کشف زیر گرافهاي متصل وب با استفاده از اتوماتاي یادگیر توزیع شده

[1]راهبه مجتهدی صفاری; [2]محمدرضا میبدی

چکیده

استفاده از تکنیک های داده کاوی به منظور استخراج اتوماتیک اطلاعات از اسناد وب را کاوش وب می گویند. یک مسأله بنیادین در زمینه کاوش وب، کشف زیر گراف های متصل اطلاعاتی وب بر اساس درخواست کاربر می باشد. مسأله کشف زیر گراف های متصل وب، خصوصاً در مواردی که کاربر الگوریتم های کاوش وب را بر روی تنها بخشی از گراف وب اجرا می کند مؤثر است و منجر به هدایت بهتر کاربر در امر جستجو و پیشنهاد اسناد مشابه با یک سند بر اساس علاقه کاربر می گردد. در این مقاله از اتوماتای یادگیر توزیع شده برای حل مسأله کشف زیر گرافهای متصل وب استفاده می شود. برای این منظور در ابتدا یک الگوریتم مبتنی بر اتوماتای یادگیر توزیع شده برای تعیین ساختار ارتباطی بین اسناد وب پیشنهاد می گردد. الگوریتم پیشنهادی کشف ساختار در مقایسه با روش های موجود دارای کارایی بالاتر می باشد. در مرحله بعد با استفاده از ساختار ارتباطی حاصل و روش خوشه بندی *kmeans* یک الگوریتم جدید برای کشف زیر گرافهای متصل وب ارائه می گردد. الگوریتم پیشنهادی کشف زیر گراف متصل تنها بر روی خوشه هایی که اسناد مورد درخواست کاربر را شامل می شوند، اجرا می گردد، بنابراین در مقایسه با روش های موجود دارای سرعت بالاتر می باشد. نتایج شبیه سازی ها نشان می دهد زیر گراف های به دست آمده از الگوریتم پیشنهادی کشف زیر گراف متصل، به دلیل استفاده همزمان از اطلاعات ساختار پیوندی و نحوه پیمایش کاربران در هنگام تعامل با وب، در مقایسه با روش های موجود کیفیت بالاتری دارند.

کلمات کلیدی

کشف زیر گراف متصل، اتوماتای یادگیر، اتوماتای یادگیر توزیع شده، خوشه بندی *k-means* درجه ارزش.

Web Connection Subgraph Discovery Using Distributed Learning Automata

R. Mojtahedi Saffari; M. R. Meybodi

Abstract

Web mining is the use of data mining techniques to discover and extract information from the web documents. The connection subgraph discovery based on user's request is a fundamental issue in exploring the web. The connection subgraph discovery problem particularly when a user performs the web mining algorithms on only a part of the web graph, is efficient and leads to better user's navigation in search and proposes documents similar with the one that satisfy the user's interest. In this paper a solution based on distributed learning automata for discovering the web connection subgraph is proposed. For this propose in the first part of this paper an algorithm based on distributed learning automata for determining the structure of web documents is proposed. It is shown that the proposed algorithm has higher performance in comparison to existing methods. In the second part of this paper, a hybrid algorithm obtained by combining the algorithm proposed in the first part and k-means clustering method, a new algorithm for discovering the web connection subgraph is proposed. The proposed algorithm uses the clusters that only include requested user's documents and therefore has higher performance over the to existing methods. Simulation results have shown that the subgraphs generated by the proposed subgraph discovery algorithm because of using the link structure and user's navigation information of web documents at the same time, have higher quality in comparison to existing methods.

Keywords

Connection Subgraph Discovery, Learning Automata, Distributed Learning Automata, K-means Clustering, Nearness Score.

---

[1]. عضوهیأت علمی دانشگاه آزاد اسلامی واحد لاهیجان، دانشکده برق و کامپیوتر، دانشگاه آزاد واحد لاهیجان، لاهیجان، ایران، mojtahedi@liau.ac.ir

[2]. عضو هیأت علمی دانشگاه صنعتی امیرکبیر، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، mmeybodi@aut.ac.ir

۱- مقدمه

یکی از مسائل مطرح در زمینه کاوش گراف وب، کشف زیرگراف‌های متصل اطلاعاتی مبتنی بر پرس وجوی کاربر می‌باشد. مسأله به اینگونه است که کاربر درخواست خود را در قالب پرس وجو مطرح نموده و بر اساس نوع پرس وجو و الگوریتم به کار رفته برای کشف زیر گراف، زیرگرافی که قوی ترین اتصال در بین اسناد پرس وجوی آن وجود دارد، ایجاد می شود. به عبارت دیگر، هدف پیدا کردن اسناد میانی و مرکزی می باشد که به طور مستقیم ویا غیر مستقیم با تمام اسناد پرس وجو و یا بیشتر آنها در ارتباط قوی تر از سایر اسناد می باشند. از زیر گرافهای به دست آمده می توان جهت شناسایی اجتماعات گوناگون وب، هدایت بهتر کاربران در امر جستجو و پیشنهاد اسناد مشابه با یک سند بر اساس علایق کاربر استفاده نمود، همچنین با استفاده از زیر گرافهای حاصل می توان الگوریتم های کاوش وب مبتنی بر پرس وجوی کاربر را سازماندهی نمود ودر امر شخصی سازی وب به کار برد.

یکی از مسایل مهم در زمینه کشف زیر گراف های متصل تعیین معیار درجه ارزش بین دوسند می باشد. درجه ارزش بین دو سند کمیتی است که میزان ارتباط بین دو سند را مشخص می نماید. این معیار، ملاک ارزیابی در کشف قوی ترین اتصال در بین اسناد گراف به شمار می رود و در مقالات، با توجه به روش های گوناگون به کار رفته برای کشف زیر گراف متصل، به گونه های متفاوت تعریف می شود. تا کنون دو روش برای کشف زیر گرافهای متصل گزارش شده است،که بطور مستقیم به حل این مسأله می پردازند. روش اول که توسط *Christos Faloutsos* و همکارانش در سال ۲۰۰۴ ارائه شده است مسأله کشف زیر گرافهای متصل را به مسأله ماکسیمم جریان در شبکه های الکتریکی نگاشت می کنند[۷]. در این روش ضمن در نظر گرفتن مفهوم حداکثر جریان منتقله در طول مسیر[۱۲]، مسیرهایی انتخاب می شوند که اسناد آن همپوشانی زیادی با اسناد موجود در مسیرهای انتخابی قبلی، دارند.

یکی از مشکلاتی که این روش دارد، این است که زیر گراف متصل را فقط به ازای هر دو سند پرس وجو پیدا می کند. همچنین در این روش زیر گراف به دست آمده، حساس به ترتیب قرارگیری اسناد پرس و جو می باشد، یعنی بسته به اینکه سندپرس وجوی انتخابی سند مبدأ یا مقصد باشد، زیر گراف تولید شده تغییر می کند. روش*Ceps*، در جهت بهبود روش[۷] مطرح شده و زیرگراف متصل بین چند سند پرس وجوی کاربر را پیدا می کند[۱۸]. در این روش، از مفهوم احتمال حالت پایدار به منظور تعیین درجه ارزش بین دو سند استفاده می گردد. احتمال حالت پایدار، احتمالی است که یک عامل تصادفی در حین گردش تصادفی خود در گراف پس از رسیدن به یک سند خاص به دست می آورد[۱۸,۱۷]. در روش *Ceps*، به منظور محاسبه احتمال حالت پایداربرای تمام اسنادموجود در گراف، از یک ماتریس ورودی که ساختار ارتباطی بین اسناد را نشان می دهد، استفاده می گردد. با افزایش تعداد اسناد وب سایز این ماتریس افزایش یافته و محاسبه احتمالات حالت پایدار دارای پیچیدگی زمانی بالایی می شود. بنابراین روش *Ceps* قابلیت گسترش برروی گرافهای بزرگ را دارا نمی باشد.

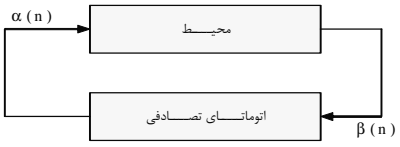
روشهای مطرح شده برای کشف زیر گرافهای متصل[۱۸,۷]، راه حلی است که بر روی گرافهای بدون جهت اجرا می شوند، همچنین در این روشها تنها از اطلاعات ساختار پیوندی بین اسناد به منظور تعیین درجه ارزش بین اسناد استفاده می گردد. به منظور رفع این مشکلات، در روش پیشنهادی از اتوماتای یادگیر توزیع شده برای حل مسأله کشف زیر گرافهای متصل استفاده می کنیم. در اتوماتای یادگیر توزیع شده به هر سند یک اتوماتای یادگیر تخصیص داده می شود، که در هر لحظه بر اساس ساختار پیوندی موجود در بین اسناد و نحوه استفاده کاربران از اسناد در هنگام تعامل با وب، ارتباطات بین اسناد را یاد می گیرد. روش پیشنهادی ضمن در نظر گرفتن جهت در الگوریتم پیشنهادی کشف زیر گراف متصل، به دلیل استفاده همزمان از اطلاعات ساختار پیوندی و اطلاعات استفاده کاربران از وب، زیر گرافهایی با کیفیت بالاتر و پیچیدگی زمانی پایین تر تولید می کند.

