کلیه حقوق مادی و معنوی این مقاله متعلق به دبیرخانه دائمی کنفرانس داده کاوی ایران است و درج آن فقط در سایت رسمی کنفرانس داده کاوی ایران مجاز بوده و هرگونه انتشار غیر از آن، خلاف عرف، اخلاق و قانون و مستوجب پیگیری است.

۱۰۸۷

کاوش ساختار وب با استفاده از اتوماتای یادگیر وگرامراحتمالی

زهره اناری[۱] ؛ محمد رضا میبدی[۲] ؛ بابک اناری[۳]

چکیده

یکی از روش‌های داده کاوی در وب، کاوش استفاده از وب می‌باشد. هدف ازاین کاوش، استخراج اطلاعات مفید از داده‌هایی است که از تعامل کاربران در هنگام استفاده از وب به‌دست می‌آید. با استخراج این اطلاعات، می‌توان میزان ارتباط بین صفحات وب را تعیین و در نتیجه عملیاتی مثل خوشه بندی و رتبه بندی صفحات وب را انجام داد. هدف در این مقاله، تعیین میزان ارتباط بین صفحات وب با استفاده از گرامر احتمالی و اتوماتای یادگیر می‌باشد. در الگوریتم پیشنهادی ابتدا از مسیرهای حرکتی ذخیره شده کاربران در لاگ فایل‌ها، گرامر احتمالی تولید شده و قوانین انجمنی استخراج می‌گردند. قوانین انجمنی استخراجی توسط اتوماتای یادگیر مورد ارزیابی قرار گرفته و میزان ارتباط بین صفحات وب توسط اتوماتای یادگیر تعیین می‌گردد. نتایج شبیه‌سازی‌ها از طریق مقایسه با روشهای AntWeb، Bollen و E- DLA نشان می‌دهد که روش پیشنهادی از کارایی قابل ملاحظه‌ای برخوردار است.

کلمات کلیدی

کاوش استفاده از وب ، اتوماتای یادگیر، گرامر احتمالی

[۱] دستیار علمی ، دانشکده کامپیوتر و فناوری اطلاعات ، دانشگاه پیام نور ، zanari323@yahoo.com
[۲] استاد و عضو هیات علمی دانشگاه صنعتی امیرکبیر mmeybodi@aut.ac.ir
[۳] عضو هیات علمی دانشگاه، دانشگاه آزاد اسلامی واحد شبستر، anari322@yahoo.com

Web Structure Mining using Learning Automata and Probabilistic Grammar

Zohreh Anari; Mohammad Reza Meybodi; Babak Anari

ABSTRACT

One of the data mining techniques on the web is web usage mining. The purpose of this mining, extracting useful information from the user interaction data that can be obtained when using the web. By extracting this information, we can determine the relationship between the web pages and then we can do operations such as clustering and ranking web pages. The purpose of this article is to determine the relationship between web pages, using probabilistic grammar and learning automata. In the proposed algorithm first probabilistic grammar generated and association rules are extracted from the user navigation paths stored in log files. Extracted association rules evaluated by learning automata and the relationship between web pages by learning automata is determined. The simulation results compared with Antweb, Bollen and E-DLA shows that the proposed method has a considerable efficiency.

KEYWORDS

Web usage mining, Learning Automata, Probabilistic Grammar

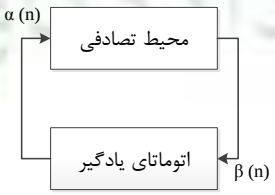
کنفرانس داده‌کاوی ایران

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 صفحات وب، به وسیله تکنیک‌های داده کاوی را کاوش وب 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 صفحات وب است. در سطح استفاده از وب، هدف، کشف اطلاعات مفید از داده - هایی است که از تعامل کاربران در هنگام استفاده از وب بدست می‌آید. بیشتر تکنیک‌های موجود برای کشف ساختار ارتباطی بین صفحات وب از اطلاعات موجود درلاگ فایل‌ها استفاده می‌کنند. لاگ فایل‌ها، فایل‌هایی هستند که مجموعه درخواستهای کاربران به صفحات وب را در خود ذخیره می‌کنند. این اطلاعات به همان ترتیبی که به سرور می‌رسند، ذخیره شده و می‌توان دوباره آنها را بازیابی‌کرد. روش‌های موجود با 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 روش‌ها عبارتند از: استفاده از قوانین یادگیری، مثل استفاده از قانون یادگیری هب در روش بولن [7]، استفاده از سیستم مورچه‌ها مثل AntWeb [19]، استفاده از اتوماتای یادگیر توزیع شده، [3,17,18]، استفاده از زنجیر مارکف [8] و استفاده از گرامرهای احتمالی ابرمتن [5]. ضعف روش گزارش شده مبتنی بر اتوماتای یادگیر توزیع شده (E-DLA) که بعدا در این مقاله مفصل‌تر به آن اشاره می‌شود، کارایی پایین آن در مقایسه با روش Ant Web است [2]. دو روش AntWeb و Bollen با وجود مزایایی که نسبت به روشهای قدیمی‌تر دارند، به دلیل اینکه از ماتریس ارتباطات بین صفحات وب استفاده می‌کنند، برای استفاده در مجموعه‌های بزرگ و قابل گسترش مناسب نمی‌باشند. نقطه ضعف استفاده از زنجیره مارکف، نیاز به محاسبات بالا به واسطه محاسبه توان n ام ماتریس انتقال می‌باشد. هر چند با روش‌های فشرده‌سازی می‌توان هزینه محاسبه را تا حدی کاهش داد [8]. نقطه ضعف استفاده از گرامرهای احتمالی ابرمتن نیز داشتن پیچیدگی O(n) به واسطه استفاده ازالگوریتم پیمایش در عمق می‌باشد. در این مقاله یک روش جدید و مبتنی بر اتوماتای یادگیر توزیع شده که از گرامرهای احتمالی استفاده خواهد کرد، برای تعیین میزان ارتباط بین صفحات وب پیشنهاد می‌گردد. در این روش به هر صفحه وب یک اتوماتای یادگیر اختصاص داده می‌شود که وظیفه‌اش یادگیری ارتباطات آن صفحه با صفحات دیگر می‌باشد. کارایی الگوریتم پیشنهادی از طریق مقایسه با سه روش Ant web وBollen و E-DLA مورد ارزیابی قرار خواهد گرفت. ادامه مقاله به این صورت سازماندهی شده است: در بخش ۲ اتوماتای یادگیر و اتوماتای یادگیر توزیع شده به اختصار شرح داده می‌شوند. در بخش۳ گرامر احتمالی ابرمتن شرح داده می‌شوند. در بخش ۴ الگوریتم پیشنهادی و در بخش ۵ نتایج آزمایشات ارایه می- گردد. بخش پایانی نتیجه‌گیری می‌باشد.

