Sistemas Híbridos de Recomendación

Cristian Cardellino - Luis Biedma

Contenido

- Introducción a sistemas híbridos
- Sistema monolítico
- Sistema mixto
- Sistema de Ensemble

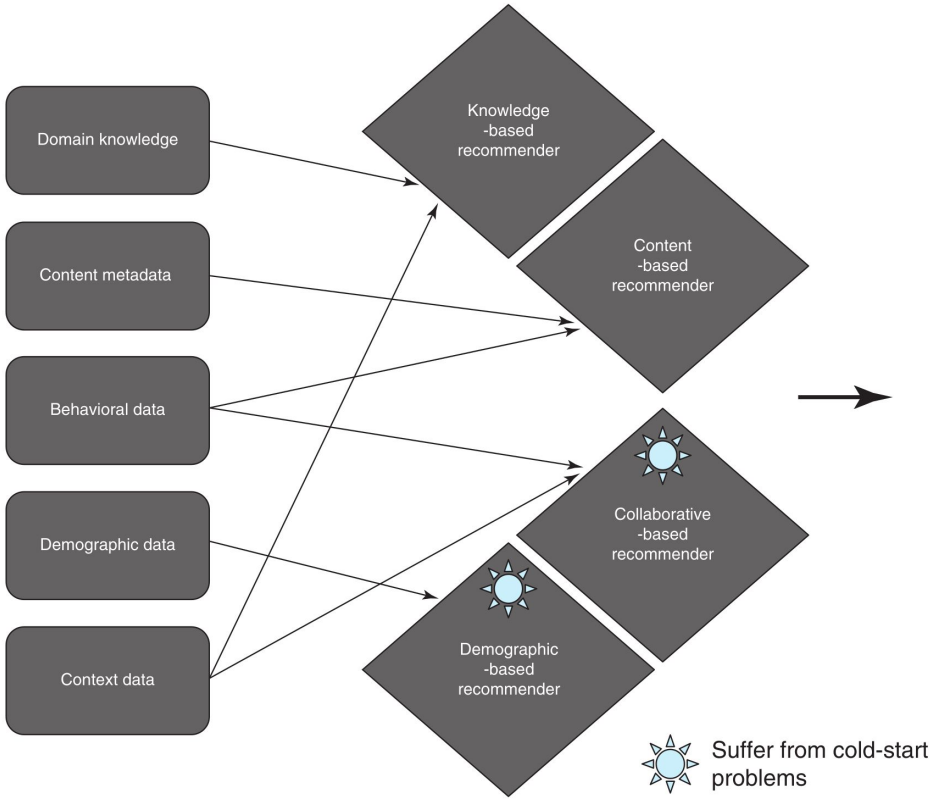
Introducción a Sistemas Híbridos de Recomendación

