دوازدهمین کنفرانس بین‌المللی انجمن کامپیوتر ایران
دانشگاه شهید بهشتی، دانشکده مهندسی برق و کامپیوتر ،تهران، ایران، ۱ تا ۳ اسفند ۱۳۸۵

داده‌کاوی استفاده از وب با استفاده از اتوماتای یادگیر توزیع‌شده

علی برادران هاشمی* محمدرضا میبدی†

چکیده

یکی از مسایل مطرح در داده‌کاوی وب، تعیین میزان شباهت اسناد با یکدیگر از طریق اطلاعات درباره چگونگی استفاده کاربران از وب می‌باشد. در این مقاله روشی مبتنی بر اتوماتای یادگیر توزیع شده که از اطلاعات چگونگی استفاده کاربران از وب استفاده می‌کند به منظور تشخیص شباهت صفحات وب پیشنهاد می‌گردد. این روش بر این ایده استوار است که اگر تعدادی از کاربران تعدادی از صفحات وب را پی در پی درخواست کنند، احتمالا این صفحات به نیازهای اطلاعاتی یکسانی پاسخ داده‌اند و در این صورت با همدیگر شباهت دارند. در این روش یک اتوماتای یادگیر به هر صفحه وب تخصیص داده می‌شود که وظیفه آن یادگیری میزان شباهت این صفحه با دیگر صفحات وب می‌باشد. از نتایج حاصل از این روش می‌توان برای ارائه صفحات پیشنهادی مشابه با یک صفحه بر اساس علایق یک یا چند کاربر و یا خوشه‌بندی صفحات مشابه استفاده نمود. نتایج شبیه‌سازیها نشان داده است که روش پیشنهادی در مقایسه با روش هب و تنها روش گزارش شده مبتنی بر اتوماتای توزیع شده در تشخیص شباهت صفحات از کارایی بالاتری برخوردار است. بطورتیکه کوریلیشن ماتریس شباهت بدست آمده با ماتریس شباهت صفحات، در الگوریتم پیشنهادی بترتیب ۰,۱ و ۰,۲ بیشتر از این مقدار در تنها روش گزارش شده مبتنی بر اتوماتای توزیع شده و بهترین الگوریتم هب آزمایش شده است. همچنین روش پیشنهادی در مقایسه با روشهای دیگر دارای پیچیدگی زمانی پایین‌تری می‌باشد و برخلاف تنها روش گزارش شده مبتنی بر اتوماتای یادگیر توزیع شده قابلیت استفاده برخط را نیز دارد.

کلمات کلیدی

داده‌کاوی استفاده از وب، اتوماتای یادگیر، اتوماتای یادگیر توزیع شده.

Web Usage Mining Using Distributed Learning Automata

Ali B. Hashemi* M. R. Meybodi†
Soft Computing Lab, Computer Engineering Department,
Amirkabir University of Technology, Tehran, Iran

Abstract

One of the most important issues in web mining is how to find out similarities between web pages. In this paper we propose a method based on distributed learning automata which take advantage of usage data to find out web pages similarities. The idea of the proposed method is that if different users request a couple of pages consistently together, then these pages are likely to correspond to the same information needs and hence can be considered similar. In the proposed method, a learning automaton is assigned to each page and is responsible for learning the similarities of that page to the other pages. It is shown that the proposed method performs better than the hebbian algorithm and the only learning automata based method reported in the literature. Furthermore, the proposed method needs lower computing time comparing to the other methods and unlike the only reported distributed learning automata based method it can be used online.

Keywords

Web usage mining, learning automata, distributed learning automata.

* آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، a_hashemi@aut.ac.ir
† استاد و عضو هیئت علمی، آزمایشگاه محاسبات نرم، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، mmeybodi@aut.ac.ir

۱ مقدمه

تشخیص شباهت بین اسناد یک مجموعه یکی از اهداف روشهای بازیابی اطلاعات می‌باشد. از اطلاعات در باره شباهت بین اسناد (صفحات وب) می‌توان برای ارائه اسناد مشابه به کاربران به منظور یافتن اطلاعات مورد نظر خود استفاده کرد. روش‌های متعددی برای تشخیص شباهت بین اسناد وجود دارد. قدیمی‌ترین روش، استفاده از نظر یک فرد خبره می‌باشد. این روش معمولا با دسته‌بندی اسناد بر اساس طبقه‌بندی موضوعی انجام می‌شود. استفاده از کلمات کلیدی در مقالات علمی یا صفحات وب برای یافتن شباهت بین اسناد نیز میتواند استفاده شود. استفاده از کلمات کلیدی دارای مشکلاتی مانند وجود کلمات مترادف[۱] (کلماتی با ظاهر متفاوت ولی معنای یکسان)، کلمات متشابه[۲] (کلماتی با ظاهر یکسان ولی در معنی متفاوت) میباشد. علاوه بر این زمانیکه که موضوعی برای کاربر با موضوع جدیدی روبرو میشود، پیدا کردن کلمات کلیدی مناسب کاری مشکل میباشد و تنها توسط افرادی که به زمینه سند آشنایی دارند می‌تواند استخراج شود. این مشکل در مورد اسناد الکترونیکی مانند صفحات وب، با استفاده از روشهای بازیابی اطلاعات تا اندازه زیادی کاهش یافته است. بعنوان مثال با استخراج کلمات اصلی یک متن، کلمات کلیدی آن را مشخص می‌کنند [7]. البته این روشها برای اسناد غیر متنی مانند تصاویر، فیلمها و اسناد صوتی کمتر استفاده شده است. از دیگر روشهای سنتی برای تعیین ارتباط اسناد مانند مقالات علمی با یکدیگر استفاده از اطلاعات در باره مراجع هر مقاله و بررسی ارتباط آنها از لحاظ کتابشناسی می‌باشد [6].

