تشخیص اجتماعات وب با استفاده از اتوماتای یادگیر توزیع شده و پارتیشن بندی گراف

مجید تاران[۱]، شهرزاد معتمدی مهر[۲]، علی برادران هاشمی[۳]، محمدرضا میبدی[۴]

چکیده

مجموعه ای از صفحات وب که درباره یک موضوع مشترک می باشند وتوسط افراد یا سازمان های مختلف که علایق مشترک درباره آن موضوع خاص دارند ایجاد شده اند، اجتماع وب نامیده می شود. از آنجا که امروزه حجم وب از یک بیلیون صفحه گذشته است و همچنان در حال افزایش است، تشخیص اجتماعات وب روز به روز دشوارتر می شود. در این مقاله روشی ترکیبی مبتنی براتوماتای یادگیر توزیع شده و پارتیشن بندی گراف برای تشخیص اجتماعات وب پیشنهاد می گردد. روش پیشنهادی همان الگوریتم HITS می باشد که درآن علاوه بر ساختار پیوند بین صفحات، رفتار کاربر در مشاهده این صفحات نیز در نظر گرفته شده است. برای این منظور از اتوماتای یادگیر توزیع شده برای یادگیری امتیازات Authority و Hub صفحات وب استفاده می گردد. اجتماع وبی که به این روش به دست می آید وابسته به ساختا ر گرافی وب نمی باشد. الگوریتم پیشنهادی با استفاده از پیوندهای بین صفحات و رفتار کاربران وب، میزان شباهت و ارتباط بین صفحات را تعیین می کند. در این مقاله از پارتیشن بندی گراف وب برای بهبود کارایی استفاده شده است. به منظور ارزیابی، روش پیشنهادی پیاده سازی گردیده و نتایج ان با نتایج الگوریتم HITS و الگوریتمی دیگر مبتنی اتوماتای یادگیر توزیع شده مقایسه شده است. نتایج آزمایشها حاکی از کارایی روش پیشنهادی دارد.

کلمات کلیدی

اتوماتای یادگیر، اجتماع وب، داده های استفاده از وب

Identification of Web Communities using Distributed Learning Automata and Graph Partitioning

Shahrzad Motamedi Mehr; Majid Taran; Ali Baradaran Hashemi; Mohammad Reza Meybodi

ABSTRACT

A collection of web pages which are about a common topic and are created by individuals or any kind of associations that have a common interest on that specific topic is called a web community. Since at present, the size of the web is about ۳ bilion pages and it is still growing very fast, identification of web communities has become an increasingly hard task. In this paper a distributed learninig automata based approach and graph partitioning for identification of web communities is proposed. The Proposed approach is a combination of web structure. Web usage and web content mining techniques. The proposed approach is based on HITS algorithm in which in addition to link structure of web pages, the users' behavior in visiting these pages is also taken into consideration for Identification of Web Communities. A distributed learning automaton is used to learn the hub and the authority scores of web pages. The web community obtained using this method is not dependent on a special structure. The proposed algorithm to determine similarity between web pages using hyperlinks and users behavior in visiting web pages. Graph partitioning improves the quality of web usage mining significantly. To evaluate the proposed approach, it is implemented and the results are compared with the results of two existing methods, HITS and a distributed learning automata based method. Experimental results show the performance of the proposed method.

KEYWORDS

Learning Automata, Web Usage Mining, Web Community

[۱] دانشکده برق و مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد قزوین، قزوین، ایران، *m_taran@isc.iranet.net*
[۲] دانشکده برق و مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد قزوین، قزوین، ایران، *motamedi@tmu.ac.ir*
[۳] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، *a_hashemi@aut.ac.ir*
[۴] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران، ایران، *mmeybodi@aut.ac.ir*

۱- مقدمه

وب طی یک فرآیند آشفته و غیر متمرکز رشد می کند و این روند منجر به تولید حجم وسیعی از مستندات متصل به یکدیگر گشته است که از هیچ گونه سازماندهی منطقی برخوردار نیستند. در حال حاضر موتور جستجوی Googleبیش از ۳ بیلیون صفحه وب را شاخص گذاری کرده است که این تعداد با نرخ ۷.۳میلیون صفحه در روز افزایش می یابد. برای بهره برداری از این حجم وسیع داده در سال های اخیر تکنیک های وب کاوی[۱] معرفی شده اند. یکی از انواع وب کاوی، کاوش ساختار وب[۲] است که از ساختار پیوندهای موجود بین صفحات وب اطلاعات راجع به این صفحات و ارتباطشان را به دست می آورد. در این نوع از وب کاوی، وب به صورت یک گراف مدلسازی می شود که در آن صفحات وب، گره های گراف و پیوندهای[۳] بین صفحات، یال های گراف هستند. کاوش ساختار وب برای اهداف متفاوتی همچون رتبه بندی صفحات وب، تشخیص اجتماعات وب، تحلیل گراف وب، مدلسازی و شبیه سازی فرآیند تولید گر اف وب به کار می رود. یک اجتماع وب مجموعه ای از صفحات وب است که درباره یک موضوع مشترک می باشند و توسط افراد یا سازمان های مختلف که علایق مشترک درباره آن موضوع خاص دارند ایجاد شده اند [۲۲]. با تشخیص یک اجتماع وب درباره یک موضوع خاص، کاربران می توانند با استفاده از صفحات اجتماع، اطلاعات مفیدی درباره آن موضوع به دست آورند. از آنجا که امروزه حجم وب از سه بیلیون صفحه گذشته است و همچنان در حال افزایش است، تشخیص اجتماعات روز به روز دشوارتر می شود.

