Optimal discovery with probabilistic expert advice

Sébastien Bubeck[1], Damien Ernst[2] and Aurélien Garivier[3]

Abstract— **Motivated by issues of security analysis for power systems, we analyze a new problem, called optimal discovery with probabilistic expert advice. We address it with an algorithm based on the optimistic paradigm and the Good-Turing missing mass estimator. We show that this strategy attains the optimal discovery rate in a macroscopic limit sense, under some assumptions on the probabilistic experts. We also provide numerical experiments suggesting that this optimal behavior may still hold under weaker assumptions.**

I. Introduction

In this paper we consider the following problem: Let $\mathcal{X}$ be a set, and $A \subset \mathcal{X}$ be a set of interesting elements in $\mathcal{X}$. One can access $\mathcal{X}$ only through requests to a finite set of probabilistic experts. More precisely, when one makes a request to the ith expert, the latter draws independently at random a point from a fixed probability distribution P_i over $\mathcal{X}$. One is interested in discovering rapidly as many elements of A as possible, by making sequential requests to the experts.

A. Motivation

The original motivation for this problem arises from the issue of real-time security analysis of a power system. This problem often amounts to identifying in a set of credible contingencies those that may indeed endanger the security of the power system and perhaps lead to a system collapse with catastrophic consequences (e.g., an entire region, country may be without electrical power for hours). Once those dangerous contingencies have been identified, the system operators usually take preventive actions so as to ensure that they could mitigate their effect on the system in the likelihood they would occur. Note that usually, the dangerous contingencies are very rare with respect to the non dangerous ones. A straightforward approach for tackling this security analysis problem is to simulate the power system dynamics for every credible contingency so as to identify those that are indeed dangerous. Unfortunately, when the set of credible contingencies contains a large number of elements (say, there are more than 10^5 credible contingencies) such an approach may not possible anymore since the computational resources required to simulate every contingency may excess those that are usually available during the few (tens of) minutes available for the real-time security analysis. One is

therefore left with the problem of identifying within this short time-frame a maximum number of dangerous contingencies rather than all of them. The approach proposed in [FB11], [FFBW10] addresses this problem by building first very rapidly what could be described as a probability distribution P over the set of credible contingencies that points with significant probability to contingencies which are dangerous. Afterwards, this probability distribution is used to draw the contingencies to be analyzed through simulations. When the computational resources are exhausted, the approach outputs the contingencies found to be dangerous. One of the main shortcoming of this approach is that usually P points only with a significant probability to a few of the dangerous contingencies and not all of them. This in turn makes this probability distribution not more likely to generate after a few draws new dangerous contingencies than for example a uniform one. The dangerous contingencies to which P points to with a significant probability depend however strongly on the set of (sometimes arbitrary) engineering choices that have been made for building it. One possible strategy to ensure that more dangerous contingencies can be identified within a limited budget of draws would therefore be to consider $K > 1$ sets of engineering choices to build K different probability distributions $P_1, P_2, \ldots, P_K$ and to draw the contingencies from these K distributions rather than only from a single one. This strategy raises however an important question to which this paper tries to answer: how should the distributions be selected up for being able to generate with a given number of draws a maximum number of dangerous contingencies? We consider the specific case where the contingencies are sequentially drawn and where the distribution selected for generating a contingency at one instant can be based on the past distributions that have been selected, the contingencies that have been already drawn and the results of the security analyses (dangerous/non dangerous) for these contingencies. This corresponds exactly to the optimal discovery problem with expert advice described above. We believe that this framework has many possible applications, such as web-based content access.