ادامه مقاله بدین صورت سازماندهی شده است: در بخش۲، اتوماتاهای یادگیر، اتوماتای یادگیر توزیع شده و یک روش برای تعیین ساختار ارتباطی بین اسناد وب پیشنهاد می گردد. در بخش۳، روش خوشه بندی*k-means* مورد بررسی قرار می گیرد. در بخش۴ الگوریتم پیشنهادی برای کشف زیر گرافهای متصل جهتدار مطرح می گردد. در بخش۵ نتایج شبیه سازیها مطرح شده است. بخش پایانی شامل نتیجه گیری می باشد.

۲- اتوماتاهاي يادگير

اتوماتای یادگیر یک مدل انتزاعی است که به طور 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 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 $E=\{\alpha,\beta,c\}$ نشان دادکه درآن $\alpha=\{\alpha_1,\alpha_2,\ldots,\alpha_r\}$ مجموعه

ورودی ها، $\beta=\{\beta_1,\beta_2,\ldots,\beta_r\}$ مجموعه خروجیها و $c=\{c_1,c_2,\ldots,c_r\}$ مجموعه احتمالات جریمه می‌باشد. c_i نشان دهنده احتمال نامطلوب بودن سیگنال تقویتی محیط در پاسخ به عمل α_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 به روز رسانی وضعیت اتوماتای یادگیر (که با اطلاع از عمل انتخاب شده و سیگنال تقویتβ، 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 تقسیم‌می‌گردند. شکل ۱ یک اتوماتای یادگیر تصادفی را نمایش می دهد.



شکل۱: اتوماتای یادگیر تصادفی

در این مقاله از اتوماتای یادگیر با ساختار متغیر استفاده شده است که در ادامه معرفی می شود. اتوماتای یادگیر با ساختار متغیر توسط چهارتایی $\{\alpha,\beta,p,T\}$نشان داده می‌شودکه در آن $\alpha=\{\alpha_1,\alpha_2,\ldots,\alpha_r\}$، مجموعه اعمال اتوماتای یادگیر، $\beta=\{\beta_1,\beta_2,\ldots,\beta_r\}$، مجموعه ورودی های اتوماتای یادگیر، $p=\{p_1,p_2,\ldots,p_r\}$ بردار احتمال انتخاب هر یک ازعمل‌هاوT، $p(n+1)=T[\alpha(n),\beta(n),p(n)]$ الگوریتم یادگیری اتوماتای یادگیر می‌باشد. الگوریتمهای یادگیری متنوعی برای اتوماتای یادگیر ارائه شده است که در ادامه یک الگوریتم‌یادگیری خطی برای اتوماتای یادگیر بیان می‌گردد. فرض کنید اتوماتای یادگیر در مرحله nاُم اقدام α_i خود را انتخاب نموده و محیط ارزیابی خود را توسط سیگنال تقویتی$\beta(n)$ به اتوماتای یادگیر اعلام کند. با استفاده از الگوریتم یادگیری خطی، اتوماتای یادگیر بردار احتمال انتخاب اقدام‌های خود را مطابق روابط ۱ بروز می‌کند.

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

در رابطه بالا، a پارامتر پاداش و b پارامتر جریمه می‌باشد. اگر a و b با هم برابر باشند، الگوریتم L_{R-P}، اگر b از a خیلی کوچکتر باشد، الگوریتم $L_{R\varepsilon P}$ و اگر b صفر باشد، الگوریتم L_{R-I} نام دارد. اتوماتای یادگیری که در بالا معرفی شد، دارای تعداد اقدام های ثابتی می باشد.در بعضی از کاربرد ها به اتوماتای یادگیر با تعداد اقدام متغیر نیاز می باشد[۱۶].

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 به صورت تصادفی انجام می شود. نحوه فعالیت این اتوماتای یادگیر به صورت زیر است. اتوماتای یادگیر برای انتخاب یک اقدام در زمانn، ابتدا مجموع احتمال اقدامهای فعال خود $(K(n))$ را محاسبه و بردار $\hat{P}(n)$ را مطابق رابطه (۲) 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 انتخابشده α_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 مطابق رابطه (۵) بروز می کند. سپس اتوماتای یادگیر بردار احتمال اقدامهای خود $P(n)$را با استفاده از بردار $\hat{P}(n+1)$و طبق رابطه (۶) بروز رسانی می کند.

$$K(n) = \sum_{\alpha_i \in V(n)} p_i(n) \qquad (۲)$$

$$\hat{P}_i(n) = prob[\alpha(n) = \alpha_i | \alpha_i \in V(n)] = \frac{p_i(n)}{K(n)} \qquad (۳)$$

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

در صورت پاسخ مطلوب:

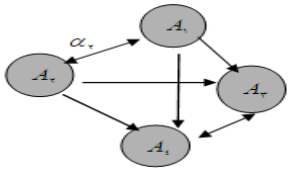
$$\begin{aligned} &\hat{P}_i(n+1) = \hat{P}_i(n) + a_i.(1-\hat{P}_i(n)) \\ &j \neq i \ \forall j \ \ \hat{P}_j(n+1) = \hat{P}_j(n) - a_i.\hat{P}_j(n) \end{aligned} \qquad (۴)$$

در صورت پاسخ نامطلوب:

$$\hat{p}_i(n+1)=(1-b).\hat{p}_i(n) \qquad \alpha(n)=\alpha$$
$$\hat{p}_j(n+1)=\frac{b}{r-1}+(1-b)\hat{p}_j(n) \quad \alpha(n)=\alpha_i,\ \forall j\ j\neq i \qquad (۵)$$
$$i,\alpha_i\in V(n)\ for\ all\ \ P_i(n+1)=\hat{P}_i(n+1).K(n) \qquad (۶)$$
$$P_j(n+1)=P_j(n) \qquad j,\alpha_j\notin V(n)$$

۲-۱- اتوماتاهاي يادگير توزيع شده

اتوماتای یادگیر توزیع‌شده شبکه‌ای از چند اتوماتای یادگیر است که برای حل یک مساله مشخص با یکدیگر همکاری می‌کنند. یک اتوماتای یادگیر توزیع‌شده را می‌توان به صورت یک گراف جهت‌دار نمایش داد، به صورتیکه 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 اتوماتا یکی از اعمال خود را انتخاب می‌کند، اتوماتایی که در دیگر انتهای یال متناظر با آن عمل قرار دارد، فعال می‌شود. درهر لحظه فقط یک اتوماتای یادگیر در اتوماتای یادگیر توزیع‌شده فعال می باشد. به عنوان مثال در شکل۲، یک اتوماتای یادگیر توزیع شده با چهار اتوماتای یادگیر را نشان می دهد. برای اطلاعات بیشتردرباره اتوماتای یادگیر توزیع شده می توان به مراجع [۱۶,۱۵,۱۴,۳]، مراجعه کرد.



شکل۲: اتوماتای یادگیر توزیع شده

۲-۲- الگوريتم پيشنهادي تعيين ساختار ارتباطي اسناد وب با استفاده از اتوماتاي يادگير توزيع شده

به منظور تعیین ساختار اسناد وب، یک اتوماتای یادگیر توزیع شده متناظر با اسناد وب ایجاد می کنیم. در روش پیشنهادی، اسناد وب و کاربران استفاده کننده از آن نقش یک محیط تصادفی را برای اتوماتاهای یادگیر موجود در اتوماتای یادگیر توزیع شده ایفا می کنند. خروجی یک اتوماتای یادگیر توزیع شده، یک دنباله از اسناد وب مرور شده توسط یک کاربر هستند که مسیر حرکت کاربر را به سمت سند وب مورد نظر نشان می دهد. 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 می شود.

