

شبکه نیمه‌ناظر خودسازمانده پویا مبتنی بر یادگیری حداکثری

علی مهریزی، هادی صدوقی یزدی و سیدجواد سیدمهدوی چابک

یادگیری منطبق با دامنه مسئله دارند، الگوریتم‌های یادگیری نیمه‌ناظر می‌گویند [۱].

از دیدگاه کار این مقاله، روش‌های نیمه‌ناظر را می‌توان به دو دسته تقسیم کرد: (الف) روش‌های نیمه‌ناظری که در راستای بهبود خوشه‌بندی داده‌ها ایجاد شده‌اند و (ب) روش‌های نیمه‌ناظری که از خوشه‌بندی نیمه‌ناظر برای طبقه‌بندی داده‌ها استفاده می‌کنند. در ادامه به نمونه‌هایی از هر کدام از این روش‌ها اشاره می‌شود.

الف- روش‌های نیمه‌ناظر بهبوددهنده خوشه‌بندی

ماکاریو^۸ و همکارانش در [۲] یک الگوریتم نیمه‌ناظر مبتنی بر الگوریتم FCM^۹ ارائه کرده‌اند. ایده اصلی این مقاله، خوشه‌بندی نیمه‌ناظر با استفاده از مزیت برچسب قطعی برخی از داده‌ها است. الگوریتم پیشنهادی این مقاله، یک جمله^{۱۰} با ناظر به تابع اصلی الگوریتم FCM اضافه می‌کند. بنابراین در صورتی که متعلق به یک کلاس باشد، تفاوت بین درجه عضویت آن به خوشه کم می‌شود. این رویکرد باعث می‌شود درجه عضویت نمونه‌هایی که دارای کلاس مشابه هستند در همه خوشه‌ها ثابت باقی بماند. در این مقاله یک راه حل وفقی برای بهبود دقت الگوریتم خوشه‌بندی ارائه شده و برای جلوگیری از رشد بدون کنترل خوشه، می‌توان از ماتریس ضرایب حجم خوشه استفاده نمود.

اولین الگوریتم خوشه‌بندی مقید برخط توسط هالکید^{۱۱} و همکارانش در [۳] ارائه گردید. مسئله‌ای که این مقاله به آن می‌پردازد پیداکردن خوشه‌هایی متمرکز و دور از هم است در شرایطی که قیود داده‌شده داده‌ها به صورت جریانی به الگوریتم وارد شود. الگوریتم هالکید^{۱۱} بر پایه الگوریتم خوشه‌بندی^{۱۲} MPCK-Means پایه‌گذاری شده است. الگوریتم MPCK-Means یک الگوریتم برون‌خط است و در محیط‌های برخط نمی‌تواند مورد استفاده قرار گیرد. به همین دلیل هالکید^{۱۱} و همکارانش روند ورود داده‌ها به الگوریتم را به صورت تکه‌هایی^{۱۳} از مجموعه کل داده‌ها در نظر گرفتند که به مرور زمان در اختیار الگوریتم قرار می‌گیرد. طبق روش ارائه‌شده توسط این مقاله عملیات خوشه‌بندی بر روی داده‌های موجود در این تکه‌ها توسط الگوریتم MPCK-Means انجام می‌شود. در روند اجرای این الگوریتم، داده‌ها به صورت یکی‌یکی به خوشه‌ها اختصاص می‌یابند. در فرایند خوشه‌بندی، در تکه واردشده، داده‌هایی که به تازگی وارد سیستم شده‌اند نسبت به داده‌های قدیمی اهمیت بیشتری دارند. الگوریتم هالکید^{۱۱} داده‌های ورودی را به صورت تکه‌هایی در نظر می‌گیرد و سپس عملیات خوشه‌بندی را به صورت برون‌خط و توسط الگوریتم MPCK-Means انجام می‌دهد. همچنین از این روش می‌توان جهت کاهش هزینه محاسباتی حافظه استفاده کرد.

چکیده: شبکه خودسازمانده پویا با یادگیری نیمه‌ناظر در بسیاری از کاربردها نظیر خوشه‌بندی داده‌ها کاربرد دارد. محاسبه پارامترهای شبکه خودسازمانده شامل شکل و ساختار لایه خوشه‌بندی، سطح فعال‌سازی و وزن‌های لایه طبقه‌بندی از جمله مسایل چالش‌برانگیز و مهم آن است. راهکارهای ارائه‌شده فعلی از روش‌های ابتکاری و با یک نگاه محلی سعی در تعیین این پارامترها دارند که در اثر آن، نتایج این الگوریتم‌ها وابستگی بالایی به شرایط دارد. این مقاله یک روش یادگیری نیمه‌ناظر مبتنی بر شبکه خودسازمانده پویا و یادگیری حداکثری را برای اولین بار مورد بررسی قرار می‌دهد. روش پیشنهادی، بدون محاسبه مستقیم پارامترهای شبکه خودسازمانده پویا و با استفاده از روش یادگیری حداکثری، کلاس هر داده را تعیین می‌کند. خطای حاصل از بازخورد سیستم، هم در یادگیری حداکثری و هم در بهینه‌سازی شبکه خودسازمانده پویا مورد استفاده قرار می‌گیرد. در این مقاله، علاوه بر بررسی تحلیلی همگرایی روش پیشنهادی، روش حداکثری تربیتی برای شبکه نیمه‌ناظر خودسازمانده پویا ارائه شده است. آزمایش‌های انجام‌شده بر روی داده‌های برخط و با برچسب جزئی نشان می‌دهند که روش پیشنهادی از نظر دقت، نسبت به روش نیمه‌ناظر خودسازمانده پویا برتری نسبی دارد.

کلیدواژه: یادگیری نیمه‌ناظر، شبکه‌های خودسازمانده پویا، یادگیری حداکثری، یادگیری برخط.

۱- مقدمه

در مسایل موجود در دنیای واقعی با داده‌هایی سروکار داریم که بدون برچسب^۱ و حجم هستند، به صورت برخط^۲ و جریانی^۳ به سیستم وارد می‌شوند، ساختار غیر خطی دارند و لازم است سیستم در زمان کوتاه به آنها پاسخ دهد. در حالت معمول برای استخراج دانش از چنین داده‌هایی از روش‌های خوشه‌بندی بدون ناظر استفاده می‌شود اما چنانچه تعداد معدودی از این داده‌ها دارای اطلاعات اضافی همچون برچسب قطعی، فاصله نسبی^۴، قیود^۵ یا ارتباطات^۶ بین آنها باشند، می‌توان از روش‌های طبقه‌بندی با ناظر و یا روش‌های نیمه‌ناظر^۷ برای یادگیری استفاده کرد. به مجموعه الگوریتم‌هایی که با استفاده از اطلاعات اضافی موجود در داده‌ها،

این مقاله در تاریخ ۱۹ آذر ماه ۱۳۹۲ دریافت و در تاریخ ۱۹ آبان ماه ۱۳۹۳ بازنگری شد.

علی مهریزی، گروه کامپیوتر، دانشگاه آزاد اسلامی، واحد مشهد، مشهد، (email: mehri@staff.um.ac.ir)

هادی صدوقی یزدی، گروه کامپیوتر، دانشگاه فردوسی مشهد، مشهد، (email: h-sadoghi@um.ac.ir)

سیدجواد سیدمهدوی چابک، گروه کامپیوتر، دانشگاه آزاد اسلامی، واحد مشهد، مشهد، (email: mahdavi@mshdiau.ac.ir)

1. Label
2. Online
3. Stream
4. Relative Distance
5. Constrain
6. Connection
7. Semi-Supervised

8. Macario
9. Fuzzy C-Means
10. Term
11. Halkidi
12. Metric Pairwise Constrained K-Means
13. Chunk

جدول ۱: خلاصه الگوریتم‌های نیمه‌ناظر.

مشکلات حل نشده	مزایا	نوع الگوریتم نیمه‌ناظر	روش ارائه شده
برای داده‌های برخط مناسب نیست	نیمه‌ناظر - خوشه‌بندی فازی - فاصله تطبیقی	بهبود خوشه‌بندی	ماکاریو و همکاران
کندی سرعت - عدم بهبود خوشه‌بندی با استفاده از بازخورد داده‌های برچسب‌دار	نیمه‌ناظر - فازی - پایداری در مقابل انواع داده	طبقه‌بندی داده‌ها	لیتی و همکاران
دریافت داده به صورت تکه‌تکه	نیمه‌ناظر - داده‌های مقید	بهبود خوشه‌بندی	هالکیدی و همکاران
دریافت داده‌ها به صورت تکه‌تکه - کندی سرعت	نیمه‌ناظر - یادگیری متریک	طبقه‌بندی داده‌ها	کونگ و همکاران
کندی سرعت ناشی از ایجاد درخت تصمیم	نیمه‌ناظر -	طبقه‌بندی داده‌ها	فرید و همکاران
برای داده‌های برخط مناسب نیست - کندی سرعت ناشی از روش همکارانه	نیمه‌ناظر - یادگیری حداکثری همکارانه	طبقه‌بندی داده‌ها	لی و همکاران
تعیین پارامترهای شبکه GSOM به صورت ابتکاری	نیمه‌ناظر - برخط - مبتنی به GSOM - بهبود خوشه‌بندی با استفاده از بازخورد داده‌های برچسب‌دار	بهبود خوشه‌بندی - طبقه‌بندی داده‌ها	اچسو و همکاران
مشکل داده‌های کم و پارامترهای تصادفی	برخط - مبتنی به GSOM - بهبود خوشه‌بندی با استفاده از بازخورد داده‌های برچسب‌دار - سرعت اجرای بالا - مستقل از تعیین پارامترهای شبکه GSOM	بهبود خوشه‌بندی - طبقه‌بندی داده‌ها	روش پیشنهادی مقاله

کار ابتدا به هر داده در مرحله یادگیری یک وزن تصادفی داده می‌شود و سپس برای این داده‌های درخت تصمیم ساخته می‌شود. بر اساس داده‌های آموزشی وزن‌های اختصاص داده شده اصلاح می‌شود.

