

## ماشین بردار پشتیبان دوگانه ساختاری چگالی گرا

رامین رضوانی خراشادی زاده<sup>1</sup>، رضا منصفی<sup>2</sup>

<sup>1</sup> دانشجوی کارشناسی ارشد، گروه آموزشی کامپیوتر، دانشگاه فردوسی مشهد، مشهد،  
raminrezvani@ymail.com

<sup>2</sup> دانشیار، گروه آموزشی کامپیوتر، دانشگاه فردوسی مشهد، مشهد،  
[Monsefi@um.ac.ir](mailto:Monsefi@um.ac.ir)

### چکیده

اطلاعات ساختاری داده ها، بخشی از دانش و اطلاعات پیشین کم تر توجه شده برای آموزش یک طبقه بند است. یکی از قوی ترین طبقه بندها در حوزه یادگیری ماشین، ماشین بردار پشتیبان (*SVM*) است که با یافتن نقاط حاشیه بین دو کلاس، این دو کلاس را از یکدیگر جدا می کند. *SVM*، به اطلاعات ساختاری و پیشین درون کلاس ها داده توجه ای ندارد و بیش تر توجه آن بر روی جدا سازی بین کلاس ها است. یکی از الگوریتم هایی که بر روی اطلاعات ساختاری توجه نموده، الگوریتم *S-TWSVM* است که از دو ابر صفحه (دو مدل) برای تشخیص کلاس داده ی جدید، استفاده می کند به صورتی که هر مدل، فقط اطلاعات ساختاری یک کلاس را در نظر می گیرد و با دور شدن از کلاس دیگر، به این کلاس مورد نظر نزدیک تر می شود. در الگوریتم فوق کلیه نقاط کلاس مورد نظر با درجه اهمیت یکسانی بررسی می شود، در حالی که ممکن است تعدادی از نقاط آن کلاس از درجه اهمیت بیشتر و تعدادی از آن ها نویز باشند که درجه اهمیت پایینی داشته باشند. الگوریتم پیشنهادی با وزندهی به نقاط نواحی متراکم، توجه الگوریتم به این نقاط را بیشتر و با وزندهی کم تر به نقاط پرت و نویز، اهمیت این نقاط کم تر می نماید به طوری که دقت الگوریتم افزایش می یابد. الگوریتم پیشنهادی بر روی داده های پایگاه *UCI* که دو کلاسی هستند، آزمایش شده است. نتایج حاصل نشان دهنده برتری روش پیشنهادی نسبت به روش همتا، *S-TWSVM* است.

### کلمات کلیدی

ماشین بردار پشتیبان دوگانه (*twin SVM*)، اطلاعات ساختاری (*structural information*)، اطلاعات محلی (*local information*)، نزدیکترین همسایگی (*nearest neighbor*)، طبقه بندهای حاشیه بزرگ *Large margin classifiers*

یکی از قوی ترین طبقه بندها در حوزه یادگیری ماشین، ماشین بردار پشتیبان (*SVM*) [8],[12] است. هدف اصلی *SVM*، یافتن ابر صفحه ای (*hyperplane*) است به طوری که بتواند دو کلاس از داده ها را با حداکثر حاشیه (*maximal margin*) از یکدیگر جدا کند. *SVM*، به اطلاعات ساختاری و پیشین درون کلاس های داده توجه ای ندارد و بیش تر تمرکز آن بر روی جدا سازی بین کلاس ها است.

### 1- مقدمه

اطلاعات ساختاری داده ها، زیر مجموعه ای از دانش و اطلاعات پیشین (*prior knowledge*) برای آموزش یک طبقه بند است. این که چگونه این اطلاعات ساختاری برای ساختن طبقه بندی خوب اعمال شود، موضوع تحقیقاتی جدیدی است که اخیراً بیش تر مورد توجه قرار گرفته است.

