

## یک مدل خود سازمانده برای ساختار اطلاعاتی اسناد با استفاده از اتوماتاهای یادگیر توزیع شده

سعید ساعتی      محمدرضا میبیدی

saeid\_saati@yahoo.co.uk

4

5

ANTWeb

کلمات کلیدی: اتوماتاهای یادگیر، اتوماتاهای یادگیر توزیع شده، خود سازمانده

### ۱- مقدمه

یکی از مسائل مهم در مجموعه های بزرگ از اسناد مانند کتابخانه های دیجیتال اخذ و سازماندهی اسناد به نحوی است که بتوان آنرا بصورت کارا بازیابی نمود. یکی از روشهای معمول برای انجام اینکار استفاده از متادیتا<sup>۴</sup> برای اندیس

1  
2  
3

<sup>4</sup> Clustering

<sup>5</sup> Ranking

<sup>6</sup> Metadata

گذاری اسناد میباشد. ولی این روش اندیس گذاری نمی تواند موضوع یک سند را بطور کامل پوشش دهد. متادیتاهای مرسوم مانند کلیدواژه ها یا پیوندهای ثابت بین اسناد که بعنوان متادیتای پیوسته استفاده می شود دارای ساختار ثابت هستند. همچنین در استفاده از کلید واژه ها مشکل ابهام کلمات بصورت جدی مطرح می باشد [1]. به همین دلیل روشهای متعددی برای بدست آوردن متادیتا بصورت پویا و با بهره گیری از الگوی استفاده کاربران بوجود آمده است که به کشف الگوهای علایق کاربران یا الگوهای حرکات آنها می پردازند [10,11]. اکثر این روشها از یک مجموعه اطلاعات فاقد عمومیت مانند تعدادی دسته از پیش تعریف شده برای دسته بندی علایق یا الگوهای حرکتی کاربران استفاده می کنند [12] و بنابراین امکان استفاده از آنها در سیستمهای پویا و قابل گسترش وجود ندارد.

افراد یک مدل ذهنی از ارتباطات بین مفاهیم مختلف در ذهن خود دارند و حرکات بین اسناد بر اساس این مدل ذهنی آنها می باشد و بنابراین می توان از الگوی حرکات کاربر بین اسناد برای نشان دادن ارتباط بین اسناد استفاده کرد. زمانیکه کاربران چندین بار به یک توالی مشخص از اسناد مراجعه میکنند نشاندهنده یک نوع ارتباط بین آن اسناد می باشد. به همین دلیل با استفاده از الگوی استفاده کاربران (حرکت کاربر بین اسناد) میتوان ساختار شبکه اطلاعاتی اسناد را بطور اتوماتیک ایجاد و سپس سازماندهی مجدد کند.

اولین روشی که با استفاده از الگوی حرکات کاربر بین اسناد ساختار اطلاعاتی خود را اصلاح می کند در [2] و توسط بولن گزارش شده است. در این روش ارتباطات بین اسناد با استفاده از روشی مانند قانون هب اصلاح میگردد. در روش بولن اینگونه عمل می شود که برای اسناد موجود یک ماتریس ارتباطات در نظر گرفته می شود که هر درایه  $a_{ij}$  آن میزان ارتباط بین اسناد  $i$  و  $j$  را نشان می دهد. در این روش با فعال شدن متوالی سه سند  $\{v_i, v_j, v_k\}$  توسط یک کاربر ارتباطات آن سندها در ماتریس ارتباطات با استفاده از سه قانون زیر اصلاح می شوند.

$$\text{Frequency: } a_{ij} = a_{ij} + r_f \quad a_{jk} = a_{jk} + r_f$$

$$\text{Symmetry: } a_{ji} = a_{ji} + r_s \quad a_{kj} = a_{kj} + r_s$$

$$\text{Transitivity: } a_{ik} = a_{ik} + r_t$$

[16]

ANTWeb

(DLA-DL)

ANTWeb<sup>7</sup>

[8]

[17]

[15]

