

بهینه‌سازی پارامترها به همراه انتخاب ویژگی برای طبقه‌بند SVM توسط الگوریتم چندهدفه‌ی PSO

ایمان بهروان¹، سید حمید ظهیری²

¹ دانشکده ی برق و کامپیوتر دانشگاه بیرجند، دانشگاه بیرجند، بیرجند، i.behravan@gmail.com

² دانشکده ی برق و کامپیوتر دانشگاه بیرجند، دانشگاه بیرجند، بیرجند، shzahiri@birjand.ac.ir

چکیده

ماشین بردار پشتیبان (SVM) یک طبقه‌بند دو کلاسه است که در سال های اخیر بسیار مورد توجه قرار گرفته است. عملکرد این طبقه‌بند به مقدار پارامترهای آن از جمله پارامتر C (Penalty factor) و پارامتر موجود در کرنل بستگی دارد. همچنین انتخاب تابع کرنل مناسب هم تاثیر بسزایی در بهبود عملکرد آن دارد. در کنار موارد ذکر شده انجام فرایند انتخاب ویژگی نه تنها می تواند باعث بهبود عملکرد طبقه‌بند مذکور شود بلکه باعث کاهش زمان آموزش و پیچیدگی محاسبات نیز می‌گردد. در این پژوهش ما از الگوریتم چندهدفه‌ی PSO (MOPSO) در بهینه‌سازی طبقه‌بند ماشین بردار پشتیبان برای دو تابع هدف نرخ بازشناسی و قابلیت اطمینان استفاده کرده‌ایم.

کلمات کلیدی

ماشین بردار پشتیبان، الگوریتم بهینه‌سازی ازدحام ذرات، بهینه‌سازی چندهدفه، نرخ بازشناسی، قابلیت اطمینان

1- مقدمه

جدید و کارآمدتر در پیش‌بینی ورشکستگی شرکت‌های اقتصادی و صنعتی ارائه کرده‌اند. آن‌ها از الگوریتم ژنتیک تنها برای بهینه‌سازی پارامترهای SVM استفاده کرده‌اند و بر خلاف پژوهش‌های قبلی انتخاب ویژگی در این پژوهش انجام نشده است. همانند الگوریتم ژنتیک، الگوریتم‌های ابتکاری دیگری نیز مانند الگوریتم PSO و SA جهت ارتقای عملکرد SVM در زمینه‌های عملی مختلفی مورد استفاده قرار گرفته‌اند همانند زمینه‌های پزشکی [5-7] و تشخیص چهره [8]. در تمام پژوهش‌های ذکر شده، محققین برای ارزیابی عملکرد روش پیشنهادی تنها از یک تابع برازندگی استفاده کرده‌اند، که غالباً این تابع نرخ بازشناسی است. اما در کنار نرخ بازشناسی محاسبه‌ی قابلیت اطمینان نیز می‌تواند روش مناسبی برای ارزیابی طبقه‌بند ارتقاء یافته باشد. قابلیت اطمینان در واقع میزان اعتبار خروجی طبقه‌بند است. غالباً این دو تابع برازندگی، با هم در تقابل هستند به طوری که در برخی مسائل اگر چه نرخ بازشناسی یک کلاس از داده‌ها بالا است اما قابلیت اطمینان خروجی طبقه‌بند در مورد آن کلاس پایین است و بالعکس. این موضوع در شکل (1) به خوبی نشان داده شده است. با توجه به این شکل نرخ بازشناسی دایره‌های توخالی 100% است در حالی که قابلیت اطمینان برای این کلاس 83% می‌باشد. این اعداد برای دایره‌های توپر به ترتیب 80% و 100% هستند. در این پژوهش ما از الگوریتم بهینه‌سازی چندهدفه‌ی PSO (MOPSO) جهت بدست آوردن ابرصفحه‌های بهینه برای دو تابع هدف یعنی نرخ بازشناسی و قابلیت اطمینان استفاده کرده‌ایم.

