

## روشی جدید مبتنی بر اتوماتاهای یادگیر توزیع شده برای کشف ساختار اسناد

### وب

راهبه مجتهدی صفاری<sup>۱</sup>، محمد رضا مبینی<sup>۲</sup>

<sup>۱</sup> عضو هیأت علمی دانشکده مهندسی کامپیوتر، دانشگاه آزاد اسلامی، واحد لاهیجان، ایران

<sup>۲</sup> عضو هیأت علمی دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران

[mmeybodi@aut.ac.ir](mailto:mmeybodi@aut.ac.ir), [mojtahedi@iau-lahijan.ac.ir](mailto:mojtahedi@iau-lahijan.ac.ir)

### چکیده:

وب یک شبکه بزرگ از اسناد مرتبط به هم می باشد که سرعت در حال رشد است. بیشتر تحقیقات انجام شده در زمینه وب شامل استفاده همزمان از اطلاعات ساختاری و اطلاعات نحوه پیمایش کاربران از اسناد وب می باشد. در این مقاله روشی جدید مبتنی بر اتوماتای یادگیر توزیع شده برای کشف ساختار ارتباطی بین اسناد وب ارائه شده که در تعیین میزان شباهت بین اسناد وب در امر خوشه بندی، رتبه بندی، شخصی سازی وب و کشف زیر گرافهای متصل وب مورد استفاده قرار می گیرد. در الگوریتم پیشنهادی کشف ساختار، مانند سایر روشهای گزارش شده به هر سند وب یک اتوماتای یادگیر تخصیص داده می شود که بر اساس اطلاعات پیمایش کاربران ارتباطات بین اسناد را یاد می گیرد. تفاوت الگوریتم پیشنهادی نسبت به سایر روشهای موجود مبتنی بر اتوماتاهای یادگیر، عملکرد تصادفی و متفاوت اتوماتاهای یادگیر در انتخاب اقدام بعدی و نحوه تخصیص پاداش به اقدامها می باشد. نتایج شبیه سازیها نشان می دهد که الگوریتم پیشنهادی به دلیل استفاده از اتوماتاهای یادگیر با تعداد اقدامهای متغیر و بروز رسانی پارامتر یادگیری در حین افزایش تعداد صفحات وب، نسبت به سایر الگوریتم های گزارش شده مبتنی بر اتوماتاهای یادگیر دارای کارایی بالاتر و پیچیدگی زمانی پایینتری می باشد.

### واژه های کلیدی:

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

### ۱- مقدمه

وب بر اساس این مدل ذهنی می باشد. زمانیکه کاربران به طور متوالی از یک ترتیب مشخصی از اسناد، دیدن می کنند، به معنای این است که در بین این اسناد نوعی ارتباط وجود دارد. بنابراین می توان با استفاده از الگوی پیمایش کاربران که در لاگ فایلها ثبت می گردد، ساختار شبکه اطلاعاتی اسناد را بصورت اتوماتیک سازماندهی مجدد کرد. برخی از روشهای موجود برای سازماندهی ساختار اسناد وب از طریق کاوش در لاگ فایلها عبارتند از: استفاده از روشهای آماری، استفاده از قوانین یادگیری، مانند استفاده از قانون یادگیری هب در روش بولن [۱۲]، استفاده از روش کلونی مورچه ها [۱۳] در Antweb، استفاده از روشهای مبتنی بر اتوماتای یادگیر توزیع شده [۳، ۲، ۱]، استفاده از زنجیره های مارکوف [۱۴]. روشهای Antweb و بولن بدلیل اینکه از ماتریس ارتباطات اسناد بعنوان ورودی الگوریتم یادگیری در تعیین ساختار ارتباطی استفاده می کنند، در مجموعه های

استفاده از تکنیک های داده کاوی به منظور استخراج اتوماتیک اطلاعات از اسناد وب را کاوش وب می گویند. کاوش وب در سه سطح مطرح می شود. در سطح محتوا، در سطح ساختار و در سطح استفاده از وب. کاوش محتوای وب در مورد کشف وبازایی اطلاعات مفید از محتوا و داده های وب است. کاوش ساختار وب بر کشف و چگونگی مدل کردن ساختار ارتباطات اسناد وب تأکید دارد. در سطح استفاده از وب، هدف کشف اطلاعاتی است که از نحوه استفاده کاربران با وب بدست می آید. بیشتر روشهای موجود برای کشف ساختار ارتباطی اسناد وب از اطلاعات موجود در لاگ فایلها استفاده می نمایند. لاگ فایلها، فایلهایی هستند که درخواستهای کاربران را به هنگام مراجعه به صفحات وب ثبت می کنند. افراد یک مدل ذهنی از ارتباطات بین مفاهیم مختلف در ذهن خود دارند و نحوه حرکت آنها در بین اسناد

