الگوریتمی مبتنی بر اتوماتاهای یادگیر سلولی برای خوشه‌بندی مورچه- فازی داده‌ها

امیر اسمعیل‌زاده
دانشکده مهندسی کامپیوتر و فناوری اطلاعات،
دانشگاه آزاد اسلامی واحد قزوین، ایران
amir.esmaeilzadeh@qazviniau.ac.ir

محمدرضا میبدی
دانشکده مهندسی کامپیوتر و فناوری اطلاعات،
دانشگاه صنعتی امیرکبیر، تهران، ایران
mmeybodi@aut.ac.ir

چکیده: امروزه خوشه‌بندی یکی از مسائل مهم و پر کاربرد در زمینه-های گوناگون می‌باشد. اکثر الگوریتم‌های موجود از روش‌های سنتی استفاده می‌کنند که در برخی موارد از دقت و کارایی لازم برخوردار نمی‌باشند. این مقاله به بررسی و ارائه راهکاری مبتنی بر اتوماتاهای یادگیر در خوشه‌بندی مورچه فازی داده‌ها، جهت کاهش خطا و افزایش کارایی می‌پردازد. در این مقاله یک الگوریتم جدید مبتنی بر اتوماتاهای یادگیر سلولی برای خوشه‌بندی بهتر داده‌ها از طریق یافتن مقادیر مناسب معیارهای خوشه‌بندی مورچه-فازی ارائه می‌گردد. برای این منظور مجموعه‌ای از اتوماتاهای یادگیر سلولی با همدیگر برای یافتن مقادیر مناسب معیارهایی خوشه‌بندی مورچه-فازی همکاری می‌نمایند. در این مجموعه در هر مرحله اتوماتاهای یادگیر فعال شده و سپس هر یک از آنها یکی از اعمال خود را که همان انتخاب یکی از مقادیر مناسب می‌باشد انتخاب می‌کنند. اعمال انتخاب شده اجرا و با توجه به نتیجه اعمال آنها پاداش یا جریمه به آنها تعلق می‌گیرد. الگوریتم پیشنهاد شده سعی می‌کند تا مناسب ترین مقدار را برای معیارهای خوشه‌بندی مورچه- فازی که دارای حداقل خطا در خوشه‌بندی می‌باشند را پیدا کند تا خوشه‌بندی با دقت بالاتر و خطای کمتری انجام گیرد. روش پیشنهادی با الگوریتم‌های خوشه‌بندی فازی مقایسه شده است و نتایج حاصل بیانگر افزایش دقت و کاهش خطا در خوشه‌بندی می‌باشد.

واژه‌های کلیدی : خوشه‌بندی فازی، تابع C-میانگین فازی، اتوماتای یادگیر، خوشه‌بندی مورچه- فازی

۱- مقدمه

امروزه خوشه‌بندی یکی از مسائل مهم و پرکاربرد در زمینه‌های گوناگون می‌باشد. خوشه‌بندی یک تکنیک دسته بندی بدون نظارت است که در آن مجموعه داده‌ها که معمولا بردارهایی در فضای چند بعدی می باشند، بر اساس یک معیار شباهت یا عدم شباهت به تعداد مشخصی خوشه تقسیم می شوند. تاکنون تعداد بسیار متنوعی الگوریتم‌های خوشه‌بندی ارائه شده‌اند، اما به دلیل آنکه هیچ کدام از الگوریتم‌های ارائه شده قادر به خوشه‌بندی هر مجموعه داده‌ای دلخواه با کیفیت مناسب نیستند، تحقیقات برروی ارائه الگوریتم‌های جدید از ارزش بالایی برخوردار می باشد.

خوشه‌بندی‌های فازی و مورچه-فازی[1] نیز، از خوشه‌بندی‌های مهم در این زمینه می‌باشند. خوشه‌بندی مورچه-فازی داده‌ها ترکیبی از الگوریتم مورچه‌ها و خوشه‌بندی فازی می‌باشد[1],[2],[3]. در خوشه-بندی مورچه-فازی داده‌ها روش کار بدین صورت است که در ابتدا n داده بر روی یک صفحه $m*m$ به طوری که $m=4*n$ می‌باشد، به صورت تصادفی پراکنده می‌شوند و داده‌های پراکنده شده توسط عامل-ها(مورچه‌ها) جابجا شده و توده‌ها را تشکیل می‌دهند. لازم به ذکر می-باشد که تعداد عامل‌ها نیز برابر $\frac{n}{3}$ می‌باشد.