در این مقاله برای تعیین ساختار ارتباطی بین اسناد وب، در یک شبکه با n سند، از یک اتوماتای یادگیر توزیع شده با n اتوماتای یادگیر و تعداد اقدامهای متغیر [۱۶]، که هر یک حداکثر $n-1$ اقدام دارند، استفاده می شود. برای هر اتوماتای یادگیر در هر زمان تنها زیرمجموعه ای از اقدامهایش فعال و می تواند قابل استفاده باشد[۱۶]. در ابتدای کار تمامی اقدامهای اتوماتای یادگیر در اتوماتای یادگیر توزیع شده غیر فعال می باشند. با استفاده از اطلاعات ذخیره شده در لاگ فایل کاربران که مسیر حرکت کاربران را به هنگام ورود به شبکه تا زمان خروج، در خود ثبت می کند، اتوماتاهای یادگیر موجود در اتوماتای یادگیر توزیع شده فعال می شوند. با ورود کاربر به شبکه و حرکت آن از سند i به سند j، اقدام j از اتوماتای متناظر با سند i فعال شده و به مجموعه اقدامهای فعال اتوماتای i اضافه می گردد، سپس اتوماتای i یک اقدام از مجموعه اقدامهای فعال خود را به طور تصادفی انتخاب می نماید، در صورتیکه اقدام انتخاب شده توسط اتوماتای i، متناظر با سند j (همان سندی که کاربر در لاگ فایل بعد از سند i مشاهده می کند) باشد، پاسخ محیط مطلوب است و اتوماتای i به اقدام انتخابی خود پاداش می دهد، در غیر این صورت پاسخ محیط نامطلوب بوده و اتوماتای i باید اقدام انتخابی را به میزان پارامتر b طبق الگوریتم یادگیری جریمه نماید. در بخش شبیه سازیها نشان خواهیم داد که الگوریتم یادگیری L_{RI} در اجرای الگوریتم پیشنهادی تعیین ساختار، نسبت به سایر الگوریتمهای یادگیری، دارای کارایی بالاتر است. بنابراین در آزمایشها مقدار پارامترجریمه(b) را صفردر نظر می گیریم، همچنین ضریب پاداش دریافتی توسط اقدام انتخابی اتوماتا، در هر مرحله بر اساس اطلاعات لاگ فایل کاربران بطور پویا تعیین می گردد. روند اجرای الگوریتم پیشنهادی به صورت شکل۳می باشد.

۱- یک اتوماتای یادگیر توزیع شده متناظر با ساختار اسناد وب ایجاد کن

۲- بردار احتمالات اتوماتاهای یادگیر موجود در اتوماتای یادگیر توزیع شده را مقدار دهی اولیه کن.

۳- تمام اقدامهای اتوماتاهای یادگیر موجود در اتوماتای یادگیر توزیع شده غیر فعال می باشند.

۴- به ازای هر کاربر موجود در لاگ فایل انجام بده

۳-۱-به ازای هرحرکت به صورت $m \to n$ در طول مسیر انجام بده

۳-۱-۱-مجموعه اقدامهای فعال اتوماتای متناظر با سند m را بروز کن.

۳-۱-۲- بردار احتمال اتوماتای متناظر با سند m را مطابق روابط زیر بروز کن.

در صورتیکه اقدام انتخابی اتوماتای m ، اقدام n باشد(پاسخ محیط مطلوب)

$$\hat{P}_n^m(n+1) = \hat{P}_n^m(n) + a_n^m(n).(1-\hat{P}_n^m(n))$$

$$\hat{P}_k^m(n+1) = \hat{P}_k^m(n) - a_n^m(n).\hat{P}_k^m(n) \qquad \forall k \qquad k \neq n$$

$$a_n^m(n) = \frac{\hat{p}_n^m(n)}{\hat{p}_n^m(n)+1}$$

در غیر اینصورت(پاسخ محیط نامطلوب) بردار احتمال اتوماتای متناظر با سندm در لحظه $n+1$ تغییری نمی کند.

شکل۳: الگوریتم پیشنهادی تعیین ساختار ارتباطی اسناد وب

با توجه به الگوریتم پیشنهادی مقدار ضریب پاداش هر اتوماتا به اقدام انتخابی خود، بر اساس مقدار احتمال اقدام انتخابی اتوماتا تعیین می گردد. به عبارت دیگر در طول مسیر حرکت کاربر، اقدامهای انتخابی که دارای مقدار احتمال بالاتر می باشند، پاداش بیشتری دریافت می کنند. تنظیم اتوماتیک ضریب پاداش سبب می شود که با افزایش تعداد اسناد وب نیاز به تنظیم مجدد ضریب پاداش نداشته باشیم. فرق الگوریتم پیشنهادی با الگوریتمهای پیشین مبتنی بر اتوماتای یادگیر توزیع شده[۱۵,۲,۱]، 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 ودر هر مرحله همان اقدامی را انتخاب می کند که کاربر از آن عبور می کند و به این ترتیب تمام اسنادی که در طول مسیر حرکت کاربر می باشند، پاداش می گیرند. 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 به صورت تصادفی اقدام خود را از میان مجموعه اقدامهای فعال انتخاب نموده و عمل بروزرسانی مقدار احتمال این اقدام را بر اساس مراجعه به لاگ فایل و مقایسه با آن، انجام می دهد. ویژگی دیگر الگوریتم مطرح شده این است که در آن فرآیند محاسبه دور مانند[۱۵] وجود ندارد و به همین دلیل در مقایسه با الگوریتم پیشنهادی[۱۵] دارای سربار محاسباتی کمتری می باشد.

۳- روش خوشه بندي *K-means*

الگوریتم *k-means*، معمولترین روش خوشه بندی می باشد، که توسط *Mac Queen* در سال ۱۹۶۷ ارائه شد. نامگذاری این الگوریتم بر گرفته از این نکته است که این روش، به صورت اتوماتیک یک مجموعه با n داده در فضای d بعدی را در داخل k گروه خوشه بندی می کند. این الگوریتم در ابتدا k مرکز خوشه را به طور تصادفی از میان n داده انتخاب نموده و فاصله بین هر داده و مرکز خوشه ها را، توسط یک تابع هدف تعیین می نماید. به عنوان مثال می توان از تابع L۲-*norm* که جمع توان های خطای بین داده ها و مراکز خوشه های آنها(واریانس داخلی خوشه)راتعریف می کند، به عنوان تابع هدف استفاده کرد. روند کلی اجرای این الگوریتم به صورت زیر است.

$$E(c) = \sum_{j=1:k} \sum_{x_i \in C_j} \left\| x_i - c_j \right\|^2 \qquad (۷)$$

۱- k مرکز خوشه را انتخاب کن.

۲-تمام داده ها را به نزدیکترین خوشه منتسب کن.

۳- مرکز خوشه ها را دوباره محاسبه کن.

۴- مراحل ۲ و۳ را تازمانیکه مرکز خوشه ها تغییر نکند، تکرارکن.

شکل ۴: الگوریتم کلی روش *k-means*

داده هایی که در این مقاله از آنها استفاده می شود، به صورت عددی است و تعداد آنها زیاد می باشد، بنابراین از روش خوشه بندی *k-means*، برای خوشه بندی اسناد استفاده می کنیم. برای مطالعه بیشتر در مورد روشهای موجود برای خوشه بندی می توان به مراجع[۱۹,۱۱,۱۰,۹,۶,۴] مراجعه کرد.

۴- الگوريتم پيشنهادي براي كشف زير گراف متصل جهتدار با استفاده از اتوماتاي يادگير توزيع شده

زمان اجرای الگوریتمهای موجود برای کشف زیر گراف متصل، در مورد گرافهای بسیار بزرگ، طولانی می باشد. روش ارائه شده در این بخش، کوششی در جهت رفع این مشکل می باشد. بنابراین با خوشه بندی گراف اصلی و اجرای الگوریتم کشف زیر گراف متصل تنها بر روی خوشه هایی که شامل اسناد پرس وجو می باشند، می توان به میزان قابل توجهی زمان پاسخ را کاهش داد. بنابراین با استفاده ازروش خوشه بندی*k-means* و ساختار ارتباطی به دست آمده از بخش۲-۲، الگوریتم پیشنهادی برای کشف زیر گراف متصل را ارائه می کنیم. الگوریتم پیشنهادی، در واقع فرم اصلاح شده روش*Ceps* می باشد[۱۸]، با این تفاوت که در الگوریتم پیشنهادی برای کشف زیر گراف متصل مسئله جهتدار بودن را در محاسبه درجه ارزش بین اسناد لحاظ می کنیم. به منظور اینکه الگوریتم پیشنهادی بر روی گرافهای جهتدار قابل اجرا باشد، اسناد پرس وجو را به دو دسته اسناد پرس وجوی مبدأ و مقصد تقسیم بندی می کنیم، بنابراین از آرایه ای بنام *Flag برای نگه داری اطلاعات مربوط به اسناد پرس و جو* استفاده می گردد. در این آرایه اسناد پرس وجوی مبدأ ومقصد به ترتیب مقادیر ۱+و۱- را دارا می باشند. با دریافت اطلاعات این آرایه توسط کاربر، زیر گراف متصل جهتدار متناسب با پرس وجوی کاربر تولید می گردد.

