



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
PROFESOR: DENIS PARRA
IIC3633- SISTEMAS RECOMENDADORES
SEGUNDO SEMESTRE - 2024

HearthStone Deck Recommendation System

Gustavo Cornejo Olea - Juanita Fernández - Fransisco Jorquera

1. Descripción y justificación del problema

El proyecto se desarrolla dentro del contexto de juegos de cartas coleccionables (CCG) y en particular en el caso del juego Hearthstone. Hearthstone es un juego *online* y *free-to-play* de la compañía Blizzard Inc. que para 2016 ya había alcanzado el total de 40 millones de cuentas registradas y que consiste en partidas entre dos jugadores, donde cada uno tendrá un mazo de 30 cartas y 30 puntos de vida y, por turnos, avanzarán durante la partida manejando sus cartas con el fin de atacar al contrincante hasta acabar sus puntos de vida.

La justificación de este proyecto viene de la cantidad de cartas y variables a considerar dentro del mazo. Es importante considerar que cada jugador elige un personaje que tendrá su propia dinámica o “poder de héroe”, la dinámica puede consistir en jugar una determinada carta o aumentar puntos de salud. Tendrán que armar un mazo con cartas compatibles con su personaje, de un total de más de 5000 cartas, las cuales se dividen en aliados y hechizos. Los aliados son personajes que se mantienen a lo largo de la partida y permiten atacar al adversario o a los aliados del mismo, mientras los hechizos tienen diversos efectos en la partida, pero desaparecen automáticamente.

2. Objetivos del proyecto

El objetivo global del proyecto es implementar un sistema de recomendación que, a partir de un mazo inicial entregado por el usuario, entregue un nuevo mazo con buen desempeño en las métricas que se propondrán.

3. Análisis descriptivo de los datos

La base de datos se obtuvo de la plataforma HearthPwn (<https://www.hearthpwn.com/decks>), esta contiene información sobre mazos creados por jugadores de Hearthstone. Los datos incluyen la clase del mazo, sus 30 cartas, el coste de creación, la puntuación, entre otros, pero dentro de las columnas más importantes está el rating que asignaron múltiples jugadores en la plataforma. En el gráfico 1 del anexo se muestra la puntuación promedio de los mazos según clase de héroe.

Para abarcar con menor complejidad el problema se decidió trabajar solo con cartas de la clase de héroe más frecuente, mago. Esto se visualiza en el gráfico 2 del anexo.

4. Definición del problema y solución

El problema se abordó con optimización combinatorial. Definimos como función objetivo maximizar el win rate de un mazo de cartas dado, sujeto a un máximo de cambios. Para definir una función de win rate se construirán modelos predictores con machine learning.

5. Desarrollo de la solución

Para el desarrollo de la solución se comenzó definiendo un conjunto de cartas con las N más repetidas en los mazos mago, y el tamaño del mazo para armar, que en el caso de Hearthstone es 30. El mazo por mejorar se definió como un vector binario con N entradas y exactamente D iguales a 1. Este vector nos indica con la i-ésima entrada si está o no (1 o 0) en el mazo la i-ésima carta del conjunto de N cartas.

El proceso para mejorar el mazo consta de aplicar múltiples veces una función de transición que modifica exactamente una carta del mazo por una carta que actualmente no está incluida. Para elegir las acciones que nos lleven a encontrar un mazo ganador usaremos el win rate de este mazo. Básicamente se implementa un algoritmo glotón que decide la siguiente acción como la que maximiza el win rate. Se realizan las acciones hasta que ya no sea conveniente seguir modificando el mazo, estas serán a los más D, ya que a lo más cambiamos todo el mazo para encontrar el óptimo.

Para hacer una función predictora del win rate entrenamos distintos modelos de machine learning, una red neuronal y random forest. Dentro de esto, implementamos random forest con $n_estimators=50, 100, 150$ y distintas 4 redes neuronales con distintas cantidades y tipos de capas. Finalmente se conservaron solo un modelo de random forest y una red neuronal, los cuales conseguían los mejores resultados en sus propias categorías, y para la implementación del algoritmo *DeckRec* se utilizó solamente la red neuronal dado que su rango de valores al predecir una nueva muestra era considerablemente más amplio.

