Estimation of the Number of Operating Sensors in Large-Scale Sensor Networks with Mobile Access

Cristian Budianu, Shai Ben-David, and Lang Tong†
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853
{cris, shai, ltong}@ece.cornell.edu

Abstract

This paper investigates the estimation of the number of operating sensors in a sensor network in which the data collection is made by a mobile access point. We propose an estimator based on the Good-Turing estimator of the missing mass and generalize it to other related problems such as the estimation of the distribution of energy available at sensors. The estimator is analyzed using the theory of large deviations. We present closed-form bounds on the large deviation exponent and characterize confidence intervals for the estimator.

Index Terms

sensor networks, sensor life-time estimation, nonparametric estimation, large deviation analysis.

EDICS: 2-ESTM (Estimation Theory and Applications), 2-PERF (Statistical Performance Analysis and Error Bounds), 3-CNET (Communication Systems and Networks).

†Corresponding author

This work was supported in part by the Multidisciplinary University Research Initiative (MURI) under the Office of Naval Research Contract N00014-00-1-0564, and Army Research Laboratory CTA on Communication and Networks under Grant DAAD19-01-2-0011. Part of this work was presented at the Asilomar Conference in Oct 2003 and ICASSP in May 2004.

I. INTRODUCTION

A. The Problem and Operation Setup

The large scale sensor networks may be used in military and civil applications to retrieve information from large and possibly unaccessible areas. In such networks the number of operating sensors can vary with time due to battery consumption and external factors. In many cases, the density of operating sensors may affect significantly some network operations, such as routing, medium access, and also information retrieval and processing. Thus knowing the number of operating nodes of a sensor network is crucial to network operation as well as network maintenance. For instance, an accurate estimation of operating sensors facilitates the decision to deploy new sensors.

Besides the number of operating sensors, one may also be interested in other quantities such as the distribution of the energy available to each sensor. Such information allows the estimation of the life-time of the sensor network and adjusting transmission strategies accordingly. A closely related problem is to estimate the number of sensors that have certain attributes, say having temperature measurements in an interval of interest $[T_1,\ T_2]$. This also generalizes to the problem of estimating class histogram of sensors observing different quantities.

We consider the problem of estimating operating sensors under a special sensor network architecture: SEnsor Network with Mobile Access (SENMA) [1]. A key feature of SENMA is the presence of the mobile access points (APs) that have high processing power and act as mobile base stations for sensors, see Fig. 1. In SENMA, the sensors may transmit the collected data to the mobile access points in the form of packets, and each packet may contain the ID of the transmitting sensor. If a class histogram is required, then the data packets will include the quantity of interest, besides the sensor ID. In Fig. 1, for example, each packet has the field "EL" (energy level). We assume a random access protocol, such as slotted ALOHA [2], by which packets collected by the mobile APs are as if they were drawn randomly from the sensor field.

A simple approach is to schedule the transmission of each sensor and count the number of sensors observed. Besides the complications of scheduling, this method requires a thorough observation of the network, at least on the order $\mathcal{O}(N)$ of the number of operating sensors. For wireless sensor networks with unreliable links, packet loss is inevitable, and retransmissions are necessary. For sensor networks with energy constraints, such a brute force approach does not scale well with the size of the network. If a random collection is used, and N is estimated by counting the number of distinct sensors observed, a

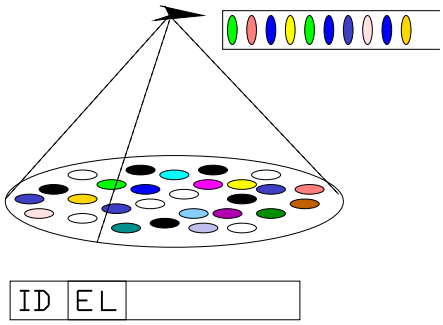

Fig. 1. The Sensor Network with Mobile Access Point; the packet structure contains an ID field and an EL "Energy Level" field.

higher number of transmissions, on the order of $\mathcal{O}(N \log N)$, is required[1]. What is needed is a technique that provides an accurate estimate but requires far less number of transmissions.

B. Summary of Results

We estimate the number of operating sensors by exploiting statistical properties of the observed data. We propose an estimator based on the Good-Turing estimator of the missing mass—a technique invented by Turing in the second world war while trying to break the Enigma code. Within the context of packet transmission in SENMA, the so-called missing mass is the conditional probability that, given a vector of observed sensor IDs (vector sample), the newly received packet comes from a new sensor. The Good-Turing nonparametric estimator [3] is the best known estimator for the missing mass of a random sample. By assuming that the samples are independent, identically distributed (i.i.d.) with uniform distribution, we express the population size as a function of the missing mass and derive an estimator for the number of operating sensors. This estimator has a simple expression and achieves a performance similar to that of the maximum likelihood (ML) estimator. It can be applied further to the estimation of class histograms; two ways for estimating fixed and time-varying class histograms are proposed and investigated through simulations.

The performance of the estimator is analyzed using the theory of large deviations for occupancy models developed recently by Dupuis, Nuzman and Whiting in [4]. This approach provides a characterization of the asymptotic behavior of the estimator proposed, when the number N of operating sensors is very large and the size of the vector sample increases proportional with N with a fixed ratio β. The large deviations

[1]This corresponds to the classical coupon collection problem.

exponent, however, can only be obtained by solving a numerical optimization problem. For a convenient performance evaluation and gaining insights into the convergence behavior, we provide closed-form upper and lower bounds, and use them to derive approximative confidence intervals for the relative error of the estimator. The simulations show that the confidence intervals derived are accurate approximations for the performance of the estimator.

C. Related work and organization

The Good-Turing nonparametric estimator, described first by Good [3], can be used to estimate probability distributions from data samples, but it works the best for estimating the probability of those elements that did not appear, *i.e.,* the missing mass. The application of Good-Turing algorithm is quite broad [5], but the use of Good-Turing algorithm for estimating the number of operating sensors was first presented in [6] with an abbreviated analysis in [7].

The analysis of the Good-Turing algorithm is much more challenging. The recent work by Dupuis, Nuzman, and Whiting [4] on the large deviation principle for the general occupancy problem forms the basis of our analysis of the proposed estimator. One can also pursue the alternative framework developed by McAllester and Schapire [8] in which confidence intervals for the Good-Turing nonparametric estimator are derived. For estimating the missing mass, the width of an ε-confidence interval is upper bounded by $\frac{2}{n} + 2\log\left(\frac{3n}{\varepsilon}\right)\sqrt{\frac{2\log(3/\varepsilon)}{n}}$, ($n$ being the number of available samples), which gives a bound on the convergence rate of the estimator. While this bound is general and applicable to cases when the received samples are not i.i.d., it can't be applied straightforward to the accuracy of the resulting approximation of N, especially when the sample size is much smaller than N.

In [9], Esty compared the asymptotic performance of the Good-Turing estimator of the missing mass and that of the ML parametric estimator. It was shown (using combinatorics) that the difference between their asymptotic performance was small. This suggests that, for the i.i.d. model, the Good-Turing estimator can be used to derive an estimator of the total number of sensors with "good" asymptotic properties. However, the asymptotic results of [9] obtained for the Good-Turing estimator of the missing mass cannot be extended directly to our problem. In subsection III-C we will justify why the results of [8] and [9] do not apply directly to our problem.

Finally, we should mention that in our paper we only use the Good-Turing estimator in its simplest variant. Many improvements have been proposed. See, for example, the paper by Orlitsky, Santhanam, and Zhang [5] where smoothed variants of the Good-Turing estimator are used for analysis.

The paper is organized as follows. The model and the motivation behind our approach are discussed

in Section II. The Good-Turing based estimator and its application to class histogram estimation are presented in Section III. The performance analysis based on the large deviations theory is presented in Section IV. The simulations and numerical results are presented in Section V. We conclude the paper in Section VI. Some proofs are deferred to the Appendix.

II. The Model

The sensor network considered has N operating sensors, each having an ID that is an element of set $\mathcal{N}$, with $|\mathcal{N}| = N$. The mobile AP collects n packets, each of them containing the ID of the transmitting sensor. We denote by $X_i \in \mathcal{N}$ the ID in the i-th received packet and by $\mathbf{X} \stackrel{\Delta}{=} (X_1, \ldots, X_n)$ the vector sample of received IDs. The unknown N is assumed to remain constant during collection.

For SENMA using a random access protocol such as ALOHA, packet collection can be modeled as an i.i.d. sampling with uniform distribution, *i.e.,* in each time slot the received packet can be from any of the sensors with equal probability:

$$\forall\ x \in \mathcal{N}\ :\ p_x \stackrel{\Delta}{=} \mathbb{P}[X_i = x] = \frac{1}{N}. \tag{1}$$

This model is identical to an urn model with replacement. Note that some of the received packets may come from the same sensor.

In a practical setup, the i.i.d. sampling assumption can be justified as follows. First, the access point (AP) broadcasts a request for the information needed - *i.e.,* want to know how many sensors are functional. Each slot, each operating sensor flips a coin to decide on transmiting a packet in that slot. The probability of transmission is the same for all sensors and is kept low enough to avoid collisions (with high probability). Also, we assume that the probability of AP detecting correctly a transmitted packet is the same for all sensors. These conditions are reasonable for sensor networks because the data is transmitted at very low data rates, and justify the i.i.d. sampling assumption.

For convenience, we introduce here some notations that will be used later. The vector sample may contain multiple packets from the same sensor. Therefore, we denote $\mathcal{S}$ as the set of received (distinct) IDs

$$\mathcal{S} \stackrel{\Delta}{=} \{x \in \mathcal{N} : \exists k \in \{1, \ldots, n\}, X_k = x\}$$

whose size $S = |\mathcal{S}|$ represents the total number of (different) sensors observed. For the observed vector sample $\mathbf{X}$, define the multiplicity function $t_{\mathbf{X}} : \mathcal{N} \to \mathbb{N}$, where $t_{\mathbf{X}}(x)$ gives the number of samples in $\mathbf{X}$ equal to x. Using the function $t_{\mathbf{X}}$, we partition $\mathcal{S}$ according to the number of times an ID appears by

denoting

$$\mathcal{S}_k \stackrel{\Delta}{=} \{x \in \mathcal{N} : t_{\mathbf{X}}(x) = k\} \quad , \quad \forall k = 0, \ldots, n.$$

Note that the sets $\mathcal{S}_k$ depend on the observed sample $\mathbf{X}$, thus they are random. We use the notation $S_k \stackrel{\Delta}{=} |\mathcal{S}_k|$ for the sizes of the sets defined. Therefore, $S_0 \stackrel{\Delta}{=} N - S$ represents the number of operating sensors that are not in the current vector sample. The problem is to estimate N from $\mathbf{X}$. Since S is known, this is equivalent to estimating S_0, the number of hidden operating sensors.

