

# Estudio y aplicación de algoritmos y estructuras de datos a los Sistemas de Recomendación

Pablo Sánchez Pérez

Universidad Autónoma de Madrid

*pablo.sanchezp@estudiante.uam.es*

8 de Junio 2016

- 1 Introducción
- 2 Estado del arte
- 3 Sistema desarrollado
- 4 Resultados
- 5 Conclusiones y trabajo futuro

- 1 Introducción
- 2 Estado del arte
- 3 Sistema desarrollado
- 4 Resultados
- 5 Conclusiones y trabajo futuro

¿Qué es un sistema de recomendación?

- Sistemas que sugieren artículos a los usuarios.
- Tienen el potencial para mejorar la calidad de decisiones de los usuarios.
- Algunos emplean una gran cantidad de datos.
- Importancia de almacenar los datos de manera eficiente.
- Existen de diversos tipos.

- 1 Introducción
- 2 Estado del arte**
- 3 Sistema desarrollado
- 4 Resultados
- 5 Conclusiones y trabajo futuro

# Sistemas de recomendación. Recommender systems handbook

- Basados en contenido.
- Sistemas de filtrado colaborativo.
- Demográficos.
- Basados en conocimiento.
- Basados en comunidad.
- Híbridos.

- Pilas.
- Colas.
- Listas.
- Árboles.
  - Árboles binarios de búsqueda.
  - Árboles rojo-negros.
  - Árboles estadísticos.
  - Árboles B.
- Heaps.
- Colas de prioridad.

- Algoritmos de ordenación.
- Algoritmos de grafos.
- LCS (Longest common subsequence).
- Multiplicación de matrices.

- ¿Cómo medir la similitud?
  - Similitud coseno y Correlación de Pearson.
  - Basado en artículos o en usuarios.

$$\text{Cosine Vector } (u, v) = \cos(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_u} r_{ui}^2 \sum_{j \in \mathcal{I}_v} r_{vj}^2}} \quad (1)$$

Figura: Similitud Coseno

$$\text{PC } (u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \bar{r}_v)^2}} \quad (2)$$

Figura: Correlación de Pearson

# Ejemplo de las medidas

U/I	I1	I2	I3	I4	I5
U1	3		5	3	
U2	3		5		2
U3	5	4	2	2	
U4	4	2	4	3	3

Tabla: Usuarios de ejemplo.

$$\cos(U1, U4) = \frac{3 \cdot 4 + 5 \cdot 4 + 3 \cdot 3}{\sqrt{(3^2 + 5^2 + 3^2) \cdot (4^2 + 2^2 + 4^2 + 4^2 + 3^2)}} = 0.85 \quad (3)$$

$$PC(U1, U4) = \frac{(3 - 3.67) \cdot (4 - 3.2) + (5 - 3.67) \cdot (4 - 3.2) + (3 - 3.67) \cdot (3 - 3.2)}{\sqrt{((3 - 3.67)^2 + (5 - 3.67)^2 + (3 - 3.67)^2) \cdot ((4 - 3.2)^2 + (4 - 3.2)^2 + (3 - 3.2)^2)}} = 0.355 \quad (4)$$

# ¿Únicas medidas de similitud?

## Probaremos el rendimiento de LCS (Longest Common Subsequence)

Dadas dos cadenas X e Y, obtener la cadena más larga tal que los caracteres de dicha cadena aparezcan en ese mismo orden en X e Y.

$$C[i, j] = \begin{cases} 0 & \text{si } i=0 \text{ o } j=0 \\ C[i-1, j-1] + 1 & \text{si } i, j > 0 \text{ y } X_i = Y_j \\ \max(C[i, j-1], C[i-1, j]) & \text{si } i, j > 0 \text{ y } X_i \neq Y_j \end{cases} \quad (5)$$

Utilizada para comparar similitudes entre dos cadenas de ADN.

# Transformación de datos empleada para LCS

- Algo novedoso en el área (hasta donde hemos encontrado).
- Necesaria una transformación de los datos.

U1: I1(3), I2(4), I3(4), I4(5)  $\rightarrow$  103,204,304,405

U2: I1(3), I2(5), I4(5)  $\rightarrow$  103,205,405

	y	103	204	304	405
x	0	0	0	0	0
103	0	1	1	1	1
205	0	1	1	1	1
405	0	1	1	1	2

Tabla: LCS sobre datos de recomendación

- Listas enlazadas.
- Tablas Hash.
- Árboles.
  - Árboles binarios de búsqueda.
  - Árboles rojo-negros.
  - Árboles estadísticos.
  - Árboles B.
- Heaps (Max-Min Heaps).
  - Colas de prioridad.

