

Sistemas Recomendadores

IIC-3633

Ensamblés Parte 1

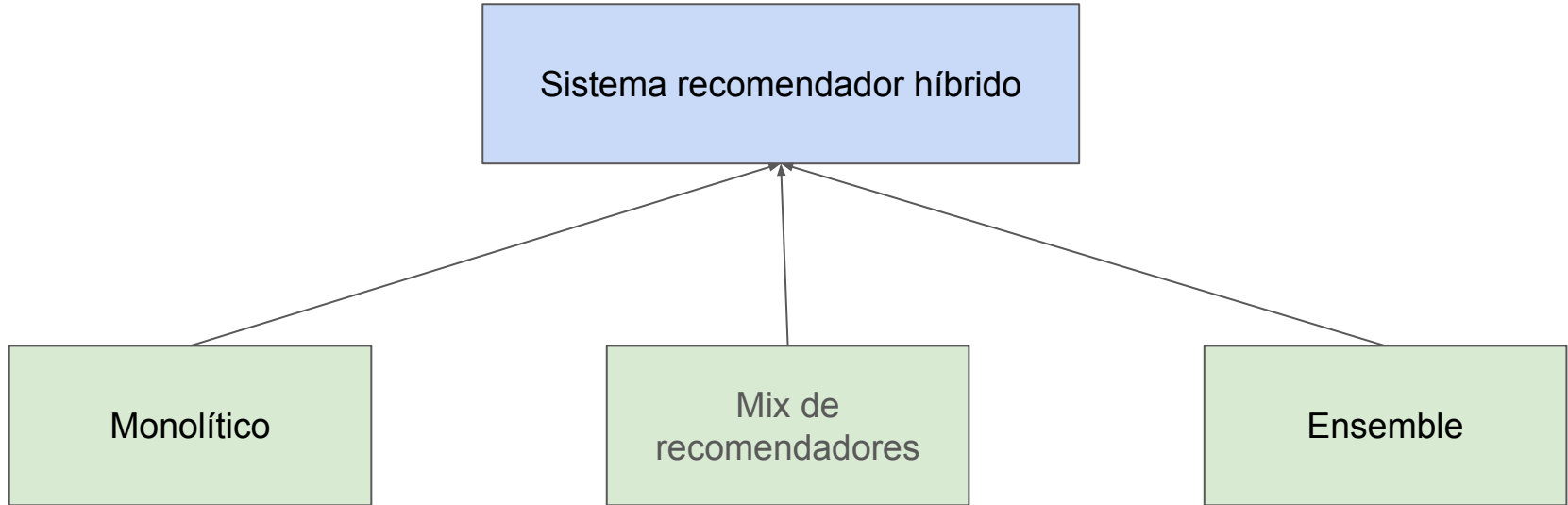
Esta clase

1. Recomendación basada en ensambles

¿Para qué me puede servir combinar varios recomendadores?

- Un recomendador puede tener mejor desempeño en ciertos patrones y otros no.
- Basado en contenido al no sufrir de cold start se podría combinar con filtrado colaborativo

Tipos de recomendación híbrida.

