

# یک الگوریتم خوشبندی مبتنی بر اتوماتای یادگیر سلولی

<sup>۱</sup>; محمد رضا میبدی<sup>۲</sup>; سید میثم حسینی سدهی

## چکیده

اتوماتای یادگیر سلولی مجموعه‌ای متشکل از اجزاء ساده بوده که رفتار هر جزء بر اساس تجربیات گذشته و رفتار همسایگانش تعیین و اصلاح می‌شود. اجزاء ساده تشکیل دهنده این مدل، از طریق تعامل با یکدیگر رفتار پیچیده‌ای از خود نشان می‌دهند. هر اتوماتای یادگیر سلولی، از یک اتوماتای سلولی تشکیل شده است که هر سلول آن به یک یا چند اتوماتای یادگیر مجهز می‌باشد. در این مقاله ابتدا نسخه‌ای از اتوماتای یادگیر سلولی که در آن اتوماتاهای یادگیر می‌توانند بین سلول‌ها حرکت کنند پیشنهاد می‌شود و سپس یک کاربرد از آن در خوشبندی ارائه می‌گردد. به منظور ارزیابی، الگوریتم خوشبندی پیشنهادی بر روی تعدادی دادگان استاندارد آزمایش و نتایج بدست آمده با نتایج حاصله برای الگوریتم‌های ASM و K-means مقایسه گردیده است. نتایج مقایسه حاکی از کارایی بالاتر الگوریتم خوشبندی پیشنهادی نسبت به الگوریتم‌های ASM و K-means می‌باشد.

## کلمات کلیدی

اتوماتای یادگیر سلولی، خوشبندی، اتوماتای یادگیر

## A Data Clustering Algorithm based on Cellular Learning Automata

Meisam Hosseini Sedehi; Mohammad Reza Meybodi

Computer Engineering and Information Technology Department

### Abstract

Cellular learning automata which is obtained by combining cellular automata and learning automata is a mathematical model for dynamical complex systems that consists of a large number of simple learning components. In this paper first a new version of cellular learning automata in which the learning automaton in each cell is capable of moving between cells introduced and then an application of this new version to data clustering is given. In order to show the performance of the proposed data clustering algorithm it is tested on several standard data sets and then the results are compared with the results obtained for ASM and K-means algorithms. The results of comparisons show the superiority of the proposed clustering algorithm over the ASM and K-means algorithms.

### Keywords

Cellular Learning automata, Clustering, Learning Automata

<sup>۱</sup> دانشجوی کارشناسی ارشد، گرایش هوش مصنوعی و رباتیک، دانشگاه امیرکبیر، Sehrestan@Hotmail.com

<sup>۲</sup> عضو هیأت علمی دانشکده کامپیوتر و فناوری اطلاعات، دانشگاه امیرکبیر، mmeybodi@Aut.ac.ir



## ۱. مقدمه

خوشبندی دادگان، کاربردهای فراوانی در علوم پزشکی، مهندسی و صنعت دارد. اهمیت خوشبندی در علوم مختلف و همچنین نوع دادگان مورد استفاده، سرعت خوشبندی، دقت و بسیاری پارامترهای دیگر باعث معرفی روش‌ها و الگوریتم‌های متنوعی از خوشبندی دادگان شده است. خوشبندی یک تکنیک دسته بنده بدون نظرات است که در آن مجموعه دادگان که معمولاً بردارهایی در فضای چند بعدی می‌باشند، بر اساس یک معیار شباهت یا عدم شباهت به تعداد مشخصی خوشبندی تقسیم می‌شوند. روش سلسله مراتبی<sup>۱</sup> و روش جزء‌بندی<sup>۲</sup> دو روش متداول در خوشبندی دادگان می‌باشند [۱]. در روش سلسله مراتبی، خوشبندی ساختاری شبیه درخت داشته که تمامی داده‌ها به اولین گره درخت متعلق بوده و هرچه در شاخه‌های درخت پیش می‌رویم، خوشبندی دقیق‌تر می‌شود. در روش‌های خوشبندی به صورت جزء‌بندی، داده‌ها بر پایه مراکز خوشبندی تقسیم شده و بر اساس شباهت، تعلق داده به یکی از خوشبندی‌ها تعیین می‌شود. خوشبندی بر اساس ساختار توری<sup>۳</sup> از روش‌های مطرح امروزی است. در این روش داده‌های موجود در دادگان به سلول‌های توری متناظر شده و بر اساس ساختار دادگان و روش پایه مورد استفاده در خوشبندی، ساختارهای مختلفی از خوشبندی معرفی شده است. از روش‌های بر پایه توری می‌توان به مدل Sting (ونگ و همکارانش ۱۹۹۷) که جستجو بر اساس اطلاعات آماری ذخیره شده در توری می‌باشد [۲]. WaveCluster (شیخ‌الاسلامی و همکارانش ۱۹۹۸) از تبدیل ویولت استفاده می‌نماید [۳] و Clique (آگراول و همکارانش ۱۹۹۸) بر مبنای ساختار چگالی دادگان در توری [۴]، اشاره نمود. الگوریتم‌های بر پایه توری عموماً از سرعت بالایی برخوردار بوده چرا که خوشبندی در آنها وابسته به تعداد سلول‌های توری می‌باشد.

در برخی مدل‌های خوشبندی امروزی از ترکیب ساختار توری و روش‌های گروه‌های هوشمند<sup>۴</sup> الهام گرفته شده از طبیعت مانند رفتار مورچگان، حرکت پرنده‌گان و چرخش ماهی‌ها استفاده می‌شود. در بعضی از این الگوریتم‌ها، محیط خوشبندی به صورت توری و در غالب اتوماتای سلولی و قوانین محلی حاکم بر آن می‌باشد. اتوماتای سلولی<sup>۵</sup> (CA)، توری است که هر سلول تعداد مشخصی حالت داشته و تعیین حالت هر سلول در هر مرحله وابسته به حالات سلول‌های همسایه و قوانین محلی حاکم می‌باشد [۵]. از این دسته می‌توان به الگوریتم‌های ASM (چن و همکارانش ۲۰۰۴) که ترکیب اتوماتای سلولی و مدل خواب مورچگان می‌باشد [۶] و Cellular Ant (وند موئر و همکارانش ۲۰۰۵) که ترکیب خوشبندی مورچگان بر اساس فرومون و اتوماتای سلولی می‌باشد [۷]، اشاره نمود.

