سومین کنفرانس فناوری اطلاعات و دانش
۶ تا ۸ آذرماه ۱۳۸۶
کنفرانس بین المللی فناوری اطلاعات و دانش IKT 2007
دانشگاه فردوسی مشهد

شخصی سازی صفحات وب با استفاده از اتوماتای یادگیر توزیع‌شده

رعنا فرصتی[۱]، محمد رضا میبدی [۲] و مهرداد مهدوی[۳]

(۱) دانشکده مهندسی برق، رایانه و فناوری اطلاعات، دانشگاه آزاد اسلامی، قزوین، ایران

(۲) دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

(۳) دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، ایران

forsati@mrl.ir, mmeybodi@aut.ac.ir, mahdavi@ce.sharif.edu

چکیده - استفاده همزمان از اطلاعات ساختاری و اطلاعات پیمایش کاربران یکی از چالش‌های مطرح در بهبود کارایی الگوریتم‌های شخصی‌سازی وب می‌باشد. در این مقاله الگوریتمی ترکیبی مبتنی بر اتوماتای یادگیر توزیع شده و الگوریتم *PageRank* ارائه شده است. الگوریتم پیشنهادی از اطلاعات پیمایش کاربران و پیوند بین صفحات به منظور پیشنهاد صفحات به کاربران استفاده می‌کند. الگوریتم ارائه شده را همچنین می‌توان برای تغییر پیوند صفحات به منظور هدایت بهتر کاربران استفاده کرد. بر خلاف الگوریتم های شخصی سازی موجود که تنها از اطلاعات پیمایش کاربران استفاده می کنند، الگوریتم ارائه‌شده اولین روش گزارش شده مبتنی بر اتوماتای توزیع شده می‌باشد که همزمان از اطلاعات پیمایش کاربران و پیوند بین صفحات برای پیشنهاد صفحات استفاده می‌کند. در الگوریتم ارائه شده یک اتوماتای یادگیر به هر صفحه وب تخصیص داده می‌شود. هر اتوماتای یادگیر، بر اساس اطلاعات پیمایش کاربران احتمال گذار بین صفحات را یاد می گیرد . بر اساس احتمالات گذار و اهمیت هر صفحه که با استفاده از الگوریتم *PageRank* محاسبه می‌شود، عملیات شخصی سازی انجام می شود. بر خلاف الگوریتم *PageRank* موجود که اهمیت هر صفحه بر اساس ساختار پیوندی صفحات محاسبه می‌شود الگوریتم ارائه شده همزمان از اطلاعات ساختار پیوندی صفحات و پیمایش کاربران برای محاسبه اهمیت صفحات استفاده می کند. نتایج شبیه‌سازی الگوریتم در داده‌های واقعی نشان داده است که کارایی الگوریتم پیشنهادی به ۹۰٪ می رسد در حالیکه پیچیدگی زمانی اجرای آن نیز پایین می باشد.

کلید واژه- اتوماتای یادگیر، الگوریتم*PageRank* ، داده‌کاوی استفاده از وب.

۱- مقدمه

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. در حال حاضر موتور جستجوی Google بیش از ۳ بیلیون صفحه وب را شاخص‌گذاری کرده است. بنابراین وب به مجموعه بزرگی از داده های ساخت یافته و نیمه ساخت یافته تبدیل شده است که کاربران آن از همپوشانی داده‌ها رنج می‌برند. برای حل این مشکل، شخصی کردن وب به یک پدیده محبوب به منظور سفارشی کردن محیط‌های وب تبدیل شده است. هدف از سیستم‌های شخصی‌ساز فراهم‌کردن نیازهای کاربران، بدون اینکه به طور صریح آن ها را بیان کنند یا نشان بدهند، می‌باشد[۳]. شخصی‌سازی وب مجموعه‌ای از عملیات است که تجربهء وب را برای یک کاربر خاص یا مجموعه ای از کاربران سازمان دهی می‌کند. شخصی‌سازی وب می‌تواند به عنوان فرآیندی برای سفارشی کردن محتوا و ساختار وب سایت بر حسب نیازهای شخصی و ویژهء هر کاربر باشد[۴]. سایت از طریق پررنگ کردن لینکهای موجود، ایجاد پویای

لینک‌های جدید که مورد نظر کاربر جاری باشد یا حتی ایجاد صفحات شاخص جدید شخصی می‌شود.