Sistemas híbridos

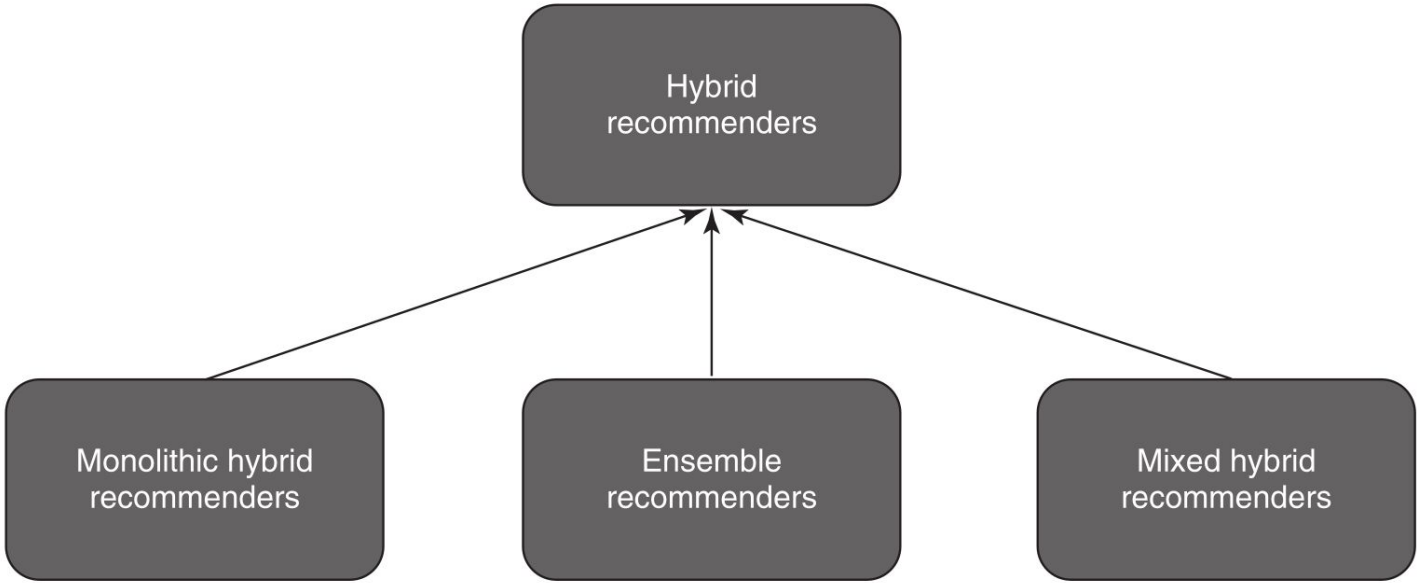
- Buscamos **tomar lo mejor de varios sistemas de recomendación distintos**.
- El objetivo es **mejorar el resultado promedio de cada sistema**.
- Trabajan mejor los **casos extremos** (e.g. Cold Start).
- Las **combinaciones posibles son muchas**.
- **Dependen más de la ingeniería** (e imaginación) que de la herramienta.

Tipos de sistemas

Distintos tipos de sistemas consumen distintos tipos de datos.



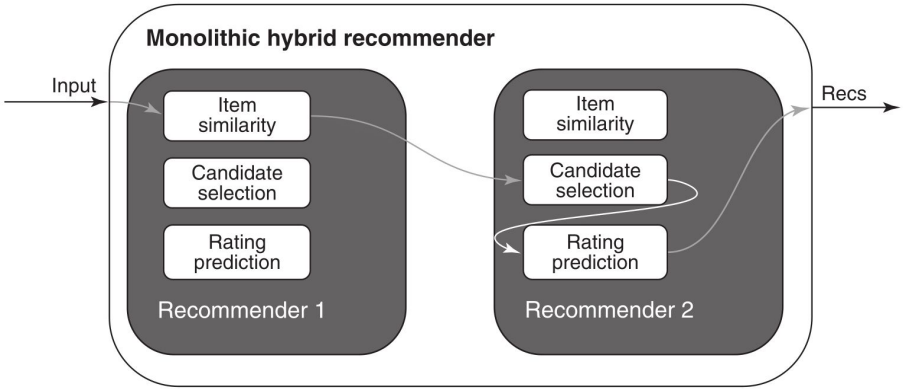
Tipos de sistemas híbridos



Sistema Monolítico

Sistema monolítico

- Es un **Frankenstein de sistemas**.
- Toma **componentes de varios sistemas y los une**.
    - La idea es tomar algo de un sistema de recomendación y unirlo a otro.
- Requieren **trabajo para adaptar recomendadores al sistema**.



Filtrado colaborativo mejorado con contenido

- Un ejemplo de un sistema monolítico es **utilizar la información de contenido** para generar nuevas relaciones entre usuarios e items.
- La idea parte del problema de **usuarios puntuando items disjuntos**.
- A partir de los vectores de contenido de los items se calculan ratings aproximados.
- Se evalúa el filtrado colaborativo sobre los nuevos valores.