6. Experimentación realizada

A modo de experimentación se comparó la implementación recién descrita con los métodos creados para la entrega anterior. Estos son random, que entrega de forma aleatoria un mazo de cartas de la base de datos; most popular, que retorna el mazo con el mejor rating de la base de datos; y best similar, que recibe un mazo y retorna un mazo similar de la base de datos con mejor rating. Para definir la similitud, se implementó una función que calcula la cantidad de cartas diferentes entre los mazos. De esta forma, el modelo también recibe como parámetro un delta que define la máxima posible diferencia entre el mazo retornado y el original.

Lo anterior, se implementó para múltiples mazos generados de forma aleatoria con distintos parámetros delta y N, que es el total de cartas sobre las que se eligen para construir un mazo.

7. Análisis preliminar de los resultados obtenidos

El Cuadro 1 muestra los resultados de win rate por método para 10 mazos distintos, la columna Initial tiene el win rate inicial para comparar con los recomendados.

Initial	Random	MP	BS	DeckRec
57.161957	53.423840	53.383629	53.383629	59.073711
52.291637	53.175045	53.383629	53.383629	59.073711
56.342117	54.633308	53.383629	51.934380	60.453484
53.423840	54.963871	53.383629	53.383629	60.912884
54.963871	51.934380	53.383629	53.383629	60.453484
52.291637	53.491936	53.383629	53.383629	59.073711
53.062859	56.342117	53.383629	54.643101	59.073711
54.643101	53.491936	53.383629	54.784740	60.912884
56.324768	53.423840	53.383629	56.324768	60.453484
54.963657	54.963871	53.383629	53.383629	60.453484

Cuadro 1: Win rate de los mazos resultantes por método

El Cuadro 2 muestra los aumentos en el win rate con respecto al mazo inicial, se calcula la razón entre el win rate del método y el inicial.

Random	MP	BS	DeckRec
0.9346	0.9339	0.9339	1.0334
1.0169	1.0209	1.0209	1.1297
0.9697	0.9475	0.9218	1.0730
1.0288	0.9992	0.9992	1.1402
0.9449	0.9712	0.9712	1.0999
1.0230	1.0209	1.0209	1.1297
1.0618	1.0060	1.0298	1.1133
0.9789	0.9770	1.0026	1.1147
0.9485	0.9478	1.0000	1.0733
1.0000	0.9713	0.9713	1.0999

Cuadro 2: Aumento relativo frente a *initial* (baseline/initial)

Ahora, con el Cuadro 3, comparamos como cambia el win rate a medida que aumentamos N. Se observa que a medida que tenemos más cartas de las cuales elegir, mejora la efectividad del mazo.

	initial	40	55	70
0	56.342117	60.912884	60.161621	65.126862
1	54.963871	60.912884	61.246162	60.161621
2	54.633308	60.453484	59.497566	62.584610
3	54.633308	60.453484	65.126862	65.271690
4	56.324768	59.073711	60.971478	63.977943
5	53.272411	59.073711	60.971478	60.790009
6	54.784740	60.453484	59.497566	60.912884
7	53.491936	60.453484	60.971478	65.536736
8	53.423840	60.453484	58.418598	63.796070
9	56.324768	59.073711	59.497566	60.453484

Cuadro 3: Tabla con win rates para distintos N

Finalmente, analizamos los tiempos de la implementación para un mazo aleatorio para cada N en el Cuadro 4.

N	Tiempo (segundos)
40	65.296554
55	97.527588
70	287.718022

Cuadro 4: Tiempo de ejecución para diferentes valores de N.

A pesar de que las tablas muestran que DeckRec logra consistentemente los mayores incrementos en win rate en comparación con los métodos random, most popular (MP) y best similar (BS), su tiempo de ejecución representa una desventaja significativa. Este tiempo de respuesta considerablemente más alto limita su aplicabilidad en contextos donde se requiere rapidez, lo que sugiere que, aunque es el método más efectivo en términos de precisión, su alta latencia reduce su eficiencia práctica en escenarios de tiempo real.

