
Bayesian Multiple Target Localization

Purnima Rajan

Department of Computer Science, Johns Hopkins University

PURNIMA@CS.JHU.EDU

Weidong Han

Department of Operations Research and Financial Engineering, Princeton University

WHAN@PRINCETON.EDU

Raphael Sznitman

ARTORG Center, University of Bern

RAPHAEL.SZNITMAN@ARTORG.UNIBE.CH

Peter I. Frazier

School of Operations Research and Information Engineering, Cornell University

PF98@CORNELL.EDU

Bruno M. Jedynek

Department of Applied Mathematics & Statistics, Johns Hopkins University

BRUNO.JEDYNAK@JHU.EDU

Abstract

We consider the problem of quickly localizing multiple targets by asking questions of the form “How many targets are within this set” while obtaining noisy answers. This setting is a generalization to multiple targets of the game of 20 questions in which only a single target is queried. We assume that the targets are points on the real line, or in a two dimensional plane for the experiments, drawn independently from a known distribution. We evaluate the performance of a policy using the expected entropy of the posterior distribution after a fixed number of questions with noisy answers. We derive a lower bound for the value of this problem and study a specific policy, named the dyadic policy. We show that this policy achieves a value which is no more than twice this lower bound when answers are noise-free, and show a more general constant factor approximation guarantee for the noisy setting. We present an empirical evaluation of this policy on simulated data for the problem of detecting multiple instances of the same object in an image. Finally, we present experiments on localizing multiple faces simultaneously on real images.

1. Introduction

The task of localizing structures of interest, or *targets*, appears in numerous applications, such as finding quasars in astronomical data (Mortlock, 2009), localizing faces in images (Ali et al., 2012) or counting synapses in microscopy volumes (Merchan-Perez et al., 2009). Once a competent detection scheme is available, the localization task often reduces to evaluating each possible location in an exhaustive fashion. Such strategies are highly effective and easy to implement, which contributes to their widespread use.

Yet such localization strategies do not scale with data size requirements since their computational complexity depends directly on the searchable domain’s size. This problem is critical in analysis of enormous microscopy volumes where localizing and counting intra-cellular structures such as mitochondria or synapses is critical to understanding brain processes (Lee et al., 2007). Similarly, efficient localization of object instances in images remains challenging given the growing amount of image data to evaluate.

To address this issue, theoretical works have established strategies to reduce the computation needed to find targets. Generally, literature considering such problems falls into two categories: those that consider a single target ($k = 1$) and those that consider multiple targets ($k \geq 1$).

For single-target localization, (Jedynek et al., 2012) considered a Bayesian setting and used the entropy of the posterior distribution to measure accuracy, as we do here. Within this context, a number of policies have been proposed, such as the dyadic policy and the greedy probabilistic bisection (Horstein, 1963), which was further stud-

ied in (Castro & Nowak, 2007; Waeber et al., 2013). (Tsiligkaridis et al., 2013) more recently generalized this probabilistic bisection policy to multiple questioners as well. A discretized version of probabilistic bisection was studied by (Burnashev & Zigangirov, 1975).

For multiple-target localization, three variations appear frequently: the Group Testing problem (Du & Hwang, 2000; Stinson et al., 2000; Eppstein et al., 2007; Harvey et al., 2007; Porat & Rothschild, 2008), the subset-guessing game associated with the Random Chemistry algorithm (Kauffman, 1996; Buzas & Warrington, 2013) and the Guessing Secret game (Chung et al., 2001). In each case, the goal is to query subsets, A , of the search space to determine an unknown set S . In the Group Testing problem, questions are of the form: “Is $A \cap S \neq \emptyset$?” In the subset-guessing game associated with the Random Chemistry algorithm, questions are of the form “Is $S \subset A$?” In the Guessing Secret game, when queried with a set A , the responder chooses an element from S according to any self-selected rule and specifies whether this chosen element is in A . The chosen element itself is not revealed and may change after each question. Thus, the answer is 1 when $S \subset A$, 0 when $A \cap S = \emptyset$, and can be 0 or 1 otherwise.

Four major practical considerations, however, severely limit the usability of existing theoretical results in real applications. First, when multiple targets need to be located, significant noise in the query answers are typically observed in real applications but ignored in most theoretical analyses. Second, existing theoretical analyses lack generality, considering very specific models for what is observed, rather than a general observational model that could be adapted to the application at hand. Third, many methods with theoretical guarantees on query complexity require a great deal of computation, and cannot be used in computation-constrained applications. Fourth, existing theoretical analysis often do not make clear the computational gain possible over repeatedly applying optimal strategies for single-target localization, making simpler strategies based on localizing single-targets more attractive.

This work addresses these concerns, by proposing and then analyzing the dyadic policy for simultaneous localization of multiple targets. This policy and our analysis uses an observational model that allows noise, and is general enough to subsume Group Testing, Random Chemistry, and a wide variety of other problems. We provide an explicit expression for the expected entropy of the posterior after N queries from this policy, and together with a simple information-theoretic lower bound on the expected entropy under the optimal policy, we show an approximation guarantee for the expected entropy reduction under the dyadic policy. Using this result, we can then demonstrate significant computation gains over repeated single-target optimal

localization. The dyadic policy can be computed quickly and is non-adaptive, making it easy to parallelize, and far simpler to implement than dynamic strategies. Moreover, it allows easy and exact computation of the expected number of targets at each location of our space.

