Recomendación de cervezas con máquinas de factorización

DANIELA FLORES and HERNÁN VALDIVIESO

En este paper se detallan todos los pasos que fueron seguidos para construir un sistema recomendador basado en máquinas de factorización. En primer lugar, se separó el *data set* en una parte para entrenamiento y otra destinada a pruebas de desempeño, según una proporción 7:3. Posteriormente, se utilizó validación cruzada con 5 *folds*, para validar hiper-parámetros, mediante la reducción de RMSE. Finalmente, se entrenó un modelo de máquinas de factorización para predecir *ratings* y, con esa información, generar recomendaciones. Para evaluar el desempeño del modelo, se utilizaron las métricas *recall*, *nDCG@10* y *MAP@10*, calculadas para dos escenarios: uno con *cold start* y uno sin él. Cuando se comparó el modelo entrenado en el marco de este proyecto con los destinados a servir de *baseline* (*Random*, BPR y *Most Popular*), se obtuvo que el desempeño del primero era muy pobre en comparación a los demás, a excepción de *Random*. A futuro, se desea intentar explicar por qué métodos no personalizados obtienen mejores resultados que el modelo propuesto. Además, resulta relevante incluir *features* de texto al modelo, para estudiar si se observa o no un incremento en su desempeño.

ACM Reference Format:
Daniela Flores and Hernán Valdivieso. 2018. Recomendación de cervezas con máquinas de factorización. 1, 1 (December 2018), 6 pages.

1. INTRODUCCIÓN

Los modelos de máquinas de factorización han sido ampliamente usados para generar recomendaciones en una gran variedad de dominios. Esto sirvió como motivación de este trabajo, que buscó estudiar qué tan bien se comporta uno de estos modelos cuando se utilizan distintos tipos de datos. Esta es la razón por la cual se escogió el *data set* de BeerAdvocate para ejecutar los experimentos, pues, como se detallará más adelante, este ofrece diferentes tipos de *ratings* numéricos que los usuario asignaron a los productos. Además, los usuarios entregaron una reseña para los ítems en forma de texto.

1.1. Data set

BeerAdvocate es un sitio web de calificación de cervezas fundado en 1996. En el *data set* extraído de la plataforma que se empleó en este proyecto, existen diferentes tipos de *ratings* numéricos, tales como una evaluación *overall* que entregó un usuario a una cerveza y evaluaciones de un aspecto particular de la misma (sabor, sensación, olor, apariencia). Además los usuarios tienen la oportunidad de escribir una reseña en lenguaje natural de la cerveza evaluada. Las columnas del *data set* son [1]:

1. `brewery_id` (valor continuo numérico): identificación de la cervecería productora.
2. `brewery_name` (valor nominal): nombre de la cervecería productora.
3. `review_time` (valor continuo de punto flotante): timestamp del review del usuario.
4. `review_aroma` (valor numérico de cinco niveles entre 1 y 5): calificación del aroma de la cerveza.
5. `review_appearance` (valor numérico de cinco niveles entre 1 y 5): calificación de la apariencia de la cerveza.

Authors' address: Daniela Flores, diflores@uc.cl; Hernán Valdivieso, hfvaldivieso@uc.cl.

Sistemas Recomendadores/2018/12-ART $15.00

6. `review_profilename` (valor nominal alfanumérico): nombre del usuario calificador. Esta columna se utiliza para los usuarios en el modelo.
7. `review_palate` (valor numérico de cinco niveles entre 1 y 5): calificación del sabor en paladar de la cerveza.
8. `review_taste` (valor numérico de cinco niveles entre 1 y 5): calificación del sabor de la cerveza.
9. `beer_abv` (valor continuo de punto flotante): alcohol por volumen de la cerveza.
10. `beer_style` (valor nominal alfanumérico): nombre del estilo de la cerveza.
11. `beer_name` (valor nominal alfanumérico): nombre de la cerveza.
12. `beer_beerid` (valor continuo): identificación de la cerveza. Esta columna se utiliza para los ítems en el modelo.
13. `review_overall` (valor numérico de seis niveles entre 0 y 5): clase indicadora del nivel de calidad de la cerveza. Este es el valor que se buscará predecir con la máquina de factorización.
14. `review_text`: comentario del usuario sobre la cerveza.

