SISTEMAS RECOMENDADORES

DENISPARRA

Profesor Asistente Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile. Doctor en Information Science, University of Pittsburgh, Estados Unidos; Ingeniero Civil en Informática, Universidad Austral de Chile. Lidera el laboratorio de investigación en Computación Social y Visualización (SocVis). Líneas de Investigación: Sistemas de Recomendación, Interfaces de Usuario Inteligentes, Minería de Datos, Visualización de Información, Análisis de Redes Sociales.
dparra@ing.puc.cl

"LO QUE LA INFORMACIÓN CONSUME ES BASTANTE OBVIO: CONSUME LA ATENCIÓN DE SUS DESTINATARIOS. POR LO TANTO, UNA GRAN CANTIDAD DE INFORMACIÓN CREA UNA POBREZA DE ATENCIÓN Y UNA NECESIDAD DE ASIGNAR ESA ATENCIÓN EFICIENTEMENTE ENTRE LA SOBREABUNDANCIA DE FUENTES DE INFORMACIÓN QUE PODRÍAN CONSUMIRLA".
HERBERT SIMON
PREMIO TURING (1975) Y PREMIO NOBEL DE ECONOMÍA (1978).

Uno de los principales efectos que ha tenido el avance de las tecnologías de la información y comunicación en los últimos treinta años –la Internet, la WWW, la tecnología móvil y sus aplicaciones– es permitirnos el acceso de manera casi instantánea a grandes volúmenes de información. Así mismo, muchas actividades que en el pasado sólo podíamos llevar a cabo movilizándonos físicamente como hacer compras, planificar viajes, hacer trámites para obtener certificados, etc., ahora podemos realizarlas de forma remota accediendo a la Web o vía aplicaciones móviles. De esta forma, las tecnologías han aumentado la disponibilidad de información, productos y servicios, y a la vez han facilitado enormemente su acceso.

A pesar de los grandes beneficios que nos trae la mayor disponibilidad y acceso a información, estos avances han producido también un problema: sobreabundancia de información. La aparición de la Web Social a mediados de la década 2000-2010 solo ayudó a exacerbar este problema, ya que cualquier persona –no solo programadores web—pueden generar contenido para la Web como blogs, vídeos o música.

Bajo este escenario, cuando queremos comprar un producto en línea y estamos a una distancia de un *click* de un catálogo de varios millones de productos, ¿cómo elegir el producto a comprar? ¿cómo asegurarnos que se están tomando en cuenta nuestras preferencias personales? Bajo este escenario es que los *sistemas recomendadores* han florecido y siguen creciendo día a día.

Tales sistemas tienen como rol principal ayudarnos a encontrar ítems relevantes dentro de una sobreabundancia de información [McNee et al, 2006] considerando nuestras preferencias individuales. Compañías tan diversas como Amazon, Netflix, Google, Booking y Spotify basan buena parte de sus funcionalidades y modelos de negocio en sistemas recomendadores. Estos sistemas se han desarrollado por más de treinta años pero han evolucionado especialmente rápido en la última década.

En este artículo presentamos estos sistemas a través de su evolución histórica, describiendo sus principales técnicas de implementación, así como algunos importantes dominios de aplicación y desafíos del área.

CONTEXTO HISTÓRICO Y EL PROBLEMA DE RECOMENDACIÓN

A comienzos de los años noventa, el tiempo requerido para filtrar mensajes de correo electrónico crecía velozmente y en Xerox PARC se estaba convirtiendo en un problema para sus funcionarios. El problema los motivó a desarrollar

	Promedio Rojo	Caluga o Menta	Machuca	. . .	Sexo con Amor
Usuario 1	5	2	1		?
Usuario 2	1	?	5		3
. . .					
Usuario n	1	3	?		1

FIGURA 1.
MATRIZ DE USUARIOS E ÍTEMS QUE PRESENTA EL PROBLEMA DE RECOMENDACIÓN COMO PREDICCIÓN DE LAS EVALUACIONES DE USUARIOS SOBRE ÍTEMS, EN ESTE CASO PELÍCULAS, AÚN NO CONSUMIDOS.

el sistema Tapestry [Goldberg et al, 1992], que les permitía filtrar mensajes usando de forma colaborativa las anotaciones de relevancia que los usuarios daban a los mensajes. Este sistema es usualmente reconocido como el primer sistema recomendador ya que introduce una de las técnicas más usadas en el área, el filtrado colaborativo, el cual explicaremos en detalle en la siguiente sección.