در [۷] که توسط لی^۷ و همکارانش معرفی گردیده است، از یادگیری حداکثری و ترکیب آن با یادگیری همکارانه جهت ارائه یک الگوریتم نیمه‌ناظر استفاده شده است. ایده اصلی یادگیری همکارانه بدین صورت است که ابتدا دو یادگیرنده به طور جداگانه با استفاده از داده‌های برچسب‌دار آموزش می‌بینند. سپس پیش‌بینی داده‌های بدون برچسب هر یادگیرنده جهت افزایش مجموعه داده‌های آموزشی یادگیرنده دیگر در اختیار او قرار می‌گیرد. نقطه ضعف یادگیری همکارانه این است که به دلیل وجود حلقه در پروسه یادگیری هم سرعت یادگیری کاهش می‌یابد و هم خطای مرحله یادگیری به سایرین منتشر می‌شود. برای برطرف کردن مشکلات روش یادگیری همکارانه در این مقاله پیشنهاد شده که از روش یادگیری حداکثری که یک روش با ناظر سریع و با دقت بالاست برای یادگیری یادگیرنده‌های روش یادگیری همکارانه استفاده شود.

در [۸] یک الگوریتم نیمه‌ناظر مبتنی بر شبکه خودسازمانده پویا توسط اچسو^۸ و همکارانش ارائه گردیده است. در این الگوریتم از یک مدل دولایه برای طبقه‌بندی داده‌های برخط و با برچسب جزئی استفاده شده است. در لایه اول مدل از شبکه‌های خودسازمانده پویا برای خوشه‌بندی داده‌ها استفاده شده و هر گاه داده برچسب‌داری وارد سیستم شد، از برچسب داده برای طبقه‌بندی داده‌ها استفاده می‌شود. در این روش همانند الگوریتم‌های خوشه‌بندی K- میانگین یا C- میانگین فازی یک عملیات تکراری وجود دارد. با این تفاوت که در هر تکرار، تنها یک داده به صورت تصادفی انتخاب و وارد شبکه می‌شود. به عبارت دیگر این الگوریتم به طور ذاتی برخط است.

جمع‌بندی مقدمه

جدول ۱ خلاصه روش‌های نیمه‌ناظر معرفی شده را نشان می‌دهد. در این جدول علاوه بر نوع الگوریتم، مزایا و مشکلات حل نشده هر الگوریتم

ب- روش‌های نیمه‌ناظر طبقه‌بندی کننده داده‌ها

لیتی^۱ و همکارانش در [۴] بیان می‌کنند که در محیط‌های غیر ثابت، ویژگی و توزیع اصلی داده‌ها با زمان تغییر می‌کند. در اصطلاحات یادگیری ماشین به این اتفاق رانش مفهوم^۲ گفته می‌شود. برای مثال در طبقه‌بندی داده‌ها، مفهوم، همان کلاس داده‌ها یا همان مرز بین داده‌هاست و رانش همان تغییرات مرز داده در طول زمان است. رانش می‌تواند تدریجی یا ناگهانی (تغییر مفهوم)، قراردادی یا گسترشی، قطعی یا تصادفی یا دوره‌ای باشد. نیاز اصلی یادگیری برخط شناسایی و مقابله با رانش داده است [۴]. روش دانه‌ای^۳ برای یادگیری نیمه‌ناظر در حالت کلی کلی به وسیله یک شبکه عصبی پنج‌لایه انجام می‌شود. در مدل پنج‌لایه، لایه دوم وظیفه خوشه‌بندی داده را دارد، لایه سوم عملیات ادغام را انجام می‌دهد و لایه چهارم لایه تصمیم‌گیری در مورد کلاس داده است. لایه اول و پنجم به ترتیب لایه ورودی و خروجی شبکه عصبی است.

در [۵] یک الگوریتم نیمه‌ناظر دیگر برای شناسایی اجزای صحنه معرفی شده است. در این مقاله کونگ^۴ و همکارانش از گراف دوطرفه برای بررسی میزان شباهت بین داده‌ها استفاده کرده‌اند و بر اساس آن یادگیری متریک را انجام می‌دهند. از داده‌های برچسب‌دار و داده‌های بدون برچسب با امتیاز بالا جهت به روز رسانی یادگیری متریک استفاده شده است. الگوریتم معرفی شده یک الگوریتم برخط است.

در [۶] الگوریتم نیمه‌ناظری توسط فرید^۵ و همکارانش معرفی گردیده که مبتنی بر روش یادگیری جمعی^۶ است. روش یادگیری جمعی در حقیقت یک روش برون‌خط است که بر اساس رأی‌گیری بین چندین طبقه‌بند، برچسب داده مشخص می‌شود. در این مقاله از روش درخت تصمیم برای طبقه‌بندی داده‌ها استفاده شده و با اصلاح این ساختار یک الگوریتم برخط مبتنی بر یادگیری ترکیبی ارائه شده است. برای انجام این

1. Leite
2. Concept Drift
3. Granular
4. Cong
5. Farid
6. Ensemble Learning

7. Li

8. Hsu

در (۱)، ζ_t^m برچسب واقعی داده t ام برای کلاس m است و PS_t^m مقدار تخمینی شبکه نیمه‌ناظر برای همان داده و کلاس m است که این دو مقدار بر اساس (۲) و (۳) تعیین می‌شوند

$$\zeta_t^m = \begin{cases} 0, & l_t \neq \hat{m}_t \text{ or missing label} \\ 1, & l_t = \hat{m}_t \end{cases} \quad (2)$$

$$PS_t^m = \sum_{c=1}^C w^{cm} \times o_t^c, \quad m = \{1, 2, \dots, M\} \quad (3)$$

در (۳) تابع o_t^c میزان شباهت داده وارد شده با گره‌های لایه خوشه‌بند را مشخص می‌کند و آن را با نام درجه فعال‌سازی^۲ می‌شناسیم. مقدار درجه فعال‌سازی برای هر گره c با استفاده از تابع گوسی که در (۴) آمده است محاسبه می‌گردد

$$o_t^c = \exp\left(-\frac{\|x_t - v_c\|^2}{\sigma_c^2}\right), \quad c \in \{1, \dots, C\} \quad (4)$$

که در آن پارامتر σ_c میانگین فاصله‌های گره c با کلیه گره‌های همسایه است و توسط (۵) محاسبه می‌شود

$$\sigma_c = \frac{\sum_{k \in Ne(c)} \|v_k - v_c\|}{|Ne(c)|} \quad (5)$$

به منظور بهینه‌سازی شبکه خودسازمانده پویا از بازخورد داده‌های برچسب‌دار، خطای گره‌ها بر اساس (۶) محاسبه می‌شود. لازم به یادآوری است که میزان خطا در شبکه خودسازمانده پویا، پارامتری است که میزان گسترش گره‌ها در این شبکه تعیین می‌کند

$$E_{BMU(t)} = \begin{cases} 0, & l_t = \hat{m}_t \\ \frac{\sqrt{D}}{2}, & l_t \neq \hat{m}_t \\ \frac{\|x_t - v_{BMU(x_t)}\|^2}{2}, & \text{missing label} \end{cases} \quad (6)$$

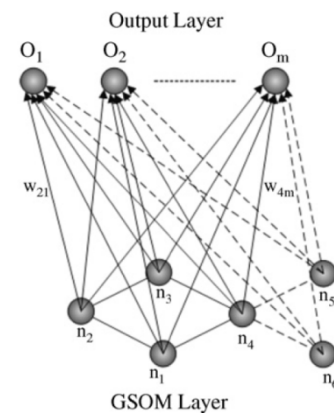
$m = \{1, 2, \dots, M\}$

در این رابطه l_t مشخص‌کننده برچسب واقعی داده وارد شده با شبکه در زمان t است و مقدار \hat{m}_t از (۷) به دست می‌آید

$$\hat{m}_t = \underset{m}{\text{Argmax}} PS_t^m, \quad m = \{1, 2, \dots, M\} \quad (7)$$

۲-۲ یادگیری حداکثری

روش یادگیری حداکثری اولین بار در سال ۲۰۰۴ توسط هوانگ^۳ معرفی گردید [۹] که این روش یک الگو برای شبکه‌های عصبی معرفی می‌کند و به کمک آن می‌توان به راحتی و در زمان بسیار کم و بدون محاسبه پارامترهای لایه مخفی به خوبی تابع هدف را تقریب زد. این روش با حداکثر کردن فاصله خطای بین دو مرحله از الگوریتم، سرعت همگرایی را حداکثر می‌کند. تعیین تعداد مراکز لایه مخفی از جمله مسایل چالشی این روش است که وابسته به نوع داده‌ها است. در حالتی که شبکه عصبی از نوع RBF^۴ گوسی باشد، پارامترهای لایه مخفی شامل مرکز و عرض تابع گوسی گره‌ها است که در حالت معمول به صورت تصادفی تعیین می‌شوند.