SVM یک روش یادگیری با نظارت است که با استفاده از داده‌های آموزش فرایند طبقه‌بندی را انجام می‌دهد [1]. این طبقه‌بند فاصله‌ی بین دو کلاس را حداکثر می‌کند. طبقه‌بند SVM از تابع کرنل برای نگاشت داده‌ها به فضای با ابعاد بی‌نهایت استفاده می‌کند با این هدف که در این فضا داده‌های مورد نظر ساده‌تر جدا شوند. عملکرد SVM در جداسازی داده‌هایی که به صورت خطی جداپذیر نیستند به انتخاب مناسب تابع کرنل و همچنین نسبت دادن مقادیر مناسب به پارامتر موجود در تابع مذکور و پارامتر C، بستگی دارد. علاوه بر این انجام فرایند انتخاب ویژگی هم می‌تواند باعث افزایش دقت بازشناسی طبقه‌بند SVM شود. در نتیجه قبل از آموزش SVM با داده‌های آموزش، کاربر باید یک تابع کرنل مناسب انتخاب کند و مقادیر بهینه‌ای را به پارامترهای ذکر شده نسبت دهد. حل این مساله می‌تواند توسط الگوریتم‌های ابتکاری انجام شود. در [2]، Huang و همکارانش از الگوریتم ژنتیک برای بهینه‌سازی پارامترهای SVM و انتخاب ویژگی به طور همزمان استفاده کرده‌اند. آن‌ها در این پژوهش تنها از کرنل RBF در تمام آزمایش‌های خود استفاده کرده‌اند. در [3]، B. Samanta و همکارانش روشی را تحت عنوان GA-SVM ارائه کرده‌اند که در آن از الگوریتم ژنتیک برای بهبود عملکرد SVM در تشخیص عیب ماشین های چرخان، بهره برده‌اند. در این پژوهش نیز الگوریتم ژنتیک علاوه بر بهینه‌سازی پارامترهای SVM، فرایند انتخاب ویژگی را هم انجام می‌دهد. در [4]، Wu و همکارانش با بهینه‌سازی SVM توسط الگوریتم ژنتیک، روشی

فاصله‌ی ابرصفحه‌ی جداکننده با بردارهای پشتیبان برابر است با: $\frac{2}{\|w\|}$. پس SVM برای یافتن ابرصفحه‌ای که بیشترین فاصله را با بردارهای پشتیبان دارد باید $\|w\|$ را می‌نیمد. برای سادگی $\frac{1}{2}\|w\|^2$ جایگزین $\|w\|$ می‌شود. پس در واقع SVM برای یافتن ابرصفحه‌ی مذکور خود یک مسأله‌ی بهینه‌سازی را حل می‌کند که در آن به دنبال یافتن نقطه‌ای است که $\frac{1}{2}\|w\|^2$ می‌نیمد. اما بر خلاف آن چه در شکل (2) قابل مشاهده است، غالباً داده‌ها به صورت خطی جداپذیر نیستند و دارای همپوشانی می‌باشند. در این موارد به نمونه‌هایی که از مرز مربوط به کلاس خود عبور کرده و در ناحیه‌ی کلاس دیگر قرار گرفته‌اند یک ضریب نسبت داده می‌شود (ζ) که برای سایر نمونه‌ها صفر می‌باشد. در نتیجه مسأله‌ی بهینه‌سازی به فرم زیر تبدیل می‌شود:

$$\text{Min } \frac{1}{2} \|w\|^2 + C \cdot \sum_{i=1}^n \zeta_i \quad (4)$$

به شرطی که:

$$\text{s.t } y_i \cdot (w \cdot x_i + b) \geq 1 - \zeta_i, \zeta_i \geq 0, 1 \leq i \leq N$$

در معادله‌ی (4) ضریب C، ضریب مجازات (Penalty factor) نامیده می‌شود که میزان خطا و فاصله‌ی ابرصفحه با بردارهای پشتیبان را کنترل می‌کند. این مسأله به کمک ضرایب نامعین لاگرانژ حل می‌شود. در نتیجه تابع تصمیم‌گیری خروجی که به نمونه‌ها برچسب می‌زند به صورت زیر است:

$$F(x) = \text{sign} \left(\sum_{i=1}^N \alpha_i \cdot y_i \cdot K(x_i, x_j) + b \right) \quad (5)$$

