

کشف زیر گرافهای متصل وب با استفاده از اتوماتای یادگیر توزیع شده

^۱راهبه مجتهدی صفاری؛ ^۲محمدرضا میبیدی

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.

^۱ عضو هیأت علمی دانشگاه آزاد اسلامی واحد لاهیجان، دانشکده برق و کامپیوتر، دانشگاه آزاد واحد لاهیجان، لاهیجان، ایران. mojtahedi@liau.ac.ir

^۲ 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 این روش زیر گراف به دست آمده، حساس به ترتیب قرارگیری اسناد پرس وجو 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 پایدار برای تمام اسناد موجود در گراف، از یک ماتریس ورودی که ساختار ارتباطی بین اسناد را نشان می دهد، استفاده می گردد. با افزایش تعداد اسناد وب سائز این ماتریس افزایش یافته و محاسبه احتمالات حالت پایدار دارای پیچیدگی زمانی بالایی می شود. بنابراین روش Ceps قابلیت گسترش بر روی گرافهای بزرگ را دارا نمی باشد.

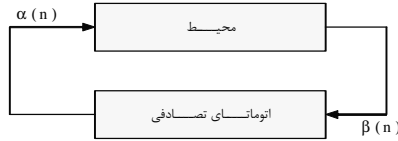
روشهای مطرح شده برای کشف زیر گرافهای متصل [۱۸، ۷]، راه حلی است که بر روی گرافهای بدون جهت اجرا می شوند، همچنین در این روشها تنها از اطلاعات ساختار پیوندی بین اسناد به منظور تعیین درجه ارزش بین اسناد استفاده می گردد. به منظور رفع این مشکلات، در روش پیشنهادی از اتوماتای یادگیر توزیع شده برای حل مسأله کشف 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 بخش ۲، اتوماتاهای یادگیر، اتوماتای یادگیر توزیع شده و یک روش برای تعیین ساختار ارتباطی بین اسناد وب پیشنهاد می گردد. در بخش ۳، روش خوشه بندی k -means مورد بررسی قرار می گیرد. در بخش ۴ الگوریتم پیشنهادی برای کشف زیر گرافهای متصل جهتدار مطرح می گردد. در بخش ۵ نتایج شبیه سازها مطرح شده است. بخش پایانی شامل نتیجه گیری می باشد.

۲- اتوماتاهای یادگیر

اتوماتای یادگیر یک مدل انتزاعی است که به طور تصادفی یک عمل از مجموعه متناهی اعمال خود را انتخاب کرده و بر محیط اعمال می کند. محیط عمل انتخاب شده توسط اتوماتای یادگیر را ارزیابی کرده و نتیجه ارزیابی خود را توسط یک سیگنال تقویتی به اتوماتای یادگیر اطلاع می دهد. سپس اتوماتای یادگیر با اطلاع از عمل انتخاب شده و سیگنال تقویتی، وضعیت داخلی خود را بروز کرده و عمل بعدی خود را انتخاب می کند. محیط را می توان توسط سه $E = \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه

ورودی ها، $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$ مجموعه خروجیها و $C = \{C_1, C_2, \dots, C_r\}$ مجموعه احتمالات جرمیه می باشد. C_i نشان دهنده احتمال نامطلوب بودن سیگنال تقویتی محیط در پاسخ به عمل α_i می باشد. در یک محیط ایستا مقادیر C_i ها ثابت هستند، حال آنکه در یک محیط غیر ایستا این مقادیر در طی زمان تغییر می کنند. بر اساس اینکه تابع به روز رسانی وضعیت اتوماتای یادگیر (که با اطلاع از عمل انتخاب شده و سیگنال تقویت β ، وضعیت بعدی اتوماتای یادگیر را محاسبه می کند) ثابت یا متغیر باشد، اتوماتای یادگیر به دو دسته اتوماتای یادگیر با ساختار ثابت و اتوماتای یادگیر با ساختار متغیر تقسیم می گردند. شکل ۱ یک اتوماتای یادگیر تصادفی را نمایش می دهد.



شکل ۱: 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 چهار تایی $\{\alpha, \beta, p, T\}$ نشان داده می شود که در آن $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه اعمال اتوماتای یادگیر، $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$ مجموعه ورودی های اتوماتای یادگیر، $p = \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از عمل ها و $T, T[\alpha(n), \beta(n), p(n)] = p(n+1)$ الگوریتم یادگیری اتوماتای یادگیر می باشد. الگوریتم های یادگیری متنوعی برای اتوماتای یادگیر ارائه شده است که در ادامه یک الگوریتم یادگیری خطی برای اتوماتای یادگیر بیان می گردد. فرض کنید اتوماتای یادگیر در مرحله n ام 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 اقدام های خود را مطابق روابط ۱ بروز می کند.

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

در رابطه بالا، a پارامتر پاداش و b پارامتر جریمه می باشد. اگر a و b با هم برابر باشند، الگوریتم L_{R-P} ، اگر b از a خیلی کوچکتر باشد، الگوریتم L_{RP} 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 کاربردها 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 زمان n ، 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 انتخاب شده 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.

