کارگاه‌های آموزشی مرکز اطلاعات علمی

آموزش مهارت های کاربردی در تدوین و چاپ مقاله

اصول تنظیم قراردادها

مقاله نویسی علوم انسانی

3rd. INTERNATIONAL CONFERENCE ON
RESEARCH IN ENGINEERING, SCIENCE AND TECHNOLOGY
Batumi GEORGIA 5 JUNE. 2016
www.restconf.org
سومین کنفرانس بین المللی پژوهش در مهندسی، علوم و تکنولوژی
باتومی - گرجستان ۱۶ خردادماه ۱۳۹۵

ارزیابی همگرایی و خوشه بندی اسناد وب با استفاده از الگوریتم الکترو مغناطیس ترکیبی بر اساس مدل رفتار کاربر

سعید مصطفی پور گندلی
دانشگاه آزاد اسلامی واحد آبادان
fdx_mg@yahoo.com

دکتر محمدحسین یکتایی
دانشگاه آزاد اسلامی واحد آبادان
my.yektaie@gmail.com

چکیده

الگوریتم الکترومغناطیس به عنوان یکی از روش های نوظهور در زمینه بهینه سازی مبتنی بر هوش دسته جمعی، هوش گروهی به حساب می آید. همچنین این روش فراابتکاری تا حدی جزء الگوریتم های جمعیت محور نیز محسوب می گردد. این الگوریتم از مکانیسم جذب و دفع بارهای الکترونیکی در مبحث تئوری های الکترومغناطیس برای تعیین پاسخ بهینه استفاده کرده و عملکرد مناسبی در حل مسائل کمینه سازی پیوسته و همچنین با تغییراتی در حل مسائل گسسته دارد. در عصر حاضر مهم‌ترین منبع اطلاعاتی صفحات وبی است که بر روی اینترنت یافت می‌شوند که این صفحات به طرز فزاینده‌ای رو به افزایش هستند. پژوهشگران تلاش‌های زیادی در خصوص دسته‌بندی و خوشه‌بندی صفحات وب انجام داده‌اند. روش‌های موجود از خصوصیات ذاتی اسناد به‌منظور خوشه‌بندی استفاده می‌کنند. به نظر ما تعامل کاربران با سیستم‌های اطلاعاتی حاوی مطالب مفیدی است. به‌عنوان مثال اگر کاربران با سلایق مشابه به دو سند دسترسی پیدا کنند، نشان‌دهنده‌ی مشابه بودن خود دو سند است. این مطالب می‌تواند کمک کنند تا اسناد براساس خصوصیات دیگری علاوه بر خصوصیات ذاتی آن‌ها (خصوصیات رفتاری کاربران) خوشه‌بندی شوند. در این پروژه تلاش در همین زمینه گردیده است و سعی شده با استفاده از استخراج اطلاعات رفتار کاربران، اطلاعات دقیق‌تر و بهتری با هزینه‌های محاسباتی کمتر از صفحات وب استخراج و در زمینه خوشه-بندی اسناد استفاده شود؛ بنابراین در این تحقیق با استفاده از خوشه‌بندی و الگوریتم الکترومغناطیسسعی شده تا هزینه‌های محاسباتی را کاهش و دقت روش خوشه‌بندی اسناد را افزایش یابد. دلیل اصلی این کار بهینه‌سازی روش خوشه‌بندی برای انتخاب مراکز بهینه برای خوشه‌ها است. نتایج این کار تحقیقاتی بهینگی مناسب در زمینه دقت تشخیص در خوشه‌بندی صفحات وب و همین‌طور کاهش هزینه محاسباتی در خوشه‌بندی صفحات وب را نشان می‌دهد.

واژگان کلیدی: رفتار کاربر، خوشه بندی، الگوریتم الکترومغناطیس

مقدمه

تحقیقاتی در حوزه‌های مختلف مانند مدل‌سازی کاربر و وب‌کاوی کاربرد وب جهت استفاده از رفتار کاربر در وب به‌صورت ضمنی انجام‌گرفته است. مدل کاربر را هم می‌توان فقط بر مبنای داده‌های کاربرد وب ساخت که درکی سطحی از الگوها می‌دهد و هم می‌توان از محتوای صفحات برای ایجاد مدل بهتر استفاده کرد. در این پروژه، بر روی آن دسته از کارها در زمینه‌ی مدل‌سازی کاربر تأکید می‌کنیم که در حوزه‌ی شخصی‌سازی وب انجام‌گرفته‌اند و از محتوای صفحات وب برای بهبود مدل بهره گرفته‌اند(B. Mobasher et al, 2000).