## ¿Cómo compararlas?

Comparaciones de memoria y tiempo cuando se utilizan dentro de un sistema de recomendación

# Predicción de rating y cálculo de error

Cuanta mayor similitud, mayor peso.

$$\hat{r}_{ui} = \frac{\sum_{v \in \mathcal{N}_i(u)} r_{vi} w_{uv}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \quad (6)$$

También se pueden normalizar

$$\text{Mean centering: } \hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in \mathcal{N}_i(u)} (r_{vi} - \bar{r}_v) w_{uv}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \quad (7)$$

## ¿Cómo medir el error?

$$\text{Mean Absolute Error MAE} = \frac{1}{|\mathcal{R}_{\text{test}}|} \sum_{r_{ui} \in \mathcal{R}_{\text{test}}} |f(u, i) - r_{ui}| \quad (8)$$

$$\text{Root Mean Squared Error RMSE} = \sqrt{\frac{1}{|\mathcal{R}_{\text{test}}|} \sum_{r_{ui} \in \mathcal{R}_{\text{test}}} (f(u, i) - r_{ui})^2} \quad (9)$$

# Generación de ranking y evaluación

Cuanta mayor similitud, mayor peso.

$$\hat{r}_{ui} = \frac{\sum_{v \in \mathcal{N}_i(u)} r_{vi} w_{uv}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \quad (10)$$

También se pueden normalizar

$$\text{Mean centering: } \hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in \mathcal{N}_i(u)} (r_{vi} - \bar{r}_v) w_{uv}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \quad (11)$$

Se emplean otras métricas

$$\text{NDCG}@p = \frac{\text{DCG}@p}{\text{IDCG}@p} \quad (12)$$

$$\text{DCG}@p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2(i)} \quad (13)$$

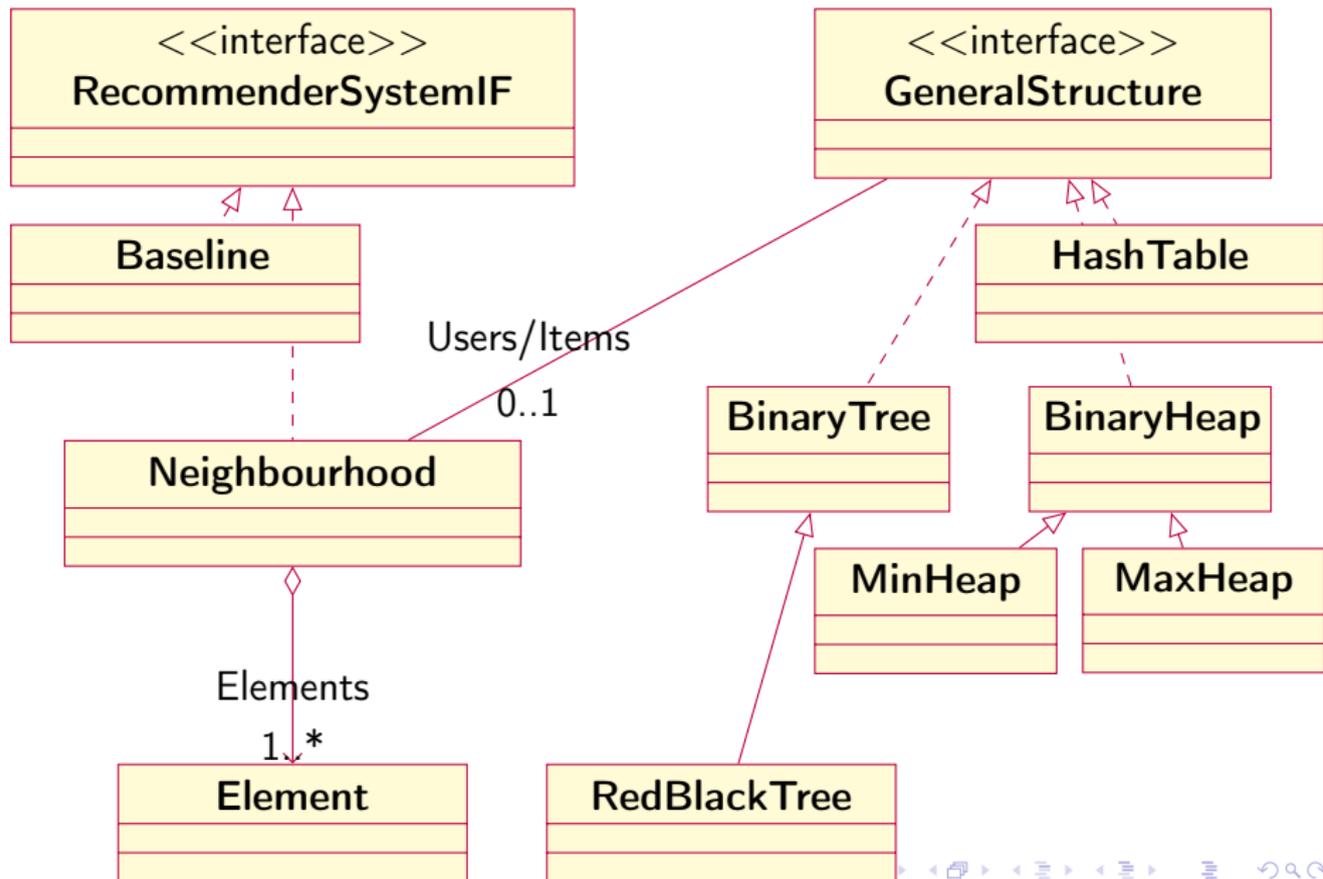
- 1 Introducción
- 2 Estado del arte
- 3 Sistema desarrollado**
- 4 Resultados
- 5 Conclusiones y trabajo futuro

