ماشین های بردار پشتیبان

Support Vector Machine

پژوهشگر: مهدی غفاری

استاد: فاطمه شریفی زاده

واحد درسی: هوش مصنوعی

دی ۱۳۹۵

چکیده

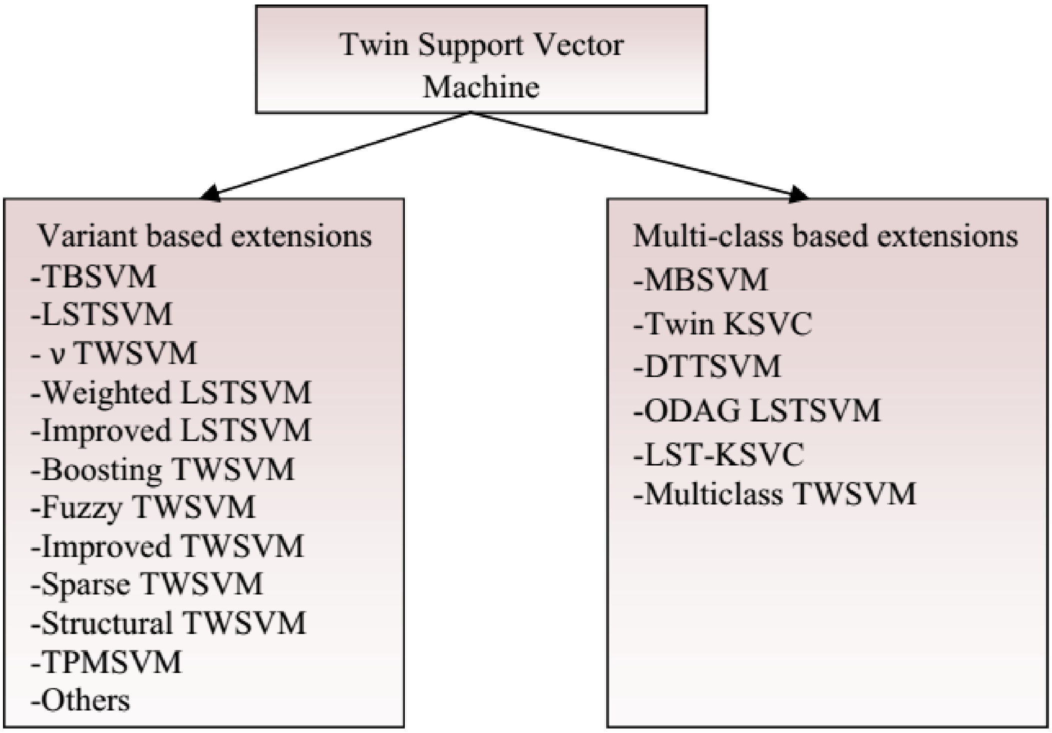
در این تحقیق هدف ما معرفی روش ماشین های بردار پشتیبان بعنوان یکی از شاخه های مورد مطالعه در زمینه هوش مصنوعی است. سپس روش ماشین های بردار پشتیبان حداقل مربعات ($:Least\ squares\ support\ vector\ machine\ LSSVM$) و ماشین های بردار پشتیبان دوقلو ($TSVM: TwinSupport\ Vector\ Machine$) که بعنوان 2 متد تکامل یافته این روش را معرفی میکنیم. پس از معرفی سیر تکاملی روش ماشین های بردار پشتیبان گفته شده و بیان روابط ریاضی آنها به مقایسه عملکرد هر 3 روش برای طبقه بندی چندکلاسه می پردازیم. برای بررسی عملکرد آنها از دیتابیس های استاندارد UCI استفاده خواهیم کرد. برای پیاده سازی این 3 روش از نرم افزار متلب R2014b استفاده خواهیم کرد و نتایج را بصورت جدول هایی جهت مقایسه عملکرد هر یک از روش های بیان شده ارایه خواهیم داد.

معرفی

ماشین بردار پشتیبانی یکی از روش‌های یادگیری بانظارت است که از آن برای طبقه‌بندی خطی و غیرخطی و نیز رگراسیون چندبعدی استفاده می‌کنند. این روش از جمله روش‌های نسبتاً جدیدی است که در سال‌های اخیر کارایی خوبی نسبت به روش‌های قدیمی‌تر برای طبقه‌بندی از جمله شبکه‌های عصبی از خود نشان داده است. اساس کاری کلاس بندی کنندهٔ SVM کلاس‌بندی خطی داده‌ها است و در کلاس بندی خطی داده‌ها خطی را انتخاب کنیم که حاشیه اطمینان بیشتری داشته باشد. مساله پیدا کردن خط بهینه برای کلاس بندی داده‌ها به وسیله روش‌های QP که در حل مسائل محدودیت‌دار شناخته شده هستند صورت می‌گیرد. برای اینکه ماشین بتواند داده‌های با پیچیدگی بالا را نیز دسته‌بندی کند داده‌ها را به وسیله یک تبدیل کرنل به فضای با ابعاد خیلی بالاتر می برند. برای اینکه بتوانیم مسئله ابعاد خیلی بالا را با استفاده از این روش‌ها حل کنیم از قضیه دوگان لاگرانژ برای تبدیلِ مسئله‌ی مینیمم‌سازی به فرم دوگان آن استفاده می‌کنیم. از توابع هسته مختلفی از جمله هسته‌های نمایی، چندجمله‌ای و سیگموید می‌توان استفاده نمود. الگوریتم SVM، جز الگوریتم های تشخیص الگو دسته بندی می شود. از الگوریتم SVM، در هر جایی که نیاز به تشخیص الگو یا دسته بندی اشیا در کلاس های خاص باشد می توان استفاده کرد[1]. روش های ماشین های بردار پشتیبان حداقل مربعات و ماشین های بردار پشتیبان دوقلو نسخه های بهبود یافته از نظر سرعت و عملکرد این متد هستند که در بخش های زیر آن ها را بصورت کامل بیان خواهیم کرد.

