JMLR: Workshop and Conference Proceedings vol 49:1–6, 2016

On the Capacity of Information Processing Systems (Extended Abstract)

Laurent Massoulie LAURENT.MASSOULIE@INRIA.FR *Microsoft Research-Inria Joint Centre, 91120 Palaiseau, France*

Kuang Xu KUANGXU@STANFORD.EDU *Stanford Graduate School of Business, Stanford, CA 94305, USA*

Abstract

We propose and analyze a family of *information processing systems*, where a finite set of experts or servers are employed to extract information about a stream of incoming jobs. Each job is associated with a hidden label drawn from some prior distribution. An inspection by an expert produces a noisy outcome that depends both on the job's hidden label and the type of the expert, and occupies the expert for a finite time duration. A decision maker's task is to dynamically assign inspections so that the resulting outcomes can be used to accurately recover the labels of all jobs, while keeping the system stable. Among our chief motivations are applications in crowd-sourcing, diagnostics, and experiment designs, where one wishes to efficiently discover the nature of a large number of items, using a finite pool of computational resources or human agents.

We focus on the *capacity* of such an information processing system. Given a level of accuracy guarantee, we ask how many experts are needed in order to stabilize the system, and through what inspection architecture. Our main result provides an adaptive inspection policy that is asymptotically optimal in the following sense: the ratio between the required number of experts under our policy and the theoretical optimal converges to one, as the probability of error in label recovery tends to zero. [1]

Keywords: sequential hypothesis testing, stochastic resource allocation, information processing system, fluid model.

1. Introduction

An increasing number of modern processing systems has been designed and deployed for the purpose of learning and information extraction. In these applications, which we refer to broadly as information processing systems, a group of experts or servers is tasked with performing noisy inspections on a large collection of jobs, with the objective of uncovering some hidden features associated with each job up to a level of desirable accuracy. For instance, in crowd-sourcing (Karger et al. (2014)), a collection of images is dispatched to a group of human agents, where an agent attaches a label to each assigned image based on her own judgment. A decision maker then aggregates the agents' responses to produce a "best" label for each image. Similar learning tasks can also arise in medical diagnostics (Gerdtz and Bucknall (2001)), where medical data of patients is reviewed by physicians or nurses with different domains of expertise, with the goal of correctly identifying the patients' diseases, or in quality management (Baker and von Beers (1996)), where a set of prod-

1. Extended abstract. Full version appears as Massoulie and Xu (2016), http://arxiv.org/abs/1603.00544.

ucts undergoes a number of different tests performed by specialized machines, to identify whether a product is faulty and the type of fault it contains.

The presence of **resource constraints** is a crucial feature shared across many of these systems: the amount of processing resources, such as human agents, machines, or computer servers, is finite, and yet, each inspection or test requires the corresponding resource to commit a non-trivial amount of effort. This raises a natural question:

How much information can we extract using a finite amount of processing resources?

The main objective of the present work is to address this question, and gain understanding of the "capacity" of an information processing system. We will approach this problem by studying the *minimum number of experts* needed in order to learn a sufficient amount of information about *every* job in a stream of arrivals, while ensuring system stability. [2]

2. Model and Metrics

Our system operates in continuous time, and consists of m experts (system size), where each expert has a *type* (expertise) that belongs to a finite set, $\mathcal{K}$, and the fraction of type-k experts is ρ_k (expert mixture). The system receives a stream of incoming jobs arriving according to a Poisson process of rate one, where the ith job is associated with a random *label*, H_i, taking values in a finite set $\mathcal{H}$. The labels are drawn i.i.d. according to some prior distribution, π, and are hidden from the decision maker.

An atomic unit of processing in our system is called an *inspection*: the decision maker may assign an expert of type k to perform an inspection on a job, which occupies the expert for a random period of time that is exponentially distributed with mean μ_k (inspection rate), during which the expert cannot inspect other jobs. The inspection leads to a noisy *outcome*, whose distribution, $p(k, h, \cdot)$, depends on both the type of the expert, $k \in \mathcal{K}$, and the true label of the job under inspection, $h \in \mathcal{H}$.[3]

At any time t, the decision maker can choose to let a job i depart from the system, at which point she must produce a *classification*, $\widehat{H}_i$, representing her belief of job i's true label. We say that there is an *error* if the classification does not match the true label, i.e., if $\widehat{H}_i \neq H_i$. The high-level goal of the decision maker is to assign inspections intelligently, and use the resulting outcomes to produce, for each job, a correct classification of its hidden label with a small probability of error.