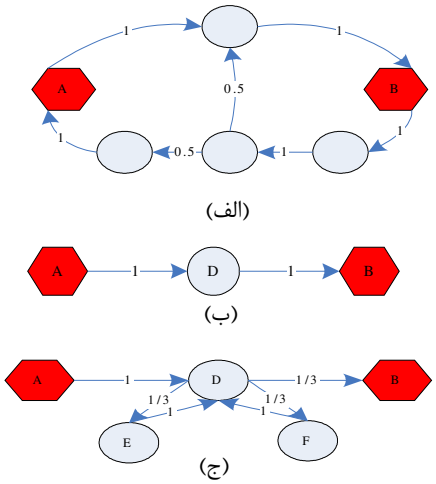
همانطور که بیان شد محاسبه درجه ارزش بین دو سند مهمترین معیار، در کشف زیر گرافهای متصل می باشد. در الگوریتم پیشنهادی کشف زیر گراف متصل بر خلاف روشهای از مفهوم احتمال گریز برای محاسبه درجه ارزش بین اسناد استفاده می کنیم. احتمال گریز از سند i به سندj، احتمالی است که یک ذره تصادفی باشروع از سندi، قبل از اینکه دوباره به سند i برگردد، سندjرا ملاقات کند و به صورت زیر تعریف می گردد.

$$ep_{i,j} = \sum_{k=1}^{n} c \times p_{i,k} \times v_k(i,j) \qquad (۸)$$

$$V_k(i,j) = \sum_{t=1}^{n} c \times P(k,t) \times V_t(i,j) \quad k \neq i,j \quad V_i(i,j) = 0, \quad V_j(i,j) = 1 \qquad (۹)$$

در رابطه(۹)، $v_k(i,j)$، احتمال رفتن ذره تصادفی از سندk به سندjاست، قبل از اینکه به سندi برگردد. منظور از $p_{i,k}$ ، احتمال انتقال مستقیم ازسندi به سندjمی باشد.c ضریب تعدیل است و دارای مقداری بین صفر و یک است.

بعنوان مثال در شکل ۵- الف با احتساب c=۰.۹ احتمال گریز از سند A به سند B و برعکس بترتیب برابربا ۰.۸۱و ۰.۴۰۵ می باشد. در شکلهای ۵- ب و ۵- ج احتمال گریز از سند A به سندB به ترتیب برابر با ۰.۸۱ و ۰.۷۴ می باشد. این مقادیر نشان می دهند که مسیرهای طولانیتر دارای اتصالات ضعیفتر می باشند.



شکل۵: مقایسه احتمال گریز از گره Aبه B وبرعکس

اگر از روش مستقیم جهت حل رابطه (۸) استفاده کنیم باید رابطه (۸) را به یک دستگاه معادلات خطی تبدیل نماییم[۵] و سپس با حل مستقیم این معادله مسأله حل می گردد. استفاده از این روش برای گرافهای بزرگ که حاوی صدها هزار سند می باشند بدلیل نیاز

به محاسبه معکوس ماتریس ارتباطات بین اسناد، بسیار زمانگیر و حتی غیر ممکن است، همچنین باید به این نکته توجه کرد که به منظور محاسبه درجه ارزش بین تمام اسناد، لازم است که معکوس ماتریس های متفاوتی را محاسبه نماییم. به عبارت دیگر اگر در گراف n سند وجود داشته باشد، به منظور محاسبه درجه ارزش بین تمام اسناد گراف نیاز به محاسبه معکوس $n\times(n-1)$ ماتریس داریم. بنابراین این روش محاسبه، برای گراف های بزرگ مانند وب رد میگردد. با استفاده از ساختار ارتباطی به دست آمده از بخش ۲-۲ و مفهوم احتمال گریز، رابطه پیشنهادی برای محاسبه درجه ارزش بین دو سند به صورت زیر است.

$$Nearness(LA_i, LA_j) = \sum_{k=1}^{n} c \times Nearness(LA_k, LA_j) \times P(LA_i, k) \qquad k \neq i \qquad (۱۰)$$

$$Nearness(LA_i, LA_i) = 0$$

$$Nearness(LA_j, LA_j) = 1$$

در رابطه فوق، اتوماتای یادگیر با شماره i (یک سند وب مثل i) به صورت LA_i ومقداراحتمال انتخاب اقدام k دراتوماتای LA_i به صورت $P(LA_i,k)$ ، نشان داده شده است، که مقدار این احتمال، درایه ای از ماتریس ارتباطات است، که از الگوریتم پیشنهادی برای تعیین ساختار که در بخش۲-۲ مطرح شده است، به دست می آید. چنانچه اقدام مربوطه در اتوماتایی وجود نداشته باشد، مقدار احتمال آن اقدام صفرخواهد بود(یعنی $P(LA_i,k)=0$). پارامتر n، تعداد اسناد وب می باشد. برای حل این نوع معادلات بازگشتی، می توان از روش های عددی استفاده کرد. ما به منظور به دست آوردن کارایی بالاتر نسبت به روش های قبلی از روش برنامه نویسی پویا، برای حل مسأله استفاده می کنیم. با استفاده ازاطلاعات نحوه پیمایش کاربران در هر مرحله بردار احتمال اتوماتای یادگیرمربوط به هر سند، بروز می شود و با استفاده از این بردار در هر مرحله درجه نزدیکی بین هر دوسند به صورت بازگشتی محاسبه میگردد. به منظور پیاده سازی این روش، ابتدا ماتریس $Nearness^{(0)}$ را با ساختار ارتباطی که در بخش۲-۲، به دست آوردیم، مقدار دهی اولیه می کنیم. سپس با توجه به رابطه(۱۰) ، درجه ارزش بین دو سند وب به صورت بازگشتی، از روی درجه ارزش سایر اتوماتاهایی که در طول مسیر این دو سند قرار دارند، به دست می آید. به این ترتیب در هر مرحله درایه های ماتریس $Nearness$ بروز رسانی می شود. به عبارت دیگر در هرمرحله از روی مقادیر ماتریس $Nearness^{(k)}$ که $0 \le k < n$ می باشد، مقادیر ماتریس $Nearness^{(k+1)}$ ، محاسبه می شود. این روند تا زمانی ادامه می یابد که درجه ارزش بین تمام اسناد محاسبه شود. سرعت این روش در مقایسه با روش مستقیم بالاتر است و نتایج آن در بخش شبیه سازی نشان داده شده است. با استفاده از مقادیر درجه ارزش بین اسناد، الگوریتم پیشنهادی کشف زیر گراف متصل جهتدار در شکل۶ نشان داده شده است.

۱- یک اتوماتای یادگیر توزیع شده متناظر با ساختار ارتباطی ارائه شده در بخش ۲-۲ ایجاد کن.

۲-مجموعه اسنادپرس وجو $(Q=\{q_1,q_2,...,q_Q\})$، حداکثر تعداد اسناد میانی (b) ، آرایه $Flag=[f_1,f_2,...,f_Q]$ وتعداد خوشه ها را از ورودی دریافت کن.

۳- الگوریتم خوشه بندی $kmeans$ را روی ساختار ارتباطی حاصل اجرا کن.

۴- خوشه هایی را که شامل اسناد پرس وجو هستند را انتخاب کن و با این خوشه ها ساختار جدید را ایجاد کن.

۵- به ازای هر سند پرس وجوی $q_i \in Q$

۵-۱- برای هر سند j موجود در ساختار ارتباطی

۵-۲- اگر $f_{q_i}=+1$ باشد، $Nearness(q_i,j)$ را محاسبه کن.

در غیر اینصورت، $Nearness(j,q_i)$ را محاسبه کن.

۶- $Nearness(Q,j)$ را به صورت رابطه زیر به دست می آوریم.

$$Nearness(Q,j)=\prod_{i=q_1}^{i=q_Q} Nearness(i,j)$$

۷- زیر گراف خروجی را با مقدار خالی مقدار دهی اولیه می کنیم.

۸- تازمانیکه زیر گراف خروجی از سایز در نظر گرفته شده کوچکتر است:

۸-۱- گره مقصد pd را به صورت زیر انتخاب کن:

۸-۱-۱- $pd=\arg\max_{j\notin H} Nearness(Q,j)$

۸-۱-۲- به ازای هر گره پرس وجوی q_i

۸-۱-۲-۱- اگر $f_{q_i}=+1$ باشد، یک مسیر ازسند پرس

وجوی q_i به سند مقصدpd را پیدا کن.

در غیر این صورت یک مسیر از سند pd به سند پرس

وجوی q_i را پیدا کن.

۸-۱-۲-۲- مسیر را به زیر گراف اضافه کن.

شکل۶: الگوریتم پیشنهادی برای کشف زیر گراف متصل جهتدار با استفاده از اتوماتای یادگیر توزیع شده

به منظور پیدا کردن یک مسیر بین سند پرس وجو q_i و سند مقصدpd، باید تمام اسناد موجود در گراف را بر اساس مقادیر نزولی $Nearness(q_i, j)\forall j \quad (j=1,2,...,n)$ مرتب نماییم. از تمام مقادیر کمتر از $Nearness(q_i, pd)$ صرفه نظر می کنیم. در الگوریتم پیشنهادی کشف مسیر، بر خلاف روشهای بکار رفته[۱۸,۷]، برای انتخاب یک سند علاوه بر ساختار پیوندی بین اسناد به نحوه استفاده و پیمایش کاربران در هنگام تعامل با اسناد وب توجه می شود. نحوه پیدا کردن مسیر بین سند پرس وجو و سند مقصد بر اساس الگوریتم پیشنهادی شکل۷ می باشد.

۱- فرض میکنیم که ماکسیمم طول مسیر مجاز $maxlength$ باشد.

