ارزیابی یک سیستم توصیه‌گر با استفاده از الگوریتم تکاملی ترکیبی GAPSO

اعظم سمیعی [1]، کمال میرزائی [2]

[1] دانشجوی کارشناسی ارشد، گروه کامپیوتر نرم‌افزار، دانشگاه آزاد اسلامی، واحد میبد
azamsameiy@yahoo.com

[2] عضو هیئت علمی، گروه کامپیوتر نرم‌افزار، دانشگاه آزاد اسلامی، واحد میبد
mirzaie_kamal@yahoo.com

چکیده

در سال‌های اخیر شاهد انفجار اطلاعات با رشد نمائی در اینترنت هستیم. اینترنت با سربار اطلاعاتی بسیار زیادی مواجه است و مشکل مواجه کاربران با انبوهی از اطلاعات وجود دارد. این‌طور به نظر می‌رسد که باید دنبال راهکاری بود که کاربران را در، یافتن آیتم‌های موردعلاقه‌شان یاری کند. سیستم‌های توصیه‌گر با تحلیل رفتار کاربران و جمع‌آوری اطلاعات آنان، توصیه‌های مفیدی متناسب با نیاز کاربران به آن‌ها پیشنهاد می‌دهند. یکی از الگوریتم‌های به کار رفته در این سیستم‌ها الگوریتم پالایش مشارکتی مبتنی بر کاربر است. مهم‌ترین بخش الگوریتم پالایش مشارکتی تعیین شباهت بین کاربران است.

در این مقاله از یک معیار شباهتی استفاده می‌شود که وزن و نسبت رتبه دهی را برای تعیین شباهت در نظر می‌گیرد و به‌منظور یافتن بردار وزن مناسب، الگوریتم‌های تکاملی ترکیبی ازدحام ذرات و ژنتیک پیشنهاد داده و پیاده‌سازی می‌شود همچنین الگوریتم‌های مذکور به‌صورت منفرد نیز پیاده‌سازی شده و مورد ارزیابی قرار می‌گیرد. نتایج به‌دست‌آمده نشان می‌دهد که روش پیشنهادی دقت پیشنهاد را بهبود می‌بخشد و باعث افزایش کیفیت پیش‌بینی و کارایی پیشنهاد می‌شود.

کلمات کلیدی

سیستم‌های توصیه‌گر، پالایش مشارکتی، معیار شباهت، الگوریتم ازدحام ذرات، الگوریتم ژنتیک.

۱- مقدمه

در عصر جدید اگرچه در اینترنت با حجم زیادی از اطلاعات روبرو هستیم ولی این مسئله فرایند تصمیم‌گیری و انتخاب اطلاعات را برای کاربران وب دشوار ساخته است. این موضوع انگیزه‌ای شد تا پژوهشگران را به یافتن راه‌حلی برای رفع مشکل مذکور وادار نماید. یکی از این رویکردها استفاده از سیستم‌های توصیه‌گر است. این سیستم‌ها در دهه‌ی ۹۰ به‌عنوان یک شاخه‌ی مستقل پا به عرصه‌ی تحقیق و پژوهش گذاردند. این سیستم‌ها تلاش می‌کنند با حذف اطلاعات اضافه یا نامطلوب، آن دسته از آیتم‌های اطلاعاتی را پیشنهاد دهند که موردعلاقه‌ی کاربر باشد[2]. تنها کارایی یک سیستم توصیه‌گر، جستجوی آیتم‌ها در زمان کمتر نیست بلکه هدف اصلی آن کشف آیتم‌ها است؛ یعنی پیشنهاد آیتمی فوق‌العاده جالب که مورد توجه شماست و در گوشه‌ای از بستر وب وجود دارد در حالیکه شما از وجود آن کوچک‌ترین اطلاعی ندارید. هسته یک سیستم توصیه‌گر، الگوریتم پالایش آن است؛ پالایش مبتنی بر کاربر و پالایش مبتنی بر محتوا. اولی بر روی این فرض استقرار یافته که افراد با ویژگی‌های شخصی مشترک (مثلاً سن، جنس، کشور، غیره) علایق مشترک خواهند داشت؛ در حالیکه پالایش مبتنی بر محتوا آیتم‌هایی مشابه به آنچه در گذشته ترجیح داده را توصیه می‌کند. تکنیک پالایش مشارکتی به‌منظور ساخت یک توصیه برای کاربر هدف، بر اساس امتیازاتی که مجموعه‌ای از کاربران (همسایگان) که امتیازدهی آن‌ها، بیشترین شباهت را با امتیازات کاربر هدف دارد، اقدام به تولید پیشنهاد می‌کند.

Archive of SID

www.SID.ir

این الگوریتم در حالت کلی، همه شباهت‌ها یا ارتباطات موجود میان کاربران را محاسبه می‌کند[3]. درنتیجه هسته‌ی اصلی این الگوریتم تعیین شباهت است.

