



حداقل مربعات ماشین بردار پشتیبان دوگانه ساختاری

(*LS-STWSVM: Least Squares Structural Twin Support Vector Machine*)

رامین رضوانی خراشادی زاده¹، رضا منصفی²، هادی صدوقی یزدی³

¹ دانشجوی کارشناسی ارشد، گروه آموزشی کامپیوتر، دانشگاه فردوسی مشهد، مشهد،

raminrezvani@stu-mail.um.ac.ir

² دانشیار، گروه آموزشی کامپیوتر، دانشگاه فردوسی مشهد، مشهد

monsefi@um.ac.ir

³ دانشیار، گروه آموزشی کامپیوتر، دانشگاه فردوسی مشهد، مشهد

[h-sadoghi\(@\)um.ac.ir](mailto:h-sadoghi(@)um.ac.ir)

چکیده

ماشین بردار پشتیبان به عنوان یکی از بهترین طبقه‌بندهای موجود در زمینه یادگیری ماشین نسبت به توزیع داده‌ها اهمیتی قائل نمی‌شود. روش‌های تعمیم‌یافته آن به لحاظ سرعت، روش ماشین بردار پشتیبان دوگانه و به لحاظ دقت و سرعت هم‌زمان، بهبودیافته آن روش ماشین بردار پشتیبان دوگانه ساختاری ارائه شده است. در این مقاله، بکارگیری حداقل مربعات (*least square*) از الگوریتم *S-TWSVM* جهت افزایش سرعت برای طبقه‌بندی‌های باینری پیشنهاد می‌شود. روش پیشنهادی الگوریتمی ساده و سریع، برای ایجاد طبقه‌بندهای باینری بر مبنای دو ابرصفحه غیر موازی فراهم می‌آورد. در این مقاله به جای حل دو مسأله *quadratic programming*، در *S-TWSVM*، دو سیستم از روابط خطی حل می‌شوند. بدین ترتیب سرعت اجرای الگوریتم پیشنهادی از *S-TWSVM* به طور قابل توجهی افزایش می‌یابد. این در حالی است که اطلاعات ساختاری داده‌ها همانند الگوریتم *S-TWSVM* به درون مسأله بهینه‌سازی وارد می‌شود تا ابرصفحه‌های الگوریتم پیشنهادی بتوانند روند توزیع داده‌ها را دنبال کنند. نتایج آزمایشات الگوریتم پیشنهادی روی پایگاه داده‌های استاندارد *UCI* مبین صحت بیشتر طبقه‌بندی این الگوریتم در برابر الگوریتم‌های *S-TWSVM* و *LSTWSVM* و سرعت بالاتر در برابر الگوریتم *S-TWSVM* است.

کلمات کلیدی

اطلاعات ساختاری (*structural information*)، حداقل مربعات (*least squares*)، ماشین بردار پشتیبان دوگانه (*twin support vector machine*)، خوشه‌بندی (*clustering*)، *ward's linkage clustering*، ماشین بردار پشتیبان دوگانه ساختاری (*structural twin SVM*)، حداقل مربعات ماشین بردار پشتیبان دوگانه (*least squares twin SVM*).

از مسائل دنیای واقعی استفاده می‌شود [1, 12]. ایده اصلی در *SVM*، جدا کردن نقاط داده‌ای مثبت و منفی با ترسیم ابرصفحه‌ای با بیشینه حاشیه (*maximum margin*) بین نقاط حاشیه‌ای دو کلاس مثبت و منفی می‌باشد.

1- مقدمه

ماشین بردار پشتیبان یکی از ابزارهای قدرتمند در زمینه طبقه‌بندی داده‌ها می‌باشد و امروزه در حوزه‌هایی مختلفی

2- پیشینه تحقیق

فرض کنید مجموعه داده‌های آموزشی برای دو کلاس داده به صورت زیر باشد:

$$X^{(i)} = [x_1^{(i)}, x_2^{(i)}, \dots, x_{N_i}^{(i)}], i = 1, 2 \quad (1)$$

با N_i نمونه n بعدی در کلاس i به صورتی که ماتریس A با $N_1 \times n$ نمونه (A_i به عنوان i امین نمونه از کلاس $+1$) در کلاس $+1$ و ماتریس B با $N_2 \times n$ نمونه در کلاس -1 می‌باشد، به طوری که ($N_1 + N_2 = N$).

