دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

جزیره کیش، خلیج فارس، ایران ۱۹ الی ۲۱ اسفند ۱۳۸۶

انجمن کامپیوتر ایران
Computer Society of Iran

مقاله کامل

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

بابک اناری
عضو هیئت علمی
دانشگاه آزاد اسلامی واحد شبستر
anari322@yahoo.com

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

زهره اناری
دانشجوی کارشناسی ارشد نرم افزار
دانشگاه آزاد اسلامی واحدشبستر
zanari323@yahoo.com

چکیده: با افزایش روز افزون تعداد صفحات وب، مسئله جستجوی اطلاعات در وب توسط کاربران اهمیت زیادی پیدا می‌کند. کاربران وب دوست دارند کوئری خود را به موتور جستجو داده و در نهایت خروجی خودشان را دریافت کنند، این خروجی که دنباله ای متوالی 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 رتبه‌بندی استفاده می‌شود. یکی از نقطه ضعفهای الگوریتمهای رتبه بندی موجود در این است که این الگوریتمها فقط به ساختار ارتباطی بین صفحات وب توجه می‌کنند و به نحوه استفاده کاربران از یالهای ارتباطی توجهی نمی‌کنند. در این مقاله روشی 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 می‌شود که وظیفه آن یادگیری ارتباط آن صفحه با سایر صفحات وب دیگر است. الگوریتم پیشنهادی، با توجه به مقادیر بردار احتمال هر اتوماتا، رتبه هر صفحه وب را بصورت بازگشتی محاسبه می‌کند. نتایج شبیه-سازیها نشان داده است که روش پیشنهادی در مقایسه با تنها روش 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 می‌باشد. هدف از رتبه‌بندی صفحات وب، پیدا کردن صفحات مرتبط به هم است. رتبه‌بندی تکنیکی است که بوسیله آن می‌توان 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 می‌شود.

یک الگوریتم رتبه‌بندی برمبنای یکسری صفحات وب شروع بکار می‌کند. برحسب اینکه این مجموعه صفحات وب، چگونه استخراج ورتبه بندی می‌شوند دو نوع الگوریتم رتبه‌بندی خواهیم داشت: الگوریتمهای مستقل از کوئری و الگوریتمهای وابسته به کوئری [3]. یک الگوریتم رتبه‌بندی را مستقل از کوئری می‌گویند، اگر هنگام اعمال رتبه‌بندی، به رتبه‌بندی تمامی صفحات وب پرداخته شود. همچنین یک الگوریتم رتبه‌بندی را وابسته به کوئری می‌گویند، اگر هنگام اعمال رتبه‌بندی، فقط به رتبه‌بندی زیر مجموعه‌ای از صفحات مرتبط با کوئری کاربر پرداخته شود نه به رتبه‌بندی تمامی صفحات وب. مثلا الگوریتم رتبه-بندی PageRank [1,2] نمونه‌ای از الگوریتمهای مستقل از کوئری است و الگوریتمHITS [3] نیز نمونه‌ای از الگوریتمهای وابسته به کوئری است.

رتبه یک صفحه مثل A درالگوریتم رتبه بندی PageRank بصورت رابطه زیر محاسبه می‌شود.

$$PR(A)=\frac{1-d}{N}+d\times\left\{\frac{PR(T_1)}{C(T_1)}+\ldots+\frac{PR(T_n)}{C(T_n)}\right\} \qquad (۱)$$

که در آن N، تعداد صفحات وب$PR(T_i)$ ، رتبه صفحهT_i که به صفحهA لینک دارند.$C(T_i)$ ، تعداد یالهای صفحهT_i ، d ضریب تعدیل که عددی بین صفر و یک است. از آنجاییکه این الگوریتم رتبه صفحات دیگر را برای محاسبه استفاده می‌کند، این مقدار بصورت بازگشتی محاسبه می‌شود. ولی در عمل از روشهای عددی برای حل این معادله استفاده می‌شود. همانطوریکه از رابطه فوق مشخص است، عیب این الگوریتم این است که فقط به یالهای ارتباطی بین صفحات وب توجه می‌کند و به تعداد گذرهای کاربران در بین این یالها توجه نمی‌کند.

