A Highly-Parameterized Ensemble To Play Gin Rummy

Masayuki Nagai, Kavya Shrivastava, Kien Ta, Steven Bogaerts, Chad Byers
Department of Computer Science
DePauw University
Greencastle, IN, USA

DEPAUW UNIVERSITY

Abstract

In this work, we describe the development and tuning of a computer Gin Rummy player. The system includes three main components to make decisions about drawing cards, discarding, and ending the game, with numerous hyperparameters controlling behavior. After the components are described, three sets of hyperparameter tuning and performance experiments are analyzed.

Tracking the State of a Round

Self tracks each card's current state in the round:

- Increasing probability that card is in hand:
$$\forall a \in Ac\ \ p(a) \leftarrow p(a) - p(a) \cdot \beta$$
- Increasing probability that card is in stock:
$$\forall a \in Ac\ , p(a) \leftarrow p(a) + (1 - p(a)) \cdot \beta$$
- At each turn t, the estimated probabilities of remaining unknown cards are updated :
$$\Delta_t = \frac{|S_t|}{|Ut|} - \frac{|S_{t\_1}|}{|Ut_{\_1}|} = \frac{S_t}{|S_t| + |Ot|} - \frac{S_{t\_1}}{|S_{t\_1}| + |O_{t\_1}|}, \forall c \in Ut, p(c) \longleftarrow \Delta_t$$

Obtainability and Meldability

- The obtainability *ob(c)* of a card c is an estimation that self can obtain c: **ob(c) = αm$_1$· p(c)**
- The meldability m(c) of a card c is the estimated likelihood that c will eventually be part of a meld in self's hand.
- If c is in a rank meld, $m_{rank}(c) = 100$, otherwise:
$$m_{rank}(c_1) = ob(c_2)ob(c_3) + ob(c_3)ob(c_4) + ob(c_2)ob(c_4)$$
- If c is already in a run meld, $m_{run}(c) = 100$, otherwise:
$$m_{run}(c_1) = ob(c_a)ob(c_b) + ob(c_b)ob(c_d) + ob(c_d)ob(c_a)$$
- We can now define the overall meldability of c:
$$m(c) = \alpha_{m2}\ mr_{ank}(c) + \alpha m_3 \cdot mrun(c)$$

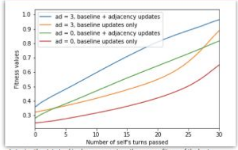
The Draw, Discard, and Knock Deciders

A player has three decisions to make: (1) whether to *draw* from the stock or discard pile, (2) which card to *discard*, and (3) whether to *knock*.

- A hybrid approach is used where in the early stage of a round ($t < \alpha d_0$), criterion (1) must be met in order to draw c_u. In the later stage, ($t > \alpha d_0$), criterion (2) is also sufficient to draw c_u.
- To make the *discard* decision, consider 11 *candidate* hands of 10 cards.
- Evaluate each h_i with an ensemble of hand evaluation policies.
- The ensemble's evaluation *eval(h)* of a hand *h* is a weighted sum of the evaluation *eval j(h)* of each member j: $eval(h) = \Sigma\ j\ eval_j(h) \cdot \alpha_{wj}$.
- Let $kn(t)$ be the knocking threshold, in which the player will only knock with hand h at turn t if $d(h) \leq kn(t)$.
- Define: $kn(t) = \begin{cases} \alpha_{t0}\ if\ t < \alpha_{t2}\ (early\ stage) \\ \alpha_{t1}\ otherwise\ (late\ stage) \end{cases}$

Tuning the State Tracking

- When opp draws face-up card c, the in-stock probabilities of adjacent cards are decreased.
$\forall a \in Ac_0, p(a) \leftarrow p(a) - p(a) \cdot \alpha_{s0}$
- When opp declines face-up card c, the in-stock probabilities of adjacent cards are increased.
$\forall a \in Ac_0\ p(a) \leftarrow p(a) + (1 - p(a)) \cdot \alpha_{s3}$



In tuning the state tracking hyperparameters, the average fitness of the best individual based on number of turns, for two fitness functions (ad= 0and3) and two prob-ability update types

- Tuning the nine hyperparameters with a genetic algorithm.
- Selected cards form the set G_{ad}, on which the fitness is calculated:
$$Fitness_{ad}(Gad) = |U_o \cap Gad| / |U_o|$$
- Eight players play 24,000 games against the simple player.
- As the number of turns increases, fitness increases, suggesting that the baseline probability adjustment is effective.
- Comparing the two lines with ad= 0, the adjacency updates result in a consistently higher fitness than the corresponding baseline. Similar results are seen for $ad = 3$.
- This suggests that adjacency updates provide additional accuracy over the baseline.

