Bayes-Relational Learning of Opponent Models from Incomplete Information in No-Limit Poker

Marc Ponsen[1], **Jan Ramon**[2], **Tom Croonenborghs**[3], **Kurt Driessens**[2] **and Karl Tuyls**[4]
1: MICC Maastricht University, Netherlands
2: Declarative Languages and Artificial Intelligence Group, Katholieke Universiteit Leuven (KUL), Belgium
3: Biosciences and Technology Department, KH Kempen University College, Belgium
4: Technische Universiteit Eindhoven, Netherlands

Abstract

We propose an opponent modeling approach for No-Limit Texas Hold'em poker that starts from a (learned) prior, i.e., general expectations about opponent behavior and learns a relational regression tree-function that adapts these priors to specific opponents. An important asset is that this approach can learn from incomplete information (i.e. without knowing all players' hands in training games).

Introduction

For many board and card games, computers have at least matched humans in playing skill. An exception is the game of poker, offering new research challenges. The complexity of the game is threefold, namely poker is (1) an imperfect information game, with (2) stochastic outcomes in (3) an adversarial multi-agent environment. One promising approach used for AI poker players applies an adaptive imperfect information game-tree search algorithm to decide which actions to take based on expected value (EV) estimates (Billings *et al.* 2006). This technique (and related simulation algorithms) require two estimations of opponent information to accurately compute the EV, namely a prediction of the opponent's outcome of the game and prediction of opponent actions. Therefore learning an opponent model is imperative and this model should include the possibility of using relational features for the game-state and -history.

In this paper we consider a relational Bayesian approach that uses a general prior (for outcomes and actions) and learns a relational regression tree to adapt that prior to individual players. Using a prior will both allow us to make reasonable predictions from the start and adapt to individual opponents more quickly as long as the choice of prior is reasonable.

Learning an Opponent Model

We learn an opponent model for players in the game of No-Limit Texas Hold'em poker. To make the model useful for an AI player, we must be able to learn this model from a limited amount of experience and (if possible) adapt the model quickly to changes in the opponent's strategy. An added, and important, difficulty in poker is that we must be able to learn this model given a large amount of hidden information. We propose to start the opponent model with a prior distribution over possible action choices and outcomes. We will allow the model to adapt to different opponents by correcting that prior according to observed experience.

Consider a player p performing the i-th action a_i in a game. The player will take into account his hand cards, the board B_i at time point i and the game history H_i at time point i. The board B_i specifies both the identity of each card on the table (i.e., the *community cards* that apply to all players) and when they appeared, and H_i is the betting history of all players in the game. The player can *fold*, *call* or *bet*. For simplicity, we consider *check* and *call* to be in the same class, as well as *bet* and *raise* and we don not consider the difference between small and large calls or bets at this point.[1]

We limit the possible outcomes of a game r_p for a player p to: 1) p folds before the end of the game ($r_p = lose$), 2) p wins without showing his cards ($r_p = win$) and 3) p shows his cards ($r_p = cards(X,Y)$). This set of outcome values also allows us to learn from examples where we did not see the opponent's cards, registering these cases as win or $lose$, without requiring the identities of the cards held by the player. The learning task now is to predict the outcome for an opponent $P(r_p|B_i, H_i)$ and the opponent action (given a guess about his hand cards) $P(a_i|B_i, H_{i-1}, r_p)$

Learning the Corrective Function

We propose a two-step learning approach. First, we learn functions predicting outcomes and actions for poker players in general. These functions are then used as a prior, and we learn a corrective function to model the behavior and statistics of a particular player. The key motivations for this are first that learning the difference between two distributions is an elegant way to learn a multi-class classifier (e.g. predicting distributions over 2+(52*53/2) possible outcomes) by generalizing over many one-against-all learning tasks, and second that even with only a few training examples from a particular player already accurate predictions are possible.

In the following description, the term example references a tuple $(i, p, a_i, r_p, H_{i-1}, B_i)$ of the action a_i performed at step i by a player p, together with the outcome r_p of the game, the board B_i and the betting history H_{i-1}.

Consider the mixture D_{p+*} of two distributions: the distribution D_* of arbitrarily drawn examples from all players and the distribution D_p of arbitrarily drawn examples from a particular player p. Then, consider the learning problem of, given a randomly drawn example x from D_{p+*}, predicting whether x originated from D_* or from D_p. For a given learning setting (either predicting actions from outcomes or predicting outcomes from actions), it is easy to generate examples from D_* and D_p, labeling them with $*$ or p, and learning the function $P(D_p|x)$, giving for each example x the probability the example is labeled with p. We do so

[1] We will consider the difference between small and large calls or bets as features in the learned corrective function.

