

Recomendación Multimodal Adversaria

Mario Carlos Mallea Ruz

Universidad Federico Santa María, Chile

mario.mallea@sansano.usm.cl

Abstract

Este paper proporciona una extensión del modelo IRGAN a problemas multimodales, integrando atributos visuales y de texto a IRGAN, el cual es un método de muestreo negativo propuesto para filtrado colaborativo. A través de una competencia donde se ven enfrentados dos modelos de factorización matricial en un juego tipo min-max, se busca conseguir la mejor factorización matricial que considere feedback implícito por parte de los usuarios, integrando las características visuales de los items y el sentimiento de las opiniones de los usuarios sobre los items comprados.

La mejor versión del tipo IRGAN desarrollada en este trabajo logra una mejora con respecto a modelos como VBPR, donde incorporando una representación latente de los gustos visuales del usuario y un sesgo de subjetividad vinculada a sus opiniones, se logra una mejora de un 4.5% en la AUC por sobre el mejor de los métodos clásicos.

Todas las implementaciones necesarias para reproducir el trabajos se encuentra en el siguiente [repositorio](#)

Keywords— IRGAN, Recomendación Multimodal, E-commers, Factorización Matricial, Muestreo Negativo. .

1 Introducción

Los métodos de factorización matricial típicos en filtrado colaborativo ofrecen una manera transparente y económica para realizar recomendaciones, existiendo métodos capaces de considerar tanto el feedback implícito por parte del usuario, como métodos más complejos que logran considerar atributos visuales en dicha factorización. Motivados por IRGAN [12], un método basado en generar una factorización matricial, pero que a través de una metodología de aprendizaje adversario, construye un muestreo negativo dinámico para el entrenamiento, ofreciendo un mejor desempeño que otros métodos clásicos de filtrado colaborativo. En este trabajo se explora la adaptación de IRGAN a un problema de recomendación multimodal, es decir, un problema donde no sólo se considere el feedback implícito por parte del usuario, sino que también sea posible considerar atributos visuales de los items, y las opiniones por parte de los usuarios.

En este trabajo, se proponen 4 extensiones de IRGAN, que consideran, adaptaciones del entrenamiento y la incorporación de más contexto y contenido a la recomendación.

2 Trabajo relacionado

El presente trabajo se relaciona con varias áreas específicas a la vez:

2.1 Modelamiento

El trabajo [5] fundó las bases de la metodología de aprendizaje adversario con el modelo GAN. Posteriormente [13] adoptó la metodología para el de generación de secuencias de tokens discretos con el modelo SeqGAN, la principal dificultad que presentaba la adaptación era que en el modelo GAN, el gradiente de la loss del discriminador (D) con respecto al output del generador (G) es usado para guiar a los parámetros de (G) para producir un cambio leve del valor generado para hacerlo más real, pero para el caso de un espacio discreto es necesario reformular la forma en la que se consigue este leve cambio. Inspirados en la idea de modelar el problema de generación de secuencias como uno de toma de decisiones, (G) es considerado como un agente de aprendizaje reforzado y se interpreta una aproximación de monte carlo para el gradiente, como una política de gradiente estocástico. La idea fue probada con éxito, mejorando significativamente algunos baselines con respecto a ciertas métricas incluyendo el juicio humano experto, esto en tareas tales como generación de poemas, música y lenguaje hablado. Luego de ello IRGAN [12] adapta las ideas de SeqGAN para problemas de information retrieval, incluyendo recomendación basada en filtrado colaborativo, considerando (D) y (G) como factorizaciones matriciales simples.

Otros modelos de recomendación basada en aprendizaje adversario han surgido tras IRGAN. Siguiendo la taxonomía de explotar la capacidad generativa en un aprendizaje adversario para recomendación, el siguiente survey [4], destaca por ejemplo a GraphGAN [11], un modelo que puede ser visto como una representación basada en grafos de IRGAN, con el objetivo de que el generador aprenda el ground truth de conectividad como modelo recomendador, mientras que el discriminador, decide si aquella conexión podría existir o no. También, por ejemplo CFGAN [1] re-adapta el problema para que el modelo generador aprenda vectores continuos, mostrando una mejora con respecto a IRGAN y GraphGAN. También existen otras propuestas que reemplazan la factorización matricial por auto encoders, auto encoders variacionales, redes convoluciones, entre otros.