The real-time operation of the system is controlled by an *inspection policy*, ψ. At any time t, the policy has the ability to: (1) let an idle expert initiate an inspection on a job; (2) let a job depart from the system, in which case the policy will have to produce a classification for the job's label. An inspection policy that does not require the knowledge of the prior distribution, π, is said to be *prior-oblivious*. There are two main performance criteria for an inspection policy that are of interest:

2. We will use the term "expert" to refer to a single unit of processing resource, with the understanding that it may represent a computer server, testing machine, or human agent, depending on the application.
3. The family of outcome distributions, $\{p(k, h, \cdot)\}_{k\in\mathcal{K}, h\in\mathcal{H}}$, is assumed to satisfy the following two properties, but can otherwise be arbitrary (see Massoulie and Xu (2016) for details): (1) no single outcome can distinguish two job labels with absolute certainty; (2) for any two distinct job labels, $h, h' \in \mathcal{H}$, there exists at least one expert type, $k \in \mathcal{K}$, such that $p(k, h, \cdot)$ and $p(k, h', \cdot)$ are distinct.

1. Accuracy: We would like an inspection policy to accurately recover the true labels of all jobs. Given an accuracy parameter, $\delta > 0$, we say that a policy is **δ-accurate**, if for all $h \in \mathcal{H}$ and $i \in \mathbb{N}$, we have that $\mathbb{P}(\widehat{H}_i = h \,\big|\, H_i = h) \geq 1 - \delta$.

2. Stability: We would like all jobs to receive a classification in finite time. Denote by $Q(t)$ the total number of jobs in the system at time t. We say that the system is **stable** under an inspection policy, if the resulting process $\{Q(t)\}_{t \in \mathbb{R}_+}$ is positive recurrent.

The main goal of this paper is to understand how to design δ-accurate inspection policies that can stablize the system with the least number of experts (m). This inspires the following notion of resource efficiency, where we compare the number of experts required under a specific inspection policy against that of a theoretical optimal, as follows. Fix an expert mixture, $\{\rho_k\}_{k \in \mathcal{K}}$, inspection rates, $\{\mu_k\}_{k \in \mathcal{K}}$, and outcome distributions, $\{p(h, k, \cdot)\}_{h \in \mathcal{H}, k \in \mathcal{K}}$. For a δ-accurate inspection policy, ψ, define $m_\psi(\delta, \pi)$ as the number of experts required under ψ in order to ensure stability:

$$m_\psi(\delta, \pi) = \min\{m \in \mathbb{N} : \text{given } \delta \text{ and } \pi, \text{ a system with } m \text{ experts is stable under } \psi\}. \tag{1}$$

Given prior distribution π and $\delta > 0$, we define $m^*(\delta, \pi)$ as the smallest number of experts for which there exists a δ-accurate inspection policy that stabilizes the system. That is, $m^*(\delta, \pi)$ represents the minimal amount of processing resources required to ensure stability under an "optimal" inspection policy. The following definition serves as our main performance metric.

Definition 1 *We say that an inspection policy, ψ, is* **resource efficient**, *if*

$$\limsup_{\delta \to 0} \frac{m_\psi(\delta, \pi)}{m^*(\delta, \pi)} = 1, \quad \textit{for all prior distribution, } \pi. \tag{2}$$

We say that ψ is **strongly resource efficient**, *if the above convergence occurs uniformly over all prior distributions:*

$$\limsup_{\delta \to 0} \sup_{\pi} \frac{m_\psi(\delta, \pi)}{m^*(\delta, \pi)} = 1. \tag{3}$$

3. Main Result

Our main result provides a prior-oblivious policy that asymptotically achieves the optimal minimum system size in the regime where the accuracy parameter, δ, tends to zero.