دو ابرصفحه (دو مدل) برای تشخیص کلاس داده جدید، استفاده می‌کند به طوری که هر مدل، فقط اطلاعات ساختاری یک کلاس را در نظر می‌گیرد و با دور شدن از کلاس دیگر، به این کلاس مورد نظر نزدیکتر می‌شود. این روش باعث می‌شود که  $S-TWSVM$  به‌طور جداگانه و کامل اطلاعات ساختاری هر کلاس را استخراج کند و بدین ترتیب تعمیم‌پذیری الگوریتم را افزایش می‌دهد. برای حالت خطی،  $S-TWSVM$  دو ابرصفحه غیرخطی را تعیین می‌کند:

$$f_1(x) = w_1^T x + b_1 = 0, \quad f_2(x) = w_2^T x + b_2 = 0$$

رابطه 1- مدل  $S-TWSVM$  برای حالت خطی که  $w_+, w_- \in R^n, b_+, b_- \in R$  است. در اینجا هر ابرصفحه به یکی از دو کلاس نزدیکتر است و حداقل به فاصله یک، از کلاس دیگر قرار دارد. در این الگوریتم هر مدل، فقط اطلاعات ساختاری مربوط به کلاسی که به ابرصفحه نزدیکتر است را دربر می‌گیرد.

مدل  $S-TWSVM$  می‌تواند به صورت زیر فرموله شود:

$$\text{Min}_{w_1, b_1, \varepsilon} \frac{1}{2} \|AW_1 + e_1 b_1\|_2^2 + c_1 e_2^T \varepsilon + \frac{1}{2} c_2 (\|W_1\|_2^2 + b_1^2) + \frac{1}{2} c_3 W_1^T \Sigma_1 W_1$$

$$\text{s.t.} \quad -(BW_1 + e_2 b_1) + \varepsilon \geq e_2, \quad \varepsilon \geq 0$$

رابطه 2- تابع هدف  $S-TWSVM$  برای داده‌های کلاس 1

$$\text{Min}_{w_2, b_2, \eta} \frac{1}{2} \|BW_2 + e_2 b_2\|_2^2 + \frac{1}{2} c_4 e_1^T \eta + \frac{1}{2} c_5 (\|W_2\|_2^2 + b_2^2) + \frac{1}{2} c_6 W_2^T \Sigma_2 W_2$$

$$\text{s.t.} \quad (AW_2 + e_1 b_2) + \eta \geq e_1, \quad \eta \geq 0$$

رابطه 3- تابع هدف  $S-TWSVM$  برای داده‌های کلاس 1

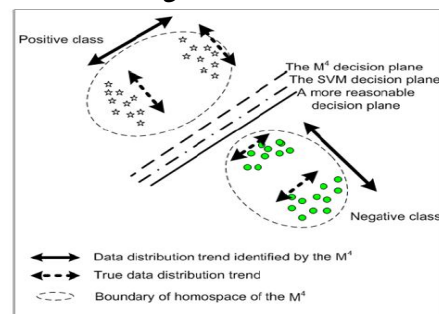
که  $c_1, \dots, c_6 \geq 0$  فاکتورهای جریمه از پیش تعیین شده هستند و  $\xi_i$ ، متغیر لغزش (slack variable) است و  $\Sigma_+ = \Sigma_{P_1} + \dots + \Sigma_{P_{C_p}}$ ،  $\Sigma_- = \Sigma_{N_1} + \dots + \Sigma_{N_{C_N}}$  و  $\Sigma_{N_j}$  به ترتیب ماتریس‌های کوواریانس برای  $i$ -امین و  $j$ -امین کلاستر در دو کلاس هستند ( $i=1, \dots, C_p, j=1, \dots, C_N$ )

### 3- ماشین بردار پشتیبان دوگانه ساختاری چگالی‌گرا Density-Oriented Structural Twin Support Vector Machine (DOS-TWSVM)

در الگوریتم  $S-TWSVM$  هر کدام از ابرصفحه‌ها سعی در نزدیک شدن به نقاط یک کلاس می‌کند و این کار را به وسیله کمینه‌سازی فاصله نقاط آن کلاس تا ابرصفحه انجام می‌دهد. چالشی که در