پژوهش های انجام شده در گذشته

این الگوریتم اولین بار در سال 1963 میلادی بوسیله (Vladimir N.Vapnik وAlexey Ya. Chervonenkis) ابداع شد و در سال 1992 میلادی توسط (Bernhard E. Boser ، Isabelle M. Guyon و Vladimir N. Vapnik) با استفاده از ترفند کرنل(هسته) برای ساخت کلاس بندی کننده غیرخطی به کمک ابرصفحه ها، پیشنهاد شد. نسخه استاندارد که امروزه مورد استفاده قرار میگیرد در سال 1993 بوسیله (Corinna Cortes و Vladimir N. Vapnik) پیشنهاد و در سال 1995 به چاپ رسید[2]. در سال 1999 میلادی روش ماشین های بردار پشتیبان حداقل مربعات توسط (Suykens و همکاران) پیشنهاد شد که از نظر زمان و عملکرد نتایج بسیار جالبی را ارایه میداد[3]. در سال 2007 روش ماشین های بردار پشتیبان دوقلو توسط (Jayadeva و همکاران) معرفی و با عملکرد آن با روش های رایج کلاس بندی مقایسه و برتری های این روش نشان داده شد[4]. دو سال بعد از ارایه این روش نسخه ترکیبی 2 روش ماشین های بردار پشتیبان حداقل مربعات و دوقلو با عنوان ماشین های بردار پشتیبان دوقلوی حداقل مربعات بوسیله (M. Arun Kumar, M. Gopal) ارایه شد که نتایج بهتری نسبت به روش های گفته شده ارایه داد[5]. نسخه های مختلفی از روش ماشین های بردار پشتیبان دوقلو از سال 2007 تا 2016 براساس همین روش با اندکی تغیر معرفی و گاهی برای داده های خاصی نتایج بهتری را ارایه دادن که در شکل زیر تعدادی از آنها نشان داده شده است.



در ادامه هریک از متدها بصورت کامل توضیح داده خواهند شد.

کلاس بندی کننده ی (SVM)

منظور ما از کلاس بندی کننده در واقع یافتن خطی بصورت $w.x + b = 0$ است که بتوانید بدرستی داده های 2 یا چند کلاس را از یکدیگر جدا کند. از آنجایی که لفظ خط بیشتر یادآور صفحه یا در بهترین حالت خم سه بعدی است برای داده هایی با ابعاد بالاتر باید از مفهوم کلی تری استفاده شود از لفظ ریاضی ابرصفحه به جای آن استفاده خواهیم کرد. در واقع به زبان ریاضی ابرصفحه زیرفضایی از فضای مورد مطالعه است که فقط به اندازه یک بعد با فضای مورد مطالعه تفاوت دارد، بعنوان مثلا یک صفحه 2 بعدی را بعنوان فضا در نظر بگیریم ابرصفحه متناظر با آن یک خط معمولی و برای فضای 3 بعدی ابرصفحه ما یک صفحه 2 بعدی است و به همین ترتیب برای ابعاد بالاتر نیز این مفهوم تعمیم میابد. شکل زیر را در نظر بگیرید.

ما مجوعه داده های آزمایش D شامل n عضو(نقطه)را در اختیار داریم که به صورت زیر تعریف می شود:

$$\mathrm{D} = \left\{ (\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1,1\} \right\}_{i=1}^{n}$$

جایی که مقدار y برابر ۱ یا -۱ و هر $\mathbf{x}_i$ یک بردار حقیقی-p بعدی است. هدف پیدا کردن ابرصفحه جداکننده با بیشترین فاصله از نقاط حاشیه ای است که نقاط با $y_i = 1$ را از نقاط با $y_i = -1$ جدا کند. هر ابر صفحه می تواند به صورت مجموعه ای از نقاط $\mathbf{x}$ که شرط زیر را ارضا می کند نوشت:

$\mathbf{w} \cdot \mathbf{x} - b = 0,$ جایی که . علامت ضرب است $\mathbf{w}$.بردار نرمال است، که به ابرصفحه عمود است. می خواهیم $\mathbf{w}$ و $\mathbf{b}$ را طوری انتخاب کنیم که بیشترین فاصله بین ابر صفحه های موازی که داده ها را از هم جدا می کنند، ایجاد شود. این ابرصفحه ها با استفاده از رابطه زیر توصیف می شوند.

