سیستم پیشنهاد دهنده وب با استفاده از اتوماتای یادگیر توزیع‌شده و پارتیشن‌بندی گراف

شهرزاد معتمدی مهر[۱]، مجید تاران[۲]، علی برادران هاشمی[۳]، محمدرضا میبدی[۴]

چکیده

هدف سیستم‌های پیشنهاددهنده وب هدایت کاربران به سمت صفحاتی است که به بهترین وجه نیازها و علایق آنها را برآورده سازد. در این مقاله یک الگوریتم جدید مبتنی بر اتوماتای یادگیر توزیع شده و پارتیشن بندی گراف پیشنهاد می‌گردد. در الگوریتم پیشنهادی یک اتوماتای یادگیر توزیع شده بر اساس داده‌های استفاده کاربران از وب و گراف پیوند بین صفحات، شباهت صفحات یک سایت با یکدیگر را مشخص می‌کند. سپس یک الگوریتم **PageRank** بر اساس این اتوماتای یادگیر توزیع شده امتیاز صفحات را محاسبه می‌کند. الگوریتم پیشنهادی با ایجاد یک مدل مارکوف بر اساس اطلاعات فوق صفحات جدیدی را برای ادامه حرکت هر کاربر در سایت به وی پیشنهاد می‌دهد. نتایج آزمایشات انجام شده نشان می دهد که الگوریتم پیشنهادی در مقایسه با روش های گزارش شده مبتنی بر قوانین انجمنی و اتوماتای یادگیر توزیع شده از دقت بیشتری برخودار است.

کلمات کلیدی

اتوماتای یادگیر، داده کاوی استفاده از وب، سیستمهای پیشنهاد دهنده

Web Recommendation using Distributed Learning Automata and Graph Partitioning

Shahrzad Motamedi Mehr; Majid Taran; Ali B. Hashemi; M.R. Meybodi

ABSTRACT

Recommendation systems aim at directing users toward the resources that best meet their needs and interests. One of the challenging tasks in improving web recommendation algorithms is the simultaneous use of users's activity log and hyperlink graph of the web site. In this paper, we propose a new recommendation algorithm based on web usage data and hyperlink graph of a web site. In the proposed algorithm, a distributed learning automata learns similarity between web pages of a web site using web usage data and hyperlink graph of the web site. Then, a usage based page rank for all pages of the web site is calculated using the probabilities of actions in the distributed learning automata. The proposed algorithm uses these information to build a Markov model which will be used to recommend new web pages for a user. Experiments show that the proposed method outperforms Association Rule Mining algorithm and the only learning automata based method reported in the literature in terms of precision and coverage.

KEYWORDS

Learning Automata, Web Usage Mining, Recommendations systems

1. مقدمه

وب، محیطی وسیع، متنوع و پویا است که کاربران متعدد اسناد خود را در آن منتشر می کنند. وب طی یک فرآیند آشفته و غیر متمرکز رشد می‌کند و این روند منجر به تولید حجم وسیعی از مستندات متصل به یکدیگر گشته است که از هیچ گونه سازماندهی منطقی برخوردار نیستند . با توجه به حجم وسیع اطلاعات در وب، مدیریت آن با ابزارهای سنتی تقریبا غیر ممکن است و ابزارها و روش هایی نو برای مدیریت آن مورد نیاز است.

برای حل این مشکل، شخصی کردن وب به یک پدیده محبوب به منظور سفارشی کردن محیط های وب تبدیل شده است. هدف از سیستم‌های

[۱] دانشکده برق و مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد قزوین، قزوین، ایران، motamedi@tmu.ac.ir
[۲] دانشکده برق و مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد قزوین، قزوین، ایران، m_taran@isc.iranet.net
[۳] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، a_hashemi@aut.ac.ir
[۴] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، mmeybodi@aut.ac.ir

شخصی‌ساز فراهم‌کردن نیازهای کاربران، بدون اینکه به طور صریح آن ها را بیان کنند یا نشان بدهند، می‌باشد[۴]. شخصی‌سازی وب مجموعه‌ای از عملیات است که تجربهء وب را برای یک کاربر خاص یا مجموعه ای از کاربران سازمان دهی می‌کند[۵].

روش های وب کاوی بر اساس آن که چه نوع داده ای را مورد کاوش قرار می دهند، به سه دسته داده کاوی محتوای وب[۱]، داده کاوی ساختار وب[۲] و داده کاوی استفاده از وب[۳] تقسیم می شوند[۲۳]. داده کاوی محتوای وب، فرآیند استخراج اطلاعات مفید از محتوای مستندات وب است . داده کاوی ساختار وب به کشف اطلاعات جدید با استفاده از پیوندهای[۴] بین صفحات وب می پردازد[۹] [۱۰] [۱۱]. داده کاوی استفاده از وب نیز داده های مربوط به استفاده کاربران از وب را مورد کاوش قرار می دهد و الگوهای استفاده از وب را به منظور درک و برآوردن بهتر نیازهای کاربران استخراج می کند. [۷] [۸]. بخش عمدهی فعالیت ها و تحقیقات انجام شده در وب کاوی به محتوای صفحات وب می پردازند. اما در سال های اخیر داده کاوی ساختار وب و داده کاوی استفاده از وب نیز مورد توجه قرار گرفته اند.

در [۵] از روش مبتنی بر قوانین انجمنی (AR[۵]) که با استفاده از کاوش آیتم های تکراری به دسته بندی صفحات می پردازد، استفاده شده است. الگوریتم ارائه شده در[۱۲] مبتنی بر آنالیز لینک ها می باشد که صفحات وب و کاربران سایت را به صورت گره و ابرپیوند مدل می کند و از الگوریتم HITS برای ارزیابی اهمیت آنها در گراف استفاده می کند و هدف آن اندازه گیری تخصص کاربران و اهمیت صفحات وب است. در[۱۳] دو متد مجزای رتبه بندی بر اساس آنالیز لینک ها ارائه داده شده است. Site Rank و Popularity Rank، الگوریتم‌های رتبه‌بندی هستندو از آنها در گراف سایت استفاده می شود. در [۱۳] توزیع و رتبه بندی دو متد ارائه و مقایسه شده است. Mobasher از درجه اتصالات بین صفحات سایت به عنوان فاکتوری تعیین کننده برای پیشنهاد بر اساس کاوش آیتم های تکرار شونده یا کشف الگوهای ترتیبی استفاده می کند[۱۴] ولی هیچ روشی تکنیک های آنالیز لینک ها را به طور کامل با فرایند شخصی سازی بوسیله استخراج اعتبار یا اهمیت صفحات وب در گراف ترکیب نکرده است. اتوماتای یادگیر توزیع شده قبلاً برای رتبه بندی صفحات وب [۲] و تعیین شباهت اسناد وب بکاربرده شده است [۳]. الگوریتم پیشنهادی مانند[۱] [۶] یک روش مبتنی بر اتوماتای یادگیر توزیعی است. نتایج بررسی ها نشان می دهد که الگوریتم[۱] در تعداد صفحات بالا نسبت به الگوریتم پیشنهادی ما کارایی پایین تری دارد.

در این مقاله با ترکیب داده‌های استفاده کاربران و داده‌های ساختاری صفحات وب الگوریتمی ترکیبی مبتنی بر اتوماتای یادگیر توزیع شده، پارتیشن بندی گراف و الگوریتم PageRank به منظور پیشنهاد صفحات ارائه شده است. در روش پیشنهادی کاربر با صرف کمترین زمان به نتایج مطلوب خود دست می یابد. در الگوریتم پیشنهادی رابطه تراگذری و استفاده از ساختار گراف وب جهت بهبود اضافه شده است. رابطه تراگذری به این معنی که اگر کاربری ابتدا صفحه a سپس b و در انتها c را مشاهده کرده است، پاداش در اتوماتای a به حرکت b و با یک ضریب ثابت کاهشی به حرکت c داده می شود. در مورد جریمه نیز به همین صورت می باشد با این تفاوت که در دادن جریمه ضریب ثابت، افزایشی می باشد. از دیگر ویژگی های این الگوریتم، استفاده از ساختار گراف وب می باشد. صفحاتی که در ساختار گراف وب به هم متصل می باشند دارای اولویت بالایی در پیمایش کاربر می باشند.

ویژگی دیگر الگوریتم پیشنهادی این است که با افزایش تعداد صفحات، مقدار دقت کم نمی شود. برای بالا بردن کارایی الگوریتم با تعداد صفحات زیاد از پارتیشن بندی گراف استفاده شده است. الگوریتم پیشنهادی بر این ایده استوار است که کاربر با ورود به شبکه اینترنت در یک محدوده خاص به دنبال اهداف خود می باشد. با پارتیشن بندی در واقع محدوده وب را برای کاربران کوچک کرده و سعی می کنیم نیاز کاربران را در آن محدوده بر طرف سازیم. به منظور کاهش اثر اطلاعات ناصحیح،کاربری که از پارتیشن خود خارج شود مسیر اشتباهی طی کرده و میزان شباهت محاسبه شده برای صفحات مسیر خارج از محدوده، با توجه به رابطهای مشخص کاهش می یابد. جریمه دیگری که در روش پیشنهادی برای کاربر در نظرگرفته شده وجود دور در مسیر پیمایشی کاربر می باشد.

برای بررسی کارایی الگوریتم پیشنهادی و مقایسه آن با سایر الگوریتم ها از صفحات واقعی وب و داده‌های واقعی کاربران وب استفاده شده است. همچنین جهت پارتیشن بندی گراف از الگوریتم های چند سطحی[۶] استفاده شده است[۲۸]. این نوع از الگوریتم های پارتیشن بندی در ۳ فاز اجرا می شوند. در فاز اول سایز گراف وب کوچک می شود[۷]. در فاز دوم[۸] گراف تبدیل شده در فاز ۱ بر اساس روشهای سنتی پارتیشن بندی شده و در فاز سوم گراف به حالت اولیه تبدیل می شود که به آن فاز پالایش[۹] گفته می شود.