یک اتوماتای یادگیر سلولی یک اتوماتای سلولی است که هر سلول آن به یک یا چند اتوماتای یادگیر<sup>۶</sup> (LA) که وظیفه تعیین حالت آن سلول را به عهده دارند، تجهیز شده است [۸]. در اتوماتای یادگیر سلولی هر سلول بر اساس اتوماتای یادگیر موجود در آن عملی را انتخاب نموده و بر پایه اعمال انتخابی همسایگان سلول و قانون محلی حاکم بر همسایگی، عمل انتخابی سلول پاداش و یا جریمه دریافت می‌نماید. در این مقاله ابتدا مدل جدیدی از اتوماتای یادگیر سلولی با نام اتوماتای یادگیر سلولی متحرک معرفی می‌گردد و سپس یک کاربرد از آن در خوشبندی دادگان ارائه می‌شود. در اتوماتای یادگیر سلولی متحرک تعداد اتوماتاهای یادگیر از تعداد سلول‌ها تبعیت نکرده و وابسته به مسئله می‌باشد. همچنین اتوماتاهای یادگیر در محیط اتوماتای سلولی، توانایی حرکت دارند.

در الگوریتم خوشبندی پیشنهادی مبتنی بر اتوماتای یادگیر سلولی متحرک، به ازای هر داده یک اتوماتای یادگیر که در حکم یک عامل می‌باشد، در نظر گفته شده است. هر عمل یک اتوماتای یادگیر، متناظر با یک جهت حرکت تعریف شده برای اتوماتای یادگیر می‌باشد. هرچه اتوماتای یادگیر با همسایگانش شباهت بیشتری داشته باشد، تمایل حرکت به سمت آنها بیشتر و هر چه میزان تشابه آنها کمتر باشد، تمایل حرکت به سمت آنها کمتر می‌شود. از تعامل هر اتوماتای یادگیر با همسایگانش، اتوماتاهای یادگیری که شباهت بیشتری داشته باشند، در یک راستای خاص حرکت نموده که با افزایش تعداد تکرار، هر اتوماتای یادگیر جهت مناسب حرکت خود که در راستای همسایگان مشابه می‌باشد را یاد می‌گیرد.

برای جلوگیری از برخورد عامل‌ها در الگوریتم‌های خوشبندی مبتنی بر اتوماتای سلولی، در هر سلول تنها اجازه حضور یک عامل وجود داشته و لذا بروزرسانی اتوماتای سلولی به صورت ناهمگام و به ترتیب خطی انجام می‌پذیرد. در الگوریتم پیشنهادی جدید در هر سلول امکان حضور بیش از یک عامل یادگیر وجود دارد که بروزرسانی سلول‌های اتوماتای یادگیر سلولی را به صورت موازی و همزمان، امکان پذیر می‌نماید.

کارایی الگوریتم خوشبندی پیشنهادی بر روی چند دادگان استاندارد، با الگوریتم‌های ASM و K-means مورد مقایسه قرار گرفته است. نتایج آزمایشات انجام شده بر روی دادگان مختلف بیانگر کارایی مطلوب این الگوریتم در مقایسه با الگوریتم‌های ASM و K-means در خوشبندی دادگان می‌باشد.

سازماندهی بخش‌های این مقاله به ترتیب زیر می‌باشد. در بخش دو الگوریتم‌های ASM و K-means به اختصار بیان شده‌اند. در بخش ۳ اتوماتاهای یادگیر، اتوماتای یادگیر سلولی به اختصار تشریح و همچنین مدل جدید اتوماتای یادگیر سلولی متحرک معرفی شده



است. در بخش ۴ الگوریتم پیشنهادی خوشبندی جدید مبتنی بر اتوماتای یادگیر سلولی متحرک شرح داده شده است. در بخش ۵ نتایج حاصل از پیاده‌سازی الگوریتم پیشنهادی و در نهایت در بخش نهایی، نتیجه‌گیری مقاله آورده شده است.

## ۲. الگوریتم‌های خوشبندی

در این بخش دو الگوریتم K-means و ASM که با الگوریتم پیشنهادی مقایسه خواهند شد، به اجمال شرح داده می‌شوند. الگوریتم خوشبندی ASM از ساختار توری در غالب اتوماتای سلولی استفاده می‌نماید. و الگوریتم K-means متعلق به الگوریتم‌های خوشبندی با روش جزء‌بندی می‌باشد.

**الگوریتم K-means:** در این الگوریتم از میانگین داده‌ها به عنوان مرکز خوشه‌ها استفاده می‌شود.تابع مورد ارزیابی، تابع مربعات خطابوده که طبق رابطه (۱) تعریف می‌شود.

$$E = \sum_{i=1}^k \sum_{x \in C_i} |x - m_i|^2 \quad (1)$$

در رابطه (۱)،  $x$  داده در فضای مورد بررسی و  $m_i$  میانگین خوشه  $C_i$  می‌باشد. الگوریتم K-means در دو فاز کار می‌نماید. در فاز اول تعلق هر داده به خوشه‌های موجود مورد بررسی قرار می‌گیرد. هر داده به خوشه‌ای تعلق داشته که نزدیکترین فاصله را با مرکز آن خوشه داشته باشد. در مرحله دوم مرکز خوشه‌ها با توجه به داده‌های موجود در هر خوشه محاسبه مجدد می‌شوند. این روند تا زمانی که تغییرات تابع ارزیابی از یک مقدار از پیش تعیین شده‌ای بیشتر باشد، ادامه می‌یابد [۱].

