ماشین بردار پشتیبان (SVM)

(Support Vector Machine)

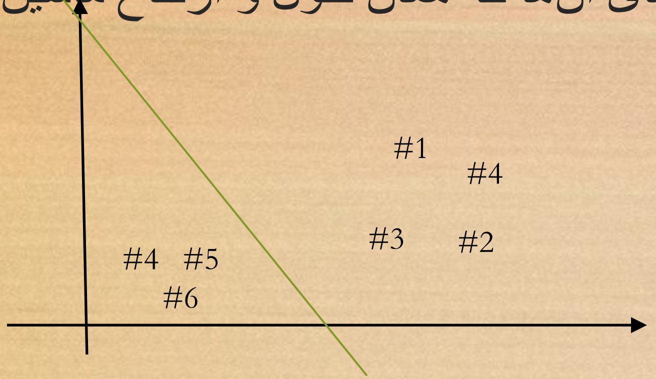
فهرست مطالب

- مقدمه
- مسئله جداسازی خطی (Linear Discrimination)
- مسئله جداسازی غیر خطی
  - بهینه سازی با حاشیه های انعطاف پذیر (نرم)
  - بکارگیری kernel trick به منظور خطی سازی مسائل غیر خطی مورد نظر
- ماشین‌های بردارپشتیبانی و شبکه‌های عصبی
- نقاط ضعف ماشین های بردار پشتیبانی

ماشین بردار پشتیبان(SVM) یک الگوریتم نظارت شده یادگیری ماشین است که هم برای مسائل طبقه بندی و هم رگرسیون قابل استفاده است، با این حال از آن بیشتر در مسائل طبقه بندی استفاده می شود. در الگوریتم SVM هر نمونه داده را به عنوان یک نقطه در فضای n بعدی روی نمودار پراکندگی داده ها ترسیم کرده و مقدار هر ویژگی مربوط به داده ها، یکی از مولفه های مختصات نقطه روی نمودار را مشخص می کند. سپس با ترسیم یک خط راست، داده های مختلف و متمایز از یکدیگر را دسته بندی می کند.

مقدمه

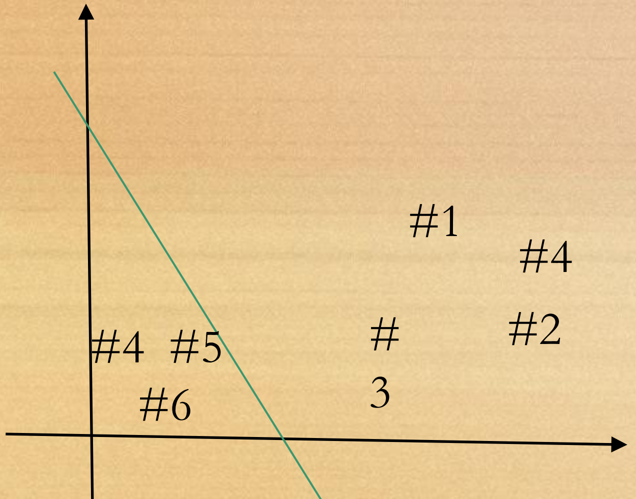
- پرسپترون در واقع یک طبقه‌بند خطی (linear classifier) است که می‌تواند تفاوت بین دو طبقه را تشخیص دهد. یک مثالِ پرسپترون به این صورت بود که می‌خواستیم تفاوت میان **پراید** و **اتوبوس** را با توجه به ویژگی‌های آن‌ها که همان طول و ارتفاع ماشین بود، به دست بیاوریم.



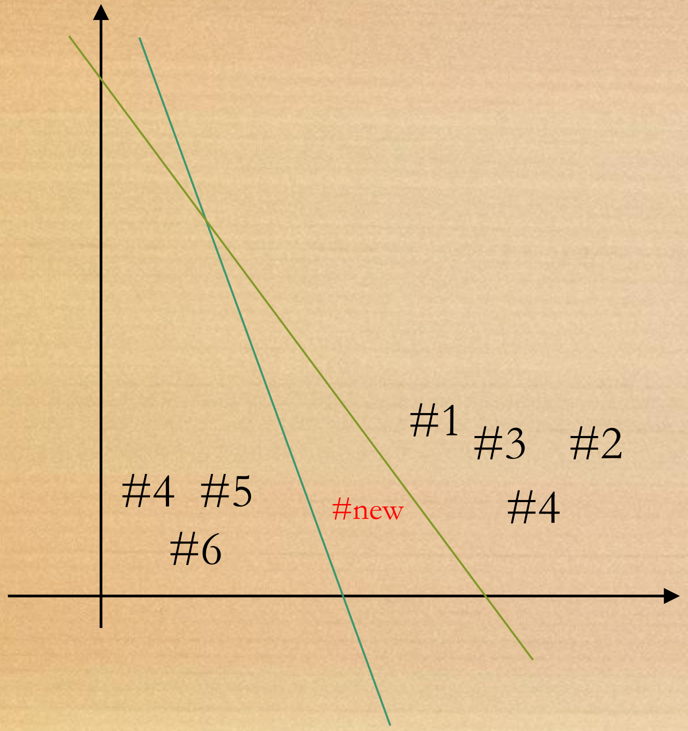
- خطِ آبی که در تصویر بالا مشاهده می‌کنید، در واقع خطی است که پرسپترون (با توجه به پارامترهای W- شیب خط و b-انحراف خط) رسم کرده است. پرسپترون با استفاده از این خط می‌تواند بفهمد نقاطی که پایین‌تر از این خط هستند، نقاط **پراید** و نقاطی که بالاتر از این خط هستند نقاط اتوبوس هستند.

- حال فرض کنید که خطِ آبیِ بالا به صورتی دیگر کشیده می‌شد. مثلاً مانند شکل زیر:

در واقع این خطی است که باز هم به صورت کامل **پراید** و **اتوبوس** را از هم دیگر جدا کرده است.



- حال فرض کنید یک نمونه جدیدِ ماشین می‌آید و الگوریتم نمی‌داند که این ماشین چیست. پس باید با توجه به خط جدا کننده‌ای که توسط داده‌های آموزشی یادگرفته است، این ماشین جدید را طبقه بندی کند.
مانند #new در شکل زیر:



اگر خطِ آبی یعنی همان خطِ جداکننده باشد، این اتومبیلِ جدید به دسته **پرایدها** می‌رود، این در حالی است که اگر خطِ نارنجی خط جداکننده باشد، این اتومبیل به دسته‌ی **اتوبوس‌ها** می‌رود. البته ما به عنوانِ یک انسان می‌توانیم بفهیم که این اتومبیل بیشتر به پرایدها شبیه است چون نزدیکِ نمونه‌های پرایدهاست. ولی الگوریتم با توجه به خطی که از داده‌های آموزشی یاد گرفته است، این تمایز و طبقه‌بندی را انجام می‌دهد. در واقع مشکلِ اصلیِ پرسپترون همین‌جاست که خطی که رسم می‌کند، خط بهینه‌ای نیست و ممکن است خطا ایجاد کند. بنابراین به روشِ به مراتب پیشرفته‌تری به اسمِ ماشینِ بردار پشتیبان می‌رسیم که این مشکل را برطرف خواهد کرد.

ماشین بردار پشتیبان (Support Vector Machine) چیست؟

- الگوریتم SVM اولیه در ۱۹۶۳ توسط وپنیک ابداع شد و در سال ۱۹۹۵ توسط وپنیک و کورتس براي حالت غيرخطي تعمیم داده شد.

- ماشین بردار پشتیبان که به اختصار به آن SVM گفته می‌شود یک الگوریتم **یادگیری ماشین با ناظر** است که نمونه‌ی داده‌هایی را به صورتی نقاطی در فضا نشان داده شده است، با استفاده از یک خط یا هایپرپلین (Hyperplane)، از هم جدا می‌کند. این جداسازی به‌گونه‌ای است که نقاط داده‌ای که در یک طرف خط هستند مشابه به‌هم و در یک گروه قرار می‌گیرند. نمونه داده‌های جدید هم بعد از اضافه شدن به همان فضا در یکی از دسته‌های موجود قرار خواهند گرفت.
- اصول نظری مربوط به این الگوریتم ها بر پایه تئوری یادگیری آماری (statistical learning theory) است. مبنای نظریه مزبور، بر قائل شدن تفاوت بین نمونه های مختلف در حین یادگیری است.
- ماشین های بردار پشتیبانی از دقیق ترین و نیرومندترین الگوریتم های داده کاوی است.
- در اغلب موارد هیچ حساسیتی نسبت به ابعاد داده ها ندارند.
- شامل دسته بندی کننده بردار پشتیبانی (SVC) و رگرسور(رگرسیون) بردار پشتیبانی (SVR) است.