Los sistemas recomendadores se continuaron desarrollando por varios años, pero siempre dentro de una comunidad de investigadores más bien limitada, hasta el año 2006 en que la empresa de arriendo de películas por correo postal Netflix anuncia el Netflix Prize [Bennet y Lanning, 2007]. Este concurso, que se llevó a cabo hasta 2009, estuvo abierto vía Web a cualquier competidor y consistía en mejorar el algoritmo de predicción de preferencias de películas llamado *Cinematch* en al menos un 10%, con un premio de un millón de dólares. En este concurso se popularizó el siguiente modelo como problema de recomendación: dada una matriz de usuarios, ítems y evaluaciones explícitas (ratings) observadas, crear una función capaz de predeci la evaluación (rating) que un usuario daría a un ítem aún no consumido. La Figura 1 ejemplifica el problema descrito.

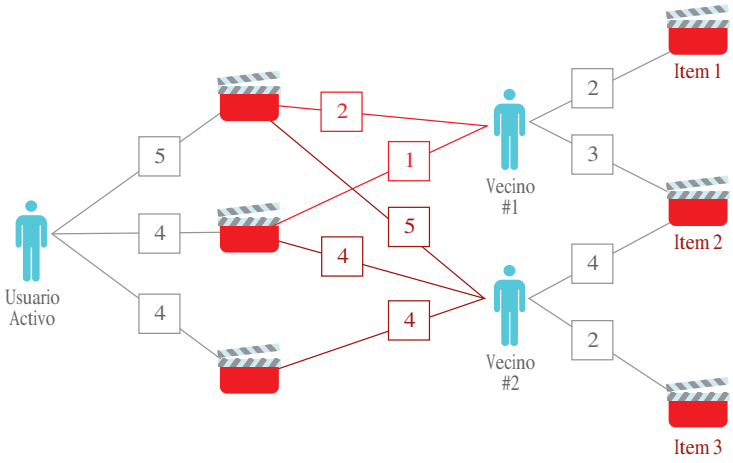

FIGURA 2.
EN EL FILTRADO COLABORATIVO BASADO EN EL USUARIO, SE CALCULAN LOS *K* VECINOS MÁS CERCANOS (VECINOS #1 Y #2), Y LUEGO SE INTENTA PREDECIR LA EVALUACIÓN QUE EL USUARIO ACTIVO DARÍA A ÍTEMS QUE NO HA CONSUMIDO, PERO QUE SUS VECINOS SÍ HAN CONSUMIDO (ÍTEMS 1, 2 Y 3).

El efecto positivo del concurso fue hacer más notoria el área de sistemas recomendadores entre ingenieros, desarrolladores y científicos de computación y otros campos, además de acelerar el desarrollo de algoritmos. Sin embargo, también generó el perjuicio de centrar la investigación de sistemas recomendadores en un dominio específico (recomendación de películas) y en un tipo específico de evaluación, como es minimizar métricas de error de predicción como RMSE y MAE [Parra y Sahebi, 2013]. En los años posteriores al Netflix Prize se ha visto un resurgimiento de otras formas de formular el problema de recomendación y de evaluar la calidad de estos sistemas.

TÉCNICAS DE RECOMENDACIÓN

Siguiendo la definición de Adomavicious y Tzuhilin [2005], podemos definir el problema de recomendación de forma general como encontrar una función de utilidad $u(c, S)$ tal que dado un usuario c y potenciales elementos en un dataset S, retorne un conjunto de ítems R que maximicen la utilidad del usuario:

$$\forall c \in C, s'_c = \underset{s \in S}{\operatorname{argmax}}\, u(c, s)$$

$$u : C \times S \rightarrow R, \textit{ funcion de utilidad}$$

Eq1: Formulación del problema de recomendación basado en Adomavicious y Tzuhilin [2005].

