دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گزارش پروژه درس شبکه‌های پیچیده

به کارگیری پیش‌بینی لینک در سیستم توصیه‌گر محصولات دیجی‌کالا

توسط:

محمد قربانی (۹۷۱۳۱۰۹۹)

استاد درس:

دکتر چهرقانی

نیم‌سال اول تحصیلی ۹۹-۹۸

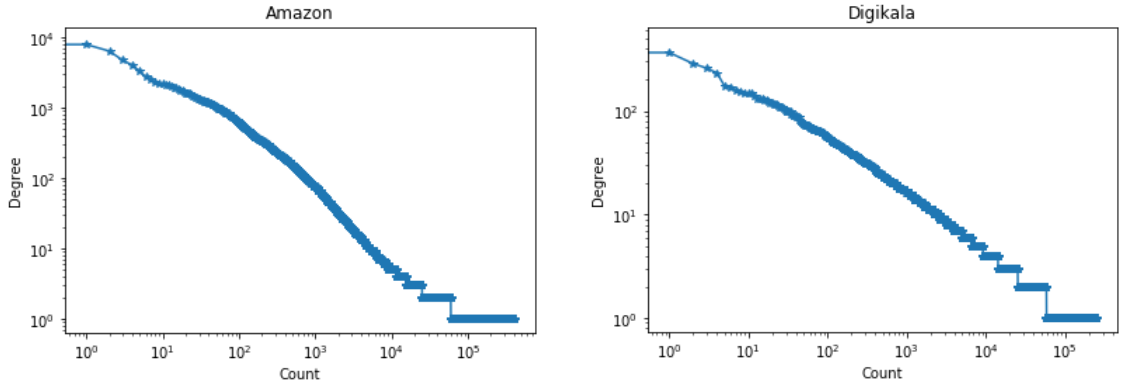
۱. مقدمه و تعریف مسئله

سیستم‌های توصیه‌گر امروزه دارای اهمیت فراوانی در بازاریابی آنلاین هم برای مشتریان و هم برای فروشندگان هستند. به طور متداول، پیشنهاد محصولات یا با استفاده از روش‌های مبتنی بر محتوا[1] که بر اساس ویژگی‌های محصولات کار می‌کند و یا با استفاده از رویکرد فیلترینگ مشارکتی[2] که بر اساس معیارهای شباهت بین کاربران/آیتم‌ها عمل می‌کند، انجام می‌شود. ما در این پروژه رویکردی شبکه‌ای نسبت به این مسئله اتخاذ خواهیم کرد؛ به این صورت که با تمرکز بر روی ویژگی‌های گرافی (گره‌ها/یال‌ها) و ساختار شبکه، پیشنهاد محصولات را انجام داده و مسئله را به صورت پیش‌بینی لینک‌های از دست رفته در شبکه‌ی محصولاتی که با یکدیگر خریداری شده‌اند پیش می‌بریم. در واقع در این پروژه ما الگوریتم‌های مختلف پیش‌بینی لینک را بر روی مجموعه دادگان مربوط به محصولات **دیجی‌کالا** و همچنین **آمازون** اجرا و تحلیل می‌کنیم. روند کاری ما برگرفته از کار شیو هوانگ [۱] و است که پیش‌بینی لینک را بر روی دادگان آمازون انجام داده است. در حالی که کارهای مرتبط زیادی هم در همین راستا صورت گرفته است [۴-۳].

۲. مجموعه دادگان

۲-۱. مرور کلی

ما از مجموعه دادگان محصولات دیجی‌کالا استفاده کردیم. این مجموعه شامل فراداده‌های مربوط به بیش از 200,000 محصول خریداری شده توسط مشتریان است و از آدرس https://www.digikala.com/opendata/#section-4 قابل دریافت است. به ازای هر سفارش خریداری شده توسط مشتری اطلاعاتی شامل ID_Order، ID_Customer، ID_Item، DateTime_CartFinalize، Amount_Gross_Order، city_name_fa و Quantity_item قابل دسترس است. همچنین در دادگان آمازون که از دسته Software استفاده کردیم، ویژگی‌های product_id، user_id و score برای بیش از 459,000 محصول خریداری شده موجود است. در شکل ۱ توزیع درجه مجموعه دادگان دیجی کالا و آمازون را مشاهده می‌کنید. همانطور که می‌بینید این نمودارها به طور طبیعی از توزیع power-law پیروی می‌کنند.