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

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

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

در صورت پاسخ مطلوب:

$$\begin{aligned} \hat{P}_i(n+1) &= \hat{P}_i(n) + a_i.(1 - \hat{P}_i(n)) \\ j \neq i \quad \forall j \quad \hat{P}_j(n+1) &= \hat{P}_j(n) - a_i.\hat{P}_j(n) \end{aligned} \quad (4)$$

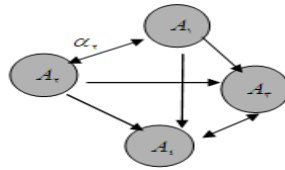
در صورت پاسخ نامطلوب:

$$\begin{aligned}
\hat{p}_i(n+1) &= (1-b) \cdot \hat{p}_i(n) & \alpha(n) &= \alpha \\
\hat{p}_j(n+1) &= \frac{b}{r-1} + (1-b) \hat{p}_j(n) & \alpha(n) &= \alpha_i, \forall j \neq i \\
i, \alpha_i &\in V(n) \text{ for all } P_i(n+1) = \hat{P}_i(n+1).K(n) \\
P_j(n+1) &= P_j(n) \quad j, \alpha_j \notin V(n)
\end{aligned} \quad (5)$$

$$\quad (6)$$

۲-۱- اتوماتاهای یادگیر توزیع شده

اتوماتای یادگیر توزیع شده شبکه‌ای از چند اتوماتای یادگیر است که برای حل یک مساله مشخص با یکدیگر همکاری می‌کنند. یک اتوماتای یادگیر توزیع شده را می‌توان 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 اتوماتای یکی از اعمال خود را انتخاب می‌کند، اتوماتایی که در دیگر انتهای یال متناظر با آن عمل قرار دارد، فعال می‌شود. در هر لحظه فقط یک اتوماتای یادگیر در اتوماتای یادگیر توزیع شده فعال می‌باشد. به عنوان مثال در شکل ۲، یک اتوماتای یادگیر توزیع شده با چهار اتوماتای یادگیر را نشان می‌دهد. برای اطلاعات بیشتر درباره اتوماتای یادگیر توزیع شده می‌توان به مراجع [۱۶، ۱۵، ۱۴، ۳]، مراجعه کرد.



شکل ۲: اتوماتای یادگیر توزیع شده

۲-۲- الگوریتم پیشنهادی تعیین ساختار ارتباطی اسناد وب با استفاده از اتوماتای یادگیر توزیع شده

به منظور تعیین ساختار اسناد وب، یک اتوماتای یادگیر توزیع شده متناظر با اسناد وب ایجاد می‌کنیم. 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 می‌کنند. خروجی یک اتوماتای یادگیر توزیع شده، یک دنباله از اسناد وب مرور شده توسط یک کاربر هستند که مسیر حرکت کاربر را به سمت سند وب مورد نظر نشان می‌دهد. محیط با استفاده از این دنباله، پاسخی برای اتوماتای یادگیر توزیع شده تولید کرده و با استفاده از این پاسخ، ساختار داخلی اتوماتای یادگیر در اتوماتای یادگیر توزیع شده، طبق الگوریتم یادگیری بروز می‌شود.

در این مقاله برای تعیین ساختار ارتباطی بین اسناد وب، در یک 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 می‌شود. برای هر اتوماتای یادگیر در هر زمان تنها زیرمجموعه‌ای از اقدامهای فعال و می‌تواند قابل استفاده باشد [۱۶]. در ابتدای کار تمامی اقدامهای اتوماتای یادگیر در اتوماتای یادگیر توزیع شده غیر فعال می‌باشند. 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 می‌کند، اتوماتاهای یادگیر موجود در اتوماتای یادگیر توزیع شده فعال می‌شوند. 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 می‌گردد، سپس اتوماتای i یک اقدام از مجموعه اقدامهای فعال خود را به طور تصادفی انتخاب می‌نماید، در صورتیکه اقدام انتخاب شده توسط اتوماتای i متناظر با سند j (همان سندی که کاربر در لاگ فایل بعد از سند i مشاهده می‌کند) باشد، پاسخ محیط مطلوب است و اتوماتای i به اقدام انتخابی خود پاداش می‌دهد، 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 سازها 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 پارامتر جریمه (b) را صفر در نظر می‌گیریم، همچنین ضریب پاداش دریافتی توسط اقدام انتخابی اتوماتا، در هر مرحله بر اساس اطلاعات لاگ فایل کاربران بطور پویا تعیین می‌گردد. روند اجرای الگوریتم پیشنهادی به صورت شکل ۳ می‌باشد.

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 می‌باشند.

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

۳-۱- به ازای هر حرکت به صورت $n \rightarrow m$ 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)

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

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

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

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

شکل ۳: 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 و در 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 بندی 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 $L2$ -norm که جمع توان های خطای بین داده ها و مراکز خوشه های آنها (واریانس داخلی خوشه) را تعریف می کند، به عنوان تابع هدف استفاده کرد. روند کلی اجرای این الگوریتم به صورت زیر است.

$$E(c) = \sum_{j=1:k} \sum_{x_i \in C_j} \|x_i - c_j\|^2 \quad (7)$$

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 و ۳ را تا زمانی که مرکز خوشه ها تغییر نکند، تکرار کن.

