الگوریتم‌های رتبه‌بندی بر اساس تحلیل لینک

حسن شجاعی مند
دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف

بهمن ۱۳۸۳

چکیده

موتورهای جستجو در دو مرحله صفحات مرتبط با درخواست کاربر را آماده می کنند. در مرحله اول از روشهای مختلف بازیابی اطلاعات(IR) صفحات مرتبط با پرس و جوی کاربر بدست می آیند، سپس در مرحله دوم نتایج بدست آمده بر اساس رتبه صفحات مرتب شده و به کاربر نمایش داده می شود.

رتبه‌بندی صفحات وب با استفاده از روشهای مختلفی می تواند انجام شود. یکی از روشهایی که بسیار مورد استفاده قرار گرفته الگوریتم‌های مبتنی بر تحلیل لینک است. این الگوریتم ها از اطلاعات موجود در ساختار اتصال صفحات وب برای رتبه‌بندی صفحات وب استفاده می کنند. از مشهورترین این الگوریتم‌ها PageRank و HITS می‌باشد. پس از این دو الگوریتم، الگوریتم‌های زیادی مطرح شدند که همه به نوعی از ایده‌های مطرح شده توسط این دو الگوریتم استفاده می کنند و هر کدام به نوعی درصدد رفع مشکلات این دو الگوریتم برآمده‌اند. تعداد الگوریتم‌های موجود افزایش یافته و وجود یک روش فرمال برای مقایسه الگوریتم‌های مختلف موجود، می تواند برای محققان کارآمد باشد. در این نوشته، یکی از روش‌های فرمال موجود برای بررسی خصوصیات الگوریتم‌ها و مقایسه آنها بیان شده است.

در این اواخر، علاوه بر کارهای انجام شده در خصوص بهبود الگوریتم‌های رتبه‌بندی در مواردی سعی شده است تا از ایده الگوریتم‌های رتبه‌بندی در مرتب سازی نتایج برگشتی از پایگاه داده‌ها استفاده شود. بخصوص در مورد پایگاه داده‌هایی که به نوعی ارتباط معنایی بین Object های آن وجود دارد. الگوریتم ObjectRank نمونه‌ای از این قبیل الگوریتم‌ها است که از روشی مشابه الگوریتم PageRank استفاده می‌کند.

کلمات کلیدی: الگوریتم‌های رتبه‌بندی، Ranking، PageRank، HITS، SALSA، ObjectRank

۱ مقدمه

بسیاری از موتورهای جستجو بدلیل حجم زیاد وب، برای یک پرس و جو هزاران جواب بر می‌گردانند و بدلیل آنکه کاربر وقت بررسی تمام صفحات نتایج را ندارد باید موتور جستجو نتایج بدست آمده را به گونه ای مرتب شده بر اساس اهمیت آنها به کاربر نمایش دهد. موتورهای جستجو با استفاده از روشهای رتبه‌بندی به هر صفحه وب رتبه‌ای را اختصاص داده و در هنگام نمایش نتایج جستجو صفحات را بر اساس رتبه آنها مرتب می‌کنند. کارایی موتور جستجو می‌تواند توسط فاکتورهای زیادی تحت تاثیر قرار گیرد. یکی از آنها تابعی است که موتور جستجو برای رتبه‌بندی صفحات وب استفاده می‌کند. توابع مختلف موجود برای رتبه‌بندی صفحات وب را به سه دسته کلی می توان تقسیم کرد:[۱]

- رتبه‌بندی مبتنی بر محتویات:

  این تابع رتبه‌بندی صفحات وب را بر اساس محتویات صفحه انجام می دهد. این تابع از کلمات بکار رفته در متن سند برای رتبه‌بندی آن استفاده می کند.

- رتبه‌بندی مبتنی بر لینک:

  در این روش الگوریتم‌های رتبه‌بندی از ساختار اتصال صفحات وب برای بهبود کارآیی رتبه‌بندی استفاده می کنند.

- رتبه‌بندی مبتنی بر ساختار:

  این روش بیشتر توسط موتورهای جستجوی تجاری مورد استفاده قرار می‌گیرد. آنها به کلمات داخل متن سند که در قسمتهای مختلف آن قرار گرفته باشند، وزنهای متفاوتی را می‌دهند. مثلاً Header,Anchor,Title و سپس از این ساختار وزن دهی برای بهبود کارآیی رتبه‌بندی استفاده می کنند.

الگوریتم مبتنی بر تحلیل ساختار اتصال صفحات وب، کار خود را با مجموعه ای از صفحات شروع می کند که بسته به اینکه این صفحات چگونه بدست آمده باشند، دو دسته الگوریتم مبتنی بر تحلیل لینک خواهیم داشت:

۱) الگوریتم وابسته به عبارت پرس و جو [۱]

۲) الگوریتم مستقل از عبارت پرس و جو [۲]

در بخش ۲ ، الگوریتم‌های موجود رتبه‌دهی بر اساس تحلیل لینک، مرور خواهند شد. در بخش ۳، یک چارچوب و روش فرمال برای بررسی خصوصیات مختلف الگوریتم‌های رتبه‌بندی و مقایسه آنها با یکدیگر بیان خواهد شد. یکی از موارد استفاده از الگوریتم‌های رتبه‌بندی بر اساس تحلیل لینک در بهینه کردن بازیابی اطلاعات از بانک اطلاعاتی می باشد. در بخش ۴، الگوریتم ObjectRank بررسی خواهد شد. این الگوریتم، با استفاده از ایده الگوریتم PageRank سعی می کند تا نتایج بدست آمده از جستجو در بانک اطلاعاتی را بهبود بخشد.