8. Problemas identificados durante el proceso

Primero destacamos dos situaciones que son propias de la formulación del problema y representan obstáculos importantes. Por un lado, está la gran cantidad de cartas y maneras de complementarlas, lo cual implica, un problema de optimización combinatorial muy complejo y que no fue posible evitar completamente. Los efectos de este obstáculo se ven en el tiempo de cómputo del algoritmo “DeckRec”, el cual fue considerablemente mayor que el de los demás algoritmos. Por otro lado, la inexistencia de una clase o estrategia ganadora hace aún más difícil la generación de un mazo adecuado para una situación aleatoria.

Entrando ahora en el detalle del proceso, el principal obstáculo encontrado fue el cálculo del winrate de un mazo. Esto fue una dificultad agregada respecto a lo expuesto por [Chen et al.2018] ya que, utilizaban el software *MetaStone* que permitía la simulación de varias partidas en que las jugadas eran realizadas por distintos modelos de inteligencia artificial pre-entrenadas. Sin embargo, este software parece haber dejado de estar disponible ya que aunque intentamos implementarlo, esto no fue posible. De aquí nació la propuesta de usar *machine learning* para la predicción del win rate. En este caso si bien desarrollamos diversos modelos, ninguno funcionó satisfactoriamente para un calculo realista.

Nos parece importante destacar que este cálculo es bastante complejo al punto de ser parte de la competencia de minería de datos AAIA’18 organizada por Silver Bullet Labs y Knowledge Pit como una parte de la conferencia the FedCSIS 2018.

9. Revisión del plan propuesto en etapa anterior y justificación de ajustes

El plan propuesto en la etapa anterior era el siguiente:

1. Definir métricas (18 al 25 de Octubre)
2. Implementar método Q-DeckRec (14 al 23 de Octubre)
3. Comparar métodos (23 de Octubre al 4 de Noviembre)
4. Escribir informe (1 al 8 de Noviembre)

En general, se realizaron ajustes a la propuesta. En primer lugar, para la métrica no se pudo usar el software *MetaStone*, por lo tanto, se tuvo que usar *machine learning* y la definición e implementación del win rate tomó más tiempo del esperado. En segundo lugar, por lo mismo, no se pudo implementar el método Q-DeckRec, por lo tanto se creó una variación de este método. Finalmente, la comparación de métodos se realizó de la forma esperada.

Referencias

- [Chen et al.2018] Chen, Z., Amato, C., Nguyen, T.-H. D., Cooper, S., Sun, Y., and El-Nasr, M. S. (2018). Q-deckrec: A fast deck recommendation system for collectible card games. page 1–8.

Anexo

Gráfico 1

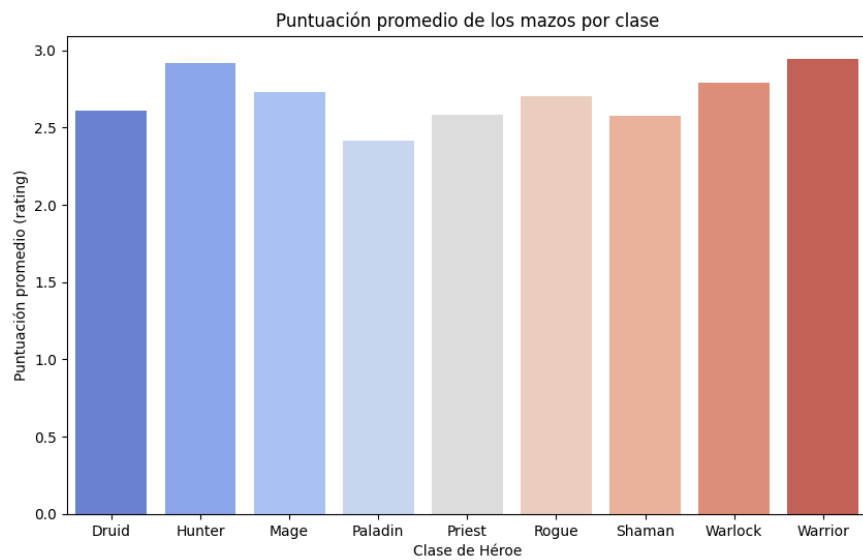


Gráfico 2