دنیای وب منبع عظیمی از اطلاعات است که روزبه‌روز بر حجم آن افزوده می‌شود. در حال حاضر با رشد روز افزون این پدیده، حجم گسترده‌ای از منابع اطلاعاتی حوزه‌های مختلف مثل کتب، مجلات، پایان‌نامه‌ها، طرح‌های پژوهشی و پایگاه‌های اطلاعاتی در بسترهای جدید و با قابلیت‌های متنوعی در مقایسه با منابع چاپی قابل‌دسترسی هستند. سردرگمی در میان این‌همه اطلاعات و حجم زیاد داده‌های ذخیره‌شده در وب و دست‌کاری داده‌ها برای یک پرس‌وجوی ساده، نیاز به ابزارهای پردازش مناسب برای استخراج اطلاعات مربوط دارد. مفهوم وب‌کاوی نخستین بار توسط *Etzioni* در سال ۱۹۹۶ مطرح شد. وب‌کاوی به بیان ساده، استفاده از روش‌های داده‌کاوی برای بازیابی، استخراج و ارزیابی (تشخیص و آنالیز) اطلاعات به‌صورت خودکار از داده‌های وب، مستندات و سرویس‌های آن است. تکنیک‌های وب‌کاوی را به سه دسته وب‌کاوی بر اساس محتوا، وب‌کاوی بر اساس ساختار و وب‌کاوی بر اساس رفتار کاربری می‌توان تقسیم‌بندی نمود. وب‌کاوی بر اساس محتوا به تشریح و کشف اطلاعات مفید و قابل‌استفاده از محتویات، داده‌ها و مستندات موجود در وب می‌پردازد، وب‌کاوی ساختاری برای کشف مدل طرح‌بندی و ساختار پیوندهای وب استفاده می‌شود و وب‌کاوی بر اساس رفتار کاربری با استفاده از داده‌های مشتق شده از تأثیرات رفتار کاربر بر روی وب، الگوهای رفتاری کاربران برای دسترسی به سرویس‌های وب را به‌صورت خودکار کشف می‌کند، (S. Chakrabarti et al, 1998) .کشف الگوها یک مؤلفه کلیدی در وب‌کاوی است که الگوریتم‌ها و تکنیک‌های مختلفی در چند زمینه تحقیقی ازجمله داده‌کاوی، یادگیری ماشینی، آمارشناسی و الگوشناسی را می‌پوشاند. یکی از کاربردهای مهم کشف الگو در وب‌کاوی خوشه‌بندی است(R O. Duda et al, 2000). در تحقیقات صورت گرفته، در زمینه داده کاوی *(J. Furnkranz, 2010)* اصولاً مدل رفتار کاربر برای خوشه‌بندی به کار گرفته نشده‌اند. بنابراین در این تحقیق تلاش خواهد شد؛ که برای خوشه‌بندی صفحات از مدل رفتار کاربر استفاده کرده و صفحات را با توجه به رفتار کاربران با استفاده از الگوریتم الکترومغناطیس، خوشه‌بندی شود. الگوریتم الکترومغناطیسم برای حل مسائل بهینه سازی کاربرد دارد. الگوریتم برای حل مسائل از خاصیت جاذبه–دافعه ذرات باردار استفاده می کند. در این الگوریتم هر پاسخ به عنوان یک ذره باردار در نظر گرفته می شود. حال ذره هایی که بهینه تر باشند، بار بیشتری دارند و می توانند ذرات دیگر را به سمت خود جذب کنند و ذراتی که بهینگی کمتری دارند، باعث دفع دیگر ذرات می شوند. ایده اصلی در این الگوریتم بر این پایه استوار است که در اطراف نقاط خوب ممکن است نقاط بهتری یافت شود. به همین دلیل نقاط ضعیف به سمت نقاط بهینه حرکت داده می شوند(J. Holland, 1978).

روش تحقیق

در این تحقیق ما برای اولین بار برای حل مسائل خوشه‌بندی اسناد وب بر اساس مدل رفتار کاربر، از تکنیک شروع مجدد استفاده خواهیم کرد الگوریتم الکترومغناطیس و همگرایی آن را با الگوریتم الکترو مغناطیس ترکیبی و k-meanes را مورد تحلیل و بررسی قرار خواهیم داد. الگوریتم الکترومغناطیس (EM) یکی از روش‌های نوظهور در زمینه بهینه‌سازی مبتنی بر هوش دسته‌جمعی است که اولین بار در سال ۲۰۰۳ توسط بیربیل و فنگ (Birbil, S. I, 2003) معرفی شد. این روش بهینه‌سازی که با الهام گیری از قوانین حاکم بر سیستم‌های الکترواستاتیکی کار می‌کند، ذاتاً برای حل مسائل کمینه‌سازی پیوسته در حالت بدون قید مناسب است. اما بااین‌حال با اعمال تغییرات جزئی در نحوه تعریف تابع هدف می‌توان از آن برای حل مسائل بهینه‌سازی تحت قید نیز استفاده کرد. تاکنون ازاین‌روش بهینه‌سازی در حل مسائل مختلف مهندسی با موفقیت استفاده‌شده است. یکی از نقاط قوت الگوریتم EM تعداد کم پارامترهای مورداستفاده در آن است که همین امر باعث می‌شود تا در اکثر مواقع بتوان مقدار مناسب آن‌ها را با یک آزمون و خطای ساده تعیین کرد. در الگوریتم EM نیز همانند سایر الگوریتم‌های بهینه‌سازی مبتنی بر هوش دسته‌جمعی، موقعیت هر ذره با استفاده از نیروی وارده از طرف سایر ذرات بر آن مرتباً تغییر داده می‌شود. شبیه‌سازی‌های مختلف نشان می‌دهند که الگوریتم EM از کارایی بسیار خوبی برای حل مسائل بهینه‌سازی پیوسته حتی در مواجهه با مسائل آزمون پیچیده برخوردار است. ازآنجایی‌که در این الگوریتم با افزایش تعداد تکرارها، ذرات همگی سرانجام در یک نقطه تجمع نمی‌یابند و از طرفی بهترین ذره به‌دست‌آمده در تکرار فعلی نیز عیناً به تکرار بعدی انتقال می‌یابد، الگوریتم از شانس بالایی برای گریز از نقاط بهینه محلی و یافتن جواب بهینه سراسری برخوردار است. در الگوریتم EM نیز همانند سایر الگوریتم‌های بهینه‌سازی مبتنی بر هوش دسته‌جمعی ابتدا تعدادی ذره به‌طور تصادفی در دامنه مسأله پخش می‌شوند. سپس موقعیت این ذرات با الهام گیری از قانونی شبیه به قانون نیروی کولمب به‌طور مرتب به‌گونه‌ای تغییر داده می‌شود که گرایش کلی آن‌ها به‌سوی نقاط بهینه‌تر باشد. برای این منظور در الگوریتم EM به هر یک از ذرات یک‌بار مجازی نسبت داده می‌شود به‌طوری‌که میزان بار نسبت داده‌شده به هر ذره متناسب با میزان بهینگی نقطه‌ای است که آن ذره در آن واقع‌شده است. همانند قوانین حاکم بر طبیعت در الگوریتم مذکور نیز میزان بار هر ذره تعیین‌کننده میزان جاذبه یا دافعه سایر ذرات نسبت به آن است. به عبارت دقیق‌تر، ذراتی که منجر به مقادیر کمتری برای تابع هزینه شوند بار بیشتری داشته و درنتیجه سایر ذرات را باقدرت بیشتری به‌سوی خود جذب می‌کنند؛ درحالی‌که ذراتی با مقادیر بزرگ تابع هزینه، سایر ذرات را از خود دفع می‌نمایند (یا کمتر جذب می‌کنند). در الگوریتم EM پس از اختصاص دادن یک‌بار مجازی به هر یک از ذرات مصنوعی، نیروی کل وارده بر هر ذره از طرف سایر ذرات را با استفاده از رابطهی شبیه به قانون نیروی کولمب محاسبه می‌کنیم. سپس هر ذره را با یک طول گام معین در جهت نیروی وارده بر آن حرکت می‌دهیم و مراحل فوق را تا زمانی که شرط لازم برای خاتمه اجرای الگوریتم فراهم شود تکرار می‌کنیم. البته در این الگوریتم به‌منظور جستجوی مؤثرتر دامنه مسأله، پس از هر بار جابجایی ذرات یک جستجوی محلی نیز در همسایگی هر یک از جواب‌ها انجام می‌شود که در ادامه بیشتر در این مورد صحبت خواهیم کرد. با توجه به توضیحات فوق، الگوریتم EM استاندارد از چهار مرحله اصلی تشکیل‌شده است که عبارت‌اند از:

