# روشی مبتنی بر اتوماتای یادگیر توزیع شده برای تعیین ساختار اسناد وب

بابک اناری[1]، محمد رضا میبدی[2]

## چکیده

**در این مقاله یک روش جدید و مبتنی براتوماتای یادگیر توزیع شده برای کشف ساختار ارتباطی بین اسناد وب پیشنهاد می‌گردد. تعیین ساختار اسناد وب باعث پیدا کردن اسناد مشابه به هم شده و می‌توان بوسیله آن به خوشه‌بندی و رتبه‌بندی این اسناد پرداخت. در روش پیشنهادی مانند روشهای گزارش شده مبتنی بر اتوماتای یادگیر توزیع شده به هر سند وب یک اتوماتای یادگیر اختصاص داده می‌شود که وظیفه آن یادگیری ارتباطات آن سند با اسناد دیگر می‌باشد. الگوریتم پیشنهادی نسبت به تنها روش گزارش شده مبتنی بر اتوماتای یادگیر توزیع شده دارای دو مزیت است: بالا بودن میزان کارایی و عدم نیاز به تنظیم پارامترهای اتوماتای یادگیر توزیع شده در صورت گسترش تعداد اسناد وب. کارایی الگوریتم پیشنهادی از طریق مقایسه با سه روش AntWeb، Bollen و روش DLA-FA مورد ارزیابی قرار خواهد گرفت.**

**کلمات کلیدی**

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

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

Babak anari, Mohammad Reza Meybodi

## Abstract

In this article a new method which is based on Distributed Learning Automata (DLA) for discovering the relational structure of web documents is proposed. Determining web documents structure leads to finding the similar documents and we can have a clustering and a ranking of these documents. Like other reported methods based on the DLA, in this method a learning automata is devoted to each document too, Its function is learning the relationships of that document with the other documents. The advantages of proposed algorithm toward only reported algorithm (DLA-FA) which is based on DLA having high performance and no needing to set the DLA parameters by developing the number of web documents. The comparison of proposed algorithm with the three methods of AntWeb, Bollen and DLA-FA will be considered.

**Keywords**

Web Usage Mining, Learning Automata, Distributed Learning Automata

[1] دانشکده مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد اراک، anari322@yahoo.com

[2] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه امیر کبیر، meybodi@ce.aut.ac.ir

## 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 می‌باشد. استخراج اطلاعات از اسناد وب بوسیله تکنیکهای داده‌کاوی را، کاوش وب[1] می‌گویند. 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 می‌باشد. 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 داده‌هایی است که از تعامل کاربران در هنگام استفاده از وب بدست می‌آید. کاوش الگوهای حرکتی ذخیره شده کاربران در لاگ فایلها، کاوش استفاده از وب محسوب می‌شود.

بیشتر تکنیکهای موجود برای کشف ساختار اسناد وب از اطلاعات موجود در لاگ فایلها استفاده می‌کنند. لاگ فایلها، فایلهایی هستند که مجموعه درخواستهای کاربران به صفحات وب را در خود ذخیره می‌کنند[3]. روشهای موجود برای کشف ساختار اسناد وب از طریق کاوش در لاگ فایلها یکسری اطلاعات آماری را از آنها استخراج می‌کنند که می‌توان 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[2]، استفاده از قوانین یادگیری، مثل استفاده از قانون یادگیری هب در روش بولن[4] ، استفاده از سیستم مورچه‌ها مثل روش AntWeb [13]، استفاده از اتوماتاهای یادگیر توزیع شده[3][1,11,12] ، استفاده از زنجیره مارکف[5] و استفاده از گرامرهای احتمالی ابرمتن[4][2] . روش گزارش شده مبتنی بر اتوماتای یادگیر توزیع شده که بعدا در این مقاله مفصلتر به آنها اشاره می‌شود دارای دو نقطه ضعف می‌باشد. 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 دوباره 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 قدیمی‌تر دارند، بدلیل اینکه از ماتریس ارتباطات اسناد استفاده می‌کنند برای استفاده در مجموعه‌های بزرگ و قابل گسترش مناسب نمی‌باشند. نقطه ضعف استفاده از زنجیره مارکف، نیاز به محاسبات بالا بواسطه محاسبه توان $n$ام ماتریس انتقال می‌باشد. هر چند با روشهای فشرده سازی می‌توان هزینه محاسبات را تا حدی کاهش داد[5]. نقطه ضعف استفاده از گرامرهای احتمالی ابرمتن نیز داشتن پیچیدگی$O(n)$ به واسطه استفاده از الگوریتم پیمایش درعمق می‌باشد. در این مقاله روشی مبتنی بر اتوماتای یادگیر توزیع شده که برای تعیین ساختار اسناد وب از اطلاعات موجود درلاگ فایلها استفاده می‌کند، پیشنهاد می‌گردد.