اکثر تحقیقات انجام شده در زمینه داده‌کاوی بر اساس تحلیل محتوای اسناد (داده‌کاوی محتوا[۳]) و یا ساختار گراف ارتباط اسناد (داده‌کاوی ساختار[۴]) بوده است. علاوه بر اطلاعات بدست آمده از این دو روش، می‌توان از اطلاعات در باره رفتار کاربران (با استفاده از فایلهای ثبت وقایع[۵] در سرویس‌دهنده‌های وب یا برنامه‌های در سمت کاربر) برای تعیین ارتباط بین اسناد [3]، پیشنهاد صفحات[19][20][21]، تغییر ساختار سایت وب [13]، شخصی کردن سرویس‌هایی مانند وب [14][15][16]، بهینه‌سازی موتورهای جستجو [17] استفاده کرد. در [18] کاربردهای اطلاعات استفاده از سیستم بطور مفصل ارائه شده است.

در تعدادی از روشهای گزارش شده مانند روش گزارش شده در [20] سیستم به صفحات وب با توجه به بازخوردهای ارائه شده توسط کاربران امتیازاتی میدهد که این امتیازات برای پیشنهاد صفحات به کاربر مورد استفاده قرار میگیرد. استفاده از بازخورد کاربران موجب ایجاد وظیفه‌ای ناخواسته برای کاربران شده و باعث نارضایتی آنها میشود. در بعضی از روشها از اطلاعات ثبت شده مشخص برای هر کاربر استفاده میشود. بعنوان مثال amazon.com بعنوان یک سایت فروش الکترونیکی بر روی وب، با استفاده از اطلاعات در باره خرید کاربران، ممکن است به کاربری که می‌خواهد جنس a را بخرد، پیشنهاد خرید جنس b را نیز بدهد چرا که کاربرانی که جنس a را خریده‌اند، معمولا جنس b را نیز خریده‌اند. هرچند، استفاده از اینگونه روشها به دلایلی مانند مسائل مرتبط با حریم شخصی کاربران یا محدودیتهای سرویس‌دهنده‌های وب امکان‌پذیر نمی‌باشد. بهمین دلیل معمولا در روشهای داده‌کاوی اطلاعات استفاده از وب از فایلهای ثبت وقایع در سرویس‌دهنده‌های وب (که تنها اطلاعات درخواستهای کاربران را در بر دارند) استفاده می‌شود [22][23]. از آنجاییکه استفاده از اطلاعات وب بصورت ناشناس صورت می‌گیرد، استفاده از چنین روشی تنها برای سایتهای محدودی امکان‌پذیر است. بهمین علت معمولا در چنین سیستم‌هایی از فایلهای ثبت وقایع در سرویس‌دهنده‌های وب و بدون دسترسی به اطلاعات شخصی هر کاربر استفاده می‌شود[22][23].

در [3] رویکرد جدیدی برای مساله *داده‌کاوی اطلاعات استفاده از وب*[۶] ارائه شده است. ایده این روش بر این اساس است که اگر دو سند به یک نیاز اطلاعاتی پاسخ دهند، آنگاه آن دو سند مشابه می‌باشند. در این روش فرض بر این است که کاربران از محتویات سندی که می‌خواهند آنرا در گام بعدی خود انتخاب کنند آگاهی نسبی دارند و بر اساس نیاز اطلاعاتی خود سند بعدی را انتخاب می‌کنند و حرکت کاربران در بین اسناد اتفاقی نیست. در واقع کاربر با استفاده از اطلاعات خود ارتباطی مجازی بین اسناد ایجاد کرده و آنها را مشاهده می‌کند. این ارتباط لزوما منطبق بر ارتباطات قابل مشاهده اسناد (مانند ارتباط اسناد بر اساس کلمات کلیدی تعریف شده یا ارتباطات کتابشناسی) نمی‌باشد بلکه می‌تواند برگرفته از مدل ذهنی کاربر باشد. از آنجاییکه فرض شده است که کاربر اطلاعات کافی در مورد اسناد مشاهده شده دارند، بنابراین انتظار می‌رود که اسناد مشابه در یک موضوع با یکدیگر مورد استفاده قرار گیرند. در این روش، با تحلیل داده‌های استفاده، بدون تلاش مضاعف کاربر یا افراد خبره (مانند کتابداران)، اطلاعات با ارزشی بدست می‌آید. در روش فوق ارتباطات بین اسناد با استفاده از روشی مانند قانون هب [32] اصلاح می‌گردد. به این صورت که با حرکت کاربر از سند i به سند j، تنها اتصال بین این دو سند ($a(i, j)$) تقویت می‌شود. که تقویت اتصال دو سند i و j متناظر با افزایش میزان شباهت این دو سند در نظر گرفته شده است. در نسخه توسعه یافته این الگوریتم، با حرکت کاربر از سند i به سند j، نه تنها اتصال بین این دو سند تقویت می‌شود، بلکه با در نظر گرفتن رابطه تراگذری، اتصال سند i به سندهای دیگری که کاربر بعد از مشاهده سند j در ادامه مسیر خود مشاهده می‌کند، با در نظر گرفتن یک ضریب کاهش(b)، تقویت می‌گردد.

