

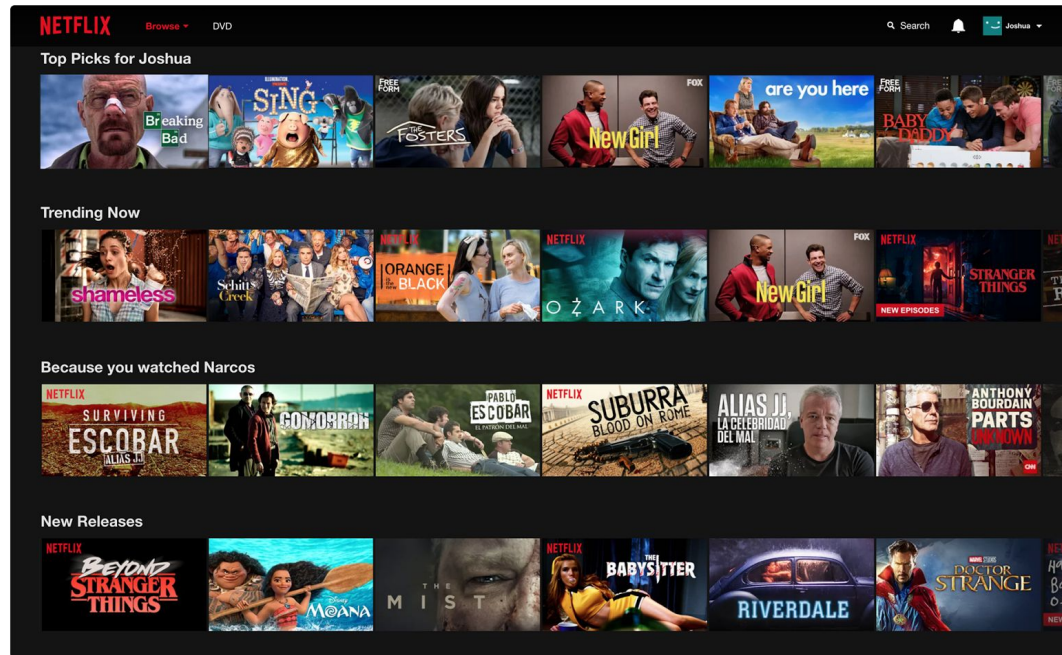


Implementación de Mínimos Cuadrados Alternados con Regularización para un Sistema de Recomendación

144089 Laura López Santibañez Jácome
183340 Dante Ruiz Martínez

Netflix

- La misión de Netflix es conectar a las personas con las películas que aman.
- Para ayudar a los clientes a encontrar esas películas, Netflix ha desarrollado un sistema de recomendación de películas de clase mundial: CinematchSM.



Sistemas de Recomendación

- El propósito de un sistema de recomendación es sugerir productos que en caso de consumirse, maximizan la utilidad del comprador y los beneficios del vendedor en un determinado momento en el tiempo.
- Los sistemas analizan y procesan la información histórica de los usuarios, como por ejemplo, compras previas, calificaciones o visitas y la usan para predecir qué producto puede ser interesante para el consumidor.
- **CinematchSM de Netflix:** Su trabajo es predecir si alguien disfrutará de una película en función de cuánto les haya gustado o no a otras personas con gustos similares.

2
Tipos



Filtros colaborativos

Filtros basados en
contenido

Objetivo

Programar de manera local en R el algoritmo de **mínimos cuadrados alternados con regularización** que se ha utilizado para ajustar sistemas de recomendación, utilizando los datos de la competencia de Netflix. Asimismo, comparar su desempeño con otras implementaciones disponibles.

