbol{y}) = \prod_{\ell=1}^{n} P_{X_iY}(x_{i\ell}, y_\ell) = \prod_{\ell=1}^{n} P_{X_i}(x_{i\ell})P_{Y|X}(y_\ell|x_{i\ell})$. 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 $Q^{(j)}_{\boldsymbol{X_iY}}(\boldsymbol{x_i}, \boldsymbol{y}) = \prod_{\ell=1}^{n} Q^{(j)}_{X_iY}(x_{i\ell}, y_\ell) = \prod_{\ell=1}^{n} \sum_{a\in\mathcal{X}} P_{X_i}(x_{i\ell})P_{X_j|X_i}(x_j = a|x_{i\ell})P_{Y|X}(y_\ell|x_j = a)$. Note that $\boldsymbol{X_i}, \boldsymbol{Y}$ are dependent here, although $\boldsymbol{Y}$ was produced by $\boldsymbol{X_j}$ because of the dependence of $\boldsymbol{X_i}, \boldsymbol{X_j}$.

Since the sensor network exhibits permutation symmetry, $P_{\boldsymbol{X_iY}}(\boldsymbol{x_i}, \boldsymbol{y})$ depends only on the type γ of $\boldsymbol{v_i}$. Thus, we write $P_{\boldsymbol{X_iY}}(\boldsymbol{x_i}, \boldsymbol{y}) = \prod_{\ell=1}^{n} P^{\gamma}_{X_iY}(x_{i\ell}, y_\ell)$ where $P^{\gamma}_{X_iY}(x_i, y) = P^{\gamma}(x_i)P_{Y|X}(y|x_i)$. Similarly, $Q^{(j)}_{\boldsymbol{X_iY}}(\boldsymbol{x_i}, \boldsymbol{y})$ 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 $Q^{\boldsymbol{\lambda}}_{X_iY}(x_i, y) = \sum_{a\in\mathcal{X}} P^{\gamma}(x_i)P^{\boldsymbol{\lambda}}(x_j = a|x_i)P_{Y|X}(y|x_j = a)$. 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 is bounded as,

Sensing Capacity Theorem.

$$C(D) \geq C_{LB}(D) = \min_{\gamma} \min_{\substack{\boldsymbol{\lambda} \\ \lambda_{01}+\lambda_{10}>D \\ \lambda_{00}+\lambda_{01}=\gamma_0 \\ \lambda_{10}+\lambda_{11}=\gamma_1}} \frac{D\left(P^{\gamma}_{X_iY}||Q^{\boldsymbol{\lambda}}_{X_iY}\right)}{H(\boldsymbol{\lambda}) - H(\gamma)} \quad (1)$$

where $\gamma = (\gamma_0, \gamma_1)$ and $\boldsymbol{\lambda} = (\lambda_{00}, \lambda_{01}, \lambda_{10}, \lambda_{11})$ are two arbitrary probability mass functions.

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 $\boldsymbol{X_i}$ 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.

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

Table 2: Joint distribution of X_j and X_i in terms of the joint type $\boldsymbol{\lambda}$ of $\boldsymbol{v_j}$ and $\boldsymbol{v_i}$. Each sensor makes $c = 2$ connections.

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.

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

We bound the expression $\Pr[\text{error}|i, \boldsymbol{x_i}, \boldsymbol{y}]$ 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,

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

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

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

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

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

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

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

where

$$\begin{aligned} S_i(D) = \{\boldsymbol{\lambda} : \boldsymbol{\lambda} \in \mathcal{P}_k(\{0,1\}^2), \lambda_{01} + \lambda_{10} > D, \\ \lambda_{00} + \lambda_{01} = \gamma_0, \lambda_{10} + \lambda_{11} = \gamma_1\} \quad (7) \end{aligned}$$

and where we choose $s_{ij} = s_{\boldsymbol{\lambda}}$ for all $\{i, j\}$ of joint type $\boldsymbol{\lambda}$. Here $\beta(i, \boldsymbol{\lambda}; k)$ is the number of vectors $\boldsymbol{v_j}$ which have a joint type $\boldsymbol{\lambda}$ with respect to $\boldsymbol{v_i}$. This is bounded as,

$$\beta(i, \boldsymbol{\lambda}; k) = \frac{\binom{k}{k\lambda_{00}, k\lambda_{01}, k\lambda_{10}, k\lambda_{11}}}{\binom{k}{k\gamma_0, k\gamma_1}} \quad \leq \quad 2^{k(H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma}))} \tag{8}$$