- مقداردهی اولیه به ذرات (به‌صورت تصادفی)
- محاسبه نیروی وارده بر هر یک از ذرات
- حرکت دادن ذرات در جهت نیروی وارده بر آن‌ها

➢ جستجوی محلی

در ادامه‌ی بحث ابتدا فرم کلی الگوریتم EM را معرفی کرده و سپس با جزئیات مربوط به پیاده‌سازی هر یک از مراحل چهارگانه فوق آشنا می‌شویم. فرض کنید می‌خواهیم مسأله کمینه‌سازی(Min) رابطه (۱)

$$\begin{aligned} & f(x) \\ & s..t. \quad x \in \Omega \end{aligned} \qquad \Omega = (X \in R^n \mid \ -\infty < l_k \le x_k \le u_k < \infty, \quad k = 1,..., n)$$

(۱)

را که در آن $f : R^n \to R$ یک تابع غیرخطی است، با استفاده از الگوریتم EM حل کنیم. برای این منظور ابتدا باید مقادیر مناسبی را به پارامترهای M، LSIter، Maxlter و δ نسبت دهیم که به‌ترتیب:

M: تعداد ذرات مورداستفاده در الگوریتم EM

Maxlter: حداکثر تعداد تکرارها به‌منظور خاتمه اجرای الگوریتم

LSIter: حداکثر تعداد مراحل جستجوی محلی

δ: پارامتر جستجوی محلی ($\delta \in [0,1]$) است. قالب کلی الگوریتم EM در حالت استاندارد به‌صورت زیر است، شکل (۱):

```
General procedure of Em (M , MaxIter .lSIter , δ)
1 : initialize ()
2 : iteration ← 1
3 : while stopping criterion is met do
4 : local (lSIter , δ)
5 : F ← CalcF ()
6 : move (F)
7 : end while
```

شکل (۱): شبه‌کد روند کلی الگوریتم EM

در خط شماره‌ی ۱ از الگوریتم فوق، بردارهای جواب با استفاده از تابع Initialize به‌طور تصادفی در دامنه مسأله پخش می‌شوند. در خطوط ۳ تا ۸ مراحل سه‌گانه جستجوی محلی (با استفاده از تابع Local)، محاسبه بردار نیروی کلی وارده بر هر یک از ذرات (با استفاده از تابع CalcF) و جابجایی ذرات در جهت نیروی وارده بر آن‌ها (با استفاده از تابع(Move، به‌طورمتوالی و به تعداد دفعات از پیش تعیین‌شده‌ای انجام می‌شوند. همان‌طور که می‌بینیم تابع Local برای انجام یک جستجوی محلی در همسایگان هر یک از جواب‌ها از پارامترهای LSIter و δ بهره می‌گیرد که در ادامه بحث به‌طور دقیق‌تر با نحوه انجام این کار آشنا خواهیم شد.

تولید بردارهای تصادفی اولیه

در الگوریتم ارائه‌شده در قسمت قبل در مرحله مقداردهی اولیه، تابع Initialize، M بردار تصادفی N بعدی به نام‌های x^iرا در دامنه مسأله‌ایجاد می‌کند. برای این منظور، این تابع مؤلفه kام از بردار iام، یعنی x_k^i $(k = 1,..., N \ ; \ i = 1,..., M)$ را برابر با یک عدد تصادفی با توزیع یکنواخت در بازه مطلوب در نظر می‌گیرد، یعنی قرار می‌دهد $x_k^i = l_k + \lambda (u_k - l_k)$ که در آن $\lambda \approx u(0,1)$. سپس این تابع مقدار تابع هزینه را در هر یک از این نقاط محاسبه