شکل ۱: ساختار شبکه‌های خودسازمانده پویای نیمه‌ناظر [۸].

نیز بیان شده است. همان طور که در جدول نیز بیان شده، روش پیشنهادی این مقاله، یک روش یادگیری نیمه‌ناظر برخط است که بدون محاسبه پارامترهای شبکه خودسازمانده و با استفاده از روش یادگیری حداکثری، به طور توأم هم خوشه‌بندی را بهبود می‌دهد و هم طبقه‌بندی داده را انجام می‌دهد. علاوه بر این، روش پیشنهادی بر خلاف مدل اچسو، پارامترهای شبکه خودسازمانده را به طور تحلیلی و بهینه تعیین می‌کند و همچنین به دلیل استفاده از یادگیری حداکثری از سرعت اجرای بالایی برخوردار است. یادآوری می‌شود که روش اچسو [۸]، قابلیت بهبود خوشه‌بندی و طبقه‌بندی داده‌های برخط را به طور توأم دارد و یک روش مبتنی بر شبکه خودسازمانده پویا است.

در فصل دوم، مقدمات لازم جهت الگوریتم پیشنهادی بیان می‌شود. در فصل سوم، الگوریتم پیشنهادی ارائه شده است. در فصل چهارم، نتایج آزمایش‌های انجام شده بر روی داده‌های مختلف آمده و در فصل پنجم، نتیجه‌گیری گفته شده است.

۲- مفاهیم اولیه

در این بخش مفاهیم اولیه لازم جهت ارائه الگوریتم پیشنهادی بیان می‌شود.

۲-۱ شبکه‌های خودسازمانده پویا

شبکه‌های خودسازمانده پویا (GSOM)^۱ یک روش یادگیری بدون ناظر برای خوشه‌بندی و نمایش ساختار داده‌های برخط است. در سال ۲۰۰۸ اچسو و همکارانش با اضافه کردن یک لایه با ناظر بر بالای این شبکه، قابلیت نیمه‌ناظر را به آن اضافه کردند. شکل ۱ ساختار این مدل را نشان می‌دهد [۸].

در مدل نیمه‌ناظر معرفی شده، تعداد گره‌ها در لایه بالا برابر با تعداد کلاس‌هاست (M) و تعداد گره‌ها در لایه پایین به تعداد خوشه‌هاست (K). بین دو لایه، اتصال کاملی از وزن‌ها وجود دارد که با w^{cm} نمایش داده می‌شود. هدف آن است این وزن‌ها به گونه‌ای آموزش داده شوند که با وارد شدن داده به لایه خوشه‌بند، نزدیک‌ترین گره به داده فعال و سپس بر اساس وزن بین دو لایه، کلاس مناسب برای داده مشخص گردد. آموزش وزن‌های بین دو لایه توسط داده‌های برچسب‌دار و با استفاده از (۱) انجام می‌شود

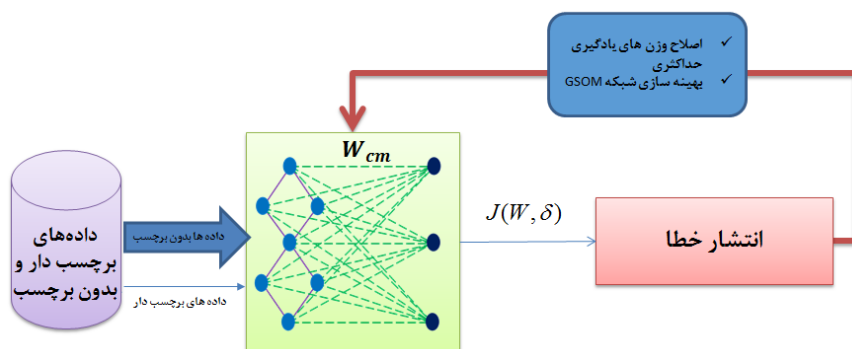
$$\Delta w^{cm} = \eta \times (\zeta_t^m - PS_t^m) \times o_t^c \quad (1)$$

2. Activation Degree

3. Huang

4. Radial Basis Function

1. Growing Self Organizing Map



شکل ۲: ساختار روش پیشنهادی.

انجام دهد.

۳-۱ ساختار روش پیشنهادی

در این مقاله یک روش نیمه‌ناظر برخط مبتنی بر شبکه خودسازمانده پویا و یادگیری حداکثری ارائه می‌شود. روش پیشنهادی از شبکه خودسازمانده پویا برای خوشه‌بندی داده‌های بدون برچسب و از روش یادگیری حداکثری برای طبقه‌بندی داده‌های برچسب‌دار استفاده می‌شود. به بیان دقیق‌تر وقتی که داده بدون برچسب وارد سیستم می‌شود، ساختار شبکه خودسازمانده به طور برخط شکل می‌گیرد. هنگامی که داده برچسب‌دار وارد سیستم می‌شود، یادگیری حداکثری با تعیین مراکز لایه مخفی خود به وسیله گره‌های شبکه خودسازمانده پویا عملیات طبقه‌بندی را انجام می‌دهد. پس از عملیات طبقه‌بندی، از خطای حاصل از آن هم برای بهبود یادگیری حداکثری و هم برای بهینه‌سازی شبکه خودسازمانده پویا استفاده می‌شود. شکل ۲ ساختار روش پیشنهادی را نشان می‌دهد.

دلیل ارائه این پیشنهاد این است که شبکه خودسازمانده که بر اساس داده‌های بدون برچسب شکل می‌گیرد مراکز را به ما معرفی می‌کند که در آنجا تجمعی از داده‌ها وجود دارد. بنابراین این مراکز، نقاط بهتری نسبت به حالت تعیین مراکز به صورت تصادفی است. از طرف دیگر، از خطای منتشرشده در لایه طبقه‌بند برای گسترش بهینه در لایه خوشه‌بند استفاده خواهد شد. بدین ترتیب یک سیستم نیمه‌ناظر برخط مبتنی بر یادگیری حداکثری به وجود می‌آید که قادر است از مزایای یادگیری حداکثری و شبکه خودسازمانده به طور توأمان استفاده نماید.

همان طور که قبلاً هم گفته شد، تعیین تعداد و محل مراکز در مدل یادگیری حداکثری به عنوان یک چالش مطرح است و چنانچه این مراکز به درستی تعیین نشوند سیستم یادگیر دچار بیش‌برازش^۲ یا کم‌برازش^۳ می‌شود. بنابراین در روش پیشنهادی که تعداد و محل مراکز توسط شبکه خودسازمانده پویا تعیین می‌گردد، این مشکل برطرف می‌گردد.

۳-۲ تحلیل همگرایی روش پیشنهادی

به منظور بررسی همگرایی روش پیشنهادی، در این بخش با استفاده از تعاریف شبکه خودسازمانده پویا و روش یادگیری حداکثری، شرایط و سرعت همگرایی را بررسی می‌نماییم.

تعریف ۱: خطا را در لحظه n ام به فرم (۱۲) تعریف می‌کنیم که در آن ζ تابع هدف و PS_n مقدار تخمین شبکه GSOM است

$$e_n = \zeta - PS_n \quad (12)$$

در ماشین یادگیر حداکثری و برای شبکه عصبی سه‌لایه، برای به دست آوردن وزن لایه خروجی (β) از (۸) تا (۱۰) استفاده می‌شود

$$\tilde{\beta} = H^+ T \quad (8)$$

$$H^+ = (H^T H)^{-1} H^T \quad (9)$$

$$\tilde{\beta} = (H^T H)^{-1} H^T T \quad (10)$$

که در آن H از (۱۱) به دست می‌آید و T بردار برچسب، داده‌های برچسب‌دار است

$$H(w_1, \dots, w_N, b_1, \dots, b_N, x_1, \dots, x_N) = \begin{bmatrix} g(w_1 x_1 + b_1) & \dots & g(w_N x_1 + b_N) \\ \dots & \dots & \dots \\ g(w_1 x_N + b_1) & \dots & g(w_N x_N + b_N) \end{bmatrix} \quad (11)$$

ماتریس H ماتریس خروجی لایه مخفی نامیده می‌شود. w و b پارامترهای تابع لایه مخفی هستند و در حالت معمول به طور تصادفی تعیین می‌شوند.

