Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
Volume 8, Number 3, Fall 2019, pp. 52-59
Received: 02/18/2019, Revised: 07/11/2019, Accepted: 07/21/2019

PageRank mutation

Javad Paksima
Department of Computer, Payame Noor University, Iran.
Paksima@pnu.ac.ir

Corresponding author's address: Javad Paksima, Faculty of Computer Engineering, Payame Noor University, Tehran, Iran.

Abstract- The PageRank algorithm is one of the web-based classification methods used by Google search engine first. The main purpose of this algorithm was to determine the popularity of Web pages. The algorithm uses the web links structure to find important pages. One of the problems of PageRank and the same algorithms based on the web graph is that the number of a page is propagated to its output pages without any control while the output screen really is not really recommended by the previous page directly or indirectly. In this paper, by changing the original formula PageRank, a method has been proposed to prevent the entry of the input bonds to a single page of publication without gaining popularity. In order to evaluate the proposed algorithm, a single web graph is constructed, which in some nodes has a rating leap. This mutation decreases after the proposed algorithm is applied.
Keywords- Ranking, PageRank, Web-Graph.

مجله علمی پژوهشی رایانش نرم و فن‌آوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
جلد ۸، شماره ۳، پاییز ۱۳۹۸، صفحه ۵۹-۵۲
دریافت: ۱۳۹۷/۱۱/۲۹، بازنگری: ۱۳۹۸/۰۴/۲۰، پذیرش: ۱۳۹۸/۰۴/۳۰

جهش رتبه در PageRank

جواد پاک سیما
گروه کامپیوتر و فناوری اطلاعات، دانشگاه پیام نور، ایران.
paksima@pnu.ac.ir

* نشانی نویسنده مسئول: جواد پاک‌سیما، تهران، دانشگاه پیام نور، گروه کامپیوتر و فناوری اطلاعات.

چکیده- الگوریتم PageRank یکی از روش‌های رتبه‌بندی مبتنی بر گراف وب است که اولین بار توسط موتور جستجوی گوگل مورد استفاده قرار گرفت. هدف اصلی این الگوریتم مشخص کردن میزان محبوبیت صفحات وب بود. این الگوریتم برای پیدا کردن صفحات مهم از ساختار پیوندها در وب استفاده می‌کند. یکی از مشکلات PageRank و الگوریتم‌های مشابه مبتنی بر گراف وب، آن است که رتبه‌ی یک صفحه به صفحات خروجی آن بدون هیچ کنترلی انتشار می‌یابد درحالی‌که ممکن است صفحه خروجی واقعاً توصیه شده توسط صفحه قبلی بطور مستقیم یا غیرمستقیم نباشد. در این مقاله با تغییر فرمول اصلی PageRank روشی ارائه گردیده است تا با نرمال‌سازی پیوندهای ورودی به یک صفحه از انتشار بدون کنترل محبوبیت جلوگیری به عمل آید. همچنین به منظور ارزیابی الگوریتم ارائه شده یک گراف وب نمونه ساخته شده است که در بعضی از گره ها دارای جهش رتبه است. این جهش رتبه بعد از اعمال الگوریتم پیشنهادی کاهش می‌یابد.

واژه‌های کلیدی: رتبه بندی، PageRank، گراف وب.

۱- مقدمه

اکثر موتورهای جستجو از الگوریتم‌هایی استفاده می‌کنند تا براساس گراف وب به صفحات امتیازدهی کنند. پیوندها بیان کننده‌ی کیفیت محتوای یک صفحه از منظر صفحات بیرونی هستند. این درحالی است که محتوای متنی صفحه کاملاً به ایجاد کننده‌ی آن وابسته است. متن پیوند معمولاً در بردارنده‌ی توصیفی از یک صفحه توسط صفحات دیگر می‌باشد. به عبارت دیگر در رتبه‌بندی براساس پیوند از محتوای صفحات دیگر برای ارزیابی یک صفحه استفاده می‌شود. اکثر روش‌های مبتنی بر گراف با این فرض طراحی شده‌اند که پیوندها توسط شخصی غیراز طراح صفحه ایجاد شده است و هدف از آن توصیه‌ی صفحه مورد نظر می‌باشد؛ ولی همیشه این‌گونه نیست.

در [۱] سه مشکل در استفاده از پیوندها بصورت زیر مطرح شده است:

- دو یا چند پیوند ممکن است از یک وب سایت به یک سایت ایجاد شده باشد.
- دو یا چند پیوند ممکن است از دو سایت مشابه، به یک سایت ایجاد شده باشد. در این حالت نیز نباید دو پیوند درنظر گرفت.
- برخی از پیوندها بصورت غیرواقعی برای صفحات اسپم ایجاد می‌شوند تا رتبه‌ی آن‌ها در موتورهای جستجو بالا ببرند.