شکل ۱. توزیع درجه مجموعه دادگان قبل از پیش پردازش

۲-۲. پیش پردازش و تقسیم داده‌ها

کار پیش‌بینی لینک می‌تواند به این صورت فرموله شود که با داشتن گراف G=(V,E) که در آن V مجموعه نودها و E مجموعه یال‌های گراف هستند، فرض کنید $M = \{(i,j): i,j \in V. (i,j) \notin E\}$، ما به دنبال یافتن تابعی مانند F هستیم که $F: M \rightarrow (0.1)$ با این تفسیر که F(i,j) = 1 اگر یالی در آینده از i به j شکل گیرد و در غیر این صورت F(i,j) = 0. با توجه به اینکه ما در حال حاضر از روی مجموعه دادگان نمی‌دانیم کدام لینک‌ها وجود ندارند و در آینده شکل می‌گیرند (یعنی چند snapshot از مجموعه دادگان در زمان‌های مختلف در اختیار نداریم)، بنابراین برای ارزیابی رویکردهای متنوع، ما به صورت

[1] Content-based
[2] Collaborative filtering

تصادفی مجبوریم مجموعه‌ای از یال‌ها را از مجموعه دادگان حذف کرده و از آن‌ها برای ارزیابی رویکردها استفاده کنیم[۲] . در واقع ما به طور تصادفی مجموعه یال‌های گراف را به دو مجموعه مجزا تقسیم می‌کنیم: یکی برای آموزش و دیگری برا تست. مجموعه آموزشی از یال‌ها نشان دهنده‌ی گراف در نقطه‌ی زمانی گذشته اند و مجموعه‌ی تست ما یال‌های آینده را شبیه سازی می‌کنند. پیش از اینکه هرگونه تحلیلی انجام دهیم، ابتدا مراحل زیر را طی می‌کنیم تا داده‌ها را پیش پردازش کرده و تقسیم کنیم:

الگوریتم ۱. پیش‌پردازش داده‌ها

اگر گراف کامل اصلی را با $G = (V, E)$ نشان دهیم خواهیم داشت:

۱. چون تعداد رکورد داده‌ها بسیار زیاد است، ما فقط از یک زیرمجموعه از گراف اصلی جهت سادگی در زمان محاسبات رویکردهای مختلف استفاده می‌کنیم. از تمام نودهای V، ما از 50% آن‌ها تصادفی نمونه برداری می‌کنیم و گراف حاصل را $G_1 = (V_1, E_1)$ می‌نامیم.

۲. یک گراف دوبخشی از کاربران و محصولات ساخته و روی محصولات این گراف یک one-mode projection اجرا می‌کنیم. بین محصول i و j لینکی وجود خواهد داشت اگر این دو محصول توسط یک کاربر با هم خریداری شده باشند. گراف حاصل را $G_2 = (V_2, E_2)$ نام‌گذاری می‌کنیم.

۳. نودهای گراف G_2 با درجه کمتر از ۳ را حذف می‌کنیم. این حذف کردن به این دلیل است که نودهای با درجه‌های کوچک که دارای تعامل کمتری با دیگر نودها هستند، برای کار پیش‌بینی لینک مناسب عمل نمی‌کنند. V_{main} را مجموعه نودهای باقی مانده در نظر گرفته و زیرگراف نتیجه شده از این فرایند را $G_{main} = (V_{main}, E_{main})$ می‌نامیم.

۴. حال، تقسیم داده‌های آموزشی را انجام می‌دهیم. از میان تمام یال‌های موجود در E_{main} ما از 10% یال‌ها نمونه‌برداری کرده و آن‌ها را از گراف کنونی حذف می‌کنیم و به عنوان داده تست در نظر می‌گیریم. در حال حاضر ما دارای گراف آموزش هستیم که آن را به صورت $G_{train} = (V_{main}, E_{train})$ نام‌گذاری می‌کنیم.

