Recomendación Basada en Grafos

Denis Parra

IIC 3633

2016

Agenda Semestral

Week	Fecha semana	Clase Martes	Clase Jueves	Presentador 1	Presentador 2	Presentador 3
I	2 - 4 Ago	Intro + CF	CF + Clustering			
II	9 - 11 Ago	CF item-based	Slope One + RecSys			
III	16 - 18 Ago	Evaluacion de RecSys	Evaluacion de RecSys			
IV	23 - 25 Ago	Content-based	Tag-based			
V	30 Ag - 1 Sept	Hybrid	Factorizacion Matricial			
VI	6 - 8 Sept	Context-aware RecSys	Implicit Feedback			
VII	13 - 15 Sept	student presentation (Context, MF	RECSYS Conf	V. Dominguez	J. Schellman	P. Lopez
VIII	20 - 22 Sept	RECSYS Conf	student presentation (IF, MF)	F. Lucchini	V. Claro	V. Castillo
IX	27 - 29 Sept	Presentaciones: Proy. Final	Presentaciones: Proy. Final			
X	4 - 6 Oct	User-centric RecSys/Interfaces	student presentation	J. Lee	C. Kutscher	R. Carmona
XI	11 - 13 Oct	Active Learning/Ranking	student presentation	F. Rojos	J. Navarro	N. Morales
XII	18 - 20 Oct	Graph-based	student presentation	P. Messina	S. Martí	J. Castro
XIII	25 - 27 Oct	Applications: Social/Trust/Music	student presentation	J.M. Herrera	V. Dragicevic	L. Zorich
XIV	1 - 3 Nov	Applications: POI/Tourism	student presentation	I. Becker	T. Hepner	M. Troncoso
XV	8 - 10 Nov	Applications: Educ/Soft.Eng.	student presentation	R. Perez	P. Sanabria	J. Diaz
XVI	15 - 17 Nov	Deep Learning	student presentation	Felipe del Río	L. Pose	G. Sepulveda
XVII	29 Nov - 1 Dic	Presentacion Final	Presentacion Final			

Problema de Recomendación

- Nuevamente revisitamos el problema de recomendación.
- Una alternativa válida a los métodos vistos hasta ahora es explotar las relaciones entre items en la forma de grafos.

Hoy

- Associative retrieval techniques to alleviate the sparsity problem in CF (Huang et al. 2004)
- The link Prediction Problem for Social Networks (Liben-Nowel, Kleinberg, 2002)

Paper 1

- Zan Huang, Hsinchun Chen, and Daniel Zeng. 2004. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Trans. Inf. Syst. 22, 1 (January 2004), 116-142.

Resumen

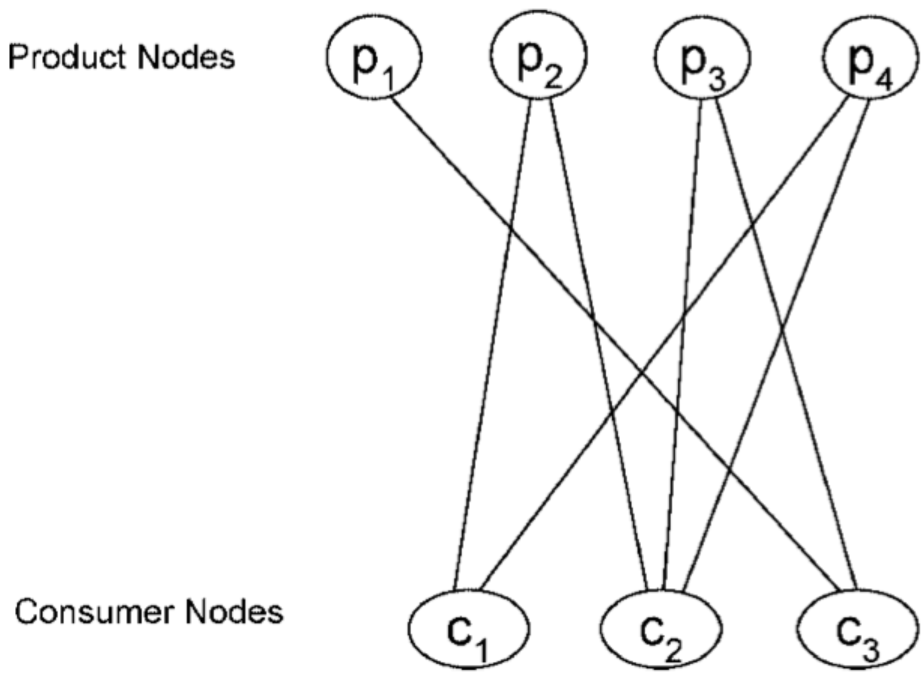
- Lidiar con el problema de escasez de evaluaciones del usuario (ratings)
- Filtrado Colaborativo es estudiado como un grafo bi-partito.
- Técnicas de recuperación asociativa son utilizadas sobre el grafo (Spreading Activation)
- RESULTADO: Cuando hay escasez de ratings, estas técnicas basadas en grafos mejoran los resultado del filtrado colaborativo.

El Problema de Escasez (Sparsity)

- Al 2004, los problemas de cold-start y new-item se habían atacado usando:
  - Item-Based CF (Sarwar 2001)
  - Reducción de Dimensionalidad (Golderg 2001)
  - Híbridos (Balanovic 2002, Basu 1998, Condliff 1999, etc.)
- Ninguno de los métodos mencionados había tenido consenso absoluto de su éxito

CF como Recuperación Asociativa

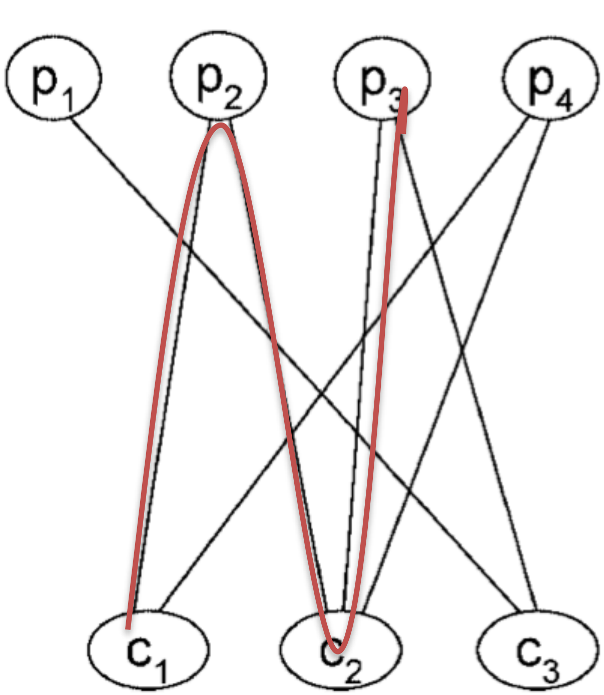
- Idea básica: construir un grafo entre usuarios e items y explorar asociaciones transitivas entre ellos.



