



بهبود روش خوشه بندی مورچه ای به کمک اتوماتاهای یادگیر

محمد رضا میبدی

آزمایشگاه محاسبات نرم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیر کبیر

تهران ایران

mmeybodi@aut.ac.ir

برنا جعفرپور

آزمایشگاه محاسبات نرم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیر کبیر

تهران ایران

jafarpour@cic.aut.ac.ir

کنند. در این روش برخلاف بسیاری از روش‌های خوشه بندی احتیاج به از پیش تعیین کردن تعداد خوشه‌ها نمی‌باشد و تعداد آنها بطور خودکار در حین فرایند خوشه بندی تعیین می‌گردد. یکی دیگر از ویژگی‌های مهم این روش اینست که خوشه بندی در یک فضای 2 بعدی به جای فضای n بعدی انجام می‌شود که امکان ارزیابی خوشه بندی توسط انسان را فراهم می‌کند. یکی از بهبودهایی که در این روش ایجاد شده است، افزودن حافظه کوتاه مدت¹ به مورچه‌ها برای یادآوری مکانهایی است که قبلاً در آنجا داده ای را قرار داده اند [3] و نشان داده شده است که این تغییر می‌تواند کارایی را در خوشه بندی مورچه ای افزایش دهد. این روش دارای کاربردهای زیادی از جمله خوشه بندی مستندات [4][5]، قسمت بندی² گراف [6][7] و قسمت بندی در VLSI [8] و غیره می‌باشد.

یک اتوماتای یادگیر [13] مانعی است که می‌تواند تعدادی متناظر عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود و نتیجه ارزیابی در قالب سیگنالی تقویتی³ به اتوماتای یادگیر داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی استفاده می‌کند و بدین ترتیب به سمت انتخاب عملی که بیشترین پاداش را از محیط می‌گیرد، میل می‌کند. سادگی و کارایی اتوماتای یادگیر آن را مناسب برای بسیاری کاربردها می‌کند. برای مثال از این ابزار در آموزش شبکه‌ی عصبی [9]، محاسبات تکاملی [10][11]، حل مسئله‌ی کوتاه مدت می‌باشد.

در این مقاله از اتوماتاهای یادگیر برای یادگیری مکانهایی که مورچه‌ها برای اندختن داده‌ها بر روی صفحه باید در نظر بگیرند استفاده شده است. در روش پیشنهادی، صفحه ناحیه بندی می‌شود و هر مورچه

چکیده: خوشه بندی مورچه ای روشی الهام گرفته از طبیعت برای خوشه بندی می‌باشد. این روش دارای مزایایی می‌باشد که آن را در سالهای اخیر مورد توجه محققان قرارداده است. در این مقاله از یک پیشنهاد برای بهبود خوشه بندی مورچه ای ارایه می‌شود. در روش پیشنهادی، از اتوماتای یادگیر برای یادگیری مکان قرار دادن داده‌ها برروی صفحه استفاده شده است. برای این منظور صفحه‌ای که مورچه‌ها بر روی آن خوشه بندی می‌کنند ناحیه بندی می‌شود و سپس هر مورچه مجهز به یک اتوماتای یادگیر است. وضیفه اتوماتای یادگیر هر مورچه یادگیری ناحیه ای می‌باشد که قرار است مورچه داده خود را در ان قرار دهد. این روش با تنها روش موجود برای تشخیص مکانهای مهم در صفحه برای خوشه بندی که "حافظه‌ی کوتاه مدت" نام دارد مقایسه شده است. نتایج آزمایشها بر روی 5 مجموعه داده استاندارد نشان می‌دهد که روش پیشنهادی با محاسبات کمتر، خوشه بندی بهتری در مقایسه با روش "حافظه‌ی کوتاه مدت" بدست می‌آورد. همچنین مقایسه‌ها برتری روش ارائه شده را نسبت به K-means نشان می‌دهد.

کلمات کلیدی: خوشه بندی مورچه ای، اتوماتاهای یادگیر، حافظه‌ی کوتاه مدت

1- مقدمه

روش خوشه بندی مورچه ای که الهام گرفته از رفتار مورچه‌ها در طبیعت می‌باشد اولین بار در [1] برای مجموعه‌ای از ربات‌ها که اشیاء را دسته بندی می‌کردند ارائه شد. این الگوریتم در [2] برای خوشه بندی داده‌ها تصحیح و ارائه شد. در روش خوشه بندی مورچه ای از تعدادی مورچه با عملکرد ساده و بدون کنترل مرکزی استفاده می‌شود. در این روش یک شبکه 2 بعدی از سلول‌ها وجود دارد که داده‌ها و مورچه‌ها برروی سلول‌های آن قرار دارند و مورچه‌ها می‌توانند بر روی آنها حرکت