روشهای مختلفی برای تشخیص اجتماعات وب گزارش شده است که انها را می توان به دو گروه روشهای مبتنی بر تحلیل پیوند [۴] و روشهای مبتنی بر تئوری گراف تقسیم کرد. از جمله روشهایی که مبتنی بر تحلیل پیوند هستند می توان به روشهای ارایه شده در [۲] و [۲۲] اشاره کرد. در روش ارایه شده در [۲۲] یک مجموعه اولیه از صفحات را به عنوان ورودی دریافت می کند و اجتماعات شامل آنها را به دست می آورد. این روش مبتنی بر الگوریتمی برای یافتن صفحات مرتب (RPA)[۵] است که صفحات مرتبط با یک صفحه را با استفاده از تحلیل پیوندها به دست می آورد. الگوریتم RPA بر هر یک از صفحات مجموعه اولیه اعمال می شود. سپس با توجه به شباهت بین نتایج به دست آمده، صفحات به گروه هایی تقسیم و اجتماعات وب به دست می آیند. در روش ارایه شده در [۲] که یکی از مهترین روش های تشخیص اجتماعات وب است مجموعه ای از صفحات Authority و Hub را به عنوان اجتماع وب معرفی می کند. یک Authority صفحه ای حاوی اطلا عات ارزشمند راجع به یک موضوع خاص می باشد. یک Hub نیز صفحه ای حاوی پیوندهایی به صفحاتی با اطلاعات ارزشمند راجع به یک موضوع خاص است. این روش با استفاده از الگوریتم HITS [۶] صفحات Hub و Authority را تشخیص می دهد.

روش ها ی گزارش شده در [۳][۴][۵][۷] از جمله روش های مبتنی بر تئوری گراف می باشند. روش های مبتنی بر تئوری گراف به تحلیل گراف وب می پردازند، اما از آن جا که وب بسیار گسترده و رو به رشد می باشد، به کارگیری الگوریتم های گراف به سادگی امکان پذیرنمی باشد. به منظور آن که این الگوریتم ها قابل استفاده در وب باشند، باید توانایی هایی همچون مقاومت در برابر داده های ناکامل و ناشناخته را داشته باشند. در این روش ها اجتماعات وب، به صورت بخش های متراکم گراف وب تعریف می شوند. اما ساختار زیر گراف متراکم در هر یک از این روش ها متفاوت است. برای مثال Kumer و همکارانش در [۳] اجتماعات موجود در وب را با تشخیص گراف های کامل دوبخشی در آن به دست آورده اند. آنها اجتماعات وب را در هنگام پیمایش وب و با استفاده از تکنیکی به نام Trawling به دست می آورند. در [۴] روش دیگری برای تشخیص اجتماعات وب با استفاده از گراف کامل دو بخشی[۶] ارائه شده است. در این روش مجموعه ای از صفحات وب به عنوان ورودی الگوریتم در نظر گرفته می شوند. ابتدا کلیه گراف های دوبخشی کامل $k_{۳,۳}$ گراف مجاورت این صفحات بدست می آید. سپس این زیرگراف هابا یکدیگر ادغام و اجتماعات را تولید می کنند. علاوه بر روش های مبتنی بر گراف کامل دوبخشی، روش هایی دیگری نیز با استفاده از تئوری گراف به تشخیص اجتماعات وب پرداخته اند. در روش های معرفی شده در [۵][۷] مجموعه ای از گره ها که تعداد پیوندهای آنها با اعضای مجموعه بیش از تعداد پیوندهای آنها با اعضای خارج از مجموعه است، به عنوان اجتماعات وب در نظر گرفته شده اند. در این روش ها یک اجتماع وب، از طریق جدا کردن یک زیرگراف از وب با استفاده از الگوریتم جریان بیشینه به دست می آید.

روش های گزارش شده برای تشخیص اجتماعات که تاکنون گزارش شده است تنها از ساختار پیوندهای بین صفحات وب استفلده می کنند. استفاده از ساختار پیوندهای بین صفحات وب به تنهایی نمی تواند ارتباط مفهومی بین صفحات وب را استخراج کند. در [۲۲] الگوریتم HITS بهبود داده شده است که علاوه بر ساختار پیوند بین صفحات از رفتار کاربران نیز استفاده می کند ولی در تعداد صفحات بزرگ وب نتایج خوبی را نشان نمی دهد در روش پیشنهادی ما از پارتیشن بندی گراف برای حل این مشکل استفاده می کنیم.

روش پیشنهادی همان الگوریتم HITS می باشد که درآن علاوه بر ساختار پیوند بین صفحات، رفتار کاربر در مشاهده این صفحات نیز در نظر گرفته شده است در این مقاله الگوریتمی ترکیبی مبتنی بر اتوماتای یادگیر توزیع شده و پارتیشن بندی گراف ارایه شده است. برای این منظور از اتوماتای یادگیر توزیع شده برای یادگیری امتیازا ت Authority و Hub صفحات وب استفاده می گردد. اجتماع وبی که از این روش به دست می آید وابسته به ساختا ر گراف وب نمی باشد. الگوریتم پیشنهادی با استفاده از اطلاعات چگونگی استفاده کاربران از وب و پارتیشن بندی گراف وب، جهت تشخیص شباهت صفحات وب پیشنهاد می‌گردد. صفحاتی که در گراف وب به هم متصل هستند به همدیگر شبیه می باشند. به منظور ارزیابی، روش پیشنهادی پیاده سازی گردیده و نتایج ان با نتایج الگوریتم HITS و الگوریتمی دیگر مبتنی اتوماتای یادگیر توزیع شده مقایسه شده است[۱].

ادامه مقاله بدین صورت سازماندهی شده است. در بخش ۲ اتوماتای یادگیر و اتوماتای یادگیر توزیع شده به اختصار معرفی می شوند. در بخش۳ پارتیشن بندی گراف بطور مختصرمعرفی شدهو در بخش ۴ الگوریتم پیشنهادی و در بخش ۵ پس از معرفی مدل استفاده شده برای شبیه سازی، نتایج شبیه سازی ارائه می شود. بخش ۶ نتیجه گیری می باشد.

۲- اتوماتای یادگیر

اتوماتای یادگیر یک مدل انتزاعی است که بطور تصادفی یک اقدام از مجموعه متناهی اقدام‌های خود را انتخاب کرده و بر محیط اعمال می‌کند. محیط اقدام 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 اقدام انتخاب شده و سیگنال تقویتی، وضعیت داخلی خود را بروز کرده و اقدام بعدی خود را انتخاب می‌کند. شکل ۱ نحوه ارتباط بین اتوماتای یادگیر و محیط را نشان می‌دهد.

