



ارائه روشی جدید برای رتبه بندی صفحات وب با استفاده از اتوماتای یادگیر توزیع شده

زهره اناری

دانشجویی کارشناسی ارشد نرم افزار
دانشگاه آزاد اسلامی واحد شیبدشت
zanari323@yahoo.com

محمد رضا میبدی

استاد و عضو هیئت علمی
دانشگاه صنعتی امیر کبیر
mmeybodi@aut.ac.ir

بابک اناری

عضو هیئت علمی
دانشگاه آزاد اسلامی واحد شیبدشت
anari322@yahoo.com

یک الگوریتم رتبه بندی بر مبنای یکسری صفحات وب شروع بکار می کند. بر حسب اینکه این مجموعه صفحات وب، چگونه استخراج و رتبه بندی می شوند دو نوع الگوریتم رتبه بندی خواهیم داشت: الگوریتمهای مستقل از کوئری و الگوریتمهای وابسته به کوئری [3]. یک الگوریتم رتبه بندی را مستقل از کوئری می گویند، اگر هنگام اعمال رتبه بندی، به رتبه بندی تمامی صفحات وب پرداخته شود. همچنین یک الگوریتم رتبه بندی را وابسته به کوئری می گویند، اگر هنگام اعمال رتبه بندی، فقط به رتبه بندی زیر مجموعه ای از صفحات مرتبط با کوئری کاربر پرداخته شود نه به رتبه بندی تمامی صفحات وب. مثلاً الگوریتم رتبه بندی PageRank [1,2] نمونه ای از الگوریتمهای مستقل از کوئری است و الگوریتم HITS [3] نیز نمونه ای از الگوریتمهای وابسته به کوئری است.

رتبه یک صفحه مثل A در الگوریتم رتبه بندی PageRank بصورت رابطه زیر محاسبه می شود.

$$PR(A) = \frac{1-d}{N} + d \times \left\{ \frac{PR(T_1)}{C(T_1)} + \dots + \frac{PR(T_n)}{C(T_n)} \right\} \quad (1)$$

که در آن N، تعداد صفحات وب T_i ، رتبه صفحه T_i که به صفحه A لینک دارند. C(T_i)، تعداد یالهای صفحه T_i ، d، ضریب تعدیل که عددی بین صفر و یک است. از آنجاییکه این الگوریتم رتبه صفحات دیگر را برای محاسبه استفاده می کند، این مقدار بصورت بازگشتی محاسبه می شود. ولی در عمل از روش های عددی برای حل این معادله استفاده می شود. همانطوریکه از رابطه فوق مشخص است، عیب این الگوریتم این است که فقط به یالهای ارتباطی بین صفحات وب توجه می کند و به تعداد گذرهای کاربران در بین این یالها توجه نمی کند. اگر تعداد گذرهای کاربران در بین صفحات وب در دسترس باشد، می توان از الگوریتم PageRank [4] بجای الگوریتم PageRank استفاده نمود. رتبه یک صفحه مثل A در الگوریتم رتبه بندی PageRank بصورت رابطه زیر محاسبه می شود.

$$PR(A) = \frac{1-d}{N} + d \times \left\{ PR(T_1) \times \frac{n(T_1, A)}{\sum_i n(T_1, i)} + \dots + PR(T_n) \times \frac{n(T_n, A)}{\sum_i n(T_n, i)} \right\} \quad (2)$$

چکیده: با افزایش روز افزون تعداد صفحات وب، مسئله جستجوی اطلاعات در وب توسط کاربران اهمیت زیادی پیدا می کند. کاربران وب دوست دارند کوئری خود را به موتور جستجو داده و در نهایت خروجی خودشان را دریافت کنند، این خروجی که دنباله ای متواالی و مرتبط به هم از صفحات وب است، باید نشان دهنده نتایج مطلوب کاربران باشد. برای ارزیابی میزان اهمیت ارتباط بین صفحات وب، از تکنیکی بنام رتبه بندی استفاده می شود. یکی از نقطه ضعفهای الگوریتمهای رتبه بندی موجود در این است که این الگوریتمها فقط به ساختار ارتباطی بین صفحات وب توجه می کنند و به نحوه استفاده کاربران از یالهای ارتباطی توجهی نمی کنند. در این مقاله روشنی مبتنی بر اتوماتای یادگیر توزیع شده برای حل این مشکل پیشنهاد شده است. در روش پیشنهادی به هر صفحه وب یک اتوماتای یادگیر تخصیص داده می شود که وظیفه آن یادگیری ارتباط آن صفحه با سایر صفحات وب دیگر است. الگوریتم پیشنهادی، با توجه به مقدار بردار احتمال هر اتومات، رتبه هر صفحه وب را بصورت بازگشتی محاسبه می کند. نتایج شبیه سازیها نشان داده است که روش پیشنهادی در مقایسه با تنها گزارش شده مبتنی بر اتوماتای یادگیر توزیع شده در تعیین رتبه بندی صفحات وب، از کارایی قابل ملاحظه ای برخوردار است.