در تشکیل توده‌ها معیارها و پارامترهای مختلفی تاثیر گذار می‌باشند که عبارتند از :

- حد آستانه تشکیل توده[2]
- حد آستانه جدا کردن یک داده از یک توده توسط عامل[3]
- احتمال از بین بردن یک توده توسط عامل[4]
- احتمال برداشتن داده توسط عامل[5]
- احتمال انداختن داده توسط عامل[6]

در روش ذکر شده انتخاب یک مقدار مناسب برای شاخص‌ها و معیارهای ذکر شده فوق در خوشه‌بندی با دقت بالا و با خطای حداقل اهمیت زیادی دارد[1],[3]. لذا مقادیر شاخص‌ها و پارامترهای ذکر شده تاثیر مستقیمی در کاهش خطا و افزایش دقت خوشه‌بندی داشته و می‌توان با تعریف اتوماتاهای یادگیر بر روی پارامترهای فوق، به یک خوشه‌بندی مناسب با حداقل خطاها دست یافت.

در این مقاله، الگوریتم جدید خوشه‌بندی مورچه- فازی مبتنی بر اتوماتاهای یادگیر سلولی ارائه و ارزیابی خواهد شد. در این روش، با استفاده از مفاهیم اتوماتای سلولی، اتوماتاهای یادگیر و خوشه‌بندی مورچه- فازی مدلی ارائه می‌گردد که داده‌های خوشه‌بندی را از ورودی

دریافت و آنها را به نحو مطلوبی خوشه‌بندی نماید. در این مجموعه در ابتدا اتوماتای سلولی فعال شده و با استفاده از قوانین حاکم بر آن بر روی داده‌هایی که به صورت تصادفی بر روی صفحه دو بعدی پراکنده شده‌اند، اعمال می‌گردد و داده‌های مشابه را به همدیگر نزدیک می‌کند و در ادامه اتوماتاهای یادگیر نیز فعال می‌گردند و سپس هر یک از آنها اعمال خود را انتخاب می‌کنند که در این قسمت اعمال اتوماتاها، همان مقادیر خوشه‌بندی مورچه- فازی می‌باشند. اعمال انتخاب شده توسط اتوماتاها، اجرا و با توجه به نتیجه آن پاداش یا جریمه به عمل انتخاب شده تعلق می‌گیرد[4],[5].

الگوریتم ارائه شده بر این نکته تاکید دارد که مناسب‌ترین مقدار را برای خوشه‌بندی مورچه- فازی که دارای حداقل خطا در خوشه‌بندی می‌باشند را پیدا کند تا خوشه‌بندی با دقت و خطای کمتری انجام شود.

روشهایی که تاکنون ارائه شده‌اند دارای ضعف‌هایی می‌باشند و اکثر الگوریتم‌های ارائه شده با دو مشکل زیر روبرو می‌باشند :

تاثیر گروه‌بندی اولیه داده‌ها در نتایج خوشه‌بندی

حساسیت به شرایط اولیه الگوریتم

در رهیافت ارائه شده در الگوریتم پیشنهادی در این مقاله، با استفاده از اتوماتای سلولی و قوانین حاکم بر آن و ترکیب آن با اتوماتای یادگیر و خوشه‌بندی مورچه- فازی سعی شده تا موارد ذکر شده بالا برطرف شود که نتیجه آن بهبود قابل توجه نتایج خوشه‌بندی مورچه-فازی می‌باشد[6],[7].

ادامه این مقاله به صورت زیر سازماندهی شده است. ابتدا در بخش ۲ اتوماتای یادگیر سلولی به عنوان ابزار یادگیری مورد استفاده در این مقاله معرفی می‌گردد. در بخش ۳، الگوریتم پیشنهادی مبتنی بر اتوماتاهای یادگیر سلولی به منظور خوشه‌بندی مورچه-فاری با تعداد خطاهای کمتر ارائه خواهد شد. در نهایت در بخش ۴ به ارزیابی و مقایسه روش ارائه شده و روش‌های موجود C-میانگین فازی و خوشه-یابی مورچه-فازی ارائه شده توسط آقای M. Kanade (FAC) پرداخته می‌شود. در نهایت در بخش ۵، نتیجه‌گیری مقاله ارائه خواهد شد.

۲- اتوماتای یادگیر و اتوماتای یادگیرسلولی

در این بخش شرح مختصری درباره اتوماتاهای یادگیر و اتوماتای یادگیر سلولی ارائه می گردد.

۲-۱- اتوماتای یادگیر

اتوماتای یادگیر[8] یک ماشین با حالات محدود است که می تواند تعداد محدودی عمل را انجام دهد. هر عمل انتخاب شده، توسط محیطی احتمالی ارزیابی می گردد و پاسخی به اتوماتای یادگیر داده می شود. اتوماتای یادگیر از این پاسخ استفاده نموده و عمل بعدی را انتخاب می کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند.

