Online computation with untrusted advice

**Areas of interest:** Optimization, Theoretical Computer Science, Machine Learning.

**Place:** Laboratoire d’Informatique de Paris 6 (LIP6), Sorbonne Université, Paris.

**Team:** Recherche Opérationnelle.

**Director of internship:** Spyros Angelopoulos (spyros.angelopoulos@lip6.fr).

**Director of host laboratory:** Fabrice Kordon (Fabrice.Kordon@lip6.fr).

**General presentation of the topic**

Suppose that you have an investment account with a significant amount in it, and that your financial institution advises you periodically on investments. One day, your banker informs you that company X will soon receive a big boost, and advises to use the entire account to buy stocks. If you were to completely trust the banker’s advice, there are naturally two possibilities: either the advice will prove correct (which would be great) or it will prove wrong (which would be catastrophic). A prudent customer would take this advice with a grain of salt, and would not be willing to risk everything. In general, our understanding of advice is that it entails *knowledge that is not foolproof*. A similar situation occurs in the field of online computation with advice. In this field, the online algorithm receives some additional information concerning the input. This information could reflect anything: from simple and useful concepts (such as the length of the input) to complex and unintuitive ones (e.g., how the optimal algorithm makes certain decisions). This has been a heavily researched topic over the last decade: See the survey [2] for a summary of results.

All this previous work assumed that the advice has no errors, and it comes from a trusted source. However, it is possible, as shown in the example above, that the advice is erroneous. We thus want algorithms that are *robust*, that is if the advice is correct, they perform very well, but they are also *consistent*: if the advice is incorrect, they do not perform much worse than the best algorithm with no advice. This framework was recently introduced in [1] which studied important online problems such as ski rental, online bidding, list update, and bin packing under untrusted advice.

Very recently, the topic of online computation with advice was also studied within the Machine Learning community. More specifically, [4] consider the *paging* problem with Machine Learned advice: here, there is an oracle that gives some information about the input to the algorithm, and we seek online algorithms whose performance degrades gently as function of the error in the advice. Examples of recent additional works are [5] and [3].
Objectives

This is a new, promising, and active topic of research, and has intersections with both Theoretical Computer Science and Machine Learning.

Specific questions of interest to this internship:

- Can we find other online problems for which the online algorithm performs very well if the advice is correct, but is very inefficient if the advice is incorrect? For such problems, what is the best tradeoff one can achieve?

- What is the impact of randomization on online algorithms with advice? For instance, what is an optimal algorithm for randomized ski rental with untrusted advice?

- Define and apply a model in which the advice is probabilistically correct.

The above is an indication of topics that are of interest to this internship. An exact list will be worked out in collaboration with the supervisor.

Qualifications

No prior knowledge of online algorithms is required, but a solid background and general interest in theoretical analysis of algorithms are essential.

References