2.1- ماشین بردار پشتیبان دوگانه ساختاری (Structural Twin Support Vector Machine)

در سال‌های اخیر الگوریتم‌های زیادی در زمینه افزودن اطلاعات ساختاری داده‌ها به درون SVM ارائه شده است [2, 3, 4, 13, 14]. در این الگوریتم‌ها اطلاعات ساختاری با کمک تکنیک‌های خوشه‌بندی استخراج می‌شود و با استفاده از ماتریس‌های کوواریانس خوشه‌ها، این اطلاعات به درون مسأله بهینه‌سازی به‌طور مؤثر وارد می‌شود و طبقه‌بندی‌های بهتری شکل می‌گیرد.

در این الگوریتم، از دست‌آوردهای مدل‌های TWSVM استفاده می‌شود بدین صورت S-TWSVM از دو ابرصفحه برای تشخیص دسته داده جدید، استفاده می‌کند. به صورتی که هر مدل در S-TWSVM، فقط اطلاعات ساختاری یک کلاس را در نظر می‌گیرد و به کلاس مزبور نزدیک‌تر می‌شود و همزمان از کلاس دیگر فاصله می‌گیرد. این روش باعث می‌شود که S-TWSVM به‌طور کامل این دانش پیشین را استخراج کند و بدین ترتیب تعمیم‌پذیری الگوریتم افزایش می‌یابد.

این الگوریتم هم مانند الگوریتم‌های مبتنی بر اطلاعات ساختاری شامل دو بخش است: (1) خوشه‌بندی (clustering) (2) یادگیری (learning). در قسمت خوشه‌بندی، اطلاعات ساختاری درون کلاس‌ها بوسیله روش خوشه‌بندی استخراج می‌شود [7]. روش خوشه‌بندی که در اینجا استفاده می‌شود مانند الگوریتم‌های قبلی Ward's linkage clustering (WIL) [7] می‌باشد. در قسمت یادگیری مدل، اگر فرض کنیم برای حالت خطی، S-TWSVM دو ابرصفحه غیرخطی رابطه (8) را تعیین می‌کند:

$$f_+(x) = w_+^T x + b_+ = 0, \quad f_-(x) = w_-^T x + b_- = 0 \quad (8)$$

تاکنون پیشرفت‌های زیادی بر روی SVM انجام شده است. پیشرفت‌هایی در زمینه افزایش سرعت SVM [5] و همچنین برای بهبود صحت طبقه‌بندی و افزایش تعمیم‌پذیری SVM نیز الگوریتم‌های مختلفی ارائه شده است [2, 3, 4, 13, 14]. در [5]، یکی از توسعه‌های SVM برای افزایش سرعت و البته دقت آن، تحت عنوان twin SVM معرفی شده است که دو ابرصفحه غیر موازی را با حل دو مسأله quadratic programming حل می‌کند. ایده TWSVM حل کردن دو مسأله QP با اندازه کوچکتر (به طوری که هر مسأله QP تنها داده‌های یک کلاس را شامل می‌باشد) از یک مسأله QP با اندازه بزرگ (تمام داده‌ها) در SVM متداول، می‌باشد.

در [4]، برای افزایش صحت طبقه‌بندی الگوریتم TWSVM، از ویژگی درون داده‌ها بهره می‌برد. الگوریتم S-TWSVM با افزودن اطلاعات ساختاری داده‌ها (نحوه توزیع داده‌ها درون هر کلاس) به تولید ابرصفحه‌های بهینه کمک می‌کند. این اطلاعات ساختاری توسط تکنیک‌های خوشه‌بندی [7] استخراج می‌شوند و با استفاده از ماتریس‌های کوواریانس کلاس‌ها اطلاعات فوق‌بدست می‌آیند. افزودن اطلاعات ساختاری داده‌ها به مسأله بهینه‌سازی TWSVM سبب افزایش صحت (accuracy) طبقه‌بندی داده‌ها در برابر TWSVM می‌شود.