در رابطه‌ی (5) α_i ها ضرایب نامعین لاگرانژ نامیده می‌شوند و $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ تابع کرنل نام دارد که برابر با حاصل ضرب توابع هسته (φ) می‌باشد. این تابع داده‌ها را به فضایی با ابعاد بی‌نهایت منتقل می‌کند. پس از یافتن ضرایب α_i ضرایب w و b از طریق معادلات (6) و (7) بدست می‌آیند.

$$W = \sum_{i=0}^N \alpha_i \cdot y_i \cdot \varphi(x_i) \quad (6)$$

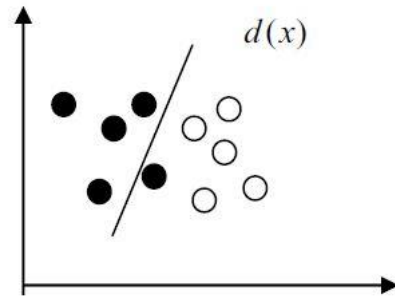
$$b = \frac{1}{N_{SV}} \sum_i (y_i - \sum_j \alpha_j \cdot y_j \cdot K(x_j, x)) \quad (7)$$

در معادله‌ی (7) N_{SV} نمایانگر تعداد بردارهای پشتیبان است و x نمونه‌ی ناشناخته‌ی ورودی است.

3- الگوریتم بهینه‌سازی ازدحام ذرات (PSO)

3-1- بهینه‌سازی تک هدفه

الگوریتم PSO اولین بار در سال 1995 معرفی شد [10]. این الگوریتم الهام گرفته شده از حرکت دسته جمعی پرندگان می‌باشد. یعنی گروهی از ذرات، با الهام از حرکت دسته جمعی پرندگان، در فضای پاسخ به دنبال یافتن جواب بهینه برای مسأله‌ی مورد نظر می‌باشند. موقعیت هر کدام از ذرات، پاسخی برای مسأله می‌باشد. علاوه بر این هر ذره دارای یک حافظه است که بهترین موقعیتی را که از ابتدای اجرای الگوریتم داشته است، در آن ذخیره می‌کند. در هر بار تکرار الگوریتم و جابجایی ذرات، ذره‌ای که بهترین موقعیت را دارد به عنوان رهبر در نظر گرفته می‌شود و سایر ذرات تمایل دارند به موقعیت آن نزدیک شوند. پس جابجایی ذرات تحت تاثیر دو عامل انجام می‌شود: بهترین موقعیت فردی از ابتدای اجرای الگوریتم و موقعیت رهبر در هر تکرار. روابط (8) و (9) به خوبی این مسأله را نشان می‌دهند.



شکل (1) - نرخ بازشناسی برای دایره‌های توخالی و توپر به ترتیب 100% و 80% می‌باشد. قابلیت اطمینان برای این دو کلاس به ترتیب برابر است با 83% و 100%

ساختار کلی مقاله در ادامه به این گونه است: بخش دوم شامل معرفی مختصر طبقه‌بند SVM می‌باشد. در بخش 3 الگوریتم‌های PSO و MOPSO معرفی می‌شوند. در بخش 4 روش پیشنهادی معرفی می‌گردد. پس از آن نتایج بدست آمده در بخش 5 ارائه می‌گردند. بخش 6 نیز متعلق به نتیجه‌گیری است.

2- طبقه‌بند ماشین بردار پشتیبان (SVM)

همان طور که گفته شد SVM یک طبقه‌بند دو کلاسه است [9]. اگر یک مجموعه‌ی شامل N داده‌ی آموزشی داشته باشیم در این صورت هر عضو از این مجموعه را می‌توان به فرم (x_i, y_i) نشان داد که:

$$1 \leq i \leq N, x_i \in R^D, y_i \in \{-1, 1\}$$

در روابط بالا D ابعاد داده‌های آموزشی و y_i کلاس آن‌ها می‌باشد. در این صورت ابرصفحه‌ی جداکننده از رابطه‌ی (1) بدست می‌آید:

$$w \cdot x_i + b = 0, 1 \leq i \leq N \quad (1)$$

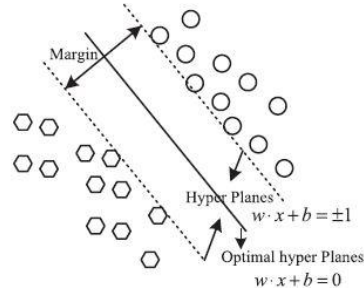
نمونه‌هایی که کمترین فاصله را تا این ابرصفحه دارند، بردار پشتیبان نامیده می‌شوند. برای بردارهای پشتیبان رابطه‌ی (1) به فرم زیر قابل نوشتن است:

$$w \cdot x_i + b = \pm 1 \quad (2)$$

و برای سایر داده‌ها:

$$y_i \cdot (w \cdot x_i + b) \geq 1 \quad (3)$$

پس برای یافتن این ابرصفحه باید ضرایب w و b را یافت که طبقه‌بند SVM این کار را به بهترین نحو انجام می‌دهد. یعنی w و b را بگونه‌ای پیدا می‌کند که ابرصفحه‌ی مذکور بیشترین فاصله را با بردارهای پشتیبان دو کلاس داشته باشد. این مطلب در شکل (2) قابل مشاهده است.



شکل (2) - ابرصفحه‌ی بهینه

$$v_{id}^{t+1} = w.v_{id}^t + c_1 \cdot \text{rand} \cdot (p_{best}^d - x_{id}^t) + c_2 \cdot \text{rand} \cdot (p_{gbest}^d - x_{id}^t) \quad (8)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^t \quad (9)$$

در روابط بالا v_{id} ، w نامین بعد از سرعت نامین ذره است. x موقعیت ذره و t شماره‌ی تکرار را نشان می‌دهند. c_1 و c_2 نیز دو ضریب کنترلی هستند. rand عددی تصادفی در بازه‌ی $[0, 1]$ با توزیع نرمال می‌باشد. p_{best} بهترین موقعیت فردی را از ابتدای تکرار الگوریتم برای هر ذره نگه‌داری می‌کند و p_{gbest} هم موقعیت رهبر را در تکرار فعلی نشان می‌دهد.

3-2- بهینه‌سازی چندهدفه (MOPSO)

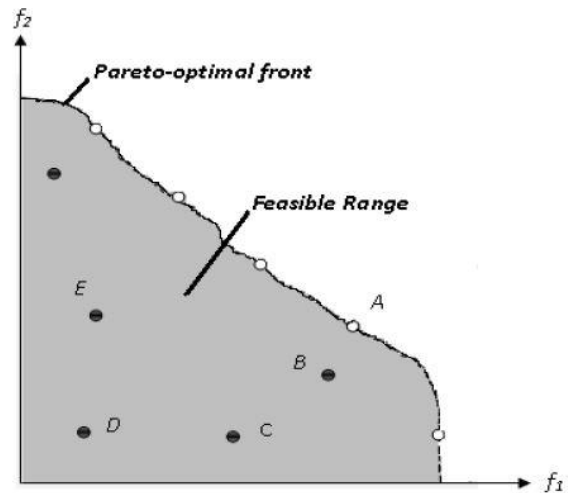
در بهینه‌سازی چندهدفه تعداد توابع هدف بیش از یک است [11]. یک مساله‌ی بهینه‌سازی چندهدفه به فرم زیر تعریف می‌شود:

$$\text{Minimize } F(x) = [f_1(x), f_2(x), \dots, f_k(x)] \quad (10)$$

به شرطی که:

$$\text{s.t } g_j(x) < 0 \text{ and } h_j(x) = 0$$

در روابط بالا $X = (x_1, x_2, \dots, x_n)$ یک جواب مساله، f_i ها توابع هدف و g_j و h_j هم محدودیت‌های مساله می‌باشند. در این گونه مسائل غالباً در بین توابع هدف تقابل وجود دارد یعنی نمی‌توان جوابی پیدا کرد که بهترین جواب برای همه‌ی توابع هدف باشد. می‌گوییم پاسخ x ، پاسخ y را غالب کرده است اگر x در همه‌ی توابع هدف از y بهتر باشد [12]. اما اگر y تنها در یک تابع هدف از x بهتر باشد، در این صورت هیچ کدام از آن‌ها بر دیگری غالب نیست. در این دسته از مسائل غالباً در پایان مجموعه‌ای از جواب‌ها را خواهیم داشت که بر هم غالب نیستند. به این مجموعه جبهه‌ی پرتو گفته می‌شود. در واقع اعضای این جبهه بر دیگر پاسخ‌ها غالب هستند اما بر یکدیگر غالب نیستند. شکل (3) این مفهوم را به خوبی نشان می‌دهد.