2. PROBLEM FORMULATION

Let $\theta = (\theta_1, \dots, \theta_k)$ be a random vector taking values in \mathbb{R}^k . θ_i represents the location of the i th target of interest, $i = 1, \dots, k$. We assume that $\theta_1, \dots, \theta_k$ are i.i.d. with density f_0 , and joint density p_0 . We refer to p_0 as the Bayesian prior probability distribution on θ . Note that even if the targets are indistinguishable, they are modeled as a vector, and not a set. This is a key requirement for simplifying the combinatorics of the probabilistic analysis. We will ask a series of $N > 0$ questions to locate $\theta_1, \dots, \theta_k$, where each question takes the form of a subset of \mathbb{R} . The answer to this question is the number of targets in this subset. However, this answer is not available to the questioner. Instead, a noisy version of this answer is available. More precisely, for each $n \in \{1, 2, \dots, N\}$, the n^{th} question is $A_n \subset \mathbb{R}$ and its noiseless answer is

$$Z_n = \mathbb{1}_{A_n}(\theta_1) + \dots + \mathbb{1}_{A_n}(\theta_k), \quad (1)$$

where $\mathbb{1}_A$ is the indicator function of the set A . The noisy observable answer is a random function of Z_n , namely

$$X_n = h(Z_n, W_n) \quad (2)$$

where h is a known function and W_n is a collection of independent random variable, which are also independent of θ . Note that our choice of the set A_n may depend upon the answers to all previous questions. Thus, the set A_n is random.

We call a rule for choosing the questions A_n a *policy*, and indicate it with the notation π . The distribution of A_n thus implicitly depends on π . When we wish to highlight this dependence, we use the notation P^π and E^π to indicate probability and expectation respectively. However, when the policy being studied is clear, we simply use P and E . We let Π be the space of all policies.

Throughout the paper, we use the notation $X_{a:b}$ for any $a, b \in \mathbb{N}$ to indicate the sequence (X_a, \dots, X_b) if $a \leq b$, and the empty sequence if $a > b$. We define $\theta_{a:b}$ and $A_{a:b}$ similarly.

We refer to the posterior probability distribution on θ after n questions and answers as p_n , so p_n is the conditional distribution of θ given $X_{1:n}$ and $A_{1:n}$.

After we exhaust our budget of N questions, we measure the quality of what we have learned via the differential entropy $H(p_N)$ of the posterior distribution p_N on the targets

at this final time,

$$H(p_N) = - \int_{\mathbb{R}^k} p_N(u_{1:k}) \log(p_N(u_{1:k})) du_{1:k}. \quad (3)$$

Throughout this paper, we use “log” to denote the logarithm to base 2. We let $H_0 = H(p_0)$, and we assume $-\infty < H(p_0) < +\infty$. The posterior distribution p_N , as well as its entropy $H(p_N)$, are random for $N > 0$, as they depend on $X_{1:N}$. Thus, we measure the quality of a policy $\pi \in \Pi$ as

$$R(\pi, N) = E^\pi[H(p_N)]. \quad (4)$$

Our goal is to characterize the solution to the optimization problem

$$\inf_{\pi \in \Pi} R(\pi, N). \quad (5)$$

Any policy that attains this infimum is called *optimal*.

Beyond theoretical interest, a policy for which $H(p_N)$ is small is of practical interest. It was shown in (Jedynak et al. 2012), section 4.3, that an optimal policy allows for localizing θ efficiently in the case of $k = 1$. We conjecture that the same occurs for arbitrary values of k . While (5) can be formulated as a partially observable Markov decision process (Frazier, 2010), and can be solved, in principle, via dynamic programming, the state space of this dynamic program (which is the space of posterior distributions over θ) is too large to allow solving it through brute-force computation. Thus, in this paper, rather than attempting to compute the optimal policy, we provide an easily computed lower bound on (5), and then study a particular policy, called the dyadic policy and defined below, whose performance is close to this lower bound.

3. THEORETICAL RESULTS

We first present an information-theoretic lower bound on the best expected entropy achievable, and a proof sketch. Proofs of all results may be found in the supplement.

Theorem 1.

$$H_0 - \log(k+1)N \leq H_0 - C_k N \leq \inf_{\pi \in \Pi} R(\pi, N) \quad (6)$$

The main arguments of the proof are as follows: First, at step n , the largest reduction in entropy that can be obtained in one question and on average occurs when the answer X_{n+1} and the targets θ have the largest mutual information given the history $I(\theta, X_{n+1}|X_{1:n})$, see Geman and Jedynak (1996). Second, since X_{n+1} depends on θ only through Z_{n+1} given $X_{1:n}$,

$$I(\theta, X_{n+1}|X_{1:n}) = I(Z_{n+1}, X_{n+1}|X_{1:n}). \quad (7)$$

Third, (7) is upper bounded by the *channel capacity*

$$C_k = \sup_q H \left(\sum_{z=0}^k q(z) f(\cdot|z) \right) - \sum_{z=0}^k q(z) H(f(\cdot|z)), \quad (8)$$

where q is a point mass function over the set $\{0, \dots, k\}$ and $f(\cdot|z)$ is the density, or point mass function of the noisy answer X_{n+1} given the noiseless answer Z_{n+1} . Since the noiseless answer Z_{n+1} is discrete and takes values in a set of size $k+1$, C_k is bounded above by $\log(k+1)$ providing the first inequality in (6).