Slightly more complicated is the *class partition model.* Suppose that the set of sensors is partitioned into classes, and each sensor transmits in each packet its class index (besides its ID). For example, each class can contain those sensors with available energy in a specific interval; we want to estimate the number of (operating) sensors in each class. Assume that the class of each sensor is fixed during data acquisition, and let C denote the total number of classes and $\phi(x)$ the class of sensor x. Denote by $N(\varphi)$ the number of sensors that belong to class φ. For each class $\varphi = 1, \ldots, C$, we define $\mathbf{X}(\varphi)$ as a vector made of those elements of $\mathbf{X}$ belonging to class φ, and the corresponding $\mathcal{S}(\varphi)$, $S(\varphi)$:

$$\begin{aligned} \mathcal{S}(\varphi) &\stackrel{\Delta}{=} \{x \in \mathcal{N} : \exists k \in \{1, \ldots, n\} \text{ s.t. } X_k = x,\ \phi(X_k) = \varphi\} \\ S(\varphi) &\stackrel{\Delta}{=} |\mathcal{S}(\varphi)|. \end{aligned}$$

We make the extra assumption that $C \ll \min(n, N)$ so that, with high probability, $S(\varphi) \gg 1$, $\forall \varphi$. This assures that all classes appear in the current vector sample so that we don't need to estimate the total number of existing classes. In this setup, the problem is to estimate how many sensors are in each class, *i.e.,* the histogram vector $\mathbf{N} = (N(1), \ldots, N(C))$ (Fig. 2). In Fig. 2, in each bar, the lower part represents the observed percentage of the nodes in each class and the upper part the hidden one. The leftmost bar represents the percentage of sensors that are not operating, and thus are assumed to have zero energy available.

In the same way as above, for each k and φ, the set $\mathcal{S}_k(\varphi)$ contains those elements that belong to class φ and appear in $\mathbf{X}$ exactly k times

$$\mathcal{S}_k(\varphi) \stackrel{\Delta}{=} \{x \in \mathcal{N} : t_{\mathbf{X}}(x) = k,\ \phi(x) = \varphi\} \quad , \quad \forall k = 0, \ldots, n, \quad \forall \varphi = 1, \ldots, C,$$

and $S_k(\varphi) \stackrel{\Delta}{=} |\mathcal{S}_k(\varphi)|$.

If the class of each sensor is determined by its available battery energy, this model is realistic only if the battery consumption during the collection of the sample vector $\mathbf{X}$ is negligible so that the class of each sensor can be considered fixed and battery levels do not affect the transmission probabilities. A

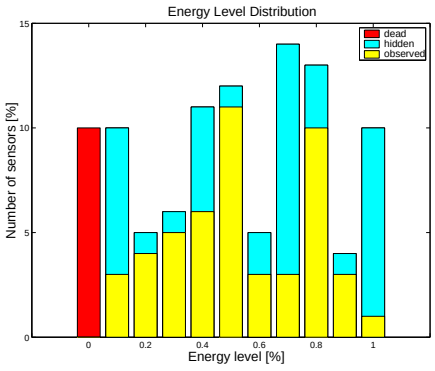

Fig. 2. The histogram of energy available to sensors. The sensors that are not operating are considered to have 0 energy available (the first bar). The figure is illustrative (is not the result of a simulation).

more general model in which the class of each sensor can vary during data collection will be introduced in subsection III-D.

III. Estimation of the Number of Operating Sensors and Class Histograms

A. Motivation: The Maximum Likelihood Estimator

The Maximum Likelihood (ML) estimator would be a natural choice. Because the observation space $\mathcal{N}$ is not known in advance, the ML estimator must be based on the vector $[S_1, \ldots, S_n]$ (see *e.g.*,[10]). Denote by $\{Y_1, \ldots Y_S\}$ the S distinct elements of $\mathcal{N}$ that appear in the vector sample $\mathbf{X}$. We have

$$\begin{aligned}\mathbb{P}[[S_1, \ldots, S_n]|N] &= \frac{\binom{N}{S_1\ S_2 \ldots S_n}}{N^n} \binom{n}{t_{\mathbf{X}}(Y_1)\ t_{\mathbf{X}}(Y_2) \cdots t_{\mathbf{X}}(Y_S)} \\ &= \frac{N!}{S_1! S_2! \ldots S_n! (N-S)! N^n} \binom{n}{t_{\mathbf{X}}(Y_1)\ t_{\mathbf{X}}(Y_2) \cdots t_{\mathbf{X}}(Y_S)},\end{aligned}$$

which gives

$$\hat{N}_{\text{ML}} = \arg\max_{N \geq S} \mathbb{P}[[S_1, \ldots, S_n]|N] = \arg\max_{N \geq S} \frac{N!}{N^n\ (N-S)!}. \tag{2}$$

The above optimization does not have a closed form solution. To perform the search, we need an upper bound on $\hat{N}_{\text{ML}}$. Taking the derivative with respect to N of the logarithm of $\frac{N!}{N^n\ (N-S)!}$, we observe that the derivative is negative for $N > \frac{nS}{n-S}$, which gives an upper bound. If $n = S$ then each sample in the collection is new and thus $N \gg n$. Therefore, the optimization problem above can be solved numerically and, in this case, the solution can be obtained easily.

For histogram estimation, the ML solution based on the vectors $[S_1(\varphi), \ldots, S_n(\varphi)],\ \varphi = 1, \ldots, C$, is given by

$$\hat{\mathbf{N}}_{ML} = \arg\max_{\mathbf{N}} \frac{1}{N^n} \prod_{\varphi=1}^{C} \frac{N(\varphi)!}{(N(\varphi) - S(\varphi))!}.$$

This problem requires a C-dimensional search, which is considerably harder than estimating N only. Moreover, in both cases, besides the difficulty associated with the numerical evaluation, the ML solutions give little insight into the problem, since the analysis is hard, if not impossible.

We note that the proposed estimator of the number of sensors based on the Good-Turing estimator of the missing mass has the performance close to that of the ML estimator, but its simple formula is easy to implement and analyze.

B. Background: the Good-Turing estimator

Consider a finite or countable set $\mathcal{N}$, a probability distribution P on this set, and a sample $\mathbf{X} = (X_1, \ldots, X_n)$, where $X_i \in \mathcal{N}$ are i.i.d random variables with distribution P. As before, for $x \in \mathcal{N}$, denote $p_x \stackrel{\Delta}{=} \mathbb{P}[X_i = x]$. Note that P need not be uniform nor $\mathcal{N}$ finite.

Recall that, for each k, the set $\mathcal{S}_k$ is composed of all the elements of $\mathcal{N}$ that appear in the vector sample $\mathbf{X}$ exactly k times. Now we define P_k to be the probability that the next sample, drawn (i.i.d.) with distribution P, belongs to set $\mathcal{S}_k$

$$P_k \stackrel{\Delta}{=} \sum_{x \in \mathcal{S}_k} p_x = \mathbb{P}[X_{n+1} \in \mathcal{S}_k | \mathbf{X}] = \mathbb{P}[t_{\mathbf{X}}(X_{n+1}) = k | \mathbf{X}].$$

For $k = 0$, P_0 is the probability that the next observed sample X_{n+1} is new, *i.e.*, $X_{n+1} \in \mathcal{S}_0$. The probability P_0 is called the missing mass and $1 - P_0$ the coverage of the sample $\mathbf{X}$. The probabilities P_k depend on the sample $\mathbf{X}$, thus they are random variables.

The following estimator for the missing mass, known as the Good-Turing estimator, was proposed in [3]

$$\hat{P_0} = \frac{S_1}{n}. \tag{3}$$

The missing mass is estimated using the number of elements that appear in the sample exactly once. Some intuition about the Good-Turing estimator is given by its behavior in some extreme situations. First, if all n elements of $\mathbf{X}$ are different, this means that the samples are drawn from a very large collection, and it is likely that the next sample will be new as well. The estimator gives $\hat{P_0} = 1$. On the other hand, if all elements of $\mathbf{X}$ appear at least twice, this suggests that the collection is complete. The estimator gives $\hat{P_0} = 0$.

C. Estimation of the Number of Operating Sensors and Class Histograms

The Good-Turing estimator can be used to estimate the number of operating sensors. Under the uniform distribution assumption, the missing mass is given by

$$P_0 = 1 - \frac{S}{N}.$$

Using the estimated value of P_0 in (3), we have the following estimator for N:

$$\hat{N} = \frac{S}{1 - \hat{P}_0} = \frac{S}{1 - \frac{S_1}{n}}. \tag{4}$$

The relative error of this estimator can be written as

$$\left| \frac{\hat{N} - N}{N} \right| = \frac{|P_0 - \hat{P}_0|}{1 - \frac{S_1}{n}}.$$

This formula explains why the results obtained for the performance of the Good-Turing estimator (the numerator) can't be applied directly in our case. The results of [8] and [9] are only for the numerator of the relative error on N. In our case, the denominator is less than one and can take very low values for small samples, resulting in an increase of the error. Moreover, there is no guarantee that the numerator and denominator are independent. A study of the properties of the denominator and of the correlation between the two terms would be needed.

The estimation approach can be used further for estimation of class histograms. For each class φ, its missing mass is the probability that the next sample is new *and* it belongs to class φ

$$P_0(\varphi) \stackrel{\Delta}{=} \mathbb{P}\{X_{n+1} \notin \mathbb{S},\ \phi(X_{n+1}) = \varphi \ |\mathbf{X}\}. \tag{5}$$

The Good-Turing estimator can be used to estimate the missing mass for each class separately

$$\hat{P}_0(\varphi) = \frac{S_1(\varphi)}{n}.$$

The formula above can be justified as follows. We label all those elements of $\mathcal{N}$ that are not in class φ with a new ID y, and, consequently, the new space of IDs $\mathcal{N}'_\varphi$ is given by :

$$\mathcal{N}'_\varphi \stackrel{\Delta}{=} \{y\} \cup \{x \in \mathcal{N} : \phi(x) = \varphi\}.$$

If the vector sample $\mathbf{X}$ contains at least two elements that do not belong to class φ, then, taking into account the relabeling, the number of elements that appear in $\mathbf{X}$ only once is equal to $S_1(\varphi)$, and the formula given before follows.