کرده و موقعیت بهترین بردار، x^{best} را که منجر به کمترین مقدار برای تابع هزینه می‌شود تعیین می‌کند. شکل (۲) نحوه پیاده‌سازی تابع Initialize را نشان می‌دهد. درک نحوه عملکرد این الگوریتم بسیار ساده بوده و نیاز به هیچ‌گونه توضیح اضافی ندارد.

```
Initialize ()
1: for i = 1 to M do
2: for k = 1 to N do
3: λ ← u(0,1)
4: x_k^i ← l_k + λ(u_k − l_k)
5: end for
6: Calculatef(x^i)
7: end for
8: x^best ← arg min{ f(x^i), ∀_i }
```

شکل (۲): تولید بردارهای اولیه و ارزیابی آن‌ها

جستجوی محلی

پس از توزیع بردارها به‌طور تصادفی در دامنه مسأله، تابع Local در شبه کد شکل (۳) با اعمال تغییرات تصادفی به هر یک از مؤلفه‌های بردار x^i (به‌طور تک‌به‌تک) به انجام یک جستجوی محلی در همسایگان هر یک از جواب‌ها می‌پردازد. در این جستجوی محلی پارامترهای δ ,lSIter به ترتیب شعاع و تعداد مراحل جستجوی محلی را مشخص می‌کنند. برای انجام جستجوی محلی در همسایگان هر یک از جواب‌ها ابتدا با استفاده از مقدار تعیین‌شده برای δ، حداکثر مقدار مجاز برای تغییرات هر یک از متغیرها را به‌صورت $\delta \times (\max_k (u_k - l_k)$ تعریف می‌کنیم. استفاده از این مقدار بدین معناست که هر یک از مؤلفه‌های x^i می‌تواند حداکثر به‌اندازه $\delta \times (\max_k (u_k - l_k))$ تغییر نماید. بدین ترتیب با انتخاب یک مقدار مناسب برای پارامتر δ می‌توان امیدوار بود که پس از اعمال یک تغییر تصادفی به هر یک از مؤلفه‌های بردار x^i، بردار حاصله همچنان در دامنه تعریف مسأله باقی بماند. در قدم بعد، بردار x^i را موقتاً در متغیر y ذخیره‌سازی می‌کنیم و سپس به‌نوبت، هر بار یکی از مؤلفه‌های y را به‌طور تصادفی با استفاده از اندازه گام به‌دست‌آمده تغییر می‌دهیم. این تغییرات تصادفی حداکثر به تعداد دفعات LSIter بار به هر یک از مؤلفه‌های بردار y اعمال می‌شوند. چنانچه اعمال هر یک از این تغییرات تصادفی متوالی به هر یک از مؤلفه‌های بردار y منجر به مقداری کوچک‌تر از $f(x^i)$ برای $f(y)$ گردد (یعنی $f(y) < f(x^i)$)، در آن صورت بردار y را جایگزین بردار x^i کرده و جستجوی محلی را در همسایگی بردار جواب بعدی یعنی x^{i+1} یا در همسایگی یکی دیگر از مؤلفه‌های بردار x^i انجام می‌دهیم پس ازآنجام جستجوی محلی در همسایگان تمام بردارهای جواب x^{best} تعیین می‌شود. شکل (۴-۵) جزئیات مربوط به نحوه انجام جستجوی محلی در همسایگان جواب‌ها را نشان می‌دهد. در خطوط ۳و۴ دو حلقه تودرتو، یکی برای جستجو در اطراف هر یک از بردارها و دیگری برای تغییر هر یک از مؤلفه‌های بردار موردنظر به‌طور جداگانه، مورداستفاده قرارگرفته‌اند. حلقه ایجادشده در خط ۶ به‌گونه‌ای طراحی‌شده که پس از LSIter بار انجام جستجوی محلی و یا یافتن جوابی بهتر از جواب فعلی اجرای آن متوقف شود. لازم به توضیح است که در خط ۸ مقدار به‌دست‌آمده برای متغیر تصادفی λ_2 همواره مثبت است. ولی برای آنکه امکان افزایش و کاهش تصادفی متغیر y_k به‌طورتوأم

وجود داشته باشد، در خطوط ۹ تا ۱۳ تصمیم‌گیری مناسب برای این منظور با استفاده از متغیر λ_1 انجام‌شده است. همچنین در پایان اجرای الگوریتم بهترین جواب به‌دست‌آمده از جستجوی محلی دربردار x^{best} ذخیره‌سازی است. توجه داشته باشید که در الگوریتم فوق جستجوی محلی بدون نیاز به گرادیان تابع هزینه انجام می‌شود. هرچند که برای انجام جستجوی محلی می‌توان از الگوریتم‌های کارآمدتر دیگری نیز استفاده کرد ولی همین الگوریتم بسیار ساده نیز در بیشتر مواقع منجر به انجام یک جستجوی مؤثر در همسایگی جواب‌ها می‌گردد.

```
local(LSIter, δ)
1: counter ← 1
2: length ← δ(max_k(u_k − l_k))
3: for i = l to M do
4:   for k = 1 to N do
5:   λ_1 ← u(0,1)
6:   while counter < lsiter do
7: y ← X^i
8: λ_2 ← u(0,1)
9:   if λ_1 > 0.5 then
10:   γ_k ← γ_k + λ_2
11:     else
12:     γ_k ← γ_k − λ_2
13:   end if
14: if f(y) < f(x^i) then
15:   x^i ← y
16:   counter ← lstie − 1
17:   end if
18: count ← counter + 1
19:   end while
20:   end for
21: end for
22: x^besr ← arg min(f(x^i), ∀_i)
```

شکل (۳): شبه‌کد مراحل جستجوی محلی در الگوریتم EM

در اینجا به این نکته نیز اشاره می‌کنیم که چون طول گام جستجوی محلی برابر با $\delta \times (\max_k (u_k - l_k))$ است لذا از انتخاب مقادیر بزرگ برای پارامتر δ (نظیر $\delta = 0.5$) باید پرهیز کرد زیرا این کار مانع ازآنجام یک جستجوی مؤثر در

همسایگی جواب‌ها می‌گردد و خطر انتخاب نقاطی بیرون از دامنه مسأله را نیز به‌طور محسوسی افزایش می‌دهد. به‌عنوان‌یک قاعده سرانگشتی مقدار پارامتر δ باید به‌اندازه کافی کوچک‌تر از $\min_k (u_k - l_k)/m$ انتخاب گردد.