در [۲]، *Wang* و همکارانش مشکل شکاف صفر و یک[۱] در *PageRank* را مطرح کردند. در *PageRank* این‌گونه فرض شده است که موج‌سوار تصادفی[۲] صفحات را یکی پس از دیگری مشاهده می‌کند. موج سوار باید در هر صفحه با یک احتمال (d) یکی از

لینک‌های خروجی را دنبال کند و یا این که با یک احتمالی (*1-d*) به صفحه دیگری پرش می‌کند. در اکثر روش‌ها احتمال *d* مقدار ثابتی مثلاً ۰/۸۵ در نظر گرفته می‌شود. یکی از مشکلات *PageRank* صفحات معلق[۳] می‌باشد. صفحات معلق، صفحاتی هستند که لینک خروجی ندارند و مانند یک چاله امتیازات را به سمت خود می‌کشند. یک روش برای محاسبه امتیاز صفحات معلق قطع ارتباطات ورودی این صفحات است. در این روش امتیاز دریافتی از صفحات ورودی صفر است و در نتیجه بین صفحاتی که لینک خروجی ندارند و صفحاتی که فقط یک لینک خروجی دارند فاصله زیادی وجود دارد. این مشکل شکاف صفر و یک نامیده می‌شود و در [۲] راه حلی برای آن ارائه شده است.

Berlt و همکارانش مشکل دیگری را برای را برای روش‌های مبتنی بر گراف بیان کردند [۳]. آن‌ها مدعی شدند که یک لینک به یک صفحه می‌تواند به عنوان یک رأی برای مشخص کردن کیفیت یا عدم کیفیت آن صفحه باشد. آن‌ها مشکلات مطرح شده در [۱] را به شکل دیگری مطرح کردند. به عنوان مثال لینک‌های صفحات اسپم را مشخص نمودند که باید در رأی‌گیری حذف شوند و یا لینک‌های تکراری را نام بردند که باید فقط یک بار در رأی‌گیری‌ها لحاظ شوند. برای حل این مشکلات آنها مدل جدیدی با نام ابرگراف[۴] را ارائه دادند. در مدل ارائه شده، ابرگراف وب از دسته‌بندی صفحات در گروه‌های مجزا ساخته شد و پیوندهای گراف اصلی برای ساخت پیوندهای ابرگراف به کار برده شد. به این ترتیب اتصالات گراف شکل منحصر به فردتری بخود گرفت.

مشکل دیگر *PageRank* به علت تراکم در گراف وب رخ می‌دهد [۴]. آزمایش‌ها نشان می‌دهد که گراف وب از یک توزیع *power law* تبعیت می‌کند. این موضوع باعث می‌شود ماتریس اتصالات، یک ماتریس خلوت شود و در نتیجه امتیاز تخصیص یافته به بسیاری از صفحات خیلی ناچیز است و صفحات تازه متولد شده نیز امتیاز بسیار ناچیزی را دریافت کنند.

قربانی و همکارش الگوریتم *PageRank* وزن‌دار را پیشنهاد کردند [۵]. در الگوریتم پیشنهادی آن‌ها، به‌جای این که صفحات خروجی یک صفحه امتیاز یکسانی را از صفحه قبلی دریافت کنند بسته به میزان اهمیتشان وزن بیشتری را دریافت می‌کردند.

مشکل دیگری که در [۶] به آن اشاره شده است مسأله دکمه‌ی برگشت *(Back)* در زمان تغییر مسیر از یک صفحه است. در الگوریتم *PageRank* این گونه فرض شده است که کاربر یکی از لینک‌های خروجی را انتخاب می‌کند یا به صفحه دیگری پرش می‌کند درحالی‌که حالت سومی هم متصور است و آن برگشت به صفحه قبل که احتمال آن صفر نیست. با توجه به این مطلب *Mathieu* و *Bouklit* با برقرارکردن پیوند بین هر صفحه و صفحه‌ی قبل آن گراف وب را اصلاح کردند.

در وب برخی از صفحات قدرتمند دارای پیوندهای ضعیفی هستند و با توجه به بررسی‌های انجام شده بسیاری از روش‌های مبتنی بر گراف قادر به شناسایی آنها نیستند. به عنوان نمونه صفحات قدرتمند زیر دارای پیوندهای ضعیف هستند:

- صفحه‌ی اول وبلاگ‌ها: معمولاً در صفحه اول وبلاگ بخشی با عنوان آخرین وبلاگ‌های بروز رسانی شده وجود دارد که حاوی پیوندهایی به آخرین وبلاگ‌هایی که دارای تغییراتی بوده‌اند است. معمولاً وبلاگ صفحه‌ای با امتیاز بالاست درحالی‌که مقصد این پیوندها ممکن است صفحات ضعیفی باشند و اگر کنترل کافی انجام نشود ممکن است امتیاز وبلاگ به آخرین وبلاگ‌های بروز شده منتشر شود. شکل ۱ صفحه اصلی سایت بلاگفا[۵] را نشان می‌دهد. این سایت در صفحه اصلی خود آخرین وبلاگ‌های بروزرسانی شده را نشان می‌دهد.
- صفحات حاصل از جستجوی یک پرس‌وجو در یک موتور جستجو: بعضی از سایت‌ها لینک‌هایی در صفحات خود برای تحقیق بیشتر راجع به یک موضوع می‌گذارند که مقصد آن آدرسی است که موتور جستجو به ازای جستجوی آن موضوع تولید می‌کند. وجود این لینک باعث بروز مشکل نمی‌شود زیرا مقصد آن یک موتور جستجو است ولی لینک بعدی یعنی صفحه‌ی مربوط به موتور جستجو دارای لینک‌های خروجی است و در برخی از روش‌های مبتنی بر گراف مثل *HostRank* ممکن است امتیاز بالایی به آنها داده شود در حالی که صفحات مقصد صفحات ضعیفی هستند.
- یک میزبان با چند کاربر: بعضی مواقع صفحات یک میزبان کاربران مختلفی دارد و در نتیجه اهمیت لینک‌های خارج شونده از این صفحات متفاوت خواهد بود. به عنوان مثال در صفحه شخصی یک استاد ممکن است به صفحات شخصی دانشجویان آن استاد پیوندهایی وجود داشته باشد. واضح است که اگر صفحه‌ی شخصی استاد دارای امتیاز بالایی باشد صفحه شخصی دانشجو هم امتیاز بالایی دریافت می‌کند. تا اینجا مشکلی وجود ندارد ولی ممکن است صفحه شخصی دانشجو دارای پیوندهای خروجی ضعیفی باشد و در نتیجه به طور غیرمستقیم به‌صورت

