



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده ریاضی و علوم کامپیوتر

تمرین کارشناسی
هوش مصنوعی و کارگاه

گزارش ۷: درک و توضیح مقاله swarm intelligence for self-organized clustering

نگارش
سارینا حشمتی
۴۰۰۱۳۰۴۳

استاد ۱
دکتر مهدی قطعی

استاد ۲
دکتر بهنام یوسفی مهر

بهمن ۱۴۰۲

چکیده

در این گزارش به بررسی یک نوع الگوریتم به نام "Databionic Swarm" (DBS) میپردازیم که برای خوشه‌بندی داده‌ها استفاده می‌شود. خوشه‌بندی یک تکنیک در تحلیل داده است که از آن برای گروه‌بندی نقاط مشابه با یکدیگر استفاده می‌شود. این الگوریتم بر اساس ایده‌ای به نام هوش ازدحامی ساخته شده است که از رفتار گروه‌های حیوانات و حشرات الهام گرفته شده است که با همکاری به‌منظور دستیابی به یک هدف مشترک فعالیت می‌کنند.

در DBS، الگوریتم شامل یک جمعیت از "عامل‌ها" است که با یکدیگر تعامل و محیط خود را حس می‌کنند. این عامل‌ها با همکاری در پیدا کردن الگوها در داده‌های بعد بالا که می‌توانند به ساختارهای مبتنی بر فاصله یا چگالی توصیف شوند، فعالیت می‌کنند. الگوریتم بر اساس یک تابع هدف پیش‌تعیین‌شده (معیاری برای اندازه‌گیری خوبی خوشه‌های شکل‌گرفته) عمل نمی‌کند، بلکه در فرایند annealing به دنبال تعادلش می‌گردد، که وضعیتی است که هیچ عاملی نمی‌تواند با تغییر استراتژی خود وضعیت خود و دیگران را بهبود ببخشد.

نویسندگان ادعا می‌کنند که DBS می‌تواند عملکرد بهتری نسبت به روش‌های خوشه‌بندی معمولاً مانند PAM، K-means، خوشه‌بندی طیفی و ... داشته باشد، به ویژه زمانی که اطلاعات قبلی درباره داده‌ها موجود نباشد.

یکی از چالش‌های خوشه‌بندی تعیین تعداد صحیح خوشه‌ها در داده است DBS. این مسئله را با استفاده از یک تکنیک مصورسازی به نام نقشه‌ی توپوگرافیک (topographic map) مورد بررسی قرار می‌دهد که به تعداد خوشه‌ها کمک می‌کند. این تکنیک قادر است شرایطی را شناسایی کند که خوشه‌بندی در داده‌های ورودی معنادار نباشد، این حالت زمانی اتفاق می‌افتد که داده‌ها شامل خوشه‌های (طبیعی) واضحی نباشد.

فهرست مطالب

صفحه

ا	چکیده
1	فصل اول مقدمه مقدمه
4	فصل دوم بررسی ویژگی‌های سیستم‌های مبتنی بر رفتار در یادگیری ماشین بدون ناظر
5	معرفی سیستم‌ها مبتنی بر رفتار در یادگیری ماشین بدون ناظر
5	PSO 2-1-
5	Self-organization 2-2-
7	فصل سوم حلقه‌ی گمشده: game theory و emergence
8	Emergence 2-3-
9	Game theory 2-3-
10	فصل چهارم Databionic swarm
11	DBS 2-4- و هدف آن
11	Pswarm 2-5-
14	The pseudo code 2-1-
15	نتیجه گیری
15	منابع و مراجع

فصل اول

مقدمه

مقدمه

در این گزارش به بررسی مقاله‌ی swarm intelligence for self-organized clustering میپردازیم. در این مقاله به معرفی یک سیستم ازدحام (swarm system) پرداخته میشود که از آن میتوان نوعی روش خوشه بندی استخراج کرد که ویژگی‌های منحصر به فردی دارد و به نسبت سایر روش‌های خوشه بندی برتری‌هایی دارد و محدودیت‌های آنها را میتواند به گونه‌ای جبران کند.

به طور کلی عملیات خوشه بندی به این اشاره دارد که داده‌هایی که با هم شباهت دارند در یک خوشه قرار بگیرند و داده‌های متفاوت در خوشه‌های مختلف باشند. گفته میشود که خوشه بندی مناسب باید سه ویژگی داشته باشد و مطالعات نشان میدهد که هیچ کدام از روش‌های خوشه بندی توانایی انجام عملیات خوشه بندی را با در نظر گرفتن این ویژگی ندارد.