را توسط یک سیگنال تقویتی به اتوماتای یادگیر اطلاع می‌دهد. سپس اتوماتای یادگیر با اطلاع از عمل انتخاب شده و سیگنال تقویتی، وضعیت داخلی خود را بروز کرده و عمل بعدی خود را انتخاب می‌کند. محیط را می‌توان توسط سه تایی  $E=\{\alpha, \beta, c\}$  نشان داد که در آن  $\alpha=\{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه ورودی ها،  $\beta=\{\beta_1, \beta_2, \dots, \beta_r\}$  مجموعه خروجیها و  $c=\{c_1, c_2, \dots, c_r\}$  مجموعه احتمالات می‌باشد.  $c_i$  نشان دهنده احتمال نامطلوب بودن سیگنال تقویتی محیط در پاسخ به عمل  $\alpha_i$  می‌باشد. در یک محیط ایستا مقادیر  $c_i$  ها ثابت هستند، حال آنکه در یک محیط غیر ایستا این مقادیر در طی زمان تغییر می‌کنند. بر اساس اینکه تابع بروز رسانی وضعیت اتوماتای یادگیر (که با اطلاع از عمل انتخاب شده و سیگنال تقویت  $\beta$ ، 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 می‌گردد. در این مقاله از اتوماتای یادگیر با ساختار متغیر استفاده شده است که در ادامه معرفی می‌شود. اتوماتای یادگیر با ساختار متغیر توسط چهار تایی  $\{\alpha, \beta, p, T\}$  نشان داده می‌شود که در آن  $\alpha=\{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه اعمال اتوماتای یادگیر،  $\beta=\{\beta_1, \beta_2, \dots, \beta_r\}$  مجموعه ورودی های اتوماتای یادگیر،  $p=\{p_1, p_2, \dots, p_r\}$  بردار احتمال انتخاب هر یک از عمل‌ها و  $T[\alpha(n), \beta(n), p(n)]$  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  $n$ م اقدام  $\alpha_i$  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 روابط ۱ بروز می‌کند.

$$\begin{aligned} p_i(n+1) &= p_i(n) + a(1-\beta(n)) \cdot (1-p_i(n)) - b\beta(n)p_i(n) \\ p_j(n+1) &= p_j(n) + a(1-\beta(n)) \cdot \frac{b\beta(n)}{r-1} - b\beta(n)p_j(n) \end{aligned} \quad (1)$$

که در آن  $a$  پارامتر پاداش و  $b$  پارامتر جریمه می‌باشد. اگر  $b$  او  $a$  با هم برابر باشند، الگوریتم یادگیری،  $L_{RP}$ ، اگر  $b$  از

بزرگ از اسناد، قابل استفاده نمی‌باشند. استفاده از روشهای مارکوف هم به دلیل نیاز به محاسبه توان  $n$ ام ماتریس انتقال، دارای پیچیدگی محاسباتی بالایی می‌باشند.

در این مقاله یک الگوریتم جدید مبتنی بر اتوماتای یادگیر توزیع شده، بمنظور کشف ساختار ارتباطی بین اسناد وب پیشنهاد می‌گردد. الگوریتم پیشنهادی، بدلیل استفاده از اتوماتاهای یادگیر با تعداد اقدامهای متغیر، در مقایسه با الگوریتم ارائه شده در [۱]، با افزایش تعداد اسناد وب، دارای همگرایی بالاتر است. در روش ارائه شده در [۱]، بعلت استفاده از اتوماتاهای یادگیر با تعداد اعمال برابر اسناد، با افزایش تعداد اسناد وب طول بردار اعمال اتوماتای یادگیر افزایش یافته که این باعث کاهش سرعت همگرایی می‌گردد. همچنین روش پیشنهادی بدلیل متغیر در نظر گرفتن پارامتر یادگیری، بر خلاف روش مبتنی بر اتوماتاهای یادگیر با تعداد اقدامهای متغیر [۳] با افزایش تعداد اسناد، نیازی به تنظیم مجدد پارامتر یادگیری ندارد. همچنین روش پیشنهادی برخلاف روش گزارش شده در [۲]، همزمان با حرکت کاربر از یک سند به سند دیگر، میزان ارتباط بین اسناد را بصورت برخط محاسبه می‌نماید. علاوه بر این، بعلت اینکه در الگوریتم پیشنهادی فرایند محاسبه دور وجود ندارد، در مقایسه با روش ارائه شده در [۲]، دارای سربار محاسباتی کمتری می‌باشد. بمنظور ارزیابی و مقایسه روش پیشنهادی با سایر روشها از مدل ارائه شده توسط Liu و همکاران [۱۱]، استفاده شده است.

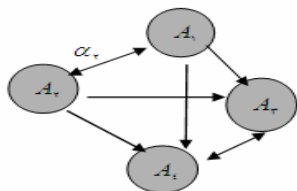
ادامه مقاله بدین صورت سازماندهی شده است. در بخش ۲ اتوماتاهای یادگیر و اتوماتای یادگیر توزیع شده به اختصار شرح داده می‌شوند. در بخش ۳ الگوریتم پیشنهادی برای کشف ساختار اسناد وب با استفاده از اتوماتای یادگیر توزیع شده ارائه می‌گردد. در بخش ۴ نتایج شبیه سازی آمده است. بخش ۵ نتیجه گیری می‌باشد.

## ۲- اتوماتاهای یادگیر

اتوماتای یادگیر یک مدل انتزاعی است که بطور تصادفی یک عمل از مجموعه متناهی اعمال خود را انتخاب کرده و بر محیط اعمال می‌کند. محیط عمل انتخاب شده توسط اتوماتای یادگیر را ارزیابی کرده و نتیجه ارزیابی خود

## ۲-۱- اتوماتای یادگیر توزیع شده

اتوماتای یادگیر توزیع شده شبکه‌ای از چند اتوماتای یادگیر است که برای حل یک مساله مشخص با یکدیگر همکاری می‌کنند. یک اتوماتای یادگیر توزیع شده را می‌توان بصورت یک گراف جهت‌دار نمایش داد. بصورتیکه مجموعه‌گره‌های آن را مجموعه‌ای از اتوماتاهای یادگیر و یالهای خروجی هر گره مجموعه اعمال متناظر با اتوماتای یادگیر متناظر با آن گره است. هنگامی که اتوماتای یکی از اعمال خود را انتخاب می‌کند، اتوماتایی که در دیگر انتهای یال متناظر با آن عمل قرار دارد، فعال می‌شود. در هر لحظه فقط یک اتوماتای یادگیر در اتوماتای یادگیر توزیع شده فعال می‌باشد. بعنوان مثال در شکل ۱، یک اتوماتای یادگیر توزیع شده با چهار اتوماتای یادگیر را نشان می‌دهد. برای اطلاعات بیشتر درباره اتوماتای یادگیر توزیع شده میتوان به مراجع [۴-۱۰]، مراجعه کرد.