شکل (3) - جبهه‌ی پرتو

در الگوریتم MOPSO جواب‌های پرتو در یک مخزن خارجی نگه‌داری می‌شوند [13]. این جواب‌ها در حقیقت خروجی‌های الگوریتم می‌باشند. انتخاب رهبر برای هر ذره از اعضای این مخزن انجام می‌شود. برای انتخاب رهبر، فضای پاسخ توسط ابر مکعب‌هایی پوشیده می‌شود که هر کدام از آن‌ها تعدادی از اعضای جبهه‌ی پرتو را بر اساس مختصات این اعضا، در بر خواهد گرفت. به هر کدام از این ابر مکعب‌ها یک برانزنگی نسبت داده می‌شود که

برابر است با تقسیم تعداد اعضای موجود در ابر مکعب بر یک عدد ثابت. در مرحله‌ی بعد از طریق مکانیزم چرخ رولت یکی از این ابر مکعب‌ها انتخاب شده و در پایان به صورت تصادفی یکی از اعضای آن به عنوان رهبر برای ذره‌ی مورد نظر انتخاب می‌شود. لازم به ذکر است که مختصات اعضاء همان مقادیر توابع برانزنگی آن‌ها می‌باشند. گام‌های زیر الگوریتم MOPSO را توصیف می‌کنند.

- 1- ایجاد جمعیت اولیه و مقداردهی اولیه به سرعت آن‌ها
- 2- ارزیابی ذرات.
- 3- ذخیره‌ی جواب‌های غیر غالب در یک مخزن.
- 4- تولید ابر مکعب‌هایی برای پوشش فضای پاسخ.
- 5- بروزرسانی حافظه‌ی هر ذره.
- 6- حلقه‌ی اصلی
 - محاسبه‌ی سرعت هر ذره.
 - بروزرسانی موقعیت ذرات.
 - ارزیابی ذرات.
 - بروزرسانی مخزن و حافظه‌ی ذرات.
- 7- پایان حلقه‌ی اصلی

4- روش پیشنهادی

در این پژوهش ما از الگوریتم MOPSO برای انتخاب تابع کرنل مناسب، تنظیم پارامتر موجود در تابع کرنل، تنظیم پارامتر C و همچنین انجام فرایند انتخاب ویژگی برای دو تابع هدف یعنی نرخ بازشناسی و قابلیت اطمینان استفاده کرده‌ایم. شکل (4) ساختار ذرات را نشان می‌دهد.

C	KN	σ	F_1	...	F_n
-----	------	----------	-------	-----	-------

شکل (4) - ساختار هر ذره

اولین متغیر، برای تنظیم کردن پارامتر C در نظر گرفته شده است. متغیر KN هم برای انتخاب تابع کرنل مناسب از بین 4 تابع: خطی، Polynomial ، RBF و Quadratic به کار رفته است. متغیر σ مربوط به پارامتر به کار رفته در توابع کرنل (به جز تابع خطی) می‌باشد. سایر قسمت‌های هر ذره برای انتخاب ویژگی در نظر گرفته شده‌اند. برای یک مجموعه داده با n تعداد ویژگی، F_1 تا F_n تعیین کننده‌ی حضور یا عدم حضور ویژگی مورد نظر در مجموعه داده‌ی آموزش هستند. اگر مقدار آن‌ها کمتر یا مساوی 0.5 باشد، ویژگی مورد نظر انتخاب نشده در غیر این صورت ویژگی مذکور برای آموزش SVM انتخاب شده است.

اگر داده‌ها را در دو کلاس "Positive" و "Negative" در نظر بگیریم، در این صورت داده‌های دسته‌بندی شده توسط طبقه‌بند را می‌توان در 4 گروه جای داد.

