

SVM فازی مبتنی بر خوشه‌بندی کوهونن چند لایه برای شناسایی ارقام دست‌نویس

مهدی صالح‌پور^۱، علیرضا بهراد^۲

^۱ دانشجوی کارشناسی ارشد دانشکده فنی مهندسی دانشگاه شاهد، salehpour@shahed.ac.ir

^۲ استادیار دانشکده فنی مهندسی دانشگاه شاهد، behrad@shahed.ac.ir

چکیده

در این مقاله ساختار طبقه‌بندی جدیدی معرفی می‌گردد به طوری که کلاس‌هایی با همپوشانی دادگان بالا، راحت‌تر و با خطای کمتری بتوانند از هم تفکیک گردند. روش پیشنهادی بر مبنای ساختار کوهونن دو لایه می‌باشد که از این ساختار به منظور استفاده از خوشه‌بندی اولیه داده‌های آموزش استفاده می‌گردد که هدف از آن افزایش بعد داده‌های ورودی به صورت هدفمند است تا اینکه این کلاس‌ها بتوانند با دقت بیشتری توسط طبقه‌بند از هم جدا شوند. به دلیل اینکه داده‌های ورودی همپوشانی بالایی دارند بنابراین به طبقه‌بندی‌کننده‌ای نیاز داریم که ماکزیمم حاشیه اطمینان را برای تفکیک کلاس‌ها از هم داشته باشد، به همین سبب از طبقه‌بند *SVM* استفاده می‌کنیم و از آنجائیکه این طبقه‌بند از خوشه‌های مختلفی استفاده می‌کند به صورت طبقه‌بندی‌کننده *SVM* فازی در آمده است. این الگوریتم را با داده‌های دهگانه ارقام دست‌نویس فارسی تست کرده و نتایج نشانگر برتری این الگوریتم در برابر روش‌های کلاسیک *SVM* و *MLP* و همچنین روش‌های مراجع [۱] و [۲] است.

کلیدواژه: ارقام دست‌نویس فارسی، خوشه‌بندی، کوهونن دو لایه، *SVM* فازی

۱- مقدمه

پایگاه‌های داده مختلف نتایج متفاوتی را خواهد داد. در [۵] برای بازشناسی حروف عربی از گشتاور هندسی یک بعدی به همراه طبقه‌بند بیز استفاده شده است، البته این روش در مورد پیاده‌سازی کاراکترهای چاپی بوده و در مورد کاراکترهای دست‌نویس کارایی مناسبی نخواهد داشت. در مرجع [۶] برای بازشناسی ارقام فارسی از روش تکرار تصاویر نرمالیزه مربوط به یک کلاس استفاده کرده و بر اساس میزان تکرار پیکسل‌های سیاه در مختصات خاص، قالبی را به وجود آوردند و در طبقه‌بندی تصاویر تست از این قالب استفاده کردند. عیب این روش این می‌باشد که وابستگی شدیدی به قرارگیری اعداد در محدوده مورد نظر داشته و نسبت به داده‌های پرت بسیار حساس می‌باشد. همچنین در حالت چرخش، به دلیل اینکه محل قرارگیری اعداد چرخش یافته نسبت به قالب به دست آمده از مرحله آموزش متفاوت می‌باشد، الگوریتم از دقت پایینی برخوردار می‌باشد که چنین مشکلاتی در الگوریتم پیشنهادی بسیار کمتر می‌باشد. روش عصبی فازی از روشهای دیگر برای شناسایی حروف دست‌نویس است که در مرجع [۷] مورد استفاده قرار گرفته شده است. در الگوریتم آنها استفاده از شخص‌خبره به عنوان یک نیاز اساسی می‌باشد که این روند در الگوریتم پیشنهادی ما مشاهده نمی‌شود. شبکه خودسازمانده کوهونن نوعی توپولوژی است که

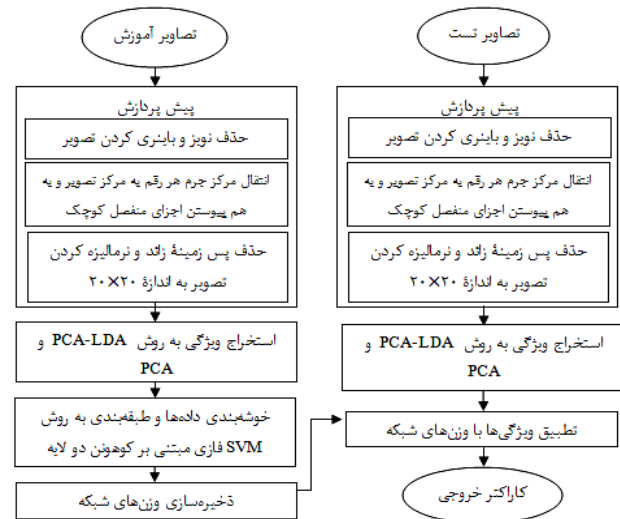
امروزه حجم زیادی از اسناد کاغذی موجود، توسط اسکنرها یا دوربین‌ها به اسناد تصویری دیجیتالی تبدیل می‌شوند. ذخیره‌سازی، بازیابی و مدیریت کارآمد این اسناد تصویری، در بسیاری از کاربردها، نظیر اتوماسیون اداری و کتابخانه‌های دیجیتالی اهمیت فراوانی دارند. در نتیجه دستیابی به الگوریتم‌های مؤثر به منظور آنالیز تصویری اسناد، یک نیاز اساسی به حساب می‌آید. شناسایی کاراکترهای دست‌نویس یکی از کاربردهای بسیار جالب در شناسایی الگو می‌باشد. در کاربردهای ^۱ OCR، شناسایی کاراکترهای دست‌نویس به خصوص ارقام، کاربردهای فراوانی دارد که می‌توان به شناسایی ارقامی در مرتب‌کردن نامه‌های پستی و فرایندهای چک در بانکها اشاره کرد [۳]. برای تشخیص ارقام از ویژگی‌ها و روشهای مختلفی استفاده شده است. در مرجع [۴]، به منظور شاخص‌گذاری و بازیابی تصاویر بافت‌گونه، بانکی از فیلترهای گابور نرمالیزه شده با پاسخ DC صفر استفاده شده است. البته هیچ تعریف استاندارد از تابع گابور دو بعدی به علت اختلاف در اندازه‌های مختلف پوش تابع گوسی و فرکانس مربوطه تابع سینوسی وجود ندارد، بنابراین الگوریتم در

¹ Optical character recognition

فضایی با ابعاد بالا را بر روی یک صفحه نگاشت می‌دهد. تحقیقات قبلی نشانگر این می‌باشد که ساختار کوهونن زمان آموزش به مراتب کمتری در مقایسه با سایر ساختارهای شبکه عصبی دارد [۸] [۹].