۲. اتوماتای یادگیر

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 و عمل خود را برای مرحله بعد انتخاب می‌کند. شکل۱ ارتباط بین اتوماتای یادگیر و محیط را نشان می‌دهد.



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط.

محیط: محیط را می توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه ورودی ها، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه‌خروجی ها و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمالهای جریمه می باشد. 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، $\beta(n)$ می‌تواند به طور

گسسته یک مقدار از مقادیر محدود در فاصله [0,1] و در محیط از نوع S، $\beta(n)$ متغیر تصادفی در فاصله [0,1] است. c_i احتمال اینکه عمل α_i نتیجه نامطلوب داشته باشد، می‌باشد. در محیط ایستا ، مقادیر c_i بدون تغییر باقی می‌مانند حال آنکه در محیط غیرایستا این مقادیر در طی زمان تغییر می-کنند. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و متغیر تقسیم بندی می‌گردند. در ادامه به شرح مختصری درباره اتوماتای یادگیر با ساختار متغیر و اتوماتای یادگیر توزیع شده که در این مقاله از آنها استفاده شده است، می‌پردازیم.

۱.۲ اتوماتای یادگیر با ساختار متغیر

اتوماتای یادگیر با ساختار متغیر توسط۴ تایی $\{\alpha, \beta, p, T\}$ نشان داده می‌شود که در آن $\alpha \equiv \{\alpha_1,\alpha_2,...,\alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta \equiv \{\beta_1,\beta_2,...,\beta_m\}$ مجموعه ورودی‌های اتوماتا، $p \equiv \{p_1, p_2,...,p_r\}$ 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 α_i در مرحله n 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 $p_i(n)$ همواره مساوی یک باقی بماند. الگوریتم زیر نمونه‌ای از الگوریتم های یادگیری خطی در اتوماتای با ساختار متغیر است .

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)\, p_j(n) \qquad \forall j \quad j \neq i \end{aligned} \qquad (۱)$$

الف- پاسخ مطلوب

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

ب- پاسخ نامطلوب

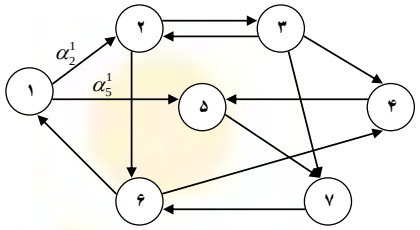
در روابط فوق a ، پارامتر پاداش و b پارامتر جریمه می‌باشد. با توجه به مقادیر a و b سه حالت را می‌توان در نظر گرفت. زمانی که a و b باهم برابر باشند، الگوریتم را L_{RP} می‌نامیم. زمانی که b از a خیلی کوچک باشد، الگوریتم را $L_{R\epsilon P}$ می‌نامیم. زمانی که b مساوی صفر باشد، الگوریتم را L_{RI} می‌نامیم. برای مطالعه بیشتر درباره اتوماتاهای یادگیرمی‌توان به [9,12,13,16,20] مراجعه کرد.

۲.۲ اتوماتای یادگیر توزیع شده (DLA)[۴]

یک اتوماتای یادگیر توزیع شده، شبکه‌ای از اتوماتاهای یادگیر است که برای حل یک مسئله خاص با یکدیگر همکاری دارند. در این شبکه از اتوماتاهای یادگیر همکار در هر زمان، تنها یک اتوماتا فعال است. مدلی که برای شبکه DLA در نظر گرفته می‌شود یک گراف جهت دار است که هر یک از رئوس آن یک اتوماتای یادگیر است. وجود یال (LA_i, LA_j) در این گراف به این معناست که انتخاب عمل α_j^i توسط LA_i باعث فعال شدن LA_j می‌گردد. تعداد اعمال قابل انتخاب توسط LA_k بصورت $p^k = \{p_1^k, p_2^k,..., p_r^k\}$ نمایش داده می‌شود. در این مجموعه عدد p_m^k نشان

[4] Distributed Learning Automata

دهنده احتمال مربوط به عمل α_m^k است. انتخاب عمل α_m^k توسط LA_k باعث فعال شدن LA_m می‌شود. r_k تعداد اعمال قابل انجام توسط اتوماتای LA_k را نشان می‌دهد. شکل ۲ مثالی از یک اتوماتای یادگیر توزیع شده با هفت اتوماتای یادگیر را نشان می‌دهد. برای کسب اطلاعات بیشتر درباره اتوماتای یادگیر توزیع شده و کاربردهای آن می‌توان به [4,1 ,14,15] مراجعه کرد .