کلمات کلیدی: رتبه بندی، اتوماتای یادگیر، اتوماتای یادگیر توزیع شده

۱- مقدمه

یکی از عناصر اصلی برای مرتب سازی نتایج جستجو در موتورهای جستجوگر، رتبه بندی می باشد. هدف از رتبه بندی صفحات وب، پیدا کردن صفحات مرتبط به هم است. رتبه بندی تکنیکی است که بوسیله آن می توان به کشف ساختار ارتباطی صفحات وب پرداخت. ایده رتبه بندی صفحات وب که توسط Page, Brin ارائه گردیده است [1,2] در موتور جستجوی گوگل به منظور مرتب کردن نتایج جستجو استفاده می شود.

بدون تغییر می‌مانند، حال آنکه در محیط غیر ایستا این مقدار در طی زمان تغییر می‌کنند. اتماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم بندی می‌گردد. در ادامه به شرح مختصه درباره اتماتاهای یادگیر با ساختار متغیر که در این مقاله از آنها استفاده شده است می‌پردازیم.

اتوماتای یادگیر با ساختار متغیر: اتماتای یادگیر با ساختار متغیر توسط ۴ تایی $\{a, b, p, T\}$ نشان داده می‌شود که در آن $\{a_1, a_2, \dots, a_r\}$ مجموعه عملهای اتماتا $a \equiv \{a_1, a_2, \dots, a_r\}$ $b \equiv \{b_1, b_2, \dots, b_m\}$ L $p \equiv \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از عملهای a و b $T[a(n), b(n), p(n)] = T$ می‌باشد. در این نوع از اتماتاهای یادگیری اگر عمل a_i در الگوریتم یادگیری می‌باشد، در مراحله انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال p_i افزایش یافته و سایر احتمالها کاهش می‌یابند و برای پاسخ نامطلوب احتمال p_j کاهش یافته و سایر احتمالها افزایش می‌یابند. در هر حال، تغییرات به‌گونه‌ای صورت می‌گیرد تا حاصل جمع p_i ها همواره مساوی یک باقی بماند. الگوریتم زیر یک نمونه از الگوریتمهای یادگیری خطی در اتماتای با ساختار متغیر است.

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1 - a)p_j(n) \quad \forall j \neq i \end{aligned} \quad (3)$$

الف- پاسخ مطلوب

$$p_i(n+1) = (1 - b)p_i(n) \quad (4)$$

$$\forall j \neq i \quad p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n)$$

ب- پاسخ نامطلوب

در روابط فوق، a پارامتر پاداش و b پارامتر جریمه می‌باشد. با توجه به مقادیر a و b سه حالت را می‌توان در نظر گرفت. زمانیکه b با هم برابر باشند، الگوریتم را L_{RP} می‌نامیم. زمانیکه b با a خیلی کوچکتر باشد، الگوریتم را L_{REP} می‌نامیم. زمانیکه b مساوی صفر باشد، الگوریتم را L_{RI} می‌نامیم. برای مطالعه بیشتر درباره اتماتاهای یادگیر می‌توان به مراجع [6,7,8,9,10] مراجعه کرد.

۱-۲- اتماتای یادگیر توزیع شده

یک اتماتای یادگیر توزیع شده (DLA)، شبکه‌ای از اتماتاهای یادگیر است که برای حل یک مساله خاص با یکدیگر همکاری دارند. در این شبکه از اتماتاهای یادگیر در هر زمان تنها یک اتماتا فعال است. تعداد اعمال قابل انجام توسط یک اتماتا در DLA برابر با تعداد اتماتاهایی است که به این اتماتا متصل شده‌اند. انتخاب یک عمل توسط اتماتا

که در آن $n(T_{j,A})$ تعداد کاربرانی است که از صفحه T_j به صفحه A رفته‌اند.

$\sum_i n(T_{j,i})$ مجموع تمامی کاربرانی که از صفحه T_j به تمام صفحات دیگر رفته‌اند. نقطه ضعف این الگوریتم این است که بدليل استفاده از ماتریس ارتباطات، استفاده از آن در مجموعه‌های بزرگ و قابل گسترش مناسب نمی‌باشد.

عیب تنها الگوریتم گزارش شده مبتنی بر اتماتای یادگیر توزیع شده برای رتبه بندی صفحات وب [5] داشتن کارایی پایین آن است. نتایج حاصل از شبیه سازی روش پیشنهادی نشان دهنده کارایی بالای آن می‌باشد. ادامه مقاله بین صورت سازماندهی شده است: در بخش ۲ اتماتای یادگیر و اتماتای یادگیر توزیع شده و تعیین ساختار ارتباطی بین صفحات وب توسط اتماتای یادگیر توزیع شده مورد بحث قرار خواهد گرفت. در بخش ۳ روش پیشنهادی ارائه خواهد شد و در بخش ۴ نتایج شبیه سازیها ارائه می‌گردد. بخش پایانی نتیجه‌گیری می‌باشد.