۲- به ازای هر سند $u_j\ (j=1,2,...,n)$ موجود در گراف

۲-۱- به ازای تمام مسیرها ی $len=[2,...,\max length]$

اگر سند u_j در زیر گراف خروجی وجود دارد، $len'=len$

در غیر اینصورت $len'=len-1$

$$Extract_{len}(q_i,u_j)=\max_{u|u\to u_j}(Extract_{len'}(q_i,u)+\Pi_{i=1}^{i=Q}Nearness(q_i,u_j)$$

۳- مسیری انتخاب می شود که دارای ماکسیمم مقدار برای عبارت زیر باشد.

$$\frac{Extract_{len}(q_i,pd)}{len} \qquad len\neq 0$$

شکل۷: الگوریتم کشف مسیر بین سند پرس وجو وسند مقصد *pd*

۵- نتایج شبیه سازیها

در این بخش ابتدا مدل شبیه سازی بطور مختصر ارائه می شود، سپس نتایج به دست آمده از الگوریتمهای پیشنهادی مورد بررسی قرار می گیرد.

۵-۱- مدل شبیه سازي

به منظور انجام شبیه سازیها از مجموعه داده های مدل مطرح شده در[۱۳] استفاده می کنیم. اعتبار این مدل توسط *Liu* و همکاران با استفاده از اطلاعات به دست آمده از چندین وب سایت بزرگ مانند مایکروسافت تأیید شده است. در این مدل، گراف وب و کاربران بر اساس توزیع های آماری، شبیه سازی می شوند و بر این اساس، در این مقاله پروفایل علاقه کاربران به صورت توزیع قانون-توانی و توزیع محتوای اسناد وب، به صورت توزیع نرمال در نظر گرفته شده است. هر سند وب در این مدل 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 موضوعات موجود نشان می دهد(رابطه۱۱). با استفاده از بردار محتوای دو سند iوj، فاصله اقلیدسی بین دو سند $(d_{i,j})$، طبق رابطه(۱۲) محاسبه می شود. سایر پارامتر های استفاده شده در مدل[۱۳] برای شبیه سازیهای انجام شده در این مقاله درجدول۱نشان داده شده است.

$$C_n=\{cw_n^1,cw_n^2,.....cw_n^M\} \qquad (11)$$

$$d_{ij} = \sqrt{\sum_{k=1}^{k=M} (cw_i^k - cw_j^k)} \qquad (۱۲)$$

جدول۱: پارامترهای شبیه سازی استفاده شده در الگوریتم پیشنهادی تعیین ساختار وب

حد آستانه اتصال	۰.۷
تعداد اسناد	۲۰
تعداد کاربران	۲۰۰۰۰
تعداد موضوعات	۵
T_c مقدارثابت سند اولیه درموضوعات مختلف	۰.۲
ΔM_t^c ضریب ثابت کاهش اشتیاق کاربر	-
ΔM_t^v ضریب متغیر کاهش اشتیاق کاربر	-
α_u پارامتر توزیع قانون- توانی توزیع احتمال علایق کاربران	۱
ϕ ضریب پاداش دریافتی از مشاهده یک سند	۱.۲
λ ضریب جذب اطلاعات از یک سند توسط یک کاربر	۰.۵
μ_m میانگین توزیع نرمال ΔM_t^v	۵.۹۷
σ_m واریانس توزیع نرمال ΔM_t^v	۰.۲۵
μ_t میانگین توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع خاص	-
α_p پارامتر توزیع قانون – توانی توزیع احتمالهای وزنهای مطالب برای هر سند	۳
σ واریانس توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع خاص	۰.۲۵
θ ضریب کاهش علاقه کاربر	۱
حداقل اشتیاق کاربر برای ادامه جستجو	۰.۲

۵-۲- ارزیابي كارايي الگوريتم پيشنهادي تعيين ساختار ارتباطي اسناد وب

به منظور ارزیابی کارایی الگوریتم پیشنهادی تعیین ساختار ارتباطی بین اسناد وب و مقایسه آن با سایر روشهای موجود، 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(۱۶) محاسبه می شود. در این رابطه، N تعداد داده هاست. مجموعهP، ساختار ایجاد شده در حالت ایده آل و P' ساختار ایجاد شده توسط الگوریتم پیشنهادی، است. در ساختار ارتباطی ایده آل از عکس فاصله اقلیدسی بین دو سندجهت تعیین ارتباط بین دو سند استفاده شده است.مقدار کورولیشن بین صفر تا یک تغییر می کند، هرقدر میزان کورولیشن دومجموعه داده به عدد یک نزدیکتر باشد این دومجموعه همبستگی بیشتری به هم دارند.

$$p = \{ p_{ij} \mid i, j = 1,2,3,\ldots,n, \quad i \neq j \} \qquad (۱۳)$$

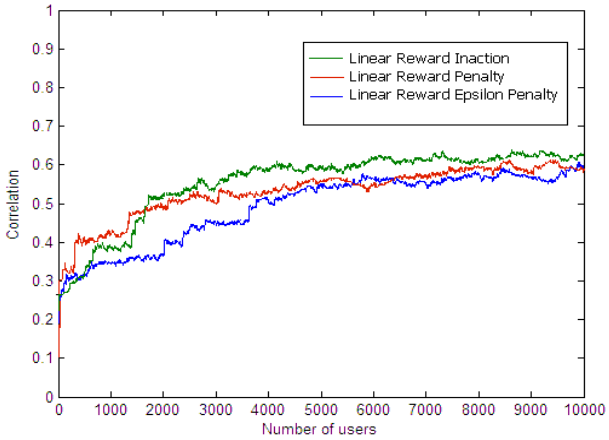
$$p_{ij} = \frac{(d_{ij})^{-1}}{\sum_{k=1}^{k=n} (d_{ik})^{-1}} \qquad (۱۴)$$

$$p' = \{ p'_{ij} \,|\, i, j = 1,2,3,\ldots,n, \quad i \neq j \} \qquad (۱۵)$$

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

$$Corr(P,P') = \frac{\sum PP' - (\sum P \sum P') / N}{\sqrt{(\sum P^2 - (\sum P^2)/N)(\sum P' - (\sum P')^2 / N)}} \qquad (۱۶)$$

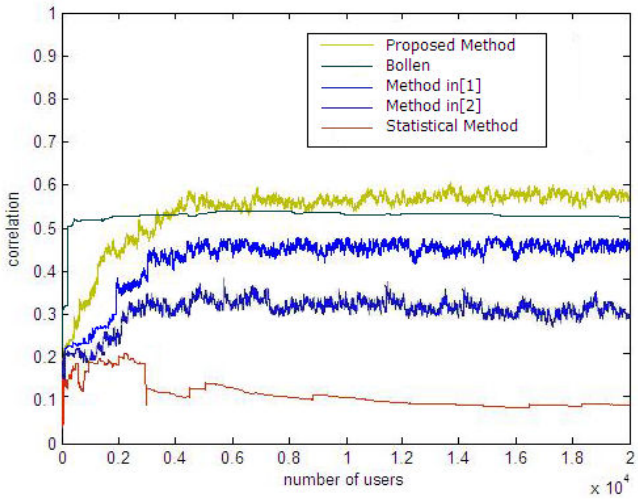
قبل از ارزیابی کارایی الگوریتم پیشنهادی تعیین ساختار، به مقایسه الگوریتمهای یادگیری استاندارد L_{RI} ، L_{RP} و $L_{R\varepsilon P}$ ، در اجرای الگوریتم پیشنهادی تعیین ساختار می پردازیم. برای این منظور شبکه ای شامل ۲۱ سند در نظر گرفتیم. تعداد موضوعات موجود در سیستم و پارامترهای یادگیری aوb را بترتیب ۰.۰۵ ، ۰.۰۵ و ۰.۰۰۱ تعریف نمودیم. کورولیشن ساختار ارتباطی پیشنهادی و ساختار ایده آل به ازای هر الگوریتم یادگیری تا تکرار ۱۰۰۰۰، در شکل۸ نشان داده شده است. همانگونه که در شکل۸ مشاهده می شود الگوریتم یادگیری L_{RI} در مقایسه با دو الگوریتم یادگیری دیگر دارای سطح کورولیشن بالاتر است، به عبارت دیگر ساختار ارتباطی به دست آمده با استفاده از این الگوریتم یادگیری، به ساختار ارتباطی ایده آل نزدیکتر می باشد. بنابراین از الگوریتم یادگیری L_{RI} به منظور تعیین ساختار ارتباطی اسناد استفاده می کنیم.