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

شکل ۲: یک اتوماتای یادگیر توزیع شده با هفت اتوماتای یادگیر

3. گرامراحتمالی ابرمتن[5] (HPG)

یک جلسه کاری کاربر، دنباله‌ای از صفحات ملاقات شده توسط همان کاربر است، به طوری که فاصله زمانی بین ملاقات دو صفحه متوالی بیش از یک حد زمانی معین نباشد (پژوهشگران نشان داده اند که این زمان تقریبا ۲۵،۵ دقیقه یا ۳۰ دقیقه می‌باشد). می‌توان جلسات کاری کاربران استنتاج شده از لاگ داده را توسط زبانی به نام HPA[10] بیان کرد که توسط HPG[5] بدست می‌آید . HPG یک گرامر با قاعده احتمالی است به طوری‌که بین مجموعه غیرپایانه‌ها و پایانه‌ها یک نگاشت یک به یک وجود دارد. هر غیرپایانه نشان دهنده یک صفحه وب و هر قانون تولید نشان دهنده یال بین صفحات است. در این مدل، سیستم ابرمتن به صورت گراف جهت دار G = (N,E) در نظر گرفته می‌شود. به طوری که N تعداد نودها و E یال‌های این گراف را نشان می‌دهد. اگر A , B دو صفحه وب باشند و کاربر از صفحه A به صفحهB حرکت کند، یک یال یک جفت (A,B)$\in E$ خواهد بود. تعداد زمان هایی که یک دنباله از دو صفحه در جلسه کاربری ظاهر شده (مانندنباله $A \to B$) معادل تعداد دفعاتی است که لینک مورد نظر (یال) ملاقات شده است. این مدل بصورت ۵ تایی $\Phi(H, M, S, F, \Gamma)$ تعریف می‌شود که در آن :

H : مجموعه نودها، $H = \{h_1, ..., h_m\}$ که هر یک از اعضای آن معرف یک صفحه وب خواهد بود.

M : یک ماتریس $m \times m$ است به طوری‌که : $M = \{p(h_i, h_j)\}$ ، $(h_i, h_j) \in H$ ، $0 \le p(h_i, h_j) \le 1$ ، $\sum_{k=1}^{m} p(h_i, h_k) = 1$

S : حالت شروع جلسه کاری کاربر و F : حالت خاتمه جلسه کاری کاربر

Γ : تابعی از $H \times H$ به [0,1] است. یعنی احتمال حرکت از صفحه h_i به صفحه h_j را نشان می‌دهد.

اگر صفحه h_i را به عنوان صفحه شروع و صفحه h_j را به عنوان صفحه مقصد در نظر بگیریم، یک قانون انجمنی $h_i \to h_j$ به این صورت تعریف می‌شود:

[5] Hypertext Probabilistic Grammar

زمانی که کاربر به صفحه h_i می‌رسد لینک بعدی که او انتخاب می‌کند با یک احتمال معین h_j خواهد بود. احتمال لینک انتخاب شده به عنوان اطمینان[۶] توصیف شده و به این صورت تعریف می‌شود:

$$P(h_i, h_j) = \frac{R_j^i}{R^i} \qquad (۳)$$

R_j^i: تعداد دفعاتی که صفحه h_j بلافاصله بعد از صفحه h_i قرار گرفته است و R^i: تعداد تمامی صفحاتی که بعد از صفحه h_i ملاقات می شوند .
احتمال رشته تولید شده توسط این گرامر، از حاصل ضرب اطمینان همه قوانین مورد استفاده برای اشتقاق این رشته به دست می‌آید. به عبارت دیگر اگر رشته تولید شده توسط این گرامر به صورت $<h_{i1},...,h_{ir}>$ باشد، احتمال رشته تولید شده به صورت رابطه زیر تعیین می‌شود:

$$P(<h_{i1},...,h_{ir}>) = \prod_{k=1}^{k=r-1} P(h_{ik}, h_{i(k+1)})$$

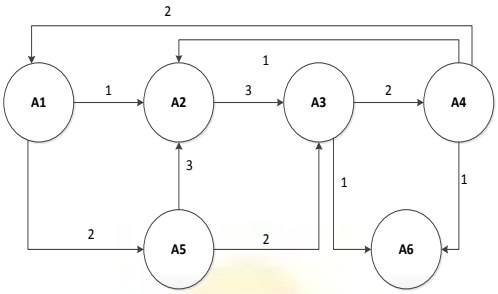
در این مدل، دنباله های حرکتی کاربران استخراج شده از لاگ داده به عنوان گرامر احتمالی ابرمتن مدل سازی می‌شوند، به طوری که رشته های تولید شده با احتمال بالا مسیرهای حرکتی را که مورد علاقه کاربران بوده را نشان می‌دهد[5]. برای مدل سازی جلسات کاربران در سیستم ابرمتن از مدل پیشنهادی[5] استفاده می‌کنیم.

