

یک روش مبتنی بر اتوماتای یادگیر توزیع شده برای تعیین ساختار اسناد وب

بابک اناری^۱، محمد رضا میبدی^۲

چکیده

در این مقاله یک روش مبتنی بر اتوماتای یادگیر توزیع شده جدید برای کشف ساختار ارتباطی اسناد وب پیشنهاد می‌گردد. این روش از الگوهای حرکتی ذخیره شده کاربران در لاگ فایلها برای تعیین ساختار اسناد وب و میزان ارتباط آنها با یکدیگر استفاده می‌نماید. برتری این روش نسبت به تنها روش گزارش شده مبتنی بر اتوماتای یادگیر توزیع شده این است که در صورت گسترش اسناد وب نیازی به تنظیم پارامترهای اتوماتای یادگیر توزیع شده نمی‌باشد. کارایی الگوریتم پیشنهادی از طریق مقایسه آن با دو روش AntWeb, Bollen مورد ارزیابی قرار خواهد گرفت.

کلمات کلیدی

کاوش استفاده از وب، اتوماتای یادگیر توزیع شده، اتوماتاهای یادگیر

A Method based on distributed learning automata for determining web documents structure

B.anari,M.R.Meybodi

Abstract

In this paper a new method based on distributed learning automata (DLA) for discovering the structure of web documents is proposed. This method uses the users stored navigation patterns on the log files in order to determine the web documents structure. The advantage of the proposed algorithm over the only DLA based algorithm reported earlier is that in the proposed method there is no need to change the parameters of the distributed learning automata when the number of documents changes. Computer simulations have been conducted to evaluate the performance of the proposed method. The proposed method is also compared with two existing methods: AntWeb, and Bollen.

Keywords: Web Usage Mining, Distributed Learning Automata, Learning automata

۱- مقدمه

استفاده از الگوهای حرکتی کاربران در بین صفحات وب، یک روش مهم برای کسب اطلاعات به منظور یاری رساندن کاربران وب در امر جستجو و حرکت در وب می باشد. استخراج اطلاعات از اسناد وب بوسیله تکنیکهای داده کاوی راه، کاوش وب^۱ می گویند. کاوش وب در سه سطح مطرح است. در سطح محتوا، در سطح ساختار و در سطح استفاده از وب. در سطح محتوا، هدف، کاوش محتوای وب می باشد. در سطح ساختار، هدف استفاده از توپولوژی ابر پیوندها برای تعیین ارتباط اسناد وب است. در سطح استفاده از وب، هدف، کشف اطلاعات مفید از داده هایی است که از تعامل کاربران در هنگام استفاده از وب بدست می آید.

بیشتر تکنیکهای موجود برای کشف ساختار اسناد وب از اطلاعات موجود در لاگ فایلها استفاده می کنند. لاگ فایلها، فایلهایی هستند که مجموعه درخواستهای کاربران به صفحات وب را در خود ذخیره می کنند [3]. روشهای موجود برای کشف ساختار اسناد وب از طریق کاوش در لاگ فایلها یکسری اطلاعات آماری را از آنها استخراج می کنند که می توان از آنها به عنوان ابزاری برای کشف ساختار اسناد وب استفاده کرد.