To estimate the number of sensors in each class, $N(\varphi) = S(\varphi) + S_0(\varphi)$, we use the relation $P_0(\varphi) = \frac{S_0(\varphi)}{N}$, and the estimates $\hat{N}$ and $\hat{P}_0(\varphi)$ obtained previously to get $\hat{S}_0(\varphi) = \hat{P}_0(\varphi)\hat{N}$, and further, $\hat{N}(\varphi) = S(\varphi) + \hat{S}_0(\varphi)$. We obtain the estimator $\hat{N}_{GT}$ (Good-Turing):

$$\hat{N}_{GT}(\varphi) = S(\varphi) + S_1(\varphi)\frac{S}{n - S_1}. \tag{6}$$

A different estimator can be obtained by assuming a histogram scaling law:

$$\frac{S(\varphi)}{S} \approx \frac{N(\varphi)}{N}.$$

In other words, it is assumed that the proportion of elements from a class that appear in the vector sample $\mathbf{X}$ is the same as the proportion of the elements of that class in set $\mathcal{N}$. Substituting the unknown quantities by the estimates, we have

$$\frac{S(\varphi)}{S} = \frac{\hat{N}(\varphi)}{\hat{N}}.$$

This gives the following estimator denoted $\hat{N}_{HS}(\varphi)$ (histogram scaling):

$$\hat{N}_{HS}(\varphi) = S(\varphi) + S(\varphi)\frac{S_1}{n - S_1} = \frac{S(\varphi)}{1 - \frac{S_1}{n}}. \tag{7}$$

We'll see in the simulations section that the histogram scaling estimator has slightly better performance than the one derived previously by applying the Good-Turing formula to estimate $P_0(\varphi)$. For both estimators, it can be verified easily that $\sum_\varphi \hat{N}(\varphi) = \hat{N}$.

A third possibility would be to estimate $N(\varphi)$ for each φ by using formula (4) for the elements of the respective class φ (and thus ignoring all elements of $\mathbf{X}$ that are not in class φ). It can be checked that the performance of this estimator is much worse than the other two given in the current section.

D. Estimation of Time-Varying Class Histograms

In this section we modify the estimators proposed before so that they can be used for the estimation of time-varying class histograms. In the setup considered, the class of each sensor can change during the collection time. It is assumed, however, that the uniform distribution of received IDs (1) still holds, *i.e.*, it is not modified by the class changing process. This assumption is suitable if the class of each sensor is determined by a quantity measured by the sensor. However, for estimation of the available battery energy, the assumption holds only if the sensors do not modify their transmission strategy as a function of the available energy; also, the collection time is assumed short enough so that the variation of the number of operating sensors can be neglected.

If the classes of sensors change in time, we use an upper index to show the variation in time of the quantities involved. For example, denote by $\phi^{(i)}(x)$ the class of sensor x in time slot i. The class of each sensor is fixed during one time slot, and the packet received in one slot i contains the class of the transmitting sensor in the current slot i. The way in which the class of each sensor changes in time is assumed known. This means that for each of the sensors observed, $\forall x \in \mathcal{S}$ and $\forall \varphi \in \{1, \ldots, C\}$ we know

$$\mathbb{P}\{\phi^{(n+1)}(x) = \varphi | \mathbf{X}, \Phi\}.$$

The conditional probabilities given above depend on the system characteristics. Some examples are discussed later in this section and in Section V-A.

For each class φ we want to estimate $N^{(n+1)}(\varphi)$, the number of sensors in class φ in time slot $n+1$. Throughout this section it is assumed that the estimation is done based on the vector sample $\mathbf{X}$ with n elements and on the corresponding vector of received states $\Phi \stackrel{\Delta}{=} (\phi^{(1)}(X_1), \phi^{(2)}(X_2), \ldots, \phi^{(n)}(X_n))$.

In the case of time-varying classes, the formula (5) of the missing mass of class φ becomes

$$P_0^{(n+1)}(\varphi) \stackrel{\Delta}{=} \mathbb{P}\{X_{n+1} \notin \mathcal{S},\ \phi^{(n+1)}(X_{n+1}) = \varphi\ |\mathbf{X}\}. \tag{8}$$

The assumption that the uniform distribution of received IDs is fixed simplifies the problem greatly: one could use the previous methods by substituting the fixed sets $\mathcal{S}_k(\varphi)$ with the corresponding sets at time $n+1$, *i.e.,* with $\mathcal{S}_k^{(n+1)}(\varphi)$, defined as

$$\mathcal{S}_k^{(n+1)}(\varphi) \stackrel{\Delta}{=} \left\{ x \in \mathcal{N} : t_{\mathbf{X}}(x) = k,\ \phi^{(n+1)}(x) = \varphi \right\} \quad , \quad \forall k = 0, \ldots, n, \quad \forall \varphi = 1, \ldots, C.$$

Note that $\mathcal{S}_k^{(n+1)}(\varphi)$ is a subset of $\mathcal{S}$, *i.e.,* is composed of elements that appeared in sample $\mathbf{X}$ (that has n elements). However, since one element $x \in \mathcal{N}$ may have different classes in different time slots, in general the class of each $x \in \mathcal{S}$ is unknown at time instant $n+1$, thus the sets $\mathcal{S}_k^{(n+1)}(\varphi)$ are also unknown.

Our approach is to use the estimators (6) and (7) given in the previous section by substituting the quantities $S(\varphi)$ and $S_1(\varphi)$ with estimates of $S^{(n+1)}(\varphi)$ and $S_1^{(n+1)}(\varphi)$, respectively. We use:

$$\hat{S}^{(n+1)}(\varphi) \stackrel{\Delta}{=} \mathbb{E}[S^{(n+1)}(\varphi)|\mathbf{X}, \Phi] = \sum_{x \in \mathcal{S}} \mathbb{P}\{\phi^{(n+1)}(x) = \varphi | \mathbf{X}, \Phi\}, \tag{9}$$

$$\hat{S}_1^{(n+1)}(\varphi) \stackrel{\Delta}{=} \mathbb{E}[S_1^{(n+1)}(\varphi)|\mathbf{X}, \Phi] = \sum_{x \in \mathcal{S}} \mathbf{1}_{\{t_{\mathbf{X}}(x)=1\}} \mathbb{P}\{\phi^{(n+1)}(x) = \varphi | \mathbf{X}, \Phi\}. \tag{10}$$

Since the expressions above are rather general, we will see how they apply in a couple of special situations.

First, if for each $x \in \mathcal{S}$ the class $\phi^{(n+1)}(x)$ can be determined exactly, *i.e.*, $\phi^{(n+1)}(x) = f(x; \mathbf{X}, \Phi)$, then the sets $\mathcal{S}_k^{(n+1)}(\varphi)$ can be determined exactly and the estimator $\hat{S}^{(n+1)}(\varphi)$ gives the exact value of $S^{(n+1)}(\varphi)$.

A second special case is when the class of each sensor changes independently of the class of other sensors. If for each sensor the class evolution process is Markov, then the probabilities $\mathbb{P}\{\phi^{(n+1)}(x) = \varphi | \mathbf{X}, \Phi\}$ used in (9) and (10) depend only on the time slot of last apparition of x in $\mathbf{X}$, $i_{max}(x) \stackrel{\Delta}{=} \max\{i : X_i = x\}$, and the class of x at that moment, $\phi^{(i_{max}(x))}(x)$:

$$\mathbb{P}\{\phi^{(n+1)}(x) = \varphi | \mathbf{X}, \Phi\} = \mathbb{P}\{\phi^{(n+1)}(x) = \varphi | i_{max}(x),\ \phi^{(i_{max}(x))}(x)\}.$$

The last expression can be determined using the transition matrix of the Markov chain. Such an example is analyzed in the simulations section.

A final comment is about the convergence properties of the two estimators (9,10). If the number of classes is finite and the class of each sensor changes independently of the class of other sensors, and if the collection is done such that $S \to \infty$, and $S_1 \to \infty$ respectively, then the convergence of the two estimators can be analyzed using a form of the strong law of large numbers [11, Corollary 7.4.1, p. 214]. If the variance of the quantities to be estimated goes to infinity, the Lindeberg-Feller central limit theorem [11, Theorem 9.8.1, p. 315] can be applied to derive more precise results.

IV. A Performance Analysis based on the Theory of Large Deviations

In this section a performance analysis of the Good-Turing-based estimator for i.i.d. sampling model is done. The analysis is based on the theory of large deviations for occupancy problems developed by Dupuis, Nuzman and Whiting in [4]. We use confidence intervals for the relative error to characterize the performance of the estimator. Choosing an interval $(c_1, c_2),\ \ c_1 < 1 < c_2$, we investigate the variation of the probability that the ratio $\frac{\hat{N}}{N}$ falls outside this interval with the number of sensors N and the number of samples n. Our analysis is targeted to the case in which the number of samples is relatively small compared to the total number of sensors, *i.e.*, , for the ratio n/N subunitary.

The large deviations approach is motivated by the complications associated with the analysis of exact combinatorial expressions. Furthermore, thanks to the large number of sensors, the asymptotic results can predict the performance of the system.

A. Large Deviations Asymptotics for Occupancy Problems - The framework of [4]

Our presentation starts with a short overview of the framework and results of [4]. In [4] the occupancy problem is explained using urns and balls. The number of operating sensors corresponds to the number

of available urns, and the number of available samples to the number of balls. There are n balls that are thrown (one by one) in N urns, each ball falling in any of the urns with equal probability. The number of balls in an urn corresponds to the number of packets received from a sensor.

Introduce a constant β, and consider $n = \lfloor \beta N \rfloor$. Fix an integer $I > 0$, and for $i = 0, \ldots, I$ denote by

$$\Gamma_i^N \triangleq \frac{S_i}{N}$$

the fraction of urns that contain exactly i balls (or sensors that appear i times in the current sample), and by

$$\Gamma_{I+}^N \triangleq 1 - \sum_{i=0}^{I} \Gamma_i^N$$

the fraction of urns that contain more than I balls (or sensors that appear more than I times in the current sample). Thus

$$\Gamma^N \triangleq [\Gamma_0^N,\ \Gamma_1^N, \ldots, \Gamma_I^N, \Gamma_{I+}^N]$$

is a random probability vector that specifies the occupancy of the urns after $\lfloor \beta N \rfloor$ balls have been thrown. The vector Γ^N takes values in the space of probability vectors on $I+2$ points

$$\Omega_I \triangleq \left\{ \gamma \in \mathbb{R}^{I+2} :\ \gamma_j \geq 0\ \forall j,\ 0 \leq j \leq I+1;\ \sum_{j=0}^{I+1} \gamma_j = 1 \right\}.$$

The behavior of the random vector Γ^N depends on the initial conditions, *i.e.*, the initial distribution of balls in urns. Empty initial conditions means that all urns are initially empty, *i.e.*, $\{\Gamma_0^N(0) = 1, \Gamma_i^N(0) = 0, \Gamma_{I+}^N(0) = 0\}$.