۲- کلیات الگوریتم

شکل ۱ بلوک دیاگرام این طرح را نشان می‌دهد. ابتدا روی داده‌های ورودی مرحله پیش‌پردازش را انجام دادیم، به این طریق که پس از بارگذاری تصاویر، آنها را باینری کرده و نویز را از تصاویر حذف کردیم سپس ابعاد اعداد را به 40×40 تغییر داده و پس از محاسبه مرکز ثقل هر عدد، با استفاده از این مختصات، عدد را به مرکز تصویرش انتقال دادیم.



شکل ۱. بلوک دیاگرام کلی سیستم

برای دستیابی به سرعت پردازش بیشتر، نقاط حاشیه‌ای سفید پس زمینه که تمایز چندانی در طبقه‌بندی ایجاد نمی‌کنند را حذف کردیم. به این صورت که با انتخاب کوچک‌ترین مربعی که کل تصویر در آن قرار دارد آن را به ابعاد 20×20 نرمالیزه می‌کنیم. از آنجایی که تغییر مقیاس معمولاً ممکن است منجر به نازک شدن تصویر و حذف پیکسل‌های اصلی تصویر گردد بنابراین در مرحله قبل از تغییر مقیاس توسط عملگر مورفولوژی گسترش رقم مورد نظر را ضخیم می‌کنیم. اندازه پنجره این عملگر به صورت تجربی 3×3 انتخاب شده است.

در این الگوریتم برای استخراج ویژگی از دو روش مجزایی استفاده کردیم. در روش اول از آنالیز اجزای اصلی استفاده شد، به این ترتیب که تمام تصاویر آموزش را به فضای دیگری انتقال داده و پس از

گرفتن کواریانس از این داده‌ها، جهت بیشترین پراکندگی را به دست آوردیم که توضیحات کامل آن در زیر بخش ۳-۱ آورده شده است. در الگوریتم بعدی از روش آنالیز تفکیک کننده خطی برای استخراج ویژگی استفاده کرده و با استفاده از تبدیل PCA^2 بُعد داده ورودی را کم کرده، سپس از طریق این نوع آنالیز جهت بیشترین تفکیک‌پذیری را به دست می‌آوریم که توضیحات کامل آن در زیر بخش ۳-۲ آمده است. سپس عمل کاهش بعد داده‌های ورودی را انجام داده و ویژگی‌هایی را که می‌توانند تمایز بیشتری را ایجاد کنند انتخاب کردیم. این ویژگی‌ها به نوعی بیانگر داده ورودی می‌باشند، بنابراین برای اینکه تفکیک‌پذیری بالایی را ایجاد نماییم باید دوباره آنها را به فضای بالاتری نگاشت دهیم که برای این نوع نگاشت از شبکه کوهونن دو لایه استفاده کردیم. به طوریکه داده‌های ورودی در داخل خوشه‌هایی قرار گرفته تا در مرحله بعدی توسط طبقه‌بند SVM^3 به صورت فازی این خوشه‌ها کلاسه‌بندی شوند.

در مرحله طبقه‌بندی نیز برای مقایسه الگوریتم پیشنهادی، از دو روش مجزای MLP^4 و SVM استفاده کردیم که توضیحات تکمیلی آنها در بخش ۴ آمده است.

۳- استخراج ویژگی

در این مقاله از دو روش متفاوت برای استخراج ویژگی استفاده می‌شود. یکی از این روش‌ها استفاده از تکنیک PCA است و روش دیگر استفاده از تکنیک $PCA-LDA$ می‌باشد که در بخش‌های ذیل توضیح داده می‌شوند.

۳-۱- استخراج ویژگی به روش PCA

در این روش کل تصاویر به صورت بردارهایی از پیکسل‌ها تبدیل شده و پس از چیدن آن در ماتریس اصلی (W)، ماتریس کواریانس (C) آنها محاسبه می‌شود. با این عمل میزان پراکندگی هر پیکسل نسبت به کل پیکسل‌های دیگر سنجیده می‌شود که روابط به صورت زیر است [۱۰]:

$$\bar{X} = \frac{\sum_{i=1}^N X_i}{N} \quad (1)$$

$$\hat{X}_i = X_i - \bar{X} \quad (2)$$

$$W = [\hat{X}_1, \hat{X}_2, \dots, \hat{X}_N] \quad (3)$$

$$C = \frac{1}{N} \sum_{i=1}^N \hat{X}_i \hat{X}_i^T = \frac{1}{N} W W^T \quad (4)$$

² Principal Component Analysis

³ Support Vector Machine

⁴ Multi Layer Perceptron

در این مرحله بردار ویژه ماتریس A را بدست آورده و M تا از بزرگترین مقادیر ویژه این ماتریس را تحت نگاشت این بردار ویژه به عنوان ویژگی به طبقه بندی کننده برای مرحله شناسایی می‌دهیم [۱۱].

۴-۱- طبقه‌بندی کننده SVM فازی مبتنی بر کوهون دو لایه

ایده اصلی طبقه‌بند کننده SVM این است که نمونه‌های ورودی را به بعد بالاتری نگاشت می‌کند که توسط این نگاشت، در فضای با بعد بالاتر جدا سازی بین کلاس‌ها، طبق قضیه پوشش^۵، بهتر انجام می‌گیرد [۳].