در اینجا مفهوم خوشه‌های طبیعی (natural clusters) مطرح میشود که در آن خوشه‌ها براساس فاصله و/یا تراکم شناسایی میشوند و به طور کلی تنها مفهوم کلی برای توضیح آنها همان شباهت داده‌های داخل یک خوشه و تفاوت میان داده‌های خارج خوشه است.

دلیل چالش برانگیز بودن شناسایی این خوشه‌ها توسط الگوریتم‌های خوشه بندی سنتی آن است که این روش‌ها برای انجام این عملیات در تلاش برای بهینه کردن یک objective function هستند که این معادل در نظر گرفتن فرض‌هایی در رابطه با ساختار داده‌های ورودی است و این فرضیات الزاماً درست نیستند پس در نتیجه روند حل به خوبی به نتیجه نمیرسد.

برای حل کردن این مشکل و رد شدن از این محدودیت مقاله مفهوم self-organization و swarm intelligence را مطرح میکند.

هوش ازدحامی (swarm intelligence) به این معناست که تعداد زیادی particle هدف دار را در محیطی با چهارچوب‌های مشخص قرار میدهیم و هر یک از این جزها در تلاش برای رسیدن به هدف خود هستند. برای انجام این کار طبعاً با یکدیگر تعاملاتی خواهند داشت و تعاملات و تغییرات وضعیت جزها به شرایطی کلی‌ای منجر خواهد شد که به صورت خود به خود و صرفاً براساس این تعاملات فردی particleها ایجاد میشود. به این روند self-organization میگویند. همچنین اگر این شرایط کلی نهایی دارای الگوهای معناداری باشد میگوییم که سیستم ازدحامی ما دارای ویژگی emergence میباشد.

از آنجایی که این سیستم معرفی شده از هیچ objective functionی برای رسیدن به شرایط بهینه استفاده نمیکند (و در نتیجه فرض ثابت نشده‌ای را برای داده‌ها در نظر نمیگیرند)، به چالش‌های روش‌های دیگر خوشه بندی که پیشتر ذکر شد برنمیخورد و انگار براساس مفهوم خوشه‌های طبیعی عملیات خوشه بندی را انجام میدهد. لازم به ذکر است که این روش در مرحله‌ای از اجرا نقشه‌ی توپوگرافیکی‌ای برای داده‌ها آماده میکند که باعث میشود بتوانیم براساس آن به صورت دستی هم عملیات خوشه بندی را انجام دهیم. (چرا که به خوبی براساس مفهوم خوشه‌های طبیعی داده‌ها را نمایش میدهد و خوشه‌ها به خوبی قابل درک و استخراج میباشند).

در ادامه به بررسی این روش و روند اجرای الگوریتم مربوط به آن میپردازیم.

فصل دوم

بررسی ویژگی‌های سیستم‌های مبتنی بر رفتار در یادگیری ماشین بدون ناظر

معرفی سیستم‌ها مبتنی بر رفتار در یادگیری ماشین بدون ناظر

در این بخش به بررسی روش‌هایی می‌پردازیم که از هوش ازدحامی (swarm intelligence) برای بررسی و تحلیل اطلاعات استفاده می‌کنند، این روش‌ها از طبیعت الهام گرفته شده‌اند. در ابتدا به بررسی معنی swarm می‌پردازیم تا بتوانیم به طور دقیق آن را تعریف بکنیم، به طور کلی پنج اصل وجود دارد که نشانگر رفتارهای ازدحامی (swarm behavior) است؛ این پنج اصلی عبارتند از:

1. Homogeneity: که به این معناست که تمام agent (particle) ها مدل رفتاری یکسانی دارند.
2. Locality: حرکات هر agent متاثر از حرکات همسایه‌هایش می‌باشد.
3. Velocity matching: هر ذره سعی می‌کند که سرعتش را با سرعت اطرافیانش همخوانی دهد.
4. Collision avoidance: به این معنی که هر ذره از برخورد با ذرات دیگر خودداری می‌کند.
5. Flock centering: به این معنی که هر ذره تلاش می‌کند تا نزدیک همسایه‌هایش باقی بماند.