۲- اتماتای یادگیر

اتوماتای یادگیر یک مدل انتزاعی است که تعداد محدودی عمل را می‌تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی به اتماتای یادگیر داده می‌شود. اتماتای یادگیر از این پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب می‌کند.

شکل ۱: ارتباط بین اتماتای یادگیر و محیط را نشان می‌دهد.



شکل ۱: ارتباط بین اتماتای یادگیر و محیط

محیط: محیط را می‌توان توسط سه تایی $E \equiv \{a, b, c\}$ نشان داد که در آن $a \equiv \{a_1, a_2, \dots, a_r\}$ مجموعه ورودیهای a می‌باشد، $b \equiv \{b_1, b_2, \dots, b_m\}$ مجموعه خروجیهای b و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالهای جریمه می‌باشد. هر گاه b مجموعه دو عضوی باشد، محیط از نوع P می‌باشد. در چنین محیطی $b_1 = 1$ به عنوان جریمه و $b_2 = 0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع Q ، $b(n)$ می‌تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله $[0,1]$ و در محیط از نوع S $b(n)$ متغیر تصادفی در فاصله $[0,1]$ است. c_i احتمال اینکه عمل c_i نتیجه نامطلوب داشته باشد می‌باشد. در محیط ایستا مقادیر c_i

یادگیر در اتوماتیک یادگیر توزیع شده طبق الگوریتم یادگیری بروز می‌شود.

هر کدام از اعمال یک اتوماتیک یادگیر، متناظر با یک صفحه در مجموعه صفحات و احتمال انتخاب این عمل در بردار احتمالات، ارتباط این صفحه با صفحه متناظر با آن عمل می‌باشد. بعبارت دیگر بردار اعمال یک اتوماتیک یادگیر می‌تواند بعنوان شناسه صفحه متناظر با آن اتوماتیک یادگیر و بردار احتمالات، میزان ارتباط این صفحه با دیگر صفحه‌ها در مجموعه صفحات در نظر گرفته شود. برای هر صفحه P_i یک اتوماتیک یادگیر LA_i در نظر می‌گیریم. انتخاب عمل z توسط اتوماتیک یادگیر LA_i به معنی فعال کردن اتوماتیک یادگیر LA_j متناظر با صفحه P_j می‌باشد. در صورتیکه عمل انتخاب شده k امین عمل اتوماتیک LA_i باشد (یعنی $a_{ik}^i = j$) احتمال متناظر این عمل

یعنی p_{ik}^i بعنوان میزان ارتباط صفحات i و j در نظر گرفته می‌شود. با ورود یک کاربر به سیستم و مشاهده صفحه P_i اتوماتیک یادگیر متناظر با آن صفحه یعنی LA_i فعال می‌شود. با حرکت کاربر از صفحه P_i به صفحه P_j ، عمل مرتبط با این انتخاب در اتوماتیک LA_i انتخاب و به محیط اعمال می‌شود.

در این مقاله برای تعیین میزان ارتباط بین صفحات و ب از الگوریتم [18] استفاده کردایم. اختلاف این الگوریتم با الگوریتمهای پیشین مبتنی بر اتوماتیک یادگیر توزیع شده [16,17] در چگونگی بروز کردن بردار احتمال اعمال اتوماتیک یادگیر در DLA می‌باشد. فرض کنید، LA_k بردار احتمال اتوماتیک یادگیر LA_k باشد که به صفحه k تخصیص داده شده است و p_m^k احتمال انتخاب عمل a_m^k و r تعداد صفحات باشد. اگر کاربر حرکت $D_k \rightarrow D_m$ را انجام دهد (از صفحه D_k به صفحه D_m حرکت کند) در این صورت اتوماتیک یادگیر LA_k بردار احتمال اعمال خود را طبق الگوریتم یادگیری زیر بروز می‌کند.

$$p_m^k(n+1) = p_m^k(n) + a_m^k [1 - p_m^k(n)] \quad (5)$$

$$p_j^k(n+1) = (1 - a_m^k) p_j^k(n) \quad j \neq m \quad \forall j \quad (6)$$

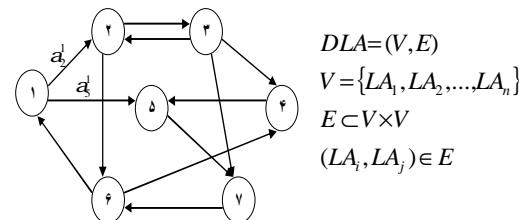
$$a_m^k = \frac{E_m^k}{1 + E_m^k} \quad (7)$$

$$E_m^k = -(p_m^k \log p_m^k + (1 - p_m^k) \log(1 - p_m^k)) \quad (8)$$

مقدار بالا برای E_m^k نشان دهنده ارتباط بیشتر بین دو صفحه - D_m و D_k می‌باشد و بالعکس هرچه این مقدار کمتر باشد نشان

یادگیر در این شبکه باعث فعال شدن اتوماتیک یادگیر متناظر با این عمل می‌گردد.