¹ Short Term Memory

² Partitioning

³ Reinforcement Signal

مقادیر بالا خواهد داشت و در نتیجه داده i با احتمال کمی برداشته می شود (فرمول (1)) و با احتمال بالایی در آن مکان انداخته می شود (فرمول (2)). تفاوت داده i با داده های همسایه اش باعث می شود که (f(i)). مقادیر پایین داشته باشد و در این حالت احتمال اینکه داده توسط مورچه از جای خود برداشته شود زیاد می شود و احتمال گذاشته شدن در آن مکان برای داده پایین می آید.تابع ارائه شده در [1] امکان مقایسه ای داده های عددی را فراهم نمی کند ولی این مشکل در [2] حل و تابع زیر ارائه شد.

$$f(i) = \max(0, \frac{1}{s^2} \sum_{j \in \text{Neigh}(i)} (1 - \frac{d(i, j)}{a})) \quad (3)$$

در محدوده $i \in [0, 1]$ قرار دارد و تفاوت داده های i و z را تعیین می کند. a پارامتر مقیاس⁴ است که در محدوده $i \in [0, 1]$ قرار دارد و باید با توجه به داده ها تعیین شود. s محدوده دید را تعیین می کند به این معنی که محدوده i دید یک مورچه با پارامتر s در هر 4 جهت برابر است با $\frac{s-1}{2}$. $\text{Neigh}(i)$ مجموعه ای داده هایی است که در همسایگی i داده i بر روی صفحه قرار دارند. برای افزایش کارایی الگوریتم "حافظه ای کوتاه مدت" در [2] ارائه شد. در این مدل مورچه ها حافظه ای محدودی دارند که تعدادی از آخرین مکانهایی را که داده ها در آنجا انداخته شده اند در آن ثبت می کنند و هر گاه که قصد انداختن داده ای را داشته باشند به حافظه ای خود رجوع می کنند و به سمت مکانی که شبیه ترین داده به داده ای که حمل می شود در آنجا انداخته شده حرکت می کند. این مدل دارای مشکلاتی نیز می باشد، از جمله اینکه مورچه باید تمام مسیر را طی کند و زمانی که به مکان مورد نظر رسید ممکن است داده ای که قبلا در آنجا قرار داده است توسط مورچه ای دیگری جابجا شده باشد و طی این مسیر بی فایده باشد. برای حل این مشکل در [3] نسخه ای جدیدی از حافظه ای کوتاه مدت ارائه شد. در این مدل هر مورچه برای انداختن داده به تمام نقاطی در که در حافظه ای خود ثبت کرده می رود و با احتمال P_{drop} داده i حمل شده را در مکانی که بیشترین f را داشته باشد قرار می دهد. برای افزایش سرعت، مورچه ها به جای طی مسیر تا نقاط مورد نظر به این نقاط پرش می کنند.

چنانچه مورچه نتواند به کمک حافظه ای خود، داده ای را در صفحه قرار دهد استفاده از حافظه ای خود را تا انداختن موفق بعدی به حالت تعلیق در آورده و به صورت تصادفی عمل می کند. از جمله دیگر تغییراتی که برای بهبود در [3] ارائه شد پارامتر $step_size$ می باشد. در این مدل مورچه ها به جای حرکت از یک سلول به سلول مجاور می توانند به سلولهایی که مجموع فاصله ای عمودی و افقی آنها حد اکثر

دارای یک اتوماتای یادگیر است که تعداد اعمال آن برابر است با تعداد ناحیه ها. اتوماتای یادگیر هر مورچه به صورت احتمالاتی ناحیه ای قرار دادن داده ها برای هر مورچه تعیین می کند. مورچه ها به صورت دست جمعی و بدون داشتن ارتباط مستقیم با یکدیگر، یاد می گیرند که چه ناحیه هایی برای انداختن داده ها مناسب تر می باشد. مقایسه ای روش پیشنهادی با تنها روش موجود برای تعیین مکان انداختن داده ها نشان می دهد که روش پیشنهادی احتیاج به محاسبات کمتری دارد و خوش بندی بهتری تولید می کند. ادامه مقاله بدین صورت سازماندهی شده است. در بخش های 2 و 3 به ترتیب خوش بندی مورچه ای و اتوماتاهای یادگیر به اختصار شرح داده می شود. در بخش 4 روش پیشنهادی و در بخش 5 نتایج آزمایشها ارایه می گردد. بخش 6 نتیجه گیری می باشد.