نتایج شبیه سازیها نشان داده است که روش پیشنهادی در مقایسه با روش های گزارش شده مبتنی بر اتوماتای توزیع شده با تعداد اسناد بیشتر در تشخیص شباهت صفحات از دقت بالاتری برخوردار است.

در ادامه ابتدا در بخش۲ اتوماتای یادگیر و اتوماتای یادگیر توزیع‌شده به اختصار معرفی می‌شوند. دربخش ۳ الگوریتم PageRank به طور اجمالی بررسی میشود و در بخش ۴ الگوریتم پیشنهادی ارائه می‌گردد. در بخش۵ پس از معرفی مدل استفاده شده برای شبیه‌سازی ، نتایج شبیه‌سازی ارائه و بررسی می‌گردد. بخش ۶ نتیجه‌گیری می‌باشد.

1. اتوماتای یادگیر

اتوماتای یادگیر یک مدل انتزاعی است که بطور تصادفی یک اقدام از مجموعه متناهی اقدام‌های خود را انتخاب کرده و بر محیط اعمال می‌کند. محیط اقدام انتخاب شده توسط اتوماتای یادگیر را ارزیابی کرده و نتیجه ارزیابی خود را توسط یک سیگنال تقویتی به اتوماتای یادگیر اطلاع می‌دهد. سپس اتوماتای یادگیر با اطلاع از اقدام انتخاب شده و سیگنال تقویتی، وضعیت داخلی خود را بروز کرده و اقدام بعدی خود را انتخاب می‌کند. شکل ۱ نحوه ارتباط بین اتوماتای یادگیر و محیط را نشان می‌دهد.

شکل ۱. ارتباط اتوماتای یادگیر با محیط

محیط را می‌توان توسط سه‌تایی $E=\{\alpha,\beta,c\}$ نشان داد که در آن $\alpha=\{\alpha_1,\alpha_2,...,\alpha_r\}$ مجموعه ورودیها، $\beta=\{\beta_1,\beta_2,...,\beta_r\}$ مجموعه خروجیها و $c=\{c_1,c_2,...,c_r\}$ مجموعه احتمالات جریمه می‌باشد. هرگاه β مجموعه دو عضوی باشد، محیط از نوع P می‌باشد. در چنین محیطی $\beta_1=1$ به عنوان جریمه و $\beta_2=0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع Q، مجموعه β دارای تعداد متناهی عضو می‌باشد و در محیط از نوع S، تعداد اعضا مجموعه β نامتناهی است. c_i نشان دهنده احتمال نامطلوب بودن سیگنال تقویتی محیط در پاسخ به اقدام α_i می‌باشد. در یک محیط ایستا[۱۰] مقادیر c_iها ثابت هستند، حال آنکه در یک محیط غیر ایستا[۱۱] این مقادیر در طی زمان تغییر می‌کنند. بر اساس اینکه تابع بروز رسانی وضعیت اتوماتای یادگیر (که با اطلاع از اقدام انتخاب شده و سیگنال تقویتی β، وضعیت بعدی اتوماتای یادگیر را محاسبه می‌کند) ثابت یا متغیر باشد، اتوماتای یادگیر به دو دسته اتوماتای یادگیر با ساختار ثابت و اتوماتای یادگیر با ساختار متغیر تقسیم می‌گردند [۱۷]. در این مقاله از اتوماتای یادگیر با ساختار متغیر استفاده شده است که در ادامه معرفی می‌شود.

اتوماتای یادگیر با ساختار متغیر توسط چهارتایی $\{\alpha,\beta,p,T\}$ نشان داده می‌شود که در آن $\alpha=\{\alpha_1,\alpha_2,\cdots,\alpha_r\}$ مجموعه اقدام‌های اتوماتا ی یادگیر، $\beta=\{\beta_1,\beta_2,\cdots,\beta_r\}$ مجموعه ورودیهای اتوماتا ی یادگیر، $p=\{p_1,p_2,\cdots,p_r\}$ بردار احتمال انتخاب هر یک از اقدامها و T، $p(n+1)=T[\alpha(n),\beta(n),p(n)]$ الگوریتم یادگیری اتوماتای یادگیر می‌باشد. الگوریتم‌های یادگیری متنوعی برای اتوماتای یادگیر ارائه شده است که در ادامه یک الگوریتم یادگیری خطی برای اتوماتای یادگیر بیان می‌گردد. فرض کنید اتوماتا ی یادگیر در مرحله nاُم اقدام α_i خود را انتخاب نموده و محیط ارزیابی خود را توسط سیگنال تقویتی $\beta(n)$ به اتوماتای یادگیر اعلام کند.

اتوماتای یادگیری که در بالا معرفی شد، دارای تعداد اقدامهای ثابتی می‌باشد. در بعضی از کاربردها به اتوماتای یادگیر با تعداد اقدام متغیر[۱۲] نیاز می‌باشد[۲۰]. یک *اتوماتای یادگیر با تعداد اقدام متغیر*، در لحظه n، اقدام خود را از یک زیر مجموعه غیر تهی از اقدامها بنام مجموعه اقدامهای فعال $V(n)$ انتخاب می‌کند. انتخاب مجموعه اقدامهای فعال اتوماتای یادگیر $V(n)$ توسط یک عامل خارجی و بصورت تصادفی انجام می‌شود. نحوه فعالیت این اتوماتای یادگیر بصورت زیر است.

اتوماتای یادگیر برای انتخاب یک اقدام در زمان n ابتدا مجموع احتمال اقدامهای فعال خود ($K(n)$) را محاسبه و بردار $\hat{p}(n)$ را مطابق رابطه (۱) ایجاد می‌کند. آنگاه اتوماتای یادگیر یک اقدام از مجموعه اقدام‌های *فعال* خود را بصورت تصادفی و بر اساس بردار احتمال $\hat{p}(n)$ انتخاب کرده و بر محیط اعمال می‌کند. در یک اتوماتای یادگیر با الگوریتم یادگیری خطی، اگر اقدام انتخاب شده α_i باشد، اتوماتای یادگیر پس از دریافت پاسخ محیط، بردار احتمال $\hat{p}(n)$ اقدامهای خود در صورت دریافت پاسخ مطلوب بر اساس رابطه (۲) و در صورت دریافت پاسخ نامطلوب طبق رابطه (۳) بِروز می‌کند. سپس اتوماتای یادگیر بردار احتمال اقدامهای خود $p(n)$ را با استفاده از بردار $\hat{p}(n+1)$ و طبق رابطه (۴) بِروز می‌کند.

$$K(n)=\sum_{\alpha_i\in V(n)} p_i(n)$$

$$\hat{p}_i(n)=prob[\alpha(n)=\alpha_i \mid \alpha_i\in V(n)]=\frac{p_i(n)}{K(n)} \qquad (۱)$$

$V(n)$ is the set of enabled actions

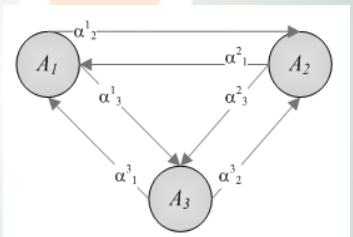
$$\hat{p}_i(n+1)=\hat{p}_i(n)+a.(1-\hat{p}_i(n))$$
$$\hat{p}_j(n+1)=\hat{p}_j(n)-a.\hat{p}_i(n) \qquad \forall j \;\; j\neq i \qquad (۲)$$

$$\hat{p}_i(n+1) = (1-b).\hat{p}_i(n)$$
$$\hat{p}_j(n+1) = \frac{b}{\hat{r}-1} + (1-b)\hat{p}_j(n) \qquad \forall j \;\; j \neq i \tag{۳}$$

$$\mathrm{p_i(n+1) = \hat{p}_i(n+1).K(n)} \qquad \text{for all i}, \alpha_i \in V(n)$$
$$\mathrm{p_j(n+1) = p_j(n)} \qquad \text{for all j}, \alpha_j \notin V(n) \tag{۴}$$

۱.۲. اتوماتای یادگیر توزیع شده

اتوماتای یادگیر توزیع‌شده [۱۸] شبکه‌ای از چند اتوماتای یادگیر است که برای حل یک مساله مشخص با یکدیگر همکاری می‌کنند. یک اتوماتای یادگیر توزیع‌شده را می‌توان با یک گراف جهت‌دار مدل کرد. بصورتی که مجموعه گره‌های آنرا مجموعه‌ای از اتوماتای یادگیر و یالهای خروجی هر گره مجموعه اقدامهای متناظر با اتوماتای یادگیر متناظر با آن گره است. هنگامی که اتوماتا یکی از اقدامهای خود را انتخاب می‌کند، اتوماتایی که در دیگر انتهای یال متناظر با آن اقدام قرار دارد، فعال می‌شود. بعنوان مثال در شکل ۲ هر اتوماتا ۲ اقدام دارد. اگر اتوماتای A_1 اقدام α_3 خود را انتخاب کند، آنگاه اتوماتای A_3 فعال خواهد شد. در گام بعد، اتوماتای A_3 یکی از اقدامهای خود را انتخاب می‌کند که منجر به فعال شدن یکی از اتوماتاهای یادگیر متصل به A_3 می‌شود. در هر لحظه فقط یک اتوماتای یادگیر در اتوماتای یادگیر توزیع‌شده فعال می‌باشد. بصورت رسمی، یک اتوماتای یادگیر توزیع‌شده با n اتوماتای یادگیر توسط یک گراف (A,E) تعریف می‌شود که $A=\{A_1,A_2,\ldots,A_n\}$ مجموعه اتوماتا و $E \subset A \times A$ مجموعه لبه‌های گراف است بطوریکه لبه (i,j) متناظر با اقدام a_j از اتوماتای A_i است. اگر بردار احتمال اقدامهای اتوماتای یادگیر A_j با p^j نشان داده شود، آنگاه p_m^j احتمال انتخاب اقدام α_m از اتوماتای یادگیر A_j را نشان می‌دهد که احتمال انتخاب لبه خروجی (j,m) از میان لبه‌های خروجی گره j می‌باشد.