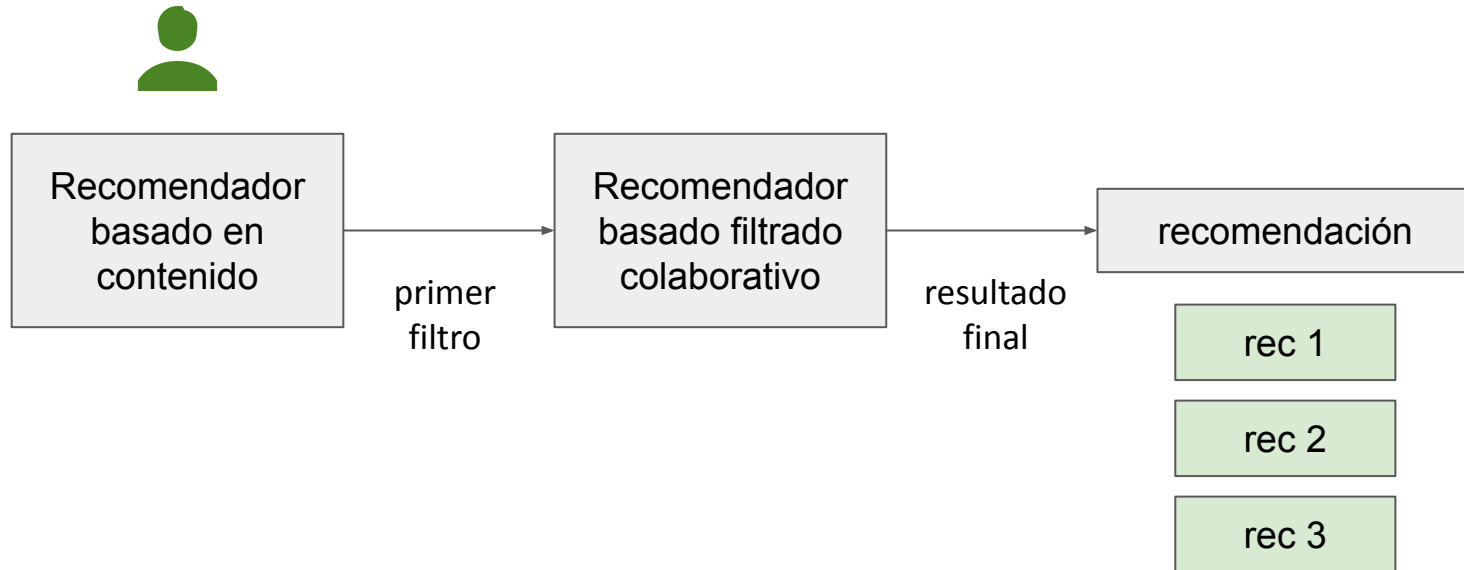


Tipos de recomendación híbrida.

1. **Monolítico:** corre uno a uno varios recomendadores y va filtrando los resultados en un determinado orden.
2. **Mix de recomendadores:** corre varios recomendadores, retorna todos los scores (normalizados) y los ordena de mayor a menor.
3. **Ensamble:** corre varios recomendadores y combina o mezcla su resultado en una sola recomendación.

1. Monolítico

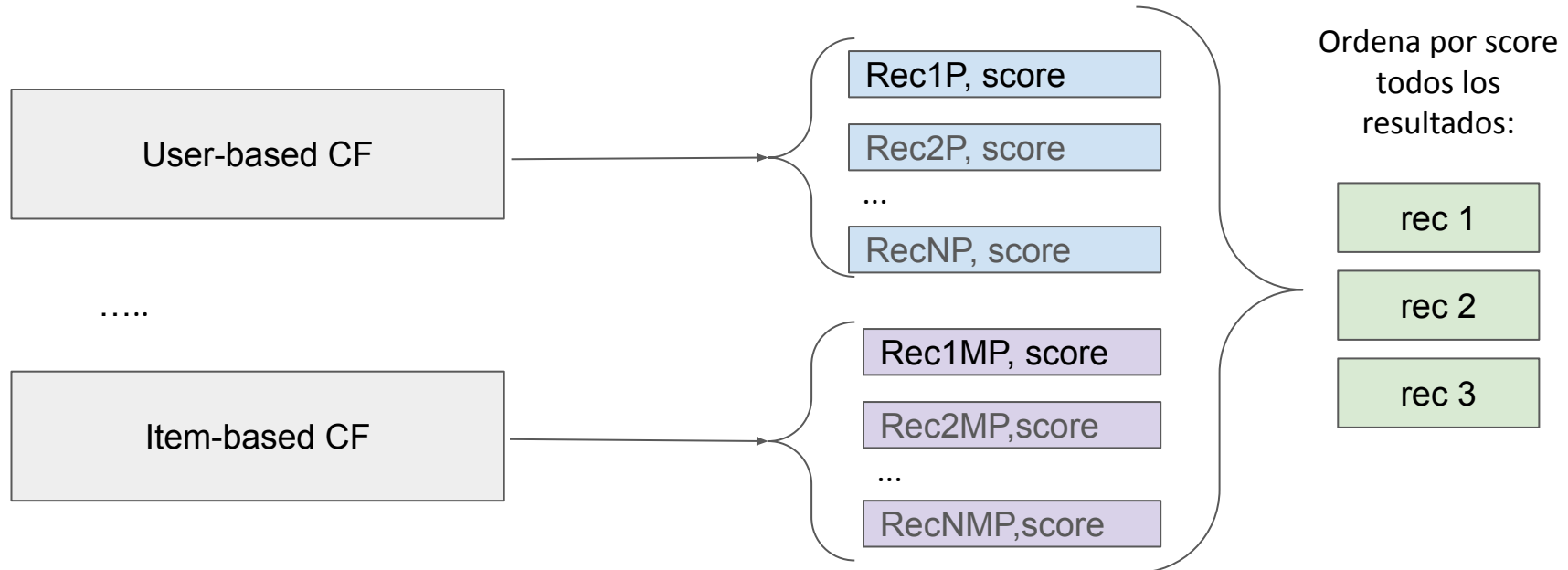
Contiene partes o módulos con diferentes tipos de recomendadores. Por ejemplo podemos usar contenido para encontrar ítems similares content-wise y luego sobre ese subset de ítems aplicar filtrado colaborativo.