۲ الگوریتم‌های موجود

در این الگوریتم‌ها، از زنجیر مارکوف برای بدست آوردن رتبه صفحات استفاده می‌شود. ماتریس احتمالی انتقال اولیه [۳] زنجیر مارکوف (P) با کمک ساختار اتصال صفحات وب ایجاد می‌شود. این ماتریس را از این به بعد P خواهیم نامید. ماتریس P از روی گراف اتصال وب به گونه‌ای ساخته می شود که برای هر صفحه یک نود در گراف داریم و بین دو نود

[۱] query dependent algorithm
[۲] query independent algorithm
[۳] primitive transition probability matrix

یالی وجود دارد اگر از صفحه مربوط به نود اول لینکی به صفحه مربوط به نود دوم وجود داشته باشد. یکی از روشهای حل مسئله زنجیر مارکوف، استفاده از روش توانی [۴] است. رتبه صفحات، برداری است که زنجیر مارکوف به آن همگرا می شود. برای تضمین وجود یک بردار پایدار π^T که زنجیر به آن همگرا شود، باید زنجیر کاهش ناپذیر باشد. ماتریس در صورتی کاهش ناپذیر است که در گراف آن هر نودی از طریق هر نود دیگری قابل دسترس باشد. یک ماتریس نامنفی و کاهش ناپذیر را ماتریس اولیه گویند اگر فقط یک مقدار ویژه داشته باشد.فروبنیوس [۵] یک روش ساده برای تست اولیه بودن ماتریس ارائه کرده است: ماتریس $A \geq 0$ اولیه است اگر و فقط اگر $\exists m : A^m > 0$ این تست مفید است برای تعیین اینکه آیا روش توانی که روی ماتریس اعمال خواهد شد همگرا خواهد شد یا خیر. [۲][۳]

۱.۲ INDEGREE

استدلال ساده که تقریباً در تمام الگوریتم‌های تحلیل لینک مورد استفاده قرار می‌گیرد، این است که صفحات بر اساس میزان مشهوریت آنها رتبه‌بندی شوند. مشهوریت صفحه بوسیله تعداد صفحاتی که به صفحه لینک داده‌اند اندازه‌گیری می‌شود. این الگوریتم را الگوریتم INDEGREE گوییم. به صورت فرمال داریم: اگر به ازای هر نود i در گراف G مجموعه B(i) را به صورت زیر تعریف می کنیم:

$$B(i) = \{j : P[j,i] = 1\}$$

که در آن P ماتریس تعریف شده بر روی گراف وب می‌باشد. رتبه نود i را با a_i نشان می دهیم و خواهیم داشت: $a_i = |B(i)|$

۲.۲ PageRank

الگوریتم PageRank اولیه در سال ۱۹۹۸ توسط Page و Brin در [۵]ارائه شد. Page و Brin ایده الگوریتم INDEGREE را توسعه داده‌اند به این صورت که تمام لینکهای ورودی دارای ارزش یکسانی نیستند وصفحات معتبرتر ارزش بیشتری را به صفحه لینک داده شده، منتقل می کنند.[۴]

در [۳] از روش فرمال و جالبی برای بیان الگوریتم PageRank استفاده شده است. ساختار اتصال صفحات وب به صورت یک گراف جهتدار در نظر گرفته می‌شود. مدل مارکوف این گراف با یک ماتریس مربعی P نمایش داده می‌شود که در آن P_{ij} برابر احتمال حرکت از نود i به نود j می‌باشد. هر توزیع احتمالی مناسب می‌تواند در سطرهای این ماتریس اعمال شود. مثلاً اگر لوگ‌های استفاده از وب نشان دهد که احتمال اینکه در نود i باشیم و این نود به نودهای j و k لینک داشته باشد آنگاه احتمال اینکه به نود j برویم دو برابر احتمال آن است که به نود k برویم در این صورت می‌توان ماتریس P را به گونه ای تغییر داد که P_{ij} دو برابر P_{ik} باشد. اگر نودی وجود داشته باشد که به هیچ نودی لینک نداشته باشد تمام عناصر سطر مربوط به آن نود صفر خواهد بود و این باعث می‌شود که ماتریس P احتمالی نباشد و چون برای همگرا شدن زنجیر مارکوف یکی از شرط‌های ماتریس این است که احتمالی باشد بنابراین باید این سطر با مقادیری مقداردهی شود تا احتمالی بودن ماتریس حفظ شود. یک راه حل این است که تمام عناصر این سطر با $\frac{1}{n}$ مقداردهی شود.این مشکل در [۵] به عنوان مشکل dangling Links عنوان شده است. ماتریس P تغییر یافته را P' می‌نامیم.این اصلاح به تنهایی برای اطمینان از وجود بردار پایدار برای زنجیر مارکوف کافی نیست و باید علاوه بر احتمالی بودن ماتریس شرط کاهش ناپذیری هم برقرار باشد. ماتریس P' را با استفاده از فرمول زیر تغییر می‌دهیم: $P'' = \alpha P' + (1-\alpha)ee^T/n$ که در آن $0 \leq \alpha \leq 1$.

با اعمال تغییر فوق در ماتریس تضمین شده است که هر نودی به نود دیگری متصل شده است و بنا به تعریف زنجیر کاهش ناپذیر است. این اصلاحی که برای کاهش ناپذیر کردن زنجیر انجام شد باعث می شود که اولیه بودن ماتریس هم

[۴] power method
[۵] feronbenius

بدست آید که این شرط برای اثبات اینکه روش توانی به بردار پایدار π^T همگرا خواهد شد کافیست. بردار سطری π^T که همان بردار PageRank می‌باشد با حل مسئله بردار ویژه $\pi^T P'' = \pi^T$ قابل محاسبه است.

روش توانی یک روش حل این مسئله است. برای هر بردار شروع $x^{(0)T}$ خواهیم داشت: (این بردار شروع می‌تواند هر مقداری داشته باشد به عنوان مثال می توان قرار داد:$x^{(0)T} = e^T/n$)

$$x^{(k)T} = x^{(k-1)T}P'' = \alpha x^{(k-1)T}P' + (1-\alpha)x^{(k-1)T}ee^T/n = \alpha x^{(k-1)T}P' + (1-\alpha)e^T/n = \alpha x^{(k-1)T}P + (\alpha x^{(k-1)T}a + (1-\alpha))e^T/n$$

با محاسبات انجام شده فوق دیگر نیازی به نگهداری ماتریس‌های 'P و "P نداریم و می توان با روش توانی و استفاده از ضرب بردار—ماتریس بر روی ماتریس بسیار پراکنده P به جواب رسید. هر ضرب بردار—ماتریس که در روش توانی مورد نیاز است می تواند با انجام nnz(P) عملیات ممیز شناور[6] انجام شود. nnz(P) تعداد عناصر غیر صفر ماتریس P می‌باشد.