PSO-1-2

یکی از اصلی‌ترین و کلاسیک‌ترین روش‌های استفاده از swarm intelligence روش PSO می‌باشد که در اکثر موارد دارای ویژگی self-organization می‌باشد ولی نکته‌ای که در مورد این روش وجود دارد آن است که براساس بهینه‌کردن یک objective function است که همانطور که پیشتر گفته شد معادلا این است که در مورد داده‌های ورودی فرض‌های را در نظر بگیریم که الزاما درست نمی‌باشند و همین باعث می‌شود که در برخی موارد به نتایج درستی نرسیم.

Self-organization-2-2

اگر ازدحام به اندازه‌ی کافی عضو داشته باشد آنگاه ممکن است که پدیده‌ی self-organization رخ دهد. این پدیده به این معناست که ازدحام به صورت خود به خودی و بدون هیچ کنترل مرکزی‌ای تشکیل الگوهایی بدهد و به طور کلی بدون هیچ رهبر یا کنترل مرکزی به نظم و ترتیبی برسد.

برای رسیدن به self-organization سیستم باید ۴ ویژگی داشته باشد؛ که عبارتند از:

1. Positive feedback.
2. Negative feedback.
3. Amplification of fluctuation.

Multiple interaction. 4

دو مورد اول به پایدار شدن سیستم و ساختار کمک میکنند.

فصل سوم

حلقه‌ی گمشده: game theory و emergence

حلقه‌ی گمشده: game theory و emergence

در این قسمت به بررسی دو حلقه‌ی گمشده در روش‌های قبل می‌پردازیم و مفاهیم آنها را به صورت مختصر توضیح می‌دهیم. در بخش آینده با به کار گرفتن این دو حلقه‌ی گمشده و ترکیب آن با روش‌ها و مفاهیم قبلی، روش DBS را معرفی می‌کنیم.

Emergence-3-2

اگر الگوهای شکل گرفته در قسمت self-organization که در قسمت قبل به توضیح آن پرداختیم، الگوهای معناداری باشند آنگاه می‌گوییم که سیستم دارای رفتار emergence می‌باشد. وجود یا عدم وجود این ویژگی در سیستم‌ها مبحثی بحث برانگیز است، چرا که ممکن است سیستم دارای آن الگوی معنا دار مورد نظر باشد ولی صرفاً چون معنای آن الگو به عنوان یک «معنی معتبر» در نظر گرفته نمی‌شود، ویژگی emergence برای سیستم تعریف نشود. نکته‌ی جالبی که در مورد این مفهوم وجود دارد آن است که مطالعات نشان می‌دهد که نه تنها سیستم‌های تشکیل شده از جزیهای حمایتگر (cooperative) می‌توانند به emergence ختم می‌شوند، بلکه سیستم‌های تشکیل شده از جزیهای خودخواه (egoistic) نیز امکان رسیدن به این نتیجه را دارند! به طور کلی ۴ فاکتور وجود دارد که باعث می‌شود سیستم به emergence ختم بشود که عبارت هستند از:

1. Randomness.

2. Temporal and structural unpredictability.

3. Multiple non-linear interactions among many agents.

4. Irreducibility.

مورد اول امکان کشف راه‌حل‌های جدید را فراهم می‌کند.

مورد دوم با روش‌های قبلی (مانند PSO که در آن هدف بهینه کردن objective function است) تداخل دارد، چرا که در این روش‌ها به دلیل objective function، فرضیاتی نسبت به ساختار داده در نظر گرفته می‌شود که به نوعی بیاثر می‌کند که ساختار داده تا حدود قابل پیش بینی است.

بحث سوم غیرخطی بودن تعاملات میانه اجزا را بیاثر می‌کند که به این معنا است که تعاملات میانه اجزا به صورت خطی قابل پیش بینی نباشد.

مورد چهارم نیز به این معنا است که ویژگی‌های یک ازدحام (swarm) تنها مخصوص ازدحام است و برای یک جز یا بخشی از اجزای آن برقرار نمی‌باشد.

Game theory-3-2

حلقه‌ی گمشده‌ی دوم مربوط به نظریه‌ی بازی است که چهار اصل self-organization برای آن برقرار است.

نظریه‌ی بازی‌ها مربوط به بازی‌هایی کلی با n بازیکن می‌باشد که تصمیم‌های هر بازیکن علاوه بر نتایج خودش، روی نتایج دیگران نیز تاثیر می‌گذارد. در اینجا مفهومی به نام payoff function معرفی می‌شود که به صورت زیر تعریف می‌شود:

$$p = (p_1, \dots, p_n): \Pi_1 \times \dots \times \Pi_n \rightarrow \mathbb{R}^n$$

در فرمول بالا ورودی‌های تابع payoff یک عضو از تمام مجموعه‌های Π_i (که مجموعه‌ی انتخاب‌های بازیکن i است) انتخاب می‌شود و نتیجه یک بردار n بعدی است که نشان دهنده‌ی نتیجه‌ی (outcome) هر بازیکن می‌باشد.