۳- روش پیشنهادی

هدف این مقاله ارائه یک راهکار نیمه‌ناظر خودسازمانده پویای مبتنی بر روش یادگیری حداکثری^۱ است. به دلایل زیر روش پیشنهادی مورد نیاز است:

- بیشتر روش‌های نیمه‌ناظر موجود نیاز دارند تمام داده‌های یادگیری را تا پایان الگوریتم در حافظه نگهداری کنند و بنابراین در این روش‌ها حافظه زیادی استفاده می‌شود. همچنین برای به روز رسانی اطلاعات کلاس‌ها و خوشه‌ها نیاز است هر بار تمامی این اطلاعات پردازش شوند. بنابراین در این روش‌ها علاوه بر حافظه، نیاز به محاسبات زیاد و تکراری است.
- پاسخ‌گویی سریع و دقیق از جمله ویژگی‌های مهم برای روش‌های نیمه‌ناظر است. بیشتر روش‌های موجود یا سریع نیستند و یا دیر همگرا می‌شوند و بنابراین نمی‌توانند در زمان کم پاسخ قابل قبولی را ارائه کنند.
- بیشتر روش‌های نیمه‌ناظر ذاتاً قابلیت برخط‌بودن را ندارند و به همین دلیل داده‌ها را به صورت تکه‌تکه ذخیره کرده و در اختیار الگوریتم قرار می‌دهند.

در این مقاله قصد داریم جهت پاسخ به نیازهای گفته‌شده، روشی را ارائه کنیم که بتواند به صورت برخط و بهینه عملیات یادگیری را

2. Over Training
3. Under Training

1. Extreme Learning Machine

$$\|e_n\| = \left\| e_{n-1} - \frac{e_{n-1} o_n^T o_n}{o_n^T o_n} \right\| \quad (۳۲)$$

$$\|e_n\| = \|e_{n-1} - e_{n-1}\| = 0 \quad (۳۳)$$

بنابراین همگرایی اثبات می‌گردد.

۳-۳ یادگیری حداکثری ترتیبی در شبکه خودسازمانده پویا

یادگیری حداکثری از جنبه‌های مختلفی دسته‌بندی شده است. از نظر معماری شبکه، یادگیری حداکثری به دو دسته معماری ثابت یا روش پایه‌ای^۱ و معماری افزایشی^۲ تقسیم می‌شود. از نظر داده‌های در دسترس، روش یادگیری افزایشی به دو دسته ترتیبی^۳ و دسته‌ای^۴ تقسیم می‌شود. در روش یادگیری افزایشی دسته‌ای، همه داده‌های یادگیری در آغاز کار در دسترس هستند و در روش یادگیری افزایشی ترتیبی، داده‌های یادگیری در ابتدا وجود ندارند و به صورت یکی‌یکی به سیستم وارد می‌شوند. از آنجایی که در این مقاله فرض می‌شود که داده‌ها به صورت برخط به سیستم وارد می‌شوند، بنابراین روش یادگیری حداکثری افزایشی ترتیبی برای مسئله مناسب است که این روش برای محاسبه سریع ماتریس ضرایب ارائه شده که در ادامه به بررسی آن برای شبکه نیمه‌ناظر مبتنی بر یادگیری حداکثری می‌پردازیم.

در مدل نیمه‌ناظر خودسازمانده پویا، ضرایب بین دو لایه را وزن لایه خروجی و سطح فعال‌سازی را برابر با تابع RBF گوسی از مدل یادگیری حداکثری در نظر می‌گیریم. بنابراین برای به دست آوردن وزن بین لایه‌ای مدل نیمه‌ناظر به روش حداکثری ترتیبی با استفاده از (۸) تا (۱۱) داریم

$$w = o^+ T \quad (۳۴)$$

$$o^+ = (o^T o)^{-1} o^T \quad (۳۵)$$

$$w = (o^T o)^{-1} o^T T \quad (۳۶)$$

به دلیل این که ایجاد ماتریس w پس از ورود هر داده برچسب‌دار زمان‌بر است و به منظور کاهش محاسبات که در روش‌های برخط از اهمیت بالایی برخوردار است، در ادامه نحوه محاسبه ماتریس w به صورت مرحله‌ای بیان می‌شود.

اگر فرض کنیم N داده برچسب‌دار داریم و ساختار داده‌ها به صورت (x, t) است آن گاه اگر اولین دسته از داده‌های ورودی به سیستم را به صورت $N_1 = \{(x_i, t_i)\}_{i=1}^{N_1}$ نشان دهیم، بر اساس (۳۵) ماتریس H به صورت زیر محاسبه می‌شود

$$O = \begin{bmatrix} G(a_1, b_1, X_1) & \dots & G(a_{N_1}, b_{N_1}, X_1) \\ \dots & \dots & \dots \\ G(a_1, b_1, X_{N_1}) & \dots & G(a_{N_1}, b_{N_1}, X_{N_1}) \end{bmatrix} \quad (۳۷)$$

$$T = \begin{bmatrix} t_1^T \\ \dots \\ t_{N_1}^T \end{bmatrix} \quad (۳۸)$$

1. Basic
2. Incremental
3. Online Sequential
4. Batch

قضیه: چنانچه تابع درجه فعال‌سازی (رابطه (۴))، تابع تکه‌ای پیوسته انتگرال‌پذیر، $R^d \rightarrow R$ و $\int_R o_n(x) dx \neq 0$ باشد آن گاه برای هر گره GSOM حداکثر همگرایی در صورتی به دست می‌آید که

$$w_n = \frac{\langle e_{n-1}, o_n \rangle}{\|o_n\|^2} \quad (۱۳)$$

اثبات: اگر بین هر دو مرحله خطای $\|e_n\|^2$ و $\|e_{n-1}\|^2$ تغییر بسیار شدیدی داشته باشد در این صورت به نظر می‌رسد تابع PS_n به سرعت به سمت ζ میل می‌کند

$$\Delta_n = \left\| \|e_{n-1}\|^2 - \|e_n\|^2 \right\| \quad (۱۴)$$

$$\Delta_n = \left\| \|e_{n-1}\|^2 - \left\| \zeta - \sum_{i=1}^n w_i o_i \right\|^2 \right\| \quad (۱۵)$$

$$\Delta_n = \left\| \|e_{n-1}\|^2 - \left\| \zeta - (PS_{n-1} + w_n o_n) \right\|^2 \right\| \quad (۱۶)$$

$$\Delta_n = \left\| \|e_{n-1}\|^2 - \left\| \zeta - PS_{n-1} - w_n o_n \right\|^2 \right\| \quad (۱۷)$$

$$\Delta_n = \left\| \|e_{n-1}\|^2 - \|e_{n-1} - w_n o_n\|^2 \right\| \quad (۱۸)$$

$$\Delta_n = \left\| \|e_{n-1}\|^2 - \langle e_{n-1} - w_n o_n, e_{n-1} - w_n o_n \rangle \right\| \quad (۱۹)$$

$$\Delta_n = \left\| \|e_{n-1}\|^2 - \|e_{n-1}\|^2 + 2w_n \langle e_{n-1}, o_n \rangle - w_n^T \|o_n\|^2 \right\| \quad (۲۰)$$

$$\Delta_n = 2w_n \langle e_{n-1}, o_n \rangle - w_n^T \|o_n\|^2 \quad (۲۱)$$

برای حداقل‌سازی Δ نسبت به w بایستی (۲۲) برقرار باشد. بر این اساس داریم

$$\frac{\partial \Delta_n}{\partial w_n} = 0 \quad (۲۲)$$

$$\frac{\partial \Delta_n}{\partial w_n} = 2w_n \langle e_{n-1}, o_n \rangle - w_n^T \|o_n\|^2 = 0 \quad (۲۳)$$

$$\Delta_n = 2w_n \langle e_{n-1}, o_n \rangle - w_n^T \|o_n\|^2 = 0 \quad (۲۴)$$

$$\Rightarrow w_n = \frac{\langle e_{n-1}, o_n \rangle}{\|o_n\|^2} \quad (۲۵)$$

تا اینجا وزن بهینه را برای بیشترین کاهش خطا به دست آوردیم. برای اثبات همگرایی لازم است (۲۶) برقرار باشد

$$\lim_{n \rightarrow \infty} \|e_n\| = \lim_{n \rightarrow \infty} \left\| \zeta - PS_n \right\| = 0 \quad (۲۶)$$