Mínimos Cuadrados Alternados

- Filtro colaborativo que matemáticamente es un problema de factorización de matrices.
- El propósito de este algoritmo es resolver un problema de optimización para encontrar las matrices U y M que multiplicadas puedan aproximar la matriz de datos original R .

$$R \approx UM.$$

3 pasos:

1. Inicializar matrices M y U .

2. Fijar M y resolver para U : $\min_{U_1} ||R - U_1 M_1^t||_{obs}^2$

3. Fijar U y resolver para M : $\min_{M_2} ||R - U_1 M_2^t||_{obs}^2$

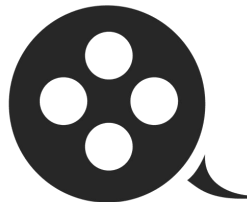
★ 2 y 3 se repiten hasta un criterio de paro

Datos

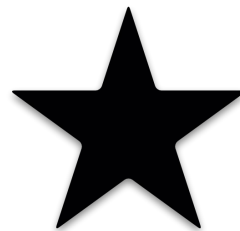
Tomamos una muestra de los datos de la competencia para poder correr el algoritmo de manera local, no distribuida y no paralelizada



129



769



99,201

En promedio, los usuarios han calificado 1.18% de todas las películas.

Mínimos Cuadrados Alternados con Regularización (Zhou et al. 2008)

- Para lidiar con los datos malos se utiliza **regularización tipo Ridge**.
- El problema de optimización se reescribe a:

$$\min_{U, V} f(U, M) = \sum_{(i,j) \text{ obs}} w_{i,j} (r_{ij} - u_i^t m_j)^2 + \lambda \left(\sum_i n_{u_i} \|u_i\|^2 + \sum_j n_{m_j} \|m_j\|^2 \right)$$

Con esta nueva función, la optimización queda:

- **Para una M fija, derivando con respecto a U:** $(M_{I_i} M_{I_i}^T + \lambda n_{u_i} E) \mathbf{u}_i = M_{I_i} R^T(i, I_i), \forall i$
 $\Rightarrow \mathbf{u}_i = A_i^{-1} V_i, \forall i$
- **Para una U fija, derivando con respecto a M:** $\mathbf{m}_j = A_j^{-1} V_j, \forall j$
donde $A = U_{I_j} U_{I_j}^t + \lambda n_{m_j} E$, $V_i = U_{I_j} R(I_j, j)$

Algoritmo implementado

1. Inicializar matrices M con la primera fila siendo el promedio de calificación de la película y el resto de la matriz números chicos y U con números chicos.

2. Fijar M y resolver para cada renglón de U: $\mathbf{u}_i = A_i^{-1} V_i, \forall i$
donde $A = M_{I_i} M_{I_i}^t + \lambda n_{u_i} E$, $V_i = M_{I_i} R^T(i, I_i)$

3. Fijar U y resolver para cada columna de M: $\mathbf{m}_j = A_j^{-1} V_j, \forall j$
donde $A = U_{I_j} U_{I_j}^t + \lambda n_{m_j} E$, $V_i = U_{I_j} R(I_j, j)$