مهم‌ترین تفاوت میان روش‌های یادگیری (مثل  $SVM$ )، دانه‌بندی‌ای ( $granularity$ ) است که داده‌های آموزشی را ساختاربندی ( $structure$ ) می‌کنند (منظور از دانه‌بندی، کوچکترین واحد است که داده‌ها به نظر می‌آیند توزیع یکسانی دارند که معمولاً این توزیع یکسان، به وسیله ماتریس کوواریانس ( $covariance matrix$ ) اندازه‌گیری می‌شود). این دامنه همگن بودن، ( $homogeneous scope$ )،  $homospace$  نامیده می‌شود. از این رو، همان‌گونه که  $SVM$  به توزیع داده‌ها حساس نیست،  $homospace$  آن، نقاط داده منفرد ( $individual data points$ ) هستند.

الگوریتم  $M^4$  [9]، یکی از الگوریتم‌هایی است که اطلاعات ساختاری را در نظر می‌گیرد. این الگوریتم داده‌های آموزشی هر کلاس را به عنوان یک ساختار واحد در نظر می‌گیرد، لذا تنها دو ساختار متفاوت را بررسی می‌کند. بنابراین  $homospace$  آن، کلاس ( $class$ ) است. بنابراین هر دوی این الگوریتم‌ها ( $M^4$  و  $SVM$ ) نمی‌توانند روند داده‌ها ( $data trend$ ) را تعیین کنند (توصیف کنند) همان‌گونه که در شکل 1 مشاهده می‌شود.



شکل 1- عملکرد الگوریتم‌های  $SVM$  و  $M^4$  [9]

در شکل 1، ملاحظه می‌شود برای نقاط ستاره شکل، روند داده‌ها به صورت عمود بر صفحه تصمیم است در حالی که نقاط دایره شکل، در طرف دیگر ابرصفحه، روند موازی با صفحه جداکننده دارند. بنابراین ما به دنبال الگوریتم‌هایی هستیم که مسائلی از این قبیل را پوشش دهند و بتوانند روند داده‌ها را توصیف کرده و در تصمیم‌گیری به‌کار گرفته و بتوانند صفحه تصمیم منطقی‌تری را ایجاد کنند.

## 2- پیشینه

### 1-2- الگوریتم Structural twin support vector machine for classification

الگوریتم  $S-TWSVM$  [6] از ویژگی‌های دو الگوریتم  $TWSVM$  [3], [11] و الگوریتم  $SRSVM$  [2], [10] بهره می‌برد. به‌صورتی‌که، از

ماتریس وزنی  $W_{ij}$ ، با بهره‌بردن از روش  $KNN[14],[13]$ ، به صورت زیر بدست می‌آید:

$$W_{ij} = \begin{cases} 1 & \text{if } x_j \text{ is } k\text{-nearest neighbors of } x_i \\ 0 & \text{otherwise} \end{cases}$$

رابطه 4- ماتریس وزنی  $W_{ij}$

به عنوان مثال برای نقاط کلاس +1، برای هر جفت از نقاط این کلاس، ماتریس  $W_{ij}$  را ایجاد می‌کنیم و ضریب وزنی  $d_j$  را به صورت زیر محاسبه می‌کنیم:

$$d_j = \sum_{i=1}^{N_1} W_{ij}, j = 1, 2, \dots, N_1$$

رابطه 5- ضریب وزنی برای هر نقطه

$N_1$  تعداد عناصر کلاس +1 می‌باشد.  $d_j$  بیانگر تعداد نقاطی است که  $x_j$  در (k-امین) نزدیکترین همسایگی آنها قرار دارد. هرچه این مقدار  $d_j$  بیشتر باشد،  $x_j$  در نزدیکی نقاط بیشتری قرار دارد، چگالی بیشتری دارد و ضریب وزنی بیشتری به خود می‌گیرد. هرچه نقاط به یکدیگر نزدیکتر باشند (چگال‌تر) این ضریب وزنی  $d_j$ ، برای آنها بزرگتر خواهد بود و این ضریب برای نقاط پرت و نویز دارای مقدار کمی می‌باشد و باعث می‌شود الگوریتم پیشنهادی به این داده‌ها توجه کمتری نشان دهد.

