

Sistemas Recomendadores

IIC-3633

Slope one

Esta clase

1. Repaso User-based e Item-Based Collaborative Filtering
2. Recomendación basada en Slope One
3. Restricted Boltzmann Machines

Comentarios del paper de filtrado colaborativo

(Table 1). Without loss of generality, a ratings matrix consists of a table where each row represents a user, each column represents a specific movie, and the number at the intersection of a row and a column represents the user's rating value. The absence of a rating score at this intersection indicates that user has not yet rated the item.

Table 1. A MovieLens ratings matrix. Amy rated the movie Sideways a 5. Matt has not seen The Matrix

	The Matrix	Speed	Sideways	Brokeback Mountain
Amy	1	2	5	
Matt		3	5	4
Paul	5	5	2	1
Cliff	5	5	5	5

idea como MovieLens me parece elegante. Captura la esencia de las recomendaciones y de las interacciones entre usuarios de una manera bastante simple y fácil de manejar.

La traducción de este problema a datos en una tabla (matriz) facilita bastante la manipulación y el almacenamiento de datos muy ricos para un sistema recomendador.

A

Una mejora a esta forma de atacar el problema sería incluir información más rica, por ejemplo, dejar que los usuarios puedan escribir reviews para poder utilizar modelos de DNNs para poder aprender representaciones ricas sobre las interacciones entre los "users" y los "items".

Para finalizar me gustaría comentar sobre las entradas faltante para la matriz que se menciona. Estas entradas faltantes con los datos necesarios podrían aprenderse por medios de los famosos LLMs, de hecho ese es uno de los rumbos actuales para abordar el problema como por ejemplo en el siguiente [artículo](#).

Vamos a ver en recomendación basada en contenido como incorporar información del texto en forma de embeddings para complementar al filtrado colaborativo.

(Table 1). Without loss of generality, a ratings matrix consists of a table where each row represents a user, each column represents a specific movie, and the number at the intersection of a row and a column represents the user's rating value. The absence of a rating score at this intersection indicates that user has not yet rated the item.

Table 1. A MovieLens ratings matrix. Amy rated the movie Sideways a 5. Matt has not seen The Matrix

	The Matrix	Speed	Sideways	Brokeback Mountain
Amy	1	2	5	
Matt		3	5	4
Paul	5	5	2	1
Cliff	5	5	5	5

The term *user* refers to any individual who provides ratings to a system. Most often, we use this term to refer to the people using a system to receive information (e.g., recommendations) although it also refers to those who provided the data (ratings) used in generating this information.

usuario en función de los ratings asignados a los ítems que consume, entiendo que es una forma básica de caracterización (se utiliza en los modelos de la sección 9.3). En ese sentido, me parece razonable incorporar otras variables que permitan enriquecer el "embedding" que finalmente representa a cada usuario/ítem. Por ejemplo, pensando en los ítems, sería valioso considerar el costo que tienen y su uso. Respecto a los usuarios, podría tomarse en consideración algunas variables demográficas, como edad o género, y variables que puedan limitar el nivel de consumo de una persona (nivel de ingresos). Por otro lado, sobre la medida rating, pueden existir diferentes puntos de vista que llevan a un grupo de usuarios a asignar una misma nota. En esa línea, se podrían evaluar técnicas de procesamiento de lenguaje natural para profundizar en la evaluación que realizan los usuarios (en caso de tener un review).

A nivel general, creo que esta forma de representar al usuario/ítem reduce demasiado la variabilidad inherente del problema. De todas maneras, esta simplificación puede ser valiosa en una etapa exploratoria para identificar potenciales relaciones entre usuarios/ítems.

M

En recomendación basada en contenido además de las interacciones consideramos metadata del usuario.