2- خوش بندی مورچه ای

روش خوش بندی مورچه ای الهام گرفته از رفتار مورچه ها در طبیعت می باشد. مورچه ها در طبیعت می توانند اشیاء مختلف را بدون وجود کنترل مرکزی در دسته های مشابه در کنار یکدیگر جمع آوری کنند. با الهام گرفتن از این مدل، خوش بندی مورچه ای توسعه داده شده است و دارای کاربردهای زیادی در علم می باشد. در این مدل، مورچه ها و داده هایی که باید خوش بندی شوند بر روی صفحه ای دو بعدی قرار داده می شوند. مورچه ها می توانند داده هایی را که به داده های اطراف خود شبیه نیستند از جای خود در صفحه برداشته و در کنار داده های مشابه قرار دهند. مورچه ها برای ارزیابی اینکه داده ای به داده ای اطراف خود شبیه است یا خیر دارای دید محدودی می باشند. این مورچه ها، همانند مورچه های واقعی از کنترل مرکزی بر خوردار نیستند و به صورت مستقل از یکدیگر عمل می کنند. در هر سلول شبکه همzمان دو داده یا دو مورچه نمی توانند قرار بگیرند و چنانچه مورچه ای از یک طرف شبکه بیرون رود از طرف دیگر وارد خواهد شد. مورچه ها با احتمالات زیر داده i را از جای خود برداشته یا در مکانی قرار می دهند [2].

$$P_{pick}(i) = \left(\frac{k_{pick}}{k_{pick} + f(i)} \right) \quad (1)$$

$$P_{drop}(i) = \left(\frac{f(i)}{k_{drop} + f(i)} \right) \quad (2)$$

k_{pick} و k_{drop} پارامترهای الگوریتم می باشند و $f(i)$ تابعی است که شایستگی داده i را برای بودن در مکانی که قرار دارد محاسبه می کند. چنانچه داده i شبیه داده هایی که در همسایگی اش هستند باشد، $f(i)$

⁴ Scaling Parameter

شده و پاسخی به اتماتای یادگیر داده می شود. اتماتای یادگیر از این پاسخ استفاده نموده و حالت درونی خود را به روز می کند و دوباره عمل خود را برای مرحله بعد انتخاب می کند [13]. اتماتا با تعامل با محیط عمل بهینه را فرآگیری می کند و به این شکل پاداش دریافتی خود را از محیط حد اکثر می کند. شکل ارتباط بین اتماتای یادگیر و محیط را نشان می دهد.



شکل ۱: ارتباط بین اتماتای یادگیر و محیط

محیط را می توان توسط یک سه تایی $E \equiv \{a, b, c\}$ نشان داد که در آن $\{a\}$ مجموعه ورودیها، $b \equiv \{b_1, b_2, \dots, b_m\}$ مجموعه احتمالهای خروجیها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالهای جریمه ها می باشد. هر گاه b مجموعه دو عضوی باشد، محیط از نوع P می باشد. در چنین محیطی b_1 به عنوان جریمه و b_2 به عنوان پاداش در نظر گرفته می شود. در محیط از نوع Q ، b می تواند به طور گسته یک مقدار از مقادیر محدود در فاصله $[0,1]$ و در محیط از نوع S متغیر تصادفی در فاصله $[0,1]$ است.

c_i احتمال اینکه عمل a_i نتیجه نامطلوب^۵ داشته باشد، می باشد. در محیط ایستا^۶ مقادیر c_i بدون تغییر می مانند، حال آن که در محیط غیر ایستا^۷ این مقادیر در طی زمان تغییر می کنند. اتماتای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم می گردد [3,13]. اتماتای یادگیر با ساختار متغیر^۸ توسط ۴ تایی $\{a, b, p, T\}$ نشان داده می شود که در آن $a \equiv \{a_1, a_2, \dots, a_r\}$ مجموعه عملهای اتماتا، $b \equiv \{b_1, b_2, \dots, b_m\}$ مجموعه ورودیهای اتماتا و $p = \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از اعمال و $p(n+1) = T[a(n), b(n), p(n)]$ الگوریتم یادگیری می باشد. در این نوع از اتماتاها، اگر عمل a_i در مرحله n انتخاب شود و این عمل، پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می یابند. برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالها افزایش می یابند. این تغییرات به گونه ای صورت می پذیرد که جمع احتمالات برابر با یک باقی بماند. فرمولهای

باشد پرش کند. شبیه کد الگوریتمی را که در این مقاله مورد بررسی قرار گرفته و برگرفته از [3] می باشد که در شکل ۱ آمده است.