شکل ۱. ارتباط اتوماتای یادگیر با محیط

محیط را می‌توان توسط سه‌تایی $E=\{\alpha,\beta,c\}$ نشان داد که در آن $\alpha=\{\alpha_1,\alpha_2,...,\alpha_r\}$ مجموعه ورودیها، $\beta=\{\beta_1,\beta_2,...,\beta_r\}$ مجموعه خروجیها و $c=\{c_1,c_2,...,c_r\}$ مجموعه احتمالات جریمه می‌باشد. 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، مجموعه β دارای تعداد متناهی عضو می‌باشد و در محیط از نوع S، تعداد اعضا مجموعه β نامتناهی است. c_i نشان دهنده احتمال نامطلوب بودن سیگنال تقویتی محیط در پاسخ به اقدام α_i می‌باشد. در یک محیط ایستا[۷] مقادیر c_i ها ثابت هستند، حال آنکه در یک محیط غیر ایستا[۸] این مقادیر در طی زمان تغییر می‌کنند. بر اساس اینکه تابع بروز رسانی وضعیت اتوماتای یادگیر (که با اطلاع از اقدام انتخاب شده و سیگنال تقویتی β ، وضعیت بعدی اتوماتای یادگیر را محاسبه می‌کند) ثابت یا متغیر باشد، اتوماتای یادگیر به دو دسته اتوماتای یادگیر با ساختار ثابت و اتوماتای یادگیر با ساختار متغیر تقسیم می‌گردند. در این مقاله از اتوماتای یادگیر با ساختار متغیر استفاده شده است که در ادامه معرفی می‌شود.

اتوماتای یادگیر با ساختار متغیر توسط چهارتایی $\{\alpha,\beta,p,T\}$ نشان داده می‌شود که در آن $\alpha=\{\alpha_1,\alpha_2,\cdots,\alpha_r\}$ مجموعه اقدام‌های اتوماتای یادگیر، $\beta=\{\beta_1,\beta_2,\cdots,\beta_r\}$ مجموعه ورودیهای اتوماتای یادگیر، $p=\{p_1,p_2,\cdots,p_r\}$ بردار احتمال انتخاب هر یک از اقدام‌ها و T، $p(n+1)=T[\alpha(n),\beta(n),p(n)]$ الگوریتم یادگیری اتوماتای یادگیر می‌باشد. الگوریتم‌های یادگیری متنوعی برای اتوماتای یادگیر ارائه شده است که در ادامه یک الگوریتم یادگیری 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 رابطه (۱) تنظیم می‌کند.

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

که a پارامتر پاداش و b پارامتر جریمه می‌باشد. اگر a و b با هم برابر باشند، الگوریتم L_{R-P}[۹]، اگر b از a خیلی کوچکتر باشد، الگوریتم $L_{R\varepsilon P}$[۱۰] و اگر b صفر باشد، الگوریتم L_{R-I}[۱۱] نام دارد [۸].

اتوماتای یادگیری که در بالا معرفی شد، دارای تعداد اقدامهای ثابتی می‌باشد. در بعضی از کاربردها به اتوماتای یادگیر با تعداد اقدام متغیر[۱۲] نیاز می‌باشد [۹]. 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 می‌کند. انتخاب مجموعه اقدامهای فعال اتوماتای یادگیر $V(n)$ توسط یک عامل خارجی و بصورت تصادفی انجام می‌شود. نحوه فعالیت این اتوماتای یادگیر بصورت زیر است.

اتوماتای یادگیر برای انتخاب یک اقدام در زمان n ابتدا مجموع احتمال اقدامهای فعال خود ($K(n)$) را محاسبه و بردار $\hat{p}(n)$ را مطابق رابطه (۲) ایجاد می‌کند. آنگاه اتوماتای یادگیر یک اقدام از مجموعه اقدام‌های *فعال* خود را بصورت تصادفی و بر اساس بردار احتمال $\hat{p}(n)$ انتخاب کرده و بر محیط اعمال می‌کند. 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 در صورت دریافت پاسخ مطلوب بر اساس رابطه (۳) و در صورت دریافت پاسخ

نامطلوب طبق رابطه (۴) بِروز می‌کند. سپس اتوماتای یادگیر بردار احتمال اقدامهای خود $p(n)$ را با استفاده از بردار $\hat{p}(n+1)$ و طبق رابطه (۵) بِروز می‌کند.

$$K(n) = \sum_{\alpha_i \in V(n)} p_i(n)$$
$$\hat{p}_i(n) = prob[\alpha(n) = \alpha_i \mid \alpha_i \in V(n)] = \frac{p_i(n)}{K(n)} \qquad (۲)$$
$V(n)$ is the set of enabled actions

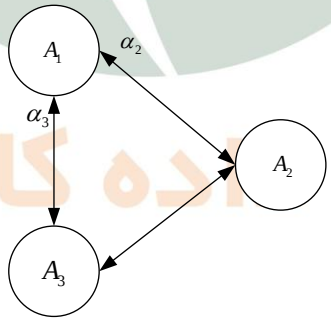
$$\hat{p}_i(n+1) = \hat{p}_i(n) + a.(1 - \hat{p}_i(n))$$
$$\hat{p}_j(n+1) = \hat{p}_j(n) - a.\hat{p}_i(n) \qquad \forall j \ \ j \neq i \qquad (۳)$$

$$\hat{p}_i(n+1) = (1-b).\hat{p}_i(n)$$
$$\hat{p}_j(n+1) = \frac{b}{\hat{r}-1} + (1-b)\hat{p}_j(n) \qquad \forall j \ \ j \neq i \qquad (۴)$$

$$\mathrm{p_i(n+1) = \hat{p}_i(n+1).K(n)} \qquad \text{for all i}, \alpha_i \in V(n)$$
$$\mathrm{p_j(n+1) = p_j(n)} \qquad \text{for all j}, \alpha_j \notin V(n) \qquad (۵)$$