2.2 Problema

Las plataformas de e-commerce son ecosistemas de constante flujo de compras que logran beneficiarse de buenos sistemas recomendadores gracias al feedback del consumo recolectado. Un ejemplo concreto es Amazon, en ese sentido [9] introduce varios conjuntos de datos de esta plataforma, que constan de un historial de compra para ciertos usuarios, incluyendo feedback como rating y opiniones que entrega cada usuario al ítem consumido, y además se tiene la imagen del artículo comprado.

Un modelo que establece un buen baseline en los problemas de recomendación que incluyen características visuales es VBPR [6], esta consiste en una factorización matricial que construye la representación latente visual del usuario en base a la similitud con la representación visual del ítem (pre entrenada), según una metodología de aprendizaje que busca capturar la relación de orden entre items. Utilizando el mismo framework DVBP [7] establece el estado del arte para este problema multimodal con respecto a la métrica AUC, pero considerando redes convolucionales siamesas para el aprendizaje directo de las características visuales de los artículos en venta, explicando que es factible desechar la integración de filtrado colaborativo y el uso de opiniones.

Con respecto a la integración de la review a través de un análisis de sentimiento de las mismas, en un sistema recomendador. Existen

muchos trabajos que utilizan diversas técnicas, el trabajo más cercano encontrado fue [3], que combina la factorización matricial y un modelo de sentimientos LSTM a través de una combinación convexa, y buscando en validación el parámetro de combinación óptimo. Si bien no se encontró una investigación con la propuesta de este trabajo para el sentimiento, esta es en extremo simple por lo que no se considera una novedad, esto consiste en utilizar la función de polaridad de la librería *TextBlob* [8] (modelo pre entrenado) para incluir un sesgo de emocionalidad del usuario a la factorización matricial.

2.3 Muestreo negativo

La selección de items como ejemplos negativos tiene una importante influencia en el entrenamiento de sistemas recomendadores que se basan en tripletas de entrenamiento (usuario, item positivo, item negativo). Considerar que si se tomase simplemente el complemento de los items consumidos o relevantes como ejemplo negativos, entonces se construiría un problema muy desbalanceado, ya que generalmente los items disponibles son muchos más que las interacciones recolectadas del consumo. Notar adicionalmente, que tal conjunto de entrenamiento contendría muchos falsos negativos (puesto que no tener feedback de items no quiere decir que estos no le puedan gustar o ser relevantes para el usuario) y triviales verdaderos negativos, cuando los items seleccionados para ser ejemplos negativos son poco similares a los recolectados, lo cual es bastante probable si su selección fuese de manera aleatoria, como lo hacen la mayoría de trabajos clásicos como BPR [10] y VBPR [6]. Esto es de vital importancia para entender la contribución de IRGAN.

Diversas heurísticas para la determinación del muestreo negativo han sido exploradas y debido a su sencillez y bajo costo computacional son las técnicas más ocupadas. Algunos ejemplos son el muestreo negativo random [10] y el basado en popularidad [2]. Otra línea de técnicas se enmarca en muestreos negativos basados en modelos, por ejemplo dynamically negative sampling [14] se basa en la idea de cambiar dinámicamente la distribución de muestreo de acuerdo a la situación actual del modelo con el fin de mejorar la calidad del muestreo en cada ronda de entrenamiento. Notar que si en cada ronda de aprendizaje, se seleccionan a través del modelo anterior los ejemplos negativos como aquellos items que fueron más valorados, entonces estos ejemplos aportan mayor información para el modelo actual, y así iterativamente.

IRGAN plantea una manera adversaria de muestrear negativamente, su objetivo es seleccionar como ejemplos negativos aquellos items que son más capaces de aumentar el valor de una pérdida adversaria. Específicamente existe una competencia entre dos modelos recomendadores, uno generativo y otro discriminativo. Mientras el recomendador generativo intenta seleccionar los items que más le pueden gustar al usuario para intentar confundir al discriminador, que al discernir entre items consumidos versus generados, fortalece su conocimiento acerca de los gustos de los usuarios, y al mismo tiempo recompensa las selecciones del generador a través de una política de gradiente estocástica.