اگر داده های آموزشی جدایی پذیر خطی باشند، می توانیم دو ابر صفحه $\mathbf{w}\cdot\mathbf{x}-b=1$ و $\mathbf{w}\cdot\mathbf{x}-b=-1.$ در حاشیه نقاط به طوری که هیچ نقطه مشترکی نداشته باشند، در نظر بگیریم و سپس سعی کنیم، فاصله آنها را، ماکسیمم کنیم. با استفاده از هندسه، فاصله این دو صفحه $\frac{2}{\|\mathbf{w}\|}$است. بنابر این ما باید $\|\mathbf{w}\|$ را مینیمم کنیم. برای اینکه از ورود نقاط به حاشیه جلو گیری کنیم، شرایط زیر را اضافه می کنیم: برای هر i و به ازای هر $\mathbf{x}_i$ متعلق به کلاس اول داشته باشیم $\mathbf{w}\cdot\mathbf{x}_i-b\geq 1$ یا به ازای هر $\mathbf{x}_i$ متعلق به کلاس دوم داشته باشیم $\mathbf{w}\cdot\mathbf{x}_i-b\leq -1$

می توان آن را به صورت زیر نیز نوشت:

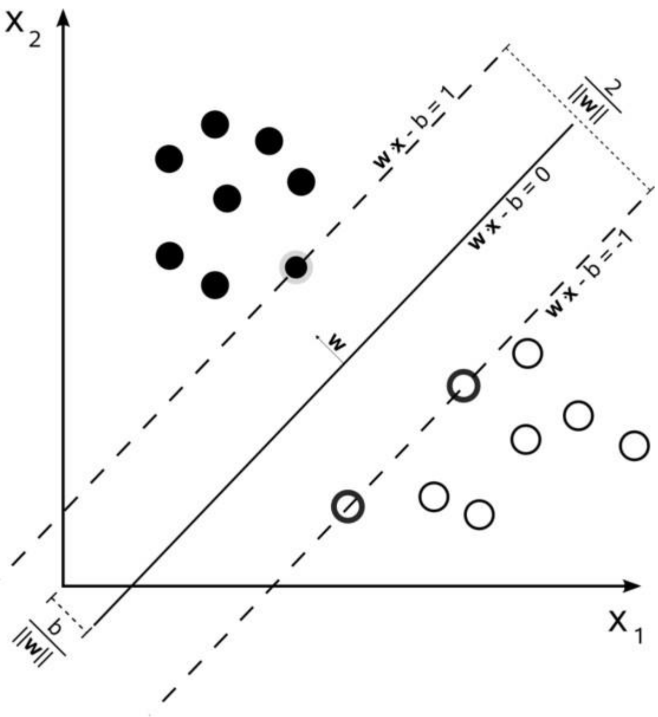
$$y_i(\mathbf{w}\cdot\mathbf{x}_i-b)\geq 1, \qquad 1\leq i\leq n. \qquad (1)$$

با کنار هم قرار دادن این دو یک مسئله بهینه سازی به دست می آید:

Minimize ($\mathbf{w},b$) $\|\mathbf{w}\|$

subject to

$(i=1,\ldots,n)\ y_i(\mathbf{w}\cdot\mathbf{x_i}-b)\geq 1.$



مسئله بهینه سازی ، مسئله سختی برای حل کردن است، زیرا به ||w||وابسته است، بدون تغییر در مسئله ||w||را با $\frac{1}{2}\|\mathbf{w}\|^2$ جانشین کنیم(عبارت ½ برای آسودگی در محاسبات ریاضی آورده شده). و آن را به برنامه ریزی غیرخطی(QP) تبدیل کنیم، یعنی:

Minimize ($\mathbf{w}$,b) $\quad \frac{1}{2}\|\mathbf{w}\|^2$

subject to

$$(i=1,\dots,n)\ y_i(\mathbf{w}\cdot\mathbf{x_i}-b)\ge 1.$$

می توان این عبارت را با استفاده از ضرایب نا منفی لاگرانژ به صورت زیر نوشت که در آن α_i ضرایب لاگرانژ هستند.

$$\min_{\mathbf{w},b}\max_{\boldsymbol{\alpha}}\{\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{n}\alpha_i[y_i(\mathbf{w}\cdot\mathbf{x_i}-b)-1]\}$$

حالا می توان این مسئله را به کمک برنامه ریزی غیرخطی استاندارد حل کرد. جواب این مساله بهینه سازی می تواند به صورت ترکیب خطی از بردارهای پشتیبان بیان شود :

$$\mathbf{w}=\sum_{i=1}^{n}\alpha_i y_i \mathbf{x_i}$$

در این مساله تعداد محدودی از α_i ها بزرگتر از صفر خواهند بود $\mathbf{x_i}$ متناظر، دقیقاً همان بردار پشتیبان خواهد بود. با توجه به این واقعیت می توان نتیجه گرفت که بردارهای پشتیبان شرط زیر را نیز ارضا می کنند:

$$y_i(\mathbf{w}\cdot\mathbf{x_i}-b)=1$$

که اجازه می دهد مقدار b به صورت زیر تعریف شود که در آن N_{SV} تعداد بردارهای پشتیبان است.