# Características del sistema I

- Sistema de recomendación basado en vecinos.
- Funcionamiento con varias estructuras de datos.
- Medidas de similitud implementadas:
  - Similitud coseno.
  - Correlación de Pearson.
  - LCS.
  - Variantes.
- Predicción de rating y obtención de listas de recomendaciones (ranking).
- Implementación de recomendadores base.

- Vecinos no iguales (en función del peso).
- Filtrado de similitudes.
- Soporte para mean-centering.
- Configuración de parámetros para algunas estructuras de datos (Tablas Hash y Árbol B).
- Lazy-instantiation.
- Posibilidad de emplear un valor por defecto en caso de no encontrar vecinos.

# Diagrama simplificado del sistema



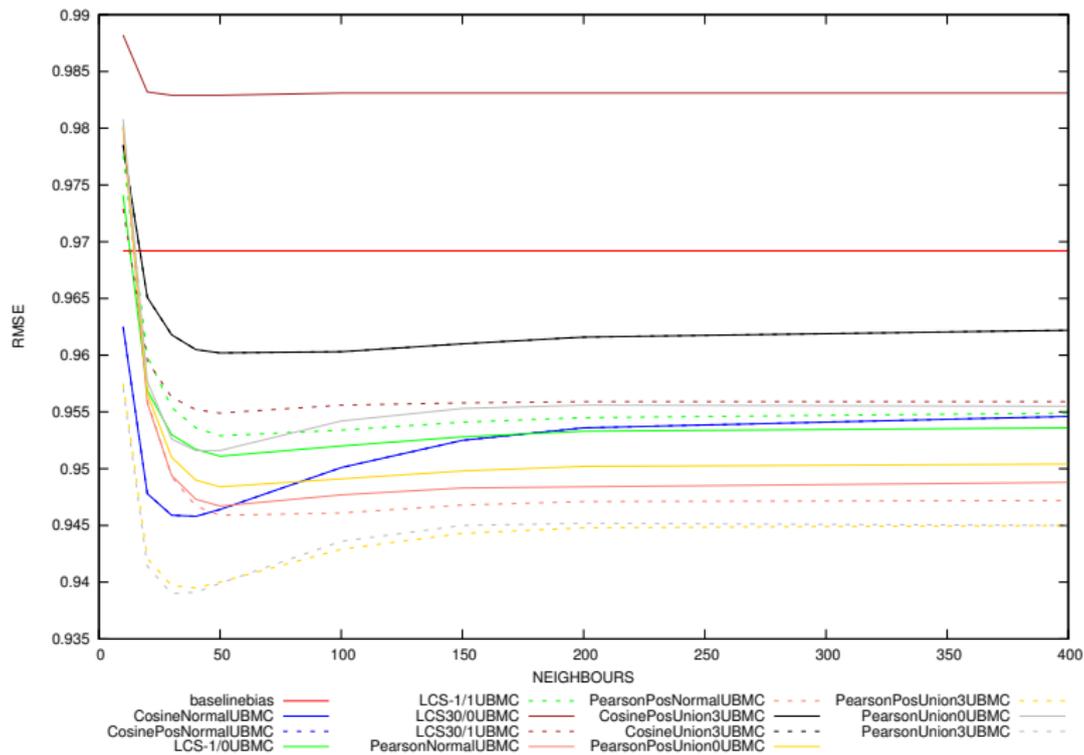
- 1 Introducción
- 2 Estado del arte
- 3 Sistema desarrollado
- 4 Resultados**
- 5 Conclusiones y trabajo futuro

# Resultados memoria.

Structure	Load training memory	Load training time	Training memory	Training time
Binary search tree	61,56	1,82	191,99	64,27
Red-black tree	61,56	0,45	216,57	53,16
Order statistic tree	61,56	0,48	242,49	54,40
Double linked list	61,56	1,85	296,36	69,08
Btree	61,56	0,58	146,12	64,14
Hash table (23 lists)	87,21	2,07	672,44	87,52
Hash table (23 red-black trees)	92,34	0,60	511,78	68,15
Hash table (list)	292,42	0,80	1957,53	331,26
Hash table (red-black tree)	202,52	1,26	2070,80	411,04

Figura: Memoria (en MB) y tiempo (en s) de distintas estructuras. Lazy instantiation.

