Filtros colaborativos (Collaborative Filtering)

El filtrado colaborativo es otro método distinto de predecir puntuaciones de usuarios a ítems. Sin embargo, en este método usamos las puntuaciones existentes de usuarios a ítems para predecir los ítems que no han sido valorados por el usuario al que queremos recomendar.

Para ello asumimos que las recomendaciones que le hagamos a un usuario serán mejores si las basamos en usuarios con gustos similares.

Los filtros colaborativos generalmente basan su lógica en las características del usuario. Los datos que se tienen del usuario se convierten en el centro de un filtro colaborativo. El sistema analiza las compras anteriores, las preferencias, las calificaciones que ha dado de otros productos, el importe medio de las compras, etc. y busca otros usuarios que se parecen a él y que han tomado decisiones parecidas. Los productos que han tenido éxito con usuarios similares, seguramente también le interesarán al nuevo usuario.

Concepto de similitud en sistemas de recomendación

Un concepto que es clave para los sistemas de recomendación es el concepto de similitud. Estos sistemas siempre hacen recomendaciones basándose en las similitudes entre entidades. Pero una cosa importante a recordar es que la similitud puede ser medida diferente dependiendo de lo que usted está midiendo.

Por ejemplo, una buena manera de medir la similitud entre entidades es usar la distancia euclídea que básicamente es medir la distancia en línea recta entre 2 punto.Tenemos otras medidas como:

- •La Distancia de Manhattan es la distancia entre dos puntos medidos a lo largo de los ejes en ángulos rectos

- •La similitud mediante el coseno del ángulo entre 2 puntos

¿Cómo obtenemos las recomendaciones para un usuario mediante filtrado colaborativo?

```
def get_movie_recommendations(user_movies):
    '''given a set of movies, it returns all the movies sorted by their correlation with the use
r'''
    movie_similarities = np.zeros(corr_matrix.shape[0])
    for movie_id in user_movies:
        movie_similarities = movie_similarities + get_movie_correlations(movie_id)
    similarities_df = pd.DataFrame({
        'movie_title': movie_index,
        'sum_similarity': movie_similarities
        })
    similarities_df = similarities_df[~(similarities_df.movie_title.isin(user_movies))]
    similarities_df = similarities_df.sort_values(by=['sum_similarity'], ascending=False)
    return similarities_df
```

Obtenemos la lista de películas que el usuario ha revisado, creamos una matriz vacía de longitud el número de películas. Para cada película de las películas del usuario, sumamos las correlaciones de esa película con la matriz que acabamos de crear. Al final ordenamos las similitudes resultantes por orden descendente.

En el filtrado colaborativo,para realizar recomendaciones se tienen en cuenta las preferencias de usuarios similares a aquél que es recomendado. La idea detrás de este tipo de filtrado es que si 2 usuarios A y B tienen intereses similares en una serie de aspectos conocidos ,entonces se puede emplear la información del usuario A para inferir información desconocida sobre el usuario B

Las reglas de recomendación en el filtrado colaborativo son del tipo "otros usuarios que compraron el producto X también compraron el producto Y"

Ventajas filtrado colaborativo

- Su dominio es abierto.
- Puede abordar aspectos de los datos que son a menudo difíciles de alcanzar y difíciles de perfilar usando filtrado basado en contenido.
- Generalmente son más precisos que las técnicas basadas en contenido.

La principal **ventaja** de este enfoque respecto al filtro basado en contenidos es que se pueden **ofrecer ítems que son completamente distintos al actual, permitiendo así ofrecer un conjunto de recomendaciones más amplio.**

Inconvenientes filtro colaborativo

- Sufren del conocido problema Cold Start, debido a su incapacidad para manejar ítems y usuarios nuevos en el sistema.

Este tipo de sistema sabe lo que puede gustar a un usuario basándose en las preferencias de otras personas con gustos similares. Como ya se ha comentado anteriormente, a este tipo de sistema pertenecen los proyectos de **Filmafinity y MovieLens** donde los usuarios puntúan películas para recibir recomendaciones sobre otras películas basándose en sus gustos

Este tipo de sistema de recomendación, se divide a su vez en dos subgrupos:

User-User collaborative filtering

Estos sistemas son capaces de recomendar ítems basándose en los gustos de otras personas que muestren gustos similares. Estos sistemas recogen medidas de interés o ratings de los distintos usuarios. El algoritmo calcula la distancia entre cada par de usuarios del sistema para medir cuánto se parecen dos usuarios que han puntuado los mismos ítems. Los usuarios cuyos gustos son afines de acuerdo a estos cálculos forman un vecindario (“neighborhood”).