In the noiseless case, both lower bounds in (6) are identical. Moreover, they are not achievable. Indeed, at $N = 1$, the target locations are independent, and so the answer to the first question is Binomial, and must have an entropy no better than $H(\text{Bin}(k, \frac{1}{2})) < \log(k+1)$. Moreover, as the expected entropy reduction is the sum of the expected entropy reduction at each question, the lower bound is not achievable for any N .

We now define an easy-to-compute policy, called the dyadic policy, and indicate it with the notation π_D . We first recall that the quantile function of θ_1 is

$$Q(p) = \inf \{u \in \mathbb{R} : p \leq F_0(u)\}, \quad (9)$$

where F_0 is the cumulative distribution function of θ_1 , corresponding to its density f_0 . Then, the *dyadic policy* consists in choosing at step $n \geq 1$ the set

$$A_n = \left(\bigcup_{j=1}^{2^{n-1}} \left(Q\left(\frac{2j-1}{2^n}\right), Q\left(\frac{2j}{2^n}\right) \right) \right) \cap \text{supp}(f_0), \quad (10)$$

where $\text{supp}(f_0)$ is the support of f_0 , i.e., the set of values $u \in \mathbb{R}$ for which $f_0(u) > 0$. This definition of the dyadic policy generalizes a definition provided in Jedynak et al. (2012) for single targets.

The dyadic policy is easy to implement, and is non-adaptive, allowing its use in parallel computing environments. Figure 1 illustrates this policy.

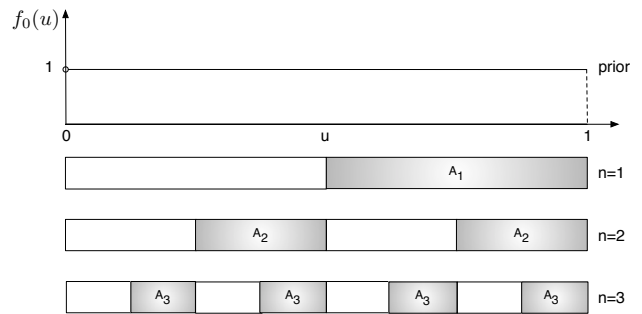


Figure 1. Illustration of the dyadic policy. The prior density f_0 displayed at top is uniform over $(0,1]$. The question set A_n is the union of the dark subsets for $n = 1, 2, 3$.

The following theorem provides an explicit expression for the expected entropy of the posterior distribution under the dyadic policy.

Theorem 2. *Under the dyadic policy π_D ,*

$$R(\pi_D, N) = H_0 - D_k N, \quad (11)$$

where

$$D_k = H \left(\frac{1}{2^k} \sum_{z=0}^k \binom{k}{z} f(\cdot|z) \right) - \frac{1}{2^k} \sum_{z=0}^k \binom{k}{z} H(f(\cdot|z)). \quad (12)$$

In the noiseless case, this simplifies to

$$D_k = H \left(\text{Bin} \left(k, \frac{1}{2} \right) \right), \quad (13)$$

the entropy of a Binomial distribution $\text{Bin}(k, \frac{1}{2})$.

This result is easier to interpret in a discrete setting. Consider, as we will in Section 4, an image of $M \times M$ pixels containing k instances of an object, located at random, uniformly and independently. The instances are our targets. The starting entropy, neglecting the fact that several instances might occupy the same location, is

$$H_0 = k \log M^2. \quad (14)$$

According to (11), the expected number of questions N^* such that the k targets are located with certainty when using the dyadic policy, i.e. $R(\pi_D, N^*) = 0$, is such that

$$N^* = \frac{k}{D_k} \log M^2. \quad (15)$$

Consider the noiseless case for simplicity. Firstly, N^* is negligible compared to the number of questions asked by a naive algorithm that queries the pixels in a fixed, predetermined order, for example line by line and column by column. This naive algorithm requires on average $\frac{M^2}{2}$ queries for a single target and more for more targets. Secondly, N^* is also better than querying optimally one target, which requires $\log M^2$ queries, and repeating this k times, for a total of $k \log M^2$ queries. Indeed,

$$N^* \sim \frac{2k}{\log \frac{\pi e k}{2}} \log M^2 \quad (16)$$

using the approximation of the Binomial distribution $B(k, \frac{1}{2})$ with the Normal distribution.

We can also compare the expression (11) for the expected entropy under the dyadic policy to the lower bound on the optimal expected entropy from Theorem 1. In both cases the expected entropy decreases linearly in the number of questions, with a reduction per question of D_k under the dyadic policy, and a reduction of C_k in the lower bound. This implies the following approximation guarantee for the entropy reduction under the dyadic policy, relative to optimal.