# Resultados métricas de error. Mean-Centering basado en usuarios.





# Resultados métricas de evaluación. NDCG@1000 (1).

Algorithm	10 Neighbours	20 Neighbours	30 Neighbours	40 Neighbours	50 Neighbours	100 Neighbours	150 Neighbours	200 Neighbours	400 Neighbours
baselinebias	0,2444								
baselinerandom	0,1539								
CosineNormalBMC	0,2503	0,2498	0,2490	0,2484	0,2479	0,2465	0,2460	0,2458	0,2457
CosineNormalBStd	0,2334	0,2308	0,2257	0,2215	0,2164	0,2024	0,1953	0,1922	0,1899
CosineNormalUBMC	0,2465	0,2448	0,2439	0,2430	0,2418	0,2386	0,2365	0,2352	0,2341
CosineNormalUBStd	0,2526	0,2532	0,2531	0,2523	0,2516	0,2492	0,2476	0,2467	0,2459
CosinePosNormalBMC	0,2503	0,2498	0,2490	0,2484	0,2479	0,2465	0,2460	0,2458	0,2457
CosinePosNormalBStd	0,2334	0,2308	0,2257	0,2215	0,2164	0,2024	0,1953	0,1922	0,1899
CosinePosNormalUBMC	0,2465	0,2448	0,2439	0,2430	0,2418	0,2386	0,2365	0,2352	0,2341
CosinePosNormalUBStd	0,2526	0,2532	0,2531	0,2523	0,2516	0,2492	0,2476	0,2467	0,2459
LCS-1/0IBMC	0,2470	0,2480	0,2477	0,2473	0,2469	0,2462	0,2460	0,2459	0,2458
LCS-1/0IBStd	0,2260	0,2270	0,2256	0,2235	0,2216	0,2153	0,2122	0,2109	0,2100
LCS-1/0UBMC	0,2467	0,2449	0,2431	0,2418	0,2406	0,2373	0,2356	0,2348	0,2342
LCS-1/0UBStd	0,2562	0,2538	0,2531	0,2522	0,2513	0,2486	0,2473	0,2466	0,2461
LCS-1/1IBMC	0,2464	0,2477	0,2476	0,2473	0,2470	0,2464	0,2462	0,2461	0,2461
LCS-1/1IBStd	0,2222	0,2253	0,2240	0,2223	0,2207	0,2145	0,2110	0,2093	0,2082
LCS-1/1UBMC	0,2481	0,2463	0,2443	0,2429	0,2418	0,2378	0,2357	0,2346	0,2337
LCS-1/1UBStd	0,2528	0,2525	0,2519	0,2513	0,2507	0,2484	0,2472	0,2466	0,2461
LCS30/0IBMC	0,2415	0,2392	0,2387	0,2384	0,2382	0,2381	0,2381	0,2381	0,2381
LCS30/0IBStd	0,2319	0,2283	0,2273	0,2268	0,2266	0,2265	0,2265	0,2265	0,2265
LCS30/0UBMC	0,0948	0,0941	0,0938	0,0937	0,0937	0,0936	0,0936	0,0936	0,0936
LCS30/0UBStd	0,0965	0,0959	0,0957	0,0957	0,0956	0,0956	0,0956	0,0956	0,0956
LCS30/1IBMC	0,2722	0,2696	0,2682	0,2674	0,2668	0,2662	0,2660	0,2660	0,2660
LCS30/1IBStd	0,2515	0,2458	0,2435	0,2417	0,2405	0,2384	0,2381	0,2381	0,2381
LCS30/1UBMC	0,1615	0,1599	0,1588	0,1581	0,1576	0,1565	0,1563	0,1562	0,1562
LCS30/1UBStd	0,1646	0,1640	0,1635	0,1630	0,1627	0,1620	0,1619	0,1619	0,1619
PearsonNormalBMC	0,2671	0,2695	0,2695	0,2690	0,2686	0,2675	0,2672	0,2671	0,2671
PearsonNormalBStd	0,1793	0,1930	0,2017	0,2072	0,2111	0,2223	0,2247	0,2243	0,2223