The large deviations theory for occupancy problems characterizes the behavior of the random vector Γ^N when $N \to \infty$ while β is constant and $n = \lfloor \beta N \rfloor$. A large deviation principle (LDP) for the random vector Γ^N is stated by [4, Corollary 2.3]. Furthermore, for the case of empty initial conditions, [4, Theorem 2.5] gives the rate function in a convenient form. To present the large deviation principle we introduce first some more notations.

For each discrete probability distribution $\omega \in \Omega_I$, define the set $F(\beta, \omega)$ to be the set of all discrete distributions γ on the non-negative integers satisfying $\gamma_i = \omega_i$ for $i = 0, \ldots, I$ and the constraint (conservation)

$$\sum_{i=0}^{\infty} i\gamma_i = \beta. \tag{11}$$

Note that the distributions in $F(\beta, \omega)$ are not restricted to Ω_I; they must agree with ω on the first $I+1$ points and the other values are free. The condition for feasibility of (β, ω) is $\sum_{i=0}^{I} i\omega_i + (I+1)\omega_{I+} \leq \beta$,

with $\omega_{I+} \stackrel{\Delta}{=} 1 - \sum_{i=0}^{I} \omega_i$. In the sequel, the notation $D(P||Q)$ denotes the Kullback-Leibler distance between two distributions and $\mathcal{P}_\beta$ is the Poisson distribution with parameter β *i.e.*, $\mathcal{P}_\beta(i) = \exp(-\beta)\frac{\beta^i}{i!}$.

The following theorem states the large deviation principle for the sequence of random probability vectors Γ^N.

Theorem 1: [4, Corollary 2.3 and Theorem 2.5]
The sequence of random vectors Γ^N satisfies the large deviations principle with rate function

$$\mathcal{J}(\beta,\omega) = \begin{cases} \min_{\gamma \in F(\beta,\omega)} D(\gamma||\mathcal{P}_\beta), & \text{if } (\beta,\omega) \text{ is feasible} \\ \infty & \text{otherwise} \end{cases}$$

The minimizing argument $\gamma^* \in F(\beta,\omega)$ is unique. In particular, for any set $A \subset \Omega_I$ that is the closure of its interior we have

$$\lim_{N\to\infty} \frac{1}{N} \log \mathbb{P}\{\Gamma^N \in A\} = -\inf\{\mathcal{J}(\beta,\omega) : \omega \in A\}.$$

□

B. The Large Deviation Principle for Estimation of the Number of Operating Sensors

The large deviation principle presented earlier can be applied to the estimator (4) if we specify the constant I and the optimization domain A that appear in Theorem 1. The relative error of estimator (4) can be written as

$$\frac{\hat{N}}{N} = \frac{S}{N}\frac{1}{1-\frac{S_1}{N}\frac{N}{n}} = \frac{1-\Gamma_0^N}{1-\Gamma_1^N\frac{1}{\beta}}.$$

The reader might note that the last equality holds only for rational β and the corresponding pairs (n, N); a rigurous statement is easy to justify and would just bring some unnecessary complications.

We are interested in the asymptotic behavior of the probability of the following events when N is large

$$\left\{\frac{\hat{N}}{N} \geq c > 1\right\}, \quad \left\{\frac{\hat{N}}{N} \leq c < 1\right\}.$$

Since only Γ_0^N and Γ_1^N appear in the formula of relative error, we have $I = 1$. The optimization domains denoted by A in Theorem 1 will be denoted by $A(\beta, c)$; as before, we have $A(\beta, c) \subset \Omega_1$, but in addition to the feasibility condition, all distributions in $A(\beta, c)$ satisfy a condition imposed on the performance

bound. If $c > 1$, $A(\beta, c)$ is given by the distributions $\gamma \in \mathbb{R}^3$ which satisfy

$$\gamma_0 + \gamma_1 + \gamma_{1+} = 1 \tag{12}$$

$$\gamma_1 + 2\gamma_{1+} \leq \beta \tag{13}$$

$$\frac{1-\gamma_0}{1-\gamma_1\frac{1}{\beta}} \geq c > 1. \tag{14}$$

If $c < 1$, $A(\beta, c)$ is given by the distributions $\gamma \in \mathbb{R}^3$ which satisfy (12), (13) and

$$\frac{1-\gamma_0}{1-\gamma_1\frac{1}{\beta}} \leq c < 1. \tag{15}$$

We introduce the notation $J(\beta, c)$ for the large deviations exponent of Theorem 1, and express it in a more convenient form :

$$J(\beta, c) \stackrel{\Delta}{=} \inf\{\mathcal{J}(\beta, \omega) \ : \ \omega \in A(\beta, c)\} = \min\{D(\gamma||\mathcal{P}_\beta) \ : \ \gamma \in \mathcal{F}(\beta, c)\}, \tag{16}$$

with $\mathcal{F}(\beta, c) \stackrel{\Delta}{=} \bigcup_{\omega \in A(\beta,c)} F(\beta, \omega)$ the set of discrete probability distributions on non-negative integers, that satisfy the performance bounds conditions (14) and (15) for $c > 1$ and $c < 1$ respectively, and the conservation condition (11):

$$\mathcal{F} = \left\{\gamma : \sum \gamma_i = 1, \quad \gamma_1 \geq \gamma_0\frac{\beta}{c} - \beta\frac{1-c}{c}, \quad \sum_{i=0}^{\infty} i\gamma_i = \beta\right\}. \tag{17}$$

The solution of the optimization problem (16) can be found using Lagrange multipliers [12]. For convenient evaluations, closed form lower and upper bounds for the function $J(\beta, c)$ are obtained in the next section. The cases $c > 1$ and $c < 1$ will be treated separately.

C. Bounds on the large deviations exponent $J(\beta, c)$

As written before, the function $J(\beta, c)$ is given by the minimization problem

$$J(\beta, c) = \min_{\gamma \in \mathcal{F}(\beta,c)} \{D(\gamma||\mathcal{P}_\beta)\}, \tag{18}$$

where the domain $\mathcal{F}(\beta, c)$ is given by the distributions γ over non-negative integers that satisfy the conservation constraint (11) and the bounds on the relative error of the estimator (14) and (15) for $c > 1$ and $c < 1$ respectively. Using the properties of the optimization region $\mathcal{F}(\beta, c)$, we derive upper and lower bounds for the exponent, *i.e.,* determine $D^*(\beta, c)$ and $D_*(\beta, c)$ such that

$$D_*(\beta, c) \leq J(\beta, c) \leq D^*(\beta, c).$$

The upper bound on the exponent can be found by considering a point $\gamma^* \in \mathcal{F}(\beta, c)$ and setting $D^*(\beta, c) \stackrel{\Delta}{=} D(\gamma^*||\mathcal{P}_\beta)$. The choice is given in Proposition 1.

Lower bounds on the exponent function $J(\beta, c)$ are obtained by enlarging the optimization domain so that for the new domains the solution can be found in closed form. More exactly, we choose a new domain $\mathcal{F}_*$ such that $\mathcal{F} \subset \mathcal{F}_*$ and set

$$D_*(\beta, c) \stackrel{\Delta}{=} \min_{\gamma \in \mathcal{F}_*} D(\gamma || \mathcal{P}_\beta).$$

Although the cases $c > 1$ and $c < 1$ can be treated together, we treat them separately in order to take advantage of a simplified bound that can be obtained for $c > 1$; the case $c < 1$ is treated in Proposition 2 and the case $c > 1$ in Proposition 3.

Here we introduce some notations that will be used in the propositions that follow. Obviously, the distributions in $\mathcal{F}(\beta, c)$ must satisfy the feasibility condition (13), which can be written as

$$\gamma_0 + \frac{1}{2}\gamma_1 \geq 1 - \frac{1}{2}\beta. \tag{19}$$

For any value of $c \neq 1$ the boundary of $\mathcal{F}(\beta, c)$ given by the performance bounds (14) and (15) is

$$\gamma_1 = \frac{\beta}{c}\gamma_0 + \beta\frac{c-1}{c}. \tag{20}$$

The domain for the pair (γ_0, γ_1) is the domain with bounds given by $\gamma_0 + \gamma_1 \leq 1$, (19) and (20). This domain is represented in Fig. 3 for $c > 1$ and in Fig. 4 for $c < 1$. Its boundary determined by the performance condition (20) is a segment with endpoints $(\gamma_{0L}, \gamma_{1L})$ and $(\gamma_{0U}, \gamma_{1U})$. These two points are given by

$$\begin{aligned}
\gamma_{0L} &\stackrel{\Delta}{=} 1 - \beta + \beta^2 \frac{1}{2c + \beta} \\
\gamma_{1L} &\stackrel{\Delta}{=} \beta - \frac{\beta^2}{c} + \frac{\beta^3}{c(2c + \beta)} \\
\gamma_{0U} &\stackrel{\Delta}{=} 1 - \beta + \beta^2 \frac{1}{c + \beta} \\
\gamma_{1U} &\stackrel{\Delta}{=} \beta - \frac{\beta^2}{c} + \frac{\beta^3}{c(c + \beta)}.
\end{aligned}$$

As mentioned before, the next proposition gives an upper bound on the large deviations exponent.

Proposition 1: We have the following upper bound on the exponent:

$$J(\beta, c) \leq D^*(\beta, c),$$

with

$$D^*(\beta, c) = D(\gamma^*(\beta, c) || \mathcal{P}_\beta),$$

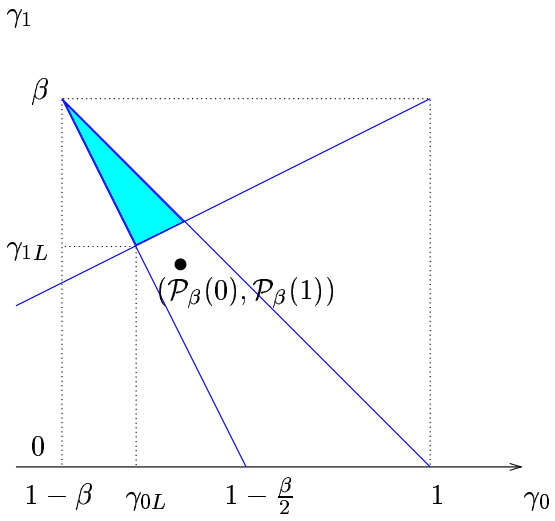

Fig. 3. The optimization region for γ_0 and γ_1, $c > 1$.