Item-Item collaborative filtering

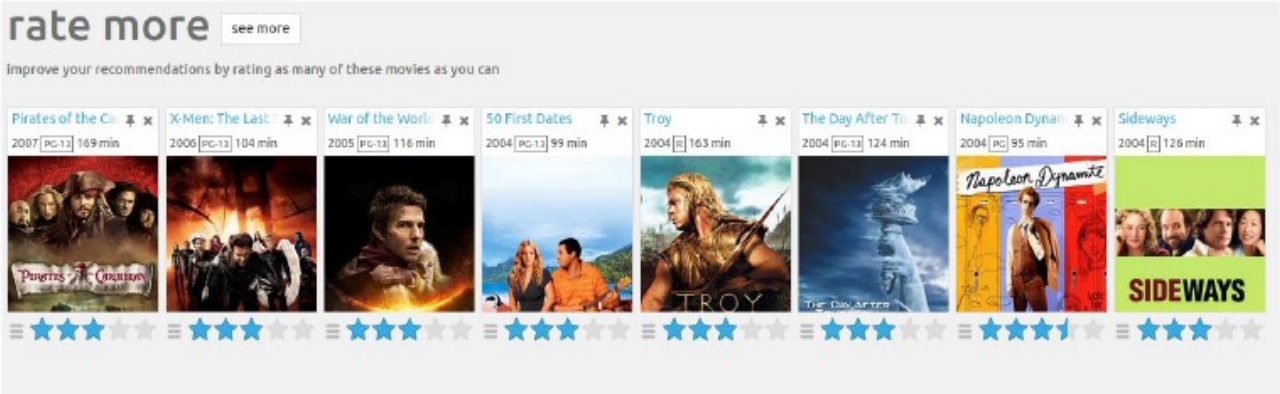
Estos sistemas calculan la distancia entre cada par de ítems basándose en la calificación que les han otorgado los distintos usuarios que comparten gustos similares. El algoritmo calcula similaridad entre pares de ítems. Esta técnica considera que las preferencias de un usuario no se pueden modelar únicamente a través de un conjunto de palabras claves que las representen, considera que los gustos y preferencias de un usuario, se pueden ver reflejados en los gustos y preferencias de otro usuario.

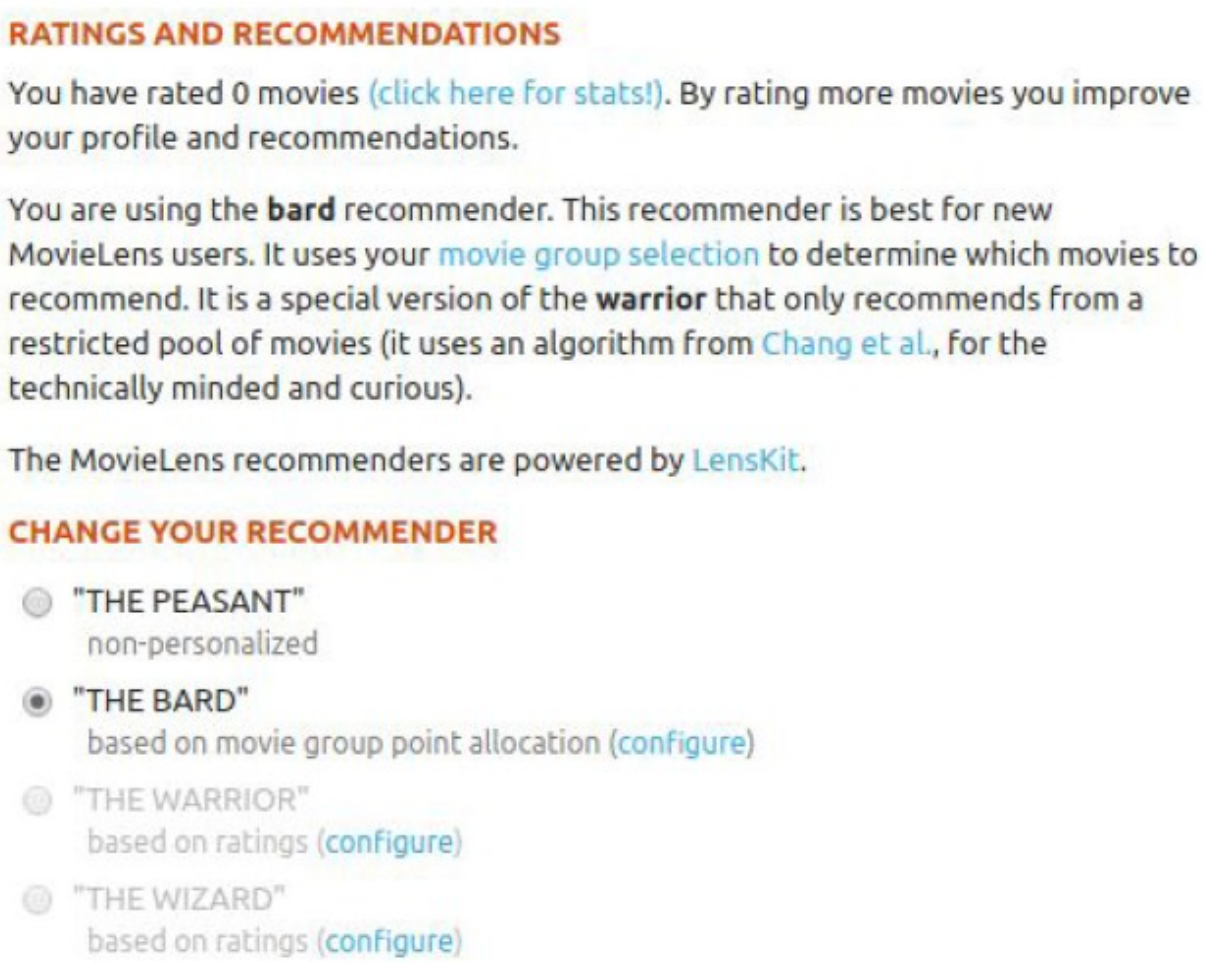
Algunos de los proyectos que usan **sistemas basados en filtrado colaborativo** ,como el proyecto GroupLens, que desarrolló distintos sistemas como Usenet News o MovieLens.