۲-۱- اتوماتای یادگیر توزیع‌شده

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 با یک گراف جهت‌دار مدل کرد. بصورتی که مجموعه گره‌های آنرا مجموعه‌ای از اتوماتای یادگیر و یالهای خروجی هر گره مجموعه اقدامهای متناظر با اتوماتای یادگیر متناظر با آن گره است. هنگامی که اتوماتا یکی از اقدامهای خود را انتخاب می‌کند، اتوماتایی که در دیگر انتهای یال متناظر با آن اقدام قرار دارد، فعال می‌شود. بعنوان مثال در شکل ۲ هر اتوماتا ۲ اقدام دارد. اگر اتوماتای A_1 اقدام α_3 خود را انتخاب کند، آنگاه اتوماتای A_3 فعال خواهد شد. در گام بعد، اتوماتای A_3 یکی از اقدامهای خود را انتخاب می‌کند که منجر به فعال شدن یکی از اتوماتاهای یادگیر متصل به A_3 می‌شود. در هر لحظه فقط یک اتوماتای یادگیر در اتوماتای یادگیر توزیع‌شده فعال می‌باشد. بصورت رسمی، یک اتوماتای یادگیر توزیع‌شده با n اتوماتای یادگیر توسط یک گراف (A,E) تعریف می‌شود که $\{A_1, A_2, \ldots, A_n\}$ مجموعه اتوماتا و $E \subset A \times A$ مجموعه لبه‌های گراف است بطوریکه لبه (i,j) متناظر با اقدام a_j از اتوماتای A_i است. اگر بردار احتمال اقدامهای اتوماتای یادگیر A_j با p^j نشان داده شود، آنگاه p_m^j احتمال انتخاب اقدام α_m از اتوماتای یادگیر A_j را نشان می‌دهد که احتمال انتخاب لبه خروجی (j,m) از میان لبه‌های خروجی گره j می‌باشد. برای کسب اطلاعات بیشتر در باره اتوماتای یادگیر توزیع شده و کاربردهای آن می توان به[۱۰][۱۱][۱۲][۱۳][۱۴][۱۵] مراجعه نمود.



شکل ۲. اتوماتای یادگیر توزیع‌شده

۳- پارتیشن بندی گراف

پارتیشن بندی گراف، یک گراف را به چندین زیرگراف تقسیم میکند. هدف اصلی یک پارتیشن بندی مینیمم کردن هزینههای تعریف شده روی لبههای متصل پارتیشن ها با کمترین برش لبه[۱۳]، می باشد. روش های زیادی برای تعیین هزینه خارجی پارتیشنبندی گراف وجود دارد اما دو نکته که اغلب بطور وسیعی بکار برده می شود، ماکزیمم وزن لبه های بین رئوس است که روی پارتیشن های مختلف قرار می گیرد و وزن کلی همه لبه های متصل به پارتیشن های مشخص است. پارتیشن بندی گراف جزو مسائل NP می باشد بهمین دلیل روشهای مختلفی برای حل وبهینه کردن این مساله مطرح شده است.

یکی از بهترین الگوریتم های پارتیشن بندی گراف، الگوریتم مطرح شده در[۱۶] می باشد بطوریکه پارتیشن بندی با یک پارتیشن دلخواه شروع می گردد و سپس برای کاهش هزینه خارجی توسط یک سری از تغییرات داخلی در زیرمجموعه های پارتیشن ها اقدام می کند این کار تا وقتی که دیگر امکان بهبودی وجود ندارد، ادامه می یابد. برای اجتناب از بهینه سازی محلی، الگوریتم بطور متوالی برای بدست آوردن تعدادی از پارتیشن های بهینه محلی از میان آنها درخواست میشود تا بهترین پارتیشن انتخاب شود. بعداً در [۸] کارایی الگوریتم مطرح شده در[۱۶] را بهبود دادند.

در پارتیشن بندی سنتی، پارتیشن بندی بر روی کل گراف انجام می شد. این الگوریتم ها با افزایش رئوس بسیار کند بوده و کیفیت پایینی داشتند. پارتیشن بندی چند سطحی روشی کاملا متفاوت نسبت به روشهای سنتی دارند [۱۷][۱۸][۱۹]. روش کار این الگوریتم ها به این صورت می باشد که گراف را بوسیله فروریختن[۱۴] رئوس و لبه ها کاهش می دهند. پس از انجام این عمل پارتیشن بندی بر روی گراف کوچک شده انجام می شود. پارتیشن بدست آمده، کوچک تر از گراف اصلی می باشد. به همین منظور جهت بازیابی گراف اصلی عملیات بازیابی گراف [۱۵] انجام می گیرد که یک پارتیشن برای گراف اصلی می سازد. طرح چندسطحی برای بهبود کارایی پارتیشن بندی یک گراف بزرگ و بهینه کردن هزینه وضعیت یک پارتیشن بکار می رود[۲۰].

۴- روش پیشنهادی

روش پیشنهادی برای تشخیص اجتماعات وب مبتنی بر روش معرفی شده در [۱][۲] می باشد. در [۲] مجموعه ای از صفحات Hub و Authority به عنوان وب معرفی شده اند که تنها از پیوندهای بین صفحات وب برای تعیین صفحات Hub و Authority استفاده می کند، این روش از دقت کافی برخوردار نمی باشد. در [۱] علاوه بر پیوند بین صفحات وب از رفتار کاربران استفاده شده است که در نتیجه میزان تعداد صفحات نامرتبط کاهش می یابد. که با افزایش تعداد صفحات وب کارایی آن کاهش می یابد. در روش پیشنهادی محتوای صفحات وب نیز در دو مرحله مورد استفاده قرار می گیرد: برای ساخت مجموعه ریشه و برای اصلاح اجتماع وب بدست آمده. یکی از مشکلات روش های کاوش استفاده از وب ، اطلاعات ناصحیح می باشد. چرا که در برخی موارد، کاربران در وب سرگردان می شوند و بدون داشتن هدف مشخصی بر روی صفحات مختلف کلیک می کنند و گاهی اوقات کاربران به صفحه ای که پیمایش را آغاز کرده بودند بر می گردند.