در این مقاله، نسخه least square از الگوریتم S-TWSVM، تحت عنوان LS-STWSVM (Least Squares Structural Twin SVM) ارائه می‌شود. در الگوریتم پیشنهادی مسائل QP در S-TWSVM، در حالت least square حل می‌شوند و محدودیت‌های مساوی جای قبلی‌های نامساوی در S-TWSVM را می‌گیرند. نتایج الگوریتم پیشنهادی با حل کردن دو سیستم از روابط خطی (در برابر حل دو سیستم درجه دو QP در S-TWSVM) بدست می‌آیند. با انجام آزمایشات و مقایسه الگوریتم پیشنهادی با S-TWSVM و LS-TWSVM، مشاهده می‌شود LS-STWSVM در بیشتر موارد از نظر صحت طبقه‌بندی از دو الگوریتم بهتر و سرعت اجرای این الگوریتم نیز از S-TWSVM بالاتر می‌باشد.

این مقاله به صورت زیر سازماندهی شده است، در قسمت 2، پیشینه تحقیق مورد بررسی قرار می‌گیرد، الگوریتم پیشنهادی این مقاله در قسمت 3 به تفصیل بیان می‌شود، در قسمت 4 نتایج آزمایشات بیان شده است و در نهایت نتیجه‌گیری‌های این مقاله در قسمت 5 آورده شده است.



3- حداقل مربعات ماشین بردار پشتیبان دوگانه ساختاری (Least Squares Structural Twin Support Vector Machine)

در این قسمت نسخه حداقل مربعات از الگوریتم *S-TWSVM* ارائه می‌شود. همان‌طور که گفته شد، *S-TWSVM* از دو مسأله *QP* برای ساخت ابرصفحه‌های غیرموازی استفاده می‌کند. با توجه به این‌که اندازه مسائل *QP* در *S-TWSVM* کاهش یافته است اما این قبیل مسائل از درجه دو هستند و زمان‌بر می‌باشند. در این مقاله، الگوریتمی با عنوان (*Least Squares Structural Twin Support Vector Machine*) یا به اختصار *LS-STWSVM* معرفی می‌شود که به‌جای مسائل *dual QP* در *S-TWSVM*، مسائل *primal QP* از *S-TWSVM* حل می‌شود. با انجام این عمل، یک الگوریتم ساده و سریع تولید می‌شود که تنها به حل روابط خطی برای ساخت طبقه‌بندها نیاز دارد. *LS-STWSVM* نیز همانند *S-TWSVM* شامل دو مرحله می‌باشد. خوشه‌بندی و یادگیری مدل.

3.1 خوشه‌بندی (*clustering*)

در این بخش اطلاعات ساختاری درون کلاس‌ها به‌وسیله روش‌های خوشه‌بندی استخراج می‌شود. متد خوشه‌بندی که در اینجا استفاده می‌شود، مانند الگوریتم‌های قبلی (*Ward's linkage clustering (WIL)*) است. روش *Ward* بدین صورت است که در ابتدا هر نقطه داده‌ای را یک کلاستر در نظر می‌گیرد. سپس بین هر دو کلاستری *Ward linkage* را محاسبه می‌کند. به‌عنوان مثال، اگر *S* و *T* دو کلاستر با میانگین‌های μ_S و μ_T باشند، *Ward's linkage W(S,T)* بین کلاسترهای *S* و *T* به صورت زیر محاسبه می‌شود:

$$W(S, T) = \frac{|S| \cdot |T| \cdot \|\mu_S - \mu_T\|^2}{|S| + |T|} \quad (13)$$

وقتی که دو کلاستر *A* و *B* با هم دیگر ترکیب می‌شوند و یک کلاستر جدید *A'* ایجاد می‌کنند، آنگاه $W(A', C)$ می‌تواند با استفاده از $W(A, C)$ ، $W(B, C)$ و $W(A, B)$ به صورت زیر بدست بیاید:

$$W(A', C) = \frac{(|A|+|C|)W(A, C) + (|B|+|C|)W(B, C) - |C|W(A, B)}{|A|+|B|+|C|} \quad (14)$$