Corollary 1.

$$\frac{H_0 - R(\pi_D, N)}{H_0 - \inf_{\pi \in \Pi} R(\pi, N)} \geq \frac{D_k}{C_k}.$$

The approximation ratio¹ D_k/C_k depends upon the noise model and the number of targets k .

In the noiseless case, this approximation ratio is $H(\text{Bin}(k, \frac{1}{2}))/\log(k+1) \geq \frac{1}{2}$ (this inequality is shown in the supplement), showing that the dyadic policy is a 2-approximation in the noiseless case. Moreover, this approximation ratio approaches 1/2 as $k \rightarrow \infty$, showing that the dyadic policy does not achieve the lower bound from Theorem 1 for large values of k . However, as previously noted, this lower bound is not achievable. The precise value of an optimal policy remains unknown.

To support algorithms described in Section 4 that use the dyadic policy as a first phase procedure for deciding the order over pixels in which to call the oracle in computer vision applications, we introduce here some additional notation and derive an explicit formula for the posterior distribution over the targets θ given the history of the noiseless answers.

Consider a fixed n , where $1 \leq n \leq N$. For each binary sequence $s = \{s_1, \dots, s_n\}$, define

$$C_s = \left(\bigcap_{1 \leq j \leq n; s_j=1} A_j \right) \cap \left(\bigcap_{1 \leq j \leq n; s_j=0} A_j^c \right) \cap \text{supp}(f_0). \quad (17)$$

The collection $\mathcal{C} = \{C_s : C_s \neq \emptyset, s \in \{0, 1\}^n\}$ provides a partition of the support of f_0 . A history of n questions provides information on which sets C_s contain which targets among $\theta_{1:k}$.

For each $C \in \mathcal{C}$, let $N(C) = \sum_{i=1}^k 1\{\theta_i \in C\}$ be the number of targets in C .

We now provide a result that shows how to compute $E[N(C)|X_{1:N}]$, the expected number of targets within one of these sets C under the posterior distribution. This is used by the algorithms presented in Section 4, at the start of a second stage following the use of the dyadic policy, in which an expensive noise-free oracle is called on some of the small sets C to establish definitively the number of targets in each. The algorithms use the value $E[N(C)|X_{1:N}]$ to determine the order over sets in which to call the oracle.

Theorem 3. *For each instance θ_i and each $C \in \mathcal{C}$, the*

¹In the version of the paper that appeared at ICML, there was a typo in Corollary 1, in which the ratio D_k/C_k was reversed.

posterior likelihood $P(\theta_i \in C | X_{1:N} = x_{1:N})$ satisfies

$$P(\theta_i \in C | X_{1:N} = x_{1:N}) = \prod_{n=1}^N \left(\frac{e_n}{k}\right)^{s_n} \left(1 - \frac{e_n}{k}\right)^{1-s_n}. \quad (18)$$

where $e_n = E[Z_n | X_n = x_n]$ and $s_n = 1\{C \subseteq A_n\}$. Moreover,

$$E[N(C) | X_{1:N} = x_{1:N}] = kP(\theta_i \in C | X_{1:N} = x_{1:N}).$$

The quantity e_n can be computed according to Bayes rule:

$$\begin{aligned} e_n &= E[Z_n | X_n = x_n] = \sum_{j=0}^k jP(Z_n = j | X_n = x_n) \\ &= \sum_{j=0}^k \frac{jP(X_n = x_n | Z_n = j)P(Z_n = j)}{P(X_n = x_n)}, \end{aligned}$$

where $Z_n \sim \text{Bin}(k, \frac{1}{2})$, $P(X_n = x_n | Z_n = j)$ can be computed directly from the noise model (2), and $P(X_n = x_n) = \sum_{j=0}^k P(X_n = x_n | Z_n = j)P(Z_n = j)$.

4. ALGORITHMS AND EXPERIMENTS

We now show how the dyadic policy, analyzed above in the continuous setting using the entropy, can be used in an idealized computer vision setting to reduce the number of oracle calls required to locate k instances of a given object within a $M \times M$ digital image.

We consider algorithms that proceed in 2 phases, eventually iterated. The first phase consists in querying the dyadic sets. As opposed to the continuous domain, there is here a limited supply of dyadic sets. Choosing for M a power of 2, there are $\log M$ dyadic horizontal queries and $\log M$ dyadic vertical queries. Figure 2 presents the dyadic questions for $M = 16$. The second phase consists in ordering the pixels and querying the oracle according to this ordering. We compare three algorithms: Posterior Rank (PR), Iterated Posterior Rank (IPR) and Entropy Pursuit (EP). We will see that all these three algorithms significantly outperform the baseline algorithm—the Index Rank (IR) algorithm—in terms of the expected number calls to the oracle (see Figure 4).

The PR algorithm computes the expected number of instances $E[N(C) | X_{1:N}]$ in each pixel C using Theorem 3, orders the pixels in decreasing order of this quantity $E[N(C) | X_{1:N}]$, and runs oracle calls according to this order until all the instances are found. This algorithm is summarized below, and a detailed implementation is provided in the supplementary file.