# Resultados métricas de evaluación. NDCG@1000 (2).

PearsonNormalUBMC	0,2132	0,2154	0,2164	0,2171	0,2175	0,2192	0,2194	0,2186	0,2152
PearsonNormalUBStd	0,2584	0,2647	0,2674	0,2691	0,2705	0,2694	0,2592	0,2445	0,2086
PearsonPosNormalIBMC	0,2669	0,2689	0,2685	0,2679	0,2676	0,2663	0,2660	0,2659	0,2659
PearsonPosNormalIBStd	0,1769	0,1820	0,1846	0,1862	0,1871	0,1895	0,1903	0,1905	0,1906
PearsonPosNormalUBMC	0,2258	0,2298	0,2313	0,2322	0,2325	0,2329	0,2329	0,2328	0,2328
PearsonPosNormalUBStd	0,2418	0,2437	0,2443	0,2443	0,2442	0,2436	0,2432	0,2431	0,2430
CosinePosUnion3IBMC	0,2434	0,2439	0,2440	0,2437	0,2437	0,2439	0,2442	0,2441	0,2443
CosinePosUnion3IBStd	0,0708	0,0647	0,0614	0,0570	0,0537	0,0475	0,0454	0,0445	0,0434
CosinePosUnion3UBMC	0,2289	0,2314	0,2313	0,2312	0,2307	0,2305	0,2304	0,2307	0,2311
CosinePosUnion3UBStd	0,2403	0,2447	0,2455	0,2460	0,2459	0,2461	0,2460	0,2456	0,2444
CosineUnion3IBMC	0,2434	0,2439	0,2440	0,2437	0,2437	0,2439	0,2442	0,2441	0,2443
CosineUnion3IBStd	0,0708	0,0647	0,0614	0,0570	0,0537	0,0475	0,0454	0,0445	0,0434
CosineUnion3UBMC	0,2289	0,2314	0,2313	0,2312	0,2307	0,2305	0,2304	0,2307	0,2311
CosineUnion3UBStd	0,2403	0,2447	0,2455	0,2460	0,2459	0,2461	0,2460	0,2456	0,2444
PearsonPosUnion0IBMC	0,2454	0,2452	0,2448	0,2446	0,2443	0,2440	0,2439	0,2439	0,2439
PearsonPosUnion0IBStd	0,1937	0,1889	0,1867	0,1857	0,1850	0,1839	0,1838	0,1837	0,1837
PearsonPosUnion0UBMC	0,2571	0,2529	0,2504	0,2488	0,2476	0,2443	0,2429	0,2423	0,2420
PearsonPosUnion0UBStd	0,2486	0,2511	0,2516	0,2518	0,2518	0,2513	0,2508	0,2505	0,2503
PearsonPosUnion3IBMC	0,2480	0,2483	0,2478	0,2474	0,2470	0,2464	0,2463	0,2462	0,2462
PearsonPosUnion3IBStd	0,2086	0,2040	0,2016	0,1998	0,1988	0,1964	0,1958	0,1957	0,1956
PearsonPosUnion3UBMC	0,2424	0,2432	0,2431	0,2428	0,2423	0,2411	0,2403	0,2399	0,2397
PearsonPosUnion3UBStd	0,2511	0,2524	0,2524	0,2521	0,2517	0,2500	0,2490	0,2486	0,2483
PearsonUnion0IBMC	0,2456	0,2469	0,2474	0,2477	0,2479	0,2481	0,2485	0,2486	0,2488
PearsonUnion0IBStd	0,1968	0,1995	0,2023	0,2046	0,2063	0,2106	0,2145	0,2171	0,2206
PearsonUnion0UBMC	0,2473	0,2442	0,2423	0,2409	0,2399	0,2386	0,2383	0,2382	0,2355
PearsonUnion0UBStd	0,2627	0,2677	0,2696	0,2711	0,2722	0,2736	0,2697	0,2627	0,2359
PearsonUnion3IBMC	0,2482	0,2499	0,2505	0,2509	0,2511	0,2517	0,2521	0,2523	0,2524
PearsonUnion3IBStd	0,2113	0,2147	0,2183	0,2205	0,2220	0,2252	0,2255	0,2259	0,2265
PearsonUnion3UBMC	0,2339	0,2341	0,2339	0,2338	0,2336	0,2335	0,2337	0,2334	0,2311
PearsonUnion3UBStd	0,2642	0,2687	0,2703	0,2714	0,2724	0,2728	0,2677	0,2602	0,2338

- 1 Introducción
- 2 Estado del arte
- 3 Sistema desarrollado
- 4 Resultados
- 5 Conclusiones y trabajo futuro

- URL del sistema desarrollado:  
<https://bitbucket.org/PabloSanchezP/dt4recsys>
- Publicación en CERI:2016. 4th Spanish Conference in Information Retrieval.
- El algoritmo de la longitud de la subcadena común más larga es un algoritmo que puede ser aplicable en sistemas de recomendación.
- Las tablas hash no son siempre óptimas en todos los aspectos de la aplicación.
- La normalización de los datos en general trae mejores resultados. Al igual que la estrategias basadas en artículos.