که در آن $w_+, w_- \in R^n, b_+, b_- \in R$ است. در این‌جا هر ابرصفحه به یکی از دو کلاس نزدیکتر است و حداقل به فاصله یک از کلاس دیگر واقع می‌شود. در این الگوریتم هر مدل، فقط اطلاعات ساختاری مربوط به کلاسی که به ابرصفحه نزدیکتر است را در برمی‌گیرد. مدل *S-TWSVM* می‌تواند به‌صورت روابط (9) و (10) فرموله شود: برای کلاس $+1$:

$$\begin{aligned} \min_{w_+, b_+, \xi} & \frac{1}{2} \|Aw_+ + e_+ b_+\|_2^2 + c_1 e_-^T \xi + \frac{1}{2} c_2 (\|w_+\|_2^2 + b_+^2) + \\ & \frac{1}{2} c_3 w_+^T \Sigma_+ w_+, \\ \text{s.t.} & -(Bw_+ + e_+ b_+) + \xi \geq e_-, \xi \geq 0, \end{aligned} \quad (9)$$

و برای کلاس -1 :

$$\begin{aligned} \min_{w_-, b_-, \eta} & \frac{1}{2} \|Bw_- + e_- b_-\|_2^2 + c_4 e_+^T \eta + \frac{1}{2} c_5 (\|w_-\|_2^2 + b_-^2) + \\ & \frac{1}{2} c_6 w_-^T \Sigma_- w_-, \\ \text{s.t.} & (Aw_- + e_- b_-) + \eta \geq e_+, \eta \geq 0, \end{aligned} \quad (10)$$

که $c_1, \dots, c_6 \geq 0$ فاکتورهای جریمه ازپیش تعیین شده هستند. ξ_i و η_i متغیرهای لغزش (*slack variables*) هستند و $\Sigma_+ = \Sigma_{P_1} + \dots + \Sigma_{P_{C_p}}$ ، $\Sigma_- = \Sigma_{N_1} + \dots + \Sigma_{N_{C_N}}$ و Σ_{N_j} به ترتیب ماتریس‌های کوواریانس برای *i*امین و *j*امین کلاستر در دو کلاس می‌باشند و $i = 1, \dots, C_p, j = 1, \dots, C_N$. اگر صفحات جداکننده را به صورت رابطه (8) در نظر بگیریم، یک نقطه داده‌ای جدید $x \in R^n$ ، به کلاس مثبت یا منفی متعلق است و این مبتنی بر این است که این نقطه به کدام یک از دو ابرصفحه، نزدیکتر باشد:

$$f(x) = \arg \min_{+,-} \{d_+(x), d_-(x)\} \quad (11)$$

که

$$d_+(x) = |w_+^T x + b_+|, \quad d_-(x) = |w_-^T x + b_-| \quad (12)$$

که $| \cdot |$ فاصله عمودی نقطه *x* از صفحات $W_+^T x + b_+$ یا $W_-^T x + b_-$ است.

با وزن ξ ، به جای نورم 1 از ξ با وزن c_1 که در رابطه (9) آمده است، استفاده می‌شود. این امر سبب می‌شود که محدودیت $\xi \geq 0$ زائد شناخته شود و به طور ضمنی در تابع هدف قرار گیرد. این تغییر خیلی ساده باعث می‌شود حل مسأله (16) به صورت حل یک سیستمی از روابط خطی نوشته شود. با جایگزینی محدودیت‌های مساوی به درون تابع هدف، مسأله QP (16) به صورت رابطه (17) درخواهد آمد:

$$\min_{w_1, b_1, \xi} \frac{1}{2} \|Aw_1 + e_1 b_1\|_2^2 + \frac{c_1}{2} e_2^T \|Bw_1 + e_2 b_1 + e_2\|_2^2 + \frac{1}{2} c_2 (\|w_1\|_2^2 + b_1^2) + \frac{1}{2} c_3 w_1^T \Sigma_1 w_1 \quad (17)$$

با مشتق گرفتن از رابطه (17) از نظر w_1 و b_1 و قرار دادن آن‌ها برابر صفر، داریم:

$$A^T(Aw_1 + e_1 b_1) + c_1 e_2^T B^T(Bw_1 + e_2 b_1 + e_2) + c_2 w_1 + c_3 w_1 \Sigma_1 = 0 \quad (18)$$

$$e_1^T(Aw_1 + e_1 b_1) + c_1 e_2^T e_2^T(Bw_1 + e_2 b_1 + e_2) + c_2 b_1 = 0 \quad (19)$$

