

بهبود روش خوشه بندی مورچه ای به کمک اتماتاهای یادگیر

برنا جعفرپور محمد رضا میبدی

آزمایشگاه محاسبات نرم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیر کبیر

تهران ایران

jafarpour@cic.aut.ac.ir, mmeybodi@aut.ac.ir

چکیده: خوشه بندی مورچه ای روشنی الهام گرفته از طبیعت برای خوشه بندی می باشد. این روش دارای مزایایی می باشد 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 اتماتای یادگیر برای یادگیری مکان قرار دادن داده ها بروی صفحه استفاده شده است. برای این منظور صفحه ای که مورچه ها بر روی آن خوشه بندی می کنند ناحیه بندی میشود و سپس هر مورچه مجهز به یک اتماتای یادگیر است. وضیفه اتماتای 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 "حافظه کوتاه مدت" نام دارد مقایسه شده است. نتایج آزمایشها بر روی ۵ مجموعه داده استاندارد نشان می دهد که روش پیشنهادی با محاسبات کمتر، خوشه بندی بهتری در مقایسه با روش "حافظه کوتاه مدت" بدست می آورد. همچنین مقایسه ها برتری روش ارائه شده را نسبت به K-means نشان می دهد.

کلمات کلیدی: خوشه بندی مورچه ای، اتماتاهای یادگیر، حافظه کوتاه مدت

Improving Ant based Clustering Technique using Learning Automata

B. Jafarpour M. R. Meybodi

Soft Computing Laboratory
Computer Engineering and Information Technology Department
Amirkabir University of Technology
Tehran Iran
jafarpour@cic.aut.ac.ir, mmeybodi@aut.ac.ir

Abstract

Ant based clustering is a bio-inspired method for clustering. This method has some interesting characteristic that has prompted special attention from the researchers in the past few years. In this paper an improvement on ant based clustering using learning automata is proposed. In improved ant based clustering every ant is equipped with a learning automaton which learns the place that the ant should drop its data item on the grid. To show the performance of the proposed method, it is tested on five standard data sets and the results are compared with the results obtained for the only reported method in the literature for determining important places in grid which is called "Short Term Memory" and K-means method. The results of comparison have shown the superiority of the proposed method over "Short Term Memory" and K-means methods in term of computational effort and accuracy.

Keywords: Ant based Clustering, Learning Automata, Short Term Memory

۱- مقدمه

روش خوشه بندی مورچه ای که الهام گرفته از رفتار مورچه ها در طبیعت می باشد اولین بار در [1] برای مجموعه ای از ربات ها که اشیاء را دسته بندی می کردند ارائه شد. این الگوریتم در [2] برای خوشه بندی داده ها 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 ۲ بعدی از سلولها وجود دارد که داده‌ها و مورچه‌ها برروی سلولهای آن قرار دارند و مورچه‌ها می‌توانند بر روی آنها حرکت کنند. در این روش برخلاف بسیاری از روش‌های خوشبندی احتیاج به از پیش تعیین کردن تعداد خوشبندی نمی‌باشد و تعداد آنها بطور خودکار در حین فرایند خوشبندی تعیین می‌گردد. یکی دیگر از ویژگی‌های مهم این روش اینست که خوشبندی در یک فضای ۲ بعدی به جای فضای n بعدی انجام می‌شود که امکان ارزیابی خوشبندی توسط انسان را فراهم می‌کند. یکی از بهبودهایی که در این روش ایجاد شده است، افزون حافظه کوتاه مدت^۱ به مورچه‌ها برای یادآوری مکانهایی است که قبل از آنچه داده ای را قرار داده اند [۳] و نشان داده شده است که این تغییر می‌تواند کارایی را در خوشبندی مورچه‌ای افزایش دهد. این روش دارای کاربردهای زیادی از جمله خوشبندی مستندات [۴][۵]، قسمت بندی^۲ گراف [۶][۷] و قسمت بندی در VLSI [۸] و غیره می‌باشد.

یک اتماتای یادگیر [۱۳] ماشینی 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 تقویتی^۳ به اتماتای یادگیر داده می‌شود و اتماتاً 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 اتماتای یادگیر آن را مناسب برای بسیاری کاربردها می‌کند. برای مثال از این ابزار در آموزش شبکه‌ی عصبی [۹]، محاسبات تکاملی [۱۰][۱۱]، حل مسئله‌ی کوتاهاستین مسیر اتفاقی [۱۲] و غیره استفاده شده است.