شکل ۸: مقایسه کورولیشن الگوریتمهای یادگیری L_{RI} ، L_{RP} و $L_{R\varepsilon P}$ به ازای تعداد اسناد ۲۱ و مراحل یادگیری ۱۰۰۰۰

شکل۹، مقایسه الگوریتم پیشنهادی تعیین ساختار را با الگوریتم ارائه شده در[۱]، [۲]، روش بولن[۸] و روش آماری مطرح شده در[۲]، نشان می دهد. در این آزمایش تعداد کاربران۲۰۰۰۰، تعداد موضوعات مرتبط با هر سند۵ و تعداد اسناد وب۲۰در نظرگرفته شده است. الگوریتم پیشنهادی برای تعیین ساختار اسناد وب، بهتر از سایر الگوریتم های ارائه شده مبتنی بر اتوماتای یادگیر، عمل می کند، به علت اینکه دو عامل مهم را در مورد آن لحاظ کرده ایم. اول اینکه تعداد اقدامهای هر اتوماتای یادگیرمطابق با[۱۶]، متغیر درنظر گرفته شده و این باعث می شود که در هر لحظه فقط زیر مجموعه ای از اقدامهای فعال یک اتوماتا، بروزرسانی شوند. دوم اینکه، ضریب پاداش داده شده به اقدام انتخابی اتوماتا به طور پویا تنظیم می گردد. ترکیب این دو عامل باعث می شود، الگوریتم پیشنهادی 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، الگوریتم ارائه شده به دلیل تنظیم اتوماتیک ضریب پاداش، نسبت به روش مبتنی بر اتوماتای یادگیر با تعداد اقدام های متغیر[۲]، با افزایش تعداد اسناد، نیاز به تنظیم مجدد پارامتر یادگیری ندارد. به منظور نمایش یک کران پایین برای کارایی الگوریتم پیشنهادی، از یک روش آماری ساده، استفاده شده است. دراین روش آماری، شباهت دوسندiوj، به صورت نسبت تعداد دفعاتی که کاربران از سندiبه سندj رجوع کرده اند$(N(i,j))$، به تعداد دفعاتی که کاربران از سندi به هر سند دیگری مراجعه کرده اند، محاسبه می شود.

$$Similarity(i,j) = \frac{N(i,j)}{\sum_{k=1}^{n} N(i,k)} \qquad (۱۷)$$



شکل۹ : مقایسه الگوریتم پیشنهادی کشف ساختار ارتباطی با سایر روشهای گزارش شده

۵-۳- ارزیابي كارايي الگوريتم پيشنهادي كشف زير گراف متصل جهتدار

الگوریتم پیشنهادی کشف زیر گراف متصل جهتدار را بر روی ساختار ارتباطی به دست آمده از بخش۲-۲ اجرا می کنیم. در این آزمایش تعداد اسناد،تعداد کاربران،تعداد موضوعات موجود در سیستم و تعداد خوشه ها را به ترتیب ۱۰۰،۵۰۰۰، ۵ و۵ در نظر گرفتیم. سایر پارامترهای شبیه سازی مطابق جدول۱ می باشد. برای ارزیابی کارایی الگوریتم پیشنهادی و کیفیت زیر گرافهای خروجی دو معیار $NRatio$ و $ERatio$ را در نظر گرفتیم. این دو معیار به صورت زیر تعریف می شوند.

$$NRatio = \frac{\sum_{j \in H} Nearness(Q, j)}{\sum_{j \in W} Nearness(Q, j)} \qquad (۱۸)$$

$$ERatio = \frac{\sum_{(j,l) \in H} Nearness(Q,(j,l))}{\sum_{(j,l) \in W} Nearness(Q,(j,l))} \qquad (۱۹)$$

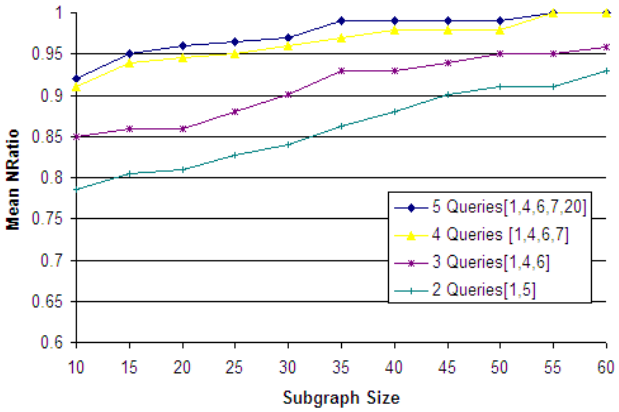
مطابق با رابطه(۱۸) مقدار $NRatio$ ازنسبت مجموع درجه ارزش تمام اسناد موجود در زیر گراف خروجی به مجموع درجه ارزش تمام اسناد موجود در گراف اصلی به دست می آید. در رابطه (۱۹) مقدار $ERatio$ از نسبت مجموع درجه ارزش تمام لبه های موجود در زیر گراف خروجی به مجموع درجه ارزش تمام لبه های موجود در گراف اصلی به دست می آید. درجه ارزش لبه (j,l) در مقابل سند پرس وجوی i مطابق رابطه (۲۰) محاسبه می شود.

$$Nearness\ (i,(j,l)) = \frac{1}{2} \times (Nearness\ (i,j) \times P_{l,j} + Nearness\ (i,l) \times P_{j,l}) \qquad (۲۰)$$

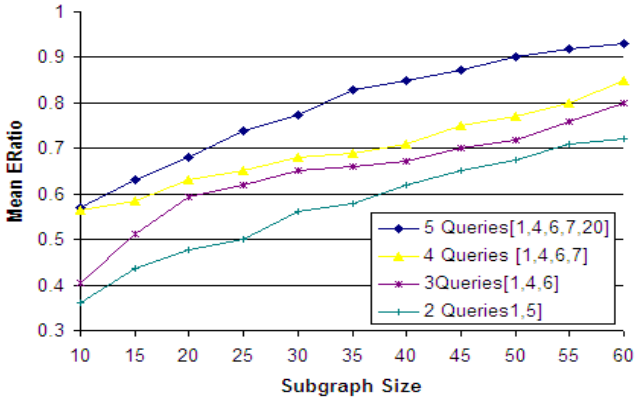
با توجه به رابطه (۲۰)، درجه ارزش لبه (j,l) در مقابل تمام اسناد پرس وجو(Q)، مطابق رابطه (۲۱) به دست می آید.

$$Nearness\ (Q,(j,l)) = \prod_{q_i=1}^{Q} Nearness\ (i,(j,l)) \qquad (۲۱)$$

دومعیار $Nratio$ و $ERatio$ را بعنوان تابعی از تعداد اسناد میانی در نظر گرفتیم. شکلهای۱۰و۱۱، متوسط مقادیر $NRatio$ و $ERatio$ را در مقابل حداکثر تعداد اسناد میانی(b)، به ازای تعداد اسناد پرس وجوی مختلف، نشان می دهد. در هر دو شکل به ازای مقادیرکمb، زیر گراف خروجی، اسناد ولبه هایی با درجه ارزش بالا را شامل می شود. بعنوان مثال در شکلهای ۱۰ و۱۱ به ازای دو سند پرس وجوی۱و۵ زیر گراف خروجی با حداکثر تعداد اسناد میانی۵۰، بطور متوسط۹۱٪ اسناد با ارزش و۶۷٪ لبه های با ارزش را شامل می شود، اما به ازای پنج سند پرس وجوی[۱و۴و۶و۷و۲۰] زیر گراف خروجی با حداکثر تعداد اسناد میانی۲۰، بطو متوسط۹۶٪ اسناد باارزش و۶۸٪ لبه های با ارزش را دارا می باشد. این نکته لازم به ذکر است که به ازای تعداد اسناد میانی برابر، زیر گراف خروجی با تعداد اسناد پرس وجوی بیشتر دارای مقادیر $NRatio$ و $ERatio$ بالاتر نسبت به زیر گراف خروجی با تعداد اسناد پرس وجوی کمتر می باشد. همچنین همانطور که در هر دو شکل مشاهده می شود، به ازای تعداد اسناد پرس وجوی متفاوت، با افزایش سایز زیر گراف، کیفیت زیر گراف استخراج شده افزایش می یابد.