**الگوریتم ASM:** این الگوریتم زیر مجموعه‌ای از الگوریتم خوشبندی مورچگان می‌باشد. محیط مورد استفاده در این الگوریتم به صورت توری در نظر گرفته شده است. در این الگوریتم هر مورچه در حکم یک داده بوده که در محیط سلولی در حرکت است. اعمال هر سلول شامل دو عمل وجود و یا عدم وجود داده در سلول می‌باشد. روند کلی الگوریتم ASM به صورت زیر می‌باشد.

هر داده بر اساس همسایگی محلی تعریف شده‌اش و دادگان موجود در این همسایگی، تابع شایستگی خود را بر اساس رابطه (۲) محاسبه می‌نماید. این تابع شایستگی متناظر با میزان مشابهت داده مورد بررسی با دادگان موجود در همسایگیش می‌باشد.

$$f(agent_i) = \max\{0, \frac{1}{(2s_x + 1) \cdot (2s_y + 1)} \sum_{agent_j \in N(agent_i)} (1 - \frac{D(agent_i, agent_j)}{\alpha_i})\} \quad (2)$$

در رابطه فوق  $s_x$  و  $s_y$  ابعاد همسایگی،  $N(agent_i)$  همسایگان عامل  $i$ ،  $D(agent_i, agent_j)$  فاصله بین دو عامل و  $\alpha_i$  میانگین فاصله عامل  $i$  از بقیه عامل‌ها می‌باشد. از تابع شایستگی برای محاسبه احتمالی به نام احتمال فعالیت بر اساس رابطه (۳)، استفاده می‌شود. رابطه (۳) بیانگر میزان احتمال سکون یک عامل در مکان فعلی خود و یا حرکت عامل به همسایگی‌های مجاورش می‌باشد.

$$P_a(i) = \frac{\beta^\lambda}{\beta^\lambda + f(agent_i)^\lambda} \quad (3)$$

در رابطه فوق  $\beta = 0.1$  و  $\lambda = 2$  در نظر گرفته شده است. محاسبه تابع شایستگی و احتمال فعالیت و اعمال آن به هر عامل تا تعداد تکرار مشخصی صورت می‌پذیرد. با گذشت زمان عامل‌های مشابه گرد هم جمع شده و تشکیل خوشه می‌دهند [۶]. در نسخه بهبود یافته این الگوریتم  $\lambda$  و  $\alpha_i$  به صورت تطبیقی محاسبه می‌شوند. همچنین به جای حرکت تصادفی عامل‌ها در محیط سلولی از الگوریتم حریصانه برای یافتن جهت مناسب حرکت استفاده شده است.

## ۳. اتوماتاهای یادگیر، اتوماتای یادگیر سلولی و اتوماتای یادگیر سلولی متحرک

در این بخش پس از شرح مختصری درباره اتوماتاهای یادگیر و اتوماتای یادگیر سلولی، اتوماتای یادگیر سلولی متحرک که نسخه جدیدی از اتوماتای یادگیر سلولی می‌باشد، معرفی می‌گردد.



### ۱.۳ اتوماتاهای یادگیر (LA)

اتوماتای یادگیر ماشینی با توانایی انجام تعداد محدودی عمل است. در هر مرحله اتوماتای یادگیر یک عمل از مجموعه اعمال خود را انتخاب می‌نماید. عمل انتخابی توسط محیط ارزیابی شده و اتوماتا از پاسخ محیط برای بروزرسانی ساختار داخلی خود و انتخاب عمل بعدی استفاده می‌نماید. در طی این فرآیند اتوماتا یاد گرفته که از بین اعمال خود بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند [10]. یک اتوماتای یادگیر از دو قسمت اصلی تشکیل شده است:

- یک اتوماتای با تعداد محدودی عمل و یک محیط که اتوماتا با آن در ارتباط است.

- الگوریتم یادگیری که اتوماتا با استفاده از آن اقدام بهینه را یاد می‌گیرد.

یک اتوماتای بصورت پنجمتایی  $SA \equiv \{\alpha, \beta, F, G, \phi\}$  تعریف می‌شود که در آن:

$$1. \quad \alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\} \quad \text{عملهای اتوماتا}$$

$$2. \quad \beta \equiv \{\beta_1, \beta_2, \dots, \beta_r\} \quad \text{مجموعه ورودی‌های اتوماتا}$$

$$3. \quad F \equiv \phi \times \beta \rightarrow \phi \quad \text{تابع تولید وضعیت جدید اتوماتا}$$

$$4. \quad G \equiv \phi \rightarrow \alpha \quad \text{تابع خروجی که وضعیت فعلی را به خروجی بعدی نگاشت می‌کند.}$$

$$5. \quad n \quad \phi(n) \equiv \{\phi_1, \phi_2, \dots, \phi_k\} \quad \text{مجموعه وضعیت‌های داخلی اتوماتا در لحظه } n$$

مجموعه  $\alpha$  شامل اعمال (خروجی‌های) اتوماتا است که اتوماتا در هر گام، یک عمل از  $r$  عمل این مجموعه را انتخاب می‌نماید.  $\beta$  مجموعه ورودی‌های اتوماتا را مشخص می‌کند. تابع  $F$  و  $G$  وضعیت فعلی ورودی را به خروجی بعدی (عمل بعدی) اتوماتا نگاشت می‌کنند. اگر نگاشتهای  $F$  و  $G$  قطعی باشند، اتوماتا، اتوماتای قطعی<sup>۷</sup> نامیده می‌شود و در صورتیکه  $F$  و  $G$  تصادفی باشند، اتوماتا، اتوماتای تصادفی نامیده می‌شود.