اثبات: طبق تعریف ۱ داریم

$$e_n = \zeta - PS_n \quad (۲۷)$$

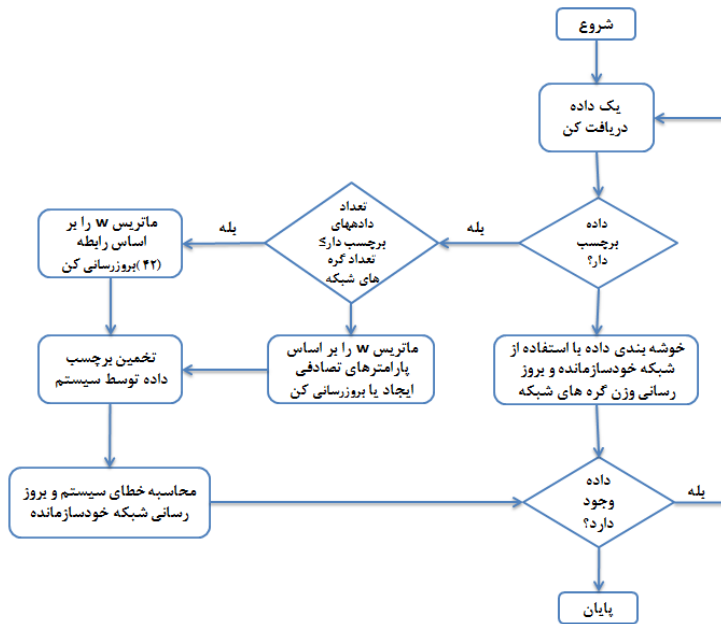
$$\|e_n\| = \left\| \zeta - \left(\sum_{i=1}^n w_i o_i \right) \right\| \quad (۲۸)$$

$$\|e_n\| = \left\| \zeta - \left(\sum_{i=1}^{n-1} w_i o_i + w_n o_n \right) \right\| \quad (۲۹)$$

$$\|e_n\| = \left\| (\zeta - PS_{n-1}) - w_n o_n \right\| \quad (۳۰)$$

با جایگذاری e_{n-1} از تعریف ۱ و استفاده از نتیجه (۲۵) داریم

$$\|e_n\| = \left\| e_{n-1} - \frac{\langle e_{n-1}, o_n \rangle}{\|o_n\|^2} o_n \right\| \quad (۳۱)$$



شکل ۳: الگوریتم یادگیری نیمه‌ناظر خودسازمانده مبتنی بر یادگیری حداکثری.

پیش می‌آید: حالت اول هنگامی است که تعداد داده‌های برچسب‌دار کمتر از تعداد گره‌های شبکه خودسازمانده پویا است. در این حالت بردار w با استفاده از پارامترهای تصادفی ایجاد یا به روز رسانی می‌شود. همچنین تعداد گره‌های لایه مخفی از روش یادگیری حداکثری برابر با تعداد داده‌های برچسب‌دار موجود در نظر گرفته می‌شود. حالت دوم هنگامی است که تعداد داده‌های برچسب‌دار بزرگ‌تر یا مساوی تعداد گره‌های شبکه خودسازمانده باشد. در این شرایط بردار w بر اساس (۴۲) محاسبه می‌شود. بعد از محاسبه وزن هر کلاس برای داده ورودی (w)، برچسب تخمینی سیستم برای داده برچسب‌دار ورودی محاسبه می‌گردد. سپس میزان خطای حاصل از تفاضل برچسب تخمینی و برچسب واقعی محاسبه می‌گردد و به لایه خوشه‌بند بازخورد داده می‌شود. بر اساس بازخورد داده‌شده از لایه طبقه‌بند، میزان خطای گره برنده از شبکه خودسازمانده پویا به روز رسانی می‌گردد. نکته قابل توجه دیگر این است که هنگام پیاده‌سازی روش پیشنهادی دو روش برای مقداردهی به پارامتر δ وجود دارد. بر اساس پیشنهاد هوانگ می‌توان مقدار تصادفی برای δ در نظر گرفت [۹] یا بر اساس پیشنهاد لین^۱ برای آن مقدار $\delta = m^{-(1/(r+d)+\epsilon)}$ را در نظر گرفت [۱۱]. در فصل چهارم به منظور آزمایش پیشنهادها از هر دو روش استفاده شده و نتیجه گزارش خواهد شد.

۳-۵ جمع‌بندی روش پیشنهادی

روش‌های نیمه‌ناظر موجود دارای اشکالاتی هستند که در ابتدای بخش ۳ به آنها اشاره شد. روش پیشنهادی این مقاله به دلایل زیر تا حد امکان این اشکالات را برطرف نموده است:

- روش پیشنهادی به دلیل استفاده از شبکه خودسازمانده پویا، تنها اطلاعات گره‌های شبکه خودسازمانده پویا را نگهداری می‌کند. بنابراین نیاز به نگهداری همه داده‌های یادگیری ندارد و در نتیجه حافظه مصرفی آن کم است. همچنین برای به روز رسانی وزن گره‌های شبکه خودسازمانده پویا، با ورود هر داده محاسبات بسیار کمی انجام می‌شود. بنابراین روش پیشنهادی علاوه بر حافظه کم، دارای سرعت پردازش مناسب نیز است.

متغیر K را به صورت زیر تعریف می‌کنیم

$$K_i = o_i^T o_i \quad (39)$$

بنابراین وزن لایه خروجی بر اساس روش یادگیری حداکثری (رابطه (۳۳)) برای داده‌های ورودی به صورت زیر به دست می‌آید

$$w^{(i)} = K_i^{-1} o_i^T T_i \quad (40)$$

حال چنانچه دسته دیگری از داده‌ها مانند $N_1 = \{(x_i, t_i)\}_{i=N_1}^{N_1+N_2}$ وارد شود آن گاه وزن لایه خروجی از (۴۱) به دست می‌آید

$$w^{(i)} = K_i^{-1} \begin{bmatrix} o_i \\ o_i \end{bmatrix}^T \begin{bmatrix} T_i \\ T_i \end{bmatrix} \quad (41)$$

که ضریب K به صورت زیر است

$$K_i = \begin{bmatrix} o_i \\ o_i \end{bmatrix}^T \begin{bmatrix} o_i \\ o_i \end{bmatrix} \quad (42)$$

$$K_i = \begin{bmatrix} o_i^T & o_i^T \end{bmatrix} \begin{bmatrix} o_i \\ o_i \end{bmatrix} = K_i + o_i^T o_i \quad (43)$$

در نهایت چنانچه $P_{k+1} = K_{k+1}^{-1}$ در نظر بگیریم و روابط را به صورت بازگشتی بازنویسی کنیم، خواهیم داشت

$$\begin{aligned} P_{k+1} &= P_k - P_k o_{k+1}^T (I + o_{k+1} P_k o_{k+1}^T)^{-1} o_{k+1} P_k \\ w^{(k+1)} &= w^{(k)} + P_{k+1} o_{k+1}^T (T_{k+1} - o_{k+1} w^{(k)}) \end{aligned} \quad (44)$$

نکته حائز اهمیت این است که در مرحله اول و هنگام محاسبه H نباید تعداد داده‌های برچسب‌دار از تعداد گره‌های لایه مخفی کمتر باشد. همچنین در صورتی که تعداد داده‌ها زیاد باشد، می‌توان به جای محاسبه ضرب $o^T o$ ، ضرب $o o^T$ را با ملاحظه تمهیداتی استفاده کرد [۱۰].

۳-۴ پیاده‌سازی روش پیشنهادی

شکل ۳ فلوجارت الگوریتم یادگیری نیمه‌ناظر خودسازمانده مبتنی بر یادگیری حداکثری را نشان می‌دهد. در این الگوریتم چنانچه داده برچسب‌دار نباشد، به وسیله شبکه خودسازمانده پویا عملیات خوشه‌بندی برای آن انجام می‌شود و در صورتی که داده برچسب‌دار باشد دو حالت

جدول ۲: مشخصات مجموعه داده‌های مصنوعی و UCI استفاده شده برای ارزیابی.

نام مجموعه داده	تعداد خوشه‌ها	تعداد ویژگی‌ها	تعداد نمونه‌ها
Two Ring	۲	۲	۲۳۶
Spiral	۲	۲	۳۰۰۰
Iris	۳	۴	۱۵۰
Sonar	۲	۶۰	۲۰۸
Ionosphere	۲	۳۴	۳۵۱
Vehicle	۴	۱۸	۸۴۶
Abalone	۳	۸	۴۱۷۷
Waveform	۳	۲۱	۵۰۰۰

تنها از الگوریتم نیمه‌ناظر اچ‌سو استفاده شده است [۸]. در این مقاله روش پیشنهادی را با نام $ELSSGSOM^3$ و روش اچ‌سو را با نام $SSGSOM^4$ می‌شناسیم.

برای آزمایش الگوریتم‌های نیمه‌ناظر، ابتدا مجموعه‌ای با ترتیب تصادفی از داده‌های بدون برچسب و تعداد اندکی داده برچسب‌دار ایجاد می‌شود. سپس این مجموعه به هر الگوریتم عرضه شده و دقت الگوریتم‌ها محاسبه می‌گردد. در ادامه، رفته‌رفته تعداد داده‌های برچسب‌دار درون مجموعه افزایش می‌یابد و دقت الگوریتم‌ها مجدداً محاسبه می‌شود. با افزایش تعداد داده‌های برچسب‌دار، انتظار می‌رود که دقت الگوریتم‌های مورد آزمایش افزایش یابد. در این روش ارزیابی، نکته اصلی آن است که در هر دوره، مجموعه داده ایجاد شده از داده‌های برچسب‌دار برای هر دو الگوریتم‌ها یکسان نگه داشته شود. با این کار مشخص می‌شود که با در اختیار داشتن اطلاعات یکسان، کدام الگوریتم توانسته است طبقه‌بندی بهتری را ارائه نماید. برای مجموعه اولیه برچسب‌دار، از ۵٪ داده برچسب استفاده خواهد شد و با شروع هر دوره جدید، مجموعه داده‌های برچسب‌دار ۵٪ افزایش می‌یابد و این کار تا زمانی که ۴۰٪ داده‌ها، برچسب‌دار شوند ادامه خواهد یافت [۱۳]. در شروع کار مقدار اولیه w به طور تصادفی تعیین می‌شود و همچون سایر تنظیمات مربوط به شبکه GSOM برای هر دو الگوریتم‌ها یکسان است.