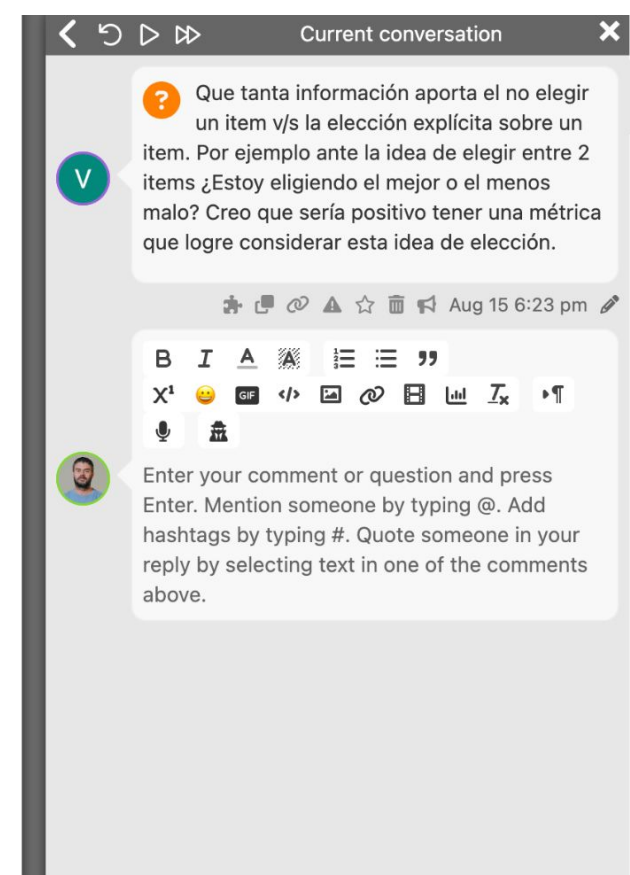
we use this term to refer to the people using a system to receive information (e.g., recommendations) although it also refers to those who provided the data (ratings) used in generating this information.

Collaborative filtering systems produce predictions or recommendations for a given user and one or more *items*. Items can consist of anything for which a human can provide a rating, such as art, books, CDs, journal articles, or vacation destinations.

Ratings in a collaborative filtering system can take on a variety of forms.

- Scalar ratings can consist of either numerical ratings, such as the 1-5 stars provided in MovieLens or ordinal ratings such as strongly agree, agree, neutral, disagree, strongly disagree.
- Binary ratings model choices between agree/disagree or good/bad.
- Unary ratings can indicate that a user has observed or purchased an item, or otherwise rated the item positively. The absence of a rating indicates that we have no information relating the user to the item (perhaps they purchased the item somewhere else).

Ratings may be gathered through explicit means, implicit means, or both. *Explicit ratings* are those where a user is asked to provide an opinion on an item. *Implicit ratings* are those inferred from a user's actions. For example, a user who visits a product page perhaps has some interest in that product while a user who subsequently purchases the product may have a much stronger interest in that product. The issues of design decisions and tradeoffs regarding collection of different types of ratings are discussed in section 9.4.



En feedback implícito existe una variable de cuanto se prefiere el ítem que incorpora al cálculo de preferencia.

Rocks and Minerals Society but also the “shallower” posting to a music bulletin board regarding one visitor’s opinion of the 1970s rock band.

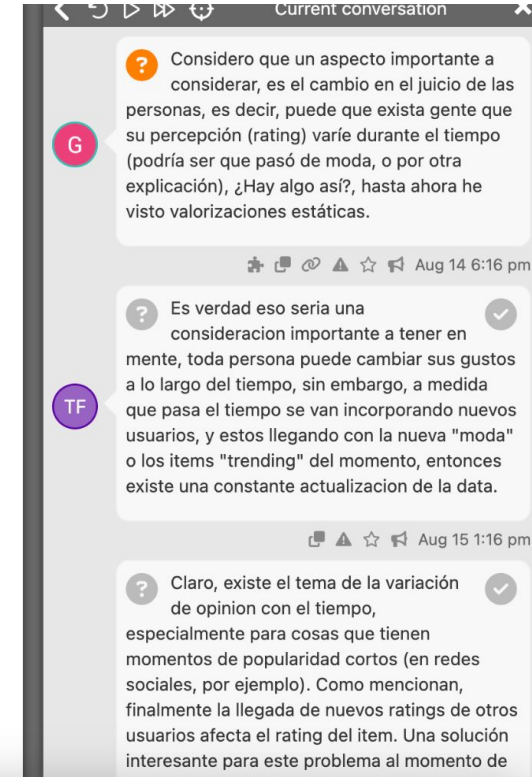
In the early 1990s there seemed to be two possible solutions to this new challenge:

1. wait for improvements in artificial intelligence that would allow better automated classification of documents, or
2. **bring human judgment into the loop.**

While the challenges of automated classification have yet to be overcome, human judgment has proved valuable and relatively easy to incorporate into semi-automated systems².