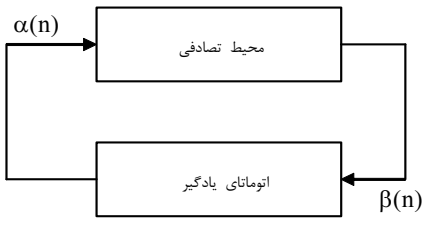
در این روش مانند روشهای گزارش شده مبتنی بر اتوماتای یادگیر توزیع شده به هر سند یک اتوماتای یادگیر اختصاص داده می‌شود که وظیفه‌اش یادگیری ارتباطات آن سند با اسناد دیگر می‌باشد. روش پیشنهادی که از مفهوم آنتروپی برای بروز رساندن بردار احتمالات اعمال اتوماتاهای یادگیر توزیع شده استفاده می‌نماید، ضمن داشتن کارایی در حد روشهای AntWeb و Bollen، پارامترهای یادگیری در اتوماتای یادگیر توزیع شده را با توجه به تعداد اسناد وب بصورت پویا تنظیم می‌کند. کارایی الگوریتم پیشنهادی از طریق مقایسه با سه روش AntWeb ، Bollen و روش DLA-FA مورد ارزیابی قرار می‌گیرد. ادامه مقاله بدین صورت سازماندهی شده است: در بخش ۲ اتوماتای یادگیر و اتوماتای یادگیر توزیع شده و روشهای مبتنی بر اتوماتای یادگیر توزیع شده به اختصار شرح داده می‌شوند. در بخش ۳ روش پیشنهادی و در بخش ۴ نتایج شبیه سازیها ارائه می‌گردد. بخش پایانی نتیجه‌گیری می‌باشد.

## ۲- اتوماتای یادگیرتوزیع شده و تعیین ساختار اطلاعاتی اسناد وب

در این بخش ابتدا به اتوماتای یادگیر و اتوماتای یادگیر توزیع شده بطور مختصر اشاره می‌شود و سپس روشهای پیشین تعیین ساختار در وب که مبتنی بر اتوماتای یادگیر توزیع شده می‌باشند، شرح داد می‌شود.

### ۲-۱- اتوماتای یادگیر[5]

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




**شکل ۱: ارتباط بین اتوماتای یادگیر و محیط**

**محیط**[6]: محیط را می‌توان توسط سه‌تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که درآن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه ورودیها، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه خروجیها

و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمالهای جریمه می‌باشد. هر گاه $\beta$ 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، $\beta(n)$ 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 $\alpha_i$ نتیجه نامطلوب داشته باشد می‌باشد. در محیط ایستا[7] مقادیر $c_i$ بدون تغییر می‌مانند، حال آنکه در محیط غیر ایستا[8] این مقادیر در طی زمان تغییر می‌کنند. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختارمتغیر تقسیم بندی می‌گردند. در ادامه به شرح مختصری درباره اتوماتای یادگیر با ساختار متغییر که در این مقاله از آنها استفاده شده است می پردازیم.