با استفاده از ایده مطرح شده در پاراگراف قبلی، در [۱] یک روش خودسازمانده مبتنی بر اتوماتای یادگیر توزیع‌شده برای تعیین شباهت اسناد در یک کتابخانه دیجیتال ارائه شده است. در این روش یک اتوماتای یادگیر توزیع‌شده متناظر با گراف ارتباطات اسناد کتابخانه

دیجیتال در نظر گرفته می‌شود. بصورتی که هر اتوماتای یادگیر در اتوماتای یادگیر توزیع شده دارای تعدادی محدودی اقدام می‌باشد و هر اقدام متناظر با یک سند در مجموعه اسناد است. در این روش تنها اسنادی که در مسیر مستقیم حرکت کاربر از سند آغازین تا آخرین سند مشاهده شده قرار دارند، مشابه در نظر گرفته می‌شوند. بر این اساس پس از خروج هر کاربر از سیستم، با بررسی مسیر حرکت او، به اقدامهای اتوماتای یادگیر متناظر با اسنادی که در مسیر حرکت کاربر از نخستین صفحه تا آخرین صفحه قرار داشته‌اند پاداش و اقدامهای متناظر با اسنادی که قسمتی از یک دور هستند، جریمه می‌شوند.

در این مقاله یک الگوریتم جدید مبتنی بر اتوماتای یادگیر توزیع‌شده که از "اطلاعات استفاده از وب" استفاده می‌کند به منظور تشخیص شباهت صفحات وب پیشنهاد می‌گردد. الگوریتم پیشنهادی برخلاف تنها الگوریتم گزارش شده مبتنی بر اتوماتای یادگیر توزیع شده[۱] همزمان با حرکت کاربر از یک سند به سند دیگر، میزان شباهت بین اسناد را محاسبه می‌کند. این ویژگی موجب می‌شود که بتوان از این الگوریتم بصورت برخط استفاده کرد. علاوه بر این بعلت اینکه در الگوریتم پیشنهادی فرایند محاسبه دور مانند [۱] وجود ندارند، الگوریتم پیشنهادی در مقایسه با الگوریتم ارائه شده در [۱] دارای سربار محاسباتی کمتری می‌باشد.

ویژگی دیگر الگوریتم پیشنهادی این است که بدون استفاده از هیچگونه اطلاعاتی در باره محتوای اسناد و صرفا با استفاده بر الگوی رفتار کاربران میزان شباهت اسناد با یکدیگر را محاسبه می‌کند. استفاده از این روش میتواند بویژه در مواقعی که اسناد مورد جستجو را نتوان براحتی بصورت کلمات کلیدی مطرح کرد (مانند اسناد چندرسانه‌ای)، مفید باشد. به این صورت که کاربر می‌تواند با استفاده از موضوعات مشابه با موضوع مورد نظر خود و استفاده از اطلاعات سایر کاربرانی که در زمینه‌های مشابه بدنبال اطلاعات بودند جستجوی خود را برای رسیدن به مطالب و صفحات مورد نظر خود استفاده کند. الگوریتم پیشنهادی بر خلاف تنها الگوریتم مبتنی بر اتوماتای یادگیر توزیع شده بدلیل استفاده از اتوماتاهای یادگیر با تعداد اقدام متغیر سریع‌تر همگرا می‌شود. برای بررسی کارایی الگوریتم پیشنهادی و همچنین مقایسه آن با سایر الگوریتمها از مدل حرکت کاربران در صفحات وب که در [5] ارائه شده است و مدل کاملتری در مقایسه با مدل استفاده شده در [۱] می‌باشد، استفاده می‌شود. نتایج شبیه سازیها نشان داده است که روش پیشنهادی در مقایسه با روش هب و تنها روش گزارش شده مبتنی بر اتوماتای توزیع شده [۱] در تشخیص شباهت صفحات از کارایی بالاتری برخوردار است. همچنین روش پیشنهادی دارای پیچیدگی زمانی پایینتری میباشد و برخلاف تنها روش گزارش شده مبتنی بر اتوماتای توزیع شده قابلیت استفاده برخط[۷] را نیز دارد.