3 Modelo

Siguiendo el framework desarrollado por IRGAN [12] para recomendación adversaria basada en filtrado colaborativo, en este trabajo se extendió dicho modelo para incluir imágenes de los ítems y las opiniones de los usuarios sobre los mismos.

IRGAN, plantea una manera adversaria de muestrear negativamente. Específicamente existe una competencia entre dos modelos recomendadores, uno generativo $p_\theta = \text{softmax}(g_\theta)$ y otro discriminativo $D_\phi = \sigma(f_\phi)$, que se entrenan con un framework adversario con el objetivo de reducir la loss:

$$\min_{\theta} \max_{\phi} \sum_{n=1}^N (E_{i \sim p_{\text{true}}(i|u_n, r)} [\log D(i | u_n)] + E_{i \sim p_\theta(i|u_n, r)} [\log (1 - D(i | u_n))])$$

Donde u denota un usuario de N disponibles, i al item, r su relevancia dada.

En lo que sigue se explicará el framework de entrenamiento, el cual depende de un modelo base, en este trabajo se considerarán de manera intercambiable tres tipos de factorizaciones matriciales, tipo BPR [10], VBPR [6] y con un sesgo de sentimiento del usuario:

$$\begin{aligned} x_{u,i} &= u_{emb}^T i_{emb} + i_{bias} \\ Vx_{u,i} &= x_{u,i} + \theta_u^T (\mathbf{E}f_i) + \beta^T f_i \\ VTx_{u,i} &= Vx_{u,i} + b_u s_u \end{aligned}$$

Donde u_{emb} y i_{emb} son representaciones latentes de los usuario e items en el mismo espacio y i_{bias} es un sesgo del item. Además, f_i (dimensión 1536) es un vector de características de la imagen del item pre entrenado por una red convolucional (EfficientNetB3), que añade los parámetros θ_u (representación visual del usuario), \mathbf{E} (matriz de reducción de dimensionalidad) y β^T (sesgo visual del item). Por otro lado, s_u es el sentimiento pre calculado promedio de la opinión que entrega el usuario u a sus ítems consumidos, que se utiliza para incluir un sesgo por usuario con el parámetro b_u . En lo que sigue Θ denotará el conjunto de parámetros según corresponda al modelo base utilizado.

Luego el algoritmo inicia como sigue:

1. Inicialización

Se inicializan random los parámetros del modelo según corresponda, fijando la dimensionalidad de los embeddings y λ (la constante de regularización).

Para el caso de la metodología propuesta por IRGAN, se construyen las tripletas (u, i^+, i^-) para todas las interacciones disponibles. Donde i^+ es un item consumido y i^- se escoge como el item más recomendable* de una pequeña muestra aleatoria de los no consumidos de tamaño 5 (para todo efecto). Así, se actualiza el modelo vía SGD para cada tripleta construida, considerando la loss:

$$\begin{aligned} prelogits &= \sigma[\sum x_{u,i^+} - x_{u,i^-}] \\ preloss &= -\sum [\log(prelogits)] + \lambda \|\Theta\|_2 \end{aligned}$$

*: Según la factorización matricial actual: $\sum x_{u,i}$

Notar que cada época de entrenamiento equivale a la cantidad de interacciones en conjunto de train, se considerarán 3. Una variante que se considerará será introduciendo una técnica bootstrap con reemplazo, inspirado en el trabajo BPR [10], es decir entrenado para tripletas random generadas por época, para una comparación justa, serán consideradas el equivalente a tres veces la cantidad de interacciones en el conjunto de train.

Finalmente, se inicializa el generador con los parámetros aprendidos, y se inicializa random el discriminador.

2. Entrenamiento adversario

Entrenamiento del discriminador

Se entrena el discriminador en d_{epoch} , en cada una se construyen las tripletas a través del generador (fijo), y la elección de los ejemplos negativos consiste en un random de entre todos los items pero dándole mayor probabilidad (temperatura de 0.2) a aquellos que son más recomendables según el recomendador generativo g_θ .