منابع الگوریتم‌های شخصی‌سازی را می‌توان به ۴ گروه تقسیم نمود[۵]. دسته اول داده‌های استفاده کاربران می‌باشد که شامل داده های جمع آوری شده از فایل ثبت که به صورت اتوماتیک توسط سرور وب انجام می‌شود، می‌باشد. دسته دوم داده‌های محتوا می‌باشند که شامل داده‌های موجود در صفحات می‌باشد. دسته سوم شرح حال کاربر می‌باشد که به صورت پروفایل‌هایی قابل استفاده می‌باشد و نیازمند تعامل مستقیم با کاربر می‌باشد. داده‌های ساختاری دسته بعدی هستند و شامل ابرپیوندهای موجود بین صفحات سایت می‌باشند. هر چه از این اطلاعات بیشتر در فرایند شخصی‌سازی استفاده شود کارایی الگوریتم ارائه شده افزایش خواهد یافت. اکثر تحقیقات انجام شده در زمینه شخصی‌سازی بر اساس تحلیل محتوای اسناد (داده‌کاوی محتوا[1]) و یا اطلاعات درباره رفتار کاربران (با استفاده از فایلهای ثبت وقایع[2] در سرویس‌دهنده‌های وب یا برنامه‌های در سمت کاربر) بوده است[6,7]. اگرچه از خصوصیات ساختار گراف وب (داده کاوی ساختار[3]) برای شخصی‌سازی نتایج جستجوی وب بسیار زیاد استفاده شده‌است[8-10] اما در فرایند شخصی سازی صفحات وب به آن کمتر توجه شده است. در صورتیکه علاوه بر اطلاعات بدست آمده از این دو روش، می‌توان از اطلاعات درباره ساختار گراف ارتباط اسناد (داده‌کاوی ساختار) برای پیشنهاد صفحات، تغییر ساختار سایت وب، شخصی کردن[4] سرویس‌هایی مانند وب استفاده کرد. الگوریتم ارائه شده در[11] مبتنی بر آنالیز لینک ها می باشد که صفحات وب و کاربران سایت را به صورت نود و ابرپیوند مدل می‌کند 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[12] دو متد مجزای رتبه بندی بر اساس آنالیز لینکها ارائه داده شده است.Site Rank و Popularity Rank که الگوریتم‌های رتبه‌بندی هستند که از آنها در گراف سایت استفاده می‌شود. در [12] به جای اینکه بر اساس آنها الگوریتم شخصی‌سازی ارائه شود توزیع و رتبه بندی ۲ متد ارائه شده مقایسه شده است.Mobasher از درجه اتصالات بین صفحات سایت به عنوان فاکتوری تعیین کننده برای پیشنهاد بر اساس کاوش آیتمهای تکرار شونده یا کشف الگوهای ترتیبی استفاده می‌کند[13] ولی هیچ روشی تکنیکهای آنالیز لینکها را به طور کامل با فرایند شخصی سازی بوسیله استخراج اعتبار یا اهمیت صفحات وب در گراف ترکیب نکرده است. در این مقاله با ترکیب داده‌های استفاده کاربران و داده‌های ساختاری صفحات وب الگوریتمی ترکیبی مبتنی بر اتوماتای یادگیر توزیع شده و الگوریتم PageRank به منظور پیشنهاد صفحات ارائه شده است که الگوریتم ارائه شده را می توان علاوه بر شخصی سازی برای تغییر پیوند صفحات و اصلاح سایت به گونه ای که اطمینان حاصل شود هر کاربر در بین ساختار وب به صورت بهینه هدایت می شود، استفاده کرد تا کاربر با صرف کمترین زمان به نتایج مطلوب خود دست می یابد. در ادامه ابتدا در بخش 2 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 می شود.

۲- اتوماتاهای 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. اتوماتای یادگیر با دریافت سیگنال و با توجه به آخرین اقدام انجام شده، 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.