ساختار ادامه مقاله بدین صورت سازماندهی شده است. در بخش ۲ اتوماتای یادگیر و اتوماتای یادگیر توزیع‌شده به اختصار معرفی می‌شوند. در بخش ۳ الگوریتم پیشنهادی ارایه می‌گردد. در بخش ۴ پس از معرفی مدل استفاده شده برای شبیه‌سازی، نتایج شبیه‌سازی ارائه و بررسی می‌گردد. بخش ۵ نتیجه‌گیری می‌باشد.

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. 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[۸] مقادیر c_i ها ثابت هستند، حال آنکه در یک محیط غیر ایستا[۹] 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.

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

که a پارامتر پاداش و b پارامتر جریمه می‌باشد. اگر a و b با هم برابر باشند، الگوریتم L_{R-P} [10]، اگر b از a خیلی کوچکتر باشد، الگوریتم $L_{R\varepsilon P}$ [11] و اگر b صفر باشد، الگوریتم L_{R-I} [12] نام دارد [10].

اتوماتای یادگیری که در بالا معرفی شد، دارای تعداد اقدامهای ثابتی می‌باشد. در بعضی از کاربردها به اتوماتای یادگیر با تعداد اقدام متغیر[13] نیاز می‌باشد [12]. 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 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) بروز می‌کند. سپس اتوماتای یادگیر بردار احتمال اقدامهای خود $p(n)$ را با استفاده از بردار $\hat{p}(n+1)$ و طبق رابطه (۵) بروز می‌کند.

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

$$\hat{p}_i(n) = prob[\alpha(n) = \alpha_i \mid \alpha_i \in V(n)] = \frac{p_i(n)}{K(n)} \qquad (۲)$$

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

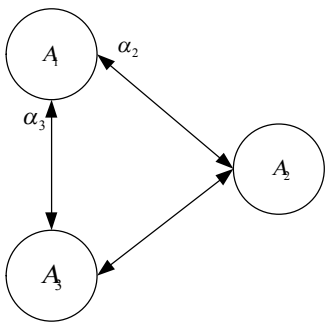
$$\hat{p}_i(n+1) = \hat{p}_i(n) + a.(1-\hat{p}_i(n))$$
$$\hat{p}_j(n+1) = \hat{p}_j(n) - a.\hat{p}_i(n) \qquad \forall j \ \ j \neq i \qquad (۳)$$

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

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

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 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. برای کسب اطلاعات بیشتر در باره اتوماتای یادگیر توزیع شده و کاربردهای آن میتوان به [24-31] مراجعه نمود.



شکل ۲. اتوماتای یادگیر توزیع‌شده

۳ الگوریتم پیشنهادی

در این بخش روشی مبتنی بر اتوماتای یادگیر توزیع شده که از اطلاعات درباره استفاده از وب استفاده می‌کند به منظور تشخیص شباهت صفحات وب پیشنهاد می‌گردد. در این روش برای تشخیص شباهت اسناد از رفتار کاربران استفاده می‌شود. از الگوریتم پیشنهادی

می‌توان بعنوان گام اولیه فرایند دسته‌بندی اسناد (قبل از اعمال نظر یک فرد خبره) یا در یک سیستم پیشنهاد دهنده استفاده نمود. در شرایطی که کلمات کلیدی اسناد مشخص نبوده یا محتوای آنها براحتی قابل استخراج[۱۴] نباشند(مانند اسناد چند رسانه‌ای) استفاده از این روش بیشتر مورد توجه می‌باشد.

در الگوریتم پیشنهادی برای تعیین شباهت بین صفحات یک سایت (اسناد) در یک مجموعه با n صفحه، از یک اتوماتای یادگیر توزیع‌شده با n *اتوماتای یادگیر با تعداد اقدامهای متغیر* [12] که هر یک n-1 اقدام دارند، استفاده می‌شود. برای هر اتوماتای یادگیر در هر زمان تنها یک زیرمجموعه از اقدامهایش فعال و می‌تواند قابل استفاده باشد [12]. تعداد اقدامهای اتوماتای یادگیر متناظر با هر صفحه مانند i برابر است با تعداد صفحاتی که کاربر می‌تواند بعد از آن صفحه مشاهده کند. هنگامی که یک کاربر پس از مشاهده صفحه i، صفحه j را مشاهده می‌کند، با فرض اینکه وجود شباهتی بین محتوای دو صفحه i و j موجب این انتخاب کاربر شده است، اقدام jام در اتوماتای iام پاداش می‌گیرد. نحوه فعالیت این الگوریتم بصورت زیر می‌باشد.