در این مقاله از اتماتاهای یادگیر برای یادگیری مکانهایی که مورچه‌ها برای انداختن داده‌ها بر روی صفحه باید در نظر بگیرند استفاده شده است. در روش پیشنهادی، صفحه ناحیه بندی می‌شود و هر مورچه دارای یک اتماتای یادگیر است که تعداد اعمال آن برابر است با تعداد ناحیه‌ها. اتماتای یادگیر هر مورچه به صورت احتمالاتی ناحیه‌ی قرار دادن داده‌ها را برای هر مورچه تعیین می‌کند. مورچه‌ها به صورت دست جمعی و بدون داشتن ارتباط مستقیم با یکدیگر، یاد می‌گیرند که چه ناحیه‌هایی برای انداختن داده‌ها مناسب‌تر می‌باشد. مقایسه‌ی روش پیشنهادی با تنها روش موجود برای تعیین مکان انداختن داده‌ها نشان می‌دهد که روش پیشنهادی احتیاج به محاسبات کمتری دارد و خوشبندی بهتری تولید می‌کند. ادامه مقاله بدین صورت سازماندهی شده است. در بخش‌های ۲ و ۳ به ترتیب خوشبندی مورچه‌ای و اتماتاهای یادگیر به اختصار شرح داده می‌شود. در بخش ۴ روش پیشنهادی و در بخش ۵ نتایج آزمایشها ارایه می‌گردد. بخش ۶ نتیجه‌گیری می‌باشد.

۲- خوشبندی مورچه‌ای

روش خوشبندی مورچه‌ای الهام گرفته از رفتار مورچه‌ها در طبیعت می‌باشد. مورچه‌ها در طبیعت می‌توانند اشیاء مختلف را بدون وجود کنترل مرکزی در دسته‌های مشابه در کنار یکدیگر جمع آوری کنند. با الهام گرفتن از این مدل، خوشبندی مورچه‌ای توسعه داده شده است و دارای کاربردهای زیادی در علم می‌باشد. در این مدل، مورچه‌ها و داده‌هایی که باید خوشبندی شوند بر روی صفحه ای دو بعدی قرار داده می‌شوند. مورچه‌ها می‌توانند داده‌هایی را که به داده‌های اطراف خود شبیه نیستند از جای خود در صفحه برداشته و در کنار داده‌های مشابه قرار دهند. مورچه‌ها برای ارزیابی اینکه داده‌ای به داده‌ی اطراف خود شبیه است یا خیر دارای دید محدودی می‌باشند. این مورچه‌ها، همانند مورچه‌های واقعی از کنترل مرکزی بر خوردار نیستند و به صورت مستقل از یکدیگر عمل می‌کنند. در هر سلول شبکه همزمان دو داده یا دو مورچه نمی‌توانند قرار بگیرند و چنانچه مورچه‌ای از یک طرف شبکه بیرون رود از طرف دیگر وارد خواهد شد. مورچه‌ها با احتمالات زیر داده n را از جای خود برداشته یا در مکانی قرار می‌دهند [۲].

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

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

¹ Short Term Memory

² Partitioning

³ Reinforcement Signal

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، $f(i)$ مقادیر بالا خواهد داشت و در نتیجه داده i با احتمال کمی برداشته می شود (فرمول (۱)) و با احتمال بالایی در آن مکان انداخته می شود(فرمول (۲)). 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 [۱] امکان مقایسه ای داده های عددی را فراهم نمی کند ولی این مشکل در [۲] حل و تابع زیر ارائه شد.