The Tapestry system, developed at Xerox PARC, took the first step in this direction by incorporating user actions and opinions into a message database and search system [19]. Tapestry stored the contents of messages, along with metadata about authors, readers, and responders. It also allowed any user to store annotations about messages, such as “useful survey” or “Phil should see this!” Tapestry users could form queries that combined basic textual information (e.g. contains the phrase “recommender systems”) with semantic metadata queries (e.g. written by John OR replied to by Joe) and annotation queries (e.g. marked as “excellent” by Chris). This model has become known as pull-active collaborative filtering, because it is the responsibility of the user who desires recommendations to actively pull the recommendations out of the database.

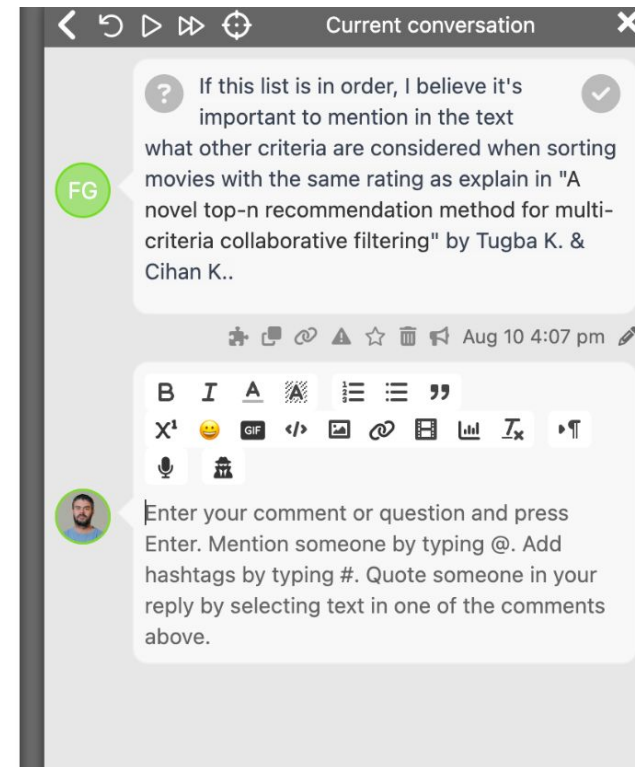
Soon after the emergence of Tapestry, other researchers began to recognize the potential for exploiting the human “information hubs” that seem to naturally occur within organizations. Maltz and Ehrlich [42] developed a push-active collaborative filtering recommender system that made it easy for a person reading a document to



Este es un problema frecuente en recomendación que es el cambio de preferencias en el tiempo. Evaluación offline con una foto del dataset no lo captura.

opinions in real time and determine not only what a much larger community thinks of an item, but also develop a truly personalized view of that item using the opinions most appropriate for a given user or group of users.

(hide) Predictions for you ↕	Your Ratings	Movie Information	Wish List
★★★★★	Not seen ▾	Secondhand Lions (2003) DVD VHS info imdb Adventure, Children, Comedy, Drama	<input type="checkbox"/>
[add tag]	Related tags: art house , classic , japan		
★★★★★	Not seen ▾	Holes (2003) DVD VHS info imdb Children, Comedy, Crime, Drama	<input type="checkbox"/>
[add tag]	Related tags: good , art study		
★★★★★	Not seen ▾	Three... Extremes (2004) info imdb Comedy, Horror - Cantonese, Japanese, Korean, Mandarin	<input type="checkbox"/>
[add tag]	Related tags: Miyazaki , get , comedy		
★★★★★	Not seen ▾	Seducing Doctor Lewis (Grande séduction, La) (2003) DVD VHS info imdb Comedy - French	<input type="checkbox"/>
[add tag]	Related tags: classic , dvd , Rome		
★★★★★	Not seen ▾	Jump Tomorrow (2001) DVD info imdb Comedy, Drama, Romance	<input type="checkbox"/>
[add tag]	Related tags: golf , teen , sports		



Alternativas de desempate: Timestamp , cantidad de ratings , metadata del ítem (popularidad, género, contenido) , metadata del usuario.