- این روش از جمله روش‌هاي نسبتاً جديدي است که در سال‌هاي اخير کارايي خوبي نسبت به روش‌هاي قديمي‌تر براي طبقه‌بندي از جمله شبکه های عصبی پرسپترون نشان داده است.
- مبناي کاري دسته بندی کننده SVM دسته بندی خطی داده ها است و در تقسيم خطي داده‌ها سعي مي‌کنيم خطي را انتخاب کنيم که حاشيه اطمينان بيشتري داشته باشد. حل معادله پيدا کردن خط بهينه براي داده‌ها به وسيله روش‌هاي QP(Quadratic Programming) که روش‌هاي شناخته شده‌اي در حل مسائل محدوديت‌دار هستند صورت مي‌گيرد.
- SVMها به خانواده ای از مدل های خطی تعمیم یافته تعلق دارد.
- SVM دسته بندی کننده ای است که جزو شاخه روش های کرنل (Kernel Methods) دریادگیری ماشین محسوب میشود.
- يکي از روش هايي که در حال حاضر به صورت گسترده براي مسئله دسته بندي (Classification) مورد استفاده قرار مي گيرد، روش ماشين بردار پشتيبان است. شايد به گونه اي بتوان محبوبيت کنوني روش ماشين بردار پشتيبان را با محبوبيت شبکه هاي عصبي در دهه گذشته مقايسه کرد. علت اين قضيه نيز قابليت استفاده اين روش در حل مسائل گوناگون مي باشد، در حاليکه روش هايي مانند درخت تصميم گيري را نمي توان به راحتي در مسائل مختلف به کار برد.

- در صورت استفاده مناسب از SVM این الگوریتم قدرت تعمیم خوبی خواهد داشت.
  - علیرغم داشتن ابعاد زیاد (high dimensionality) از overfitting پرهیز میکند. این خاصیت ناشی از optimization این الگوریتم است
- فشرده سازی اطلاعات:
  - بجای داده های آموزشی از بردارهای پشتیبان استفاده میکند.

➢ هدف این دسته الگوریتم ها تشخیص و متمایز کردن الگوهای پیچیده در داده هاست

➢ مسایل مطرح:

- الگوهای پیچیده را چگونه نمایش دهیم
- چگونه از مسئله overfitting پرهیز کنیم

کاربردهای SVM:

- تشخیص های پزشکی
- بیوانفورماتیک
- پردازش تصویر
- متن کاوی

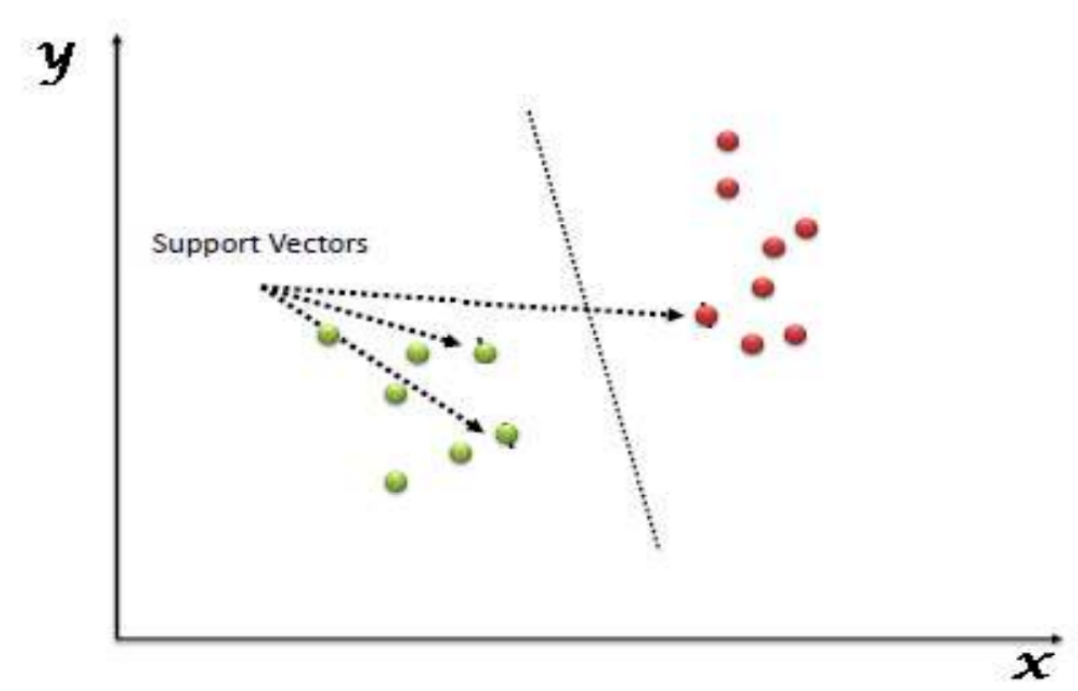
ماشین های بردار پشتیبانی مشابه با شبکه های عصبی قادرند تا برای هر تابع چند متغیره تقریب هایی را با درجه دقت دلخواه به دست بیاورند. بنابراین به منظور مدل کردن سیستم ها و فرآیند های غیر خطی و بسیار پیچیده می توان از SVMها استفاده کرد
.

- با فرض اینکه دسته ها بصورت خطی جداپذیر باشند، ابرصفحه هائی با حداکثر حاشیه (maximum margin) را بدست می آورد که دسته ها را جدا کنند.
- در مسایلی که داده ها بصورت خطی جداپذیر نباشند داده ها به فضای با ابعاد بیشتر نگاشت پیدا میکنند تا بتوان آنها را در این فضای جدید بصورت خطی جدا نمود.

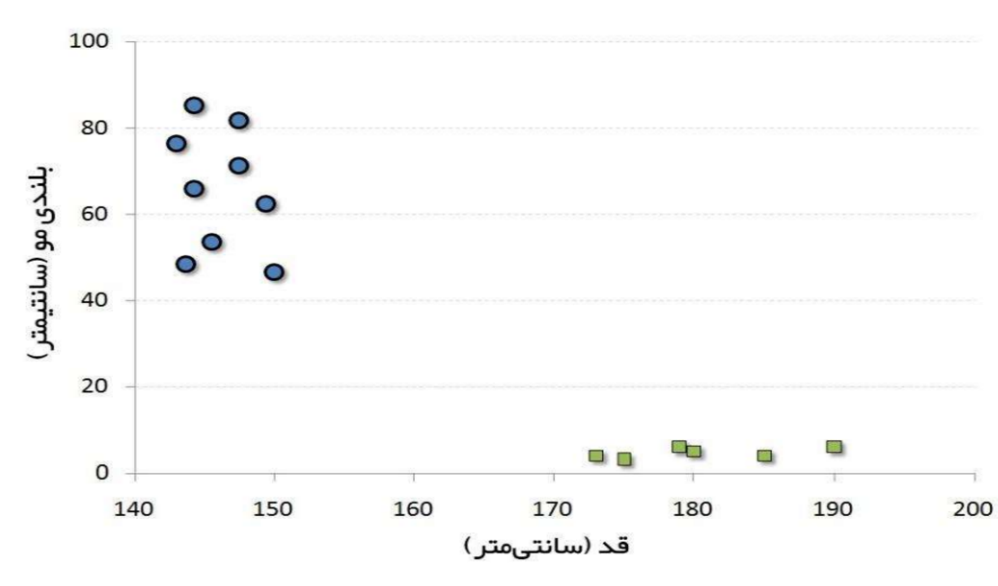
✓ هدف عبارت است از یافتن ابر صفحه جداکننده نقاط داده ای متعلق به دو کلاس با حاشیه ماکسیمال و بهترین توانایی تعمیم.

✓ حاشیه از دیدگاه هندسی عبارت است از فاصله موجود بین ابرصفحه و نزدیک ترین نمونه آموزشی. از یک زاویه دیگر حاشیه اینگونه تعریف می شود : مقدار فضا یا جدایی موجود میان دو کلاس که توسط ابرصفحه تعریف می شود.