Procedure DLA_usage_minig
variables:
DLA: Distributed Learning Automata which contains n LA having n-1 actions.
user_log: Array of [Number of Users][Users Path]
/ user log, documents viewed by each user. each row contains trace of a user. */*
begin
for *all users* **do**
doc_id = 1;
while *user is browsing the site*
cur_doc = user_log[user_id][doc_id];
doc_id = doc_id + 1;

/ find next document (α) visited by current user */*
α = user_log[user_id][doc_id];
if *action α of DLA(cur_doc) is disabled* **then**
enable action α of DLA (cur_doc);
end
set *β =0; // rewarding action cur_doc*
reward action α of DLA(cur_doc) according to eq. (۳)
end
end
end

شکل ۳. شبه‌کد الگوریتم پیشنهادی

در ابتدای الگوریتم، تمامی اقدام‌های اتوماتاهای یادگیر در اتوماتای یادگیر توزیع‌شده غیر فعال می‌باشند. با حرکت یک کاربر از سند i که در حال مشاهده می‌باشد، به سند j، اقدام متناظر با آن سند (اقدام j) در اتوماتای یادگیر i فعال می‌شود. در این حالت اتوماتای یادگیر i به اقدام j خود پاداش می‌دهد. آنگاه اتوماتای یادگیر j در اتوماتای یادگیر توزیع‌شده فعال می‌شود و مراحل فوق تا پایان حرکت کاربر در مجموعه صفحات ادامه می‌یابد. در هر زمان، شباهت دو سند i و j برابر با احتمال انتخاب اقدام j در اتوماتای i است. در صورتیکه اقدام مورد نظر غیرفعال باشد، شباهت دو سند صفر در نظر گرفته می‌شود. شبه‌کد الگوریتم پیشنهادی در شکل ۳ نشان داده شده است.

۴ نتایج شبیه‌سازیها

در این بخش نتایج بدست آمده از شبیه‌سازی الگوریتم پیشنهادی ارائه می‌شود.

۴,۱ مدل شبیه‌سازی

برای شبیه‌سازی الگوریتم پیشنهادی و مقایسه آن با سایر روشها از مدل معرفی شده در [5] برای نشان دادن ساختار صفحات وب و چگونگی استفاده کاربران، استفاده شده است. اعتبار این مدل توسط Lui و همکاران[5] با استفاده از اطلاعات استفاده از وب چندین سایت وب بزرگ مانند مایکروسافت، تایید شده است. بر این اساس، در این مقاله مطابق با مدل رفتار کاربران، پروفایل علاقه کاربران بصورت توزیع قانون-توانی[۱۵] و توزیع محتوای صفحات وب بصورت توزیع نرمال در نظر گرفته شده است. سایر پارامترهای استفاده شده در مدل [5] برای شبیه‌سازیهای انجام شده در این مقاله در جدول ۱ نشان داده شده است.

حد آستانه ایجاد اتصال	**۰/۷**
تعداد کاربران	۱۰۰۰۰
تعداد اسناد	۲۶
تعداد موضوعها	۴
T_c مقدار ثابت سند اولیه (صفحه اولیه سایت) در موضوعات مختلف	۰/۲
ΔM_t^c ضریب ثابت کاهش اشتیاق کاربر	-
ΔM_t^v ضریب متغیر کاهش اشتیاق کاربر	-
α_u پارامتر توزیع قانون-توانی توزیع احتمال علایق کاربران	۱
ϕ ضریب پاداش دریافتی از مشاهده یک سند	۱/۲
λ ضریب جذب اطلاعات از یک سند توسط یک کاربر	۰/۵
μ_m میانگین توزیع نرمال ΔM_t^v	۵/۹۷
σ_m واریانس توزیع نرمال ΔM_t^v	۰/۲۵
μ_t میانگین توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع خاص	-
α_p پارامتر توزیع قانون-توانی توزیع احتمال وزنهای مطالب برای هر سند	۳
σ_t واریانس توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع خاص	۰/۲۵

ضریب کاهش علاقه کاربر θ	۱
حداقل اشتیاق کاربر برای ادامه جستجو	۰/۲

جدول ۱: پارامترهای شبیه‌سازیها

۴،۲ شاخص ارزیابی

در این مقاله معیار شباهت دو سند عکس فاصله این دو سند تعریف شده است. فاصله دو سند i و j (درایه d_{ij} ماتریس D)، فاصله اقلیدسی بردارهای محتوای آنها(C_i و C_j)، طبق رابطه (۷) محاسبه می‌شود. در الگوریتم پیشنهادی در صورت فعال بودن اقدام j در اتوماتای i، شباهت آنها (d'_{ij}) برابر با p_j^i (احتمال اقدام j در اتوماتای i) و در غیر اینصورت شباهت دو سند i و j صفر قرار داده می‌شود (رابطه (۸)). شاخص ارزیابی کارایی الگوریتمهای شبیه‌سازی شده، کوریلیشن بردار فاصله دو سند i و j (d_{ij}) و بردار شباهت آنها (d'_{ij}) می‌باشد(رابطه (۹)). از آنجاییکه شباهت دو سند عکس فاصله آنها تعریف شده است، در صورت تشخیص درست، مقدار این کوریلیشن منفی می باشد. هر چه این مقدار به ۱- نزدیکتر باشد، الگوریتم در تشخیص شباهت اسناد بهتر عمل کرده است.