اگر در روابط بالا $c_7 = c_1 e_2^T$ را در نظر بگیریم، با تبدیل روابط (18) و (19) به فرم ماتریسی داریم:

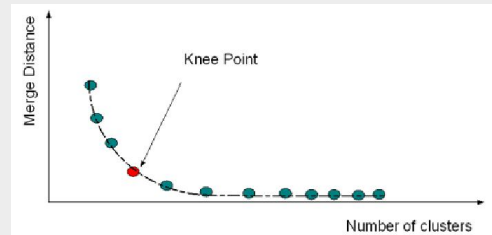
$$\begin{bmatrix} A^T A & A^T e_1 \\ e_1^T A & m_1 \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + c_7 \begin{bmatrix} B^T B & B^T e_2 \\ e_2^T B & m_2 \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} c_2 \\ c_2 \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + c_3 J \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + c_7 \begin{bmatrix} B^T e_2 \\ m_2 \end{bmatrix} = 0 \quad (20)$$

که در نتیجه:

$$\frac{1}{c_7} \begin{bmatrix} A^T A & A^T e_1 \\ e_1^T A & m_1 \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} B^T B & B^T e_2 \\ e_2^T B & m_2 \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \frac{c_2}{c_7} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \frac{c_3}{c_7} J \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} B^T e_2 \\ m_2 \end{bmatrix} = 0 \quad (21)$$

با کمی محاسبات جبری مقادیر w_1 و b_1 مطابق رابطه (22) بدست می‌آیند:

در طول خوشه‌بندی سلسله مراتبی، *Ward's linkage* بین کلاسترهایی که ترکیب می‌شوند افزایش می‌یابد، درحالی‌که تعداد کلاسترها کاهش پیدا می‌کند. یک منحنی به نام *merge distance curve* برای نمایش این فرآیند، در شکل (1) مشاهده می‌شود. با پیدا کردن *knee point* (نقطه زانو) [8] تعداد کلاسترهای بهینه بدست می‌آید و عملیات خوشه‌بندی (*Clustering*) متوقف می‌شود.



شکل (1): منحنی *merge distance curve* و نقطه *knee point* به عنوان تعداد بهینه کلاسترها [8]

3.2- یادگیری مدل (*model learning*) در قسمت اول الگوریتم N و P کلاستر به ترتیب در دو کلاس P و N بدست می‌آیند، به صورتی که:

$$P = P_1 \cup \dots \cup P_i \cup \dots \cup P_{C_p}, N = N_1 \cup \dots \cup N_j \cup \dots \cup N_{C_n} \quad (15)$$

فرض کنید که نقاط متعلق به کلاس مثبت توسط $A \in R^{m_1 \times n}$ ، که هر سطر $A_i \in R^n$ بیان گر یک نقطه داده‌ای است، نشان داده شود. به همین ترتیب، $B \in R^{m_2 \times n}$ تمام داده‌های کلاس منفی را نشان دهد. برای حالت خطی دو ابرصفحه غیرموازی (8) تعیین می‌شوند.

در این قسمت، مسأله (9) از S -*TWSVM*، به صورت رابطه (16) در حالت *least square*، با تبدیل محدودیت‌های نامساوی به محدودیت‌های مساوی به صورت زیر تبدیل می‌شود.

$$\min_{w_1, b_1, \xi} \frac{1}{2} \|Aw_1 + e_1 b_1\|_2^2 + \frac{c_1}{2} e_2^T \xi^T \xi + \frac{1}{2} c_2 (\|w_1\|_2^2 + b_1^2) + \frac{1}{2} c_3 w_1^T \Sigma_1 w_1, \quad (16)$$

$$s.t. \quad -(Bw_1 + e_2 b_1) + \xi = e_2$$

همان‌طور که در رابطه (16) مشاهده می‌شود، در این مسأله QP از مربع نورم 2 از متغیرهای لغزشی ξ ،