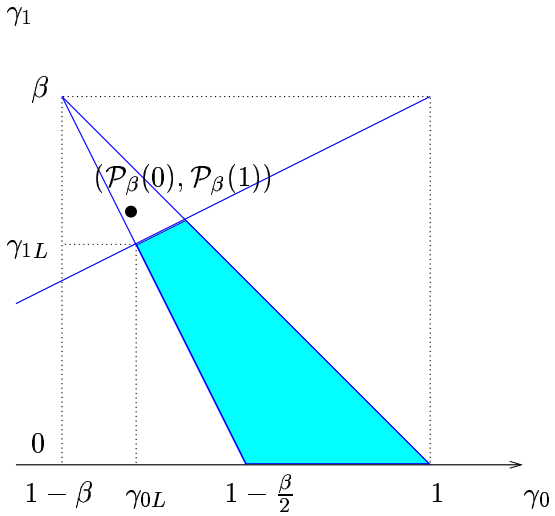

Fig. 4. The optimization region for γ_0 and γ_1, $c < 1$.

and

$$\begin{aligned}
\gamma^*(\beta, c) &\triangleq [\gamma_0^*, \ldots, \gamma_4^*, 0, \ldots, 0, \ldots] \\
\gamma_0^* &\triangleq \left(1 - \frac{\beta}{6}\right)\gamma_{0,L} + \frac{\beta}{6}\gamma_{0,U} \\
\gamma_1^* &\triangleq \left(1 - \frac{\beta}{6}\right)\gamma_{1,L} + \frac{\beta}{6}\gamma_{1,U} \\
\gamma_2^* &\triangleq 1 - \gamma_0^* - \gamma_1^* - \left(1 - \frac{\beta}{6}\right)\frac{\beta^3}{6(c+\beta)} \\
\gamma_3^* &\triangleq \left(1 - \frac{\beta}{3}\right)\frac{\beta^3}{6(c+\beta)} \\
\gamma_4^* &\triangleq \frac{\beta}{3}\frac{\beta^3}{6(c+\beta)}.
\end{aligned}$$

□

One can check that the point γ^* proposed belongs to the optimization domain, with the pair (γ_0^*, γ_1^*) on the boundary of the domain. One can consider instead the simpler variant $\gamma^* = [\gamma_{0,L}, \gamma_{1,L}, 1-\gamma_{0,L}-\gamma_{1,L}, 0, \dots]$, which gives a simpler expression. However, this last choice would give a tight bound only for small values of β, and it is quite useless for deriving approximative confidence intervals for the estimator proposed. The bound given in the proposition is tight in a large interval (up to $\beta \approx \frac{1}{2}$) and provides an excellent approximation for the confidence intervals; these are discussed in the simulations results - Subsection V-B.

Introduce the notations :

$$\bar{\gamma}_{01L} \stackrel{\Delta}{=} 1-\gamma_{0L}-\gamma_{1L} \quad , \quad \bar{\gamma}_{1L} \stackrel{\Delta}{=} 1-\gamma_{1L}$$
$$\bar{\mathcal{P}}_{\beta,01} \stackrel{\Delta}{=} 1-\mathcal{P}_\beta(0)-\mathcal{P}_\beta(1) \quad , \quad \bar{\mathcal{P}}_{\beta,1} \stackrel{\Delta}{=} 1-\mathcal{P}_\beta(1).$$

The following quantities defined will be used in Proposition 2 and their significance is explained in its proof :

$$\begin{aligned} \tilde{\gamma}_0 &\stackrel{\Delta}{=} \frac{\beta(1-c)+c\exp(-\beta)(1+\beta)}{c+\beta} \\ \tilde{\gamma}_1 &\stackrel{\Delta}{=} \mathcal{P}_\beta(0)+\mathcal{P}_\beta(1)-\tilde{\gamma}_0 \quad \text{and} \\ \tilde{\gamma}_{*,0} &\stackrel{\Delta}{=} 1-\frac{\beta}{2}-\frac{1}{2}\tilde{\gamma}_1; \end{aligned} \tag{21}$$

$$\Delta_0 \stackrel{\Delta}{=} \gamma_{0,U}-\gamma_{0,L} \; ; \quad \Delta_1 \stackrel{\Delta}{=} \gamma_{1,U}-\gamma_{1,L}$$
$$C_1 \stackrel{\Delta}{=} \left(\frac{\tilde{\gamma}_0}{\mathcal{P}_\beta(0)}\right)^{\Delta_0}\left(\frac{\tilde{\gamma}_1}{\mathcal{P}_\beta(1)}\right)^{\Delta_1} \tag{22}$$
$$C_2 \stackrel{\Delta}{=} \frac{1-\tilde{\gamma}_1}{1-\mathcal{P}_\beta(1)}\mathcal{P}_\beta(0)-\left(1-\frac{\beta}{2}-\frac{1}{2}\tilde{\gamma}_1\right). \tag{23}$$

Proposition 2: If $c<1$, we have the following lower bound on the exponent

$$D_*(\beta,c) \le J(\beta,c)$$

where $D_*(\beta,c)$ is given by

$$D_*(\beta,c) = \begin{cases} D_0 & if \;\; C_1 < 1 \\ D_1 & if \;\; C_1 > 1 \; and \;\; C_2 > 0 \\ D_{corner} & if \;\; C_1 > 1 \; and \;\; C_2 < 0 \end{cases},$$

with

$$\begin{aligned} D_0 &\triangleq \tilde{\gamma}_0 \log \frac{\tilde{\gamma}_0}{\mathcal{P}_\beta(0)} + (1-\tilde{\gamma}_0)\log\frac{1-\tilde{\gamma}_0}{1-\mathcal{P}_\beta(0)} \\ D_1 &\triangleq \tilde{\gamma}_1 \log \frac{\tilde{\gamma}_1}{\mathcal{P}_\beta(1)} + (1-\tilde{\gamma}_1)\log\frac{1-\tilde{\gamma}_1}{1-\mathcal{P}_\beta(1)} \\ D_{corner} &\triangleq \tilde{\gamma}_{*,0} \log \frac{\tilde{\gamma}_{*,0}}{\mathcal{P}_\beta(0)} + \tilde{\gamma}_1 \log \frac{\tilde{\gamma}_1}{\mathcal{P}_\beta(1)} + (1-\tilde{\gamma}_{*,0}-\tilde{\gamma}_1)\log\frac{1-\tilde{\gamma}_{*,0}-\tilde{\gamma}_1}{1-\mathcal{P}_\beta(0)-\mathcal{P}_\beta(1)}. \end{aligned}$$

Proof: see the Appendix. □

Proposition 3: For any $\beta \in [0,1)$ and $c > \frac{\beta(1+\exp(-\beta))}{2(1-\exp(-\beta))}$, we have the following lower bound on the exponent. In particular, the bound holds for all $c > 1.0821$:

$$D_*(\beta,c) = \begin{cases} \gamma_{0L}\log\frac{\gamma_{0L}}{\mathcal{P}_\beta(0)} + \gamma_{1L}\log\frac{\gamma_{1L}}{\mathcal{P}_\beta(1)} + \bar{\gamma}_{01L}\log\frac{\bar{\gamma}_{01L}}{\bar{\mathcal{P}}_{\beta,01}} & if \quad \mathcal{P}_\beta(0)\frac{1-\gamma_{1L}}{1-\mathcal{P}_\beta(1)} < \gamma_{0L} \\ \gamma_{1L}\log\frac{\gamma_{1L}}{\mathcal{P}_\beta(1)} + \bar{\gamma}_{1L}\log\frac{\bar{\gamma}_{1L}}{\bar{\mathcal{P}}_{\beta,1}} & otherwise \end{cases}.$$

Proof: see the Appendix. □

Note that Proposition 3 does not provide a lower bound for all pairs (β, c). Although a solution similar to the one given in Proposition 2 can be given using an identical technique, the constraint imposed in Proposition 3 holds for most practical situations, the bound obtained is tight and has a relatively simple expression.

Using the bounds on the error exponent, we can study its behavior for small β, by taking the limits of the bounds [7]. For $c > 1$, we have the following behavior of $J(\beta, c)$:

$$\lim_{\beta\to 0}\frac{J(\beta,c)}{\beta^2} = \frac{c-1-\ln(c)}{2c} \triangleq B. \tag{24}$$

For $c < 1$, the upper bound is identical :

$$\lim_{\beta\to 0}\frac{J(\beta,c)}{\beta^2} \leq \frac{c-1-\ln(c)}{2c}.$$

The lower bound obtained here for $c < 1$ is not tight (it gives $\lim_{\beta\to 0}\frac{J(\beta,c)}{\beta^3} \in (0,\infty)$). This can be seen in the simulations section.

However, it can be shown that (24) holds in this case as well. The first step of the proof is to compute the limits when $\beta \to 0$ for D_{corner}/β^2 and for $D(\gamma^*||(\mathcal{P}_\beta(0), \mathcal{P}_\beta(1), \bar{\mathcal{P}}_{\beta,01}))/\beta^2$, with γ^* defined after Proposition 1. Both limits are equal to the RHS of (24). Then the result can be extended to the rest of the distributions of interest (those that can be lead to the minimum value) that lie on the boundary of the extended domain considered ($\tilde{\mathcal{F}}_*(\beta, c)$, in the appendix, the proof of Proposition 3).

Also, the simulations revealed that if a certain performance is required, then when the number of sensors is increased, the ratio β necessary to achieve a certain performance decreases. In fact, in the

simulations made, the remarkable relations hold ($\tilde{\mathbb{P}}$ is the empirical probability):

$$\tilde{\mathbb{P}}\left\{\frac{\hat{N}}{N} \geq c > 1\right\} < \exp\left(-NJ(\beta, c)\right), \tag{25}$$

$$\tilde{\mathbb{P}}\left\{\frac{\hat{N}}{N} \leq c < 1\right\} < \exp\left(-NJ(\beta, c)\right). \tag{26}$$

This suggests that the right hand side expression that uses the exponent could be an upper bound for the true probability $\mathbb{P}\left\{\frac{\hat{N}}{N} \geq c\right\}$. If this is true, then by using the asymptotic behavior of $J(\beta, c) = \beta^2 B + o(\beta^2)$, with B defined in (24), a strong large deviations result will follow. The main implication of such a result is that one can achieve reliable estimation using only $n = O(\sqrt{N})f(N)$ samples, with $f(N) \to \infty$.