$$C_n = [cw_n^1 \quad cw_n^2 \quad \cdots \quad cw_n^M] \tag{۶}$$

$$d_{ij} = \sqrt{\sum_{k=1}^{M} \left(cw_i^k - cw_j^k\right)^2} \tag{۷}$$

$$d'_{ij} = \begin{cases} p_j^i & \text{if } \alpha_j^i \text{ is enabled} \\ 0 & otherwise \end{cases} \tag{۸}$$

$$Correlation(D,D') = \frac{Cov(D,D')}{\sigma_D \sigma_{D'}} = \frac{\sum DD' - (\sum D \sum D')/N}{\sqrt{(\sum D^2 - (\sum D)^2/N)(\sum D'^2 - (\sum D')^2/N)}}$$
$$D = \{d_{ij} \mid i,j = 1,2,\cdots,n, \quad i \neq j\}$$
$$D' = \{d'_{ij} \mid i,j = 1,2,\cdots,n, \quad i \neq j\} \tag{۹}$$

۴،۳ نتایج شبیه‌سازی

برای ارزیابی الگوریتم پیشنهادی، نتایج شبیه‌سازی این الگوریتم با روش هب ساده و تعمیم‌یافته [3] و تنها روش گزارش شده مبتنی بر اتوماتای توزیع شده [۱] مقایسه می‌گردد. همچنین برای نشان دادن یک کران پایین برای کارایی این الگوریتمها، از یک روش آماری ساده نیز استفاده می‌شود. 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(s(i,j)) به تعداد دفعاتی که کاربران از سند i به هر سند دیگری مانند k حرکت نموده‌اند، محاسبه می‌شود(رابطه (۱۰)).

$$similarity(i,j) = \frac{s(i,j)}{\sum_{k=1}^{n} s(i,k)} \tag{۱۰}$$

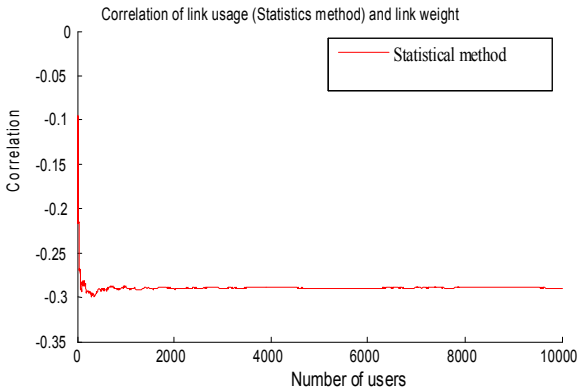
برای شبیه‌سازی الگوریتم هب توسعه‌یافته ضریب کاهش بر اساس یکی از روابط (۱۱)، (۱۲) یا (۱۳) در نظر گرفته می‌شود.

$$compute\_dec\_factor(i,k) = 1 \tag{۱۱}$$

$$compute\_dec\_factor(i,k) = 1/(steps\ between\ i\ and\ k\ in\ the\ current\ path) \tag{۱۲}$$

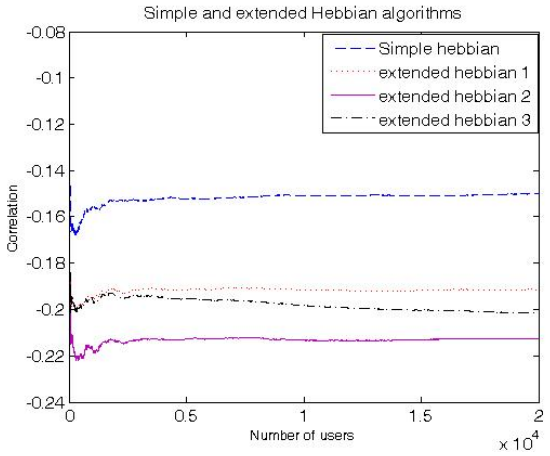
$$compute\_dec\_factor(i,k) = \sum_{m \geq i, n \leq k} a(m,n) \tag{۱۳}$$

در شکل ۴ کارایی روش آماری نشان داده شده است. از آنجاییکه میزان استفاده از اتصال بین دو سند i و j با فاصله بردار محتوای آنها نسبت عکس دارد، انتظار می‌رود که کوریلیشن این دو مقدار منفی گردد. همانطور که در شکل ۴ نیز مشاهده می‌شود، این مقدار منفی می‌باشد.