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

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

Since we are bounding a probability, the following bound also holds for $\rho_{\boldsymbol{\lambda}} \in [0, 1]$ and $s_{\boldsymbol{\lambda}} = \frac{1}{1+\rho_{\boldsymbol{\lambda}}}$.

$$P_{e,i} \leq \sum_{\boldsymbol{x_i} \in \mathcal{X}^n} \sum_{\boldsymbol{y} \in \mathcal{Y}^n} P^{\boldsymbol{\gamma},n}(\boldsymbol{x_i}) P_{\mathbf{Y}|\mathbf{X}}(\boldsymbol{y}|\boldsymbol{x_i}) \cdot \sum_{\boldsymbol{\lambda} \in S_i(D)} \Bigg(2^{k(H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma}))}$$
$$\cdot \sum_{\boldsymbol{x_j} \in \mathcal{X}^n} P^{\boldsymbol{\lambda},n}(\boldsymbol{x_j}|\boldsymbol{x_i}) \frac{P_{\mathbf{Y}|\mathbf{X}}(\boldsymbol{y}|\boldsymbol{x_j})^{\frac{1}{1+\rho_{\boldsymbol{\lambda}}}}}{P_{\mathbf{Y}|\mathbf{X}}(\boldsymbol{y}|\boldsymbol{x_i})^{\frac{1}{1+\rho_{\boldsymbol{\lambda}}}}} \Bigg)^{\rho_{\boldsymbol{\lambda}}} \tag{9}$$

Using the independence of the sensor outputs, the joint p.m.f.s can be simplified as below.

$$P_{e,i} \leq \sum_{\boldsymbol{\lambda} \in S_i(D)} 2^{\rho_{\boldsymbol{\lambda}} k(H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma}))}$$
$$\cdot \Bigg(\sum_{a_i \in \mathcal{X}} \sum_{b \in \mathcal{Y}} P^{\boldsymbol{\gamma}}(a_i) P_{Y|X}(b|a_i)^{\frac{1}{1+\rho_{\boldsymbol{\lambda}}}}$$
$$\cdot \Bigg(\sum_{a_j \in \mathcal{X}} P^{\boldsymbol{\lambda}}(a_j|a_i) P_{Y|X}(b|a_j)^{\frac{1}{1+\rho_{\boldsymbol{\lambda}}}} \Bigg)^{\rho_{\boldsymbol{\lambda}}} \Bigg)^n \tag{10}$$

We define the following quantity.

$$E(\rho_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) = -\log \Bigg(\sum_{a_i \in \mathcal{X}} \sum_{b \in \mathcal{Y}} P^{\boldsymbol{\gamma}}(a_i) P_{Y|X}(b|a_i)^{\frac{1}{1+\rho_{\boldsymbol{\lambda}}}}$$
$$\cdot \Bigg(\sum_{a_j \in \mathcal{X}} P^{\boldsymbol{\lambda}}(a_j|a_i) P_{Y|X}(b|a_j)^{\frac{1}{1+\rho_{\boldsymbol{\lambda}}}} \Bigg)^{\rho_{\boldsymbol{\lambda}}} \Bigg) \tag{11}$$

Since the number of types of $\boldsymbol{\lambda}$ is upper bounded by $(k+1)^4$, and $k = \lceil nR \rceil$, implying $k < nR + 1$, (10) is bounded as,