V. Simulations and Numerical Results

A. The Performance of Algorithms Presented in Section III

In this subsection we investigate by simulations the performance of the algorithms presented in Section III. For each simulation the total number of (operating) sensors N is fixed. The performance measure used is the confidence interval for the relative estimation error. In figures, the x-axis represents the ratio $\beta \stackrel{\Delta}{=} \frac{n}{N}$ between the lenght n of the vector sample and N. For a fixed $\varepsilon \in (0, 1)$, $\varepsilon \ll 1$, and for each vector sample length n, we determined experimentally two quantities c_U (upper bound, "UB" in plots legends) and c_L (lower bound, "LB" in plots legends) such that

$$\tilde{\mathbb{P}}\left\{\frac{\hat{N}}{N} > c_U > 1\right\} = \varepsilon,$$

$$\tilde{\mathbb{P}}\left\{\frac{\hat{N}}{N} < c_L < 1\right\} = \varepsilon,$$

where we denoted by $\tilde{\mathbb{P}}$ the observed empirical probability of an event. Thus, for given N, n and ε, we have

$$\tilde{\mathbb{P}}\left\{\frac{\hat{N}}{N} \in (c_L, c_U)\right\} = 1 - 2\varepsilon.$$

The plots presented can be used to determine the size of the vector sample necessary to achieve a required performance.

In Fig. 5 the performance of the Good-Turing estimator is compared to the performance of the ML estimator given by (2). For the situation analyzed, *i.e.*, $N = 1000$, 10000 Monte-Carlo runs, and $\varepsilon = 0.01$,

the confidence intervals for the two methods are virtually identical. Other combinations of parameters showed that the performance loss by using the Good-Turing estimator instead of the ML one is negligible.

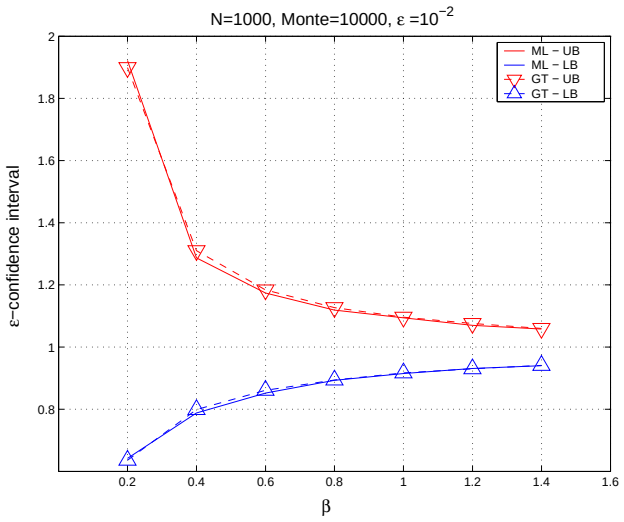

Fig. 5. ML vs Good-Turing; the performance difference is negligible

For the histogram estimation case, we consider again $N = 1000$; the sensors belong to 4 classes, the ratios $\frac{N(\varphi)}{N}$ for $\varphi = 1, \ldots, 4$ are given in the vector $\mathbf{C}_d = [0.1, 0.2, 0.3, 0.4]$.

In Fig. 6 are represented only the performance plots for $\varphi = 1$ ($N(1) = 100$) and $\varphi = 4$ ($N(4) = 400$), as well as the performance plots for estimation of N. The plots reveal that the performance of the proposed estimators for the number of operating sensors in each class is better when the number of sensors in each class is larger. Also, one can see that the performance of the estimator for the "larger" classes is very close to the performance of the Good-Turing estimator of the total number of samples. The performance plots also reveal that the performance of the estimator (7) based on histogram scaling is slightly better than the one of the estimator (6) derived by applying the Good-Turing formula once again.

In the next example the number of sensors in each class varies during data collection. In the setup considered each sensor can belong to one of 5 classes, and initially all sensors are in class 1.

The class of each sensor is a Markov chain with 4 transient states and with one absorbing state, as represented in Fig. 7. Each sensor can change its class in any time slot, but only by increasing its class index by one. The sensors which belong to class 5 can't change the class anymore. Each sensor changes its class independently of the other sensors and of the reception process, with a fixed probability $p_0 = 5 \times 10^{-4}$. This model can be a good approximation for the variation of the remaining battery

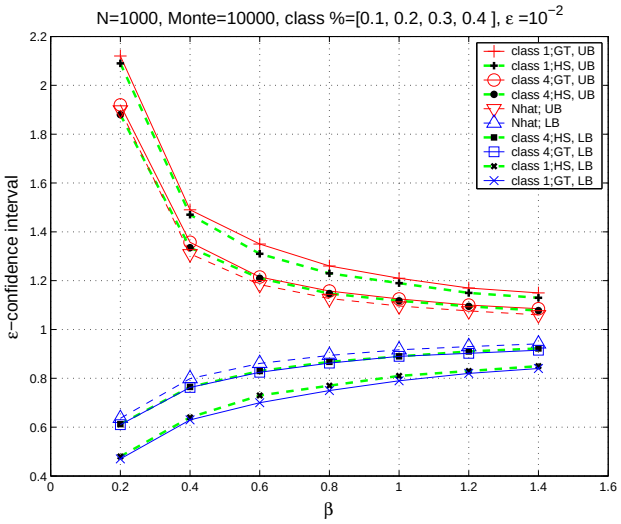

Fig. 6. Histogram estimation using estimators (6) - "GT" and (7) - "HS"

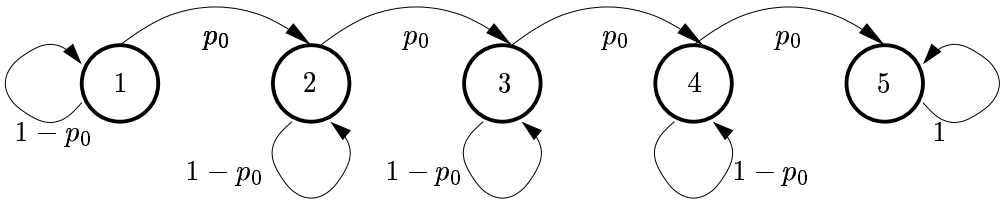

Fig. 7. The Markov Process representing the variation of the class of one sensor

energy of the sensors; each class corresponding to a certain interval for the energy. The randomness of transitions between classes is created by a MAC protocol which varies the transmission power from slot to slot (*e.g.*,function of the quality of the uplink wireless channel).

The total number of sensors is $N = 1000$. Confidence intervals for the estimation of the number of sensors in classes 1, 4 and 5 using the variant of the estimator (7) are given in Figs. 8, 9, and 10. It can be observed that for the transient states 1 and 4 the relative error for estimation is not monotonic with the increase of the number of samples. This happens because when the time passes and more samples are collected, fewer sensors have classes corresponding to the transient states of the Markov chain. The relative error can take large values if the estimated variables are extremely small. On the other hand, for class 5 that corresponds to the unique absorbing state of the Markov chain, the variation of the relative error with the number of samples available is similar to the variation of the relative error for the total

number of operating sensors, as expected.

For the time-varying case we represented only the performance of the estimator based on histogram scaling, but the observation made previously that the estimator (6) has a slightly worse performance holds for time-varying classes as well.

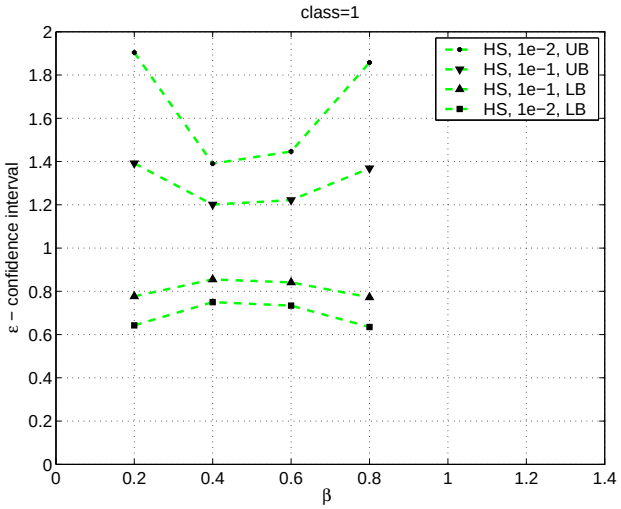

Fig. 8. Histogram estimation for time-varying classes; class corresponding to the transient state 1

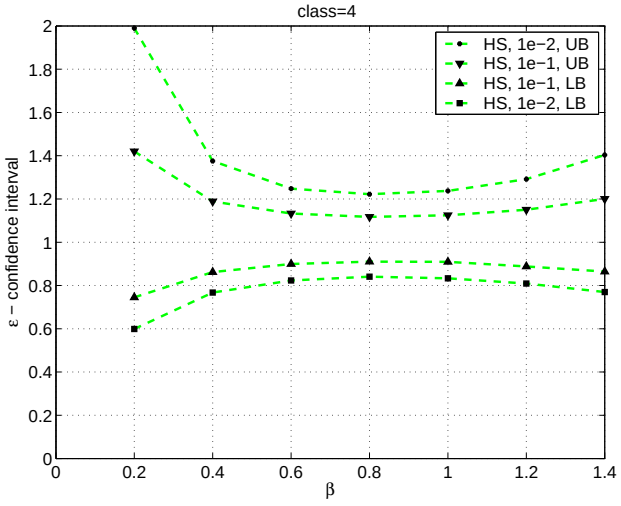

Fig. 9. Histogram estimation for time-varying classes; class corresponding to the transient state 4

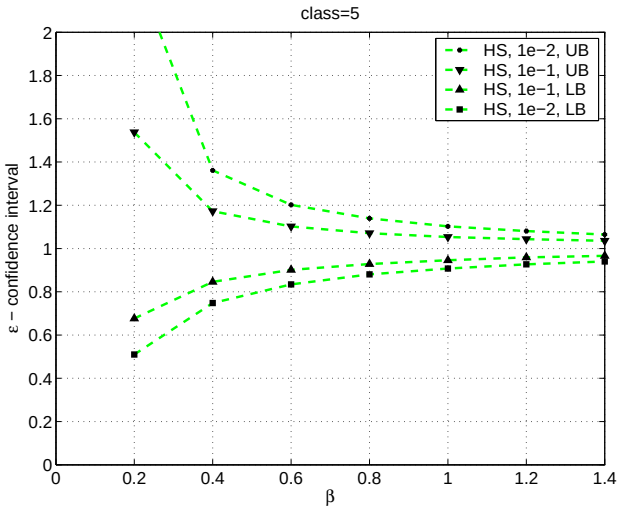

Fig. 10. Histogram estimation for time-varying classes; class corresponding to the absorbing state 5

B. Performance Bounds Using the Large Deviation Approximation

In Fig. 11 the confidence intervals for the relative error of the estimator are represented for $N = 16000$ and $\varepsilon = 0.001$. The way curves were obtained was explained in Subsection V-A. Three more pairs of curves are represented in Fig. 11, the elements of a pair corresponding to the two cases $c > 1$ and $c < 1$.