اگر تعداد گذرهای کاربران در بین صفحات وب در دسترس باشد، می‌توان از الگوریتم PageRate [4] بجای الگوریتمPageRank استفاده نمود. رتبه یک صفحه مثل A درالگوریتم رتبه بندی PageRate بصورت رابطه زیر محاسبه می‌شود.

$$PR(A)=\frac{1-d}{N}+d\times\left\{PR(T_1)\times\frac{n(T_1,A)}{\sum_i n(T_1,i)}+\ldots+PR(T_n)\times\frac{n(T_n,A)}{\sum_i n(T_n,i)}\right\} \qquad (۲)$$

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 بدلیل استفاده از ماتریس ارتباطات، استفاده از آن در مجموعه‌های بزرگ و قابل گسترش مناسب نمی‌باشد.

عیب تنها الگوریتم گزارش شده مبتنی بر اتوماتای 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 بدین صورت سازماندهی شده است: در بخش ۲ اتوماتای یادگیر و اتوماتای یادگیر توزیع شده و تعیین ساختار ارتباطی بین صفحات وب توسط اتوماتای یادگیر توزیع شده مورد بحث قرار خواهد گرفت. در بخش۳ روش پیشنهادی ارائه خواهد شد و در بخش 4 نتایج شبیه سازیها ارائه می‌گردد. بخش پایانی نتیجه‌گیری می‌باشد.

۲- اتوماتای 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 اتوماتای یادگیر داده می‌شود. اتوماتای یادگیر از این پاسخ استفاده نموده وعمل خود را برای مرحله بعد انتخاب می‌کند. شکل ۱ ارتباط بین اتوماتای یادگیر و محیط را نشان می‌دهد.

شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

محیط: محیط را می‌توان توسط سه‌تایی $E \equiv \{a, b, c\}$ نشان داد که در آن $a \equiv \{a_1, a_2, ..., a_r\}$ مجموعه ورودیها، $b \equiv \{b_1, b_2, ..., b_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, ..., c_r\}$ 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 a_i نتیجه نامطلوب داشته باشد می‌باشد. در محیط ایستا مقادیر c_i بدون تغییر می‌مانند، حال آنکه در محیط غیر ایستا این مقادیر در طی زمان تغییر می‌کنند. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختارمتغیر تقسیم بندی می‌گردند. در ادامه به شرح مختصری درباره اتوماتای یادگیر با ساختار متغیر که در این مقاله از آنها استفاده شده است می پردازیم.

اتوماتای یادگیر با ساختار متغیر : اتوماتای یادگیر با ساختار متغیر توسط ۴ تائی $\{a, b, p, T\}$ نشان داده می‌شود که در آن $a \equiv \{a_1, a_2, \mathbf{L}, a_r\}$ مجموعه عملهای اتوماتا $b \equiv \{b_1, b_2, \mathbf{L}, b_m\}$ مجموعه ورودیهای اتوماتا $p \equiv \{p_1, p_2, \mathbf{L}, p_r\}$ بردار احتمال انتخاب هر یک از عملها و $p(n+1) = T[a(n), b(n), p(n)]$ الگوریتم یادگیری می‌باشد. در این نوع از اتوماتاها، اگر عمل a_i در مرحله nام انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می‌یابند و برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالها افزایش می‌یابند. در هر حال، تغییرات به‌گونه‌ای صورت می‌گیرد تا حاصل جمع $p_i(n)$ ها همواره مساوی یک باقی بماند. الگوریتم زیر یک نمونه از الگوریتمهای یادگیری خطی در اتوماتای با ساختار متغیر است.

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \qquad (۳)$$
$$p_j(n+1) = (1-a)\,p_j(n) \qquad \forall j \quad j \neq i$$

الف- پاسخ مطلوب

$$p_i(n+1) = (1-b)\,p_i(n) \qquad (۴)$$
$$\forall j \;\; j \neq i \;\; p_j(n+1) = \frac{b}{r-1} + (1-b)\,p_j(n)$$

ب-پاسخ نامطلوب

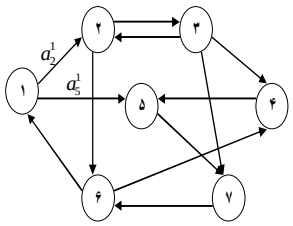
در روابط فوق، a پارامتر پاداش و b پارامتر جریمه می‌باشد. با توجه به مقادیر a و b سه حالت را می توان در نظر گرفت. زمانیکه a و b با هم برابر باشند، الگوریتم را L_{RP} می نامیم. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را L_{ReP} می نامیم. زمانیکه b مساوی صفر باشد، الگوریتم را L_{RI} می نامیم. برای مطالعه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