برخی از این روشهای تعیین ساختار اسناد وب با استفاده از لاگ فایلها عبارتند از: استفاده از روشهای آماری^۲، استفاده از قوانین یادگیری، مثل استفاده از قانون یادگیری هب در روش بولن [4]، استفاده از کلونی مورچه ها در روش AntWeb [13]، استفاده از اتوماتاهای یادگیر توزیع شده³ [1,11,12]، استفاده از زنجیره مارکف [5] و استفاده از گرامرهای احتمالی ابرمتن^۴ [2]. روشهای گزارش شده مبتنی بر اتوماتاهای یادگیر توزیع شده که بعداً در این مقاله مفصلاً به آن اشاره می شود دارای دو نقطه ضعف می باشد. کارایی آن در مقایسه با روش AntWeb پایینتر است و همچنین در اثر افزایش تعداد اسناد وب، برای حفظ کارایی روش، پارامترهای اتوماتاهای یادگیر بایستی مجدداً تنظیم شود. نقطه ضعف دوم در هر دو روش AntWeb و Bollen نیز وجود دارد. دو روش AntWeb و Bollen با وجود مزایایی که نسبت به روشهای قدیمی تر دارند، بدلیل اینکه از ماتریس ارتباطات اسناد استفاده می کنند برای استفاده در مجموعه های بزرگ و قابل گسترش مناسب نمی باشند. نقطه ضعف استفاده از زنجیره مارکف، محاسبات بالا بواسطه محاسبه توان n ماتریس انتقال می باشد. هر چند با روشهای فشرده سازی می توان هزینه محاسبات را در این روش تا حدی کاهش داد [5].

در این مقاله روشی مبتنی بر اتوماتای یادگیر توزیع شده جدید که برای تعیین ساختار اسناد وب از اطلاعات موجود در لاگ فایلها استفاده می کند، پیشنهاد می گردد.

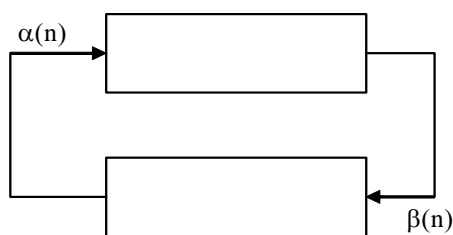
در این روش مانند روشهای گزارش شده مبتنی بر اتوماتای یادگیر توزیع شده به هر سند یک اتوماتای یادگیر اختصاص داده می شود که وظیفه اش یادگیری ارتباطات آن سند با اسناد دیگر می باشد. روش پیشنهادی که از مفهوم آنتروپی برای بروز رساندن بردار احتمالات اعمال اتوماتاهای یادگیر توزیع شده استفاده می نماید، ضمن داشتن کارایی در حد روشهای AntWeb و Bollen، پارامترهای یادگیری در اتوماتای یادگیر توزیع شده را با توجه به تعداد اسناد وب بصورت پویا تنظیم می کند. کارایی الگوریتم پیشنهادی از طریق مقایسه با دو روش AntWeb و Bollen مورد ارزیابی قرار می گیرد. ادامه مقاله بدین صورت سازماندهی شده است: در بخش ۲ اتوماتاهای یادگیر و اتوماتای یادگیر توزیع شده و روشهای مبتنی بر اتوماتای یادگیر توزیع شده به اختصار شرح داده می شوند. در بخش ۳ روش پیشنهادی و در بخش ۴ نتایج شبیه سازیها ارائه می گردد. بخش پایانی نتیجه گیری می باشد.

۲- اتوماتای یادگیر توزیع شده و تعیین ساختار اطلاعاتی اسناد وب

در این بخش ابتدا به اتوماتاهای یادگیر و اتوماتای یادگیر توزیع شده بطور مختصر اشاره می شود و سپس روشهای پیشین تعیین ساختار در وب که مبتنی بر اتوماتای یادگیر توزیع شده می باشند، شرح داد می شود

۲-۱- اتوماتاهای یادگیر^۵

اتوماتای یادگیر یک مدل انتزاعی است که تعداد محدودی عمل را می تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی به اتوماتای یادگیر داده می شود. اتوماتای یادگیر از این پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب می کند. شکل ۱ ارتباط بین اتوماتای یادگیر و محیط را نشان می دهد.