یک DLA توسط یک گراف که هر یک از رئوس آن یک اتوماتیک یادگیر است، نشان داده می‌شود. وجود یال (LA_i, LA_j) در این گراف بین معنایت که انتخاب عمل j توسط LA_i باعث فعال شدن j LA می‌گردد. تعداد اعمال قابل انتخاب توسط LA_k بصورت $\{p_1^k, p_2^k, \dots, p_{r_k}^k\}$ نمایش داده می‌شود. در این مجموعه عدد p_m^k نشان دهنده احتمال مربوط به عمل a_m^k است. انتخاب عمل a_m^k توسط LA_k باعث شدن LA_m می‌شود. r_k تعداد اعمال قابل انجام توسط اتوماتیک LA_k را نشان می‌دهد. برای کسب اطلاعات بیشتر در باره اتوماتیک یادگیر توزیع شده و کاربردهای آن می‌توان به [11,12,13,14,15] مراجعه نمود.



شکل ۲: یک اتوماتیک یادگیر توزیع شده با ۷ اتوماتیک یادگیر

۲-۲- تعیین ساختار ارتباطی بین صفحات و ب استفاده از اتوماتیک یادگیر توزیع شده

اگر تعدادی از کاربران، تعدادی از صفحات و ب را پی در پی درخواست کنند، احتمالاً این صفحات به نیازهای اطلاعاتی یکسانی پاسخ داده‌اند و در این صورت با هم دیگر ارتباط دارند. با استفاده از اتوماتیک یادگیر توزیع شده می‌توان میزان ارتباط بین صفحات و ب را با توجه به اطلاعات حرکتی کاربران در بین آن صفحات مشخص کرد [16,17,18]. با تعیین میزان ارتباط بین صفحات و ب می‌توان به رتبه‌بندی و یا خوشبندی صفحات و ب پرداخت. در روش‌های مبتنی بر اتوماتیک یادگیر توزیع شده، برای تعیین میزان ارتباط بین صفحات و ب، به هر صفحه و ب یک اتوماتیک یادگیر اختصاص داده می‌شود که وظیفه آن یادگیری میزان ارتباط آن صفحه با تمامی صفحاتی است که با آن صفحه ارتباط دارند. صفحات و ب و کاربران استفاده کننده از آن نقش یک محیط تصادفی را برای اتوماتیک یادگیر موجود در DLA ایفا می‌کنند. خروجی DLA بک دنباله از صفحات و ب مور شده توسط یک کاربر هستند که مسیر حرکت کاربر را به سمت یک صفحه و ب موردنظر نشان می‌دهد. محیط با استفاده از این دنباله پاسخی برای - DLA تولید می‌کند. با استفاده از این پاسخ ساختار داخلی اتوماتیک

$$PR(LA_k) = \frac{1-d}{n} + d \times \left\{ \sum_{i=1, i \neq k}^n PR(LA_i) \times P(LA_i, k) \right\} \quad (9)$$

که در رابطه فوق، اتوماتای یادگیر با شماره k (یک صفحه وب مثل k) بصورت LA_k و مقدار احتمال اقدام k در اتوماتای LA_i بصورت $P(LA_i, k)$ نشان داده شده است. پارامتر n ، تعداد صفحات وب را نشان می‌دهد. پارامتر d که عددی بین صفر و یک است بنام ضریب تغییر معروف است. الگوریتم پیشنهادی که از ایده الگوریتم PageRank [4] استفاده می‌کند، بصورت بازگشتی تبیه یک اتوماتا را از روی رتبه سایر اتوماتاها، بدست می‌آورد. اگر بخواهیم از رابطه فوق در سایر انواع اتوماتای یادگیر، مثلاً اتوماتای یادگیر با اقدام متغیر [10] استفاده نماییم، چنانچه اقدام مربوطه در اتوماتایی وجود نداشته باشد، مقدار احتمال آن اقدام صفر خواهد بود. (یعنی $P(LA_i, k) = 0$). معمولاً برای حل این نوع معادلات بازگشتی می‌توان از روش‌های عددی، مثل روش تکرار خطی استفاده کرد [4]. بطور مثال اگر بردار اعمال و بردار احتمال اتوماتای یادگیر LA_1 ، (A_2, A_3) ، $(2/3, 1/3)$ ، بردار اعمال و بردار احتمال اتوماتای یادگیر LA_2 ، $(0, 1)$ و بردار اعمال و بردار احتمال اتوماتای یادگیر LA_3 ، $(1, 0)$ و (A_1, A_2) باشند و ضریب تغییر $n = 3$ باشد، رتبه‌ها پس از حل معادله بازگشتی، بصورت زیر بدست می‌آینند. عدد بزرگتر، نشان دهنده بالاترین رتبه در بین تمام صفحات است.