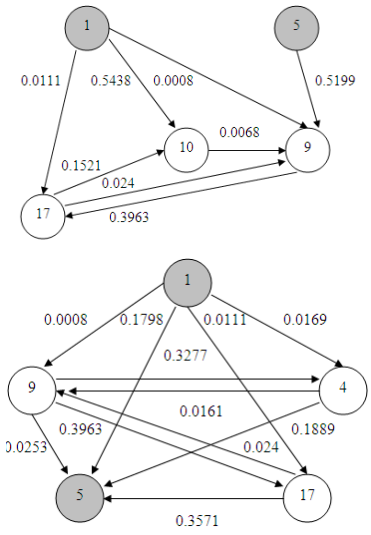


شکل۱۰: متوسط مقادیر *NRatio* در مقابل حداکثر تعداد اسنادمیانی(*b*) و برای تعداد اسناد ۱۰۰به ازای تعداد اسنادپرس وجوی مختلف روی داده های مدل

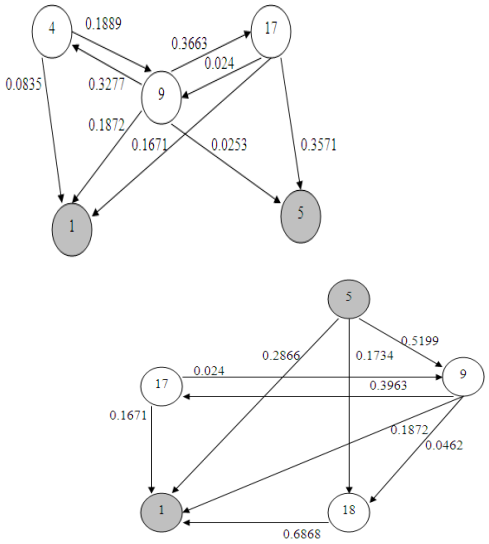


شکل۱۱: متوسط مقادیر *ERatio* در مقابل حداکثر تعداداسنادمیانی(*b*) برای تعداد اسناد ۱۰۰و به ازای تعداد اسنادپرس وجوی مختلف روی داده های مدل

شکل۱۱، نتیجه اجرای الگوریتم پیشنهادی کشف زیر گراف را برای۲۰ سند وب و حداکثر تعداد اسناد میانی ۵ نشان می دهد. در این آزمایش اسناد۱و۵ را بعنوان اسناد پرس وجو انتخاب نمودیم. همانطور که در شکل۱۲ نشان داده شده است به ازای تنظیم مقادیر مختلف *Flag*، انواع زیر گرافها که ارتباط دو سندپرس وجوی ۱و۵ را مشخص می کنند، استخراج می شوند.



الف.[+۱,+۱] **ب. [+۱,-۱]**



پ. [−۱,−۱] ت. [+۱,−۱]

شکل۱۲: زیر گراف کشف شده بین دو سند پرس وجوی ۱و۵ به ازای تنظیم مقادیر مختلف *Flag*

جدول۲، نتایج اجرای الگوریتم پیشنهادی کشف زیر گراف را بدون اعمال روش خوشه بندی، به ازای اسناد پرس وجوی[۷,۶,۴,۱]نشان می دهد. در این آزمایش یک گراف با ۵۵ سند و۱۰۰۰ کاربر مطابق مدل[۱۳] ایجاد نمودیم و حداکثر تعداد اسناد میانی را ۱۰در نظر گرفتیم.

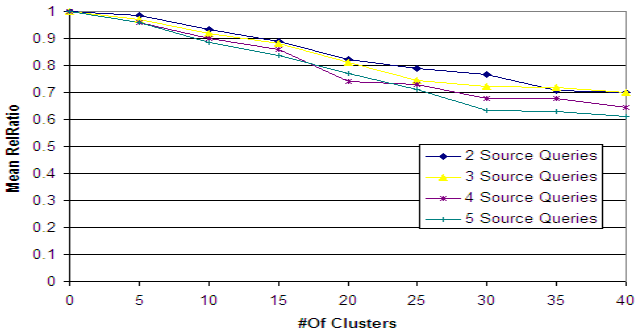
جدول ۲: نتیجه اجرای الگوریتم کشف زیر گراف بر روی مدل [۱۳] ، به ازای اسناد پرس وجوی[۷,۶,۴,۱]

Flag	NRatio	ERatio	Running Time(s)	Nodes in Subgraph
[+۱,+۱,−۱,−۱]	۰.۹۰۴	۰.۲۸۷	۶۱.۲۸	[۱,۴,۶,۷,۹,۱۰,۱۳,۱۹,۲۵,۳۲,۳۸,۴۱,۵۰]
[+۱,+۱,−۱,+۱]	۰.۹۸۵	۰.۸۹	۶۴.۳۶	[۱,۳,۴,۶,۷,۲۷,۳۳,۴۷,۵۲,۵۳]
[−۱,−۱,−۱,+۱]	۰.۶۲	۰.۳۵	۶۲.۶۸	[۱,۲,۳,۴,۶,۷,۲۳,۲۶,۴۲,۵۰,۵۲]
[−۱,−۱,−۱,−۱]	۰.۶۳	۰.۲۳	۶۱.۵۲	[۱,۲,۳,۴,۶,۷,۸,۱۲,۱۳,۳۳,۳۶,۳۹,۴۵]
[−۱,−۱,+۱,−۱]	۰.۶۷	۰.۳۰	۶۳.۳۲	[۱,۲,۳,۴,۶,۷,۹,۱۲,۲۲,۳۲,۳۳,۴۲,۵۲,۵۳]
[−۱,+۱,+۱,−۱]	۰.۶۴	۰.۲۶	۶۵.۲۵	[۱,۲,۴,۶,۷,۹,۱۲,۲۰,۴۲,۵۲]
[+۱,+۱,+۱,−۱]	۰.۹۳۶	۰.۷۰	۶۵.۲۲	[۱,۲,۴,۶,۷,۱۲,۳۳,۳۶,۵۱,۵۲]
[−۱,+۱,−۱,+۱]	۰.۷۰	۰.۳۰	۶۳.۷۶	[۱,۲,۴,۶,۷,۹,۱۳,۲۰,۲۷,۳۲,۳۸,۴۷,۵۳]
[+۱,+۱,+۱,+۱]	۰.۹۹	۰.۸۷	۶۳.۹۸	[۱,۳,۴,۶,۷,۹,۲۰,۲۷,۴۲,۴۷]
[+۱,−۱,+۱,+۱]	۰.۹۶	۰.۹۲	۶۳.۵۱	[۱,۳,۴,۶,۷,۱۲,۲۰,۲۶,۲۷,۴۲,۵۳]
[+۱,−۱,−۱,−۱]	۰.۸۶	۰.۲۲	۶۳.۹۰	[۱,۳,۴,۶,۷,۹,۲۰,۳۳,۳۶,۳۸]
[+۱,−۱,−۱,+۱]	۰.۹۰۷	۰.۵۲۴	۶۳.۴۲	[۱,۳,۴,۶,۷,۱۳,۲۷,۴۲,۵۲,۵۳]
[+۱,−۱,+۱,−۱]	۰.۹۱	۰.۴۲	۶۳.۷۱	[۱,۲,۳,۴,۶,۷,۲۰,۲۳,۴۱,۴۲,۵۲]
[−۱,+۱,−۱,−۱]	۰.۹۷۶	۰.۶۸	۶۵.۱۷	[۱,۲,۳,۴,۶,۷,۹,۱۰,۲۵,۳۴,۴۲]
[−۱,+۱,+۱,+۱]	۰.۹۸	۰.۹۳	۶۳.۷۵	[۱,۲,۳,۴,۶,۷,۱۵,۲۰,۲۳,۴۲]
[−۱,−۱,+۱,+۱]	۰.۴۵	۰.۲۷	۶۳.۵۶	[۱,۲,۳,۴,۶,۷,۱۳,۲۷,۴۱,۵۰]

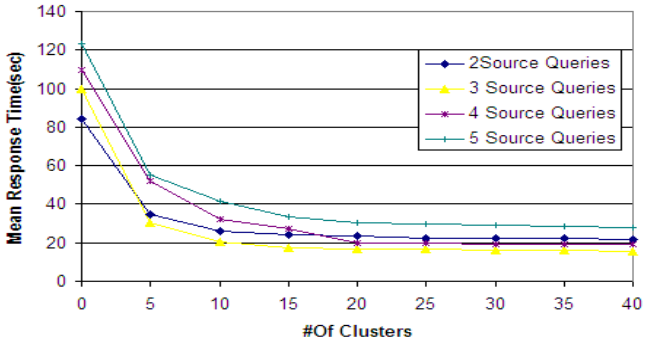
به منظور بررسی تأثیر خوشه بندی اسناد وب بر روی سرعت اجرای الگوریتم پیشنهادی کشف زیر گراف، آزمایش دیگری انجام شد. در این آزمایش نحوه پیمایش ۲۰۰۰ کاربر را از ۱۰۰ سند با استفاده از مدل استخراج نموده و الگوریتم پیشنهادی را بدون اعمال روش خوشه بندی بر روی کل گراف وب اجرا کردیم. حداکثر تعداد اسناد میانی را۲۰ومقادیر آرایه *Flag* را به ازای تمام اسناد پرس وجو۱+ در نظر گرفتیم. برای مقایسه کیفیت زیر گراف های خروجی با بکارگیری روش خوشه بندی و بدون اعمال آن، از معیار *Relratio* استفاده شده است. این معیار طبق رابطه (۲۲) محاسبه می شود.

$$\mathrm{Re}\,lRatio = \frac{N\hat{R}atio}{NRatio} \qquad (۲۲)$$

در عبارت بالا منظور از $N\hat{R}atio$ و $NRatio$ به ترتیب کیفیت زیر گراف خروجی با بکارگیری و بدون بکارگیری روش خوشه بندی روی گراف وب می باشد. شکل۱۳، متوسط مقادیر *RelRatio* را به ازای تعداد خوشه های متفاوت و برای اسنادپرس وجوی مختلف نشان می دهد. همانطور که در شکل۱۳، نشان داده شده است، بدیهی است که با افزایش تعداد خوشه ها کیفیت زیر گراف خروجی که با اعمال روش خوشه بندی حاصل شده کاهش می یابد، به دلیل اینکه با افزایش تعداد خوشه ها، احتمال از دست دادن اسناد با درجه ارزش بالا که در خوشه های انتخابی(خوشه هایی که اسنادپرس وجو در آن قرار دارند) وجود ندارند، بیشتر می شود. اما در قبال از دست دادن درصدی از کیفیت زیر گراف، افزایش سرعت چشمگیری در اجرای الگوریتم کشف زیر گراف مشاهده می شود. همانطور که در شکل۱۴، نشان داده شده است در ازای تعداد خوشه های کم، متوسط زمان اجرای الگوریتم پیشنهادی کشف زیر گراف متصل بشدت کاهش می یابد.