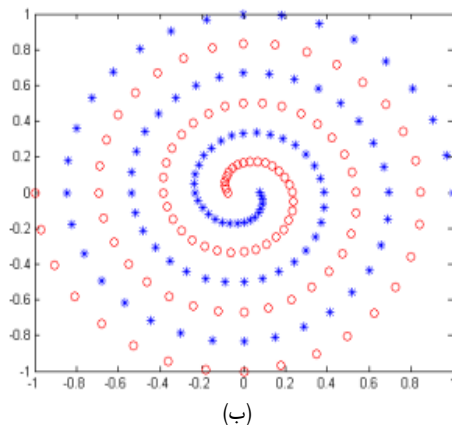
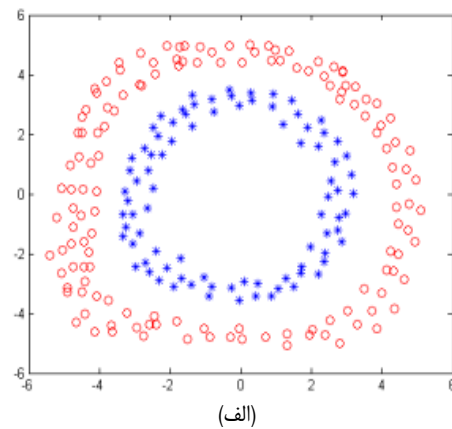
در ادامه ابتدا مجموعه داده‌های مورد استفاده معرفی می‌شوند و سپس نتایج اجرای الگوریتم پیشنهادی و الگوریتم رقیب بر روی این مجموعه داده‌ها نشان داده می‌شود. در آزمایش‌های انجام شده در بخش‌های ۳-۴ و ۴-۴ از داده‌های نرمال شده استفاده گردیده و همچنین مقدار δ به طور تصادفی تعیین شده و نتایج حاصل از متوسط ۵۰ بار اجرای الگوریتم‌ها است.

۴-۱ معرفی مجموعه داده‌های استفاده شده

مجموعه داده‌های مصنوعی استفاده شده برای ارزیابی در شکل ۴ نشان داده شده است. در این دسته از داده‌ها، برای امر خوشه‌بندی مجموعه داده حلزونی به علت شکل پیچیده و به شدت غیر محدب خود یک مجموعه داده سخت است. همچنین در جدول ۲، مجموعه داده‌های UCI مورد استفاده همراه با مشخصات کامل نشان داده شده است.

۴-۲ معرفی معیارهای ارزیابی

معروف‌ترین معیار ارزیابی در یادگیری ماشین معیار دقت است. این معیار یکی از ساده‌ترین معیارهای ارزیابی در یادگیری ماشین است. مطابق



شکل ۴: ساختار مجموعه داده‌های مصنوعی استفاده شده برای ارزیابی، (الف) دو حلقه و (ب) حلزونی.

- سرعت روش پیشنهادی به دلیل حجم کم محاسبات در لایه خوشه‌بند و همچنین سرعت بالای یادگیری حداکثری ترتیبی در لایه طبقه‌بند، نسبت به روش‌های دیگر برتری دارد. روش پیشنهادی به علت این که در لایه طبقه‌بند از یادگیری حداکثری ترتیبی استفاده می‌کند، سریع همگرا می‌شود (روش حداکثری ترتیبی همانند روش RLS عمل می‌کند و ثابت شده که روش RLS نسبت به روش LMS سریع‌تر همگرا می‌گردد [۱۲]). در نتیجه می‌تواند در زمان کم جواب قابل قبول ارائه کند.
- ساختار شبکه خودسازمانده به صورت برخط شکل می‌گیرد و از این نظر برای ورود داده‌ها به صورت برخط به سیستم بسیار مناسب است. بنابراین به دلیل این که روش پیشنهادی مبتنی بر شبکه خودسازمانده پویا و یادگیری حداکثری ترتیبی است، یک روش کاملاً برخط است.

در مجموع می‌توان گفت روش پیشنهادی، یک روش یادگیری نیمه‌ناظر، برخط و سریع است که نیاز به حافظه کمی دارد.

۴- ارزیابی و آزمایش

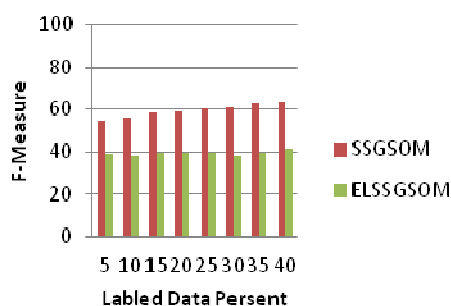
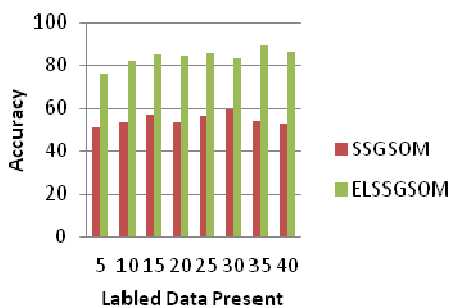
در این بخش با استفاده از معیارهای دقت، F-Measure و آنالیز مقاومت^۲ بر روی مجموعه داده‌های مصنوعی و UCI، به ارزیابی الگوریتم پیشنهادی می‌پردازیم. به دلیل این که روش پیشنهادی و روش اچ‌سو [۸] تنها الگوریتم‌هایی هستند که هم طبقه‌بندی داده و هم بهبود خوشه‌بندی را انجام می‌دهند، بنابراین برای مقایسه الگوریتم پیشنهادی،

3. Extreme Learning Semi Supervised GSOM

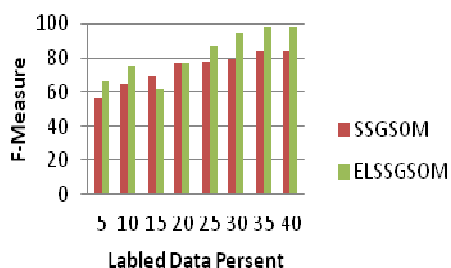
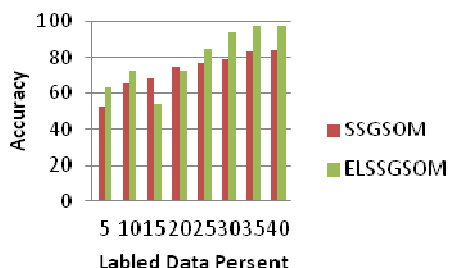
4. Semi Supervised GSOM

1. Accuracy

2. Robustness Analysis



شکل ۵: نتایج اجرای دو الگوریتم بر روی مجموعه داده Spiral.



شکل ۶: نتایج اجرای دو الگوریتم بر روی مجموعه داده Two Ring.

استفاده از معیارهای درستی و یادآوری دارای این اشکال است که برای هر الگوریتم دو مقدار بازگردانده خواهد شد. این مسئله تصمیم‌گیری در مورد آن که کدام الگوریتم بهتر است را با مشکل مواجه می‌نماید. برای رفع این مشکل F-Measure مورد استفاده قرار می‌گیرد. این معیار از محاسبه میانگین همساز بین دو معیار درستی و یادآوری به صورت (۴۸) به دست می‌آید

$$FM_q = \frac{PR_q \times RE_q}{PR_q + RE_q} \quad (48)$$

برای مقایسه کارایی و میزان برتری الگوریتم‌های مورد مقایسه از روشی به نام تحلیل مقاومت که توسط گنگ^۴ و همکاریانش ارائه شده استفاده خواهد شد. در این روش میزان کارایی نسبی الگوریتم q برای یک مجموعه داده با استفاده از (۴۹) و به صورت نسبت صحت خوشه‌بندی الگوریتم q به بیشترین میزان صحت در میان Q الگوریتم رقیب اجرا شده بر روی مجموعه داده مورد نظر محاسبه می‌شود

$$RA_q = \frac{Acc_q}{\max Acc_j}, \quad \forall j \in \{1, 2, \dots, Q\} \quad (49)$$

۳-۴ نتایج ارزیابی بر روی مجموعه داده‌های مصنوعی

شکل ۵ و ۶ نتایج اجرای الگوریتم پیشنهادی و رقیب را بر روی دو مجموعه داده مصنوعی نشان می‌دهد. همان طور که در شکل مشخص

