Agents in the era of big data: What the "end of theory" might mean for agent systems

Aditya Ghose

Decision Systems Laboratory
School of Computer Science and Software Engineering
University of Wollongong, NSW 2522 Australia
aditya@uow.edu.au

1 Introduction

The "data deluge" has been the topic of much recent discourse. Over the past decade (but, in some cases, even earlier) we have come to recognize that we have crossed a sort of tacit threshold in our capacity to collect, store and analyze large volumes of data. This has spawned a large industry in big data analytics, much of it focused on leveraging business insights (such as customer buying patterns) from large and real-time data streams. But we have also started asking whether the big data phenomenon might perhaps represent a more fundamental shift in the process of human inquiry. For one, we have the wherewithal to generate large bodies of statistical correlations from these data streams. That has not been without controversy (see, for instance, the recent Chomsky-Norvig debate [1]). For another, the statistically mined knowledge is turning out, in some cases, to be transient. The rules that we are able to mine change with the arrival of new data (although that is no surprise). This has led some to posit the "end of theory" [2]. The time-honoured scientific method of constructing largely stable predictive models on the basis of carefully curated data is being supplanted by machinery that generates and updates (potentially less reliable) models from data very quickly and without demanding as much effort from the human in the loop. To quote Anderson [2]: "Correlation supersedes causation, and science can advance without coherent models, unified theories, or really any mechanistic explanation at all"'.

My main argument here is that this is a game-changer for the agents research community. I will discuss some of the opportunities in broad outline, offer some specific examples, and list some challenges.

Much like the bulk of the AI community, agents researchers can now leverage the speed, scale and ubiquity of knowledge acquisition machinery. The devil, as always, is in the detail. The kinds of knowledge that agent systems rely on require specialized machinery, both for knowledge mining and by way of instrumentation for data collection. Let us consider a simple taxonomy for the types of knowledge that agent systems rely on. *Prescriptions* specify behaviour, and is to be found in constructs such as agent plans, policies, strategies and so on. *Drivers* specify agent motivation, and can be found in constructs such as goals, optimization objectives, event triggers and so on, *Constraints* circumscribe the space of valid behaviours, and can be found in constructs such as norms, commitments or even the constraint theories that multi-agent optimization techniques use. Each category in this taxonomy requires bespoke data and machinery. Prescriptions are typically mined from behaviour logs, process logs and their ilk. Drivers must be mined from data that describes the *impact* of agent behaviour (in our work, we have used *effect logs* that describe state transitions of objects impacted by agent behaviour). Sometimes, agent behaviour manifests as choices between competing alternatives, which can be captured in *decision logs*. Constraints must also be mined from a record of the impact of agent behaviour, or from a record of speech acts that contain clues about the commitments that an agent might have made [3].

In the following, I will offer two examples of our recent work in leveraging data in building and maintaining agent systems, before discussing some open questions.

2 Mining agent programs

In recent work [4], we addressed the knowledge acquisition bottleneck in the context of BDI agent programs. Given that organizations often avoid building agent-based solutions (even when there is a clear need and a good fit) because of the investment required and the perceived complexity of agent programming, we set out to build an *agent mining* tool that would simplify the task of building agent programs (and thus improve agent programmer productivity). The tool infers "first-cut" agent plans from *process logs* (time-stamped records of task execution) and plan contexts (preconditions) from *effect logs* (time-stamped records of object states). The tool also leverages norm mining techniques to refine agent plans from process log entries tagged as erroneous (either because of obligatory actions that were not performed or prohibited actions that were performed). Initial experimental results on the extent to which we could improve agent programmer productivity were promising.

More generally, this suggests that data-driven generation of agent behaviour (prescriptions) might become fairly standard in the future. Indeed, we might conceive of machinery that maintained agent programs to best fit the available data, incorporating human oversight at appropriate intervals and for key decision points.

3 The *Know-How Miner*

In recent work [5], we built the *Know-How Miner* to extract know-how descriptors (patterns of the form: `to <GOAL>, <STEP1>, <STEP2>...` - the steps to be separated by an AND or an OR) from the textual content of the web. Each know-how descriptor specifies the steps that (under conjunctive or disjunctive composition) help achieve a goal. The goal as well as the steps are extracted as text. The following is an example of a know-how descriptor that was extracted:

```
<to create a robust business plan>
take a comprehensive view of the enterprise
AND
incorporate management-practice knowledge from every first-semester
course
```

Implemented on fairly standard hardware, the tool is quite efficient. In one experiment, a single crawler, was able to produce 22 useful know-how descriptors in the space of about 30 minutes from 446 webpages.

The original intent of the *Know-How Miner* was harvesting process innovation. Organizations can use the tool to look outside their enterprise boundaries (and leverage, in the limit, all textual descriptions of know-how in the web) to identify alternative ways of doing things, which in turn would serve as triggers for process innovation.

It isn't hard to see that know-how descriptors can also be viewed as agent plans (or plan snippets). We can imagine a not-too-distant future where agent programmers would routinely rely on tools such as this to help them write better agent programs, or improve the quality of existing agent programs. We can also imagine a future where not just text, but multimedia content on the web could be mined for know-how.

4 Open questions

For agent researchers in the era of big data, the possibilities are endless. Some of our other on-going work looks at how we might extract the objective functions that drive optimizing agents from *decision logs* (where each entry specifies the set of options available to an agent, plus the option selected). Norm learning machinery already exists. A future where data-driven agents become routine appears to be within reach.

There are some hard problems to solve, however. Unlike many other applications of knowledge mining, the knowledge comprising agent systems is not easily amenable to the traditional generate-evaluate-update cycle that is used to quality assure the knowledge that is mined. How much testing, verification or validation should we subject a mined body of behavioural knowledge to, before we are confident of deploying it in an operational agent system? If we spend too long, the "best-fit" behavioural specifications that can be mined from the data might have changed, requiring another evaluation cycle. If norms are to be data-driven, what would the evaluate-update part of the cycle for normative multi-agent systems look like? And finally, where and how should we position the human in the loop? How do we decide how much human oversight would be appropriate for an adaptive agent system?

The problem of *context* bedevils many big data applications. Much of the data that we have access to does not come with detailed specifications of the context within which the data was generated. Yet that contextual information makes all the difference to the quality of the theories (or agent programs, or customer insights) that we generate. Sometimes we may have the capability to deploy lightweight instrumentation to acquire some contextual information. How much is enough?

If we are staring at the end of theory, we will be left in a world with no constants. What kind of a world would that be for agents, both of the human and machine variety?

References

1. Norvig, P.: On chomsky and the two cultures of statistical learning. On-line essay in response to Chomskys remarks in [2]. Available from: http://norvig. com/chomsky. html (2011)
2. Anderson, C.: The end of theory. Wired Magazine **16** (2008)
3. Gao, X., Singh, M.P., Mehra, P.: Mining business contracts for service exceptions. Services Computing, IEEE Transactions on **5**(3) (2012) 333–344
4. Xu, H., Savarimuthu, B.T.R., Ghose, A.K., Morrison, E., Cao, Q., Shi, Y.: Automatic bdi plan recognition from process execution logs and effect logs. In: Proc. of the EMAS-2013 Workshop, held in conjunction with AAMAS-2013, Springer LNAI (2013 (To appear))
5. Ghose, A.K., Morrison, E., Gou, Y.: A novel use of big data analytics for service innovation harvesting. In: Proc. of the 2013 International Conference on Service Science and Innovation, IEEE Computer Society Press (2013)