[1] Content mining
[2] Log files
[3] Structure mining
[4] Personalization

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[5] مقادیر c_iها ثابت هستند، حال آنکه در یک محیط غیر ایستا[6] 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. 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 اتوماتای یادگیر، $\beta=\{\beta_1,\beta_2,\cdots,\beta_r\}$ مجموعه ورودیهای اتوماتای 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 اتوماتای یادگیر در مرحله nأم اقدام α_i خود را انتخاب نموده و محیط ارزیابی خود را توسط سیگنال تقویتی $\beta(n)$ به اتوماتای یادگیر اعلام کند. با استفاده از الگوریتم یادگیری خطی، اتوماتای یادگیر بردار احتمال انتخاب اقدامهای خود را مطابق رابطه (۱) تنظیم می‌کند.

$$p_i(n+1)=p_i(n)+a.(1-\beta(n)).(1-p_i(n))-b.\beta(n).p_i(n) \qquad (۱)$$
$$p_j(n+1)=p_j(n)+a(1-\beta(n)).p_j(n)+\frac{b.\beta(n)}{r-1}-b.\beta(n).p_j(n) \quad if\ j\neq i$$

که a پارامتر پاداش و b پارامتر جریمه می‌باشد. اگر a و b با هم برابر باشند، الگوریتم L_{R-P}[7] اگر b از a خیلی کوچکتر باشد، الگوریتم $L_{R\varepsilon P}$[8] و اگر b صفر باشد، الگوریتم L_{R-I}[9] نام دارد [14].

اتوماتای یادگیری که در بالا معرفی شد، دارای تعداد اقدامهای ثابتی می‌باشد. در بعضی از کاربردها به اتوماتای یادگیر با تعداد اقدام متغیر[10] نیاز می‌باشد [15]. 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 (۲) ایجاد می‌کند. آنگاه اتوماتای یادگیر یک اقدام از مجموعه اقدامهای فعال خود را

[5] Stationary
[6] Non-Stationary
[7] Linear Reward-Penalty
[8] Linear Reward epsilon Penalty
[9] Linear Reward Inaction
[10] Learning automata with changing number of actions

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 (۳) و در صورت دریافت پاسخ نامطلوب طبق رابطه (۴)بِروز می‌کند. سپس اتوماتای یادگیر بردار احتمال اقدامهای خود $p(n)$ را با استفاده از بردار $\hat{p}(n+1)$ و طبق رابطه(۵) بِروز می‌کند.

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

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

$V(n)$ is the set of enabled actions (۲)

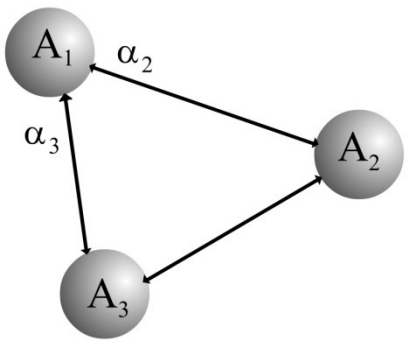
$$\hat{p}_i(n+1) = \hat{p}_i(n) + a.(1 - \hat{p}_i(n))$$
$$\hat{p}_j(n+1) = \hat{p}_j(n) - a.\hat{p}_i(n) \qquad \forall j \;\; j \neq i \quad (۳)$$

$$\hat{p}_i(n+1) = (1-b).\hat{p}_i(n)$$
$$\hat{p}_j(n+1) = \frac{b}{\hat{r}-1} + (1-b)\hat{p}_j(n) \qquad \forall j \;\; j \neq i \quad (۴)$$

$$\begin{aligned} &p_i(n+1) = \hat{p}_i(n+1).K(n) && \text{for all i}, \alpha_i \in V(n) \\ &p_j(n+1) = p_j(n) && \text{for all j}, \alpha_j \notin V(n) \end{aligned} \quad (۵)$$

۲-۱- اتوماتای یادگیر توزیع‌شده

اتوماتای یادگیر توزیع‌شده 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 اوتوماتاهای 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 $\underline{p}^j$ 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. برای کسب اطلاعات بیشتر در باره اتوماتای یادگیر توزیع شده میتوان به [16-19] مراجعه نمود. اتوماتای یادگیر توزیع شده قبلا برای رتبه بندی صفحات وب [1] و تعیین شباهت اسناد وب بکاربرده شده است[2].