شکل ۲. اتوماتای یادگیر توزیع شده

۳. الگوریتم PageRank

الگوریتم PageRank معروفترین الگوریتم تحلیل پیوند صفحات وب می باشد که در سال ۱۹۹۸ ارائه گردید [۱۹] [۲۰]. این الگوریتم با انتساب وزن به هر صفحه نتایج جستجو را بر اساس این وزن ها مرتب می کند. از دید الگوریتم PageRank صفحه ای مهم و معتبر است که از صفحات مهم و معتبر دیگر مورد اشاره باشد. در این دیدگاه تنها به وجود پیوندها توجه نمی شود، بلکه کیفیت آنها نیز مورد استفاده قرار می گیرد. این الگوریتم جستجو در وب را بصورت یک حرکت تصادفی در نظر می گیرد که در آن گردشگر بطور تصادفی یک لینک را انتخاب نموده و لینک ها را پشت سر هم دنبال می کند تا به هدف برسد، همچنین در صورت شروع یک مسیر دیگر برای پیمایش، به صفحه جدید پرش می کند. به عبارت دیگر الگوریتم PageRank به عنوان مدلی از حرکت تصادفی بر روی وب در نظر گرفته می شود[۱۹]. این مدل معادل با یک زنجیره مارکوف می باشد. حالت‌های این زنجیره صفحات وب هستند و گذار از یک حالت به حالت دیگر معادل انتخاب یک پیوند در صفحه جاری و رفتن به صفحه مورد اشاره است.

فرض کنید $G = (V, E)$ گراف متناظر وب باشد. $u \to v$ را به عنوان وجود پیوند از صفحه u به v و $\deg(u)$ را برابر با درجه خروجی صفحه u در نظر می‌گیریم. فرض کنیم p ماتریس انتقال گذار بین صفحات باشد. اگر گردش تصادفی در لحظه k در صفحه u قرار داشته باشد در لحظه $k+1$ به احتمال مساوی طبق رابطه (۵) به یکی از صفحات مجاور $\{v \mid u \to v\}$ خواهد رفت.

$$p_{ij} = \begin{cases} \dfrac{1}{\deg(u_i)} : if \quad (u_i \to u_j) \in E \\ 0 \qquad\quad : otherwise \end{cases} \tag{۵}$$

برای اینکه زنجیره مارکوف دارای توزیع پایدار باشد، هیچ صفحه‌ای نباید درجه خروجی صفر داشته باشد. در گراف وب صفحات زیادی وجود دارند که بدون پیوند هستند. برای رفع این مشکل فرض می‌شود وقتیکه گردش‌گر تصادفی به چنین صفحاتی می‌رسد، به طور تصادفی و با احتمال برابر از صفحه‌ای دیگر شروع به گردش می‌کند. اگر n تعداد صفحات وب و $\vec{v}$ بردار ستونی n بعدی توزیع یکنواخت احتمال باشد و $\vec{d}$ بردار

ستونی n بعدی مشخص‌کننده رئوس با درجه خروجی صفر باشد($d_u = 1$ به معنای درجه خروجی صفر برای راس u می‌باشد) در این صورت ماتریس p' طبق رابطه (۶) بدست می آید.

$$D = \vec{d}\,\vec{v}^T$$
$$P' = P + D \qquad (۶)$$

شرط بعدی لزوم توزیع احتمالی پایدار در زنجیره‌های مارکوف، کاهش‌ناپذیر بودن و به عبارت دیگر به شدت همبند بودن گراف وب است. برای رفع این کمبود فرض می‌کنیم که گردشگر با یک احتمال یکنواخت و بسیار کم از هر صفحه به هر صفحه دیگر می‌تواند جهش داشته باشد. با افزودن این ویژگی گراف وب به شدت همبند شده و دارای توزیع احتمالی پایدارخواهد بود. در این صورت ماتریس گذار طبق رابطه (۷) بدست می آید.

$$E = [1]_{n\times 1}\,\vec{v}^T$$
$$P'' = cP' + (1-c)E \quad ,0 \le c \le 1 \qquad (۷)$$

که در آن c ضریب تعدیل با مقدار معمول ۰.۸۵ می‌باشد. $\vec{v}$ بردار شخصی‌سازی نامیده می‌شود که می‌تواند به نفع صفحات خاص با مقداردهی بیشتر در درایه مورد نظر آن صفحه، تنظیم شود[۲۱].

در [۲۲] Eirinaki و Vazirgiannis الگوریتم PageRank شخصی سازی مبتنی بر استفاده (UPR[۱۳]) را که بر اساس رفتار کاربران قبلی عمل می کند، برای رتبه بندی صفحات وب پیشنهاد کرده اند. w_i وزن صفحه i است که مدت زمان مشاهده صفحه i و w_{ij} وزن انتقال از صفحه i به j است و مدت زمان انتقال به صفحه j که بلافاصله بعد از صفحه i مشاهده می شود را نشان می دهد. در UPR احتمال انتقال از صفحه i به j متناسب با لینک به صفحه j درصفحه i است و طبق رابطه (۸) بدست می آید.

$$p_{ij}^{UPR} = \frac{w_{ij}}{\sum_{x_k \in out(k)} w_{ik}} \qquad (۸)$$

بطوری که $out(i)$ مجموعه صفحات لینک شده از صفحه i را مشخص می کند. بردار شخصی سازی $\vec{v}$ بر اساس رابطه (۹) تعریف می شود.

$$v_{ij}^{UPR} = \frac{w_i}{\sum_{\forall k} w_k} \qquad (۹)$$

4 . الگوریتم پیشنهادی

الگوریتم ارائه شده, صفحات وب جدید را بر اساس پیمایش کاربر, استفاده کاربران قبلی و اطلاعات پیوندی گراف سایت به کاربر جاری پیشنهاد می دهد. بدین منظور اتوماتای یادگیر توزیع شده شباهت بین صفحات وب را با استفاده از اطلاعات پیمایش کاربران قبلی و پیوند گراف وب سایت تعیین می کند. سپس یک رتبه بندی جدید صفحه مبتنی بر استفاده از وب, با بکاربردن شباهت تعیین شده محاسبه می شود. رتبه صفحه و شباهت بین صفحات محاسبه شده برای ایجاد مدل زنجیره مارکوف انتقال های کاربران در وب سایت استفاده می گردد. این مدل مارکوف برای پیش بینی احتمال ملاقات صفحات جدید برای پیشنهاد به کاربر استفاده می شود. جزئیات الگوریتم پیشنهادی در زیر آمده است.

۱.۴. تعیین شباهت بین صفحات

الگوریتم پیشنهادی، یک اتوماتای یادگیر توزیع شده را بکار می گیرد که شباهت بین صفحات را با استفاده از فعل وانفعال کاربر و گراف پیوندها تعیین می کند. الگوریتم ارائه شده در زیر تشریح شده است.

ابتدا گراف وب سایت با استفاده از نرم افزار Metis به k پارتیشن تقسیم می شود. Metis ابزاری مفید برای پارتیشن بندی گراف می باشد و بصورت رایگان در دسترس است. این نرم افزار با بکاربردن الگوریتم های چند سطحی بر روی گراف بدون ساختار کار می کند.

برای تعیین شباهت بین صفحات در یک مجموعه با n صفحه، از یک اتوماتای یادگیر توزیع شده با n اتوماتای یادگیر که هر یک n-۱ عمل دارند، استفاده می‌شود. برای هر اتوماتای یادگیر در هر زمان تنها یک زیرمجموعه از عمل هایش فعال و قابل استفاده هستند. هر کدام از اعمال یک اتوماتای یادگیر، متناظر با یکی از صفحات در مجموعه صفحات و احتمال انتخاب این عمل در بردار احتمالات، ارتباط این صفحه با صفحه متناظر با آن عمل می باشد. برای هر صفحه j یک اتوماتای یادگیر در نظر می گیریم. انتخاب عمل j توسط اتوماتای یادگیر i به معنی فعال کردن اتوماتای یادگیر j متناظر با صفحه j می باشد. در صورتیکه عمل انتخاب شده k امین عمل اتوماتای i باشد یعنی ($a^i_k = j$) احتمال متناظر این عمل یعنی p^i_k بعنوان میزان ارتباط صفحه های i و j در نظر گرفته می شود.

هنگامی که کاربری در صفحه i قرار دارد، اتوماتای یادگیر متناظر با آن فعال است. حرکت کاربر از صفحه i به صفحه j، به منزله انتخاب عمل j از اتوماتای i می باشد (α_j^i) که منجر به فعال شدن اتوماتای یادگیر j می شود. این عمل از اتوماتای یادگیر توسط محیط پاداش یا جریمه داده می شود.

در صورت وجود پیوند در گراف وب سایت، عمل متناظر با آن در ماتریس گذار فعال و در غیر اینصورت غیرفعال می باشد. در ابتدا احتمال اعمال بطور یکنواخت است. سپس این احتمالات با استفاده از پیمایش هر کاربر و طبق یک الگوریتم یادگیری به روز رسانی می شوند.

اگر در مسیر حرکت کاربر از صفحه i به j دور وجود داشته باشد و عمل اتوماتای متناظر با آن فعال باشد، به این حرکت جریمه داده می شود. در این حالت از صفحه i به j لینک وجود دارد. اعمالی که قسمتی از یک دور باشند نشان دهنده حرکت اشتباه کاربر، سرگردانی وی در وب و یا عدم رضایت او از اطلاعات صفحات مشاهده شده می باشند و مجازات می شوند. هر چه طول این دور بیشتر باشد، میزان جریمه بیشتر خواهد بود. ضریب جریمه طبق رابطه (۱۰) محاسبه می گردد.

$$b_{i,j}^{cycle} = (\text{distance between page } i \text{ and page } j \text{ in the cycle}).b_0^{cycle} \qquad (۱۰)$$

بطوری که b_0^{cycle} ضریب ثابت است.