شکل ۱: اتوماتای یادگیر توزیع شده

## ۳- الگوریتم پیشنهادی

در روش پیشنهادی، اسناد وب و کاربران استفاده کننده از آن نقش یک محیط تصادفی را برای اتوماتای یادگیر موجود در اتوماتای یادگیر توزیع شده ایفا می‌کنند. خروجی یک اتوماتای یادگیر توزیع شده، یک دنباله از اسناد وب مرور شده توسط یک کاربر هستند که مسیر حرکت کاربر را به سمت یک سند وب مورد نظر نشان می‌دهد. محیط با استفاده از این دنباله، پاسخی برای اتوماتای یادگیر توزیع شده تولید کرده و با استفاده از این پاسخ، ساختار داخلی اتوماتای یادگیر در اتوماتای یادگیر توزیع شده، طبق الگوریتم یادگیری بروز می‌شود. در این مقاله برای تعیین ساختار ارتباطی بین اسناد وب، در یک مجموعه با  $n$  سند، از یک اتوماتای یادگیر توزیع شده با  $n$  اتوماتای یادگیر با تعداد اقدامهای متغیر [۱۰]، که هر یک حداکثر  $n-1$  اقدام دارند، استفاده می‌شود. برای هر اتوماتای یادگیر در هر زمان تنها

$a$  خیلی کوچکتر باشد، الگوریتم یادگیری،  $L_{Rep}$  و اگر  $b$  صفر باشد، الگوریتم یادگیری،  $L_{RI}$  نام دارد. اتوماتای یادگیری که در بالا معرفی شد، دارای تعداد اقدام های ثابتی می باشد. در بعضی از کاربردها به اتوماتای یادگیر با تعداد اقدام متغیر نیاز می باشد [۱۰]. 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 می کند. انتخاب مجموعه اقدامهای فعال اتوماتای یادگیر  $V(n)$  توسط یک عامل خارجی و بصورت تصادفی انجام می شود. 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 (۳) ایجاد می کند. آنگاه اتوماتای یادگیر یک اقدام از مجموعه اقدامهای فعال خود را به صورت تصادفی و بر اساس بردار احتمال  $\hat{P}(n)$  انتخاب کرده و بر محیط اعمال می کند. در یک اتوماتای یادگیر با الگوریتم یادگیری خطی، اگر اقدام انتخاب شده  $\alpha_i$  باشد، اتوماتای یادگیر پس از دریافت پاسخ محیط، بردار احتمال  $\hat{P}(n)$  اقدامهای خود را در صورت دریافت پاسخ مطلوب بر اساس رابطه (۴) و در صورت دریافت پاسخ نامطلوب طبق رابطه (۵) بروز می نماید، سپس اتوماتای یادگیر بردار احتمال اقدامهای خود  $P(n)$  را با استفاده از بردار  $\hat{P}(n+1)$  و طبق رابطه (۶) بروز می کند.

$$K(n) = \sum_{\alpha_i \in V(n)} p_i(n) \quad (2)$$

$$\hat{P}_i(n) = \text{prob}[\alpha(n) = \alpha_i | \alpha_i \in V(n)] = \frac{p_i(n)}{K(n)} \quad (3)$$

که در فرمول بالا منظور از  $V(n)$  تعداد اقدامهای فعال می باشد.

$$\hat{P}_i(n+1) = \hat{P}_i(n) + a \cdot (1 - \hat{P}_i(n)) \quad (4)$$

$$\hat{P}_j(n+1) = \hat{P}_j(n) - a \cdot \hat{P}_j(n) \quad \forall j \neq i$$

$$\hat{P}_i(n+1) = \hat{P}_i(n) + (1-b) \cdot (1 - \hat{P}_i(n)) \quad (5)$$

$$\hat{P}_j(n+1) = \frac{b}{\hat{r}-1} + (1-b) \hat{P}_j(n) \quad \forall j \neq i$$

$$P_i(n+1) = \hat{P}_i(n+1) \cdot K(n) \quad (6)$$

$$P_j(n+1) = P_j(n) \quad j, \alpha_j \notin V(n)$$

کد این الگوریتم بصورت زیر می باشد. مقدار بیشتر برای  $E_{\alpha}^{cur\_page}$ ، نشان دهنده ارتباط بیشتر دو سند  $\alpha$  و  $cur\_page$  می باشد و بالعکس مقدار کمتر این معیار نشان دهنده ارتباط کمتر این دو سند است.