شکل ۲. اتوماتای یادگیر توزیع‌شده

۳- الگوریتم PageRank

الگوریتم PageRank معروفترین الگوریتم تحلیل پیوند می‌باشد که در سال 1998 ارائه شد [20] و در موتور جستجوی گوگل مورد استفاده قرار گرفت. این الگوریتم با انتساب وزن به هر صفحه نتایج جستجو را بر اساس این وزن‌ها مرتب می‌کند. شهود اساسی PageRank را از دو دیدگاه می‌توان بررسی کرد: دیدگاه اول تعریف «درجه اعتبار» صفحات است. مسلما صفحه‌ای مهم و معتبر است که از صفحات مهم و معتبر دیگری مورد اشاره باشد. در واقع در این شهود به صرف وجود پیوندها توجه نمی‌شود، بلکه کیفیت آنها نیز مورد استفاده قرار می‌گیرد. دیدگاه دوم مدل «گردش‌گر تصادفی» است. در این مدل فرض می‌کنیم که یک کاربر اینترنت به طور تصادفی در حال گردش در گراف ایجاد شده از صفحات اینترنت است. گردش‌گر با ورود

به هر سایت، با احتمال مساوی یکی از پیوندهای خروجی صفحه جاری را انتخاب می‌کند[21].

از دید ریاضی، دیدگاه دوم معادل با یک زنجیره مارکوف می‌باشد. 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. کمیت قابل محاسبه در زنجیره مارکوف، توزیع احتمالاتی حالت پایدار است که نشان دهنده احتمال حضور گردش‌گر در گام بی‌نهایت در هر صفحه می‌باشد. فرض کنیم $G = (V, E)$ 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 $\frac{1}{\deg(v)}$ به یکی از صفحات مجاور $\{v \mid u \to v\}$ خواهد رفت. بر این اساس عنصر p_{uv} برابر $\frac{1}{\deg(v)}$ و در صورت عدم وجود پیوند $P_{uv} = 0$ خواهد بود. 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 که 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، $\vec{v} = \left[\frac{1}{n}\right]_{n\times 1}$ 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)، آنگاه ماتریس $P' = P + D$ به دست می‌آید که در آن $D = \vec{d}\,\vec{v}^T$. شرط بعدی لزوم توزیع احتمالی پایدار طبق قضیه ارگودیک در زنجیره‌های مارکوف، کاهش‌ناپذیر بودن و به عبارت دیگر قویا 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 قویا همبند شده و دارای توزیع احتمالی پایدارخواهد بود. ماتریس گذار به صورت $P'' = cP' + (1-c)E$ درمی‌آید که در آن $E = \vec{e}\vec{v}^T$ و $\vec{e} = (1,1,...,1)$ و 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[22]. برای محاسبه احتمالات توزیع در حالت پایدار از روش نمایی استفاده می‌شود. فرض کنیم $\vec{x}$ بردار توزیع احتمال در حالت پایدار باشد. با حل را بطه بازگشتی $\vec{x}^{(k+1)} = P''\vec{x}^{(k)} = P''^{(k)}\vec{x}^{(0)}$ احتمالات محاسبه می‌شود. $\vec{x}^{(0)}$ مقدار اولیه توزیع می‌باشد که به صورت یکنواخت بین همه صفحات توزیع می‌شود[21].

۴- الگوریتم پیشنهادی

در این بخش روشی مبتنی بر اتوماتای یادگیر توزیع شده- که از اطلاعات پیمایش کاربران استفاده میکند- و الگوریتم PageRank به منظور پیشنهاد صفحات ارائه می‌کنیم. الگوریتم ارائه شده اطلاعات پیوندی صفحات و استفاده کاربران را با هم ترکیب می‌کند. در الگوریتم ارائه شده، ماتریس p و بردار شخصی سازی $\vec{v}$ به جای اطلاعات پیوندی گراف سایت بر اساس اطلاعات پیمایش کاربران تولید می‌شوند. بدین منظور ابتدا اتوماتای یادگیر توزیع شده احتمال گذار بین صفحات را بر اساس اطلاعات پیمایش کاربران محاسبه می‌کند. همچنین در الگوریتم PageRank بردار شخصی‌سازی بر اساس تعداد بازدید هر صفحه مقداردهی می‌شود. با داشتن ماتریس گذار بین صفحات و بردار شخصی‌سازی، با استفاده از رابطه الگوریتم PageRank توزیع احتمالات پایدار صفحات در بی‌نهایت محاسبه می‌شود که در پیشنهاد صفحات به کاربران مورد استفاده قرار می‌گیرد. الگوریتم ارائه شده علاوه بر پیشنهاد صفحات می‌تواند برای تغییر ساختار یک سایت، حذف و اضافه کردن پیوند بین صفحات، مورد استفاده قرار گیرد. در ادامه به بررسی کامل مراحل الگوریتم می‌پردازیم.