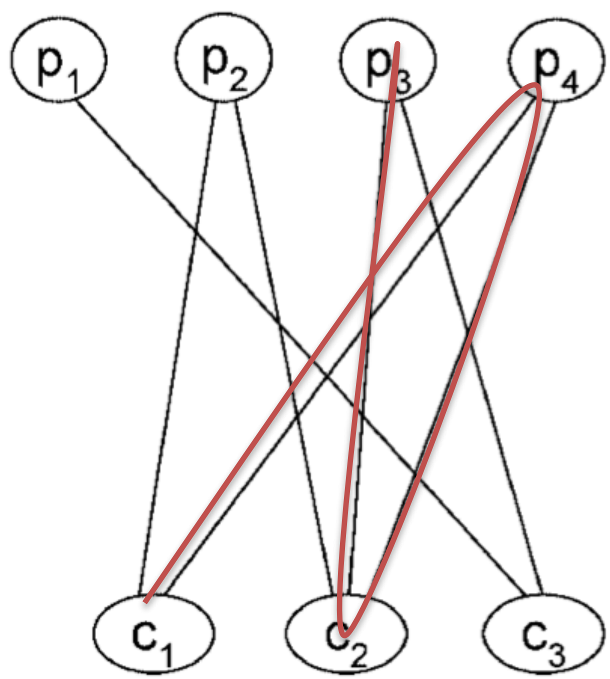
$$\begin{array}{c c} & \begin{array}{cccc} p_1 & p_2 & p_3 & p_4 \end{array} \\ \begin{array}{c} c_1 \\ c_2 \\ c_3 \end{array} & \left[\begin{array}{cccc} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{array}\right] \end{array}$$

CF como Recuperación Asociativa

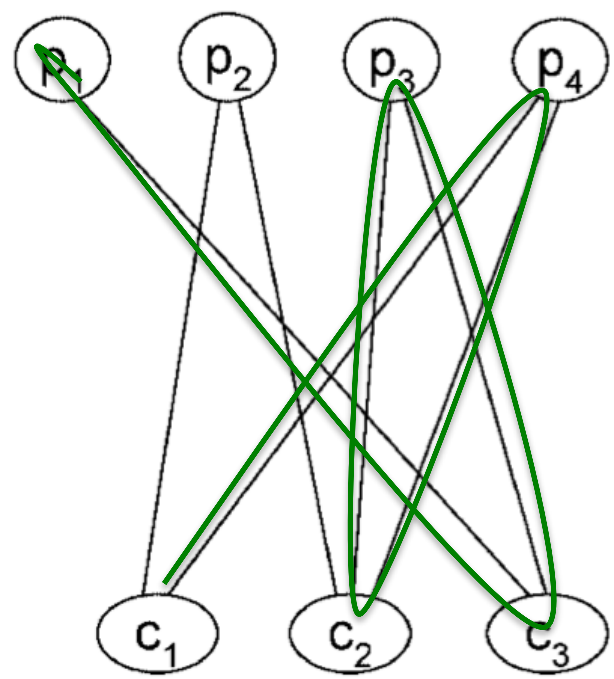
- Idea básica: construir un grafo entre usuarios e items y explorar asociaciones transitivas entre ellos.



3 hops



3 hops



5 hops

Notación Matricial

- Consideremos la matriz consumidor/producto A
- Parámetros: M: hops, α= decaimiento (peso asociado al enlace

$$A_\alpha^M = \begin{cases} \alpha A, & \text{if } M = 1, \\ \alpha^2 A \cdot A^T \cdot A_\alpha^{M-2}, & \text{if } M = 3, 5, 7, \ldots \end{cases}$$

Ejemplo

- Dado A

$$\begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & p_2 & p_3 & p_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

- Luego, para M = 3, α= 0.5

$$A_{0.5}^{3} = \begin{bmatrix} 0 & 0.5 & 0.25 & 0.5 \\ 0.125 & 0.625 & 0.5 & 0.625 \\ 0.25 & 0.125 & 0.375 & 0.125 \end{bmatrix}$$