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

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,

$$P_{e,\max} \leq 2^{-n(-o_1(n) + E_r(R,D)})$$
$$E_r(R, D) = \min_{\boldsymbol{\gamma}} \min_{\boldsymbol{\lambda} \in S(D)} \max_{0 \leq \rho_{\boldsymbol{\lambda}} \leq 1} (E(\rho_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) - \rho_{\boldsymbol{\lambda}} R(H(\boldsymbol{\lambda}) – H(\boldsymbol{\gamma})))$$
$$o_1(n) = \frac{4\log(nR+2)}{n} + \frac{1}{n}\rho_{\boldsymbol{\lambda}}(H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma})) \tag{12}$$

where $\boldsymbol{\gamma} = (\gamma_0, \gamma_1)$ is over all p.m.f.s, 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 $R(H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma}))$. We write this derivative below.

$$\left.\frac{\partial E(\rho_{\boldsymbol{\lambda}}, \boldsymbol{\lambda})}{\partial \rho_{\boldsymbol{\lambda}}}\right|_{\rho_{\boldsymbol{\lambda}}=0} = \sum_{a_i \in \mathcal{X}} \sum_{b \in \mathcal{Y}} P^{\boldsymbol{\gamma}}(a_i) P_{Y|X}(b|a_i)$$
$$\cdot \frac{P^{\boldsymbol{\gamma}}(a_i) P_{Y|X}(b|a_i)}{\sum_{a_j \in \mathcal{X}} P^{\boldsymbol{\gamma}}(a_i) P^{\boldsymbol{\lambda}}(a_j|a_i) P_{Y|X}(b|a_j)}$$
$$= \sum_{a_i \in \mathcal{X}} \sum_{b \in \mathcal{Y}} P^{\boldsymbol{\gamma}}_{X_iY}(a_i, b) \frac{P^{\boldsymbol{\gamma}}_{X_iY}(a_i, b)}{Q^{\boldsymbol{\lambda}}_{X_iY}(a_i, b)} \quad = \quad D(P^{\boldsymbol{\gamma}}_{X_iY} \| Q^{\boldsymbol{\lambda}}_{X_iY})$$

Using this derivative in the analysis above, and dropping the condition $\boldsymbol{\lambda} \in \mathcal{P}_k(\{0,1\}^2)$ from the definition (7) of $S(D)$ (thus, weakening the bound), we see that the sensor network can achieve any rate R bounded as below.

$$R \leq \min_{\boldsymbol{\gamma}} \min_{\substack{\boldsymbol{\lambda} \\ \lambda_{01}+\lambda_{10}>D \\ \lambda_{00}+\lambda_{01}=\gamma_0 \\ \lambda_{10}+\lambda_{11}=\gamma_1}} \frac{D\left(P^{\boldsymbol{\gamma}}_{X_iY} \| Q^{\boldsymbol{\lambda}}_{X_iY}\right)}{H(\boldsymbol{\lambda}) - H(\boldsymbol{\gamma})} \tag{13}$$

□

IV. Capacity bound examples

We compute the capacity bound $C_{LB}(D)$ in (1) for various distortions, noise levels, and sensor resolutions. The sensor noise model assumed is that the probability of counting 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. Also, 'sensor resolution' in bits is simply $\log_2(c+1)$. In Figure 2, we demonstrate $C_{LB}(D)$ for various sensor noise levels and sensor resolutions. In all cases, $C_{LB}(D = 0) = 0$, since each sensor only has a fixed number of connections. Other obvious conclusions about the effect of noise and resolution can also be drawn.

Figure 3 shows $C_{LB}(D)$ at $D = 0.1$, as a function of sensor noise level. This figure demonstrates that the random sensor network is more efficient than a strategy of simple sensor replication, which is a popular practical method to minimize error probability. For example, for 2-bit sensors, a rate of 0.26 is achievable at noise level 0.2. 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 falls to $0.26/3 = 0.087$. For a noise level of 0.1, $C_{LB}(0.1)$ equals 0.43 for a 2-bit sensor. Thus, the bound indicates that cooperative sensor strategies are significantly more efficient than sensor replication.

V. Empirical Evaluation of Capacity

We used the belief propagation algorithm [14] to decode randomly generated sensor networks in order to empirically demonstrate a capacity effect. Though this algorithm is suboptimal for graphs with cycles such as our sensor network model, belief propagation has proven effective in decoding other graphs with cycles, such as the graphs associated with LDPC codes.