2. Mix de recomendadores

Retorna simplemente la UNIÓN de todos los resultados y ordena por score.

Por ejemplo se toman algoritmos de recomendación personalizada, obtiene un score normalizado y retorna resultados ordenados.



3. Ensamble

Ejemplo:

Si tienes dos recomendadores en producción, por ejemplo uno basado en contenido y otro en filtrado colaborativo, ejecuta ambos de manera simultánea y luego combina los resultados.

Collaborative
Filtering?



Content Based?

NOTA:

La principal diferencia entre un ensamble y un recomendador mixto es que el ensamble en el resultado no muestra las predicciones de todos los recomendadores sino que los combina en una sola, en cambio el sistema mixto siempre retorna todo y luego ordena.

Ensamble

Tipos de ensambles:

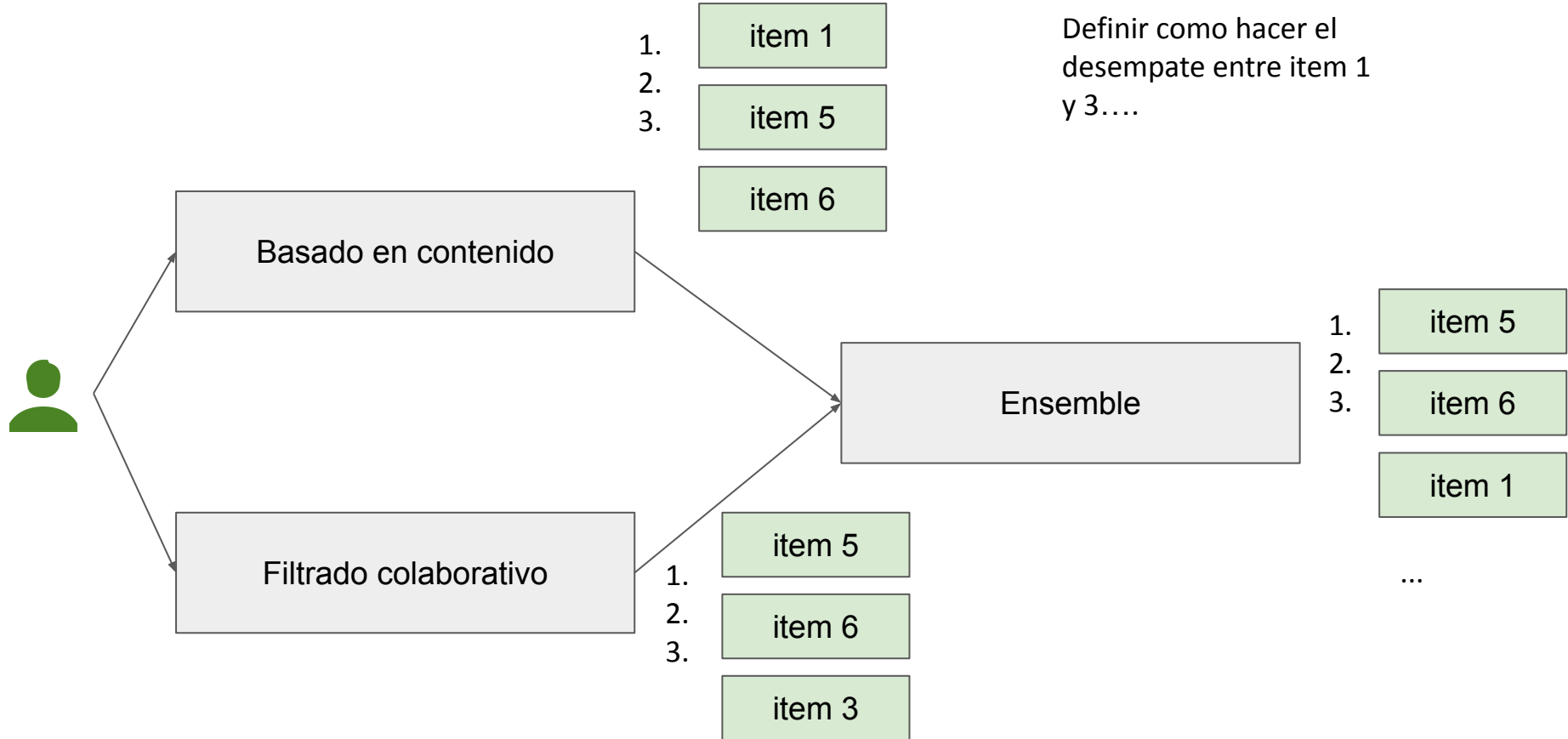
1. Votación entre distintos algoritmos de recomendación.
2. Utilizar heurísticas para decidir qué algoritmo de recomendación utilizar.
3. Pesos (aprendidos o como meta-parámetro) asignado a cada recomendador.
4. Feature ensemble: agrega features adicionales a un algoritmo de recomendación.