- Nuevas formas de plantear el algoritmo de LCS (utilizando otros datos como la fecha de visionado).
- Plantear si es posible aplicar otros algoritmos (p.e. Set Covering).
- Diseñar una estructura específica para los sistemas de recomendación.

# Final

# Memoria y tiempo Completo

Structure	Load training memory	Load training time	Training memory	Training time
Binary search tree	61,01	1,86	311,97	65,10
Red-black tree	61,01	0,49	379,48	55,25
Order statistic tree	61,01	0,51	512,62	56,74
Double linked list	61,01	1,85	412,82	70,06
Hash table (23 lists)	85,96	0,54	1263,77	63,92
Hash table (23 red-black trees)	90,95	0,60	1239,18	69,36
Btree	61,01	0,65	365,35	66,07

Structure	Load test memory	Load test time	Test memory	Test time
Binary search tree	15,53	0,19	41,38	4700,38
Red-black tree	15,53	0,05	59,56	437,34
Order statistic tree	15,53	0,05	70,22	453,49
Double linked list	15,31	0,22	NaN	7644,48
Hash table (23 lists)	15,53	0,05	NaN	423,54
Hash table (23 red-black trees)	10,54	0,05	NaN	376,36
Btree	20,52	0,05	104,95	700,09

Figura: Memoria y tiempo de distintas estructuras. Todos los algoritmos. Incluido el test.

# Artículos predichos

Algorithm	10 Neighbours	20 Neighbours	30 Neighbours	40 Neighbours	50 Neighbours	100 Neighbours	150 Neighbours	200 Neighbours	400 Neighbours
baselinebias	1996,6000								
baselinerandom	1996,6000								
CosineNormalIBMC	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000
CosineNormalIBStd	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000
CosineNormalUBMC	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000
CosineNormalUBStd	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000
CosinePosNormalIBMC	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000
CosinePosNormalIBStd	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000
CosinePosNormalUBMC	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000
CosinePosNormalUBStd	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000	1996,6000
LCS-1/0IBMC	19963,4000	19963,4000	19963,4000	19963,4000	19963,4000	19963,4000	19963,4000	19963,4000	19963,4000
LCS-1/0IBStd	19963,4000	19963,4000	19963,4000	19963,4000	19963,4000	19963,4000	19963,4000	19963,4000	19963,4000
LCS-1/0UBMC	19963,6000	19963,6000	19963,6000	19963,6000	19963,6000	19963,6000	19963,6000	19963,6000	19963,6000
LCS-1/0UBStd	19963,6000	19963,6000	19963,6000	19963,6000	19963,6000	19963,6000	19963,6000	19963,6000	19963,6000
LCS-1/1IBMC	19965,4000	19965,4000	19965,4000	19965,4000	19965,4000	19965,4000	19965,4000	19965,4000	19965,4000
LCS-1/1IBStd	19965,4000	19965,4000	19965,4000	19965,4000	19965,4000	19965,4000	19965,4000	19965,4000	19965,4000
LCS-1/1UBMC	19966,4000	19966,4000	19966,4000	19966,4000	19966,4000	19966,4000	19966,4000	19966,4000	19966,4000
LCS-1/1UBStd	19966,4000	19966,4000	19966,4000	19966,4000	19966,4000	19966,4000	19966,4000	19966,4000	19966,4000
LCS30/0IBMC	9838,0000	9838,0000	9838,0000	9838,0000	9838,0000	9838,0000	9838,0000	9838,0000	9838,0000
LCS30/0IBStd	9838,0000	9838,0000	9838,0000	9838,0000	9838,0000	9838,0000	9838,0000	9838,0000	9838,0000
LCS30/0UBMC	9211,0000	9211,0000	9211,0000	9211,0000	9211,0000	9211,0000	9211,0000	9211,0000	9211,0000
LCS30/0UBStd	9211,0000	9211,0000	9211,0000	9211,0000	9211,0000	9211,0000	9211,0000	9211,0000	9211,0000
LCS30/1IBMC	15203,0000	15203,0000	15203,0000	15203,0000	15203,0000	15203,0000	15203,0000	15203,0000	15203,0000
LCS30/1IBStd	15203,0000	15203,0000	15203,0000	15203,0000	15203,0000	15203,0000	15203,0000	15203,0000	15203,0000