$$\begin{aligned} PR(LA_1) &= \frac{0.5}{3} + 0.5 \times \{PR(LA_1) \times 1\} \\ PR(LA_2) &= \frac{0.5}{3} + 0.5 \times \{PR(LA_1) \times \frac{2}{3}\} \\ PR(LA_3) &= \frac{0.5}{3} + 0.5 \times \{PR(LA_1) \times \frac{1}{3} + PR(LA_2) \times 1\} \end{aligned}$$

$$\begin{aligned} PR(LA_1) &= \frac{7}{20} \approx 0.35 \\ PR(LA_2) &= \frac{17}{60} \approx 0.28 \\ PR(LA_3) &= \frac{11}{30} \approx 0.37 \end{aligned}$$

دهنده ارتباط کمتر است. بطور مثال اگر بردار اعمال و بردار احتمال اتوماتای یادگیر LA_1 به ترتیب (A_2, A_3, A_4) و $(0.2, 0.5, 0.3)$ باشند و کاربر حرکت $D_1 \rightarrow D_2$ را انجام دهد -

$$\begin{aligned} E_2^1 &= -(0.2 \log 0.2 + 0.8 \log 0.8) = 0.21 \\ a_2^1 &= 0.21 / (1 + 0.21) = 0.17 \quad \text{و در نتیجه بردار احتمال اعمال } LA_1 \text{ به } (0.35, 0.42, 0.23) \text{ تغییر پیدا می‌کند. اگر کاربر حرکت } D_1 \rightarrow D_3 \text{ را انجام دهد در این صورت } a_3^1 = 0.3 / (1 + 0.3) = 0.23 \text{ و در نتیجه بردار احتمال اعمال } LA_1 \text{ به } (0.154, 0.615, 0.231) \text{ تغییر پیدا می‌کند. این الگوریتم بصورت شکل ۳ می‌باشد.} \end{aligned}$$

۱- یک DLA متناظر با ساختار صفحات وب ایجاد کن

۲- بردار احتمالات اتوماتاها یادگیر در DLA را مقداردهی اولیه کن.

۳- برای هر کاربر در Log فایل انجام بده

۳-۱- برای هر مسیر کاربر در لاگ فایل انجام بده

۳-۱-۱- برای هر حرکت $D_k \rightarrow D_m$ در طول مسیر انجام بده

۳-۱-۱-۱- بردار احتمال اعمال اتوماتای یادگیر LA_k را طبق

الگوریتم یادگیری زیر بروز کن

$$\begin{aligned} p_m^k(n+1) &= p_m^k(n) + a_m^k [1 - p_m^k(n)] \\ p_j^k(n+1) &= (1 - a_m^k) p_j^k(n) \quad j \neq m \quad \forall j \\ a_m^k &= E_m^k / (1 + E_m^k) \\ E_m^k &= -(p_m^k \log p_m^k + (1 - p_m^k) \log(1 - p_m^k)) \end{aligned}$$

شکل ۳: الگوریتم موجود [18]

۳- روش پیشنهادی

پس از تعیین میزان ارتباط بین صفحات وب، می‌توان از مقداری بردار احتمال هر اتوماتای یادگیر استفاده کرد و به ترتیب بندی صفحات وب پرداخت. رابطه پیشنهادی برای انجام عمل رتبه‌بندی روی صفحات وب بصورت زیر است:

جدول ۲: پارامترهای شبیه سازی

0.7	حد آستانه ایجاد اتصال
2000	تعداد کاربران
30	تعداد صفحات
5	تعداد موضوعها
0.2	T مقدار ثابت صفحه اولیه (صفحه اولیه سایت) در موضوعات مختلف
-	ΔM ضریب ثابت کاهش اشتیاق کاربر
-	ΔM ضریب متغیر کاهش اشتیاق کاربر
1	a پارامتر توزیع قانون-توانی توزیع احتمال علایق کاربران
1.2	ضریب پاداش دریافتی از مشاهده یک صفحه f
0.5	ضریب جذب اطلاعات از یک صفحه توسط یک کاربر I
5.97	m_m میانگین توزیع نرمال ΔM
0.25	S_m واریانس توزیع نرمال ΔM
-	m میانگین توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع خاص
3	a پارامتر توزیع قانون-توانی توزیع احتمالهای وزنهای مطالب برای هر صفحه
0.25	S واریانس توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع خاص
1	ضریب کاهش علاقه کاربر q
0.2	حداقل اشتیاق کاربر برای ادامه جستجو

در این مدل، هر صفحه وب با یک بردار محتوا نشان داده می شود. طول این بردار برابر با تعداد موضوعهای موجود در سیستم است. هر عضو این بردار میزان ارتباط صفحه متناظر با آن بردار را با یکی از این صفحات نشان می دهد. با استفاده از بردار محتوای هر کدام از صفحات، شباهت بین هر دو صفحه موجود در سیستم محاسبه می شود. ماتریس شباهت بدست آمده بعنوان ماتریس ارتباطات ایدهآل بین صفحات و وب در شبیه سازیها به منظور ارزیابی کارایی روش پیشنهادی برای رتبه بندی استفاده می شود.