۴-۱- محاسبه احتمال انتقال بین صفحات

در الگوریتم PageRank، در ماتریس گذار احتمال رفتن از هر صفحه به صفحه‌های خروجی آن، به طور یکنواخت بین همه صفحات توزیع می‌شود. فرض کنیم که صفحه‌ای از

میان صفحات خروجی یک صفحه نسبت به صفحات خروجی دیگر بیشتر مشاهده شده است. در نتیجه به نظر می‌رسد که آن صفحه برای کاربران نسبت به صفحات خروجی دیگر جالب‌تر بوده و بهتر است این صفحه را به کاربر پیشنهاد کنیم. در واقع شهود اینکه از بین صفحات کاندید برای پیشنهاد، صفحه‌ای پیشنهاد شود که قبلا توسط کاربران بیشتری مشاهده شده است، در ذهن تداعی می‌نماید. بر اساس این شهود، مبتنی بر اتوماتای یادگیر توزیع شده احتمال گذار از یک صفحه uبه صفحات دیگر که معادل با احتمال پیشنهاد صفحات دیگر هنگامی که کاربر در صفحه u می‌باشد، را محاسبه می‌نماییم. بدین منظور از یک اتوماتای یادگیر توزیع‌شده با n اتوماتای یادگیر با تعداد اقدامهای متغیرکه هر یک $n-1$اقدام دارند، استفاده می‌شود. برای هر اتوماتای یادگیر در هر زمان تنها زیرمجموعه‌ای از اقدامهایش فعال و میتواند قابل استفاده باشد. تعداد اقدامهای اتوماتای یادگیر متناظر با هر صفحه مانند i برابر است با تعداد صفحاتی که ممکن است کاربر بعد از آن صفحه مشاهده کند. هنگامیکه یک کاربر پس از مشاهده صفحه i، صفحه j را مشاهده می‌کند اقدام jام در اتوماتای iام پاداش می‌گیرد. در ابتدای الگوریتم، تمامی اقدام‌های اتوماتاهای یادگیر در اتوماتای یادگیر توزیع‌شده غیر فعال می‌باشند. با حرکت یک کاربر از صفحه i که در حال مشاهده میباشد، به صفحه j ، اقدام متناظر با آن صفحه (اقدام j) در اتوماتای یادگیر i فعال شده و اتوماتای یادگیر i به اقدام j خود پاداش می‌دهد. آنگاه اتوماتای یادگیر j در اتوماتای یادگیر توزیع‌شده فعال می‌شود. مراحل فوق تا پایان حرکت کاربر در مجموعه صفحات ادامه مییابد. بعد از اتمام یادگیری از اطلاعات پیمایش تمام کاربران، احتمال اقدام jام در اتوماتای iام در درایه P_{ij} قرار می‌گیرد که بیانگر احتمال مشاهده دو صفحه iام و jام به طور متوالی است .