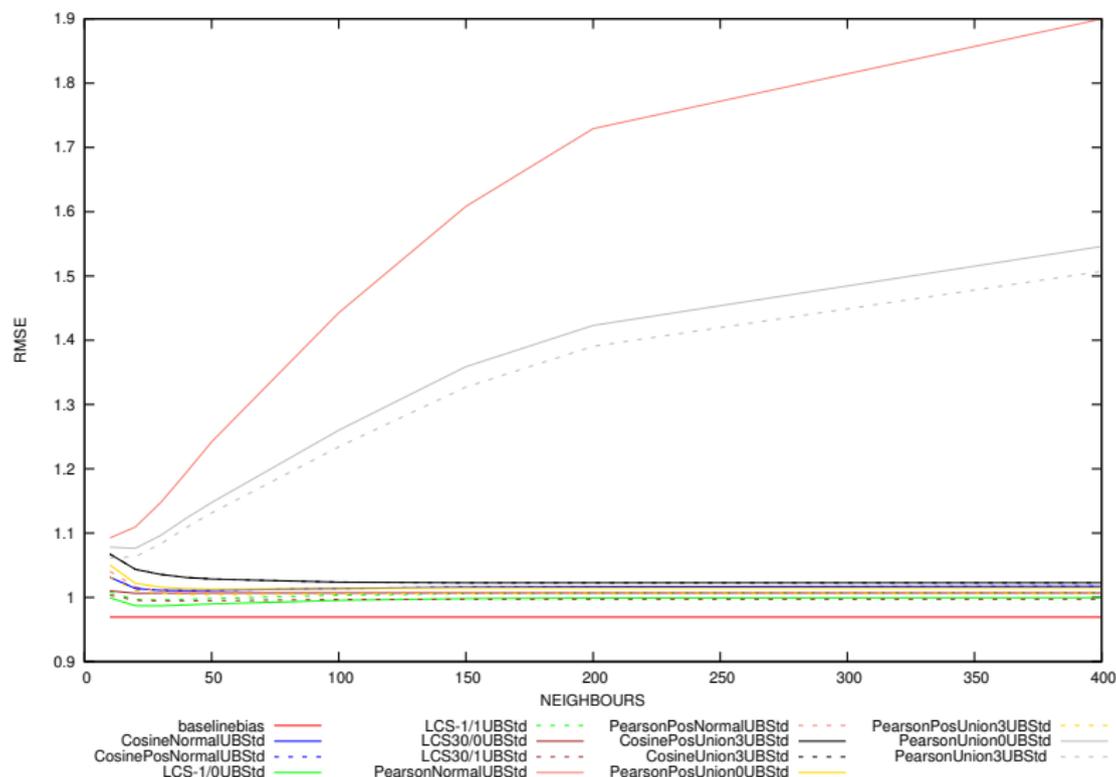
Figura: Número de artículos predichos por cada algoritmo (1).