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- الگوریتم پیشنهادی برای کشف زیر گراف متصل جهتدار با استفاده از اتوماتای یادگیر توزیع 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 از روش خوشه بندی k -means و ساختار ارتباطی به دست آمده از بخش ۲-۲، 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 وجو استفاده می گردد. در این آرایه اسناد پرس وجوی مبدأ و مقصد به ترتیب مقادیر ۱ و ۱- را دارا می باشند. با دریافت اطلاعات این آرایه توسط کاربر، زیر گراف متصل جهتدار متناسب با پرس وجوی کاربر تولید می گردد.

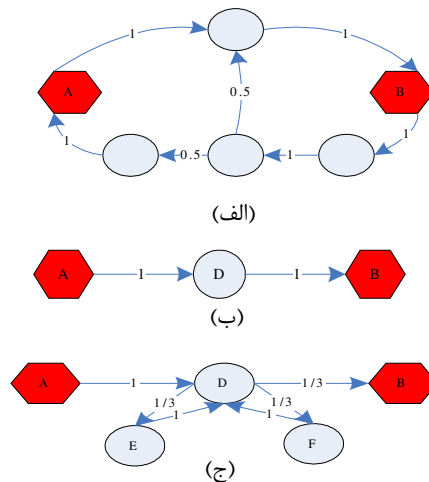
همانطور که بیان شد محاسبه درجه ارزش بین دو سند مهمترین معیار، در کشف زیر گرافهای متصل می باشد. در الگوریتم پیشنهادی کشف زیر گراف متصل بر خلاف روشهای از مفهوم احتمال گریز برای محاسبه درجه ارزش بین اسناد استفاده می کنیم. احتمال گریز از سند i به سند j ، احتمالی است که یک ذره تصادفی با شروع از سند i ، قبل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

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

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

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

بعنوان مثال در شکل ۵- الف با احتساب $c=0.9$ احتمال گریز از سند A به سند B و برعکس بترتیب برابر با ۰.۸۱ و ۰.۴۰۵ می باشد. در شکلهای ۵- ب و ۵- ج احتمال گریز از سند A به سند B به ترتیب برابر با ۰.۸۱ و ۰.۷۴ می باشد. این مقادیر نشان می دهند که مسیرهای طولانیتر دارای اتصالات ضعیفتر می باشند.



شکل ۵: مقایسه احتمال گریز از گره A به B و برعکس

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 $(n-1) \times n$ 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.

$$Nearness(LA_i, LA_j) = \sum_{k=1}^n c \times Nearness(LA_k, LA_j) \times P(LA_i, k) \quad k \neq i \quad (10)$$

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

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

در رابطه فوق، اتوماتای یادگیر با شماره i (یک سند وب مثل i) به صورت LA_i و مقدار احتمال انتخاب اقدام k در اتوماتای 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 بخش ۲-۲ مطرح شده است، به دست می آید. چنانچه اقدام مربوطه در اتوماتایی وجود نداشته باشد، مقدار احتمال آن اقدام صفر خواهد 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 از اطلاعات نحوه پیمایش کاربران در هر مرحله بردار احتمال اتوماتای یادگیر مربوط 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 بخش ۲-۲، 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 هر مرحله از روی مقادیر ماتریس $Nearness^{(k)}$ که $0 \leq k < n$ 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 شکل ۶ نشان داده شده است.

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

۲- مجموعه اسناد پرس وجو ($Q = \{q_1, q_2, \dots, q_Q\}$)، حداکثر تعداد اسناد میانی (b)، آرایه $Flag = [f_1, f_2, \dots, f_Q]$ و تعداد 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 $q_i \in Q$

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

۲-۵- 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:

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

$$pd = \arg \max_{j \in H} Nearness(Q, j) \quad 1-1-8$$

۸-۲-۱- به ازای هر گره پرس وجوی q_i

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

وجوی q_i به سند مقصد pd را پیدا کن.

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

وجوی q_i را پیدا کن.

۸-۲-۱- مسیر را به زیر گراف اضافه کن.

شکل ۶: 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 مقصد pd باید تمام اسناد موجود در گراف را بر اساس مقادیر نزولی $Nearness(q_i, j) \forall j (j = 1, 2, \dots, n)$ 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 [۷، ۱۸]، 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 شکل ۷ می باشد.