محاسبه بردار نیروی کلی

با توجه به قانون کولمب، نیروی وارده به هر یک از دو ذره باردار در یک سیستم الکترواستاتیکی متناسب با حاصل‌ضرب بار آن دو و نیز متناسب با عکس مجذور فاصله بین دو ذره است. همان‌طور که پیش‌تر نیز به آن اشاره کردیم، در الگوریتم EM، به هر یک از ذرات یک‌بار مجازی نسبت داده می‌شود با این تفاوت که برخلاف سیستم‌های رایج الکترواستاتیکی در اینجا بار مجازی نسبت داده‌شده به ذرات در طی اجرای برنامه تغییر می‌کند. پس از نسبت دادن یک‌بار مجازی به هر ذره، نیروی کل وارده بر آن را با استفاده از قانونی شبیه به قانونی نیروی کولمب محاسبه می‌کنیم. تابع CalcF در شبه کد شکل (۴) برای محاسبه نیروی وارده بر یک‌ذره مجازی از طرف سایر ذرات گروه مورداستفاده قرارگرفته است. برای انجام این کار در تابع CalcF ابتدا به ذره i ام بار مجازی q^i را نسبت می‌دهیم. از این بار مجازی در ادامه برای محاسبه‌شدت جاذبه یا دافعه ذره باردار واقع در نقطه x^i استفاده خواهیم کرد. مقدار q^i با توجه به میزان بهینگی بردار x^i با استفاده از رابطه (۲) زیر محاسبه می‌شود:

$$q^i = \exp\left(-n \frac{f(x^i) - f(x^{best})}{\sum_{k=1}^{m} f(x^k) - f(x^{best})}\right) \qquad i=1, 2...M \qquad (۲)$$

```
Calcf ()
1 : for i = 1 to M do
2 : q^i ← exp( −n (f(x^i) − f(x^best)) / (Σ_{k=1}^{m} f(x^k) − f(x^best)) )
3 : F^i ← 0
4 : enf for
5 : for i = 1 to M do
6 : for j = 1 to M do
7 : if f(X^j) < f(X^i) then
8 : F^i ← Fi + (X^j − X^i) q^i q^j / ||x^j − x^i||^2
9 : else
10 : F^i ← Fi + (X^j − X^i) q^i q^j / ||x^j − x^i||^2
11 : end if
12 : end for
13 : endfor
```

شکل (۴): شبه‌کد محاسبه بردار نیروی کلی وارد به هر ذره

در رابطه (۲) دلیل ضرب کردن کسر درون پرانتز در ضریب n این است که در مسائلی با تعداد متغیرهای زیاد معمولاً مقادیر نسبتاً بزرگی به پارامتر M نسبت داده می‌شود که همین امر ممکن است باعث به دست آمدن مقادیر بسیار کوچکی برای کسر درون پرانتز شده و به‌تبع آن مشکلاتی در اجرای صحیح برنامه به وجود آید. رابطه (۲) به‌گونه‌ای تعریف‌شده که ذرات واقع در نقاط بهتر بار بیشتری به دست آورند. بدیهی است که در حالت کلی به‌جای استفاده از این رابطه می‌توان از هر رابطه دیگری که در آن به نقاط بهتر بار بیشتری نسبت داده شود نیز استفاده کرد. بااین‌حال نتایج حاصل از شبیه‌سازی‌های مختلف نشان می‌دهند که در اکثر مسائل استفاده از رابطه (۲) منجر به نتایج رضایت بخشی می‌شود. توجه داشته باشید که در الگوریتم EM بار مجازی نسبت داده‌شده به‌تمامی ذرات همواره مثبت است و نیروی وارده بر هر ذره تنها با استفاده از مقادیر به‌دست‌آمده برای تابع هزینه توسط سایر ذرات محاسبه می‌شود. پس از محاسبه بار مجازی نسبت داده‌شده به هر یک از ذرات با استفاده از رابطه (۳)، کل نیروی وارده بر ذره i ام، F^i را که برابر با مجموع نیروهای وارده از طرف سایر ذرات بر این ذره است، محاسبه می‌کنیم. برای این منظور اگر F_j^i بیانگر نیروی وارده از طرف ذره j ام بر ذره i ام باشد F^i از رابطه زیر محاسبه می‌شود:

$$F^i = \sum_{j \neq i}^{M} F_j^i, i = 1,2,.., M \qquad (۳)$$

که در آن:

$$F_j^i = \begin{cases} (x^j - x^i)\dfrac{q^i q^j}{\left\|x^j - x^i\right\|^2} & f(x^j) < f(x^i) \\ (x^j - x^i)\dfrac{q^i q^j}{\left\|x^j - x^i\right\|^2} & f(x^j) \leq f(x^i) \end{cases} \qquad (٤)$$

