Informe Final
Implicit to explicit regression mapping of user preferences for personalized recomendations.

Ignacio Becker Troncoso

November 28, 2016

Abstract

Métodos de recomendación basados en feedback implícito han aparecido hace relativamente poco tiempo, pero han tenido un progreso explosivo. Existen métodos de factorización matricial que sólo incorporan la senñal implícita sin responder la pregunta de la razón por la que funcionan. En este trabajo se incorpora el feedback explícito y la información temporal en una regresión logística. Empleando la métrica Mean Percentile Ranking el método implementado en este trabajo obtiene 45.76% en comparación con 41.31% de métodos tradicionales. Se explica por la sobresimplificación de la regresión logística en el bineo de datos. Se proponen modelos que incluyan la información temporal directamente en la función de costo.

1 Introducción

Actualmente, la cantidad de sitios de e-commerce cubre prácticamente todo tipo de productos, desde objetos físicos a cosas intangibles, como música o juegos. La cantidad de elementos ofrecidos ha explotado también, por lo que algoritmos de recomendación son indispensables para poder hacer de la experiencia del usuario cómoda y expedita.

En un inicio los algoritmos desarrollados se basaban en calificaciones de los usuarios sobre algún ítem consumido, pero en la actualidad ese método no es aplicable debido a varias razones. Una de las más importantes es que

el tiempo del usuario es limitado, por lo que pedirle que evalúe items no es viable, menos cuando esta cantidad puede llegar a ser muy grande como música, donde la experiencia es completamente distinta a la de una película o la de un libro, donde el usuario tendría que evaluar decenas o incluso cientos de elementos en un día.

Por otro lado, el feedback implícito es aquel que no requiere una evaluación explícita del usuario y se puede recolectar automáticamente. Este tipo de feedback puede tomar diversas formas, generalmente como reproducciones de una pieza de música, cantidad de clicks en una página web, tiempo que pasa el usuario en cierta página, entre otras. Este tipo de información es mucho más abundante pero a la vez mucho más confusa ya que no hay una conexión explícita entre feedback explícito e implícito.

Recomendadores de música han aumentado su popularidad con los servicios streamming de Spotify o GooglePlay Music, sin dejar de lado servicios como LastFM o MusicBrainz. Los recomendadores de música en si mismos son un área de investigación, ya que pueden orientarse a recomendar artistas, álbums, listas de reproducción o tracks independientes.

1.1 Estado actual

Este trabajo se apoya mayoritariamente en el trabajo de [Hu et al., 2008], en el cual utilizan el formalismo de factorización matricial para predecir listas de recomendación rankeadas. Lo innovador de su trabajo fue incluír pesos para cada observación basadas en la cantidad de reproducciones de cierto ítem. Trabajos más recientes como el de [Johnson, 2014] han expandido la idea anterior, esta vez haciendo una regresión logística sobre los datos, para obtener de igual forma factores latentes y bias tanto para usuarios como para ítems. En este trabajo se habla de escalar el algoritmo para ser implementado en el framework MapReduce. Lamentablemente la implementación de este algoritmo no considera el uso de matrices sparse por lo que no es posible usarlo directamente.

También [Rendle et al., 2009] plantea el problema desde el formalismo bayesiano, maximizando la distribución posterior para cada usuario, cuyos parámetros son los factores latentes. Muestran que su método de optimización supera a los métodos de factorización matricial y de vecinos cercanos.

Si bien estos trabajos apuntan a obtener mejores recomendaciones, la conexión entre feedback implícito y explícito no está clara ni ha sido bien

estudiada. El trabajo de Parra el at.(en preparación) muestra que la inclusión de información temporal puede explicar mejor las observaciones, basadas en encuestas hechas a usuarios de LastFM en donde evaluaron en escala de 1 a 5 distintos albums obtenidos de sus listas de reproducción. El estudio consistió en 114 usuarios y 6037 albums en total, en donde se hizo una regresión para reproducir la calificación en escala de 1 a 5. El problema principal es el poco overlap en los albums de cada usuario, lo que limita la efectividad de distintos algoritmos y prohibe aplicar métodos colaborativos.

2 Solución

En este trabajo se propone usar un método de factorización matricial para feedback implícito, enriquecido con información sobre lo reciente del consumo de cada álbum. Específicamente se remplazará la matriz de confianza C por la probabilidad de "like" obtenida de la regresión logística hecha sobre los datos.