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

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

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

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

در غیر اینصورت $len' = len - 1$

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

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

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

شکل ۷: الگوریتم کشف مسیر بین سند پرس وجو و سند مقصد pd

۵- نتایج شبیه سازی

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 سازی

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 (رابطه ۱۱). با استفاده از بردار محتوای دو سند u_i و u_j فاصله اقلیدسی بین دو سند $(d_{i,j})$ ، طبق رابطه (۱۲) محاسبه می شود. سایر پارامترهای استفاده شده در مدل [۱۳] برای شبیه سازیهای انجام شده در این مقاله در جدول ۱ نشان داده شده است.

$$C_n = \{cw_n^1, cw_n^2, \dots, cw_n^M\} \quad (11)$$

$$d_{ij} = \sqrt{\sum_{k=1}^{k=M} (cw_i^k - cw_j^k)^2} \quad (12)$$

جدول ۱: 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 مقدار ثابت سند اولیه در موضوعات مختلف	۰.۲
ΔM_i^c ضریب ثابت کاهش اشتیاق کاربر	-
ΔM_i^v ضریب متغیر کاهش اشتیاق کاربر	-
α_u پارامتر توزیع قانون - توانی توزیع احتمال	۱
علاقه کاربران	
ϕ ضریب پاداش دریافتی از مشاهده یک سند	۱.۲
λ ضریب جذب اطلاعات از یک سند توسط	۰.۵
یک کاربر	
μ_m میانگین توزیع نرمال ΔM_i^v	۵.۹۷
σ_m واریانس توزیع نرمال ΔM_i^v	۰.۲۵
μ_i میانگین توزیع نرمال برای مقدار افزایش	-
یک گره برای یک موضوع خاص	
α_p پارامتر توزیع قانون - توانی توزیع احتمالهای	۳
وزنهای مطالب برای هر سند	
σ واریانس توزیع نرمال برای مقدار افزایش یک	۰.۲۵
گره برای یک موضوع خاص	
θ ضریب کاهش علاقه کاربر	۱
حداقل اشتیاق کاربر برای ادامه جستجو	۰.۲

۵-۲- ارزیابی کارایی الگوریتم پیشنهادی تعیین ساختار ارتباطی 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. مجموعه P ، 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 سند جهت تعیین ارتباط بین دو سند استفاده شده است. مقدار کورولیشن بین صفر تا یک تغییر می کند، هر قدر میزان کورولیشن دو مجموعه داده به عدد یک نزدیکتر باشد این دو مجموعه همبستگی بیشتری به هم دارند.

$$p = \{p_{ij} | i, j = 1, 2, 3, \dots, n, \quad i \neq j\} \quad (13)$$

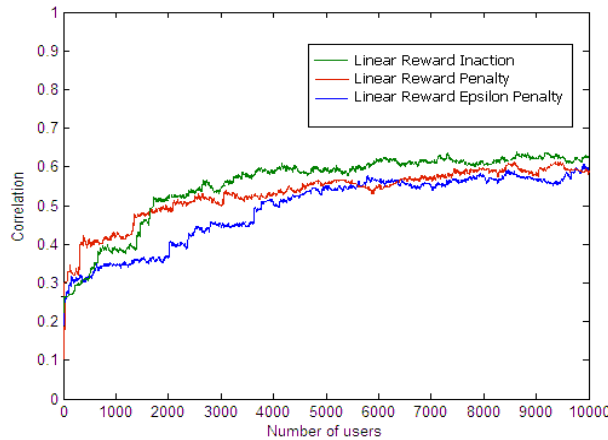
$$p_{ij} = \frac{(d_{ij})^{-1}}{\sum_{k=1}^{k=n} (d_{ik})^{-1}} \quad (14)$$

$$p' = \{p'_{ij} | i, j = 1, 2, 3, \dots, n, \quad i \neq j\} \quad (15)$$

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

$$Corr(P, P') = \frac{\sum PP' - (\sum P \sum P') / N}{\sqrt{(\sum P^2 - (\sum P)^2 / N)(\sum P'^2 - (\sum P')^2 / N)}} \quad (16)$$

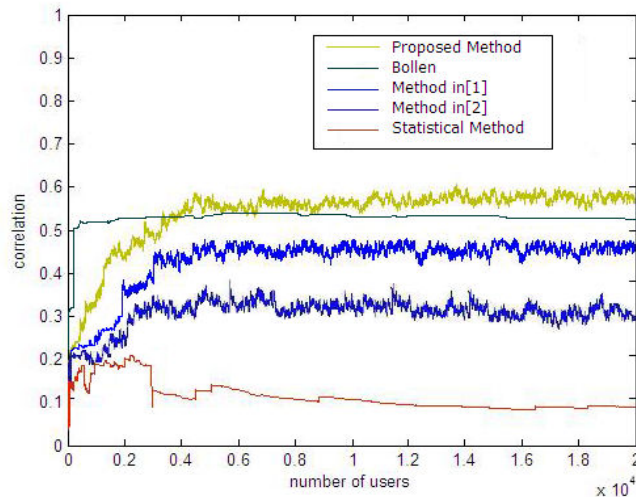
قبل از ارزیابی کارایی الگوریتم پیشنهادی تعیین ساختار، به مقایسه الگوریتمهای یادگیری استاندارد L_{RI} ، L_{RP} و L_{REP} ، در اجرای الگوریتم پیشنهادی تعیین ساختار می پردازیم. برای این منظور شبکه ای شامل ۲۱ سند در نظر گرفتیم. تعداد موضوعات موجود در سیستم و پارامترهای یادگیری ba را به ترتیب ۵، ۰.۰۵ و ۰.۰۰۱ تعریف نمودیم. کورولیشن ساختار ارتباطی پیشنهادی و ساختار ایده آل به ازای هر الگوریتم یادگیری تا تکرار ۱۰۰۰۰، در شکل ۸ نشان داده شده است. همانگونه که در شکل ۸ 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_{REP} به ازای تعداد اسناد ۲۱ و مراحل یادگیری ۱۰۰۰۰