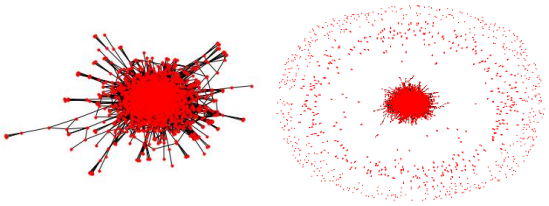
جدول زیر نشان‌دهنده‌ی تعداد نودها و یال‌ها گراف در گام‌های مختلف پیش‌پردازش داده‌ها می‌باشد و شکل زیر G_{main} را نشان می‌دهد:

	No. of Nodes	No. of Edges
G	244787	199857
G_1	78875	49989
G_2	34293	7439
G_{main}	1240	2612
G_{train}	1240	2351

جدول ۱. تعداد نودها و یال‌های گراف دیجی کالا

	No. of Nodes	No. of Edges
G	396809	450577
G_1	119552	114324
G_2	12699	11021
G_{main}	1861	8539
G_{train}	1861	7686

جدول ۲. تعداد نودها و یال‌های گراف آمازون



شکل ۲. گراف آموزش دیجی کالا و آمازون

۳. روش‌های مبتنی بر شباهت[1]

روش‌های مبتنی بر شباهت جزو بزرگترین دسته روش‌های مربوط به پیش‌بینی لینک هستند. فرض اساسی این روش‌ها این است که دو موجودیت دارای تعامل بیشتری با هم هستند اگر به هم شبیه‌تر باشند؛ بر همین مبنا یک تابع Sim(x,y) تعریف می‌شود که این تابع یک امتیاز برای هر جفت نود x و y منتسب می‌کند. ما در این پروژه از روش‌های شباهت محلی مانند از جمله Common Neighbors (CN)، Jaccard Index (JA)، Preferential Attachment (PA)، Adamic/Adar (AA) و Resource Allocation (RA) جهت اجرای عملیات پیش‌بینی لینک اجرا می‌کنیم.

[1] Similarity-based methods

ما هر روش را با استفاده از معیار Prec@K ارزیابی می‌کنیم. شبه کد زیر مراحل ارزیابی هر معیار شباهت را نشان می‌دهد:

Algorithm 2 Compute Similarity Measures

for all pair of nodes $u,v \in V_{main}$ and $u \neq v$ **do**
    if $(u,v) \in E_{train}$ **then**
        continue
    end if
    if u,v has no common neighbors **then**
        continue
    end if
    s = Sim(u,v) based on G_{train}
    Store (u,v) and s as a tuple ((u,v), s) in a list simScoreList
end for

Algorithm 3 Evaluate Similarity Measures using Prec@K

1. Sort simScoreList in descending order of similarity score
2. output the top K (u,v) pairs as our predicted edges, call this set of edges $E_{pred,K}$
3. Now compare $E_{pred,K}$ with the withheld test set E_{test}, We have Prec@K = $|E_{pred,K} \cap E_{test}|/K$

۴. تکمیل ماتریس[1]

پیش‌بینی لینک می‌تواند به عنوان یک مسئله تکمیل ماتریس شکل داده شود. تکمیل ماتریس مسئله‌ای است که در آن با داشتن تعدادی از درایه‌های یک ماتریس به دنبال یافتن مقادیر از دست رفته آن هستیم. فرض کنید یک شبکه‌ی مشاهده شده داریم که با ماتریس مجاورت A بازنمایی شده که زیرمجموعه‌ای از شبکه‌ی اصلی ما یعنی است. در این پروژه، G^* معادل G_{main} و A از G_{train} استخراج شده است، E_{train} نیز مجموعه یال‌های مشاهده شده در A و E_{test} مجموعه یال‌های مخفی هستند که ما می‌خواهیم آن ها را بازیابی کنیم. برای ارزیابی این روش نیز مانند روش‌های پیشین از Prec@K استفاده خواهیم کرد. نتایج تجربی را می‌توانید در بخش ۶ مشاهده کنید.

۵. دسته بندی دودویی با نظارت[2]

پیش از این ما با روش‌های بدون نظارت کار کردیم. کلاس دیگری از رویکردها آموزش یک دسته‌بند با نظارت روی گراف است. ما ابتدا روی یال‌های شبکه یک تقسیم آموزش-تست انجام داده تا $E_{train,+}$ و $E_{test,+}$ را به دست آوریم (یال‌های موجود الان نمونه‌های مثبت ما هستند). سپس یک مجموعه با اندازه یکسان از نمونه‌های منفی (جفت نودهایی که به هم متصل نیستند) به مجموعه آموزشی اضافه می‌کنیم تا $E_{train} = E_{train,+} \cup E_{train,-}$ به دست آید. باقی یال‌های منفی به همراه $E_{test,+}$ بخشی از مجموعه تست خواهند بود.