Se crea un conjunto de entrenamiento para el discriminador que consiste en desarmar las tripletas en pares etiquetados con 1 para positivo (consumido) y 0 para negativo (generado), por minibatch actualiza parámetros con SGD, para la cross-entropy:

$$\begin{aligned} f_\phi(u, i) &= \sum x_{u,i} \\ preloss &= label[-\log(\sigma(f_\phi(u, i)))] + \\ &\quad (1 - label)[-\log(1 - \sigma(f_\phi(u, i)))] + \lambda \|\Theta\|_2 \end{aligned}$$

Obteniendo así un discriminador, que es una factorización matricial ajustada, tal que logre discernir entre interacciones recolectadas y generadas.

Notar que el entrenamiento, en base a reducir la pérdida se interpreta como sigue:

- Cuando un ejemplo es positivo: La loss es $-\log(prob)$, entonces se quiere que la probabilidad que viene de la sigmoide de los valores de la factorización matricial sea cercana a 1, o equivalentemente, estos provienen de un producto punto que es un coseno que se maximiza para vectores de embeddings (regularizados) muy cercanos en ángulo.
- Cuando un ejemplo es negativo: La loss es $-\log(1 - prob)$, entonces se quiere que la probabilidad sea cercana a cero, equivalentemente, la factorización matricial debe de retornar valores negativos, osea, formar un mayor ángulo para el par de vectores, y que por tanto tienen coseno negativo.

Entrenamiento del generador:

Luego con este discriminador fijo, se entrena el generador para g_{epoch} , el cual por cada usuario es:

$$g_{\theta}(u, i_k) = \sum x_{u,i}$$

Además transforma estos en una probabilidad p_{θ} vía softmax que es utilizada para seleccionar aleatoriamente el doble de items a los consumidos.

$$p_{\theta}(i_k|u, r) = softmax(g_{\theta}(u, i_k))$$

Y con estos items, junto al usuario dado, se obtiene el reward desde el discriminador:

$$reward = 2\sigma(f_{\phi}(u, i_k)) - 1$$

Entonces para cada usuario, su muestra de items y su recompensa, se actualizan los parámetros de g_{θ} vía SGD, con la pérdida derivada del aprendizaje adversario discreto:

$$g_{loss} = - \sum (p_{\theta}(i_k|u, r) \cdot reward) + \lambda ||\Theta||_2$$

Notar entonces que la actualización de parámetros vía reducir la pérdida, equivale a maximizar tanto la probabilidad p_{θ} como $\sigma(f_{\phi})$, es decir, que para el usuario dado se quiere que el generador aprenda a recomendar aquellos items que el discriminador actual prefiere, osea aquellos items que son parecidos a su feedback recolectado y que lo pueden estar confundiendo con mayor probabilidad.

Al terminar los pasos para cada usuario, se guarda el mejor modelo según la AUC de entrenamiento.

Finalmente se vuelve al paso 2.

4 Experimentos

La métrica de evaluación de los modelos será la misma AUC, la cual para cada item a evaluar se calcula versus una muestra random proporcional a 50 veces la cantidad de interacciones del usuario en entrenamiento, esto a partir del trabajo DVBPR [7]. Para efectos del calculo de la AUC se deben evitar los items con cold start, los cuales para ambos problemas no es una cantidad significativa puesto que los conjuntos de entrenamiento y test se construyeron de esa manera a partir de la data original. Por otro lado, de manera transversal a todos los modelos, el set up consistió en un coeficiente de regularización de 0.001, learning rate 0.1, e inicialización de sesgos en ceros y en caso de las representaciones latente y la matriz de reducción de dimensionalidad se inicializan con distribución normal entre $[-0.1, 0.1]$.

Los modelos utilizados son los siguientes:

Por un lado como baseline se considerarán:

Random (R): Un modelo random que asigna de manera aleatoria los ratings latentes a cada usuario.

Most Popular (MP): Un modelo que asigna los rating en escala proporcional al número de interacciones del item en train.

BPR: Una reproducción propia de [10] entrenado en 10.000 épocas con dimensión latente de 10. Seleccionado según el análisis de parámetros de la grilla de VBPR. Esto para reducir el tiempo computacional en los experimentos. Además para la grilla [10,20,35,50,75,100] la diferencia de AUC en entrenamiento alcanzó un máximo para 100 dimensiones latentes que sólo superaba al modelo con menos parámetros sólo por un 0.04 de AUC en ambos data sets.