### 3-3- یادگیری مدل

در قسمت اول این الگوریتم  $P$  و  $N$  کلاستر به ترتیب در دو کلاس  $P$  و  $N$  بدست می‌آید، به صورتی که:

$$P = P_1 \cup \dots \cup P_i \cup \dots \cup P_p, N = N_1 \cup \dots \cup N_j \cup \dots \cup N_{C_N}$$

فرض کنید که نقاط متعلق به کلاس

مثبت توسط  $A \in R^{m_1 \times n}$ ، که هر سطر  $A_i \in R^n$  بیانگر یک نقطه داده‌ای است، نشان داده شود. به همین ترتیب،  $B \in R^{m_2 \times n}$  تمام داده‌های کلاس منفی را نشان می‌دهد. برای حالت خطی، دو ابرصفحه غیرموازی تعیین می‌شود:

$$f_1(x) = w_1^T x + b_1 = 0, f_2(x) = w_2^T x + b_2 = 0$$

رابطه 6- ابرصفحه‌های الگوریتم DOS-TWSVM

که  $w_+, w_- \in R^n, b_+, b_- \in R$  است.

مدل DOS-TWSVM می‌تواند به صورت زیر فرموله شود:

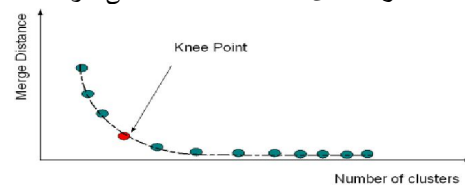
این امر وجود دارد این است که تمام نقاط کلاس مورد نظر با درجه اهمیت یکسانی بررسی می‌شود، درحالی‌که ممکن است تعدادی از نقاط آن کلاس از درجه اهمیت بیشتر و تعدادی از آنها نویز باشند که درجه اهمیت پایینی داشته باشند و نزدیک شدن ابرصفحه به این نقاط کم اهمیت، مطلوب نخواهد بود. بنابراین فقدان اطلاعات محلی (local) در الگوریتم S-TWSVM یکی از چالش‌های موجود در این حوزه است.

### 3-1- استخراج اطلاعات ساختاری درون کلاسها

الگوریتم پیشنهادی شامل 3 بخش است. در بخش اول اطلاعات ساختاری درون کلاسها استخراج می‌شود، در بخش دوم اطلاعات محلی درون داده‌ها و در بخش سوم یادگیری صورت می‌گیرد.

در این قسمت برای استخراج اطلاعات ساختاری داده‌های آموزشی را خوشه‌بندی می‌کنیم. این کار می‌تواند بوسیله الگوریتم‌های مختلفی صورت بگیرد. همانند روش S-TWSVM، الگوریتمی که در این تحقیق استفاده می‌شود، الگوریتم Ward's linkage clustering (WLC)[7]، که یکی از الگوریتم‌های خوشه‌بندی سلسله مراتبی است، می‌باشد.

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 2 مشاهده می‌شود.



شکل 2- منحنی merge distance curve و نقطه knee point به عنوان تعداد بهینه کلاسترها [1]

با پیدا کردن knee point (نقطه زانو) [4] تعداد کلاسترهای بهینه بدست می‌آید و عملیات خوشه‌بندی (Clustering) متوقف می‌شود.

### 3-2- استخراج اطلاعات محلی درون داده‌ها (چگالی نقاط)

در این قسمت برای هر کدام از ابرصفحه‌ها یک گراف درون-کلاسی  $W_{ij}$  ایجاد می‌شود که با دادن هر جفت از نقاط  $(x_i, x_j)$  در هر کدام از کلاسها،

که

$$P=[Ae_2], F=\begin{bmatrix} \Sigma_2 & 0 \\ 0 & 0 \end{bmatrix}, Q=[Be_1]$$

و بردار  $v_2=[w_2b_2]^T$  به صورت زیر بدست می‌آید

$$v_2=-(Q^TKQ+c_5I+c_6F)^{-1}P^T\beta$$

رابطه 12- بردار  $v_2$

هنگامی که بردارهای  $v_1$  و  $v_2$  توسط رابطه‌های فوق بدست آمدند، صفحات جداکننده  $W_1^Tx+b_1=0, W_2^Tx+b_2=0$  ایجاد می‌شوند.