(۴۵) منظور از نرخ دقت الگوریتم q ، نسبت تعداد نمونه‌هایی که به طور صحیح خوشه‌بندی شده‌اند به تعداد کل نمونه‌ها است

$$Acc_q = \frac{TP}{TP + TN + FP + FN} = \frac{TP}{|X|} \quad (45)$$

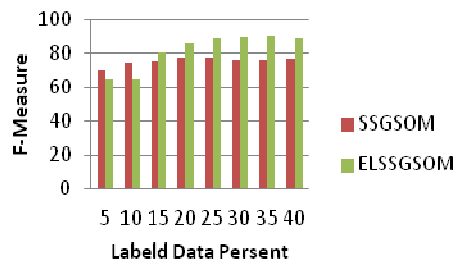
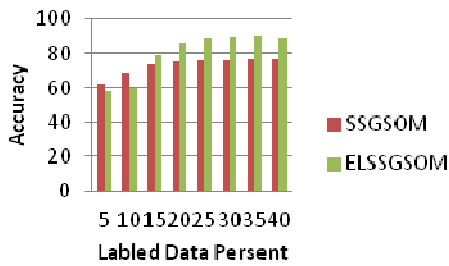
که در آن $|X|$ نماینده تعداد عناصر و X مجموعه کل داده‌های مسئله است. TP بیانگر تعداد داده‌هایی است که دارای برچسب k هستند و در خوشه k دسته‌بندی شده‌اند. TF بیانگر تعداد داده‌هایی است که دارای برچسب k نیستند و در خوشه k قرار گرفته نشده‌اند. FP نشان‌دهنده تعداد داده‌هایی است که دارای برچسب k نیستند ولی در خوشه k طبقه‌بندی شده‌اند. FN نشان‌دهنده تعداد داده‌هایی است که دارای برچسب k هستند ولی در خوشه k قرار نگرفته‌اند.

به علت سادگی و همچنین قابلیت تفسیر بالا، این معیار در بسیاری از زمینه‌ها خصوصاً یادگیری ماشین کاربرد زیادی یافته است اما این معیار توجه زیادی به تعداد پیش‌بینی‌های درست نموده و به اشتباهات مرتکب‌شده از سوی الگوریتم توجه اندکی می‌کند. برای رفع این مشکل از معیارهای درستی^۱ و یادآوری^۲ استفاده می‌شود. این دو معیار بیشتر در بازیابی اطلاعات^۳ و شناسایی الگو به کار می‌روند و به علت آن که به نوعی مکمل یکدیگر هستند، معمولاً با هم به کار می‌روند. معیار درستی، درصدی از داده‌های بازیابی شده است که واقعاً در خوشه مورد نظر قرار دارند و معیار یادآوری، درصدی از داده‌های خوشه مورد نظر است که توسط الگوریتم بازگردانده شده است. همین طور که از تعریف دو معیار برمی‌آید هر دو مبتنی بر بازیابی اطلاعات از یک منبع پایه‌گذاری شده‌اند. این دو معیار در شناسایی الگو به صورت (۴۶) و (۴۷) تعریف می‌گردند

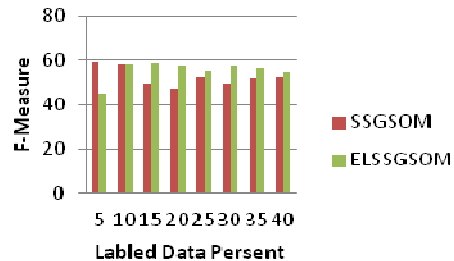
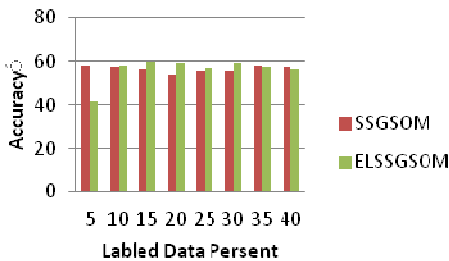
$$PR_q = \frac{TP}{TP + FP} \quad (46)$$

$$RE_q = \frac{TP}{TP + FN} \quad (47)$$

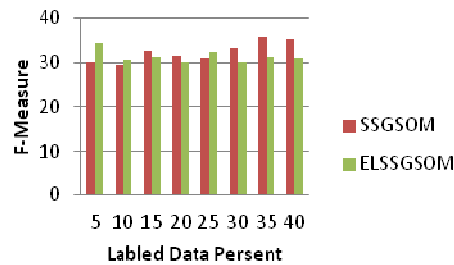
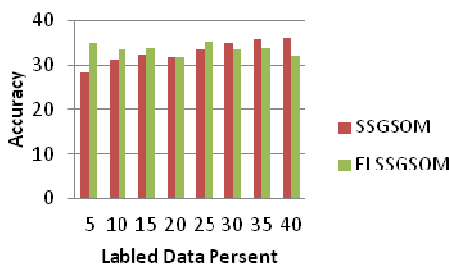
1. Precision
2. Recall
3. Information Retrieval



شکل ۷: نتایج اجرای دو الگوریتم بر روی مجموعه داده Iris.



شکل ۸: نتایج اجرای دو الگوریتم بر روی مجموعه داده Ionosphere.



شکل ۹: نتایج اجرای دو الگوریتم بر روی مجموعه داده Vehicle.

کمتر است که دلیل آن مشکل داده‌های کم و پارامترهای تصادفی در الگوریتم پیشنهادی است (شکل ۸).

با انتخاب مجموعه داده‌های بزرگ‌تر، مشکل داده‌های کم و پارامترهای تصادفی در الگوریتم پیشنهادی برطرف می‌شود. در این شرایط الگوریتم پیشنهادی در شرایطی از الگوریتم رقیب بهتر است که تعداد داده‌های برچسب‌دار کمتر باشد. شکل ۹ نتایج بررسی را بر روی مجموعه داده Vehicle نشان می‌دهد.

۴-۵ بررسی‌های بیشتر و بررسی سرعت

در این بخش علاوه بر بررسی مجموعه داده‌های بیشتر و بزرگ‌تر، به بررسی سرعت اجرای الگوریتم پیشنهادی و رقیب در عمل می‌پردازیم. برای انجام این کار در شرایط یکسان (پارامترهای شبکه GSOM، مقدار اولیه w و ...) هر الگوریتم را ۳۵ بار بر روی داده‌های طبقه‌بند انجام داده و سپس متوسط پارامترهای به دست آمده، گزارش شده است. حجم داده‌های برچسب‌دار در این سناریو ۳۰٪ از کل داده‌های هر مجموعه داده است. سخت‌افزار مورد استفاده برای اجرا، یک دستگاه کامپیوتر شخصی با پردازنده intel iV QV۴۰ با فرکانس ۱٫۷۳ GHz و ۴ GB RAM است. در این بخش مقدار اولیه برای δ بر اساس پیشنهاد لین برابر با $\delta = m^{-(1/(r+d)+\epsilon)}$ در نظر گرفته شده است [۱۱].

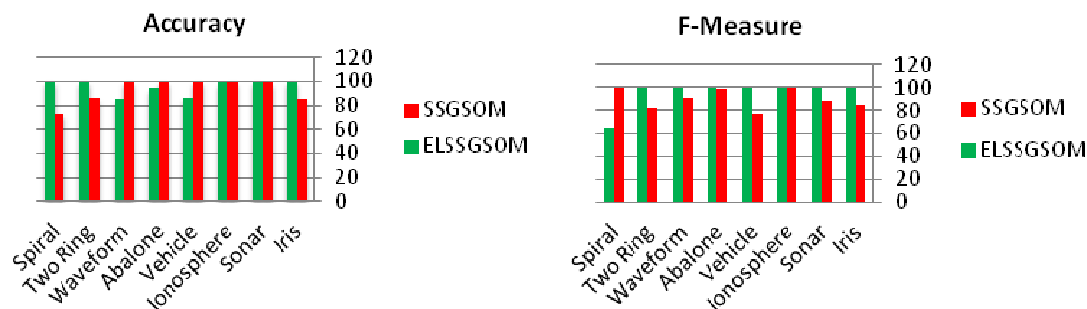
با مشاهده نتایج به دست آمده در جدول ۳، مشخص است که روش پیشنهادی از نظر پارامتر F-Measure یک پارامتر کامل و دقیق است دارای برتری تقریباً کامل است. همچنین این برتری با زمان کمتر به دست می‌آید و از نظر پارامتر دقت نیز نتایج بسیار به هم نزدیک است. لازم به ذکر است که بر اساس پیشنهاد لین بایستی تعداد گره‌ها در روش

دقت الگوریتم پیشنهادی بر روی هر دو مجموعه داده نسبت به الگوریتم رقیب برتری دارد (شکل ۷). همچنین F-Measure الگوریتم پیشنهادی برای مجموعه داده two ring نیز برتری دارد اما به دلیل این که مجموعه داده Spiral غیر خطی است و همچنین تعداد داده‌های این مجموعه نیز کم است، F-Measure الگوریتم پیشنهادی نسبت به الگوریتم رقیب کمتر بوده و در این شرایط میزان اشتباهات الگوریتم پیشنهادی نسبت به الگوریتم رقیب بیشتر است.