(منظور از $\left\|x^j - x^i\right\|$ نرم ۲ بردار $x^j - x^i$ است.) با یک استدلال ساده می‌توان دید که نتیجه کلی استفاده از معادلات فوق این است که همواره ذراتی با میزان بهینگی کمتر به‌سوی ذراتی با میزان بهینگی بیشتر جذب می‌شوند. شکل (۴) جزئیات مربوط به نحوه پیاده‌سازی تابع (CalcF) را نشان می‌دهد. در شبه کد فوق خطوط ۷ و ۸ برای محاسبه جاذبه و خطوط ۹ و۱۰ برای محاسبه دافعه بین دو ذره به‌کاررفته‌اند. توجه داشته باشید که x^{best} به‌عنوان یک نقطه جاذب برای تمام ذرات دیگر عمل می‌کند. با کمی دقت در الگوریتم فوق می‌توان دید که محاسبه جهت نیروی کلی وارده بر هر ذره، ازنظر مفهومی تا حدی شبیه به محاسبه گرادیان تابع f است. البته تفاوت‌های نیز بین این دو وجود دارد ، مثلاً در خطوط ۸ و ۱۰ از الگوریتم فوق مقدار به‌دست‌آمده برای F^i متأثر از فاصله بین دو ذره است که همین امر باعث می‌شود تا در صورت نزدیکی بیش‌ازحد دو ذره به یکدیگر، آن دو در جهتی کاملاً متفاوت از گرادیان تابع f در آن نقطه بر یکدیگر نیرو وارد کنند. پس از تعیین بردار نیروی کلی وارده بر هر یک از ذرات نوبت به جابجایی آن‌ها در دامنه مسأله می‌رسد. برای این منظور هر ذره را در جهت نیروی کل وارده بر آن با گامی با طول تصادفی جابه‌جا می‌کنیم. طول این گام تصادفی که در ادامه آن را با λ نشان می‌دهیم، برابر با یک متغیر تصادفی با توزیع یکنواخت در بازه صفر و یک است. البته بدیهی است که برای این منظور می‌توان از گام‌های تصادفی با توزیع یکنواخت نیز استفاده کرد. با توجه به توضیحات فوق و شکل (۵)، تابع (Move) با استفاده از رابطه زیر موقعیت ذره i ام را تغییر می‌دهد:

$$x^i \leftarrow x^i + \lambda \frac{F^i}{\left|F^i\right|^2}(RNG) \quad i = 1,..,M, \quad i \neq best \tag{۵}$$

در عبارت فوق RNG برداری است که محدوده مجاز تغییرات هر یک از متغیرها را تعریف می‌کند. در واقع استفاده از این بردار تضمین می‌کند که مقدار به‌دست‌آمده برای هر یک از مؤلفه‌های بردار X^i همواره در محدوده مجاز یعنی (u_k, l_k) قرار داشته باشد. توجه داشته باشید که با توجه به رابطه $X_0^i = LB + rand(UB - LB)$، هر ذره در جهت بردار نیروی وارده بر آن حرکت می‌کند. همچنین همان‌طور که در این رابطه می‌بینیم، بهترین ذره یعنی X^{best} از موقعیت خود جابه‌جا نشده و عیناً به تکرار بعدی انتقال می‌یابد. به همین دلیل می‌توان در شبه کد نشان داده‌شده در شکل (۴) بردار کل نیروی وارده بر X^{best} را اصلاً محاسبه نکرد، هرچند که این کار منجر به بهبود چندانی در بار محاسباتی کل الگوریتم نمی‌گردد. شکل (۵) شبه کد مربوط به پیاده‌سازی تابع (Move) را نشان می‌دهد. درک نحوه عملکرد الگوریتم فوق ساده و آسان است و فقط اشاره مختصری به خطوط ۶ تا ۱۱ می‌کنیم. خطوط ۷و۹از الگوریتم فوق نحوه پیاده‌سازی رابطه $V_0^i = \frac{LB + rand(UB-LB)}{\Delta t}$ را در عمل نشان می‌دهند. در عبارت داده‌شده در خط ۷ پارامترهای λ, F_k^i در محدوده بین صفر و یک قرار دارند. بنابراین حاصل عبارت $x_k^i \leftarrow x_k^i + \lambda F_k^i (U_K - x_k^i)$ حداکثر برابر با u_k و حداقل برابر با x_k^j است. به‌طور مشابه می‌توان دید که در عبارت داده‌شده در خط ۹ حاصل عبارت $x_k^i + \lambda F_k^i (u_K - l_K)$ حداکثر برابر با x_k^i و حداقل برابر با x_k^j است. بنابراین خطوط ۶ تا ۱۰ اولاً باعث می‌شوند هر ذره در امتداد نیروی کل وارده بر آن جابه‌جا شود و ثانیاً نحوه انجام این جابجایی به‌گونه‌ای است که ذره در دامنه تعریف مسأله باقی می‌ماند.

```
Move (F)
1 : for i = to M do
2 : if i ≠ best then
3 : λ ← u(0,1)
4 : Fi ← Fi / |Fi|
5 : for k = 1 to N do
6 : if F_k^i > 0 then
7 : x_k^i ← x_k^i + λF_k^i (U_K − x_k^i)
8 : ELSE
9 : x_k^i ← x_k^i + λF_k^i (U_K − L_K)
10 : end if
11 : end for
12 : end if
13 : end for
```

شکل (۵): شبه‌کد تغییر مکان ذرات در دامنه مسأله

یافته ها :

۱- پیدا کردن پاسخ بهینه در فضای جستجو در زمان کمتر نسبت به رویکردهایی با بررسی تمامی داده‌ها

۲- کاهش هزینه محاسباتی برای یافتن پاسخ بهینه در فضای جستجو نسبت به رویکردهایی با بررسی تمامی داده‌ها

۳- افزایش دقت الگوریتم خوشه‌بندی با توجه به داده‌های پیداشده از طریق الگوریتم فرا ابتکاری

برای ساخت دیتاست این تحقیق از *log* فایل های ذخیره شده در پایگاه داده سایت http://masjedsoleyman-samacollege.ir استفاده شده است.این صفحات شامل ۱۰۰۰ صفحه است که ۱۶۲ کاربر در یک دوره زمانی ۲ ماه بر اساس رفتارهایشان به آن‌ها نرخ داده‌اند سپس بر اساس محتوای صفحات به‌وسیله افراد خبره این صفحات به ۵ دسته کاربری هم علاقه دسته‌بندی گردیدند که تمام الگوریتم‌های خوشه‌بندی مقایسه شده شامل ۵ کلاس مختلف برای مقایسه با نتایج قضاوت افراد خبره انتخاب‌شده‌اند.زمان جستجو برای همه آزمایش‌ها یکسان در نظر گرفته می‌شود که برابر با ۴۵۰ ثانیه است.