Ejemplo de aplicación de modelo híbrido

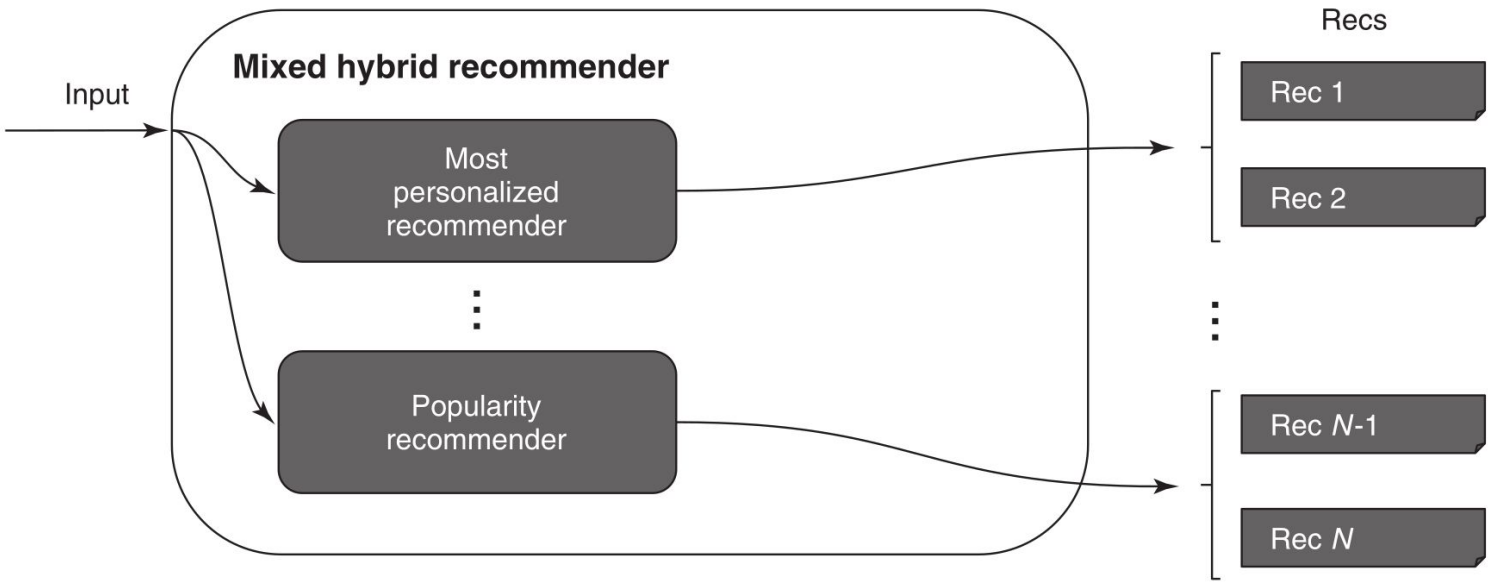
	Sci-fi 1	Sci-fi 2	Sci-fi 3	Sci-fi 4
User 1	4	4		
User 2	5	4		
User 3			2	4

Sistema mixto

Sistema mixto de recomendación

- El sistema mixto **devuelve una unión de recomendaciones** de otros sistemas.
- Se puede pensar en base a **escalas de personalización**.
    - Los sistemas se ordenan de acuerdo a que tan personalizados sean.
    - A medida que se sube en la escala las recomendaciones son menos personalizadas.
    - Más personalización equivale a menos recomendaciones.
    - La última escala utiliza recomendaciones no personalizadas (e.g. popularidad)
- Si se tienen varios sistemas similares se devuelve una lista de recomendaciones por puntuación.
    - La puntuación entre distintos sistemas tiene que estar normalizada.