محیط را می‌توان توسط سه‌تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه ورودی‌های محیط، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_r\}$ مجموعه خروجی‌های محیط و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمال‌های جریمه می‌باشند. ورودی محیط یکی از r عمل انتخاب شده توسط اتوماتای یادگیر است. خروجی (پاسخ) محیط برای هر عمل i توسط β_i مشخص می‌شود. اگر β_i یک پاسخ دودویی باشد، محیط نوع P نامیده می‌شود. در چنین محیطی $\beta_i(n) = 1$ بعنوان پاسخ نامطلوب یا شکست و $\beta_i(n) = 0$ بعنوان پاسخ مطلوب یا موفقیت در نظر گرفته می‌شوند. در محیط نوع Q، $\beta_i(n)$ شامل تعداد محدودی از مقادیر موجود در بازه [۰، ۱] می‌باشد. در محیط نوع S مقادیر $\beta_i(n)$ یک متغیر تصادفی در بازه [۰، ۱] می‌باشد. مجموعه c احتمالات جریمه (شکست) پاسخ‌های محیط را مشخص می‌کند.

الگوریتم بیان شده در روابط (۱) و (۲) بیانگر نمونه‌ای از الگوریتم خطی یادگیری در محیط از نوع P می‌باشد.

$$p_i(n+1) = p_i(n) - \beta(n).b.p_i(n) + (1-\beta(n)).a.[1-p_i(n)] \quad (۱)$$

$$p_j(n+1) = p_j(n) + \beta(n).(\frac{b}{r-1} + -b.p_j(n)) - (1-\beta(n)).a.(1-p_j(n)) \quad \forall j,\ j \neq i \quad (۲)$$

در روابط فوق r تعداد اقدام‌های اتوماتا، $\beta(n)$ سیگنال تقویتی تولیدی محیط ، a پارامتر پاداش و b پارامتر جریمه می‌باشد.

۲-۲- اتوماتای یادگیر سلولی

- اتوماتای یادگیر سلولی مجموعه‌ای متشکل از اجزاء ساده بوده که رفتار هر جزء بر اساس تجربیات گذشته و رفتار همسایگانش تعیین و اصلاح می‌شود. هر اتوماتای یادگیر سلولی، از یک اتوماتای سلولی تشکیل شده است که هر سلول آن به یک یا چند اتوماتای یادگیر مجهز می‌باشد. همانند اتوماتای سلولی، قانون محلی در محیط حاکم بوده، و این قانون تعیین می‌کند که عمل انتخاب شده توسط یک اتوماتا در سلول باید پاداش داده شود و یا اینکه جریمه دریافت نماید.

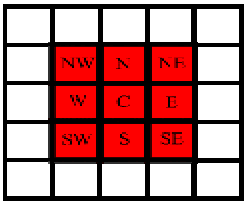
در اتوماتای یادگیر سلولی می‌توان از ساختارهای مختلفی برای همسایگی استفاده نمود. معمول ترین آنها همسایگی اسمیت، کول، ون

نیومن و مور می‌باشد. عملکرد اتوماتای یادگیر سلولی را می‌توان به شرح زیر بیان نمود. در هر لحظه هر اتوماتای یادگیر در اتوماتای یادگیر سلولی یک عمل از مجموعه اعمال خود را انتخاب می‌کند. عمل انتخاب شده با توجه به اعمال انتخاب شده توسط سلول‌های همسایه و قانون حاکم بر اتوماتای یادگیر سلولی پاداش و یا جریمه داده می‌شود. با توجه به اینکه عمل انتخاب شده پاداش گرفته و یا جریمه شده است، اتوماتا رفتار خود را تصحیح کرده و ساختار داخلی اتوماتا بهنگام می‌گردد. بعد از بروزرسانی، هر اتوماتای یادگیر در اتوماتای یادگیر سلولی دوباره یک عمل از مجموعه اعمال خود را انتخاب می‌نماید. فرآیند انتخاب عمل و دادن پاداش و یا جریمه تا زمانی که سیستم به حالت پایدار برسد و یا یک معیار از قبل تعریف شده‌ای برقرار شود، ادامه می‌یابد[4],[5].

۳- الگوریتم پیشنهادی

در این بخش الگوریتم پیشنهادی برای یافتن مقدار مناسب پارامتر برای شاخص‌ها و معیارهای خوشه‌بندی مورچه – فازی که دارای حداقل خطا در دسته‌بندی می‌باشد، ارائه می‌گردد. در بخش اتوماتای سلولی که در این قسمت پیاده‌سازی شده است از روش ارائه شده توسط آقایان A. Vande و J. Clayden استفاده شده است[6] ,[7] .