Algorithm 1 Posterior Rank (PR) Algorithm

- 1: Compute the answers to the screening questions.
 - 2: Compute the posterior rank r according to (18).
 - 3: Run the oracle on the pixels according to r until all the instances are found.
-

The IPR algorithm is a variation of the PR Algorithm. As before, the pixels are searched in decreasing order of the expected number of instances. When the oracle locates an instance (or instances) at a pixel, the expected number of unlocalized instances at each pixel is recomputed using (20) as described below, and provides an updated ranking for the remainder of the search.

Computing the expected number of unlocalized instances at a pixel, given the locations of $0 \leq i < k$ previously localized instances, is most straightforward in the case of additive noise, i.e.,

$$h(Z_n, W_n) = Z_n + W_n \quad (19)$$

In this case, this computation is accomplished by *masking* the instances already found, i.e., by subtracting those instances localized in A_n from X_n , subtracting the overall number of localized instances from k , and recomputing using (18).

More generally, we re-use the expression (18), but replace the number of unlocalized instances k by $k' = k - i$, and alter e_n to account for those previously localized instances residing in the queried set A_n . Letting $N'(C) = N(C) - \sum_{j=1}^i 1\{\theta_j \in C\}$ be the number of unlocalized instances in C , we have

$$E[N'(C) | X_{1:N}, \theta_{1:i}] = k' \prod_{n=1}^N \left(\frac{e'_n}{k'}\right)^{s_n} \left(1 - \frac{e'_n}{k'}\right)^{1-s_n}. \quad (20)$$

Here, $e'_n = E[Z'_n | X_n = x_n, \theta_{1:i}]$, with $Z'_n = Z_n - \sum_{j=1}^i 1\{\theta_j \in C\}$ being the number of unlocalized instances in the queried set A_n . The quantity e'_n can be computed as,

$$e'_n = \sum_{j=0}^{k'} \frac{jP(X_n = x_n | Z'_n = j, \theta_{1:i})P(Z'_n = j)}{P(X_n = x_n | \theta_{1:i})},$$

where $Z'_n \sim \text{Bin}(k', \frac{1}{2})$, $P(X_n = x_n | Z'_n = j, \theta_{1:i})$ can be computed from the noise model (2), and $P(X_n = x_n | \theta_{1:i}) = \sum_{j=0}^{k'} P(X_n = x_n | Z'_n = j, \theta_{1:i})P(Z'_n = j)$.

In the special case when no instances have been localized, so $i = 0$, (20) recovers (18).

We now summarize the IPR algorithm:

Algorithm 2 Iterated Posterior Rank (IPR) Algorithm

- 1: Compute the answers to the screening questions.
- 2: **repeat**
- 3: Compute the posterior rank r according to (20).
- 4: Run the oracle on the pixels according to r until one (several) instance(s) is (are) found at a pixel.
- 5: **until** all the instances are found.

IPR’s Step 4 may request the oracle’s feedback on a pixel that was already queried in a previous stage. This is because we condition on previously localized instances, but not on previous negative reports from the oracle that there were no instances at a particular pixel. When this occurs, we simply report the oracle’s previous value, rather than re-running the oracle. We do not condition on all previous oracle results because computing the posterior expected number of instances is much more challenging computationally.

Figure 2 and 3 illustrate the procedures in the IPR algorithm for a 16×16 image with $k = 4$ instances. Figure 2 illustrates the screening questions under the dyadic policy, with light regions marking the questions sets. The first line of Figure 3 shows the true but unknown locations of the instances in each iteration of the IPR algorithm. The second line shows the expected number of instances within each pixel computed after screening questions in each iteration, respectively, with lighter regions having a higher expected number of instances.

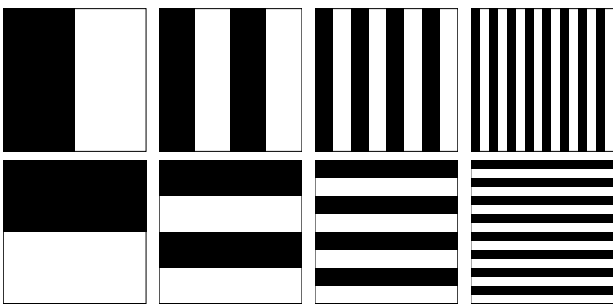


Figure 2. The queried regions under the dyadic policy for a 16×16 image shown in white.

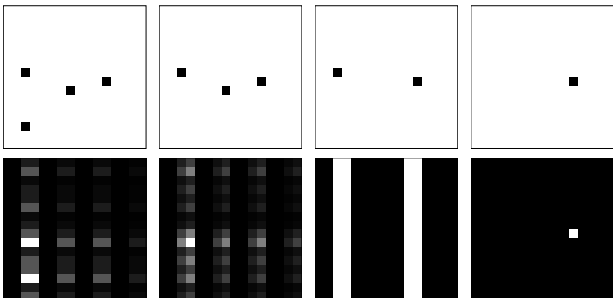


Figure 3. (row 1) Example image with 4 instances of the object initially, one instance is found after each iteration of the IPR algorithm. (row 2) The corresponding posterior distribution after each iteration. Light regions indicate pixels more likely to contain the object instance, while dark regions are less likely.