خوشبختانه چون ماتریس P یک ماتریس خلوت است و میانگین تعداد عناصر غیر صفر آن در هر سطر بین ۳ تا ۱۰ عنصر است پس خواهیم داشت:$O(nnz(P)) \approx O(n)$

یکی از معایب این الگوریتم آن است که این الگوریتم کل صفحات وب را در نظر می گیرد به عبارت دیگر رتبه‌بندی را مستقل از عبارت پرس و جو انجام می‌دهد. بدلیل هم حجم زیاد صفحات هم بازیابی آنها زمان زیادی لازم دارد و هم ساخت ماتریس احتمالی انتقال که این باعث می شود موتورهای جستجویی که از این الگوریتم استفاده می‌کنند، نمی‌توانند در فاصله‌های زمانی کوتاه آنرا اجرا کنند. مثلاً Google هر یک ماه یکبار الگوریتم PageRank را اجرا می کند. در صورتیکه ممکن است صفحاتی خیلی زودتر از این فاصله زمانی و گاه روزانه تغییر کنند. تحقیقات جدید بر روی روشهای بروز رسانی بردار PageRank انجام می شود تا بتوان بروز رسانی رتبه‌بندی را در زمان کمتری انجام داد. صفحاتی که PageRank کمتری دارند سریعتر به مقدار نهایی خود همگرا می شوند و بنابراین می توان در حین انجام الگوریتم PageRank آن نودهایی را که به مقدار نهایی خود همگرا شده اند را از محاسبات حذف کرد. به این الگوریتم Adaptive PageRank گویند. روشهای مختلف موجود در این زمینه در [۸] بررسی شده است. الگوریتم مبنایی Adaptive، به نقل از همین مرجع به صورت زیر می باشد:

$$\begin{aligned}
&functionAdaptivePageRank(A, z^{\circ})\\
&repeat\\
&z_N^{k+1} = A_N z^k;\\
&z_C^{k+1} = z_C^k;\\
&[N, C] = detectConverged(z^k, z^{k+1}, \epsilon);\\
&periodically, \delta = \|Az^k - z^k\|\\
&until, \delta < \epsilon
\end{aligned}$$

که در آن z_N نودهای همگرا نشده و z_C نودهایی است که همگرا شده‌اند.

در [۱۱] یکی دیگر از مشکلات الگوریتم PageRank بررسی شده است. در مدل اولیه الگوریتم مشکل dangling links مطرح شده بود و راه حل در آنجا به این صورت در نظر گرفته شده بود که می توان این نودها را حذف کرد و پس از همگرا شدن بردار PageRank آنها را به بردار اضافه کرده و سپس چند مرحله دیگر الگوریتم اجرا می شود. در این مقاله به این مشکل و راه حلهای ممکن پرداخته شده است.

۳.۲ HITS

همزمان با Page و Brin، Kleinberg یک روش دیگری را برای تعیین اهمیت صفحات وب پیشنهاد کرد.[۵] ایده Kleinberg این بود که حتماً لازم نیست که صفحات معتبر خوب به یکدیگر اشاره کنند بلکه می‌توان یکسری صفحه

[6] flops

داشت که به عنوان کانون عمل کرده و به صفحات معتبر خوب اشاره می‌کنند. در این الگوریتم ابتدا برای پرس و جوی کاربر با استفاده از یک موتور جستجوی متنی مانند AltaVista مجموعه جواب مشخص می‌شود، سپس از بین مجموعه جواب مشخص شده، k صفحه اول به عنوان مجموعه ریشه[۷] در نظر گرفته می‌شود. حال این مجموعه ریشه با اضافه کردن نودهایی که به نودهای موجود در این مجموعه لینک داده‌اند یا نودهای موجود در این مجموعه به آنها لینک داده‌اند، توسعه داده می‌شود. به این مجموعه توسعه یافته، مجموعه پایه گفته می‌شود. برای هر نود در گراف ساخته شده از مجموعه پایه دو مشخصه در نظر گرفته می‌شود، یکی مشخصه اعتبار و دیگری مشخصه کانون. صفحات وب با استفاده از این دو مشخصه به دو دسته صفحات معتبر و صفحات کانون تقسیم می‌شوند. صفحات کانون صفحاتی هستند که مشخصه کانون آنها، مقدار خوبی را دارد و صفحات معتبر صفحاتی که مشخصه اعتبار آنها، مقدار خوبی را داشته باشد. ایده در محاسبه مشخصه اعتبار و کانون صفحات این است که یک صفحه خوب صفحه‌ای است که توسط تعداد زیادی صفحه کانون خوب، به آن اشاره شده باشد و همچنین یک صفحه کانون خوب، صفحه‌ای است که به تعداد زیادی صفحه معتبر خوب، اشاره کرده باشد. این رابطه متقابل بین صفحه معتبر و کانون در محاسبه مشخصه اعتبار و کانون صفحات، در نظر گرفته می‌شود. مجموعه F(i) را برای نود i به صورت $F(i) = \{j : P_{ij} = ۱\}$ تعریف می کنیم و مشخصه کانون نود i را با h_i و مشخصه اعتبار آنرا با a_i نمایش می‌دهیم. در این صورت روابطمربوط به محاسبه مشخصه اعتبار و کانون نود i به صورت زیر خواهد بود:

$a_i = \sum_{j \in B(i)} h_j$ و $h_i = \sum_{j \in F(i)} a_j$

در معادله فوق، B(i) مجموعه نودهایی است که به نود i لینک داده‌اند که در بخش‌های قبلی، تعریف آن بیان شد. این دو عمل آنقدر تکرار می شود تا مقادیر a و h به a^* و h^* همگرا شوند. بردارهای a^* و h^* به ترتیب، بردارهای ویژه P^TP و PP^T هستند.[۷]