مراحل روش پیشنهادی به شرح زیر است:

• **ایجاد مجموعه ریشه:** در مرحله اول، موضوع اجتماع وب مورد نظر کاربر، به عنوان ورودی به الگوریتم ارائه می شود . سپس مجموعه ای ازصفحات مرتبط با این موضوع انتخاب شده و مجموعه ریشه ساخته می شود.

• **ایجاد مجموعه پایه :** در این مرحله مجموعه ریشه که در مرحله قبل ایجاد شد، با استفاده از صفحاتی که اعضای مجموعه با آنها پیوند دارند، گسترش می یابد و مجموعه پایه را می سازند. برای این منظور ابتدا صفحاتی که صفحات مجموعه ریشه به آنها اشاره می کنند، به مجموعه پایه اضافه می شوند. سپس صفحاتی که به صفحات مجموعه ریشه اشاره می کنند، به این مجموعه اضافه می شوند.

• **ایجاد اتوماتای یادگیر توزیع شده :** در این مرحله برای هر یک از صفحات مجموعه پلیه یک اتوماتای یادگیر ایجاد می شود . اعمال هر یک از این اتوماتاهای یادگیر متناظر با صفحاتی است که صفحه جاری (صفحه مربوط به این اتوماتا) به آنها اشاره می کند. در ابتدا مولفه های بردار احتمال هر اتوماتا به صورت مساوی مقدار دهی اولیه می شوند.

•**محاسبه امتیازHub و Authoruty** : عملیات زیر تا رسیدن به نتیجه قابل قبول تکرار می گردد.

- کاربران به پیمایش صفحات وب می پردازند و مسیرهای پیمایش شده توسط آنها، در سیستم ثبت می شود.
- کلیه مسیرهای پیمایش شده استخراج و تعداد دفعات پیمایش آنها به دست می آید.
- بردار احتمالات اتوماتاهای یادگیر هر یک از صفحات مجموعه پایه به شرح زیر به روز می شوند.

در این قسمت، الگوریتمی ترکیبی مبتنی بر اتوماتای یادگیر توزیع شده و پارتیشن بندی گراف به منظور تشخیص شباهت صفحات وب پیشنهاد می‌گردد. در روش ارایه شده برای اسناد با تعداد بیشتر نتایج بهتری تولید می شود. برای تعیین شباهت بین صفحات در یک مجموعه با n صفحه، از یک اتوماتای یادگیر توزیع شده با n اتوماتای یادگیر که هر یک n-۱ عمل دارند، استفاده می‌شود. برای هر اتوماتای یادگیر در هر زمان تنها یک زیرمجموعه از عمل هایش فعال و قابل استفاده هستند. هر کدام از اعمال یک اتوماتای یادگیر، متناظر با یکی از صفحات در مجموعه صفحات و احتمال انتخاب این عمل در بردار احتمالات، ارتباط این صفحه با صفحه متناظر با آن عمل می باشد. برای هر صفحه j یک اتوماتای یادگیر در نظر می گیریم. انتخاب عمل j توسط اتوماتای یادگیر i به معنی فعال کردن اتوماتای یادگیر j متناظر با صفحه j می باشد. در صورتیکه عمل انتخاب شده k امین عمل اتوماتای i باشد یعنی ($a^i_k = j$) احتمال متناظر این عمل یعنی p^i_k بعنوان میزان ارتباط صفحه های i و j در نظر گرفته می شود.

هنگامی که کاربری در صفحه i قرار دارد، اتوماتای یادگیر متناظر با آن فعال است. حرکت کاربر از صفحه i به صفحه j، به منزله انتخاب عمل j از اتوماتای i می باشد که منجر به فعال شدن اتوماتای یادگیر j می شود. این عمل از اتوماتای یادگیر توسط محیط پاداش یا جریمه داده می شود. پاداش دادن به اعمال انتخاب شده توسط اتوماتای یادگیر به سه عامل بستگی دارد:

۱. مسیرهای طی شده توسط کاربران
۲. فاصله صفحات در مسیرهای طی شده
۳. پیوند بین صفحات در گراف وب

شکل ۳. مسیر کاربر برای محاسبه رابطه تعدی

نحوه پاداش و جریمه در این الگوریتم با در نظر گرفتن رابطه تراگذری می باشد. مثلاً اگر کاربر مسیری را که در شکل ۳ نشان داده شده است، طی کند با فرض اینکه وجود شباهتی بین محتوای این صفحات موجب این انتخاب کاربر شده است، به اعمال هر یک از اتوماتاهای متناظر با این صفحات پاداش داده می شود.

- عمل j از اتوماتای i، عمل k از اتوماتای j، عمل l از اتوماتای k
- عمل k از اتوماتای i، عمل l از اتوماتای j
- عمل l از اتوماتای i

پاداشی که به هر یک از اعمال j, k, l از اتوماتای یادگیر i داده میشود، طبق یک ضریب کاهشی، به ترتیب کاهش داده میشود. چراکه فاصله صفحات متناظر آنها در مسیر کاربر به ترتیب افزایش مییابد. همچنین در صورتی که صفحه i, j در گراف وب به یکدیگر متصل و صفحه j و k غیرمتصل باشند، پاداشی که عمل j در اتوماتای i می گیرد، بیشتر از پاداشی است که عمل k در اتوماتای j می گیرد.

جریمه دادن اعمال انتخاب شده توسط اتوماتای یادگیر که در این مقاله مطرح شده، به دو عامل بستگی دارد:

۱. وجود دور در مسیر حرکت کاربر
۲. خارج شدن از زیر گراف مربوط به اتوماتا

اگر در مسیر حرکت، کاربر از زیر گرافی که اتوماتا در آن قرار داشت خارج شود نشان دهنده سرگردانی وی در وب و یا عدم رضایت او از اطلاعات صفحات مشاهده شده می باشند و مجازات می شوند. در حالت دوم، وقتی جریمه به حرکت کاربر داده می شود که دور درمسیر حرکتی آن باشد. همچنین اعمالی که قسمتی از یک دور باشند نشان دهنده حرکت اشتباه کاربر، سرگردانی وی در وب و یا عدم رضایت او از اطلاعات صفحات مشاهده شده می باشند و مجازات می شوند. هر چه طول این دور بیشتر باشد، میزان جریمه بیشتر خواهد بود.