$$b=\frac{1}{N_{SV}}\sum_{i=1}^{N_{SV}}(\mathbf{w}\cdot\mathbf{x_i}-y_i)$$

فرم دوگان مساله

استفاده از این واقعیت که $\|\mathbf{w}\|^2=w\cdot w$ و جانشینی $\mathbf{w}=\sum_{i=1}^{n}\alpha_i y_i \mathbf{x_i}$ می توان نشان داد که دوگان SVM به مسئله بهینه سازی زیر ساده می شود:

Maximize (α_i)

$$\tilde{L}(\boldsymbol{\alpha})=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j)$$

$$(i=1,\dots,n)\ \alpha_i\ge 0,$$

$$\sum_{i=1}^{n}\alpha_i y_i=0.$$

چیزی که بیان کردیم ساده ترین حالت ممکن برای بردارهای پشتیبان است و عملا بیشتر مناسب کلاس بندی هایی است که از یکدیگر براحتی بوسیله یک ابرصفحه از هم جدا می شوند. حال فرض کنیم که داده های ما جدایی پذیر نباشند و بعبارت ساده تر یک ابرصفحه که قادر باشد همه داده های 2 کلاس را از یکدیگر جدا کند وجود نداشته باشد، در این حالت از متغیر کمکی ξ_i و ترفند تابع هسته یا همان کرنل استفاده میکنیم و داریم:

$$\begin{cases} y_i\left[w^T\phi(x_i)+b\right]\ge 1-\xi_i, & i=1,\dots,N,\\ \xi_i\ge 0, & i=1,\dots,N. \end{cases}$$

براساس قاعده کاهش ریسک ساختاری، حاشیه های ریسک بوسیله مساله مینیم سازی زیر کاهش میابد:

$$\min J_1(w,\xi)=\frac{1}{2}w^T w+c\sum_{i=1}^{N}\xi_i,$$

$$\text{Subject to }\begin{cases} y_i\left[w^T\phi(x_i)+b\right]\geq 1-\xi_i, & i=1,\dots,N,\\ \xi_i\geq 0, & i=1,\dots,N,\end{cases}$$

تابع لاگرانژ را بصورت زیر تشکیل میدهیم:

$$L_1(w,b,\xi,\alpha,\beta)=\frac{1}{2}w^T w+c\sum_{i=1}^{N}\xi_i+\sum_{i=1}^{N}\alpha_i\left\{y_i\left[w^T\phi(x_i)+b\right]-1+\xi_i\right\}+\sum_{i=1}^{N}\beta_i\xi_i,$$

که در آن $\alpha_i\geq 0,\beta_i\geq 0\ (i=1,\dots,N)$ ضرتیب لاگرانژ هستند. نقطه بهینه این مساله متناظر با نقطه گره زینی تابع لاگرانژ ما خواهد شد، آنگاه داریم

$$\begin{cases} \dfrac{\partial L_1}{\partial w}=0 & \rightarrow\quad w=\sum_{i=1}^{N}\alpha_i y_i\phi(x_i),\\ \dfrac{\partial L_1}{\partial b}=0 & \rightarrow\quad \sum_{i=1}^{N}\alpha_i y_i=0,\\ \dfrac{\partial L_1}{\partial \xi_i}=0 & \rightarrow\quad 0\leq\alpha_i\leq c,\ i=1,\dots,N.\end{cases}$$

و با جایگذاری w بدست آمده در تابع لاگرانژ مساله بهینه سازی ما به یک مساله بهینه سازی غیرخطی بصورت زیر تبدیل می شود:

$$\max Q_1(\alpha)=-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)+\sum_{i=1}^{N}\alpha_i,$$

با حل این مسله بهینه سازی با توجه به محدودیت های مربوط به ضرایب لاگرانژ، ابرصفحه ای در فضای ویژگی با ابعاد بالاتر بدست خواهد آمد در حالی که کلاس بندی کننده ما در همان فضای اولیه است. در عبارت بالا $K(x_i,x_j)=\left\langle\phi(x_i),\phi(x_j)\right\rangle$ یک تابع کرنل است که در ادامه نسخه هایی از آنها را که در این ماشین های بردار پشتیبان مورد استفاده قرار میگرند را معرفی خواهیم کرد.