The first pair is given by the quantity c obtained using the large deviations formula $\frac{1}{N}\log\varepsilon = -J(\beta, c)$. The other two pairs are obtained in the same way but using instead of $J(\beta, c)$ the upper and lower bounds $D^*(\beta, c)$ and $D_*(\beta, c)$ derived before, *i.e.*, solving $\frac{1}{N}\log\varepsilon = -D^*(\beta, c)$ and $\frac{1}{N}\log\varepsilon = -D_*(\beta, c)$ for $c > 1$ and $c < 1$.

One might note that for $c > 1$ the curves obtained using the bounds on the error exponent are excellent approximations to the curve obtained using the computed exponent. On the other hand, for $c < 1$, the curve obtained using D^* is tight (if $\beta < 0.5$), while the one obtained using D_* is quite loose for small β and reasonable tight for large β. Actually, it can be shown that with the technique used (replacing the conservation condition (11) with the equivalent condition (19) for (γ_0, γ_1)), the best lower bound on the exponent is loose for small β.

For the numerical results represented in Fig. 11 the relations (25) and (26) mentioned in the end of Section IV hold:

$$\tilde{\mathbb{P}}\left\{\frac{\hat{N}}{N} \geq c > 1\right\} < \exp\left(-N\,J(\beta, c)\right),$$

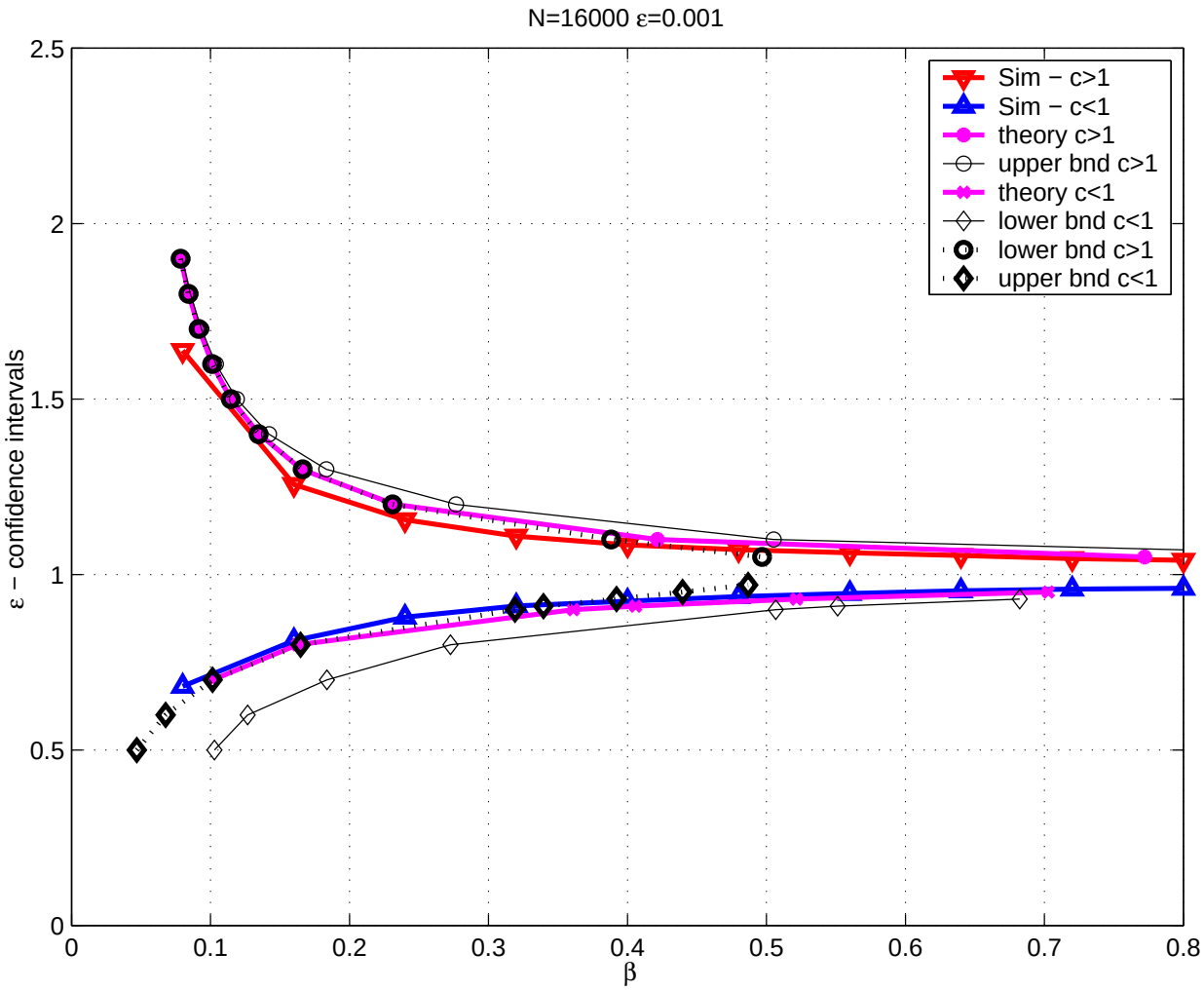

Fig. 11. Confidence intervals for the performance of proposed estimator. The way they were obtained is given in the legend.

$$\tilde{\mathbb{P}}\left\{\frac{\hat{N}}{N} \leq c < 1\right\} < \exp\left(-NJ(\beta, c)\right).$$

If these relations are true in general, then a strong large deviation result will follow.

VI. Conclusions

The estimator of the number of operating sensors based on the Good-Turing estimator was shown to achieve a performance similar to the ML estimator. It can also be used to solve more complicated problems like class histogram estimation. Its simple expression allowed us to perform a performance analysis using the principle of large deviations. We provide closed form upper and lower bounds for the large deviations exponent, which are used further to characterize the behavior of the exponent for small

β and to derive approximative confidence intervals for the performance of the estimator proposed.

The large deviations analysis and the simulations suggested that one can achieve reliable estimation using only $n = O(\sqrt{N})f(N)$ samples, with $f(N) \to \infty$. In fact, the last statement (but not the large deviation one) was already shown in [13]. In contrast, under the same i.i.d. random collection model with uniform distribution, the number of samples necessary to achieve a complete collection (with high probability) is $N \log N + O(N)$. Thus, an estimation approach can reduce dramatically the number of necessary samples.

An accurate modeling of a specific communication system would require changes in the basic model considered in this paper. An example was investigated in [6], where the vector sample is collected using a receiver with multi-packet reception (MPR) capability. For the same number of available samples, the model mismatch introduces a slight degradation of estimator's performance. However, for a certain required performance, the MPR capability of the mobile access point reduces dramatically the necessary sample collection time.

Finally, we note that the proposed algorithms are not restricted to SENMA. For other types of sensor networks, for example the multihop ad hoc sensor networks with gateway nodes, the proposed algorithm can be easily implemented at the gateway nodes or fusion center. However, the performance analysis that depends on the i.i.d. random collection model of SENMA may not apply.

APPENDIX

Proof of Proposition 2

Consider the following domain

$$\tilde{\mathcal{F}}_*(\beta, c) = \left\{ \gamma \ : \ \sum_{i=0}^{\infty} \gamma_i = 1; \ \gamma_1 \leq \gamma_0 \frac{\beta}{c} - \beta \frac{1-c}{c}, \ \gamma_0 + \frac{1}{2}\gamma_1 \geq 1 - \frac{\beta}{2} \right\}.$$

The difference between $\tilde{\mathcal{F}}_*(\beta, c)$ and $\mathcal{F}(\beta, c)$ is that the conservation condition (11) in the definition of $\mathcal{F}(\beta, c)$ is replaced by condition (19) derived for the pair (γ_0, γ_1). From the convexity property of Kullback-Leibler distance [12], we know that the optimizing solution $\tilde{\gamma}_*$ must be on the boundary of the optimization domain. Taking into account the position of the Poisson distribution with respect to this domain (Fig. 12), in our case, the solution must satisfy

$$\gamma_1 = \gamma_0 \frac{\beta}{c} - \beta \frac{1-c}{c}. \tag{27}$$

Moreover, if the first two elements (γ_0, γ_1) of a distribution γ are given, the distribution that minimizes $D(\gamma||\mathcal{P}(\beta))$ and the corresponding minimized value are known in closed form (for $i > 1$ the elements

$\gamma_{*,i}$ of the solution are proportional to $\mathcal{P}_\beta(i)$). Thus the optimization problem

$$\tilde{\gamma}_* = \arg\min_{\gamma \in \tilde{\mathcal{F}}_*} D(\gamma_* || \mathcal{P}_\beta)$$

reduces to an optimization problem with only one parameter $\lambda \in [0, 1]$, that indicates the position of the pair $(\tilde{\gamma}_{*,0}, \tilde{\gamma}_{*,1})$ on the boundary of the domain. This optimization problem can be solved only numerically.

A simpler bound can be obtained by replacing $\tilde{\mathcal{F}}_*(\beta, c)$ with a domain that is not convex anymore, but is the union of two convex domains. The domain that contains the solution can be found by testing one simple condition. The detailed steps are given below; the domains are illustrated in Fig. 12.