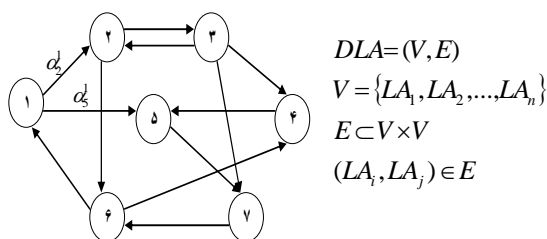
:

محیط^۶: محیط را می توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودیه ها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجیه ها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالات جرمیه می باشد. هر گاه β مجموعه دو عضوی باشد، محیط از نوع P می باشد. در چنین

۲-۲- اتوماتای یادگیر توزیع شده

یک اتوماتای یادگیر توزیع شده شبکه‌ای از اتوماتاهای یادگیر است که برای حل یک مساله خاص با یکدیگر همکاری دارند. در این شبکه از اتوماتاهای یادگیر همکار 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 متصل شده به این اتوماتای یادگیر متناظر با این عمل گردد. به عبارت دیگر انتخاب یک عمل توسط یک اتوماتای یادگیر در این شبکه متناظر با فعال شدن یک اتوماتای یادگیر دیگر در این شبکه است.

یک DLA توسط یک گراف که هر یک از رئوس آن یک اتوماتای یادگیر نشان داده می‌شود. وجود یال (LA_i, LA_j) در این گراف بدین معناست که انتخاب عمل α_j^i توسط LA_i باعث فعال شدن LA_j می‌گردد. تعداد اعمال قابل انتخاب توسط LA_k بصورت $p^k = \{p_1^k, p_2^k, \dots, p_{r_k}^k\}$ نمایش داده شود. در این مجموعه عدد p_m^k نشان دهنده احتمال مربوط به عمل a_m^k است. انتخاب عمل a_m^k توسط LA_k باعث 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 کاربرد های آن می‌توان به [15,16,17,18,19,20,21,22] مراجعه نمود.



$$\begin{aligned} DLA &= (V, E) \\ V &= \{LA_1, LA_2, \dots, LA_n\} \\ E &\subset V \times V \\ (LA_i, LA_j) &\in E \end{aligned}$$

:

۲-۳- تعیین ساختار اطلاعاتی اسناد با استفاده از

اتوماتای یادگیر توزیع شده

در روشهای مبتنی بر اتوماتای یادگیر توزیع شده [1,11] برای ایجاد یک ساختار اطلاعاتی پویا در مجموعه‌های بزرگ از اسناد مانند صفحات وب، اسناد و کاربران استفاده کننده از آن نقش یک محیط تصادفی را برای اتوماتاهای یادگیر موجود در DLA ایفا می‌کنند. خروجی DLA یک دنباله از اسناد مرور شده توسط یک کاربر هستند که مسیر حرکت کاربر را به سمت یک سند مورد نظر نشان می‌دهد. محیط با استفاده از این دنباله پاسخی برای DLA تولید می‌کند. با استفاده از این پاسخ ساختار داخلی اتوماتاهای یادگیر در اتوماتای یادگیر توزیع شده طبق الگوریتم یادگیری بروز می‌شود.

محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع $Q, \beta(n)$ می‌تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله $[0,1]$ و در محیط از نوع $S, \beta(n)$ متغیر تصادفی در فاصله $[0,1]$ است. c_i احتمال اینکه عمل α_i نتیجه نامطلوب داشته باشد می‌باشد. در محیط ایستای^۷ مقادیر c_i بدون تغییر می‌مانند، حال آنکه در محیط غیر ایستای^۸ این مقادیر در طی زمان تغییر می‌کنند. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم بندی می‌گردند. در ادامه به شرح مختصری درباره اتوماتای یادگیر با ساختار متغیر که در این مقاله از آنها استفاده شده است می‌پردازیم.

اتوماتای یادگیر با ساختار متغیر: اتوماتای یادگیر با ساختار متغیر توسط ۴ تائی $\{\alpha, \beta, p, T\}$ نشان داده می‌شود که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عملهای اتوماتا $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ $p \equiv \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از عملها و $T[\alpha(n), \beta(n), p(n)]$ $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. در این نوع از اتوماتاها، اگر عمل α_i در مرحله n ام انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالات کاهش می‌یابند و برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالات افزایش می‌یابند. در هر حال، تغییرات به گونه‌ای 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.