Usenet News es un sistema que permite a sus usuarios valorar las distintas noticias,y es capaz de predecir qué tipo de noticia le gustará leer a cada usuario basándose en las preferencias de otros usuarios que comparten las mismas opiniones (Nearest Neighbor Aproach).

MovieLens es un sistema de recomendación de películas que igualmente permite a sus usuarios crear opiniones. Este sistema obtiene la correlación de cada usuario con respecto al resto de usuarios del sistema, basándose en dichas opiniones, para encontrar un grupo de usuarios afines a dicho usuario, de tal modo que el sistema pueda realizar recomendaciones a dicho usuario basándose en los gustos de dicho grupo.

https://movielens.org/home





Los desarrolladores de sistemas de recomendación actuales han aplicado distintos enfoques para manejar y procesar todos los datos, pero el enfoque que se ha asentado principalmente en todos ellos es el de la recomendación colaborativa personalizada. Este tipo de sistema de recomendación es el corazón de Amazon, Netflix, las recomendaciones de amigos en la famosa red social Facebook y Last.fm.

Estos sistemas se denominan “personalizados” porque rastrean el comportamiento de cada usuario, páginas visitadas, compras y puntuaciones, para generar recomendaciones adaptadas a las necesidades y gustos de cada usuario.

Del mismo modo, son “colaborativas” porque relacionan los distintos artículos basándose en el hecho de que varios usuarios hayan comprado el mismo artículo o bien muestran cierta preferencia por ellos.

Movielens dataset

A continuación están los datasets soportados hasta el momento y el código necesario para cargarlos:

- **Sencillo - 100k** . Ideal para pruebas y algoritmos. [5 MB] - 100,000 ratings, 6000 usuarios, 4000 películas
- http://files.grouplens.org/datasets/movielens/ml-100k.zip

- **Grande - 20M** . Seguramente correrá bastante lento. Se puede usar para entrarle a rollos de optimización. [138 MB] - 20,000,000 ratings, 138,000 usuarios, 27,000 películas
- http://files.grouplens.org/datasets/movielens/ml-20m.zip

- **Grande actualizado - 21M** . Dataset similar al anterior, pero es constantemente actualizado (i.e. seguirá creciendo). [+144 MB] - 21,000,000 ratings, 230,000 usuarios, 30,000 películas
- http://files.grouplens.org/datasets/movielens/ml-latest.zip

El conjunto de datos de movielens contiene un millón de puntuaciones anónimas, con valoraciones entre 1 y 5, sobre aproximadamente 3900 películas realizadas por 6040 usuarios recogidas en MovieLens en el año 2000, siendo la densidad de la matriz de puntuaciones del 4.25 %.

El conjunto de datos se encuentra formado por 3 ficheros:

movies.dat: La información relativa a cada una de las películas disponible en el conjunto de datos se encuentra en este fichero proporcionado en formato csv con las siguientes columnas mostradas en la tabla.

- **MovieID:** Identificador numérico único de la película.
- **Title** : Título de la película junto con el año de estreno. Los títulos son idénticos a los títulos proporcionados por la base de datos IMBD
- **Genres** : Lista de géneros separados por el carácter “|” al que pertenece la película:
  - Action
  - Adventure
  - Animation
  - Children’s
  - Comedy
  - Crime
  - Documentary
  - Drama
  - Fantasy
  - Film-Noir
  - Horror
  - Musical
  - Mystery
  - Romance

- Sci-Fi
- Thriller
- War
- Western

ratings.dat: La información relativa a cada una de las puntuaciones de un usuario sobre una determinada película se encuentra disponible en este fichero proporcionado en formato csv con las columnas mostradas en la Tabla. Cada usuario tiene al menos 20 puntuaciones.

- **UserIDs** : Los identificadores de los distintos usuarios (Entre 1 y 6040)
- **MovieIDs** : Los identificadores de las películas (Entre 1 y 3952)
- **Ratings** : Las puntuaciones dadas por cada uno de los usuarios (Entre 1 y 5).
- **Timestamp** : Fecha y hora en la que se realizó la puntuación representada en segundos

	user_id	movie_id	rating	timestamp
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291