در این مدل هر مورچه برای برداشتن داده ها به لیستی که از داده های آزاد (داده هایی که توسط دیگر مورچه ها حمل نمی شوند) وجود دارد مراجعه می کند و یکی از داده ها را به صورت اتفاقی انتخاب می کند و آن را با احتمالی که فرمول (۱) تعیین می کند بر می دارد. این کار تا زمانی ادامه پیدا می کند که مورچه موفق به برداشتن یک داده بشود. پس از آن مورچه در تکرار های بعدی سعی خواهد کرد که داده اش را در مکان مناسبی قرار دهد. در هنگام انداختن داده چنانچه حافظه ای کوتاه مدت مورچه فعال باشد، مورچه به تمام نقاطی که در حافظه اش ثبت شده مراجعه می کند و با احتمالی که فرمول (۲) تعیین می کند داده اش را در مکانی که بیشترین f را تولید می کند قرار می دهد. چنانچه حافظه ای کوتاه مدت مورچه به علت تلاش ناموفق برای انداختن داده غیر فعال باشد، مورچه اقدام به جهش های اتفاقی با حداکثر اندازه $step_size$ می کند و از فرمول (۲) برای تصمیم گیری در مورد انداختن داده ای کرد و سلول مورد نظر توسط داده ای دیگری اشغال شده بود، مورچه داده را به صورت اتفاقی در یکی از سلول های همسایه می اندازد و چنانچه تمام سلوهای همسایه داده در خود داشتند مورچه از انداختن داده صرف نظر می کند.

Algorithm Ant Clustering

```

1: INITIALIZATION PHASE
2: Randomly scatter data items on the toroidal grid
3: for each  $j$  in 1 to  $\#agents$  do
4:    $i :=$  random select(remaining items)
5:   pick up(agent(j), i)
6:    $g :=$  random select(remaining empty grid locations)
7:   place agent(agent(j), g)
8: end for
9: for each  $it\_ctr$  in 1 to  $\#iterations$  do
10:    $j :=$  random select(all agents)
11:   step(agent(j), stepsize)
12:    $i :=$  carried item(agent(j))
13:    $drop :=$  drop item?(f(i)) // see equation 2
14:   if drop = TRUE then
15:     while pick = FALSE do
16:        $i :=$  random select(free data items)
17:       pick := pick item? // see equation 1
18:     end while
19:   end if
20: end for

```

الگوریتم خوش بندی مورچه ای

۳- اتماتاهای یادگیر

یک اتماتای یادگیر یک مدل انتزاعی است که تعداد محدودی عمل را می تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی

⁵Unfavorable

⁶Stationary

⁷Non-Stationary

⁸ Variable Structure Learning Automata

دهنده‌ی قسمتی از ناحیه است که برای انداختن داده در نظر گرفته می‌شود. چنانچه مورچه موفق به انداختن داده در ناحیه‌ی مورد نظر بشود اتوماتای یادگیر پاداش می‌گیرد) $b = 0$ (و در غیر اینصورت جریمه می‌شود) $b = 1$ (برای جلوگیری از همگرایی زودرس اتوماتای یادگیر، عمل یادگیری را بعد از گذشت یک پنجم از تعداد تکرارها شروع می‌کنیم. برای یادگیری از قانون $L_{R\text{-}ep}$ استفاده کرده‌ایم. دلیل این انتخاب اینست که تعداد تلاش‌های موفق برای انداختن داده‌ها معمولاً بسیار کمتر از تعداد تلاش‌های ناموفق است و اتوماتای یادگیر بیشتر جریمه می‌شود تا پاداش بگیرد.

4	8	12	16
3	7	11	15
2	6	10	14
1	5	9	13

شکل 2: تقسیم بندی صفحه به 16 قسمت

بدین ترتیب اتوماتای یادگیر هر مورچه ناحیه‌هایی را که در آنها داده قرار داده است با احتمال بیشتری مورد بررسی قرار می‌دهد. به این شکل یک پرتاپ ناموفق باعث کنار گذشتن اطلاعات موجود نخواهد شد (برخلاف روش حافظه‌ی کوتاه مدت که یک پرتاپ ناموفق باعث غیر فعال شدن حافظه خواهد شد) و تنها احتمال بررسی آن ناحیه کمتر خواهد شد. مزیت دیگر این اینست که ناحیه‌هایی که در آنها داده‌های بیشتری قرار دارند، با احتمال بیشتری بررسی می‌شوند و دیگر نیازی به ارزیابی تعداد زیادی مکان (که در روش حافظه‌ی کوتاه مدت نگه داری می‌شوند) برای انداختن داده در صفحه نیست، در نتیجه تعداد محاسبه f کاهش خواهد یافت. تفاوت دیگر اینست که به جای بررسی امکان انداختن در یک نقطه‌ی خاص، امکان انداختن در یک ناحیه بررسی می‌شود. این ویژگی باعث می‌شود که داده‌های یک کلاس که به خوشه‌های کوچک تقسیم شده با یکدیگر ترکیب شده و خوشه‌های بهتری تشکیل دهند. به این شکل الگوریتم ارائه شده تعداد محاسبه f را کاهش خواهد داد و خوشه‌بندی بهتری تولید می‌کند.

5- نتایج شبیه سازیها

برای ارزیابی خوشه‌بندی انجام شده توسط روش پیشنهادی (Ant+Memory) و روش موجود (Ant+LA) که از حافظه‌ی کوتاه مدت استفاده می‌کند از 4 معیار که در ادامه به آنها اشاره می‌شود استفاده خواهد شد.