مثال۱ : اگر حرکت های کاربر در یک جلسه کاری کاربر به صورت جدول ۱ باشد، در این مثال ما ۶ جلسه کاربری داریم. سیستم ابرمتن آن بصورت شکل ۳ خواهد بود. به عنوان مثال درشکل3 وزن یال $A_2 \rightarrow A_3$ برابر ۳ می‌باشد زیرا این دنباله حرکتی درجلسات کاربران ۳ بارظاهر شده است. همچنین با توجه به شکل ۳ احتمال رشته $(A_1 \rightarrow A_2 \rightarrow A_3)$ به صورت زیر محاسبه می‌شود :

جدول ۱: دنباله‌ای از حرکت های کاربران

Session ID	User trail
1	$A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_4$
2	$A_1 \rightarrow A_5 \rightarrow A_3 \rightarrow A_4 \rightarrow A_1$
3	$A_5 \rightarrow A_2 \rightarrow A_4 \rightarrow A_6$
4	$A_5 \rightarrow A_2 \rightarrow A_3$
5	$A_5 \rightarrow A_2 \rightarrow A_3 \rightarrow A_6$
6	$A_4 \rightarrow A_1 \rightarrow A_5 \rightarrow A_3$

[6] Confidence



شکل ۳: ایجاد سیستم ابرمتن از حرکتهای کاربر

ابتدا از حرکت های کاربر یک سیستم ابر متن می سازیم (شکل ۳). بعد از ایجاد سیستم ابرمتن می‌توان به تولید رشته ها از آن پرداخت. الگوریتمی وجود دارد که استخراج این رشته ها را انجام می‌دهد. الگوریتم استخراج رشته ها از سیستم ابرمتن را استخراج قوانین انجمنی می‌گویند [5]. این الگوریتم در ادامه بحث در بخش 1.3آورده شده است.

1.3 الگوریتم استخراج قوانین انجمنی از سیستم ابرمتن

هدف از استخراج قوانین انجمنی [5] این است، که در گراف جهت دار موجود تمامی قوانینی را تولید کنیم که مقدار احتمال قوانین استخراجی از مقدار نقطه برش[7] بیشتر باشد. این الگوریتم از هر نود این گراف شروع کرده و رشته هایی که مقدار احتمال آنها از نقطه برش بیشتر باشند را تولید می کند. اگر CR [8] نشان دهنده قانون انتخاب باشد و x یک دنباله باشد. forward-neigh(x) همسایه رو به جلو و ملاقات نشده را نشان می‌دهد. با یک دنباله x و یک لینک e ، x+e الحاق دو رشته x و e را نشان می‌دهد. این الگوریتم حالت خاصی از الگوریتم DFS[9] می‌باشد. از هر یال این گراف درخت DFS ساخته شده و هر شاخه این درخت متناظر با یک قانون انتخاب(CR) است. هر درخت DFS مشخص کننده همه قوانین انتخابی است، که یال شروع شده از ریشه این درخت، اولین یال آن را نشان می‌دهد. کاوش یک شاخه زمانی پایان می‌پذیرد که به نودی برسیم که همه یال‌های آن از قبل ملاقات شده باشد یا مقدار اطمینان دنباله پایین تر از حد آستانه C باشد. بعلاوه کاوش هر شاخه مستقل از هر شاخه دیگر است. الگوریتم استخراج قوانین انجمنی در شکل ۴ آورده شده است.

کنفرانس داده‌کاوی ایران

[7] Cut-Point
[8] Candidate Rules
[9] Depth-First-Search

شکل ۴ : الگوریتم استخراج قوانین انجمنی از سیستم ابر متن

Algorithm 1.1(Modified_DFS (G, C))
1. **begin**
2. **for** each $\{e \in E: C_e \geq C\}$

3. Explore (e, C_e)

4.**end for**

5. **end.**

Algorithm 1.2(Explore ($trail, C_{trail}$))

1. **begin**
   $CR = CR \cup trail$

3. **for** each $(forw - neigh(trail))$

4. **if** $(C_{fn} \times C_{trail} \geq C)$ **then**

5. Explore$(trail + f_n\ , C_{trail})$

6. **end for**
7. **end**

مثال2: در مثال ۱ با فرض این که مقدار نقطه برش $\lambda = 0.33$ باشد در این صورت نحوه استخراج قوانین انجمنی با شروع از A_1 به صورت زیر انجام می‌شود. از A_1 می‌توان به A_2 و A_5 رفت ولی $P(A_1 \to A_2) = \frac{1}{3} < \lambda$ است، پس مسیر $A_1 \to A_2$ را نادیده می‌گیریم. حال از صفحه A_5 کار را ادامه می‌دهیم. از این صفحه می‌توان به صفحات A_2 و A_3 رفت. مسیر $A_1 \to A_5 \to A_3$ نیز رد می‌شود زیرا خواهیم داشت $P(A_1 \to A_5 \to A_3) = \frac{2}{3} \times \frac{2}{5} < \lambda$ پس مسیر استخراجی با شروع از A_1 عبارت است از: $A_1 \to A_5 \to A_2$ جدول۲ تمامی قوانین استخراجی را برای شکل۳ به دست می‌دهد و برای هر کدام از نودها (صفحات وب) نیز داریم:

$$A_1: \begin{cases} p(A_1 \to \\ p(A_1 \to \end{cases}$$

جدول۲ : قوانین انجمنی استخراج شده از شکل۳

Session ID	User Session
1	$A_4 \rightarrow A_1$
2	$A_5 \rightarrow A_2 \rightarrow A_3$
3	$A_5 \rightarrow A_3$
4	$A_3 \rightarrow A_4$
5	$A_2 \rightarrow A_3 \rightarrow A_4$
6	$A_1 \rightarrow A_5 \rightarrow A_2$

λ به نقطه برش معروف است. هر چه مقدار λ بزرگتر باشد، طول قوانین استخراج شده نیز کمتر خواهد بود و بالعکس هرچه این مقدار کمتر باشد، طول قوانین استخراجی نیز بیشتر خواهد بود. برای مقدار کمتر از۰,۳ ، تعداد تکرارهای الگوریتم بالا است ولی برای مقدار بزرگتر از ۰,۳ این تکرارها کم می‌شود. پیچیدگی این الگوریتم خطی و به تعداد نودها وابسته است. به عبارت دیگر با افزایش تعداد نودها، تعداد تکرارهای الگوریتم نیز به صورت خطی رشد می‌کند. با افزایش تعداد نودها، زمان اجرایCPU به صورت توانی رشد می‌کند. همچنین با افزایش درجه گراف، تعداد قوانین تولید شده نیز بیشتر خواهد بود[6].

