A Graphical Modeling Viewpoint on Queueing Networks

Charles Sutton and Michael I. Jordan
{casutton,jordan}@cs.berkeley.edu
Computer Science Division
University of California, Berkeley

Diagnosis of performance problems in computer systems is a rich application area for machine learning, because data about system performance can be readily obtained. Many such diagnostic questions concern system performance in the face of load. For example: *"Five minutes ago, a brief spike in workload occurred. Which parts of the system were the bottleneck during that spike?"* A second type of question is diagnosis of slow requests: *"During the execution of the 1% of requests that perform poorly, which system components receive the most load?"* The bottleneck for slow requests could be very different than the bottleneck for typical requests if, for example, a storage or network resource is failing intermittently.

However, classical approaches to machine learning are not an entirely good fit: Supervised learning requires labeling failure data, a time-consuming task that may need to be performed anew for each application to be diagnosed. On the other hand, a fully unsupervised approach fails to exploit the known structure of the system. An appealing alternative is a model-based approach, in which we design a performance model that can be learned directly from measurements of system performance and that incorporates the structure of the system as an inductive bias. A class of performance model that is particularly well studied is queueing models. Queueing models predict the explosion in system latency under high workload in a way that is often reasonable for real systems, allowing the model to extrapolate from performance under low load to performance under high load. Queueing theory has been studied for over a hundred years, but the theory concerns approximating future behavior of the system, not inference about past system behavior or learning from incomplete data.

In this work, we present a new family of analysis techniques for queueing models, based on a statistical viewpoint. We collect a training set by sampling a small set of arrival and departure times from the system, treating the times that were not sampled as missing data. The issue of missing data is unavoidable in real systems, either because full instrumentation is too expensive, or because the true bottlenecks in the system are unknown. To learn the model parameters, we sample from the posterior distribution over missing data and parameters in a Bayesian fashion, using approximate inference algorithms for graphical models. Essentially, we view a queueing network as a structured probabilistic model, a novel viewpoint that combines queueing networks and graphical models.

Specifically, we develop a slice sampler for networks of G/G/K queues, resampling the latent arrivals and departures one at a time. Algorithmically, the sampler is significantly more complex for queueing networks than for standard graphical models, for two reasons. First, the local conditional distribution over a single departure can have many singularities, corresponding to when other tasks arrive and depart. Second, the Markov blanket for a departure can be arbitrarily large, because a delay in one departure can tie up arbitrarily

---

Category: graphical models
Preference: oral
Presenter: Charles Sutton

many later jobs in the queue.

We test the sampler on log data generated by Cloudstone [1], a recently-proposed benchmark that is designed to model Web 2.0 applications. Cloudstone has been implemented on several platforms for Web development by professional Web developers, with the intention of reflecting common design idioms of real-world Web applications. We demonstrate the ability to localize bottlenecks to a component of the system, and to determine whether performance degradation is due to the intrinsic performance of the system, or to increased workload. Furthermore, we demonstrate the ability to perform accurate localization with 25% of the overhead of full instrumentation.

A particularly interesting application of this framework is model selection. Beginning with a network that reflects programmers' beliefs about the system's performance, we can greedily attempt to add additional queues to the network to find a structure that better fits observed. If such a structure exists, this can indicate a hidden resource bottleneck, of which the programmer was previously unaware.

Queueing models have been long studied in telecommunications, operations research, and performance modeling of computer networks and systems. Queueing theory focuses on analytic approximations to the long-run behavior of the system—such as the steady-state distribution or large-deviations bounds—but does not consider the problem of inferring system behavior from incomplete data. Despite the long history of queueing models, we are unaware of any existing work that treats them as latent-variable probabilistic models, and attempts to approximate the posterior distribution directly. Furthermore, we are unaware of any technique for estimating the parameters of a queueing model from an incomplete sample of arrivals and departures.

References

[1] Will Sobel, Shanti Subramanyam, Akara Sucharitakul, Jimmy Nguyen, Hubert Wong, Sheetal Patil, Armando Fox, and David Patterson. Cloudstone: Multi-platform, multi-language benchmark and measurement tools for Web 2.0. In *First Workshop on Cloud Computing and its Applications (CCA)*, 2008.