Este es un paso intermedio, ya que el objetivo final es incluir la información temporal directamente en la función de pérdida ya que es la manera más directa de incluir información adicional.

3 Dataset

En este trabajo se usó el dataset $30Music$ [Turrin et al.,] publicado recientemente. Contiene información de $45K$ usuarios, $217K$ albums, $600K$ artistas, $5.6M$ tracks, y $276K$ tags obtenidos de usuarios de LastFM. Además contiene los historiales de reproducción de los usuarios en un período de un año, a contar del 20 de Enero del 2014, sumando en total $31M$ de eventos de reproducción. Adicionalmente se incluyen $4.1M$ de evaluaciones explícitas en forma de "love". Dicha evaluación se hace directamente en el sitio de LastFM, lo que significa un esfuerzo adicional por parte del usuario. Esto indica un alto nivel de preferencia, por lo que la seguridad en este valor es muy alta y correspondería a una evaluación de 5 estrellas en un contexto de feedback explícito.

La información relacionada a los tracks incluye la duración, el album relacionado si es que pertenece a alguno, la cantidad total de reproducciones,

y los tags asociados a dicho track.

La información de cada reproducción incluye el ID del usuario y el ID del track, el timestamp y el tiempo escuchado. Es importante notar que todos los elementos en este dataset fueron escuchado a lo menos un 50% de la duración o un mínimo de 4 minutos.

La información de los usuarios incluye el nombre del usuario, ID del usuario, género, edad, fecha de creación de la cuenta y país.

Para no disminuír la cantidad de elementos en el set de entrenamiento, se fabricó un set de test que acumula las reproducciones por álbum por usuario a lo largo de un año a partir del 20 de Enero del 2015. Debido a que la API de LastFM no permite recuperar datos con periodos arbitrarios, la recolección de datos tuvo que ser hecha crawleando los datos directamente de la página de cada usuario. Se recuperaron los primeros 500 albums por usuario, guardando el artista, el album y la cantidad de reproducciones. Lamentablemente no todos los álbums de este set están incluidos en el dataset, por lo que el comportamiento del usuario no está del todo representado en el set de test.

El set de test consistió finalmente en 5358 usuarios, que tenían al menos 50 albums en su lista (ver figura 1).. Se realizó este filtro ya que para listas muy pequeñas, los ítems se parecen mucho y en general tienen pocas reproducciones, lo que hace que no sea un buen indicador.

3.1 Preprocesamiento

Como parte del procesamiento, se eliminaron los tracks que no pertenecían a ningún álbum alcanzando un total de 1469K tracks. Posteriormente se condensaron las reproducciones por track en reproducciones por álbum para después eliminar los álbums con menos de 200 reproducciones totales como se aprecia en la figura 2, obteniendo un total de 13121 álbums. Esto último fue hecho para reducir el ruido producido por elementos muy poco frecuentes y desconocidos que probablemente no fueron escuchados por muchos usuarios.

Del total de usuarios, los que habían hecho algún "like" a algunos de los tracks con álbum fueron 36583. Luego se eliminaron los usuarios con menos de 20 álbums en total, quedando un total de 29027 para el set final.

La sparcity de la matriz final es de 0.76% ya que nos aseguramos de incluir álbums relativamente populares, descartando la mayor cantidad de elementos que pertenecían a la larga cola exponencial.

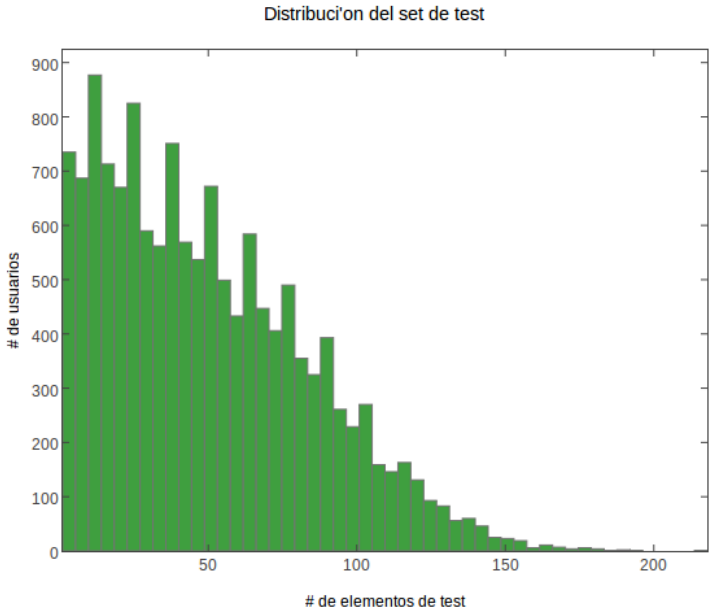

Figure 1: SE observa que la mayoría de los usuarios tienen menos de 50 elementos.