۲- کارهای مرتبط

در سال‌های اخیر معیارهای شباهت متفاوتی در سیستم‌های توصیه‌گر استفاده‌شده است نظیر ضریب همبستگی پیرسون، این معیار از امتیازات کاربران به‌طور مستقیم استفاده می‌کند. اسپیرمن، که ترتیب رتبه‌بندی‌ها را در نظر می‌گیرد. تفاضل مربعات میانگین در مقایسه با ضریب همبستگی پیرسون محدود می‌باشد، زیرا همبستگی‌های منفی بین ترجیحات کاربر یا درک آیتم‌های مختلف را در نظر نمی‌گیرد؛ در حالی که وجود چنین همبستگی‌های منفی ممکن است باعث بهبود دقت پیش‌بینی رتبه‌بندی شود. معیار شباهت جاکارد برای استفاده از دیتاست‌هایی مناسب می‌باشد که بسیار پراکنده است. ضریب همبستگی پیرسون به‌عنوان یک معیار شباهت، بهتر از سایر دیدگاه‌های سنتی عمل می‌کند[4]. یکی از چالش‌های الگوریتم پالایش مشارکتی که از همبستگی پیرسون برای محاسبه شباهت بین دو کاربر استفاده می‌کند این است که ارزش تمامی امتیازهای کاربران به آیتم‌ها، یکسان فرض می‌شود این در صورتی است که در واقعیت امتیازهای کاربران روی آیتم‌های مختلف ارزش‌های متفاوت دارد[5]. برای رفع این مشکل می‌توان از معیار شباهتی استفاده نمود که وزنی را برای امتیازات کاربران در نظر می‌گیرد.

۳- معرفی معیار شباهت

انتخاب صحیح یک تابع شباهت برای تعیین شباهت میان کاربران یک فاکتور مهم در الگوریتم‌های پالایش مشارکتی است زیرا دقت توصیه را تحت تأثیر قرار می‌دهد.

تابع شباهت پیشنهادی، برای محاسبه شباهت میان هر دو کاربر a, u در پایگاه داده، به‌صورت رابطه (۱) تعریف شده‌است که با استناد به مقاله‌ی[6] استخراج شده است:

$$sim_w(a,u) = \frac{1}{M-m+1}\sum_{i=0}^{M-m} W^{(i)} V_{a,u}^{(i)} \qquad (۱)$$

در رابطه (۱)، M، بالاترین رتبه و m پایین‌ترین رتبه تعریف‌شده در سیستم است، که کاربر می‌تواند به هر یک از آیتم‌ها بدهد. $V_{a,u}$ بردار مقادیر برای هر زوج کاربر a,u است و W بردار وزن و هم‌اندازه با بردار V است که در ادامه روش محاسبه هر دو بردار تشریح می‌شود.

۳-۱- محاسبه بردار مقادیر(V)

به‌منظور مقایسه دو بردار r_a, r_u، که درواقع رتبه‌های داده‌شده به آیتم‌ها توسط دو کاربر a,u هستند، بردار $V_{a,u}$ به‌صورت $\mathrm{V_{a,u}} = \left(v_{a,u}^{(0)}, \dots, v_{a,u}^{(M-m)}\right)$ تعریف می‌شود که تعداد عناصر آن برابر حداکثر رتبه‌ای که یک کاربر می‌تواند روی یک آیتم ایجاد کند.

هر عنصر $v_{a,u}^{(i)}$ از بردار $\mathrm{V_{a,u}}$ به‌صورت کسر $\frac{A}{B}$ تعریف می‌شود که در آن B بیان‌گر تعداد آیتم‌های رتبه‌بندی شده توسط هر دو کاربر(یعنی $r_a^{(i)} \neq 0, r_u^{(i)} \neq 0$) و A بیان‌گر تعداد آیتم‌های رتبه‌بندی شده توسط هر دو کاربر است با این شرط که تفاضل مطلق رتبه‌بندی‌های کاربر a و کاربر u برابر i باشد$\left(|r_a^j - r_u^j| = i\right)$.

۳-۲- بردار وزن

بر طبق این تابع شباهت به ازای هر بردار V یک بردار W با همان اندازه باید تعریف شود که هر عنصر $w^{(i)}$، نشان‌دهنده اهمیت عنصر $v_{a,u}^{(i)}$ در محاسبه شباهت میان دو کاربر است.

ویژگی هر عنصر بردار $W=(w^{(0)},\dots,w^{(M-m)})$ که عناصر آن در بازه $[-1,1]$ قرار دارند می‌توان با در نظر گرفتن یک بردار نمونه $W=(1,0.5,0,-0.5,-1)$ به‌صورت زیر تشریح کرد:

- $w(0)= 1$، آیتم‌هایی که مشابه هم رتبه دهی شدند را در محاسبه شباهت بسیار مثبت ارزیابی می‌کند.
- $w(1)= 0.5$، آیتم‌هایی را که اختلاف بین رتبه‌های آن‌ها ۱ است را با یک مقدار متوسط، مثبت ارزیابی می‌کند.
- $w(2)= 0$، آیتم‌هایی را که اختلاف رتبه دهی دو کاربر به آن‌ها ۲ است را در محاسبه شباهت در نظر نمی‌گیرد.
- $w(3)= -0.5$، آیتم‌هایی را که اختلاف رتبه آن‌ها ۳ است را با یک مقدار متوسط منفی ارزیابی می‌کند.
- $w(4)= -1$، آیتم‌هایی که متضاد هم رتبه‌دهی شدند را در محاسبه شباهت بسیار منفی ارزیابی می‌کند.