Escala de personalización

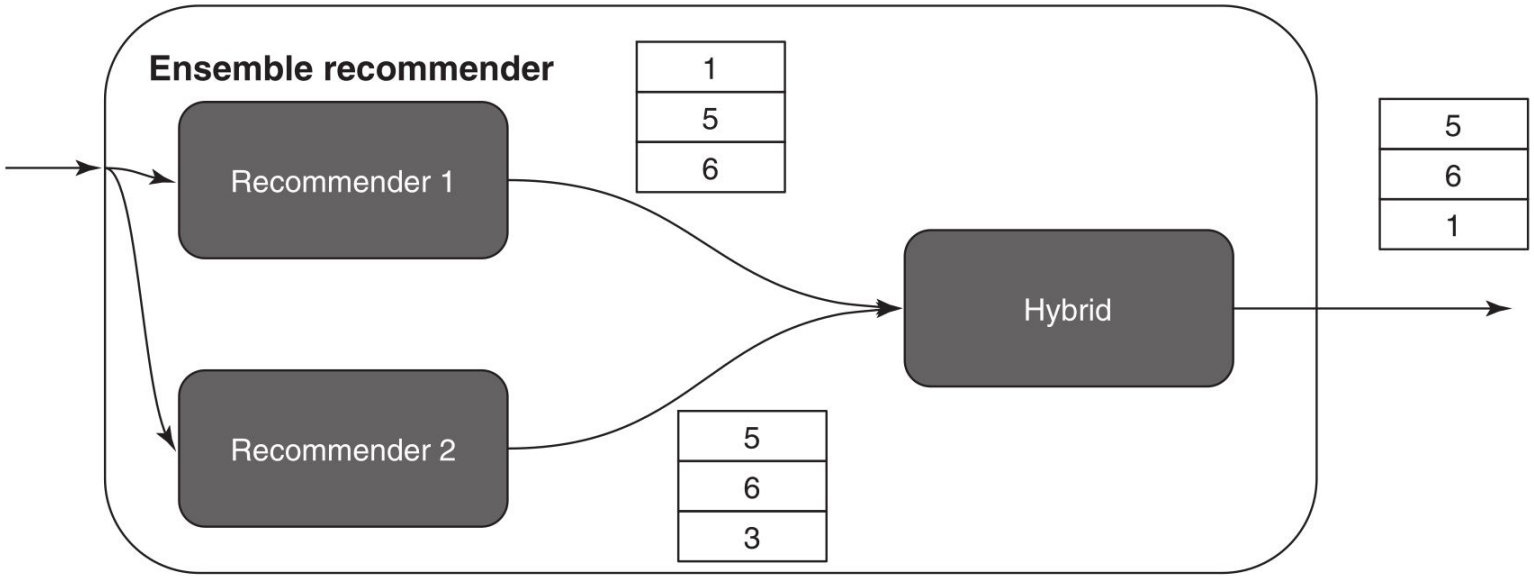


Sistema de Ensemble

Sistema de Ensemble

- Se basa en la **idea de aprendizaje automático con ensemble**.
- Utiliza varios sistemas y **combina sus predicciones**.
- Los sistemas **se utilizan en simultáneo y devuelven una sola predicción**.
    - Esto los diferencia de sistemas mixtos que combinan rankings de predicciones.
- Existen **varias maneras de combinar**.
    - E.g. sistema de votación donde se le da más presencia a aquellas recomendaciones compartidas