**اتوماتای یادگیر با ساختار متغیر**[9]: اتوماتای یادگیر با ساختار متغیر توسط ۴ تائی $\{\alpha, \beta, p, T\}$ نشان داده می‌شود که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \Lambda, \alpha_r\}$ مجموعه عملهای اتوماتا $\beta \equiv \{\beta_1, \beta_2, \Lambda, \beta_m\}$ مجموعه ورودیهای اتوماتا $p \equiv \{p_1, p_2, \Lambda, p_r\}$ بردار احتمال انتخاب هر یک از عملها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. در این نوع از اتوماتاها، اگر عمل $\alpha_i$ در مرحله nام انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می‌یابند و برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالها افزایش می‌یابند. در هر حال، تغییرات به‌گونه‌ای صورت می‌گیرد تا حاصل جمع $p_i(n)$ ها همواره مساوی یک باقی بماند. الگوریتم زیر یک نمونه از الگوریتمهای یادگیری خطی در اتوماتای با ساختار متغییر است.

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$
$$p_j(n+1) = (1-a)p_j(n) \quad j \neq i \quad \forall j \qquad (۱)$$

**الف- پاسخ مطلوب**

$$p_i(n+1) = (1-b)\,p_i(n)$$
$$p_j(n+1) = \frac{b}{r-1} + (1-b)\,p_j(n) \quad j \neq i \ \forall j \qquad (۲)$$

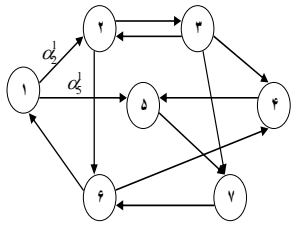
**ب-پاسخ نامطلوب**

در روابط فوق، a پارامتر پاداش و b پارامتر جریمه می‌باشد. با توجه به مقادیر a و b سه حالت را می توان در نظر گرفت. زمانیکه a و b با هم برابر باشند، الگوریتم را $L_{RP}$[10] می نامیم. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را $L_{R\varepsilon P}$[11] می نامیم. زمانیکه b مساوی صفر باشد، الگوریتم را $L_{RI}$[12] می نامیم. برای مطالعه بیشتر در باره اتوماتاهای یادگیر می توان به مراجع برای مطالعه بیشتر در باره اتوماتاهای یادگیر می‌توان به[6,8,9,10,14] مراجعه کرد.

## 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. دراین 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 یادگیراست، نشان داده می‌شود. وجود یال $(LA_i, LA_j)$ در این گراف بدین معناست که انتخاب عمل $\alpha_j^i$ توسط $LA_i$ باعث فعال شدن $LA_j$ می‌گردد. تعداد اعمال قابل انتخاب توسط $LA_k$ بصورت $p^k = \{p_1^k, p_2^k, ..., p_{r_k}^k\}$ نمایش داده شود. دراین مجموعه عدد $p_m^k$ نشان دهنده احتمال مربوط به عمل $\alpha_m^k$ است. انتخاب عمل $\alpha_m^k$ 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 و کاربرد های آن می‌توان به[15,16,17,18,19,20] مراجعه نمود.




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

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

## ۲-۳- تعیین ساختار اطلاعاتی اسناد بااستفاده‌از 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 صفحات‌وب، اسناد و کاربران استفاده کننده از آن نقش یک محیط تصادفی را برای اتوماتاهای یادگیر موجود درDLA 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.

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 براین است که با افزایش تعداد اسناد، تعداد اسنادی که یک سند به آن مرتبط می‌باشد ثابت باقی می‌ماند و بهمین 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 متغییر فرض شده است و به همین دلیل به هر سند یک اتوماتای یادگیر با تعداد اعمال متغییر تخصیص داده می‌شود. روش (DLA-FA) 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 می باشد. بعبارت دیگر بردار اعمال یک اتوماتای یادگیر می تواند بعنوان شناسه سند متناظر با آن اتوماتای یادگیر و بردار احتمالات میزان ارتباط این سند با دیگر سندها در مجموعه اسناد در نظر گفته شود. بنابراین برای هر سند $Doc_i$ یک اتوماتای یادگیر $LA_i$ در نظر می‌گیریم که تعداد عملهای آن تعداد ثابتی می باشد. 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 $\alpha_k^i = j$) 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 $\alpha_j^i$ قرار می‌گیرد و احتمال متناظر با این عمل برابر صفر قرار داده می‌شود. سپس احتمال عمل حذف شده بین احتمالهای اعمال توزیع می شود تا مجموع احتمالها 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 می باشد 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.