Ayudantía

LUNES 21 DE AGOSTO

MÓDULO 5 , SALA B18

LIBRERÍA SUPRISE PARA FILTRADO COLABORATIVO BASADO EN USUARIO
E ITEMS

Recomendación no personalizada, semi-personalizada y personalizada.

No
personalizada

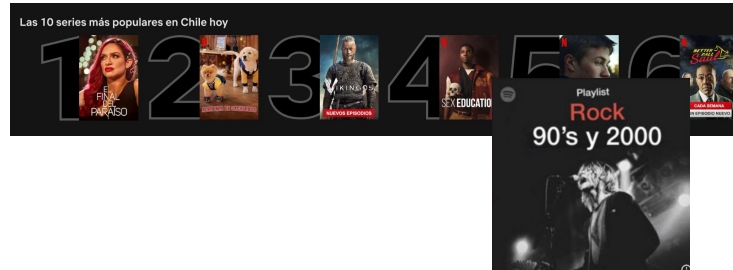
Semi- personalizada

Personalizada

- Más populares
- Random
- L2R



Basada en regla
general
(ej País,
Segmentación por
edad, intereses)



Basada en
intereses
personales

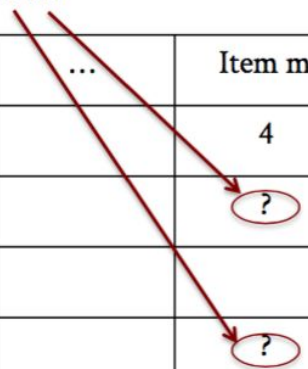
Filtrado colaborativo
Factores latentes
Basada en contenido
Ensamblados / Híbridos

Predicción de Ratings

- Los algoritmos que veremos el resto de esta clase simplifican el problema de recomendación considerándolo como un problema de predicción de ratings basado en coocurrencia :

Predict!

	Item 1	Item 2	...	Item m
User 1	1	5		4
User 2	5	1		?
...				
User n	2	5		?