by using the relational probability tree learner TILDE (Blockeel & De Raedt 1998). From this learned 'differentiating' model, we can compute the probability $P(x|D_p)$, for every example x by using Bayes' rule:

$$P(x|D_r) = P(D_r|x) \cdot P(x)/P(D_r) \qquad (1)$$

Since we have chosen to generate as many examples for D_* as for D_p in the mixture,

$$\begin{aligned} P(D_p) = P(D_*) &= 1/2 && (2) \\ P(x) = P(D_*)P(x|D_*) &+ P(D_p)P(x|D_p) && (3) \end{aligned}$$

and substituting (2) and (3) into (1) gives:

$$\begin{aligned} P(x|D_p) &= \left(P(D_p|x) \cdot \left(\frac{1}{2}P(x|D_p) + \frac{1}{2}P(x|D_*)\right)\right) / \frac{1}{2} \\ &= P(D_p|x)P(x|D_p) + P(D_p|x)P(x|D_*). \end{aligned}$$

From this, we easily get:

$$P(x|D_p) = \frac{P(x|D_*) \cdot P(D_p|x)}{1 - P(D_p|x)} \qquad (4)$$

Here, $P(x|D_*)$ is the learned prior and $P(D_p|x)$ is the learned differentiating function.

Having now explained how to learn a player-specific prediction function given a prior, the question remains as how to learn the prior. We learn the prior by (again) learning a differentiating function between a uniform distribution and the distribution formed by all examples collected from all players. Even though the uniform distribution is not accurate, this is not really a problem as sufficient training examples are available.

Experiments and Results

We observed cash games (max 9 players per game) played in an online poker room and extracted examples for players who played more than 2300 games. We randomly selected 20% of the games for the test set, while the remaining games were used to learn an opponent model. We learned one decision tree for all examlpes in the preflop phase and another for all remaning examples from the other phases, i.e, the *postflop* phases. The language bias used by TILDE (i.e., all possible tests for learning the decision tree) includes tests to describe the game history H_i at time i (e.g. game phase, number of remaining players, pot odds, previously executed actions etc.), board history B_i at time i, as well as tests that check for certain types of opponents that are still active in the game. For example, we may find tests such as "there is an opponent still to act in this round who is aggressive after the flop, and this player raised earlier in this game".

To evaluate our learning strategy, we report the log-likelihoods of the learned distributions and compare them with reasonable priors. A model with a higher likelihood directly allows an algorithm to sample and estimate the actions and outcome more accurately.

Figure 1 and 2 plot log-likelihoods averaged over 8 players for different training set sizes. The priors are clearly better than uninformed priors (i.e. not using learning). After having observed 200 games, in general the likelihood improves with the size of the training set.

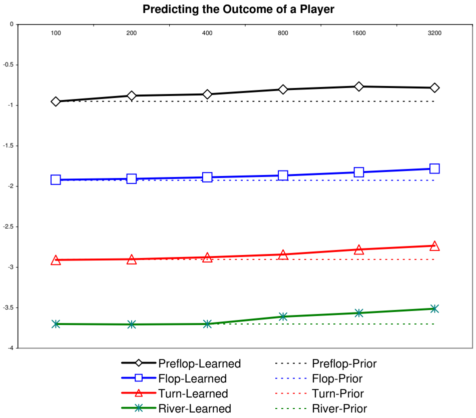

Figure 1: Experiment predicting the outcome, given board and game history. The *x*-axis represents the number of games used for the training set, and the *y*-axis the averaged log-likelihood scores on examples in the test set.

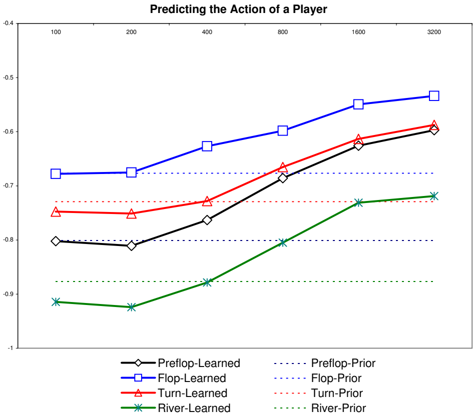

Figure 2: Experiment predicting the action, given outcome, board and game history. The axis are similar to those in Figure 1.

Conclusions

We presented a Bayes-relational opponent modeling system that predicts both actions and outcomes for human players in the game of No-Limit Texas Hold'em poker. Both these sources of opponent information are crucial for simulation and game-tree search algorithms, such as the adaptive tree search method by (Billings *et al.* 2006). The Bayes-relational opponent modeling approach starts from prior expectations about opponent behavior and learns a relational regression tree-function that adapts these priors to specific opponents. Our experiments show that our model adapts to specific player strategies relatively quickly.[2]

References

Billings, D.; Davidson, A.; Schauenberg, T.; Burch, N.; Bowling, M.; Holte, R. C.; Schaeffer, J.; and Szafron, D. 2006. Game-tree search with adaptation in stochastic imperfect-information games. In van den Herik, H. J.; Björnsson, Y.; and Netanyahu, N. S., eds., *The 4th Computers and Games International Conference (CG 2004), Revised Papers*, volume 3846 of *Lecture Notes in Computer Science*, 21–34. Ramat-Gan, Israel: Springer.

Blockeel, H., and De Raedt, L. 1998. Top-down induction of first order logical decision trees. *Artificial Intelligence* 101(1-2):285–297.

[2] Marc Ponsen is sponsored by the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. Jan Ramon and Kurt Driessens are post-doctoral fellow of the Research Foundation - Flanders (FWO).