هسته های متداول به صورت زیر هستند:

خطی:

$$k(x,x_i)=x_i^T x,$$

چندجمله ای (همگن) درجه d :

$$k(\mathbf{x_i},\mathbf{x_j})=(\mathbf{x_i}\cdot\mathbf{x_j})^d$$

چندجملهای (ناهمگن) درجه d :

$$k(\mathbf{x_i},\mathbf{x_j})=(\mathbf{x_i}\cdot\mathbf{x_j}+1)^d$$

گوسیین :

$$k(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2), \gamma > 0., \gamma = 1/2\sigma^2$$

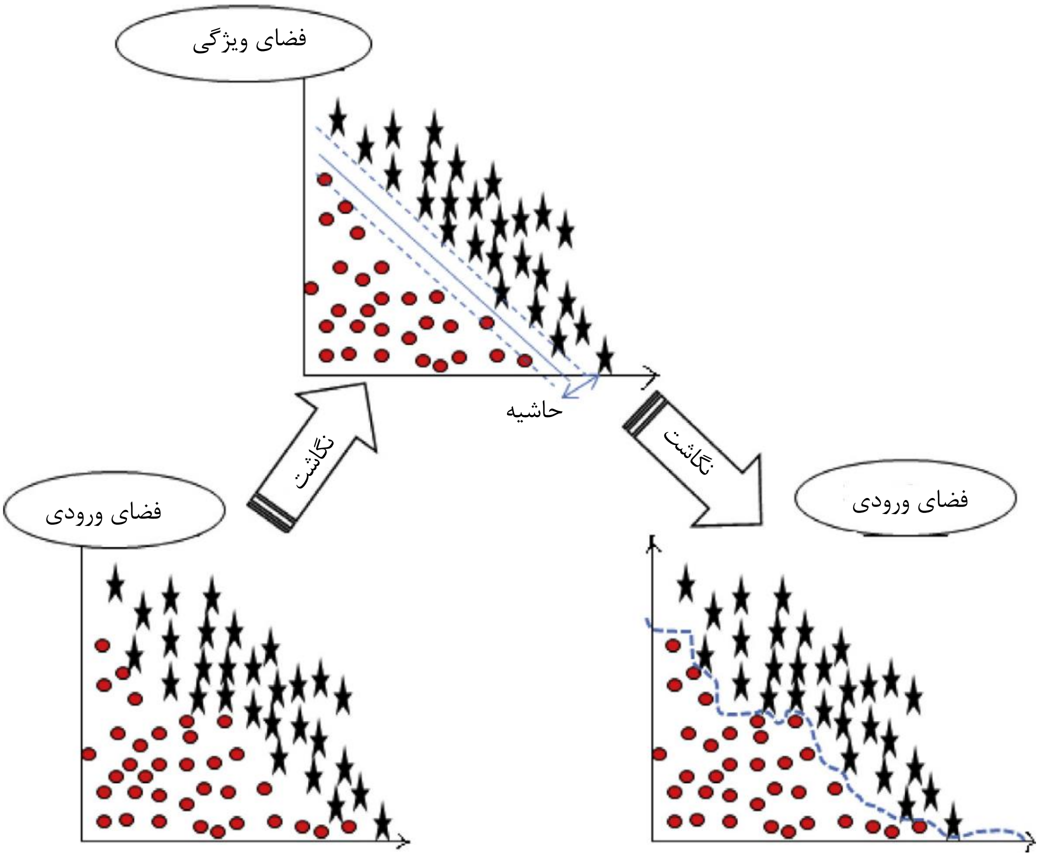
یا بصورت زیر

$$k(x, x_i) = \exp\left(-\|x - x_i\|^2 / \sigma^2\right),$$

تانژانت هذلولی:

$$k(x_i, x_j) = \tanh(kx_i \cdot x_j + c)$$

شکل زیر بصورت ساده نحوه تاثیر تابع هسته بر روی فضای ویژگی ها و نتیجه مرتب با آن را بخوبی نشان می دهد.



کلاس بندی کننده ی (LSSVM)

نسخه حداقل مربعات ماشین های بردار پشتیبان بوسیله بازنویسی مساله مینیم سازی بصورت زیر حاصل می شود:

$$\min J_2(w, b, e) = \frac{\mu}{2} w^T w + \frac{\zeta}{2} \sum_{i=1}^{N} e_{c,i}^2 ,$$

subject to:

$$y_i\left[w^T\phi(x_i)+b\right]=1-e_{c,i}, \quad i=1,\ldots,N.$$

کلاس بندی کننده ی حداقل مربعات بالا بطور ضمنی برای درونیابی رگراسیون با مقادیر هدف $y_i=\pm 1$ فرمول بندی شده است و با استفاده از ترفند کرنل برای داده های با فضای ورودی پیچیده نیز قابل استفاده است با بکار بردن $y_i=1$ داریم:

$$\sum_{i=1}^{N}e_{c,i}^2=\sum_{i=1}^{N}(y_i e_{c,i})^2=\sum_{i=1}^{N}e_i^2=\sum_{i=1}^{N}\left(y_i-(w^T\phi(x_i)+b)\right)^2,$$

که

$$e_i=y_i-(w^T\phi(x_i)+b).$$

است. بنابراین فرمول بندی کلاس بندی کننده ی حداقل مربعات با رابطه زیر معادل است.

$$J_2(w,b,e)=\mu E_W+\zeta E_D$$

که در آن

$$E_W=\frac{1}{2}w^Tw$$

و

$$E_D=\frac{1}{2}\sum_{i=1}^{N}e_i^2=\frac{1}{2}\sum_{i=1}^{N}\left(y_i-(w^T\phi(x_i)+b)\right)^2.$$

است.