B. Setting and notation

In this paper we restrict our attention to finite or countably infinite sets $\mathcal{X}$. We denote by K the number of experts. For each $i \in \{1, \ldots, K\}$, we assume that $(X_{i,n})_{n \geq 1}$ are random variables with distribution P_i such that the $(X_{i,n})_{i,n}$ are independent. Sequential discovery with probabilistic expert advice can be described as follows: at each time step $t \in \mathbb{N}^*$, one picks an index $I_t \in \{1, \ldots, K\}$, and one observes

[1]S. Bubeck is with the Department of Operations and Financial Engineering, Princeton University, Princeton, NJ 08544, USA sbubeck@princeton.edu

[2]D. Ernst is with Université of Liege, Liegem B-4000, Belgium dernst@ulg.ac.be

[3]A. Garivier is with LTCI, CNRS and Telecom ParisTech, Paris, France garivier@telecom-paristech.fr

$X_{I_t,n_{I_t,t}}$, where

$$n_{i,t} = \sum_{s \le t} \mathbb{1}_{\{I_s = i\}} \ .$$

The goal is to choose the $(I_t)_{t \ge 1}$ so as to observe as many elements of A as possible in a fixed horizon t, or equivalently to observe all the elements of A within as few time steps as possible. The index I_{t+1} may be chosen according to past observations: it is a (possibly randomized) function of $(I_1, X_{1,I_1}, \ldots, I_t, X_{I_t,n_{I_t,t}})$. We are mainly interested in the number of interesting items found by the strategy after t time steps:

$$\sum_{x \in A} \mathbb{1}\left\{ x \in \{X_{1,1}, \ldots, X_{1,n_{1,t}}, \ldots, X_{K,1}, \ldots, X_{K,n_{K,t}}\} \right\}.$$

Note in particular that it does not help to observe twice the same interesting item.

While Algorithm Good-UCB, presented in Section II, can be used in a more general setting (as illustrated in Section III), for the mathematical analysis we restrict our attention to the case of probabilistic experts with the following properties:

(i) non-intersecting supports: $A \cap \mathrm{supp}(P_i) \cap \mathrm{supp}(P_j) = \emptyset$ for $i \ne j$,

(ii) finite supports with the same cardinality: $|\mathrm{supp}(P_i)| = N, \forall i \in \{1, \ldots, K\}$,

(iii) uniform distributions: $P_i(x) = \frac{1}{N}, \forall x \in \mathrm{supp}(P_i), \forall i \in \{1, \ldots, K\}$.

These asumptions are made in order to be able to compare the performance of the Good-UCB algorithm to an "oracle" policy, i.e., a virtual algorithm that would be aware, at each time, of the probability of each item under each distribution, and would thus be able to sample optimally. Indeed, here such an oracle strategy consists in making a request, at each time step, to one of the experts with the highest number of still undiscovered interesting items.

In this setting it is convenient to reparametrize slightly the problem (in particular we make explicit the dependency on N for reasons that will appear later). Let $\mathcal{X}^N = \{1, \ldots, K\} \times \{1, \ldots, N\}$, $A^N \subset \mathcal{X}^N$ the set of interesting items of $\mathcal{X}^N$, and $Q^N = |A^N|$ the number of interesting items. We assume that, for expert $i \in \{1, \ldots, K\}$, P_i^N is the uniform distribution on $\{i\} \times \{1, \ldots, N\}$. We also denote by $Q_i^N = \left|A^N \cap (\{i\} \times \{1, \ldots, N\})\right|$ the number of interesting items accessible through requests to expert i.

C. Main result

This paper contains the description of a generic algorithm for the optimal discovery problem with probabilistic expert advice, and a theoretical result of optimality in a particular setting. In Section II, we first depict our strategy, termed Good-UCB. This algorithm relies on the *optimistic paradigm* (which led to the UCB (Upper Confidence Bound) algorithm for multi-armed bandits, [ACBF02]), and on a finite-time analysis of the Good-Turing estimator for the missing mass. In order to analyze and quantify the performance of this strategy, we compare it with the *oracle* policy described above.

The analysis is performed under the non-intersecting and uniform draws assumptions [(i), (ii), (iii)] described above, and in a macroscopic limit sense, that is when the size of the set $\mathcal{X}$ grows to infinity while maintaining a constant proportion of interesting items. First we show that one can assess the performance of a strategy in this macroscopic limit thanks to the following theorem. Let $F^N(t)$ be the number of interesting items found by the oracle policy after t time steps.