واسطی برای انتقال رتبه صفحه قدرتمندی مثل صفحه استاد به صفحات ضعیف عمل کند.

شکل ۱: صفحه اصلی بلاگفا که بصورت لحظه ای آخرین وبلاگ‌های بروزشده را نشان می‌دهد

در این مقاله ابتدا الگوریتم *PageRank* توضیح داده شده است. سپس الگوریتم‌هایی برای حل مشکل جهش رتبه ارائه می‌گردد. به منظور ارزیابی الگوریتم اصلی ارائه شده نیاز به یک گراف واقعی با تعداد صفحات زیاد است و با توجه به این‌که چنین گرافی در دسترس نبود یک گراف فرضی با شرایط مشابه ساخته شد تا نتایج الگوریتم را بتوان مقایسه کرد.

۲- الگوریتم PageRank

الگوریتم *PageRank* به‌صورت مستقل از پرس‌وجو عمل می‌کند و در موتور جستجوی گوگل مورد استفاده قرار گرفته است. این الگوریتم روی کل گراف وب اجرا می‌شود و مقدار رتبه‌ی هر صفحه برابر جمع وزن‌دار رتبه‌های صفحات ورودی آن است؛ یعنی صفحه‌ای دارای رتبه‌ی بالا است که تعداد صفحات زیادی به آن اشاره کند یا صفحات اشاره‌کننده به آن رتبه بالایی داشته باشند [۷], [۸].

در *PageRank* به روابط بین صفحات توجه شده است. برای مثال، اگر صفحه‌ی p_1 اتصالی به صفحه‌ی p_2 دارد، پس موضوع p_2 احتمالاً جالب توجه برای خالق صفحه‌ی p_1 بوده است؛ بنابراین تعداد پیوند ورودی به صفحات وب نشان‌دهنده‌ی درجه علاقه‌ی صفحه برای دیگران است. واضح است که درجه علاقه‌ی صفحه با افزایش تعداد پیوندهای ورودی افزایش می‌یابد. علاوه بر این، وقتی صفحه‌ی وب پیوندهایی از صفحه‌ای مهم دریافت می‌کند، پس طبیعتاً باید رتبه‌ی بالاتری داشته باشد. *PageRank* صفحه j را با P_j نشان می‌دهند:

$$P_j = \frac{1-d}{n} + d \times \sum_{i \epsilon B(j)} \frac{P_i}{O(i)} \qquad (۱)$$

که $O(i)$ نشان‌دهنده‌ی تعداد پیوند خروجی از صفحه i و $B(j)$ نشان‌دهنده‌ی مجموعه صفحات که به صفحه j اشاره می‌کنند است.

بنابراین، *PageRank* صفحه j برابر با مجموع *PageRank*-های صفحات ورودی تقسیم بر درجه خروجی است. تقسیم *PageRank* ورودی صفحات به درجه‌ی خروجی‌شان یعنی $O(i)$ دو اثر دارد. اول، توزیع *PageRank* به همه خروجی‌ها به‌طور منصفانه است و دوم، جمع اثر هر صفحه و بردار رتبه‌ی آن صفحه نرمال می‌شود.

از پارامتر d برای مشخص کردن احتمال پرش به صفحات استفاده می‌شود که در واقع معادل رفتار پیمایش‌گر تصادفی است. وقتی کاربر به یک صفحه بدون پیوند خروجی رسید؛ به صفحه دیگری به‌صورت تصادفی پرش می‌کند؛ بنابراین، وقتی یک کاربر روی صفحه وب باشد با احتمال d یکی از پیوندهای خروجی را بطور تصادفی انتخاب می‌کند یا با احتمال $1-d$ به صفحات دیگری پرش می‌کند. به دلیل اینکه این روش مستقل از پرس‌وجو است؛ تمام صفحات با موضوعات مختلف با یکدیگر رقابت می‌کنند و باعث پایین آمدن دقت می‌شود. این روش از مشکل غنی‌تر شدن اغنیاء [۹] رنج می‌برد. همچنین ضریب سودمندی پایین این الگوریتم برای نداشتن گراف وب و محدود بودن تعداد پرس‌وجوها است. بزرگ‌ترین مزیت *PageRank* این است که هیچ ربطی به پرس‌وجو ندارد؛ بنابراین تمام ارزش‌های *PageRank* به‌صورت برون‌خط[۶] محاسبه می‌شوند. و محاسبات برخط[۷] را کاهش می‌دهد. بااین‌حال بزرگ‌ترین عیب الگوریتم *PageRank* این است که ارتباط موضوع با اطلاعات را نادیده می‌گیرد. به‌بیان‌دیگر، صفحات با *PageRank* متفاوت می‌تواند وجود داشته باشد که محتوایی مشابه داشته باشند [۱۰].