در این اتوماتای سلولی ارائه شده در این قسمت پارامتر آستانه شباهت داده‌ها از اهمیت بسیار بالایی برخوردار می‌باشد. در اعمال روش پیشنهادی، فرض بر این است که هر مورچه هیچ اطلاعی از نسبت به کل مجموعه داده‌ها نداشته و فقط می‌تواند با مورچه‌های همسایه خود ارتباط برقرار نماید. در اینجا همان طوری که در شکل ۱ مشاهده می‌گردد از همسایگی مور استفاده می‌گردد و وضعیت بعدی همه مورچه‌ها به وجود مورچه‌های همسایه و مقدار داده آنها بستگی دارد.



شکل۱: همسایگی مور

قبل از ارائه شبه کد الگوریتم پیشنهادی چند مفهوم و قوانینی که در اتوماتای سلولی مورد استفاده قرار گرفته، بررسی می‌گردد.

در هر تکرار، رفتارهای مورچه‌ها توسط استنتاج‌های زیر تعیین می‌شو [6]:

این الگوریتم داده‌های دیگر را مشابه به حساب می‌آورد در صورتی که فاصله بین مقادیر داده آنها از یک مقدار آستانه t کمتر باشد. میزان تشابه دو شی داده از رابطه(۳) تعیین می‌شود:

$$data_t = (z_{i1}, z_{i2}, \ldots, z_{ip}) \in R^p, p \in Z^+$$

$$d_{ij} = d(data_i - data_j) = \| data_i - data_j \|_p \quad (۳)$$

$$d_{ij} < t \Rightarrow similar(ant_i, ant_j) = true$$

مورچه‌هایی که به صورت تصادفی حرکت می‌کنند، خوشه‌های جدا و کوچکی ایجاد می‌کنند که چسبندگی ناپایداری به یکدیگر دارند. به همین دلیل از اتوماتای سلولی برای تعیین عمل بعدی مورچه استفاده شده است که قوانین آن به همسایه‌های هر مورچه وابسته است. بنابراین یک مورچه با تعداد کمتر از چهار همسایه مشابه، به یک سلول خالی منتقل می‌شود که ۱) هیچ همسایه نامشابهی نداشته باشد و ۲) در کنار مشابه‌ترین مورچه باشد. این قانون باعث می‌شود تا مورچه‌ها خوشه‌های بزرگ و پایدار ایجاد کنند.

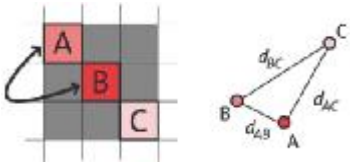
مورچه‌ها در حالت تنظیمات مناسب، یعنی با تعداد شش یا بیشتر مورچه همسایه مناسب، باز هم باید تلاش کنند تا از یک یا دو همسایه نامشابه باقیمانده فاصله بگیرند. این قانون باعث می‌شود تا خوشه‌های بزرگ یکدیگر را در حاشیه‌های بیرونی‌شان دفع کنند و بین آنها سلولهای خالی بوجود آید. این عمل باعث می‌شود تا داده‌ها بتوانند جایشان را با یکدیگر عوض کنند و در نتیجه به وضعیت مناسب‌تری برسند. علاوه بر این جابجایی باعث می‌شود تا مورچه‌هایی که در موقعیت نامناسبی گیر افتاده‌اند، بتوانند سریع‌تر از آن موقعیت خارج شده و به حاشیه‌های خارجی خوشه برسند. این خاصیت از ویژگی اتوماتای سلولی گرفته شده است که هر سلول می‌تواند ویژگی‌های سلول‌های اطرافش را دریافت کند و بر اساس آن موقعیت بعدی خود را تعیین کند. در هر تکرار، هر مورچه یکی از چهار جهت افقی، عمودی، مایل راست یا مایل چپ را به صورتتصادفی انتخاب کرده و فاصله d_{ij} را بین همه عناصر این سه سلول محاسبه می‌کند. بر اساس این فاصله، مورچه تصمیم می‌گیرد که آیا نیاز به جابجایی وجود دارد و یا اینکه مکان جاری مکان مناسبی می‌باشد. این جابجایی در این سه عنصر باید به گونه‌ای باشد که عناصر مشابه‌تر در کنار یکدیگر باشند. به عبارت دیگر این سه عنصر باید در جهتی که انتخاب شده است، مرتب شده باشند شکل ۲. قانون جابجایی تضمین می‌کند که در بین هر سه مورچه که به صورت خطی با یکدیگر همسایه هستند، دو عنصر بزرگتر از یکدیگر دو می‌باشند. این قانون را می‌توان توسط رابطه (۴) نشان داد [6].

$$d_{AC} > d_{AB}, d_{AC} > d_{BC} \Rightarrow ok \quad (۴)$$

$$d_{AC} < d_{AB}, d_{BC} < d_{AB} \Rightarrow swap(ant_B, ant_C)$$

$$d_{AC} < d_{AB}, d_{BC} > d_{AB} \Rightarrow swap(ant_A, ant_B)$$

در شکل ۲، فاصله دو عنصر B و C از یکدیگر بیشتر است، بنابراین در جابجایی باید این دو از هم دورتر باشند. در نتیجه برای مرتب کردن این سه عنصر، باید جای دو عنصر A و B را با یکدیگر عوض کرد. الگوریتم اتوماتای سلولی استفاده شده در الگوریتم پیشنهادی در شکل ۳ نشان داده شده است.