- 1- داده‌هایی که متعلق به کلاس "Positive" هستند و به درستی دسته‌بندی شده‌اند. (TP)
- 2- داده‌هایی که متعلق به کلاس "Positive" هستند اما به اشتباه به کلاس "Negative" نسبت داده شده‌اند. (FN)
- 3- داده‌هایی که متعلق به کلاس "Negative" هستند و به درستی دسته‌بندی شده‌اند. (TN)

زبان ساده‌تر مقادیر برازندگی برای هر ذره، نرخ باز شناسی و قابلیت اطمینان روی داده‌های آموزش می‌باشد. برای طبقه‌بندی چند کلاسه از روش ((یک در مقابل همه)) استفاده می‌شود. در این روش برای جدا کردن هر کلاس از سایر کلاس‌ها یک ابرصفحه بدست می‌آید و در نهایت با ترکیب نتایج ابرصفحه‌های مربوط به کلاس‌های مختلف به نمونه‌های ناشناخته برچسب زده می‌شود. شکل (5) این روش را برای یک مجموعه داده‌ی 7 کلاسه نشان می‌دهد.

5- آزمایشات انجام شده

روش پیشنهادی روی سه مجموعه داده از کتابخانه‌ی UCI تست شده است [14]. ویژگی‌های این داده‌ها در جدول (1) آورده شده است. جداول (2)، (3) و (4) نیز ابرصفحه‌های بهینه‌ی بدست آمده برای جدا کردن کلاس‌های هر کدام از این سه مجموعه داده را نشان می‌دهند. جدول (5) هم شامل نتایج نهایی می‌باشد.

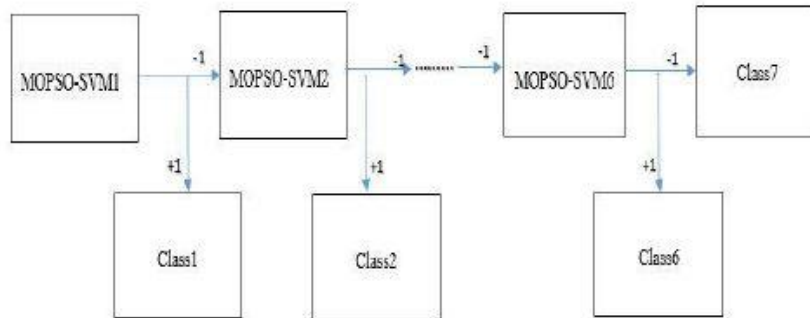
4- داده‌هایی که متعلق به کلاس "Negative" هستند اما به اشتباه به کلاس "Positive" نسبت داده شده‌اند. (FP) با این تفاسیر می‌توان توابع برازندگی را به فرم زیر تعریف کرد:

$$\text{Recognition Score} = \frac{TP+TN}{TP+TN+FN+FP} \quad (11)$$

$$\text{Pos-reliability} = \frac{TP}{TP+FN} \quad (12)$$

$$\text{Neg-reliability} = \frac{TN}{TN+FP} \quad (13)$$

رابطه‌ی (11) بیان‌کننده‌ی نرخ بازشناسی و روابط (12) و (13) به ترتیب بیان‌کننده‌ی قابلیت اطمینان برای کلاس‌های مثبت و منفی می‌باشند. برای محاسبه‌ی مقادیر برازندگی هر ذره، SVM باید با ویژگی‌ها و تابع کرنل انتخاب شده و مقادیر در نظر گرفته شده برای پارامترهای C و σ آموزش دیده و پس از آن مقادیر برازندگی بر طبق روابط (11) تا (13) محاسبه شود. در این مرحله داده‌های آزمایش همان داده‌های آموزش می‌باشند. و در واقع ابرصفحه‌هایی که بهترین عملکرد را روی داده‌های آموزش داشته باشند برگزیده می‌شوند تا در مرحله‌ی بعد در معرض داده‌های جدید قرار بگیرند. به