4 Metodología

4.1 Modelos

Siguiendo la línea de Parra et al., se hizo la separación en bines temporales y bines de feedback implícito.

Para los bines temporales, se obtuvo el timestamp de cada evento dentro del año que abarcaba el set de datos y se impusieron tres intervalos, asignando el valor de 3 para álbums que se escucharon durante el último mes, 2 para los escuchados durante el tercer y segundo mes, y finalmente 1 el resto, escuchados del tercer mes hacia atrás.

Para el feedback implícito, se evaluó la cantidad total de reproducciones por usuario, y se asignó un valor de 3 a los álbums escuchados que componen el primer tercio de la suma acumulada, 2 para el siguiente tercio y 1 los álbums del final de la lista.

Por cada elemento de del dataset se aplicaron los modelos:

$$\textbf{Model 1: } l_{iu} \quad = \quad \beta_0 + \beta_1 \cdot if_{iu} + \beta_2 \cdot r_{iu} \tag{1}$$

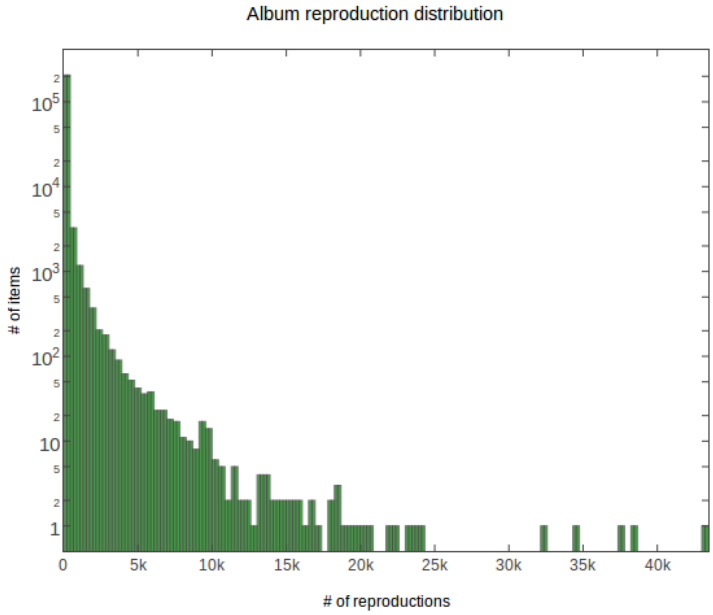

Figure 2: Distribución de la cantidad de reproducciones. Notar que la cantidad de elementos con menos de 200 reproducciones es $\sim 10^5$.

$$\textbf{Model 2: } l_{iu} \quad = \quad \beta_0 + \beta_1 \cdot if_{iu} + \beta_2 \cdot r_{iu} + \beta_3 \cdot if_{iu} \cdot r_{iu}. \tag{2}$$

En el trabajo de HK08, minimizan la expresión

$$min \sum_{u,i} c_{ui}(p_{ui} - x_u^t y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right), \tag{3}$$

donde p_{ui} es una variable binaria que toma el valor de 1 si el usuario i consumió el ítem, y 0 en caso contrario. c_{ui} es la confianza en la la observación, que toma la forma

$$c_{ui} \quad = \quad 1 + \alpha r_{ui}, \tag{4}$$

$$c_{ui} \quad = \quad 1 + \alpha \log\left(1 + r_{ui}/\epsilon\right). \tag{5}$$

Donde ajustan el valor de $\alpha = 40$. Mientras mayor sea la señal, mayor peso tendrá en la minimización por lo que dará más peso a dicha observación.

Según el trabajo en [Frederickson, 2016], la matriz de confianza puede ser obteinda usando los pesos BM25 que toman la forma

$$c_{ui} = \frac{r_{ui} \cdot (k1 + 1)}{k_1 * b1 + r_{ui}} \cdot IDF, \tag{6}$$

donde r_{ui} es el feedback implícito, $k_1 = 100$, $b1 = (1 - B) + B\frac{S_i}{D}$, donde $B = 0.8$, S_i es la cantidad total de reproducciones para el album i y D es la media de la cantidad de reproducciones del dataset, por album.