SVM یک طبقه‌بندی کننده دو کلاسی یا باینری است. اگر مجموعه داده‌های ورودی $X_i, i=1, 2, \dots, n$ متعلق به یکی از دو کلاس ω_1 و ω_2 باشد آنگاه یک خط جدا کننده به صورت زیر خواهیم داشت [۱۱].

$$G(x) = \omega^T x + \omega_0 \quad (11)$$

$$\omega^T x + \omega_0 \begin{cases} > 0 \\ < 0 \end{cases} \rightarrow x \begin{cases} \omega_1 & y_i = +1 \\ \omega_2 & y_i = -1 \end{cases} \quad (12)$$

روش پیشنهادی در این مقاله الگوریتم SVM فازی بر مبنای خوشه‌بندی کوهون دو لایه می‌باشد. مزیت این الگوریتم در برابر طبقه‌بند SVM در این می‌باشد که در حالت فازی، هر داده منحصرأ به یک طبقه منتسب نمی‌شود. بلکه داده‌ها با استفاده از توابع عضویتشان به دسته‌های مختلفی منتسب می‌شوند. با این کار مشکل نواحی طبقه‌بندی نشده کمتر می‌شود.

۴-۲- کوهون دو لایه

شبکه کوهون تک لایه، نوعی طبقه‌بندی کننده بدون سرپرست می‌باشد. در این شبکه نرون‌ها به صورت رقابتی در یک صفحه در کنار هم قرار گرفته و نرون‌های برنده بر روی همسایه‌های خودشان تاثیر می‌گذارند. هر یک از بردارهای ورودی بر اساس معیار شباهت، یکی از این نرون‌ها را فعال می‌کنند، بنابراین نرون‌هایی که در صفحه خروجی در کنار هم فعال شدند بیشترین شباهت را نسبت به هم دارند و در مرحله وزن‌دهی در SVM باید وزن بیشتری را به آنها اختصاص داد. معیار شباهت بر اساس فاصله اقلیدسی بین بردار ورودی با بردار وزن شبکه است و توسط رابطه (۱۳) بیان می‌شود.

$$\|x - w_c\| = \min_i \|x - w_i\| \quad (13)$$

که بردار X_i بیانگر هر کدام از اعداد می‌باشد، N تعداد کل ارقام مربوط داده‌های آموزش، \bar{X} بردار میانگین داده‌های آموزش، W ماتریس اصلی و C ماتریس کواریانس می‌باشد.

سپس تبدیل آنالیز مولفه اصلی PCA به ماتریس کواریانس اعمال می‌شود. با توجه به این که تبدیل PCA جهت بیشترین پراکندگی‌ها را در داده اصلی نشان می‌دهد، در عین حال برای کاهش بعد نیز به کار می‌رود [۱۰]. M بردار اول ماتریس حاصل از PCA، که متناظر با بزرگترین مقادیر ویژه این ماتریس می‌باشد، به عنوان ویژگی به طبقه بندی داده می‌شود.

۳-۲- استخراج ویژگی به روش PCA_LDA

در این روش نیز مانند روش قبلی ماتریس کواریانس داده‌های ورودی محاسبه شده و تبدیل PCA به منظور کاهش ابعاد داده‌های ورودی اعمال می‌شود. یعنی تعداد ویژگی‌های هر مشاهده که برابر تعداد پیکسل‌های آن تصویر مربوط به ارقام بود را کم می‌کنیم. سپس با استفاده از الگوریتم LDA که در زیر توضیح آن آمده ویژگی‌های مورد نظر را استخراج می‌کنیم. بدین صورت که از تمام ستون‌های ماتریس جدیدی که از PCA به دست آمد، طبق رابطه (۱) میانگین می‌گیریم به طوری که در اینجا X_i همان سطرهای ماتریس جدید می‌باشند. سپس از تصاویر مربوط به هر کلاس نیز به صورت جداگانه طبق رابطه (۵) میانگین می‌گیریم.

$$\bar{Y}^k = \frac{\sum_{i=1}^{N_k} Y_i^k}{N_k} \quad k=1,2,\dots,10 \quad (5)$$

که N_k تعداد داده‌های آموزشی در کلاس k می‌باشد. Y_i^k میانگین مربوط به کلاس (رقم) k است. در مرحله بعدی همه بردارهای تصاویر مربوط به هر کلاس را از میانگین آن کلاس طبق رابطه (۶) کم کرده و طبق رابطه (۷) ماتریس پراکندگی هر کلاس را تشکیل می‌دهیم تا بتوانیم ماتریس پراکندگی درون کلاسی را بر طبق (۸) تشکیل دهیم.

$$\hat{Y}_i^k = Y_i^k - \bar{Y}^k \quad k=1,2,\dots,10 \quad (6)$$

$$S^k = \sum_{i=1}^{N_k} (\hat{Y}_i^k \cdot \hat{Y}_i^{kT}) \quad k=1,2,\dots,10 \quad (7)$$

$$S_w = \sum_{k=1}^{10} S^k \quad (8)$$

سپس برای به دست آوردن ماتریس پراکندگی بین کلاسی از رابطه (۹) استفاده می‌کنیم.

$$S_B = \frac{N}{10} \sum_{k=1}^{10} (\bar{Y}^k - \bar{X})(\bar{Y}^k - \bar{X})^T \quad (9)$$

$$A = S_w^{-1} S_B \quad (10)$$

⁵ Cover's Theorem

که w_i بردار وزن مربوط به نرون در لایه رقابتی است و x بردار ویژگی ورودی بوده و $\| \cdot \|$ نرم اقلیدسی می‌باشد. به روزسازی وزن‌ها توسط رابطه (۱۴) بیان می‌شود [۱۲].

$$w_{ij}^s(t+1) = \begin{cases} w_{ij}^s(t) + \eta(t) (x_k^s(t) - w_{ij}^s(t)) & , i, j \in N_c \\ w_{ij}^s(t) & otherwise \end{cases} \quad (14)$$