Proposed\_DLA\_Algorithm:  
**Variables:** **route** : Array of [number of users][user's path] that is obtained from Liu model  
**L[number of user]**: Array of [number of pages][number of pages].  
**DLA**: distributed learning automata which contains n LA having n-1 actions.  
**Begin**  
**For** all users **do**  
    Step\_id=1;  
    Cur\_page=route[user\_id][Step\_id];  
    **While** (Step\_id<=Max\_forage & S<Maxthd & S>Minthd) **do**  
        **If** (Dist(Cur\_page,  $\alpha$ )<Threshold)  
            **If**  $A_{\alpha}^{cur\_page}=0$  **then**  
                 $A_{\alpha}^{cur\_page}=1$ ;  
            **End // If**  
            **If** (L[user\_id](Cur\_page,  $\alpha$ )==1)  
                Set  $\beta=0$   
                
$$E_{\alpha}^{cur\_page} = -(\hat{p}_{\alpha}^{cur\_page} \log \hat{p}_{\alpha}^{cur\_page} + (1 - \hat{p}_{\alpha}^{cur\_page}) \log (1 - \hat{p}_{\alpha}^{cur\_page}))$$

$$a_{\alpha} = \frac{E_{\alpha}^{cur\_page}}{E_{\alpha}^{cur\_page} + 1}$$

$$\hat{p}_{\alpha}^{cur\_page}(n+1) = \hat{p}_{\alpha}^{cur\_page}(n) + a_{\alpha}(1 - \hat{p}_{\alpha}^{cur\_page}(n))$$

$$\hat{p}_j^{cur\_page}(n+1) = \hat{p}_j^{cur\_page}(n) - a_{\alpha} \hat{p}_j^{cur\_page}(n) \quad \forall j \neq \alpha$$
                **Else**  
                     $\hat{p}_{\alpha}^{cur\_page}(n+1) = \hat{p}_{\alpha}^{cur\_page}(n)$   
                     $\hat{p}_j^{cur\_page}(n+1) = \hat{p}_j^{cur\_page}(n) \quad \forall j \neq \alpha$   
                **End // If**  
                Step\_id= Step\_id+1;  
                Cur\_page=  $\alpha$  ;  
            **Else**  
                 $p_{\alpha}^{cur\_page}$  probability is divided between other Cur\_page's Actions.  
                 $p_{\alpha}^{cur\_page} = 0$ ;  
            **End // If**  
            **End // While**  
            **End // For**  
**End**

شکل ۲: شبه کد الگوریتم پیشنهادی

#### ۴- نتایج آزمایشها

در این بخش ابتدا مدل شبیه سازی و سپس نتایج شبیه سازی الگوریتم پیشنهادی و مقایسه آنها با الگوریتم های موجود ارائه می شوند.

زیرمجموعه ای از اقدامهای فعال و می تواند قابل استفاده باشد [۱۰]. در روش پیشنهادی، در ابتدای کار تمامی اقدامهای اتوماتای یادگیر در اتوماتای توزیع شده غیر فعال می باشند. با شروع حرکت کاربر از سند، اتوماتای متناظر با آن سند در اتوماتای یادگیر توزیع شده فعال می شود سپس اتوماتای فعال شده در اتوماتای یادگیر توزیع شده، یک اقدام را بطور تصادفی انتخاب نموده و در صورتیکه فاصله اقلیدسی بین سند متناظر با اتوماتای  $i$  و سند متناظر با اقدام انتخابی بیشتر از یک مقدار آستانه اتصال باشد اقدام حذف شده و احتمال آن بین بقیه تقسیم می شود. در صورتیکه فاصله اقلیدسی بین سند متناظر با اتوماتای  $i$  و سند متناظر با اقدام انتخابی کمتر از یک مقدار آستانه اتصال باشد آنگاه اگر ارتباط بین این دو سند در لاگ فایل، به ازای کاربر مورد نظر موجود باشد، پاسخ محیط مطلوب بوده و اتوماتای  $i$  به اقدام انتخابی خود پاداش می دهد، در غیر اینصورت پاسخ محیط نامطلوب بوده و اتوماتای  $i$  باید اقدام انتخابی را به میزان پارامتر  $b$  طبق الگوریتم یادگیری جریمه نماید، بدلیل استفاده از الگوریتم یادگیری  $L_{RI}$ ، مقدار  $b$  صفر می باشد، همچنین ضریب پاداش دریافتی توسط اقدام  $\alpha$ ، بر اساس مفهوم آنتروپی، بصورت  $a_{\alpha}$ ، تعریف می شود. آنتروپی تعیین می کند که آیا پروفایل علاقه هر کاربر متوازن است یا نه. پروفایل علایق کاربران نشان دهنده رفتار آنها در جستجو می باشد. فرق الگوریتم پیشنهادی با الگوریتمهای پیشین مبتنی بر اتوماتای یادگیر توزیع شده، در نحوه عملکرد اتوماتاهای یادگیر در اتوماتای یادگیر توزیع شده می باشد. در تمام الگوریتمهای موجود، هر اتوماتا برای انتخاب عمل خود به لاگ فایل کاربران نگاه می کند و در هر مرحله همان اقدامهایی را فعال می کند که کاربر از آنها عبور می کند. بعبارت دیگر نحوه عملکرد اتوماتای تخصیص داده شده به هر سند بر اساس اطلاعات نحوه حرکت کاربر موجود در آن سند می باشد، بنابراین در تمام الگوریتمهای موجود، عملکرد انتخاب اقدامهای اتوماتا قابل پیش بینی و بر اساس مدل حرکات کاربر است، اما در الگوریتم پیشنهادی اتوماتای یادگیر در هر لحظه بصورت تصادفی اقدام خود را انتخاب نموده و عمل بروزرسانی مقدار احتمال این اقدام را بر اساس مراجعه به لاگ فایل و مقایسه با آن، انجام می دهد. شبه