<sup>7</sup> Bollen

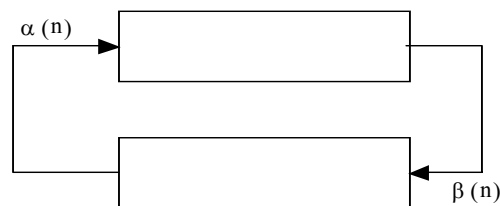
برای ارزیابی عملکرد ساختار پیشنهادی از مقایسه کورلیشن<sup>۸</sup> ساختار اطلاعاتی بدست آمده توسط ساختار پیشنهادی با کورلیشن ساختار اطلاعاتی ایده ال استفاده می کنیم. کورلیشن معیاری برای بدست آوردن وابستگی خطی بین این دو ماتریس می باشد که بصورت زیر تعریف می شود.

$$\text{Corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_x \sigma_y} = \frac{\sum XY - (\sum X \sum Y) / N}{\sqrt{(\sum X^2 - (\sum X)^2 / N)(\sum Y^2 - (\sum Y)^2 / N)}}$$

ادامه مقاله بدین صورت سازماندهی شده است. در بخش ۲ اتوماتاهای یادگیر و اتوماتای یادگیر توزیع شده شرح داده می شوند. الگوریتم پیشنهادی در بخش ۳ و نتایج شبیه سازی در بخش ۴ آمده است. در بخش ۵ به تعدادی از کاربردهای ساختار پیشنهادی اشاره شده است. بخش پایانی نتیجه گیری میباشد.

## ۲- اتوماتای یادگیر<sup>۹</sup>

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.



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

**محیط<sup>۱۰</sup>:** 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 ورودیها،  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$  مجموعه خروجیها و  $c \equiv \{c_1, c_2, \dots, c_r\}$  مجموعه احتمالهای جریمه می باشد. هر گاه  $\beta$  مجموعه دو عضوی باشد، محیط از نوع P می باشد. در چنین محیطی  $\beta_1 = 1$  به عنوان جریمه و  $\beta_2 = 0$  به عنوان پاداش در نظر گرفته می شود. در محیط از نوع Q،  $\beta(n)$  می تواند 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 می باشد. در محیط ایستا<sup>۱۱</sup> مقادیر  $c_i$  بدون تغییر می مانند، حال آنکه در محیط غیر ایستا<sup>۱۲</sup> این مقادیر در طی زمان تغییر می کنند. 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 می پردازیم.

<sup>8</sup> Correlation

<sup>9</sup> Learning Automata

<sup>10</sup> Environment

<sup>11</sup> Stationary

<sup>12</sup> Non-Stationary

**اتوماتای یادگیر با ساختار متغیر<sup>۱۳</sup>:** اتوماتای یادگیر با ساختار متغیر توسط ۴ تائی  $\{\alpha, \beta, p, T\}$  نشان داده می شود که در آن  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه عملهای اتوماتا،  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$  مجموعه ورودیهای اتوماتا،  $p \equiv \{p_1, p_2, \dots, p_r\}$  بردار احتمال انتخاب هر یک از عملها، و  $p(n+1) = T[\alpha(n), \beta(n), p(n)]$  الگوریتم یادگیری می باشد. در این نوع از اتوماتاها، اگر عمل  $\alpha_i$  در مرحله  $n$  انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال  $p_i(n)$  افزایش یافته و سایر احتمالات کاهش می یابند. و برای پاسخ نامطلوب احتمال  $p_i(n)$  کاهش یافته و سایر احتمالات افزایش می یابند. در هر حال، تغییرات به گونه ای صورت می گیرد 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.