Ensamble

Tipos de ensambles:

- 1. Votación entre distintos algoritmos de recomendación.**
2. Utilizar heurísticas para decidir qué algoritmo de recomendación utilizar.
3. Pesos (aprendidos o como meta-parámetro) asignado a cada recomendador.
4. Feature ensemble: agrega features adicionales a un algoritmo de recomendación

Ensamble basado en votación



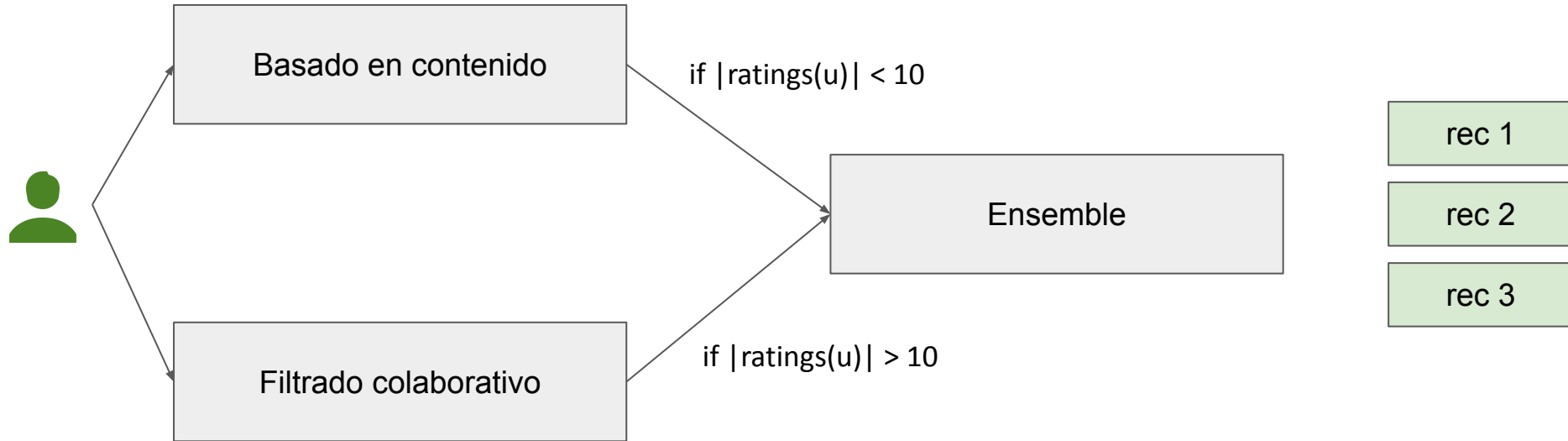
Ensamble

Tipos de ensambles:

1. Votación de rating entre distintos algoritmos de recomendación.
- 2. Utilizar heurísticas para decidir qué algoritmo de recomendación utilizar**
3. Pesos (aprendidos o como meta-parámetro) asignado a cada recomendador.
4. Feature ensemble: agrega features adicionales a un algoritmo de recomendación.