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \quad (1)$$

$$p_j(n+1) = (1-a)p_j(n) \quad j \neq i \quad \forall j$$

$$p_i(n+1) = (1-b)p_i(n)$$

$$p_j(n+1) = \frac{b}{r-1} + (1-b)p_j(n) \quad j \neq i \quad \forall j \quad (2)$$

در روابط فوق، a پارامتر پاداش و b پارامتر جریمه می‌باشد. با توجه به مقادیر a و b سه حالت را می‌توان در نظر گرفت. زمانی که a و b با هم برابر باشند، الگوریتم را LRP ^{۱۰} می‌نامیم. زمانی که b از a خیلی کوچکتر باشد، الگوریتم را LRP ^{۱۱} می‌نامیم. زمانی که b مساوی صفر باشد، الگوریتم را LR ^{۱۲} می‌نامیم. برای مطالعه بیشتر در باره اتوماتاهای یادگیر می‌توان به مراجع برای مطالعه بیشتر در باره اتوماتاهای یادگیر می‌توان به [6,8,9,10,14] مراجعه کرد.

تا به حال دو روش مبتنی بر اتوماتای یادگیر توزیع شده برای تعیین ساختار اطلاعاتی اسناد پیشنهاد شده است. در روش اول (DLA-FA) فرض بر این است که با افزایش تعداد اسناد، تعداد اسنادی که یک سند به آن مرتبط می‌باشد ثابت باقی می‌ماند و به همین دلیل به هر سند یک اتوماتای یادگیر با تعداد اعمال ثابت تخصیص داده می‌شود. در روش دوم (DLA-VA) تعداد اسنادی که یک سند به آن مرتبط می‌باشد متغیر فرض شده است و به همین دلیل به هر سند یک اتوماتای یادگیر با تعداد اعمال متغیر تخصیص داده می‌شود. این دو روش در ادامه شرح داده می‌شود.

در روش (DLA-FA) اندازه بردار احتمال برای هر اتوماتای یادگیر در DLA با افزایش تعداد اسناد در مجموعه اسناد تغییر پیدا نمی‌کند. هر کدام از اعمال یک اتوماتای یادگیر، متناظر با یکی از اسناد در مجموعه اسناد و احتمال انتخاب این عمل در بردار احتمالات، ارتباط این سند با سند متناظر با آن عمل می‌باشد. بعبارت دیگر بردار اعمال یک اتوماتای یادگیر می‌تواند بعنوان شناسه سند متناظر با آن اتوماتای یادگیر و بردار احتمالات میزان ارتباط این سند با دیگر سندها در مجموعه اسناد در نظر گرفته شود. بنابراین برای هر سند Doc_i یک اتوماتای یادگیر LA_i در نظر می‌گیریم که تعداد عملهای آن تعداد ثابتی می‌باشد. انتخاب عمل j توسط اتوماتای یادگیر LA_i به معنی فعال کردن اتوماتای یادگیر LA_j متناظر با سند Doc_j می‌باشد. در صورتیکه عمل انتخاب شده k امین عمل اتوماتای LA_i باشد (یعنی $a_k^i = j$) احتمال متناظر این عمل یعنی p_k^i بعنوان میزان ارتباط سندهای i و j در نظر گرفته می‌شود.