به‌این‌ترتیب، می‌توان به‌صورت منطقی انتظار داشت که تابع شباهت، یک مقدار مثبت بالا $w^{(0)}$ و یک مقدار منفی بالا از $w^{(M-m)}$ داشته باشد.

موضوع اصلی، تعیین مناسب‌ترین تابع شباهت ارائه‌شده توسط یکی از بردارهای W است که در این روش بستگی به ذات خاص داده‌ها در سیستم توصیه‌گر دارد. از میان کلیه‌ی بردارهای W تولیدشده برداری انتخاب می‌شود که تابع هدف یعنی میانگین خطای مطلق (MAE) را کمینه کند؛ بنابراین برای یافتن بهینه‌ترین بردار W که مناسب‌ترین تابع شباهت را ارائه می‌دهد، از الگوریتم تکاملی ترکیبی GAPSO استفاده می‌شود.

٤- الگوریتم ترکیبی GAPSO

یکی از ویژگی‌های الگوریتم‌های تکاملی این است که برای حل مسئله نیاز به یک جمعیت اولیه دارد.

٤-۱- جمعیت اولیه

جمعیت اولیه همان بردارهای وزن است که به‌صورت تصادفی تولید می‌شود. با توجه به اینکه $W \in [-1,1]$ است طول بازه به‌صورت فرمول (۲) به دست می‌آید[6]:

$$\text{بازه طول} = \frac{1-(-1)}{M-m+1} \qquad (۲)$$

بنابراین طول هر بازه با در نظر گرفتن M=5 و m=1 برابر ۰/۴ محاسبه‌شده و محدوده‌های مناسب برای قرار گرفتن جمعیت اولیه به‌صورت زیر تعیین می‌شود:

$w0 \in [1,0.6], w1 \in [0.6,0.2], w2 \in [0.2,-0.2], w3 \in [-0.2,-0.6], w4\ [-0.6,-1]$

بهینه‌ترین بردار W، برداری خواهد بود که مناسب‌ترین وزن‌ها را برای هر یک از عناصر بردار V در نظر بگیرد تا با محاسبه دقیق‌تر میزان شباهت میان کاربران بتواند مقدار خطای میان رتبه‌های واقعی داده‌شده توسط کاربران و رتبه‌های پیش‌بینی‌شده توسط سیستم را کاهش داده و درنتیجه دقت و صحت توصیه‌ها را در سیستم توصیه‌گر افزایش دهد؛ بنابراین برای یافتن

Archive of SID

www.SID.ir

بهینه‌ترین بردار W، توسط الگوریتم GAPSO یک تابع برازندگی مناسب برای آن تعریف می‌شود.

٤-٢- تابع برازندگی

در این مسئله، تابع برازندگی بر اساس دقت توصیه و مؤثر بودن آن ارائه می‌شود. معیارهای مختلفی برای دقت توصیه مطرح است که میانگین خطای مطلق (MAE) در نظر گرفته می‌شود زیرا MAE یکی از مهم‌ترین معیارها برای سنجش دقت توصیه است.

الگوریتم GAPSO سعی می‌کند از میان تمام بردارهای W که درواقع جمعیت اولیه‌ی الگوریتم را تشکیل می‌دهند، W ای را پیدا کند که منجر به کم‌ترین میانگین خطای مطلق(MAE) شود. درواقع تابع برازندگی استفاده‌شده در الگوریتم، تابع MAE است که به‌صورت معادله (۳) تعریف می‌شود[6]:

$$\text{fitness} = \text{MAE} = \frac{1}{\#U}\sum_{u\in U}\frac{\sum_{i\in I_u}|p_u^i - r_u^i|}{\#I_u} \qquad (۳)$$

در معادله (۳)، $\#U$ و $\#I_u$ به ترتیب تعداد کاربران آموزشی و تعداد آیتم‌های آموزشی رتبه‌بندی شده به‌وسیله کاربر u است. p_u^i و r_u^i به ترتیب رتبه واقعی کاربر u به آیتم i و رتبه پیش‌بینی‌شده توسط سیستم توصیه‌گر برای آیتم i است.

MAE، میزان خطای رتبه‌های واقعی کاربران با رتبه‌های پیش‌بینی‌شده سیستم است؛ هر چه این مقدار کمتر باشد عملکرد سیستم دقیق‌تر خواهد بود.

الگوریتم GAPSO برای محاسبه MAE، گام‌های زیر را برای هر کاربر هدف (a) نیاز دارد:

۱. بر اساس تابع شباهت (sim_w) تعریف‌شده در معادله (۱)، شباهت میان کاربر هدف (a) و سایر کاربران محاسبه می‌شود. در حقیقت مجموعه همسایه‌های کاربر a، با عنوان k_a (k، تعداد همسایه‌های شبیه‌تر به کاربر هدف است) به دست می‌آیند که با تغییر مقدار k، تعداد k همسایه برای کاربر a، از میان کاربرانی با بالاترین مقدار شباهت، انتخاب می‌شوند.