شکل (4) - دسته‌بندی یک مجموعه داده‌ی 7 کلاسه

جدول (1) - ویژگی‌های مجموعه داده‌های مورد بررسی

مجموعه داده	تعداد کلاس‌ها	تعداد نمونه‌ها	تعداد ویژگی‌ها
Glass	6	214	9
Iris	3	150	4
wine	3	175	13

جدول (2) - نتایج بدست آمده برای مجموعه داده‌ی Glass

کلاس	C	تابع کرنل انتخاب شده	σ	ویژگی‌های انتخاب شده
1	151.3586	polynomial	5.9744	1,3,6,7,8
2	161.0391	polynomial	5.3396	1,2,3,4,6,7
3	176.5744	polynomial	5.9970	1,4,6,8,9
4	141.1287	quadratic	6.1331	1,2,3,4,5,7,8
6	152.5003	quadratic	5.3975	1,3,5,6,,8
7	176.8546	quadratic	2.8688	1,2,3,5,8

جدول (3) - نتایج بدست آمده برای داده‌های Iris

ویژگی‌های انتخاب شده	σ	تابع کرنل انتخاب شده	C	کلاس
1,4	-	خطی	152.2266	1
1,2,3	10.7930	quadratic	159.6416	2
2,3,4	13.3159	quadratic	165.0780	3

جدول (4) - نتایج بدست آمده برای داده‌های Wine

ویژگی‌های انتخاب شده	σ	تابع کرنل انتخاب شده	C	کلاس
1,3,4,6,10,11,12,13	6.4732	quadratic	121.3559	1
1,2,5,6,10,11,13	10.9194	quadratic	142.7297	2
3,6,7,8,9,11	9.7302	quadratic	151.9882	3

جدول (5) - نرخ بازشناسی و میانگین قابلیت اطمینان

توابع برازندگی	Glass	Iris	Wine
نرخ بازشناسی	81.31%	94.67%	97.75%
میانگین قابلیت اطمینان کلاس‌ها	92.94%	97.93%	100%

بدست آمدن قابلیت اطمینان بالا، باعث افزایش اعتماد کاربر به نتیجه‌ی ارائه شده توسط طبقه‌بند می‌شود.

6- نتیجه گیری

در این پژوهش ما از الگوریتم MOPSO جهت بهینه‌سازی SVM برای دو تابع هدف نرخ بازشناسی و قابلیت اطمینان استفاده کرده‌ایم. نتایج بدست آمده در جدول (5)، حاکی از توان بالای MOPSO در حل مسائل چندهدفه است. به خصوص این که خروجی MOPSO یک مجموعه از جواب‌هاست که باعث می‌شود کاربر حق انتخاب داشته باشد. علاوه بر این توان بالای MOPSO در بهینه‌سازی تابع قابلیت اطمینان، نشان می‌دهد که این طبقه‌بند توان بالایی در حل مسائلی دارد که در آن‌ها قابلیت اطمینان هم اهمیت دارد همانند مسائل پزشکی و تشخیص انواع بیماری‌ها، تومورها، تشخیص سیگنال‌های ECG و مسائل نظامی.

مراجع

- [1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer Verlag, 1995
- [2] C.L.Huang, C.J.Wang "A GA-based feature selection and parameters optimization for support vector machines", *Expert Systems with Applications*, Vol.31, pp. 231-240, 2006.
- [3] B.Samanta, K.R.Al-Balushi, S.A.Al-Araimi, "Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection", *Engineering Applications of Artificial Intelligence*, Vol.16, pp. 657-665, 2003
- [4] C.H.Wu, G.H.Tzeng, Y.J.Goo, W.C.Fang, "A real-valued genetic algorithm to optimize the parameters of