Ensamble basado en heurísticas

Para usuarios que han dado menos de 10 ratings, utilizar basado en contenido, en caso contrario usar filtrado colaborativo.

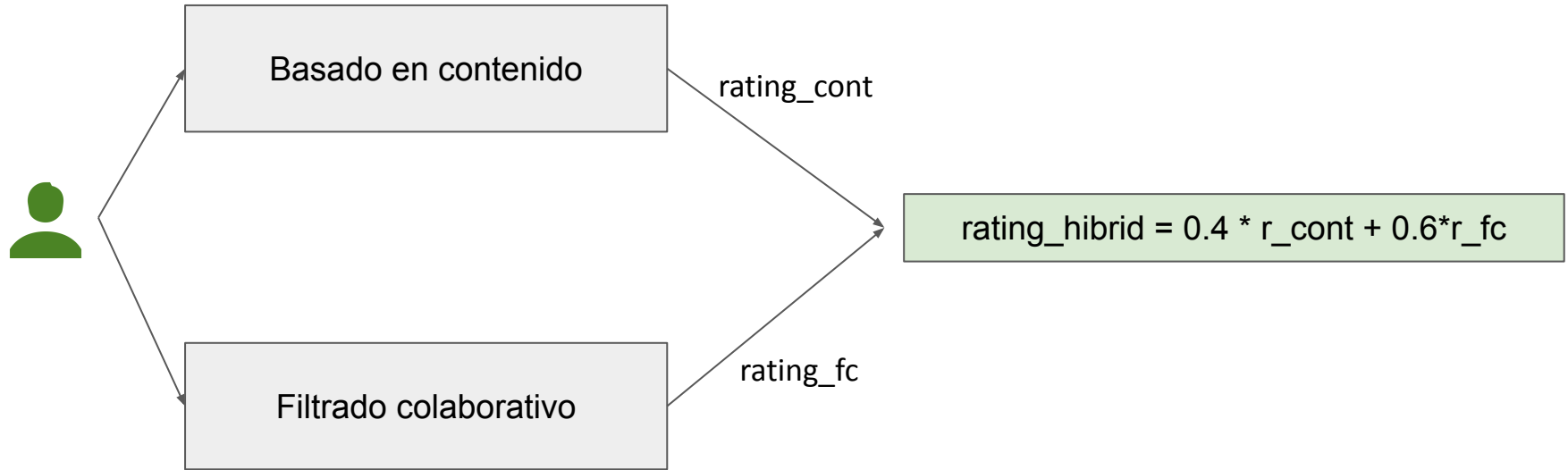


Ensamble

Tipos de ensambles:

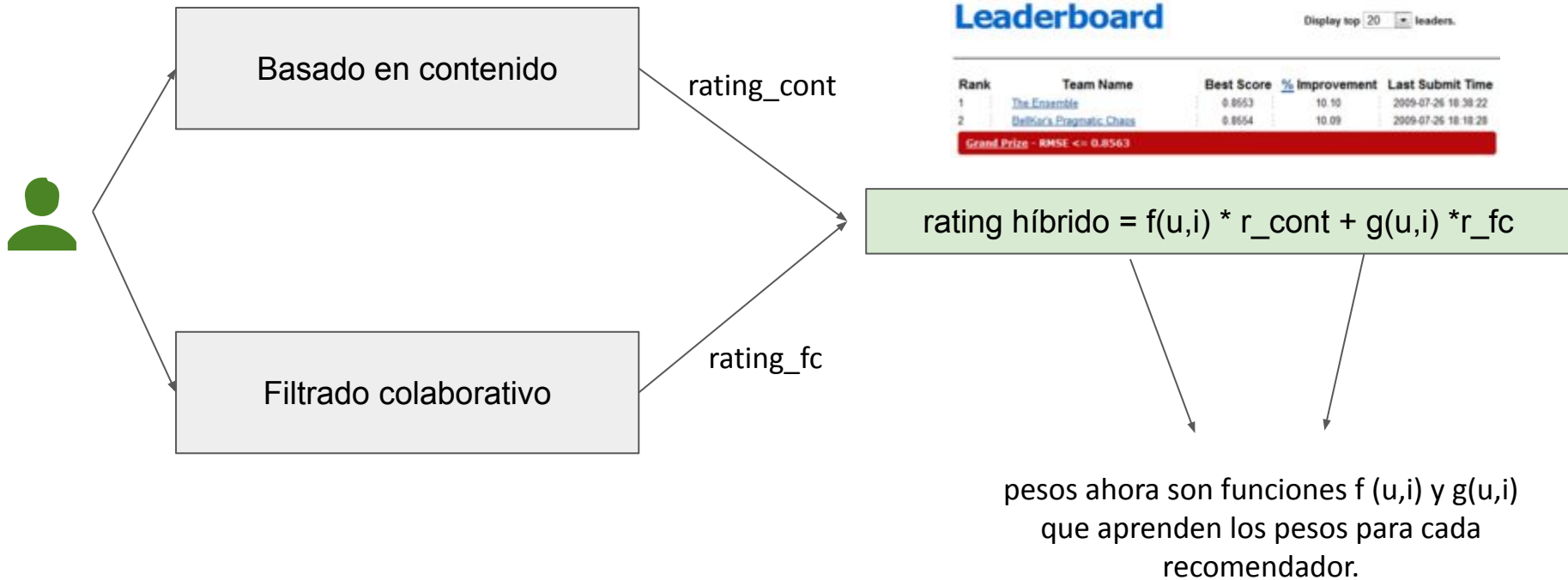
1. Votación de rating entre distintos algoritmos de recomendación.
2. Utilizar heurísticas para decidir qué algoritmo de recomendación utilizar.
3. **Pesos (aprendidos o como meta-parámetro) asignados a cada recomendado**
4. Feature ensemble: agrega features adicionales a un algoritmo de recomendación.

Ensamble basado en pesos



Ensamble basado en pesos entregados por un método de aprendizaje

Netflix Prize uso un ensemble de múltiples métodos (FM y RBM)



Ensamble

Tipos de ensamble:

1. Votación de rating entre distintos algoritmos de recomendación.
2. Utilizar heurísticas para decidir qué algoritmo de recomendación utilizar.
3. Pesos (aprendidos o como meta-parámetro) asignados a cada recomendador.
4. **Feature ensemble: agrega features adicionales a un algoritmo de recomendación.**

Máquinas de factorización

Máquinas de factorización

- Inspiradas en SVM, permiten agregar un número arbitrario de features (user, item, contexto) pero funcionan bien con “sparse data” al incorporar variables latentes factorizadas (inspiradas en Factorización Matricial). No se necesitan vectores de soporte para optimizar el modelo.
- Generalizan diversos métodos de factorización matricial.
- Disminuyen la complejidad de aprendizaje del modelo de predicción respecto de métodos anteriores.

Motivación

- Cada tarea de recomendación (implicit feedback, agregar tiempo, contexto, etc..) requiere rediseño del modelo de optimización y re-implementación del algoritmo de inferencia.
- Se podrían calcular de alguna forma vectores de features.
- Pero para mantener datos tan dispersos como la matriz de usuario ítem, se podrían mantener las factorizaciones.

Ejemplo

- Supongamos los siguientes usuarios, items y transacciones

$$U = \{\text{Alice (A), Bob (B), Charlie (C), } \dots \}$$

$$I = \{\text{Titanic (TI), Notting Hill (NH), Star Wars (SW),} \\ \text{Star Trek (ST), } \dots \}$$

Let the observed data S be:

$$S = \{(\text{A, TI, 2010-1, 5}), (\text{A, NH, 2010-2, 3}), (\text{A, SW, 2010-4, 1}) \\ (\text{B, SW, 2009-5, 4}), (\text{B, ST, 2009-8, 5}), \\ (\text{C, TI, 2009-9, 1}), (\text{C, SW, 2009-12, 5})\}$$

Recomendación tradicional

Example for data:

		Movie				
		TI	NH	SW	ST	...
User	A	5	3	1	?	...
	B	?	?	4	5	...
	C	1	?	5	?	...