#### ۴-۱- مدل شبیه سازی

برای ارزیابی الگوریتم پیشنهادی، از مدل معرفی شده در [۱۱]، برای نشان دادن ساختار وب و چگونگی استفاده کاربران از وب، استفاده می شود. اعتبار این مدل توسط Liu و همکاران با استفاده از اطلاعات بدست آمده از چندین وب سایت بزرگ مانند مایکروسافت تأیید شده است. بر این اساس، در این مقاله پرو فایل علاقه کاربران بصورت توزیع قانون-توانی و توزیع محتوای اسناد وب، بصورت توزیع نرمال در نظر گرفته شده است. سایر پارامترهای استفاده شده در مدل [۱۱] برای شبیه سازیهای انجام شده در این مقاله در جدول ۱ نشان داده شده است. در این مدل، هر سند وب با یک بردار محتوا نمایش داده می شود. طول این بردار برابر با تعداد موضوعات موجود در سیستم است. هر عضو این بردار، میزان ارتباط سند متناظر با آن بردار را با یکی از این اسناد نشان می دهد. با استفاده از بردار محتوای دو سند  $(d_{i,j})$ ، طبق رابطه (۱۰) محاسبه می شود. در این مقاله معیار شباهت دو سند در حالت ایده آل، بصورت عکس فاصله اقلیدسی این دو سند تعریف شده است. برای ارزیابی الگوریتم پیشنهادی، با سایر روشها، از معیاری به نام کورولیشن استفاده می کنیم. کورولیشن، میزان همبستگی خطی بین داده ها را مشخص می کند. کورولیشن دو مجموعه داده  $P$  و  $P'$  طبق رابطه (۱۲) محاسبه میشود. در این رابطه،  $N$  تعداد داده هاست. مجموعه  $P$ ، ساختار ایجاد شده در حالت ایده آل توسط مدل شبیه سازی و  $P'$  ساختار ایجاد شده توسط الگوریتم پیشنهادی، است.

$$C_n = \{cw_n^1, cw_n^2, \dots, cw_n^M\} \quad (7)$$

$$p = \{p_{ij} | i, j = 1, 2, 3, \dots, n, \quad i \neq j\} \quad (8)$$

$$p_{ij} = \frac{(d_{ij})^{-1}}{\sum_{k=1}^{k=n} (d_{ik})^{-1}} \quad (9)$$

$$d_{ij} = \sqrt{\sum_{k=1}^{k=M} (cw_i^k - cw_j^k)^2} \quad (10)$$

$$p' = \{p'_{ij} | i, j = 1, 2, 3, \dots, n, \quad i \neq j\} \quad (11)$$

$p'_{ij}$  = احتمال انتخاب اقدام  $j$  از اتوماتای  $i$  است، در صورتیکه این اقدام غیر فعال باشد مقدار این احتمال صفر می باشد.

$$Cor(P, P') = \frac{\sum PP' - (\sum P \sum P') / N}{\sqrt{(\sum P^2 - (\sum P)^2 / N)(\sum P'^2 - (\sum P')^2 / N)}} \quad (12)$$

جدول ۱: پارامترهای شبیه سازی

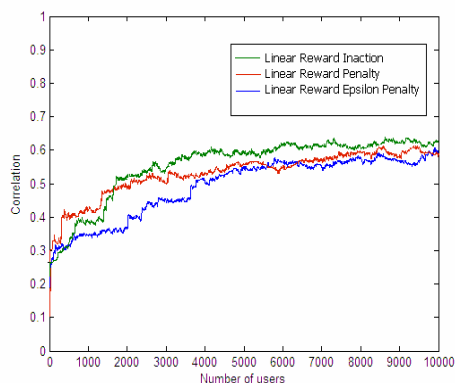
۰.۷	حد آستانه اتصال
۲۰	تعداد اسناد
۲۰۰۰۰	تعداد کاربران
۵	تعداد موضوعات
۰.۲	$T_c$ مقدار ثابت سند اولیه در موضوعات مختلف
-	$\Delta M_i^c$ ضریب ثابت کاهش اشتیاق کاربر
-	$\Delta M_i^v$ ضریب متغیر کاهش اشتیاق کاربر
۱	$\alpha_{ij}$ پارامتر توزیع قانون-توانی توزیع احتمال
	علاقه کاربران
۱.۲	$\phi$ ضریب پاداش دریافتی از مشاهده یک سند
۰.۵	$\lambda$ ضریب جذب اطلاعات از یک سند توسط یک کاربر
۵.۹۷	$\mu_m$ میانگین توزیع نرمال $\Delta M_i^v$
۰.۲۵	$\sigma_m$ واریانس توزیع نرمال $\Delta M_i^v$
-	$\mu_i$ میانگین توزیع نرمال برای مقدار افزایش
	یک گره برای یک موضوع خاص
۳	$\alpha_p$ پارامتر توزیع قانون-توانی توزیع احتمالات
	وزنهای مطالب برای هر سند
۰.۲۵	$\sigma$ واریانس توزیع نرمال برای مقدار افزایش یک
	گره برای یک موضوع خاص
۱	$\theta$ ضریب کاهش علاقه کاربر
۰.۲	حداقل اشتیاق کاربر برای ادامه جستجو

#### ۴-۲- ارزیابی کارایی الگوریتمهای یادگیری

قبل از مقایسه الگوریتم پیشنهادی تعیین ساختار با سایر روشهای موجود، به بررسی و ارزیابی کارایی الگوریتمهای یادگیری استاندارد  $L_{RI}$ ،  $L_{RP}$  و  $L_{ReP}$  می پردازیم. این آزمایش بمنظور بررسی تأثیر مقادیر پارامتر یادگیری بر روی تعداد اسناد انجام شد. به ازای تعداد اسناد مختلف و تنظیم مقادیر متفاوت برای پارامتر یادگیری، حداکثر مقدار کورولیشن و تعداد تکرارهای لازم برای رسیدن به این سطح از کورولیشن، برای الگوریتم یادگیری  $L_{RI}$ ، در جدول ۲ نشان داده شده است.