۲. پیش‌بینی رتبه‌بندی کاربر a روی آیتم i، با عنوان $\text{p}_\text{a}^\text{i}$ است. این پیش‌بینی از طریق رابطه انحراف از میانگین (DFM) که در رابطه (۴) با استفاده از رتبه‌ای که همسایگان کاربر a به آیتم i داده‌اند به دست می‌آید[4].

$$\text{p}_\text{a}^\text{i} = \bar{r}_a + \frac{\sum_{u\in k_a}[sim_w(a,u)\times(r_u^i - \bar{r}_u)]}{\sum_{u\in k_a} sim_w(a,u)} \qquad (۴)$$

در معادله (۴)، $\bar{r}_a$ متوسط رتبه‌های ایجادشده توسط کاربر a است، r_u^i رتبه‌ای است که کاربر u به آیتم i داده است، $\text{sim}_\text{w}(a,u)$ شباهت کاربر هدف با همسایه‌اش می‌باشد.

با محاسبه تابع شباهت sim_w بین هر زوج کاربر و مشخص شدن k تا از شبیه‌ترین همسایه‌ها به کاربر هدف با استفاده از امتیازات همسایه‌ها برای آیتم‌های کاربر هدف، پیش‌بینی انجام می‌شود؛ سپس MAE با استفاده از معادله (۳) محاسبه می‌شود. درنهایت بهینه‌ترین بردار وزن مربوط به کم‌ترین MAE توسط الگوریتم GAPSO محاسبه می‌شود که از آن در فاز تست برای ارزیابی استفاده می‌شود.

٤-٣- معرفی الگوریتم ژنتیک

الگوریتم ژنتیک از قانون گزینش طبیعت مطرح‌شده توسط داروین، برای یافتن جواب بهینه استفاده می‌کند. طبق این قانون تنها گونه‌هایی از یک جمعیت ادامه نسل می‌دهند که بهترین خصوصیات را جهت سازگاری با طبیعت داشته باشند. با توجه به اینکه فرزندان نسل بعد، از ترکیب کروموزوم‌های والدین پدید می‌آیند. درنتیجه خصوصیات والدین را به ارث می‌برند. بنابراین گونه‌های سازگار از جمعیت در نسل‌های بعد تکثیر یافته و مابقی موجودات (ناسازگار) به‌تدریج در طول زمان منقرض‌شده و از بین می‌روند. در این بین، گاهی اوقات جهش‌هایی نیز در کروموزوم‌های برخی از فرزندان روی می‌دهد که ممکن است باعث سازگاری بیشتر آن با طبیعت شود. الگوریتم ژنتیک (GA) با استفاده از عملگرهای ژنتیکی انتخاب، ترکیب و جهش، اقدام به حل مسائل می‌کند[7].

ابتدا GA کار خود را با یک جمعیت اولیه تصادفی از کروموزوم‌ها شروع می‌کند، هر کروموزوم رشته‌ای از ژن‌ها است که شامل یک راه‌حل، برای مسئله‌ای که GA سعی در حل آن دارد می‌باشد. جمعیت ایجادشده توسط یک تابع برازندگی، ارزیابی می‌شود. سپس جمعیت فعلی(مجموعه راه‌حل‌ها) با استفاده از برخی عملگرهای ژنتیک، در تولید نسل بعدی که اغلب راه‌حل‌های بهتری برای مسئله می‌باشند، استفاده می‌شود. ابتدا عملگر انتخاب از بین کروموزوم‌های موجود در جمعیت، دو کروموزوم (والد) را بر اساس شایستگی برای تولید مثل انتخاب می‌کنند؛ سپس عملگر ترکیب اجراشده، فرزندانی به دست می‌آید که خصوصیات هر دو والد را به ارث برده که اغلب جواب‌های بهتری هستند و عملگر جهش با تغییر برخی از ژن‌های کروموزوم‌های فرزندان باعث ایجاد تنوع ژنتیکی در جمعیت شده سپس برازندگی افراد نسل جدید به‌دست‌آمده، مورد ارزیابی قرار می‌گیرد و این مراحل تکرار می‌شود تا شرط خاتمه الگویتم برقرار گردد؛ درنهایت بهترین کروموزوم نسل آخر، جواب مسئله است[۱].

٤-٣-١- نمایش کروموزوم‌ها

در این مسئله هر کروموزوم به‌صورت بردار w با مقادیر اعشاری تصادفی در نظر گرفته می‌شود که شامل پنج ژن است که بازه مقادیر هر ژن نیز تصادفی و متفاوت است.

٤-٣-٢- عملگر انتخاب

انتخابی که در اینجا مورد استفاده قرارگرفته، انتخاب نخبه‌گرا است. با استفاده از این عملگر بهترین عضو هر جمعیت، زنده می‌ماند و در جمعیت بعد حضور می‌یابد. به‌عبارت‌دیگر عضوی که بالاترین تطابق را دارد به‌طور خودکار به جمعیت جدید منتقل می‌شود. این روش ابتدا در سال ۱۹۷۵ توسط (کندی جونز) معرفی شد. اعمال نخبه‌سالاری در الگوریتم ژنتیک، معمولاً باعث بهبود کارایی آن می‌شود.