با ورود یک کاربر به سیستم و مشاهده سند Doc_i ، اتوماتای یادگیر متناظر با آن سند یعنی LA_i فعال می‌شود. با حرکت کاربر از سند Doc_i به سند Doc_j ، عمل مرتبط با این انتخاب در اتوماتای LA_i انتخاب می‌شود و به محیط اعمال می‌شود. با توجه به ثابت بودن تعداد اعمال اتوماتاهای متناظر اسناد، ممکن است عمل مرتبط با انتخاب سند Doc_j در بردار اعمال اتوماتای یادگیر Doc_i وجود نداشته باشد. در این شرایط در اتوماتای یادگیر متناظر با سند Doc_i عملی که دارای کمترین احتمال است حذف و بجای آن عمل جدید a_j^i قرار می‌گیرد و احتمال متناظر با این عمل برابر صفر قرار داده می‌شود. سپس احتمال عمل حذف شده بین احتمالات اعمال توزیع می‌شود تا مجموع احتمالات همچنان ۱ باقی بماند. این مراحل تا پایان حرکت کاربر بین اسناد برای هر دو سند متوالی مشاهده شده توسط وی انجام می‌شود. همچنین ممکن است کاربر دوباره به Doc_i برگردد که این حرکت یک دور در مسیر حرکت او می‌باشد و نشاندهنده عدم رضایت از حرکت قبلی به سمت سند Doc_j می‌باشد. پس از اینکه کاربر سیستم را ترک کرد، با توجه به مسیر حرکت کاربر، اعمال انتخاب شده توسط اتوماتاهای یادگیر در طول مسیر طی شده در

صورتیکه جزیی از یک دور نباشند، پاداش داده می‌شوند. هرچه مسیر طی شده توسط کاربر کوتاهتر باشد میزان پاداش داده شده توسط الگوریتم یادگیری به اعمال انتخاب شده در طول این مسیر بیشتر می‌باشد. اعمالی که قسمتی از یک دور باشند نشاندهنده حرکت اشتباه کاربر هستند و مجازات می‌شوند. با این مراحل هر کاربر یک رشته از اتوماتاهای یادگیر را فعال نموده و احتمال اعمال آنها توسط سیستم اصلاح شده است که در نتیجه ارتباطات اسناد متناظر آن اتوماتاها اصلاح می‌شود.

روش دوم (DLA-VA) برای شرایطی که تعداد اسنادی که یک سند به آن مرتبط می‌باشد ثابت نمی‌باشد پیشنهاد شده است. در این روش، مانند روش (DLA-FA) بردار احتمالات اتوماتاهای یادگیر، برای نشان دادن میزان ارتباط بین اسناد استفاده می‌شود. نحوه استفاده از این بردار دقیقاً مانند روش DLA-FA می‌باشد. تفاوت این روش در نحوه حذف اعمال می‌باشد که در ادامه توضیح داده می‌شود. با ورود یک کاربر به سیستم و مشاهده سند Doc_i ، اتوماتای یادگیر متناظر با آن سند یعنی LA_i فعال می‌شود. با حرکت کاربر از سند Doc_i به سند Doc_j ، عمل مرتبط با این انتخاب در اتوماتای LA_i انتخاب می‌شود و به محیط اعمال می‌شود. در صورتیکه عمل مرتبط با انتخاب سند Doc_j در بردار اعمال اتوماتای یادگیر Doc_i وجود نداشته باشد، در اتوماتای یادگیر متناظر با سند Doc_i عمل جدید a_j^i اضافه می‌شود و احتمال متناظر با این عمل برابر پارامتر الگوریتم یادگیری اتوماتاهای یادگیر قرار داده می‌شود. این مراحل تا پایان حرکت کاربر بین اسناد برای هر دو سند متوالی مشاهده شده توسط وی انجام می‌شود. پس از اینکه کاربر سیستم را ترک کرد، احتمال اعمال انتخاب شده مانند روش DLA-FA بروز می‌شود. پس از این مرحله در صورتیکه در هر یک از اتوماتاهای یادگیر متناظر با اسناد مرور شده توسط کاربر، اعمالی وجود داشته باشند که دارای احتمال کوچکی باشند این اعمال از مجموعه اعمال آن اتوماتای یادگیر حذف می‌شوند و احتمال اعمال حذف شده بین احتمالات بقیه اعمال آن اتوماتای یادگیر توزیع می‌شود تا مجموع احتمالات همچنان ۱ باقی بماند.