✓ منظور از توانایی تعمیم آن است که دسته بندی کننده علاوه بر داشتن عملکرد خوب در دسته بندی داده های آموزشی (دقت)، دارای یک دقت پیش بینی بالا در قبال داده های مشاهده نشده (که توزیع آن ها با توزیع داده های آموزشی یکسان است) نیز باشد.



به بیان ساده، بردارهای پشتیبان در واقع مختصات یک مشاهده منفرد هستند. ماشین بردار پشتیبان مرزی است که به بهترین شکل دسته‌های داده‌ها را از یکدیگر جدا می‌کند.



دایره‌های آبی موجود در نمودار نماینده زنان و مربع‌های سبز نماینده مردان هستند. برخی از برداشت‌هایی که می‌توان از این نمودار داشت عبارتند از:

1) مردان در جمعیت مثال ما، میانگین قد بلندتری دارند.

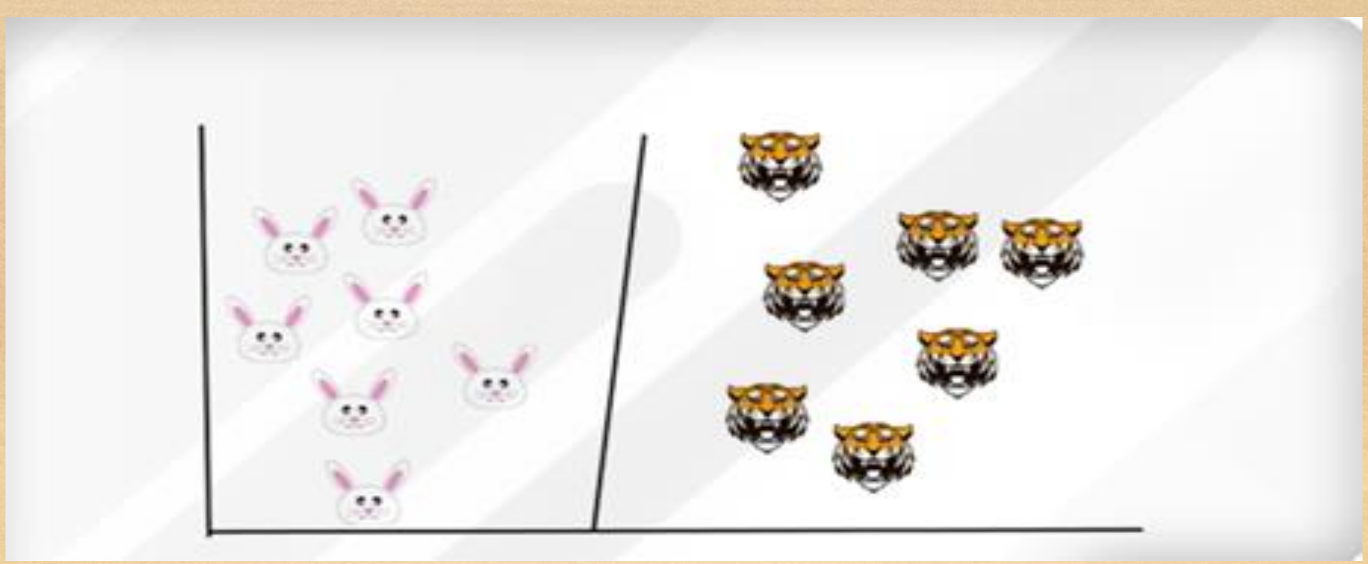
2) زنان در جمعیت مثال ارائه شده، بلندی موی بیشتری دارند.

بر این اساس اگر فردی با قد ۱۸۰ سانتی‌متر و طول موی ۴ سانتی‌متر در جمعیت وجود داشته باشد، بهترین حدسی که می‌توان زد آن است که فرد در دسته مردان قرار می‌گیرد.

ماشین بردار پشتیبان چطور کار می‌کند؟

برای درک نحوه‌ی عملکرد ماشین بردار پشتیبان فرض کنید صاحب مزرعه‌ای هستید و بنا به دلایلی می‌خواهید حصاری برای محافظت از خرگوش‌های خود دربرابر ببرها ایجاد کنید. **حصار خود را کجا بسازیم؟**

- یکی از راه‌حل‌های این مشکل این است که یک طبقه‌بندی براساس موقعیت خرگوش‌ها و ببرها ایجاد کنیم. می‌توانیم گروه خرگوش‌ها را به‌عنوان یک گروه و گروه ببرها را به‌عنوان گروه دیگر طبقه‌بندی کنیم.
- در حال حاضر، اگر سعی کنیم یک مرز میان خرگوش‌ها و ببرها بکشیم، یک خط مستقیم خواهد شد. ماشین بردار پشتیبان نیز دقیقاً به این شکل عمل می‌کند؛ یک مرز تصمیم‌گیری ترسیم می‌کند که درواقع یک هایپرپلین میان دو کلاس است تا آن‌ها را از هم جدا و طبقه‌بندی کند.

