Cognition 178 (2018) 67–81

Contents lists available at ScienceDirect

Cognition

journal homepage: www.elsevier.com/locate/cognit

Original Articles

Remembrance of inferences past: Amortization in human hypothesis generation☆

Ishita Dasgupta[a,*], Eric Schulz[b], Noah D. Goodman[c], Samuel J. Gershman[d]

[a] Department of Physics and Center for Brain Science, Harvard University, USA
[b] Harvard University, Department of Psychology, Cambridge, MA, USA
[c] Department of Psychology, Stanford University, USA
[d] Department of Psychology and Center for Brain Science, Harvard University, USA

ARTICLE INFO

Keywords:
Amortization
Hypothesis generation
Bayesian inference
Monte Carlo

ABSTRACT

Bayesian models of cognition assume that people compute probability distributions over hypotheses. However, the required computations are frequently intractable or prohibitively expensive. Since people often encounter many closely related distributions, selective reuse of computations (amortized inference) is a computationally efficient use of the brain's limited resources. We present three experiments that provide evidence for amortization in human probabilistic reasoning. When sequentially answering two related queries about natural scenes, participants' responses to the second query systematically depend on the structure of the first query. This influence is sensitive to the content of the queries, only appearing when the queries are related. Using a cognitive load manipulation, we find evidence that people amortize summary statistics of previous inferences, rather than storing the entire distribution. These findings support the view that the brain trades off accuracy and computational cost, to make efficient use of its limited cognitive resources to approximate probabilistic inference.

"*Cognition is recognition.*"

Hofstadter (1995)

1. Introduction

Many theories of probabilistic reasoning assume that human brains are equipped with a general-purpose inference engine that can be used to answer arbitrary queries for a wide variety of probabilistic models (Griffiths, Vul, & Sanborn, 2012; Oaksford & Chater, 2007). For example, given a joint distribution over objects in a scene, the inference engine can be queried with arbitrary conditional distributions, such as:

- What is the probability of a microwave given that I've observed a sink?
- What is the probability of a toaster given that I've observed a sink and a microwave?
- What is the probability of a toaster and a microwave given that I've observed a sink?

The nature of the inference engine that answers such queries is still an open research question, though many theories posit some form of approximate inference using Monte Carlo sampling (e.g., Dasgupta, Schulz, & Gershman, 2017; Denison, Bonawitz, Gopnik, & Griffiths, 2013; Gershman, Vul, & Tenenbaum, 2012; Sanborn & Chater, 2016; Thaker, Tenenbaum, & Gershman, 2017; Vul, Goodman, Griffiths, & Tenenbaum, 2014; Ullman, Goodman, & Tenenbaum, 2012). According to these theories, probability distributions are mentally represented with a set of samples, which are generated using a general-purpose inference engine that can operate on arbitrary probability distributions.

The flexibility and power of such a general-purpose inference engine trades off against its computational efficiency: by treating each query distribution independently, an inference engine forgoes the opportunity to reuse computations across queries, thus reducing time complexity (but possibly increasing space complexity). Every time a distribution is queried, past computations are ignored and answers are produced anew—the inference engine is memoryless, a property that makes it statistically accurate but inefficient in environments with overlapping queries.

Continuing the scene inference example, answering the third query should be easily computable once the first two queries have been computed. Mathematically, the answer can be expressed as:

☆ A preliminary version of this work was previously reported in Dasgupta, Schulz, Goodman, and Gershman (2017).

* Corresponding author at: Department of Physics and Center for Brain Science, Harvard University, 52 Oxford Street, Room 295.08, Cambridge, MA 02138, United States.
E-mail address: idasgupta@physics.harvard.edu (I. Dasgupta).

https://doi.org/10.1016/j.cognition.2018.04.017
Received 8 December 2017; Received in revised form 19 April 2018; Accepted 20 April 2018
0010-0277/ © 2018 Elsevier B.V. All rights reserved.

$$P(\text{toaster} \wedge \text{microwave}|\text{sink}) = P(\text{toaster}|\text{sink},\text{microwave})P(\text{microwave}|\text{sink}). \quad (1)$$

Even though this is a trivial example, standard inference engines do not exploit these kinds of regularities because they are memoryless—they have no access to traces of past computations.

An inference engine may gain efficiency by incurring some amount of bias due to reuse of past computations—a strategy we will refer to as *amortized inference* (Gershman & Goodman, 2014; Stuhlmüller, Taylor, & Goodman, 2013). For example, if the inference engine stores its answers to the "toaster" and "microwave" queries, then it can efficiently compute the answer to the "toaster or microwave" query without re-running inference from scratch. More generally, the posterior can be approximated as a parametrized function, or *recognition model*, that maps data in a bottom-up fashion to a distribution over hypotheses, with the parameters trained to minimize the divergence between the approximate and true posterior.[1] By sharing the same recognition model across multiple queries, the recognition model can support rapid inference, but is susceptible to "interference" across different queries, a property that we explore below.

One way to construct a recognition model is using Monte Carlo sampling: the recognition model can be viewed as a kind of data-driven sampler whose parameters are optimized so that the samples resemble the true posterior. In an amortized architecture, these parameters are shared across different inputs (i.e., data) and hence the samples will be correlated, introducing a systematic bias. If the sampling process corresponds to a Markov chain Monte Carlo algorithm (see below), this bias will disappear with a sufficiently large number of samples, but since humans appear to take a relatively small number of samples (Dasgupta et al., 2017; Vul et al., 2014), we expect this bias to be measurable.