الگوریتم یادگیری	۱۰۰۰	۲۰۰۰	۵۰۰۰	۱۰۰۰۰
$L_{RI}$	۰,۳۷۵	۰,۵۰۳	۰,۵۵۴	۰,۶۱۸۲
	۹۴۱	۱۹۹۹	۴۴۷۰	۶۶۱۷
$L_{ReP}$	۰,۳۶۵	۰,۴۳۷	۰,۵۷۲	۰,۵۸۳
	۹۰۴	۱۵۶۰	۴۸۴۵	۸۳۷۱
$L_{RP}$	۰,۴۰	۰,۴۲۵	۰,۵۴۳	۰,۵۹۴
	۹۲۳	۱۵۷۷	۴۶۲۹	۶۲۰۴

همانگونه که در جدول بالا مشاهده می شود با افزایش تعداد مراحل یادگیری (تعداد کاربران)، میزان کورولیشن به ازای هر یک از الگوریتمهای یادگیری افزایش می یابد. به منظور ارزیابی کارایی الگوریتمهای یادگیری  $L_{RI}$ ،  $L_{ReP}$  و  $L_{RP}$  در نحوه بروز رسانی بردار احتمالات اتوماتاهای یادگیر و تعیین ساختار ارتباطی بین اسناد، شبکه شامل ۲۱ سند در نظر گرفتیم و طول بردار اعمال اتوماتاهای یادگیر را برابر با تعداد اسناد تنظیم نمودیم. کورولیشن بین ساختار ارتباطی بدست آمده از هر الگوریتم یادگیری و ساختار ارتباطی ایده آل را تا تکرار ۱۰۰۰۰ محاسبه نمودیم. در انجام محاسبات مقدار پارامترهای یادگیری  $a$  و  $b$  را به ترتیب ۰,۰۵ و ۰,۰۱ تعریف نمودیم. نتایج آزمایش در شکل ۳ نشان داده شده است و مشخص می کند که الگوریتم یادگیری  $L_{RI}$ ، در مقایسه با دو الگوریتم یادگیری دیگر دارای سطح کورولیشن بالاتری است، بعبارت دیگر ساختار ارتباطی بدست آمده با استفاده از این الگوریتم یادگیری، به ساختار ارتباطی ایده آل نزدیکتر می باشد.



شکل ۳: مقایسه کورولیشن الگوریتمهای یادگیری

$L_{RI}$ ،  $L_{ReP}$  و  $L_{RP}$  به ازای تعداد اسناد ۲۱ و مراحل یادگیری ۱۰۰۰۰

جدول ۲: حداکثر مقدار کورولیشن بدست آمده و تعداد مراحل

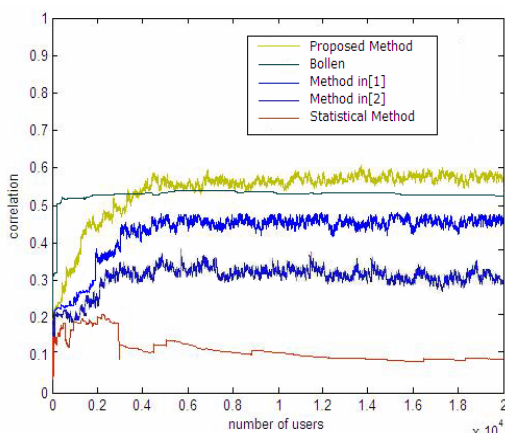
یادگیری الگوریتم یادگیری  $L_{RI}$  برای اسناد مختلف به ازای تعداد کاربران ۵۰۰۰

تعداد اسناد	۱۰	۳۱	۵۰	۷۱	۹۱
پارامتر یادگیری					
$a = 0.001$	۰,۴۱۱	۰,۲۹۴	۰,۲۲۲	۰,۱۸۸	۰,۱۷
	۴۶۵۷	۴۶۱۵	۴۸۴۶	۴۹۸۳	۴۹۹۰
$a = 0.01$	۰,۴۵۶	۰,۳۷۹	۰,۲۹۹	۰,۲۲۷	۰,۲۰
	۴۴۰۹	۴۸۶۶	۴۹۴۶	۴۹۸۹	۴۹۰۱
$a = 0.05$	۰,۵۴۳	۰,۵۵۴	۰,۳۹۴	۰,۲۶۶	۰,۲۵
	۴۹۳۹	۴۶۷۰	۴۹۴۱	۴۹۹۹	۴۹۴۳
$a = 0.07$	۰,۵۸۲	۰,۵۴۴	۰,۴۳۳	۰,۲۷۱	۰,۲۳۳
	۴۷۸۳	۴۵۳۷	۴۹۸۳	۴۸۴۳	۴۳۸۸
$a = 0.1$	۰,۶۰۱	۰,۵۹۴	۰,۴۵۸	۰,۲۵۷	۰,۲۱
	۴۳۶۰	۴۳۶۷	۴۸۱۶	۴۸۸۲	۳۹۰۶