Archive of SID

www.SID.ir

٤-٣-٣- عملگر ترکیب

از عملگر ترکیب محاسباتی، برای تولید فرزندان استفاده می‌شود. در این روش یک ضریب وزنی بین ۰ و ۱ به‌اندازه طول والدین در نظر گرفته می‌شود و با استفاده از فرمول‌های (۵) و (۶) دو کروموزوم جدید یا دو فرزند به وجود می‌آیند.

$$y_1 = \alpha.* x_1 + (1-\alpha).* x_2 \quad (۵)$$

$$y_2 = \alpha.* x_2 + (1-\alpha).* x_1 \quad (۶)$$

که در اینجا X_1 و X_2 بردارهای مقادیر اعشاری هستند که کروموزوم والدین را نشان می‌دهند؛ α ضریب وزنی و y_1 و y_2 کروموزوم‌های فرزندان، حاصل از ترکیب والدین هستند.

٤-٣-٤- عملگر جهش

عملگر جهش به روش گوسی پیاده‌سازی می‌شود. بدین‌صورت که ابتدا یک کروموزوم به‌صورت تصادفی از جمعیت انتخاب می‌شود و سپس یک یا چند مؤلفه از آن، بر اساس تابع توزیع گوسی با استفاده از فرمول (۷) تغییر داده می‌شود.

$$y_1 = x_1 + r_1 * N(0,1) \quad (۷)$$

X_1 بردار مقادیر وزن است که کروموزوم والد را نشان می‌دهد. r_1 یک عدد تصادفی در بازه ۰ و ۱ می‌باشد. N(0,1) یک عدد تصادفی توزیع‌شده با استفاده از توزیع گوسی است.

٤-٤- معرفی الگوریتم PSO

مشابه GA، الگوریتم PSO نیز با یک جمعیت اولیه از جواب‌ها که به‌طور تصادفی تولیدشده‌اند شروع می‌شود ولی برخلاف الگوریتم GA عملگر ژنتیکی همانند ترکیب و جهش ندارد. در این روش هر جواب مسئله یک‌ذره نامیده می‌شود که دارای موقعیت و سرعت است. موقعیت، در حقیقت نقطه‌ای از فضای جستجو را نشان می‌دهد و سرعت، تعیین‌کننده جهت حرکت ذره است. همچنین ذره، دارای یک حافظه است که به نگهداری بهترین موقعیت قبلی آن کمک می‌کند. ذرات فضای اطراف خود را برای پیدا کردن کمینه یا بیشینه جستجو می‌کنند در طی جستجو هر ذره موقعیت خود را بر اساس تجربه‌های خود و تجربه‌های بهترین ذره اصلاح می‌کند[8].

PSO ابتدا یک جمعیت اولیه تصادفی از ذرات ایجاد می‌کند؛ یعنی موقعیت ذرات با انتخاب تصادفی در فضای جستجو ایجاد می‌شود و سرعت هر ذره صفر است. سپس مقدار تابع هدف برای هر ذره محاسبه می‌شود.

در شروع الگوریتم، موقعیت فعلی هر ذره به‌عنوان بهترین موقعیت آن، در حافظه هر ذره ذخیره می‌شود. در مراحل بعدی، الگوریتم بر اساس مقایسه‌ای که بین مقادیر شایستگی موقعیت فعلی و موقعیت ذخیره‌شده در حافظه ذره انجام می‌دهد، بهترین موقعیت هر ذره را مشخص می‌کند. بهترین موقعیت سراسری با توجه به مقادیر شایستگی مرتب‌شده ذرات، تعیین می‌شود؛ سپس یک مقدار سرعت جدید برای هر ذره با توجه به سرعت فعلی، فاصله از بهترین موقعیت سراسری و بهترین موقعیت محلی(بهترین موقعیت فعلی ذره) تعیین می‌شود به‌عبارتی‌دیگر موقعیت جدید به‌صورت مجموع موقعیت قدیم و سرعت جدید محاسبه می‌شود. سرعت و موقعیت هر ذره با استفاده از فرمول‌های (۸) و (۹) به‌روزرسانی می‌شود.