اتوماتاهای یادگیر به دو گروه اتوماتای با ساختار ثابت<sup>۸</sup> و اتوماتای با ساختار متغیر<sup>۹</sup> تقسیم بندی می‌گردند. در اتوماتای یادگیر با ساختار ثابت، احتمال عملهای اتوماتا ثابت هستند. در حالیکه در اتوماتای یادگیر با ساختار متغیر احتمال عملهای اتوماتا در هر تکرار بروز می‌شوند. در اتوماتای یادگیر با ساختار متغیر، تغییر احتمال‌های اعمال بر اساس الگوریتم یادگیری انجام می‌شود. همچنین وضعیت داخلی  $\phi$  اتوماتا توسط احتمال عملهای اتوماتا بازنمایی می‌شوند. وضعیت داخلی  $\phi(n)$  اتوماتا در لحظه  $n$  با بردار احتمال اقدام‌های اتوماتا  $P(n)$  که در رابطه (۴) آمده است، نشان داده می‌شود.

$$\sum_{i=1}^r p_i(n) = 1, \quad \forall n \quad , \quad p_i(n) = \text{Prob}[\alpha(n) = \alpha_i] \quad P(n) \equiv \{p_1(n), p_2(n), \dots, p_r(n)\} \quad (4)$$

محیط را می‌توان توسط سه‌تایی  $E \equiv \{\alpha, \beta, c\}$  نشان داد که در آن  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه ورودی‌های محیط،  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_r\}$  مجموعه خروجی‌های محیط و  $c \equiv \{c_1, c_2, \dots, c_r\}$  مجموعه احتمال‌های جریمه می‌باشد. ورودی محیط یکی از  $r$  عمل انتخاب شده توسط اتوماتای یادگیر است. خروجی (پاسخ) محیط برای هر عمل  $i$  توسط  $\beta_i$  مشخص می‌شود. اگر  $\beta_i$  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  $Q$  شامل تعداد محدودی از مقادیر موجود در بازه  $[0, 1]$  می‌باشد. در محیط نوع  $S$  مقادیر  $\beta_i(n)$  یک متغیر تصادفی در بازه  $[0, 1]$  می‌باشد. مجموعه  $c$  احتمالات جریمه (شکست) پاسخ‌های محیط را مشخص می‌کند و همانند رابطه (۵) تعریف می‌شود.  $c_i$  احتمال اینکه اقدام  $i$  پاسخ نامطلوبی از محیط دریافت کند را نشان می‌دهد. در محیط‌های ایستا<sup>۱۰</sup>، مقادیر احتمال جریمه‌ها ( $c_i$ ) ثابت هستند. در حالیکه در محیط‌های غیر ایستا احتمالات جریمه در طول زمان تغییر می‌کند.

$$c_i = \text{Prob}\{\beta(n) = 1 | \alpha(n) = \alpha_i\}, \quad i = \{1, 2, \dots, r\} \quad (5)$$

ارتباط اتوماتای تصادفی با محیط در شکل (۱) نشان داده شده است. از این مجموعه به همراه الگوریتم یادگیری تحت عنوان اتوماتای یادگیر تصادفی<sup>۱۱</sup> نام برده می‌شود. اتوماتای یادگیر تصادفی را می‌توان با چهارتایی  $LA \equiv \{\alpha, \beta, p, T\}$  نشان داد که  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه عملهای اتوماتا و  $r$  تعداد عملهای اتوماتا،  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_r\}$  مجموعه ورودی‌های اتوماتا،  $p \equiv \{p_1, p_2, \dots, p_r\}$  بردار احتمال عملهای اتوماتا و  $T \equiv \{\phi_1, \phi_2, \dots, \phi_k\}$  الگوریتم یادگیری می‌باشد. الگوریتم‌های یادگیری تأثیری زیادی بر عملکرد اتوماتای یادگیر دارند. اگر  $T$  یک عملگر خطی باشد، الگوریتم یادگیری، خطی نامیده می‌شود. در غیر اینصورت الگوریتم یادگیری غیرخطی نامیده می‌شود.





شکل(۱) اتوماتای یادگیر تصادفی

ایده اصلی تمام الگوریتم‌های یادگیری به این صورت است که: اگر اتوماتای یادگیر در تکرار  $n$ -ام، یک عمل خود مانند  $i$  را انتخاب کند و یک پاسخ مطلوب از محیط دریافت نماید،  $p_i(n)$  (احتمال عمل  $i$ ) افزایش و احتمال سایر عمل‌ها کاهش می‌یابد. بالعکس در صورت نامطلوب بودن پاسخ دریافتی از محیط، احتمال عمل  $i$  کاهش و احتمال سایر عمل‌های اتومات افزایش می‌یابد. در هر حال، تغییرات به گونه‌ای صورت می‌گیرد تا حاصل جمع  $p_i(n)$  ها همواره ثابت و مساوی یک باقی بماند. الگوریتم افزایش می‌شده در روابط (۶) و (۷) بیانگر نمونه‌ای از الگوریتم خطی یادگیری در محیط از نوع  $S$  می‌باشد.

$$p_i(n+1) = p_i(n) - \beta(n).b.p_i(n) + (1 - \beta(n)).a.[1 - p_i(n)] \quad (6)$$

$$p_j(n+1) = p_j(n) + \beta(n).\left(\frac{b}{r-1} + -b.p_j(n)\right) - (1 - \beta(n)).a.(1 - p_j(n)) \quad \forall j, \quad j \neq i \quad (7)$$

در روابط فوق  $r$  تعداد اقدام‌های اتومات،  $(n)$  سیگнал تقویتی تولیدی محیط در بازه  $[1, r]$ ،  $a$  پارامتر پاداش و  $b$  پارامتر جریمه می‌باشد. در محیط از نوع  $S$  هر عمل بر مبنای میزان  $(n)$   $\beta$ ، توأمان پاداش و جریمه دریافت می‌نماید. با توجه به مقادیر  $a$  و  $b$  در روابط فوق، سه الگوریتم مشهور زیر تعریف می‌شوند:

اتوماتای یادگیر  $L_{RP}$ <sup>۱۲</sup>: مقادیر  $a$  و  $b$  برابر می‌باشند.

اتوماتای یادگیر  $L_{RI}$ <sup>۱۳</sup>:  $b$  مساوی با صفر است.

اتوماتای یادگیر  $L_{RP}$ <sup>۱۴</sup>:  $b << a$  است.