این انتخاب می‌تواند براساس یک یا چند ویژگی از ویژگی‌های رفتارهای ازدحامی که در فصل دوم به آنها پرداخته شد باشند.

برای بازی‌های non-cooperative (بازی‌هایی که در آن تصمیم‌گیری هر بازیکن به صورت مستقل انجام می‌شود و بازیکن‌های با یکدیگر همکاری (تیمی) ندارند) نش ثابت کرده که تعادلی به نام تعادل نش وجود دارد که در آن هر یک از بازیکن‌ها با تعویض استراتژی خود نمی‌توانند payoff خود یا هیچ یک از بازیکن‌های دیگر را بالا ببرند.

تعادل نشی ضعیف شناخته می‌شود اگر بیش از یک مجموعه استراتژی (انتخاب) برای بازیکن‌ها وجود داشته باشد که payoff مورد نظر را داشته باشد و تعادل نش قوی زمانی اتفاق می‌افتد که تنها یک ترکیب از استراتژی‌ها وجود داشته باشد که بتواند ما را به آن payoff مورد نظر برساند.

فصل چهارم

Databionic swarm (DBS)

Databionic swarm

در این قسمت به بررسی روش ارائه شده توسط مقاله میپردازیم و سعی میکنیم ویژگی‌ها و روند اجرای آن را به صورت کامل توضیح دهیم.

DBS-4-2 و هدف آن

روش Databionic Swarm یک نوع روش یادگیری ماشین بدون ناظر است که اقدام به انجام عملیات خوشه بندی میکند. این روش اینکار را بدون هیچ دانش قبلی از داده های ورودی انجام میدهد و هدف آن این است که رویکردی جایگزین با استفاده از self-organization, swarm intelligence و emergence برای انجام خوشه بندی ارائه کند به طوری که این روش بتواند خود را با داده‌هایی با ابعاد بالا که براساس فاصله و تراکم با یکدیگر ارتباط دارند و بدون استفاده از هیچ global objective function انطباق دهد. این روش به طور مشخص مناسب داده‌هایی است که هیچ اطلاعات قبلی‌ای درمورد داده های ورودی در دسترس نداریم.

روش DBS به طور کلی چند مرحله دارد؛ ابتدا داده‌های ورودی گرفته میشود و با استفاده از الگوریتم Pswarm که در ادامه به توضیح آن میپردازیم به دو بعد کاهش پیدا میکنند. سپس با استفاده از تکنیکی به نام generalized U-matrix داده را براساس فاصله‌های نقاط در حالت چندین بعدی و تراکم آنها در حالت دو بعدی به یک نقشه‌ی سه بعدی تبدیل میکند. در اینجا خوشه بندی انجام شده است و برای انجام کارهای نهایی مانند label گذاری و موارد دیگر میتوان از دخالت‌های انسانی کمک گرفت.

Pswarm-5-2

روش polar swarm یا همان Pswarm یک الگوریتم مورد استفاده در DBS است که در ادامه به بررسی آن میپردازیم.

در این روش، ازدحامی از agentها که به طور مشخص DataBot هستند وجود دارد که در یک مختصات polar موجود هستند. این مختصات به این گونه است که هر نقطه روی آن با دو عدد نمایش داده میشود، r و ϕ . اولی نشان دهنده‌ی فاصله‌ی نقطه تا مرکز مختصات است و دومی نشان دهنده‌ی زاویه ای است که خط اتصال دهنده‌ی مرکز مختصات و نقطه با خط مرجع میسازد.

هر databot یک scent (بو) مخصوص به خود دارد که توسط سایر دیتاباتها قابل شناسایی است و هر کدام از این دیتاباتها یک بردار n بعدی مخصوص به خود دارند که نشان دهنده‌ی فاصله‌ی تمام دیتاباتها با آن میباشد.

به هر ازدحامی از این دیتاباتها در این نوع مختصات polar swarm میگوییم. روش مبتنی بر pswarm شامل سه ایده ی اصلی swarm intelligence و self-organization همراه با non-cooperative game theory است و برتری اصلی این روش نسبت به روش های دیگر این است که

تکیه ی آن به emergence میباشد و به بهینه کردن یک objective function که این باعث میشود که ساختار را در تمامی داده ها (که در آنها هویت داده براساس فاصله و/یا تراکم است) حفظ کند.