$$vid_{(new)} = w * vid_{(old)} + c_1 * r_1 \left(p_{id} - x_{id\,(old)}\right) + c_2 * r_2 * \left(p_{gb} - x_{id\,(old)}\right) \quad (۸)$$

$$x_{id\,(new)} = x_{id\,(old)} + \mathrm{vid}_{(\mathrm{new})} \quad (۹)$$

در این روابط W، وزن اینرسی است که برای کنترل تأثیر سرعت قبلی بر روی سرعت فعلی استفاده می‌شود. $\mathrm{vid}_{(\mathrm{old})}$ سرعت قبلی ذره، r_1 و r_2 اعداد تصادفی تولیدشده‌ی یکنواخت در بازه [۰و۱] هستند. C_1 ضریب یادگیری مربوط به اطلاعات شخصی هر ذره است و C_2 ضریب یادگیری مربوط به اطلاعات کل جمعیت است. P_{id} بهترین موقعیت هر ذره نسبت به موقعیت‌های قبلی خودش، p_{gb} بهترین موقعیت سراسری یا بهترین موقعیتی که یک ذره نسبت به همه‌ی ذرات کسب کرده و x_{id} موقعیت هر ذره را در فضای جستجو نشان می‌دهد. بعد از به‌روزرسانی، ممکن است ذره به دلیل فاصله از بهترین موقعیت قبلی و بهترین موقعیت سراسری، سرعت زیادی داشته باشد و از فضای جستجو خارج شود، به همین دلیل برای محدود کردن سرعت هر ذره، یک سرعت بیشینه v_{max} برای کنترل سرعت ذرات استفاده می‌شود و بهترین ذره (ازنظر شایستگی) در نسل آخر جواب مسئله است [۱][9].

٤-٥- روش ترکیب الگوریتم GAPSO

الگوریتم ترکیبی، مزایای هر دو الگوریتم را دارا است. شبه کد الگوریتم ترکیبی به‌صورت زیر می‌باشد:

گام ۱. تولید نسل اولیه ذرات به‌صورت تصادفی

گام ۲. ارزیابی نسل فعلی بر اساس تابع برازندگی و رتبه‌بندی آن‌ها

گام ۳. انتخاب بهترین موقعیت ذره و انتخاب بهترین موقعیت سراسری

گام ۴. توقف در صورت برقراری شرط توقف

گام ۵. درصورت ایستایی تابع برازندگی (مشکل بهینه محلی) اجرای گام ۶ در غیر این صورت اجرای گام ۷

گام ۶. اجرای عملگرهای ژنتیک(جهش و ترکیب) و ایجاد نسل جدید و رفتن به گام ۲

گام ۷. به‌روزرسانی سرعت و موقعیت ذرات و ایجاد ذرات جدید و اجرای گام ۲

خروجی این الگوریتم ترکیبی بهینه‌ترین بردار وزن می‌باشد که جهت ارزیابی در فاز تست استفاده می‌شود.

٥- آزمایش‌ها

٥-١- مجموعه داده

برای انجام آزمایش‌ها از مجموعه داده Movielens که به‌صورت آنلاین از وب‌سایت گروپ‌لنز قابل دریافت می‌باشد، استفاده می‌شود. این مجموعه داده شامل ۱۰۰۰۰۰ امتیاز است که توسط ۹۴۳ کاربر به ۱۶۸۲ فیلم داده است (ماتریس کاربر- آیتم از ۹۴۳ سطر و ۱۶۸۲ ستون تشکیل می‌شود). این امتیازدهی با ساختار 5-Fold به دو قسمت تقسیم می‌شود: ٪۸۰ داده‌ها (۸۰۰۰۰ امتیاز) برای مجموعه آموزش و ٪۲۰ داده‌ها (۲۰۰۰۰) جهت مجموعه تست استفاده می‌شود. رتبه‌بندی‌ها شامل یک مقیاس عددی ۵ نقطه‌ای است که ۱ و ۲ بیانگر رتبه‌بندی‌های منفی، ۴ و ۵ رتبه‌بندی‌های مثبت و ۳ رتبه‌بندی میانه را نشان می‌دهد.

Archive of SID

www.SID.ir

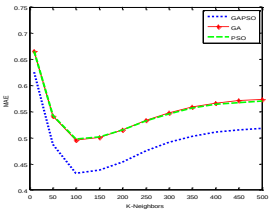
ابتدا برای تعیین تابع شباهت بهینه، فقط بخشی از مجموعه داده؛یعنی کاربران آموزشی استفاده می‌شود بدین منظور ۸۰ درصد از کاربران و آیتم‌ها، به‌عنوان کاربران وآیتم‌های آموزشی در نظر گرفته می‌شوند.

در فاز آموزش با اجرای متعدد الگوریتم GAPSO تابع شباهتی متناظر با بردار W ارائه می‌شود که تابع هدف MAE را کمینه می‌کند. به‌منظور بررسی و مقایسه‌ی نتایج، سیستم توصیه‌گر مبتنی بر الگوریتم‌های ژنتیک و ازدحام ذرات نیز پیاده‌سازی می‌شود تا با استفاده از این الگوریتم‌ها بهترین W که میانگین خطای مطلق کمینه را تولید می‌کند محاسبه شود. در فاز تست از W بهینه فاز آموزش استفاده می‌شود تا معیارهای ارزیابی نظیر میانگین خطای مطلق، فراخوانی و دقت، محاسبه شود.