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

در محدوده $[0,1]$ قرار دارد و تفاوت داده های i و j را تعیین می کند. α پارامتر مقیاس^۴ است که در محدوده $[0,1]$ قرار دارد و باید با توجه به داده ها تعیین شود. σ محدوده دید را تعیین می کند به این معنی که محدوده i دید یک مورچه با پارامتر σ در هر 4 جهت برابر است با $\frac{(\sigma-1)}{2}$. i $\text{Neigh}(i)$ مجموعه ای داده هایی است که در همسایگی σ داده i n بر روی صفحه قرار دارند. برای افزایش کارایی الگوریتم "حافظه ای کوتاه مدت" در [۲] 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 جایجا شده باشد و طی این مسیر بی فایده باشد. برای حل این مشکل در [۳] 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 ای 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 موردنظر 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 تعليق در آورده و به صورت تصادفی عمل می کند. از جمله دیگر تغییراتی که برای بهبود در [۳] ارائه شد پارامتر $step_size$ می باشد. در این مدل مورچه ها به جای حرکت از یک سلو به سلو مجاور می توانند به سلوهایی که مجموع فاصله ای عمودی و افقی آنها حد اکثر $step_size$ باشد پرش کنند. شبیه کد الگوریتمی را که در این مقاله مورد بررسی قرار گرفته و برگرفته از [۳] 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 فعل ای 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 فعل ای 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 تصمیم می گیرد. چنانچه مورچه تصمیم به انداختن داده ای کرد و سلو مورد نظر توسط داده ای دیگری اشغال شده بود، مورچه داده را به صورت اتفاقی در یکی از سلو 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(<i>remaining items</i>)
5: pick up(<i>agent(j)</i> , i)
6: $g :=$ random select(<i>remaining empty grid locations</i>)
7: place agent(<i>agent(j)</i> , g)

⁴ Scaling Parameter

```

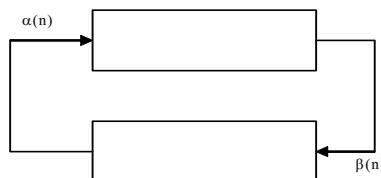
8: end for
9: for each it_ctr in 1 to #iterations do
10:   j := random select(all agents)
11:   step(agent(j), stepsize)
12:   i := carried item(agent(j))
13:   drop := drop item?(f(i)) // see equation 2
14:   if drop = TRUE then
15:     while pick = FALSE do
16:       i := random select(free data items)
17:       pick := pick item? // see equation 1
18:     end while
19:   end if
20: end for

```

الگوریتم خوش بندی مورچه ای

۳- اتوماتاهای یادگیر

یک اتوماتای یادگیر یک مدل انتزاعی است که تعداد محدودی عمل را می تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی به اتوماتای یادگیر داده می شود. اتوماتای 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]. اتوماتا با تعامل با محیط عمل بهینه را فراگیری می کند و به این شکل پاداش دریافتی خود را از محیط حد اکثر می کند. شکل ارتباط بین اتوماتای یادگیر و محیط را نشان می دهد.



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

محیط را می توان توسط یک سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودیها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالهای جریمه α ها می باشد. هر گاه β مجموعه دو عضوی باشد، محیط از نوع P می باشد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می شود. در محیط از نوع Q ، $\beta(n)$ می تواند به طور گستته یک مقدار از مقادیر محدود در فاصله $[0, 1]$ و در محیط از نوع S ، β متغیر تصادفی در فاصله $[0, 1]$ است. c_i احتمال اینکه عمل α_i نتیجه نامطلوب^۴ داشته باشد، می باشد. در محیط ایستا^۵ مقادیر c_i بدون تغییر می مانند، حال آن که در محیط غیر ایستا^۶ این مقادیر در طی زمان تغییر می کنند. اتوماتای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم می گردد [3],[13]. اتوماتای یادگیر با ساختار متغیر^۷ توسط ۴ تایی $\{\alpha, \beta, p, T\}$ نشان داده می شود که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عملهای اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودیهای اتوماتا و $p = \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از اعمال و T الگوریتم یادگیری می باشد. در این نوع از اتوماتاهای اگر عمل α_i در مرحله n ام انتخاب شود و این عمل، $p_i(n+1) = T[\alpha(n), \beta(n), p(n)]$ 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 (۴) و (۵) یکی از الگوریتمهای یادگیری خطی در اتوماتای با ساختار متغیر را نشان می دهد.

الف- پاسخ مطلوب برای عمل α :

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

^۴Unfavorable

^۵Stationary

^۶Non-Stationary

^۷Variable Structure Learning Automata

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

ب- پاسخ نامطلوب برای عمل \hat{a} :

$$\begin{aligned} p_i(n+1) &= p_i(n) - (1-b)p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) \quad \forall j \quad j \neq i \end{aligned} \quad (5)$$

a و b به ترتیب پارامتر پاداش و پارامتر جریمه می‌باشد. زمانی که a و b با هم برابر باشند، الگوریتم L_{R-EP} ^۹، زمانیکه b از a خیلی کوچکتر باشد، الگوریتم L_{R-EI} ^{۱۰} و زمانی که b مساوی صفر باشد، الگوریتم L_{R-I} ^{۱۱} نامیده می‌شود.