## ۲.۳. اتوماتاهای یادگیر سلوالی (CLA)

اتوماتای یادگیر سلوالی مجموعه‌ای متشکل از اجزاء ساده بوده که رفتار هر جزء بر اساس تجربیات گذشته و رفتار همسایگانش تعیین و اصلاح می‌شود [8]. اجزاء ساده تشکیل دهنده این مدل، از طریق تعامل با یکدیگر رفتار پیچیده‌ای از خود نشان می‌دهند. هر اتوماتای یادگیر سلوالی، از یک اتوماتای سلوالی تشکیل شده است که هر سلوول آن به یک یا چند اتوماتای یادگیر مجهز می‌باشد. همانند اتوماتای سلوالی، قانون محلی در محیط حاکم بوده، و این قانون تعیین می‌کند که عمل انتخاب شده توسط یک اتوماتا در سلوول باید پاداش داده شود و یا اینکه جریمه دریافت نماید. دادن پاداش و یا جریمه، باعث بروز شدن ساختار اتوماتای یادگیر سلوالی به منظور نیل به یک هدف مشخص می‌گردد. ایده اصلی اتوماتای یادگیر سلوالی، که زیر مجموعه‌ای از اتوماتاهای یادگیر سلوالی تصادفی محسوب می‌شود، استفاده از اتوماتاهای یادگیر برای محاسبه احتمال انتقال حالت در اتوماتای سلوالی می‌باشد. اتوماتاهای یادگیر سلوالی را می‌توان به دو دسته آسنکرون و سنکرون تقسیم کرد. در مدل سنکرون تمام سلوول‌ها با یک ساعت سراسری هماهنگ شده و به طور همزمان اجرا می‌شوند. اتوماتای یادگیر سلوالی  $d$ -بعدی یک چندتایی ( $CLA = (Z^d, \phi, A, N, F)$  است به طوری که:

$Z^d$  شبکه‌ای از  $d$ -تاپی‌های مرتب از اعداد صحیح می‌باشد. این شبکه می‌تواند متناهی، نیمه متناهی یا نامتناهی باشد.

$\phi$  یک مجموعه متناهی از حالت‌ها می‌باشد.

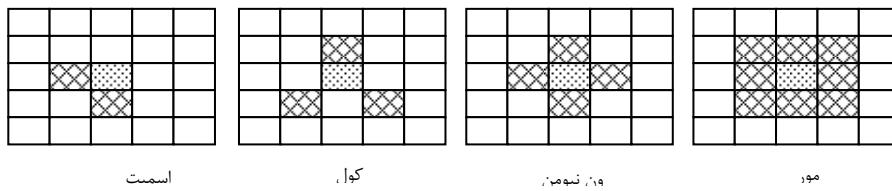
$A$  یک مجموعه از اتوماتاهای یادگیر (LA) است که هر اتوماتای یادگیر به یک سلوول نسبت داده می‌شود.

$N = \{\bar{x}_1, \dots, \bar{x}_{\bar{m}}\}$  یک زیر مجموعه متناهی از  $Z^d$  می‌باشد که بردار همسایگی خوانده می‌شود.

$F: \underline{\phi}^{\bar{m}} \rightarrow \underline{\beta}$  قانون محلی اتوماتای یادگیر سلوالی می‌باشد به طوریکه  $\underline{\beta}$  مجموعه مقادیری است که می‌تواند به عنوان سیگنال تقویتی (سیگنال خروجی محیط) پذیرفته شود.

در اتوماتای یادگیر سلوالی می‌توان از ساختارهای مختلفی برای همسایگی استفاده نمود. در حالت کلی هر مجموعه مرتب از سلوول‌ها را می‌توان به عنوان همسایه در نظر گرفت؛ اما معمولترین آنها همسایگی اسمیت، کول، ون نیومون و مور می‌باشد که به نزدیکترین همسایگان مشهور می‌باشند. این همسایگی‌ها در شکل (۲) نشان داده شده‌اند.





شکل (۲) همسایگی اسمیت، کول، ون نیومن و مور

عملکرد اتماتای یادگیر سلوی را می‌توان به شرح زیر بیان نمود. در هر لحظه هر اتماتای یادگیر سلوی یک عمل از مجموعه اعمال خود را انتخاب می‌کند. این عمل می‌تواند بر اساس مشاهدات قبلی و یا به صورت تصادفی انتخاب شود. عمل انتخاب شده با توجه به اعمال انتخاب شده توسط سلوی‌های همسایه و قانون حاکم بر اتماتای یادگیر سلوی پاداش و یا جریمه داده می‌شود. با توجه به اینکه عمل انتخاب شده پاداش گرفته و یا جریمه شده است، اتماتاً رفتار خود را تصحیح کرده و ساختار داخلی اتماتاً بهنگام می‌گردد. بعد از بروزرسانی، هر اتماتای یادگیر در اتماتای یادگیر سلوی 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. عمل بهنگام‌سازی ساختار اتماتاهای موجود در اتماتای یادگیر سلوی توسط الگوریتم یادگیری انجام می‌شود.

### ۲.۳. اتماتاهای یادگیر سلوی متحرک (M-CLA)

اتماتای یادگیر سلوی متحرک زیر مجموعه‌ای از اتماتاهای یادگیر سلوی بوده که دارای مشخصات زیر می‌باشد:

- اتماتاهای یادگیر مستقل از سلوی‌ها می‌باشند.
- تعداد اتماتاهای یادگیر می‌تواند کمتر از سلوی‌های موجود در شبکه باشد.
- در هر سلوی امکان وجود بیش از یک اتماتای یادگیر وجود دارد.
- بروزرسانی ساختار داخلی اتماتاهای یادگیر بر مبنای اتماتاهای یادگیر موجود در همسایگی سلوی می‌باشد.
- هر اتماتای یادگیر، امکان جایجایی از یک سلوی به سلوی دیگر دارد.
- بروزرسانی تمامی سلوی‌ها به صورت همزمان انجام می‌باشد.
- هر اتماتای یادگیر همانند اتماتاهای یادگیر سلوی تصادفی باز [9] می‌تواند پارامتر سراسری دریافت نماید.