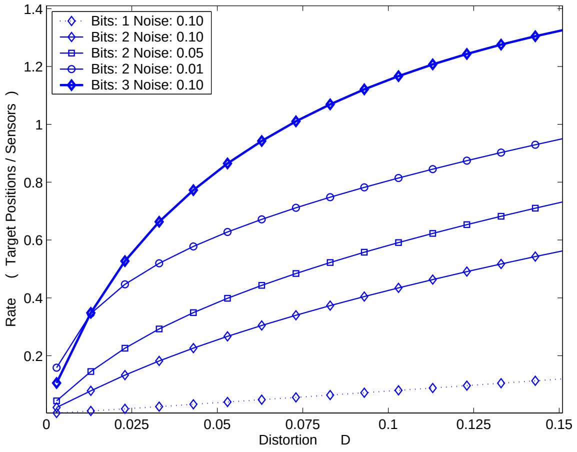

Figure 2: $C_{LB}(D)$ at various noise levels and resolutions.

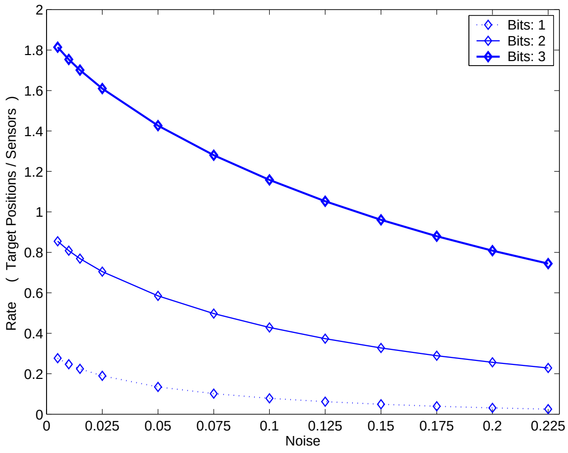

Figure 3: $C_{LB}(D = 0.1)$ as a function of sensor noise.

Borrowing from [14], 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 $\mathcal{L}(m)\backslash\ell$. The algorithm consists of two parts, where two sets of quantities, $q_{m\ell}$ and $r_{m\ell}$, are iteratively updated. For $v \in \{0, 1\}$, the quantity $q^v_{m\ell}$ is the probability that the value v_m of target m in $\boldsymbol{v}$ has the value v given the observations of all sensors except sensor ℓ. The quantity $r^v_{m\ell}$ corresponds to the probability of sensor ℓ generating the observed value y_ℓ given that target m is fixed at the value v and the other targets have a separable distribution given by the probabilities $\{q_{m'\ell} : m' \in \mathcal{M}(\ell)\backslash m\}$. Finally, let $p^0_m = P(v_m = 0)$ and $p^1_m = P(v_m = 1)$ represent the prior probabilities of the target bits of the target vector. We now proceed to describe the belief propagation algorithm for our sensor network model.

We initialize the algorithm by letting $q^0_{m\ell} = p^0_m$ and $q^1_{m\ell} = p^1_m$. In the *sensor* step of the algorithm we compute the $r_{m\ell}$ quantities using the following expressions.

$$r^0_{m\ell} = \sum_{b=0}^{c} P_{Y|X}(y_\ell|b) \cdot \sum_{\boldsymbol{v}' \in \{v_{m'}:m' \in \mathcal{M}(\ell)\backslash m\}} P(x_\ell = b | v_m = 0, \boldsymbol{v}') \prod_{m' \in \mathcal{M}(\ell)\backslash m} q^{v_{m'}}_{m'\ell}$$

$$r^1_{m\ell} = \sum_{b=0}^{c} P_{Y|X}(y_\ell|b) \cdot \sum_{\boldsymbol{v}' \in \{v_{m'}:m' \in \mathcal{M}(\ell)\backslash m\}} P(x_\ell = b | v_m = 1, \boldsymbol{v}') \prod_{m' \in \mathcal{M}(\ell)\backslash m} q^{v_{m'}}_{m'\ell}$$