الف- پاسخ مطلوب

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \quad (2)$$

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

ب- پاسخ نامطلوب

$$p_i(n+1) = (1-b)p_i(n) \quad (3)$$

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

در روابط فوق، پارامتر پاداش و  $a$  پارامتر پاداش و  $b$  پارامتر جریمه می باشد. با توجه به مقادیر  $a$  و  $b$  سه حالت را می توان در نظر گرفت. زمانی که  $a$  و  $b$  با هم برابر باشند، الگوریتم را  $L_{RP}$ <sup>۱۴</sup> می نامیم. زمانی که  $b$  از  $a$  خیلی کوچکتر باشد، الگوریتم را  $L_{REP}$ <sup>۱۵</sup> می نامیم. زمانی که  $b$  مساوی صفر باشد، الگوریتم را  $L_{RI}$ <sup>۱۶</sup> می نامیم. برای مطالعه بیشتر در باره اتوماتاهای یادگیر می توان به [4],[5],[6],[7] مراجعه کرد.

**اتوماتای یادگیر توزیع شده<sup>۱۷</sup>:** یک اتوماتای یادگیر توزیع شده شبکه ای 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 شده اند. انتخاب یک عمل توسط اتوماتای یادگیر در این شبکه باعث فعال شدن اتوماتای یادگیر متصل شده به این اتوماتای یادگیر متناظر با این عمل می 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.

مدلی که برای شبکه DLA در نظر می گیریم یک گراف است که هر یک از رئوس آن یک اتوماتای یادگیر است. وجود یال  $(LA_i, LA_j)$  در این گراف بدین معناست که انتخاب عمل  $\alpha_j^i$  توسط  $LA_i$  باعث فعال شدن  $LA_j$  می گردد. تعداد اعمال قابل انتخاب توسط  $LA_k$  بصورت  $p^k = \{p_1^k, p_2^k, \dots, p_r^k\}$  نمایش داده می شود. در این مجموعه

<sup>13</sup> Variable Learning Automata

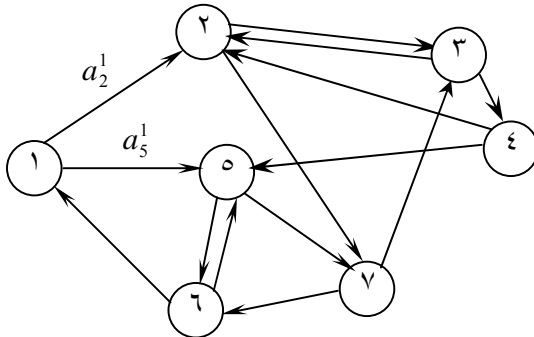
<sup>14</sup> Linear Reward Penalty

<sup>15</sup> Linear Reward Epsilon Penalty

<sup>16</sup> Linear Reward Inaction

<sup>17</sup> Distributed Learning Automata

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  $LA_m$  می شود.  $r_k$  تعداد اعمال قابل انجام توسط اتوماتای  $LA_k$  را نشان می دهد.



شکل ۲: یک اتوماتای یادگیر توزیع شده با ۷ اتوماتای یادگیر و طول بردار اعمال = ۲

$$DLA = (V, E)$$

$$V = \{LA_1, LA_2, \dots, LA_n\}$$

$$E \subset V \times V$$

$$(LA_i, LA_j) \in E$$

### ۳- خود سازماندهی در ساختار اطلاعاتی اسناد با استفاده از اتوماتای یادگیر توزیع

شده (DLA-DL)

در روش مبتنی بر اتوماتای یادگیر توزیع شده برای ایجاد یک ساختار اطلاعاتی پویا در مجموعه های بزرگ از اسناد مانند کتابخانه های دیجیتال، مجموعه اسناد و کاربران استفاده کننده از آن نقش یک محیط تصادفی را برای اتوماتاهای یادگیر موجود در DLA ایفا می کنند. خروجی DLA یک دنباله از اسناد مرور شده توسط یک کاربر هستند که مسیر حرکت کاربر را به سمت یک سند مورد نظر نشان می دهد. محیط با استفاده از این دنباله پاسخی برای DLA تولید می کند. با استفاده از این پاسخ ساختار داخلی اتوماتاهای یادگیر در اتوماتای یادگیر توزیع شده طبق الگوریتم یادگیر بروز می شود.