Amortization has a long history in machine learning; the *locus classicus* is the Helmholtz machine (Dayan, Hinton, Neal, & Zemel, 1995; Hinton, Dayan, Frey, & Neal, 1995), which uses samples from the generative model to train a recognition model. More recent extensions and applications of this approach (e.g., Kingma & Welling, 2013; Paige & Wood, 2016; Rezende, Mohamed, & Wierstra, 2014; Ritchie, Thomas, Hanrahan, & Goodman, 2016) have ushered in a new era of scalable Bayesian computation in machine learning. We propose that amortization is also employed by the brain (see Yildirim, Kulkarni, Freiwald, & Tenenbaum, 2015, for a related proposal), flexibly reusing past inferences in order to efficiently answer new but related queries. The key behavioral prediction of amortized inference is that people will show correlations in their judgments across related queries.

We report 3 experiments that test this prediction using a variant of the probabilistic reasoning task previously studied by Dasgupta et al. (2017). In this task, participants answer queries about objects in scenes, much like in the examples given above. Crucially, the hypothesis space is combinatorial because participants have to answer questions about sets of objects (e.g., "All objects starting with the letter S"). This renders exact inference intractable: the hypothesis space cannot be efficiently enumerated. In our previous work (Dasgupta et al., 2017), we argued that people approximate inference in this domain using a form of Monte Carlo sampling. Although this algorithm is asymptotically exact, only a small number of samples can be generated due to cognitive limitations, thereby revealing systematic cognitive biases such as anchoring and adjustment, subadditivity, and superadditivity (see also Lieder, Griffiths, Huys, & Goodman, 2017b, 2017a; Vul et al., 2014).

We show that the same algorithm can be generalized to reuse inferential computations in a manner consistent with human behavior. First we describe how amortization might be used by the mind. We consider two crucial questions about how this might be implemented: what parts of previous calculations do people reuse—all previous memories or summaries of the calculations— and when do they choose to reuse their amortized calculations. Next we test these questions empirically. In Experiment 1, we demonstrate that people *do* use amortization by showing that there is a lingering influence of one query on participants' answers to a second, related query. In Experiment 2, we explore what is reused, and find that people use summary statistics of their previously generated hypotheses, rather than the hypotheses themselves. Finally, in Experiment 3, we show that people are more likely to reuse previous computations when those computations are most likely to be relevant: when a second cue is similar to a previously evaluated one.

2. Hypothesis generation and amortization

Before describing the experiments, we provide an overview of our theoretical framework. First, we describe how Monte Carlo sampling can be used to approximate Bayesian inference, and summarize the psychological evidence for such an approximation. We then introduce amortized inference as a generalization of this framework.

2.1. Monte Carlo sampling

Bayes' rule stipulates that the posterior distribution is obtained as a normalized product of the likelihood $P(d|h)$ and the prior $P(h)$:

$$P(h|d) = \frac{P(d|h)P(h)}{\sum_{h' \in \mathscr{H}} P(d|h')P(h')}, \quad (2)$$

where $\mathscr{H}$ is the hypothesis space. Unfortunately, Bayes' rule is computationally intractable for all but the smallest hypothesis spaces, because the denominator requires summing over all possible hypotheses. This intractability is especially prevalent in combinatorial space, where hypothesis spaces are exponentially large. In the scene inference example, $\mathscr{H} = \mathscr{H}_1 \times \mathscr{H}_2 \times \cdots \mathscr{H}_K$ is the product space of latent objects, so if there are K latent objects and M possible objects, $|\mathscr{H}| = M^K$. If we imagine there are $M = 1000$ kinds of objects, then it only takes $K = 26$ latent objects for the number of hypotheses to exceed the number of atoms in the universe.

Monte Carlo methods approximate probability distributions with samples $\theta = \{h_1,...,h_N\}$ from the posterior distribution over the hypothesis space. We can understand Monte Carlo methods as producing a recognition model $Q_\theta(h|d)$ parametrized by θ (see Saeedi, Kulkarni, Mansinghka, & Gershman, 2017, for a systematic treatment). In the idealized case, each hypothesis is sampled from $P(h|d)$. The approximation is then given by:

$$P(h|d) \approx Q_\theta(h|d) = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}[h_n = h], \quad (3)$$

where $\mathbb{I}[\cdot] = 1$ if its argument is true (and 0 otherwise). The accuracy of this approximation improves with N, but from a decision-theoretic perspective even small N may be serviceable (Vul et al., 2014; Lieder, Griffiths, Huys, & Goodman, 2017a; Gershman, Horvitz, & Tenenbaum, 2015).

The key challenge in applying Monte Carlo methods is that generally we do not have access to samples from the posterior. Most practical methods are based on sampling from a more convenient distribution, weighting or selecting the samples in a way that preserves the asymptotic correctness of the approximation (MacKay, 2003). We focus on Markov chain Monte Carlo (MCMC) methods, the most widely used class of approximations, which are based on simulating a Markov chain whose stationary distribution is the posterior. In other words, if one samples from the Markov chain for long enough, eventually h will be sampled with frequency proportional to its posterior probability.

[1] Formally, this is known as *variational inference* (Jordan, Ghahramani, Jaakkola, & Saul, 1999), where the divergence is typically the Kullback-Leibler divergence between the approximate and true posterior. Although this divergence cannot be minimized directly (since it requires knowledge of the true posterior), a bound (variational free energy) can be tractably optimized for some classes of approximations.

Download English Version:

https://daneshyari.com/en/article/7285225

Download Persian Version:

https://daneshyari.com/article/7285225

Daneshyari.com