پاداش و جریمه دادن به اعمال اتوماتاهای یادگیر طبق یک الگوریتم یادگیری انجام میشود که با استفاده از آن احتمال اعمال اتوماتای یادگیر به روز رسانی میشوند. در این مقاله از الگوریتم یادگیری $L_{R\varepsilon P}$ استفاده شده است. ضریب پاداش و جریمه در این الگوریتم با فرمول (۶) محاسبه می شود.

$$a = \frac{1}{\text{steps between h and i in the path}}\omega + \lambda \qquad (۶)$$

که در آن ω=۰.۰۲ پارامتر ثابت، h و i دو صفحه که در یک مسیر توسط کاربر مشاهده شده اند (عمل i از اتوماتای h) و λ=۰.۰۰۲ ثابتی است که اگر صفحه h و i غیرمتصل باشند برابر با صفر و در غیر این صورت برابر با یک مقدار ثابت است.

پارامتر b را نیز که پارامتر جریمه می باشد، طبق رابطه (۷) محاسبه می کنیم:

$$b = (\text{steps in cycle containing k and l}) * \beta \qquad (۷)$$

که در آن β=۰.۰۰۲ مقدار ثابت می باشند.

در ابتدای الگوریتم، اقدام اتوماتاهایی که سند متناظر آن در گراف به هم متصل شده اند فعال و مابقی غیرفعال می باشند. با حرکت یک کاربر از سند i به سند j، اقدام متناظر با آن سند (اقدام j) در اتوماتای یادگیر i فعال می‌شود. در این حالت اگر هر دو سند در یک پارتیشن قرار داشته باشند و در مسیر دوری نباشد ، اتوماتای یادگیر i به اقدام j خود پاداش می‌دهد در غیر این صورت جریمه می شود و این عمل با استفاده از رابطه تراگذری تا انتهای مسیر ادامه پیدا می کند. سپس اتوماتای یادگیر j در اتوماتای یادگیر توزیع‌شده فعال می‌شود و مراحل فوق تا پایان حرکت کاربر در مجموعه صفحات ادامه می‌یابد. در صورت وجود دور، اگر اقدام متناظر با آن سند(اقدام j) در اتوماتای یادگیر i فعال باشد به عمل متناظر در سند جریمه داده می شود(در جریمه نیز رابطه تراگذری در نظر گرفته می شود). در هر زمان، شباهت دو سند i و j برابر با احتمال انتخاب اقدام j در اتوماتای i است. در صورتیکه اقدام مورد نظر غیرفعال باشد، شباهت دو سند صفر در نظر گرفته می‌شود.

– پس از اصلاح احتمال انتخاب اعمال اتوماتای یادگیر توزیع شده، امتیاز Hub و Authority هر یک از صفحات مجموعه پایه طبق فرمول (۸) و (۹) بروز می شود.

$$Hub(i) = \sum_{\forall i \to j} r(i, j) \times Authority(j) \quad (۸)$$

$$Authority(i) = \sum_{\forall j \to i} r(j, i) \times Hub(j) \quad (۹)$$

که r(i,j) میزان رابطه صفحه i و j در مدل ساختاری می باشد. از آنجا که این مقدار با توجه به نحوه پیمایش کاربران به دست آمده است، در محاسبه امتیاز Hub و Authority علاوه بر لینک های بین صفحات، از نحوه پیمایش کاربران نیز استفاده شده است.

تشخیص اجتماع وب: در این مرحله پس از محاسبه امتیاز Hub و Authority صفحات وب مجموعه پایه، تعداد ثابتی از صفحات (WC_Num) با بیشترین امتیاز Hub و تعداد ثابتی از صفحات (WC_Num) با بیشترین امتیاز Authority به عنوان صفحات اجتماع وب در نظر گرفته می شوند.

اصلاح اجتماع وب: همان طور که در بالا اشاره شد، از آنجا که صفحات Hubی که به صفحاتی با موضوعات مختلف اشاره می کنند، چندان از لحاظ مفهومی با موضوع اجتماع وب در ارتباط نیستند، در این مرحله اجتماع وب به دست آمده را اصلاح می کنیم به آن معنی که صفحات Hubی که به صفحات مرتبط با بیش از T موضوع اشاره می کنند، از اجتماع وب حذف می شوند.

۵- نتایج شبیه سازی

در این بخش نتایج بدست آمده از شبیه‌سازی الگوریتم پیشنهادی ارائه می‌شود.

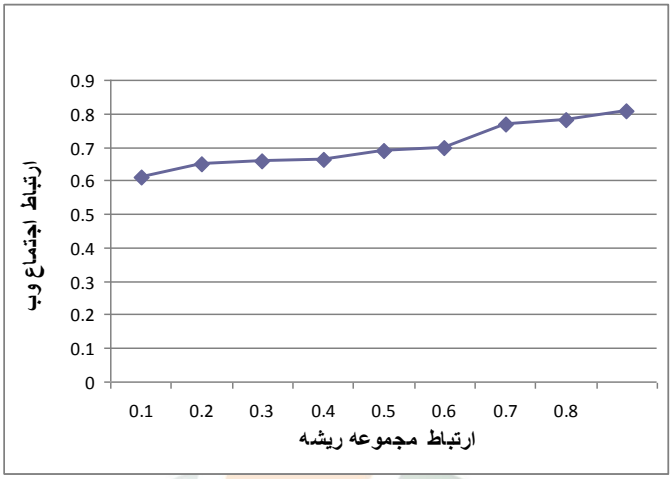
۵-۱- مدل شبیه‌سازی

برای شبیه‌سازی الگوریتم پیشنهادی و مقایسه آن با سایر روشها از مدل معرفی شده در [۶] برای نشان دادن ساختار صفحات وب و چگونگی استفاده کاربران، استفاده شده است. اعتبار این مدل توسط Lui و همکاران [۶] با استفاده از اطلاعات استفاده از وب چندین سایت وب بزرگ مانند مایکروسافت، تایید شده است. بر این اساس، در این مقاله مطابق با مدل رفتار کاربران، پروفایل علاقه کاربران بصورت توزیع قانون-توانی[۱۶] و توزیع محتوای صفحات وب بصورت توزیع نرمال در نظر گرفته شده است. سایر پارامترهای استفاده شده در مدل[۶] برای شبیه سازیهای انجام شده در این مقاله در جدول ۱ نشان داده شده است. همچنین پارتیشن بندی گراف توسط نرم افزار Metis انجام شده است[۲۱]. در این نرم افزار پارتیشن بندی با استفاده از روشهای چند سطحی انجام می شود. این نرم افزار از الگوریتمهایی برای پارتیشن بندی گراف استفاده می کند که در سریعترین زمان، زیر گراف هایی با کیفیت بالا ایجاد کند.