Matrix Factorization:

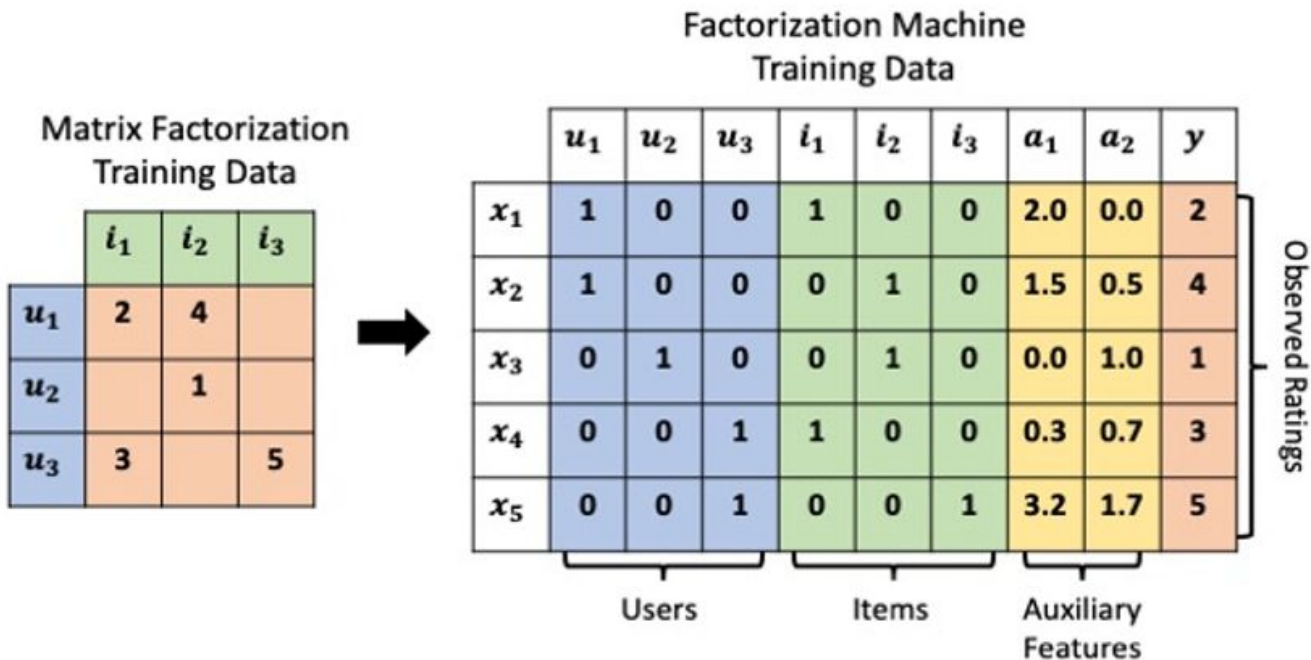
$$\hat{Y} := W H^t, \quad W \in \mathbb{R}^{|U| \times k}, H \in \mathbb{R}^{|I| \times k}$$

$$\hat{y}(u, i) = \hat{y}_{u,i} = \sum_{f=1}^k w_{u,f} h_{i,f} = \langle \mathbf{w}_u, \mathbf{h}_i \rangle$$

k is the rank of the reconstruction.