FILTRADO COLABORATIVO

El método conocido como filtrado colaborativo se basa en la intuición de que usuarios con gustos similares pueden servir de base para predecir gustos en el futuro [Resnick et al. 1994]. La versión más común de este método es el filtrado colaborativo basado en el usuario, que se puede ver en la Figura 2. Este método comprende dos pasos principales: (a) dado un "usuario activo" a quien se le desean realizar recomendaciones, encontrar

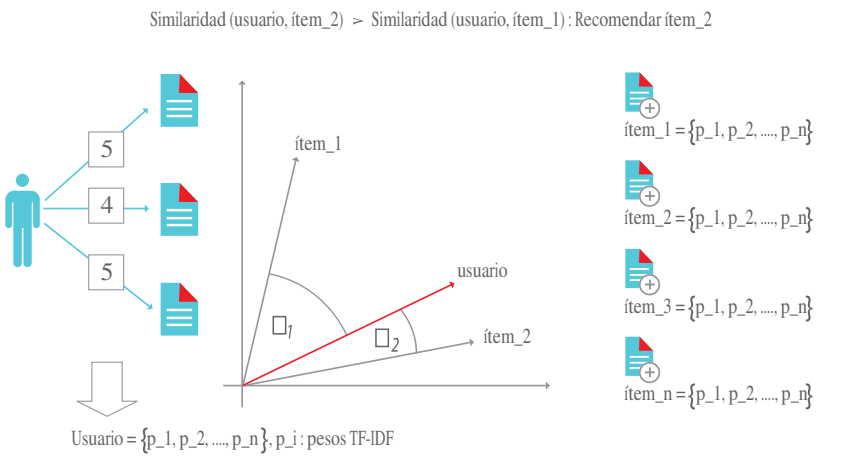

FIGURA 3.
REPRESENTACIÓN DEL FILTRADO BASADO EN CONTENIDO. EL PERFIL DEL USUARIO ES REPRESENTADO POR LOS ÍTEMS QUE HA CONSUMIDO Y SE BUSCAN ÍTEMS A RECOMENDAR EN LA BASE DE DATOS EN BASE A UNA FUNCIÓN DE SIMILARIDAD, QUE PODRÍA SER EL COSENO DEL ÁNGULO ENTRE EL VECTOR DEL USUARIO Y EL VECTOR DEL ÍTEM.

a los usuarios con mayor similaridad en cuanto a sus ratings, es decir, sus vecinos más cercanos, y luego (b) predecir el rating que el "usuario activo" dará a un ítem multiplicando la similaridad del usuario central con el vecino por el rating dado por el vecino.

Si bien este modelo es adecuado para datasets pequeños, en la práctica es poco escalable así es que suele usarse la alternativa llamada filtrado colaborativo basado en ítems [Sarwar et al, 2001]. Además, este método presenta tres grandes problemas: (a) partida en frío *(cold start)*: usuario nuevo que ha dado muy pocas o ninguna valoración a ítems, (b) nuevo ítem: ítem nuevo en la base de datos que pocos usuarios han evaluado, y (c) escasez de datos *(sparsity)*: es esperable que pocos usuarios den valoración explícita, pero cuando menos de un 2% de las *n x m* celdas de una matriz usuario-ítem posee datos, el algoritmo de filtrado colaborativo tiende a fallar en las predicciones [Parra y Sahebi, 2013].

FILTRADO BASADO EN CONTENIDO

La técnica de recomendación basada en contenido, si bien menos precisa en la predicción de ratings, se suele usar para aliviar problemas del filtrado colaborativo como el "nuevo ítem" y la escasez de datos. En esta técnica se suele representar al usuario como un vector de las características de los ítems que ha evaluado positivamente. Como los ítems no consumidos por el usuario se pueden representar igualmente como vectores, es posible hacer recomendaciones calculando la similaridad de un usuario y un ítem en base, por ejemplo, a su distancia coseno [Pazzani et al, 2007], como se ejemplifica en la Figura 3.