۴- روش پیشنهادی

همانطور که قبلا در بخش ۳ اشاره شد، "حافظه‌ی کوتاه مدت" یکی از روش‌هایی است که برای سرعت بخشیدن به همگرایی و افزایش کیفیت خوش بندی در خوش بندی مورچه‌ای ارایه شده است [2] [3]. هر چند این روش کارایی خوش بندی مورچه‌ای را افزایش می‌دهد ولی دارای معایبی می‌باشد. اولین عیب این روش اینست که زمانی که مورچه‌ای از حافظه‌ی کوتاه مدت استفاده می‌کند باید به تعداد خانه‌های حافظه اش تابع f را محاسبه کند. محاسبه‌ی f یکی از زمان برترین محاسبات این الگوریتم می‌باشد و فراخوانی مکرر این تابع می‌تواند محاسبات الگوریتم را افزایش دهد. عیب دیگر روش حافظه‌ی کوتاه مدت این است که توانایی ترکیب خوش‌های کوچک تشکیل شده را ندارد. در ادامه این بخش روشی که بر پایه‌ی اتوماتای یادگیر مبایش براي 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 یادگیریش تعیین می‌کند و نقطه‌ای را به صورت اتفاقی در مربع کوچکتری که در آن ناحیه در نظر گرفته می‌شود انتخاب می‌کند. اضلاع مربع داخلی برابر با $1/8$ ضلع ناحیه می‌باشد و در وسط آن قرار دارد. شکل یک ناحیه بندی ۱۶ تایی را نشان می‌دهد. خطوط سیاه ناحیه‌ها را از یکدیگر جدا می‌کنند و مربع‌های خاکستری نشان 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 ($\beta = 0$) و در غیر اینصورت جریمه می‌شود ($\beta = 1$). 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-EP} استفاده کرده ایم. 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 ۲: تقسیم بندی صفحه به ۱۶ 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 غیرفعال 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

^۹Linear Reward Penalty

^{۱۰}Linear Reward Epsilon Penalty

^{۱۱}Linear Reward Inaction

امکان انداختن در یک نقطه‌ی خاص، امکان انداختن در یک ناحیه برسی می‌شود. این ویژگی باعث می‌شود که داده‌های یک کلاس که به خوشه‌های کوچک تقسیم شده با یکدیگر ترکیب شده و خوشه‌های بهتری تشکیل دهنند. به این شکل الگوریتم ارائه شده تعداد محاسبه‌ی f را کاهش خواهد داد و خوشه‌بندی بهتری تولید می‌کند.

۵- نتایج شبیه سازیها

برای ارزیابی خوشه‌بندی انجام شده توسط روش پیشنهادی (Ant+LA) و روش موجود (Ant+Memory) که از حافظه‌ی کوتاه مدت استفاده می‌کند از ۴ معیار که در ادامه به آنها اشاره می‌شود استفاده خواهد شد.

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.

$$\begin{aligned} p(i, j) &= \frac{n_{ij}}{n_j} & r(i, j) &= \frac{n_{ij}}{n_i} \\ F(i, j) &= \frac{(b^2 + 1) \cdot p(i, j) \cdot r(i, j)}{b^2 \cdot p(i, j) + r(i, j)} & (6) \\ F &= \sum_i \frac{n_i}{n} \max_j \{F(i, j)\} \end{aligned}$$

در فرمولهای فوق، n_i و n_j به ترتیب برابرند با تعداد داده‌های کلاس i و j (بدست آمده توسط الگوریتم خوشه‌بندی) و تعداد داده‌های کلاس i در خوشه‌ی j p و r نیز به ترتیب نشان دهنده‌ی مفاهیم دقت و یادآوری می‌باشند. در این مقاله از $b=1$ استفاده شده است.