# Artículos predichos

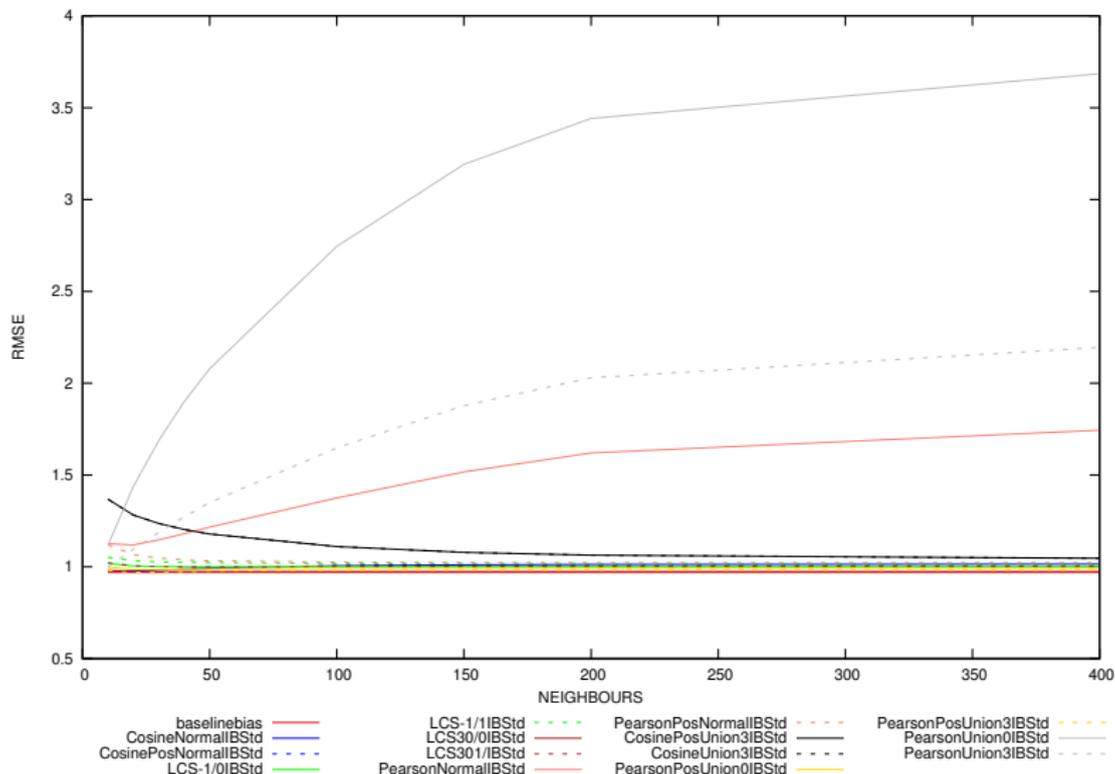
LCS30/1UBMC	15339,4000	15339,4000	15339,4000	15339,4000	15339,4000	15339,4000	15339,4000	15339,4000	15339,4000
LCS30/1UBStd	15339,4000	15339,4000	15339,4000	15339,4000	15339,4000	15339,4000	15339,4000	15339,4000	15339,4000
PearsonNormalIBMC	19901,4000	19901,4000	19901,4000	19901,4000	19901,4000	19901,4000	19901,4000	19901,4000	19901,4000
PearsonNormalIBStd	19901,4000	19901,4000	19901,4000	19901,4000	19901,4000	19901,4000	19901,4000	19901,4000	19901,4000
PearsonNormalUBMC	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
PearsonNormalUBStd	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
PearsonPosNormalIBMC	19901,2000	19901,2000	19901,2000	19901,2000	19901,2000	19901,2000	19901,2000	19901,2000	19901,2000
PearsonPosNormalIBStd	19901,2000	19901,2000	19901,2000	19901,2000	19901,2000	19901,2000	19901,2000	19901,2000	19901,2000
PearsonPosNormalUBMC	19944,8000	19944,8000	19944,8000	19944,8000	19944,8000	19944,8000	19944,8000	19944,8000	19944,8000
PearsonPosNormalUBStd	19944,8000	19944,8000	19944,8000	19944,8000	19944,8000	19944,8000	19944,8000	19944,8000	19944,8000
CosinePosUnion3IBMC	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
CosinePosUnion3IBStd	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
CosinePosUnion3UBMC	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
CosinePosUnion3UBStd	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
CosineUnion3IBMC	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
CosineUnion3IBStd	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
CosineUnion3UBMC	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
CosineUnion3UBStd	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
PearsonPosUnion0IBMC	19961,2000	19961,2000	19961,2000	19961,2000	19961,2000	19961,2000	19961,2000	19961,2000	19961,2000
PearsonPosUnion0IBStd	19961,2000	19961,2000	19961,2000	19961,2000	19961,2000	19961,2000	19961,2000	19961,2000	19961,2000
PearsonPosUnion0UBMC	19958,0000	19958,0000	19958,0000	19958,0000	19958,0000	19958,0000	19958,0000	19958,0000	19958,0000
PearsonPosUnion0UBStd	19958,0000	19958,0000	19958,0000	19958,0000	19958,0000	19958,0000	19958,0000	19958,0000	19958,0000
PearsonPosUnion3IBMC	19950,2000	19950,2000	19950,2000	19950,2000	19950,2000	19950,2000	19950,2000	19950,2000	19950,2000
PearsonPosUnion3IBStd	19950,2000	19950,2000	19950,2000	19950,2000	19950,2000	19950,2000	19950,2000	19950,2000	19950,2000
PearsonPosUnion3UBMC	19954,8000	19954,8000	19954,8000	19954,8000	19954,8000	19954,8000	19954,8000	19954,8000	19954,8000
PearsonPosUnion3UBStd	19954,8000	19954,8000	19954,8000	19954,8000	19954,8000	19954,8000	19954,8000	19954,8000	19954,8000
PearsonUnion0IBMC	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
PearsonUnion0IBStd	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
PearsonUnion0UBMC	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
PearsonUnion0UBStd	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
PearsonUnion3IBMC	19950,6000	19950,6000	19950,6000	19950,6000	19950,6000	19950,6000	19950,6000	19950,6000	19950,6000
PearsonUnion3IBStd	19950,6000	19950,6000	19950,6000	19950,6000	19950,6000	19950,6000	19950,6000	19950,6000	19950,6000
PearsonUnion3UBMC	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000
PearsonUnion3UBStd	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000	19966,6000

Figura: Número de artículos predichos por cada algoritmo (2).

# Rating prediction UB-Estándar. Movielens.



# Rating prediction IB-Estándar. Movielens.



# Rating prediction preliminares Lastfm.