شکل ۹، 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 کاربران ۲۰۰۰، تعداد موضوعات مرتبط با هر سند ۵ و تعداد اسناد ۲۰ در نظر گرفته شده است. الگوریتم پیشنهادی برای تعیین ساختار اسناد وب، بهتر از سایر الگوریتم های ارائه شده مبتنی بر اتوماتای یادگیر، عمل می کند، به علت اینکه دو عامل مهم را در مورد آن لحاظ کرده ایم. اول اینکه تعداد اقدامهای هر اتوماتای یادگیر مطابق با [۱۶]، متغیر در نظر گرفته شده و این باعث می شود که در هر لحظه فقط زیر مجموعه ای از اقدامهای فعال یک اتوماتا، بروزسانی 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. در این روش آماری، شباهت دو سند i و j به صورت نسبت تعداد دفعاتی که کاربران از سند i و j رجوع کرده اند $N(i, j)$ ، به تعداد دفعاتی که کاربران از سند i به هر سند دیگری مراجعه کرده اند، محاسبه می شود.

$$Similarity(i, j) = \frac{N(i, j)}{\sum_{k=1}^n N(i, k)} \quad (17)$$



شکل ۹: مقایسه الگوریتم پیشنهادی کشف ساختار ارتباطی با سایر روشهای گزارش شده

۵-۳- ارزیابی کارایی الگوریتم پیشنهادی کشف زیر گراف متصل جهتدار

الگوریتم پیشنهادی کشف زیر گراف متصل جهتدار را بر روی ساختار ارتباطی به دست آمده از بخش ۲-۲ اجرا می کنیم. در این آزمایش تعداد اسناد، تعداد کاربران، تعداد موضوعات موجود در سیستم و تعداد خوشه ها را به ترتیب ۵۰۰، ۵، ۵ و ۵ در نظر گرفتیم. سایر پارامترهای شبیه سازی مطابق جدول ۱ می باشد. برای ارزیابی کارایی الگوریتم پیشنهادی و کیفیت زیر گرافهای خروجی دو معیار $ERatio$ و $NRatio$ را در نظر گرفتیم. این دو معیار به صورت زیر تعریف می شوند.

$$NRatio = \frac{\sum_{j \in H} Nearness(Q, j)}{\sum_{j \in W} Nearness(Q, j)} \quad (18)$$

$$ERatio = \frac{\sum_{(j,l) \in H} Nearness(Q, (j,l))}{\sum_{(j,l) \in W} Nearness(Q, (j,l))} \quad (19)$$

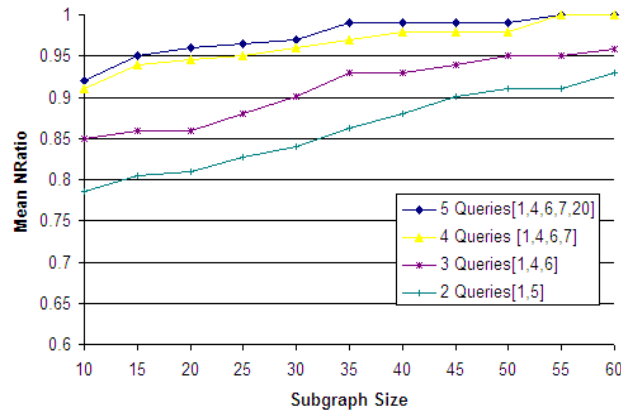
مطابق با رابطه (۱۸) مقدار $NRatio$ از نسبت مجموع درجه ارزش تمام اسناد موجود در زیر گراف خروجی به مجموع درجه ارزش تمام اسناد موجود در گراف اصلی به دست می آید. در رابطه (۱۹) مقدار $ERatio$ از نسبت مجموع درجه ارزش تمام لبه های موجود در زیر گراف خروجی به مجموع درجه ارزش تمام لبه های موجود در گراف اصلی به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

$$Nearness(i, (j,l)) = \frac{1}{2} \times (Nearness(i, j) \times P_{i,j} + Nearness(i, l) \times P_{j,l}) \quad (20)$$