۲-۱- اتوماتای یادگیر توزیع شده

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

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

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



$DLA=(V,E)$
$V=\{LA_1, LA_2, ..., LA_n\}$
$E \subset V \times V$
$(LA_i, LA_j) \in E$

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

۲-۲-تعیین ساختار ارتباطی بین صفحات وب بااستفاده از اتوماتای یادگیر توزیع شده

اگر تعدادی از کاربران، تعدادی از صفحات وب را پی در پی درخواست کنند، احتمالا این صفحات به نیازهای اطلاعاتی یکسانی پاسخ داده‌اند و در این صورت باهمدیگر ارتباط دارند. با استفاده از اتوماتای یادگیر توزیع شده می‌توان میزان ارتباط بین صفحات وب را با توجه به اطلاعات حرکتی کاربران در بین آن صفحات مشخص کرد[16,17,18]. با تعیین میزان ارتباط بین صفحات وب می‌توان به رتبه‌بندی و یا خوشه‌بندی صفحات وب پرداخت. در روشهای مبتنی بر اتوماتای یادگیر توزیع شده، برای تعیین میزان ارتباط بین صفحات وب، به هر صفحه وب یک اتوماتای یادگیر اختصاص داده می‌شود که وظیفه آن یادگیری میزان ارتباط آن صفحه با تمامی صفحاتی است که با آن صفحه ارتباط دارند. صفحات وب و کاربران استفاده کننده از آن نقش یک محیط تصادفی را برای اتوماتای یادگیر موجود در DLA ایفا می کنند. خروجی DLA یک دنباله ازصفحات وب مرور شده توسط یک کاربر هستند که مسیر حرکت کاربر را به سمت یک صفحه وب مورد نظر نشان می‌دهد. محیط با استفاده از این دنباله پاسخی برای DLA تولید می‌کند. با استفاده از این پاسخ ساختار داخلی اتوماتای یادگیر در اتوماتای یادگیر توزیع شده طبق الگوریتم یادگیری بروز می شود.

هر کدام از اعمال یک اتوماتای 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 اتوماتای یادگیر می‌تواند بعنوان شناسه صفحه متناظر با آن اتوماتای یادگیر و بردار احتمالات، میزان ارتباط این صفحه با دیگر صفحه‌ها در مجموعه صفحات در نظر گرفته شود. برای هر صفحه P_i یک اتوماتای یادگیر LA_i در نظر می‌گیریم. انتخاب عمل j توسط اتوماتای یادگیرLA_i به معنی فعال کردن اتوماتای یادگیر LA_j متناظر با صفحه p_j می‌باشد. در صورتیکه عمل انتخاب شده k امین عمل اتوماتایLA_i باشد (یعنی $a_k^i = j$) احتمال متناظر این عمل یعنی p_k^i بعنوان میزان ارتباط صفحات i و j در نظر گرفته می‌شود.

با ورود یک کاربر به سیستم و مشاهده صفحه P_i، اتوماتای 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 اتوماتای LA_i انتخاب و به محیط اعمال می‌شود.

در این مقاله برای تعیین میزان ارتباط بین صفحات وب از الگوریتم [18] استفاده کرده‌ایم. اختلاف این الگوریتم با الگوریتمهای پیشین مبتنی بر اتوماتای یادگیر توزیع شده[16,17] در چگونگی بروز کردن بردار احتمال اعمال اتوماتای یادگیر در DLA می‌باشد. فرض کنید، $p^k = \{p_1^k, p_2^k, ..., p_r^k\}$ بردار احتمال اتوماتای 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 $D_k \to D_m$ را انجام دهد (از صفحه D_k به صفحه D_m حرکت کند) در این صورت اتوماتای یادگیر LA_k بردار احتمال اعمال خود را طبق الگوریتم یادگیری زیر بروز می‌کند.