- Luego, para M = 5, α= 0.5

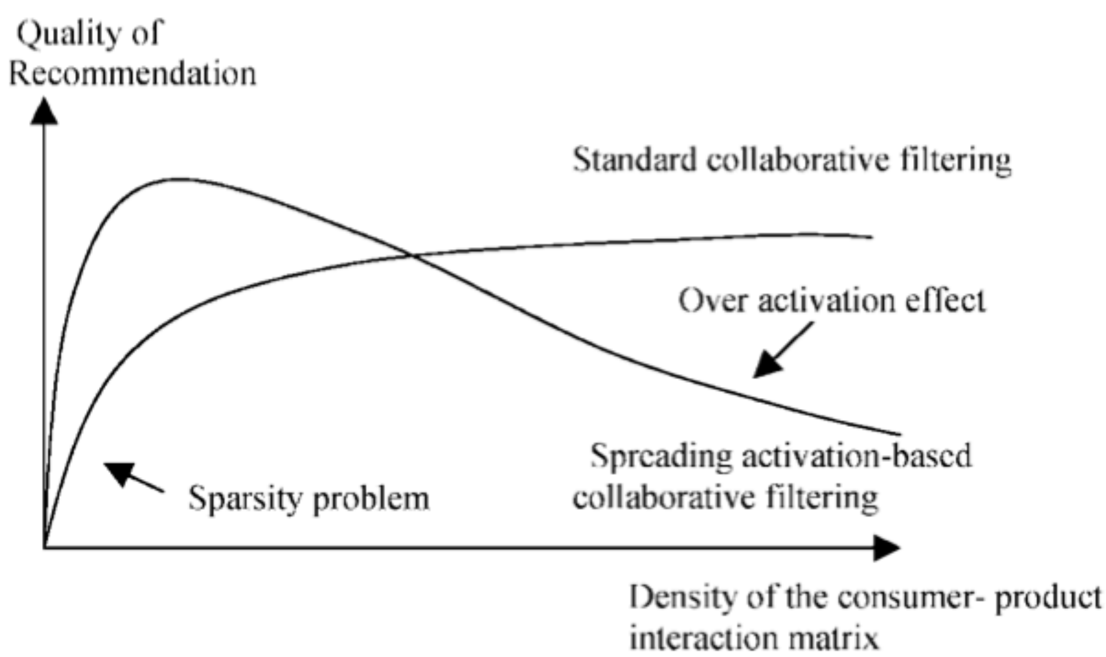
$$A_{0.5}^{5} = \begin{bmatrix} 0.0625 & 0.5625 & 0.375 & 0.5625 \\ 0.15625 & 0.75 & 0.59375 & 0.75 \\ 0.15625 & 0.21875 & 0.3125 & 0.21875 \end{bmatrix}$$

Dificultades

- Calcular la potencia de una matriz puede ser muy costoso para un “c” y un “n” muy grandes, lo cual motiva los 3 métodos probados por Huang et al. en el paper.

Supuesto de la Investigación

- Los métodos de Spreading Activation funcionarán mejor cuando la red tiene muy baja densidad, en caso contrario puede ocurrir sobre-activación.



Modelos

- Constrained Leaky Capacitor Model (LCM)
- Branch-and-Bound
- Hopfield Net

LCM

- Propuesto por Anderson (1983)

$$R(r \times r) = \begin{pmatrix} I(|P| \times |P|) & A^T(|C| \times |P|) \\ A(|P| \times |C|) & I(|C| \times |C|) \end{pmatrix}$$

- Pasos:
  - Identificar nodo-vector inicial V, setear D(0)
  - Cálculo de nivel de activación

$$D(t) = V + M'D(t-1), \quad M = (1-\gamma)I + \alpha R,$$

Donde (1-γ): speed of decay (0.8), α: efficiency (0.8)

  - Condición de detención: en el paper = 10, top 50

Branch-and-Bound

- Implementación basada en (Chen & Ng 1995)
- Paso 1, Inicialización: Nodo correspondiente al usuario es activado (1), los otros = 0. Cola $Q_{priority}$ se inicializa con nodo usuario activo.
- Paso 2, Cálculo de activación: Sacar nodos de $Q_{priority,}$ por cada nodo vecino calcular

$$\mu_j(t+1) = \mu_i(t) \times t_{ij}$$

y agregar/actualizar nodo activado a Q_{output}

- Paso 3, detención: determinada empiricamente (70)

Holpfield Net

- Paralelo con red neuronal. Usuarios e items son neuronas. Sinapsis son las activaciones.
- Inicialización: igual que las anteriores
- Calculo de activación:

$$\mu_j(t+1) = f_s\left[\sum_{i=0}^{n-1} t_{ij}\mu_i(t)\right], 0 \leq j \leq n-1 \qquad f_s(x) = \frac{1}{1+\exp((\theta_1 - x)/\theta_2)}$$

- Condición de detención:

$$\sum_j \mu_j(t+1) - \sum_j \mu_j(t) < \varepsilon \times t\,.$$

Estudio Experimental

- Tienda de libros en linea de China

9,695 libros / 2,000 usuarios / 18,771 transacciones

- Métricas de evaluación:

  Precision, Recall, F-1

- Y utility rank

$$R_i = \sum_j \frac{p(i,j)}{2^{(j-1)/(h-1)}} \qquad R = 100 \frac{\sum_i R_i}{\sum_i R_i^{\max}},$$

$$\text{where } p(i,j) = \begin{cases} 1, & \text{if item } j \text{ is in user } i\text{'s future purchase list}, \\ 0, & \text{otherwise.} \end{cases}$$

Recordemos hipótesis

- **H1**. Spreading activation-based CF **can achieve higher recommendation quality** than the 3-hop, User-based (Correlation), User-based (Vector Similarity), and Item-based approaches.
- **H2**. Spreading activation-based CF can achieve higher recommendation quality than the 3-hop, User-based (Correlation), User-based (Vector Similarity), and Item-based approaches **for new users** (the cold-start problem).
- **H3**. The recommendation quality of spreading activation-based CF **decreases when the density** of user–item interactions **is beyond a certain level** (the over-activation effect).

Resultados

Table I. Experimental Results for H1

Algorithm	Precision	Recall	F′-measure	Utility score
Hopfield	**0.0266**	**0.1519**	**0.0407**	**7.94**
3-hop	**0.0155**	0.0705	**0.0230**	**3.51**
User-based (Correlation)	0.0181	**0.1064**	0.0279	4.57
User-based (Vector Similarity)	0.0187	0.1089	0.0288	4.56
Item-based	0.0082	0.0516	0.0126	0.65

H1: Comparación de algoritmos bajo condiciones normales

Table II. Experimental Results for H2

Algorithm	Precision	Recall	F′-measure	Utility score
Hopfield	**0.0054**	**0.1122**	**0.0102**	**9.78**
3-hop	**0.0017**	0.0315	**0.0031**	**2.36**
User-based (Correlation)	0.0027	**0.0525**	0.0051	3.86
User-based (Vector Similarity)	0.0027	**0.0525**	0.0051	3.86
Item-based	0.0014	0.0282	0.0027	0.43