(4) و (5) یکی از الگوریتمهای یادگیری خطی در اتوماتای با ساختار متغیر را نشان می‌دهد.

الف- پاسخ مطلوب برای عمل i :

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \quad (4)$$

$$p_j(n+1) = (1 - a)p_j(n) \quad \forall j \neq i$$

ب- پاسخ نامطلوب برای عمل i :

$$p_i(n+1) = p_i(n) - (1 - b)p_i(n) \quad (5)$$

$$p_j(n+1) = \frac{b}{r-1} + (1 - b)p_j(n) \quad \forall j \neq i$$

و b به ترتیب پارامتر پاداش و پارامتر جریمه می‌باشد. زمانی که a و b با هم برابر باشند، الگوریتم $L_{R\text{-}P}$ ⁹، زمانیکه b از a خیلی کوچکتر باشد، الگوریتم $L_{R\text{-}ep}$ ¹⁰ و زمانی که b مساوی صفر باشد، الگوریتم $L_{R\text{-}I}$ ¹¹ نامیده می‌شود.

4- روش پیشنهادی

همانطور که قبلا در بخش 3 اشاره شد، "حافظه‌ی کوتاه مدت" یکی از روش‌هایی است که برای سرعت بخشیدن به همگرایی و افزایش کیفیت خوشه‌بندی در خوشه‌بندی مورچه‌ای ارایه شده است [2]. هر چند این روش کارایی خوشه‌بندی مورچه‌ای را افزایش می‌دهد ولی دارای معایبی می‌باشد. اولین عیب این روش اینست که زمانی که مورچه‌ای از حافظه‌ی کوتاه مدتی استفاده می‌کند باید به تعداد خانه‌های حافظه اش تابع f را محاسبه کند. محاسبه f یکی از زمان بر ترین محاسبات این الگوریتم می‌باشد و فراخوانی مکرر این تابع می‌تواند مدت این است که توانایی ترکیب خوشه‌های کوچک تشکیل شده را ندارد. در ادامه این بخش روشی که بر پایه‌ی اتوماتای یادگیر می‌باشد برای حل مشکلات ذکر شده پیشنهاد می‌گردد. در این روش صفحه را به n^2 ناحیه‌ی مربعی شکل مساوی تقسیم می‌کنیم و به هر مورچه یک اتوماتای یادگیر اختصاص می‌دهیم. این اتوماتاهای یادگیر دارای n^2 عمل می‌باشند که هر کدام از آنها متناظر با یکی از نواحی اختصاص داده شده می‌باشد. به این شکل هر مورچه برای انتخاب مکانی برای انداختن داده‌ی خود یک ناحیه را توسط اتوماتای یادگیرش تعیین می‌کند و نقطه‌ای را به صورت اتفاقی در مربع کوچکتری که در آن ناحیه در نظر گرفته می‌شود انتخاب می‌کند. اصلاح مربع داخلی برابر با $0/8$ ضلع ناحیه می‌باشد و در وسط آن قرار دارد. شکل یک ناحیه‌بندی 16 تایی را نشان می‌دهد. خطوط سیاه ناحیه‌ها را از یکدیگر جدا می‌کنند و مربع‌های خاکستری نشان

⁹Linear Reward Penalty

¹⁰Linear Reward Epsilon Penalty

¹¹Linear Reward Inaction

۵ داده‌ی مورد استفاده عبارتند از: **Iris** که دارای ۱۵۰ داده‌ی ۴ بعدی می‌باشد و در ۳ کلاس به شکل مساوی تقسیم شده‌اند، **Wisconsin** که از ۶۸۳ داده‌ی ۱۰ بعدی تشکیل شده و دارای ۲ کلاس با تعداد داده‌های ۲۳۹ و ۴۴۴ می‌باشد، **Ecoli** که از ۳۳۶ داده‌ی ۷ بعدی تشکیل شده و دارای ۷ کلاس با تعداد داده‌های ۱۴۳، ۳۵، ۵۲، ۷۹، ۲۰، ۵ و ۲ می‌باشد، **Data1** که از ۱۰۰۰ داده‌ی ۲ بعدی تشکیل شده است که به شکل مساوی در ۴ کلاس با توزیع نرمال با میانگین‌های (۵، ۰.۲)، (۵، ۰.۲)، (۵، ۰.۲) و (۵، ۰.۲) و انحراف معیار ۲ تقسیم شده‌اند و **Glass** که شامل ۲۱۴ داده‌ی ۹ بعدی است و به ۶ کلاس که دارای ۱۳، ۱۷، ۲۹، ۷۰، ۷۶ و ۲۹ داده‌ی می‌باشند تقسیم شده است. تمام داده‌ها در تمام ابعاد قبل از استفاده نرمال شده‌اند. برای اینکه ماتریس d (د.ک. فرمول (۳)) اعدادی بین ۰ و ۱ داشته باشد، اعداد این ماتریس نرمال شده‌اند. برای محاسبه‌ی ماتریس d برای تمام داده‌ها به غیر از Iris از فاصله‌ی اقلیدسی استفاده شده است و برای داده‌ی Iris از فاصله‌ی کسینوسی استفاده شده است. پارامترهای وابسته به داده‌ها که در هر دو روش به صورت یکسان مورد استفاده قرار گرفته‌اند در جدول ذکر شده‌اند.