در اتماتای یادگیر سلوی متحرک اتماتاهای یادگیر مستقل از سلوی‌ها بوده و سلوی که شامل یک یا چند اتماتای یادگیر باشد، سلوی فعال نامیده می‌شود. همانند اتماتای یادگیر سلوی، ساختار سلوی و قانون محلی حاکم در اتماتای یادگیر سلوی متحرک، مؤثر در بروزرسانی ساختار داخلی اتماتاهای یادگیر موجود در سلوی‌های فعال می‌باشد. تنها سلوی‌های فعال موجود در همسایگی یک سلوی فعال در روند یادگیری عمل بهینه هر اتماتای یادگیر مؤثر می‌باشند.

باتوجه به نکات بیان شده عملکرد اتماتای یادگیر سلوی متحرک را می‌توان به شرح زیر بیان نمود. در هر لحظه هر اتماتای یادگیر در اتماتای یادگیر سلوی یک عمل از مجموعه اعمال خود را بر اساس بردار احتمال اعمال انتخاب می‌نماید. عمل انتخاب شده باعث جایجایی اتماتای یادگیر از یک سلوی به سلوی در همسایگیش می‌شود. در چنین صورتی اگر سلوی قبلی که جایگاه اتماتای یادگیر بوده شامل هیچ اتماتای یادگیری نباید، غیر فعال می‌شود. همچنین سلوی جدید که جایگاه فعلی اتماتای یادگیر می‌باشد در صورت غیر فعال بودن به صورت فعل تبدیل می‌شود. اعمال اتماتای یادگیر موجود در هر سلوی با توجه به اعمال اتماتاهای یادگیر موجود در سلوی‌های فعل همسایه و قانون محلی حاکم پاداش و یا جریمه دریافت می‌نمایند. پاداش و جریمه دریافتی موجب بروزرسانی ساختار داخلی اتماتای یادگیر موجود در سلوی فعل می‌شود. فرآیند دادن پاداش و جریمه تا زمانی که سیستم به حالت پایدار برسد و یا یک معیار از قبل تعریف شده‌ای برقرار شود، ادامه می‌یابد. عمل بهنگام‌سازی ساختار اتماتاهای موجود در اتماتای یادگیر سلوی متحرک توسط الگوریتم یادگیری و به صورت همزمان انجام می‌شود.

### ۴. الگوریتم خوشبندی با استفاده از مدل اتماتای یادگیر سلوی متحرک

پیش از بررسی الگوریتم خوشبندی با استفاده از اتماتای یادگیر سلوی متحرک چند اصطلاح زیر تعریف می‌شود:

- عامل: به ازای هر داده  $n$  بعدی یک اتماتای یادگیر در نظر گرفته شده که عامل نامیده می‌شود.
- جهت حرکت: همسایگی‌های مجاز هر عامل برای حرکت، جهت حرکت نامیده می‌شود.



- **همسوبی:** دو عامل متشابه تمایل به حرکت در یک راستای مشخص را دارند که این تشابه راستای حرکت همسوبی نامیده می‌شود.
- **غیرهمسوبی:** دو عامل غیر متشابه تمایل به حرکت در جهات مختلف دارند که این عدم تشابه راستای حرکت غیر همسوبی نامیده می‌شود.

برای خوشبندی دادگان، هر عامل بر اساس بردار احتمال جهات حرکتش به جهتی حرکت می‌نماید. عامل اگر عامل دیگری را در همسایگی مسیر حرکت خود مشاهده نماید، بر اساس میزان مشابهت با عامل همسایه، جهت حرکت خود را تغییر می‌دهد. اگر تشابه با عامل همسایه زیاد باشد، عامل مورد بررسی جهت حرکت خود را به جهت حرکت عامل مشابه نزدیک می‌نماید؛ به بیانی عامل مورد بررسی یاد می‌گیرد که جهت صحیح حرکت برای یافتن عامل‌های مشابه، همسوبی با جهت حرکت عامل مشابه موجود در همسایگی اش می‌باشد. همچنین اگر شابهای بین عامل مورد بررسی و عامل موجود در همسایگی کم باشد، عامل مورد بررسی جهت حرکت خود را تغییر می‌دهد؛ به بیانی یاد می‌گیرد که جهت حرکت عامل غیرمشابه جهت مناسبی برای یافتن عامل‌های مشابه خود نیست. همسوبی و غیرهمسوبی عامل برای یافتن عامل‌های مشابه باعث یادگیری جهت حرکت مطلوب بر اساس اطلاعات محلی می‌شود. مشابه و یا غیر مشابه بودن یک عامل با عامل‌های همسایه‌اش از رابطه تشابه بدست می‌آید.

در هر مرحله اگر عامل مورد نظر در همسایگی خود عاملی را مشاهده نماید، از جهت حرکت عامل همسایه برای بهبود جهت حرکت خود استفاده می‌نماید. یادگیری هر عامل بر اساس تشابه و یا عدم تشابه با عامل‌های همسایه به صورت زیر انجام می‌شود:

- اگر عامل همسایه غیر مشابه باشد: جهت حرکت پیشنهادی عامل همسایه (جهت حرکت با احتمال بیشتر) در عامل مورد بررسی جریمه دریافت می‌نماید.

- اگر عامل همسایه مشابه باشد: جهت حرکت پیشنهادی عامل همسایه در عامل مورد بررسی پاداش دریافت می‌نماید.