Al contrario que el filtrado colaborativo, los sistemas de recomendación basados en contenido permiten recomendar ítems que no han sido evaluados por ningún usuario, eliminando el problema del "nuevo ítem". Sin embargo, tienden a la sobreespecialización y corren el riesgo de caer en una burbuja de filtrado. Por ejemplo, si hago siempre "like" a noticias de mi equipo de fútbol favorito, el recomendador tenderá a mostrar sólo esas noticias y no las de otros equipos. Este ejemplo parece no ser tan nocivo, pero en el caso de política, por ejemplo, esto puede reforzar nuestra posición o sesgo, evitando que pudiéramos tener acceso a contenido más diverso.

SISTEMAS BASADOS EN REGLAS Y SISTEMAS HÍBRIDOS

En ocasiones los usuarios de sistemas recomendadores tienen restricciones o limitaciones específicas. Por ejemplo, para comprar una cámara, un usuario podría tener un presupuesto específico o preferencias por una cámara en particular. En estos casos, los sistemas basados en reglas [Jannach et al, 2010] tienen una ventaja importante. Sin embargo, estos sistemas tienen la debilidad de ser difíciles de mantener actualizados ya que los usuarios cambian sus preferencias en el tiempo y las reglas suelen fijarse de forma relativamente manual, algo que en inglés se denomina "user interest drift" [Koren et al, 2009].

Otra solución es la de los sistemas híbridos, los cuales combinan las fortalezas de distintos métodos y así permitir sistemas más robustos a la escasez de datos o a la sobreespecialización. El trabajo de Burke [2007] muestra distintas técnicas de combinación de recomendadores, que se pueden agrupar en 3: a) monolítico, b) paralelizado y c) pipeline. Para más detalle, revisar [Jannach et al, 2010].

FACTORIZACIÓN MATRICIAL

La factorización matricial es una de las técnicas más comunes para implementar sistemas de recomendación debido a su buen poder predictivo y a su rapidez en generar recomendaciones en línea. Se hizo popular luego de ser uno de los principales métodos usados para ganar el Netflix Prize [Koren et al, 2009]. En esta técnica, la intuición se basa en que tanto usuarios como ítems pueden ser llevados a un espacio "latente" común en base a sus interacciones, como se ve en la Figura 4.

La forma de obtener los valores de los usuarios y de los ítems en el espacio latente es a través de una optimización como la descrita en la ecuación siguiente, donde se intenta obtener los valores

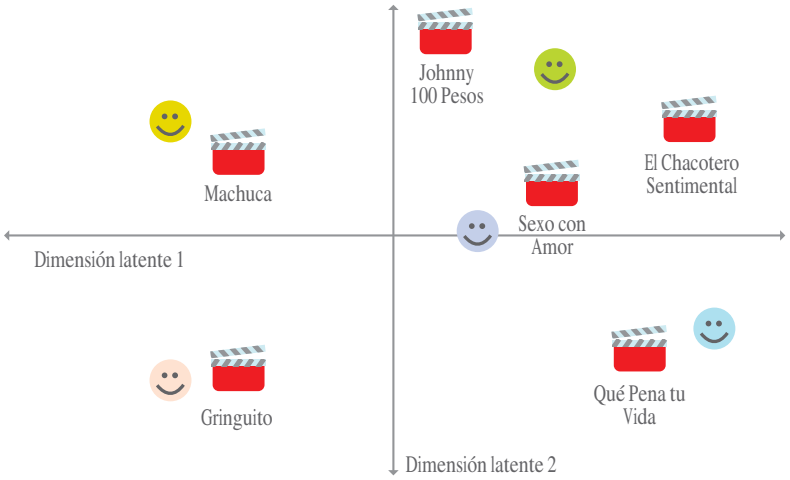

FIGURA 4.
ÍTEMS Y USUARIOS DISPUESTOS EN UN ESPACIO LATENTE COMÚN LUEGO DE UNA FACTORIZACIÓN MATRICIAL.

de los vectores q^* y p^* que minimizan el error de predicción

$$\min_{q^*,p^*} \sum_{(u,i)\in K} (r_{ui} - q_i^T p_u)^2 + \lambda(|| q_i ||^2 + || p_u ||^2)$$

Eq. 2 Modelo de factorización regularizado. El objetivo es aprender los valores de los vectores q_i y p_u, los cuales al multiplicarse permiten hacer una predicción. Los términos a la derecha corresponden a la regularización, que permite penalizar valores muy altos dentro de los vectores q_i y p_u y así evitar el sobreentrenamiento.