٤. الگوریتم پیشنهادی

صفحات وب و کاربران استفاده کننده ازآن نقش یک محیط تصادفی را برای اتوماتای یادگیر موجود در DLA ایفا می‌کنند. خروجی DLA یک دنباله از صفحات وب مرورشده توسط کاربر هستند که مسیر حرکت کاربر را به سمت یک صفحه وب مورد نظر نشان می‌دهد. محیط با استفاده از این دنباله پاسخی برای DLA تولید می‌کند. با استفاده از این پاسخ ساختار داخلی اتوماتای یادگیر دراتوماتای یادگیر توزیع شده طبق الگوریتم یادگیری بروز می‌شود. در الگوریتم پیشنهادی از اتوماتای یادگیر با ساختار متغیر استفاده شده است. برای این کار در درروش پیشنهادی برای هرصفحه p_i یک اتوماتای یادگیر LA_i در نظر می‌گیریم. اگر n تعداد صفحات وب باشد در این حالت اتوماتا n-1 عمل دارد. زمانی که کاربر از صفحه p_i به صفحه p_j حرکت می‌کند j امین عمل از اتوماتای LA_i درDLA فعال شده و به محیط اعمال می‌شود. در صورتی که عمل انتخاب شده k امین عمل اتوماتای LA_i باشد(یعنی $\alpha_k^i = j$) احتمال متناظر این عمل یعنی p_k^i به عنوان میزان ارتباط صفحات i و j در نظر گرفته می-شود. در این الگوریتم ابتدا یک لاگ فایل برای ذخیره حرکات کاربران در نظر می‌گیریم. سپس حرکات کاربران در این فایل ذخیره می‌شود، بعد از پر شدن این لاگ‌فایل، یک سیستم ابر متن از این فایل ساخته می‌شود و قوانین انجمنی با مقدار نقطه برش با استفاده از الگوریتم و1.2و۱,۱(شکل ۴) از آن استخراج می‌شوند. این قوانین استخراجی، مسیرهایی را نشان می‌دهند که شدت دنباله (مقدار احتمال) آنها بالاتر است، در نهایت فقط به این قوانین استخراجی توسط اتوماتای یادگیر پاداش‌داده می‌شود. دوباره لاگ فایل پاک شده حرکت های جدید در آن وارد شده و دوباره همین روال انجام می‌گیرد. الگوریتم پیشنهادی بصورت زیر است:

ورودی : دنباله های حرکتی کاربران

خروجی : استخراج قوانین انجمنی

۱- یک DLA متناظر با ساختار اسناد ایجاد کن.

۲- بردار احتمالات اتوماتاهای یادگیردر DLA را مقدار دهی اولیه کن.

۳- یک لاگ فایل جهت ذخیره حرکت های کاربران ایجاد کن.

۴- تا زمانی‌که لاگ فایل پر نشده، حرکت های کاربران را در آن ذخیره کن.

۵- اگرلاگ فایل پر شد، از اطلاعات آن یک سیستم ابرمتن بساز.

۶- با استفاده از الگوریتم ۱،۱و1.2 (شکل ۴) تمامی قوانین انجمنی با نقطه برش معلوم را از این سیستم ابرمتن استخراج کن .

۷- برای هر قانون استخراجی انجام بده

۱-۷- برای هر حرکت $D_k \rightarrow D_m$ (حرکت ازصفحه D_k به صفحه D_m) کاربر در طول قانون استخراجی، انجام بده

2-۷- بردار احتمال اعمال اتوماتای یادگیر LA_k را طبق الگوریتم یادگیری زیر بروزکن

$$p_m(n+1) = p_m(n) + a[1 - p_m(n)]$$
$$p_j(n+1) = (1-a)\, p_j(n) \qquad j \neq m \quad \forall j$$

۸- لاگ فایل را پاک کرده و برو به ۴

۹- پایان الگوریتم.

شکل ۵ : الگوریتم پیشنهادی

۵. ارزیابی کارایی الگوریتم پیشنهادی و نتایج آزمایشات

برای انجام شبیه‌سازی از مدل ارایه شده در[11] برای تولید مجموعه صفحات وب و حرکات کاربران استفاده می‌شود. هرصفحه با بردار محتوای $C_n = \left[cw_n^1 \quad cw_n^2 \quad \cdots \quad cw_n^M \right]$ نشان داده می‌شود که درآن M تعداد موضوعات در سیستم می‌باشد. هر عضو این بردار (cw_n^i) میزان ارتباط صفحه متناظر با آن بردار را با یکی از این موضوعات نشان می‌دهد. با استفاده از بردار محتوای هر کدام از صفحات، شباهت بین هر دو صفحه موجود در سیستم محاسبه می‌شود. ماتریس شباهت به دست آمده به عنوان ماتریس ارتباطات ایده ال بین صفحات وب در شبیه سازی‌ها به منظور ارزیابی کارایی روش پیشنهادی استفاده می‌شود. در این مدل توزیع موضوعات بین صفحات وب بصورت توزیع نرمال و پروفایل علاقه کاربران بصورت توزیع قانون توانی در نظر گرفته شده است که با تغییر پارامتر این توزیع و تعداد صفحات، سیستم‌های اطلاعاتی متفاوتی می‌توان ایجاد نمود.