## 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. فرض‌کنید $p^k = \{p_1^k, p_2^k, ..., p_r^k\}$ بردار احتمال اتوماتای یادگیر $LA_k$ که به سند k تخصیص داده شده است باشد که $p_m^k$ احتمال انتخاب عمل $\alpha_m^k$ و r تعداد اسناد می‌باشد. اگر کاربر حرکت $D_k \to D_m$ را انجام دهد (از سند $D_k$ به سند $D_m$ حرکت کند) در این صورت اتوماتای یادگیر $LA_k$ بردار احتمال اعمال خود را طبق الگوریتم یادگیری زیر بروز می‌کند.

$$p_m^k(n+1) = p_m^k(n) + a_m^k[1 - p_m^k(n)] \qquad (۳)$$

$$p_j^k(n+1) = (1 - a_m^k)\, p_j^k(n) \quad j \neq m \;\forall j \qquad (۴)$$

$$a_m^k = \frac{E_m^k}{1 + E_m^k} \qquad (۵)$$

$$E_m^k = -(p_m^k \log p_m^k + (1 - p_m^k)\log(1 - p_m^k)) \qquad (۶)$$

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 $D_1 \to D_2$ 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 $D_1 \to D_3$ را انجام دهد در این صورت $a_3^1 = 0.3/(1+0.3) = 0.23$ و در نتیجه بردار ااحتمال اعمال

$LA_1$ به(0.154, 0.615, 0.231) تغییر پیدا می‌کند. الگوریتم پیشنهادی بصورت شکل۳ می باشد.

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

۳-۱-۱-۱- بردار احتمال اعمال اتوماتای یادگیر $LA_k$ را طبق الگوریتم یادگیری زیر بروز کن

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

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

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

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

**شکل۳: الگوریتم پیشنهادی**

## ۴- ارزیابی کارایی الگوریتم پیشنهادی و نتایج شبیه‌سازیها

برای انجام شبیه‌سازی از مدل ارایه شده در [7] برای تولید مجموعه اسناد و حرکات کاربران استفاده می‌شود. هر سند با بردار محتوای $C_n = \lfloor cw_n^1 \quad cw_n^2 \quad \Lambda \quad cw_n^M \rfloor$ 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[13] 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 در نظر گرفته‌ایم.

برای ارزیابی الگوریتم پیشنهادی از معیاری بنام کورولیشن استفاده خواهیم نمود. کورولیشن دو مجموعه داده مانند P,P' بصورت رابطه ۷ حساب می‌شودکه در آن N تعداد داده‌ها است. مجموعه P ساختار ایجاد شده توسط مدل شبیه‌سازی[7] و P' نیز ساختار بدست آمده توسط الگوریتم پیشنهادی را نشان می‌دهد.

$$Corr(P,P') = \frac{\sum PP' - (\sum P \sum P')/N}{\sqrt{(\sum P^2 - (\sum P)^2/N)(\sum P'^2 - (\sum P')^2/N)}} \quad (۷)$$

$$P = \{P_{ij} \mid i,j = 1,2,...,n, \quad i \neq j\} \quad (۸)$$

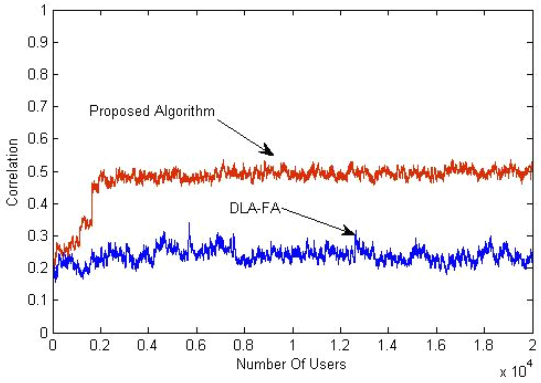
$$P' = \{P'_{ij} \mid i,j = 1,2,...,n, \quad i \neq j\} \quad (۹)$$