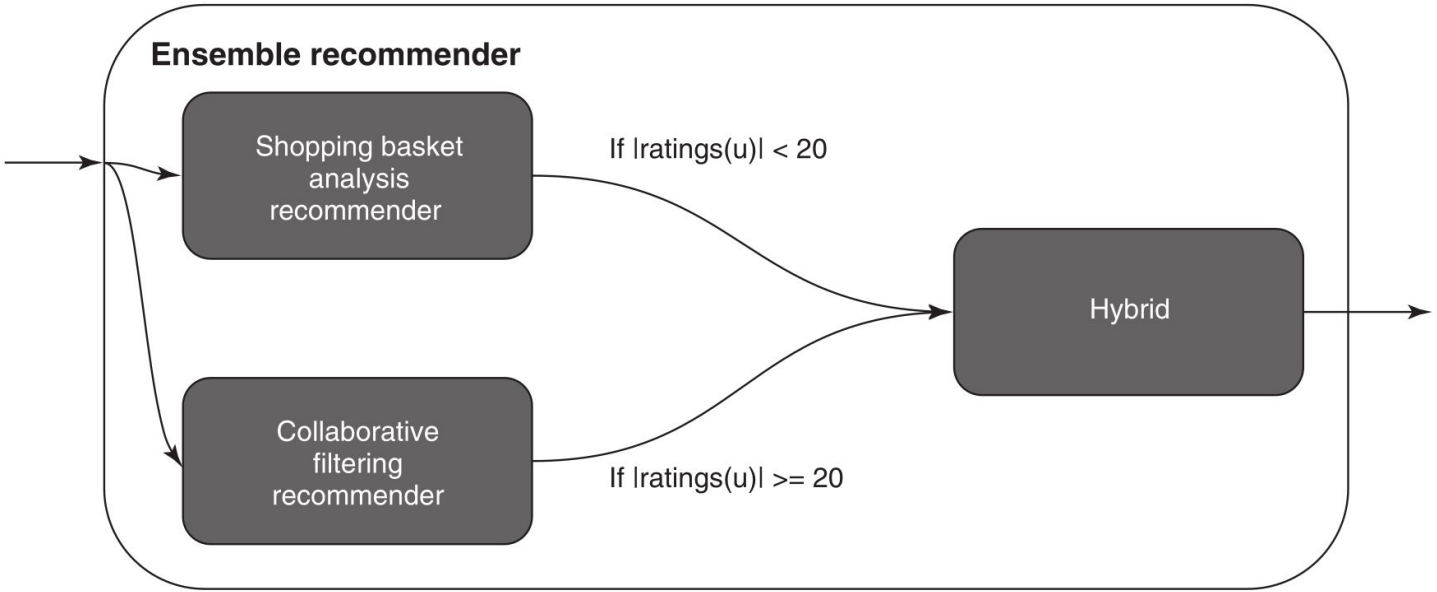
Ejemplo de ensemble



Ensemble por Contexto (Switching Ensemble)

- Dados dos o más sistemas el **ensemble decidirá cuál usar**.
- La **decisión se hace en base al contexto**. Pueden ser varias.
    - Geográfica: basado en el país/ubicación (e.g. Trending Topics en Twitter).
    - Temporal: basado en la hora del día (e.g. Listas de Spotify).
    - Categórico: basado en categorías del contenido (e.g. Recomendaciones en un sistema de noticias de acuerdo a la sección)
    - De Usuario: basado en el conocimiento del usuario (e.g. Un usuario que puntuó más de 20 películas vs uno que puntuó menos).