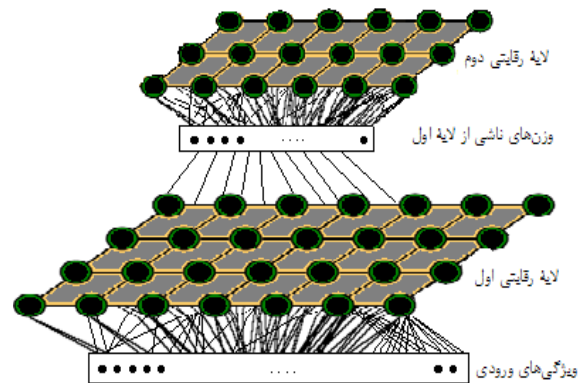
که η عدد ثابت و مثبتی است که در طول زمان کاهش یافته و منجر به همگرایی شبکه می‌گردد و N_c همسایگی در اطراف نرون پیروز C است که این همسایگی با گذشت زمان کاهش می‌یابد. x_k^s ورودی شبکه در لایه S ام بوده و k تعداد مجموع کل داده آموزش را بیان می‌کند. i و j تعداد نرون در لایه S می‌باشد.

بعد از آموزش وزن‌های شبکه در لایه اول، \bar{w}^s به دست آمده و با استفاده از رابطه (۱۵) ورودی شبکه در لایه بعدی ایجاد می‌گردد.

$$in^{s+1} = \bar{w}^s \times y^s \quad s=1,2 \quad (15)$$

در اینجا s معرف لایه کوهون بوده و in بردار ورودی لایه و y خروجی شبکه کوهون بعد از آموزش تمام نرون‌های شبکه در لایه مورد نظر می‌باشد.

برای طبقه‌بندی یا به نوعی خوشه‌بندی هدفمند، اگر تعداد ورودی و ابعاد ویژگی داده‌ها زیاد باشد آنگاه پراکندگی بسیار بالا رفته و به نوعی کلاس‌ها در هم ادغام می‌شوند، بنابراین کوهون تک لایه کارایی بالایی نخواهد داشت. به همین منظور ایده استفاده از لایه‌های بیشتر برای تفکیک‌پذیری اولیه کلاس‌ها مطرح می‌شود. شکل ۲ نمونه‌ای از شبکه کوهون دو لایه را نشان می‌دهد.



شکل ۲. نمایی از ساختار کوهون دو لایه

به خاطر اینکه بعد ویژگی‌های داده ورودی و تعداد داده‌ها نیز بالاست بنابراین در لایه اول تعداد بیشتری نرون در نظر می‌گیریم. تعداد نرون در لایه دوم بر اساس میزان موفقیت نرون‌های پیروز کاهش می‌یابد. وزن‌های لایه اول بعد از آموزش نهایشان که در واقع به نوعی

میانگین ورودی‌هایی هستند که به آن نرون متصل هستند با استفاده از رابطه (۱۵) به عنوان ورودی به نرون‌های لایه دوم متصل می‌گردند باید دقت کرد که این روش به شرطی که تعداد ویژگی‌های ورودی بالا باشد مانند نمونه‌های صوت و تصاویر دارای چندین کلاس، دارای کارایی بالایی است. برای تعداد کم ویژگی و حالتی که تعداد کمی نرون در نظر گرفته شده، این امکان وجود دارد که به جای میانگین‌گیری از داده‌های یک کلاس، از چند کلاس میانگین گرفته شود و در لایه‌های بالاتر کلاس‌ها در هم ادغام شوند، بنابراین تفکیک ورودی‌ها برای شبکه SVM فازی مشکل گردد.

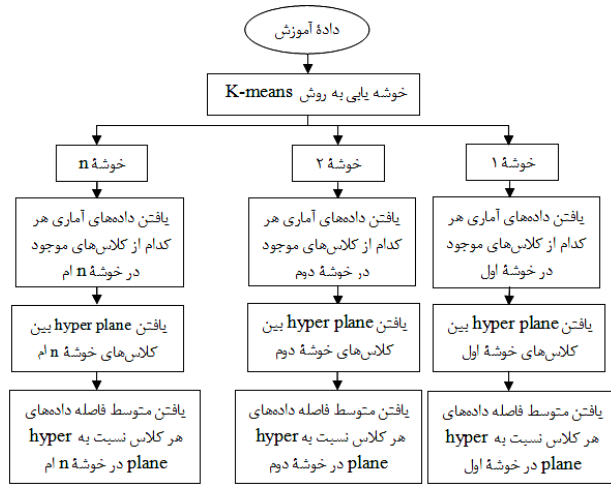
از مزیت‌های این روش نسبت به سایر خوشه‌بندها در این می‌باشد که خوشه‌بندهایی نظیر K-means یا Fuzzy C-means به علت پراکندگی بالای کلاس‌ها، در هر خوشه کلاس‌های متفاوتی موجود می‌باشد، و این سبب می‌گردد که تفکیک کلاس‌ها با خطای بیشتری صورت پذیرد. البته کوهون تک لایه هم در حالتی که پراکندگی کلاس‌ها زیاد باشد همین مشکل را دارد. ولی در خوشه‌بندی به روش کوهون دو لایه تعداد کلاس‌های موجود در هر خوشه خیلی کمتر است و این عمل در تفکیک‌پذیری کلاس‌ها تاثیر بسزایی دارد.

از خاصیت‌های مهم کوهون دو لایه می‌توان به فشردگی داده‌های ورودی اشاره کرد که این فشردگی در حالت دو لایه بیشتر نمود می‌یابد. در حالت ایده‌آل انتظار می‌رود که در طبقه خروجی کوهون دو لایه، هر نرون نمایانگر یکی از کلاس‌ها باشد که البته این امر به دلیل پراکندگی بالای کلاس‌ها و تعداد زیاد داده ورودی تا حدودی دور از انتظار است.