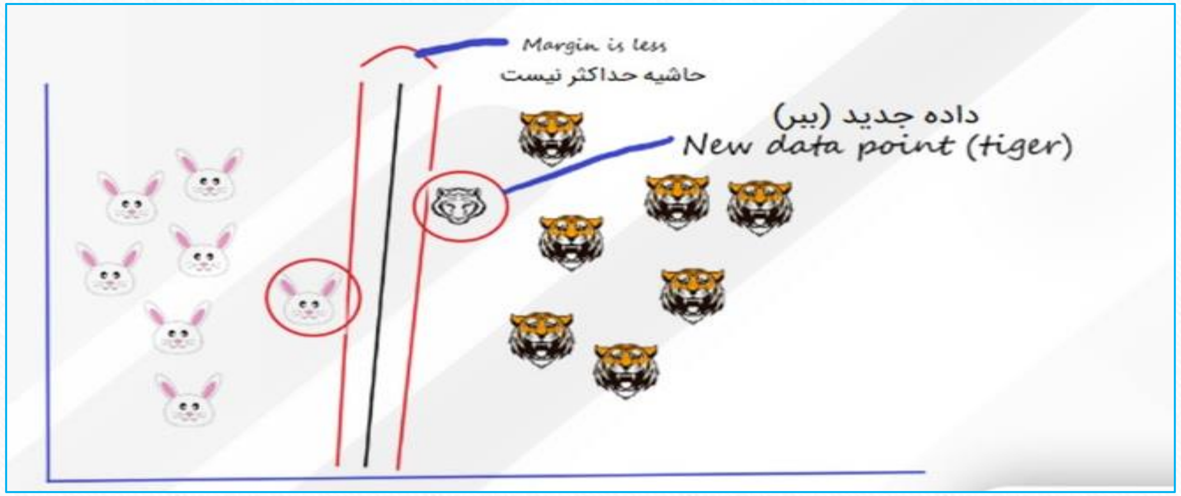


چگونه بدانیم هایپرپلین را کجا باید بکشیم؟

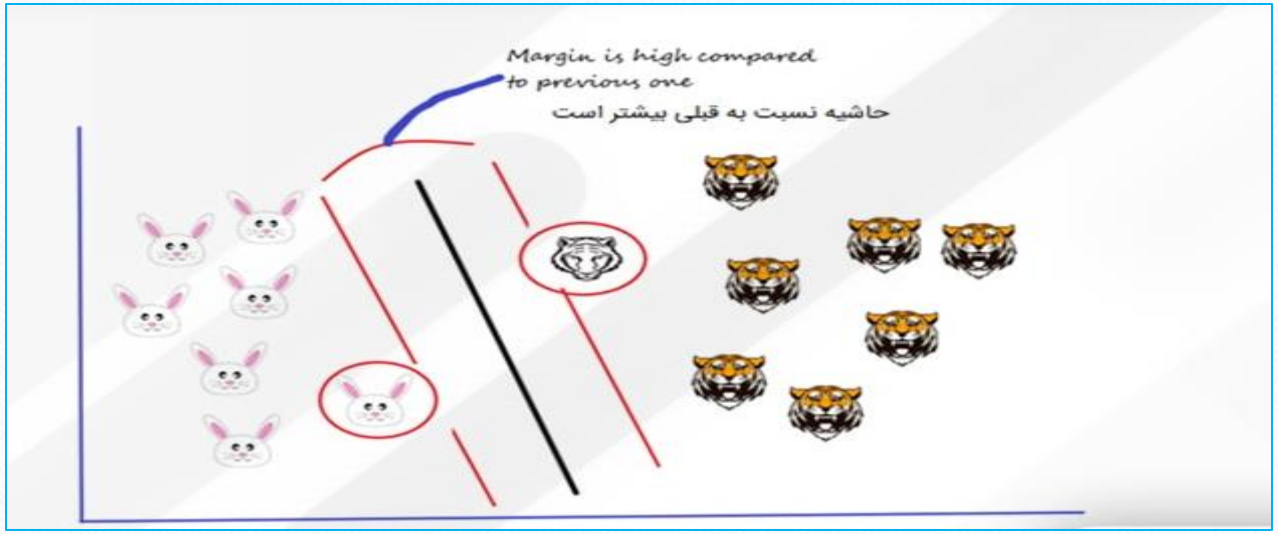
اصل اساسی ماشین بردار پشتیبان این است که یک هایپرپلین ترسیم کنیم که دو کلاس را به بهترین شکل از یکدیگر جدا کند. حال درمورد مثال ما دو کلاس خرگوش و ببر هستند؛ بنابراین ما با ترسیم یک هایپرپلین رندوم شروع می‌کنیم و سپس فاصله‌ی میان هایپرپلین و نزدیک‌ترین نقاط داده‌ی هر کلاس را بررسی می‌کنیم.

این نقاط داده نزدیک به هایپرپلین بردارهای پشتیبان (Support Vectors) نامیده می‌شوند؛ به‌همین دلیل است که به این الگوریتم ماشین بردار پشتیبان گفته می‌شود. اساساً هایپرپلین براساس این بردارهای پشتیبان ترسیم می‌شود. به‌طور معمول هایپرپلینی که بیشترین فاصله از بردارهای پشتیبان را داشته باشد بهینه‌ترین هایپرپلین است. این فاصله‌ی میان هایپرپلین و بردارهای پشتیبان حاشیه (Margin) نامیده می‌شود.

حال بیایید فرض کنیم یک نقطه‌ی داده‌ی جدید اضافه کنیم (در این مثال ببر دیگری اضافه شده است). اکنون می خواهم یک هایپرپلین بکشیم تا این دو کلاس را به‌بهترین شکل از هم جدا کند؛ بنابراین، با ترسیم هایپرپلین، همان‌طور که در تصویر نشان داده شده است، شروع می کنیم؛ سپس فاصله‌ی میان این هایپرپلین و بردارهای پشتیبان را بررسی می کنیم که آیا حاشیه‌ی این هایپرپلین حداکثر است یا خیر. در این شکل حاشیه‌ی خیلی هم زیاد نیست.



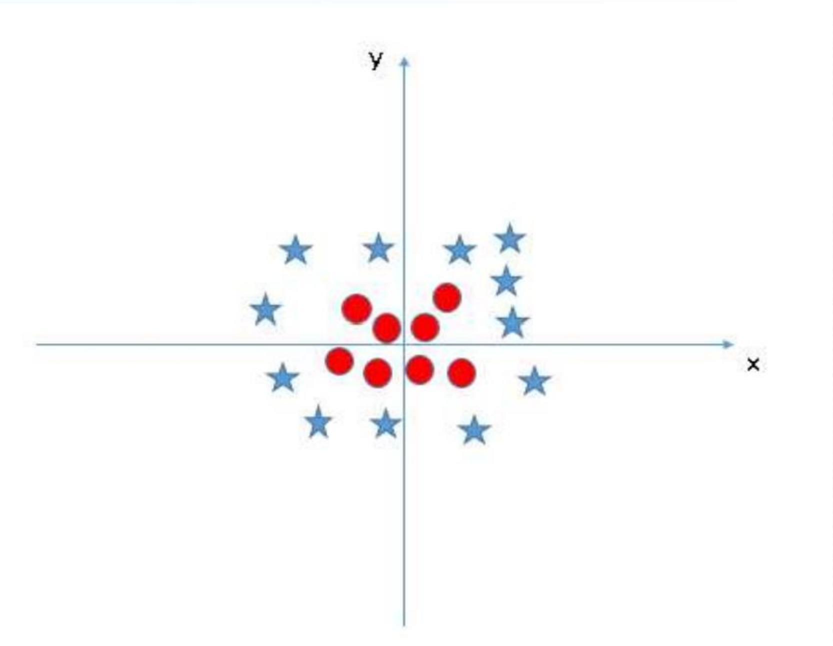
در سناریوی دوم، یک هایپرپلین متفاوت، مانند تصویر زیر، ترسیم می کنیم و سپس فاصله‌ی میان هایپرپلین و بردارهای پشتیبان را بررسی می‌کنیم که آیا حاشیه‌ی این هایپرپلین حداکثر است یا خیر؟

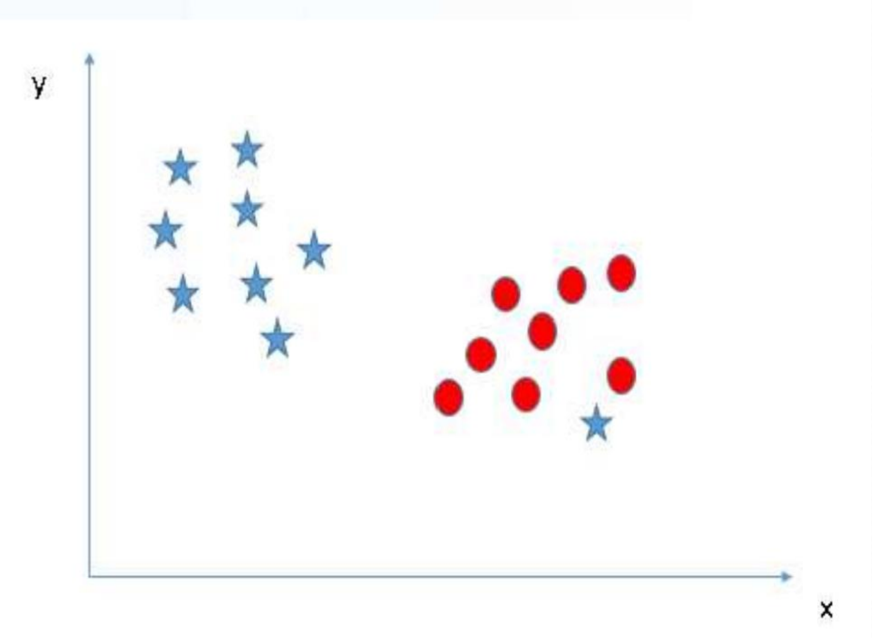


می‌بینیم که حاشیه در مقایسه با هایپرپلین قبلی بسیار زیاد است؛ بنابراین ما این هایپرپلین را انتخاب می‌کنیم.

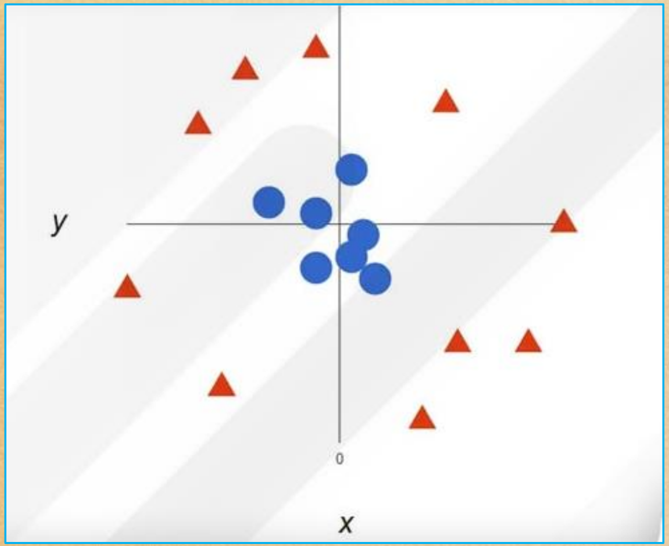
ماشین بردار پشتیبان و داده‌های غیرخطی

- در مثال قبل دیدیم که داده‌های ما تا کنون به‌صورت خطی تفکیک‌پذیر بودند، یعنی می‌توانستیم یک خط مستقیم برای جداکردن دو کلاس بکشیم، اما اگر نقاط داده‌ی ما به‌شکل زیر باشد، چه کنیم؟ این دو کلاس با یک خط مستقیم از هم جدا نمی‌شوند.
- در الگوریتم ماشین بردار پشتیبان، هنگام ساختن یک هایپرپلین برای نقاط داده‌ی تفکیک‌پذیر خطی کار آسانی است، اما وقتی داده‌ها غیرخطی تفکیک‌پذیر باشند، کار بسیار چالش‌برانگیز خواهد بود.