یکی دیگر از مشکلات *PageRank* صفحات معلق[۸] می‌باشد [۱۱]. همه صفحات وب پیوند خروجی ندارد مثل تصاویر، فایل‌های *PDF* و برخی صفحات توضیحی و موارد مشابه. صفحات معلق، صفحاتی هستند که پیوند خروجی ندارند و مانند یک چاله امتیازات را به سمت خود می‌کشند.

Patchmuthu و همکارانش روشی را برای تعیین اسپم پیوند[۹] صفحات معلق پیشنهاد کردند. در این روش به‌طور تصادفی یک صفحه هدف انتخاب می‌گردید و با استفاده از بردار ویژه[۱۰] و مقدار ویژه[۱۱] هرزنامه شناسایی می‌شد؛ سپس با اضافه کردن و حذف پیوند مشکل برطرف

می‌شود. نهایتاً الگوریتم *PageRank* بر روی گراف تغییریافته اعمال می‌شود. مشکل عمده این روش زمان اجرای بالای آن است و پیچیدگی محاسباتی آن‌که عملاً برای گراف‌های بزرگ قابل‌اجرا نخواهد بود [۱۲]. در [۱۳]، الگوریتم ساده‌ی برای محاسبه *PageRank* ارائه شد. این الگوریتم تمام صفحات وب معلق را به‌عنوان یک صفحه در نظر می‌گیرد و نشان می‌دهد که می‌توان رتبه صفحات وب غیرمعلق را مستقل از رتبه صفحات معلق محاسبه کرد. عملکرد آنها باعث شد رتبه‌بندی روی ماتریس کوچک‌تری اجرا شود. آنها نشان دادند که در *PageRank* صفحات معلق وابستگی شدیدی به صفحات وب غیرمعلق دارد اما برعکس آن برقرار نیست. مزایای این روش پیاده‌سازی ساده آن است و اینکه به حداقل ذخیره‌سازی نیاز دارد.

پنگ و همکارانش برای رفع مشکل رانش موضوع[۱۲] و تأکید صفحات قدیمی‌تر (همان مشکل غنی‌تر شدن اغنیا) الگوریتم *PageRank* را با بکارگیری محتوای صفحات و فاکتور زمان بهبود دادند [۱۴]. برای کاهش مشکل غنی‌تر شدن اغنیا، ستایش و همکارانش نسخه‌ی جدیدی از الگوریتم *PageRank* ایجاد کردند که از علایق کاربران صفحه وب و الگوریتم کلنی مورچه[۱۳] استفاده می‌کند [۱۵].

الگوریتم *Norm-PageRank* نسخه جدیدی از *PageRank* است. این الگوریتم در هر مرحله امتیاز *PageRank* صفحات را برای افزایش سرعت همگرایی نرمال می‌کند [۱۶]. الگوریتم *TrustRank* برای کاهش اسپم پیوند توسط گوگل در سال ۲۰۰۵ ارائه شده است [۱۷]. این الگوریتم صفحات *seed* را صفحات قابل‌اعتماد و شناخته‌شده در نظر می‌گیرد تا امتیاز به صفحه اسپم نشت پیدا نکند.

زینگ[۱۴] و قربانی الگوریتم *PageRank* وزن‌دار یا به اختصار *WPR*[۱۵] را پیشنهاد کردند [۵]. در الگوریتم پیشنهادی آن‌ها، به‌جای این‌که صفحات خروجی یک صفحه امتیاز یکسانی را از صفحه قبلی دریافت کنند بسته به میزان اهمیتشان وزن بیشتری را دریافت می‌کردند.

مشکل دیگری که در [۶] به آن اشاره شده است مسئله دکمه‌ی *Back* در زمان تغییر مسیر از یک صفحه می‌باشد. در الگوریتم *PageRank* این‌گونه فرض شده است که کاربر یکی از پیوندهای خروجی را انتخاب می‌کند یا به صفحه دیگری پرش می‌کند درحالی‌که حالت سومی هم متصور است و آن برگشت به صفحه قبل که احتمال آن صفر نیست. متیو[۱۶] و بوکلیت[۱۷] در [۶] با برقرار کردن پیوند بین هر صفحه و صفحه‌ی قبل آن گراف وب را اصلاح کردند.

در موقع ورود به برخی از صفحات، به‌طور خودکار مسیر صفحه عوض می‌شود[۱۸]. این موضوع نیز یکی از چالش‌ها در گراف وب می‌باشد و تحقیقاتی نیز در این زمینه انجام شده است [۱۸].