این روش در DBS برای کم کردن ابعاد داده های چندین بعدی به دو بعد استفاده میشود. البته این روش به طور کلی میتواند برای کم کردن ابعاد استفاده بشود ولی اینجا در DBS این استفاده مورد نظر ماست.

نکته ای که در مورد pswarm وجود دارد آن است که هیچ پارامتری برای انجام کارش نیاز ندارد و صرفاً براساس فاصله ی اقلیدسی بین نقاط در حالت اولیه کارش را انجام میدهد، البته این فاصله میتواند براساس یک ماتریس ورودی که کاربر خودش آماده میکند هم باشد.

در این الگوریتم هر iteration نوعی annealing میباشد که براساس یک ثابت R (شعاع همسایگی) که در هر iteration به صورت نزولی تغییر میکند، انجام میشود.

ایده ی اصلی پشت روش Pswarm آن است که به آنالیز داده مانند یک بازی نگاه میکند و هر دیتابات یک بازیکن در این بازی هستند. در راند بازی (iteration) یک ثابت R مشخص میشود که براساس آن هر بازیکن (دیتابات) بازیکن هایی که در شعاع R خود هست در نظر میگیرد و در چهارچوب قوانین مختصات، که در ادامه به آنها میپردازیم، به تعادلش خود را میسراند. به این معنی که در آخر هر راند (گام) تمام دیتابات ها در وضعیتی هستند که با تغییر وضعیتشان میزان payoff خودشان یا همسایه هایشان (که تعادل براساس آنها در نظر گرفته شده است) بهتر نمیشود. که این همان تعادل ضعیف نش میباشد. بعد از اینکه یک راند بازی به این حالت رسید، راند بعدی شروع میشود؛ با این تفاوت که مقدار ثابت R کوچک تر میشود.

قوانین این محیط بازی، که یک شبکه ی مشبک (grid) هست از این قرار است؛ اولاً، هیچ دو دیتاباتی حق بودن در یک مختصات مشترک در یک زمان را ندارند و دوماً بهترین حالت قرار گرفتن این دیتابات ها کنار ها مانند یک ۶ ضلعی است، بنابراین برای اینکه بهترین استفاده را از این روش ببریم مختصات خود به صورت شبکه ی شش ضلعی (مانند کندوی عسل) در نظر میگیریم.

نکته ی دیگری که در این روش مورد توجه قرار میگیرد و در روش های دیگر به آن توجه نمیشود مساله ی مربوط به نواحی مرزی است. این مساله به این معنا است که وقتی داده ها را در یک صفحه ی دو بعدی تصویر میکنیم، دیتاباتی که در نواحی مرزی قرار میگیرند ارتباطشان با یکدیگر مشخص نیست، و این مشخص نبودن باعث میشود که اطلاعاتی از روابط بین این ها از دست برود. برای حل کردن این مشکل از فضای خارجی toroidal استفاده میکنیم، این فضا نوعی فضای دونات شکل میباشد که در آن نواحی مرزی به نوعی به یکدیگر متصل هستند و در نتیجه مشکلات فضای مرزی نداریم.

از آنجایی که تمام دیتابات ها در طی محاسبات باید به مختصات قطبی برده شوند (چون محاسبات مربوطه در این مختصات بسیار بهینه تر انجام میشود) و در نهایت به شکل مختصات کارتزین نمایش داده میشود، از روابط زیر برای انجام این تبدیل ها استفاده میکنیم.

$$r = x^2 + y^2$$

$$\phi = \tan^{-1}\left(\frac{y}{x}\right) * \frac{180}{\pi}$$

اندازه‌ی شبکه‌ی مورد نظر به این شکل و با توجه به سه شرط زیر مشخص میشود، اگر حالتی پیش بیاید که معادلات جواب نداشته باشد (شرطها امکان برقراری نداشته باشند) جوابی تقریبی برای معادلات پیدا کرده و از آن استفاده میکنیم.

$$\frac{\sqrt{C^2 + L^2}}{1} \geq \frac{p_{99}(\tilde{D})}{p_{01}(\tilde{D})} =: A$$

$$L * C \geq \alpha * N$$

$$\frac{L}{C} = \frac{\beta}{1}$$

ماتریس D ماتریس فاصله‌های بین دیتاب‌ها است، الف‌ها نشان‌دهنده‌ی تعداد jump position های ممکن است و N تعداد دیتاب‌ها میباشد. همچنین بتا مقدار بین ۰.۵ و ۱ میباشد.