۲-۴- شاخص ارزیابی

با اجرای الگوریتم رتبه بندی بر روی مجموعه ای از صفحات، رتبه های متناظر با این صفحات بدست می آید. برای مقایسه رتبه های ایجاد شده توسط دو الگوریتم مختلف، از معیاری بنام کورولیشن ترتیب استفاده می شود. کورولیشن ترتیب بصورت زیر تعریف می شود:

در جدول ۱ از روش تکرار خطی استفاده کردند. مقدار رتبه اولیه صفحات در ابتدا برابر با $1/n$ در نظر گرفته شده و تا زمانیکه همگرایی حاصل نشده در طی تکرارهای مختلف، محاسبه رتبه ها ادامه یافته است. همگرایی زمانی حاصل می شود که در دو تکرار متوالی، اعضای بردار جواب، بیشتر از حد آستانه مشخصی تغییر نکرده باشند. برای مثال در جدول ۱ الگوریتم در تکرار ششم خاتمه می باید زیرا نتایج در تکرار ششم با نتایج در تکرار پنجم یکی است.

۴- نتایج شبیه سازیها

در این بخش نتایج بدست آمده از شبیه سازی الگوریتم پیشنهادی ارائه می شود.

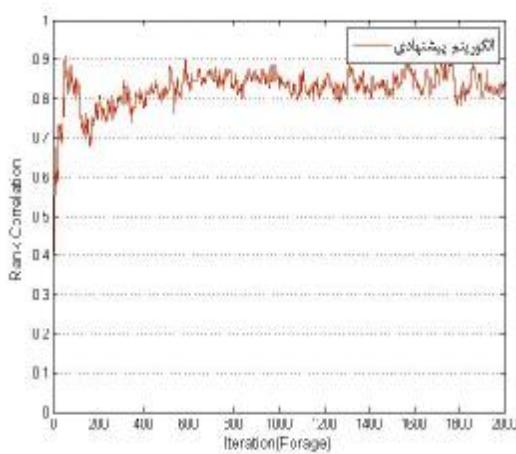
۴-۱- مدل شبیه سازی

برای شبیه سازی الگوریتم پیشنهادی، از مدل معروفی شده در [19] برای نشان دادن ساختار صفحات وب و چگونگی استفاده کاربران، استفاده شده است. اعتبار این مدل توسط Lui و همکاران [19] با استفاده از اطلاعات بدست آمده از چندین سایت وب بزرگ مانند مایکروسافت تایید شده است. بر این اساس، در این مقاله مطابق با مدل رفتار کاربران، پروفایل علاقه کاربران بصورت توزیع قانون-توانی و توزیع محتوای صفحات وب بصورت توزیع نرمال در نظر گرفته شده است. سایر پارامترهای استفاده شده در مدل [19] برای شبیه سازیها

جدول ۱: استفاده از روش تکرار خطی برای محاسبه رتبه صفحات

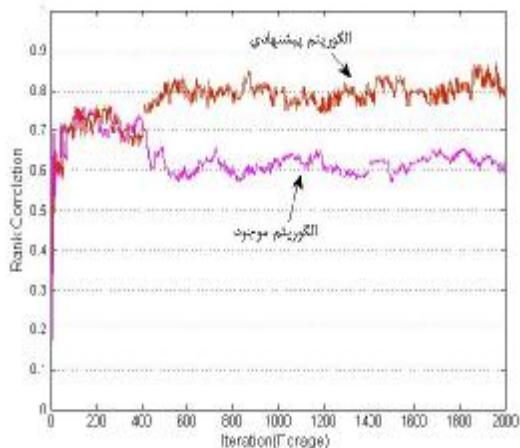
Iteration	PR(LA_1)	PR(LA_2)	PR(LA_3)
۰	۰.۳۳۳	۰.۳۳۳	۰.۳۳۳
۱	۰.۳۳۴	۰.۲۵۰	۰.۴۱۷
۲	۰.۳۷۶	۰.۲۵۱	۰.۳۷۳
۳	۰.۳۵۴	۰.۲۶۱	۰.۳۸۵
۴	۰.۳۶۰	۰.۲۵۶	۰.۳۸۴
۵	۰.۳۵۹	۰.۲۵۷	۰.۳۸۴
۶	۰.۳۵۹	۰.۲۵۷	۰.۳۸۴

انجام شده در این مقاله در جدول ۲ نشان داده شده است.



شکل ۴: نتیجه اجرای الگوریتم پیشنهادی

در این مقاله، برای انجام شبیه سازی، اندازه بردار احتمال برای هر اتوماتای یادگیر در DLA برابر با تعداد صفحات وب در نظر گرفته شده است. همانطوریکه از شکل ۴ مشخص است، کورولیشن ترتیب الگوریتم پیشنهادی بالا است. به منظور ارزیابی کارایی الگوریتم پیشنهادی، مقایسهای بین تنها الگوریتم موجود [5] برای رتبه بندی صفحات وب با استفاده از اتوماتای یادگیر توزیع شده و الگوریتم پیشنهادی، صورت گرفت. در این شبیه سازی پارامترها طبق جدول ۱ بطور مشابه برای هر دو الگوریتم یکسان فرض شد. شکل ۵ نتیجه این مقایسه را نشان می دهد.