در وب ممکن است گروهی از صفحات به یکدیگر پیوندهایی داشته باشند و هیچ پیوندی به بیرون نداشته باشند به این مشکل دام عنکبوتی[۱۹] گفته می‌شود [۱۹] و روش برطرف کردن آن مشابه صفحات معلق می‌باشد.

همچنین در [۲۰] مقایسه‌ای بین الگوریتم‌های مبتنی بر *PageRank* انجام شده است و روش‌هایی که اشکالات *PageRank* را برطرف کرده‌اند مورد بررسی قرار گرفته است. ازجمله الگوریتم *DistanceRank* که مبتنی بر یادگیری تقویتی است و یکی از مشکلات *PageRank* یعنی غنی‌تر شدن اغنیاء[۲۰] را کاهش می‌دهد [۲۱].

۳- ارائه روش‌هایی برای حل مشکل جهش PageRank

در این بخش روش‌هایی برای حل مشکل جهش رتبه ارائه می‌شود. یکی از ساده‌ترین روش‌ها روش دستی برای محدود کردن انتشار رتبه صفحاتی است که مشابه بلاگفا عمل می‌کنند. البته در مقیاس بزرگی مثل وب شاید عملا امکان‌پذیر نباشد. در این روش ابتدا به‌طور خودکار یا دستی صفحات قوی با پیوندهای خروجی ضعیف شناسایی می‌شوند و سپس به‌صورت دستی پیوندهای خروجی صفحات قوی حذف می‌شوند یا بخشی از آن منتقل می‌شود.

روش دیگر استفاده از تگ‌های *HTML* است. یکی از این تگ‌ها که برای موتورهای جستجو اهمیت دارد *nofollow* می‌باشد. این تگ که بعضا در پیوندهای خروجی یک صفحه به کار برده می‌شود به خزش‌گر موتور جستجو می‌گوید که به دلایلی طراح سایت مایل نیست که پیوند خروجی دنبال شود. این روش قطعی نیست زیرا ممکن است طراح سایت به هر دلیل این تگ را استفاده نکند. مثلا سایت بلاگفا از تگ *nofollow* برای وبلاگ‌هایی که اخیرا بروز شده است و در صفحه اصلی سایت نشان داده می‌شود استفاده نکرده است.

روش دیگر که برای سایت‌هایی مثل بلاگفا مفید است؛ خزش مجدد است. در این روش خزش‌گر درصورتی امتیاز بالای یک صفحه را به صفحات دیگر منتشر می‌کند که در حداقل دو یا سه خزش متوالی آن پیوند وجود داشته باشد. این روش برای سایت‌هایی مثل بلاگفا مفید است اما برای یک استاد که برای دانشجویانش یک صفحه اختصاص داده است مناسب نیست.

روش پیشنهادی دیگر محدود کردن افزایش امتیاز یک صفحه است. در این روش حداکثر امتیاز دریافتی از یک سند محدود می‌شود. به

عنوان مثال می‌توان این‌گونه تعریف کرد که حداکثر امتیاز دریافتی از یک صفحه بیشتر از α درصد امتیاز اولیه سند نباشد. برای این منظور ابتدا رابطه (۱) را به صورت رابطه (۲) تبدیل می‌کنیم تا مشخص شود برای محاسبه P_j از هر صفحه ورودی چقدر امتیاز دریافت می‌شود (مقدار داخل سیگما در رابطه (۲) مشخص‌کننده امتیازی است که از صفحه i برای محاسبه P_j لحاظ گردیده است).

$$P_j = \sum_{i \epsilon B(j)} \left(d \times \frac{P_i}{O(i)} + \frac{1-d}{n \times |B(j)|} \right) \qquad (۲)$$

سپس تابع f را طبق رابطه ۳ تعریف می‌کنیم تا دریافت بیش از حد امتیاز از یک صفحه جلوگیری شود:

$$f(P_i, O_i, P_j) = \qquad (۳)$$

$$\begin{cases} d \times \frac{P_i}{O(i)} + \frac{1-d}{n \times |B(j)|} & d \times \frac{P_i}{O(i)} + \frac{1-d}{n \times |B(j)|} \leq (\alpha)\frac{1}{n} \\ (\alpha)\frac{1}{n} & d \times \frac{P_i}{O(i)} + \frac{1-d}{n \times |B(j)|} > (\alpha)\frac{1}{n} \end{cases}$$

$$P_j = \sum_{i \epsilon B(j)} f(P_i, O_i, P_j) \qquad (۴)$$

در رابطه های ۳ و ۴ تابع f کنترل کننده میزان امتیازی است که صفحه مقصد از صفحه مبدا دریافت می‌کند و روی $(1+\alpha)\frac{1}{n}$ متوقف می‌شود. لازم به ذکر است که امتیاز اولیه تمام صفحات $\frac{1}{n}$ است.