VBPR: Una reproducción propia de [6] entrenada también en 10.000 épocas con dimensiones 10 y 500 para el embedding de filtrado colaborativo y visual, respectivamente. Estos parámetros fueron seleccionados sobre la grilla [(10,200),(20,200),(10,500),(20,500)]. Ver figura 3 en el anexo.

IRGAN: Siguiendo la implementación del trabajo original, consiste en el modelo explicado anteriormente considerando $x_{u,i}$ como factorización matricial con 10 de dimensión latente, inicializando como se explicó (3 épocas completas) y entrenando según la recomendación de los autores (en menor proporción) con 15 épocas adversarias en las cuales en cada una se entrena 10 veces el discriminador y 5 el generador.

Además se implementó las siguientes modificaciones a IRGAN:

(VI): Como el mismo modelo de IRGAN pero considerando la factorización matricial que incluye las características visuales de los items $Vx_{u,i}$.

(VIb): Como el mismo modelo VI pero con inicialización bootstrap.

(VIb*): Como el mismo modelo VIb pero considerando un entrenamiento de 15 épocas adversarias, en la cual cada una implica 1 época de entrenamiento del discriminador y otra época de entrenamiento del generador. Esta es considerada puesto que intuitivamente el entrenamiento del generador está buscando replicar la distribución de probabilidad latente generada por el discriminador, por lo que un ajuste de la recompensa en un paso se adapta mejor a dicha interpretación.

(VTIb*): Como el modelo que se entrena como VIb* pero considerando las opiniones en la factorización matricial, es decir $VTx_{u,i}$.

4.1 Conjuntos de datos

Los data set utilizados son los de productos del website de Amazon, categorizados por hombre y mujer[9]. Estos, cuentan con los ratings, las imágenes de los items y el texto de la review o comentario proporcionado por el usuario. Las distribuciones del promedio del sentimiento de estos se pueden ver en la figura 4 en el anexo, la cual nos indica que existen diferentes perfiles de usuarios con respecto al grado de polaridad promedio de sus comentarios.

Para enmarcar el problema en un contexto más general, que aplique por ejemplo a feedback implícito, donde generalmente sólo se tienen opiniones positivas de los items por parte de los usuarios, se decide realizar un filtrado de datos según el rating de los productos, dejando sólo aquellos items considerados relevantes, en este caso, productos con ratings mayores que 3.

Para el procesamiento y construcción de los conjuntos de train y test, sólo se consideran una serie de categorías del total disponible en el dataset, las cuales son para el caso de men: *T-Shirts, Shirts, Casual Button-Down Shirts, Polos, Outerwear, Active*. Y para el

caso de women: *T-Shirts, Shirts, Casual Button-Down Shirts, Polos, Outerwear, Tops & Tees, Knits & Tees*. Con esto se consiguen conjuntos de datos más homogéneos en items y se construyen los conjuntos de tal manera que el coldstart sea el menor posible. Notar entonces que se está trabajando con un subconjunto de construcción propio de los datos originales, con el principal propósito de permitir una experimentación más rápida.

Una vez finalizado el procesamiento de datos para ambos dataset se obtienen conjuntos de entrenamiento con las siguientes características.

	Summary Men	Summary Women
User	3834	3158
Item	9106	7245
Total Interactions	12884	11350
Min Interactions	2	2
Max Interactions	38	90
Sparcity	0.037%	0.05%
ColdStart	2.9%	2.7%