بالعکس اگر در مسیر حرکت کاربر از صفحه i به j دور وجود نداشته باشد ابتدا عمل متناظر با آن در ماتریس گذار فعال و سپس احتمالات به روز رسانی خواهد شد. به این حرکت کاربر پاداش تعلق می گیرد.

ایده در نظر گرفته شده این است که پس از پارتیشن بندی گراف وب سایت دو صفحه ای که در یک پارتیشن قرار دارند نسبت به دو صفحه ای که در پارتیشن های متفاوتی قرار دارند به یکدیگر شباهت بیشتری دارند. از اینرو اگر صفحه i و j در پارتیشن یکسانی باشند به عمل α_j^i طبق رابطه (۱۱) پاداش و در غیر اینصورت با یک ضریب ثابت b. جریمه داده می شود.

$$a = \frac{1}{\text{distance between page } i \text{ and page } j \text{ in the session}}\omega + a_0 \qquad (۱۱)$$

که در آن ω پارامتر ثابت، i و j دو صفحه که در یک مسیر توسط کاربر مشاهده شده اند (عمل j از اتوماتای i) و λ ثابتی است که اگر صفحه i و j غیرمتصل باشند برابر با صفر و در غیر این صورت برابر با یک مقدار ثابت a. است.

در ابتدای الگوریتم، اقدام اتوماتاهایی که سند متناظر آن در گراف به هم متصل شده اند فعال و مابقی غیرفعال می باشند. با حرکت یک کاربر از سند i به سند j، اقدام متناظر با آن سند (اقدام j) در اتوماتای یادگیر i فعال می‌شود. در این حالت اگر هر دو سند در یک پارتیشن قرار داشته باشند و در مسیر دوری نباشد ، اتوماتای یادگیر i به اقدام j خود پاداش می‌دهد در غیر این صورت جریمه می شود و این عمل با استفاده از رابطه تراگذری تا انتهای مسیر ادامه پیدا می کند. سپس اتوماتای یادگیر j در اتوماتای یادگیر توزیع‌شده فعال می‌شود و مراحل فوق تا پایان حرکت کاربر در مجموعه صفحات ادامه می‌یابد. در صورت وجود دور، اگر اقدام متناظر با آن سند(اقدام j) در اتوماتای یادگیر i فعال باشد به عمل متناظر در سند جریمه داده می شود(در جریمه نیز رابطه تراگذری در نظر گرفته می شود). در هر زمان، شباهت دو سند i و j برابر با احتمال انتخاب اقدام j در اتوماتای i (α_j^i) است. در صورتیکه اقدام مورد نظر غیرفعال باشد، شباهت دو سند صفر در نظر گرفته می‌شود. شبه کد تعیین شباهت بین صفحات وب در شکل ۳ نشان داده شده است.

نحوه پاداش و جریمه در این الگوریتم با در نظر گرفتن رابطه تراگذری می باشد. مثلا اگر کاربر در یک مسیر به ترتیب صفحات i۳, i۲, i۱ را مشاهده کرده باشد، با فرض اینکه وجود شباهتی بین محتوای این صفحات موجب این انتخاب کاربر شده است، به اعمال هر یک از اتوماتاهای متناظر با این صفحات پاداش داده می شود.

- عمل i۲ از اتوماتای i۱، عمل i۳ از اتوماتای i۲
- عمل i۳ از اتوماتای i۲

پاداشی که به هر یک از اعمال i۳, i۲ از اتوماتای یادگیر i۱ داده میشود، طبق یک ضریب کاهشی، به ترتیب کاهش داده میشود. چرا که فاصله صفحات متناظر آنها در مسیر کاربر به ترتیب افزایش مییابد. همچنین در صورتی که صفحه i۲, i۱ در گراف وب به یکدیگر متصل و صفحه i۲ و i۳ غیرمتصل باشند، پاداشی که عمل i۲ در اتوماتای i۱ می گیرد، بیشتر از پاداشی است که عمل i۳ در اتوماتای i۲ می گیرد.

پاداش دادن به اعمال انتخاب شده توسط اتوماتای یادگیر به سه عامل بستگی دارد:

۱. مسیرهای طی شده توسط کاربران
۲. فاصله صفحات در مسیرهای طی شده
۳. پیوند بین صفحات در گراف وب

جریمه دادن اعمال انتخاب شده توسط اتوماتای یادگیر که در این مقاله مطرح شده، به دو عامل بستگی دارد:

۱. وجود دور در مسیر حرکت کاربر

۲. خارج شدن از زیر گراف مربوط به اتوماتا

Procedure DLA-GP
n: number of pages of the web site
HG: hyperlink graph of the website
K: number of partitions of the hyperlink graph of the web site
userLog: list of users navigation graph
s: is an n×n matrix which will hold the similarity of every two pages *i* and *j*.
α_j^i : is the action *j* of learning automaton *i*.

begin

Partition the *HG* into *K* partitions

DLA← Create a distributed learning automata with *n* learning automata with changing number of actions; each has *n*-1 actions which all are disabled by default.

for each hyperlink from $page_i$ to $page_j$ in *HG* **do**
enable action α_j^i of the DLA
end-for

for each $user_u$ in userLog **do**
if navigation graph of $user_u$ contains a cycle **then**
for each $cycle_c$ found in the navigation graph of $user_u$ **do**
for each ($page_i$,$page_j$) in $cycle_c$ which $page_i$ is visited before $page_j$ **do**
if action α_j^i is enabled **do**
penalize action α_j^i according to eq. 10
end-if
end-for
end-for
else
for each ($page_i$,$page_j$) in navigation graph of $user_u$ which $page_i$ is visited before $page_j$ **do**
if α_j^i is disabled **do**
enable action α_j^i
end-if
if $page_i$ and $page_j$ are in the same partition **then**
reward action α_j^i according to eq. 11
else
penalize action α_j^i according to eq. 10
end-if
end-for
end-if
end-for
for each ($page_i$,$page_j$) **do**

$$s(i,j)=\begin{cases}0 & \text{if action } \alpha_j^i \text{ is disabled}\\ \dfrac{\hat{p}_j^i}{\sum_{j=1}^{n}\hat{p}_j^i} & \text{if action } \alpha_j^i \text{ is enabled}\end{cases}$$

end-for
end

شکل۳. شبه کد الگوریتم تعیین شباهت مبتنی بر استفاده

۲.۴. رتبه بندی صفحات مبتنی بر استفاده

در [۶] فرصتی و میبدی روش جدیدی با استفاده از اتوماتای یادگیر توزیع شده برای محاسبه PageRank شخصی سازی مبتنی بر استفاده، ارائه کرده اند. در این روش، احتمال انتقال از صفحه i به j برابر با احتمال انتخاب اقدام j در اتوماتای i (رابطه (۱۲)) و بردار شخصی سازی $\vec{v}$ مطابق با رابطه (۹) است.

$$p_{ij}^{DLA-UPR}=s(i,j) \qquad (۱۲)$$

در الگوریتم پیشنهادی، ماتریس انتقال p و بردار شخصی سازی $\vec{v}$ در الگوریتم اصلی PageRank بجای استفاده از ساختار پیوندها، مبتنی بر داده کاوی استفاده از وب محلسبه می شود. به همین دلیل اتوماتای یادگیر توزیع شده (DLA[۱۴]) ماتریس احتمال انتقال p از طریق رفتار کاربران موجود در لاگ فایل های [۱۵] سایت، یاد بگیرند. تعداد دفعات مشاهده هر صفحه توسط کاربران به عنوان معیاری برای مقداردهی بردار

شخصی سازی $\vec{v}$ استفاده می‌گردد. این مقدار دهی بر اساس تعداد مشاهدات هر صفحه است و به درستی یک بردار احتمال است زیرا که مجموع همه عناصر آن برابر ۱ می‌باشد. بدیهی است که هر چه صفحه‌ای دفعات زیادی مشاهده شده باشد، نسبت به صفحات دیگر مهم‌تر می‌باشد. با داشتن ماتریس احتمال انتقال p و بردار شخصی سازی $\vec{v}$ که از طریق اطلاعات پیمایش های کاربران قبلی بدست می آیند, الگوریتم PageRank طبق رابطه (۱۳) برای محاسبه رتبه هر صفحه استفاده می شود.

$$DLA-UPR(p_i)=d.\sum_{p_j\in in(p_i)}\frac{DLA-UPR(p_j)}{|out(p_j)|}+(1-d)\frac{1}{N}$$
$$DLA-UPR(p_i)=d.\sum_{p_j\in in(p_i)}s(j,i)+(1-d)\frac{1}{N} \qquad (۱۳)$$

بطوری که $out(p_j)$ به مجموعه صفحات لینک شده از صفحه j اشاره دارد و $in(p_j)$ مجموعه صفحاتی است که به صفحه j لینک دارند. N تعداد صفحات وب, $UPR(p_j)$ رتبه صفحه p_j و d ضریب تعدیل می باشد که عددی بین صفر و یک است و ۰.۸۵ مقداردهی شده است.