قضیه ۱: اگر در رابطه ۳ آلفا مقدار کمی درنظر گرفته شود امتیاز هر صفحه بصورت رابطه ۵ خواهد بود.

$$P_j = |B(j)| \times \frac{1}{\sum_{i=1}^{n} O(i)} \qquad (۵)$$

اثبات: اگر آلفا کوچک انتخاب شود به ازای همه ورودی‌ها شرط دوم رابطه ۳ برقرار می‌شود و خواهیم داشت:

$$f(P_i, O_i, P_j) = (\alpha)\frac{1}{n} \qquad (۶)$$

با قراردادن رابطه ۶ در رابطه ۴ رابطه ۷ بدست می‌آید:

$$P_j = |B(j)| \times \alpha \times \frac{1}{n} \qquad (۷)$$

با نرمال‌سازی رابطه ۷ رابطه رابطه ۵ بدست می‌آید و قضیه ثابت می‌شود.

***

قضیه ۲: اگر در رابطه ۳ آلفا مقدار بزرگی درنظر گرفته شود امتیاز هر صفحه همان امتیاز *PageRank* صفحه هست.

اثبات: در رابطه ۳ اگر آلفا بزرگ باشد هیچگاه شرط دوم برقرار نخواهد شد و در نتیجه مقدار تابع f بصورت زیر خواهد بود:

$$f(P_i, O_i, P_j) = d \times \frac{P_i}{O(i)} + \frac{1-d}{n \times |B(j)|} \qquad (۸)$$

اگر رابطه ۸ را در رابطه ۴ جاگذاری کنیم رابطه ۲ بدست می‌آید که همان فرمول *PageRank* می‌باشد.

***

قضیه ۱ و ۲ نشان داد که انتخاب آلفا اهمیت دارد و اگر آلفا به درستی انتخاب نشود یا امتیاز حاصل یک فرایند تصادفی ساده است یا همان *PageRank* قبلی که مشکل جهش رتبه را دارد. برای انتخاب آلفا می‌توان بدین‌گونه عمل کرد که نتایج حاصل از آلفای خیلی کوچک را بدست آورد. سپس نتایج مربوط به آلفای خیلی بزرگ محاسبه کرد و میانگین امتیازها را بدست آورد. سپس با استفاده از معیار خطای میانگین مربعات[22]یا MSE^{21} آلفای مناسب را بدست آورد.

بنابراین بطور خلاصه مراحل زیر برای محاسبه آلفا دنبال می‌شود:

۱. ابتدا با آلفای خیلی کم مقدار امتیازها محاسبه می‌گردد.
۲. سپس امتیازها با آلفای خیلی بزرگ محاسبه می‌گردد. (طبق قضیه ۲ مقدار آلفا در این حالت همان *PageRank* صفحات است.)
۳. در این مرحله میانگین امتیازها محاسبه می‌شود. برای محاسبه میانگین امتیازهای محاسبه شده از دو مرحله قبل با هم جمع می‌شوند و بر دو تقسیم می‌شوند.
۴. نهایتاً آلفا در رابطه ۳ به‌گونه‌ای محاسبه می‌شود که *MSE* میانگین امتیازها در مرحله قبل و امتیاز در رابطه ۴ حداقل شود.

۴- ارزیابی

به منظور ارزیابی الگوریتم اصلی ارائه شده نیاز به یک گراف واقعی با تعداد صفحات زیاد است و با توجه به‌این‌که چنین گرافی در دسترس نبود یک گراف فرضی با شرایط مشابه ساخته شد تا نتایج

الگوریتم را بتوان مقایسه کرد. بدین‌منظور گرافی ساخته شد که در آن گره‌ای مثل B دارای تعداد زیادی ورودی باشد. سپس یکی از خروجی‌های آن گره C فرض شد که تنها ورودیش از B باشد.

اگر *PageRank* را برای گراف جدول ۱ به‌دست آوریم جدول ۲ حاصل می‌شود. همان‌طور که جدول ۲ نشان می‌دهد مقدار امتیاز دریافت شده توسط صفحه C با تنها یک ورودی از B برابر با ۰/۱۴ می‌باشد که نسبت به گره‌های مشابه امتیاز بالایی را دریافت کرده است. مثلا گره A با چهار ورودی امتیازش ۰/۱۴ هست. البته گره H هم امتیاز بالای دریافت کرده است و دلیل اصلی آن ورودی داشتن از گره B می‌باشد. بنابراین می‌توان نتیجه گرفت که گره قوی B می‌تواند به‌راحتی امتیاز گره‌های دیگر را تحت تاثیر قرار دهد.

جدول ۱: ماتریس مجاورت گراف وب جهت ارزیابی الگوریتم

	A	B	C	D	E	F	G	H	K	L
A	-	1	0	0	0	0	0	0	1	0
B	0	-	1	0	0	0	0	1	0	0
C	0	0	-	1	1	0	0	0	0	0
D	0	1	0	-	0	1	1	1	0	0
E	1	0	0	0	-	0	0	0	0	0
F	0	1	0	0	0	-	1	0	0	0
G	0	1	0	0	0	0	-	0	0	1
H	0	1	0	0	0	0	0	-	1	0
K	1	1	0	0	0	0	0	0	-	0
L	1	1	0	0	0	0	0	0	0	-