جدول ۱: پارامترهای مشترک Ant+(Memory, LA)

	S	a	تعداد تکرارها
IRIS	5	0.03	50,000
Wisconsin	7	0.5	350,000
Ecoli	7	0.35	350,000
Data1	7	0.2	400,000
Glass	5	0.4	150,000

تعداد مورچه‌ها و k_{drop} در تمام آزمایش‌ها به ترتیب برابر با ۱۰، ۰.۱ و ۰.۱۵ قرار داده شده است. صفحه در تمام آزمایش‌ها مریعی با ضلع $\sqrt{10 \times N_{data}}$ می‌باشد. $step_size$ نیز برابر با یک پرش به تمام نقاط صفحه دسترسی دارد. حافظه‌ی کوتاه مدت Ant+Memory ظرفیت نگه داری ۱۰ مکان اندختن داده را دارد. در روش Ant+LA صفحه را به ۱۶ ناحیه تبدیل کرده‌ایم، بنابراین اتوماتیک یادگیری هر مورچه دارای ۱۶ عمل خواهد بود. Ant+LA از قانون یادگیری (۱) $L_{R-\theta P}$ ($a=0.01$ ، $b=0.0001$) استفاده می‌کند و جدول میانگین ۴ معیار را برای ۵ داده در ۲۰ تکرار گزارش می‌کند.

جدول ۲: مقایسه‌ی خوشبندی روش‌های Ant+LA و Ant+Memory و Kmeans

Wisconsin	<i>k-means</i>	Ant + Memory	Ant + LA
F-Measure	0.9699	0.8714	0.9653
Rand Index	0.9240	0.7274	0.8970
Entropy	0.1054	0.2994	0.1039
F Evaluations	-	440,631.20	375,152.05
IRIS	<i>k-means</i>	Ant + Memory	Ant + LA
F-Measure	0.8478	0.9533	0.9733

۱. **F-Measure**: این معیار دو مفهوم دقت^{۱۲} و به یاد آوری^{۱۳} که از مفاهیم بازیابی اطلاعات هستند، استفاده می‌کند. این معیار به این شکل تعریف می‌شود و مقداری بین ۰ و ۱ دارد و مقدادیر نزدیک به ۱ این معیار مطلوب هستند.

$$p(i, j) = \frac{n_{ij}}{n_j} \quad r(i, j) = \frac{n_{ij}}{n_i}$$

$$F(i, j) = \frac{(b^2 + 1) \cdot p(i, j) \cdot r(i, j)}{b^2 \cdot p(i, j) + r(i, j)} \quad (6)$$

$$F = \sum_i \frac{n_i}{n} \max_j \{F(i, j)\}$$

در فرمولهای فوق، n_{ij} و n_j و n_i به ترتیب برابراند با تعداد داده‌های کلاس i و تعداد داده‌های خوشه‌ی j (بیست آمده توسط الگوریتم خوشه‌بندی) و تعداد داده‌های کلاس i در خوشه‌ی j و p نیز به ترتیب نشان دهنده‌ی مفاهیم دقت و یادآوری می‌باشد. در این مقاله از $b=1$ استفاده شده است.

۲. **Rand Index**: فرض کنید که V آرایه‌ای باشد که خوشه‌ی داده‌ها (بیست آمده توسط الگوریتم خوشه‌بندی) و U آرایه‌ای باشد که کلاس داده‌ها را نشان می‌دهد. این معیار به این شکل تعریف می‌شود و مقداری بین ۰ و ۱ دارد. مقدادیر نزدیک به ۱ این معیار مطلوب هستند.

$$a = \{i, j \mid U(i) = U(j) \wedge V(i) = V(j)\}$$

$$b = \{i, j \mid U(i) = U(j) \wedge V(i) \neq V(j)\}$$

$$c = \{i, j \mid U(i) \neq U(j) \wedge V(i) = V(j)\}$$

$$d = \{i, j \mid U(i) \neq U(j) \wedge V(i) \neq V(j)\} \quad (7)$$

۳. **Entropy**: این پارامتر اعدادی بین ۰ و ۱ تولید می‌کند و اعداد نزدیک به صفر مطلوب می‌باشند.