۴.۲ اصلاح الگوریتم HITS

گفتیم که ایده نهفته در الگوریتم HITS آن است که صفحه معتبر خوب، صفحه‌ای است که توسط تعداد زیادی صفحه کانون خوب به آن اشاره شده باشد و صفحه کانون خوب، صفحه است که به تعداد زیادی صفحه معتبر خوب، اشاره کرده باشد.این ارتباط بین وزن اعتبار و وزن کانون توسط Kleinberg با یکسری عملیات جمع بیان شد: وزن کانون صفحه، برابر است با مجموعه وزن اعتبار صفحاتی که به آنها لینک داده است و وزن اعتبار یک صفحه، برابر است با مجموع وزن کانون صفحاتی، که به این صفحه لینک داده‌اند. این نحوه تعریف دو خاصیت ضمنی را در خود دارد:[۴]

- تقارن[۸] : وزن کانون و اعتبار صفحات به یک گونه محاسبه می‌شود.
- تساوی نگری[۹] : یعنی وقتی وزن کانون یک صفحه محاسبه می‌شود، با وزن اعتبار تمام صفحاتی که به آنها لینک داده است، به طور مشابه برخورد می‌کند و همچنین در مورد محاسبه وزن اعتبار صفحه.

این دو خاصیت گاهی اوقات باعث بروز نتایج نامعقولی می‌شوند. برای طرف کردن مشکلات، راه حل‌هایی پیشنهاد شده است که در ادامه بررسی می‌شوند.

۱.۴.۲ HUBAVG

فرض کنید که گراف وب شامل دو component باشد. در اولی تعداد زیادی صفحه کانون به یک صفحه معتبر لینک داده‌اند. در دومی یک صفحه کانون به تعداد زیادی صفحه معتبر لینک داده‌اند. اگر الگوریتم HITS بر روی این گراف اجرا شود تمام وزن اعتبار را به صفحات معتبر موجود در component دوم می‌دهد و وزن اعتبار صفحه معتبر موجود در

۷ root set
۸ symmetric
۹ egalitarian

component اول صفر خواهد شد. در حالیکه ما انتظار داریم وزن صفحه معتبر موجود در component اول بیشتر از وزن صفحات معتبر موجود در دومی باشد. دلیل آن این است که صفحه کانون component دوم، به عنوان بهترین کانون در نظر گرفته می‌شود که باعث می شود صفحات معتبر موجود در component دوم، وزن اعتبار بیشتری را بدست آورند، در صورتیکه باید وزن صفحه معتبر اول باید بیشتر باشد. در این مثال خاصیت دوم یعنی یکسان نگری، باعث بروز این مشکل شده است. یعنی در محاسبه وزن کانون صفحه، وزن اعتبار تمام صفحات معتبر یکسان شرکت داشته‌اند.

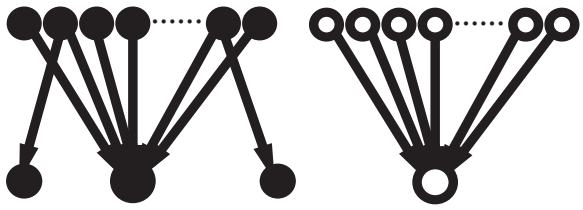
برای حل این مشکل، تغییری در الگوریتم HITS اعمال می کنیم، به این صورت که وزن کانون یک نود، به صورت میانگین وزن صفحات معتبری که به آنها لینک داده است در نظر گرفته می شود. محاسبه وزن اعتبار نودها به همان صورت انجام شده در الگوریتم HITS انجام می‌شود.

$a_i = \sum_{j \in B(i)} h_j \quad h_i = \frac{1}{|F(i)|} \sum_{j \in F(i)} a_j$

این عدم وجود تقارن، بین بهنگام‌سازی وزن کانون و وزن اعتبار بااختلاف کیفی این دو قابل توجیه است.

۲.۴.۲ AT(k)

الگوریتم HUBAVG کاستی‌هایی دارد. مثلاً به شکل ۱ توجه کنید.



شکل ۱: مشکل الگوریتم HUBAVG

در این گراف دو component داریم که کاملاً شبیه به یکدیگر هستند با این تفاوت که در component سیاه، تعدادی از کانون‌هابه تعدادی صفحه معتبر اضافه اشاره می کنند. اگر الگوریتم HUBAVG برای این گراف اجرا شود، صفحه معتبر سفید از صفحات معتبر سیاه وزن بیشتری بدست خواهد آورد. دلیل آن این است که الگوریتم HUBAVG به کانون هایی که فقط به صفحات معتبر خوب اشاره می‌کنند، ارزش بیشتری می‌دهد و ارزش صفحات کانونی که به صفحات معتبر بدی هم اشاره می‌کنند را کاهش می‌دهد. درحالیکه منطقی به نظر می‌رسد که صفحه معتبر سیاه باید ارزش یکسانی با صفحه معتبر سفید داشته باشد.

یک راه حل ساده برای این مشکل، آن است که از یک آستانه اعتبار[۱۰] استفاده شود. ایده در این الگوریتم این است که یک صفحه، کانون خوبی است اگر به حداقل k صفحه معتبر خوب اشاره کند. این الگوریتم وزن صفحه معتبر را برابر مجموع وزن k صفحه معتبر با ارزش بالاتر، قرار می‌دهد که این کانون به آنها اشاره می‌کند. مقدار k به عنوان پارامتر به الگوریتم داده می‌شود. به صورت فرمال $F_k(i)$ زیر مجموعه‌ای از F(i) است که شامل k نود باارزش تر F(i) می‌باشد. یعنی برای هر نود $p \in F(i)$ که $p \notin F_k(i)$، خواهیم داشت $a_p \leq a_q$ برای هر $q \in F_k(i)$.

اگوریتم آستانه اعتبار، وزن کانون و اعتبار صفحات را به صورت زیر محاسبه می‌کند:

$a_i = \sum_{j \in B(i)} h_j \quad h_i = \sum_{j \in F_k(i)} a_j$

واضح است که اگر d_{out} ماکزیمم درجه خروجی نودهای گراف وب باشد، این الگوریتم همان الگوریتم HITS است.