The conditional probabilities of the sensor output given the target bits in the above expressions equal one if the target bits connected to sensor ℓ contain b ones, and zero otherwise. The *target* step takes the computed $r_{m\ell}$ values and uses them to compute new $q_{m\ell}$ values as shown in the expressions below.

$$q^0_{m\ell} = \alpha_{m\ell} p^0_m \prod_{\ell' \in \mathcal{L}(m)\backslash\ell} r^0_{m\ell'}$$

$$q^1_{m\ell} = \alpha_{m\ell} p^1_m \prod_{\ell' \in \mathcal{L}(m)\backslash\ell} r^1_{m\ell'}$$

$$\text{where } \alpha_{m\ell} = q^0_{m\ell} + q^1_{m\ell}$$

After a fixed number of iterations one can halt the algorithm and compute the probabilities of each target bit as shown below. These probabilities can be used to decode the target vector.

$$q^0_m = \alpha_m p^0_m \prod_{\ell \in \mathcal{L}(m)} r^0_{m\ell}$$

$$q^1_m = \alpha_m p^1_m \prod_{\ell \in \mathcal{L}(m)} r^1_{m\ell}$$

$$\text{where } \alpha_m = q^0_m + q^1_m$$

Using our decoding algorithm we empirically examined sensor network performance as a function of rate. We generated sensor networks of various rates by setting the number of targets at 200, and varying the number of sensors. We chose the number of connections per sensor to be three, the distortion level to be 0.1, and the noise level to be 0.1 ($P(Y \neq X) = 0.1$, with $\mathcal{Y} = \mathcal{X}$). As in the previous section, we assume that the probability of counting error decays exponentially with error magnitude. For each possible target vector type, $\boldsymbol{\gamma}$, we empirically evaluated the average error rate of a set of randomly generated sensor networks using belief propagation. We plotted the maximum error rate over all types for each rate value as shown in Figure 4. In addition, we plotted the average error rate over a set of randomly generated test vectors without regard to test vector types. As expected, the average error converges to zero more quickly than the maximum type error curve. The capacity value C_{LB} for the model used in this experiment is 0.434. Since belief propagation is suboptimal, and given that the error curves converge to zero at rates above 0.434, it appears that our capacity lower bound is not tight.

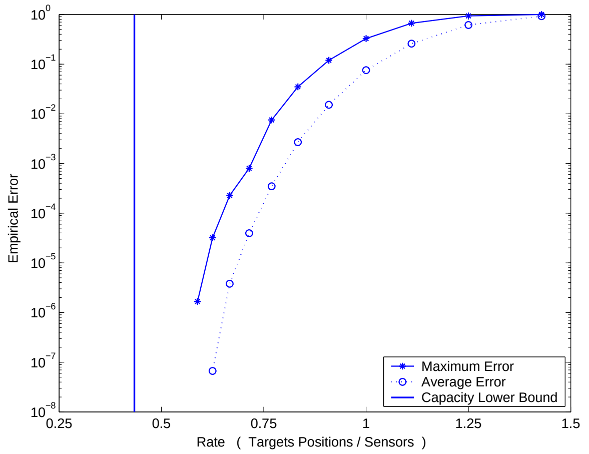

Figure 4: Maximum (over all target vector types) and average empirical error rate of belief propagation based decoding of sensor networks as a function of rate, and the corresponding lower bound on sensing capacity.

VI. Extensions of the Sensor Network Model

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 $\mathcal{A}$, and a sensor model in which a sensor can indicate either the number of occurrences of each letter of $\mathcal{A}$ that it senses, or the sum of those letters, we obtain the capacity bound below.