En la ecuación anterior, p_u y q_i son vectores latentes del usuario u y del ítem i, respectivamente, y λ corresponde al factor de regularización. La regularización permite controlar el sobreentrenamiento (*overfitting*). A partir de este modelo de optimización, decenas de otros métodos han extendido esta formulación ya sea incorporando variables nuevas (*feedback* implícito, contactos en una red social, efecto del tiempo), nuevos tipos de regularización (L_1 y $L_{1,2}$) y formas de aprendizaje de los factores latentes (SGD, ALS, etc.)

Uno de los principales problemas de la factorización matricial tradicional es su falta de versatilidad para incluir variables de contenido o contexto nuevas para hacer los modelos (por ejemplo, la hora en que se realizó el consumo). Parte de este problema fue resuelto por Steffan Rendle, quien introdujo las máquinas de factorización [Rendle, 2012]. Este modelo generaliza muchos otros modelos publicados anteriormente y es capaz de replicarlos sólo con cambiar el formato de los datos, además de introducir una forma eficiente de aprender varios de los parámetros del modelo. El modelo de Rendle ha sido tan exitoso que se ha ocupado en varias tareas de minería de datos más allá de sistemas de recomendación, con excelentes resultados en muchos concursos en línea como los del portal Kaggle[1].

DESAFÍOS DE LOS SISTEMAS DE RECOMENDACIÓN

Los sistemas de recomendación han avanzado rápidamente en la última década. El problema inicial de recomendación basado en predicción de ratings ha dado paso a muchas variantes. Es importante considerar, por ejemplo, que la recomendación en distintas áreas tiene particularidades que requieren de modelos diferentes. Probablemente, un usuario de un servicio de música, como Spotify o iTunes, no tendría problema en escuchar la misma canción en repetidas ocasiones. Sin embargo, se aburriría si Netflix le recomendara la misma película varias veces. Por otro lado, recomendar un restaurant o un café puede tener un aspecto geográfico importante relacionado, ya que un usuario podría tener restricciones de tiempo y distancia para acceder al servicio más allá de una predicción de rating. Los siguientes puntos detallan algunos desafíos de los sistemas de recomendación.

Evaluación. Considerar sólo métricas objetivas como el RMSE para evaluar un recomendador es nocivo porque muchas variables inciden en la decisión de un usuario de aceptar una recomendación. Factores como la diversidad de las recomendaciones, la transparencia del sistema para explicar las recomendaciones o la sensación de control del usuario respecto de la interfaz han mostrado ser muy importantes, al punto que mejoras pequeñas en RMSE no están en absoluto correlacionadas con una mayor satisfacción del usuario. Un gran desafío en estos sistemas es buscar la mejor manera de evaluarlos de forma holística [Konstan and Riedl, 2012], considerando una evaluación centrada en el usuario.

Contexto. Un área importante en la investigación y desarrollo de sistemas de recomendación es el contexto bajo el cual se consume un ítem. Un ejemplo típico es considerar con quién se va a ver una película: si el sistema tiene información de que se verá la película con un hermano menor de siete años, debería producir una recomendación distinta a si se ve con la pareja. El área de sistemas recomendadores basados en Contexto ha tenido un gran desarrollo en los últimos años y sigue creciendo en la actualidad [Adomavicius et al, 2015].

Aprender a Rankear. Un tema que inició su desarrollo en el área de recuperación de información pero que permeó rápidamente a los sistemas de información es la de "aprender a rankear" (Learning to Rank). Este tema produce un cambio de

1. https://www.kaggle.com/

paradigma importante, ya que el foco en predicción de ratings se transforma en producir directamente una lista ordenada de recomendaciones considerando las preferencias parciales de los usuarios y usando métricas distintas al RMSE, como nDCG, MAP, o MRR [Karatzoglou et al., 2013].