$$p_m^k(n+1) = p_m^k(n) + a_m^k[1 - p_m^k(n)] \qquad (۵)$$

$$p_j^k(n+1) = (1 - a_m^k)\, p_j^k(n) \quad j \neq m \;\forall j \qquad (۶)$$

$$a_m^k = \frac{E_m^k}{1 + E_m^k} \qquad (۷)$$

$$E_m^k = -(p_m^k \log p_m^k + (1 - p_m^k)\log(1 - p_m^k)) \qquad (۸)$$

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

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 $D_1 \to D_2$ را انجام دهد -

$$E\frac{1}{2} = -(0.2 \log\ 0.2 + 0.8 \log\ 0.8) = 0.21$$

و $a_2^1 = 0.21/(1+0.21) = 0.17$ و در نتیجه بردار احتمال اعمال LA_1 به $(0.35,\ 0.42,\ 0.23)$ تغییر پیدا می‌کند. اگر کاربر حرکت $D_1 \to D_3$ را انجام دهد در این صورت $a_3^1 = 0.3/(1+0.3) = 0.23$ و در نتیجه بردار احتمال اعمال LA_1 به$(0.154, 0.615, 0.231)$ تغییر پیدا می‌کند. این الگوریتم بصورت شکل۳ می باشد.

۱- یک DLA متناظر با ساختار صفحات وب ایجاد کن
۲- بردار احتمالات اتوماتاهای 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 $D_k \to D_m$ در طول مسیر انجام بده
۳-۱-۱-۱- بردار احتمال اعمال اتوماتای یادگیر LA_k را طبق الگوریتم یادگیری زیر بروز کن

$$p_m^k(n+1) = p_m^k(n) + a_m^k[1 - p_m^k(n)]$$
$$p_j^k(n+1) = (1 - a_m^k)\, p_j^k(n)\ \ j \neq m\ \forall j$$
$$a_m^k = E_m^k/(1 + E_m^k)$$
$$E_m^k = -(p_m^k \log p_m^k + (1 - p_m^k)\log(1 - p_m^k))$$

شکل ۳: الگوریتم موجود[18]

۳- روش پیشنهادی

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

$$PR(LA_k) = \frac{1-d}{n} + d \times \left\{ \sum_{i=1, i\neq k}^{n} PR(LA_i) \times P(LA_i, k) \right\} \qquad (۹)$$

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 تعدیل معروف است. الگوریتم پیشنهادی که از ایده الگوریتمPageRate [4] استفاده می‌کند، بصورت بازگشتی رتبه یک اتوماتا را از روی رتبه سایر اتوماتاها، بدست می‌آورد. اگر بخواهیم از رابطه فوق در سایر انواع اتوماتای یادگیر، مثلا اتوماتای یادگیر با اقدام متغیر[10] استفاده نماییم، چنانچه اقدام مربوطه در اتوماتایی وجود نداشته باشد، مقدار احتمال آن اقدام صفر خواهد بود. (یعنی $P(LA_i, k) = 0$). معمولا 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) و$(2/3, 1/3)$، بردار اعمال و بردار احتمال اتوماتای یادگیر LA_2 (A_1, A_3) و$(0,1)$و بردار اعمال و بردار احتمال اتوماتای یادگیر LA_3 (A_1, A_2) و$(1,0)$ باشند و ضریب تعدیل 0.5 و $n = 3$ باشد، رتبه‌ها پس از حل معادله بازگشتی، بصورت زیر بدست می‌آیند. عدد بزرگتر، نشان دهنده بالاترین رتبه در بین تمام صفحات است.

$$PR(LA_1) = \frac{0.5}{3} + 0.5 \times \{PR(LA_1) \times 1\}$$
$$PR(LA_2) = \frac{0.5}{3} + 0.5 \times \left\{ PR(LA_1) \times \frac{2}{3} \right\}$$
$$PR(LA_3) = \frac{0.5}{3} + 0.5 \times \left\{ PR(LA_1) \times \frac{1}{3} + PR(LA_2) \times 1 \right\}$$

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

در جدول ۱ از روش تکرار خطی استفاده کرده‌ایم. 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 رتبه‌ها ادامه یافته است. همگرایی زمانی حاصل می‌شود که در دو تکرار متوالی، اعضای بردار جواب، بیشتر از حد آستانه مشخصی تغییر نکرده باشند. برای مثال در جدول ۱ الگوریتم در تکرارششم خاتمه می‌یابد زیرا نتایج در تکرار ششم با نتایج در تکرار پنجم یکی است.

۴- نتایج شبیه‌سازیها

در این بخش نتایج بدست آمده از شبیه سازی الگوریتم پیشنهادی ارا ئه می‌شود.