$$R = \frac{a+d}{a+b+c+d}$$

$$p_{ij} = I(i, j) / n_i$$

$$e_i = - \sum_j p_{ij} \log p_{ij} \quad (8)$$

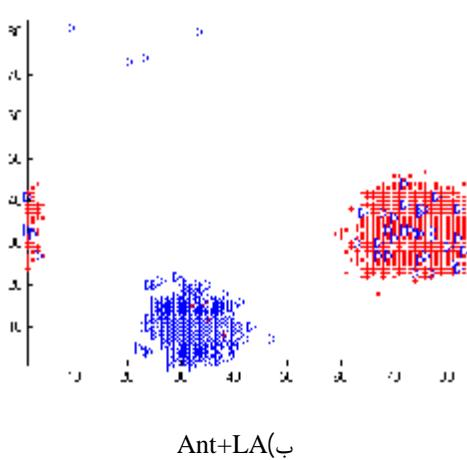
$$E = \sum_i e_i n_i / n$$

که در آن $I(i, j)$ تعداد داده‌های مشترک بین خوشه‌ی i و کلاس j می‌باشد و n_i و n_j به ترتیب تعداد داده‌های خوشه‌ی i و تعداد کل داده‌ها می‌باشد.

۴. **f** تعداد فراخوانی‌های تابع f (F Evaluation): محاسبه‌ی تابع f یکی از زمان بر ترین محاسبات این الگوریتم می‌باشد و هرچه کمتر f محاسبه شود سرعت الگوریتم بیشتر خواهد بود.

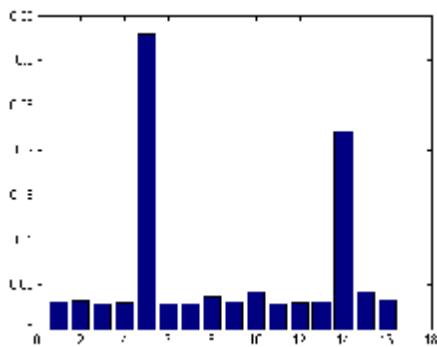
¹² Precision

¹³ Recall



ب)

همانطور که در شکل مشاهده میشود روش پیشنهادی توانسته است به خوبی خوشه ها را تشخیص داده و خوشه های کوچکی را که در طول اجرا تولید می شوند با هم ترکیب کند. این شکل همچنین نشان می دهد که *k-means* خوشه ها را به شکل صحیح تشخیص نداده است و به جای 2 خوشه 4 خوشه تشکیل داده است. شکل نشان می دهد که روش پیشنهادی چگونه توانسته است خوشه بندی صحیح را بدست آورد. در این شکل میانگین بردار احتمالات اتوماتای یادگیر مورچه ها را در روش Ant+LA در انتهایی یک اجرا نشان می دهد.

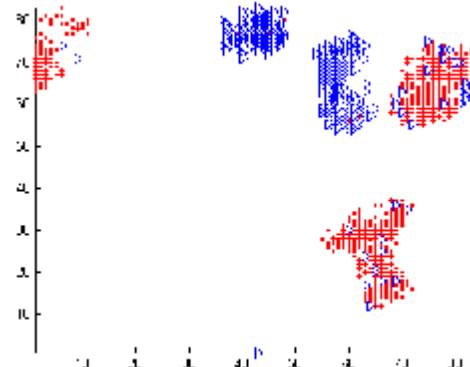


شکل 4: میانگین بردار احتمالات اتوماتای یادگیر در مورچه ها

همانطور که در شکل مشاهده میشود میانگین بردار احتمالات اتوماتای یادگیر مورچه ها نشان می دهد که آنها با احتمال زیاد دو ناحیه را برای انداختن داده ها استفاده می کنند. چنانچه شکل را با شکل **Error! Reference source not found.**-ب مقایسه کنیم می بینیم که داده ها در ناحیه های 5 و 14 تجمع کرده اند که نمودار شکل حاکی از این انتخاب توسط مورچه ها می باشد.

Rand Index	0.8627	0.9404	0.9656
Entropy	0.2519	0.1864	0.1051
F Evaluations	-	141,276.30	67,234.60
Data1	<i>k-means</i>	<i>Ant + Memory</i>	<i>Ant + LA</i>
F-Measure	0.9488	0.7567	0.9720
Rand Index	0.9526	0.8310	0.9727
Entropy	0.1787	0.4434	0.1324
F Evaluations	-	538,864.80	434,432.15
Ecoli	<i>k-means</i>	<i>Ant + Memory</i>	<i>Ant + LA</i>
F-Measure	0.6512	0.6230	0.7743
Rand Index	0.7991	0.7746	0.8325
Entropy	0.5156	0.7180	0.6226
F Evaluations	-	405,223.95	357,974.55
Glass	<i>k-means</i>	<i>Ant + Memory</i>	<i>Ant + LA</i>
F-Measure	0.4333	0.3850	0.4972
Rand Index	0.6692	0.6678	0.5743
Entropy	1.0769	1.2043	1.1190
F Evaluations	-	181,114.60	155,982.95