Theorem 2 *Fix an expert mixture, $\{\rho_k\}_{k \in \mathcal{K}}$, inspection rates, $\{\mu_k\}_{k \in \mathcal{K}}$, and outcome distributions, $\{p(h, k, \cdot)\}_{h \in \mathcal{H}, k \in \mathcal{K}}$. There exists a prior-oblivious, strongly resource efficient inspection policy, ψ. In particular, there exist $c_0, \delta_0 > 0$, such that*

$$\sup_{\pi} \frac{m_\psi(\delta, \pi)}{m^*(\delta, \pi)} \leq 1 + c_0 \sqrt{\frac{\ln \ln(1/\delta)}{\ln(1/\delta)}}, \quad \forall \delta \in (0, \delta_0). \tag{4}$$

The inspection policy in Theorem 2 will be given explicitly as part of the proof, and it also inspires a much simpler heuristic policy (see Appendix of Massoulie and Xu (2016)). We highlight two important features of the theorem. First, the inspection policy is strongly resource efficient, which implies that its performance guarantee in comparison to the theoretical optimal holds *independently* of the prior distribution. Second, the inspection policy is *prior-oblivious* so that it can operate without any knowledge of the prior distribution of the job labels. This feature is especially important for our problem, since knowing the prior distribution would have likely required first learning the labels of the incoming jobs, which is the very task that we are trying to solve. Moreover, because a prior-oblivious policy automatically adapts to any prior distribution, it is also more robust against any shifts in the prior distribution over time, which can occur in many applications.

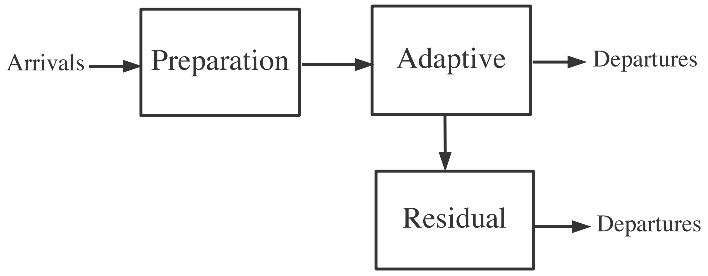

Figure 1: Overall architecture associated with the proposed inspection policy.

4. Inspection Policy and Proof Ideas

We now describe, at a high level, the inspection policy and proof techniques used to show Theorem 2. A complete proof can be found in Massoulie and Xu (2016).

Inspection policy. The main structure of our inspection policy is driven by two intrinsic features of the problem:

1. Expert heterogeneity. Different types of experts can have different expertise, which is reflected in their distinct outcome distributions, $\{p(k, h, \cdot)\}_{k\in\mathcal{K},h\in\mathcal{H}}$. In particular, for a pair of job labels, one expert type may be more efficient than another in telling them apart. Therefore, it is important for an inspection policy to quickly obtain “rough guesses” for the job labels early on, so that it can leverage these guesses to assign the majority of the inspections in an efficient manner.

2. Resource contention. The second feature is more nuanced and stems from the combined effect of the resource constraint and expert heterogeneity: to utilize the experts most efficiently, the “optimal” course of inspections for an individual job does not only depend on its true label, but also on the *prior distribution* governing the proportions of labels among its fellow jobs. This is due to the contention among jobs as they “compete” for the shared processing resources, and the mixture of inspections that a job receives may shift as the prior distribution changes. Therefore, a good inspection policy cannot be overly centered around individual jobs, and must be aware of the overall arrival pattern.

Our inspection policy makes use of a three-stage architecture, as is illustrated in Figure 1.

1. In the first stage (Preparation), the policy “bootstraps”[4] each incoming job, by performing a small number of inspections using randomly chosen experts, with the goal of generating a *coarse estimate* of its true label.