شکل ۵: مقایسه روش پیشنهادی با تنها روش موجود [5]

همانطوریکه از شکل ۵ مشخص است، کارایی الگوریتم پیشنهادی در مقایسه با تنها روش موجود بالاتر می باشد. دلیل بالا بودن کارایی الگوریتم پیشنهادی در این است که روش پیشنهادی در محاسبه رتبه ها بسیار بهتر عمل می کند. همچنین نتیجه شبیه سازیها [18] نشان می دهد که الگوریتم یادگیری مورد استفاده [18] در روش پیشنهادی از

$$\text{RankCorrelation} = 1 - \left[\frac{6 \times \sum_{i=1}^n d_i}{n^3 - n} \right] \quad (10)$$

که d_i تفاوت بین رتبه عضو i ام مجموعه رتبه های ایجاد شده توسط دو الگوریتم متفاوت می باشد. پارامتر n تعداد صفحات وب را نشان می دهد. عدد بدست آمده که بین صفر و یک می باشد، نشان دهنده میزان نزدیک بودن رتبه های دو مجموعه از رتبه ها به یکدیگر شبیه تر می باشند. در شبیه سازی های انجام شده ابتدا ماتریس ارتباطات ایده آل رتبه بندی شده و نتیجه با رتبه های بدست آمده توسط الگوریتم پیشنهادی مورد مقایسه قرار می گیرد.

۳-۴- نتایج شبیه سازی

با افزایش تعداد حرکت کاربران در بین صفحات وب، ساختار ارتباطی بین صفحات وب، به مرور طبق الگوریتم یادگیری مشخص شده در شکل ۳ توسط اتوماتای یادگیر توزیع شده ایجاد می شود. این ساختار ارتباطی ایجاد شده باید به ساختار ارتباطی ایده آل که توسط مدل ارائه شده در [19] تعیین شده است، نزدیک باشد. شکل ۴ نتیجه شبیه سازی الگوریتم پیشنهادی را نشان می دهد. محور افقی مراحل یادگیری را تا تکرار ۲۰۰۰ (۲۰۰۰ کاربر به سیستم وارد شده است) نشان می دهد. برای مشخص کردن کورولیشن ترتیب، ابتدا رتبه های ساختار ارتباطی ایده آل را طبق فرمول (۹) تعیین می کنیم. برای انجام اینکار در فرمول (۹)، LA_i را صفحه i و $P(LA_i, k)$ را میزان ارتباط صفحه i با صفحه k طبق مقدار بدست آمده از مدل [19] فرض می کنیم. سپس در هر ۲۰۰ تکرار (۲۰۰ کاربر به سیستم اضافه می شود) رتبه های بدست آمده توسط الگوریتم پیشنهادی طبق رابطه (۹) محاسبه شده و کورولیشن ترتیب بین دو مجموعه از رتبه ها طبق رابطه (۱۰) مشخص می شود. محور عمودی، کورولیشن ترتیب بین دو مجموعه را نشان می دهد.

- Science Research Center(IPM),Computer Science Research Lab.,Tehran,Iran,PP.467-473, Iran, May 24-26.2006.
- [6] Lakshmivarahan, S., "Learning algorithms: theory and applications," New York: Springer-Verlag, 1981.
- [7] Mars, p., Chen, J.R, and Nambir, R., "Learning Algorithms: Theory and Applications in Signal Processing," Control, and Communication, CRC Press Inc., 1996.
- [9] Meybodi, M.R., and Lakshmivarahan, S., "On a class of Learning Algorithms which have Symmetric Behavior under Success and Failure," pp.145-155.Lecture Notes in Statistics, Berlin: SpringerVerlag, 1984.
- [10] Narendra, K. S., and Thathachar, M. A. L., "Learning automata: An introduction," Prentice Hall, 1989.
- [11] Thathachar, M.A.L, and Bhaskar, R.Harita,"Learning Automata with changing number of actions," IEEE Transaction on System, man and cybernetics, vol.SMC-17, No.6, Nov.1987.
- [12] Beigy, H. and Meybodi, M. R., "Utilizing Distributed Learning Automata to Solve Stochastic Shortest Path Problem", International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, World Scientific Publishing Company, to appear.
- [13] Meybodi, M. R. and Beigy, H., "Solving Stochastic Path Problem Using Distributed Learning Automata", Proceedings of The Sixth Annual International CSI Computer Conference, CSICC2001, Isfahan, Iran, pp. 70-86, Feb. 20- 22 , 2001
- [14] Beigy, H. and Meybodi, M. R., "A New Distributed Learning Automata Based Algorithm For Solving Stochastic Shortest Path Problem", Proceedings of the Sixth International Joint Conference on Information Science, Durham, USA, pp. 339-343, 2002
- [15] Alipour, M. and Meybodi, M. R., "Solving Traveling Salesman Problem Using Distributed Learning Automata", Proceedings of 10th Annual CSI Computer Conference, Computer Engineering Department, Iran Telecommunication Research Center, Tehran, Iran, pp. 759-761 Feb. 2005
- [16] Alipour, M. and Meybodi, M. R., "Solving Probabilistic Traveling Sales Man Problem Using Distributed Learning Automata", Proceedings of 11th Annual CSI Computer Conference of Iran, Fundamental Science Research Center (IPM), Computer Science Research Lab., Tehran, Iran, pp. 673-678, Jan. 24-26, 2006
- [17] Baradarani Hashemi, A., and Meybodi, M.R., "Web Usage Mining Using Distributed Learning Automata," Proceedings of the 12th Annual International CSI Computer Conference, CSICC2007, Tehran, Iran, pp.553-560, Feb020-22, 2007.
- [18] Saati, s. and Meybodi, M.R., "A Self Organizing Model for Document Structure Using Distributed Learning Automata," Proceedings of the Second International Conference on Information and Knowledge Technology (IKT2005), Tehran, Iran, May 24-26.2005.
- [19] Anari,B. and Meybodi,M.R., "A new method based on distributed learning automata for determining web documents structure," Proceedings of the 12th Annual International CSI Computer Conference,CSICC2007, Tehran,Iran,pp.2276-2281,Feb.20-22,2007.
- [20] Lui,J.,Zhang,S.,and Yang,J.,"Characterizing web usage Regularities with information Foraging Agents," IEEE no.4, pp.566-584,April 2004.