زمان جستجو برای همه آزمایش‌ها یکسان در نظر گرفته می‌شود که برابر با ۴۵۰ ثانیه است. در ادامه اسامی نه الگوریتم به‌طور خلاصه آمده است:

EM: الگوریتم الکترومغناطیس

HEM: الگوریتم الکترومغناطیس ترکیبی

K-Means: الگوریتم K-Means

شاخص‌ترین و پرکاربردترین الگوریتم خوشه‌بندی مبتنی برافراز الگوریتم *K-Means* است، که در آن، هـر خوشه با میانگین اشیای آن (مرکز خوشه) نمایش داده می‌شود. این الگوریتم هنگامی‌که خوشه‌ها به‌صورت ابرهای فشرده‌ی مجـزا از هـم هـستند به‌خوبی کار می‌کند. این روش برای پایگاه‌های داده بزرگ نسبتاً کارا و ارتقا پذیر است، ولی اغلب به یک بهینـه محلـی منتهـی می‌شود (T. Kanungo et al, 2002).

با توجه به محاسبات انجام‌شده، مقدار پارامترهای بهینه در همه الگوریتم‌ها تعیین می‌گردد که به شرح زیر است:

جدول ۱: پارامترهای بهینه الگوریتم *EM*

سایز جمعیت	نرخ تغییر	ماکزیمم تعداد جستجوی محلی	جستجوی محلی
۴۰	۰/۱	۳۰	ساده

جدول ۲: پارامترهای بهینه الگوریتم *HEM*

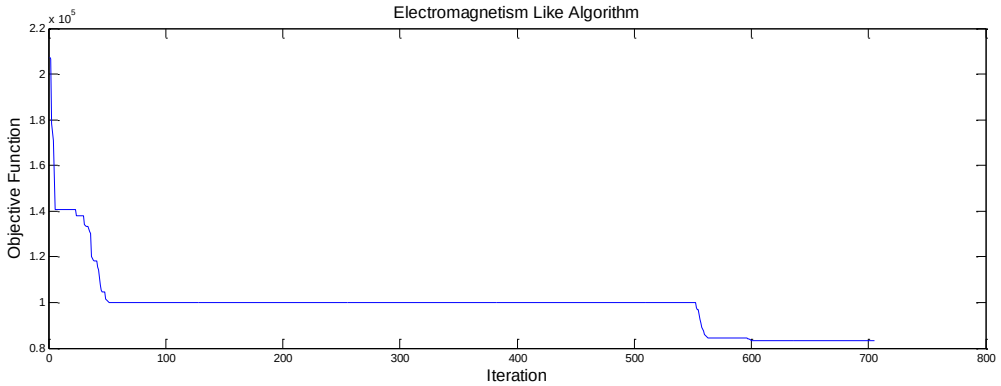
سایز جمعیت	نرخ تغییر	ماکزیمم‌تعداد جستجوی محلی	جستجوی محلی	حداکثر مقدار عدم بهبود
۴۰	۰/۳	۳۰	ساده	۵۰

جدول ۳: مقادیر تابع هدف الگوریتم‌ها برای تابع هدف به ازای پارامترهای بهینه

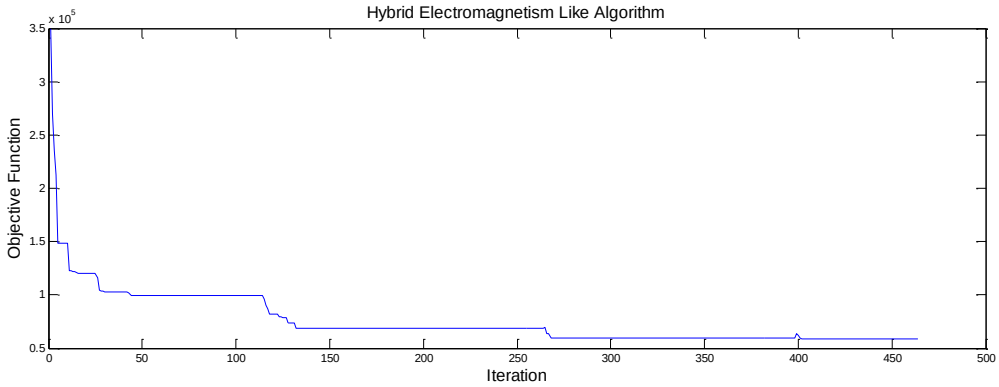
EM	HEM	K-Means	آزمایش
83134. 04	55821. 04	49968. 474	1
78073. 293	54254. 502	54739. 18	2
68662. 989	54989. 078	125785. 519	3
90240. 053	57506. 264	91097. 523	4
63147. 642	57495. 663	49968. 474	5
108276. 071	54174. 057	72200. 224	6
88006. 574	61023. 578	86312. 571	7
96591. 506	55908. 413	49968. 474	8
79962. 679	52724. 977	55122. 124	9
78904. 207	54409. 632	50548. 24	10
83499.905	55830.72	68571.08	میانگین

نمودار همگرایی

در این قسمت در هر تکرار از الگوریتم مطرح شده، بهترین مقدار تابع هدف ارزیابی می‌شود. نمودارهای ۱و۲، نمودار همگرایی درازای یک‌زمان مشخص برای دو الگوریتم پیشنهادی است.