۳- الگوریتم پیشنهادی

اختلاف الگوریتم پیشنهادی با الگوریتمهای پیشین مبتنی بر اتوماتای یادگیر توزیع شده در چگونگی بروز کردن بردار احتمال اعمال اتوماتاهای یادگیر در DLA می‌باشد. فرض کنید $p^k = \{p_1^k, p_2^k, \dots, p_r^k\}$ بردار احتمال اتوماتای یادگیر LA_k که به سند k تخصیص داده شده است باشد که p_m^k احتمال انتخاب عمل a_m^k و r تعداد اسناد می‌باشد. اگر کاربر حرکت $D_m^k \rightarrow D_k$ را انجام دهد (از سند D_k به سند D_m حرکت کند) در این صورت اتوماتای

۴- نتایج شبیه‌سازیها

برای انجام شبیه‌سازی از مدل ارائه شده در [7] برای تولید مجموعه اسناد و حرکات کاربران استفاده می‌شود. هر سند با بردار محتوای $C_n = [cw_n^1 \quad cw_n^2 \quad \dots \quad cw_n^M]$ نشان داده می‌شود که M تعداد موضوعات در سیستم می‌باشد. هر عضو این بردار میزان ارتباط سند متناظر با آن بردار را با یکی از این موضوعات را نشان می‌دهد. در این مدل توزیع موضوعات بین اسناد بصورت توزیع نرمال و پروفایل علاقه کاربران بصورت توزیع قانون توانی^{۱۳} در نظر گرفته شده است که با تغییر پارامتر این توزیع و تعداد اسناد، سیستم‌های اطلاعاتی متفاوتی می‌توان ایجاد نمود. در این مدل، انگیزه و استراتژی حرکتی کاربران نیز از طریق توزیعهای آماری مدل شده‌اند. با تغییر پارامترهای این توزیعهای آماری می‌توان کاربرانی با علایق، انگیزه‌ها و استراتژیهای متفاوت ایجاد نمود. در شبیه‌سازیها، پارامترهای تعداد موضوعها، تعداد کاربران، علایق، انگیزه‌ها و پارامترهای توزیع آنها را در طول شبیه‌سازی ثابت می‌باشد.

برای ارزیابی الگوریتم پیشنهادی از معیاری بنام کورولیشن استفاده می‌شود. کورولیشن دو مجموعه داده مانند P, P' طبق رابطه ۷ محاسبه می‌شود که در این رابطه N تعداد داده‌ها می‌باشد. مجموعه P ساختار ایجاد شده توسط مدل شبیه‌سازی [7] و P' نیز ساختار بدست آمده توسط الگوریتم پیشنهادی را نشان می‌دهد.

$$Corr(P, P') = \frac{\sum PP' - (\sum P \sum P') / N}{\sqrt{(\sum P^2 - (\sum P)^2 / N)(\sum P'^2 - (\sum P')^2 / N)}} \quad ()$$

$$P = \{P_{ij} | i, j = 1, 2, \dots, n, \quad i \neq j\} \quad ()$$

$$P'_{ij} = \{P'_{ij} | i, j = 1, 2, \dots, n, \quad i \neq j\} \quad ()$$

$$P_{ij} = \frac{(d_{ij})^{-1}}{\sum_{k=1}^n (d_{ik})^{-1}} \quad ()$$

$$P'_{ij} = \text{probability of selection of action } j \text{ of } LA_i \quad ()$$

$$d_{ij} = \sqrt{\sum_{k=1}^M (cw_i^k - cw_j^k)^2} \quad ()$$

در رابطه فوق d_{ij} فاصله اقلیدسی بین دو سند i, j از یکدیگر در مدل شبیه‌سازی می‌باشد. شکل ۴ مقایسه الگوریتم پیشنهادی با روش DLA-FA را نشان می‌دهد. در این شبیه‌سازی تعداد کاربران 20,000، تعداد موضوعات مرتبط با هر سند 5 و تعداد اسناد وب 30 در نظر گرفته شده است. مقایسه الگوریتم پیشنهادی با

یادگیر LA_k بردار احتمال اعمال خود را طبق الگوریتم یادگیری زیر بروز می‌کند.