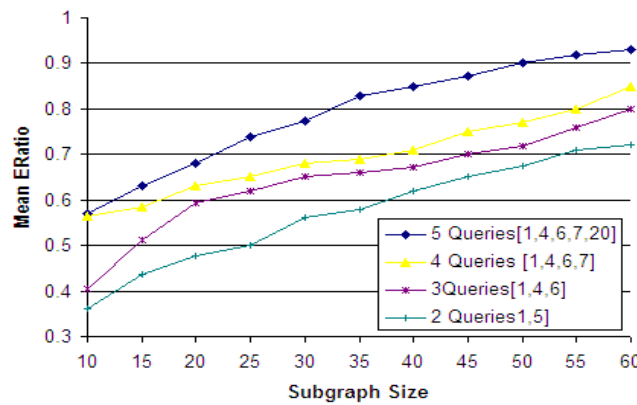
با توجه به رابطه (۲۰)، درجه ارزش لبه (j,l) در مقابل تمام اسناد پرس وجو (Q)، مطابق رابطه (۲۱) به دست می آید.

$$Nearness(Q, (j,l)) = \prod_{q_i=1}^Q Nearness(i, (j,l)) \quad (21)$$

دو معیار $ERatio$ و $NRatio$ را بعنوان تابعی از تعداد اسناد میانی در نظر گرفتیم. شکلهای ۱۰ و ۱۱، متوسط مقادیر $ERatio$ و $NRatio$ را در مقابل حداکثر تعداد اسناد میانی (b)، به ازای تعداد اسناد پرس وجوی مختلف، نشان می دهد. در هر دو شکل به ازای مقادیر کم b ، زیر گراف خروجی، اسناد ولبه هایی با درجه ارزش بالا را شامل می شود. بعنوان مثال در شکلهای ۱۰ و ۱۱ به ازای دو سند پرس وجوی ۵ و ۲۰ زیر گراف خروجی با حداکثر تعداد اسناد میانی ۵۰، بطور متوسط ۹۱٪ اسناد با ارزش ۶۷٪ لبه های با ارزش را شامل می شود، اما به ازای پنج سند پرس وجوی [۲۰ و ۷۶ و ۴۱ و ۲۰] زیر گراف خروجی با حداکثر تعداد اسناد میانی ۲۰، بطور متوسط ۹۶٪ اسناد با ارزش ۶۸٪ 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 $ERatio$ و $NRatio$ 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.

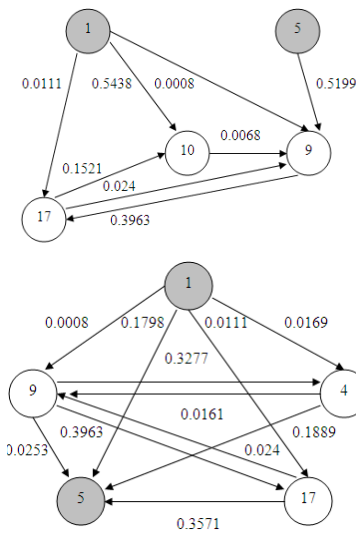


شکل ۱۰: متوسط مقادیر $NRatio$ در مقابل حداکثر تعداد اسناد میانی (b) و برای تعداد اسناد ۱۰۰ به ازای تعداد اسناد پرس وجوی مختلف روی داده های مدل



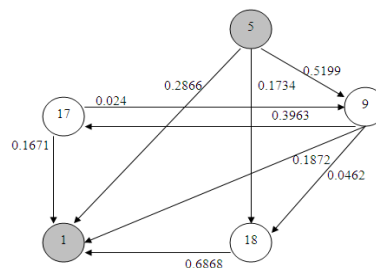
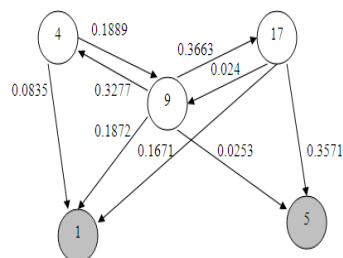
شکل ۱۱: متوسط مقادیر $ERatio$ در مقابل حداکثر تعداد اسناد میانی (b) برای تعداد اسناد ۱۰۰ به ازای تعداد اسناد پرس وجوی مختلف روی داده های مدل

شکل ۱۱، نتیجه اجرای الگوریتم پیشنهادی کشف زیر گراف را برای ۲۰ سند وب و حداکثر تعداد اسناد میانی ۵ نشان می دهد. در این آزمایش اسناد ۱ و ۵ را بعنوان اسناد پرس وجو انتخاب نمودیم. همانطور که در شکل ۱۲ نشان داده شده است به ازای تنظیم مقادیر مختلف $Flag$ ، انواع زیر گرافها که ارتباط دو سند پرس وجوی ۱ و ۵ را مشخص می کنند، استخراج می شوند.



ب. $[-1, +1]$

الف. $[+1, +1]$



ت. $[+1, -1]$

پ. $[-1, -1]$

شکل ۱۲: زیر گراف کشف شده بین دو سند پرس وجوی ۵ به ازای تنظیم مقادیر مختلف *Flag*

جدول ۲، نتایج اجرای الگوریتم پیشنهادی کشف زیر گراف را بدون اعمال روش خوشه بندی، به ازای اسناد پرس وجوی $[7, 6, 4, 1]$ نشان می دهد. در این آزمایش یک گراف با ۵۵ سند و ۱۰۰۰ کاربر مطابق مدل [۱۳] ایجاد نمودیم و حداکثر تعداد اسناد میانی را ۱۰ در نظر گرفتیم.