2. In the second stage (Adaptive), the policy performs the majority of the inspections in an adaptive manner, with the goal of verifying whether the coarse estimates are correct. For most of the jobs with a correct coarse estimate, the Adaptive stage “boosts” their classification

4. The term is not related to the bootstrap re-sampling method in statistics.

accuracy to $1 - \delta$, and allows them to depart from the system. The adaptivity of this stage addresses simultaneously the two features mentioned earlier: the combination of inspections to be received by job i is determined dynamically by solving a linear optimization problem, whose parameters depend on (1) the coarse label estimate of job i from the Preparation stage, which allows for leveraging expert heterogeneity, and (2) what other jobs are currently in the system, which coordinates the contention for processing resources among the jobs.

3. The third stage (Residual) treats those jobs whose coarse estimates were erroneous to ensure that they, too, will receive an accurate classification. The inspections in this stage are carried out in the same fashion as in the Preparation stage, i.e., by giving each job a fixed number of inspections using randomly chosen experts.

At a high level, we can also interpret this three-stage architecture through a learning versus verification dichotomy: all jobs are first inspected by some "generalists" (i.e., random experts) to *learn* a coarse label estimate. The system then enlists the "specialists" to *verify* the validity of these estimates to a high accuracy. If a coarse estimate is deemed incorrect by the "specialists," the job is then sent *back* to the "generalists" to perform learning thoroughly to reach an accurate estimate, albeit in a less efficient manner.

Proof techniques. The proof of Theorem 2 consists of two main components. First, we employ a linear program to derive a lower-bound on the minimum system size that must be satisfied by *any* stable δ-accurate inspection policy. In the second part, we show that our proposed inspection policy is δ-accurate, and is stable under a system size that asymptotically matches the lower-bound derived earlier, hence proving Theorem 2.

The key idea of the proof is to make use of a connection between *information need* and *service requirement*: inspired by Wald (1945) and Chernoff (1959), we show that obtaining accurate classifications is essentially equivalent to meeting a certain requirement on cumulative log-likelihood ratios, which can be further viewed as a form of *workload* to be "drained" via performing inspections. A main technical difficulty lies in analyzing the dynamics associated with these workloads under our inspection policy. To this end, we build on a program pioneered by Rybko and Stolyar (1992) and Dai (1995) in the context of queueing networks, in which we approximate the system's aggregate workload with a certain fluid limit, and use a contraction property of the fluid limit to derive the stability of the original system. More specifically, we show that, in the fluid limit, our inspection policy is effectively inducing a negative drift on a certain potential function of the aggregate system workload, which further implies the desired contraction property.

5. Concluding Remarks

The main objective of this work is to understand the design principles and fundamental limitations involved as one tries to efficiently classify a large set of items using a finite amount of processing resources. The main result demonstrates a prior-oblivious inspection architecture that asymptotically uses the minimum number of experts, in the regime where the required classification error tends to zero. More broadly speaking, our result is an attempt towards understanding how to build effective processing architectures for large-scale statistical learning or information extraction tasks, given limited resources or processing power. We believe that there are many other problems in this domain, situated at the intersection between stochastic modeling and statistics, that may be of interest for future research.

References

Keith Baker and J. von Beers. Shmoo plotting: The black art of IC testing. In *Proceedings of IEEE International Test Conference*, pages 932–933, 1996.

Herman Chernoff. Sequential design of experiments. *Annals of Mathematical Statistics*, 30(3): 755–770, 1959.

Jim G. Dai. On positive harris recurrence of multiclass queueing networks: a unified approach via fluid limit models. *The Annals of Applied Probability*, pages 49–77, 1995.

Marie F. Gerdtz and Tracey K. Bucknall. Triage nurses' clinical decision making. An observational study of urgency assessment. *Journal of Advanced Nursing*, 35(4):550–561, 2001.

David R. Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

Laurent Massoulie and Kuang Xu. On the capacity of information processing systems. *arXiv preprint arXiv:1603.00544v2*, 2016. URL http://arxiv.org/abs/1603.00544.

Aleksandr N. Rybko and Alexander L. Stolyar. Ergodicity of stochastic processes describing the operation of open queueing networks. *Problemy Peredachi Informatsii*, 28(3):3–26, 1992.

Abraham Wald. Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(2): 117–186, 1945.