H2: Comparación de algoritmos con usuarios sparse

Resultados 2

Table III. Recommendation Recall for Regular Users and New Users

Algorithm	Regular users	New users	Decrease	Decrease Percentage	t-test p-value
Hopfield	0.1568	0.1122	0.0446	28.44%	0.0060
3-hop	0.0728	0.0315	0.0413	56.73%	0.0001
User-based (Correlation)	0.1064	0.0525	0.0539	50.66%	0.0002
User-based (Vector Similarity)	0.1089	0.0525	0.0564	51.79%	0.0000
Item-based	0.0516	0.0282	0.0234	45.35%	0.0318

H2: Comparación de algoritmos en base a Recall, con usuarios cold-start

Resultados 3

Table IV. Characteristics of the Graphs of Varying Degree of Sparsity

Graph	Number of links	Density	Average degree of customer node	Standard deviation of customer node degree	Average degree of book node	Standard deviation of book node degree
G1	4278	0.000031	1.607	4.378	0.124	0.422
G2	6382	0.000047	2.152	5.603	0.235	0.667
G3	9690	0.000071	3.011	7.182	0.409	1.069
G4	12952	0.000095	3.868	9.253	0.580	1.621
G5	16256	0.000119	4.732	11.106	0.750	2.095
G6	19376	0.000142	5.595	13.057	0.915	2.231
G7	21494	0.000157	6.189	14.569	1.026	2.321
G8	25526	0.000187	7.279	16.619	1.228	4.649
G9	28692	0.000210	8.120	17.831	1.386	4.921
G10	31826	0.000233	8.950	18.431	1.540	5.042
G11	35038	0.000256	9.805	19.358	1.700	5.143

Resultados 3.2

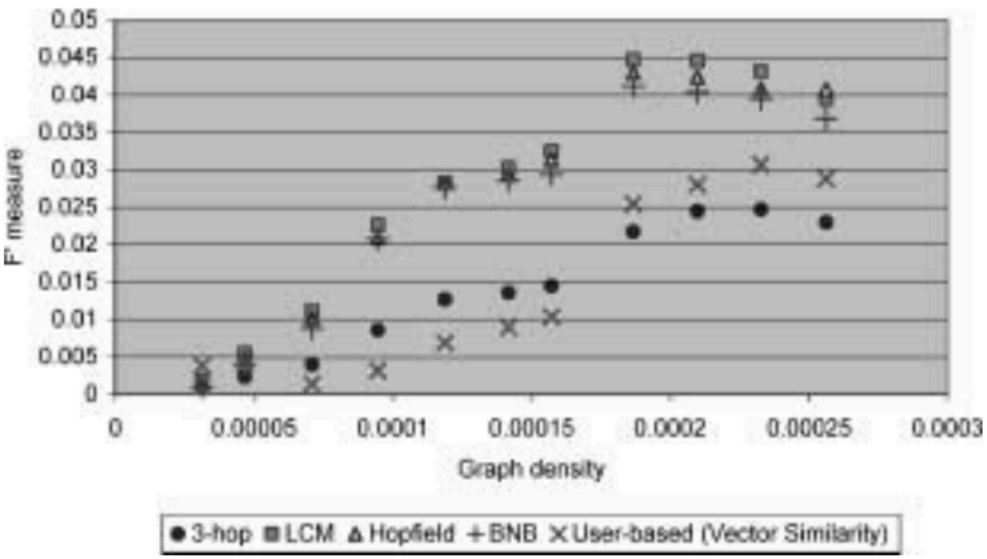


Fig. 3. Over-activation effect (G1–G11).

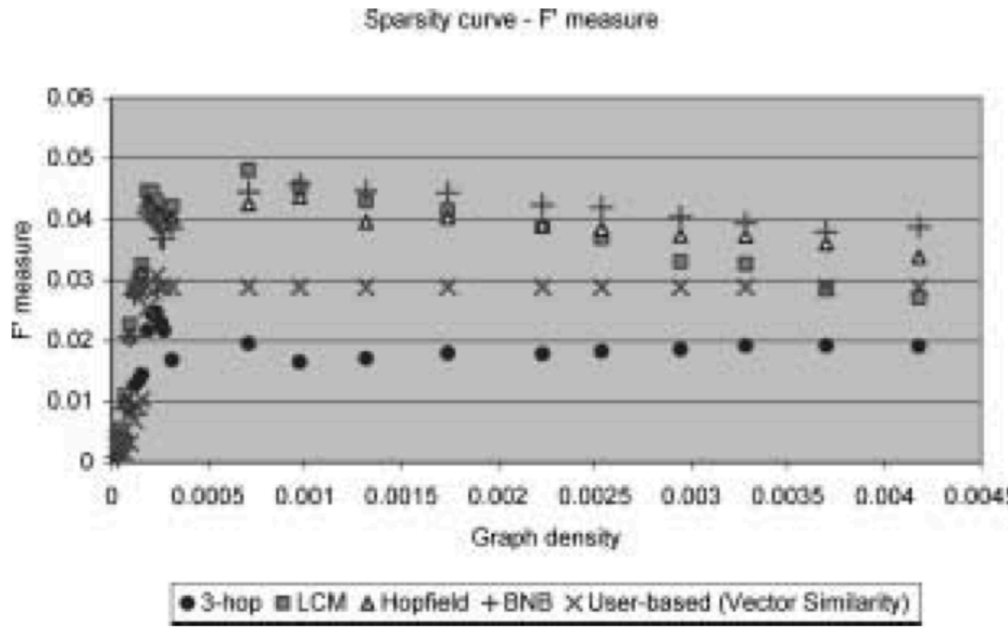


Fig. 4. Over-activation effect (with graphs enhanced by item associations).