از آنجاییکه بیشتر اسناد با یکدیگر ارتباط ندارند ماتریس ارتباطات اسناد یک ماتریس خلوت می باشد [3]. همچنین فرض می کنیم که با افزایش تعداد اسناد، تعداد اسنادی که یک سند به آن مرتبط می باشد ثابت باقی می ماند. در سیستم پیشنهادی، به هر سند یک اتوماتای یادگیر با ساختار متغیر تخصیص داده می شود. با توجه به فرض در نظر گرفته شده، تعداد اعمال و در نتیجه اندازه بردار احتمال برای هر اتوماتای یادگیر با افزایش تعداد اسناد در مجموعه اسناد تغییر پیدا نمی کند. 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. عبارت 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 گرفته شود. بنابراین برای هر سند  $Doc_i$  یک اتوماتای یادگیر  $LA_i$  در نظر می گیریم 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  $Doc_j$  می باشد. 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 می شود.

با ورود یک کاربر به سیستم و مشاهده سند  $Doc_i$ ، اتوماتای یادگیر متناظر با آن سند یعنی  $LA_i$  فعال می شود. با حرکت کاربر از سند  $Doc_i$  به سند  $Doc_j$ ، عمل مرتبط با این انتخاب در اتوماتای  $LA_i$  انتخاب می شود و به محیط اعمال می شود. 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 می گیرد و احتمال متناظر با این عمل برابر صفر قرار داده می شود. سپس احتمال عمل حذف شده بین احتمالهای اعمال توزیع می شود 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 می شود. 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  $Doc_j$  می باشد. 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. هر چه مسیر طی شده توسط کاربر کوتاهتر باشد میزان پاداش داده شده توسط الگوریتم یادگیری به اعمال انتخاب شده در طول این بیشتر می باشد. اعمالی که قسمتی از یک دور باشند نشاندهنده حرکت اشتباه کاربر هستند و مجازات می شوند. با این مراحل هر کاربر یک رشته از اتوماتاها را فعال نموده و احتمال اعمال آنها توسط سیستم اصلاح شده است که در نتیجه ارتباطات اسناد متناظر آن اتوماتاها اصلاح می شود. بحث فوق در الگوریتم زیر خلاصه شده است.

۱- یک DLA متناظر با ساختار اسناد ایجاد کن

۲- بردار احتمالات همه اتوماتاها را مقداردهی اولیه کن

۳- بازای هر کاربر

۳-۱- مسیر حرکت او را استخراج کن

۳-۲- بازای همه اتوماتاهای مسیر حرکت مراحل زیر را انجام بده

۳-۱-۲- عمل  $j$  از اتوماتای  $i$  که منجر به انتخاب اتوماتای بعد شده را پیدا کن

۳-۲-۲- در صورتیکه این عمل قسمتی از یک دور است

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

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

۳-۲-۳- در غیر اینصورت

۳-۱-۳-۲-۳- در صورت وجود نداشتن این عمل در بردار اعمال اتوماتای  $i$

• عملی که دارای کمترین احتمال انتخاب است را حذف کن و احتمال آنرا بین بقیه اعمال توزیع کن.

• بجای آن عمل جدید انتخاب  $j$  با به بردار اعمال اتوماتای  $i$  اضافه کن

۳-۲-۳-۲- بردار اعمال اتوماتای  $i$  را طبق فرمولهای زیر اصلاح کن.

$$p_i(t+1) = p_i(t) + a[1 - p_i(t)]$$

$$p_k(t+1) = (1-a)p_k(t)$$

$$k \neq i \quad k = 1, 2, \dots, r$$

#### ۴- شبیه سازی سیستم پیشنهادی

برای انجام شبیه سازی از مدل ارایه شده در [9] برای تولید مجموعه اسناد و حرکات کاربران استفاده میشود. هر سند با یک بردار محتوا نمایش داده می شود. طول این بردار برابر تعداد موضوع های موجود در سیستم است و هر عضو این بردار میزان ارتباط سند متناظر با آن بردار را با یکی از این موضوعات را نشان می دهد. هر از موضوع ها با یک توزیع احتمالاتی خاصی بین اسناد توزیع شده اند. با تغییر پارامتر این توزیع و تغییر تعداد اسناد، سیستم های اطلاعاتی متفاوتی میتوان ایجاد نمود. در این مدل پروفایل علایق کاربران<sup>18</sup>، انگیزه<sup>19</sup> و استراتژی حرکت<sup>20</sup>ی آنها از طریق