Rand Index. فرض کنید که V آرایه‌ای باشد که خوشه‌ی داده‌ها (بدست آمده توسط الگوریتم خوشه‌بندی) و U آرایه 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.

$$\begin{aligned} a &= \{i, j \mid U(i) = U(j) \wedge V(i) = V(j)\} \\ b &= \{i, j \mid U(i) = U(j) \wedge V(i) \neq V(j)\} \\ c &= \{i, j \mid U(i) \neq U(j) \wedge V(i) = V(j)\} \\ d &= \{i, j \mid U(i) \neq U(j) \wedge V(i) \neq V(j)\} \\ R &= \frac{a+d}{a+b+c+d} & (7) \end{aligned}$$

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

$$\begin{aligned} p_{ij} &= I(i, j) / n_i \\ e_i &= - \sum_j p_{ij} \log p_{ij} & (8) \\ E &= \sum_i e_i n_i / n \end{aligned}$$

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.

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.

Wisconsin Iris. که دارای ۱۵۰ داده‌ی ۴ بعدی می‌باشد و در ۳ کلاس به شکل مساوی تقسیم شده اند، 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، ۳۵، 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, ۵, ۲), (۵, ۲, -۵) و (-۵, ۲, -۲) و انحراف معیار ۲ تقسیم شده اند و

¹² Precision

¹³ Recall

که شامل ۲۱۴ داده‌ی ۹ بعدی است و به ۶ کلاس که دارای ۷۶، ۷۰، ۲۹، ۱۳، ۰ و ۹ داده‌می باشند تقسیم شده است. تمام داده‌ها در تمام ابعاد قبل از استفاده نرمال شده‌اند. برای اینکه ماتریس d (ر.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)

	σ	α	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

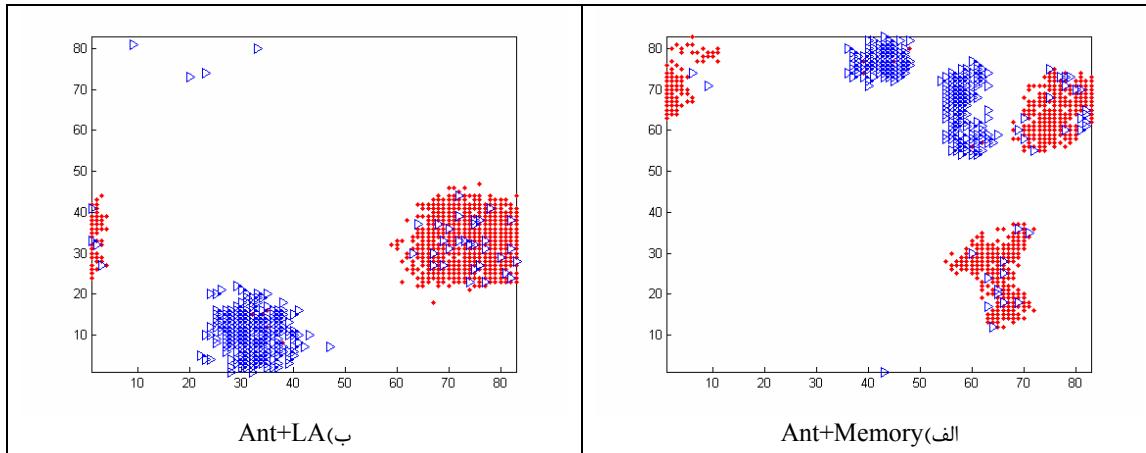
تعداد مورچه‌ها و k_{drop} در تمام آزمایش‌ها به ترتیب برابر با ۱۰، ۰.۱ و ۰.۱۵ قرار داده شده است. صفحه‌در تمام آزمایش‌ها مربعی با ضلع $\sqrt{10 \times N_{data}}$ می‌باشد. step_size نیز برابر با $\sqrt{10 \times N_{data}}$ ۲ در نظر گرفته شده است. به این شکل یک مورچه با یک پرش به تمام نقاط صفحه دسترسی دارد. حافظه‌ی کوتاه مدت Ant+Memory ظرفیت نگه داری ۱۰ مکان انداختن داده را دارد. در روش صفحه را به ۱۶ ناحیه تبدیل کرده‌ایم، بنابراین اتوماتای یادگیری هر مورچه دارای ۱۶ عمل خواهد بود. Ant+LA از قانون یادگیری استفاده می‌کند و جدول میانگین ۴ معیار را برای ۵ داده در ۲۰ تکرار گزارش می‌کند.