در فاز تست برای آزمایش W بهینه و استفاده از آن در تابع شباهت، ۲۰ درصد از کاربران را که در الگوریتم ترکیبی پیشنهادی GAPSO استفاده‌نشده‌اند (کاربران آزمایشی) در نظر گرفته می‌شوند. ابتدا با انتخاب کاربر هدف، نسبت رتبه‌بندی‌ها محاسبه شده و از بردار وزن بهینه به‌دست‌آمده فاز آموزش، شباهت بین زوج کاربران استفاده می‌شود و k همسایه‌ی شبیه‌تر به کاربر هدف مشخص می‌شود. در مرحله‌ی آخر پیش‌بینی رتبه‌بندی کاربر هدف روی آیتم‌ها با توجه به k همسایه‌اش به دست می‌آید که میانگین خطای مطلق با توجه به تمام کاربران فاز تست محاسبه می‌شود.

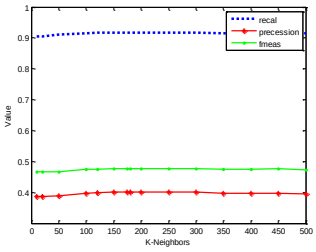
انتخاب درست تعداد همسایه‌ها در فاز تست نیز، بسیار مهم است. ثابت k (تعداد همسایه‌های هر کاربر)، بین ۱۰ تا ۵۰۰ همسایه تغییر می‌کند که به ما کمک می‌کند تا روند تغییرات نمودار مشاهده شود. مقدار k با بررسی مقالات مرجع و اجرای متعدد برنامه به‌دست‌آمده است.

در فاز تست با افزایش تعداد همسایگان از ۱۰ همسایه تا مقدار ۱۲۰ همسایه با هر سه الگوریتم، مقدار خطای حاصل از معیار شباهت بهینه، کاهش یا بهبودیافته است. با افزایش تعداد همسایگان از ۱۲۰ همسایه به بعد، میزان خطا به‌تدریج افزایش می‌یابد. بهترین میانگین خطای مطلق (کمترین خطا) در استفاده از مقدار میانی همسایه به دست می‌آید. شکل (۱) میانگین خطاهای به‌دست‌آمده از الگوریتم GA، GAPSO، PSO با مقادیر متفاوت K نشان می‌دهد.



شکل (۱) میانگین خطا در مقابل تعداد همسایه‌ها

جهت محاسبه معیارهای Recall، precision و Fmeasure باید مقدار مناسب برای k درنظر گرفته شود که با توجه شکل (۲) مقدار k دربازه ۱۸۰ تا ۲۰۰ مناسب است. برای محاسبه معیارهای ارزیابی، k =180 در نظرگرفته می‌شود.



شکل (۲): مقدار معیارهای ارزیابی در مقابل تعداد همسایه‌ها

۵-۲- ارزیابی دقت و فراخوانی

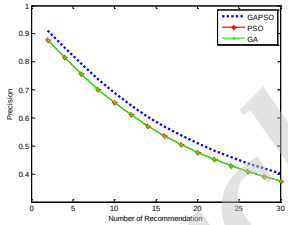
معیار دقت یا Precision بیان می‌کند که از مجموعه پیشنهادها به کاربر، چند درصد درست هستند. معیار فراخوانی یا Recall بیان می‌کند که چند درصد از آیتم‌های موردعلاقه کاربر به او پیشنهادشده است.

پارامترهای مورداستفاده برای محاسبه معیارهای ارزیابی در جدول(۱) آمده است.

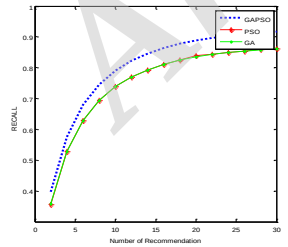
جدول (۱) : پارامترهای عددی آزمایش‌ها

MAE	Precision/Recall	
K-Ranges	N	K
{۱۰,...,۵۰۰}	{۲,...,۳۰}	۱۸۰

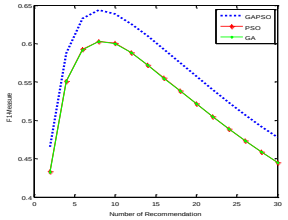
شکل‌های(۳)و (۴) و (۵) به‌ترتیب، دقت و فراخوانی , Fmeasure سیستم‌توصیه‌گر را نشان می‌دهد. این معیارها مقیاس‌های ارزیابی کیفیت توصیه هستند.



شکل (۳): دقت در برابر تغییر تعداد پیشنهادها



شکل (٤): فراخوانی در برابر تغییر تعداد پیشنهادها



شکل (۵): Fmeasure در برابر تغییر تعداد پیشنهادها

Archive of SID

www.SID.ir

۵-۳- نتایج کارایی

نتـایج حاصـل از روش GAPSO نسـبت بـه روش‌هـای ژنتیـک(GA) و بهینه‌سازی ازدحام ذرات (PSO) با هر تعداد توصیه (N)، بهبودیافته است.

در فاز آموزش، الگوریتم GAPSO نسبت به الگوریتم ژنتیک و الگوریتم ازدحام ذرات، فضای راه‌حل را سریع‌تر جستجو کرده و راه‌حـل بهینـه را ارائـه می‌دهد؛ درنتیجـه سـرعت رسـیدن بـه جـواب بهینـه و همگرایـی الگـوریتم GAPSO بسیار مناسب‌تر است زیرا GAPSO با تکرار کمتر بـه میـانگین خطای کمتر، دست‌یافته است.