و

$$d_+(x) = |w_+^T x + b_+|, \quad d_-(x) = |w_-^T x + b_-| \quad (28)$$

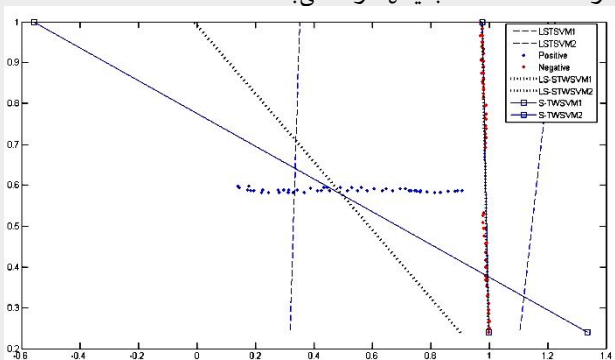
که در این رابطه $| \cdot |$ ، مبدین فاصله عمودی نقطه جدید x از ابرصفحه‌ها می‌باشد.

4- نتایج آزمایشات

قبل از آزمایش الگوریتم‌ها روی مجموعه داده‌های استاندارد مثل UCI ، این روش‌ها روی یک مجموعه داده ساختگی (toy data)، برای درک بهتر خروجی الگوریتم‌ها اجرا می‌شود.

4.1 Toy data

در این قسمت برای مقایسه کارایی سه الگوریتم $LS-TSVM$ ، $S-TWSVM$ و LS ، مجموعه داده ساختگی $the 2-D toy$ بکار گرفته شده است. این مجموعه داده ساختگی شامل دو کلاس مثبت (با ساختار توزیع افقی) و کلاس منفی (با ساختار توزیع عمودی)، همان‌طور که در شکل (2) مشاهده می‌شود، می‌باشد. همان‌طور که گفته شد اطلاعات ساختاری درون کلاس‌ها برای طبقه‌بندی داده‌ها بسیار مهم هستند. از آنجایی که الگوریتم $LSTSVM$ این اطلاعات ساختاری را در نظر نمی‌گیرد، نمی‌تواند به درستی روند داده‌ها را تشخیص داده و طبقه‌بندی ضعیف‌تری ارائه می‌دهد. از طرف دیگر با افزودن این اطلاعات در مسئله بهینه‌سازی $LS-STWSVM$ ، نتایج بهتری از $LSTSVM$ بدست می‌آید. خروجی ابرصفحه‌های دو الگوریتم $S-TWSVM$ و LS ، $STWSVM$ تقریباً مشابه هم‌دیگر می‌باشند، با این تفاوت که سرعت اجرای $LS-STWSVM$ از $S-TWSVM$ بیشتر می‌باشد.



شکل (2) : نتایج اجرای الگوریتم‌های $S-TWSVM$ ، $LSTSVM$ و $LS-STWSVM$ بر روی مجموعه داده ساختگی

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = \left(\begin{bmatrix} \frac{1}{c_7} A' A + B' B & \frac{1}{c_7} A' e_1 + B' e_2 \\ \frac{1}{c_7} e' A + e_2' B & \frac{1}{c_7} m_1 + m_2 \end{bmatrix} + \frac{c_2}{c_7} I + \frac{c_3}{c_7} J \right)^{-1} \begin{bmatrix} -B' e_2 \\ -m_2 \end{bmatrix} \quad (22)$$

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = \left(\frac{1}{c_7} \begin{bmatrix} A' \\ e_1' \end{bmatrix} \begin{bmatrix} A & e_1 \end{bmatrix} + \begin{bmatrix} B' \\ e_2' \end{bmatrix} \begin{bmatrix} B & e_2 \end{bmatrix} + \frac{c_2}{c_7} I + \frac{c_3}{c_7} J \right)^{-1} \begin{bmatrix} -B' e_2 \\ -m_2 \end{bmatrix} \quad (23)$$

با تعریف $E = [A \ e_1]$ و $F = [B \ e_2]$

، مقادیر w_1 و b_1 به صورت رابطه (24) می‌شوند:

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = - \left(\frac{1}{c_7} E' E + F' F + \frac{c_2}{c_7} I + \frac{c_3}{c_7} J \right)^{-1} F' e_2 \quad (24)$$

که $J = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}$ و I ماتریس یکه با ابعاد مناسب می‌باشد. به صورت کاملاً مشابه، با فرض $c_8 = c_4 e_1^T$ راه حل مسأله QP (25) می‌تواند به صورت رابطه (26) نمایش داده شود.