برای نقطه داده‌ای جدید  $x \in R^n$  نزدیکترین ابرصفحه به آن نقطه را بدست می‌آوریم و آن نقطه متعلق به کلاس آن ابرصفحه خواهد بود

$$f(x)=\arg \min_{1,2}\{d_1(x), d_2(x)\}$$

رابطه 13- کلاس نمونه داده جدید  $x \in R^n$  که

$$d_1(x)=|W_1^Tx+b_1|, d_2(x)=|W_2^Tx+b_2|$$

رابطه 14- فاصله داده  $x$  تا ابرصفحه‌ها

که  $|| \cdot ||$ ، فاصله عمودی نقطه  $x$  از

صفحات  $W_1^Tx+b_1, W_2^Tx+b_2$  است.

#### 4-نتایج

در این قسمت دو الگوریتم *S-TWSVM* را با الگوریتم پیشنهادی روی مجموعه داده‌های مختلفی، مقایسه می‌کنیم.

برای سادگی،  $c_1=c_4, c_2=c_5, c_3=c_6$  را در نظر می‌گیریم. صحت آزمایشات توسط *10-fold cross validation* استاندارد، محاسبه می‌شود.

$c_1, c_2, c_3$  همگی از مجموعه  $\{2^i | i=-7, \dots, 7\}$  توسط *10-fold cross validation* روی مجموعه داده تنظیم‌کننده، که شامل 20 درصد از داده‌های آموزشی است، انتخاب می‌شوند. به محض اینکه پارامترها انتخاب شدند، مجموعه تنظیم‌کننده به مجموعه داده‌های آموزشی برمی‌گردد تا تابع تصمیم نهایی استخراج شود.

همه الگوریتم‌ها توسط *Matlab 2011[15]* پیاده‌سازی شده‌اند و محیط آزمایش بدین صورت است: *Intel Core i7-2600 CPU, 4 GB memory*

در این قسمت ما این دو الگوریتم را روی مجموعه داده‌های *UCI[5]* اعمال می‌کنیم. برای هر مجموعه داده‌ای، ما به صورت تصادفی تعداد یکسانی از داده‌های هر کلاس را انتخاب می‌کنیم و یک مجموعه داده می‌سازیم. پنجاه درصد از هر مجموعه داده‌ی استخراج شده، برای آموزش و 50

$$\begin{aligned} \text{Min}_{w_1, b_1, \varepsilon} & \frac{1}{2}D\|AW_1+e_1b_1\|_2^2+c_1e_2^T\varepsilon+\frac{1}{2}c_2(\|W_1\|_2^2+b_1^2)+\frac{1}{2}c_3W_1^T\Sigma_1W_1 \\ \text{st.} & -(BW_1+e_2b_1)+\varepsilon \geq e_2, \varepsilon \geq 0 \end{aligned}$$

رابطه 7: مدل پیشنهادی برای داده‌های کلاس +1

که  $d_j = \sum_{i=1}^{N_1} W_{ij}, j=1, 2, \dots, N_1$  و  $N_1$  تعداد عناصر کلاس +1 می‌باشد و  $D$  فرم ماتریسی  $d_j$  است.

$$\begin{aligned} \text{Min}_{w_2, b_2, \eta} & \frac{1}{2}K\|BW_2+e_2b_2\|_2^2+\frac{1}{2}c_4e_1^T\eta+\frac{1}{2}c_5(\|W_2\|_2^2+b_2^2)+\frac{1}{2}c_6W_2^T\Sigma_2W_2 \\ \text{st.} & (AW_2+e_1b_2)+\eta \geq e_1, \eta \geq 0 \end{aligned}$$

رابطه 8: مدل پیشنهادی برای داده‌های کلاس -1

که  $k_j = \sum_{i=1}^{N_2} W_{ij}, j=1, 2, \dots, N_2$  و  $N_2$  تعداد عناصر کلاس -1 می‌باشد و  $K$  فرم ماتریسی  $k_j$  است.