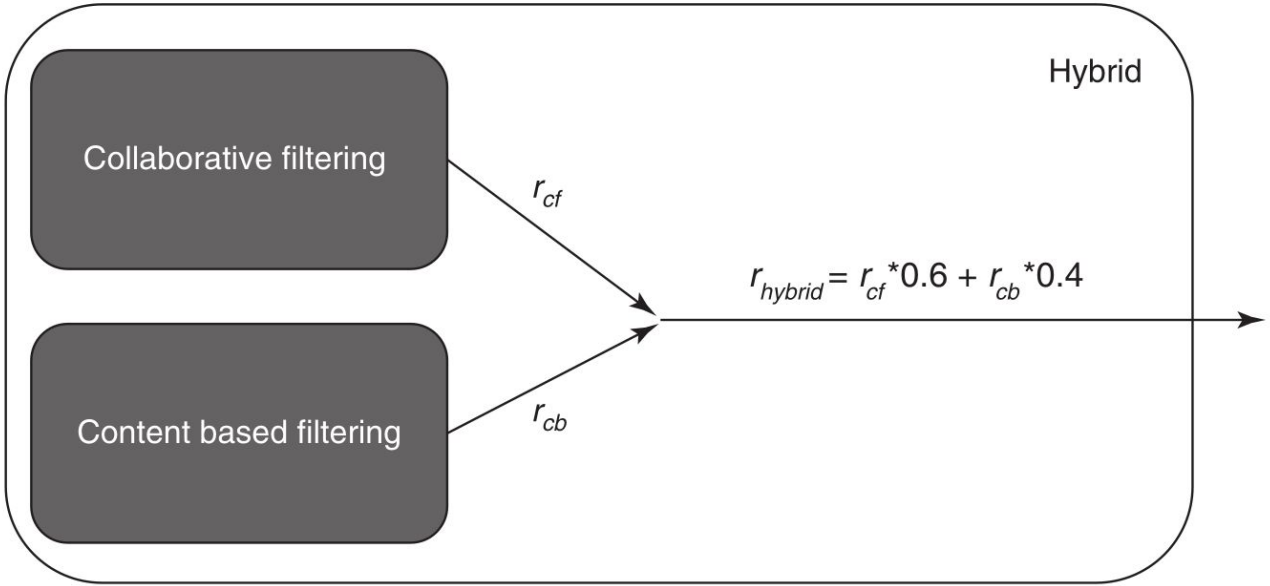
Ejemplo: Ensemble De Usuario



Ensemble Ponderado (Weighted Ensemble)

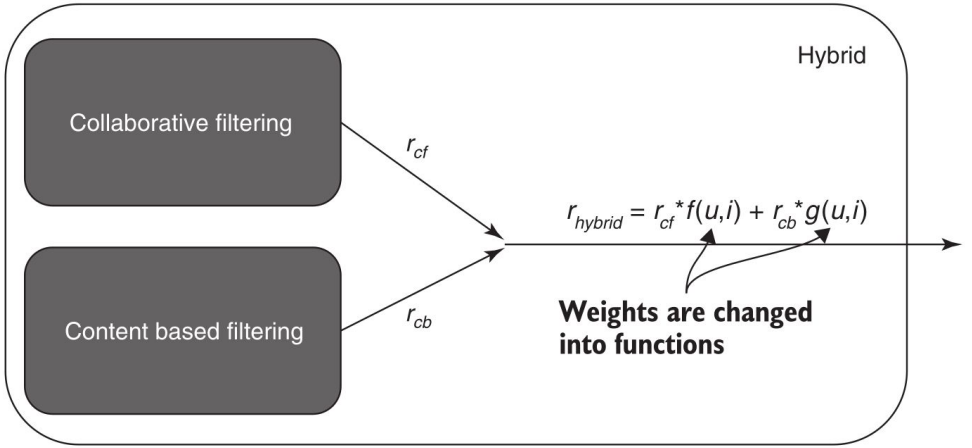
- La puntuación de la recomendación **se calcula en base a otras recomendaciones**.
- Distintos **sistemas de recomendación tiene cierto peso** asociado en la puntuación final.
- Estos **sistemas son “meta-atributos” de una operación** que los engloba.
    - Se los llama “atributos recomendadores” (feature recommenders).
- El **peso de los “meta-atributos” se debe calcular**.
    - Regresión lineal.
    - Evaluación A/B con distintos pesos.

Ejemplo: Ensemble Ponderado



Feature Weighted Linear Stacking

- Se basa en la idea de entrenar una **regresión lineal con los meta-atributos**.
- Los pesos son **meta-funciones**.
  - Se pueden sacar a partir del conocimiento de dominio.
  - Muchas veces pueden ser en base a ingeniería de atributos.
  - El Netflix Prize winner consta de 25 meta-funciones.