5 Resultados

En el anexo se presentan la comparativa del desempeño de todos los modelos presentados en etapa de inferencia o test. Para ambos casos se puede notar como particularmente el método por popularidad es bastante fuerte en AUC, esto debido a la manera de extraer una muestra del dataset completo con el fin de presentar una mayor prolijidad en la experimentación, en particular se le puede atribuir a la selección de categorías específicas. A pesar de ello IRGAN muestra ser un modelo que mejora los métodos clásicos incluso en su versión que no considera las imágenes de los productos, siendo más notorio para el caso de Amazon Women. Además, notar que la versión de IRGAN que incluye las características visuales mejora la versión de filtrado colaborativo para el caso de Amazon Men pero no así en Amazon Women. En el caso de Amazon Men es la introducción de la metodología bootstrap en la inicialización la que mejora permite mejorar de manera más clara el modelo, al igual que como lo reporta el trabajo clásico [10]. Paa el caso de Amazon Women queda en evidencia que las características visuales son más complejas de representar y que por tanto aporte en el estudio de los gustos de los usuarios. Por otro lado la modificación del procedimiento de entrenamiento adversario mejora levemente la AUC en el caso de Amazon Men pero no así en el caso de Amazon Women, pero cabe destacar que es considerablemente más eficiente al considerar solo 30 (15 + 15) épocas totales versus las 225 (15 · 10 + 15 · 5) propuesta. Estos hechos evidencian que la metodología adversaria propuesta es difícil de calibrar para obtener un entrenamiento óptimo, y en ese escenario, es más factible ver al método como un ajuste fino del modelo obtenido en el paso de inicialización, razón por la cual es recomendable considerar una versión liviana de entrenamiento como la propuesta en 30 épocas totales. Finalmente, la integración del sesgo de subjetividad para cada usuario aporta en mejorar consistentemente la AUC. Para medir el impacto de la metodología adversario sobre estos últimos dos modelos, se presenta las siguientes gráficas, que muestra la diferencia de AUC entre el modelo resultante como el mejor en la metodología adversaria y su inicialización correspondiente:

De ambas gráficas se puede extraer que para ambos métodos la metodología adversaria permite generalizar mejor, puesto que en cada caso la diferencia de AUC aumenta a favor del modelo adversario por sobre su inicialización y aquel incremento es mayor en test que en entrenamiento. Por ejemplo para el caso de Amazon Men, en el modelo que incluye las opiniones de los usuarios, el modelo que inicializa la metodología adversaria, es decir la factorización matricial $VTx_{u,i}$, es mejor que el resultante a través de la competencia, por lo que no se ve recompensado en entrenamiento el uso de la metodología adversaria. Sin embargo, el refinamiento producido por la metodología adversaria permite al modelo revertir la situación al momento de recomendar en el conjunto de test, es decir logra mejorar el modelo de inicialización a través de brindarle