۴-۴ نتایج ارزیابی بر روی مجموعه داده‌های UCI

در این بخش، ابتدا مجموعه داده Iris مورد ارزیابی قرار گرفته است. در این مجموعه داده، هم معیار دقت و هم F-Measure الگوریتم پیشنهادی نسبت به الگوریتم رقیب برتری دارد. تنها در حالتی که داده‌های با برچسب ۵٪ و ۱۰٪ است، کارایی الگوریتم پیشنهادی کمی از الگوریتم رقیب کمتر است و این به دلیل آن است که در شروع کار الگوریتم پیشنهادی و تا هنگامی که تعداد داده‌های برچسب‌دار از تعداد گره‌های شبکه خودسازمانده کمتر باشد، الگوریتم پیشنهادی به صورت کاملاً تصادفی عمل می‌کند. ما در این مقاله نام این مشکل را مشکل داده‌های کم و پارامترهای تصادفی می‌نامیم. نکته حائز اهمیت این است که با افزایش تعداد داده‌های برچسب‌دار کارایی الگوریتم پیشنهادی افزایش پیدا می‌کند.

مجموعه داده بعدی که مورد بررسی قرار گرفته است، مجموعه داده Ionosphere است. نتایج بررسی مجموعه داده Ionosphere نشان می‌دهد که الگوریتم پیشنهادی بر الگوریتم رقیب برتری نسبی دارد و تنها در حالتی که داده‌های برچسب‌دار ۵٪ است، کارایی الگوریتم پیشنهادی



شکل ۱۰: نتایج آنالیز مقاومت برای ۸ مجموعه داده مورد استفاده برای دو پارامتر دقت و F-Measure.

جدول ۳: نتایج به دست آمده برای مجموعه داده مختلف و مقایسه زمان اجرای الگوریتم‌ها.

مجموعه داده	الگوریتم	دقت	F-Measure	زمان اجرا
Sonar	SSGSOM	۵۲/۸۰	۵۹/۵۵	$3,020,094e^{-1}$
	ELMGSOM	۵۳/۱۹	۶۷/۴۸	$2,908,465e^{-1}$
Ionosphere	SSGSOM	۶۲/۱۷	۶۶/۹۷	$6,273,166e^{-1}$
	ELMGSOM	۶۱/۵۵	۶۶/۵۵	$5,762,781e^{-1}$
Vehicle	SSGSOM	۳۲/۵۷	۲۹/۲۲	$2,314,561e^{+3}$
	ELMGSOM	۲۸/۱۵	۳۷/۹۳	$1,971,417e^{+3}$
Abalone	SSGSOM	۳۹/۲۶	۴۸/۷۹	$6,433,007e^{+1}$
	ELMGSOM	۳۶/۹۷	۴۹/۸۲	$3,269,975e^{+1}$
Waveform	SSGSOM	۴۲/۱۴	۴۳/۱۵	$1,449,995e^{+2}$
	ELMGSOM	۳۵/۷۳	۴۷/۵۲	$1,259,348e^{+2}$

[2] V. Macario and F. D. A. de Carvalho, "An adaptive semi-supervised fuzzy clustering algorithm based on objective function optimization," *Proc. IEEE Int. Conf. on Fuzzy Systems, FUZZIEEE*, 8 pp., 10-15 Jun. 2012.

[3] M. Halkidi, M. Spiliopoulou, and A. Pavlou, "A semi-supervised incremental clustering algorithm for streaming data," in *Proc. of the 16th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining, PAKDD'12*, vol. 1, pp. 578-590, 2012.

[4] D. Leite, P. Costa, and F. Gomide, "Evolving granular neural network for semi-supervised data stream classification," in *Proc. Int. Joint Conf. on Neural Networks, IJCNN'10*, pp. 1877-1884, Barcelona, Spain, 18-23 Jun. 2010.

[5] Y. Cong, J. Liu, J. Yuan, and J. Luo, "Self-supervised online metric learning with low rank constraint for scene categorization," *IEEE Trans. Image Process.*, pp. 3179-3191, vol. 22, no. 8, Aug. 2013.

[6] D. M. Farid, et al., "An adaptive ensemble classifier for mining concept drifting data streams," *Expert Syst. Appl.*, vol. 40, no. 15, pp. 5895-5906, 1 Nov. 2013.

[7] K. Li, J. Zhang, H. Xu, S. Luo, and H. Li, "A semi-supervised extreme learning machine method based on co-training," *J. of Computational Information Systems*, vol. 9, no. 1, pp. 207-214, Jan. 2013.

[8] A. Hsu and S. K. Halgamuge, "Class structure visualization with semi-supervised growing self-organizing maps," *Neurocomputing*, vol. 71, no. 16-18, pp. 3124-3130, Oct. 2008.

[9] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proc. 2004 IEEE Int. Joint Conf. on Neural Networks*, vol. 2, pp. 985-990, 25-29 Jul. 2004.

[10] L. Guo, J. H. Hao, and M. Liu, "An incremental extreme learning machine for online sequential learning problems," *Neurocomputing*, vol. 128, no. 27, pp. 50-58, Mar. 2013.

[11] S. Lin, X. Liu, J. Fang, and Z. Xu, *Is Extreme Learning Machine Feasible? A Theoretical Assessment (Part II)*, arXiv preprint arXiv:1401.6240, 2014.

[12] S. S. Haykin, *Adaptive Filter Theory*, Pearson Education India, 2008.

[13] N. Grira, M. Crucianu, and N. Boujemaa, "Active semi-supervised fuzzy clustering," *Pattern Recognition*, vol. 41, no. 5, pp. 1834-1844, May 2008.

[14] X. Geng, D. C. Zhan, and Z. H. Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 6, pp. 1098-1107, Dec. 2005.

یادگیری حداکثری بایستی $d/2 \leq r \leq d$ باشد که در آن d ابعاد داده است [۱۱] اما با آزمایش‌های انجام‌شده در این مقاله با تنظیم تعداد گره‌ها به این مقدار نتایج مناسبی به دست نیامد. بنابراین تعداد گره‌ها برابر با تعداد گره‌های شبکه GSOM در نظر گرفته شد.

۴-۶ نتایج آنالیز مقاومت

شکل ۱۰ آنالیز مقاومت برای مجموعه داده‌های بحث‌شده را در حالتی که ۳۰٪ از کل داده‌ها دارای برچسب است نشان می‌دهد. نتایج نشان می‌دهد که اگر میزان داده‌های برچسب‌دار افزایش یابد، کارایی الگوریتم پیشنهادی نسبت به الگوریتم رقیب نیز افزایش می‌یابد که این را می‌توان به عنوان یک برتری برای الگوریتم پیشنهادی محسوب کرد. به بیان دیگر با افزایش داده‌های یادگیری، کارایی الگوریتم پیشنهادی نیز افزایش می‌یابد که نشان‌دهنده یادگیری بهتر الگوریتم پیشنهادی است.

۵- نتیجه‌گیری

در این مقاله برای اولین بار روش یادگیری نیمه‌ناظر خودسازمانده پویای مبتنی بر یادگیری حداکثری ارائه گردید و ثابت شد که این الگوریتم همگراست. سپس با استفاده از روش یادگیری حداکثری افزایشی ترتیبی که برای داده‌های برخط مناسب است، نحوه محاسبه سریع ضرایب بین دو لایه، در مدل نیمه‌ناظر خودسازمانده پویا بیان گردید. ارزیابی الگوریتم پیشنهادی بر روی داده‌های مصنوعی و UCI نشان می‌دهد در شرایطی که تعداد داده‌های برچسب‌دار برای یادگیری کافی باشد الگوریتم پیشنهادی ضمن برخورداری از کارایی مناسب، دارای سرعت بالاتری نیز است.

مراجع

[1] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*, Morgan & Claypool, 2009.

سیدجواد سیدمهدوی چاپک در سال ۱۳۸۱ مدرک کارشناسی مهندسی برق الکترونیک خود را از صنعتی شریف و کارشناسی ارشد و دکتری مهندسی برق الکترونیک خود را به ترتیب در سال‌های ۱۳۸۳ و ۱۳۸۹ از دانشگاه علم و صنعت دریافت نمود. ایشان در حال استادیار گروه مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد مشهد می‌باشد و زمینه‌های علمی مورد علاقه نام‌برده شامل طراحی سیستم‌های مطمئن، رایانش نرم و مدارهای مجتمع دیجیتال می‌باشد.

علی مهریزی در سال ۱۳۸۴ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه آزاد اسلامی واحد مشهد و در سال ۱۳۹۰ مدرک کارشناسی ارشد مهندسی کامپیوتر خود را از دانشگاه آزاد اسلامی واحد مشهد دریافت نمود. زمینه‌های علمی مورد علاقه نام‌برده شامل موضوعاتی مانند الگوریتم‌های یادگیری، یادگیری نیمه‌ناظر، محاسبات نرم و بهینه‌سازی می‌باشد.

هادی صدوقی یزدی هم‌اکنون استاد گروه مهندسی کامپیوتر در دانشگاه فردوسی مشهد می‌باشد. ایشان در سال ۱۳۷۳ مدرک کارشناسی مهندسی برق خود را از دانشگاه فردوسی مشهد و کارشناسی ارشد و دکتری مهندسی برق را به ترتیب در سال‌های ۱۳۷۵ و ۱۳۸۴ از دانشگاه تربیت مدرس دریافت نمود. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: شناسایی آماری الگو، یادگیری ماشین، بینایی ماشین، پردازش سیگنال، داده کاوی و بهینه‌سازی.