شکل ۳ نرون‌های لایه خروجی کوهون دو لایه را نشان می‌دهد. در این شکل نرون‌های مربوط به هر کلاس با یک رنگ نشان داده شدند. کلاس‌هایی که در کنار هم فعال شدند از لحاظ ساختاری شباهت بیشتری نسبت به هم دارند و می‌توان وزن بیشتری را به این کلاس‌ها اختصاص داد تا تمرکز بیشتری برای جداسازی این کلاس‌ها صورت گیرد. خوشه‌بندی به این روش سبب گردید تا تعداد کلاس‌هایی که در یک خوشه قرار دارند نسبت به حالت K-means کاهش داشته باشد که این عمل هم بر سرعت و هم بر دقت الگوریتم پیشنهادی می‌افزاید.

همانگونه که در شکل ۳ نشان داده شد، برخی از ارقام به دلیل تنوع در شیوه نگارش، به بیش از یک نرون تخصیص یافته‌اند. به عنوان مثال رقم ۴ که با رنگ سبز پر رنگ نشان داده شده در ۳ تا از نرون‌ها فعال شده است. اکثر کلاس‌ها در یک نرون فعال شدند، از جمله کلاس‌های مربوط به ارقام ۱، ۵، ۷ و ۹. در ۷۵٪ از کل نرون‌های خروجی فقط داده‌های مربوط به یک کلاس قرار دارند

این پارامترها در فرایند تست استفاده می‌شود.



شکل ۴. بلوک دیاگرام مربوط به فرایند آموزش در SVM فازی بر مبنای خوشه‌بندی کوهونن دو لایه

۴-۳-۲- توصیف فرایند تست

فرایند تست از ساختار یکی در مقابل دیگری که یکی از ساختارهای SVM چند کلاسی است استفاده می‌کند. در ابتدا با استفاده از رابطه (۱۷) وزن مربوط به این داده تست را با استفاده از پارامترهای آماری هر خوشه که در بخش آموزش به دست آمده بود، محاسبه کرده که درجه عضویت این داده در خوشه نامیده می‌شود.

$$w_n = \frac{1}{\sqrt{2\pi}|\Sigma|^{1/2}} \exp\left(-\frac{(x_t - \bar{x}_n)^T \Sigma_n^{-1} (x_t - \bar{x}_n)}{2}\right) \quad (17)$$

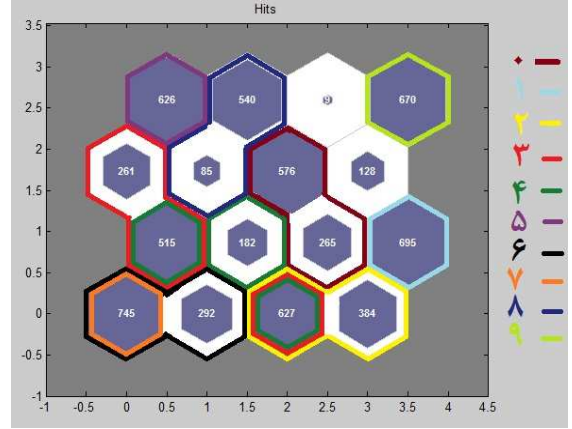
که \bar{x}_n و Σ_n بردار میانگین و کواریانس در i امین خوشه می‌باشند. در این رابطه، n تعداد خوشه و w_n میزان عضویت این داده را در هر کدام از خوشه‌ها نشان می‌دهد. سپس با استفاده از رابطه (۱۸) درصد عضویت این داده تست را در هر کدام از خوشه‌های n نرمالیزه می‌کنیم.

$$\tilde{w}_n = \frac{w_n}{\sum_{m=1}^N w_m} \quad (18)$$

$$C_n^m = \left(\sum_{i=1}^N (x_i - \bar{x}_i^m)^2 \right)^{\frac{1}{2}} \quad (19)$$

$$\tilde{C}_n^m = \frac{C_n^m}{\sum_{k=1}^{10} C_n^k} \quad (20)$$

سپس C_n^m که فاصله اقلیدسی بین داده تست و متوسط داده‌های کلاس n ام که مربوط به خوشه m ام بودند و در مرحله آموزش ذخیره شده بودند را توسط رابطه (۱۹) محاسبه می‌کنیم و توسط



شکل ۳. نرون‌های لایه خروجی برای خوشه‌بندی به روش کوهونن دو لایه

۴-۳-۱- توصیف فرایند آموزش

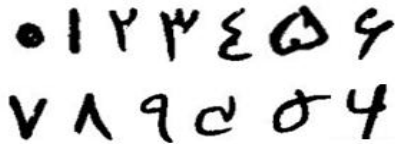
بلوک دیاگرام فرایند آموزش در شکل ۴ آورده شده است. در این فرایند داده‌های آموزشی با استفاده از الگوریتم کوهونن دو لایه به N خوشه تقسیم می‌شوند، سپس ویژگی‌های آماری هر خوشه، یعنی میانگین و کواریانس داده‌های مربوط به هر کلاس ذخیره می‌شود.

از آنجایی که در هم رفتگی بین کلاس‌ها زیاد می‌باشد، بنابراین استفاده از طبقه‌بندی کننده‌ای که بتواند ماکزیمم جداسازی را بین کلاس‌ها را ارائه دهد مناسب خواهد بود. بنابراین استفاده از طبقه بندی کننده SVM نتیجه بهتری را خواهد داد زیرا SVM، حاشیه اطمینان ماکزیمم را بین دو کلاس ارائه می‌دهد. فرایند آموزش در هر خوشه با استفاده از داده‌های آن و از طریق الگوریتم SVM فازی و با تابع هسته کوادراتیک انجام می‌شود. تمام کلاس‌ها به روش یک کلاس در مقابل کلاس دیگر طبقه‌بندی می‌گردند که در هر خوشه حداکثر ترکیب $\binom{N}{2}$ Hyper plane خواهیم داشت. فاصله متوسط داده‌های یک کلاس در هر خوشه را نسبت به Hyper plane مربوط به آن کلاس را محاسبه و ذخیره می‌کنیم. خروجی فرایند آموزش، شامل پارامترهایی از قبیل میانگین، کواریانس، متوسط فاصله داده‌های مربوط به هر کلاس در داخل هر خوشه نسبت به صفحه Hyper plane و وزن‌های مربوط به آن است. این صفحه توسط SVM، ورودی‌ها را برای تفکیک‌پذیری بهتر به داخل فضای ویژگی با ابعاد بالاتری نگاشت می‌دهد که این نگاشت $\phi(x)$ نامیده می‌شود.