جدول ۲: مقایسه‌ی خوشبندی روشهای Ant+Memory و Ant+LA و Kmeans برای ۵ داده و ۴ معیار

Wisconsin	<i>k-means</i>	<i>Ant + Memory</i>	<i>Ant + LA</i>
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>	<i>Ant + Memory</i>	<i>Ant + LA</i>
F-Measure	0.8478	0.9533	0.9733
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

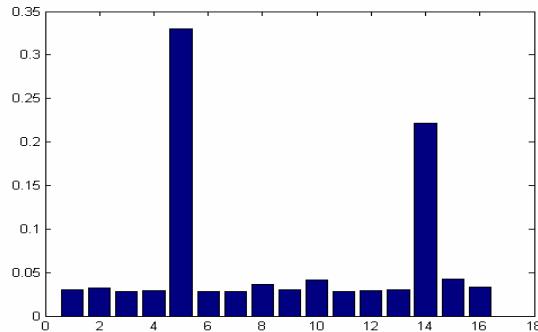
نتایج ارایه شده در جدول فوق نشان می‌دهد که استفاده از اتوماتای یادگیر برای تعیین کدن مکان انداختن داده بر روی صفحه می‌تواند کیفیت خوشبندی را افزایش دهد (با توجه به معیارهای F-Measure، Rand Index و Entropy). همچنین معیار تعداد محاسبه‌ی f نشان می‌دهد که الگوریتم پیشنهادی احتیاج به محاسبات کمتری دارد و این معیار بین ۱۲ تا ۵۲ درصد نسبت به حافظه‌ی کوتاه مدت

کاهش می یابد. شکل نتایج خوش بندی داده‌ی Ant+(Memory,LA) را برای دو روش Wisconsin نشان می‌دهد. جدول بالا همچنین کارایی روش ارائه شده را در مقایسه با K-Means نشان می‌دهد.



شکل ۳: مقایسه‌ی خوش بندی تولید شده برای داده‌ی Wisconsin توسط روش (الف) Ant+Memory و (ب) Ant+LA

همانطور که در شکل مشاهده می‌شود روش پیشنهادی توانسته است به خوبی خوش‌ها را تشخیص داده و خوش‌های کوچکی را که در طول اجرا تولید می‌شوند با هم ترکیب کند. این شکل همچنین نشان می‌دهد که Ant+Memory خوش‌ها را به شکل صحیح تشخیص نداده است و به جای ۲ خوش‌هه ۴ خوش‌هه تشکیل داده است. شکل نشان می‌دهد که روش پیشنهادی چگونه توانسته است خوش بندی صحیح را بدست آورد. در این شکل میانگین بردار احتمالات اتوماتای یادگیر مورچه‌ها را در روش Ant+LA در انتهای یک اجرا نشان می‌دهد.



شکل ۴: میانگین بردار احتمالات اتوماتای یادگیر در مورچه‌ها

همانطور که در شکل مشاهده می‌شود میانگین بردار احتمالات اتوماتای یادگیر مورچه‌ها نشان می‌دهد که آنها با احتمال زیاد دو ناحیه را برای انداختن داده‌ها استفاده می‌کنند. چنانچه شکل را با شکل سه ب مقایسه کنیم می‌بینیم که داده‌ها در ناحیه‌های ۵ و ۱۴ تجمع کرده اند که نمودار شکل حاکی از این انتخاب توسط مورچه‌ها می‌باشد.

۶- نتیجه‌گیری

در این مقاله صفحه‌ای که مورچه‌ها در روش خوش بندی مورچه‌ای بر روی آن خوش بندی می‌کنند ناحیه بندی کردیم و به هر مورچه یک اتوماتای یادگیر که تعداد اعمال آن برابر با تعداد ناحیه هاست تشخیص دادیم. هدف از انجام این کار، فراهم آوردن مکانیزمی برای مورچه‌ها بود تا بتوانند ناحیه‌هایی را در صفحه که داده‌ها بیشتر در آنجا تجمع دارند، شناسایی کرده و با احتمال بیشتری توسط مورچه‌ها مورد بررسی بگیرند. آزمایشها نشان دادند که روش پیشنهادی، از تها روش موجود برای تشخیص مکانهای مهم در صفحه برای خوش بندی احتیاج به محاسبات کمتری دارد و خوش بندی بهتری تولید می‌کند. 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. Faieta, "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 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.