$$P_{ij} = \frac{(d_{ij})^{-1}}{\sum_{k=1}^{n} (d_{ik})^{-1}} \quad (۱۰)$$

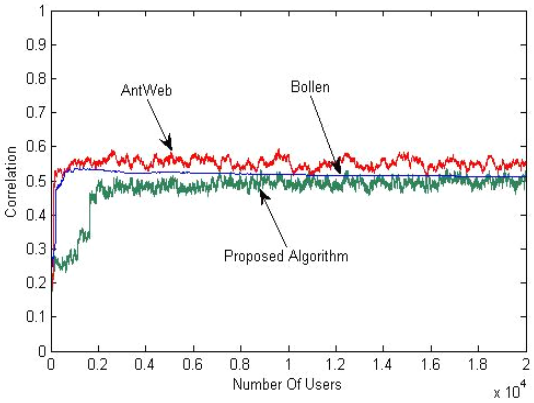
$$P'_{ij} = probabilit\,y\ of\ \ action\ \ j\ \ from\ DLA(i) \quad (۱۱)$$

$$d_{ij} = \sqrt{\sum_{k=1}^{M} \left(cw_i^k - cw_j^k\right)^2} \quad (۱۲)$$

در رابطه فوق $d_{ij}$ فاصله اقلیدسی بین دوسند i,j از یکدیگر در مدل شبیه‌سازی می‌باشد. شکل۴ مقایسه الگوریتم پیشنهادی با روش DLA-FA را نشان می‌دهد. در این شبیه سازی تعداد کاربران20,000، تعداد موضوعات مرتبط با هر سند5 و تعداد اسناد وب 30 در نظر گرفته شده است. مقایسه الگوریتم پیشنهادی با الگوریتمهایAntWeb,Bollen در شکل ۵ آمده است.




**شکل۴: مقایسه الگوریتم پیشنهادی با روش DLA-FA**




**شکل۵: مقایسه نتیجه حاصل از الگوریتم پیشنهادی با دو روش AntWeb,Bollen**

همانطوریکه از شکلهای فوق مشخص است، الگوریتم پیشنهادی ضمن داشتن کورولیشن در حد روشهای AntWeb,Bollen از روش DLA-FA بسیار بهتر بوده، ضمن اینکه نیاز به تنظیم پارامتر یادگیری با گسترش اسناد وب وجود نخواهد داشت. یکی از مهمترین ضعفهای تقریبا تمامی الگوریتمهای موجود، تنظیم پارامتر یادگیری در آنها با گسترش اسناد وب برای داشتن کورولیشن بالاتر است. برای حل مشکلات فوق می‌توان از الگوریتم پیشنهادی استفاده کرد

## ۵- نتیجه‌گیری

در این مقاله الگوریتم جدیدی با استفاده از اتوماتای یادگیر توزیع‌شده و مفهوم آنتروپی برای داده‌کاوی اطلاعات استفاده از وب ارائه شد. ایده اصلی در الگوریتم پیشنهادی این بود که اسنادی که باهمدیگر شباهت بیشتری را دارند، پاداش بیشتری را دریافت می نمایند. از نتایج بدست آمده از این الگوریتم می‌توان به عنوان ابزاری برای تعیین میزان تشابه اسناد وب استفاده کرد.

## 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] 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

[17] 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

[18] 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

[19] 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

[20] 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.

## زیرنویس‌ها

[1]Web Mining
[2]Statistical Method
[3]Distributed Learning Automata(DLA)
[4]Hypertext probabilistic grammar
[5]Learning Automata
[6]Environment
[7]Stationary
[8]Non-Stationary
[9]Variable Learning Automata
[10]Linear Reward Penalty
[11]Linear Reward Epsilon Penalty
[12]Linear Reward Inaction
[13]Power-Law