شکل۲

```
Procedure CA()
Begin CA
    Initialize all agents at random locations
    for each agent a do
      for each neighborhood cell n do
        if (n != empty) then
          if (|n.data – a.data| < tolerance t) then
            store n in array N
           neighbors = neighbors + 1
          else
            store n in array E
        if (neighbors = 0) then  /* no neighbors */
          for each cell c in E do
            move random
        if (neighbors < 4) then /* less than four data similar neighbors */
          for each cell c in E do
            l = c.clockwiseNeighbor
            r = c.counterclockwiseNeighbor
           if (l != empty and r != empty and |l.data–a.data| < tolerance and  |r.data–a.data| <  tolerance)
          then
             array diffs = |c.data – a.data|
        best = c with lowest diff
        move a to best
        if (neighbors > 5) then /* six or more similar neighbors */
         for each cell c in N do
           if (|N.data – a.data| >  tolerance) then
              tooclose = true
           if (tooclose) then
         for each cell c in E do
         l = c.clockwiseNeighbor
         r = c.counterclockwiseNeighbor
         if (l != empty && r!= empty && |l.data–a.data| < tolerance or  |r.data–a.data| < tolerance)
         then
            array diffs = |c.data – a.data|
        best = c with lowest diff
        move a to best
      dir = random direction
      l = cell on left side of a in direction dir
      r = cell on right side of a in direction dir
      if (l != empty and r != empty) then
        if (|l.data–a.data| < |r.data–a.data|) then
            swap l,a
        else
            swap r,a
End CA
```

شکل ۳: الگوریتم اتوماتای سلولی استفاده شده

الگوریتم ارائه شده در شکل ۴ شامل پنج مرحله مـی‌باشـد کـه در ادامه این مراحل را مورد بررسی قرار خواهیم داد.

```
Procedure  CLA-FAC
Begin LA-FAC
   Call CA()
   Construct the learning automata
   Initialize  PV_i(n) = 1/r ,  for  1 ≤ i ≤ r for all probabilistic vectors
   Initialize all fuzzy ant parameters
   While  N_iteration Do
    Begin
       Select randomly  PV_i(n)  for  T_Create
       NewErrorCount  =  Call FAC()   //calls Fuzzy Any Clustering (FAC)
       If iteration count <>first  iteration then
          If  NewErrorCount<LastErrorCount   then
             Reward the action selected by the Active LA  according to the  LRP  learning Algorithm
          Else
              penalize the action selected by the Active LA according to the LRP learning Algorithm
          End if
          LastErrorCount=NewErrorCount
       End if
       Present the best  PV_i,  for  1 ≤ i ≤ r`
     End
End CLA-FAC
```

شکل۴: الگوریتم پیشنهادی CLA-FAC

الگوریتم پیشنهادی را CLA-FAC می‌نامیم. مراحل الگوریتم به شـرح زیر است :

مرحله اول:

در ابتدا تمامی داده‌ها در سطح دو بعدی به صـورت تصـادفی پراکنـده می‌شوند و اتوماتای سلولی فعال می‌گردد و قوانین حاکم بر آن به تعداد مشخصی تکرار می‌شوند و داده‌های مشابه در سلول‌های نزدیک به هـم قرار می‌گیرند.

مرحله دوم:

در این مرحله یک اتوماتای یادگیر ایجاد می‌گردد. این اتوماتای یادگیر از الگوریتم یادگیری $L_{R\varepsilon P}$ اسـتفاده مـی‌کنـد. اقـدامهای اتوماتـای یادگیر بازه‌های مختلف مقادیر پارامترها می‌باشند که اتوماتـای یـادگیر می‌تواند انتخـاب کنـد. در اینجـا $tc = \{tc_1, tc_2, ..., tc_r\}$ اعمـال اتوماتای یادگیر می‌باشند که r تعداد اعمال قابل انتخاب توسط اتوماتا مـــی‌باشـــند. لـــذا بـــردار احتمـــال بـــه صـــورت $PV^j = \left(PV_1^j, PV_2^j, ..., PV`_r^j\right)$ تعریف مـی‌گـردد. در ابتدا بردار احتمال $\frac{1}{r}$ بوده و اعمال بـه صـورت تصـادفی انتخـاب مـی گردند و ورودی محیط یکی از r عمل انتخاب شده توسط اتوماتا می‌باشد. پاسخ محیط به عمل iام نیز توسط β_i مشخص مـی‌گـردد. اگـر $\beta_i(n) = 1$ باشد آن به صورت یک پاسخ نامطلوب یا شکست در نظر گرفته می‌شود و در غیر این صورت اگر $\beta_i(n) = 0$ باشـد آن بـه عنوان یک پاسخ مطلوب یا موفقیت محسوب می‌گردد.

