Project Proposal: Hanabi

Xiuhan Wang, Jing Wei

January 7, 2021

1 Problem Definition

Hanabi is a game for two to five players. Each player holds four cards in his hand (or five if less than four players). Everyone can see all others' cards but not that of himself. Each card has a rank (1–5) and a color (red, green, blue, yellow or white). There are three 1s, two 2s, 3s, 4s and a 5 of each color; hence 50 cards in total. The players are fully cooperative—they want to play cards to build five stacks, one for each color, going consecutively from 1. The goal is to maximize the score, which is the total number of cards in the stack.

Players take turns to play. In one's turn, he must take exactly one of the following actions:

- **Hint.** To give a hint to others, the player chooses a color or a rank, then point out all cards matching that color or rank in another player's hand. Only ranks and colors that are present in the other player's hand can be chosen. To limit the number of hints, the group has initially eight information tokens and a token is consumed when giving a hint.
- **Discard.** When there are less than eight information tokens, he can discard a card from his hand, draw a new card from the deck and recover an information token. The discarded card is visible to all players and the newly drawn card is visible only to all other players.
- **Play.** The player can choose a card from his hand and try to put it in the stacks. It's successful if the rank is exactly one larger than the top of the stack corresponding to the color. If it's successful, put the card on the top of that stack. Recover one information token if there are less than eight and the rank played is 5. If the play is unsuccessful, discard the card and the group loses one life. When three lives are lost, the game ends immediately. Whether successful or not, the player draws a new card from the deck.

The game ends in one of three ways: when the stacks are completed, when three lives are lost or when the last card in the deck is drawn and every player has taken one final turn. The score is 0 if three lives are lost; otherwise it's the number of cards in the stacks, which is at most 25.

2 A Survey on Previous Works

Hanabi is a benchmark challenge in fully cooperative games of imperfect information; it's very different from adversarial two-player zero-sum games which have been studied more.[2] But non-zero-sum scenarios are more important in real life.

The Hanabi game was proved NP-hard even if all hidden information is revealed.[1] The hand-coded mathematical "hat guessing game" strategy[4] held state-of-the-art results from March 2016 until December 2019. In five-player games, it almost achieved the best performance resulted by a cheating strategy where each player sees his own cards and uses the information strategy. However, it performed badly in the two-player setting[3]; and it is not generalizable to similar games. A Monte-Carlo tree search algorithm is given by [9]; but the performance is still not very good.

Foerster and Song proposed a method of Bayesian action decoder (BAD)[5]. The key idea is a joint public belief over players' private features to resolve recursive belief over beliefs, and sampling a deterministic partial policy to resolve the dilemma between informative actions and stochastic actions for exploration. BAD worked well in the two-player setting. Hu and Foerster continued to optimize BAD to SAD[6] and incorperated search into SPARTA[8] to achieve a new state-of-the-art result.

Then they come up with the method called "other-play" (OP)[7] that work well on the zero-shot coordination scenario of the game. Previous studies on MARL usually focus on the self-play (SP) setting where agents are trained and tested both against with themselves, but agents naively trained by SP could do arbitrarily bad when paired with a partner they're not trained with. OP trains the agent with a copy of itself that is randomly permuted according to the symmetries in the game; hence it reduces the tendency towards arbitrary symmetry breaking. Agents trained with OP perform well when tested by cross-play, and also cooperate well with human.

3 Our Goals and Ideas

First we have to make the open-sourced algorithms and models of SAD with OP[7] (and SPARTA[8] if necessary) run on our machines. Since our computation resources are limited, our results might be not as good as that in the paper.

Then we can try to make some improvements. We have some ideas as following:

3.1 Automatically Detect Symmetries

One limitation of OP is that the symmetries in the underlying MDP must be given. What if we don't know the symmetries initially? Can we learn the symmetries while learning the policies?

Symmetries are essentially graph automorphisms. During searching we can apply hashing to calculate some graph automorphisms and learn the symmetries. In addition, embeddings of neighborhood structures calculated by graph convolution may produce a similarity meassure between nodes.

3.2 Better RL Algorithm for Zero-Shot Coordination

3.2.1 Symmetric Network

What OP essentially does is preventing agents away from policies that become very different when permuted. But we can do more—we can force the model to be invariant under symmetries. We propose a symmetric network where every nueron has its counterparts that share the same weights but take input from the previous layer according to a different permutation.

For the same reason why people design CNN to cope with the translation invariance in pictures, a symmetric NN should work well on this symmetry-invariant problem when we also want a symmetry-invariant answer. We suppose it can meanwhile reduce the number of parameters.

3.2.2 Incoporate Searching

The OP paper[7] didn't try to apply SPARTA-style searching. But we believe there is no difficulty to do it; and it can give a large boost also for cross-play settings, because we have already assumed our agents would learn similar policies.

3.2.3 Train a Population of Agents

Symmetries contribute a lot to the incompatibility issue between policies (and therefore the bad cooperation), but those are not all. For example, an OP agent may use rank hints to indicate playing and color hints to indicate discarding, while another agent may indicate the opposite. We think this is mainly because there are multiple local optimal solutions that are almost equally good with regard to OP (though they're unlikely exactly equal because they're not symmetric). Experiments in [7] shows that most of the OP agents come up with compatible policies; but a few agents cooperate badly and seem like local optimal solutions described above.

Perhaps we can train 2 or 3 agents by self-play and cross-play at the same time. The teammates that go to the more possible optimal points will have a good chance of pulling the bad cooperator out of its bias. The OP paper[7] tried population based approaches but failed. But we think it worth a trial to add on top of OP. Especially if we totally prevent agents from exploiting symmetries by symmetric network, the team trainning may have some effect in reducing exploitation of non-symmetries.

References

Baffier, J.-F., Chiu, M.-K., Diez, Y., Korman, M., Mitsou, V., van Renssen, A., … Uno, Y. (2017). Hanabi is np-hard, even for cheaters who look at their cards . *Theoretical Computer Science*, *675*, 43–55.

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., … Bowling, M. (2020). The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, *280*, 103216.

Bouzy, B. (2017). Playing hanabi near-optimally. In *Advances in computer games* (pp. 51–62).

Cox, C., Silva, J. D., Deorsey, P., Kenter, F. H. J., Retter, T., & Tobin, J. (2015). How to make the perfect fireworks display: Two strategies for hanabi. *Mathematics Magazine*, *88*(5), 323–336.

Foerster, J. N., Song, H. F., Hughes, E., Burch, N., Dunning, I., Whiteson, S., … Bowling, M. (2018). Bayesian action decoder for deep multi-agent reinforcement learning. In *International conference on machine learning* (pp. 1942–1951).

Hu, H., & Foerster, J. N. (2020). Simplified action decoder for deep multi-agent reinforcement learning. In *Iclr 2020 : Eighth international conference on learning representations.*

Hu, H., Lerer, A., Peysakhovich, A., & Foerster, J. (2020). "other-play" for zero-shot coordination. *arXiv preprint arXiv:2003.02979*.

Lerer, A., Hu, H., Foerster, J. N., & Brown, N. (2020). Improving policies via search in cooperative partially observable games. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(5), 7187–7194.

Walton-Rivers, J., Williams, P. R., Bartle, R., Perez-Liebana, D., & Lucas, S. M. (2017). Evaluating and modelling hanabi-playing agents. In *2017 ieee congress on evolutionary computation (cec)* (pp. 1382–1389).