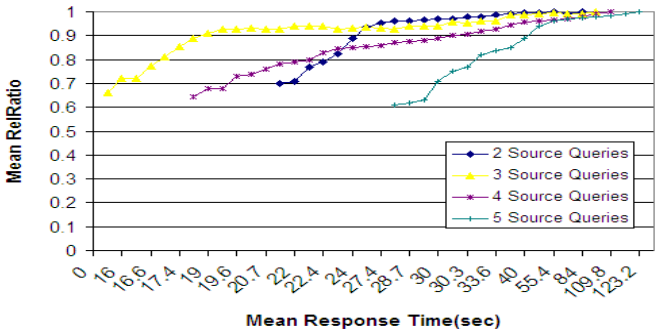


شکل۱۳: ارزیابی کیفیت روش پیشنهادی برای کشف زیر گراف متصل به ازای اسناد پرس وجوی مختلف



شکل۱۴: بررسی تأثیر تعداد خوشه ها در متوسط زمان اجرای الگوریتم پیشنهادی کشف زیر گراف متصل

شکل۱۵، متوسط مقادیر *RelRatio* را به ازای متوسط زمان پاسخ نشان می دهد. همانطور که در شکل مشاهده می شود با از دست دادن یک میزان کم از کیفیت زیر گراف، پروسه زمان پاسخ بسرعت کاهش می یابد. برای مثال با از دست دادن تقریباً۱۱٪ کیفیت، زیر گراف خروجی به ازای۲ گره پرس وجو بطور متوسط در مدت زمان ۲۴ ثانیه و به ازای ۵ گره پرس وجو در مدت زمان ۴۰ ثانیه ایجاد می شود، در حالیکه بدون استفاده از روش خوشه بندی در الگوریتم پیشنهادی زیر گراف، مدت زمان اجرا به ازای این پرس وجوها به ترتیب ۸۴ و۱۲۳.۲ثانیه می باشد.



شکل۱۵: متوسط مقادیر *RelRatio* در مقابل متوسط زمان پاسخ

نتیجه گیری

در این مقاله ابتدا یک الگوریتم جدید مبتنی بر اتوماتای یادگیر توزیع شده برای تعیین ساختار ارتباطی بین اسناد وب پیشنهاد گردید، ساختار ارتباطی پیشنهادی به علت استفاده از اتوماتاهای یادگیر با تعداد اقدامهای متغیر و نحوه تخصیص دینامیک ضریب پاداش به اقدام انتخابی اتوماتا، نسبت به سایر روشهای موجود دارای کارایی بالاتری می باشد. در مرحله بعد با استفاده از ساختار ارتباطی حاصل و روش خوشه بندی *kmeans* الگوریتمی جدید برای کشف زیر گرافهای متصل جهتدار مبتنی بر در خواست کاربر ارائه شد. الگوریتم پیشنهادی کشف زیر گراف متصل به دلیل استفاده همزمان از ساختار پیوندی و نحوه استفاده کاربران از اسناد در ایجاد زیر گرافهای با کیفیت بالاتر مؤثر است، همچنین به دلیل استفاده از تکنیک خوشه بندی اسناد، الگوریتم پیشنهادی کشف زیر گراف متصل دارای سرعت بالایی می باشد. از زیر گرافهای کشف شده می توان جهت شناسایی اجتماعات گوناگون وب، هدایت بهتر کاربران در امر جستجو و پیشنهاد اسناد مشابه با یک سند بر اساس علاقه کاربر استفاده نمود. با استفاده از زیر گرافهای حاصل می توان الگوریتم های کاوش وب مبتنی بر پرس وجوی کاربر را سازماندهی نمود ودر امر شخصی سازی وب بکار برد، این راهکار به عنوان محور اصلی مطالعات بعدی مد نظر قرار می گیرد. در انتهای این مقاله نتایج شبیه سازی ها و ارزیابی کیفیت زیر گرافهای به دست آمده توسط روش پیشنهادی را بر روی داده های مدل نشان دادیم.

مراجع

[۱] اناری، بابک؛ میبدی، محمدرضا؛ **"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.

[۲] Baradaran hashemi, A.; Meybodi, M. R.; *"Web Usage Mining Using Distributed Learning Automata"*, Computer Engineering Department, Technical Report, ۲۰۰۵.

[۳] Beigy, H.; 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. ۳۳۹- ۳٤۳, ۲۰۰۲.

[٤] Cutting, D. R.; Karger, D. R.; Pedersen, J. O.; Tukey, J. W.; *"Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections"*, proceedings of the ۱۵th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Copenhagen, Denmark, pp. ۳۱۸-۳۲۹, June ۲۱-۲٤, ۱۹۹۲.

[۵] Doyle, P.; Snell, J.; *"Random walks and electric networks"*, Mathematical Association America, New York, ۱۹۸٤.

[٦] Dubes, R. C.; Jain, A. K.; *Algorithms for Clustering Data,* Prentice Hall, New York, ۱۹۸۸.

[۷] Faloutsos, C.; McCurley, K. S.; Tomkins, A.; *"Fast Discovery of Connection Subgraphs"*, Proceedings of the ۱۰th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, pp. ۱۱۸-۱۲۷, August ۲۲-۲۵, ۲۰۰۴.

[۸] Heylighen, F.; Bollen, J.; *"Hebbian Algorithms for a Digital Library Recommendation System"*, Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'۰۲), Vancouver, pp. ٤۳۹-٤٤٦, Aguest ۱۸-۲۱, ۲۰۰۲.

[۹] Jain, A. K.; Murty, M. N.; Flynn, P. J.; *"Data Clustering: A Review"*, ACM Computing Surveys, vol. ۳۱, no. ۳, pp. ۲٦٤-۳۲۳, ۱۹۹۹.

[۱۰] Kanungo, T.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; *"An Efficient k-Means Clustering Algorithm: Analysis and Implementation"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. ۲٤, no.۷, pp. ۱۱۰۱-۱۱۱۲, ۲۰۰۲.

[۱۱] Kaufman, L.; Rousseeuw, P. J.; *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley and Sons, ۱۹۹۰.

[۱۲] Koren, Y.; North, S. C.; Volinsky, C.; *"Measuring and Extracting Proximity in Networks"*, Proceedings of the ۱۲th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pennsylvania, USA., pp. ۲٤٤-۲٥٥, ۲۰۰٦.

[۱۳] Liu, J.; Zhang, S.; Yang, J.; *"Characterizing Web Usage Regularities with Information Foraging Agents"*, IEEE Transactions on Knowledge and Data Engineering, vol. ۱٦, no. ٤, pp. ٥٦٦-٥۸٤, April ۲۰۰٤.

[۱٤] Narendra, K. S.; Thathachar, M. A. L.; *Learning Automata: An Introduction*, Prentice Hall, ۱۹۸۹.

[۱٥] Saati, s.; 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 ۲۰۰٥), Tehran, Iran, May ۲٤-۲٦, ۲۰۰٥.

[۱٦] Thathachar, M. A. L.; Bhaskar, R. H.; *"Learning Automata with Changing Number of Actions"*, IEEE Transactions on Systems Man and Cybernetics, vol. ۱۷, no.٦, pp. ۱۰۹٥-۱۱۰۰, Nov. ۱۹۸۷.

[۱۷] Tong, H.; Faloutsos, C.; Pan, J.; *"Fast RandomWalk with Restart and Its Applications"*, Proceeding of the ٦th IEEE International Conference on Data Mining, Hong-Kong, China, pp. ٦۱۳-٦۲۲, Dec. ۱۸-۲۲, ۲۰۰٦.

[۱۸] Tong, H.; Faloutsos, C.; *"Center-piece Subgraphs: Problem Definition and Fast Solutions"*, Proceeding of the ۱۲th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, USA, pp. ٤۰٤-٤۱۳, August ۲۰-۲۳, ۲۰۰٦.

[۱۹] Zamir, O.; Etzioni, O.; Madani, O.; Karp, R. M.; *"Fast and Intuitive Clustering of Web Documents"*, The Proceedings of the ۳th International Conference on Knowledge Discovery and Data Mining, USA, pp. ۲۸۷-۲۹۰, ۱۹۹۷.