Theorem 1: Assume that, for all $i \in \{1, \ldots, K\}, Q_i^N/N$ converges to $q_i \in (0, 1)$ as N goes to infinity. Then, almost surely, the sequence of mappings $t \mapsto F^N([Nt])/N$ converges uniformly on $\mathbb{R}_+$ to a limit denoted F as N goes to infinity.

The proof of this theorem (as well as all the other results in this paper) can be found in the extended version [BEG11]. Moreover, explicit expressions for the limit F can also be found in [BEG11].

Our main result is that Good-UCB is a macroscopically optimal policy, that is, the performances of Good-UCB tends to the performances of the oracle policy. More precisely let $\tilde{F}^N(t)$ be the number of interesting items found by Good-UCB after t time steps.

Theorem 2: Assume that, for all $i \in \{1, \ldots, K\}, Q_i^N/N$ converges to $q_i \in (0, 1)$ as N goes to infinity. Then, almost surely, the sequence of mappings $t \mapsto \tilde{F}^N([Nt])/N$ converges uniformly on $\mathbb{R}_+$ to the limiting proportion F found during the same time by the oracle policy.

These results are illustrated in Section III by a few simulations. In particular the experiments show that the efficiency of Good-UCB is spectacular, and applies more generally than under the assumptions [(i), (ii), (iii)].

II. THE GOOD-UCB ALGORITHM

We describe here the Good-UCB strategy. This algorithm is a sequential method estimating at time t, for each expert $i \in \{1, \ldots, K\}$, the total probability of the interesting items that remain to be discovered through requests to expert i. This estimation is done by adapting the so-called Good-Turing estimator for the missing mass. Then, instead of simply using the distribution with highest estimated missing mass, which proves hazardous, we make use of the *optimistic paradigm* (see [Agr95], [ACBF02] and references therein), a heuristic principle well-known in reinforcement learning, which entails to prefer using an *upper-confidence bound* (UCB) of the missing mass instead. At a given time step, the Good-UCB algorithm simply makes a request to the expert with highest upper-confidence bound on the missing mass at this time step. We start with the Good-Turing estimator and a brief study of its concentration properties. Then we describe precisely the Good-UCB strategy.

A. Estimating the missing mass

Our algorithm relies on an estimation at each step of the probability of obtaining a new interesting item by making a

request to a given expert. A similar issue was addressed by I. Good and A. Turing as part of their efforts to crack German ciphers for the Enigma machine during World War II. In this subsection, we describe a version of the Good-Turing estimator adapted to our problem. Let Ω be a discrete set, and let A be a subset of interesting elements of Ω. Assume that $X_1, \ldots, X_n$ are elements of Ω drawn independently under the same distribution P, and define for every $x \in \Omega$:

$$O_n(x) = \sum_{m=1}^{n} \mathbb{1}\{X_m = x\},$$
$$Z_n(x) = \mathbb{1}\{O_n(x) = 0\},$$
$$U_n(x) = \mathbb{1}\{O_n(x) = 1\}.$$

Let $p_{\max} = \max\{P(x) : x \in \Omega\}$, let $R_n = \sum_{x \in A} Z_n(x) P(x)$ denote the missing mass of the interesting items, and let $U_n = \sum_{x \in A} U_n(x)$ the number of elements of A that have been seen exactly once (in linguistics, they are often called *appaxes*). The idea of the Good-Turing estimator ([Goo53], see also [MS00], [OSZ03] and references therein) is to estimate the (random) "missing mass" R_n, which is the total probability of all the interesting items that do not occur in the sample $X_1, \ldots, X_n$, by the "fraction of appaxes" $\hat{R}_n = U_n/n$. This estimator is well-known in linguistics, for instance in order to estimate the number of words in some language [GS95]. For our particular needs, we derive (using similar techniques as in [MS00]) the following upper-bound on the estimation error:

Proposition 1: With probability at least $1 - \delta$,

$$\hat{R}_n - \frac{1}{n} - \sqrt{\frac{(2/n + p_{\max})^2 n \log(2/\delta)}{2}}$$
$$\leq R_n$$
$$\leq \hat{R}_n + \sqrt{\frac{(2/n + p_{\max})^2 n \log(2/\delta)}{2}}$$