حال اگر با استفاده از رابطه ۴ امتیازها را حساب کنیم جدول ۳ به‌دست می‌آید. در جدول ۳ مشخص هست که با افزایش آلفا، به امتیازهای جدول ۲ نزدیک می‌شویم زیرا با استفاده از آلفا برش انجام شود و وقتی آلفا زیاد باشد برش کمتری صورت می‌پذیرد. وقتب که آلفا کم باشد عملا امتیاز از صفحات قبلی به صفحات بعدی منتشر نمی‌شود و فقط تعداد پیوندهای ورودی تاثیرگذار می‌باشد. مثلا جدول ۱ دارای ۲۰ پیوند می‌باشد و سهم هر پیوند ۰/۰۵ می‌باشد و درنتیجه امتیازاتی که به ازای آلفا ۰/۰۰۱ نتیجه می‌شود تعداد پیوندهای ورودی ضرب در ۰/۰۵ می‌باشد. نتیجه تجربی این موضوع تایید کننده قضیه ۱ نیز می‌باشد.

جدول ۲: امتیاز *PageRank* صفحات مربوط به گراف جدول ۱ در حالت عادی

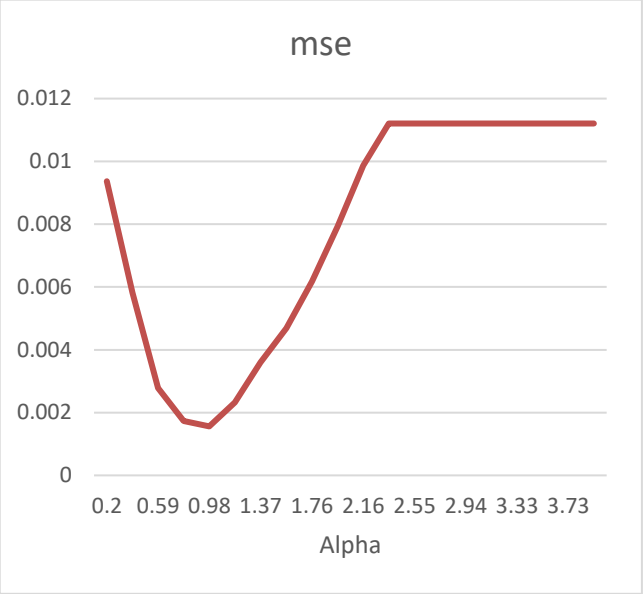
Node	*PageRank Score*
A	0.09
B	0.18
C	0.12
D	0.08
E	0.08
F	0.03
G	0.05
H	0.15
K	0.17
L	0.03

جدول ۳: امتیاز *PageRank* صفحات مربوط به گراف جدول ۱ با استفاده از رابطه ۴

A	*0.001*	*0.01*	*0.5*	*0.75*	*1*	*1.5*	*2*	*100*
A	0.15	0.14	0.15	0.14	0.13	0.11	0.10	0.09
B	0.30	0.33	0.28	0.25	0.24	0.21	0.19	0.17
C	0.05	0.05	0.06	0.07	0.08	0.09	0.11	0.12
D	0.05	0.05	0.06	0.07	0.07	0.07	0.08	0.08
E	0.05	0.05	0.06	0.07	0.07	0.07	0.08	0.08
F	0.05	0.05	0.04	0.04	0.04	0.04	0.04	0.04
G	0.10	0.10	0.08	0.07	0.07	0.06	0.06	0.06
H	0.10	0.10	0.09	0.10	0.11	0.12	0.14	0.15
K	0.10	0.10	0.12	0.13	0.15	0.17	0.16	0.16
L	0.05	0.05	0.06	0.07	0.06	0.05	0.05	0.05

شکل ۲ مقدار *MSE* را برحسب آلفاهای متفاوت نشان می‌دهد.

براساس این شکل مقدار *MSE* در آلفای حدود *0.98* کمینه است.



شکل ۲: نمودار تغییرات *MSE* براساس مقادیر مختلف آلفا

براساس شکل ۲ و جدول ۳ مقدار یک برای آلفا مقدار مناسبی است. همانطور که جدول ۳ نشان می‌دهد به ازای آلفای برابر با یک مقدار امتیاز دریافتی گره‌های مختلف هموارتر گردیده است و عملا جهش رتبه مشاهده نمی‌شود.

۵- خلاصه و چشم انداز آینده

در این مقاله یکی از مشکلات رتبه‌بندی مبتنی بر گراف وب یعنی جهش رتبه بررسی شد و روش‌هایی برای حل این مشکل پیشنهاد گردید. مشکل جهش رتبه بدین علت رخ می‌دهد که رتبه‌ی یک صفحه به صفحات خروجی آن بدون هیچ کنترلی انتشار می‌یابد درحالی‌که ممکن است صفحه خروجی واقعاً توصیه شده توسط صفحه قبلی بطور مستقیم یا غیرمستقیم نباشد. برای حل مشکل جهش رتبه در *PageRank* یک فرمول جدیدی برای محاسبه *PageRank* ارائه گردید. در این فرمول جدید با تعریف متغیرهایی، انتشار امتیاز از یک صفحه دیگرکنترل گردید.