Tuning the Knock Decider

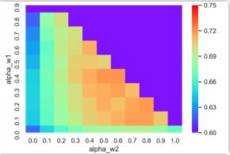
	α_{t1} 0	1	2	3	4	5	6	7	8	9
α_{t0} 1	0.48									
2	0.50	0.45								
3	0.51	0.46	0.430							
4	0.50	0.45	0.40	0.40						
5	0.50	0.47	0.41	0.40	0.36					
6	0.52	0.45	0.43	0.39	0.37	0.33				
7	0.51	0.49	0.44	0.44	0.40	0.35	0.33			
8	0.54	0.50	0.45	0.43	0.39	0.34	0.33	0.32		
9	0.54	0.53	0.47	0.45	0.41	0.37	0.36	0.34	0.32	
10	0.57	0.53	0.50	0.48	0.43	0.39	0.38	0.34	0.33	0.32

In tuning the knock decider hyperparameters, win rate for each allowed α_{t0} and α_{t1} pair, for fixed α_{t2}= 6

- Grid search, one-stage knock decider, setting α_{t2}= 0 and vary α_{t1}.
- Consider every combination of two such players: 11·11 = 121 pairings corresponding to the 11 possible values for each player's αt1
- The best one-stage knocking strategy is α_{t1}= 0: knock only on gin.
- For tuning the hyperparameters of two-stage knock decider, the players with two modified simple players are created: one using a two-stage knock decider, and one knocking only on gin (α_{t2}= 0,α_{t1}= 0).
- For the two-stage player, consider 825 different possibilities by varying the integer hyperparameters with $\alpha_{t2} \in [5,19]$ and $\alpha_{t0}, \alpha_{t1} \in [0,10]$subject to αt1< αt0. Each pair of these players plays 2,000 games.
- The highest winning rate, 0.57, is found with (α_{t0},α_{t1},α_{t2}) = (10, 0,6).
- Best two-stage player beat best one-stage player (knock-on-gin) 57% of the time, suggesting that two-stage knocking with the correct hyper parameterization is more effective than one-stage knocking.

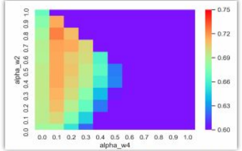
Tuning Remaining Hyperparameters

- hyperparam. subsets: (1) meldability (2) adjacency (3) deadwood penalty by turn count (4) deadwood penalty by opp cards known (5) ensemble weights.
- The initially guessed hyperparameters are called base$_1$. The base$_s$ values and the variation of values of each α∈ As generates a set of value assignments Vs.



Correlation between α_{w1} (meldability hand evaluation policy) and α_{w2} (lowering deadwood hand evaluation policy)

- For each v ∈ V_s, player v plays 5,000 games against the simple player.
- Seeking the hyperparameter values that maximize the win proportion.
- After completing iteration 1, the win rate of the best hyperparameter values is 0.739 over the 0.701 win rate using base$_1$



Correlation between α_{w2}(lowering deadwood hand evaluation policy) and α_{w4} (hand evaluation policy)

- The hypothesized pre-eminence of reducing deadwood is supported in these results, in which the combined weight of these three policies $\alpha_{w2}+\alpha_{w5}+\alpha_{w6}$= 0.6.
- The weight of meldability (policy 1),α_{w1}= 0.2, α_{w3}= 0.1 for the ace-two bonus, and α_{w4}= 0.1 for in-hand adjacency (policy 6).

Conclusion

This research presents an effective strategy for a computer Gin Rummy player composed of three deciders: a draw decider, discard decider, and knock decider. The draw decider considers whether the face-up card is expected to be melded, and in later turns, also whether the card would simply reduce deadwood. The discard decider is an ensemble of hand evaluation policies, considering a wide range of factors including deadwood points, expected melds based on the observed state of the round, and adjacency of cards in hand. The knock decider determines whether to end the round, based on a deadwood point threshold that varies depending on whether the round is in an early or late stage. Finally, three sets of experiments were conducted to optimize the hyperparameters that govern these decisions: (1) a genetic algorithm for tuning the state tracking hyperparameters, (2) a grid search to find the thresholds of deadwood points for knocking at both round stages, and when to transition from early to late stage, and (3) a grid search for the hyperparameters of the hand evaluation policies and their weights in the ensemble.

Acknowledgments

We offer sincere thanks to the DePauw Faculty Development program for funding for this work, and to Dr. Michael Roberts and Dr. Naima Shifa for their assistance in statistical analysis.