B. The Good-UCB algorithm

Following the example of the well-known Upper-Confidence Bound procedure for multi-armed bandit problems, we propose Algorithm 1, which we call Good-UCB in reference to the estimated procedure it relies on. For every arm $i \in \{1, \ldots, K\}$ and for every $t \in \mathbb{N}$, denote

$$O_{i,t}(x) = \sum_{s=1}^{n_{i,t}} \mathbb{1}\{X_{i,s} = x\},$$
$$O_t(x) = \sum_{i=1}^{K} O_{i,t}(x),$$
$$U_{i,t}(x) = \mathbb{1}\{O_{i,t}(x) = O_t(x) = 1\}, \quad U_{i,t} = \sum_{x \in A} U_{i,t}(x).$$

For each arm $i \in \{1, \ldots, K\}$, the index at time t is composed of the estimate

$$\hat{R}_{i,t-1} = \frac{U_{i,t-1}}{n_{i,t-1}}$$

of the missing mass

$$\sum_{x \in A \setminus \{X_{I_1, n_{I_1,1}}, \ldots, X_{I_{t-1}, n_{I_{t-1}, t-1}}\}} P_i^N(x)$$

inflated by a confidence bonus of order $\sqrt{\log(t)/n_{i,t-1}}$. Good-UCB relies on a tuning parameter c which is discussed below.

Algorithm 1 Good-UCB
1: For $1 \leq t \leq K$ choose $I_t = t$.
2: **for** $t \geq K + 1$ **do**
3:    Choose $I_t = \arg\max_{1 \leq i \leq K}\left\{\hat{R}_{i,t-1} + c\sqrt{\frac{\log(t)}{n_{i,t-1}}}\right\}$
4:    Observe X_t distributed as P_{I_t} and update $\hat{R}_{1,t}, \ldots, \hat{R}_{K,t}$ accordingly
5: **end for**

Note that the Good-UCB algorithm is designed for more general probabilistic experts than those satisfying assumptions [(i), (ii), (iii)]. In particular since we do not make the non-intersecting supports assumption (i), the missing mass of a given expert i depends explicitly on the outcomes of *all* requests (and not only requests to expert i). Note also that the bounds of Proposition 1 hold for all discrete distributions. The experiments of Section III validate these observations, and show that Good-UCB behaves very well even when assumptions [(i), (ii), (iii)] are not met. However, for the theoretical analysis of our algorithm, we focus on large values of N under the non-intersecting and uniform draws assumptions [(i), (ii), (iii)]: indeed in that case the performance of the oracle strategy is simple and deterministic, so that the optimality of the Good-UCB algorithm can be analyzed. More precisely, Theorem 2 shows that, in the macroscopic limit, the number of items found at each time by Good-UCB converges to the number of items found by the closed-loop oracle strategy that knows the number of interesting items to find with each expert, at every time, and that may use this information to make its choice.

III. SIMULATIONS AND CONCLUSION

We provide a few simulations illustrating the behaviour of the Good-UCB algorithm in practice. In order to illustrate the convergence properties shown in Section I-C, we first consider an example with $K = 7$ different sampling distributions satisfying assumptions [(i),(ii),(iii)], with respective proportions of interesting items $q_1 = 51.2\%, q_2 = 25.6\%, q_3 = 12.8\%, q_4 = 6.4\%, q_5 = 3.2\%, q_6 = 1.6\%$ and $q_7 = 0.8\%$. Figure 1 displays the number of items found as a function of time by the Good-UCB (solid), the oracle (dashed) and a balanced sampling scheme simply alternating between experts (dotted). The results are presented for sizes $N = 128, N = 500, N = 1000$ and $N = 10000$, each time for one representative run (averaging over different runs removes the interesting variability of the process). The convergence of Theorem 2 is obvious. Moreover, it can be seen that even for very moderate values of N the Good-UCB, although clearly distanced by the oracle, significantly outperforms uniform sampling.

For these simulations, the parameter c of Algorithm Good-UCB has been taken equal to $1/2$, which is a rather conservative choice. In fact, it appears that during all rounds of

all runs all upper-confidence bounds did contain the actual missing mass. Of course, a bolder choice of c can only improve the performance of the algorithm, as long as the confidence level remains sufficient.