The Entropy Pursuit (EP) algorithm is a greedy algorithm aimed at reducing the expected entropy on the joint location of the instances. It has been studied and used for locating and tracking instances in (Sznitman & Jedynak, 2010; Jedynak et al., 2012; Geman & Geman, 1984; Sznitman et al., 2013b;a). This algorithm can be related to the IPR algorithm. The differences between EP and IPR are: i) EP uses a different ordering criterion; ii) EP updates the ordering each time after running the oracle at a pixel instead of after an instance being found. Specifically, EP computes for each pixel the expected entropy reduction in the distribution of the location of the instances which would be achieved by running the oracle at this pixel. It then selects the pixel for which this quantity is maximal. A detailed implementation is given in the supplementary file.

We use simulations to compare the performances of the three algorithms described above with a baseline algorithm, called Index Rank (IR). IR sweeps the image from left to right, top to bottom, until all the instances of the object are found. For the sake of simplicity, the object to be found in our simulation is a dot of size 1 pixel. We use 100 random assignments for the locations of the object instances for each k and each image size in the simulation, and measure the number of calls to the oracle required in each case.

4.1. Noiseless answers to the queries

We consider first the situation where the answers to the screening questions are noiseless, which is consistent with the theoretical analysis presented in the previous sections.

Figure 4, top row compares the algorithms for $k = 2$, $k = 3$ and $k = 10$ object instances for image sizes $\{8 \times 8, 16 \times 16, \dots, 1024 \times 1024\}$. Algorithms PR, IPR and EP require a smaller average number of calls to the oracle compared to the baseline IR. An example will show how dramatic this is for large size images. In the case of 1024×1024 pixel images and $k = 2$ instances, IR requires 2^{20} evaluations of the oracle while IPR requires less than 2^8 on average. IPR is also much more efficient than PR. IPR and EP show similar performances, however, IPR is superior to EP in terms of the computational complexity. Due to the EP algorithm’s large computational and memory requirements, we have only plotted EP for $k = 2$ and $k = 3$, and have only gone up to 512×512 image for $k = 3$.

4.2. Noisy answers to the queries

In an actual computer vision setting, screening questions would be answered by an image processing algorithm, trained using labeled data. These answers would then be noisy. We performed experiments to measure the effectiveness of the Posterior Rank (PR) and the Iterated Posterior Rank (IPR) algorithms in this case, while the oracle is still considered perfect. We use the additive model presented in