<sup>18</sup> Interest Profile

توزیعهای آماری مدل شده اند با تغییر پارامترهای این توزیعهای آماری می توان کاربرانی با علائق، انگیزه ها و استراتژی های متفاوت ایجاد نمود. برای اطلاعات بیشتر درباره این مدل می توان به [9] مراجعه نمود. در شبیه سازیها پارامترهای تعداد موضوع ها، تعداد کاربران، علائق، انگیزه ها و پارامترهای توزیع آنها را در طول شبیه سازی ثابت در نظر گرفته شده اند. بمنظور بدست آوردن میزان تطابق ساختار اطلاعاتی پیشنهادی با ساختار اطلاعاتی ایده ال از کوریلیشن ماتریس های این دو ساختار اطلاعاتی استفاده می شود که هر چه این میزان به ۱ نزدیکتر باشد نمایانگر تطابق بیشتر این دو ساختار اطلاعاتی می باشد. در اینجا منظور از ساختار اطلاعاتی ایده ال، ساختار اطلاعاتی است که با توجه به محتواهای اسناد در مجموعه اسناد تولید شده بایستی وجود داشته باشد که بعنوان یک ساختار ایده ال و برای صحت سنجی ساختار اطلاعات بدست آمده توسط سیستم پیشنهادی استفاده میشود.

نتایج شبیه سازیهای سیستم پیشنهادی با نتایج بدست آمده برای روش بولن [2] و ANTWeb [16] مقایسه شده است. برای مقایسه این دو روش با روش پیشنهادی شبکه ای از ۱۰۰ سند در نظر گرفته شده است. سپس یادگیری ساختار اطلاعاتی در این شبکه و با روش بولن و ANTWeb و DLA-DL و با تعداد اعمال ثابت ۴۰ شبیه سازی شده است.

نتیجه کوریلیشن بین ساختار اطلاعاتی بدست آمده و ساختار ایده ال، که معیار نزدیک شدن به ساختار اصلی است برای سه روش در شکل ۳ آمده است. همانطور که دیده می شود در روش بولن بدلیل استفاده از ماتریس ارتباطات اسناد که شامل اطلاعات ارتباط بین همه اسناد می باشد کوریلیشن بدست آمده کمی بهتر است. برای اینکه در روش DLA-DL هم بتوان به این سطح از کوریلیشن رسید میتوان طول بردار اسناد را برابر تعداد کل اسناد در شبکه در نظر گرفت که در نتیجه کوریلیشنی در حد روش بولن بدست می آید. همچنین روش ANTWeb بدلیل قرار دادن پیوند به اسناد مرتبط در هر سند که امکان دنبال کردن مسیر دیگر کاربران را فراهم می کند، با تکرار کمتری به کوریلیشن های مورد قبول دست پیدا می کند.

در روش DLA-DL فرض شده است که تعداد اسناد مرتبط با یک سند محدود میباشد. که این محدودیت توسط مسوول سیستم و با توجه به حداکثر تعداد اسنادی که با یک سند میتواند رابطه داشته باشد تعیین شود. اگر این مقدار حداکثر به طور مناسب تعیین شود این محدودیت در عمل مشکلی ایجاد نمی کند. ولی در عوض روش پیشنهادی از نظر سربار محاسبات و حافظه مورد نیاز در مقایسه با دو روش ANTWeb و Bollen در شرایط بهتری قرار میگیرد. به منظور مطالعه و بررسی تاثیر پارامترهای اتوماتاهای یادگیر از آزمایش دیگری انجام گرفت. در این آزمایش رابطه بین طول بردار اعمال اتوماتاهای یادگیر مرتبط با اسناد و پارامتر یادگیری بهینه آن مورد بررسی قرار گرفت. در اینجا منظور از پارامتر یادگیری بهینه، پارامتر یادگیری است که به ازای آن کوریلیشن ساختار اطلاعات با ساختار ایده ال در کمترین تکرار به ۰,۹ می رسد. نتایج این آزمایش در جدول ۲ آمده است. همانطور که از جدول ۲ دیده می شود پارامتر یادگیری بهینه رابطه معکوس با طول بردار اعمال اتوماتاهای یادگیر مرتبط با اسناد در کتابخانه دارد. بنابراین پارامتر یادگیری اتوماتاهای یادگیر موجود در DLA باید با افزایش تعداد اسناد، اصلاح شود.

