

How to Make Decisions (Optimally)

Siddhartha Sen

Microsoft Research New York City, USA
sidsen@microsoft.com

Abstract

Distributed systems are constantly faced with difficult decisions to make, such as in scheduling, caching, and traffic routing, to name a few. In most of these scenarios, the optimal decision is unknown and depends heavily on context. How can a system designer know if they have deployed the best decision-making policy, or if a different policy would perform better? As a community, we have developed a few methodologies for answering this question, some of them offline (e.g., simulation, trace-driven modeling) and some of them online (e.g., A/B testing). Neither approach is satisfactory: the offline methods suffer from bias and rely heavily on domain knowledge; the online methods are costly and difficult to deploy. What system designers ideally seek is the ability to ask “what if” questions about a policy without ever deploying it, which is called *counterfactual evaluation*. In this talk, I will show how reinforcement learning and causal inference can be synthesized to counterfactually evaluate a distributed system. We will apply this methodology to infrastructure systems in Azure, and face fundamental challenges and opportunities along the way. This talk will serve as an introduction to reinforcement learning and the counterfactual way of thinking, which I hope will interest and inspire the OPODIS community.

I will start by introducing reinforcement learning (RL) as the right framework for modeling decisions in a distributed system. In RL, an agent learns by interacting with its environment: i.e., making decisions and receiving feedback for them. This is a stark contrast to traditional (supervised) learning, where the correct answer, or “label”, is known. Since an RL agent does not know the correct answer, it must constantly explore its world by randomizing some of its decisions. Now it turns out that this randomization, if used correctly, can give us a special superpower: the ability to *evaluate policies that have never been deployed*. As magical as this may sound, we can use statistics to show that this evaluation is indeed correct.

Unfortunately, applying this methodology to distributed systems is far from straightforward. Systems are complex, stateful amalgamations of components that navigate large decision spaces. We will need to wear both an RL hat and a systems hat to address these challenges. On the other hand, systems also present exciting opportunities. Many systems already use randomization in their decisions, e.g., to distribute data or work over replicas, or to manage resource contention. Sometimes, a conservative decision can implicitly yield feedback for other decisions: for example, when waiting for a timeout to expire, we automatically get feedback for what would have happened if we waited for any shorter amount of time. I will show how we can harvest this randomness and implicit feedback to achieve more effective counterfactual evaluation.

We will apply all of the above ideas to two production infrastructure systems in Azure: a machine health monitor that decides when to reboot unresponsive machines, and a geo-distributed edge proxy that chooses the TCP configuration of each proxy machine. In both cases, we are able to counterfactually evaluate arbitrary policies with estimates that match the ground truth. Production environments raise interesting constraints and challenges, some of which are preventing us from scaling up our methodology. I will describe a possible path forward, and invite others in the community to contemplate these problems as well.

2012 ACM Subject Classification Computing methodologies → Reinforcement learning, Software and its engineering → Software organization and properties

Keywords and phrases reinforcement learning, distributed systems, counterfactual evaluation

Digital Object Identifier 10.4230/LIPICs.OPODIS.2018.3

Category Keynote



© Siddhartha Sen;

licensed under Creative Commons License CC-BY

22nd International Conference on Principles of Distributed Systems (OPODIS 2018).

Editors: Jiannong Cao, Faith Ellen, Luis Rodrigues, and Bernardo Ferreira; Article No. 3; pp. 3:1–3:1

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany