Pontificia Universidad Católica de Chile
Departamento de Ciencias de la Computación
Curso: IIC3633 - Sistemas Recomendadores
Semestre: 2023 - 2
Profesores: Andrés Carvallo

Recomendación Multi Modal para Juegos de Mesa

Jueves 14 de Diciembre

Sara Godoy
Álvaro Postigo

Índice

1. Abstract

Los juegos de mesa son una forma clásica de pasar tiempo jugando en persona con familia y amigos, sin embargo, por la naturaleza física de estos, ellos se están quedando atrás con la digitalización, especialmente con los video juegos y sus plataformas de venta y recomendación online. Por lo anterior, se creó un modelo de recomendación multi modal de juegos de mesa en base a los datos recopilados de BoardGamesGeek, sitio web que tiene información de ratings de usuarios hacia distintos juegos y características de estos juegos. Estas se encuentran en formatos de imagen, descripciones en texto y datos tabulados. Para aprovechar toda la información disponible es que se propuso una red neuronal capaz de utilizar toda la información disponible para predecir los rating que un usuario de daría a cada juego. Esta red es compuesta por múltiples sub-redes capaces de lidiar con cada formato de data. Se utiliza una red convolucional para extraer features de las imagenes, el encoder de un transformer encoder-decoder para extraer features de las descripciones textuales y MLPs para extraer features de la información tabulada. La red resultante combina todas las features extraídas para luego combinarlo con un vector de usuario para predecir el rating que este le daría. El modelo es entrenado en una muestra de un conjunto de millones de rating y obtiene resultados de predicciones considerablemente mejores que metodos tradicionales, llegando incluso a un RMSE de *0.16*.

2. Introducción y Estado del Arte

El contexto del problema se centra en la limitación de las opciones disponibles para descubrir nuevos juegos de mesa. Debido a la naturaleza física de estos juegos y la falta de sistemas de streaming o plataformas digitales dedicadas, no se han desarrollado sistemas de recomendación basados en filtrado colaborativo, lo que resulta en una escasez de herramientas para ayudar a los entusiastas de los juegos de mesa a descubrir nuevos títulos que se adapten a sus preferencias. En lugar de depender de sistemas de recomendación automatizados, los jugadores a menudo se ven obligados a elegir nuevos juegos basándose únicamente en las categorías que ya conocen y disfrutan, lo que puede llevar a pasar por alto oportunidades de descubrimiento de juegos que podrían resultar de su agrado. Este problema ilustra la necesidad de desarrollar soluciones que permitan a los amantes de los juegos de mesa acceder a una mayor diversidad de opciones y descubrir títulos que se ajusten a sus gustos de manera más eficiente y efectiva.

3. Solución

La solución que se propone, es realizar un sistema de recomendación basado en la predicción de ratings que un usuario le daría a un juego en particular. Nuestro modelo debería aprender los gustos y preferencias de cada usuario de manera implícita por medio de ratings pasados. Para ello, nuestro modelo debe ser capaz de entender todas las cualidades de un juego de mesa para tratar de capturar las razones de un rating para un

usuario en particular. Para capturar de manera completa todas las cualidades de un juego de mesa es que se plantea un trabajo multimodal entre información categorica, información numerica, texto e imagenes. Para esto se pueden combinar distintas estructuras de modelos de *Deep Learning* para luego combinar sus distintos resultados en una red profunda final [Gadzicki et al., 2020]. Entonces, a partir de toda esta información de un juego y un ID de usuario, el modelo debe entregar un rating que debiese ser equivalente al rating que ese usuario le daría a ese juego basados en sus gustos. Si el modelo es capaz de comprender bien los detalles extraídos por los *embeddings*, entonces al entrenarse debiese poder aprender en sus pesos los gustos de cada usuario, como se ha logrado en estructuras similares [Tato and Nkambou, 2023]. La descripción mas detallada de cada sección y el modelo completo se encuentran en la sección de "Metodología".

4. Dataset

Utilizamos un dataset con información extraída de BoardGameGeek el cual posee caracteristicas categoticas y numericas de cada juego, ademas de una descripción textual y una imagen de la caja o portada. Entre las características están, las categorías, mecánicas del juego, subcategorías, temas, entre otras. Ademas de tiene información determinada por la comunidad, tales un rating promedio, cantidad de jugadores recomendado, tiempos recomendados, etc.

Para entrenar las predicciones que llevaran a las recomendaciones, el dataset también incluye una tabla de historial de ratings hechos por usuarios. Esta tabla posee mas de 19 millones de ratings por mas de 411 mil usuarios. Debido al surplus de datos posibles a utilizar, se realizará un corte para poder controlar tiempos de entrenamiento. También debido a la falta de distribuciones de entrenamiento y prueba, estos se tuvieron que construir tambien. Para ello se extrayeron usuarios al azar pero que cumplieran dos criterios: tener al menos 10 reviews y que su promedio de reviews fuese mayor a 5. De esta manera cada usuario tendrá suficientes ratings para que el modelo pueda aprender sus gustos y también filtramos ciertos usuarios que solo poseian reviews negativas; es dificil recomendarle algo que le guste a una persona si solo le disgustan cosas. Una vez tubiesemos 100 usuarios que cumpliesen estos requisitos, entonces extraemos todos sus reviews para crear nuestra muestra inicial. Ahora, por cada usuario extraemos uno de sus ratings mayor a 7 y así creamos nuestra data de prueba. Estos seran ratings que nuestros 100 usuarios dieron que el modelo nunca aprenderá. El resto de la muestra, entonces, funcionará como nuestro set de entrenamiento. El set resultante posee alrededor de 25 mil reviews de alrededor de 7 mil distintos juegos.

5. Metodología

El primer paso para realizar este proyecto, es vectorizar los datos para mediante a un *embedding* para poder tener la información en el mismo formato y poder hacer predicciones en base a estos. Estos *embeddings* seran

modelos distintos dependiendo del input que se trata de vectorizar: texto, imagen, información categorica, información numerica. Basandonos en le cocepto de *transfer learning*, si utilizamos modelos pre-entrenados en tareas genericas, estos poseen pesos y extraen *feature maps* ricos en información; esta es una tecnica estandar de *Deep Learning* y sigue siendo utilizada hasta el dia de hoy[Iman et al., 2023]. Por lo tanto, se utilizaran modelos especificos a cada input correspondiente con sus pesos congelados de tal manera que cumplan el papel de *encoder* solamente.

Para las descripciones que son texto, se utilizará un modelo basado en *transformers* de estructura *encoder-decoder*. Mas específicamente, se utilizará un modelo T5 [Raffel et al., 2023] pre-entrenado en tareas de NLP como *encoder* para las descripciones de cada juego. Ya que solo queremos utilizar este modelo como *embedding*, solo nos quedaremos con el *encoder* y se utilizará el output de su última capa oculta como output del *embedding* de texto.

Luego para las fotografías se utilizará una ResNet50 [He et al., 2015] pre-entrenada en ImageNet [Russakovsky et al., 2015] para codificar la imagen y poder utilizarla en el sistema de recomendación.

Para terminar con los tipos de datos, se entrenará un MLP que distinga entre las variables numericas y categoricas de las tablas de información, de tal manera que nos de un *embedding* acorde. Este tendrá que ser entrenado desde cero.

Una vez con las tres redes mencionadas anteriormente, se juntarán los resultados de estos en un transformador multi modal , denominado *Multimodal Combiner* para concatenar la información de las tres redes y presentar la información de una manera manejable. De manera mas especifica, los distintos output de los *encoders* serán transformados a vectores de igual tamaño (256), por medio de capas lineales o convuluciones dependiendo de la estructura, para luego ser concatenados entre ellos. Todos los parametros de esta pieza son entrenables en el ciclo de entrenamiento. Se puede apreciar en el siguiente cuadro un resumen de que dato obtendrá cada *encoder*.

Tipo de dato	***Encoder***
Características Numéricas	MLP
Características Categóricas	MLP
Descripciones (texto)	T5
Imágenes	ResNet50

Cuadro 1: Resumen tipo de dato y *encoders*.

El resultado del *Multimodal Combiner*, entonces, es concatenado al vector de usuario para luego ser introducido a nuestra red final para predecir un rating. Como vector de usuario se entiende un vector de largo 256 que identifique al usuario entregado por un *embedding* que funciona principalmente como una *look up table*. Esto se hace así para que el modelo no aprenda el ID de usuario como un valor numérico. El rating predicho es comparado con el rating real dado por el usuario para así calcular un error y propagarlo por toda la red. Como optimizador se utilizó *Adam* [Kingma and Ba, 2017] para un set up mas simple.

En las siguientes figuras se puede apreciar gráficamente el transformador multi modal y como se relacionaría con la información del usuario y el modelo de predicción.

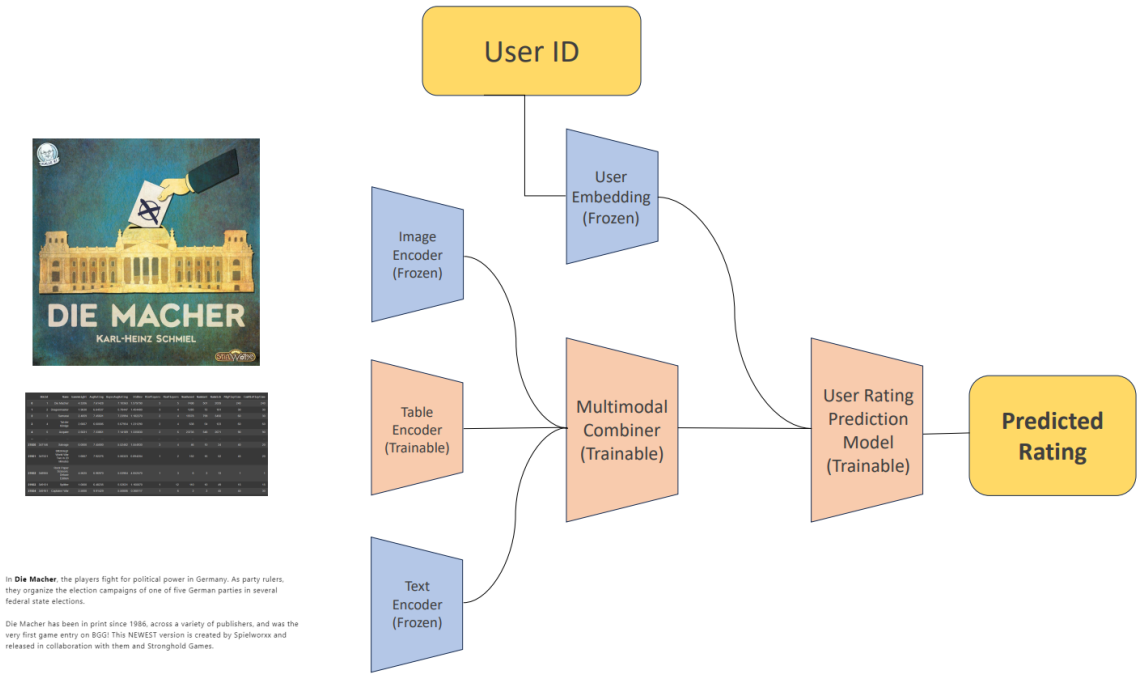

Figura 1: Diagrama Modelo

Adicionalmente se incluye la totalidad del código utilizado como un notebook en Google Colab. Este codigo incluye toda la re-estructuración de la data, la creación del modelo y sus respectivas secciones, el entrenamiento y testeo. Todo esta modularizado de tal manera que pueda fácilmente modificado a distintos tipos de encoders o estructuras de MLP.

`https://colab.research.google.com/drive/1HpF40SbSM-ixld7nMMgYs7aD5UtwtJ12?usp=sharing`

5.1. Evaluación

En la evaluación, primero se establecerá una *baseline* con recomendaciones *Bayes Average Rating* en que se obtendrían los juegos con el valor más alto de este atributo, que toma en cuenta el *rating* promedio del juego y la cantidad de *ratings* que tiene este. Además del *Bayes Average Rating* se utilizará la Factorización Matricial, estos resultados se obtendrán como base con el objetivo de mostrar que nuestro modelo es efectivo y realiza buenas recomendaciones.

Finalmente, como se tiene información sobre el rating de los usuarios, se puede separar los datos en *train* y *test*, con esa separación, podemos ver la precisión del modelo al realizar las predicciones. Si esta presición es lo suficientemente alta, se puede concluír que el modelo puede recomendar correctamente juegos a partir de la información que se expone en el *dataset*. Vale mencionar que se deberá separar la data para recomendar como para entrenar al modelo. Para ello, luego de realizar los cortes planteados, se deben extraer ratings positivos para tener un *testing* de que debería salir en las recomendaciones. Con lo que queda de la data, se

debe separar nuevamente en *train* y *test* para poder medir la predcción del modelo. Así tenemos data en que testear las predicciones y las recomendaciones, que el modelo nunca verá al entrenarse para predecir.

Finalmente es importante mencionar que se utilizará la métrica de *Root-mean-squared error* o RMSE para evaluar el desempeño de nuestro modelo y de las *baselines*.

6. Resultados

Los resultados de tanto las *baselines* como el modelo se ven a continuación.