جواب مساله بالا بعد از تشکل تابع لاگرانژ زیر و حل آن بدست خواهد

$$\begin{cases} L_2(w,b,e,\alpha)=J_2(w,e)-\sum_{i=1}^{N}\alpha_i\left\{\left[w^T\phi(x_i)+b\right]+e_i-y_i\right\}, \\ \qquad =\frac{1}{2}w^Tw+\frac{\gamma}{2}\sum_{i=1}^{N}e_i^2-\sum_{i=1}^{N}\alpha_i\left\{\left[w^T\phi(x_i)+b\right]+e_i-y_i\right\}, \end{cases}$$

که $\alpha_i\in\mathbb{R}$ ضرایب لاگرانژ و شرایط بهینه برای آن بصورت زیر هستند

$$\begin{cases} \frac{\partial L_2}{\partial w}=0 \quad \rightarrow \quad w=\sum_{i=1}^{N}\alpha_i\phi(x_i), \\ \frac{\partial L_2}{\partial b}=0 \quad \rightarrow \quad \sum_{i=1}^{N}\alpha_i=0, \\ \frac{\partial L_2}{\partial e_i}=0 \quad \rightarrow \quad \alpha_i=\gamma e_i, i=1,\ldots,N, \\ \frac{\partial L_2}{\partial \alpha_i}=0 \quad \rightarrow \quad y_i=w^T\phi(x_i)+b+e_i, i=1,\ldots,N. \end{cases}$$

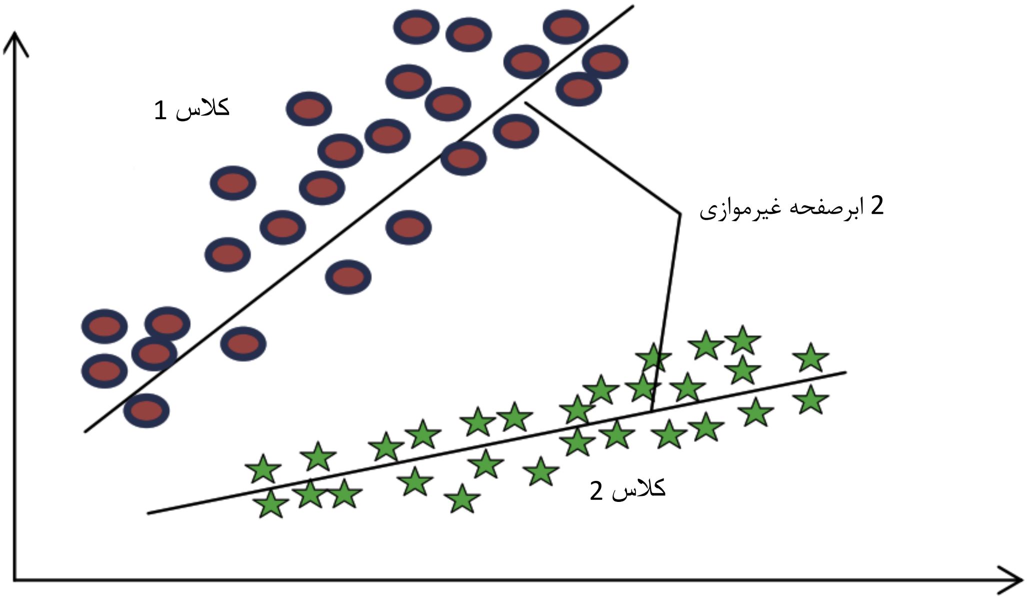
بعد از حذف w و e یک سیستم خطی بجای یک سیستم برنامه ریزی غیرخطی درجه دوم بصورت زیر حاصل می شود:

$$\begin{bmatrix} 0 & 1_N^T \\ 1_N & \Omega+\gamma^{-1}I_N \end{bmatrix}\begin{bmatrix} b \\ \alpha \end{bmatrix}=\begin{bmatrix} 0 \\ Y \end{bmatrix},$$

که $Y=[y_1,\ldots,y_N]^T, 1_N=[1,\ldots,1]^T, \alpha=[\alpha_1,\ldots,\alpha_N]^T$ در اینجا I_N یک ماتریس همانی $I_N \times I_N$ و $\Omega \in \mathbf{R}^{N\times N}$ کرنل تعریف شده توسط $\Omega_{ij}=\phi(x_i)^T\phi(x_j)=k(x_i,x_j)$ است.

کلاس بندی کنند ی (TwinSVM or TSVM)

نسخه ماشین های بردار پشتیان دوقلو راهکار متفاوتی را برای کلاس بندی مورد استفاده قرار میدهد. در این متد بجای بدست آوردن یک ابر صفحه جدا کننده برای کلاس بندی ما 2 ابر صفحه غیر موازی را به نحوی انتخاب میکنیم که هرکدام کمترین فاصله ممکن را از داده های یک کلاس و بیشترین فاصله ممکن را از کلاس دیگر داشته باشد. بصورت بسیار ساده برای داده های با 2 کلاس هدف ما یافتن ابرصفحه های بصورت شکل زیر است. کلاس بندی بوسیله آن هم بسیار ساده است و عملا هر داده آزمایشی که داخل هریک از ابرصفحه مقدار آن محاسبه می شود و کلاس مربوط به داده آزمایشی متناظر با ابرصفحه با کمترین مقدار است