در [۷] علاوه بر این روش، الگوریتم‌های مشابه‌ای ارائه شده است که در آنها همین روش محاسبه برای وزن اعتبار صفحات، در نظر گرفته شده است (Authority Threshold و Full Threshold).

[۱۰] Authority Threshold

۳.۴.۲ الگوریتم MAX

الگوریتم MAX یک حالت خاص از الگوریتم آستانه اعتبار است وقتی که مقدار k برابر ۱ باشد. ایده در این الگوریتم این است که یک کانون به خوبی بهترین صفحه معتبری است که به آن اشاره می کند. این الگوریتم یکسری خصوصیات جالبی دارد. یکی اینکه نودهایی که وزن بیشتری بدست آورده‌اند، نودهایی هستند که درجه ورودی بالاتری داشته‌اند. به این نودها، نودهای seed گفته می شود.

۵.۲ SALSA

این الگوریتم توسط Lampel و Moran پیشنهاد شد.[۹] در الگوریتم HITS گراف G را به صورت یک گراف دو بخشی تصور می کنیم که کانون ها به صفحات معتبر اشاره می کنند. الگوریتم SALSA یک پیمایش تصادفی را بر روی گراف دوبخشی بین کانون ها و صفحات معتبر انجام می‌دهد. در ابتدا با یک صفحه معتبر که به صورت تصادفی انتخاب می شود شروع می کند. وقتی که در یک صفحه معتبر هستیم، الگوریتم یک لینک ورودی را به صورت تصادفی انتخاب می کند و به یک کانون می رود. و وقتی که در یک کانون است یک لینک خروجی را به صورت تصادفی انتخاب کرده و به یک صفحه معتبر می رود. ماتریس احتمالی انتقال زنجیر مارکوف متناظر به صورت زیر خواهد بود:

$$P_a(i,j) = \sum_{k:k\in B(i)\cap B(j)} \frac{1}{|B(i)|}\frac{1}{|F(k)|}$$

گراف $G_a(A, E_a)$ گراف اعتبار نامیده می شود که نودهای آن صفحات معتبر هستند و یک یال بین دو نود وجود دارد اگر کانون مشترکی به آنها اشاره کند.

زنجیر مارکوف این الگوریتم، متناظر با پیمایش تصادفی گراف اعتبار است که در آن $P_a(i,j)$ برابر احتمال حرکت از صفحه معتبر i به صفحه معتبر j است. W_r را ماتریس مشتق شده از ماتریس W در نظر بگیرید که به صورت سطری نرمال سازی شده است که مجموع عناصر آن ۱ باشد و W_c ماتریس متناظر با W که به صورت ستونی نرمال سازی شده است. در این صورت توزیع پایدار الگوریتم SALSA، بردار ویژه چپ ماتریس $W_c^T W_r$ است.

۱.۵.۲ الگوریتم BFS

در این الگوریتم ایده الگوریتم های SALSA و HITS با هم ترکیب شده‌اند. در الگوریتم SALSA میزان مشهوریت صفحه تنها در میان همسایه های بالافصل آن در نظر گرفته می‌شود و بقیه گراف در نظر گرفته نمی شود، در طرف مقابل الگوریتم HITS تمام گراف را برای محاسبه رتبه یک نود در نظر می‌گیرد. الگوریتم BFS[۱۱] مشهوریت یک نود را در میان همسایه‌هایی که با فاصله n با این نود همسایه هستند در نظر گرفته می شود.

$(BF)^n(i)$ مجموعه نودهایی است که از طریق نود i با دنبال کردن مسیرهای $(BF)^n$ بدست می آیند. وزن نود i در این الگوریتم به صورت زیر محاسبه می شود:

$$a_i = ۲^{n-۱}|B(i)| + ۲^{n-۲}|BF(i)| + ۲^{n-۳}|BFB(i)| + ... + |(BF)^n(i)|$$

۳ چارچوبی برای مطالعه الگوریتم‌های رتبه‌بندی

بعد از ارائه دو الگوریتم PageRank و HITS ، الگوریتم های زیادی برای رتبه‌بندی صفحات وب ارائه شده است و این باعث شده تا انتخاب یکی از الگوریتم‌ها به سادگی امکان پذیر نباشد. مثلاً بررسی اینکه دو الگوریتم رتبه بندی چقدر بهم شبیه هستند؟ برای چه گراف‌هایی دو الگوریتم،رتبه‌بندی مشابهی تولید می کنند؟آیا مجموعه‌ای از خصوصیات وجود دارد که الگوریتم را توصیف کند؟ بنابراین وجود یک روش بررسی خصوصیات و ویژگیهای الگوریتم‌های موجود و

[۱۱] Backward forward Steps

امکان مقایسه آنها الزامی به نظر می‌رسد.در [۴] روشی برای معرفی یک چارچوب بررسی الگوریتم ها عنوان شده است. برای تعریف یک چارچوب بررسی الگوریتم ها ابتدا باید یک تعریف فرمال از الگوریتم رتبه‌بندی مبتنی بر لینک داشته باشیم. فرض کنید که $\wp_n$ مجموعه تمام گرافهایی باشد که اندازه آنها n باشد.تعریف می کنیم که $\overline{\wp_n} \subseteq \wp_n$ آنگاه الگوریتم رتبه‌بندی A را به صورت تابع زیر می توان تعریف کرد:

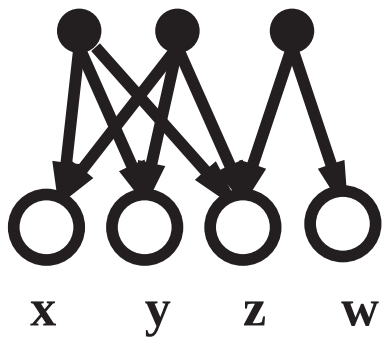
$A : \overline{\wp_n} \to R^n$

این تابع یک گراف $G \in \overline{\wp_n}$ رابه یک بردار n بعدی اعداد حقیقی نگاشت می‌کند.بردار A(G) بردار وزن اعتبار تولید شده توسط الگوریتم A برای گراف G است. این بردار را بردار LAR یا بردار اعتبار گویند.