الگوریتم یادگیری موجود[17] بسیار بهتر عمل می کند. به منظور بررسی تاثیر تعداد صفحات وب و تعداد کاربران، آزمایش دیگر انجام شد. در این آزمایش حداکثر مقدار کورولیشن ترتیب به ازای تعداد صفحات وب و تعداد کاربران محاسبه شده است.

جدول ۳: حداکثر مقدار کورولیشن ترتیب به ازای تعداد صفحات و کاربران

Pages Users	۲۰	۶۰	۱۰۰	۲۰۰	۳۰۰
۱۰۰۰	۰.۷۶۴۱	۰.۷۶۲۵	۰.۷۶۰۵	۰.۷۵۶۵	۰.۷۵۱۰
۱۵۰۰	۰.۸۵۱۱	۰.۸۱۸۱	۰.۷۷۴۱	۰.۷۴۸۱	۰.۷۴۲۷
۳۰۰۰	۰.۸۶۳۲	۰.۸۰۱۲	۰.۷۴۹۸	۰.۷۳۵۸	۰.۷۱۳۵
۵۰۰۰	۰.۸۵۱۱	۰.۸۰۵۹	۰.۷۳۷۵	۰.۷۲۳۶	۰.۷۱۱۱

جدول ۳ نتیجه این آزمایش را نشان می دهد. همانگونه که از جدول ۳ مشخص است، افزایش تعداد صفحات وب، در میزان کارایی تاثیر گذار بوده و مقدار کورولیشن ترتیب را کم می کند. دلیل این امر این است که با افزایش تعداد صفحات وب، طول بردار اتوماتا افزایش یافته و میزان همگرایی کاهش می یابد. یک راه حل برای افزایش میزان همگرایی، استفاده از اتوماتای یادگیر با اقدام متغیر[10] می باشد.

۵- نتیجه گیری

در این مقاله، روش جدیدی با استفاده از اتوماتای یادگیر توزيع شده برای رتبه بندی صفحات وب ارائه شد. روش پیشنهادی بصورت بازگشتی رتبه هر اتوماتا را بر اساس رتبه سایر اتوماتاهای متصل به آن محاسبه می کند. از روش پیشنهادی می توان برای رتبه بندی سایر انواع اتوماتای یادگیر، مثلا اتوماتای یادگیر با اقدام متغیر[10] استفاده کرد. کارایی روش پیشنهادی در مقایسه با تنها روش موجود بالا می باشد.

مراجع

- [1] Brin,S. and Page,L., " The Anatomy of a Large-Scale Hypertextual web search Engine," Computer Networks and ISDN System,vol.30,pp.107-117,1998.
- [2] Page,L.,Brin,S.,Motwan,R. and Winograd," The page rank Citation Ranking:Bringing order to the web," Technical Report,StandfordDigital Libraries SIDL-WP-1999-0120,1999.
- [3] Brodin,A.,Robers,G.O.Rosenthal,J.S, and Tsaparas,P., " Link Analysis Ranking:Algorithms, Theory, and Experiments," ACM Transactions on Internet Technology, Vol.5,NO.1,PP.231-297,February,2005.
- [4] jianhan,zhu., "Mining Web Site Link Structures for Adaptive Web Site Navigation and Search," Ph.D Thesis, University of Ulster at Jordanstown,October 2003.
- [5] Saati, s. and Meybodi, M.R., "Document Ranking Using Distributed Learning Automata," Proceedings of 11th Annual CSI Computer Conference of Iran, Fundamental