جدول ۳ : پارامترهای شبیه سازی

حد آستانه ایجاد اتصال	۰,۷
تعداد کاربران	۱۰۰۰۰و2۰۰۰0
تعداد صفحات وب	۳۰
تعداد موضوع ها	۵
T_c :مقدار ثابت صفحه اولیه سایت در موضوعات مختلف	۰.۲
ΔM_t^c :ضریب ثابت کاهش اشتیاق کاربر	-
ΔM_t^v :ضریب متغیر کاهش اشتیاق کاربر	-
$\propto_u$:پارامتر توزیع قانون- توانی (توزیع احتمال علایق کاربران)	۱
$\emptyset$:ضریب پاداش دریافتی از مشاهده یک صفحه	1.2
λ: ضریب جذب اطلاعات از یک صفحه توسط یک کاربر	۰.۵
μ_m :میانگین توزیع نرمال ΔM_t^v	5.79
σ_m :واریانس توزیع نرمال ΔM_t^v	0.25
μ_t :میانگین توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع خاص	-
α_p :پارامتر توزیع قانون- توانی (توزیع احتمال‌های وزن‌های مطالب برای هر صفحه)	۳
σ :واریانس توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع	0.25
θ : ضریب کاهش علاقه کاربر	۱
حداقل اشتیاق کاربر برای ادامه جستجو	0.2

در این مدل، انگیزه و استراتژی حرکتی کاربران نیز از طریق توزیع های آماری مدل شده‌اند. با تغییر پارامترهای این توزیع‌های آماری می‌توان کاربرانی با علایق، انگیزه‌ها و استراتژیهای متفاوت ایجاد نمود. در شبیه‌سازی‌ها، پارامترهای تعداد موضوع ها، تعداد کاربران، علایق، انگیزه‌ها و پارامترهای توزیع آنها را در طول شبیه‌سازی ثابت در نظر گرفته‌ایم. برای ارزیابی الگوریتم پیشنهادی از معیاری به نام کورولیشن استفاده خواهیم نمود. کورولیشن دو مجموعه داده مانند X ,Y بصورت رابطه ۶ محاسبه می‌شود که در آن N تعداد داده ها است. مجموعه X ساختار ایجاد شده توسط مدل شبیه سازی [11] و Y نیزساختار بدست آمده توسط الگوریتم پیشنهادی را نشان می‌دهد.

$$Corr\ (X,Y)=\frac{Cov\ (X,Y)}{\sigma x\sigma y}=\frac{\sum XY-(\sum X\sum Y)/N}{\sqrt{(\sum X^2-(\sum X)^2/N)(\sum Y^2-(\sum y)^2/N)}} \qquad (۵)$$

$$X=\{P_{ij}\,|\,i,j=1,2,\cdots,n,\quad i\neq j\},$$
$$Y=\{P'ij|i,j=1,2,\ldots,n,\quad i\neq j\},$$

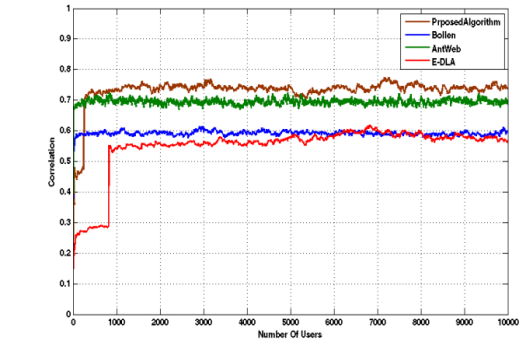
$$P_{ij}=\frac{(d_{ij})^{-1}}{\sum_{k=1}^{n}(d_{ik})^{-1}}$$

$$P'_{ij}=probability\quad of\ \ action\ \ j\ \ from\ \ DLA\ (i) \qquad (٦)$$

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

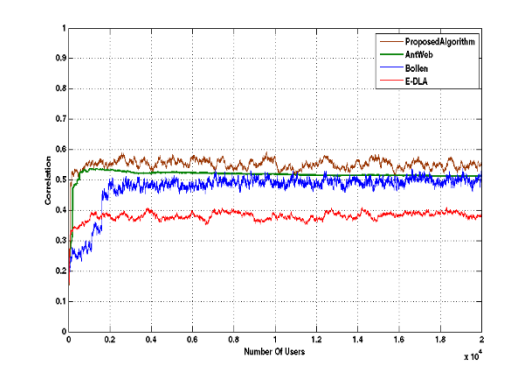
در رابطه فوق d_{ij} فاصله اقلیدسی بین دو صفحه i, j از یکدیگر در مدل شبیه‌سازی می‌باشد. پارامترهای شبیه‌سازی الگوریتم پیشنهادی مطابق با مقادیر جدول ۳ می باشد.

آزمایش ۱: شکل ۶ نتیجه مقایسه اجرای الگوریتم پیشنهادی (Proposed Algorithm) را با الگوریتم‌های AntWeb و Bollen و E-DLA نشان می‌دهد. در این شبیه سازی تعداد کاربران استفاده کننده از وب ۱۰۰۰۰، تعداد موضوعات مرتبط با هر سند ۵ ، تعداد صفحات وب ۳۰ و مقدار نقطه برش نیز $\lambda = ۰,۰۱$ در نظر گرفته شده است.