۱.۳ یکنواختی

اولین خصوصیتی که برای الگوریتم‌های LAR در نظر گرفته می شود، خاصیت یکنواختی[۱۲] است. یکنواختی یعنی اگر همه کانون‌هایی که به نود j اشاره می کنند به نود k هم اشاره کنند، آنگاه باید وزن اعتبار حداقل برابر وزن اعتبار j داشته باشد. به صورت فرمال آنرا به صورت زیر تعریف می کنیم:

تعریف: الگوریتم A، بر روی کلاسی از گراف‌های $\overline{\wp_n}$ یکنواخت است اگر برای هر گراف $G \in \overline{\wp_n}$ و هر جفت نود j و k در این گراف G اگر $B(j) \subseteq B(k)$ آنگاه $A(G)[j] \leq A(G)[k]$. یکنواختی یک خصوصیت معقول برای الگوریتم‌هاست، اما می توان الگوریتم‌های معقولی هم داشت که‌خاصیت یکنواختی را ندارند.



شکل ۲: عدم وجود خاصیت یکنواختی در الگوریتم AuthorityAVG

در گراف شکل ۲ داریم که $B(x) \subset B(z)$ و $B(y) \subset B(z)$. در صورت اجرای الگوریتم AuthorityAVG، نودهای x و y وزن بیشتری از نود z بدست می آورند. یک الگوریتم دیگر که خاصیت یکنواختی را ندارد الگوریتم HUBTHESHOLD می باشد.

تئوری : الگوریتم‌های INDEGREE,HITS,PageRank,SALSA,HUBAVG,AT(k),BFS همگی یکنواخت هستند.

اثبات:فرض کنید که j و k دو نود در گراف G باشند به گونه‌ای که $B(j) \subseteq B(k)$.در الگوریتم INDEGREE یکنواختی واضح است، چون وزن های اعتبار برابر درجه ورودی آنهاست و درجه ورودی j کمتراز درجه ورودی k است. همین خصوصیت در الگوریتم SALSA در هر component وجود دارد.

در الگوریتم PageRank اگر a_j و a_k وزن نودهای j و k باشند، خواهیم داشت:

$a_j = \sum_{i=1}^{n} MPR[i,j]a_i \quad a_k = \sum_{i=1}^{n} MPR[i,k]a_i$

که در آن M_{PR} ماتریس الگوریتم PageRank است. در این ماتریس با توجه به اینکه $B(j) \subseteq B(k)$ پس به ازای تمام i ها $M_{PR}[i,j] \leq M_{PR}[i,k]$ و بنابراین خواهیم داشت: $a_j \leq a_k$ و تئوری اثبات است.

[۱۲] monotonicity

برای الگوریتم‌های HITS,HUBAVG,AT(k) چون در هر مرحله t رابطه زیر برقرار است:

$\overline{a}_j^t = \sum_{j \in B(j)} h_i \leq \sum_{j \in B(k)} h_i = \overline{a}_k^t$

و بنابراین $a_j \leq a_k$. البته رابطه فوق برای وزن نودها قبل از مرحله نرمال سازی است ولی از آنجاییکه نرمال سازی می تواند باعث همگرا شدن وزن ها به صفر شود ولی هیچگاه شرایطی بوجود نخواهد آمد که پس از مرحله نرمال سازی داشته باشیم: $a_j \leq a_k$ پس نرمال سازی مشکلی را بوجود نخواهد آورد.

در الگوریتم BFS چون $B(j) \subseteq B(k)$ پس هر نودی که از طریق نود j قابل دسترس باشد از طریق نود k هم قابل دسترس است و بنابراین $a_j \leq a_k$.

۲.۳ اندازه گیری فاصله دو بردار LAR

فرض کنید که a_1 و a_2 دو بردار LAR باشد که بر روی گراف یکسانی تعریف شده باشند. فاصله دو بردار a_1 و a_2 را به صورت $d(a_1, a_2)$ تعریف می کنیم که d تابعی است به صورت $d: R^n \times R^n \to R^n$.

چند تابع مختلف برای اندازه گیری فاصله دو بردار وجود دارد که در ادامه دو روش را بررسی می کنیم.

۱.۲.۳ فاصله هندسی

بردارهای LAR می توانند به صورت نقاطی در فضای n بعدی در نظر گرفته شوند و در این صورت می توان از اندازه گیری های معمول فاصله هندسی استفاده کرد.

$d_1(a_1, a_2) = min_{\gamma_1, \gamma_2 \geq 0} \sum_{i=1}^{n} |\gamma_1 a_1(i) - \gamma_2 a_2(i)|$

که در آن γ_1, γ_2 ثابت‌هایی هستند که برای از بین بردن خطایی که ممکن است پس از نرمال سازی در فاصله دو بردار بوجود آید، بکار می روند.مثلاً اگر دو بردار مورد نظر به صورت $v = (1, 1, ..., 1)$ و $w = (1, 1, ..., 2)$ آنگاه اگر از نرم L_∞ برای نرمال سازی استفاده شود خواهیم داشت که $\sum_{i=1}^{n} |w_\infty(i) - v_\infty(i)| = (n-1)/2 = \theta(n)$.یعنی ماکزیمم فاصله دو بردار از $\theta(n)$ است. حال اگر از نرم L_1 برای نرمال سازی استفاده شود خواهیم داشت: $\sum_{i=1}^{n} |w_1(i) - v_1(i)| = \frac{2(n-1)}{n(n+1)} = \theta(1/n)$.

بر خلاف محاسبه اولی، در دومی نزدیکی دو بردار نشان داده است. از ثابتهای γ_1, γ_2 برای جلوگیری از بروز این اختلافات استفاده می‌شود.

۲.۲.۳ فاصله رتبه

این تابع میزان شباهت رتبه‌بندی های تولید شده توسط دو بردار LAR را مشخص می کند. فرض کنید که a_1 و a_2 دو بردار LAR باشند که بر روی گراف G تعریف شده‌اند. اگر تمام وزن‌ها با یکدیگر متفاوت باشد آنگاه رتبه‌بندی تولید شده یک مجموعه ترتیب کلی بر روی گراف G می باشد. به رتبه‌بندی کلی، جایگشت[۱۳] می گوییم. فاصله رتبه را تعداد اختلاف بین دو رتبه‌بندی تعریف می کنیم.