تابع تصمیم‌گیری توسط رابطه ۱۶ بیان می‌شود که w بردار نرمال مربوط به Hyper plane و b هم بایاس می‌باشد [۱۱].

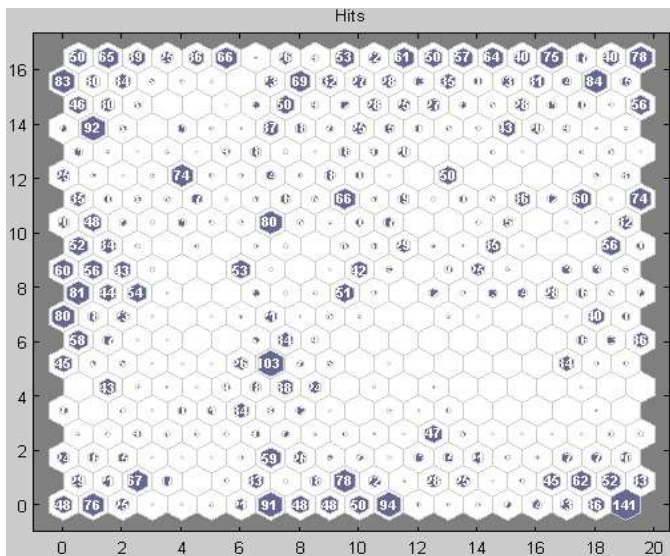
$$f(x) = \text{Sign}(\omega \phi(x) + b) \quad (16)$$

نمونه موجود می‌باشد. نمونه‌ای از ارقام این پایگاه داده در شکل ۶ نشان داده شده است.



شکل ۶. نمونه‌ای از ارقام دستنویس فارسی در پایگاه داده

برای انجام فرایند آموزش از هر رقم ۶۶۰ نمونه و برای فرایند تست نیز ۲۰۰ نمونه از هر رقم مورد استفاده قرار گرفته است. اولین لایه رقابتی در کوهونن شامل $20 * 20$ نرون و لایه دوم نیز که اساس میزان موفقیت لایه نخست تنظیم می‌گردد معمولاً $4 * 4$ نرون را دارد. در خوشه‌بندی به روش کوهونن دو لایه، نرون‌های لایه اول از پراکندگی بالایی برخوردارند. زیرا تعداد کلاس‌ها و پراکندگی داده‌گان و تنوع نگارش این ارقام سبب گردید که نرون‌های یک کلاس در مکان‌های مختلفی فعال گردند. شکل ۷ نرون‌های لایه اول را برای تمام کلاس‌ها در فرایند آموزش نشان می‌دهد. در این شکل تعداد دفعاتی که هر نرون فعال شده است را نشان می‌دهد.



شکل ۷. نرون‌های لایه اول در خوشه‌بندی به روش کوهونن دو لایه

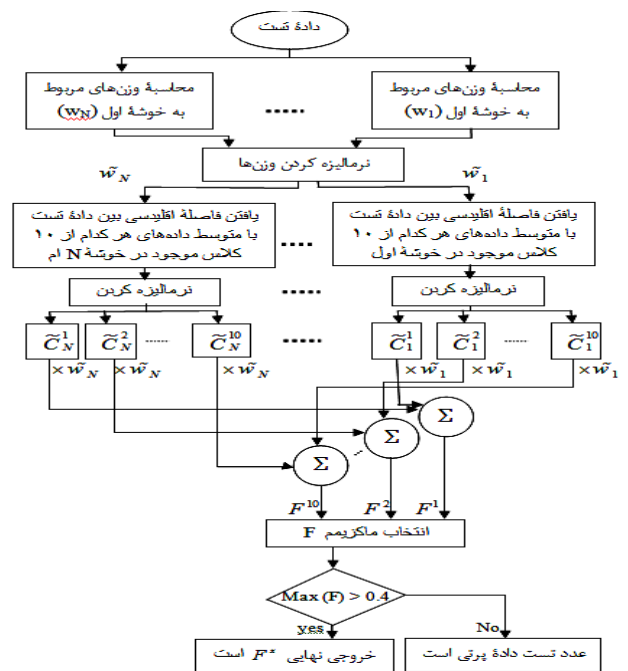
میزان پراکندگی داده‌گان هر کلاس در کوهونن دو لایه بسیار کمتر از کوهونن استاندارد می‌باشد. به عنوان مثال تعداد نرون‌هایی که توسط داده‌های کلاس مربوط به ارقام ۱ فعال شدند در لایه نخست و لایه خروجی در شکل ۸ نشان داده شده است.

رابطه (۲۰) این فواصل را نرمالیزه می‌کنیم. این عمل میزان عضویت داده تست را نسبت به کلاس‌های موجود در هر کدام از خوشه‌ها تعیین می‌کند.

$$m=1,2,\dots,10 \quad (21)$$

$$F^m = \sum_{n=1}^N \tilde{w}_n \tilde{C}_n^m$$

در خوشه‌های مختلف مربوط به یک کلاس می‌باشند را با احتساب درصد عضویت این داده در آن خوشه با هم جمع می‌کنیم. عدد حاصله به صورت نرمالیزه می‌باشد. از بین ۱۰ عدد حاصل، ماکزیم مقدار را انتخاب کرده و با مقدار آستانه که به صورت تجربی به دست آمده مقایسه کرده و در صورتی که از مقدار آستانه کمتر گردد آنگاه به صورت داده پرت شناخته می‌شود. در غیر این صورت به عنوان کلاس پیروز شناخته می‌گردد. شکل ۵ بلوک دیاگرام فرایند تست را نشان می‌دهد.