Computational Efficiency

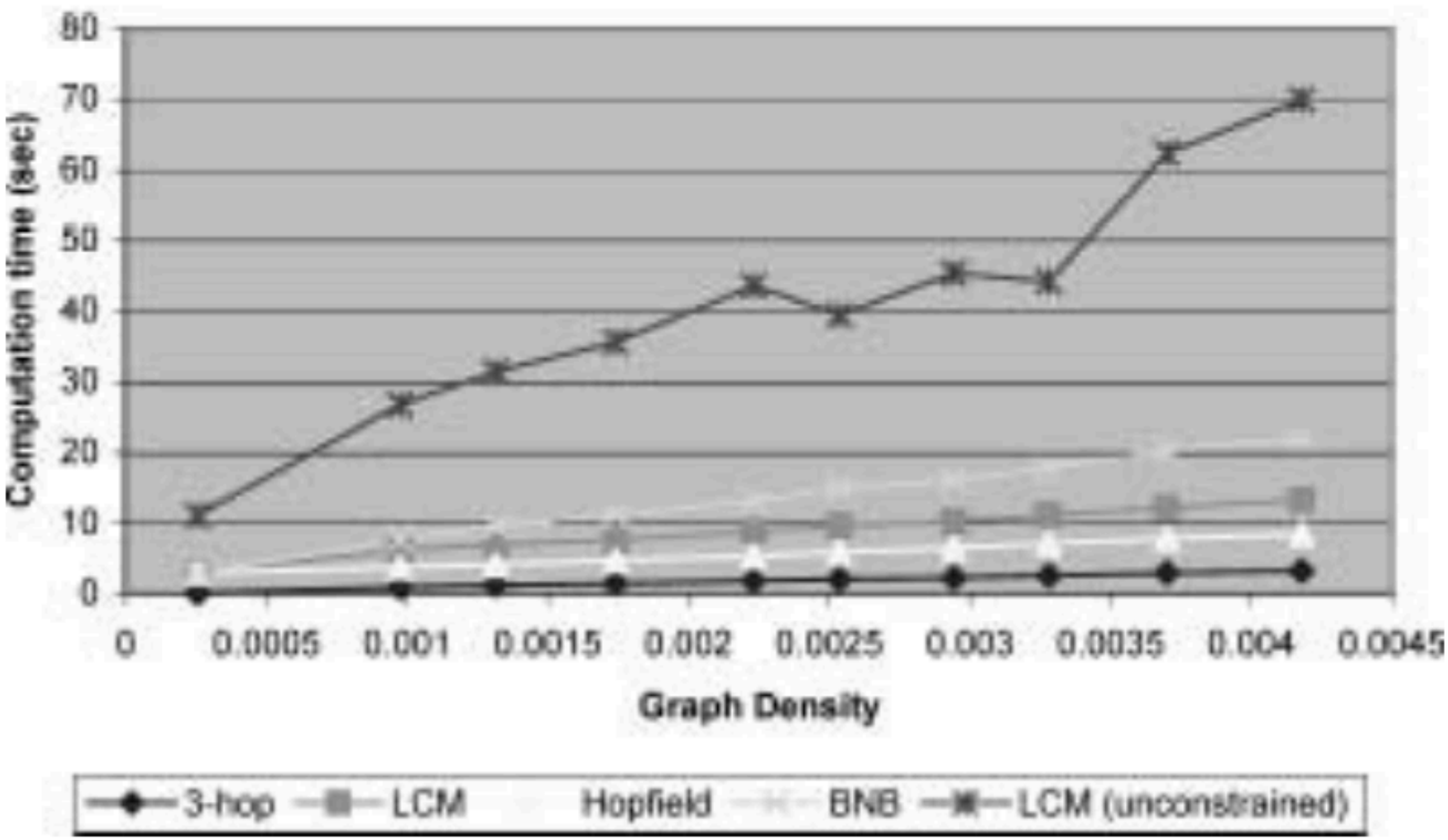


Fig. 5. Computational efficiency analysis of spreading activation algorithms.

Lecciones

- H1, H2 y H3 se demuestran
- Sensibilidad de los parámetros:
  - LCM: no es muy sensible (alfa, gama e iteraciones)
  - BNB: diferencia en 70 y 100 iteraciones es baja, sobre 100 baja drásticamente
  - Hopfield Net: poca diferencia entre parámetros

Paper 2

- Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. Journal of the American society for information science and technology, 58(7), 1019-1031.

El Problema

- **Link prediction problem:** Given the links in a social network at time t or during a time interval I, we wish to predict the links that will be added to the network during the later time interval from time t' to a some given future time.

- **Main approach:** Use measures of network-proximity adapted from graph theory, computer science, and the social sciences to determine which unconnected nodes are 'close together' in the topology of the network.

Definiciones

$G = \langle V, E \rangle$ ← Social Network

$e = \langle u, v \rangle \in E$ ← Interaction between u and V

$G[t_0, t_1]$ ← Given Subgraph as training set

$G[t_2, t_3]$ ← Infer new Edges/used for testing

$Score(u, v)$ ← Liklihood that u and V share an edge (Proximity or Similarity)

Also, for a node x, Γ(x) represents the set of neighbors of x. degree(x) is the size of the Γ(x).

Imagen desde http://be.amazd.com/link-prediction/

Notación para arXiv de Física

- $[t_0, t_0']$ are the three years $1993 - 1996$
- $[t_1, t_1']$ are the three years $1997 - 1999$
- $G[1993, 1996] = G_{collab} = \langle A, E_{old} \rangle$
- E_{new} is the set of edges $\langle u, v \rangle$ such that authors u and v co-authored an article sometime during $1997 - 1999$ but not during $1993 - 1996$
- Each link predictor p outputs a ranked list L_p of pairs in $A \times A - E_{old}$. List is ordered according to decreasing values of $\mathrm{score}(x, y)$ for $\langle x, y \rangle \in A \times A - E_{old}$