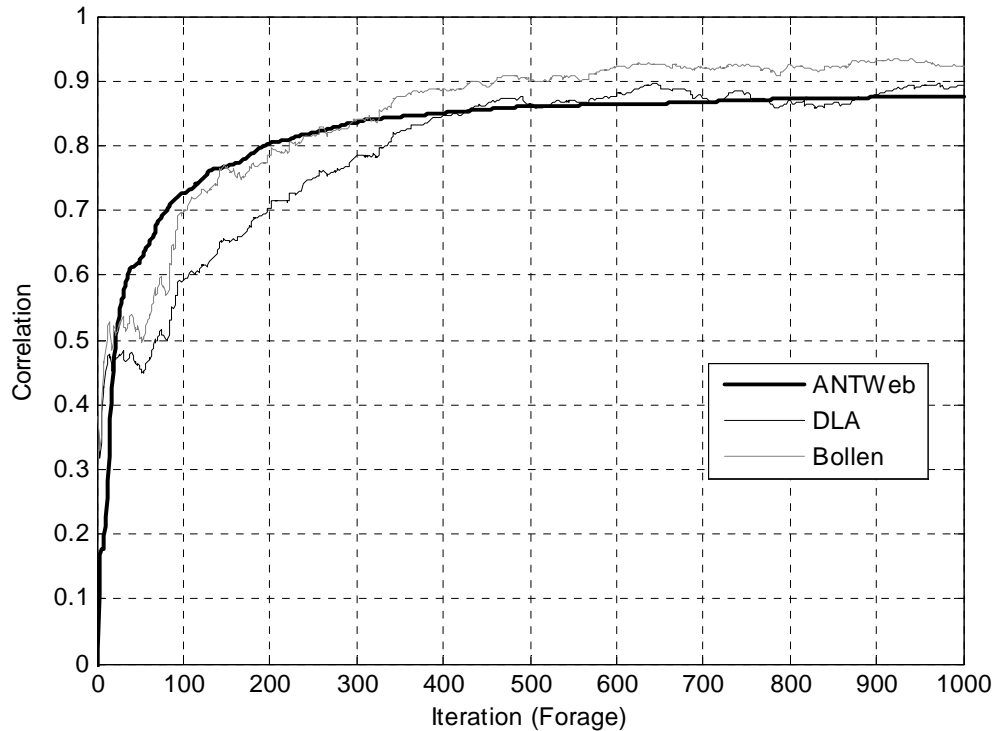
## ۶- نتیجه گیری

در این مقاله یک ساختار خود سازمانده برای مجموعه های بزرگ از اسناد مبتنی بر اتوماتای یادگیر توزیع شده (DLA-DL) ارائه گردید. این روش به جای استفاده از ماتریس ارتباطات اسناد که میتواند بسیار بزرگ باشد با استفاده

<sup>19</sup> Motivation

<sup>20</sup> Browsing Strategy

از یک اتوماتای یادگیر با تعداد محدود عمل برای هر سند، استفاده آن برای سیستمهای بزرگ را ممکن میکند. برای ارزیابی عملکرد روش پیشنهادی، این روش با روشهای بولن و ANTWeb مقایسه گردید.