En este trabajo se usó la implementación de Ben Frederickson del algoritmo en la librería en python *Implicit*, que hace uso de matrices sparse e implementa alternating least squares en Cython, lo que lo hace ordenes de magnitud más rápido que implementaciones nativas en python.

4.2 Métrica de valuación

Debido a que en el marco de feedback explícito, las métricas orientadas a precisión no pueden ser aplicadas por la aucencia de ratings, métricas de ranking son preferidas ya que entregan la lista de ítems preferidos por el usuario y el algoritmo sólo tiene que ordenarlas.

Para este trabajo se elijió Mean Percentile Ranking (MPR) [Parra and Sahebi, 2013] ya que está ideada para evaluar rankings basados en feedback implícito, está dado por

$$\overline{rank} = \frac{\sum_{u,i} r_{ui}^t \cdot rank_{ui}}{\sum_{u,i} r_{ui}}, \tag{7}$$

donde $rank_{ui}$ es el ranking basado en el percentil en el que se ubica el ítem en una lista y r_{ui} es la señal implícita. Valores cercanos a 0% son deseados, lo que indica que los elementos más consumidos fueron ubicados en los primeros lugares de la lista.

5 Análisis de parámetros

A continuación se presentan los métodos utilizados para definir los parámetros para cada uno de los algoritmos utilizados.

5.1 Hu & Koren 2008

Este método se usó como baseline, teniendo como parámetros el valor de λ y la cantidad de factores latentes. Para obtenerlos se hizo validación cruzada de 5 folds para cada usuario. De esta forma todos los elementos fueron evaluados con la métrica MPR. Aún con la librería optimizada y utilizando 8 threads, el tiempo para realizarla fue alrededor de un día y medio para una grilla de 5×5, por lo que experimentos en detalle requieren mucho más

tiempo. Se presentan los resultados en la figura 3. Los valores elegidos fueron

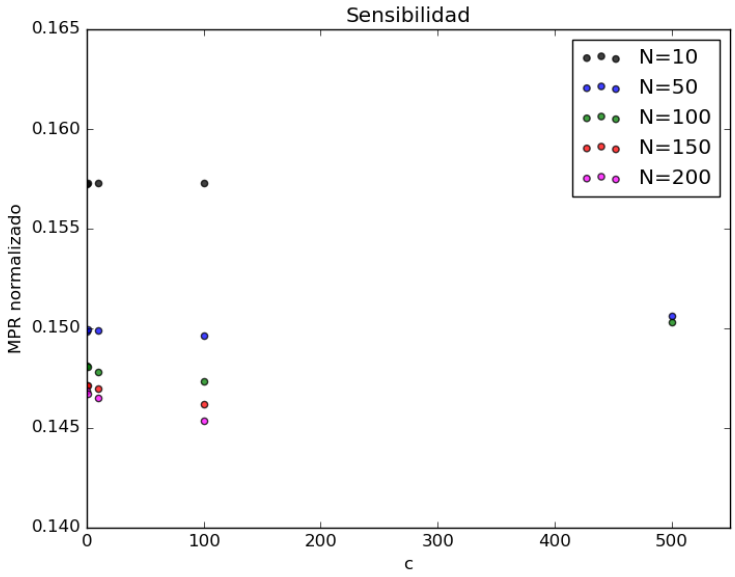

Figure 3: Sensibilidad de los parámetros al factor de regularización y cantidad de factores latentes. Notar que para $c = 500$ el error aumentaba, por lo que se detuve la búsqueda para $N = 150$.

$\lambda = 100$ y el número de parámetros wn 200. Más podrían haber significado una mejora en la precisión, pero el incremento en tiempo de cálculo hacía el método muy difícil de testear con el hardware disponible.

5.2 Regresión logística

Para la regresión logística se usó la implementación en python de scikit-learn. Se usó el método de descenso de gradiente estocástico y para los modelos de 1 y 2. Se probó con distintos valores de la constante de normalización, desde $\log c = -4$ hasta $\log c = 5$ sin lograr un impacto en el ajuste. Se ponderó los datos de acuerdo al inverso de la frecuencia, ya que de otra forma el modelo se ajustaba a los elementos sin likes. Los parámetros de ajuste se entregan a continuación:

	β_0	β_1	β_2	β_3
Model 1	-0.7531	0.5016	-0.0080	———
Model 2	-0.7241	0.4823	-0.0199	0.0068

Notar que el valor de β_0 y β_1 se mantiene casi constante, pero el valor de β_2 varía ante la influencia del término de interacción entre el implicit feedback y recentness β_3.