y
x


y
x

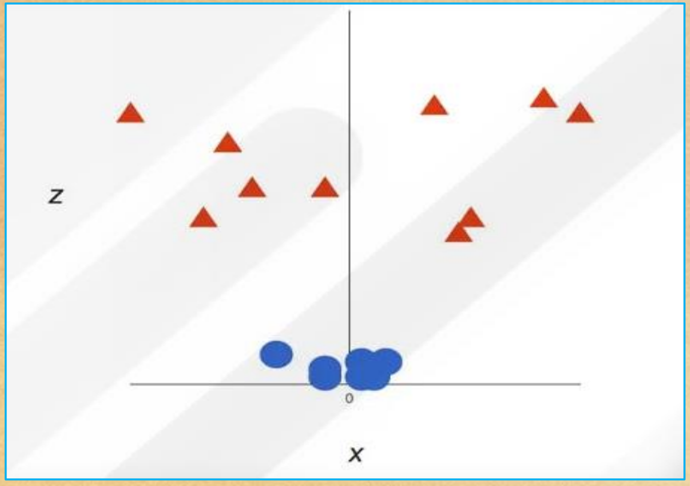
هنگامی‌که نقاط داده را نمی‌توان با یک خط مستقیم یا یک هایپرپلین مستقیم جدا کرد، مسئله غیرخطی نامیده می‌شود. در چنین شرایطی کرنل‌های ماشین بردار پشتیبان وارد عمل می‌شوند و ابعاد فضا را افزایش می‌دهند تا نقاط داده به‌صورت خطی تفکیک‌پذیر شوند.

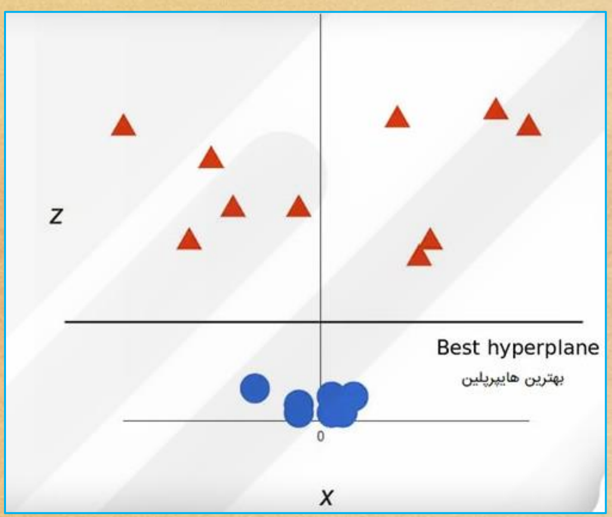


پس اگر داده‌های اولیه ما به شکل روبرو باشند، لازم است که ما بعد سوم را اضافه کنیم. تابه‌حال ما دو بعد داشتیم: x و y حال یک بعد z جدید ایجاد می‌کنیم که به‌این شکل محاسبه می‌شود:

$z = x^2 + y^2$ (اگر دقت کنید این معادله یک دایره است).

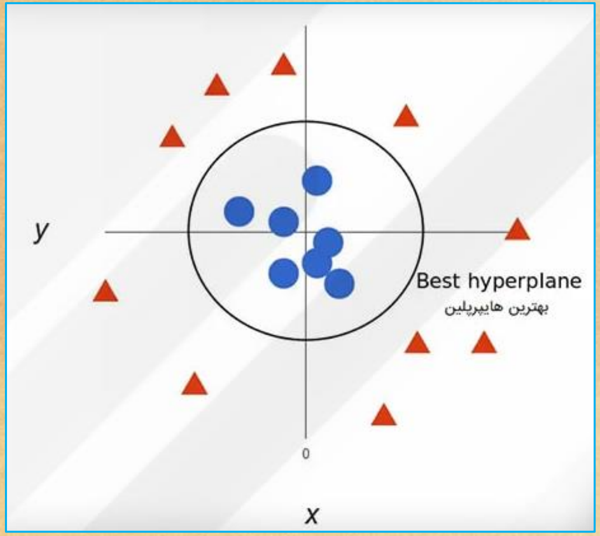
این کار به ما فضایی سه‌بعدی می‌دهد که به این شکل می‌توان آن را در اینجا نشان داد:



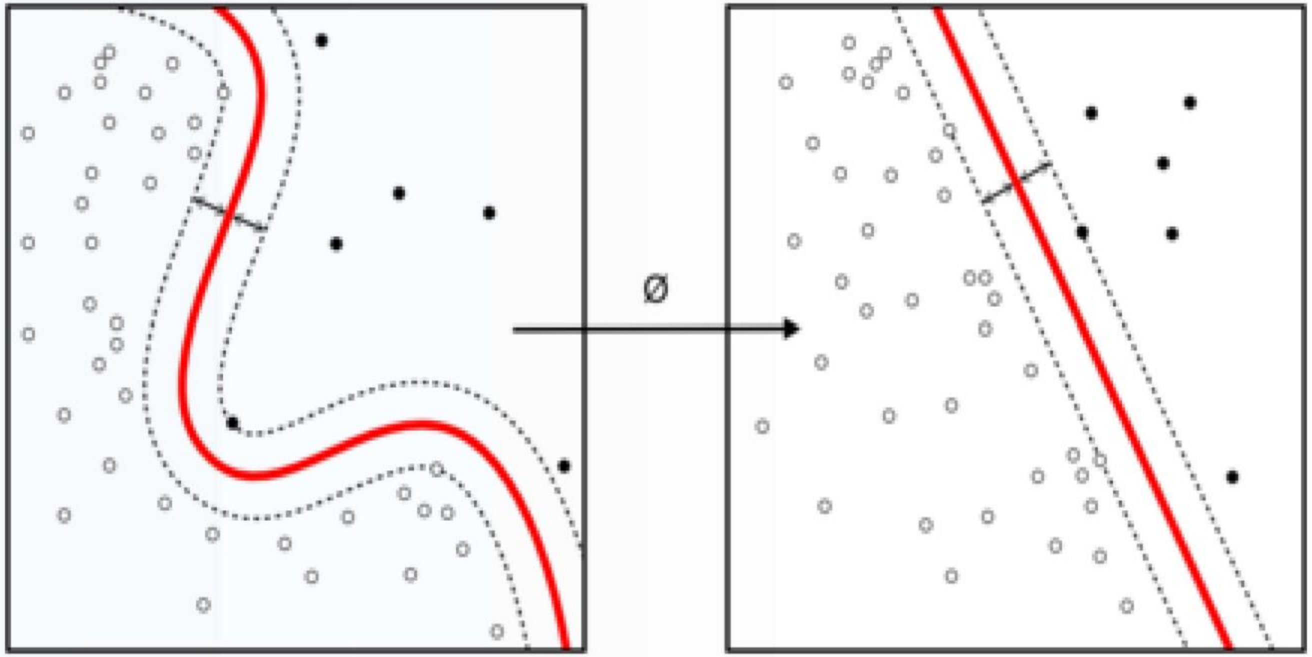


اما ماشین بردار پشتیبان الان چطور تفکیک را انجام می‌دهد؟ اجازه بدهید با هم ببینم:

پس نقاط داده‌ی ما در حال حاضر به‌راحتی با یک خط تفکیک شدند. آنچه باقی مانده ترسیم مجدد آن به‌شکل دوبعدی است:

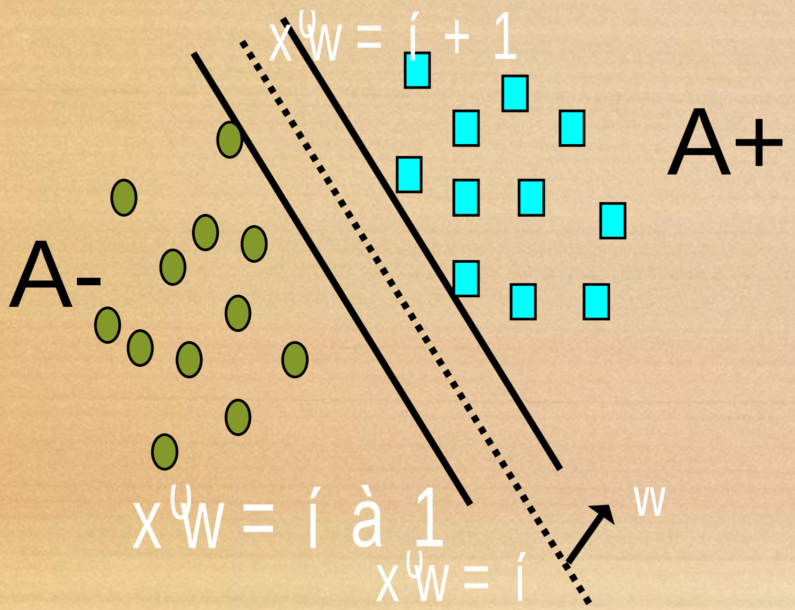


در حال حاضر کار طبقه‌بندی داده‌ها به پایان رسیده است. درواقع به این فرایند که توضیح داده شد حقه‌ی کرنل (Kernel Trick) گفته می‌شود.


Ø

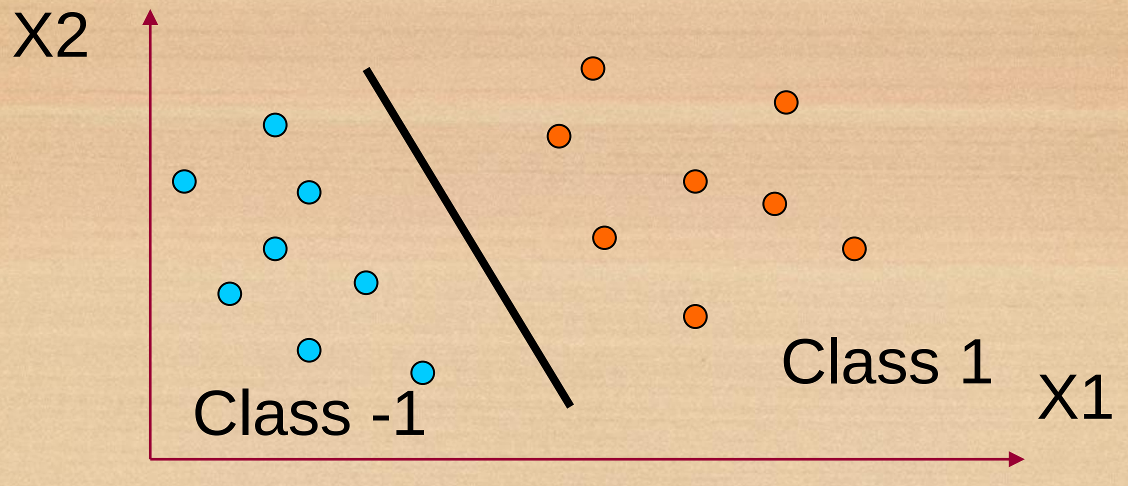
مسئله جداسازی خطی: Linear Discrimination

- اگر دو دسته وجود داشته باشند که بصورت خطی از هم جداپذیر باشند، بهترین جدا کننده این دو دسته چیست؟
- الگوریتم های مختلفی از جمله پرسپترون میتوانند این جداسازی را انجام دهند.
- آیا همه این الگوریتم ها به خوبی از عهده اینکار بر می آیند؟



Separating Surface:

خط یا ابر صفحه جدا کننده



✓ هدف: پیدا کردن بهترین خط (ابر صفحه) که دو دسته را از هم جدا کند. در حالت دو بعدی معادله این خط بصورت زیر است:

$$w_1X_1 + w_2X_2 + b = 0$$

در ابرصفحه بهینه w و b به ترتیب عبارتند از بردار وزن و یک اسکالر.

✓ در حالت n بعدی خواهیم داشت:

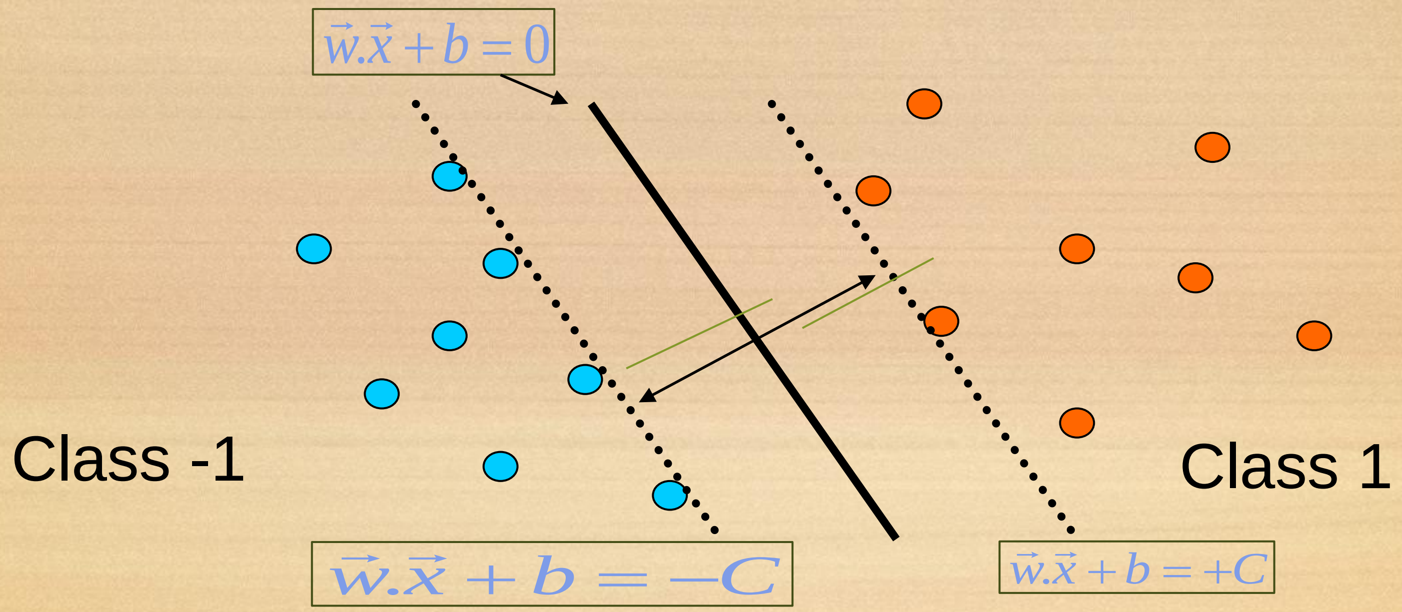
$$\sum_{i=0}^{n} wi.xi + b = 0$$

$$\vec{w}^T.\vec{x} + b = 0$$

ایده SVM برای جدا سازی دسته ها:

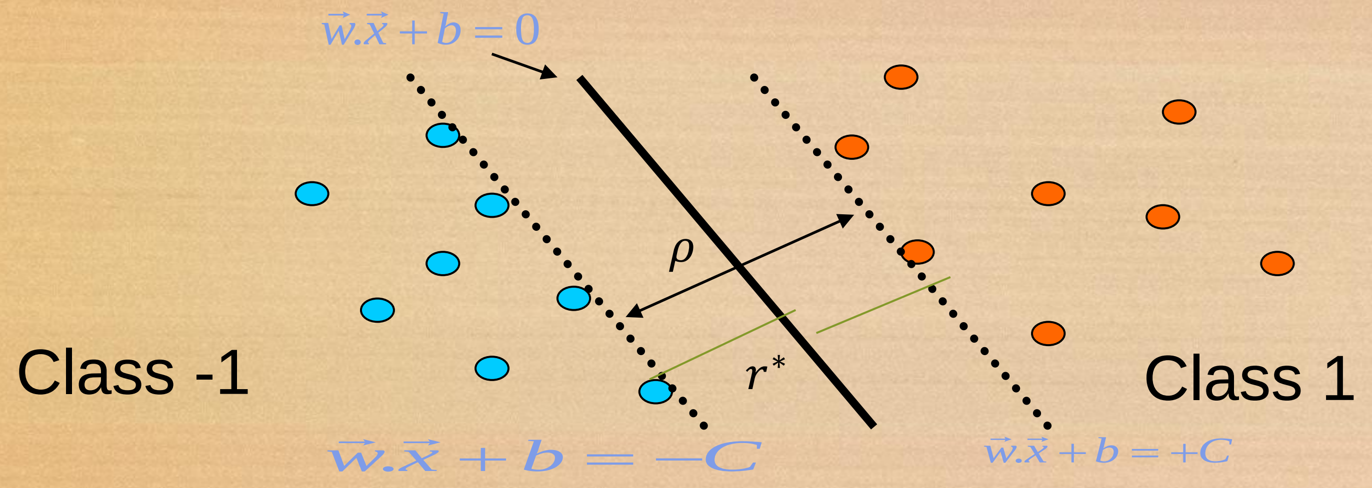
دو صفحه مرزی بسازید :