۴-۱- مدل شبیه سازی

برای شبیه سازی الگوریتم پیشنهادی، از مدل معرفی 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 سازیهای

جدول۱ : استفاده از روش تکرار خطی برای محاسبه رتبه صفحات وب

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.

جدول ۲: پارامترهای شبیه سازی

حد آستانه ایجاد اتصال	0.7
تعداد کاربران	2000
تعداد صفحات	30
تعداد موضوعها	5
T_c مقدار ثابت صفحه اولیه(صفحه اولیه سایت)در موضوعات مختلف	0.2
ΔM_t^c ضریب ثابت کاهش اشتیاق کاربر	-
ΔM_t^v ضریب متغیر کاهش اشتیاق کاربر	-
a_u پارامتر توزیع قانون-توانی توزیع احتمال علایق کاربران	1
f ضریب پاداش دریافتی از مشاهده یک صفحه	1.2
l ضریب جذب اطلاعات از یک صفحه توسط یک کاربر	0.5
m_m میانگین توزیع نرمال ΔM_t^v	5.97
s_m واریانس توزیع نرمال ΔM_t^v	0.25
m_t میانگین توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع خاص	-
a_p پارامتر توزیع قانون-توانی توزیع احتمالهای وزنهای مطالب برای هر صفحه	3
s واریانس توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع خاص	0.25
q ضریب کاهش علاقه کاربر	1
حداقل اشتیاق کاربر برای ادامه جستجو	0.2

در این مدل، هر صفحه وب با یک بردار محتوا نشان داده می‌شود. 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 می‌دهد. با استفاده از بردار محتوای هر کدام از صفحات، شباهت بین هر دو صفحه موجود در سیستم محاسبه می‌شود. ماتریس شباهت بدست آمده بعنوان ماتریس ارتباطات ایده‌آل بین صفحات وب در شبیه‌سازیها به منظور ارزیابی کارایی روش پیشنهادی برای رتبه‌بندی استفاده می‌شود.

۴-۲- شاخص ارزیابی

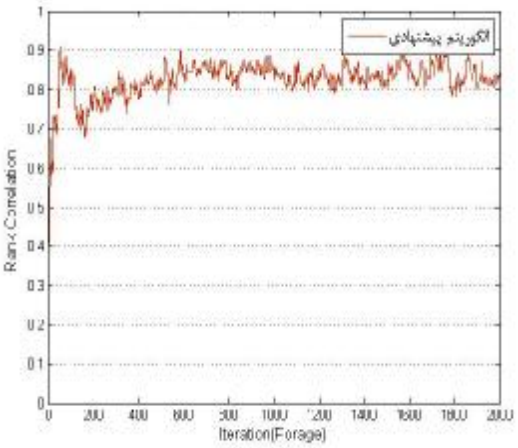
با اجرای الگوریتم رتبه‌بندی بر روی مجموعه‌ای از صفحات، رتبه‌های متناظر با این صفحات بدست می‌آید. برای مقایسه رتبه‌های ایجاد شده توسط دو الگوریتم مختلف، از معیاری بنام کورولیشن ترتیب استفاده می‌شود. کورولیشن ترتیب بصورت زیر تعریف می‌شود:

$$RankCorrelation = 1 - \left[\frac{6 \times \sum_{i=1}^{n} d_i}{n^3 - n} \right] \qquad (۱۰)$$

که d_i تفاوت بین رتبه عضو iام مجموعه رتبه های ایجاد شده توسط دو الگوریتم متفاوت می‌باشد. پارامتر n تعداد صفحات وب را نشان می‌دهد. عدد بدست آمده که بین صفر و یک می باشد، نشان دهنده میزان نزدیک بودن رتبه های دو مجموعه می باشد. هر چه این عدد به یک نزدیکتر باشد، این دو مجموعه از رتبه ها به یکدیگر شبیه تر می‌باشند. در شبیه‌سازیهای انجام شده ابتدا ماتریس ارتباطات ایده‌آل رتبه‌بندی شده و نتیجه با رتبه‌های بدست آمده توسط الگوریتم پیشنهادی مورد مقایسه قرار می‌گیرد.