جدول (2) شامل بهترین ذراتی است که کلاس مورد نظر خود را در مجموعه داده‌ی Glass، از سایر کلاس‌ها جدا می‌کنند. همان طور که گفته شد، این ذرات از مخزن الگوریتم MOPSO استخراج شده‌اند. در حقیقت برای یافتن هر کدام از این ذرات الگوریتم MOPSO باید یک بار اجرا شود. جداول (3) و (4) نیز همین اطلاعات را به ترتیب در مورد داده‌های Iris و Wine ارائه می‌کنند. با توجه به جدول (3) مشاهده می‌شود که کلاس اول مجموعه داده‌ی Iris با استفاده از یک کرنل ساده‌ی خطی قابل تفکیک است و نیازی به استفاده از توابع کرنل پیچیده نیست. ولی در مورد مجموعه‌ی Glass چنین چیزی وجود ندارد، دلیل آن هم پیچیده‌تر بودن داده‌های Glass نسبت به Iris و Wine می‌باشد. همچنین با توجه به جداول (2)، (3) و (4) می‌توان دریافت که تابع کرنل Quadratic یک تابع کارآمد است که می‌تواند برای مسائل مختلف به کار برده شود. این در حالی است که در اکثر پژوهش‌های معرفی شده در بخش اول مقاله، تنها از کرنل RBF در تمام آزمایشات صورت گرفته، استفاده شده بود و این یعنی کرنل Quadratic می‌تواند عملکردی حتی بهتر از RBF داشته باشد. علاوه بر این با اندکی دقت در این جداول می‌توان به اهمیت فرایند انتخاب ویژگی پی برد. چرا که با بررسی جدول (2) می‌بینیم که از بین 6 توپ بدست آمده برای جدا کردن 6 کلاس مجموعه داده‌ی Glass، 4 توپ دارای 5 ویژگی، یک توپ 6 ویژگی و یک توپ دارای 7 ویژگی است. این در حالی است که مجموعه داده‌ی Glass دارای 9 ویژگی است. این نکته یعنی کاهش قابل توجه حجم محاسبات و همچنین افزایش راندمان سیستم بازشناسی الگو. این مطلب برای مجموعه داده‌ی Wine نیز که دارای 13 ویژگی است، صادق است. جدول (5) هم شامل نرخ بازشناسی و میانگین قابلیت‌های اطمینان کلاس‌های مختلف برای هر مجموعه داده می‌باشد. در تمام آزمایش‌ها نمی‌توانیم از داده‌ها برای آموزش و تمام داده‌ها برای آزمایش به کار رفته‌اند. اعداد و ارقام نشان داده شده در جدول (5) نیز قبل تامل است. چرا که

- support vector machine for predicting bankruptcy”, *Expert Systems with Applications*, Vol. 32, PP. 397–408, 2007.
- [5] F.Melgani, Y.Bazi, “ *Classification of electrocardiogram signals with support vector machines and particle swarm optimization*”, *IEE transactions on information technology in Biomedicine*, Vol.12, NO.5, PP.667-677, September 2008.
- [6] J.Salimi sartahtti, M.H.Zangooei, K.Mozafari, “*Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (SVM-SA)*”, *Computer method and programs in Biomedicine*, Vol.108, pp .570-579, 2012.
- [7] Q.Shen, W.Min shi, W.Kong, B.Ye, “ *A combination of modified particle swarm optimization algorithm and support vector machine for gene selection and tumor classification*”, *Talanta*, Vol. 71,pp. 1679–1683, 2007
- [8] J.Weii, Z.Jiang-qi, Z.Xiang, “ *Face recognition method based on support vector machine particle swarm optimization*” *Expert Systems with Applications*, Vol. 38, pp. 4390–4393, 2011.
- [9] Burgers, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, Vol. 2, pp.121–167.
- [10] Kennedy, J., & Eberhart, R. C. Particle swarm optimization. *Proceedings of IEEE Conference on Neural Network*, Vol. 4, pp. 1942–1948, 1995.
- [11] A. Abraham, L. Jain, and R. Goldberg, *Evolutionary Multiobjective Optimization, Theoretical Advances and Applications*, Springer-Verlag London, 2005.
- [12] V. Pareto, *Cours D’Economie Politique*, volume I and II, F. Rouge, Lausanne, 1896.
- [13] M. Reyes-Sierra, C. A. Coello Coello, “Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art”, *International Journal of Computational Intelligence Research*, ISSN 0973-1873 Vol.2, No.3 (2006), pp. 287–308.
- [14] Hettich, S., Blake, C., & Merz, C. (1998). UCI repository of machine information and computer sciences, available at <http://www.ics.uci.edu/~mllearn/MLRepository.html>.