شکل ۵. بلوک دیاگرام فرایند تست در SVM فازی بر مبنای خوشه‌بندی به روش کوهونن دو لایه

۵- پیاده‌سازی

الگوریتم ارائه شده توسط برنامه MATLAB و جعبه‌ابزار Neural Network پیاده‌سازی شده و با استفاده از پایگاه داده‌ای از ارقام دستنویس فارسی تست شد. این پایگاه داده [۱] و [۲] شامل ۸۶۰۰ نمونه از ده رقم دستنویس فارسی از ۰ تا ۹ می‌باشد و از هر رقم ۸۶۰

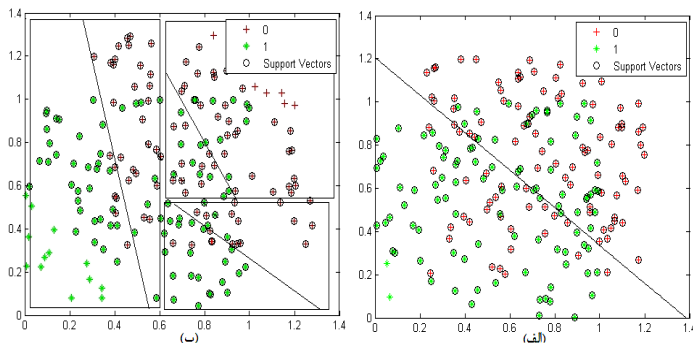
جدول ۳. نرخ بازشناسی ارقام دست‌نویس بر اساس تعداد ویژگی استخراج شده به روش PCA-LDA توسط طبقه‌بندهای مختلف

طبقه بندی کننده	تعداد ضرایب PCA			
	۲۰	۳۰	۴۰	۵۰
TLK-FSVM	۹۵.۶	۹۷.۱	۹۸.۴	۹۸.۱
SVM	۹۲.۵	۹۴.۳	۹۵.۳	۹۳.۵
MLP	۹۱.۳	۹۳.۵	۹۵	۹۶.۲

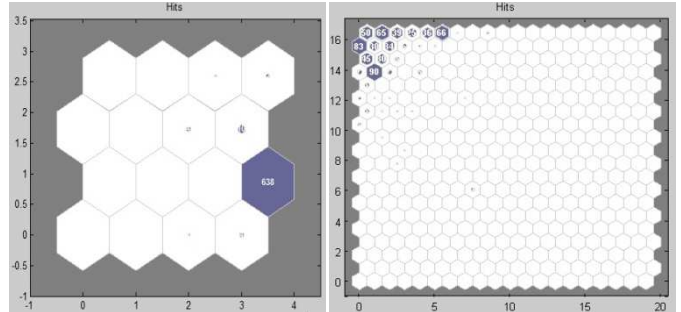
جدول ۴. مقایسه نرخ بازشناسی ارقام دست‌نویس توسط روش پیشنهادی با روش‌های استاندارد بر طبق نوع ویژگی استخراجی

روش استخراج ویژگی	طبقه بندی کننده		
	TLK-FSVM	SVM	MLP
PCA-LDA	۹۸.۴	۹۳.۳	۹۴
PCA	۹۶.۵	۹۱.۴	۹۵.۲

در جدول ۳ نرخ بازشناسی ارقام را با طبقه‌بندی کننده SVM و MLP و روش پیشنهادی خومان را بر حسب تعداد ویژگی‌های آموزش دیده با هم مقایسه کردیم. نتایج نشانگر توانمندی الگوریتم در تفکیک کلاس‌هایی با درهم رفتگی بالا می‌باشد. زیرا همانطوری که در شکل ۹ نشان داده شده است، روش‌مان برای تفکیک کلاس‌هایی با همپوشانی بالا، از چندین hyper plane استفاده کرده و همچنین از ساختار کوهونن چند لایه به منظور تفکیک‌پذیری اولیه بهره می‌برد. که این نیز به نوبه خود موجب قدرتمندی این الگوریتم شده است. در جدول ۴ نیز نرخ شناسایی را با طبقه‌بندهای مختلف بر اساس نوع ویژگی استخراجی با هم مقایسه کردیم. نتایج نشانگر این می‌باشد که استخراج ویژگی به روش PCA-LDA از تفکیک‌پذیری بالاتری برخوردار است. علت این امر در ذات LDA می‌باشد. زیرا PCA جهت بیشترین پراکندگی‌ها را می‌دهد ولی بردارهای LDA جهت بیشترین تفکیک‌پذیری می‌باشند که استفاده از تکنیک PCA در جهت کاهش ابعاد می‌تواند کمک زیادی به تفکیک‌پذیری LDA کند.



شکل ۹. مقایسه بین SVM استاندارد (الف) و SVM فازی (ب) برای حالت دو کلاس با همپوشانی بالا.



شکل ۸. فعال‌سازی نرون‌های لایه اول و دوم کوهونن برای ارقام کلاس یک

به منظور مقایسه الگوریتم پیشنهاد شده با سایر الگوریتم‌ها، دو روش ارائه شده در مراجع [۱] و [۲] نیز پیاده سازی شده‌اند. روش ارائه شده در مرجع [۱] با استفاده از همین پایگاه داده از ترکیب وزن‌دار برای بازشناسی ارقام دست‌نویس استفاده کرده است. در این روش از سه شبکه پرسپترون چند لایه با تعداد متفاوت نرون در لایه میانی به عنوان طبقه بند پایه استفاده شده است. در این مقاله برای تعیین تعداد ضرایب PCA مورد نیاز و تعداد نرون لایه میانی از یک پرسپترون چند لایه استفاده شده است. مرحله آزمایش شامل ۱۰۰ تصویر از هر رقم و مرحله تست نیز شامل ۲۰ تصویر از هر رقم می‌باشد. جدول ۱ نتایج این الگوریتم را نشان می‌دهد.