شکل ۳: کورلیشن ساختار بدست آمده با ساختار اصلی با استفاده از روش DLA-DL با ۱۰۰ سند و طول بردار اعمال ۴۰ و روشهای بولن و روش ANTWeb

تعداد اسناد	۵۰	۱۰۰	۲۰۰	۵۰۰	نوع روش
ماتریس ارتباطات اسناد	۴,۵۳	۸,۱۷	۱۵,۲۴	۲۹,۶۱	ماتریس ارتباطات اسناد
DLA	۸,۳۲	۹,۱۲	۹,۹۸	۱۱,۳۲	DLA

جدول ۱: زمان (بر حسب ms) برای محاسبه شباهت بین دو سند با استفاده از اطلاعات ماتریس ارتباطات اسناد و اطلاعات بردارهای اعمال و احتمالات DLA با تعداد اعمال ثابت ۴۰

طول بردار اعمال	۱۰	۲۰	۴۰	۸۰	۱۶۰	پارامتر یادگیری
۰,۰۰۲	-	-	-	-	۳۱۰۰	۰,۰۰۲
۰,۰۰۵	-	-	-	۱۴۰۰	۱۷۰۰	۰,۰۰۵
۰,۰۱	۲۰۰	۲۲۵	۵۵۰	۸۰۰	۱۷۰۰	۰,۰۱



۳۹۰۰	۱۹۰۰	۴۵۰	۱۷۵	۱۵۰	۰,۰۲
-	۱۹۰۰	۴۵۰	۱۵۰	۱۲۵	۰,۰۳
-	-	-	۱۷۵	۱۵۰	۰,۰۵

جدول ۲: مقایسه تعداد تکرارهای لازم برای رسیدن به سطح کورلیشن ۰,۹ برای اتوماتاهای یادگیر اسناد با طول بردار اعمال مختلف و پارامترهای یادگیری مختلف.

## مراجع

- [1] F. Heylighen "Design of A Hypermedica Interface Translating between Associative and Formal Representations", International Journal of Man-Machine Studies 35, pp. 491-515, 1990.
- [2] F. Heylighen, J. Bollen. "Hebbian Algorithms for a Digital Library Recommendation System", Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'02), 2002.
- [3] 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, pp. 15-44, 1995.
- [4] S. Lakshmirarahan, "Learning Algorithms: Theory and Applications", New York: Springer-verlag, 1981.
- [5] M. R. Meybodi and S. Lakshmirarahan, "On a Class of Learning Algorithms which have Symmetric Behavior under Success and Failure", Lecture Notes in Statistics, Berlin: Springer-Verlag, pp. 145-155, 1984.
- [6] P. Mars, J. R. Chen, and R. Nambir, "Learning Algorithms: Theory and Applications in Signal Processing", Control, and Communication", CRC Press Inc., 1996.
- [7] K. S. Narendra and K. S. Thathachar, "Learning Automata: An Introduction", New York: Prentice-Hall, 1989.
- [8] J. Kleinberg. "Authoritative sources in a hyperlinked environment", Proceedings of 9th ACM-SIAM Symposium on Discrete Algorithms 1998.
- [9] 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. 5, may 2004.
- [10] R. Colley. "Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data". Ph.D. Thesis, University of Minnesota, May 2000.
- [11] C. Shahabi, A.M. Zarkesh, J. Adibi and V. Shah. "Knowledge Discovery from User's Web-page Navigation", Proceedings 7th IEEE International Conference On Research Issues in Data Engineering, pp. 20-29, 1997.
- [12] Johan Bollen. "A Cognitive Model of Adaptive Web Design and Navigation: A Shared Knowledge Perspective", Phd Thesis, Vrije Universiteit Brussel, 2001.
- [13] F. Heylighen. "Bootstrapping Knowledge Representations: from Entailment Meshes via Semantic Nets to Learning webs", Kybernetes, 30 (5/6), pp. 691-722, 2001.
- [14] 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, p.15-44.
- [15] S. Brin and L. Page. "The Anatomy of a Large-scale Hypertextual Web Search Engine" In 7th WWW Conference, Brisbane, Australia, April 1998.
- [16] W. Teles, L. Weigang, C. Ralha. "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.
- [17] S. Saati and M. R. Meybodi, " Learning Automata and Its Application to Web Page Ranking", Technical Report, Computer Engineering and Information Technology Department, Amirkabir University, Tehran, Iran, 2005.