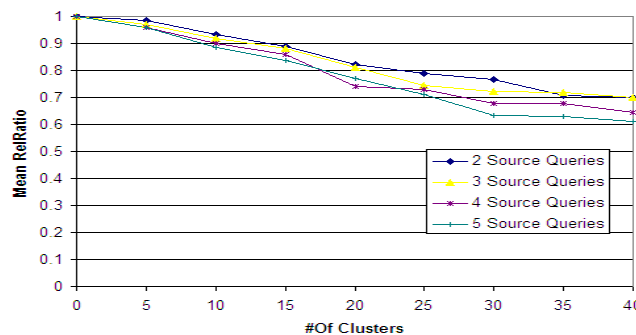
جدول ۲: نتیجه اجرای الگوریتم کشف زیر گراف بر روی مدل [۱۳]، به ازای اسناد پرس وجوی $[7, 6, 4, 1]$

Flag	NRatio	ERatio	Running Time(s)	Nodes in Subgraph
$[+1, +1, -1, -1]$	۰.۹۰۴	۰.۲۸۷	۶۱.۲۸	$[1, 4, 6, 7, 9, 10, 13, 19, 25, 32, 38, 41, 50]$
$[+1, +1, -1, +1]$	۰.۹۸۵	۰.۸۹	۶۴.۳۶	$[1, 3, 4, 6, 7, 27, 33, 47, 52, 53]$
$[-1, -1, -1, +1]$	۰.۶۲	۰.۳۵	۶۲.۶۸	$[1, 2, 3, 4, 6, 7, 23, 26, 42, 50, 52]$
$[-1, -1, -1, -1]$	۰.۶۳	۰.۲۳	۶۱.۵۲	$[1, 2, 3, 4, 6, 7, 8, 12, 13, 23, 36, 39, 45]$
$[-1, -1, +1, -1]$	۰.۶۷	۰.۳۰	۶۳.۳۲	$[1, 2, 3, 4, 6, 7, 9, 12, 22, 32, 33, 42, 52, 53]$
$[-1, +1, +1, -1]$	۰.۶۴	۰.۲۶	۶۵.۲۵	$[1, 2, 4, 6, 7, 9, 12, 20, 42, 52]$
$[+1, +1, +1, -1]$	۰.۹۳۶	۰.۷۰	۶۵.۲۲	$[1, 2, 4, 6, 7, 12, 33, 36, 51, 52]$
$[-1, +1, -1, +1]$	۰.۷۰	۰.۳۰	۶۳.۷۶	$[1, 2, 4, 6, 7, 9, 13, 20, 27, 32, 38, 47, 53]$
$[+1, +1, +1, +1]$	۰.۹۹	۰.۸۷	۶۳.۹۸	$[1, 3, 4, 6, 7, 9, 20, 27, 42, 47]$
$[+1, -1, +1, +1]$	۰.۹۶	۰.۹۲	۶۳.۵۱	$[1, 3, 4, 6, 7, 12, 20, 26, 27, 42, 53]$
$[+1, -1, -1, -1]$	۰.۸۶	۰.۲۲	۶۳.۹۰	$[1, 3, 4, 6, 7, 9, 20, 33, 36, 38]$
$[+1, -1, -1, +1]$	۰.۹۰۷	۰.۵۲۴	۶۳.۴۲	$[1, 3, 4, 6, 7, 13, 27, 42, 52, 53]$
$[+1, -1, +1, -1]$	۰.۹۱	۰.۴۲	۶۳.۷۱	$[1, 2, 3, 4, 6, 7, 20, 23, 41, 42, 52]$
$[-1, +1, -1, -1]$	۰.۹۷۶	۰.۶۸	۶۵.۱۷	$[1, 2, 3, 4, 6, 7, 9, 10, 25, 34, 42]$
$[-1, +1, +1, +1]$	۰.۹۸	۰.۹۳	۶۳.۷۵	$[1, 2, 3, 4, 6, 7, 15, 20, 23, 42]$
$[-1, -1, +1, +1]$	۰.۴۵	۰.۲۷	۶۳.۵۶	$[1, 2, 3, 4, 6, 7, 13, 27, 41, 50]$

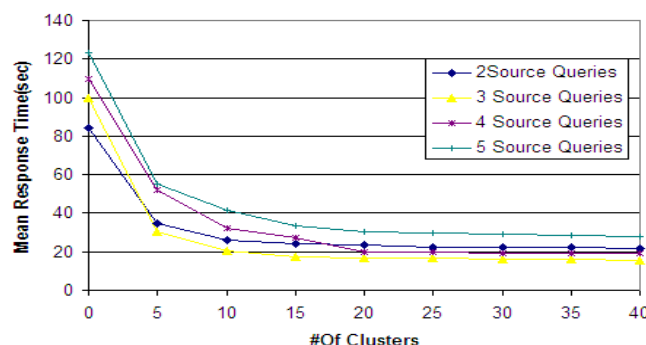
به منظور بررسی تأثیر خوشه بندی اسناد وب بر روی سرعت اجرای الگوریتم پیشنهادی کشف زیر گراف، آزمایش دیگری انجام شد. در این آزمایش نحوه پیمایش ۲۰۰۰ کاربر را از ۱۰۰ سند با استفاده از مدل استخراج نموده و الگوریتم پیشنهادی را بدون اعمال روش خوشه بندی بر روی کل گراف وب اجرا کردیم. حداکثر تعداد اسناد میانی را ۲۰ و مقادیر آرایه *Flag* را به ازای تمام اسناد پرس وجو +۱ 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.