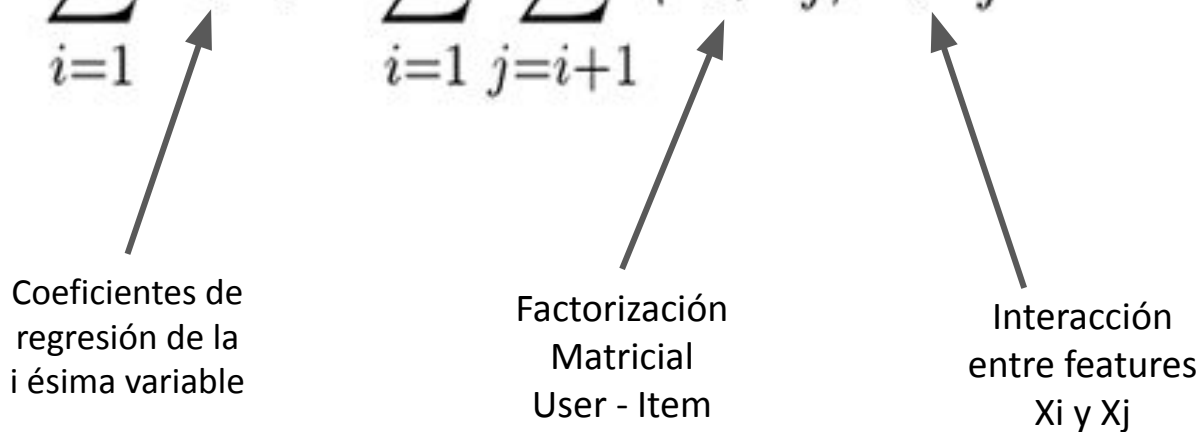
$$\min_{q^*, p^*} \sum_{(u,i) \in K} (r_{ui} - q_i^T \cdot p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$$

Factorization machines



Factorization machines

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$



Coeficientes de
regresión de la
i ésima variable

Factorización
Matricial
User - Item

Interacción
entre features
 X_i y X_j

Ejemplos

- A. Dos variables categóricas
- B. Tres variables categóricas
- C. Dos variables categóricas y tiempo como predictor continuo
- D. Dos variables categóricas y tiempo discretizado en bins

A. Dos variables categóricas

Feature vector \mathbf{x}										
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	
	A	B	C	...	TI	NH	SW	ST	...	
	User				Movie					

- Así, modelo corresponde a MF con biases

$$\hat{y}(\mathbf{x}) = w_0 + w_u + w_i + \underbrace{\langle \mathbf{v}_u, \mathbf{v}_i \rangle}_{\text{MF}}$$

B. Tres variables categóricas

- Predicción de tripletas RDF con FM

Feature vector \mathbf{x}															
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	1	0	0	0	...	
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0	1	0	0	...	
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0	0	0	1	...	
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	1	0	...	
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	1	0	...	
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	1	0	0	0	...	
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0	0	0	1	...	
	S1	S2	S3	...	P1	P2	P3	P4	...	O1	O2	O3	O4	...	
	Subject				Predicate					Object					

- Equivalente a PITF (recomendación de tags)

$$\hat{y}(\mathbf{x}) := w_0 + w_s + w_p + w_o + \langle \mathbf{v}_s, \mathbf{v}_p \rangle + \langle \mathbf{v}_s, \mathbf{v}_o \rangle + \langle \mathbf{v}_p, \mathbf{v}_o \rangle$$

C. Dos variables categóricas y tiempo como predictor continuo

Feature vector \mathbf{x}											
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.2	
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.6	
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.61	
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0.3	
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0.5	
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.1	
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.8	
	A	B	C	...	TI	NH	SW	ST	...		
	User				Movie					Time	

- Modelo corresponde a:

$$\hat{y}(\mathbf{x}) := w_0 + w_i + w_u + t w_{\text{time}} + \langle \mathbf{v}_u, \mathbf{v}_i \rangle + t \langle \mathbf{v}_u, \mathbf{v}_{\text{time}} \rangle + t \langle \mathbf{v}_i, \mathbf{v}_{\text{time}} \rangle$$

D. Dos variables categóricas y tiempo discretizado en bins

Feature vector \mathbf{x}												
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	1	0	0
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0	1	0
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0	1	0
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	1	0	0
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	1	0
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	1	0	0
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0	0	1
	A	B	C	...	T1	NH	SW	ST	...	T1	T2	T3
	User				Movie					Time		

- Modelo corresponde a:

$$\hat{y}(\mathbf{x}) := w_0 + w_i + w_u + w_{b(t)} + \langle \mathbf{v}_u, \mathbf{v}_i \rangle + \langle \mathbf{v}_u, \mathbf{v}_{b(t)} \rangle + \langle \mathbf{v}_i, \mathbf{v}_{b(t)} \rangle$$

Modelos de factorización

- Ventaja:
 - Permiten estimar interacciones entre dos (o más) variables incluso si la interacción no es observada explícitamente.
- Desventajas:
 - Modelos específicos para cada problema
 - Algoritmos de aprendizaje e implementaciones están diseñados para modelos individuales