نتایج الگوریتم GAPSO در فاز تسـت نشـان مـی‌دهـد کـه توصـیه‌هـا و پیش‌بینی‌های حاصل‌شده بهتر از نتایج الگوریتم PSO و GA است.

٦- نتیجه‌گیری و پیشنهاد

سیستم‌های توصیه‌گر سیستم‌های هوشمندی هستند که در فضای اینترنت بـا شناسایی علایق و اولویت‌های هر کاربر، اطلاعات موجود را پـالایش کـرده و پیشنهادهای مناسب و مرتبط را بـه کـاربران ارائـه مـی‌دهنـد. پرکـاربردترین الگوریتم بکار رفته در سیستم‌های توصیه‌گر، الگوریتم پالایش مشارکتی است. انتخاب صحیح معیار شباهت یک فـاکتور مهـم در پـالایش مشـارکتی اسـت زیرا دقت توصیه را تحت تأثیر قرار می‌دهد. در معیارهای شباهت سنتی ارزش همه‌ی امتیازات یکسان در نظر گرفته می‌شود درصورتی‌که در واقعیـت، ارزش آیتم‌ها متفاوت است. در این مقاله تابع شباهتی استفاده مـی‌شـود کـه از یـک بردار وزن و بردار مقادیر استفاده می‌کند که بردار مقادیر بـرای هـر جفـت از کاربران بر اساس اختلاف رتبه آیتم‌ها میان هر دو کاربر، محاسبه می‌شود.

وزن‌های بهینه اعدادی در بازه‌ی [۱و۱-] هستند که نشان‌دهنده اهمیـت هر یک از عناصر بردار مقادیر می‌باشند. برای محاسبه شباهت بهینه میـان دو کاربر از یک الگوریتم تکاملی بـه نـام الگـوریتم ترکیبـی GAPSO اسـتفاده می‌شود. این الگوریتم از بین بردارهای وزنی که تولید می‌شود یک بـردار وزن مناسب به‌دست‌آورده و از آن در محاسبه شباهت استفاده می‌شود.

نتایج به‌دست‌آمده نشان می‌دهد سیستم پیشنهادی دارای صحت و دقت قابل‌قبول در ارائه پیشنهادها به کاربر است. روش ارائه‌شده در این مقالـه ایـن مزیت دارد که به اطلاعات اضافی ارائه‌شده توسط کاربر نیـاز نـدارد و تنهـا از رتبه‌دهی کاربران استفاده می‌کند به همین دلیـل، مـی‌تـوان از آن در همـه‌ی سیستم‌های توصیه‌گر مبتنی بر تکنیک‌های پالایش مشارکتی اسـتفاده نمـود. روش ارائه شده، تنها از رتبه دهی کاربران استفاده می‌کند که حداقل اطلاعات ممکن در هر سیستم توصیه‌گر است.

در آینده می‌توان برای رفع مشکل شروع سـرد (کـاربر جدیـد) از ترکیـب اطلاعات کاربر و محتوا، در الگوریتم‌های پالایش مشارکتی استفاده نمود.

مراجع

[۱] رضایی،راحتی،وحید،امین، **"ترکیب الگـوریتم ژنتیـک و ازدحـام ذرات برای حل مسائل بهینه‌سازی پیوسـته"**، اولـین کنفـرانس ملـی ریاضـیات صنعتی تبریز، تبریز،۱۳۹۳.

[2] Y. S. Kim, B.-J. Yum, J. Song, and S. M. Kim, *"Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites,"* Expert Systems with Applications, vol. 28, pp. 381-393, 2005.

[3] F. Ortega, J. Bobadilla, A. Hernando, and A. Gutiérrez, *"Incorporating group recommendations to recommender systems: Alternatives and performance,"* Information Processing & Management, vol. 49, pp. 895-901, 2013.

[4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "*Recommender systems survey*," Knowledge-Based Systems, vol. 46, pp. 109-132, 2013

[5] K. Choi and Y. Suh, "*A new similarity function for selecting neighbors for each target item in collaborative filtering*," Knowledge-Based Systems, vol. 37, pp. 146-153, 2013

[6] J. Bobadilla, F. Ortega, A. Hernando, and J. Alcalá, *"Improving collaborative filtering recommender system results and performance using genetic algorithms,"* Knowledge-based systems, vol. 24, pp. 1310-1316, 2011.

[7] X. Shi, Y. Lu, C. Zhou, H. Lee, W. Lin, and Y. Liang, "*Hybrid evolutionary algorithms based on PSO and GA,"* in Evolutionary Computation,. CEC'03. The Congress on, pp. 2393-239, 20039. , 2003.

[8] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "*A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments*," in Advanced Information Networking and Applications (AINA), 24th IEEE International Conference on, pp. 400-40, 2010.

[9] W. Wang, Q. Sun, X. Zhao, and F. Yang, "*An improved particle swarm optimization algorithm for QoS-aware web service selection in service oriented communication,"*International Journal of Computational Intelligence Systems, vol. 3, pp. 18-30, 2010.

Archive of SID

www.SID.ir