Coopetitions in machine learning: case studies

Xavier Baró[1,2], **Sergio Escalera**[2,3,5], **Isabelle Guyon**[4,5], **Julio Jacques Jr.**[2,3], **Lukasz Romazco**[5], **Lisheng Sun**[4], **Sébastien Treguer**[6], **Evelyne Viegas**[7]
[1] EIMT/IN3 at the Open University of Catalonia, Barcelona, Spain.
[2] Computer Vision Center, Campus UAB, Barcelona, Spain
[3] Dept. Mathematics, University of Barcelona, Spain
[4] UPSud Paris-Saclay, Paris, France
[5] ChaLearn, California, USA
[6] La Paillasse, France
[7] MSR, USA

The notion of coopetition has been used in game theory, econometrics, and political science to describe systems in which agents have a partial congruence of interest and cooperate with each other to reach a higher value than by merely competing. We want to use the notion of coopetition in Machine Learning by organizing challenges in which participants collaborate to some extent, with the objective of collectively improving the challenge outcome (the best solutions to the proposed task). Classical challenges in Machine Learning are typically built around competitive winner-take-all models, in which mutually exclusive teams try to independently solve a problem to win a prize. We have been exploring several means of implementing coopetitions with the Codalab platform (http://codalab.org). This paper will compare and contrast designs we have been experimenting with, including new designs making use of Codalab worksheets, a collaborative code-sharing environment.

Background

The most pervasive means of implementing a notion of coopetition in “classical" Machine Learning challenges are simply **running recurrent challenges** or **encouraging team mergers**. When running recurring challenge, cooperation is achieved by disseminating the results of each competition by means of workshops and proceedings. The organizers may impose to the winners to make their code available as "open source" to qualify for their prizes. Examples of recurring challenges include TREC organized by NIST (since 1992, originally focused on text information retrieval, later expanding to other areas such as video processing), CASP (since 1994, focused on protein structure prediction), Robocup (since 1997, a competition of robotics), DREAM (since 2007, on the theme of DNA microarray analysis), VOC (2005-2012) and ImageNet challenges (since 2010), both focusing on image recognition and sometimes run jointly, and the ChaLearn gesture challenges (since 2011 on multi-modal gesture recognition). The second typical way of encouraging collaboration is to allow coalitions or team mergers. The organizers provide on-line feedback on progress made and a discussion forum to let competitors freely exchange ideas and authorize the merger of teams. In the Netflix prize, the challenge protocol imposed an absolute performance threshold to be exceeded to win the grand prize. This could only be achieved with a team merger after a long period of performance stagnation. In the recurring challenges, we can see how the performance achieved by the winners increases in every run of the challenge, even when the data becomes more challenging. In the Netflix challenge, performance leaps were obtained after team merges. Therefore, there is value in coopetitions to encourage a broader base of participants with complementary skills to contribute. However, to the best of our knowledge, there has not been yet any formal attempt by competition organizers to encourage collaboration between competitors by structuring the rules of the challenge in a particular way and/or facilitating information exchange. Additionally, “classical ways" of encouraging collaboration (with recurring challenges and team mergers) do not provide a means of rewarding participants for partial contributions: each team still has to solve end-to-end the task(s) of the challenge and only the participants ranking at the top at the end of the challenge win a prize.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

New coopetition designs

We are interested in encouraging two types of partial contributions:

- Sub-task contribution: A team may contribute with a module or a key idea, which, alone, is insufficient to get good overall performance. In a complex challenge, domain-specific skills that are complement one another are welcome.
- Snowball effect: Teams that break the ice by entering early in the competition or that make a sudden leap in performance make an important contribution: they attract interest and push all other teams to match the new best result (which often happens within hours).

We are exploring several types of coopetition designs to implement such ideas. The simplest design (implemented in the 2015 Microsoft Machine Learning Summer School `http://automl.chalearn.org/mlischool-petersburg`, and the Personality Trait challenge (second round, ICPR 2016 `https://competitions.codalab.org/competitions/10751`), consists in encouraging code sharing during the competition by rewarding number of downloads. The second design we will present at the workshop is presently being implemented for a follow up challenge to the AutoML challenge (`http://codalab.org/AutoML/`), in which the participant will team up to beat the winners of the AutoML challenge. In this second design, the participants' submissions will be included into a heterogeneous ensemble of methods. The results will be reported at the workshop. We will also describe more elaborate design under implementation in which participants will be able to contribute and share modules of an overall solution making use of Codalab worksheets (`https://worksheets.codalab.org/`).

Reward mechanisms and game theoretic backing

To reward participants for partial contributions we need to work both on changing the typical rules of machine learning challenges and adding more flexibility to challenge platforms. In this direction, we found that it is useful to use concepts developed in game theory to put a formal framework around the design of coopetitions. Game theory provides a framework to reason about problems in economics, political science, psychology, and biology that goes well beyond the study of recreational games. It can be thought of as an extension of decision theory to systems of intelligent rational decision-makers. According to Eric B. Rasmusen[1], the essential elements of a game are players, actions, payoffs, and information (aka PAPI). Challenges are games and challenge participants are **P**layers. The task of the organizers is to devise appropriate rules and facilitate game playing. In this line, **A**ctions include the submission of an entry to the competition; **P**ayoffs include prizes, gaining visibility (academic credit, jobs) and work dissemination opportunities (workshops and publications); and **I**nformation comprise the results rated on a leaderboard, additional data acquired during the challenge, knowledge gained from the literature or other sources and software available. For our new coopetition designs we reformulate some of the elements of the game. For **P**layers we introduce some asymmetries, enforcing participants with different roles. For **A**ctions we introduce resource sharing (data, knowledge, software). For **P**ayoff we introduce credit points earned, spent, and redeemable in various ways. And finally, as part of **I**nformation, resource ratings are computed.

Implementation

Most challenge platforms are too restrictive to implement coopetitions: the participants' actions are generally limited to submitting prediction results. We want to use the capabilities of the new platform under development Codalab to provide competition organizers with a broader PAPI vocabulary. The main benefits of Codalab are derived from the possibility of submitting generic "bundles" as challenge entries. A bundle is an archive that may contain data (prediction results being only a particular case) and code (executable or source code). Microsoft is providing access to its cloud computing facility Azure so that code bundles can be executed when submitted to the platform. Bundles can also be thought of as re-usable modules, which may be shared among participants. The present alpha version of Codalab is an open source software written in Python, which can be run either under Windows or Linux. It supports executable bundles written in Python, but the plan is to extend support for execution of any type of executable code (interpreted or compiled), isolating each user on a different virtual machine. A bundle execution time is limited by a maximum value and there is a limit on the number for submissions per day. Over this platform, we have described a number of building blocks that will constitute the new PAPI vocabulary of challenge design, beginning with the new type of actions.

[1] Eric Rasmussen. Games and Information. An Introduction to Game Teory. Oxford. 1989.