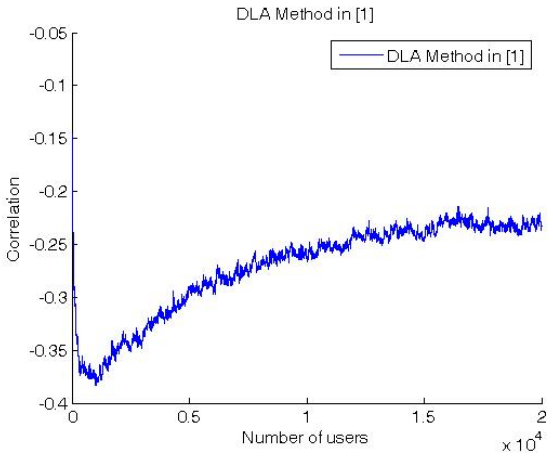


شکل ۴. کارایی روش آماری ساده

کوریلیشن بدست آمده با استفاده از الگوریتم هب ساده و تعمیم‌یافته (بترتیب با سه تابع (۱۱)، (۱۲) و (۱۳)) در شکل ۵ نشان داده شده است. در روش هب ساده و توسعه یافته مقدار a_{ij} با حرکت یک کاربر بر روی لینک (i,j) افزایش می‌یابد و بنابراین انتظار می‌رود که زمانیکه فاصله اقلیدسی بین دو گره i و j کم باشد (کوچک بودن مقدار d_{ij}) مقدار a_{ij} بزرگ شود. با فرض اینکه کاربران اتصالات با وزن (فاصله) کمتر را برای حرکت بعدی خود انتخاب می‌کنند انتظار می‌رود که کوریلیشن مقادیر a_{ij} با مقادیر واقعی وزنهای (فواصل) گره‌ها d_{ij} منفی باشد. همانطور که در شکل ۵ مشاهده می‌شود این انتظار بر آورده شده و مقدار کوریلیشن ماتریس A و D مقداری منفی می‌باشد. کوریلیشن ماتریس شباهت الگوریتم معرفی شده در [۱] با ماتریس شباهت اسناد در شکل ۶ نشان داده شده است. همانطور که مشاهده می‌شود، کوریلیشن بدست آمده از این الگوریتم در مقایسه با کوریلیشن بدست آمده برای الگوریتم هب کمتر است.



شکل ۵. کوریلیشن برای الگوریتم هب ساده و تعمیم‌یافته

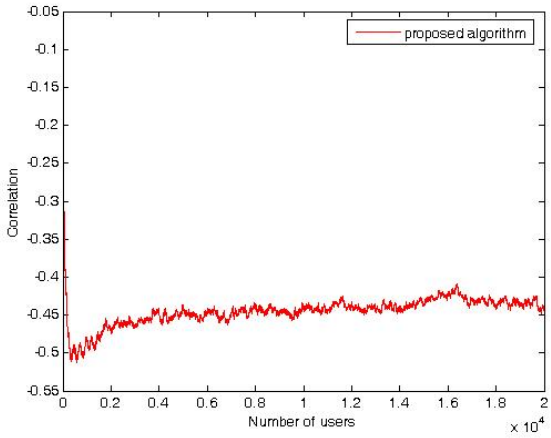


شکل ۶. کوریلیشن برای الگوریتم ارائه شده در [۱]

همانطور که در شکل ۷ نشان داده شده است کوریلیشن بدست آمده با استفاده از روش پیشنهادی بسیار بهتر از کوریلیشن بدست آمده برای روش آماری ساده (شکل ۴)، الگوریتم هب (شکل ۵) و الگوریتم معرفی شده در [۱] می‌باشد. مشاهده می‌شود که در الگوریتم پیشنهادی پس از ورود کمتر از ۵۰۰ کاربر مقدار کوریلیشن ماتریس شباهت بدست آمده به ماتریس شباهت واقعی بسیار نزدیک می‌شود و بعد از ورود ۲۰۰۰ کاربر، این مقدار تقریبا ثابت باقی می‌ماند که در مقایسه با الگوریتم معرفی شده در [۱] بسیار کمتر است.

۵ نتیجه‌گیری

در این مقاله روشی مبتنی بر اتوماتای یادگیر توزیع شده که از اطلاعات در باره استفاده از وب استفاده می‌کند به منظور تشخیص شباهت صفحات وب پیشنهاد گردید. نتایج شبیه سازیها نشان داد که روش پیشنهادی در مقایسه با روش هب و تنها روش گزارش شده مبتنی بر اتوماتای توزیع شده در تشخیص شباهت صفحات از کارایی بالاتری برخوردار است. بصورتیکه کوریلیشن ماتریس شباهت بدست آمده با ماتریس شباهت اسناد، در الگوریتم پیشنهادی بترتیب ۰,۱ و ۰,۲ بیشتر از این مقدار در الگوریتم معرفی شده در [۱] و بهترین الگوریتم هب آزمایش شده است. همچنین از آنجاییکه بر خلاف تنها روش گزارش شده مبتنی بر اتوماتای توزیع شده [۱]، در روش پیشنهادی نیازی به محاسبه دور در مسیر کاربران وجود ندارد، روش پیشنهادی دارای پیچیدگی زمانی پایین تری می‌باشد.



شکل ۷. کوریلیشن برای الگوریتم پیشنهادی

مراجع

[۱] سعید ساعتی و محمدرضا میبدی، "یک مدل خودسازمانده برای ساختار اطلاعاتی اسناد با استفاده از اتوماتای یادگیر توزیع‌شده،" مجموعه مقالات دومین کنفرانس بین‌المللی فناوری اطلاعات و دانش، تهران، ایران، ۱۳۸۴.

[2] R. Colley, Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data, Ph.D. Dissertation, University of Minnesota, May 2000.