۳.۴. پیشنهاد صفحات

هدف از شخصی سازی بر اساس اطلاعات پیمایش کاربران محاسبه یک مجموعه پیشنهادی، rs ، برای نشست کاربر جاری می‌باشد[۲۴] [۲۵]. که بیشترین تطابق را با علایق کاربر داشته باشد. این جز تنها جز برخط سیستم بوده و باید از کارایی و دقت بالایی برخوردار باشد. فرض کنیم که کاربری که در حال گردش در سایت است و مسیر $p_1\to p_2\to p_3\to\ldots\to p_k$ را پیموده است. تعداد آخرین صفحاتی را که توسط کاربر مشاهده شده و برای پیشنهاد صفحات جدید مورد استفاده قرارمی‌گیرد را پنجره پیشنهاد می‌نامیم و اندازه آن را با rw نشان می‌دهیم که حداکثر برابر با تمام صفحات مشاهده شده و حداقل برابر با آخرین صفحه مشاهده شده می‌باشد. برای پیشنهاد صفحه P_{k+1} به کاربر از خاصیت مارکوف گراف استفاده می‌کنیم. طبق قاعده زنجیر مارکوف احتمال انتخاب مسیر در گراف دارای ویژگی مارکوف، از رابطه (۱۴) به دست می‌آید :

$$\Pr(p_1\to p_2\to p_3\cdots\to p_k)=\Pr(p_1)\times\prod_{i=2}^{k}\Pr(p_i\mid p_{i-m}\cdots p_{i-1}) \qquad (۱۴)$$

به عنوان مثال، احتمال مسیر $p_1\to p_2\to p_3$ برابراست با :

$$\Pr(p_1\to p_2\to p_3)=\Pr(p_1)\Pr(p_2\mid p_1)\Pr(p_3\mid p_2)=\Pr(p_1)\frac{\Pr(p_1\to p_2)\Pr(p_2\to p_3)}{\Pr(p_1)\Pr(p_2)} \qquad (۱۵)$$

که در آن $\Pr(\bullet\to\bullet)$ برابر با احتمال گذار بین دو صفحه است و $\Pr(\bullet)$ احتمال حالت پایدار صفحه متناظر می‌باشد که در دو بخش قبل به ترتیب در ماتریس p و بردار $\bar{x}$ محاسبه شدند. برای پیشنهاد صفحه به کاربر، به ازای صفحات مختلف P_{k+1} که در مسیر $p_1\to p_2\to p_3\to\ldots\to p_k$ ملافات نشده اند، احتمال مسیر $p_1\to p_2\to p_3\to\ldots\to p_k\to p_{k+1}$ را محاسبه می‌نماییم. احتمال هر مسیر امتیاز صفحه P_{k+1} را برای پیشنهاد به کاربر نشان می‌دهد. با مرتب‌کردن صفحات بر اساس امتیاز آنها، صفحاتی با بیشترین امتیاز به کاربر پیشنهاد می‌شود. برای هر یک از تعداد صفحات پیشنهادی دقیقاً فقط d صفحه (برابر با تعداد صفحات پیشنهادی) که بیشترین امتیاز را دارند, به کاربر پیشنهاد می شود. در شکل ۴ شبه کد الگوریتم پیشنهاد صفحات ارائه شده است.

Procedure *ProposedRecommendationAlgorithm*
n: number of pages in the web site
UPR(page$_i$): Usage-based Page Rank of page *i*
s(*i*,*j*) : similarity between page *i* and *j* determined in previous section.
w: windows size, i.e. *w* past number pages user has visited.
history$_u$(*t*): Pages user *u* has visited until time step *t*.
page^u(*t*): t^{th} page user *u* has visited in the web site.
D: Number of recommendation pages
begin
  p(page$_i$) = UPR(page$_i$) *i*=۱, ۲, …, N
  p(page$_i$ →page$_j$)= *s*(*i*,*j*) *i*,*j*=۱, ۲, …, N
  for each page$_i$ ∉ history$_u$ **do**
    Calculate recommendation values for page$_i$:
    $r_i^u(t)$ = Pr(page^u(*t*-*w*)→ page^u(*t*-*w*+۱)→…→ page^u(*t*)→ page$_i$)
  end-for
  Depending on the request,
    either set the recommendation list for user *u* to top *D* pages with highest recommendation value in $r^u(t)$.
end

شکل۴. شبه کد الگوریتم پیشنهاد صفحات

۵. ارزیابی الگوریتم پیشنهادی

در این قسمت ابتدا مدل بکار رفته برای تولید داده استفاده از وب و معیار ارزیابی تشریح می گردد. سپس نتایج آزمایشات الگوریتم پیشنهادی با الگوریتم های قبلی مقایسه می گردد.

۱.۵. مدل شبیه سازی

دو روش عمده برای ارزیابی الگوریتم‌هایی که از اطلاعات پیمایش کاربران استفاده می‌کنند وجود دارد. روش اول، استفاده از صفحات وب واقعی و داده‌های واقعی کاربران وب موجود در فایل‌های ثبت رخداد سایت‌ها می‌باشد. روش دوم مدل ارائه شده در[۲۶] می‌باشد. در این روش Liu و همکارانش نظم موجود در رفتارهای کاربران در محیط وب را با استفاده از یک مدل مبتنی بر عامل ، مشخص و اعتبار مدل خود را با استفاده از چندین سایت وب بزرگ مانند مایکروسافت، تایید کرده‌اند. در این مقاله ما از داده های استاندارد سایت CTI DePaul استفاده می‌کنیم. این مجموعه داده اطلاعات نشست کاربران را به مدت ۲ هفته در سایت CTI DePaul در سال ۲۰۰۲ شامل می‌شود[۲۷]. این اطلاعات پیش‌پردازش شده و نشست‌های با اندازه ۱ و غیر استاندارد از آن حذف شده‌اند و در نهایت اطلاعات ۱۳۷۴۵ کاربر که از ۶۸۳ صفحه دیدن کرده‌اند در فایل‌های جداگانه قرار داده شده است.

برای انتخاب دو مجموعه آموزشی و تست از معیار *K-Fold* استفاده شده است [۳۰]. در این روش نمونه اصلی بطور تصادفی به k زیر نمونه تقسیم می گردد. برای هر k زیر نمونه، تنها یکی از زیر نمونه های بدست آمده به عنوان داده تست و $k-1$ زیر نمونه باقیمانده به عنوان داده آموزشی استفاده می گردد. این پروسه به تعداد k بار تکرار می شود و به این ترتیب دقیقاً همه k زیر نمونه موجود، به عنوان داده تست استفاده می گردد. در نهایت برای تولید یک تخمین واحد، بین k نتیجه میانگین گرفته می شود. مزیت روش *K-Fold* این است که با تکرار زیر نمونه های تصادفی تمام مجموعه هم به عنوان داده آموزشی و هم داده تست استفاده می شوند. این روش در شکل ۵ نشان داده شده است.

شکل ۵. روش K-Fold

در اتوماتای یادگیر توزیع شده، احتمال انتخاب عمل j در اتوماتای i ، میزان ارتباط دو صفحه i ام و j ام را نشان می دهد. ساختار ارتباطی پیشنهادی با یک ماتریس $n \times n$ به نام P بازنمایی می شود. در صورت فعال بودن عمل j در اتوماتای i، درایه P_{ij} این ماتریس برابر با احتمال عمل j در اتوماتای i و در غیر اینصورت برابر با صفر قرار داده می‌شود. از آنجاییکه فرایند آموزش ترتیب توالی دسترسی ها را در نظر می گیرد نتایج فرایند یادگیری با یک ماتریس نامتقارن ($p_{ij} \neq p_{ji}$) بازنمایی می شود. ماتریس نامتقارن تولید شده مبتنی بر آتوماتای یادگیر را ماتریس انتقال صفحات می نامیم. از حاصلضرب ماتریس نامتقارن P با ماتریس ترانهاده[۱۶] آن، ماتریس متقارن جدیدی به نام ماتریس شباهت S طبق رابطه (۱۶) حاصل می شود.

$$S = P \cdot P^T$$
$$s_{ij} = \sum_k a_{ik} a_{kj} \qquad (۱۶)$$

درایه s_{ij} در این ماتریس، درجه شباهت دو صفحه i ام و j ام را نشان می دهد.

۲.۵. معیار ارزیابی

برای ارزیابی، دو معیار "پوشش[۱۷]" و "دقت[۱۸]" معرفی می شوند. این دو معیار، بسیار شبیه به معیارهای متداول در بازیابی اطلاعات یعنی "فراخوانی[۱۹]" و "دقت" بازیابی اسناد هستند.

دقت پیشنهادها برابر با "نسبت پیشنهادهای درست به کل پیشنهاد ها" است. منظور از پیشنهاد درست، پیشنهادی است که با توجه به بخش دیده شده (پیشوند) یک جلسه کاربر تولید شده و در ادامه جلسه کاربر (پسوند) رخ دهد. اگر تعداد U جلسه کاربر را در نظر بگیریم، برای هر جلسه مثل u به ترتیب صفحات بازدید شده را، یک به، یک، به مجموعه صفحات بازدید شده اضافه می کنیم. سپس، با دیدن هر صفحه p پیشنهادهایی تولید می کنیم. این مجموعه پیشنهاد را $R(p)$ (مجموعه صفحات پیشنهاد شده پس از بازدید کاربر از صفحه p) می نامیم. سپس $R(p)$با قسمت باقیمانده از جلسه کاربر، که آن را $Tail(p)$ یا به اختصار $T(p)$ می نامیم، مقایسه می شود .دقت پیشنهادات برابر با درصد اشتراک $T(p)$و $R(p)$ خواهد بود و طبق رابطه (۱۷) محاسبه می گردد.

$$\Pr ecision = \frac{T(p) \cap R(p)}{R(p)} \quad (۱۷)$$

پوشش پیشنهادها، قدرت سیستم در پیش بینی تمام صفحاتی که ممکن است مورد نظر کاربران باشد را اندازه گیری می کند. این عدد برابر با "نسبت صفحات درست صفحات پیش بینی شده در ادامه جلسه یا $T(p)$ به کل صفحات باقیمانده (تعدادصفحات $T(p)$) در جلسه در هر قدم است" و طبق رابطه (۱۸) محاسبه می گردد.

$$Coverage = \frac{T(p) \cap R(p)}{T(p)} \quad (۱۸)$$

هر چه مقدار دقت و پوشش بالاتر باشد، کارایی الگوریتم, مطلوب تر است. بررسی این وضعیت با استفاده از معیار F۱ ساده تر می باشد (رابطه (۱۹)).

$$F1 = \frac{2 \times Coverage \times \Pr ecision}{Coverage + \Pr ecision} \quad (۱۹)$$

همانطور که رابطه (۱۹) نشان می دهد, هرچه مقدار دقت و پوشش بالاتر باشد، مقدار F۱ نیز افزایش می یابد.