Métricas 1: distancia en el grafo

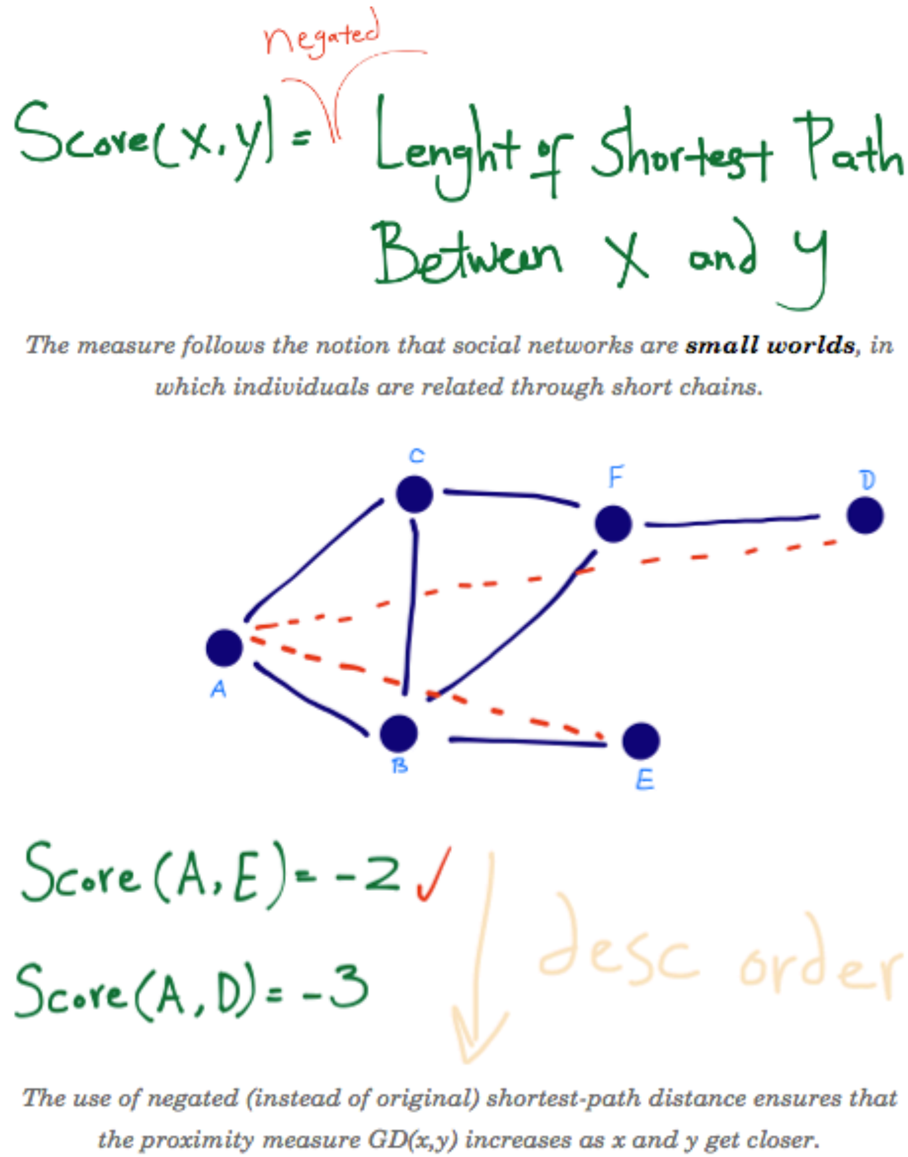


Imagen desde http://be.amazd.com/link-prediction/

Vecinos en Común

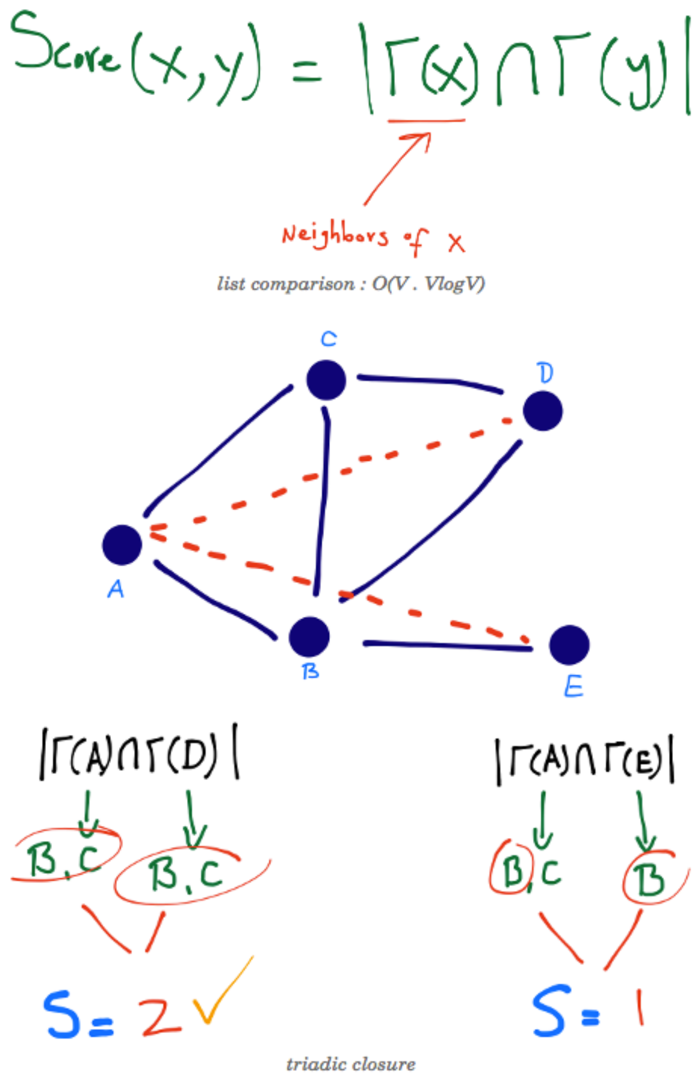


Imagen desde http://be.amazd.com/link-prediction/

Jaccard

Common friends

$$\mathrm{Score}(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

total friends

This metric solves the problem where two nodes could have many common neighbors becuase they have lots of neighbors, not becuase they are strongly related