Este ajuste tiende a tener recall 1 ya que sobre estima la cantidad de elementos que poseen like explícito. Ahora bien, este modelo por sí sólo puede predecir un ranking. Como el objetivo de estre trabajo es realizar un mapeo, estos datos se usarán como la matriz C en el modelo de HK08.

5.3 Nuestro modelo

Dado que los datos son del mismo orden para nuestro método y para el de HK08, se usó la misma constante de regularización y el número de parámetros. Dado que el ajuste mejora con la inclusión de más parámetros, compararlos con los mismos parámetros entrega luces del rendimiento de ambos métodos a falta de validación cruzada.

6 Resultados

A pesar de que resultados previos indicaban que los ajustes mejoraban al incluir la interacción entre implicit feedback y recentness, los resultados obtenidos usando nuestro modelo no logra superar lo obtenido por HK08.

Para nuestro modelo, el valor de MPR es 45.76% mientras que para HK08 se obtuvo 41.31%. Esta diferencia es considerable, dado que el valor ideal se ubica alrededor de 25%. Esta aproximación no es del todo adecuada ya que el modelo está sobresimplificado al incluir dos clases, a diferencia del trabajo previo en donde existían cinco niveles de preferencia. Otra causa puede ser que el modelo baseline no penaliza el feedback imlpícito de la forma en la que lo hace nuestro modelo y como se puede ver en los coeficientes de la regresión, es la variable más importante, por lo que binarizarla quizás no es lo más adecuado en este caso.

Por otro lado, el objetivo final es integrar la información de recentness e interacciones en la función de costo directamente ya sea mediante factorization machines o por un término independiente.

7 Conclusiones

Este trabajo mostró que utilizar el output de una regresión logística en implicit feedback y recentness no compite contra el método original de HK08. Como se puede apreciar de los coeficientes de la regresión logística, el feedback implícito es la variable más importante, por lo que una de las posibles causas es la simplificación del implicit feedback en bines con valores de 1 a 3, a difrencia del baseline que hace una ponderación contínua, de esta forma mantiene la mayoría de la información.

No sólo la cantidad de reproducciones puede ser tomado como señal implícita, sino que tambien puede ser útil usar el tiempo de reproducción. Si bien no es de la misma área, trabajos en recomendación de lugares de interés [Lim et al., 2015] han demostrado tener éxito usando tiempo en vez de frecuencia. También se propone incluir parámetros adicionales en la función de costo, ya sea mediante factorization machines o mediante una función de costo definida previamente con la dependencia temporal explícita.

La métrica de evaluación MPR está disenñada para ser usada con feedback implícito pero funciona mejor cuando la mayoría de las interacciones se concentran en pocos ítems y el tamaño de la lista es cercano a 20 elementos. Eso impone restricciones al tipos de datos que se pueden usar, ya que requiere un usuario muy activo, con cientos de reproducciones y que posea preferencias marcadas donde la mayoría de las reproducciones se concentren en los primeros elementos de la lista. Eso no se cumplió en su totalidad en este trabajo, por lo que los resultados no fueron los mejores. Se propone usar una versión de MPR que tome en cuenta la heterogeneidad de los distintos usuarios, restando al MPR el valor del ideal.

References

[Frederickson, 2016] Frederickson, B. (2016). Finding similar music using matrix factorization.

[Hu et al., 2008] Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. Ieee.

[Johnson, 2014] Johnson, C. C. (2014). Logistic matrix factorization for implicit feedback data. In *NIPS 2014 Workshop on Distributed Machine Learning and Matrix Computations*.

[Lim et al., 2015] Lim, K. H., Chan, J., Leckie, C., and Karunasekera, S. (2015). Personalized tour recommendation based on user interests and points of interest visit durations. *Under Submission*.

[Parra and Sahebi, 2013] Parra, D. and Sahebi, S. (2013). Recommender systems: Sources of knowledge and evaluation metrics. In et al. (Eds.), J. V., editor, *Advanced Techniques in Web Intelligence-2: Web User Browsing Behaviour and Preference Analysis*, chapter 7, pages 149—-175. Springer-Verlag, Berlin Heidelberg.

[Rendle et al., 2009] Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. (2009). Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 452–461. AUAI Press.

[Turrin et al.,] Turrin, R., Quadrana, M., Condorelli, A., Pagano, R., and Cremonesi, P. 30music listening and playlists dataset.