Modelo	**RMSE**
Bayes Average Rating	1.1667
Factorización Matricial	1.1289
Trained Model	0.1637

Cuadro 2: Resultados obtenidos.

Con el cuadro anterior se puede observar que los resultados entregados por las *baselines* nos dicen que los *ratings* varían entre un 1.13 y un 1.17 puntos aproximadamente con estos modelos. Esto si bien no es una cifra tan alta para la escala de 1-10, sigue siendo un resultado que puede ser mejorado bastante como se ve con el valor de RMSE del modelo propuesto que es el *Trained Model*, obteniendo una variación de 0.1637 puntos de *rating*, considerablemente menor a los otros.

La gran cantidad de información que obtiene el *Trained Model* lo ayuda a desempeñar de mejor forma que los otros con el mismo *test set*. Vale mencionar que la distribución de los ratings por usuario no era equivalente, es decir, algunos usuarios poseían mas cantidad de ratings que otros. Esto usualmente debería causar un error mucho mas alto en el testeo, pero como podemos ver esto no afecto de gran manera al modelo. Se propone una mayor investigación de este efecto en los resultados.

Ahora, si bien los resultados obtenidos son increíbles, estos tomaron un tiempo considerable de entrenamiento. La teoría nos dice que entre mayor cantidad de data, mejores resultados se obtendrán. En este caso, esto parece estar demostrado. A cambio, el modelo requiere de mucho tiempo para entrenarse, la mayoría de este tiempo siendo dedicado la búsqueda de información para ser entrenado el modelo. Por ello, estos resultados sirven como prueba de concepto de nuestra estructura pero para ver toda su capacidad se requiere de una arquitectura computacional mas fuerte que la utilizada.

7. Conclusiones

Este método continua los avances vistos en los últimos años: entre más información, mejores modelos. Esto viene con un costo, el lidiar con esta información. Al momento de entrenar el constante traspaso de información entre CPU y GPU causa una demora inevitable que aumenta en tiempo correspondiente al tamaño de datos. Esto se ve claramente cuando al entrenar el modelo pasa más tiempo buscando los datos que actualizando los pesos del modelo. Esto se puede solucionar con hardware especializado para disminuir este *overhead*.

Si bien el *dataset* es increíblemente completo, ciertas estructuras de este dificultan la creación de tanto modelos de recomendación, como también el aprendizaje de máquinas. Hay mucha data que limpiar, las imágenes no se encuentran junto al *dataset*, y no hay matrices de comparación.

Aun así, se obtuvieron resultados que validan el poder de la estructura propuesta. Mas aún, debido a la estructura modular de este modelo, se presta mucha para la investigación de como estas redes neuronales funcionan. Se propone como futura investigación revisar los aportes hechos por cada sección de manera individual. Así podríamos ver que tan útil fue cada sección al momento de predecir el rating final. Incluso llendo a mas detalle, debido a los mapas de atención utilizados por el encoder de texto, se podría investigar de cada palabra en particular por medio de *saliency maps*. Algo similar se podría aplicar también a las redes convolucionales utilizadas en las imagenes por medio de metodos como *GradCAM*.

Hablando de posibles aplicaciones de este modelo, este podría ser usado por la misma página *BoardGamesGeek* para realizar recomendaciones a sus usuarios y que el sistema perdure en el tiempo a diferencia del sistema en base a ratings que intentaron de implementar pero falló. El poder predecir el rating que cada usuario le daría a cada juego es una fuerte herramienta para aprender del gustos de este. El contrapeso a esto es que se debe aplicar a cada juego para poder ver aquellos que posee mayor pontajes, esto requiere una inteligente estructura para hacerse de manera eficiente pero que daría recomendaciones óptimas si el modelo realmente aprendió los gustos de los usuarios; como los resultados nos dicen.

Referencias

[Gadzicki et al., 2020] Gadzicki, K., Khamsehashari, R., and Zetzsche, C. (2020). Early vs late fusion in multimodal convolutional neural networks. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–6.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

[Iman et al., 2023] Iman, M., Arabnia, H. R., and Rasheed, K. (2023). A review of deep transfer learning and recent advancements. *Technologies*, 11(2):40.

[Kingma and Ba, 2017] Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

[Raffel et al., 2023] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2023). Exploring the limits of transfer learning with a unified text-to-text transformer.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge.

[Tato and Nkambou, 2023] Tato, A. and Nkambou, R. (2023). Towards a multi-modal deep learning architecture for user modeling. In *The International FLAIRS Conference Proceedings*, volume 36.