$$RelRatio = \frac{N\hat{Ratio}}{NRatio} \quad (22)$$

در عبارت بالا منظور از *NRatio* و *NRatio* به ترتیب کیفیت زیر گراف خروجی با بکارگیری و بدون بکارگیری روش خوشه بندی روی گراف وب می باشد. شکل ۱۳، متوسط مقادیر *RelRatio* را به ازای تعداد خوشه های متفاوت و برای اسناد پرس وجوی مختلف نشان می دهد. همانطور که در شکل ۱۳، 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 اسناد پرس 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 شکل ۱۴، نشان داده شده است در ازای تعداد خوشه های کم، متوسط زمان اجرای الگوریتم پیشنهادی کشف زیر گراف متصل بشدت کاهش می یابد.

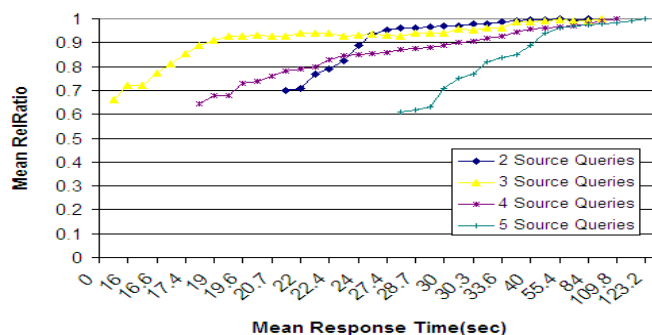


شکل ۱۳: ارزیابی کیفیت روش پیشنهادی برای کشف زیر گراف متصل به ازای اسناد پرس وجوی مختلف



شکل ۱۴: بررسی تأثیر تعداد خوشه ها در متوسط زمان اجرای الگوریتم پیشنهادی کشف زیر گراف متصل

شکل ۱۵، 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 تقریباً ۱۱٪ کیفیت، زیر گراف خروجی به ازای ۲ 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 و ۱۲۳.۲ ثانیه می باشد.



شکل ۱۵: 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 درخواست 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 و جوی کاربر را سازماندهی نمود و در 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. ۳۳۹-۳۴۳, ۲۰۰۲.
- [۴] 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 ۲۱-۲۴, ۱۹۹۲.
- [۵] Doyle, P.; Snell, J.; "Random walks and electric networks", Mathematical Association America, New York, ۱۹۸۴.
- [۶] 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'۰۶), Vancouver, pp. ۴۲۹-۴۴۶, August ۱۸-۲۱, ۲۰۰۶.

- [9] Jain, A. K.; Murty, M. N.; Flynn, P. J.; “*Data Clustering: A Review*”, ACM Computing Surveys, vol. 31, no. 3, pp. 264-323, 1999.
- [10] 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. 24, no. 5, pp. 1101-1114, 2002.
- [11] Kaufman, L.; Rousseeuw, P. J.; *Finding Groups in Data: an Introduction to Cluster Analysis*, John Wiley and Sons, 1990.
- [12] Koren, Y.; North, S. C.; Volinsky, C.; “*Measuring and Extracting Proximity in Networks*”, Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pennsylvania, USA., pp. 244-250, 2006.
- [13] Liu, J.; Zhang, S.; Yang, J.; “*Characterizing Web Usage Regularities with Information Foraging Agents*”, IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 4, pp. 566-584, April 2004.
- [14] Narendra, K. S.; Thathachar, M. A. L.; *Learning Automata: An Introduction*, Prentice Hall, 1989.
- [15] 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 2006), Tehran, Iran, May 24-26, 2006.
- [16] Thathachar, M. A. L.; Bhaskar, R. H.; “*Learning Automata with Changing Number of Actions*”, IEEE Transactions on Systems Man and Cybernetics, vol. 19, no. 6, pp. 1400-1404, Nov. 1989.
- [17] Tong, H.; Faloutsos, C.; Pan, J.; “*Fast RandomWalk with Restart and Its Applications*”, Proceeding of the 11th IEEE International Conference on Data Mining, Hong-Kong, China, pp. 613-624, Dec. 18-22, 2006.
- [18] Tong, H.; Faloutsos, C.; “*Center-piece Subgraphs: Problem Definition and Fast Solutions*”, Proceeding of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, USA, pp. 444-453, August 20-24, 2006.
- [19] Zamir, O.; Etzioni, O.; Madani, O.; Karp, R. M.; “*Fast and Intuitive Clustering of Web Documents*”, The Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining, USA, pp. 287-296, 1997.