مرحله سوم:

در این مرحله یکی از اقدام‌ها انتخاب می‌گردد و با توجه مقادیر انتخاب شده الگوریتم خوشه‌بندی مورچه- فازی اجرا می‌گردد. در این الگوریتم عامل‌ها(مورچه‌ها) و داده‌های تستی مورد به صـورت تصـادفی در یـک صفحه دو بعدی پراکنده می‌گردند. روند این الگوریتم بدین صورت است که عامل‌ها در این صفحه حرکت می‌کنند زمانی که یک عامل به یـک داده‌ای رسیدند در صورتی که خود عامل داده یا توده‌ای را حمل نمی-کند آن داده یا توده را با احتمالی که P_{pickup} نامیده می‌شود آن را بر می‌دارد و در غیر این صورت اگر داده یا توده‌ای را حمل می‌کند آن را با احتمـال P_{drop} آن را مـی‌انـدازد. در ادامـه الگـوریتم خوشـه‌هـای تشکیل شده به عنوان یک شی تنها در نظر گرفته می‌شوند و الگـوریتم به تعداد لازم بر روی آنها اجرا می‌گـردد و در پایـان تـابع C-میـانگین فازی بر روی خوشه‌ها اجرا می‌گردد و تعداد خطاهای خوشـه‌بنـدی بـه برنامه فراخوانی کننده برگردانده می‌شود.

مرحله چهارم:

در این مرحله خطای برگشتی توسط تابع خوشه‌بندی مورچه- فازی با خطای قبلی مقایسه و ارزیابی می‌گردد و در صورتی که تعـداد خطاهـا

کمتر شده باشد به آن عمل انتخاب شده اتوماتا پاداش داده می‌شود و در غیر این صورت جریمه به آن تعلق خواهد گرفت.

مرحله پنجم:

مراحل دوم تا چهارم بسته به نوع داده‌های تستی به تعداد لازم تکرار می‌گردند تا اتوماتاهای یادگیر مقادیر مناسب را جهت خوشه بندی مورچه- فازی استخراج کنند. شبه کد الگوریتم‌های ذکر شده وتابع اتوماتای سلولی استفاده شده در این الگوریتم در شکل ۳ و شکل ۴ نشان داده شده‌اند.

۴- ارزیابی و شبیه‌سازی

الگوریتم پیشنهاد شده(CLA-FAC) در بخش قبلی بر روی داده‌های تستی ایریس و چند داده تستی دیگر اعمال گردیده است که نتایج آن با روش خوشه‌بندی C-میانگین فازی و الگوریتم خوشه‌یابی مورچه- فازی مقایسه و ارزیابی شده است. با توجه به این نکته که هر کدام از داده‌های تستی دارای ویژگیها و ابعاد محتلفی می‌باشند، لذا مقادیر این ویژگیها را می‌توان با استفاده از روشهای مختلف نرمال نمود تا از آنها نتایج مطلوب را بدست آید. یکی از روشهای معمول استفاده از روش مینکوفسکی رابطه () و استفاده از روش اقلیدسی رابطه

(۶) می‌باشد که با استفاده از روابط ذکر شده داده نرمال و مورد استفاده قرار می‌گیرند.

$$d(i,j)=\left(|x_{i1}-x_{j1}|^q+|x_{i2}-x_{j2}|^q+\ldots+|x_{ip}-x_{jp}|^q\right)^q \quad (۵)$$

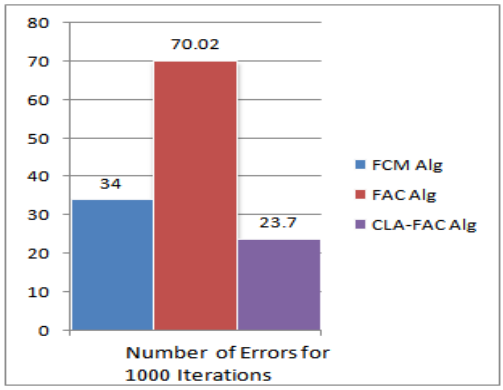
$$d(i,j)=\left(|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+\ldots+|x_{ip}-x_{jp}|\right) \quad (۶)$$