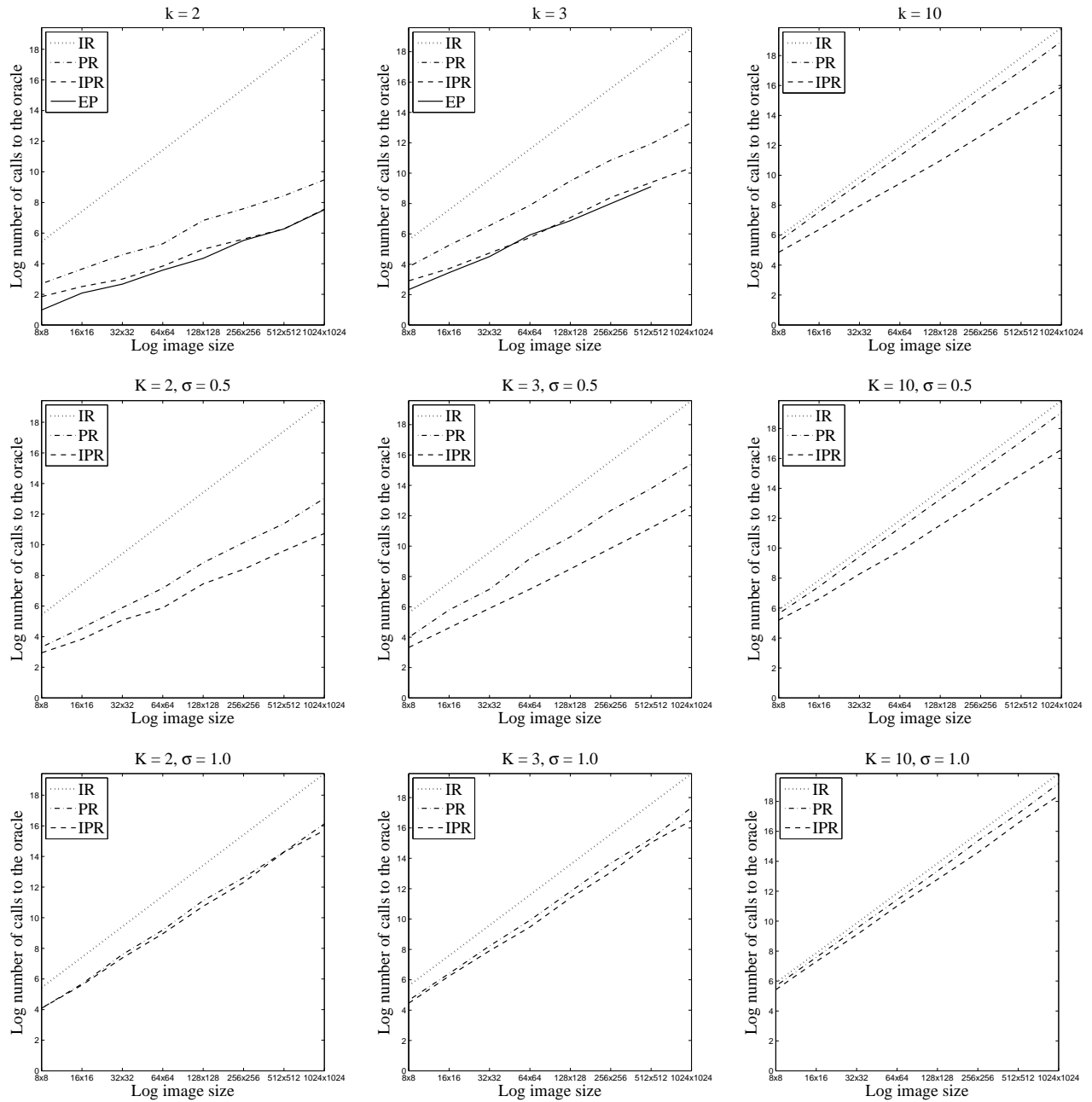


Figure 4. The mean number of calls to the oracle over 100 samples plotted against the image size for $k = 2$, $k = 3$ and $k = 10$ object instances using the algorithms described in section 4. (row 1): No noise in the screening answers. (row 2): Gaussian noise with $\sigma = 0.5$ in the screening answers. (row 3): Gaussian noise with $\sigma = 1.0$ in the screening answers.

(19) and we choose W_n to be independent, Normally distributed random variable with standard deviation σ . Figure 4, second and third row compares the Posterior Rank (PR), Iterated Posterior Rank (IPR) and the Index Rank (IR) algorithms for $k = 2$, $k = 3$ and $k = 10$ object instances for two levels of noise $\sigma = 0.5, 1$. We note that both algorithms outperform the default IR algorithm in all cases. The IPR algorithm is more robust to noise than the PR algorithm. As expected, the performances of both algorithms decrease as the amount of noise increases.

4.3. Face detection

To illustrate potential benefits of IP and IPR, we evaluate their performance in the context of finding faces in images. In particular, we consider how these strategies can be used as cost effective ways to determine regions where faces may be located such that high-performing (and computationally expensive) classifiers may then be more effectively be used.

We begin by training an extremely efficient but poorly-performing face classifier, with the intention of evaluating it at each location of an image. To this end, we train a Boosted classifier, as described in (Ali et al., 2012), but only with 50 stumps (*i.e.* 5% compared to state-of-the-art classifiers), using 4000 faces and 5 million background samples of size 30×30 . Once trained, we evaluated our poor-classifier at multiple scales on 35 images from the MIT+CMU face dataset. For each image location, a pixel was scored as the sum of weighted stump outputs of our trained classifier (*e.g.* Fig. 5 (2nd row) illustrate these scores: higher values in white and lower values in black). From these response image, we calculated the answers to the screening questions using an integral image representation followed by the PR algorithm (Alg. 1)

Fig. 5 depicts for 2 images with 4 faces each, the corresponding response maps for the poor classifier and the posterior distribution after a new face has been located using the PR algorithm (*i.e.* higher values in white and lower values in black).

No. of Pixels	Oracle Calls	Oracle Calls per face
12713984	347238	2671

Table 1. Results for PR algorithm on a dataset with 35 images containing 130 faces in total.

5. CONCLUSION

In this work, we have considered the problem of localizing several targets simultaneously. We have derived a close-to-optimal policy within a Bayesian framework using the expected entropy of the posterior as a value function. We

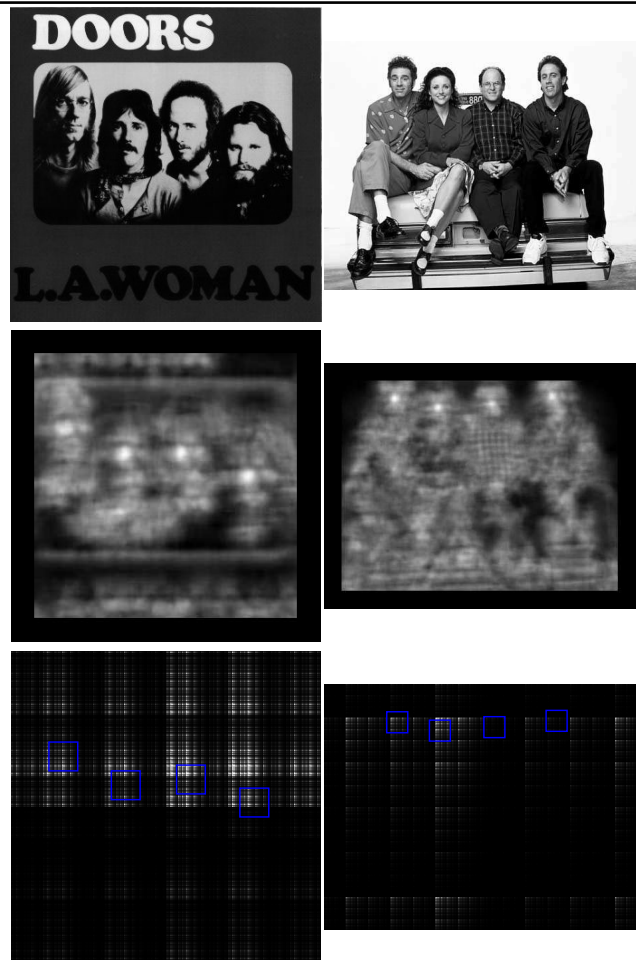


Figure 5. Face detection result. (row 1): 2 example images with 4 faces each. (row 2): Result from the poor-classifier. row(3): The posterior at found by the PR algorithm. Light regions indicate pixels more likely to contain the face, while dark regions are less likely. The locations of the faces detected are indicated as a blue rectangles.