شکل ۶ : مقایسه نتایج الگوریتم های موجود

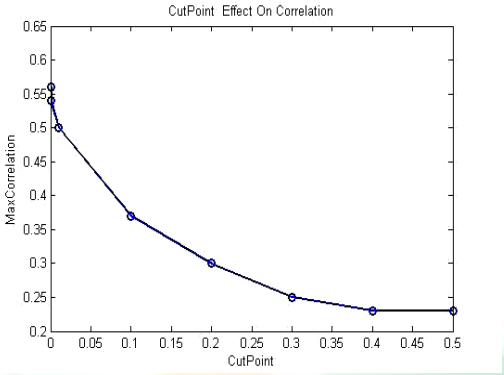
آزمایش ۲: شکل ۷ مقایسه نتیجه اجرای الگوریتم پیشنهادی را با الگوریتم‌های AntWeb و Bollen و E-DLA نشان می‌دهد. در این شبیه سازی تعداد کاربران استفاده کننده از وب ۲۰۰۰۰ ، تعداد موضوعات مرتبط با هر سند ۵ ، تعداد صفحات وب ۳۰ و مقدار نقطه برش نیز $\lambda = ۰,۰ ۱$ در نظر گرفته شده است.



شکل ۷: مقایسه نتایج الگوریتمهای موجود

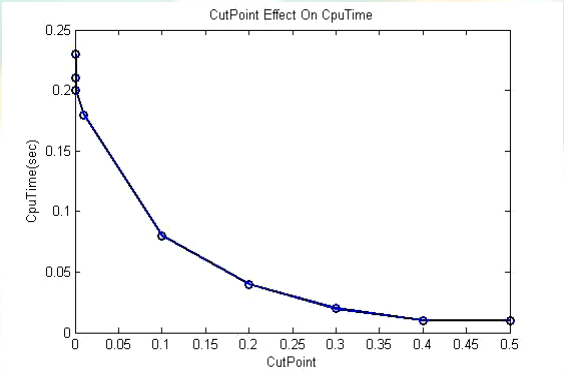
همان‌طوری که از شکل‌های ۶ و۷ مشخص است، الگوریتم پیشنهادی دارای کارایی بالایی می‌باشد و می‌توان از این الگوریتم برای کشف ساختار ارتباطی بین صفحات وب بهره جست. با استفاده از مقدار نقطه برش λ ، در هر بار قوانین انجمنی استخراج شده و به این قوانین انجمنی پاداش داده شده است. نتایج آزمایشات ۱ و۲ نشان می‌دهد که الگوریتم پیشنهادی در تعداد تکرارهای کمتری به کورولیشن مورد نظر می‌رسد.

آزمایش ۳ : در این آزمایش، تاثیر پارامتر نقطه برش بر روی کورولیشن نشان داده شده است. λ به نقطه برش معروف است. همان طوری که از شکل ۸ مشخص است، مقدار نقطه برش تاثیر زیادی در مقدار کورولیشن دارد. به عبارت دیگر با افزایش مقدار نقطه برش، مقدار کورولیشن نیز کمتر می شود. علت این امر را می‌توان در طول قوانین استخراجی دانست. هر چه مقدار λ بزرگتر باشد، طول قوانین استخراج شده نیز کمتر خواهد بود و بالعکس هرچه این مقدار کمتر باشد، طول قوانین استخراجی نیز بیشتر خواهد بود. نتایج این شبیه سازی در شکل ۸ آمده است.



شکل ۸: تاثیر نقطه برش برروی‌کورولیشن

آزمایش ۴ : تاثیرنقطه برش بر روی زمان و کارایی الگوریتم پیشنهادی نیز انجام شد وتاثیر این پارامتر بر روی الگوریتم مشخص شد. نتایج این شبیه سازی در شکل ۹ آمده است. شکل 9 تاثیر مقدار نقطه برش را در زمان اجرای الگوریتم نشان می‌دهد. همان طوری که از شکل 9 مشخص است، مقدار نقطه برش تاثیر زیادی در زمان اجرا دارد. به عبارت دیگر با افزایش مقدار نقطه برش، مقدار زمان اجرا نیز کمتر می‌شود. علت این امر را می‌توان در طول قوانین استخراجی دانست. هر چه مقدار λ بزرگتر باشد، طول قوانین استخراج شده نیز کمتر خواهد بود و بالعکس هرچه این مقدار کمتر باشد، طول قوانین استخراجی نیز بیشتر خواهد بود.



شکل 9: تاثیر نقطه برش بر روی زمان اجرا (استخراج قوانین انجمنی و پاداش به اتوماتا)

٦. نتیجه‌گیری

در این مقاله یک روش جدید برای تعیین میزان ارتباط بین صفحات وب با استفاده از اتوماتای یادگیر توزیع شده که از گرامرهای احتمالی ابر-متن استفاده می‌نماید، پیشنهاد گردید. کارایی الگوریتم پیشنهادی با روش‌های Ant Web و Bollen و E-DLA مورد مقایسه قرار گرفت و نشان داده شد که کورولیشن الگوریتم پیشنهادی بهتر از این روش‌ها است. مزیت الگوریتم پیشنهادی در این است که الگوریتم پیشنهادی به دلیل در نظر گرفتن قوانین انجمنی به حرکت‌هایی با کیفیت بالا پاداش می‌دهد. بنابراین الگوهایی با کیفیت پایین در نظر گرفته نشده و این امر باعث افزایش کارایی الگوریتم پیشنهادی نسبت به سایر الگوریتم ها می‌شود. الگوریتم پیشنهادی به دلیل استفاده از الگوریتم پیمایش در عمق (DFS)

دارای پیچیدگی زمانی O(n) بود که در آن n تعداد نودها می‌باشد. بنابر این اگر تعداد نودهایی که درلاگ فایل قرار می‌گیرند، n باشد و K تعداد دفعات تکرار باشد، در آن صورت هزینه الگوریتم پیشنهادی برابر با $O(nK)$ خواهد بود. همچنین تاثیر پارامتر نقطه برش را در زمان اجراء و کورولیشن بحث کردیم و دیدیم که الگوریتم پیشنهادی ما تحت تاثیر این پارامتر قرار دارد به عبارت دیگر تعیین مقدار اولیه نقطه برش حائز اهمیت است. از کاربردهای الگوریتم پیشنهادی می‌توان به استخراج قوانین انجمنی، پیش بینی صفحات وب مورد علاقه کاربران، فراهم کردن وب سایتهای تطبیقی برای کاربران، خوشه بندی کاربران و رتبه بندی صفحات وب اشاره کرد.