[3] F. Heylighen and J. Bollen, "Hebbian Algorithms for a Digital Library Recommendation System," Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'02), 2002, pp. 439-446.

[4] F. Heylighen, "Mining Associative Meanings from the Web: from Word Disambiguation to the Global Brain," Proceedings of the International Colloquium: Trends in Special Language and Language Technology, 1995, pp. 15-44.

[5] J. Liu, S. Zhang, and J. Yang, "Characterizing Web Usage Regularities with Information Foraging Agents," IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 4, April 2004, pp. 566-584.

[6] José Manuel Barrueco Cruz and Thomas Krichel, "Automated Extraction of Citation Data in a Distributed Digital Library," Proceedings of the 2nd International Workshop on New Developments in Digital Libraries, 2002, pp 51-62.

[7] Junichiro Mori, Yutaka Matsuo, Mitsuru Ishizuka, and Boi Faltings, "Keyword Extraction from the Web for FOAF Metadata," Proceeding of 1st International Workshop on Friend of a Friend, Social Networking and the Semantic Web, Galway, Ireland, 2004, pp 1-8.

[8] J. Mori, Y. Matsuo, M. Ishizuka, and B. Faltings, "Keyword Extraction from the Web for Creation of Person Metadata," in Poster Abstracts 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan, 2004, pp. 45-46.

[9] J. Mori, Y. Matsuo, M. Ishizuka, and B. Faltings, "Keyword Extraction from the Web for Personal Metadata Annotation," in 4th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2004) in conjunction with 3rd Int'l Semantic Web Conference (ISWC2004)), Hiroshima, Japan, 2004, pp. 51-60.

[10] K.S. Narendra and M.A.L. Thathachar, Learning Automata: An Introduction, Prentice Hall, 1989.

[11] Robert Korfhage, Information Storage and Retrieval, John Wiley and Sons, 1997.

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

[13] Mike Perkowitz and Oren Etzioni, "Adaptive Web Sites," Communications of ACM, vol. 43, no. 8, 2000, pp. 152-158.

[14] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic Personalization Based on Web Usage Mining," Communications of the ACM, vol. 43, no. 8, 2000, pp. 142–151.

[15] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization," Data Mining and Knowledge Discovery, vol. 6, no. 1, 2002, pp. 61-82.

[16] Pierrakos, G. Paliouras, C. Papatheodorou, and C.D. Spyropoulos, "Web Usage Mining as a Tool for Personalization: A Survey," User Modeling and User-Adapted Interaction, vol. 13, no. 4, 2003, pp. 311-372.

[17] T. Joachims, "Optimizing Search Engines Using Click Through Data," Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02), 2002, pp. 133–142.

[18] J. Srivastava, R. Cooley, M. Deshpande, and P.N. Tan, "Web Usage Mining: Discovery and Applications of Usage Ppatterns from Web Data," SIGKDD explorations, vol. 1, no. 2, 2000, pp. 12–23.

[19] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell, "WebWatcher: A Learning Apprentice for the World Wide Web," Proceedings of AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, AAAI Press, 1995, pp 6–12.

[20] M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & Webert: Identifying Interesting Web Sites," Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96), AAAI Press, 1996, pp. 54–61.

[21] M. Balabanovic and Y. Shoham, "Learning Information Retrieval Agents: Experiments with Automated Web Browsing," Proceedings of AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, AAAI Press, 1995, pp. 13-18.

[22] D. Mladenis, Personal WebWatcher: Implementation and Design. Technical Report IJS-DP-7472, Department of Intelligent Systems, Joz, es Stefan Institute, 1996.

[23] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic Personalization Based on Web Usage Mining," Communications of the ACM, vol. 43, no. 8, 2000, pp. 142-151.

[24] H. Beigy and M. R. Meybodi, "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

[25] M. R. Meybodi and H. Beigy, "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

[26] M. R. Meybodi and H. Beigy, "Solving Stochastic Shortest Path Problem Using Monte Carlo Sampling Method: A Distributed Learning Automata Approach", Springer-Verlag Lecture Notes in Advances in Soft Computing: Neural Networks and Soft Computing, pp. 626-632, 2003.(ISBN: 3-7908-0005-8)

[27] H. Beigy and M. R. Meybodi, "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

[28] M. Alipour and M. R. Meybodi, "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

[29] M. Alipour and M. R. Meybodi, "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

[30] M. Alipour and 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, pp. 673-678, Jan. 24-26, 2006

[31] M. Alipour and M. R. Meybodi, "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.

[32] D. O. Hebb, The organization of behavior: A neuropsychological theory, Wiley-Interscience, New York,1949.

زیرنویس‌ها

[1] Synonym
[2] Homonym
[3] Content mining
[4] Structure mining
[5] Log files
[6] Web usage mining
[7] Online
[8] Stationary
[9] Non-Stationary
[10] Linear Reward-Penalty
[11] Linear Reward epsilon Penalty
[12] Linear Reward Inaction
[13] Learning automata with changing number of actions
[14] Minable
[15] Power-law