جدول ۱. تغییرات نرخ بازشناسی با توجه به تغییرات تعداد ضرایب و تعداد نرون‌ها در لایه میانی پرسپترون واحد مربوط به مرجع [۱].

تعداد ضرایب PCA	تعداد نرون در لایه میانی MLP			
	۴۰۰	۶۰۰	۷۰۰	۸۰۰
۱۰	۶۵	۷۳.۱	۷۵.۳	۷۲.۵
۲۰	۵۵.۴	۷۴.۷	۸۰.۱	۷۷.۱
۳۰	۶۵.۲	۷۷.۲	۷۷	۷۸.۲

در مقاله [۲] نیز از همین پایگاه داده استفاده شده است. در این روش که از ۱۰ طبقه بند دو کلاس MLP استفاده کرده است از طبقه بند دو کلاسی در یک سیستم مرکب به عنوان طبقه‌بند پایه استفاده شد. در مرحله آموزش هر شبکه برای بازشناسی یک رقم از سایر ارقام آموزش می‌بیند. جدول ۲ نتایج ناشی از الگوریتم [۲] را نشان می‌دهد.

جدول ۲. تغییرات نرخ بازشناسی با توجه به تغییرات تعداد ضرایب و PCA و تعداد نرون‌ها در لایه میانی پرسپترون واحد مربوط به مرجع [۲].

تعداد ضرایب PCA	تعداد نرون در لایه میانی MLP					
	۱۰	۱۵	۲۰	۲۵	۳۰	۳۵
۱۰	۹۱.۳	۹۲.۵	۹۳.۲	۹۳.۵	۹۳.۹	۹۴
۲۰	۹۲.۲	۹۳.۴	۹۴.۴	۹۵	۹۵.۴	۹۵.۳
۳۰	۹۲.۲	۹۳.۴	۹۳.۳	۹۳.۶	۹۴.۵	۹۴.۱

بحث و نتیجه‌گیری

در این مقاله برای طبقه‌بندی، روش قدرتمندی ارائه گردید که در تفکیک کردن کلاس‌هایی با همپوشانی بین کلاسی بالا، این طبقه‌بند بسیار مناسب می‌باشد. این روش از ساختار کوهونن چند لایه به منظور خوشه‌بندی اولیه استفاده کرده و در نهایت با استفاده از SVM فازی، کلاس‌ها را از هم تفکیک می‌کند. مزیت استفاده از کوهونن چند لایه در خوشه‌بندی هدفمند است تا بتواند کلاس‌هایی با درهم-رفتگی بالا را راحت‌تر تفکیک کند. نتایج نشانگر این می‌باشند که روش پیشنهادی نسبت به روش‌های استاندارد SVM و MLP بهبود قابل ملاحظه‌ای دارد. همچنین نشان دادیم که روش‌مان نسبت به روش‌های [۱] و [۲] که از همین پایگاه داده استفاده کرده بودند بهبود قابل توجهی دارد.

بخشی از خطاها مربوط به تصاویری با دقت تصویر برداری کم و بخش دیگری هم مربوط به نامناسب بودن نمونه‌های نوشته شده و بخشی از آن نیز مربوط به طبقه‌بند می‌باشد، که با افزایش تعداد نمونه‌های آموزشی و تصحیح دستی برخی از ارقامی که حتی توسط انسان هم نمی‌توانند تشخیص داده شوند، می‌توان نرخ بازشناسی را افزایش داد.

مراجع

- [۱] سید حسن نبوی کریزی، رضا ابراهیم پور، احسان الله کبیر، "کاربرد ترکیب طبقه‌بندها در بازشناسی ارقام دستنویس فارسی" سومین کنفرانس ماشین بینایی ایران. دانشگاه تهران، صفحه ۱۱۵ تا ۱۱۹، ۱۳۸۳
- [۲] محمد نحوی، مهدی رفیعی، رضا ابراهیم پور، احسان الله کبیر، "ترکیب طبقه بند دو کلاسی برای بازشناسی ارقام دستنویس فارسی" شانزدهمین کنفرانس مهندسی برق ایران، دانشگاه تربیت مدرس، صفحه ۲۰۳ تا ۲۰۷، اسفند ۱۳۸۷.
- [۳] A. Webb. *Statistical pattern recognition*, 2nd ED., John Wiley & Sons, 2003.
- [۴] B. S. Manjunath, and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 18, No. 8, pp. 837-842, august 1996.
- [۵] H. Al-Usefi And S. Udpa, "Recognition of arabic characters," *IEEE Pattern Analysis and Machine Intelligence*, Vol. 14, No. 8, pp. 853-857, august 1992.
- [۶] M. Khosravi, and E. Kabir, "Introducing a very large dataset of handwritten Farsi digits and a study on their varieties," *Pattern Recognition. Letter*, Vol. 28, No. 10, pp. 1133-1141, feb, 2007.
- [۷]. بحری پیمانی. "شناسایی حروف دستنویس فارسی به کمک شبکه عصبی فازی". پایان نامه کارشناسی ارشد، فتحی محمود، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، ۱۳۷۶
- [۸] Kohonen, T., "Self-Organizing Maps", in *Springer Series in Information Sciences*, vol. 30, 1995.

[۹] T. Kohonen, T. Barna, and R. Chrisley, "Statistical pattern recognition with neural networks: benchmarking studies", in *Proc. ICNN*, vol. 1, pp. 61-68, July, 1988.

[۱۰] M. Turk, and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71-86, march 1991.

[۱۱] S. Abe, *Support vector machines for pattern recognition*, springer, verlog london limited, 2005.

[۱۲] L. Wang, E. Ambikairajah, E. H.C. Choi, "A Comparisonal study of the multi-layer kohonen self-organizing feature maps for spoken language identification," in *Proc. IEEE ASRU workshop*, Japan, pp. 402-407, 2007.