در خوشه‌بندی مورچه- فازی داده‌ها پارامترها و معیارهای مختلفی دخالت دارند که از جمله مهمترین و تاثیر گذارترین آنها معیار ساخت توده می‌باشد. در آزمایشاتی که توسط آقای M.Kanade به صورت تجربی بر روی داده‌های تستی مختلف انجام گردیده بهترین مقدار برای ساخت توده برابر ۰.۵ پیشنهاد شده است که خود این مقدار نیز، با خطای نسبتا زیادی همراه می‌باشد ولی در صورتی که از الگوریتم پیشنهادی بخش قبل استفاده گردد، نتیجه بهتری حاصل می‌گردد. در این بخش روش C-میانگین فازی و روش ارائه شده توسط آقای Kanade را با روش پیشنهادی در این بخش مقایسه و ارزیابی می‌گردد[1].

مجموعه داده‌های ایریس یک مثال استاندارد به منظور استفاده در سیستم‌های خوشه‌بندی می‌باشند. برای پیدا کردن مناسب‌ترین مقدار برای معیار ساخت توده(T_Create) از اتوماتاهای یادگیر استفاده می‌گردد. در این روش بازه بین صفر و یک را به ۲۰ قسمت مساوی تقسیم می‌گردد و به عنوان اقدامهای اتوماتا در نظر گرفته می‌شود و با توجه به اینکه بازه به ۲۰ قسمت تقسیم شده، مقدار بردار احتمال هرکدام نیز در ابتدا برابر $\frac{1}{20}$ قرار داده می‌شود و در چند مرحله که هر کدام در میانگین تعداد اجراها متفاوت هستند، مورد بررسی و ارزیابی قرار می‌گیرند. نتایج حاصل در جدول ۱ و نمودار شکل ۵ نشان داده شده‌اند.

جدول ۱: تعداد خطا برای داده‌های تستی ایریس

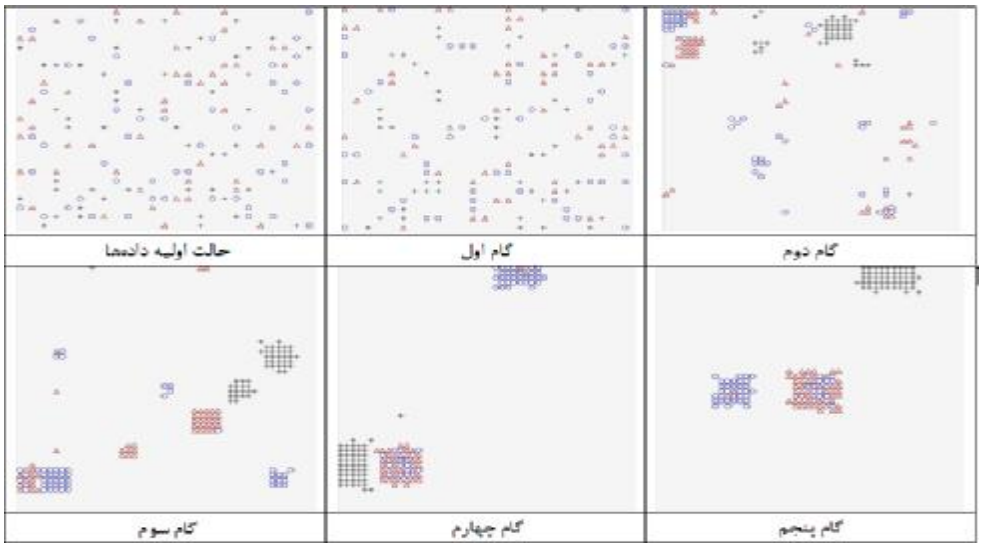
الگوریتم	تعداد تکرار	تعداد خطا
FCM	---	34
FAC	1000	70.02
CLA-FAC	1000	23.7
FAC	2000	61.76
CLA-FAC	2000	18.5
FAC	5000	64.32
CLA-FAC	5000	20.2
FAC	10000	66.56
CLA-FAC	10000	32.8



شکل ۵: مقایسه تعداد خطا برای الگوریتم پیشنهادی و الگوریتم‌های دیگر

از نتایج جدول۱ و نمودار شکل ۵ مشاهده می‌گردد که تعداد خطاهای الگوریتم پیشنهادی(CLA-FAC) به مراتب کمتر از تعداد سایر الگوریتم‌های موجود می‌باشد.

همچنین در شکل۶ مراحل اعمال الگوریتم به صورت دو بعدی نشان داده شده است.



شکل ۶: مراحل خوشه‌بندی داده‌های تستی ایریس با ۱۰۰۰ تکرار

الگوریتم پیشنهادی بر داده‌هایی با حجم بزرگتر نیز بررسی گردید که نتایج آنها را در جدول ۲ نشان داده است.