به ازای هر جفت نود در شبکه، ما یک بردار ویژگی استخراج خواهیم کرد. ما در اینجا از ویژگی‌های استاندارد گراف و برخی از معیارهای شباهت شامل درجه نود، درجه نود مقصد، همسایه های مشترک، معیار جاکارد و ... تا یک بردار ویژگی را تشکیل دهیم. سپس ما یک الگوریتم SVM را با هسته خطی جهت پیش‌بینی روی مجموعه تست از جفت نودها اجرا کردیم (با احتمالی که نشان‌دهنده‌ی اطمینان وجود یک یال است). سپس این پیش‌بینی‌ها را رتبه‌بندی کرده و مانند قبل Prec@K را محاسبه کردیم. همچنین از روش دسته بندی Logistic Regression (LR) نیز به عنوان یک روش دسته بندی با نظارت دیگر در کنار روش SVM استفاده کرده‌ایم.

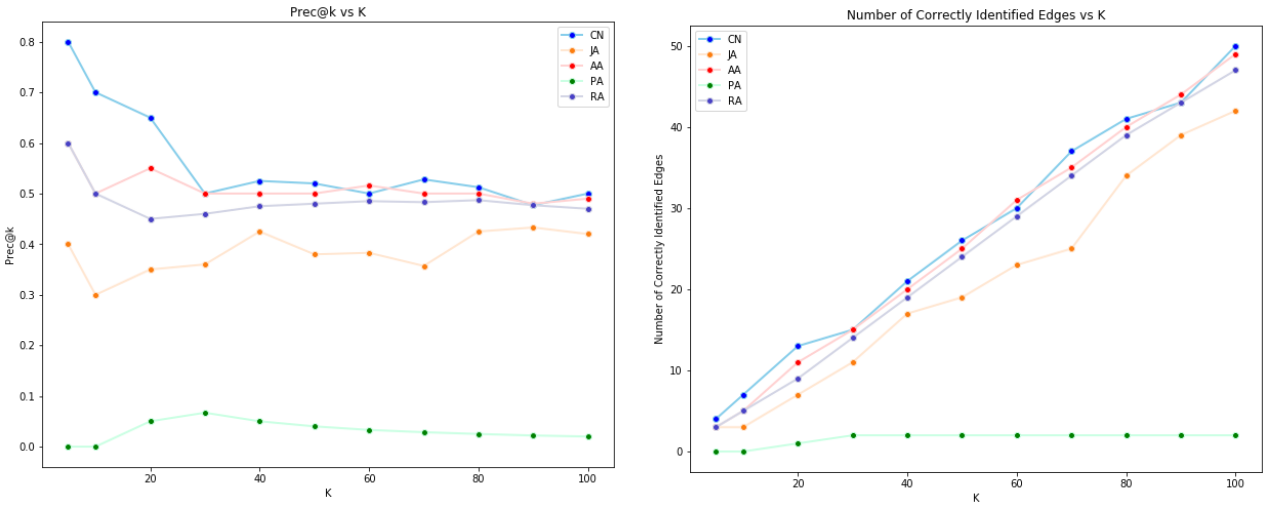
[1] Matrix completion
[2] Supervised Binary Classification

۶. نتایج تجربی[1]

در این بخش نتایج تجربی مربوط به روش‌های انجام شده جهت پیش‌بینی لینک را مشاهده خواهید کرد.