★ 2 y 3 se repiten hasta alcanzar el número de iteraciones

Resultados

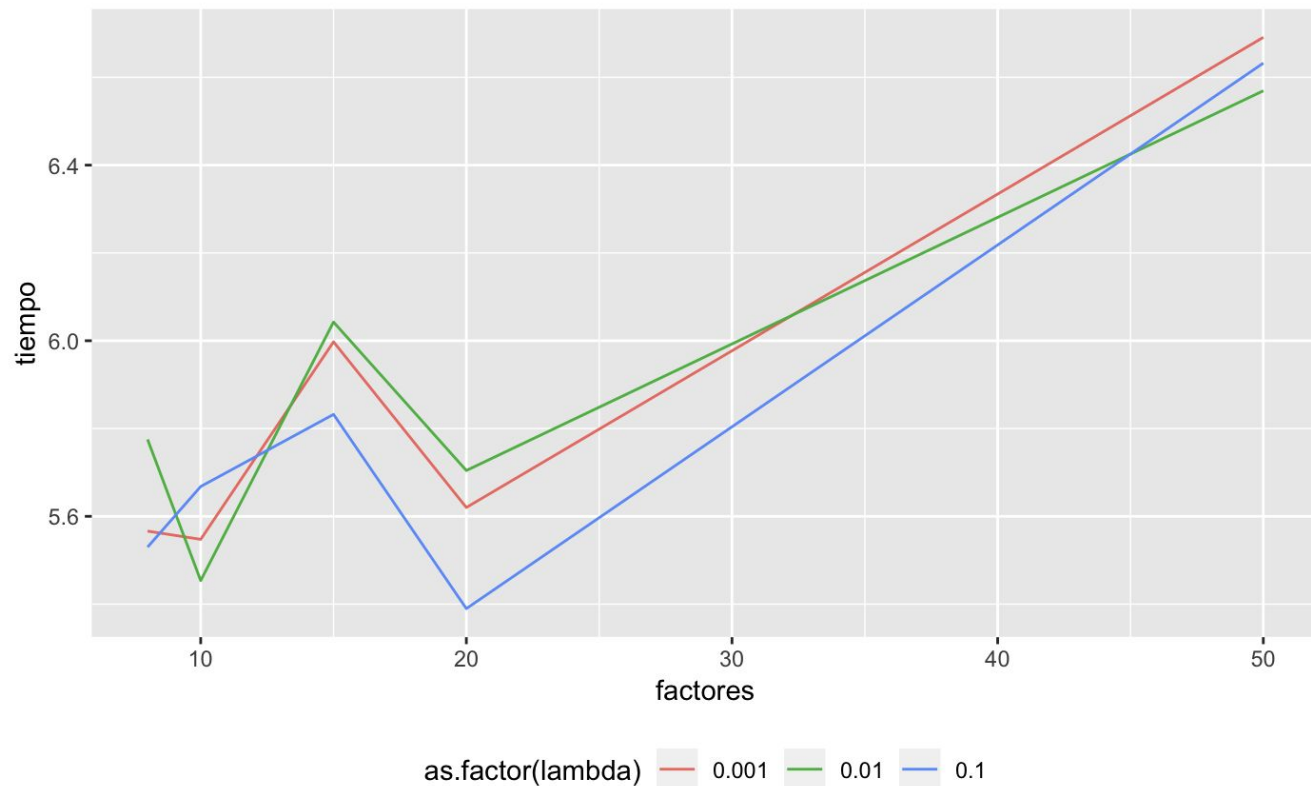
Table 2: Recomendaciones para el usuario id 10

nombre	rating
Gladiator	72
Black Sheep	72
Resident Evil	72
The Devil's Advocate	72
Look Who's Talking	78
Keeping the Faith	81
Mallrats	90
Star Trek: Deep Space Nine: Season 2	90
The Matrix	90
The Best of Friends: Vol. 2	97

