Discussion of "A conditional game for comparing approximations"

Vincent Conitzer
Duke University

Abstract

This brief paper discusses the paper by Eaton mentioned in the title [1].

1 MOTIVATION

How should we assess the quality of an approximate inference algorithm? One obvious approach is to see how it performs on instances that are small enough to solve exactly. However, this seems to be a poor way of evaluating approximate inference algorithms, because that is precisely where we do not need them. Indeed, an easy way to perform well on such an evaluation could be to design an algorithm as follows: run an exact algorithm for a pre-specified amount of time, and if it times out, run a very fast (possibly bad) approximate algorithm. While it would be natural to prohibit such a strategy in (say) a competition, some algorithms may have roughly the same behavior without explicitly following such a two-phase strategy, resulting in a difficult and perhaps subjective decision of whether the algorithm violates the rule. It would be much better to evaluate approximate inference algorithms on large instances that are out of the reach of exact methods; for this, after all, they are designed.

The problem is that it is hard to evaluate on such instances, because, by their fundamental characteristic, we do not know the ground truth for them. Eaton proposes a method for addressing this. Rather than attempting to compare the output of approximate inference algorithms to ground truth, Eaton proposes to have a pair of approximate inference algorithms compete head-to-head in a game designed for this purpose.

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15. Copyright 2011 by the authors.

2 THE GAME

(This description closely follows Section 3 of the paper [1].) In the basic version of the game, we have a factor graph that specifies a distribution over (discrete) variables $x_1, \ldots, x_n$, where

$$P(x) = \frac{1}{Z} \prod_{\alpha} \psi_\alpha(x_\alpha)$$

and each α is a subset of variables corresponding to a factor.

There are two players in the game, the *marginal player* and the *conditional player*.[1] In the i-th stage of the game, first, the marginal player gives a marginal distribution $q_i(x_i)$ over the i-th variable. The conditional player then chooses a value x_i^* to which this variable will be fixed for the rest of the game, and play proceeds to the next variable. The final payoff to the marginal player is

$$\log \frac{\prod_{i=1}^{n} q_i(x_i^*)}{\prod_\alpha \psi_\alpha(x_\alpha^*)}$$

(and the conditional player gets the negative of this, the game being zero-sum). The idea is that if the q_i distributions are exact conditional distributions ($q_i(x_i) = P(x_i|x_{1:i-1})$), then the payoff to the marginal player is $-\log Z$ regardless of the conditional player's actions; however, if the conditional player believes that the marginal player has underestimated the probability of a particular value, he can take advantage of this by setting the variable to this value, driving down the marginal player's payoff. Eaton shows how an approximate inference algorithm can be converted in a natural way to a strategy for either player in the game.

3 DESIGN CHOICES

As the author makes clear, different designs of the game would also be possible. He suggests one where

[1] I will use "she" to refer to the marginal player and "he" to refer to the conditional player. This is intended to make it easier to determine which player is being discussed, rather than to make any kind of statement about which gender is more natural for each role.

the conditioning player gets to choose not only the value for the next variable, but also *which* variable to fix next. This does not change the payoff for the marginal player if she plays optimally (i.e., exact inference), but if exact inference is not feasible, it does seem to bestow a further advantage on the conditional player. At worst, the conditional player could choose at random which variable to fix; but if the conditional player has any insight into for which variable the marginal player's approximation is particularly weak (at a given point in the game), he can take advantage of this. In fact, this is in the same vein as letting the conditional player choose the variable's value: the conditional player could just choose the value at random,[2] but if he has any insight into on which values the marginal player's approximation is particularly weak, he can take advantage of this. Of course, a randomized strategy for the conditional player makes sense if he must move *at the same time* as the marginal player.

Another aspect of the game that can be changed is the precise formula for the payoffs to the players. The author discusses some of the properties of the function he chose at the end of Subsection 3.2. Future research may be devoted to arguing more formally that this (or another) version of the game is "optimal" in some sense.

4 OTHER APPLICATIONS

Another direction for future research is to apply this approach to other computational problems. The fact that the numbers being computed are probabilities does not seem essential to the game; for example, we never sample from the distribution (rather, the conditional player fixes the values of the variables). It also does not seem essential to have a factored representation. Rather, what seems to be needed is that the (probability or other) function value can be calculated efficiently once every variable has has been fixed to a value, and that the values to be calculated at intermediate stages of the game are some type of aggregation (product, sum, ...) of the function values that can result from specifying values for the remaining variables. #P-complete problems seem a natural class to which to apply this approach. For example, consider the problem #SAT of counting the number of satisfying assignments of a given Boolean formula. The "marginal" player could estimate the number of satisfying assignments that would remain for each way of fixing the next Boolean variable to a value, and the "conditional" player would then choose the value of this variable. (Perhaps the game can be specified indirectly based on a reduction from the #SAT problem to the inference problem.)

5 CONCLUSION

I believe that the specific methodology in the discussed paper will be exciting to those in the approximate inference community, because it gives a way to address a concrete problem that researchers in that community face. However, I also think that the approach is exciting from a broader perspective, as an initial step towards a more general theory of how to evaluate algorithms that attempt to give approximate solutions to hard problems, by letting them compete against each other. This could be an attractive alternative in settings where worst-case approximation ratios are not satisfactory or hard to obtain. The notion of algorithms that compete against each other is one that has been receiving attention elsewhere recently [2]. Future research may pursue a general theory that allows us to rigorously make design choices such as those discussed above.

ACKNOWLEDGMENTS

I thank Frederik Eaton for useful feedback. This discussion paper was written while visiting Amsterdam's Centrum Wiskunde & Informatica (CWI).

References

[1] Frederik Eaton. A conditional game for comparing approximations. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, Ft. Lauderdale, FL, USA, 2011.

[2] Nicole Immorlica, Adam Tauman Kalai, Brendan Lucier, Ankur Moitra, Andrew Postlewaite, and Moshe Tennenholtz. Dueling algorithms. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC-11)*, San Jose, CA, USA, 2011.

[3] Flemming Topsøe. Information theoretical optimization techniques. *Kybernetika*, 15(1):8–27, 1979.

[2] Note that "at random" here does *not* mean according to the actual distribution, since this is presumed to be hard; rather, he could just choose it uniformly at random, or according to his own approximation of the distribution (see also the discussion of the code-length game [3] in Subsection 4.3).