که  $c_1, \dots, c_6 \geq 0$  فاکتورهای جریمه از پیش تعیین شده‌اند.  $e_+, e_-$  بردارهایی با ابعاد مناسب از مقادیر یک،  $\xi_j$  متغیرهای لغزش،  $\Sigma_+ = \Sigma_{P_1} + \dots + \Sigma_{P_{C_p}}, \Sigma_- = \Sigma_{N_1} + \dots + \Sigma_{N_{C_N}}$  به ترتیب ماتریس‌های کوواریانس  $i$ -امین و  $j$ -امین کلاستر در دوکلاس هستند ( $i=1, \dots, C_p, j=1, \dots, C_N$ )

*Wolfe dual* برای رابطه 7 به صورت زیر بدست می‌آید:

$$\begin{aligned} \text{Max}_{\alpha} & e_2^T\alpha - \frac{1}{2}\alpha^TG(H^TDH+c_2I+c_3J)^{-1}G^T\alpha \\ \text{st.} & 0 \leq \alpha \leq c_1e_2 \end{aligned}$$

رابطه 9- فرم *wolfe dual* برای رابطه 7 که

$$H=[Ae_1], G=[Be_2], J=\begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}$$

و بردار  $v_1=[w_1^Tb_1^T]^T$  به صورت زیر حاصل می‌شود:

$$v_1=-(H^TDH+c_2I+c_3J)^{-1}(G^T\alpha)$$

رابطه 10- بردار  $v_1$

$I$  یک ماتریس یکه با ابعاد مناسب است. به‌طور مشابه فرم دوگانه رابطه 8 به صورت زیر است:

$$\begin{aligned} \text{Max}_{\beta} & e_1^T\beta - \frac{1}{2}\beta^TP(Q^TKQ+c_5I+c_6F)^{-1}P^T\beta \\ \text{st.} & 0 \leq \beta \leq c_4e_1 \end{aligned}$$

رابطه 11- فرم *wolfe dual* برای رابطه 8

درصد برای آزمایش استفاده می‌شوند. فرآیند انتخاب پارامترهای مدل‌ها توسط روش *10-fold cross validation* انجام می‌گردد. نتایج نهایی در جدول 1 نشان داده می‌شود.

جدول 1: نتایج اجرای الگوریتم‌ها روی

مجموعه داده‌های مختلف

Dataset	DOS-TWSVM Accuracy Time(s)	S-TWSVM Accuracy Time(s)
Australian	0.8506 1.1014	0.8359 1.2667
Bupa	0.6138 0.0905	0.5845 0.0905
Diabetes	0.7131 0.9235	0.7037 1.3198
German	0.6958 1.3541	0.6442 1.0951
Heart	0.8271 0.0624	0.8063 0.0624

## 5- نتیجه‌گیری

الگوریتم ماشین بردار پشتیبان یکی از بهترین الگوریتم‌ها برای طبقه‌بندی داده‌ها است و تا کنون توسعه‌های زیادی روی این الگوریتم صورت گرفته است. یکی از جدیدترین دریچه‌هایی که بر روی SVM و سایر الگوریتم‌های طبقه‌بندی باز شده است، استفاده از اطلاعات ساختاری موجود در داده‌های آموزشی در فرآیند یادگیری است که می‌تواند یادگیری را هدفمندتر نموده و دقت طبقه‌بندی را افزایش دهد. در این نوشتار برای افزایش دقت طبقه‌بندی، علاوه بر اطلاعات ساختاری، از اطلاعات محلی نقاط نیز استفاده شده است. بدین ترتیب الگوریتم پیشنهادی نسبت به نقاط نویز (چگالی کم‌تر) از حساسیت پایین و نسبت به نقاط متراکم (چگالی زیاد) حساسیت بالایی دارد. بدین صورت یادگیری به صورت هدفمند انجام می‌شود. برای مقایسه این روش با روش S-TWSVM دو الگوریتم فوق بر روی مجموعه داده‌های پایگاه UCI آزمایش نمودیم به طوری که خروجی نشان‌دهنده برتری روش پیشنهادی نسبت به روش S-TWSVM است.