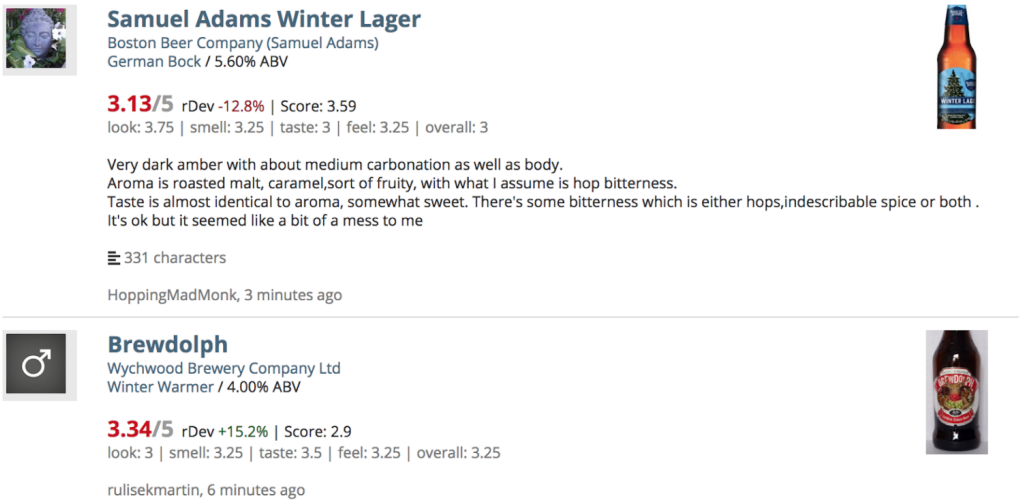

Fig. 1. Ejemplo de Reviews en BeerAdvocate.

2. ANÁLISIS EXPLORATORIO DE LOS DATOS

Antes de entrenar el modelo, se hizo necesario estudiar la composición de *data set* que se utilizaría para los experimentos. Para cumplir con este cometido, se usó la librería `pandas` de Python, con la que se obtuvo información básica de los datos, como cantidad de reseñas, cervezas y usuarios. Además, se generaron gráficos que hicieron más fácil la comprensión de la composición del *data set*. La información y los gráficos obtenidos se muestran a continuación.

Cantidad de cervezas valoradas	*Sparcity* del *data set*	Largo promedio de las reseñas	Filas con valores `NaN`
508358	99.85 %	693.51 caracteres	20512

Cuadro 1. Información básica del *data set* de BeerAdvocate escogido.

Además, se estudió la cantidad de usuarios por número de reviews realizadas. El siguiente gráfico expone como distribuye dicha relación:

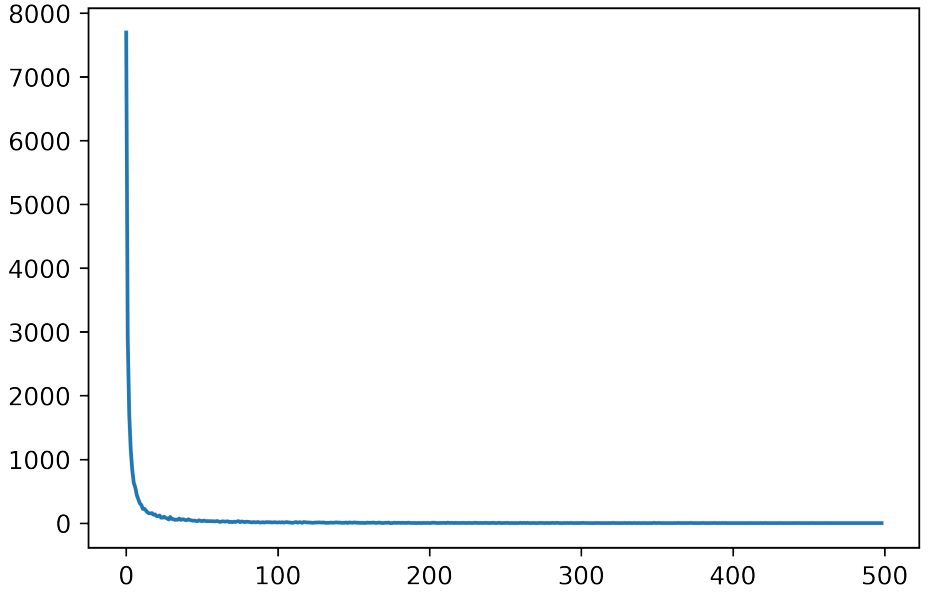

Fig. 2. Cantidad de usuarios que realizaron cierta cantidad de *reviews*.

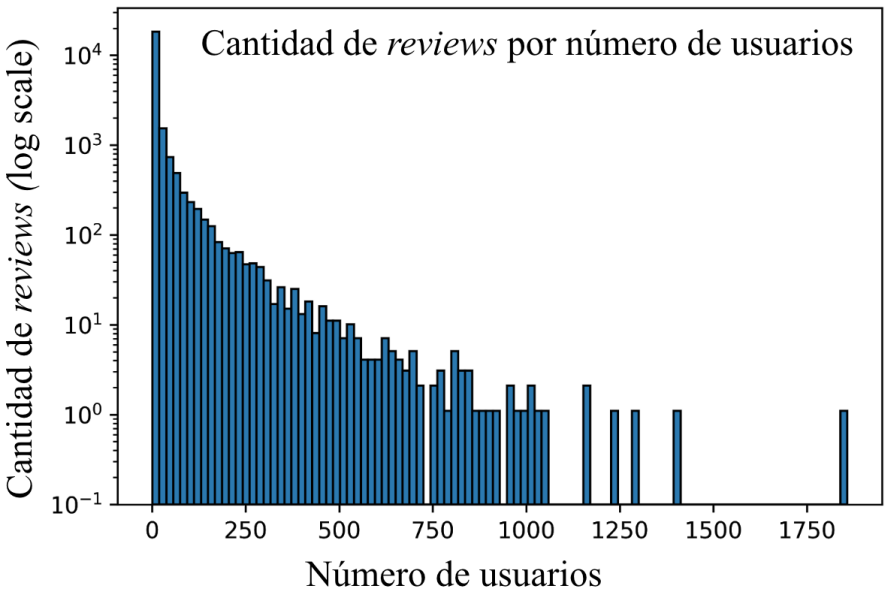

Fig. 3. Cantidad de usuarios que realizaron cierta cantidad de *reviews* escala logarítmica.

Es posible observar que una gran cantidad de usuarios solo ha realizado una *review*, pero si se analiza con más detalle la tabla que generó dicho gráfico y se considera que hay 22801 usuarios distintos, es posible concluir que al menos 40 % de los usuarios realiza 5 o más reviews.

Toda la información presentada anteriormente permitió determinar el punto en el que se dividirían los datos en conjuntos de entrenamiento y de prueba. Como se deseaba recrear un escenario de *cold start*, se decidió realizar los datos de forma temporal, es decir, se tomaría el 70 % de los datos más antiguos para conformar el conjunto de entrenamiento y el 30 % restante correspondería a datos de prueba. En particular, la fecha de corte entre entrenamiento y prueba correspondió al 10 de septiembre de 2004. Así, todos los datos pertenecientes a esa fecha o antes, corresponden a datos para entrenar el modelo. Los restantes se utilizan para probar el desempeño de las máquinas de factorización.

3. DEFINICIÓN DE EXPERIMENTOS

3.1. Calibración de híper-parámetros

Una vez que se completó la exploración de los datos, se procedió a calibrar los híper-parámetros del modelo. El objetivo de esta acción fue determinar los mejores híper-parámetros para el modelo, de forma de disminuir el valor de RMSE. Así, podría asegurarse una mejor calidad de las recomendaciones sobre el conjunto de prueba. Para esta tarea, se dividió el *data set* de entrenamiento en 5 *fold* de forma aleatoria y se iteró por diferentes combinaciones de parámetros con *cross-validation* y así llegar a los mejores parámetros.