فرم اولیه مساله بهینه سازی برای روش ماشین های بردار پشتیبان دوقلو بصورت زیر است:

$$\begin{aligned} \min(w_1,b_1,\xi) \quad & \frac{1}{2}\|X_1w_1+e_1b_1\|^2+c_1e_2^T\xi \\ \text{s.t.} \quad & -(X_2w_1+e_2b_1)+\xi \geqslant e_2, \xi \geqslant 0 \end{aligned}$$

$$\begin{aligned} \min(w_2,b_2,\eta) \quad & \frac{1}{2}\|X_2w_2+e_2b_2\|^2+c_2e_1^T\eta \\ \text{s.t.} \quad & (X_1w_2+e_1b_2)+\eta \geqslant e_1, \eta \geqslant 0 \end{aligned}$$

که در آن ξ و η متغیرهای کمکی و c_1, c_2 پارامترهای جریمه هستند. e_1 و e_2 نشانده 2 بردار با بعد مناسب که دارای مقادیر 1 هستند.
فرم لاگرانژ معادله اول بصورت زیر است:

$$L(w_1, b_1, \xi, \alpha, \beta) = \frac{1}{2}\|X_1 w_1 + e_1 b_1\|^2 + c_1 e_2^T \xi + \alpha^T((X_2 w_1 + e_2 b_1) - \xi + e_2) - \beta^T \xi$$

که در آن α و β ضرایب لاگرانژ هستند و شرایط KKT (Karush–Kuhn–Tucker) برای تابع لاگرانژ بصورت زیر هستند:

$$\frac{\partial L}{\partial w_1} = X_1^T(X_1 w_1 + e_1 b_1) + X_2^T \alpha = 0$$

$$\frac{\partial L}{\partial b_1} = e_1^T(X_1 w_1 + e_1 b_1) + e_2^T \alpha = 0$$

$$\frac{\partial L}{\partial \xi} = c_1 e_2^T - \beta^T - \alpha^T = 0$$

$$-(X_2 w_1 + e_2 b_1) + \xi \geqslant e_2, \xi \geqslant 0$$

$$\alpha^T((X_2 w_1 + e_2 b_1) - \xi + e_2) = 0,\ \beta^T \xi = 0$$

$$\alpha \geqslant 0,\ \beta \geqslant 0$$

که منجر به رابطه زیر می شود

$$\begin{bmatrix} X_1^T \\ e_1^T \end{bmatrix} [X_1 \quad e_1] \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} X_2^T \\ e_2^T \end{bmatrix} \alpha = 0$$

با قرار دادن $A = [X_1, e_1]$ و $B = [X_2, e_2]$ و $u_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$ در این داریم:

$$A^T A u_1 + B^T \alpha = 0$$
$$u_1 = -(A^T A)^{-1} B^T \alpha$$

برای اینکه مقدار $A^T A$ وارون پذیر باشد و از سینگولار شدن آن جلوگیری شود آن را بصورت $(A^T A\ + \delta I)$ در نظر میگیریم که I یک ماتریس همانی هم انداز ماتریس $A^T A$ است. که خواهیم داشت:

$$u_1 = -(A^T A + \delta I)^{-1} B^T \alpha$$

با روندی کاملا مشابه برای ابرصفحه دوم به عبارت زیر خواهیم رسید

$$u_2 = (B^T B + \delta I)^{-1} A^T \gamma$$

که در نهایت کلاس بندی کننده ی ما برای 2 کلاس بصورت($i = min|x^T w_i + b_i|$ کلاس) است. نکته ای که در هر سه روش معرفی شده و همه روش کلاس بندی کننده براساس ماشین های بردار بعنوان وجه مشترک می توان به آن اشاره کرد، 2 کلاسه بودن داده های مورد استفاده است. برای اینکه بتوان از این متدهای برای کلاس بندی های چندکلاسه استفاده کرد از دو رویکرد زیر استفاده می شود.

1- یک کلاس در مقابل بقیه کلاس ها : (One vs All)

2- یک کلاس در مقابل یک کلاس دیگر یا کلاس بندی دودویی: (One vs One)

روش پیشنهادی

با استفاده از سه روش معرفی شده یک سیستم هوشمند را آموزش میدهیم و سپس سه روش را برای مقایسه میزان دقت کلاس بندی روی دیتابیس های استاندارد UCI مورد بررسی قرار خواهیم داد.