اولین شرط تضمین میکند که شبکه بتواند کوتاه‌ترین و بلندترین فاصله‌ی بین دیتاب‌ها را هندل کند، دومین شرط تضمین میکند که به اندازه‌ی کافی برای دیتاب‌ها فضا برای جا به جایی وجود داشته باشد و در نهایت شرط آخر تضمین میکند که شبکه بیشتر مستطیلی باشد تا مربعی.

The pseudo code-1-2

```

function Positions  $O=Pswarm(matrix D(l, j))$ 

  for all  $z_i \in I$ : assign an initial random polar position  $i_\phi(r) \in O$  on the grid
    to generate DataBots  $b_i \in B$ 
  for  $R=\{Rmax=Lines/2, ..., Rmin\}$  do
    calculate chance  $c(R)$ 
    Repeat for each iteration

       $c = sample(C(R), B), c \subseteq B$ 

       $m_k(c) = uniform(1, Rmax), with k = 1, ... \alpha, m_k(c) \in O$ 

       $l(c) = \underset{j \in \{i, m_k(c)\}}{\operatorname{argmax}} (\lambda(b_j, R))$ 

       $l(!c) = i$ 

       $S = \sum_l \lambda_l(b_l, R)$ 

    Until  $\frac{\partial S(e, \lambda(R))}{\partial e} = 0$ 

  return  $O$  in Cartesian coordinates

end function Pswarm

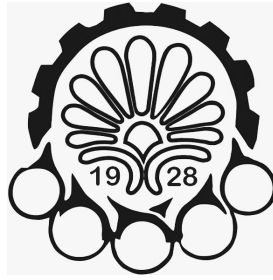
```

همانطور که از سودوکد بالا مشخص است در تابع pswarm الگوریتم به این صورت است که ابتدا به هر دیتابات یک موقعیت رندم در مختصات قطبی داده میشود و سپس الگوریتم روی مجموعه ی نزولی از R ها حرکت میکند که بیشترین مقدار آذ بزرگترین شعاع تقسیم بر دو است (که دلیل آذ این است که بخاطر شکل toroidal مختصات ناحیه ی شامل شعاع با خودش تداخل نداشته باشد). و براساس مقدار R ، مقدار $c(R)$ محاسبه میشود که در واقع یک عدد است که کسری از N است و نشان دهنده ی تعداد دیتابتهایی است که میتوانند در این iteration موقعیت خود را تغییر دهند. سپس یک زیرمجموعه تصادفی از دیتابتهای انتخاب میشود تا بتوانند به دنبال موقعیت جدید بگردند. سپس الگوریتم میزان payoff هر یک از این موقعیتهای را میسنجد و اگر اوضاع را بهتر میکردند تغییر را اعمال میکند. در نهایت آخرین موقعیتهای پیدا شده به مختصات کارتزین برگردانده میشوند و به عنوان خروجی داده میشوند.

نتیجه گیری

خوشه‌بندی DBS ساختارهای بعد بالا را حفظ می‌کند. DBS یک چارچوب خوشه‌بندی انعطاف‌پذیر و قوی است که از سه ماژول مستقل تشکیل شده است. ماژول اول روش کاهش ابعاد (projection) بدون پارامتر Pswarm است که از مفاهیم self-organization و emergence، نظریه بازی و هوش گروهی بهره می‌برد. ماژول دوم، یک تکنیک بصری‌سازی بدون پارامتر برای داده‌های بعد بالا است که نقاطی را روی یک نقشه توپوگرافیک ایجاد می‌کند و بر اساس یک ماتریس generalized-U است. ماژول سوم، خود روش خوشه‌بندی است. خوشه‌بندی می‌تواند توسط بصری‌سازی نیز انجام شود. DBS می‌تواند عملکرد بهتری نسبت به روش‌های خوشه‌بندی معمولاً مانند K-means ، PAM ، خوشه‌بندی طیفی و ... داشته باشد، به ویژه زمانی که اطلاعات قبلی درباره داده‌ها موجود نباشد.

منابع و مراجع



**Amirkabir University of Technology
(Tehran Polytechnic)**

Department of Mathematics & Computer Science

BSc report

Swarm Intelligence for Self-Organized Clustering

**By
Sarina Heshmati**

**Supervisor 1
Dr. Mehdi Ghatte**

**Supervisor 2
Dr. Behnam Yousefimehr**

January 2024