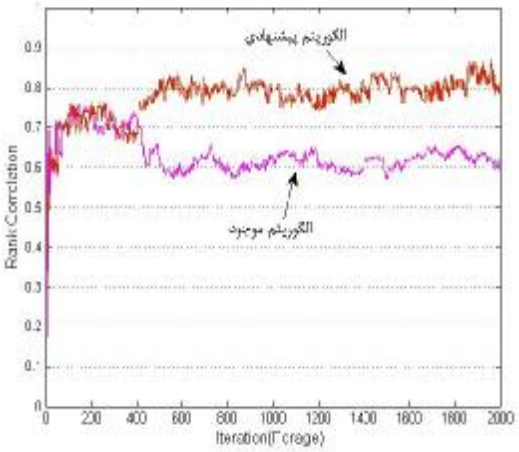
۴-۳- نتایج شبیه سازی

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 می‌شود. این ساختار ارتباطی ایجاد شده باید به ساختار ارتباطی ایده‌آل که توسط مدل ارائه شده در[19] تعیین شده است، نزدیک باشد. شکل۴ نتیجه شبیه سازی الگوریتم پیشنهادی را نشان می‌دهد. محور افقی مراحل یادگیری را تا تکرار ۲۰۰۰(۲۰۰۰ کاربر به سیستم وارد شده است) نشان می‌دهد. برای مشخص کردن کورولیشن ترتیب، ابتدا رتبه‌های ساختار ارتباطی ایده‌آل را طبق فرمول (۹) تعیین می‌کنیم. 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 می‌کنیم. سپس در هر ۲۰۰ تکرار(۲۰۰ کاربربه سیستم اضافه می‌شود) رتبه‌های بدست آمده توسط الگوریتم پیشنهادی طبق رابطه (۹) محاسبه شده و کورولیشن ترتیب بین دو مجموعه از رتبه‌ها طبق رابطه(۱۰) مشخص می‌شود. محور عمودی، کورولیشن ترتیب بین دو مجموعه را نشان می-دهد.



شکل ۴: نتیجه اجرای الگوریتم پیشنهادی

در این مقاله، برای انجام شبیه سازی، اندازه بردار احتمال برای هر اتوماتای یادگیر در DLA برابربا تعداد صفحات وب در نظر گرفته شده است. همانطوریکه از شکل۴ 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 شبیه‌سازی پارامترها طبق جدول۱ بطور مشابه برای هر دو الگوریتم یکسان فرض شد. شکل ۵ نتیجه این مقایسه را نشان می‌دهد.



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

همانطوریکه از شکل۵ مشخص است، کارایی الگوریتم پیشنهادی در مقایسه با تنها روش موجود بالاتر می‌باشد. دلیل بالا بودن کارایی الگوریتم پیشنهادی در این است که روش پیشنهادی در محاسبه رتبه ها بسیار بهتر عمل می‌کند. همچنین نتیجه شبیه‌سازیها[18] نشان می‌دهد که الگوریتم یادگیری مورد استفاده[18] در روش پیشنهادی از

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

جدول۳: حداکثر مقدار کورولیشن ترتیب به ازای تعداد صفحات و کاربران

Pages / Users	۲۰	۶۰	۱۰۰	۲۰۰	۳۰۰
۱۰۰۰۰	۰.۷۶۴۱	۰.۷۶۲۵	۰.۷۶۰۵	۰.۷۵۶۵	۰.۷۵۱۰
۱۵۰۰۰	۰.۸۵۱۱	۰.۸۱۸۱	۰.۷۷۴۱	۰.۷۴۸۱	۰.۷۴۲۷
۳۰۰۰۰	۰.۸۶۳۲	۰.۸۰۱۲	۰.۷۴۹۸	۰.۷۳۵۸	۰.۷۱۳۵
۵۰۰۰۰	۰.۸۵۱۱	۰.۸۰۵۹	۰.۷۳۷۵	۰.۷۲۳۶	۰.۷۱۱۱

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

۵- نتیجه‌گیری

در این مقاله، روش جدیدی با استفاده از اتوماتای یادگیر توزیع 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 می‌کند. از روش پیشنهادی می‌توان 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 می‌باشد.

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.,Bring,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 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*

[8] *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 cybernetice, 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] Baradaran Hashemi, A., and Meybodi, M.R.,"*Web Usage Mining Using Distributed Learning Automata,*" Proceedings of the 12th Annual International CSI Computer Conferenc, 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.