Generalización de la metodología adversaria c/r a la inicialización

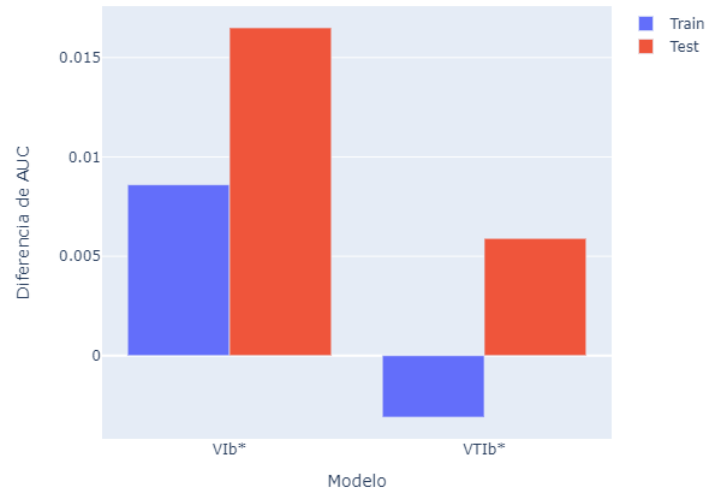


Figure 1: Impacto del entrenamiento minmax en Amazon men

Generalización de la metodología adversaria c/r a la inicialización

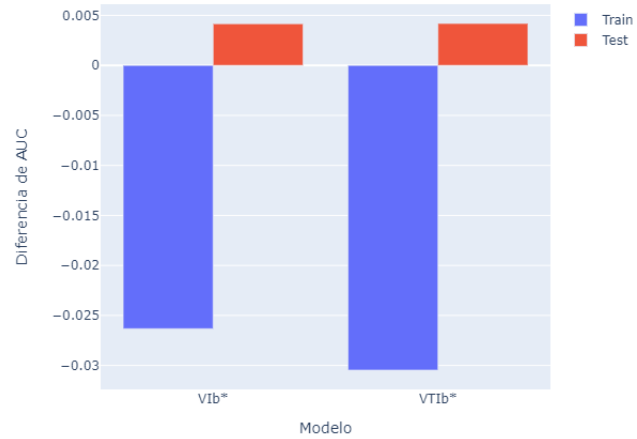


Figure 2: Impacto del entrenamiento minmax en Amazon women

mayor flexibilidad. Lo cual es un punto a favor de que la metodología adversaria puede ser utilizada como un ajuste fino posterior a un modelo de recomendación de factorización matricial, brindándole mayor capacidad de generalización, flexibilizando su ajuste al conjunto de entrenamiento a través de la recompensa que el modelo discriminador es capaz de transferir al modelo generador, ajustando su distribución de probabilidad en contra del sobre ajuste.

De esta manera es posible atribuir los mejores resultados de AUC a la integración del sesgo de emocionalidad del usuario, pero sin dejar de lado que es la metodología adversaria la que consistentemente mejora el modelo de factorización matricial al momento de inferir.

Todos lo métodos expuesto en un set up de Google Colab gratuito es posible ejecutarlos de manera completa en aproximadamente 1 hora o menos.

Por otro lado cabe mencionar que la integración del modelo del estado del arte DVBPR [7] se intentó a partir del código y recomendaciones de los autores, pero que tras 20 horas de entrenamiento para el caso de Amazon Men se registró un AUC en test de aproximadamente 0.67, muy lejos de lo presentado por los autores, por lo que se

considera un error sin una posible explicación y por ello no se presenta.

6 Conclusiones y trabajo futuro

En el presente trabajo se extendió el modelo de IRGAN, integrando características visuales de los items y opiniones de los usuarios. Esta extensión presentó una mejora de la AUC de manera consistente por sobre el resto de modelos más simples. Adicionalmente se estudió el impacto de la metodología adversaria como un ajuste fino de la factorización matricial de inicialización.

Este trabajo podría continuar con un estudio más detallado del impacto de las técnicas utilizadas en el entrenamiento de esta metodología adversaria discreta. De manera preliminar, dada la experimentación pareciera que el modelo discriminador es un clasificador demasiado fuerte como para que el modelo generador se beneficie de él, dicha situación se podría arreglar cambiando la pérdida del generador al estilo de una focal loss de manera de balancear el entrenamiento a favor de los items que selecciona el generador u encontrar los mejores hiperparámetros para balancear el problema de clasificación.

Otro factor que requiere de mayor estudio es el de la recompensa propuesta por los autores de IRGAN, puesto que de manera preliminar pareciera que la optimización de la pérdida del modelo generativo consiste en promover que dicho modelo copie la distribución de probabilidad actual del discriminador, por lo que el beneficio de encontrar un óptimo queda en duda. De la misma manera se debería estudiar bajo las distribuciones discretas adecuadas, el problema del colapso modal y su impacto según la manera de entrenamiento de la competencia, este hecho es importante estudiarlo puesto que puede estar influyendo directamente en la diversidad de la recomendación.

Otro estudio propuesto a futuro sería el reemplazo de factorizaciones matriciales como modelos base de entrenamiento por redes neuronales con embeddings uni, bi o tri modal según sea el escenario de recomendación, esto con la idea de que dichas representaciones latentes pueden llegar a beneficiarse también de la metodología adversaria.

Finalmente queda pendiente el correcto entrenamiento de DVBPR para establecer la comparación con el estado del arte, pero de todas formas el modelo propuesto en este trabajo es considerablemente más eficiente, y manteniendo las proporciones entregadas por el trabajo original con respecto al beneficio que saca por sobre VBPR podría ser esperable que VTib* sea estado del arte.