In order to illustrate the efficiency of the Good-UCB algorithm in a more difficult setting, which does not satisfy any of the assumptions (i), (ii) and (iii), we also considered the following (artificial) example: $K = 5$ probabilistic experts draw independent sequences of geometrically distributed random variables, with expectations $100, 300, 500, 700$ and 900 respectively. The set of interesting items is the set of prime numbers. We compare the oracle policy, Good-UCB and uniform sampling. The results are displayed in Figure 2. Even if the difference remains significative between Good-UCB and the oracle, the former still performs significantly better that uniform sampling during the entire discovery process. In this example, choosing a smaller parameter c seems to be preferable; this is due to the fact that the proportion of interesting items on each arm is low; in that case, one can show by using tighter concentration inequalities that the concentration of the Good-Turing estimator is actually better than suggested by Proposition 1. In fact, this experiment suggests that the value of c should be chosen smaller when the remaining missing mass is small.

Perspectives

This paper introduced an original problem, optimal discovery with probabilistic expert advice. We proposed an algorithm to solve this problem, and showed its efficiency under some assumptions on the experts.

This work can be extended along several directions. First, it would be interesting to analyze the behaviour of Good-UCB under less restrictive assumptions on the experts. Note that assumptions (ii) and (iii) are used only to (considerably) simplify the analysis of the oracle policy, and hence to prove the optimality of Good-UCB. Removing assumption (ii) seems fairly straightforward up to the addition of another level of notations. On the other hand it is not clear how the analysis could be carried out if assumption (iii) were to be removed, though it seems reasonable to assume that Good-UCB will still be macroscopically optimal. Assumption (i) seems to be the most challenging to remove. Good-UCB is designed to work even when assumption (i) is not satisfied, but the analysis becomes more complicated because of explicit dependency between the missing mass of the different experts.

Second, we may wonder whether it would be possible to obtain optimal *rates of convergence* (in the macroscopic limit sense) for this problem, and whether Good-UCB is optimal in that sense too.

Finally, another macrosopic limit deserves to be investigated, where the number of interesting items for each arm remains constant, while N and n go to infinity; then, a Poisson regime appears. The analysis of Good-UCB might be possible by using a better concentration bound for the Good-Turing estimator such as the Boucheron-Massart-Lugosi inequality [BLM09]. This could also contribute to explain