۴-۲- رتبه بندی صفحات

در الگوریتم PageRank برای محاسبه احتمالات در حالات پایدار باید ماتریس p و بردار شخصی سازی $\vec{v}$ مقدار دهی شوند. نحوه تولید ماتریس p در مرحله مبتنی بر اتوماتای یادگیر و استفاده از اطلاعات پیمایش کاربران در بخش قبل بررسی شد. با مقداردهی غیریکنواخت بردار شخصی‌سازی، می‌توان احتمالات حالت پایدار صفحات خاص را افزایش داد. هر چه مقدار این بردار برای صفحه‌ای بیشتر باشد، اهمیت آن صفحه نیز بیشتر خواهد بود. ما در اینجا از تعداد دفعات مشاهده هر صفحه توسط کاربران به عنوان معیاری برای مقداردهی بردار $\vec{v}$ استفاده می‌کنیم. فرض کنیم که $w(i)$ تعداد دفعاتی باشد که در اطلاعات پیمایش کاربران، صفحه i مشاهده شده است. مقدار $\vec{v}(i)$ را با $\frac{w(i)}{\sum_{j\in V} w(j)}$ مقداردهی می‌کنیم. 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 مقدار دهی p و $\vec{v}$ می‌توان با استفاده از روش نمایی احتمالات حالت پایداری صفحات را بدست آورد. احتمال پایداری هر صفحه اهمیت هر صفحه را نسبت به صفحات دیگر براساس اطلاعات پیمایش کاربران نشان می‌دهد.

۴-۳- پیشنهاد صفحات

هدف از شخصی سازی بر اساس اطلاعات پیمایش کاربران محاسبه یک مجموعه پیشنهادی، rs، برای نشست کاربر جاری می‌باشد[۲۳و۲۲] 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 زیر به دست می‌آید

$$\Pr(p_1 \to p_2 \to p_3 \ldots \to p_k) = \Pr(p_1) \times \prod_{i=2}^{k} \Pr(p_{i|}\ p_{i-m} \ldots p_{i-1}) \quad (۷)$$

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

با :

$$\Pr(p_1 \to p_2 \to p_3) = \Pr(p_1)\Pr(p_2|p_1)\Pr(p_3|p_2) = \Pr(p_1)\frac{\Pr(p_1 \to p_2)\Pr(p_2 \to p_3)}{\Pr(p_1)\Pr(p_2)} \quad (۷)$$

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 $\vec{x}$ 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 کنترل تعداد صفحات پیشنهادی از فاکتور حدآستانه پیشنهاد استفاده می‌شود که با rt نشان می‌دهیم. صفحاتی که امتیاز آنها از rt بیشتر باشد به کاربر پیشنهاد می‌شود. بدیهی است که با افزایش rt تعداد صفحات مجموعه rs کاهش می‌یابد. rt را می‌توان علاوه بر تنظیم بر اساس امتیاز، بر اساس تعداد صفحات پیشنهادی نیز مقدار دهی نمود.

۵- ارزیابی الگوریتم پیشنهادی

۵-۱- مدل شبیه‌سازی

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. برای استفاده از این روش مجموعه داده‌های استانداردی که از چند سایت معتبر استخراج شده‌اند در دسترس می‌باشد. روش دوم مدل ارائه شده در [۲۴] می‌باشد. در این روش Lui 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 اطلاعات استفاده از وب چندین سایت وب بزرگ مانند مایکروسافت، تایید کرده‌اند. در این مقاله ما از داده‌های 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[۲۵]. 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.

۵-۲- معیار و متدولوژی ارزیابی

پارامترهای تاثیرگذار در کارایی الگوریتم عبارتند از: rw اندازه پنجره پیشنهاد، rt حد آستانه پیشنهاد. برای بررسی دقت الگوریتم ارائه شده رویه زیر اتخاذ شده است. ابتدا با استفاده از مجموعه یادگیری الگوریتم را اجرا می‌کنیم. بر اساس مقدار rw ، از هر نشست در مجموعه تست که اندازه آن حداقل $rw+rt$ می‌باشد، rw صفحه متوالی را انتخاب کرده و به الگوریتم می‌دهیم. معیار ارزیابی رابطه معرفی شده در [۲۶] است. فرض کنیم مجموعه $rp=\{x_{rw+1}, x_{rw+2}, \ldots, x_{rw+|rs|}\}$ صفحات مشاهده شده توسط کاربر در ادامه نشست واقعی باشد. درجه شباهت مجموعه پیشنهادی و مجموعه صفحات واقعی از رابطه زیر به دست می‌آید:

$$Sim(rs, rp) = \frac{|rs \cap rp|}{|rs|} \quad (۸)$$

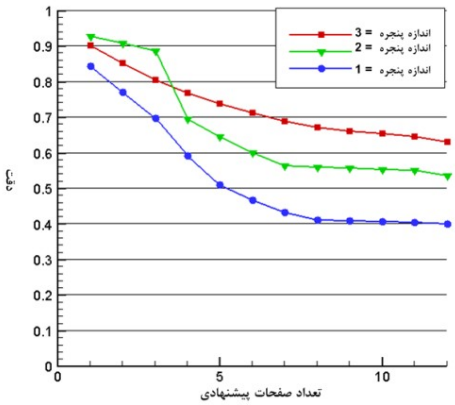
این معیار همپوشانی دو مجموعه را نشان می‌دهد. در این رابطه ترتیبی برای صفحات پیشنهادی در نظر گرفته نشده است. در صورتی که تعداد صفحات پیشنهادی را به یک صفحه محدود کنیم، کافی است تا صفحه پیشنهادی الگوریتم را با صفحه $rw+1$ام در نشست کاربر مقایسه می‌کنیم. در این صورت دقت الگوریتم درصد نشست‌هایی می‌باشد که در آنها صفحه پیشنهادی الگوریتم با صفحه واقعی یکسان می‌باشد.

۵-۳- نتایج شبیه‌سازی

برای ارزیابی دقت الگوریتم مجموعه داده‌ها را به دو دسته به نسبت ۲ به ۱ تقسیم می‌کنیم که مجموعه اول برای یادگیری و مجموعه دوم برای تست و ارزیابی مورد استفاده قرار می‌گیرد. برای محاسبه بردار $\vec{x}$ با روش نمایی، تعداد تکرارها را برابر ۲۰۰ قرار دادیم که این مقدار تکرار برای همگرایی کافی بود. شکل ۳ نتایج دقت الگوریتم را نسبت به

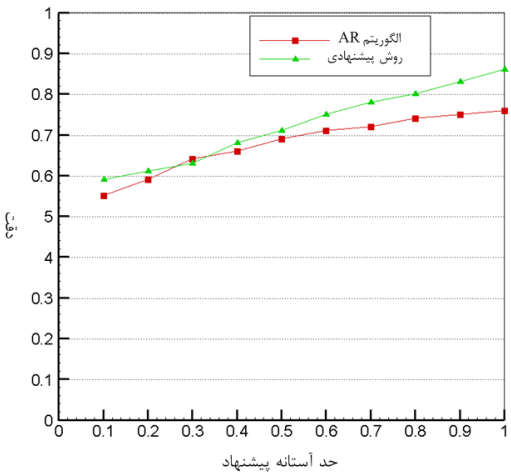
تعداد صفحات پیشنهادی برای ۳ حالت اندازه پنجره پیشنهاد نشان می‌دهد. همانطور که از شکل ۳ پیداست با افزایش تعداد صفحات در پنجره پیشنهاد، rw، از ۱ به ۳ ، دقت پیشنهاد بالا می‌رود. برای حالتی که تعداد صفحات پیشنهادی کمتر از۳ می‌باشد، دقت با اندازه پنجره ۲ از پنجره با اندازه ۳ بیشتر است که این استثنا به دلیل محتویات صفحات سایت و رفتار کاربران می‌باشد. با توجه به رابطه ۶ ، با افزایش طول پنچره حجم محاسبات بالا رفته و سرعت الگوریتم نسبت به حالتی که طول پنجره پیشنهاد کمتر است بیشتر می‌شود.

همان طور که در شکل ۳ مشخص است برای حالتی که تعداد صفحات پیشنهادی برابر ۱ و 3 صفحه از آخرین صفحات ملاقات شده توسط کاربر برای پیشنهاد استفاده می‌شود دقت الگوریتم بیش از ۹۰٪ است.