نمودار ۱: نمودار همگرایی برای الگوریتم *EM*



نمودار ۲: نمودار همگرایی برای الگوریتم *HEM*

بحث و نتیجه‌گیری

به همین دلیل همان‌طور که مشخص است هزینه جستجو در فضای رفتارهای کاربران بسیار کاهش پیداکرده است؛ و به همین دلیل دقت خوشه‌بندی با استفاده از تکنیک جستجو تصادفی الگوریتم الکترمغناطیس ترکیبی بسیار کاهش‌یافته است. زمانی که فضای عددی کوچک باشد این بهینگی کمتر مشخص خواهد بود، اما اگر فضای عددی در مسأله‌ای دارای تعداد زیادی باشد و در این فضا مقادیر دارای ترتیب خاصی نباشند، در این صورت میزان کاهش هزینه‌های محاسباتی برای پیدا کردن مقادیر بهینه، برای این فضای عددی بسیار بیشتر و چشم‌گیرتر از قبل خواهد بود. آنچه مسلم است روش‌های خوشه-بندی دارای هزینه‌های کمتری در محاسبات و همین‌طور قابلیت انعطاف خوب در زمینه رویکردهایی بدون دانش اولیه در زمینه‌های گوناگون، همانند دستگاه‌های تشخیص تقلب و تشخیص نفوذ و یا خوشه‌بندی کاربران و یا صفحات و غیره می‌توان نام برد. مهم‌ترین دلیل انتخاب رویکردهای خوشه‌بندی برای مسائل جداسازی، انعطاف بالای این رویکرد است. چراکه آنچه این رویکردها برای جداسازی از آن بهره می‌برند متریک فاصله است؛ که این قابلیت به مسائل گوناگون این امکان را می‌دهد که با استفاده از تبدیل تابع هدف مسأله به فاصله بتواند تقریباً هرگونه مسأله جداسازی را حل نماید. از طرف دیگر دانش اولیه در بسیاری از مسائل دارای اهمیت فراوانی است و به وجود آوردن این دانش برای روش‌های دسته‌بندی یا عملاً غیرممکن است یا اینکه هزینه‌های بسیار زیادی ازجمله هزینه‌های زمانی و مالی و تخصصی دارد. برای همین در روش‌هایی که عملاً ایجاد دانش اولیه غیرممکن است، باعث می‌شود تنها رویکرد قابل انجام، رویکردهای بر مبنای خوشه‌بندی و روش‌های مرتبط با این زمینه باشد؛ اما مسأله‌ی مهم در این رویکردها، گیر افتادن هر بهینه محلی و بالا بودن هزینه محاسباتی برای انتخاب مرکز بهینه هر خوشه است؛ و در ضمن آنچه می‌تواند دقت این رویکردها را به میزان چشمگیری افزایش دهد، استفاده از روش‌های جدیدی و هوشمندانه یا تکاملی برای پیدا کردن مراکز بهینه با هزینه کم محاسباتی و زمانی است. در این تحقیق برای اولین بار ازالکترومغناطیس ترکیبی جهت حل مسأله،خوشه بندی اسناد وب براساس مدل رفتار کاربر استفاده شده است. همچنین در این مطالعه برای نخستین بار جهت اجرای هرچه بهتر الگوریتم‌ها از مکانیسم شروع مجدد استفاده شد. در ادامه

تحقیق مقایسه‌ای جامع بین انواع پارامترها و عملگرهای مربوط به هریک از الگوریتم‌های ارائه‌شده، جهت یافتن و تنظیم بهترین عملکرد برای هر الگوریتم صورت پذیرفت. بعد از یافتن بهترین پارامترها و عملگرها برای اجرای بهتر هر الگوریتم، روش‌های فراابتکاری توسعه داده‌شده با یکدیگر به رقابت پرداختند که نتایج محاسباتی و تحلیل نتایج حاکی از این بود که الگوریتم الکترومغاطیس ترکیبی با مکانیسم شروع مجدد دارای عملکرد بهتری است. اما در یک مقایسه کلی و جامع در بین سه الگوریتم ارائه‌شده، مشاهده می‌شود که الگوریتم الکترومغناطیس ترکیبی با میانگین تابع هدف ۵۵۸۳۰,۷۲ دارای عملکرد خوبی نسبت به سایر الگوریتم های مطرح شده دارد.

منابع

[1] B. Mobasher, R. Cooley and J. Srivastava, Automatic Personalization based on Web Usage Mining, Communications of the ACM, 2000, vol. 43, 142-151.

[2] S. Chakrabarti, B. Dom, P. Indyk, Enhanced hypertext categorization using hyperlinks, SIGMOD 1998, 307-318.

[3] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification And Scene Analysis, John Wiley & Sons, 2000R. Web, Statistical Pattern Recognition, John Wiley & Sons, 2002.

[4] J. Furnkranz. Web mining. The Data Mining and Knowledge Discovery Handbook, pages. Springer, 2010, 899- 920.

[5] Holland, J. H. (1975). "Adaptation in natural and artificial systems: An introductory analysis with applications to biology." control and artificial intelligence. University of Michigan Press.

[6] Birbil, S. I., & Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. Journal of Global Optimization 25, 263–282.

[7] T. Kanungo, Mount D. M., Netanyahu N., Piatko C., Silverman R. and Wu A. Y., 2002. A local search approximation algorithm for K-Means clustering. Computational Geometry: Theory and Applications, SoCG'02, pp. 89-112.

کارگاه‌های آموزشی مرکز اطلاعات علمی

آموزش مهارت های کاربردی در تدوین و چاپ مقاله

اصول تنظیم قراردادها

مقاله نویسی علوم انسانی