3.2. Entrenamiento del modelo con híper-parámetros finales

Luego de obtener los híper-parámetros que entregan los mejores resultados, se optó por entrenar 3 modelos distintos con todo el *data set* de entrenamiento para luego recomendar diferentes cantidad de ítems para cada usuario y validar esa recomendación con el *data set* de *testing*. Para identificar los ítems relevantes, se decidió interpretar todos aquellos ítems cuya *rating* sea mayor al promedio de *rating*, que es 3.8, como relevante. Por lo tanto, si un usuario puso de *rating* mayor a 3.8, entonces ese ítem será considerado como relevante. Dada esta interpretación, se entrenó cada modelo:

- Bayesian Personalized Ranking (BPR): en este modelo, se armó de forma manual una matriz *sparse* que indicara si el usuario consumió un ítem o no, donde consumir hace referencia a que realizó una review de este y es un ítem relevante.
- Maquinas de factorización (MF): de forma similar al método anterior, se construyó una matriz *sparse* de los datos donde cada fila era un usuario y cada columna era un ítem distinto.
- Maquinas de factorización con metadata (MF + metadata): en este modelo se agregó nuevas *features* en forma de `OneHotEncoder` para que el modelo aprenda nuevas asociaciones. Las *features* ingresadas fueron el grado alcohólico de la cerveza (ABV), el estilo de la cerveza y quien la confeccionó. Como se puede notar, algunas *features* son numéricas y otras son categóricas.

Los modelos de *baseline* corresponden a BPR, Random y Most Popular.

3.3. Prueba del modelo entrenado con y sin *cold start*

Para la prueba con los modelos, se tomó el *rating* promedio de las valoraciones y se decidió que todo ítem con un *rating* mayor al promedio será relevante (1) y toda bajo el promedio no será relevante (0). Luego de identificar cada ítem relevante y no relevante de los usuarios, se crearon 2 escenarios distintos. El primero consiste en utilizar toda la información del conjunto de pruebas y para cada usuario, recomendar 1, 5, 10, 20, 50 y 100 ítems. En este escenario, existirán ítems nuevos

que el modelo no conoce y usuarios nuevos que el modelo nunca vio en el entrenamiento. De los usuarios en el conjunto *testing*, un 60 % son nuevos mientras que en los ítems hay un 25 % nuevo. El otro escenario es uno sin *cold start*, es decir, se retira todo ítem y usuario nuevo. Esto es para evaluar la eficiencia del modelo cuando solo recomienda a usuarios con los que ya fue entrenado y solo se compara con ítems que conoce. Para ambos escenarios, se utilizaron los mismos parámetros para las máquinas de factorización y BPR, de tal modo de poder realizar una comparación más exhaustiva entre ambos escenarios.

4. EXPOSICIÓN DE RESULTADOS

Coldstart	Random		BPR		Most Popular		MF		MF + Metadata	
	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall
1 item	0.001	0.000	0.068	0.010	0.000	0.000	0.001	0.000	0.000	0.000
5 items	0.004	0.000	0.174	0.035	0.000	0.000	0.002	0.000	0.001	0.000
10 items	0.007	0.001	0.249	0.060	0.032	0.032	0.004	0.000	0.003	0.000
20 items	0.035	0.003	0.337	0.099	0.395	0.130	0.032	0.002	0.027	0.002
50 items	0.059	0.006	0.451	0.172	0.512	0.213	0.075	0.009	0.037	0.003
100 items	0.084	0.009	0.532	0.238	0.606	0.306	0.120	0.018	0.052	0.004

Sin Coldstart	Random		BPR		Most Popular		MF		MF + Metadata	
	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall
1 item	0.002	0.000	0.109	0.009	0.000	0.000	0.000	0.000	0.000	0.000
5 items	0.007	0.000	0.290	0.041	0.000	0.000	0.002	0.000	0.000	0.000
10 items	0.017	0.001	0.386	0.069	0.018	0.018	0.006	0.000	0.002	0.000
20 items	0.052	0.003	0.491	0.118	0.435	0.104	0.054	0.004	0.038	0.003
50 items	0.087	0.005	0.603	0.200	0.554	0.174	0.101	0.010	0.054	0.004
100 items	0.121	0.010	0.673	0.285	0.650	0.265	0.168	0.020	0.090	0.008

Fig. 4. Resultados de recomendación de 1, 5, 10, 20, 50 y 100 ítems con 5 modelos.