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

where $\boldsymbol{\gamma} = (\gamma_a, a \in \mathcal{A})$ *and* $\boldsymbol{\lambda} = (\lambda_{ab}, (a,b) \in \mathcal{A}^2)$ *are two arbitrary probability mass functions.*

The second extension considers the case of heterogenous sensors, where each class of sensor has a different output alphabet $\mathcal{Y}$ and noise model $P_{Y|X}$. Let the sensor of class l be used with a given relative frequency α_l. For such a model,

$$C(D) \geq C_{LB}(D) = \min_{\boldsymbol{\gamma}} \min_{\substack{\boldsymbol{\lambda} \\ \lambda_{01}+\lambda_{10}>D \\ \lambda_{00}+\lambda_{01}=\gamma_0 \\ \lambda_{10}+\lambda_{11}=\gamma_1}} \frac{\sum_l \alpha_l D\left(P^{\boldsymbol{\gamma},l}_{X_iY}\|Q^{\boldsymbol{\lambda},l}_{X_iY}\right)}{H(\boldsymbol{\lambda})-H(\boldsymbol{\gamma})}$$

where $\boldsymbol{\gamma} = (\gamma_0, \gamma_1)$ *and* $\boldsymbol{\lambda} = (\lambda_{00}, \lambda_{01}, \lambda_{10}, \lambda_{11})$ *are two arbitrary probability mass functions.*

VII. Conclusions

We introduced a notion of sensing capacity for discrete target detection. We proved a lower bound to this ‘sensing capacity’ (as opposed to ‘channel capacity’) and computed the bound for an illustrative example at various sensor resolutions, noise levels, and tolerable distortions. By examining this bound, we concluded that under some situations, simple sensor replication is inefficient compared to sensor cooperation. We derived a belief propagation algorithm for decoding our sensor network model, and used it to empirically evaluate capacity in order to compare this quantity to our lower bound. Future work will concentrate on generalizing the sensor network model and improving the bound.

References

[1] P. Gupta and P. R. Kumar, “The Capacity of Wireless Networks,” *IEEE Trans. Inform. Theory*, Vol. 46, No. 2, 2000.

[2] I. Wang and S. Jones, “Scalability of a Class of Wireless Sensor Networks,” *SPIE ITCom Conf. Designs Modeling Wireless Networks*, August 2001.

[3] E. J. Duarte-Melo and M. Liu, “Data-gathering wireless sensor networks: organization and capacity,” *Computer Networks: Special Issue on Wireless Sensor Networks*, Vol 43, pp. 519-537, November 2003.

[4] Z. Hu and B. Li, “Fundamental Performance Limits of Wireless Sensor Networks,” to appear in *Ad Hoc and Sensor Networks*, Yang Xiao and Yi Pan, Editors, Nova Science Publishers, 2004.

[5] S. D. Servetto, “On the Feasibility of Large-Scale Wireless Sensor Networks,” *40th Annual Allerton Conference on Communication, Control, and Computing*, 2002.

[6] A. Scaglione, S. D. Servetto, “On the interdependence of routing and data compression in multi-hop sensor networks,” *MOBICOM*, 2002

[7] Q. Fang, F. Zhao, and L. Guibas, “Counting Targets: Building and Managing Aggregates in Wireless Sensor Networks,” *Palo Alto Research Center Technical Report* P2002-10298, June 2002.

[8] D. Yang, H. Gonzalez-Banos, and L. Guibas, “Counting People in Crowds with a Real-Time Network of Image Sensors,” *ICCV*, 2003.

[9] M. C. Burl, B. C. Sisk, T. P. Vaid, N. S. Lewis, “Classification performance of carbon black-polymer composite vapor detector arrays as a function of array size and detector composition,” *Sensors and Actuators B*, Vol. 87, pp. 130-149, 2002.

[10] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, “Coding Theory Framework for Target Location in Distributed Sensor Networks,” *ITCC*, 2001.

[11] R. G. Gallager, *Information Theory and Reliable Communication*, New York: Wiley, 1968.

[12] G. Miller, and D. Burshtein, “Bounds on the maximum-likelihood decoding error probability of low-density parity-check codes,” *IEEE Trans. Inform. Theory*, pp. 2696-2710, Nov. 2001.

[13] I. Csiszar, “The Method of Types,” *IEEE Trans. Inform. Theory*, Vol. 44, No. 6, 1998.

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