$$\min_{w_-, b_-, \eta} \frac{1}{2} \|B w_- + e_- b_-\|_2^2 + \frac{c_8}{2} e_1^T \eta^T \eta + \frac{1}{2} c_5 (\|w_-\|_2^2 + b_-^2) + \frac{1}{2} c_6 w_-^T \Sigma_- w_- \quad (25)$$

s.t. $(A w_- + e_+ b_-) + \eta = e_+$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = \left(E' E + \frac{1}{c_8} F' F + \frac{c_5}{c_8} I + \frac{c_6}{c_8} K \right)^{-1} E' e_1 \quad (26)$$

که $K = \begin{bmatrix} \Sigma_2 & 0 \\ 0 & 0 \end{bmatrix}$ و I ماتریس یکه با ابعاد مناسب می‌باشد. بعد از آن که توسط روابط (24) و (26)، دو ابرصفحه غیر موازی $x' w_1 + b_1 = 0$ و $x' w_2 + b_2 = 0$ بدست آمدند، برای یک نقطه داده‌ای جدید x ، هرگاه فاصله عمودی این نقطه، تا ابرصفحه $+1$ ، مینیمم باشد، متعلق به کلاس $+1$ خواهد بود و بالعکس. به عبارت دیگر:

$$f(x) = \arg \min_{+,-} \{d_+(x), d_-(x)\} \quad (27)$$

4.2 - UCI datasets

پیشنهادی با وجود سادگی بالا در تولید طبقه‌بندی‌های باینری با استفاده از دو ابرصفحه غیرموازی، صحت (*accuracy*) طبقه‌بندی بالاتری نسبت به *S-TWSVM* و *LSTSVM* و همچنین سرعت اجرای قابل توجهی نسبت به *S-TWSVM* نشان می‌دهد. در الگوریتم پیشنهادی این مقاله، از دو مسأله *primal* از *S-TWSVM*، به‌جای دو مسأله *dual* که در *S-TWSVM* حل می‌شود، استفاده می‌شود.

LS-STWSVM فقط نیاز به حل دو سیستم از مسائل خطی، در مقایسه با *S-TWSVM* دارد. درحالی‌که *S-TWSVM* دو مسأله *quadratic programming (QP)* را علاوه بر دو سیستم از روابط خطی، حل می‌کند. نتایج محاسباتی *LS-STWSVM* بر روی پایگاه داده‌های *UCI*، نشان‌دهنده صحت طبقه‌بندی قابل توجه این الگوریتم در برابر *LSTSVM* و *S-TWSVM* و همچنین سرعت اجرای بالای این الگوریتم نسبت به *S-TWSVM* می‌باشد.

6. مراجع

- C. Cortes, V.N. Vapnik, Support-vector networks, Machine Learning 20 (3) 273-297 (1995).
- D. Yeung, D. Wang, W. Ng, E. Tsang, X. Wang, "Structured large margin machines: sensitive to data distributions", Machine Learning 68 (2) 171-200 (2007).
- H. Xue, S. Chen, Q. Yang, Structural regularized support vector machine: a framework for structural large margin classifier, IEEE Transactions on Neural Networks 22 (4) 573-587, <http://dx.doi.org/10.1109/TNN.2011.2108315> (2011).
- Z. Qi, Y. Tian, Y. Shi, Structural twin support vector machine for classification, Knowledge-Based Systems 43 74-81 (2013).
- Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (5) 905-910 (2007).
- Cover, T. M., & Hart, P. E., Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 21-27 (1967).
- J. H. Ward, Hierarchical grouping to optimize an objective function, Journal of the American Statistical Association 58 (301) 236-244 (1963).
- S. Salvador, P. Chan, Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms, In Proceedings of the 16th IEEE international conference on tools with AI 576-584 (2004).
- F.R. Gantmacher, Matrix Theory, New York, Chelsea, (1990).
- Jayadeva, R. Khemchandani, S. Chandra, Fuzzy multi-category proximal support vector classification via generalized eigenvalues, Soft Computing 11 685-769 (2007).
- G. Rätsch, Benchmark Repository, datasets, <http://icla.first.fhq.de/projects/bench/benchmarks.htm>, (2000).
- N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods Cambridge, U.K Cambridge Univ. Press (2000).
- K. Huang, H. Yang, I. King, and M. R. Lyu, Learning large margin classifiers locally and globally, in Proc. 21st Int. Conf. Mach. Learn. Banff, AB, Canada, pp. 1-8, (2004).
- H. Xue, S. Chen, Q. Yang, Structural support vector machine, in: The 15th International Symposium on Neural Networks, pp. 501-511, (2008).
- Y.-H. Shao, C.-H. Zhang, X.-B. Wang, N.-Y. Deng, Improvements on twin support vector machines, IEEE Transactions on Neural Networks 22 (6) 962-968 (2011).
- Y. Wu, K. Ianakiev & V. Govindaraju, Improved k-nearest neighbor classification. Pattern Recognition 35 2311-2318 (2002).
- Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) 2278-2344 (1998).
- Kumar, M. A., & Gopal, M., Least squares twin support vector machines for pattern classification. Expert Systems with Applications, 36, 7535_7543 (2009).