شکل ۳. مقایسه دقت الگوریتم با اندازه پنجره پیشنهاد و تعداد صفحات پیشنهادی متفاوت

در نهایت کارایی روش پیشنهادی با الگوریتم AR [۴] مقایسه گردیده است. در الگوریتم AR روند کار به این ترتیب است که آیتمهای تکرار شونده در یک گراف مستقیم بدون دور ذخیره می شوند. این گراف از سطح ۰ تا K (ماکزیمم سایز آیتمهای تکرار شونده) سازماندهی می شود. هر نود در عمق d این گراف متناظر با آیتم ست I با سایز d است که به آیتم ست های سطح d+1 ای که شامل آیتمهای I هستند لینک دارد. ریشه این گراف نیز در سطح صفر شامل آیتم ست خالی است. بدین ترتیب اگر نشست کاربر با پنجره ای به طول rw و آیتمهای تکرار شونده به عنوان ورودی به الگوریتم داده شوند، الگوریتم آیتم ست های تکرار شونده به طول $rw+1$ ام که شامل نشست جاری هستند را با جستجوی اول عمق پیدا می کند و ارزش صفحات کاندید(جدید) را بر اساس محاسبه ضریب اطمینان قوانین انجمنی که شامل آن صفحه می باشند محاسبه می کند در نهایت صفحاتی که ارزش آنها بیش از حد آستانه باشد در لیست صفحات پیشنهادی قرار می گیرند. شکل ۴ مقایسه دقت الگوریتم پیشنهادی با الگوریتم AR را نمایش می دهد.



شکل ۴. مقایسه دقت الگوریتم پیشنهادی با الگوریتم ارائه شده در [۴] بر اساس حد آستانه پیشنهادی متفاوت و اندازه پنجره ۲

۶- نتیجه‌گیری

در این مقاله روشی مبتنی بر اتوماتای یادگیر توزیع شده که از اطلاعات پیمایش کاربران استفاده می‌کند و الگوریتم تحلیل پیوند PageRank به منظور پیشنهاد صفحات وب ارائه گردید. الگوریتم ارائه شده همچنین می‌تواند برای بهبود ساختار استاتیک پیوندهای موجود بین صفحات سایت مورد استفاده قرار گیرد. الگوریتم ارائه شده برای پیشنهاد تعداد صفحات کمتر ، از دقت بالایی برخوردار بوده و مناسب برای شخصی سازی وب برخط می‌باشد. در ادامه کار، هدف استفاده از اطلاعات دیگر پیمایش کاربران مانند زمان صرف شده توسط هر کاربر در هر صفحه، شباهت بین صفحات و اطلاعات معنایی سایت مانند دانش هستان سایت در بهبود الگوریتم می‌باشد.

مراجع

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

[۲] 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.

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

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

[5] H. Dai, B. Mobasher, ”Integrating Semantic Knowledge with Web Usage mining for Personalization”, 2004.

[6] 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.

[7] 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.

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

[9] T. Haveliwala, “Topic-Sensitive PageRank”, Proceeding of WWW Conference, Hawaii, 2002.

[10] M. S. Aktas, M. A. Nacar, F. Menczer, “Personalizing PageRank Based on Domain Profiles”, Proceeding of WEBKDD Workshop, Seattle, 2004.

[11] 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.

[12] 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.

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

[14] K. S. Narendra, M. A. L. Thathachar, “Learning Automata: An Introduction”, Prentice Hall, 1989.

[15] 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.

[16] M. Alipour, M. R. Meybodi, "Solving Maximal independent Set Problem Using Distributed Learning Automata”, Proceedings of 14th Iranian Electrical Engineering Conference (ICEE2006), Tehran, Iran, 2006.

[17] M. Alipour, M. R. Meybodi, "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, 2006.

[18] M. Alipour, M. R. Meybodi, "Solving Traveling Salesman Problem Using Distributed Learning Automata", Proceedings of 10th Annual CSI Computer Conference, Computer Engineering Department, Tehran, Iran, pp. 759-761, 2005.

[19] M. R. Meybodi, H. Beigy, "Solving Stochastic Shortest Path Problem Using Monte Carlo Sampling Method: A Distributed Learning Automata Approach", SpringerVerlag Lecture Notes in Advances in Soft Computing: Neural Networks and Soft Computing, pp. 626-632, 2003.

[20] L. Page, S. Brin, R. Motwani, T. Wingord, “The PageRank Citation Ranking: Bringing Order to the Web”, Stanford University, 1998.

[21] A. N. Langville, C. D. Meyer, “Deeper Inside PageRank”, Internet Mathematics, pp. 335–400, 2004.

[22] 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.

[23] 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.

[24] 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.

[25] http://maya.cs.depaul.edu/~classes/ect584/data/cti-data.zip

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