جدول ۲ : نتایج بدست آمده بر روی داده‌های تستی بالانس و یاایست

نوع داده تستی	الگوریتم	تعداد تکرار	تعداد خطا
Balance Scale	FCM	---	359
	FAC	1000	332.36
		2000	324.98
		5000	320.62
		10000	314.96
	CLA-FAC روش پیشنهادی	1000	285.8
		2000	284
		5000	311.5
		10000	289.8
Yeast	FCM	---	1167
	FAC	1000	1089
		2000	1202.24
		5000	1165.16
		10000	1158.66
	CLA-FAC روش پیشنهادی	1000	1023.2
		2000	1018
		5000	1022
		10000	1028.4

۴-۱- نتیجه‌گیری

در این مقاله، الگوریتمی مبتنی بر اتوماتاهای یادگیر سلولی که برای خوشه‌بندی فازی داده مورد استفاده قرار می‌گیرد، معرفی گردید. این الگوریتم مساله را در شرایطی که نوع کلاس و تعداد داده‌ها از قبل مشخص نیست، حل می‌کند به نحوی که مناسب‌ترین مقدار را برای پارامترهای مورد نظر که دارای حداقل خطا باشند، انتخاب می‌کند.

با توجه به نتایج بدست آمده بر روی داده‌های تستی ایریس، الگوریتم پیشنهادی را بر داده‌های تستی با تعداد داده بیشتری مورد ارزیابی قرار گرفت که تعداد خطا در آنها نیز کمتر از روش‌های معمول می‌باشد.

با توجه به نتایج بدست آمده برای داده‌های تستی فوق می‌توان نتیجه‌گیری کرد که الگوریتم فوق برای داده‌ها با حجم بیشتر نیز، دارای خروجی بهتری و خطای کمتری نسبت به روش‌های قبلی می‌باشد.

۵- مراجع

[1] M. Kanade and L. Hall, "Fuzzy ants as a clustering concept," Fuzzy Information Processing Society, 2003. NAFIPS 2003. 22nd International Conference of the North American, pp. 227-232, 2003.

[2] M. Kanade and L.Hall, "Fuzzy Ants and Clustering," *Proceedings of Systems Man and Cybernetics IEEE International Conference,* Vol 37, pp. 758-769, 2007.

[3] V. Rozin and M. Margaliot, "Fuzzy ants," *IEEE Computational Intelligence Magazine*, Vol 8, pp. 18-28, 2006.

[4] M. Thathachar and P. Sastry, "Varieties of Learning Automata: An Overview" *IEEE Transaction on Systems*, Vol. 32, No. 6, pp. 711-722, 2002.

[5] H. Beigy and M.R. Meybodi, "A mathematical framework for cellular learning automata" *Advances in complex systems*, Vol. 7, Nos. 3&4, pp. 295-319, 2004.

[6] L. Chen and Xiaohua, "A novel ant clustering algorithm based on cellular automata" *Web Intelligence and Agent Systems*, Vol. 5, No. 1, pp.1-14, 2007.

[7] M. Sato and L. C. Jain, "Innovations in Fuzzy Clustering", Springer-Verlag Berlin Heidelberg, Volume 205, 2006.

[8] P. Sarkar, "A Brief History of Cellular Automata", *ACM Computing Surveys*, Vol 32, Issue 1, pp. 80-107, 2000.

[9] M. Kanade and L. Hail, "Fuzzy Ants Clustering by Centroid Positioning", Proceedings of Fuzzy Systems IEEE International Conference, Vol 1, pp. 371-376, 2004.

[10] J. Handl and B. Meyer, "Ant-Based and Swarm-Based Clustering", *Swarm Intelligence*, No. 1, pp. 95–113, 2007.

[11] A. V. Moere and J. J. Clayden, "Cellular Ants: Combining Ant-based Clustering with Cellular Automata", *17th IEEE international conference on tools with artificial intelligence*, *ICTAI 2005*, pp. 177–184, 2005.

[12] A. Moere and J.Clayden, "Cellular Ants: Combining Ant-Based Clustering with Cellular Automata", *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence*, Vol 3409, Issue 1, 2005.

[۱۳] برنا جعفر پور، محمدرضا میبدی، "بهبود روش خوشه‌بندی مورچه ای به روش اتوماتای یادگیر"، سیزدهمین کنفرانس ملی انجمن کامپیوتر ایران، جزیره کیش، خلیج فارس، ایران ۱۹ الی ۲۱ اسفند ۱۳۸۶.

زیرنویس‌ها

[1] Fuzzy Ant Clustering

[2] Threshold for Create Heap

[3] Threshold for Remove Data from Heap

[4] Probability of Destroy Heap

[5] Probability of Pickup

[6] Probability of Drop