Methods	K=5	K=10	K=20	K=30	K=40	K=50	K=60	K=70	K=80	K=90	K=100
CN	0.8	0.7	0.65	0.5	0.525	0.52	0.5	0.528	0.5125	0.477	0.5
JA	0.4	0.3	0.35	0.36	0.425	0.38	0.383	0.357	0.425	0.433	0.42
AA	0.6	0.5	0.55	0.5	0.5	0.5	0.516	0.5	0.5	0.48	0.49
PA	0.0	0.0	0.05	0.067	0.05	0.04	0.033	0.285	0.025	0.022	0.02
RA	0.6	0.5	0.45	0.46	0.475	0.48	0.483	0.485	0.487	0.477	0.47

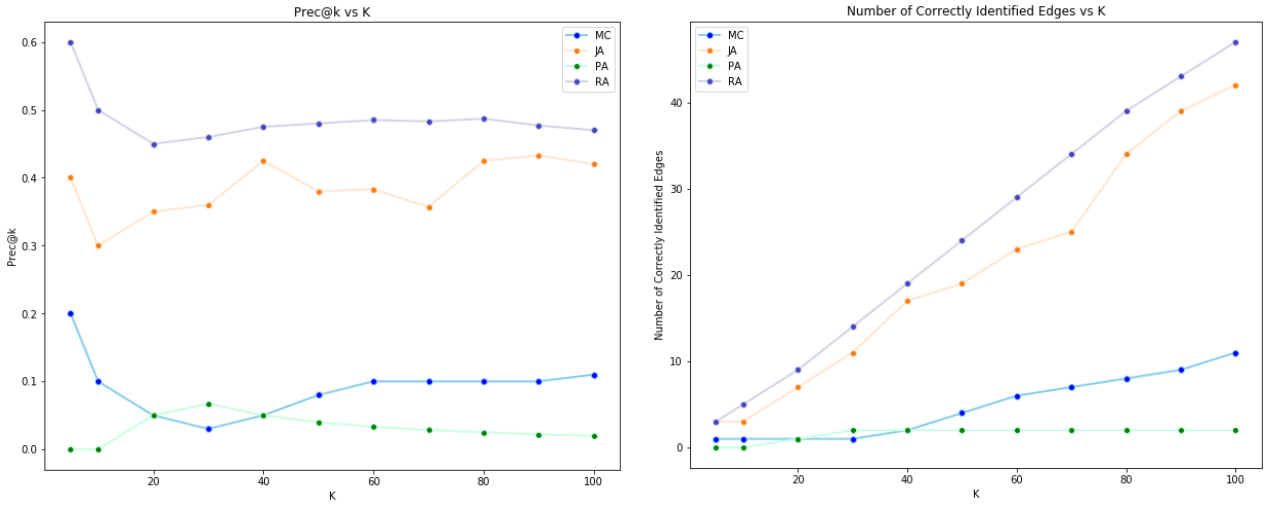
جدول ۳. Prec@K برای معیارهای شباهت محلی



شکل ۳. نتایج مربوط به روش‌های مبتنی بر معیارهای شباهت

Methods	K=5	K=10	K=20	K=30	K=40	K=50	K=60	K=70	K=80	K=90	K=100
JA	2	3	7	11	17	19	23	25	34	39	42
PA	0	0	1	2	2	2	2	2	2	2	2
RA	3	5	9	14	19	24	29	34	39	43	47
MC	1	1	1	1	2	4	6	7	8	9	11