همانگونه که در جدول ۲ نشان داده شده است با افزایش تعداد اسناد وب، و با ثابت نگه داشتن حداکثر تعداد مراحل یادگیری، میزان رسیدن به حداکثر کورولیشن کاهش می یابد. آن به این دلیل است که با افزایش تعداد اسناد، طول بردار اعمال اتوماتا افزایش یافته و میزان همگرایی کاهش می یابد. به همین دلیل لازم است که پارامتر یادگیری مناسب را به ازای افزایش تعداد اسناد وب تنظیم نمود، منظور از پارامتر یادگیری مناسب، پارامتر یادگیری می باشد که در ازای آن میزان کورولیشن ساختار ارتباطی و ساختار ایده آل در کمترین تکرار، بیشترین مقدار را دارد. نتایج آزمایش نشان می دهد که مقدار پارامتر یادگیری رابطه عکس با تعداد اسناد (طول بردار اعمال اتوماتا) دارد. بعبارت دیگر همانطور که در جدول ۲ مشاهده می شود با افزایش تعداد اسناد و با تنظیم مقدار کمتر برای پارامتر یادگیری به حداکثر سطح کورولیشن می رسیم. بنابراین بمنظور رسیدن به مقدار حداکثر کورولیشن، لازم است که دو عامل پارامتر یادگیری و تعداد مراحل یادگیری را بطور مناسب انتخاب نماییم. برای بررسی تأثیر تعداد مراحل یادگیری در میزان کورولیشن آزمایش دیگری انجام گرفت، نتایج این آزمایش در جدول ۳ نشان داده شده است. در این آزمایش تعداد اسناد را ۲۱ پارامتر یادگیری  $a$  و  $b$  را به ترتیب ۰,۰۵ و ۰,۰۱ در نظر گرفتیم.

جدول ۳: تأثیر تعداد مراحل یادگیری در حداکثر مقدار کورولیشن بدست آمده به ازای هر یک از الگوریتمهای یادگیری





شکل ۴: مقایسه الگوریتم پیشنهادی با سایر روشهای گزارش شده

به منظور بررسی تأثیر تعداد اسناد وب و تعداد کاربران بر روی کارایی الگوریتم پیشنهادی و الگوریتم ارائه شده در [۱] (از لحاظ کارایی بهتر از سایر روشهای موجود مبتنی بر اتوماتای یادگیر عمل می کند)، آزمایش دیگری انجام دادیم. در این آزمایش، حداکثر مقدار کورولیشن بدست آمده از دو روش ارائه شده در [۱] و روش پیشنهادی، معیار ارزیابی ما می باشد. همانطور که در جداول ۴ و ۵ دیده می شود، الگوریتم پیشنهادی، همواره دارای کورولیشن بالاتری نسبت به الگوریتم ارائه شده در [۱] است. با افزایش تعداد اسناد، کورولیشن در هر دو الگوریتم کاهش می یابد، دلیل آن این است که با افزایش تعداد اسناد، طول بردار اعمال اتوماتا افزایش می یابد (جدول ۴). هرچه تعداد کاربران بیشتر می شود، انتظار می رود که به سطح کورولیشن بهتری دست یابیم، این مسأله در جدول ۵ مشاهده می شود.

جدول ۴: حداکثر کورولیشن بدست آمده برای روشهای مختلف به ازای تعداد کاربران ۲۰۰۰ و تعداد اسناد ۲۰ و ۱۰۰ و ۳۰

تعداد صفحات روش	۱۰	۲۰	۳۰	۴۰
روش پیشنهادی	۰.۷۹۵۳	۰.۶۱۲۱	۰.۵۶۳۱	۰.۵۳۷۸
روش ارائه شده در [۱]	۰.۶۸۸۷	۰.۵۳۸۷	۰.۴۸۰۲	۰.۴۳۸۲

جدول ۵: حداکثر کورولیشن بدست آمده برای روش های مختلف به ازای تعداد اسناد ۲۰ و تعداد کاربران ۲۰۰۰ و ۱۰۰۰۰ و ۳۵۰۰۰

### ۳-۴- ارزیابی کارایی الگوریتم پیشنهادی تعیین ساختار

الگوریتم یادگیری استفاده شده در روشهای تعیین ساختار برای بروزرسانی بردار احتمالات اتوماتاهای یادگیر،  $L_{RI}$  می باشد. شکل ۴، مقایسه الگوریتم پیشنهادی را با الگوریتم ارائه شده در [۱]، [۲]، روش بولن و روش آماری مطرح شده در [۴]، نشان می دهد. در این آزمایش تعداد کاربران ۲۰۰۰، تعداد موضوعات مرتبط با هر سند ۵ و تعداد اسناد وب ۲۰ در نظر گرفته شده است. الگوریتم پیشنهادی برای تعیین ساختار اسناد وب، بهتر از سایر الگوریتم های ارائه شده مبتنی بر اتوماتای یادگیر، عمل می کند، بعلاوه اینکه دو عامل مهم را در مورد آن لحاظ کرده ایم. اول اینکه تعداد اقدامهای هر اتوماتای یادگیر مطابق با [۱۰]، متغیر در نظر گرفته شده و این باعث می شود که در هر لحظه فقط زیر مجموعه ای از اقدامهای فعال یک اتوماتا، بروزرسانی شوند. دوم اینکه، نحوه بروز رسانی الگوریتم یادگیری بر اساس مفهوم آنتروپی [۱]، می باشد. ترکیب این دو عامل باعث می شود، نسبت به الگوریتمی که طول بردار اعمال آن برابر با تعداد اسناد وب است و در هر لحظه کل بردار اعمال بروز می گردد [۱]، سریعتر همگرا شود. همچنین، روش ارائه شده به دلیل اینکه از مفهوم آنتروپی استفاده می کند نسبت به روش مبتنی بر اتوماتای یادگیر با اقدام های متغیر، با افزایش تعداد اسناد، نیاز به تنظیم پارامتر یادگیری ندارد. به منظور نمایش یک کران پایین برای کارایی این دو الگوریتم، از یک روش آماری ساده، استفاده شده است. در این روش آماری، شباهت دوسند  $i$  و  $j$  به صورت نسبت تعداد دفعاتی که کاربران از سند  $i$  به سند  $j$  رجوع کرده اند  $(N(i, j))$ ، به تعداد دفعاتی که کاربران از سند  $i$  به هر سند دیگری مراجعه کرده اند، محاسبه می شود.