Aprendizaje Profundo (Deep Learning). El impacto que han tenido en los últimos cuatro años las redes neuronales profundas en el avance de modelos de visión por computador, de procesamiento de lenguaje natural y de juegos ha llegado de forma tardía pero con alto impacto a los sistemas de recomendación. En la última conferencia del área realizada en Boston en 2016 (ACM RecSys 2016), varios modelos incorporaron redes neuronales profundas con resultados de gran calidad, destacando el modelo de recomendación de vídeos del sitio YouTube presentado por Google [Convington et al, 2016].

CONCLUSIÓN

LOS SISTEMAS RECOMENDADORES CUMPLEN UN ROL ESENCIAL BAJO EL ESCENARIO ACTUAL DE SOBRECARGA DE INFORMACIÓN, Y SON UNO DE LOS MEJORES EJEMPLOS DE LOS AVANCES DE LA INTELIGENCIA ARTIFICIAL. SIN EMBARGO, HAY MUCHO TRABAJO PENDIENTE EN SU DESARROLLO. RECIENTEMENTE, EL GOBIERNO DE ESTADOS UNIDOS LIBERÓ UN INFORME CON MUCHOS DESAFÍOS COMO RESULTADO DE LOS AVANCES DE LA INTELIGENCIA ARTIFICIAL[2]. EL INFORME MENCIONA TEMAS CON DIRECTO IMPACTO EN EL ÁREA DE SISTEMAS DE RECOMENDACIÓN, POR EJEMPLO: (A) TRANSPARENCIA EN LOS SISTEMAS, (B) PRIVACIDAD, Y (C) ROBUSTEZ DE LOS SISTEMAS DE RECOMENDACIÓN. ESPERAMOS QUE LOS SISTEMAS RECOMENDADORES SE SIGAN DESARROLLANDO COMO HERRAMIENTAS ÚTILES Y QUE LOS ALGORITMOS A DESARROLLAR APUNTEN A UN ACCESO INCLUSIVO DE LA INFORMACIÓN PARA EVITAR SITUACIONES COMO EL FILTER BUBBLE[3], EL CUAL PODRÍA DAÑAR NUESTRA SOCIEDAD REFORZANDO NUESTROS SESGOS DE OPINIÓN Y COGNITIVOS EN LUGAR DE VOLVERNOS MÁS INCLUSIVOS. ■

REFERENCIAS

Bennett, J., & Lanning, S. (2007). The Netflix Prize. In Proceedings of KDD cup and workshop (Vol. 2007, p. 35).

Parra, D., & Sahebi, S. (2013). Recommender systems: Sources of knowledge and evaluation metrics. In Advanced Techniques in Web Intelligence-2 (pp. 149-175). Springer Berlin Heidelberg.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, 17(6), 734-749.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM conference on Computer supported cooperative work (pp. 175-186). ACM.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In The adaptive web (pp. 325-341). Springer Berlin Heidelberg.

Burke, R. (2007). Hybrid web recommender systems. In The adaptive web (pp. 377-408). Springer Berlin Heidelberg.

McNee, S. M., Kapoor, N., & Konstan, J. A. (2006). Don't look stupid: avoiding pitfalls when recommending research papers. In Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work (pp. 171-180). ACM.

Rendle, S. (2012). Factorization machines with libfm. ACM Transactions on Intelligent Systems and Technology (TIST), 3(3), 57.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8).

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). Recommender systems: an introduction. Cambridge University Press.

Adomavicius, G., & Tuzhilin, A. (2015). Context-aware recommender systems. In Recommender systems handbook (pp. 191-226). Springer US.

Covington, P., Adams, J., & Sargin, E. (2016, September). Deep neural networks for Youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems (pp. 191-198). ACM.

Konstan, J. A., & Riedl, J. (2012). Recommender systems: from algorithms to user experience. User Modeling and User-Adapted Interaction, 22(1-2), 101-123.

Karatzoglou, A., Baltrunas, L., & Shi, Y. (2013). Learning to rank for recommender systems. In Proceedings of the 7th ACM conference on Recommender systems (pp. 493-494). ACM.

2. https://obamawhitehouse.archives.gov/sites/default/files/whitehouse_files/microsites/ostp/NSTC/preparing_for_the_future_of_ai.pdf
3. http://www.ted.com/talks/eli_pariser_beware_online_filter_bubbles/transcript?language=en