have then empirically evaluated this policy for a toy problem related to the localization of several instances of the same object in computer vision. We have shown dramatic performance increases compared to a baseline method. We have also shown that the method is robust to a reasonable level of noise.

Acknowledgments

Peter Frazier was partially supported by NSF CAREER CMMI-1254298, NSF IIS-1247696, AFOSR FA9550-12-1-0200, AFOSR FA9550-15-1-0038, and the ACSF AVF. Bruno Jedynak was partially funded by NASA Early Stage Innovations Grant NNX14AB04G and by the Science of Learning Institute at the Johns Hopkins University for the award: Spatial Localization Through Learning: An Information Theoretic Approach. Purnima Rajan was supported

by NASA Early Stage Innovations Grant NNX14AB04G.

References

- Ali, K., Fleuret, F., Hasler, D., and Fua, P. A real-time deformable detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2), 2012.
- Burnashev, M V and Zigangirov, K S. On One Problem of Observation Control. *Problemy Peredachi Informatsii*, 11(3):44–52, 1975. ISSN 0555-2923.
- Buzas, J. and Warrington, G. Optimized random chemistry. Technical Report 1302.2895, ArXiv e-prints, 2013.
- Castro, R. and Nowak, R. Active sensing and learning. *Foundations and Applications of Sensor Management*, 2007.
- Chung, F., Graham, R., and Leighton, T. Guessing secrets. *The Electronic Journal of Combinatorics*, 8(13), 2001.
- Du, D. and Hwang, F. *Combinatorial Group Testing and its Applications*. World Scientific Pub Co, 2000.
- Eppstein, D., Goodrich, M, and Hirschberg, D. Improved combinatorial group testing algorithms for real-world problem sizes. *SIAM Journal on Computing*, 36:1360–1375, 2007.
- Geman, S. and Geman, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
- Harvey, N., Patrascu, M., Wen, Y., Yekhanin, S., and Chan, V. Non-adaptive fault diagnosis for all-optical networks via combinatorial group testing on graphs. In *IEEE International Conference on Computer Communications*, pp. 697–705, 2007.
- Horstein, M. Sequential transmission using noiseless feedback. *IEEE Transactions on Information Theory*, 9(3): 136–143, Jul 1963. ISSN 0018-9448. doi: 10.1109/TIT.1963.1057832.
- Jedynak, B., Frazier, P., and Sznitman, R. Twenty questions with noise: Bayes optimal policies for entropy loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:114–136, 2012.
- Kauffman, S. *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity*. Oxford University Press, 1996.
- Lee, D., Lee, K., Ho, W., and Lee, S. Target cell-specific involvement of presynaptic mitochondria in post-titanic potentiation at hippocampal mossy fiber synapse. *Journal of Neuroscience*, 27(50):13603–13613, 2007.
- Merchan-Perez, A., Rodriguez, J., Alonso-Nanclares, L., Schertel, A., and DeFelipe, J. Counting synapses using fib/sem microscopy: A true revolution for ultrastructural volume reconstruction. *Frontiers in Neuroanatomy*, 3 (18), 2009.
- Mortlock, D. Astronomy: The age of the quasars. *Nature*, 514:43–44, 2009.
- Porat, E. and Rothschild, A. *Altomata, Languages and Programming*, volume 5125, chapter Explicit Non-adaptive Combinatorial Group Testing Schemes, pp. 748–759. Springer, Berlin Heidelberg, 2008.
- Stinson, D., Trung, T., and Wei, R. Secure flameproof codes, key distribution patterns, group testing algorithms and related structures. *Journal of Statistical Planning and Inference*, 86:595–617, 2000.
- Sznitman, R. and Jedynak, B. Active testing for face detection and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1914–1920, 2010.
- Sznitman, R., Lucchi, A., Frazier, P., Jedynak, B., and Fua, P. An optimal policy for target localization with application to electron microscopy. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1–9, 2013a.
- Sznitman, R., Richa, R., Taylor, R., Jedynak, B., and Hager, G. Unified detection and tracking of instruments during retina microsurgery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1263–1273, 2013b.
- Tsiligkaridis, T., Sadler, B. M., and Hero, A. O. Collaborative 20 questions for target localization. *CoRR*, abs/1306.1922, 2013.
- Waeber, R., Frazier, P. I., and Henderson, S. G. Bisection search with noisy responses. *SIAM J. Control and Optimization*, 51(3):2261–2279, 2013.