- دو صفحه مرزی موازی با صفحه دسته بندی رسم کرده و آن دو را آنقدر از هم دور می‌کنیم که به داده ها برخورد کنند.
- صفحه دسته بندی که بیشترین فاصله را از صفحات مرزی داشته باشد، بهترین جدا کننده خواهد بود.



حداکثر حاشیه

- بر طبق قضیه ای در تئوری یادگیری اگر داده های آموزشی بدرستی دسته بندی شده باشند، از بین جداسازهای خطی، آن جداسازی که حاشیه داده های آموزشی را حداکثر میکند خطای تعمیم را حداقل خواهد کرد.

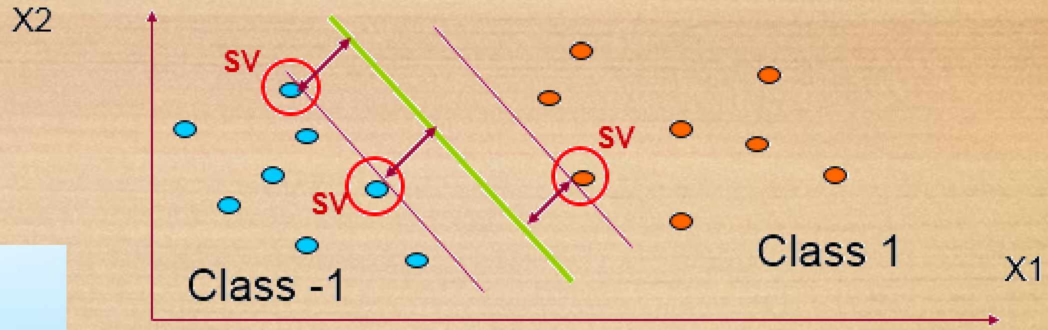


بنابراین در ارتباط با ابرصفحه بهینه پارامترهای w و b را می بایست به نحوی تعیین کرد که حاشیه جداسازی (ρ) بیشینه گردد . این حاشیه با توجه به کوتاه ترین فواصل هندسی r^* از دو کلاس تعیین می شود . با توجه به مطالب بیان شده SVC را دسته بندی کننده با حاشیه ماکسیمال می نامند.

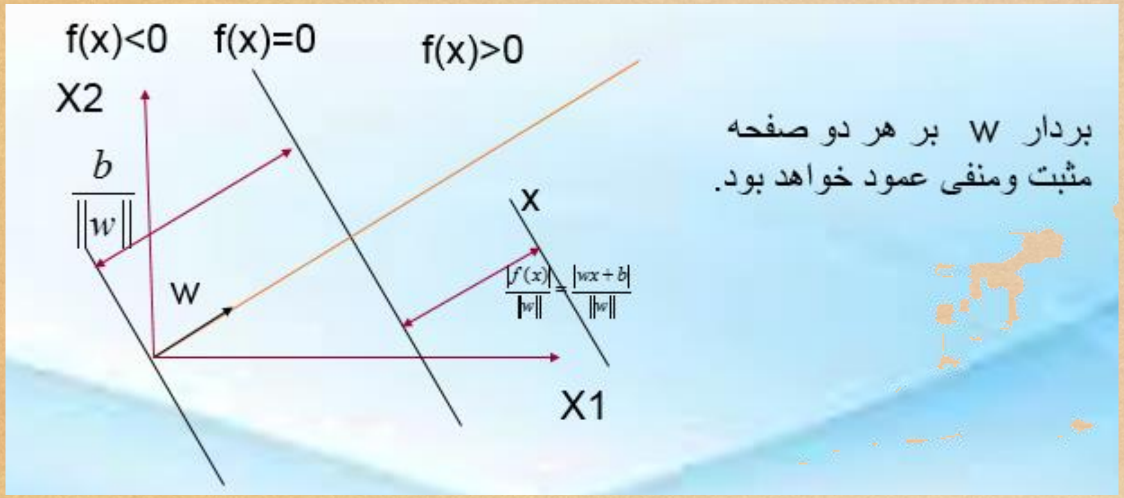
بردار پشتیبان

- نزدیکترین داده های آموزشی به ابر صفحه های جدا کننده بهینه بردار پشتیبان نامیده میشوند. نقاط داده ای ویژه ای که نامساوی های زیر را به تساوی تبدیل می کنند.

$$w^T x_i + b \geq 1 \quad for \quad y_i = +1$$
$$w^T x_i + b \leq -1 \quad for \quad y_i = -1$$



حل مسئله برای حالت دو بعدی



- فاصله خط جداکننده از مبدا برابراست با
- فاصله نمونه ای مثل x از خط جدا کننده برابر است با

$$\frac{b}{\|w\|}$$

$$\frac{|f(x)|}{\|w\|} = \frac{|wx+b|}{\|w\|}$$

تعیین حاشیه بین خطوط جدا کننده

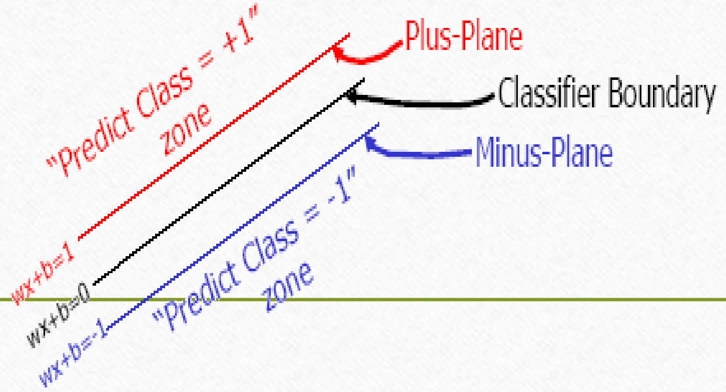
Plus-plane $= \{ \mathbf{x} : \mathbf{w}^T . \mathbf{x} + b = +1 \}$

Minus-plane $= \{ \mathbf{x} : \mathbf{w}^T . \mathbf{x} + b = -1 \}$

Classify as..

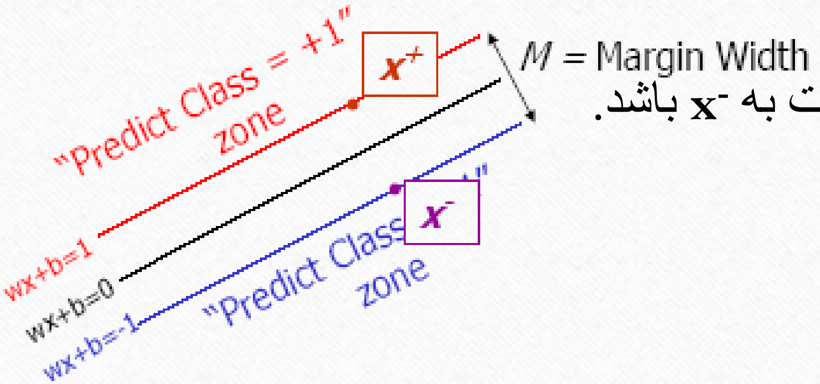
-1 if $\mathbf{w}^T . \mathbf{x} + b <= -1$

+1 if $\mathbf{w}^T . \mathbf{x} + b >= 1$



محاسبه پهنای حاشیه

- صفحه مثبت و منفی را بصورت زیر در نظر میگیریم:
  - Plus-plane $= \{ \mathbf{x} : \mathbf{w}^T . \mathbf{x} + b = +1 \}$
  - Minus-plane $= \{ \mathbf{x} : \mathbf{w}^T . \mathbf{x} + b = -1 \}$
- بردار w بر صفحه مثبت ومنفی عمود خواهد بود.
- فرض کنید $\mathbf{x}^-$ نقطه ای در صفحه منفی بوده و $\mathbf{x}^+$ نزدیکترین نقطه در صفحه مثبت به $\mathbf{x}^-$ باشد.