5. DISCUSIÓN

Debido a que se intentó recomendar con 5 técnicas diferentes (Most Popular, BPR, Random y máquinas de factorización con y sin Metadata), es posible rescatar diferentes observaciones de estos resultados. Para empezar, es posible notar que los modelos que mejor rendimiento tienen en la tarea de recomendar ítems relevantes y ocupar estos los primeros lugares en las listas de recomendación, son Most Popular y BPR, mientras que el peor modelo en recomendar es Random. Esto permite observar que el modelo de máquinas de factorización logra aprender alguna noción de los usuarios, pero es mínima en comparación a lo aprendido por BPR o a utilizar el tradicional ítem más popular para recomendar.

Otro hecho a destacar con estos resultados, es la disminución de rendimiento de Most Popular cuando se pasa de un escenario con *cold start* a uno sin este, es decir, a un escenario sin ítems nuevos o usuarios nunca antes vistos. Si se toma en cuenta que un 60 % de los usuarios del set de *testing* son nuevos y estos ya no están en el segundo escenario, es posible concluir que existe una tendencia en el dominio de las cervezas que consiste en consumir las más populares primero. Esta observación permite explicar la disminución de rendimiento, porque no hubo instancias de recomendar a nuevos usuarios los ítems que por tendencia iban a consumir y provocarían un aumento en el rendimiento de la recomendación con Most Popular.

6. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se presentó de forma detallada el desarrollo del proyecto semestral que buscó utilizar máquinas de factorización para generar recomendaciones en base a distintos tipos de datos de *input*. Lamentablemente, los resultados no fueron los esperados. De hecho, fueron tan malos, que, como se pudo constatar en las secciones anteriores, modelos no personalizados como Random y Most Popular vencieron en varias ocasiones al entrenado en el marco de este trabajo.

Entre las lecciones que quedan para los autores, se cuenta el hecho de que se perdió, por un breve momento, el foco de esta investigación. En las etapas iniciales, se destinó mucho tiempo y recursos a realizar *feature engineering*, cuando tal vez hubiera sido una mejor idea probar inmediatamente los modelos de *baseline* y saber de antemano los valores de las métricas que se debía superar. Esta forma de abordar el trabajo podría haber permitido un enfoque mayor en las mejoras del modelo propuesto. Esto tiene como consecuencia una mejora en las recomendaciones.

Por otro lado, se puede rescatar de esta experiencia que los trabajos del área de sistemas recomendadores requieren un trabajo sistemático y constante para obtener resultados. En el caso del desarrollo de este proyecto, el tiempo fue un gran limitante. El haber abordado este proyecto junto a otros cursos (también con proyectos) y considerar el tiempo requerido para terminar de ejecutar cada experimento, contribuyó en menor o mayor medida a que los resultados no fueran los esperados. En adelante, los autores de este trabajo esperan ser más capaces de equilibrar sus tiempos para que los resultados de sus investigaciones y experimentos asociados sean de gran calidad.

Finalmente, como trabajo futuro se espera poder agregar las *features* de las reseñas en forma de texto, que no alcanzaron a ser procesadas para incluirlas en el modelo presentado en este trabajo. Se espera, con estos atributos, poder mejorar el desempeño del modelo en su forma actual. Además, parece relevante estudiar por qué los métodos no personalizados (como Most Popular) alcanzaron un rendimiento superior a un modelo más robusto, ¿podrá ser que los datos utilizados tienen algún sesgo que no fue detectado en esta investigación? O bien, ¿hay algo en el dominio de la recomendación de bebidas alcohólicas (o de productos ingeribles en general, tal vez) que fomente el desempeño de dichos modelos?

REFERENCIAS

[1] L. Orellana Altam. [n. d.]. Análisis de Dataset Beer Reviews. http://www.academia.edu/36415850/Análisis_de_Dataset_Beer_Reviews