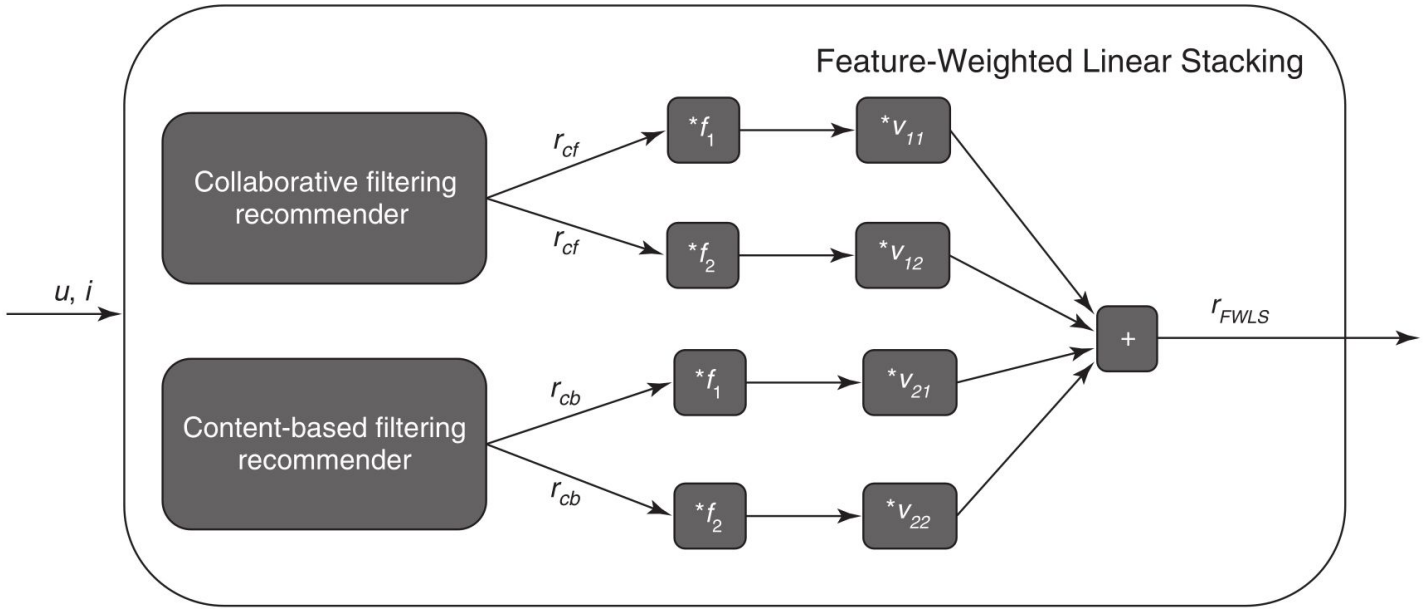


Funciones hipótesis y objetivo

$$\mathrm{r}_{FWLS}(i,u) = \sum_{j=1}^{L} \sum_{k=1}^{M} \theta_{kj} f_k(u,i) r_j(u,i) \quad \mathrm{L}_{FWLS}(\theta) = \sum_{i \in items} \sum_{u \in users} (\mathrm{r}_{FWLS}(i,u) - r_{u,i})^2$$

- θ = Parámetros a entrenar
- *f* = Meta-funciones
- r_j = Recomendador (entrenado)
- $r_{u,i}$ = Rating del usuario *u* al item *i*

Ejemplo: Ensemble FWLS



Entrenamiento de FWLS

- Dividir los datos (entrenamiento, validación, evaluación).
- Entrenar los recomendadores base.
- Generar los ratings para los datos de entrenamiento.
- Ejecutar las meta-funciones sobre los datos de entrenamiento.
- Calcular el producto entre cada rating y cada meta-función.
- Se usa regresión lineal para calcular los pesos.
- Se utilizan los datos de validación/evaluación para mejorar el algoritmo.

Entrenamiento de FWLS