در مدل اتوماتیک یادگیر سلوی متوجه هر اتوماتیک یادگیر معادل یک عامل بوده که در بردارنده یک داده  $n$  بعدی می‌باشد. هر سلوی که شامل یک عامل باشد یک سلوی فعال نامیده شده است. همچنین در هر سلوی امکان حظور بیش از یک عامل وجود دارد که باعث بر طرف شدن مشکل برخورد بین عامل‌ها و بروزرسانی همزمان تمامی سلوی‌ها می‌شود. اتوماتیک سلوی به عنوان محیط حرکت عامل‌ها در نظر گرفته شده است. هر عامل بر اساس جهت‌های تعریف شده در همسایگی اش و همچنین احتمال حرکت جهات مختلف، یک جهت را برای حرکت انتخاب می‌نماید. پس از حرکت، اگر عامل‌ای در مجاورت عامل مورد بررسی وجود داشته باشد، بردار احتمال جهات حرکت عامل بروزرسانی می‌شود. این بروزرسانی بر اساس پاداش و یا جریمه دریافتی حاصل از میزان تشابه عامل مورد بررسی و عامل همسایه آن صورت می‌پذیرد. قوانین زیر برای خوشبندی در نظر گرفته شده است:

١. عامل‌های همسایه هر عامل به صورت مجزا بر روی جهت حرکت عامل مورد بررسی تأثیر گذار می‌باشند.
٢. تشابه بین هر دو عامل بر اساس رابطه فاصله اقلیدسی و یا فاصله کسینوسی محاسبه می‌شود. اگر تشابه دو عامل زیاد باشد، عامل مورد بررسی با دریافت پاداش همسو شده و در صورت تشابه کم، عامل مورد بررسی با دریافت جریمه غیر همسو می‌شود.

## ۵. نتایج پیاده‌سازی خوشبندی

الگوریتم خوشبندی پیشنهادی با استفاده از اتوماتیک یادگیر سلوی متوجه بر روی سه مجموعه داده مورد آزمایش قرار گرفته است. در الگوریتم پیشنهاد شده از یادگیری مدل S با ظایاب یادگیری  $b = 1e - 1$  و  $a = 1e - 5$  استفاده شده است. همچنین تعداد سلوی مورد استفاده در الگوریتم پیشنهادی برابر  $(\sqrt{N} * \sqrt{N})$  بوده که  $N$  بیانگر تعداد داده‌های هر دادگان می‌باشد. نتایج حاصل با نتایج حاصل از دو الگوریتم K-means و ASM مقایسه شده‌اند. دادگان اول، شامل ۲۰۰ داده دو بعدی که توسط توزیع نرمال  $(\mu, \sigma^2)$  تولید شده و متعلق به چهار خوشه با مشخصات  $\{N(0.2, 0.1^2), N(0.2, 0.1^2), N(0.8, 0.1^2), N(0.8, 0.1^2)\}$  می‌باشد. دادگان دوم دادگان Iris بوده که شامل ۱۵۰ داده چهار بعدی متعلق به سه خوشه ۵۰ داده‌ای می‌باشد. در دادگان Iris، خوشه اول به صورت خطی از دو خوشه دیگر جدا شده و دو خوشه دیگر به صورت غیر خطی از یکدیگر جدا می‌شوند. خوشه‌های دوم و سوم امتزاج داده‌ای زیادی دارند. و در نهایت دادگان سوم، دادگان Wine که شامل ۱۷۸ داده سیزده بعدی می‌باشد. این دادگان شامل سه کلاس ۰.۵۹، ۰.۷۰ و ۰.۳۹ داده‌ای بوده که کلاس دوم با کلاس اول و سوم، دادگان هم مرز دارد. نتایج بدست آمده حاصل از ۱۰۰ بار اجرای هر یک از سه الگوریتم می‌باشد.



در جدول (۱) نتایج حاصل از مقایسه خطاکاری الگوریتم‌ها بر روی مجموعه دادگان اول نشان داده شده است. در الگوریتم معرفی شده جهات حرکت از همسایگی و نیومن با شعاع یک استفاده می‌نماید. شرط توقف الگوریتم پیشنهادی خوشبندی تعداد تکرار ۳۰۰ در نظر گرفته شده است. چنانچه در نتایج جدول (۱) مشاهده می‌شود، الگوریتم K-means تنها در ۶۰٪ موارد دادگان مجموعه اول را به چهار خوشبندی تقسیم نموده و در ۴۰٪ موارد تعداد خوشبندی‌های بدست آمده در این الگوریتم کمتر از چهار خوشبندی می‌باشد. در ۶۰٪ خوشبندی صحیح الگوریتم K-means کمینه و بیشینه در صد خطای صفر می‌باشد. همچنین الگوریتم ASM با شرط توقف ۱۰۰۰۰ تکرار، تنها در ۹۰٪ موارد دادگان را به چهار خوشبندی تقسیم می‌نماید. بر اساس نتایج جدول (۱) الگوریتم پیشنهادی مجموعه دادگان لول را بهتر از دو الگوریتم دیگر خوشبندی می‌نماید.

جدول (۱) مقایسه خطاکاری الگوریتم خوشبندی بر روی مجموعه دادگان اول

| الگوریتم | در صد کمینه خطای خطا | در صد بیشینه خطای خطا | در صد میانگین خطای خطا | در صد درستی خوشبندی |
|----------|----------------------|-----------------------|------------------------|---------------------|
| K-means  | ٪۰                   | ٪۰                    | ٪۰                     | ٪۶۰                 |
| ASM      | ٪۰                   | ٪۱۵                   | ٪۰۵                    | ٪۹۰                 |
| M-CLA    | ٪۰                   | ٪۰                    | ٪۰                     | ٪۱۰۰                |

در جدول (۲) و جدول (۳) به ترتیب نتایج حاصل از خوشبندی دو دادگان Iris و Wine نشان داده شده است. الگوریتم معرفی شده در خوشبندی این دو دادگان از همسایگی کول استفاده نموده است. خطای حاصل از الگوریتم معرفی شده با شرط توقف ۱۵۰۰ تکرار در مجموعه دادگان Iris به طور میانگین در حدود ٪۲۳۶ می‌باشد. الگوریتم K-means در ٪۷۵ موارد خطای ٪۴ داشته و در ٪۲۵ موارد دادگان را به دو خوشبندی تقسیم می‌نماید. خطای الگوریتم ASM با شرط توقف ۵۰۰۰ تکرار ٪۲۱۳ می‌باشد.