آزمایش و بررسی

برای بررسی عملکرد هر سه روش ماشین های بردار پشتیبان از 3 دیتابیس استاندار که اصل دیتابیس ها و توضیحات مربوط به آنها بصورت کامل در فایل ضمیمه گنجانده شده است. در این بررسی از 85 درصد داده ها برای تعلیم سیستم و از 15 درصد داده ها برای آزمایش سیستم استفاده شده است. برای بدست آوردن دقت نسبی هر یک از روش ها هرکدام از روش ها 10 بار تکرار شده و در هر تکرار تعداد داده های تعلیم و آزمایس بصورت تصادفی و براساس توضیح نرمال انتخاب شده اند. و در نهایت میانگین درصد درستی و زمان اجرا برای هر روش محاسبه شده است. نتایج حاصل بروی سیستم عامل ویندوز ماکروسافت 8.1 و با مشخصات سخت افزاری سی پی یو corei5 2.2GHz , Ram :4Gb و با استفاده از نسخه R2014b متلب بدست آمده است. برای روش svm از تولباکس متلب استفاده شده است و کد های مربوط به 2 روش دیگر براساس مقالات مربوط به آنها طراحی شده است. نتایج حاصل در جدول های زیر آمده است.

DATA name : ionosphere

R	LSSVM Accu	LSSVM Time	TwinSVM Accu	TwinSVM Time	SVM Accu	SVM Time	N Test	N Train
1	94.3396	0.1092	58.4906	14.4815	88.6792	0.19236	53	298
2	90.566	0.1248	75.4717	13.2128	90.566	0.17204	53	298
3	92.4528	0.078	73.5849	13.2509	88.6792	0.1405	53	298
4	92.4528	0.1092	73.5849	13.9477	88.6792	0.18752	53	298
5	92.4528	0.1248	60.3774	14.7561	90.566	0.13141	53	298
6	90.566	0.0624	64.1509	14.3847	90.566	0.19277	53	298
7	94.3396	0.078	62.2642	14.3193	88.6792	0.19481	53	298
8	92.4528	0.078	66.0377	15.136	90.566	0.15726	53	298
9	83.0189	0.0624	69.8113	14.6669	88.6792	0.18111	53	298
10	90.566	0.1248	54.717	14.4465	88.6792	0.19624	53	298
ave:	91.3208	0.095161	65.8491	14.2602	89.434	0.1746	53	298

DATA name : heart

R	LSSVM Accu	LSSVM Time	TwinSVM Accu	TwinSVM Time	SVM Accu	SVM Time	N Test	N Train
1	82.9268	0.0312	48.7805	7.8407	78.0488	0.088038	41	229
2	85.3659	0.093601	53.6585	7.852	75.6098	0.11399	41	229
3	87.8049	0.093601	51.2195	7.461	85.3659	0.094701	41	229
4	85.3659	0.093601	48.7805	8.085	65.8537	0.10339	41	229
5	82.9268	0.0312	53.6585	10.3664	75.6098	0.098591	41	229
6	82.9268	0.0624	56.0976	8.504	82.9268	0.096557	41	229
7	78.0488	0.0624	51.2195	8.433	65.8537	0.096994	41	229
8	87.8049	0.078	48.7805	8.9963	75.6098	0.098683	41	229
9	90.2439	0.0468	48.7805	8.3985	70.7317	0.092684	41	229
10	85.3659	0.0624	53.6585	8.2852	78.0488	0.097592	41	229
ave:	84.878	0.06552	51.4634	8.4222	75.3659	0.098122	41	229

DATA name : bupa

R	LSSVM Accu	LSSVM Time	TwinSVM Accu	TwinSVM Time	SVM Accu	SVM Time	N Test	N Train
1	67.3077	0.1248	42.3077	14.0934	59.6154	0.16126	52	293
2	69.2308	0.078001	53.8462	12.9608	65.3846	0.11091	52	293
3	80.7692	0.078001	53.8462	14.1087	78.8462	0.12942	52	293
4	76.9231	0.1248	59.6154	13.2661	69.2308	0.1096	52	293
5	59.6154	0.1248	46.1538	13.7842	69.2308	0.11844	52	293
6	71.1538	0.078	53.8462	13.8513	65.3846	0.16601	52	293
7	65.3846	0.1248	57.6923	13.2794	61.5385	0.17353	52	293
8	69.2308	0.1092	42.3077	13.9663	69.2308	0.14427	52	293
9	73.0769	0.0156	26.9231	13.1219	69.2308	0.17655	52	293
10	75	0.078	51.9231	13.1397	73.0769	0.1705	52	293
ave:	70.7692	0.093601	48.8462	13.5572	68.0769	0.14605	52	293

نتیجه گیری

باتوجه به به نتایج حاصل از بررسی روش ماشین های بردار پشتیبان حداقل مربعات دارای عملکرد بهتر هم از نظر دقت کلاس بندی و هر از نظر زمان اجرا نسبت به 2 روش دیگر است.

منابع

[1] https://en.wikipedia.org/wiki/Support_vector_machine

[2] C. Corinna and V. Vapnik, "Support-vector networks", Machine learning, vol. 20, no. 3, **(1995)**, pp. 273-297.

[3] Suykens J.A.K., Van Gestel T., De Brabanter J., De Moor B., Vandewalle J., Least Squares Support Vector Machines, World Scientific, to appear. **1999**

[4] Jayadeva, Khemchandani, R., & Chandra, S. (**2007**). Twin support vector machines for pattern classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(5), 905–910.

[5] K. M. Arun and M. Gopal, "Least squares twin support vector machines for pattern classification", Expert Systems with Applications, vol. 36, **(2009)**, pp. 7535–7543.