در این قسمت سه الگوریتم *LSTSVM*، *S-TWSVM* و *LS-STWSVM* روی مجموعه داده‌های *UCI* [11] آزمایش می‌شوند. از این مجموعه داده‌ها، 50 درصد داده‌ها برای آموزش و 50 درصد دیگر برای آزمایش اعمال می‌شوند.

پارامترهای c_1, c_2, \dots, c_8 برای مدل‌ها، از میان مجموعه $\{2^i | i = -7, \dots, 7\}$ ، با استفاده از متد *10-fold cross validation* روی مجموعه داده تنظیم‌کننده، که شامل 10 درصد از مجموعه آموزشی می‌باشد، انتخاب می‌شوند. برای سادگی بیش‌تر $c_1 = c_3, c_2 = c_4, c_3 = c_6$ را در نظر می‌گیریم. نتایج آزمایشات با استفاده از *10-fold cross validation* برای همه مجموعه‌های داده، در جدول (1) نشان داده می‌شوند. در این جدول علاوه بر صحت طبقه‌بندی (*accuracy*) الگوریتم‌ها، زمان اجرای هر کدام نیز آورده شده است.

جدول (1) : صحت طبقه‌بندی‌ها و زمان‌های آموزش روی مجموعه داده *UCI*

Datasets	LS-STWSVM Accuracy Time (s)	LS-TSVM Accuracy Time (s)	S-TWSVM Accuracy Time (s)
Banana	89.26 ± 0.02 1.15	85.00 ± 0.015 0.1	85.74 ± 0.022 7.14
Breast cancer	71.30 ± 0.027 0.52	70.91 ± 0.037 0.004	71.04 ± 0.041 1.93
Diabetis	76.40 ± 0.02 1.21	76.33 ± 0.017 0.007	76.37 ± 0.013 11.42
Flare solar	66.65 ± 0.022 1.79	65.47 ± 0.021 0.009	66.45 ± 0.031 24.59
German	75.97 ± 0.015 1.93	74.90 ± 0.014 0.03	75.83 ± 0.013 40.58
Heart statlog	83.30 ± 0.042 0.50	83.00 ± 0.040 0.003	83.80 ± 0.036 1.87
Ringnorm	75.45 ± 0.012 1.25	74.45 ± 0.019 0.16	75.61 ± 0.010 6.80
Splice	83.05 ± 0.006 3.30	82.77 ± 0.005 0.26	82.11 ± 0.005 80.22

همان‌طور که در جدول (1) مشاهده می‌شود، صحت طبقه‌بندی الگوریتم پیشنهادی نسبت به *LSTSVM* و *S-TWSVM* بیش‌تر است. از طرف دیگر سرعت اجرای *LS-STWSVM* در برابر *S-TWSVM* به‌طور قابل توجهی افزایش یافته است و این به خاطر حل دو مسأله خطی به جای دو مسأله *quadratic programming* در الگوریتم پیشنهادی است.

5. نتیجه‌گیری

در این مقاله الگوریتم *S-TWSVM* ارتقاء داده شد و *LS-least square S-TWSVM (LS-STWSVM)* را ارائه کردیم. الگوریتم