مراجع

[1] Z. Dou, R. Song, J.-Y. Nie, and J.-R. Wen, “Using anchor texts with their hyperlink structure for web search,” in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 227–234.

[2] X. Wang, T. Tao, J.-T. Sun, A. Shakery, and C. Zhai, “Dirichletrank: Solving the zero-one gap problem of pagerank,” *ACM Trans. Inf. Syst.*, vol. 26, no. 2, p. 10, 2008.

[3] K. Berlt, E. S. de Moura, A. Carvalho, M. Cristo, N. Ziviani, and T. Couto, “Modeling the web as a hypergraph to compute page reputation,” *Inf. Syst.*, vol. 35, no. 5, pp. 530–543, 2010.

[4] A. N. Nikolakopoulos and J. D. Garofalakis, “NCDawareRank: a novel ranking method that exploits the decomposable structure of the web,” in *Proceedings of the sixth ACM international conference on Web search and data mining*, 2013, pp. 143–152.

[5] W. Xing and A. Ghorbani, “Weighted pagerank algorithm,” in *Proceedings of Second Annual Conference on Communication Networks and Services Research*, 2004, pp. 305–314.

[6] F. Mathieu and M. Bouklit, “The effect of the back button in a random walk: application for pagerank,” in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, 2004, pp. 370–371.

[7] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Rankin: Bringing Order to the Web,” *World Wide Web Internet Web Inf. Syst.*, vol. 54, no. 1999–66, pp. 1–17, 1998.

[8] M. Bianchini, M. Gori, and F. Scarselli, “Inside pagerank,” *ACM Trans. Internet Technol.*, vol. 5, no. 1, pp. 92–128, 2005.

[9] A.-L. Barabási and R. Albert, “Emergence of Scaling in Random Networks,” *Science (80-.).*, vol. 286, no. October, pp. 509–512, 1999.

[10] L. Z. Xiang, “Research and Improvement of PageRank Sort Algorithm Based on Retrieval Results,” in *Intelligent Computation Technology and Automation (ICICTA), 2014 7th International Conference on*, 2014, pp. 468–471.

[11] A. N. Langville and C. D. Meyer, “A reordering for the PageRank problem,” *SIAM J. Sci. Comput.*, vol. 27, no. 6, pp. 2112–2120, 2006.

[12] R. K. Patchmuthu, A. K. SINGH, and A. Mohan, “A new algorithm for detection of link spam contributed by zero-out link pages,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 24, no. 4, pp. 2106–2123, 2016.

[13] I. C. F. Ipsen and T. M. Selee, “PageRank computation, with special attention to dangling nodes,” *SIAM J. Matrix Anal. Appl.*, vol. 29, no. 4, pp. 1281–1296, 2007.

[14] P. Zha, X. Xu, and M. Zuo, “An Efficient Improved Strategy for the PageRank Algorithm,” in *2011 International Conference on Management and Service Science*, 2011, pp. 1–4.

[15] S. Setayesh, A. Harounabadi, and A. M. Rahmani, “Presentation of an Extended Version of the PageRank Algorithm to Rank Web Pages Inspired by Ant Colony Algorithm,” *Int. J. Comput. Appl.*, vol. 85, no. 17, 2014.

[16] K. Mohan and J. Kurmi, “A Technique to Improved Page Rank Algorithm in perspective to Optimized Normalization Technique,” *Int. J.*, vol. 8, no. 3, 2017.

[17] N. L. Amy and D. M. Carl, “Google’s PageRank: The Math Behind the Search Engine,” *Pricet. Univ. Press. Nol*, vol. 3, pp. 335–380, 2004.

[18] M. Zhukovskii, G. Gusev, and P. Serdyukov, “URL redirection accounting for improving link-based ranking methods,” in *Advances in Information Retrieval*, Springer, 2013, pp. 656–667.

[19] Z. Bahrami Bidoni, R. George, and K. Shujaee, “A Generalization of the PageRank Algorithm,” in *ICDS 2014, The Eighth International Conference on Digital Society*, 2014, pp. 108–113.

[20] A. K. Singh, “A comparative study of page ranking algorithms for information retrieval,” *Int. J. Electr. Comput. Eng.*, vol. 4, pp. 469--480, 2009.

[21] A. M. Zareh Bidoki and N. Yazdani, “DistanceRank: An intelligent ranking algorithm for web pages,” *Inf. Process. Manag.*, vol. 44, no. 2, pp. 877–892, 2008.

[22] J. Mao, Y. Liu, N. Kando, C. Luo, M. Zhang, and S. Ma, “Investigating Result Usefulness in Mobile Search,” in *European*

Conference on Information Retrieval, 2018, pp. 223–236.

پاورقی‌ها:

[1] Zero-One Gap
[2] Random Surfer
[3] Dangling Pages
[4] Hyper Graph
[5] www.blogfa.com
[6] offline
[7] online
[8] Dangling Pages
[9] Link Spam
[10] Eigen Vector
[11] Eigen Value
[12] Topic Drift
[13] Ant Colony Algorithm
[14] Xing
[15] Weighted PageRank
[16] Mathieu
[17] Bouklit
[18] Redirect
[19] Spider-Trap
[20] Rich-Get-Richer
[21] Mean Squared Error