اگر P مجموعه تمام جفت نودها در گراف G باشد و a_1 و a_2 دو بردار LAR مورد نظر، مجموعه ناقض[۱۴] به صورت زیر تعریف می شود:

$v(a_1, a_2) = \{(i,j) \in \mathrm{P} : (\mathrm{a_1(i) < a_1(j) \wedge a_2(i) > a_2(j)}) \vee (\mathrm{a_1(i) > a_1(j) \wedge a_2(i) < a_2(j)})\}$

این مجموعه جفت نودهایی است که ترتیب رتبه‌بندی آنها در دو جایگشت با هم متفاوت است.

یک تابع را با استفاده از مجموعه فوق به صورت زیر تعریف می کنیم:

---

۱۳ permutation

۱۴ violating set

$$\mathrm{I_{a_1a_2}(i,j)} = \begin{cases} ۱ & if\ (i,j) \in v(a_۱, a_۲) \\ ۰ & otherwise \end{cases}$$

حال عدد $\sum_{(i,j)\in \mathrm{P}} \mathrm{I_{a_1a_2}(i,j)}$ نشان دهنده فاصله رتبه دو بردار است که برابر خواهد بود با تعداد swap های لازم در الگوریتم bubble sort تا دو جایگشت به یکدیگر تبدیل شوند. بزرگترین مقدار آن می تواند $n(n-۱)/۲$ باشد که آن وقتی است که دو رتبه‌بندی عکس یکدیگر باشند.

تعریف فوق در صورتی معتبر است که در دو بردار وزن مشابه نداشته باشیم. در صورت داشتن وزن نود مشابه در بردارها به احتمال وقوع آن زیاد است، الگوریتم یک مجموعه ترتیب حزئی بر روی نودهای گراف ایجاد خواهد کرد. و در این صورت یک مجموعه دیگر هم به موارد اختلاف رتبه‌بندی باید اضافه شود. این مجموعه را مجموعه ناقض ضعیف[۱۵] نامیده و آنرا به صورت زیر تعریف می کنیم:

$$\mathrm{W(a_1, a_2) = \{(i,j) \in P : (a_1(i) = a_1(j) \wedge a_2(i) \neq a_2(j)) \vee (a_1(i) \neq a_1(j) \wedge a_2(i) = a_2(j))\}}$$

هر جفت نودی که در مجموعه فوق باشد را با عددی مانند p جریمه می کنیم که $p \in [۰, ۱]$. تابع زیر را بر روی این مجموعه و مجوعه ناقض که قبلاً تعریف شد، تعریف می کنیم:

$$\mathrm{I^p_{a_1a_2}(i,j)} = \begin{cases} ۱ & if\ (i,j) \in v(a_۱, a_۲) \\ p & if\ (i,j) \in \mathrm{W(a_1, a_2)} \\ ۰ & otherwise \end{cases}$$

حال عدد $\sum_{(i,j)\in \mathrm{P}} \mathrm{I^p_{a_1a_2}(i,j)}$ نشان دهنده فاصله رتبه دو بردار خواهد بود. در صورتیکه $p = ۰$ باشد یعنی برای رتبه‌بندی ناسازگار جریمه‌ای در نظر گرفته نشده است و آنرا روش ملایم[۱۶] گوییم. اگر $p = ۱$ باشد یعنی بیشترین جریمه برای رتبه‌بندی ناسازگار در نظر گرفته شده‌است و آنرا روش سخت[۱۷] گوییم.

فاصله رتبه ضعیف را به صورت زیر تعریف می کنیم:

$$d_r^{(۰)}(a_۱, a_۲) = \frac{۱}{n(n-۱)/۲} \sum_{(i,j)\in \mathrm{P}} \mathrm{I^0_{a_1a_2}(i,j)}$$

و فاصله رتبه شدید را به صورت زیر:

$$d_r^{(۱)}(a_۱, a_۲) = \frac{۱}{n(n-۱)/۲} \sum_{(i,j)\in \mathrm{P}} \mathrm{I^1_{a_1a_2}(i,j)}$$

فاصله رتبه، با عدد $n(n-۱)/۲$ نرمال می شود. این عدد همانطور که در بالا گفته شد بزرگترین عددی است که دو رتبه‌بندی با هم اختلاف دارند.

هنگامیکه بخواهیم اثبات کنیم دو رتبه‌بندی خیلی با هم اختلاف دارند از فاصله رتبه‌بندی ضعیف استفاده می کنیم و هنگامیکه بخواهیم اثبات کنیم دو رتبه بندی به یکدیگر نزدیک هستند از فاصله رتبه‌بندی شدید.

۳.۲.۳ مشابهت دو الگوریتم

با استفاده از توابع فاصله تعریف شده در قسمت قبلی، می توان تعاریفی را برای مشابهت دو الگوریتم تعیین کرد.

تعریف: دو الگوریتم $A_۱$ و $A_۲$ که نرمال سازی را با نرم L_p انجام می دهند و $۱ \leq p \leq \infty$ بر روی یک کلاس از گرافها($\wp_n$) مشابه هستند اگر وقتی $n \to \infty$ داشته باشیم:

$$max_{G\in\wp_n} d_۱(A_۱(G), A_۲(G)) = O(n^{۱-۱/p})$$

تعریف: دو الگوریتم $A_۱$ و $A_۲$ بر روی یک کلاس از گرافها($\wp_n$) مشابهت ضعیف رتبه‌بندی دارند اگر وقتی $n \to \infty$ داشته باشیم:

$$max_{G\in\wp_n} d_r^{(۰)}(A_۱(G), A_۲(G)) = O(۱)$$

و به همین صورت برای مشابهت شدید بین دو الگوریتم رتبه‌بندی نیز می توان تعریف مربوطه را انجام داد.

در [۴] اثبات شباهت و عدم شباهت الگوریتم‌های مختلف رتبه‌بندی با استفاده از تعریف فوق آمده است.