## ضمایم

در این قسمت برای نقاط کلاس  $(+I)A$  برتری الگوریتم DOS-TWSVM را نسبت به S-TWSVM نشان می‌دهیم. برای نقاط کلاس  $(-I)B$  نیز همین اثبات قابل اعمال است.

در الگوریتم S-TWSVM :

$$\min \frac{1}{2} \|AW_1 + e_1 b_1\|_2^2 + \frac{1}{2} c_2 (\|W_1\|_2^2 + b_1^2) + \frac{1}{2} c_3 W_1^T \sum_1 W_1$$

$$\begin{aligned} \min \frac{1}{2} \|AW_1 + e_1 b_1\|_2^2 + \xi \\ \|AW_1 + e_1 b_1\|_2^2 + \xi < \Omega \\ \|AW_1 + e_1 b_1\|_2^2 < \Omega - \xi \\ AW_1 + e_1 b_1 < \Omega \end{aligned}$$

در الگوریتم DOS-TWSVM :

$$\min \frac{1}{2} D \|AW_1 + e_1 b_1\|_2^2 + \frac{1}{2} c_2 (\|W_1\|_2^2 + b_1^2) + \frac{1}{2} c_3 W_1^T \sum_1 W_1$$

$$\begin{aligned} \min \frac{1}{2} D \|AW_1 + e_1 b_1\|_2^2 + \xi \\ D \|AW_1 + e_1 b_1\|_2^2 + \xi < \Omega \\ \|AW_1 + e_1 b_1\|_2^2 < \frac{\Omega - \xi}{D} \\ AW_1 + e_1 b_1 < \frac{\Omega}{D} \end{aligned}$$

با تعرف شباهت به صورت زیر داریم :

$$\text{Similarity}(A, L_j) = \sum_{x_i \in A} P(x_i) S(x_i, L_j) = P(A) S(A, L_j)$$

$$S(A, L_j) = \exp\left(-\frac{d(A, L_j)}{\lambda}\right)$$

A : نقاط آموزشی کلاس +I

L<sub>1</sub> : ابرصفحه متناظر نقاط A

W<sub>1</sub> : وزن‌های ابرصفحه L<sub>1</sub>

d(A, L<sub>1</sub>) : فاصله نقاط A از ابرصفحه L<sub>1</sub>

در الگوریتم S-TWSVM و DOS-TWSVM :

$$d_{s-twsvm}(A, L_1) = \frac{\|AW_1 + e_1 b_1\|}{\|W_1\|} < \frac{\Omega}{\|W_1\|}$$

$$d_{dos-twsvm}(A, L_1) = \frac{\|AW_1 + e_1 b_1\|}{\|W_1\|} < \frac{\Omega}{D \|W_1\|}$$

$$\frac{d_{s-twsvm}}{\lambda} > \frac{d_{dos-twsvm}}{\lambda}$$

$$\exp\left(-\frac{d_{s-twsvm}}{\lambda}\right) < \exp\left(-\frac{d_{dos-twsvm}}{\lambda}\right)$$

$$S_{s-twsvm}(A, L_1) < S_{dos-twsvm}(A, L_1)$$

$$P(A) S_{s-twsvm}(A, L_1) < P(A) S_{dos-twsvm}(A, L_1)$$

$$\text{Similarity}_{s-twsvm}(A, L_1) < \text{Similarity}_{dos-twsvm}(A, L_1)$$

بنابراین در الگوریتم DOS-TWSVM، ابرصفحه، نقاط آموزشی A را بهتر می‌تواند مدل کند و شباهت داده‌های A با ابرصفحه الگوریتم DOS-TWSVM، بیشتر از ابرصفحه الگوریتم S-TWSVM است.