۳.۵. تنظیمات

در آزمایشات، برای پارتیشن بندی گراف وب سایت از ابزار Metis، [۲۹]، استفاده شده است که بر اساس الگوریتم های پارتیشن بندی چند سطحی [۲۷] کار می کند. تعداد پارتیشن ها k=۲۵ در نظر گرفته شده است. همچنین در الگوریتم پیشنهادی، a . و b_0^{cycle}، ۰.۰۰۲ و ω و. b به ترتیب ۰.۰۲و ۰.۳ مقداردهی می گردد.

همچنین برای آماده سازی داده تست و آموزشی مقدار پارامتر k در روش $K\text{-}Fold$، ۲۰ در نظر گرفته شده است.

۴.۵. پارامتر های موثر در ارزیابی

اندازه طول پنجره پیشنهاد (rw) و تعداد صفحات پیشنهادی که دو معیار "دقت" و "پوشش" برحسب آنها اندازه گیری می شوند، پارامترهای تاثیرگذار در کارایی الگوریتم هستند. در واقع براساس طول پنجره که مسیر پیمایش کاربر می باشد مسیر بعدی کاربر را پیش بینی می گردد.

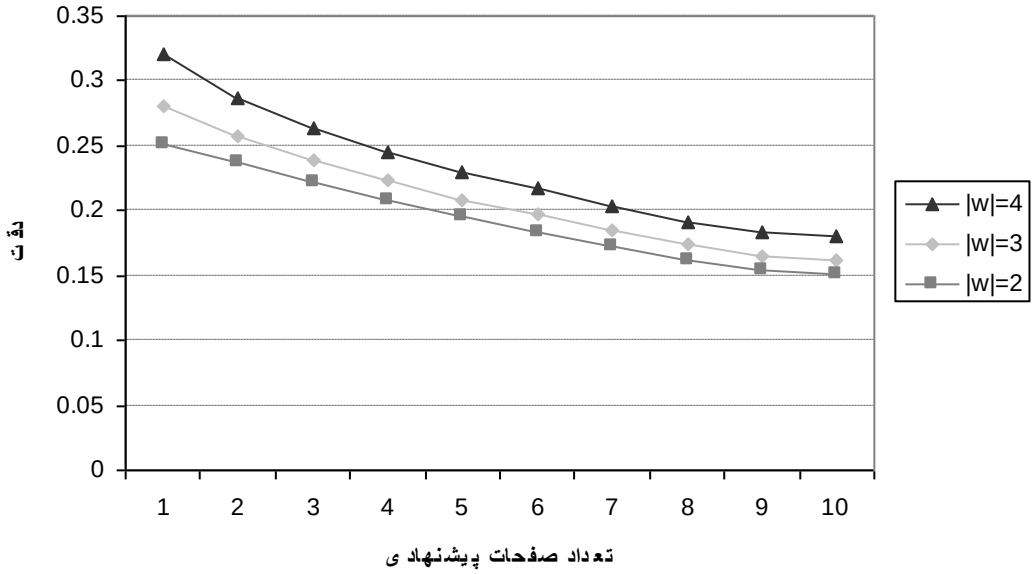
ابتدا با استفاده از مجموعه یادگیری الگوریتم اجرا شده و سپس بر اساس مقدار طول پنجره، rw صفحه متوالی را انتخاب کرده و به الگوریتم داده می شود. معیار ارزیابی رابطه معرفی شده در[۱۰] است. فرض کنیم مجموعه $rp = \{x_{rw+1}, x_{rw+2}, ..., x_{rw+|rs|}\}$ صفحات مشاهده شده توسط کاربر در ادامه نشست واقعی باشد. درجه شباهت مجموعه پیشنهادی و مجموعه صفحات واقعی از رابطه (۲۰) به دست می‌آید.

$$Sim(rs \cap rp) = \frac{rs \cap rp}{rp} \quad (۲۰)$$

۵.۵. نتایج آزمایشات

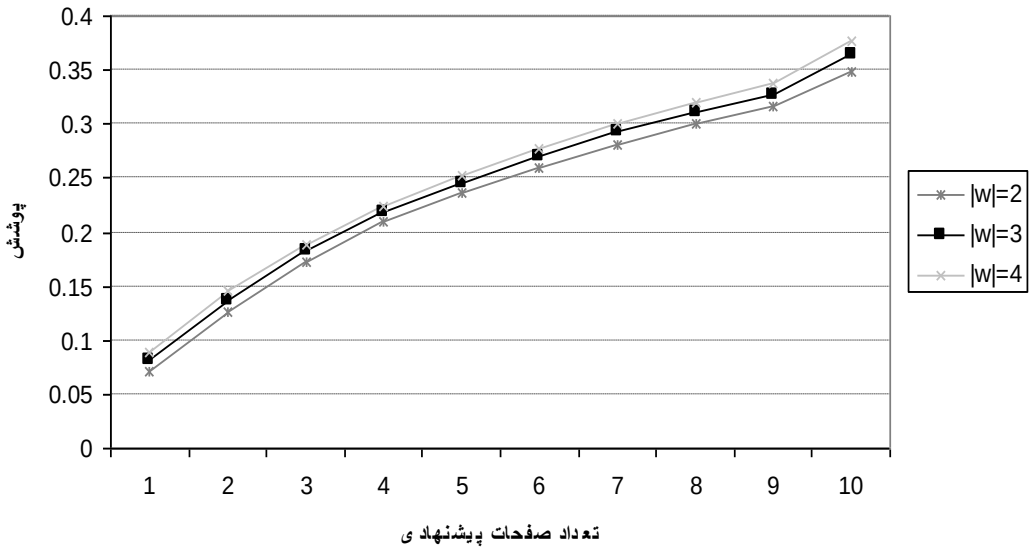
الگوریتم پیشنهادی با روش های مبتنی بر اتوماتای یادگیر توزیع شده [۱][۳][۴][۶]که از یک روش آماری ساده نیز استفاده می کنند، مقایسه می گردد. در این روش آماری شباهت دو سند i و j، بر اساس نسبت تعداد دفعاتی که کاربران از سند i به سند j حرکت کرده‌اند به تعداد دفعاتی که کاربران از سند i به هر سند دیگری مانند k حرکت نموده‌اند، طبق رابطه (۲۱) محاسبه می‌شود.

$$simpleSimilarity(i, j) = \frac{visited(i, j)}{\sum_{k=1}^{n} visited(i, k)} \quad (۲۱)$$



شکل ۶. مقایسه دقت الگوریتم پیشنهادی با اندازه های متفاوت پنجره نسبت به تعداد صفحات پیشنهادی

شکل ۶ نتایج دقت الگوریتم را نسبت به تعداد صفحات پیشنهادی مختلف برای ۳ حالت اندازه پنجره پیشنهاد نشان می‌دهد. در شکل مشخص است که دقت با تعداد صفحات پیشنهادی نسبت عکس دارد به طوری که با افزایش تعداد صفحات پیشنهادی، دقت کاهش می یابد.

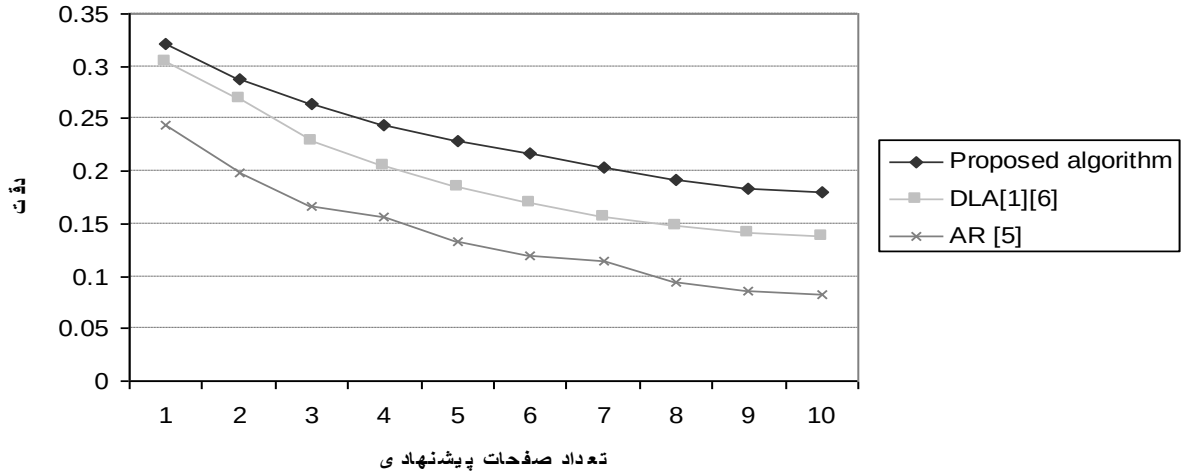


شکل ۷. مقایسه پوشش الگوریتم پیشنهادی با اندازه های متفاوت پنجره نسبت به تعداد صفحات پیشنهادی