---

۱۵ weakly violating set

۱۶ lenient approach

۱۷ strict approach

در [۱۰] روشی به نام ObjectRank مطرح شده است. ObjectRank در پایگاه داده‌هایی که جریان طبیعی از اعتبار بین Object های آن وجود دارد، برای جستجوی کلمات کلیدی از روشی مشابه الگوریتم رتبه‌بندی PageRank استفاده کرده و نتایج برگشتی را بر اساس اهمیت مرتب می‌کند. در ابتدا ساختار پایگاه داده‌ها به صورت یک گراف مدل می شود. مثلاً اگر پایگاه داده‌های مربوط به مقالات را در نظر بگیریم، در جدول مقاله هر مقاله به کنفرانسی در جدول کنفرانس و نویسنده‌ای در جدول نویسنده و همچنین به خود جدول مقاله برای مشخص کردن مقالات مرجع لینک دارد. با استفاده از مدل گردشگر تصادفی الگوریتم را می توان به صورت زیر بیان کرد:

ابتدا مجموعه Object هایی که کلمه مورد نظر کاربر را دارند مشخص می‌شوند. گردشگر تصادفی با شروع از یک نود، گراف پایگاه داده را به این صورت پیمایش می کند که یا به یکی از نودهایی که نود فعلی به آن متصل است می رود یا به صورت تصادفی از بین مجموعه نودهای موجود نودی را انتخاب کرده و به آن پرش می کند. احتمال آنکه یک پویش اتفاق افتد به چند فاکتور بستگی دارد که نوع یال را شامل می شود(استفاده از نوع یال ایده جدیدی است که در این الگوریتم بکار رفته است).حال حالات مختلف پویش گردشگر و احتمال آنها را بررسی می کنیم. اگر گردشگر در نود مقاله باشد با احتمال α یکی از لینک‌های موجود در مقاله را دنبال خواهد کرد. لینک‌های موجود در مقاله عبارتند از لینک به نویسنده، لینک به مرجع و لینک به کنفرانس. احتمال حرکت از طریق این لینک‌ها با یکدیگر متفاوت است. احتمال آنکه گردشگر از طریق یال مرجع حرکت کند را ۷۰ درصد، حرکت از طریق یال نویسنده را ۲۰ درصد و حرکت از طریق یال کنفرانس را ۱۰ درصد در نظر می گیریم. این درست مانند آن است که در الگوریتم PageRank ماتریس احتمالی انتقال را بر اساس نیاز تغییر دهیم. ماتریس زنجیر مارکوف ساخته می‌شود و با بدست آوردن بردار ویژه پایدار آن ObjectRank مربوط به Object های پایگاه بدست خواهد آمد.

۵ نتیجه گیری

الگوریتم‌های پایه‌ای ارائه شده در سال ۱۹۹۸ برای رتبه‌بندی صفحات بر اساس تحلیل لینک، ایده اصلی الگوریتم‌هایی است که بعد از این دو الگوریتم در این زمینه مطرح شده است. پس از ارائه الگوریتم های PageRank و HITS ، پایه در کارهای انجام شده بعدی، بر ایده‌های مطرح شده در این دو الگوریتم قرار داده شده است و سعی کرده اند با انجام اصلاحاتی در این الگوریتم‌ها، آنها را بهبود بخشند. نتیجه این تحقیقات الگوریتم‌های مختلف زیادی است که هر یک مزایا و معایبی دارند. با افزایش الگوریتم‌ها، تحقیقاتی در خصوص ایجاد روشهایی برای اندازه گیری ویژگیهای الگوریتم‌های مختلف و مقایسه آنها با یکدیگر انجام شده است که یک نمونه از کار انجام شده در این نوشته بررسی شد. ایده رتبه‌بندی نتایج جستجو، مخصوصاً الگوریتم PageRank کاربردهایی در مواردی بجز رتبه‌بندی صفحات وب ایجاد کرده است که از جمله آنها الگوریتم ObjectRank می باشد که سعی کرده است، از ایده الگوریتم PageRank برای مرتب کردن نتایج جستجوی پایگاه داده های خاص استفاده کند.

در تحقیقات جدید، به نظر می‌رسد که توجه به استفاده از روشهایی دیگر در کنار روشهای تحلیل لینک مثل تحلیل محتویات صفحات بیشتر شده است.

مراجع

[1] Weiguo Fan, Michael D. Gordon, Praveen Pathak, Wensi Xi and Edward A. Fox. Ranking Function Optimization For Effective Web Search By Genetic Programming: An Emprical Study. Proceeding of 37th Hawaii International Conference on Sytsem Science, 2004.

[2] William J.Stewart. Introduction to the Numerical Solution of Markov Chains. Princeton University Press, 1994

[3] Amy N. Langville and Carel D. Meyer. Deeper Inside PageRank. Technical Report, NCSU Center for Res. Sci Comp, 2003.

[4] Allan Brodin, Gareth O. Roberts, Jeffrey S. Rosenthal, Panayiotis Tsaparas. Links Analysis Ranking Algorithms, Theory, and Experiments. March 25, 2004.

[5] S. Brin, L. Page, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web.Technical report.Stanford Digital Library Technologies Project, 1998.

[6] J. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of ACM, 1999.

[7] Dr. Bruce Litow. A Review of World Wide Web searching techniques, focusing on HITS and related algorithms that utilise the link topology of the World Wide Web to provide the basis for a structure based search topology. June 2001.

[8] Sepandar Kamvar, Taher Haveliwala,Gene Golub. Adaptive Methods for the Computation of PageRank. Technical Report, April 2003.

[9] R. Lampel,S. Moran. The stochastic approach for link structure analysis(SALSA) and the TDK effort. Proc. 9th International World Wide Web Conference, 2000.

[10] Andrey Balmin, Vagelis Hristidis, Yannis Papakonstantinou. ObjectRank: Authority-Based Keyword Search in Databases. Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.

[11] Nadav Eiron, Kevin S. McCurley, John A. Tomlin. Ranking the Web Frontier. In Proceeding of the 13th Conference on World Wide Web, 2004.