$$Similarity(i, j) = \frac{N(i, j)}{\sum_{k=1}^n N(i, k)} \quad ( )$$

- [4] Meybodi, M.R. and Beigy, H., "Solving Stochastic Path Problem Using Distributed Learning Automata," Proceedings of The Sixth Annual International CSI Computer Conference, CSIC2001, Isfahan, Iran, pp. 86-70, Feb. 20-22, 2001.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] Alipour, M. and Meybodi, M.R., "Solving Maximal independent Set Problem Using Distributed Learning Automata", 14<sup>th</sup> Iranian Electrical Engineering Proceedings of (ICEE2006), Amirkabir University, Tehran, Iran, May 16-18, 2006.
- [9] Narendra, K.S. and Thathachar, M.A.L., "Learning Automata: An Introduction", Prentice Hall, 1989.
- [10] Thathachar, M.A.L. and Harita Bhaskar, R., "Learning Automata with Changing Number of Actions", IEEE Transactions on Systems Man and Cybernetics, vol. 17, no.6, pp. 1095-1100, Nov. 1987.
- [11] Liu, J., Zhang, S. and Yang, J., "Characterizing Web Usage Regularities with Information Foraging Agents", IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 4, pp. 566-584, April 2004.
- [12] Heylighen, F. and Bollen, J., "Hebbian Algorithms for a Digital Library Recommendation System", Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'02), 2002, pp. 439-446.
- [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] Jianhan, Zhu., "Mining Web Site Link Structures for Adaptive Web Site Navigation and Search", Ph.D Thesis, University of Ulster at Jordanstown, October 2003.

تعداد کاربران				
روش	۱۰۰۰۰	۲۰۰۰۰	۳۵۰۰۰	۴۵۰۰۰
روش پیشنهادی	۰.۶۰۲۳	۰.۶۱۲۱	۰.۶۸۴۳	۰.۷۱۲۴
روش ارائه شده در [۱]	۰.۵۳۳۵	۰.۵۳۸۷	۰.۵۴۹۰	۰.۶۰۴۸

## نتیجه‌گیری

در این مقاله یک روش جدید مبتنی بر اتوماتای یادگیر توزیع شده برای کشف ساختار ارتباطی بین اسناد وب پیشنهاد گردید الگوریتم پیشنهادی، بدلیل استفاده از اتوماتاهای یادگیر با تعداد اقدامهای متغیر و بروز رسانی پارامتر یادگیری با افزایش تعداد اسناد وب، دارای سه مزیت می باشد که عبارتند از: ۱- به دلیل استفاده از اتوماتاهای یادگیر با تعداد اقدامهای متغیر، الگوریتم پیشنهادی، نسبت به روشی که تعداد اقدامهای آن برابر با تعداد اسناد می باشد، با افزایش تعداد اسناد وب، دارای کورولیشن بالاتر است ۲- عدم نیاز به تنظیم مجدد پارامترهای یادگیری اتوماتاهای یادگیر، با تغییر تعداد اسناد وب، به دلیل استفاده از مفهوم آنتروپی. ۳- قابلیت استفاده بر خط. نتایج شبیه سازیها نشان می دهند روش پیشنهادی به دلیل عملکرد کاملاً تصادفی اتوماتاها درانتخاب اقدام بعدی و دادن پاداش از طریق مقایسه سند متناظر این اقدام با سند انتخابی کاربر در لاگ فایل، متفاوت از سایر روشهای موجود مبتنی بر اتوماتای یادگیر می باشد و نسبت به آنها دارای کارایی بالاتر است. از ساختار ارتباطی بدست آمده می توان بمنظور تعیین شباهت بین اسناد وب، رتبه بندی، خوشه بندی اسناد و کشف زیر گرافهای متصل وب استفاده نمود.

## مراجع

- [۱] بابک اناری و محمد رضا میبیدی، "یک روش مبتنی بر اتوماتای یادگیر توزیع شده برای تعیین ساختار اسناد وب"، مجموعه مقالات دوازدهمین کنفرانس بین المللی انجمن کامپیوتر ایران، تهران، ایران، ۱۳۸۵.
- [2] 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 (IKT 2005), Tehran, Iran, May 24-26, 2005.
- [3] Hashemi, A.B. and Meybodi, M.R., "Web Usage Mining Using Distributed Learning Automata", Computer Engineering Department, Technical Report, 2005.