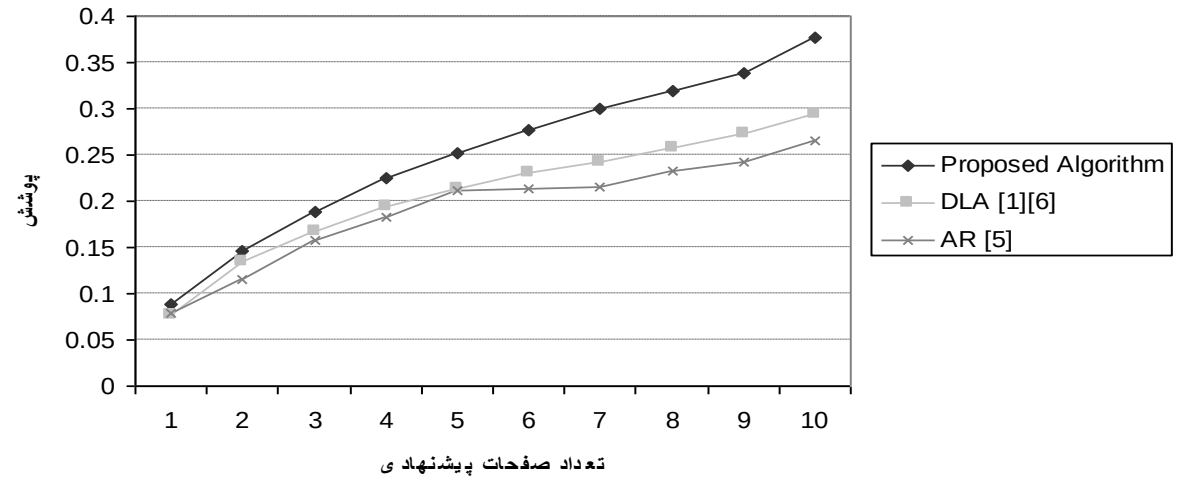
شکل ۷ نتایج پوشش الگوریتم را نسبت به تعداد صفحات پیشنهادی مختلف برای ۳ حالت اندازه پنجره پیشنهاد نشان می‌دهد. پوشش با تعداد صفحات پیشنهادی نسبت مستقیم دارد به طوری که با افزایش تعداد صفحات پیشنهادی، پوشش افزایش می یابد.

همانطور که در شکل های ۶ و ۷ مشاهده می شود, برای طول پنجره ۴ بالاترین دقت و پوشش را داریم.

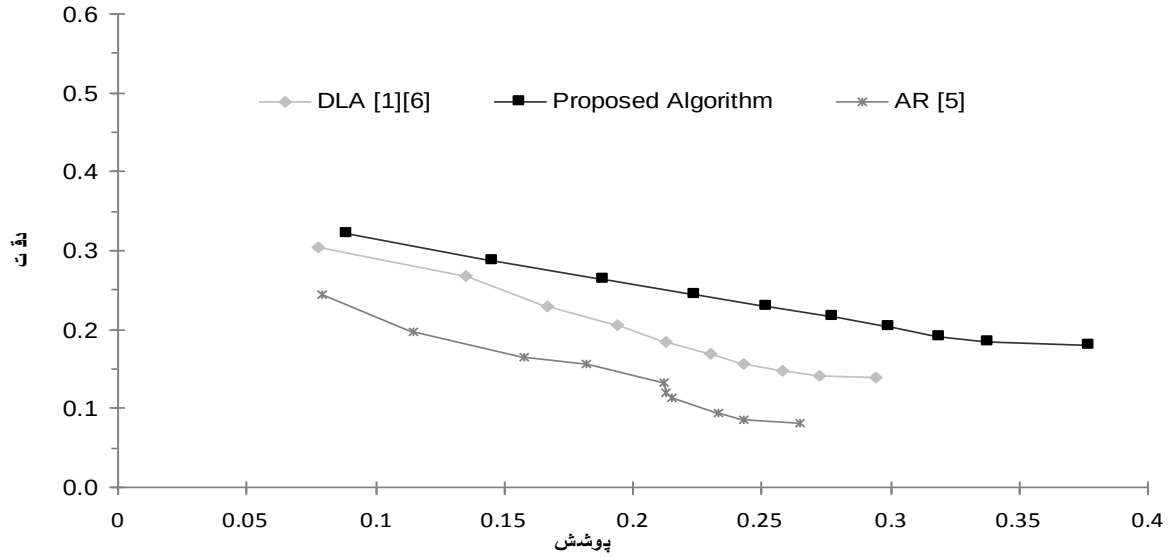
شکل ۸ و ۹ به ترتیب دقت و پوشش الگوریتم پیشنهادی با الگوریتم های مبتنی بر اتوماتای یادگیر توزیع شده [۱][۶] و قوانین انجمنی[۵] نسبت به تعداد صفحات پیشنهادی مختلف و با طول پنجره ۴, مقایسه شده است. همانطور که مشاهده می شود, الگوریتم پیشنهادی از دقت و پوشش بالاتری برخوردار است.



شکل ۸. مقایسه دقت الگوریتم پیشنهادی با الگوریتم های DLA [۱][۶] و AR [۵]



شکل ۹. مقایسه پوشش الگوریتم پیشنهادی با الگوریتم های DLA [۱][۶] و AR [۵]

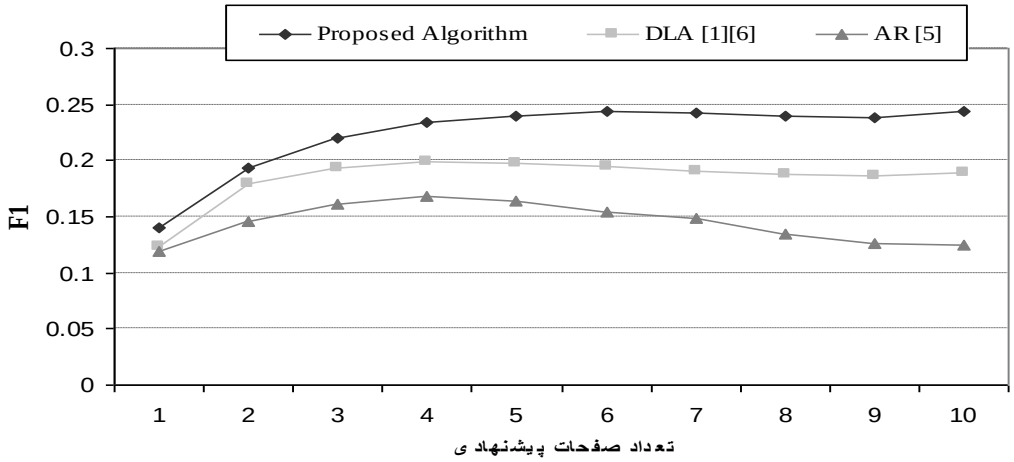


شکل ۱۰. بررسی الگوریتم پیشنهادی با الکوریتم های DLA [۱][۶] و AR [۵] براساس نسبت دقت به پوشش با طول پنجره ۴

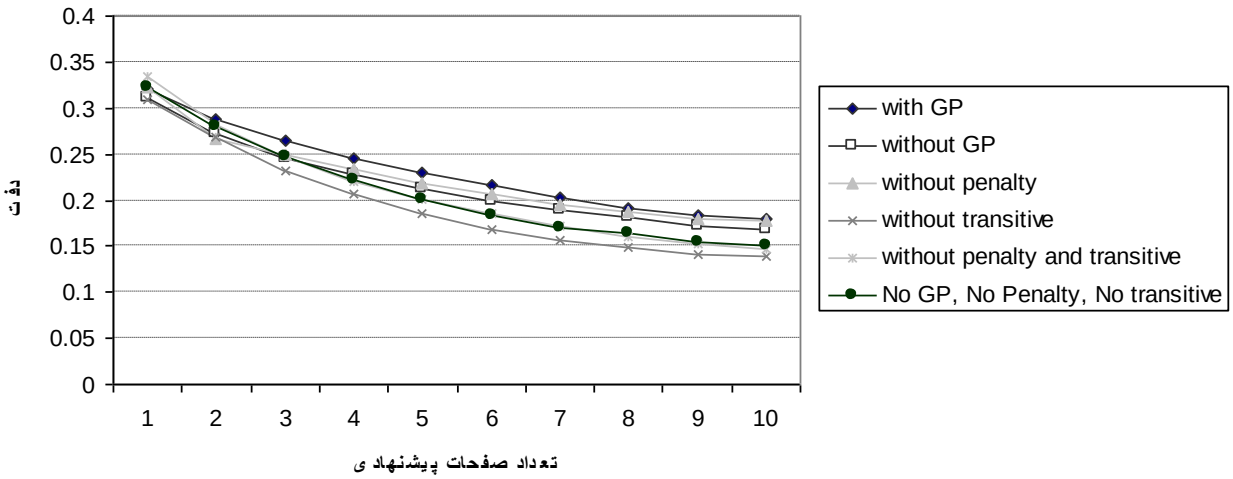
در شکل ۱۰ الگوریتم پیشنهادی با الگوریتم های DLA [۱][۶] و AR [۵] مقایسه شده است که براساس مقدار دقت به پوشش و با طول پنجره ۴ به دست آمده است. همانطور که مشاهده می شود الگوریتم پیشنهادی نسبت به دو الگوریتم فبلی نتایج بهتری را ارائه می دهد.

شکل ۱۱ مقایسه الگوریتم پیشنهادی با الگوریتم های DLA [۱][۶] و AR [۵] براساس معیار F۱ و با طول پنجره ۴ است و نشان می دهد که الگوریتم پیشنهادی نسبت به دو الگوریتم فبلی نتایج از دقت و پوشش بالاتری برخوردار است.