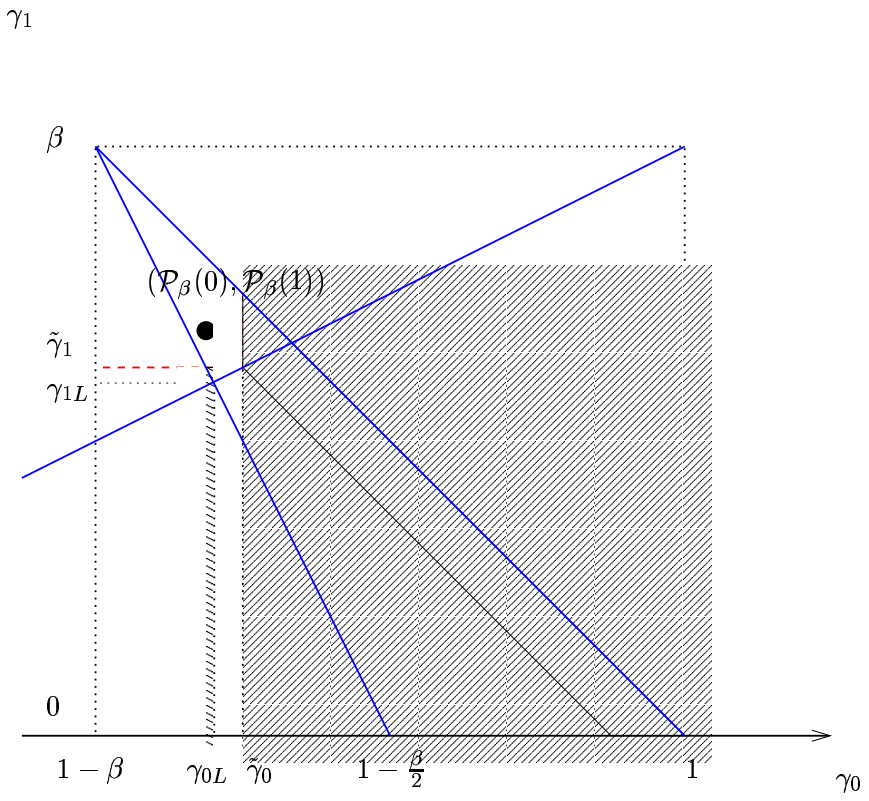

Fig. 12. The optimization region for the pair (γ_0, γ_1), $c < 1$, detail.

Choose $(\tilde{\gamma}_0, \tilde{\gamma}_1)$ on the boundary (20) such that

$$\tilde{\gamma}_0 + \tilde{\gamma}_1 = \mathcal{P}_\beta(0) + \mathcal{P}_\beta(1).$$

This gives the solution

$$\begin{aligned}\tilde{\gamma}_0 &= \frac{\beta(1-c)+c\exp(-\beta)(1+\beta))}{c+\beta}\\ \tilde{\gamma}_1 &= \mathcal{P}_\beta(0)+\mathcal{P}_\beta(1)-\tilde{\gamma}_0.\end{aligned}$$

Consider the following domain $\mathcal{F}_*(\beta,c)$:

$$\mathcal{F}_*(\beta,c)=\mathcal{F}_{*,1}(\beta,c)\cup\mathcal{F}_{*,2}(\beta,c),$$

with

$$\mathcal{F}_{*,1}(\beta,c)=\left\{\gamma\ :\ \gamma_1\le\tilde{\gamma}_1,\ \sum_{i=0}^{\infty}\gamma_i=1,\ \gamma_0+\frac{1}{2}\gamma_1\ge 1-\frac{\beta}{2},\ \gamma_0+\gamma_1\le\tilde{\gamma}_0+\tilde{\gamma}_1\right\},$$

and

$$\mathcal{F}_{*,2}(\beta,c)=\left\{\gamma\ :\ \gamma_0\ge\tilde{\gamma}_0,\ \sum_{i=0}^{\infty}\gamma_i=1,\ \gamma_0+\frac{1}{2}\gamma_1\ge 1-\frac{\beta}{2},\ \gamma_0+\gamma_1\ge\tilde{\gamma}_0+\tilde{\gamma}_1\right\}.$$

The optimization solutions over each of domains $\mathcal{F}_{*,1}$, $\mathcal{F}_{*,2}$ provide the solutions given in the text of the lemma.

The test on C_1 given by (22) gives which of the domains contains the minimum. As mentioned before, given any point (γ_0,γ_1) on the boundary (20), we know the optimizing distribution; the position of this point can be parametrized using only one parameter, $\lambda\in[0,1]$. The derivative with respect to the parameter λ of the minimized Kullback-Leibler distance $D(\tilde{\gamma}(\lambda)||\mathcal{P}_\beta)$ can be computed and analyzed easily (but the value λ for which it vanishes can't be computed in closed form). The test on C_1 is a test on the sign of the derivative mentioned for λ corresponding to the pair $(\tilde{\gamma}_0,\tilde{\gamma}_1)$ is positive or negative; this determines which of the two domains contains the solution of the minimization problem over the domain $\mathcal{F}_*(\beta,c)$

The test on C_2 given by (23) is necessary to assure that once $\tilde{\gamma}_1$ is fixed, the optimizing value of γ_0 (without any other constraints) does not fall outside of the optimization region, *i.e.*, the feasibility condition (19) is satisfied. If this is not the case, one can choose the value given by condition (19), value denoted by $\tilde{\gamma}_{*,0}$. A similar discussion, but more detailed, is given in the proof of Proposition 3.

Proof of Proposition 3

As before, the lower bound is obtained by finding a convex domain $\mathcal{F}_*$, $\mathcal{F}\subset\mathcal{F}_*$, and solving the optimization problem over $\mathcal{F}_*$. The choice made is

$$\mathcal{F}_*=\left\{\gamma:\sum\gamma_i=1,\ \gamma_1\ge\gamma_{1,L},\ \ \gamma_0+\frac{1}{2}\gamma_1\ge 1-\frac{\beta}{2}\right\}.$$

Denote

$$\gamma_* \stackrel{\Delta}{=} \arg\min_{\gamma \in \mathcal{F}_*} D(\gamma||\mathcal{P}_\beta),$$

and

$$D_*(\beta, c) \stackrel{\Delta}{=} \min_{\gamma \in \mathcal{F}_*} D(\gamma||\mathcal{P}_\beta) = D(\gamma_*||\mathcal{P}_\beta).$$

First, we need to check that $\mathcal{P}_\beta \notin \mathcal{F}_*$. This means

$$\mathcal{P}_\beta(1) \quad \leq \quad \gamma_{1L},$$

or, $\beta \exp(-\beta) \leq \beta \frac{c - \frac{\beta}{2}}{c + \frac{\beta}{2}}$, which gives

$$c > \frac{\beta(1 + \exp(-\beta))}{2(1 - \exp(-\beta))}.$$

A calculation of the RHS will give that for $\beta \in (0, 1]$ the condition is true for all $c > 1.0821$, which is enough for most practical situations. In this case, $\gamma_{1,*} = \gamma_{1L}$, which simplifies the solution.

If the condition $\gamma_0 + \frac{1}{2}\gamma_1 \geq 1 - \frac{\beta}{2}$ from the definition of $\mathcal{F}_*$ is ignored, then the minimum of $D(\gamma||\mathcal{P}_\beta)$ is

$$D_*(\beta, c) = \gamma_{1L} \log \frac{\gamma_{1L}}{\mathcal{P}_\beta(1)} + \bar{\gamma}_{1L} \log \frac{\bar{\gamma}_{1L}}{\bar{\mathcal{P}}_{\beta,1}}.$$

The constraint $\gamma_0 + \frac{1}{2}\gamma_1 \geq 1 - \frac{\beta}{2}$ is irrelevant if the optimizing $\gamma_{0,*}$ belongs to the domain $\mathcal{F}_*$, *i.e.*, :

$$\gamma_{0,*} \stackrel{\Delta}{=} \mathcal{P}_\beta(0) \frac{1 - \gamma_{1L}}{1 - \mathcal{P}_\beta(1)} \geq \gamma_{0L}.$$

If the condition above is not satisfied, then we have $\gamma_{0,*} = \gamma_{0L}$ and the rest of the distribution is determined accordingly, which gives the first formula used.

The condition (to use the first formula) is

$$\frac{\mathcal{P}_\beta(0)}{1 - \mathcal{P}_\beta(1)} < \frac{\gamma_{0L}}{1 - \gamma_{1L}}$$

or,

$$\frac{\beta^2}{\beta^2 - 2\beta c + 2c + \beta} < 1 - \frac{\mathcal{P}_\beta(0)}{1 - \mathcal{P}_\beta(1)} \stackrel{\Delta}{=} T$$

This is

$$c > \frac{\beta^2(1 - T) - \beta T}{2T(1 - \beta)} \stackrel{\Delta}{=} H(\beta).$$

One can find

$$\lim_{\beta \to 0} \frac{H(\beta) - 1}{\beta} = \frac{1}{6}.$$

Thus, an approximative condition for small $c \geq 1$ is $\beta < 6(c - 1)$.

REFERENCES

[1] L. Tong, Q. Zhao, and S. Adireddy, "Sensor Networks with Mobile Agents," in *Proc. 2003 Intl. Symp. Military Communications*, (Boston, MA), Oct. 2003.

[2] P. Venkitasubramaniam, S. Adireddy, and L. Tong, "Sensor Networks with Mobile Agents: Optimal Random Access and Coding," *IEEE Journal on Sel. Areas in Comm.: Special Issue on Sensor Networks*, Sep. 2004.

[3] I. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, pp. 237–264, 1953.

[4] P. Dupuis, C. Nuzman, and P. Whiting, "Large Deviations Asymptotics for Occupancy Problems." `http://cm.bell-labs.com/cm/ms/who/nuzman/`, 2003.

[5] A. Orlitsky, N. Santhanam, and J. Zhang, "Always Good Turing: Asymptotically Optimal Probability Estimation," *Science*, vol. 302, pp. 427–431, Oct. 2003.

[6] C. Budianu and L. Tong, "Estimation of the Number of Operating Sensors in a Sensor Network," in *Proc of 2003 Asilomar Conference on Signals, Systems and Computers*, (Monterey, CA), Nov. 2003. `http://acsp.ece.cornell.edu/pubC.html/`.

[7] C. Budianu and L. Tong, "Good-Turing Estimation of the Number of Operating Sensors : a Large Deviations Analysis," in *ICASSP 2004*, May 2004.

[8] D. McAllester and R.E.Schapire, "On the Convergence Rate of Good-Turing Estimators," in *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, 2002.

[9] W. Esty, "The Efficiency of Good's Nonparametric Coverage Estimator," *The Annals of Statistics*, vol. 14, pp. 1257–1260, Sept. 1986.

[10] M. Finkestein, H. Tucker, and J. Veeh, "Confidence Intervals for the Number of Unseen Types," *Statistics and Probability Letters*, vol. 37, pp. 423–430, 1998.

[11] S. Resnick, *A Probability Path*. Boston: Birkhäuser, 1998.

[12] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[13] C. Budianu, *Estimation in Wireless Communication Systems*. PhD thesis, Cornell University, Ithaca, NY, August 2004.