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

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

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

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

مقدار بالا برای E_m^k نشان دهنده ارتباط بیشتر بین دو صفحه سند D_k به سند D_m می‌باشد و بالعکس هرچه این مقدار کمتر باشد نشان دهنده ارتباط کمتر است. بطور مثال اگر بردار اعمال و بردار احتمال اعمال اتوماتای یادگیر LA_1 به ترتیب (A_2, A_3, A_4) و $(0.2, 0.5, 0.3)$ باشند و کاربر حرکت $D_1 \rightarrow D_2$ را انجام دهد

$$E_2^1 = -(0.2 \log 0.2 + 0.8 \log 0.8) = 0.21$$

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 $D_1 \rightarrow D_3$ 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. الگوریتم پیشنهادی بصورت شکل ۳ می‌باشد.

۱- یک DLA متناظر با ساختار اسناد 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 $D_k \rightarrow D_m$ در طول مسیر انجام بده

• بردار احتمال اعمال اتوماتای یادگیر LA_k را طبق الگوریتم یادگیری زیر بروز کن

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

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

$$a_m^k = E_m^k / (1 + E_m^k)$$

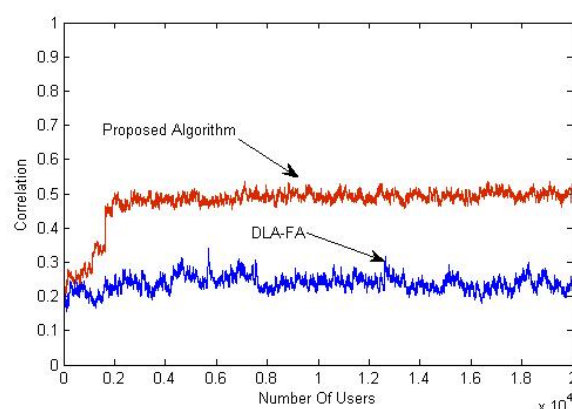
$$E_m^k = -(p_m^k \log p_m^k + (1 - p_m^k) \log(1 - p_m^k))$$

:

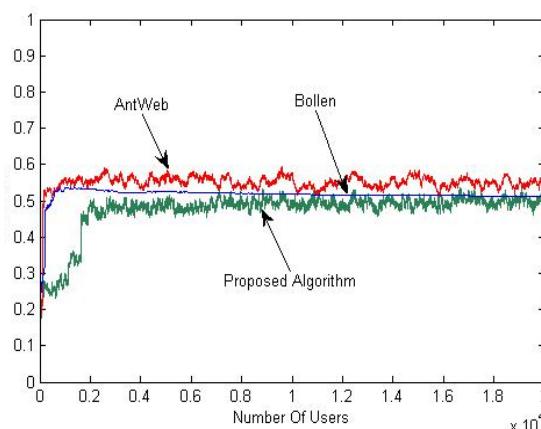
مراجع

- [1] Baradaran Hashemi, A., and Meybodi, M.R., "Web Usage Mining Using Distributed Learning Automata," Computer Engineering Department. Technical Report, 2005.
- [2] Borgers, J., and Leven, M., "Data Mining of user navigation patterns," In Proceeding of the Web Usage Analysis and User Profiling, volume 1, PP.31-36, 1999.
- [3] Borges, J., and Levene, M., "Mining Association rules in hypertext databases," In Proc. of the fourth International Conference on knowledge Discovery and Data Mining, PP.149-153, August 1998.
- [4] Heylighen, F., and Bollen, J., "Hebbian Algorithm for a Digital Library Recommendation System," Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'02) IEEE, 2002.
- [5] Jianhan, Zhu., "Mining Web Site Link Structures for Adaptive Web Site Navigation and Search," Ph.D Thesis, University of Ulster at Jordanstown, October 2003.
- [6] Lakshmivarahan, S., "Learning algorithms: theory and applications," New York: Springer-Verlag, 1981.
- [7] Liu, J., Zhang, S., and Yang, J., "Characterizing web Usage Regularities with information Foraging Agents," IEEE Transaction in Knowledge and data engineering, vol.16, no.5, may 2004.
- [8] Mars, p., Chen, J.R., and Nambir, R., "Learning Algorithms: Theory and Applications in Signal Processing," Control, and Communication, CRC Press Inc., 1996.
- [9] Meybodi, M.R., and Lakshmivarahan, S., "On a class of Learning Algorithms which have Symmetric Behavior under Success and Failure," pp.145-155. Lecture Notes in Statistics, Berlin: SpringerVerlag, 1984.
- [10] Narendra, K. S., and Thathachar, M. A. L., "Learning automata: An introduction," Prentice Hall, 1989.
- [11] 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.
- [12] 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.
- [13] Teles, W., Weigang, L., and Ralha, C., "AntWeb-The Adaptive Web Server Based on the Ants' Behavior," Proceeding of IEEE/WIC international Conference on Web Intelligence (WI'03), PP.558-564, 2003.
- [14] 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.
- [15] 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.
- [16] 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
- [17] Meybodi, M. R. And Beigy, H., "Solving Stochastic Shortest Path Problem Using Monte Carlo Sampling Method: A Distributed Learning Automata

الگوریتمهای AntWeb و Bollen در شکل ۵ آمده است. همانطوریکه مشاهده میشود، کورولیشن الگوریتم پیشنهادی در حد روشهای AntWeb, Bollen میباشد.



DLA-FA



Bollen AntWeb

۵- نتیجه گیری

در این مقاله یک الگوریتم جدید مبتنی بر اتوماتای یادگیر توزیع شده و مفهوم آنتروپی برای داده کاوی اطلاعات استفاده از وب ارائه شد. از نتایج بدست آمده از این الگوریتم می توان به عنوان ابزاری برای تعیین میزان تشابه اسناد وب استفاده کرد. الگوریتم پیشنهادی دارای سه مزیت میباشد که عبارتند از داشتن کورولیشن در حد روشهای موجود، عدم نیاز به تنظیم مجدد پارامترهای اتوماتاهای یادگیر با تغییر تعداد اسناد وب و استفاده بر خط از آن. کارایی الگوریتم پیشنهادی از طریق مقایسه آن با دو روش AntWeb, Bollen مورد ارزیابی قرار گرفت. نتایج این ارزیابی حاکی از کارایی بالای روش پیشنهادی داشته است.

- Approach*", Springer-Verlag Lecture Notes in Advances in Soft Computing: Neural Networks and Soft Computing, pp. 626-632, 2003. (ISBN: 3-7908-0005-8)
- [18] 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
 - [19] 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
 - [20] Alipour, M. and Meybodi, M. R., "Solving Dynamic Traveling Salesman Problem Using Responsive Distributed Learning Automata", Proceedings of the Second International Conference on Information and Knowledge Technology (IKT2005), Tehran, Iran, May 24-26, 2005
 - [21] 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
 - [22] Alipour, M., Meybodi, M. R., "Solving Maximal independent Set Problem Using Distributed Learning Automata", Proceedings of 14th Iranian Electrical Engineering Conference(ICEE2006), Amirkabir University, Tehran, Iran, May 16-18, 2006.

زیر نویس ها

¹Web Mining

²Statistical Method

³Distributed Learning Automata(DLA)

⁴Hypertext probabilistic grammar

⁵Learning Automata

⁶Environment

⁷Stationary

⁸Non-Stationary

⁹Variable Learning Automata

¹⁰Linear Reward Penalty

¹¹Linear Reward Epsilon Penalty

¹²Linear Reward Inaction

¹³Power-Law