اگر نزدیکترین نقطه از مجموعه نقاط آموزشی A، به نقطه تست  $a' = (x_1', x_2')$

## مرجع

- [1] 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).
- [2] 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).
- [3] 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)
- [4] Salvador, S., & Chan, P. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Proceedings of the 16th IEEE international conference on tools with AI* (pp. 576–584). (2004)
- [5] A. Asuncion, D. Newman, *UCI Machine Learning Repository* (2007).
- [6] Z. Qi, Y. Tian, Y. Shi, Structural twin support vector machine for classification, *Knowledge-Based Systems*(2013)
- [7] Ward, J. H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58, 236–244. (1963).
- [8] 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).
- [9] 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).
- [10] H. Xue, S. Chen, Q. Yang, Structural support vector machine, in: *The 15th International Symposium on Neural Networks*, pp. 501–511. (2008).
- [11] 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)
- [12] N. Deng, Y. Tian, C. Zhang, *Support Vector Machines Optimization based Theory, Algorithms and Extensions*, Taylor and Francis, (2012).
- [13] Cover, T. M., & Hart, P. E.. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13, 21–27(1967).
- [14] Wu, Y., Ianakiev, K., & Govindaraju, V. Improved k-nearest neighbor classification. *Pattern Recognition*, 35, 2311–2318 (2002).
- [15] <http://www.mathworks.com>, 2007

را نقطه  $a = (x_1, x_2)$  در نظر بگیریم داریم:

$$d(a', L_1) = d(a', a) + d(a, L_1)$$

$$d(a', L_1) = d(a', a) + \frac{|aW_1 + e_1b_1|}{\|W_1\|}$$

$$d_{dos-twsvm}(a', L_1) < \frac{\Omega}{\alpha\|W_1\|} + d(a', a)$$

که  $\alpha$  به عنوان یکی از عناصر قطر اصلی ماتریس  $D$ ، متناظر با نقطه‌ی  $a$  می‌باشد. دقت در الگوریتم‌های *twin SVM* بدین صورت عنوان می‌شود که اگر برچسب واقعی یک داده تست  $a' = (x'_1, x'_2)$ ،  $+1$  باشد آنگاه باید:

$$d(a', L_1) < d(a', L_2)$$

که  $L_1$ : ابرصفحه نقاط  $(+1)A$  و  $L_2$ : ابرصفحه نقاط  $(-1)B$  می‌باشند. رابطه فوق را می‌توان به صورت زیر بازنویسی کرد:

$$\frac{\Omega}{\alpha\|W_1\|} + d(a', a) < d(a', L_2)$$

لذا برای افزایش دقت الگوریتم، باید سمت چپ نامساوی فوق کوچکتر شود. برای نشان دادن این امر، حالات مختلف را بررسی می‌کنیم.

**حالت اول:** نقطه تست  $a'$  در نزدیکی نقاط متراکم  $a$  قرار گرفته است. در این حالت ضریب  $\alpha$  برای نقطه  $a$ ، بزرگ است و  $d(a', a)$  کوچک می‌باشد لذا سمت چپ نامساوی کاهش می‌یابد و بدین صورت دقت افزایش می‌یابد.

**حالت دوم:** نقطه تست  $a'$  در فاصله نسبتاً دوری از نقاط متراکم  $a$  قرار گرفته است. در این حالت ضریب  $\alpha$  برای نقطه  $a$ ، بزرگ می‌باشد ولی  $d(a', a)$  افزایش یافته است. اما در مجموع، با توجه به میزان تراکم نقاط در اطراف نقطه  $a$ ، سمت چپ نامساوی کاهش می‌یابد و کلاس نقطه  $a'$  به درستی تشخیص داده می‌شود.

برای سایر حالات که نقطه تست  $a'$  در نزدیکی یا فاصله دوری از نقاط پرت و کم تراکم قرار گرفته باشد، ضریب  $\alpha$  برای آن نقاط پرت، یک می‌باشد و عملکرد این الگوریتم مشابه با عملکرد الگوریتم *S-TWSVM* برای این نقاط می‌باشد.