References

- [1] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. Cfgan: A generic collaborative filtering framework based on generative adversarial networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18*, page 137–146, New York, NY, USA, 2018. Association for Computing Machinery.
- [2] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. On sampling strategies for neural network-based collaborative filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17*, page 767–776, New York, NY, USA, 2017. Association for Computing Machinery.
- [3] Cach Dang, María Moreno García, and Fernando De La Prieta. An approach to integrating sentiment analysis into recommender systems. *Sensors*, 21:5666, 08 2021.
- [4] Yashar Deldjoo, Tommaso Di Noia, and Felice Antonio Merra. A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks. *ACM Comput. Surv.*, 54(2), mar 2021.
- [5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 06 2014.
- [6] Ruining He and Julian McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *AAAI*, 2016.
- [7] Wang-Cheng Kang, Chen Fang, Zhaowen Wang, and Julian McAuley. Visually-aware fashion recommendation and design with generative image models. 11 2017.
- [8] Steven Loria. textblob documentation. *Release 0.15*, 2, 2018.
- [9] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. pages 188–197, 01 2019.
- [10] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI '09*, page 452–461, Arlington, Virginia, USA, 2009. AUAI Press.
- [11] Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets, 2017.
- [12] J Wang, L Yu, W Zhang, Yu Gong, Y Xu, Benyou Wang, P Zhang, and D Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. 08 2017.
- [13] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. 09 2016.
- [14] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. Optimizing top-n collaborative filtering via dynamic negative item sampling. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013.

7 Anexo

Dataset	R	MP	BPR	VBPR	IRGAN	VI	VIb	VIb*	VTIb*
Men	0.4916	0.7617	0.7449	0.7642	0.7939	0.7962	0.8085	0.8090	0.8302
Women	0.4970	0.8141	0.8355	0.8077	0.8635	0.8244	0.8246	0.8062	0.8708

Table 1: AUC de los diferentes modelos en test

Selección de dimensionalidad de los embeddings

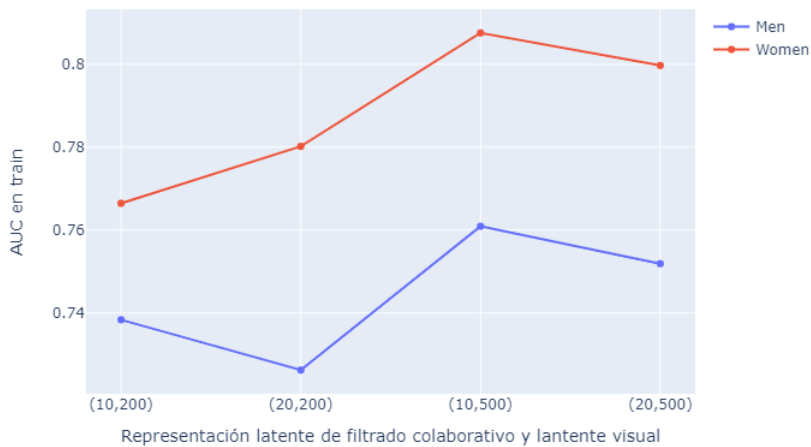


Figure 3: Selección de hiperparámetros

Polaridad promedio de review por usuario

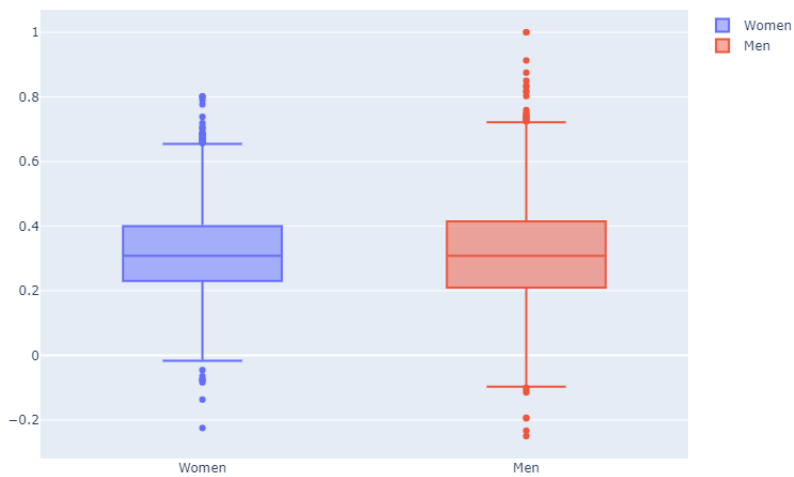


Figure 4: Distribución del análisis de sentimientos