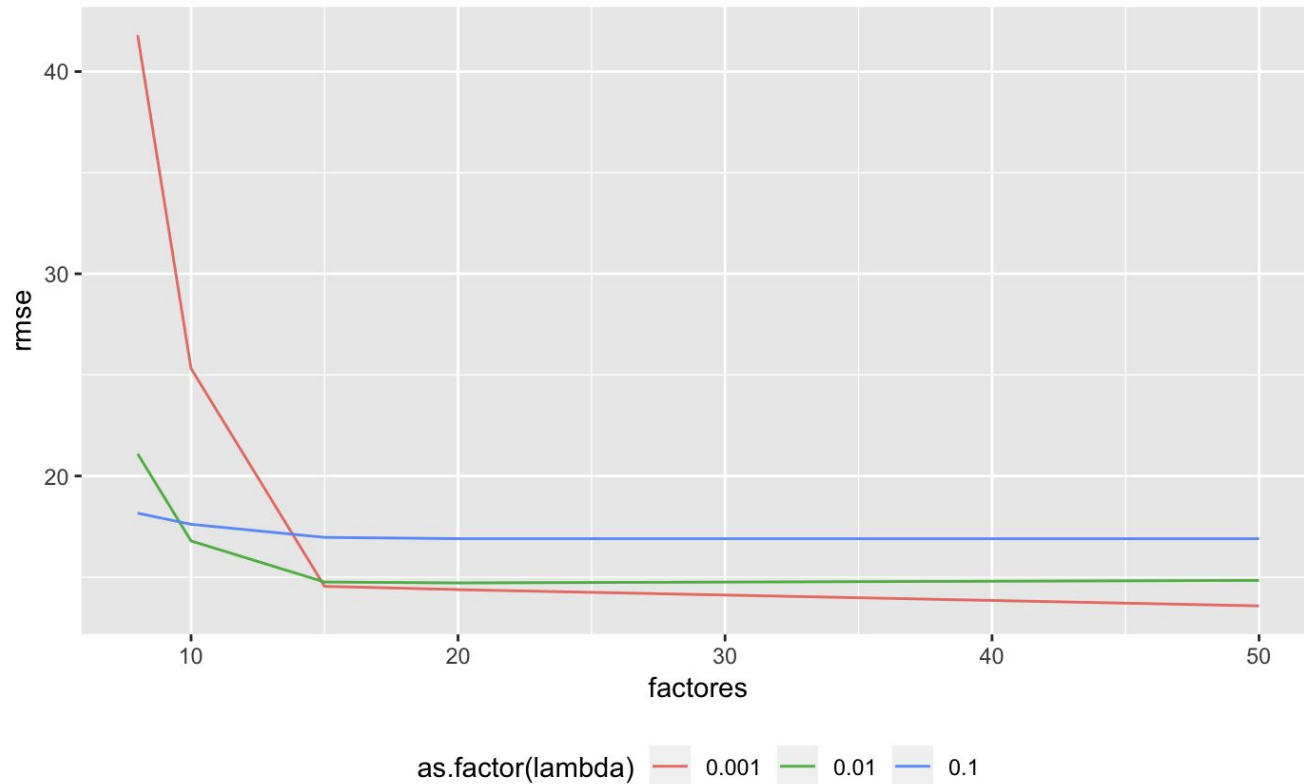
Table 3: Recomendaciones para el usuario id 8

nombre	rating
Clerks: Uncensored	10
Mallrats	10
Alien 3: Collector's Edition	10
Resident Evil	10
The Devil's Advocate	10
Kill Bill: Vol. 2	11
GoldenEye	11
Pirates of the Caribbean: The Curse of the Black Pearl	12
Gladiator	12
The Matrix	12

Evaluación del algoritmo



Evaluación del algoritmo



Comparación de modelos

modelo	lambda	factores	tiempo	costo1	costo2	rmse
1	0.100	20	5.389794	57398.279	57398.271	16.90128
2	0.010	20	5.704400	7452.808	7450.291	14.71185
3	0.001	20	5.620160	3230.613	3228.805	14.38531
4	0.100	50	6.632453	57387.204	57387.198	16.89945
5	0.010	50	6.569537	7458.932	7456.271	14.84034
6	0.001	50	6.691347	2989.641	2987.709	13.57805
7	0.100	10	5.667921	65950.974	65950.960	17.61210
8	0.010	10	5.453610	14239.891	14235.059	16.78945
9	0.001	10	5.547596	8477.014	8472.895	25.31401
10	0.100	8	5.529866	85314.863	85314.847	18.16429
11	0.010	8	5.775080	33107.149	33100.874	21.09796
12	0.001	8	5.566361	29423.222	29419.508	41.79916
13	0.100	15	5.832239	57743.315	57743.296	16.96545
14	0.010	15	6.042544	7630.530	7627.904	14.75821
15	0.001	15	5.997478	3739.275	3737.059	14.54376

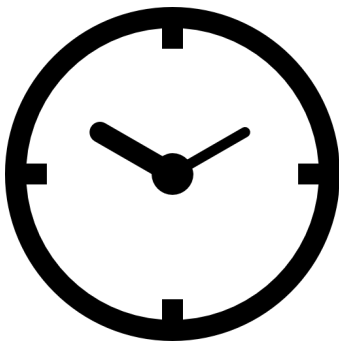
Comparación con cómputo en paralelo

lambda = 0.001

factores = 50

iteraciones 500

5.52



minutos

2.28

LOCAL

SPARK

Gracias!

Anexos