نتایج ارایه شده در جدول فوق نشان می دهد که استفاده از اتوماتای یادگیر برای تعیین کردن مکان انداختن داده بر روی صفحه می تواند کیفیت خوشه بندی را افزایش دهد) با توجه به معیار های F-Measure Rand Index Measure (Entropy. همچنین معیار تعداد محاسبه ها نشان می دهد که الگوریتم پیشنهادی احتیاج به محاسبات کمتری دارد و این معیار بین 12 تا 52 درصد نسبت به حافظه ی کوتاه مدت کاهش می یابد. **Error! Reference source not found.** نتایج خوشه بندی داده ی Wisconsin را برای دو روش Ant+(Memory,LA) نشان می دهد. جدول بالا همچنین کارایی روش ارائه شده را در مقایسه با K-Means با نشان می دهد.



الف) Ant+Memory)

from Nature, volume 2439 of LNCS, pp. 913–923. Springer-Verlag, Berlin, Germany, 2002.

[5] K. Hoe, W. Lai, and T. Tai. “Homogenous ants for web document similarity modeling and categorization”, Proceedings of the Third International Workshop on Ant Algorithms (ANTS 2002), volume 2463 of LNCS, pp. 256–261. Springer-Verlag, Berlin, Germany, 2002.

[6] P. Kuntz and D. Snyers. “New results on an ant-based heuristic for highlighting the organization of large graphs”, Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1451–1458. IEEE Press, Piscataway, NJ, 1999.

[7] P. Kuntz, D. Snyers, and P. Layzell. “A stochastic heuristic for visualizing graph clusters in a bi-dimensional space prior to partitioning”, Journal of Heuristics, Vol. 5, No. 3, pp. 327–351, 1998.

[8] P. Kuntz, P. Layzell and D. Snyers, “A colony of ant-like agents for partitioning in VLSI technology”, 4th European Conference on Artificial Life. MIT Press, July 1997.

[9] M. R. Meybodi, and H. Beigy, “A note on learning automata based schemes for adaptation of BP parameters”, Journal of Neurocomputing, Vol. 48, No. 4, pp. 957-974, October 2002.

[10] R. Rastegar, M. R. Meybodi, and K. Badie, “A New Discrete Binary Particle Swarm Optimization based on Learning Automata”, Proceedings of the International Conference on Machine Learning and Applications, ICMLA '04, pp. 456-462, 2004.

[11] M. Sheybani and M. R. Meybodi, “PSO-LA: A New Model for Optimization”, Proceedings of 12th Annual International Computer Society of Iran Computer Conference CSICC2007, Iran, pp. 1162-1169, 2007.

[12] H. Beigy, and M. R. Meybodi, “Utilizing Distributed Learning Automata to Solve Stochastic Shortest Path Problem”, International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, World Scientific Publishing Company, Vol. 14, No. 5, pp. 591-617, 2006

[13] M. A. L. Thathachar and P. S. Sastry, “Varieties of Learning Automata: An Overview”, IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, pp. 711-722, 2002.

6- نتیجه گیری

در این مقاله صفحه ای که مورچه ها در روش خوش بندی مورچه ای بر روی آن خوش بندی می کنند ناحیه بندی کردیم و به هر مورچه یک اتوماتای یادگیر که تعداد اعمال آن برابر با تعداد ناحیه هاست تخصیص دادیم. هدف از انجام این کار، فراهم آوردن مکانیزمی برای مورچه ها بود تا بتوانند ناحیه هایی را در صفحه که داده ها بیشتر در آنجا تجمع دارند، شناسایی کرده و با احتمال بیشتری توسط مورچه ها مورد بررسی بگیرند. آزمایشها نشان دادند که روش پیشنهادی، از تنها روش موجود برای تشخیص مکانهای مهم در صفحه برای خوش بندی احتیاج به محاسبات کمتری دارد و خوش بندی بهتری تولید می کند. همچنین مقایسه با روش K-Means برتری روش پیشنهادی را نشان داد.

مراجع

- [1] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chr'etien, “The dynamics of collective sorting: Robot-like ants and ant-like robots”, J.-A. Meyer and S. Wilson, editors, Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 1, pp. 356–365. MIT Press, Cambridge, MA, 1991.
- [2] E.D. Lumer ,B. Faieza, “Diversity and adaptation in populations of clustering ants” Proc. of the Third International Conference on The Simulation of Adaptive Behavior: From Animals to Animats 3, MIT Press, pp. 449–508, 1994.
- [3] J. Handl, J. Knowles, and M. Dorigo. “Strategies for the increased robustness of ant-based clustering”, Lecture Notes in Computer Science, Vol. 2977, pp. 90–104, 2004.
- [4] J. Handl and B. Meyer. “Improved ant-based clustering and sorting in a document retrieval interface”, Proceedings of the Seventh International Conference on Parallel Problem Solving