جدول ۱: پارامترهای شبیه‌سازیها

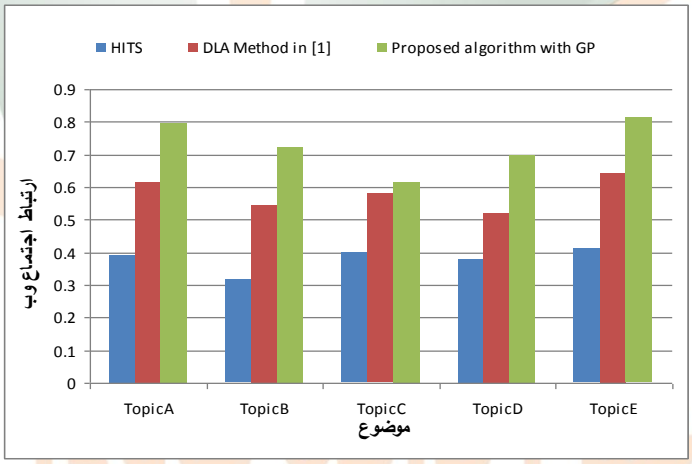
حد آستانه ایجاد اتصال	۰/۷
تعداد کاربران	۵۰۰۰ ۰
تعداد اسناد	۵۰۰
تعداد موضوع‌ها	۵
T_c مقدار ثابت سند اولیه (صفحه اولیه سایت) در موضوعات مختلف	۰/۲
ΔM_t^c ضریب ثابت کاهش اشتیاق کاربر	–
ΔM_t^v ضریب متغیر کاهش اشتیاق کاربر	–
α_u پارامتر توزیع قانون-توانی توزیع احتمال علایق کاربران	۱
ϕ ضریب پاداش دریافتی از مشاهده یک سند	۱/۲
λ ضریب جذب اطلاعات از یک سند توسط یک کاربر	۰/۵
μ_m میانگین توزیع نرمال ΔM_t^c	۵/۹۷
σ_m واریانس توزیع نرمال ΔM_t^c	۰/۲۵
μ_t میانگین توزیع نرمال برای مقدار افزایش یک گره برای یک موضوع خاص	–
α_p پارامتر توزیع قانون-توانی توزیع احتمال وزنهای مطالب برای هر سند	۳
σ_t 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
تعداد صفحات در هرپارتیشن	۲۵

آزمایش ۱ : در این آزمایش تاثیر نحوه انتخاب مجموعه ریشه بر میزان ارتباط اجتماع وب به دست آمده با موضوع اعلام شده توسط کاربر، بررسی شده است. برای این منظور میانگین میزان ارتباط اسناد مجموعه ریشه با موضوع مورد نظر تغییر داده شده است و در هر حالت میانگین میزان ارتباط صفحات اجتماع وب با این موضوع محاسبه شده است. نتایج در شکل ۴ نشان داده شده است.



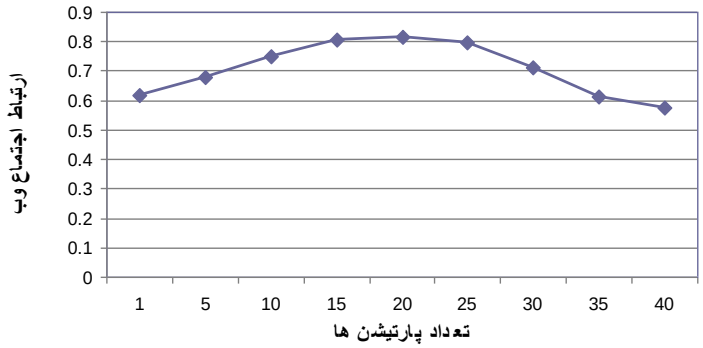
شکل ۴. تاثیر مجموعه ریشه در تشخیص اجتماع وب

آزمایش ۲ : در این آزمایش روش پیشنهادی با روش ارایه شده در [۱] و الگوریتم HITS مقایسه شده است. این آزمایش ۵۰ بار تکرار و از میانگین نتایج استفاده شده است. معیار ارزیابی میانگین میزان ارتباط صفحات اجتماع وب تولید شده با موضوع مورد نظر است. همان طور که در شکل۵ مشاهده می شود، میزان ارتباط اجتماع وب تولید شده، با استفاده از الگوریتم پیشنهادی نسبت به الگوریتم ارایه شده در [۱] و HITS افزایش یافته است.



شکل ۵. مقایسه روش پیشنهادی با الگوریتم [۱] و HITS

آزمایش ۳: در این آزمایش تعداد صفحات موجود در یک پارتیشن را مورد بررسی قرار می دهیم. با افزایش تعداد صفحات در پارتیشن روش پیشنهادی نتایجی مشابه نتایج الگوریتم ارایه شده در [۱] خواهد داشت. شکل ۶ نشان می دهد که بیش از اندازه کوچک شدن پارتیشن نیز کارایی الگوریتم را کم خواهد کرد. هر چقدر تعداد پارتیشن افزایش یابد نشان دهنده این است که تعداد صفحات در هر پارتیشن کاهش یافته است.