جدول ۴. تعداد یال‌هایی که به درستی تشخیص داده شده‌اند



شکل ۴. نتایج مربوط به روش‌های مبتنی بر معیارهای شباهت

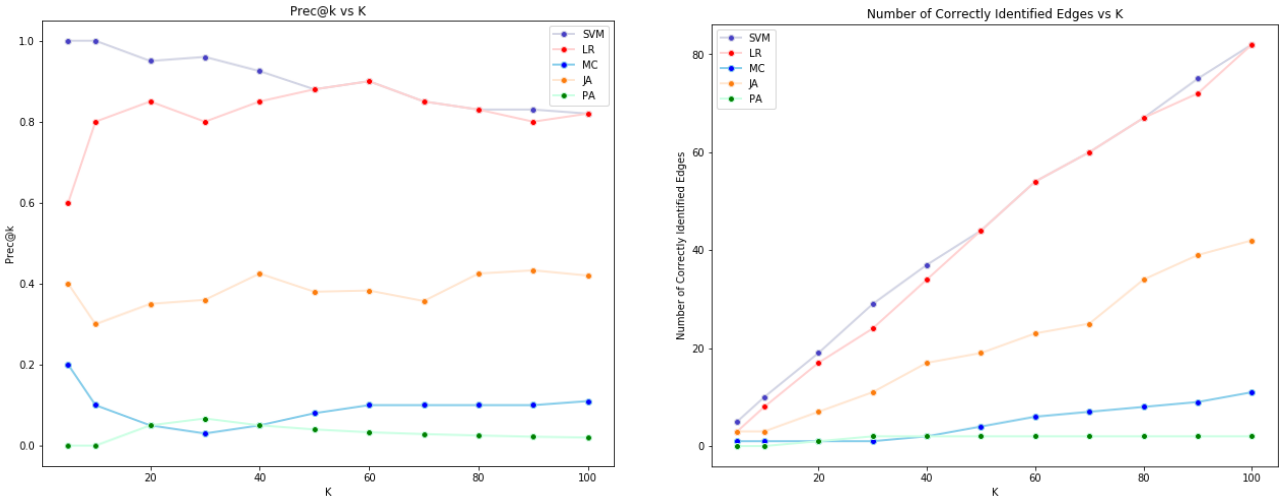
[1] Experimental Results

Methods	K=5	K=10	K=20	K=30	K=40	K=50	K=60	K=70	K=80	K=90	K=100
JA	2	3	7	11	17	19	23	25	34	39	42
PA	0	0	1	2	2	2	2	2	2	2	2
RA	3	5	9	14	19	24	29	34	39	43	47
MC	1	1	1	1	2	4	6	7	8	9	11
SVM	**5**	**10**	**19**	**29**	**37**	**44**	**54**	**60**	**67**	**75**	**82**
LR	**3**	**8**	**17**	**24**	**34**	**44**	**54**	**60**	**67**	**72**	**82**

جدول ۵. تعداد یال‌هایی که در روش‌های با نظارت به درستی تشخیص داده شده‌اند در مقایسه با دیگر روش ها

Methods	K=5	K=10	K=20	K=30	K=40	K=50	K=60	K=70	K=80	K=90	K=100
JA	0.4	0.3	0.35	0.36	0.425	0.38	0.383	0.357	0.425	0.433	0.42
PA	0.0	0.0	0.05	0.067	0.05	0.04	0.033	0.285	0.025	0.022	0.02
RA	0.6	0.5	0.45	0.46	0.475	0.48	0.483	0.485	0.487	0.477	0.47
MC	0.2	0.1	0.05	0.03	0.05	0.08	0.1	0.1	0.1	0.1	0.11
SVM	**1.0**	**1.0**	**0.95**	**0.96**	**0.925**	**0.88**	**0.90**	**0.85**	**0.83**	**0.83**	**0.82**
LR	**0.6**	**0.8**	**0.85**	**0.8**	**0.85**	**0.88**	**0.9**	**0.85**	**0.83**	**0.8**	**0.82**

جدول ۶. Prec@K برای روش‌های با نظارت در مقایسه با دیگر روش‌ها



شکل ۵. نتایج مربوط به روش‌های با نظارت در مقایسه با چند روش از دیگر روش ها

در بین معیارهای مختلفی که تست کردیم، همانطور که قابل مشاهده است معیارهای مبتنی بر شباهت ساده ای مانند Common Neighbor به خوبی اجرا می‌شوند، در حالی که بسیار ساده هم هستند. نکته ی قابل توجه روش Matrix Completion است که نسبت به روش های دیگر مطلوبیت چندانی نداشت. روش های دسته‌بندی با نظارتی مانند SVM و Logistic Regression از روش های مبتنی بر شباهت پیشی گرفته و بهترین عملکرد را در کار پیش‌بینی لینک انجام دادند. در مقایسه بین این دو روش می‌توان نتیجه گرفت به ازای افزایش داده‌ها روش LR می‌تواند از روش SVM کارایی بهتری داشته باشد.

مراجع

[1] S. Huang, "Applying Link Prediction to Amazon Product Recommendation," CS224 Project Final Report, 2018. Available from http://snap.stanford.edu/class/cs224w-2018/reports/CS224W-2018-45.pdf

[2] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In SDM06: workshop on link analysis, counter-terrorism and security, 2006.

[3] Aaron Clauset, Cristopher Moore, and Mark EJ Newman. Hierarchical structure and the prediction of missing links in networks. Nature, 453(7191):98, 2008.

[4] William Cukierski, Benjamin Hamner, and Bo Yang. Graph-based features for supervised link prediction. In Neural Networks (IJCNN), The 2011 International Joint Conference on, pages 1237-1244. IEEE, 2011.