Imagen desde http://be.amazd.com/link-prediction/

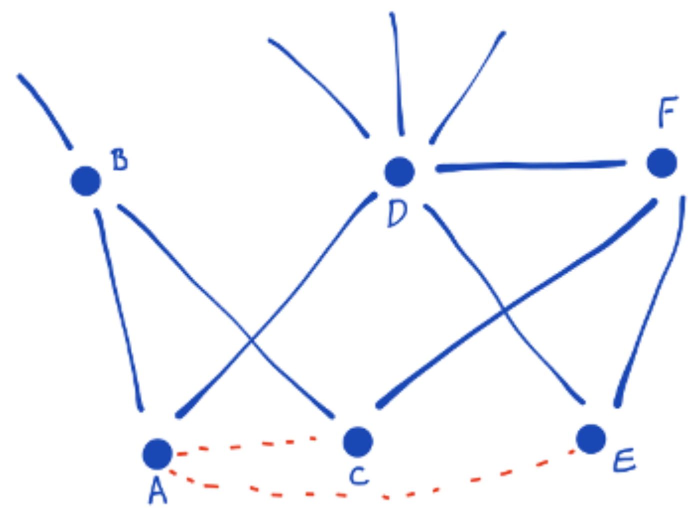
Adamic-Adar

$$Score(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log|\Gamma(z)|}$$

Frequency of Z

weighting rare features more heavily

list comparison : O(V . VlogV)



$$\Gamma(A) \cap \Gamma(C) = B$$

$$\frac{1}{\log(\Gamma(B))} = \frac{1}{\log 3} = 2.09$$

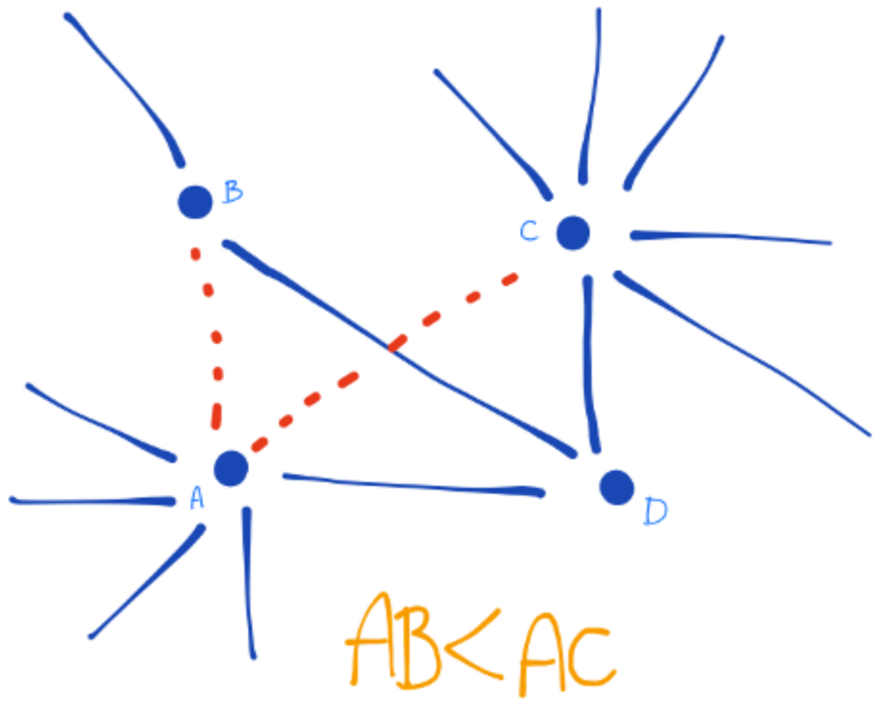
$$\Gamma(A) \cap \Gamma(E) = D$$

$$\frac{1}{\log(\Gamma(D))} = \frac{1}{\log 6} = 1.2$$

triadic closure

Preferential Attachment

$$Score(x,y) = |\Gamma(x)| \cdot |\Gamma(y)|$$



The link between A and C is more probable than the link between A and B as C have many more neighbors than B

Imagen desde http://be.amazd.com/link-prediction/

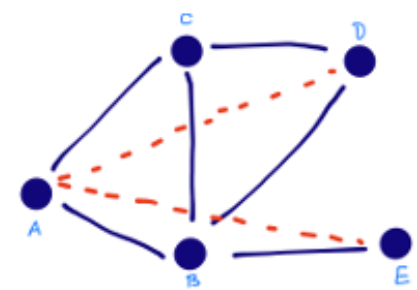
Katz

$$\text{Score}(x,y) = \sum_{l=1}^{\infty} \beta^{l} \cdot \left|\text{Path}^{l}_{x,y}\right|$$

exponentially damped by lenght

Set of all lenght l Paths from x to y

A very small β yields predictions much like common neighbors, because paths of length three or more contribute very little to the summation.



# of Hops

$\text{Path}^{2}_{A,D} = 2$ $\quad$ $\text{Path}^{3}_{A,D} = 2$

$S = \frac{1}{2} \cdot 2 + \frac{1}{4} \cdot 2 + \cdots$

Damping Factor

$\text{Path}^{2}_{A,E} = 1$ $\quad$ $\text{Path}^{3}_{A,E} = 1$

$S = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 1 + \cdots$

Espectrales/Random Walk

- Hitting Time

  - Consider a *random walk* on G_{collab} which starts at x and iteratively moves to a neighbour of x chosen uniformly at random from $\Gamma(x)$.
  - The **Hitting Time** $H_{x,y}$ from x to y is the expected number of steps it takes for the RW starting at x to reach y.

$$\text{score}(x,y) = -H_{x,y}$$

- Rooted Page Rank

  - **Rooted PageRank:**

$score(x,y) =$ stationary distribution weight of y under this scheme

- SimRank

  - **SimRank**$_\gamma$: Let $\text{similarity}(x,y)$ be a fixed point of

$$\text{similarity}(x,y) = \gamma \frac{\sum_{a\in\Gamma(x)} \sum_{b\in\Gamma(y)} \text{similarity}(a,b)}{|\Gamma(x)||\Gamma(y)|}$$