۷. تقدیم و تشکر

از استاد بزرگوار جناب آقای دکتر محمد رضا میبدی که با زحمات بی‌شائبه خویش ما را در کسب علم و معرفت یاری نمودند تشکر و قدردانی می‌نماییم.

این مقاله مستخرج از طرحی است که با حمایت مالی دانشگاه پیام نور با عنوان کاوش ساختار وب با استفاده از اتوماتای یادگیر وگرامر احتمالی به تصویب رسیده است.

کنفرانس داده‌کاوی ایران

۸. منابع

[1] M. Alipour; M. R. Meybodi, *"Solving probabilistic traveling sales man problem using distributed learning automata"*, Proc. of 11th Annual CSI Computer Conference of Iran, Fundamental Science Research Center(IPM), Computer Science Research Lab, Tehran, Iran, pp.673-67, Jan. 24-26, 2006.

[۲] B. Anari; M. R. Meybodi, *"A new method based on distributed learning automata for determining web documents structure"*, Proc. of 12th Ann. Int. CSI Computer Conf. CSICC 2007, Tehran, Iran, pp. 2276- 2281, Feb. 20-22, 2007.

[۳] A. Baradaran Hashemi ; M.R., Meybodi, *"Web usage mining using distributed learning automata"*, Proc. Int. Conf. on Computer Engineering Department, Technical Report, 2005.

[۴] H. Beigy; M. R. Meybodi, *"Utilizing distributed learning automata to solve stochastic shortest Path Problem",* Int. Jour. of Uncertainty, Fuzziness and Knowledge-based Systems, World Scientific Publishing Company, Vol. 14, No. 5, pp. 591-617, October 2006 .

[۵] J. Borges ; M. Leven, *"Data Mining of user navigation patterns",* Proc. of the Web Usage Analysis and User Profiling, vol. 1, PP. 31-36, 1999.

[۶] J. Borges; M. Levene, "Mining association rules in hypertext databases*"*, Proc. of the 4th Int. Conf. On Knowledge Discovery and Data Mining, pp. 149-153, August 1998.

[۷] F. Heylighen; J. Bollen, *"Hebbian Algorithm for a Digital Library Recommendation System",* Proc. Int. Conf. on Parallel Processing Workshops (ICPPW'02) IEEE, 2002.

[۸] Z. Jianhan, *"Mining web site link structures for adaptive web site navigation and search",* Ph.D. Thesis, university of Ulster at jordanstown, October 2003.

[۹] S. Lakshmivarahan, *"Learning algorithms: theory and applications",* New York: Springer- Verlag, 1981.

[۱۰] M. Levene; G. Loizou, *"A Probabilistic approach to navigation in hypertext",* Information Sciences, pp. ۱۱۴-۱۶۵, 1999.

[۱۱] J. Liu ; S. Zhang ; J. Yang, *"Characterizing web usage regularities with information foraging agents",* IEEE Trans. in knowledge and data engineering, Vol. 16, No. 5, May 2004.

[۱۲] P. Mars; J.R Chen; R. Nambir, *"Learning Algorithms: Theory and Applications in Signal Processing"*, Control, and Communication, CRC Press Inc, 1996.

[۱۳] M. R. Meybodi; S. Lakshmivarahan, *"On a class of Learning Algorithms which have Symmetric Behavior under Success and Failure"*, Lecture Notes in Statistics, Berlin: SpringerVerlag, pp. 145-155, 1984.

[۱٤] M. R. Meybodi; 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).

[۱۵] M. R. Meybodi; H. Beigy, *"Solving stochastic path problem using distributed learning automata"*, Proc. of 6th Annual International CSI Computer Conference (CSICC2001), Isfahan, Iran, pp. 70-86, Feb. 20- 22, 2001.

[۱۶] K. S. Narendra; M. A. L Thathachar, *"Learning automata: An introduction,"* Prentice Hall, 1989.

[۱۷] S. Saati; M.R. Meybodi, *"Document ranking using distributed learning automata,"* Proc. of 11th Annual CSI Computer Conf. of Iran, Fundamental Science Research Center(IPM), Computer Science Research Lab, Tehran, Iran, pp.467- 473, Iran, May 24-26 2006.

[۱۸] S. Saati; M.R Meybodi, *"A self-organizing model for document structure using distributed learning automata"*, Proc. The Second Int. IEEE/WIC Conf. on Information and knowege Technology (IKT2005), Tehran, Iran, May 24-26, 2005.

[۱۹] W. Teles; L. Weigang; C. Ralha, *"AntWeb-The adaptive web server based on the ants behavior",* Proc. IEEE/WIC Int. Conf. on Web Intelligence (WI'03), PP.558-564, 2003.

[۲۰] M. A. L Thathachar Baskar; R. Harita, *"Learning automata with changing number of actions"*, IEEE Trans. on System, Man and Cybernetic, vol. SMC-17, No. 6, Nov. 1987.