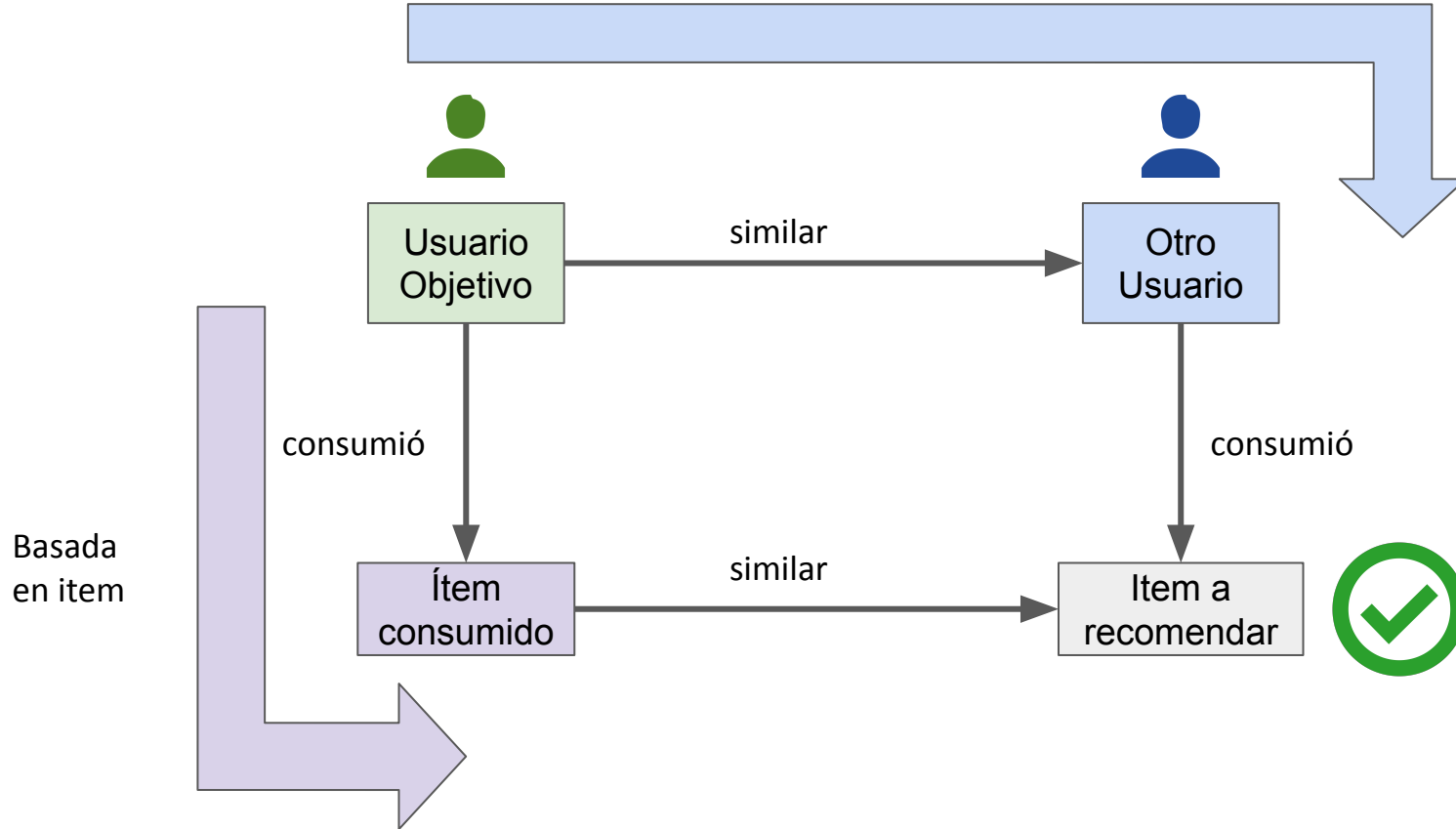


- ... y qué tan buena es mi predicción?

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{r}_{ui} - r_{ui})^2}{n}}$$

Filtrado colaborativo (resumen)

Basada en usuario



Slope One

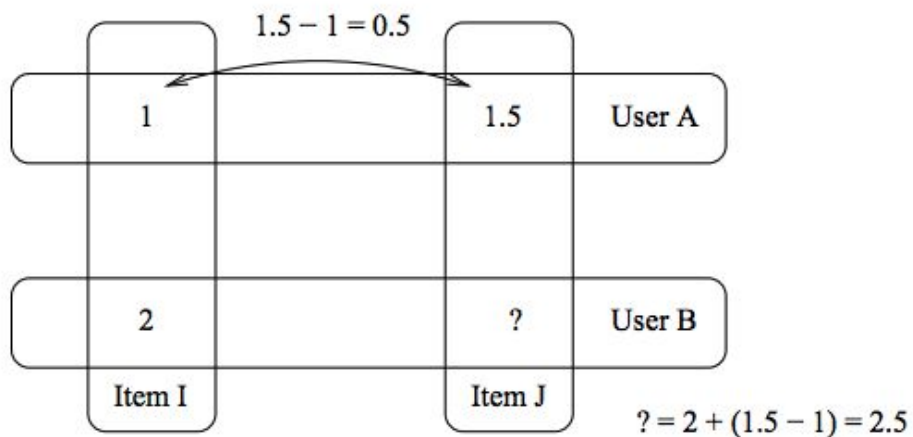
Recomendador de pendiente uno (Slope One)

Método basado en filtrado colaborativo.

- Enfoques de los autores:
 - Método fácil de implementar y mantener
 - Actualizable en línea
 - Eficiente al momento de consulta
 - Requiere poco *feedback* del usuario
 - Razonablemente preciso: pequeña ganancia en exactitude no signifique un gran sacrificio de simplicidad y escalabilidad

Slope One: intuición

- Diferencias de *ratings* entre pares de ítems generan una buena predicción



Slope One: formulación

$$P(u, i) = \frac{\sum_{j \in R(u) \setminus \{i\}} \text{freq}(i, j) \times (\text{dev}(i, j) + r(u, j))}{\sum_{j \in R(u) \setminus \{i\}} \text{freq}(i, j)}$$

Donde:

- $P(u, i)$: es la valoración predicha del ítem i para el usuario u .
- $R(u)$: es el conjunto de ítems que el usuario u ha valorado.
- $r(u, j)$: es la valoración real del ítem j para el usuario u .
- $\text{dev}(i, j)$: es la diferencia promedio de valoraciones entre los ítems i y j para todos los usuarios que han valorado ambos ítems. Se calcula como:

$$\text{dev}(i, j) = \frac{\sum_{u \in U} (r(u, i) - r(u, j))}{\text{freq}(i, j)}$$

Donde U es el conjunto de todos los usuarios que han valorado ambos ítems i y j .