where $\gamma \in [0,1]$

$$\text{score}(x,y) = \text{similarity}(x,y)$$

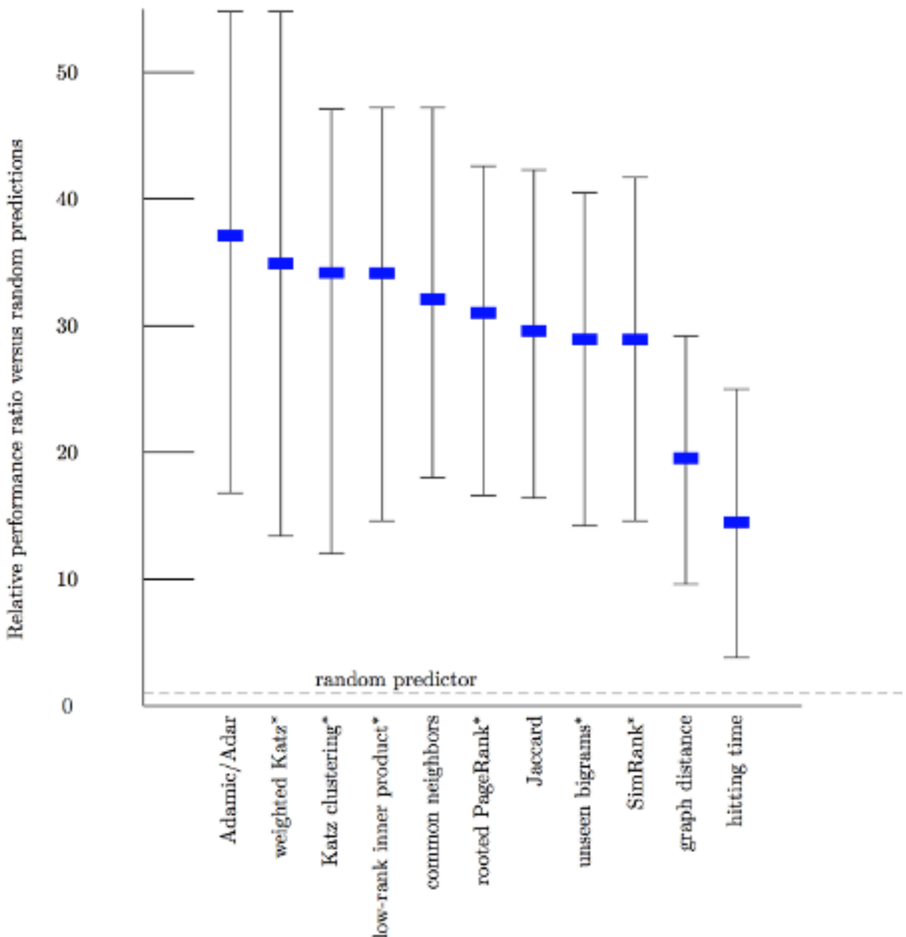
Resultados

- Prediction accuracy will be tabulated in terms of relative improvement over a *random predictor*
- The random predictor simply predicts randomly selected pairs of authors from Core who did not collaborate during the training interval $1993 - 1996$.
- The probability the random prediction is correct is

$$\frac{1}{\binom{|\mathrm{Core}|}{2} - |E_{old}|}$$

- This value ranges from 0.15% in $\mathrm{cond\text{-}mat}$ to 0.48% in $\mathrm{astro\text{-}ph}$

Resultados 2



The number on the left is the number of factor of improvements over the random prediction. i.e. the Adamic/Adar mesure is ***about 37 times more accurate*** *than the random predictor*

predictor		astro-ph	cond-mat	gr-qc	hep-ph	hep-th
probability that a random prediction is correct		0.475%	0.147%	0.341%	0.207%	0.153%
graph distance (all distance-two pairs)		9.6	25.3	21.4	12.2	29.2
common neighbors		18.0	41.1	27.2	27.0	47.2
preferential attachment		4.7	6.1	7.6	15.2	7.5
Adamic/Adar		16.8	54.8	30.1	33.3	50.5
Jaccard		16.4	42.3	19.9	27.7	41.7
SimRank	$\gamma = 0.8$	14.6	39.3	22.8	26.1	41.7
hitting time		6.5	23.8	25.0	3.8	13.4
hitting time, stationary-distribution normed		5.3	23.8	11.0	11.3	21.3
commute time		5.2	15.5	33.1	17.1	23.4
commute time, stationary-distribution normed		5.3	16.1	11.0	11.3	16.3
rooted PageRank	$\alpha = 0.01$	10.8	28.0	33.1	18.7	29.2
	$\alpha = 0.05$	13.8	39.9	35.3	24.6	41.3
	$\alpha = 0.15$	16.6	41.1	27.2	27.6	42.6
	$\alpha = 0.30$	17.1	42.3	25.0	29.9	46.8
	$\alpha = 0.50$	16.8	41.1	24.3	30.7	46.8
Katz (weighted)	$\beta = 0.05$	3.0	21.4	19.9	2.4	12.9
	$\beta = 0.005$	13.4	54.8	30.1	24.0	52.2
	$\beta = 0.0005$	14.5	54.2	30.1	32.6	51.8
Katz (unweighted)	$\beta = 0.05$	10.9	41.7	37.5	18.7	48.0
	$\beta = 0.005$	16.8	41.7	37.5	24.2	49.7
	$\beta = 0.0005$	16.8	41.7	37.5	24.9	49.7

Figure 3-3: *Performance of the basic predictors on the link-prediction task defined in Section 3.2. See Sections 3.3.1, 3.3.2, and 3.3.3 for definitions of these predictors. For each predictor and each arXiv section, the displayed number specifies the factor improvement over random prediction. Two predictors in particular are used as baselines for comparison: graph distance and common neighbors. Italicized entries have performance at least as good as the graph-distance predictor; bold entries are at least as good as the common-neighbors predictor. See also Figure 3-4.*

Chart showing the numerical results on multiple sections of the arVix coauthorship network. Different sections of arXiv yield different results.

Referencias

- Zan Huang, Hsinchun Chen, and Daniel Zeng. 2004. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Trans. Inf. Syst. 22, 1 (January 2004), 116-142.
- Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. Journal of the American society for information science and technology, 58(7), 1019-1031.
- G. Jeh and J. Widom. SimRank: A measure of structural-context similarity. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada,July 2002.
- Nguyen, P., Tomeo, P., Di Noia, T., & Di Sciascio, E. (2015, May). An evaluation of SimRank and Personalized PageRank to build a recommender system for the Web of Data. In Proceedings of the 24th International Conference on World Wide Web Companion (pp. 1477-1482). International World Wide Web Conferences Steering Committee.