شکل ۶. تاثیر تعداد پارتیشن در تشخیص اجتماعات وب

۶- نتیجه گیری

در این مقاله با استفاده از ترکیب تکنیک های کاوش ساختار وب، کاوش محتوای وب، کاوش استفاده از وب و با به کارگیری اتوماتای یادگیر و پارتیشن بندی گراف، روش ارایه شده در[۱] بهبود داده شد. برای این منظور گراف بزرگ وب با استفاده از تکنیک های پارتیشن بندی کوچک و پردازش های لازم بر روی این ساختار انجام گردید. نتایج حاصل نشان از بهبود الگوریتم تشخیص اجتماعات وب معرفی شده در [۱] را نشان می دهد. نتایج مقایسه روش پیشنهادی با روش [۱]برای تشخیص اجتماعات وب نشان داد که پارتیشن بندی گراف موجب افزایش کارایی الگوریتم مطرح شده در [۱] می گردد.

ویژگی ه‌ای این الگوریتم عبار تند از:

۱- ترکیب تکنیک های کاوش ساختار وب، کاوش استفاده از وب و کاوش محتوای وب، ۲- بهبود الگوریتم HITS با در نظر گرفتن رفتار کاربران علاوه بر ساختار پیوند بین صفحات، ۳- به کارگیری اتوماتای یادگیر توزیع شده برای یادگیری امتیازات Hub و Authority، ۴- استفاده از پارتیشن بندی گراف جهت یاد گیری بهتر، ۵- کاهش تاثیر اطلاعات ناصحیح موجود در نحوه پیمایش کاربران، ۶- به کارگیری روش های کاوش محتوای وب برای اصلاح اجتماعات تشخیص داده شده و ۷- عدم وابستگی به یک ساختار خاص برای تشخیص اجتماع وب

7- مراجع

[1] سارا مطیعی و محمدرضا میبدی، **" تشخیص اجتماعات وب با استفاده از اتوماتای یادگیر توزیع شده"**، اولین کنفرانس داده کاوی ایران، تهران، ایران

[2] Gibson, D., Kleinberg, J. M. and Raghavan, P., "*Inferring Web Communities from Link Topology*", In Proc. of the 9th ACM Conference on Hypertext and Hypermedia. Pittsburgh, PA, pp. 225-234, 1998.

[3] Kumar, R., Raghavan, P., Rajagopalan, S. and Tomkins, A., "*Trawling the Web for Emerging Cyber-Communities*", Proc. of the 8th WWW Conference, 1999.

[4] Imafuji, N. and, Kitsuregawa, M., "*Effects of Maximum Flow Algorithm on Identifying Web Community*", Proc. of the 4th international Workshop on Web information and Data Management (McLean, Virginia, USA, November 08 - 08, 2002). WIDM '02. ACM Press, New York, NY, pp. 43-48, 2002.

[5] Flake, G., Lawrence, S. and, Giles, C.L., "*Efficient Identification of Web Communities*", the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston, MA, pp. 150-160, 2000.

[6] J. Liu, S. Zhang, and J. Yang, "*Characterizing Web Usage Regularities with Information Foraging Agents*," IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 4, April 2004, pp. 566-584.

[7] Flake, G. W., Lawrence, S., Giles, C. L. and Coetzee, F. M., "*Self-Organization and Identification of Web Communities*", IEEE Computer, Vol. 35, No. 3, pp. 66-71, 2002.

[8] C. M. Fiduccia and R. M. Mattheyses, "*A linear-time heuristic for improving network partitions*". In Proceedings of the 19th Conference on Design Automation. 1982: IEEE Press.

[9] M. A. L. Thathachar and R. Harita Bhaskar, "*Learning Automata with Changing Number of Actions*," IEEE Transactions on Systems Man and Cybernetics, vol. 17, no. 6, Nov. 1987, pp 1095-1100.

[10] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell, "*WebWatcher: A Learning Apprentice for the World Wide Web*," Proceedings of AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, AAAI Press, 1995, pp 6–12.

[11] Mike Perkowitz and Oren Etzioni, "*Adaptive Web Sites*," Communications of ACM, vol. 43, no. 8, 2000, pp. 152-158.

[12] B. Mobasher, R. Cooley, and J. Srivastava, "*Automatic Personalization Based on Web Usage Mining*," Communications of the ACM, vol. 43, no. 8, 2000, pp. 142–151.

[13] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "*Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization*," Data Mining and Knowledge Discovery, vol. 6, no. 1, 2002, pp. 61-82.

[14] Pierrakos, G. Paliouras, C. Papatheodorou, and C. D. Spyropoulos, "*Web Usage Mining as a Tool for Personalization: A Survey*," User Modeling and User-Adapted Interaction, vol. 13, no. 4, 2003, pp. 311-372.

[15] M. R. Meybodi and 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).

[16] B. W. Kernighan and S. Lin, "*An efficient heuristic procedure for partitioning graphs*". The Bell System Technical journal, 1970. 49(2): p. 291-307.

[17] C. K. Cheng and Y. C. A Wei, "*An improved two-way partitioning algorithm with stable performance*". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1991. 10(12): p. 1502-1511.

[18] L. Hagen and A. B. Kahng, "*A new approach to effective circuit clustering*". In Proceedings of the IEEE/ACM International Conference on Computer-Aided Design. 1992. Santa Clara, CA, USA.

[19] T. N. Bui and C. A. Jones, "*heuristic for reducing fill in sparse matrix factorization*". in Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing. 1993. Norfolk, Virginia, USA.

[20] A. Pothen, H. D. Simon and K. P. Liou, "*Partitioning sparse matrices with eigenvectors of graphs*". SIAM Journal on Matrix Analysis and Applications, 1990. 11(3): p. 430-452.
[21] G. Karypis and V. Kumar, "*METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*". Version 5.0pre2. 2007: Minneapolis.
[22] Toyoda, M. and Kitsuregawa, M., "*Creating a Web Community Chart for Navigating Related Communities*", In Proc. Hypertext 2001, pp.103-112, 2001.

---

۱ Web mining
۲ Web Structure Mining
۳ Links
٤ Link Analysis
٥ Related Page Algorithm
٦ Bisection
۷ Stationary
۸ Non-Stationary
۹ Linear Reward-Penalty
۱۰ Linear Reward-Penalty
۱۱ Linear Reward Inaction
۱۲ Learning automata with changing number of actions
۱۳ Minimum edge-cut
۱٤ Collapsing
۱٥ Uncoarsen
۱٦ Power-law