- $\text{freq}(i, j)$: es el número de usuarios que han valorado ambos ítems i y j .


Reflexión

¿Qué información captura el modelo de pendiente uno o *slope one*?

¿Qué representa *dev*? ¿Cómo se generan las recomendaciones?

Slope One: ejercicio

Predecir el rating para el par (U3, spiderman)

Usuario	Harry Potter	Batman	Spiderman
U1	5.0	3.0	4.0
U2	-	2.0	4.0
U3	4.0	2.0	? 

Slope One: ejercicio

Para Spiderman y Harry Potter:

$$\text{dev}(\textit{Spiderman}, \textit{HarryPotter}) = -1.0$$

$$\text{freq}(\textit{Spiderman}, \textit{HarryPotter}) = 1$$

$$r(U3, \textit{HarryPotter}) = 4.0$$

$$\text{numerator}_{HP} = 1 \times (-1.0 + 4.0) = 3.0$$

$$\text{denominator}_{HP} = 1$$

Para Spiderman y Batman:

$$\text{dev}(\textit{Spiderman}, \textit{Batman}) = 1.5$$

$$\text{freq}(\textit{Spiderman}, \textit{Batman}) = 2$$

$$r(U3, \textit{Batman}) = 2.0$$

$$\text{numerator}_{Batman} = 2 \times (1.5 + 2.0) = 7.0$$

$$\text{denominator}_{Batman} = 2$$

Usuario	Harry Potter	Batman	Spider man
U1	5.0	3.0	4.0
U2	-	2.0	4.0
U3	4.0	2.0	?

Slope One: ejercicio

Usuario	Harry Potter	Batman	Spider man
U1	5.0	3.0	4.0
U2	-	2.0	4.0
U3	4.0	2.0	?

Sumando ambos numeradores y dividiendo por la suma de ambos denominadores obtenemos:

$$P(U3, Spiderman) = \frac{3.0+7.0}{1+2}$$

$$P(U3, Spiderman) = \frac{10.0}{3}$$

$$P(U3, Spiderman) = 3.33$$

Ejemplo de implementación de slope 1 en python

https://colab.research.google.com/drive/1QJSjcYAMywFz7b9v4I_f-Eyt7uaNgtY_?usp=sharing

Resultados Slope1 comparado con otros algoritmos de filtrado colaborativo.

Algoritmo	EachMovie	MovieLens
Slope One	0.200	0.188
User-based CF	0.209	0.198
item-based CF	0.231	0.208

MAE de distintos algoritmos de filtrado colaborativo.

Ventajas y Limitaciones

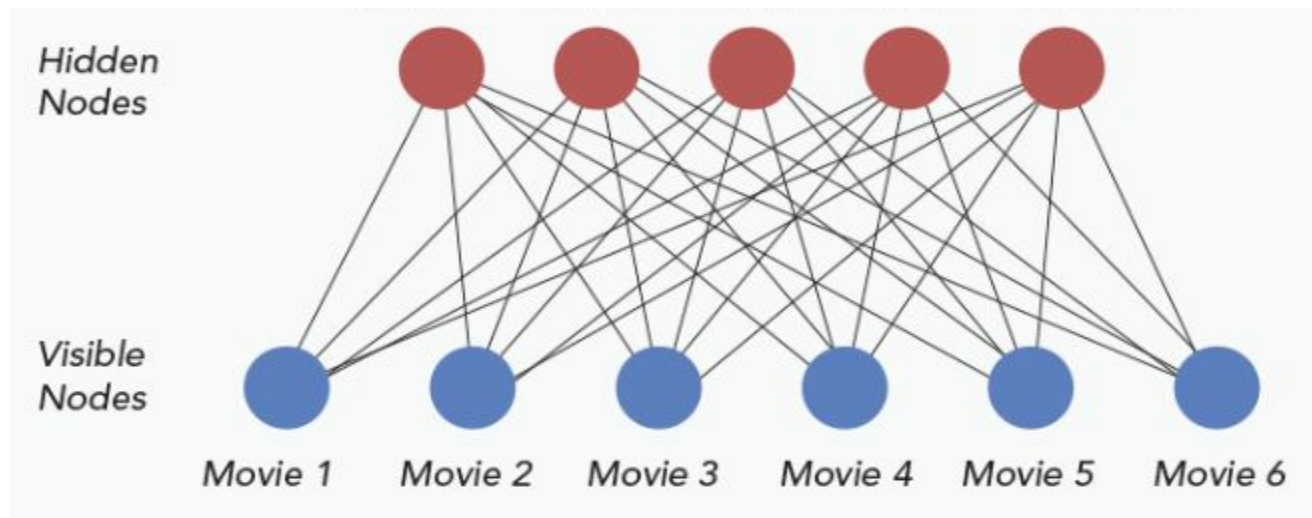
Ventajas de Slope 1:

Se supera en parte el costo computacional y se mejora el desempeño de Filtrado colaborativo basado en usuarios e ítems.

Limitaciones:

- Cold start
- Matriz de ratings sparsed

Restricted Boltzmann Machines (RBM)



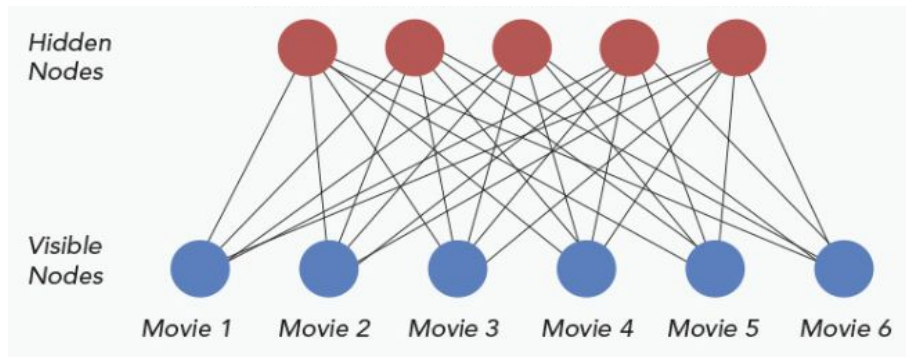
	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6
RATING	5	3	1	UNK	UNK	4

Entrenamiento

Existen

- variables visibles (ratings).
- variables ocultas (hidden features).

La tarea de entrenamiento es descubrir cómo estos dos conjuntos de variables están conectados entre sí.



Optimización

1. **Forward Pass:** Se introduce un conjunto de datos en la capa visible. Esta información se propaga a la capa oculta usando pesos iniciales.
2. **Reconstrucción:** Desde la capa oculta, se intenta reconstruir los datos originales en la capa visible.
3. **Contrastive divergence:** La diferencia entre el dato original y su reconstrucción se utiliza para calcular un gradiente. Este gradiente indica cómo cambiar los pesos y sesgos para mejorar la reconstrucción en futuras iteraciones.
4. **Actualización de pesos:** Se ajustan los pesos de la RBM en la dirección que sugiere el gradiente para reducir el error de reconstrucción.

Este proceso se repite muchas veces hasta que el error de reconstrucción converja a un valor mínimo o hasta que se cumpla un número predefinido de iteraciones.

Gibbs Sampling

Repite este proceso para **unidades visibles y ocultas**:

1. primero calcula el producto de pesos aleatorios (que se actualizan con el gradiente) con las unidades.
2. calcula la sigmoide luego de pasar el resultado anterior por una función de activación y retorna una probabilidad.
3. utiliza una distribución de bernoulli que samplea unidades basadas en la probabilidad obtenida.

Predicción

Dado un nuevo usuario con estos ratings:

[1,4 , UNK , 1, 5, 5 ...]

La RBM reconstruye el vector de ratings incluyendo aquellos que el usuario no ha calificado.

Referencias

Lemire, D., & Maclachlan, A. (2005, April). Slope one predictors for online rating-based collaborative filtering. In Proceedings of the 2005 SIAM International Conference on Data Mining (pp. 471-475). Society for Industrial and Applied Mathematics.