جدول (۲) مقایسه خطاکاری الگوریتم خوشبندی بر روی مجموعه دادگان Iris

| الگوریتم | در صد کمینه خطای خطا | در صد بیشینه خطای خطا | در صد میانگین خطای خطا | در صد درستی خوشبندی |
|----------|----------------------|-----------------------|------------------------|---------------------|
| K-means  | ٪۴                   | ٪۴                    | ٪۴                     | ٪۴                  |
| ASM      | ٪۱۳                  | ٪۳۳                   | ٪۱۳                    | ٪۲۱۳                |
| M-CLA    | ٪۱۳                  | ٪۴                    | ٪۱۳                    | ٪۲۳۶                |

الگوریتم معرفی شده با شرط توقف ۱۵۰۰ تکرار دادگان Wine را با ٪۷۹ به ۳ خوشبندی تقسیم نموده است. این الگوریتم در ٪۳ موارد دادگان را به دو خوشبندی تقسیم می‌نماید. خطای الگوریتم K-means در خوشبندی دادگان Wine ٪۸۴۳ و خطای الگوریتم ASM با شرط توقف ۵۰۰۰ در ٪۹۱ موارد ٪۱۳۶ بوده و در ٪۹ موارد این دادگان را به دو خوشبندی تقسیم می‌نماید.

جدول (۳) میانگین مقایسه الگوریتم خوشبندی بر روی مجموعه دادگان Wine

| الگوریتم | در صد کمینه خطای خطا | در صد بیشینه خطای خطا | در صد میانگین خطای خطا | در صد درستی خوشبندی |
|----------|----------------------|-----------------------|------------------------|---------------------|
| K-means  | ٪۸۴۳                 | ٪۸۴۳                  | ٪۸۴۳                   | ٪۱۰۰                |
| ASM      | ٪۶۴۷                 | ٪۱۴۰۱                 | ٪۱۲۳                   | ٪۹۱                 |
| M-CLA    | ٪۵۶                  | ٪۱۰۶۷                 | ٪۷۷۹۰                  | ٪۹۷                 |

## ۶. نتیجه‌گیری

در این مقاله یک نسخه جدید از اتماتای یادگیر سلولی معرفی و سپس یک کاربرد از آن در خوشبندی ارائه گردید. به منظور ارزیابی، الگوریتم خوشبندی پیشنهادی بر روی تعدادی دادگان استاندارد آزمایش و نتایج بدست آمده با نتایج حاصله برای الگوریتم‌های ASM و K-means مقایسه شد.

الگوریتم پیشنهادی در تعداد تکرار کمتر نسبت به الگوریتم ASM دادگان را خوشبندی می‌نماید. همچنین در مقایسه با دو الگوریتم K-means و ASM، الگوریتم معرفی شده در تعداد دفعات بیشتری دادگان را به تعداد خوشبندی صحیح دسته‌بندی می‌نماید. میزان خطای حاصل از الگوریتم پیشنهادی در مقایسه با خطای دو الگوریتم دیگر در دادگان تولید شده از توزیع نرمال و دادگان Wine، K-means می‌باشد. در دادگان Iris خطای الگوریتم پیشنهادی نسبت به الگوریتم‌های ASM بیشتر و نسبت به الگوریتم K-means کمتر می‌باشد.



## مراجع

- [1] Han, J., Kamber, M. and Tung, K. H.; “*Spatial Clustering Methods in Data Mining: A Survey*”, Geographic Data Mining and Knowledge Discovery, pp. 1–29, 2001.
- [2] Wang, W., Yang, J. and Muntz, R.; “*STING: A Statistical Grid Approach to Spatial Data Mining*”, Int. Conf. Very large Data Basses (VLDB’97), pp. 186-195, Aug. 1997.
- [3] Sheikholeslami, G., Chaterjee, S. and Zhang, A.; “*Wave Cluster: A Multi Resolution Clustering Approach for Very Large Spatial Databases*”, Int. Conf. Very large Data Basses (VLDB’98), pp. 428-439, Aug. 1998.
- [4] Agrawal, R., Gehrke, J., Gunopulos, D. and Raghavan, P.; “*Automatic Subspace Clustering of High Dimensional Data for Data Mining Application*”, ACM SIGMOD Int. Conf. Management of Data (SIGMOD’98), pp. 94-105, June 1998.
- [5] Wolfram, S.; “*Cellular Automata*”, Los Alamos Science, vol. 9, pp. 2-21, fall 1983.
- [6] Chen, L., Xu, X., Chen, Y. and He, P.; “*A Novel Ant Clustering Algorithm Based on Cellular Automata*”, Intelligent Agent Technology (IAT 2004), pp.148-154, 2004.
- [7] Vande Moere and A., Clayden, J.; “*Cellular Ants: Combining Ant-Based Clustering with Cellular Automata*”, IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI’05), pp. 177-184, Hong Kong, Nov. 2005.
- [8] Beigy, H. and Meybodi, M.R.; “*A Mathematical Framework for Cellular Learning Automata*”, Advances in Complex Systems, Vol. 7, Nos. 3-4, pp. 295-320, September/December 2004.
- [9] Beigy, H., Meybodi, M.R.; “*Open Synchronous Cellular Learning Automata*”, Proceedings of the 8th world Multi-conference on Systemics- Cybernetics and Informatics(SCI2004), pp. 9-15, USA, July 2004.
- [10] Thathachar, M.A.L. and Sastry, P.S.; “*Varieties of Learning Automata: An Overview*”, IEEE Transaction on Systems, and Cybernetics-Part B: Cybernetics Vol. 32, No. 6, pp. 711-722, 2002.

<sup>1</sup> Hierarchical

<sup>2</sup> Partitioning

<sup>3</sup> Grid

<sup>4</sup> Swarm intelligence

<sup>5</sup> Cellular Automata (CA)

<sup>6</sup> Learning Automata (LA)

<sup>7</sup> Deterministic Automata

<sup>8</sup> Fixed Structure Automata

<sup>9</sup> Variable Structure Automata

<sup>10</sup> Stationary

<sup>11</sup> Stochastic Learning Automata

<sup>12</sup> Linear Reward Penalty

<sup>13</sup> Linear Reward Inaction

<sup>14</sup> Linear Reward Epsilon Penalty