Fig. 1. Number of items found by Good-UCB (solid), the oracle (dashed), and uniform sampling (dotted) as a function of time for sizes $N = 128, N = 500, N = 1000$ and $N = 10000$ in a 7-experts setting.
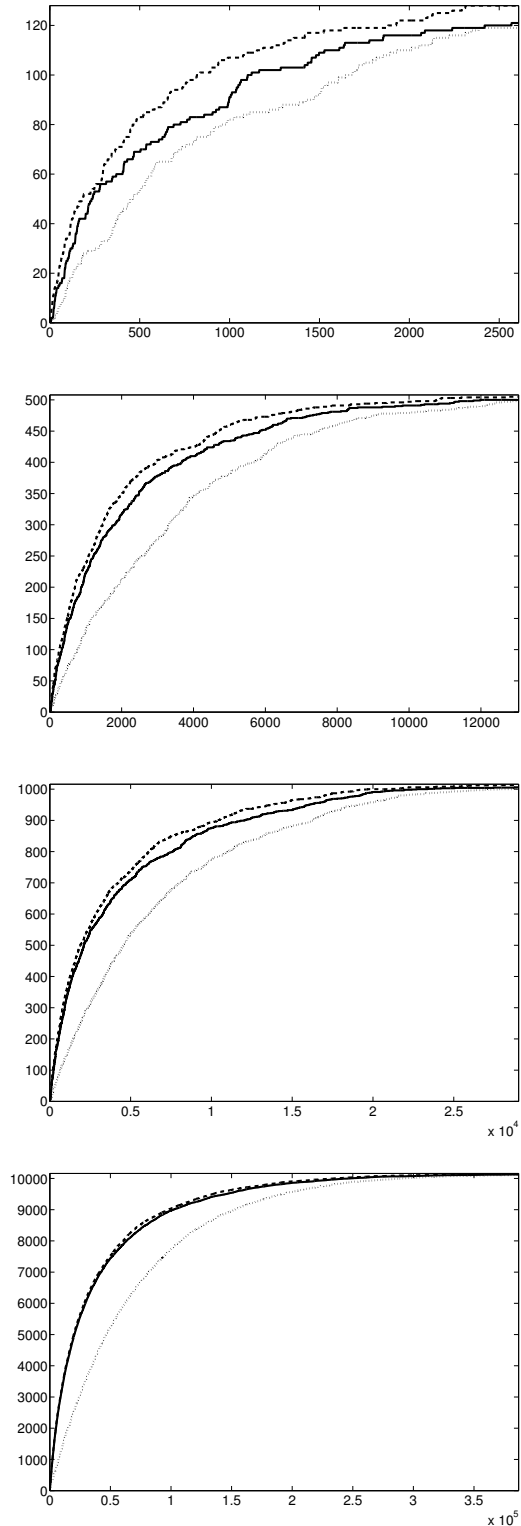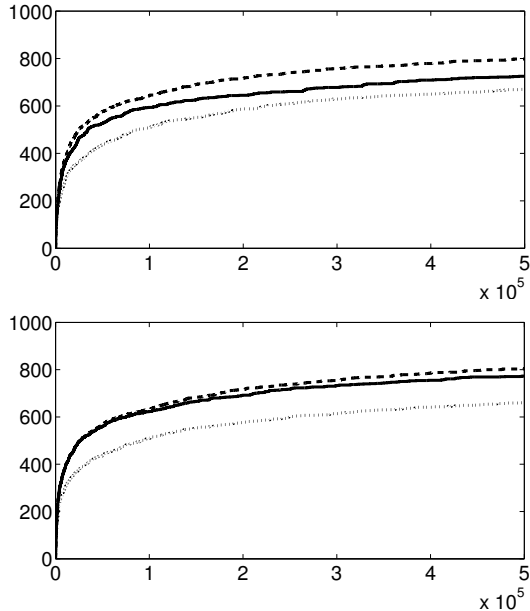
Fig. 2. Number of prime numbers found by Good-UCB (solid), the oracle (dashed), and uniform sampling (dotted) as a function of time, using geometric experts with means $100, 300, 500, 700$ and 900, for $c = 0.1$ (left) and $c = 0.02$ (right).

why, in the second experiment presented in Section III, the parameter c should be chosen decreasing with time.

REFERENCES

[ACBF02] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

[Agr95] R. Agrawal. Sample mean based index policies with O(log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995.

[BEG11] S. Bubeck, D. Ernst, and A. Garivier. Optimal discovery with probabilistic expert advice. *Arxiv preprint arXiv:1110.5447*, 2011.

[BLM09] Stéphane Boucheron, Gábor Lugosi, and Pacal Massart. On concentration of self-bounding functions. *Electron. J. Probab.*, 14:no. 64, 1884–1899, 2009.

[FB11] F. Fonteneau-Belmudes. *Identification of dangerous contingencies for large scale power system security assessment*. PhD thesis, University of Liège, 2011.

[FFBW10] C. Druet P. Panciatici F. Fonteneau-Belmudes, D. Ernst and L. Wehenkel. Consequence driven decomposition of large-scale power system security analysis. In *Proceedings of the 2010 IREP Symposium - Bulk Power Systems Dynamics and Control - VIII*, Buzios, Rio de Janeiro, Brazil, 2010.

[Goo53] I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953.

[GS95] William A. Gale and Geoffrey Sampson. Good-turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 2(3):217–237, 1995.

[MS00] David A. McAllester and Robert E. Schapire. On the convergence rate of good-turing estimators. In *COLT*, pages 1–6, 2000.

[OSZ03] Alon Orlitsky, Narayana P. Santhanam, and Junan Zhang. Always Good Turing: Asymptotically Optimal Probability Estimation. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 179+, Washington, DC, USA, 2003. IEEE Computer Society.