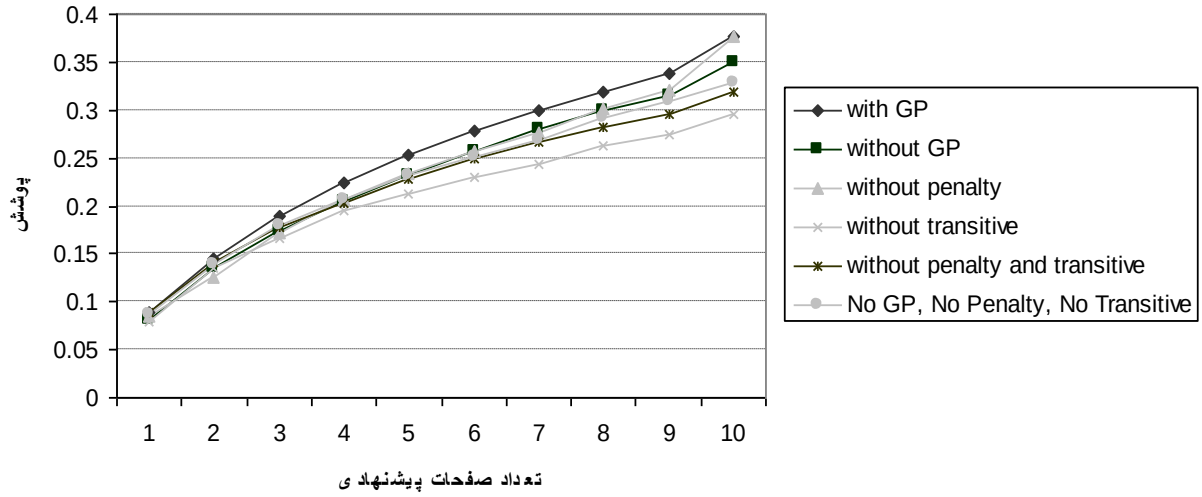
شکل ۱۲ و ۱۳ به ترتیب دقت و پوشش الگوریتم پیشنهادی را در شرایط مختلف, بدون پارتیشن بندی, رابطه تعدی و جریمه و با طول پنجره ۴ نشان می دهد. همانطور که نتایج نشان می دهد.



شکل ۱۱. مقایسه الگوریتم پیشنهادی با الگوریتم های DLA [۱][۶] و AR [۵] براساس معیار F۱ و با طول پنجره ۴



شکل ۱۲. مقایسه دقت الگوریتم پیشنهادی در شرایط مختلف و با طول پنجره ۴



شکل ۱۳. مقایسه پوشش الگوریتم پیشنهادی در شرایط مختلف و با طول پنجره ۴

۶. نتیجه گیری

در این مقاله سعی کردیم تا روش [۱][۶] که مبتنی بر اتوماتای یادگیر توزیع شده می باشد، بهبود دهیم. همچنین سعی کردیم الگوریتم پیشنهادی را برای صفحات بیشتر تعمیم داده و نتیجه آنرا بهبود دهیم. نتایج شبیه سازیها نشان داد که روش پیشنهادی در مقایسه با روش های گزارش شده مبتنی بر اتوماتای توزیع شده و قوانین انجمنی در تشخیص شباهت صفحات و پیشنهاد به کاربر از کارایی بالاتری برخوردا ر است. بصورتیکه دقت و پوشش بدست آمده با اندازه پنجره ۴، در الگوریتم پیشنهادی بیشتر از این مقدار در الگوریتم معرفی شده در [۱] [۶] و [۵] می باشد.

7. مراجع

[۱]. رعنا فرصتی، محمدرضا میبدی، مهرداد مهدی، **"شخصی سازی وب با استفاده از اتوماتای یادگیر توزیع شده "**، سومین کنفرانس فناوری اطلاعات و دانش، مشهد، ایران، ۱۳۸۶.

[۲]. سعید ساعتی، محمدرضا میبدی، **"رتبه بندی اسناد با استفاده از اتوماتای یادگیر توزیع شده"**، یازدهمین کنفرانس بین المللی انجمن کامپیوتر ایران، تهران، ایران، ۱۳۸۴.

[3]. علی براردران هاشمی، محمد رضا میبدی، **"داده کاوی استفاده از وب با استفاده از اتوماتای یادگیر توزیع شده "**، دوازدهمین کنفرانس بین المللی انجمن کامپیوتر ایران، تهران، ایران، ۱۳۸۵ .

[***4***]. B. Mobasher, R. Cooley, J. Srivastava, "*Automatic Personalization Based on Web Usage Mining*", Communications of the ACM, Vol. ***43***, No. ***8***, ***2000***.

[***5***]. B. Mobasher, H. Dai, T. Luo, M. Nakagawa, "*Effective personalization based on association rule discovery from web usage data*", Proceedings of the ***3***rd ACM Workshop on Web Information and Data Management, ***2001***.

[***6***]. R. Forsati, M. R. Meybodi, "*Effective page recommendation algorithms based on distributed learning automata and weighted association rules*". Expert Systems with Applications: An International Journal, ***37***, ***2*** (***2010***), ***1316-1330*** .

[***7***]. B. Mobasher, H. Dai, Y. Sun, J. Zhu, "*Integrating Web Usage and Content Mining for More Effective Personalization*", Proceeding of the EC-WEB Conference, ***2003***.

[***8***]. B. Mobasher, H. Dai, T. Luo, M. Nakagawa, "*Using sequential and non-sequential patterns for predictive web usage mining tasks*", Proceedings of the IEEE International Conference on Data Mining, Maebashi City, Japan, ***2002***.

[***9***]. M. Richardson, P. Domingos, "*The Intelligent Surfer: Probabilistic Combination of Link and Content Informationin PageRank*", in Neural Information Processing System, ***2002***.

[***10***]. T. Haveliwala, "*Topic-Sensitive PageRank*", in Proceedings of the ***11***thInternational Conference on World Wide Web, New York: ACM Press, pp. ***517–526***, ***2002***.

[***11***]. M. S. Aktas, M. A. Nacar, F. Menczer, "*Personalizing PageRank Based on Domain Profiles*", Proceeding of WEBKDD Workshop, Seattle, ***2004***.

[***12***]. J. Wang, Z. Chen, L. Tao, W. Ma, L. Wenyin, "*Ranking User's Relevance to a Topic through Link Analysis on Web Logs*", Proceeding of the WIDM '***02***, ***2002***.

[***13***]. J. Borges, M. Levene, "*Data Mining of User Navigation Patterns*", in Revised Papers from the International Workshop on Web Usage Analysis and User Profiling, pp. ***92-111***, ***2000***.

[***14***]. M. Nakagawa, B. Mobasher, "*A Hybrid Web Personalization Model Based on Site Connectivity*", in Proceeding of the ***5***th WEBKDD Workshop, Washington DC, ***2003***.

[***15***]. K. S. Narendra, M. A. L. Thathachar, "*Learning Automata: An Introduction*", Prentice Hall, ***1989***.

[***16***]. M. A. L. Thathachar, R. Harita Bhaskar, "*Learning Automata with Changing Number of Actions*", IEEE Transactions on Systems Man and Cybernetics, vol. ***17***, no. ***6***, pp. ***1095-1100***, ***1987***.

[17]. K. S. Narendra, M. A. L. Thathachar, "*Learning automata: An introduction*". Prentice Hall, **1989**.
[18]. H. Beigy, M. R. Meybodi, "*Utilizing Distributed Learning Automata to Solve Stochastic Shortest Path Problem*". International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, **14**, **5** (**2006**), **591-617** .L. Page, S. Brin, R. Motwani, T. Wingord, "The PageRank Citation Ranking: Bringing Order to the Web", Stanford University, **1998**.
[19]. S. Brin, L. Page, "*The anatomy of a large-scale hypertextual Web search engine*". Computer Networks and ISDN Systems, **30**, **1-7** (**1998**), **107-117** .
[20]. L. Page, S. Brin, R. Motwani, T. Winograd, "*The PageRank citation ranking: bringing order to the web*". Computer Science Department, Stanford University, Palo Alto, California, USA, **1999**.
[21]. S. Kamvar, T. Haveliwala, G. Golub, "*Adaptive methods for the computation of PageRank*". Linear Algebra and its Applications, **386**(**2004**), **51-65**.
[22]. M. Eirinaki, M. Vazirgiannis, "*Web site personalization based on link analysis and navigational patterns*". ACM Transactions on Internet Technology, **7**, **4** (**2007**), **21**.
[23]. H. Dai, B. Mobasher, "*Integrating Semantic Knowledge with Web Usage mining for Personalization*", **2004**.
[24]. B. Mobasher, H. Dai, T. Luo, M. Nakagawa, "*Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization*", Data Mining and Knowledge Discovery, pp. **61**-**82**, **2002**.
[25]. H. Liue, V. Keselj, "*Combined mining of Web server logs and web contents for classifying user navigation patterns and redicting users' future requests*", Data & Knowledge Engineering, **2007**.
[26]. J. Liu, S. Zhang, J. Yang, "*Characterizing Web Usage Regularities with Information Foraging Agents*", IEEE Transactions on Knowledge and Data Engineering, pp. **566**-**584**, **2004**.
[27]. http://maya.cs.depaul.edu/~classes/ect**584**/data/cti-data.zip
[28]. G. Karypis, V. Kumar, "*Multilevel k-way partitioning scheme for irregular graphs*". Journal of Parallel and Distributed Computing, **48**(**1**):**96**–**129**, **1998**.
[29]. G. Karypis, V. Kumar, "*METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*". Minneapolis, City, **2007**.
[30]. G. J. McLachlan, K. A. Do, C. Ambroise, "*Analyzing microarray gene expression data*". Wiley, **2004**.

---

۱ Web Content Mining
۲ Web Structure Mining
۳ Web Usage Mining
۴ Hyperlink
۵ Association Rule
۶ Multilevel
۷ Coarsening Phase
۸ Initial Partitioning Phase
۹ Refinement Phase
۱۰ Stationary
۱۱ Non-Stationary
۱۲ Learning automata with changing number of actions
۱۳ Usage-based Personalization PageRank
۱۴ Distributed Learning Automata
۱۵ Log File
۱۶ Transpose
۱۷ Coverage
۱۸ Precision
۱۹ Recall