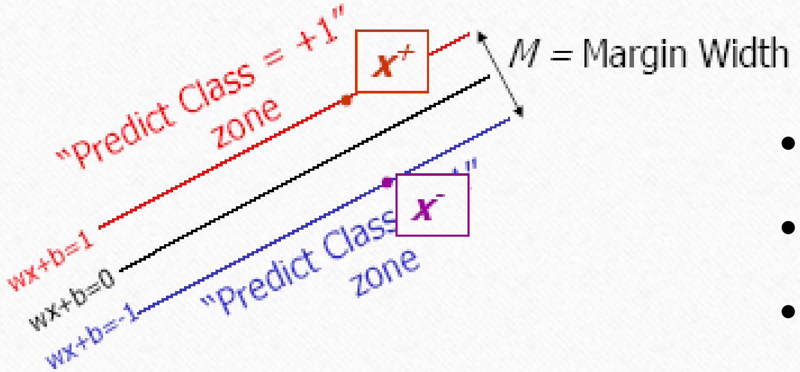


محاسبه پهنای حاشیه

- خطی که x^- رابه x^+ وصل میکند بر هر دو صفحه عمود خواهد بود. لذا فاصله بین دو صفحه مضربی از W خواهد بود.

در اینصورت خواهیم داشت:

$\mathbf{x}^+ = \mathbf{x}^- + \lambda\, \mathbf{w}$ for some value of λ.



- میدانیم که:
  - $\mathbf{w} \cdot \mathbf{x}^+ + b = +1$
  - $\mathbf{w} \cdot \mathbf{x}^- + b = -1$
  - $x^+ = \mathbf{x}^- + \lambda\, \mathbf{w}$
  - $|\, \mathbf{x}^+ - \mathbf{x}^- \,| = M$
- لذا میتوان M را برحسب W و b محاسبه کرد.

محاسبه پهنای حاشیه

- $\mathbf{w} \,.\, \mathbf{x}^{+} + b = +1$
- $\mathbf{w} \,.\, \mathbf{x}^{-} + b = -1$
- $\mathbf{x}^{+} = \mathbf{x}^{-} + \lambda\, \mathbf{w}$
- $|\, \mathbf{x}^{+} - \mathbf{x}^{-} \,| = M$

$$\boxed{\mathbf{w}.(\,\mathbf{x}^{-} + \lambda\, \mathbf{w}) + b = +1}$$

$$\Downarrow$$

$$\boxed{\mathbf{w}.\mathbf{x}^{-} + \lambda\, \mathbf{w}.\mathbf{w} + b = +1}$$

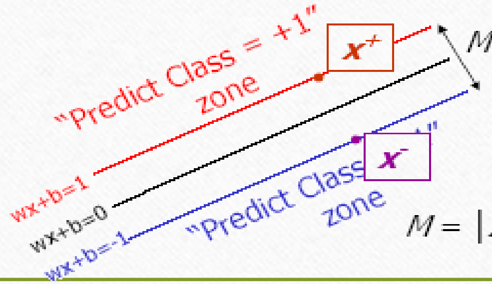
$$\Downarrow$$

$$\boxed{-1 + \lambda\, \mathbf{w}.\mathbf{w} = +1}$$

$$\Downarrow$$

$$\boxed{\lambda = 2/\, \mathbf{w}.\mathbf{w}}$$

محاسبه پهنای حاشیه



M = Margin Width = $\frac{2}{\sqrt{\mathbf{w.w}}}$

$$M = |\boldsymbol{x}^+ - \boldsymbol{x}^-| = |\lambda \boldsymbol{w}| =$$

$$= \lambda\,|\,\mathbf{w}\,| = \lambda\sqrt{\mathbf{w.w}}$$

$$= \frac{2\sqrt{\mathbf{w.w}}}{\mathbf{w.w}} = \frac{2}{\sqrt{\mathbf{w.w}}}$$

What we know:

- $\boldsymbol{w} \cdot \boldsymbol{x}^+ + b = +1$
- $\boldsymbol{w} \cdot \boldsymbol{x}^- + b = -1$
- $\boldsymbol{x}^+ = \boldsymbol{x}^- + \lambda\, \boldsymbol{w}$
- $|\boldsymbol{x}^+ - \boldsymbol{x}^-| = M$
- $\lambda = \frac{2}{\mathbf{w.w}}$

محدودیت

- اگر برای مثال دو بعدی فوق مقدار دسته ها را با 1 و -1 مشخص کنیم داریم:
- $\langle w,x_i \rangle + b \geq 1$ for $y_i=1$
- $\langle w,x_i \rangle + b \leq -1$ for $y_i = -1$
- که میتوان آنها را بصورت زیر نوشت:
- $y_i\,(\langle w,x_i \rangle + b) \geq 1$ for all i

- در SVM بدنبال حل همزمان معادلات زیر هستیم:
- با داشتن داده های آموزشی $(\mathbf{x}i, yi)$ که $i=1,2,\ldots N;\ yi \in \{+1,-1\}$

- <u>Minimise</u> $||w||^2$
- <u>Subject to</u> : $y_i\ (\langle w, x_i \rangle + b) \geq 1$ for all i

  -Note that $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w}$

- این یک مسئله quadratic programming با محدودیت هائی بصورت نامعادلات خطی است. روشهای شناخته شده ای برای چنین مسئله هائی بوجود آمده اند.

$$\min_{w,b}\left(\frac{1}{2}\|w\|^2\right)$$

Subject to :

$$y_i\left(w^T x_i + b\right) \geq 1 \qquad i = 1,2,\cdots,n$$

معمولا به منظور حل مساله ای که در بالا مطرح شده است مساله اولیه از روش ضرایب لاگرانژ استفاده می کنیم .

معمولا به منظور حل مساله ای که در بالا مطرح شده است مساله اولیه از روش ضرایب لاگرانژ استفاده می کنیم . تابع لاگرانژ زیر را تشکیل می دهیم :

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{l}\alpha_i\left(y_i\cdot((x_i\cdot w)+b)-1\right)$$

α_i عبارت است از ضریب لاگرانژ با توجه به iامین نامساوی .

از $L(w,b,\alpha)$ نسبت به w و b مشتق گرفته و نتایج حاصل را برابر با صفر قرار می دهیم . در نهایت دو شرط زیر را برای بهینگی به دست می آوریم :

$$\begin{cases}\dfrac{\partial L(\mathrm{w},b,\alpha)}{\partial \mathrm{w}}=0\\[2ex]\dfrac{\partial L(\mathrm{w},b,\alpha)}{\partial b}=0\end{cases}$$

$$\begin{cases}\mathrm{w}=\sum_{i=1}^{n}\alpha_i y_i \mathrm{x}_i\\[2ex]\sum_{i=1}^{n}\alpha_i y_i=0\end{cases}$$

$$\max_{\alpha} W(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathrm{x}_i^T \mathrm{x}_j$$

$$Subject\ to:$$

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

$$\alpha_i \geq 0 \qquad i=1,2,\cdots,n$$

همزمان:

$$\alpha_i\left[y_i\left(\mathrm{w}^T \mathrm{x}_i+b\right)-1\right]=0 \qquad i=1,2,\cdots,n$$

بنابراین تنها آن بردارهای پشتیبانی (x_i, y_i) که کم ترین فاصله را از ابرصفحه بهینه داشته و حاشیه ماکسیمال را تعیین می کنند متناظر با α_iهای غیر صفر می باشند . بقیه α_i ها برابر با صفر هستند .

پس از تعیین ضرایب لاگرانژ بهینه، می توانیم بردار وزن بهینه w^* را محاسبه نماییم :

$$\mathrm{w}^*=\sum_{i=1}^{n}\alpha_i^* y_i \mathrm{x}_i$$

سپس به کمک بردار پشتیبانی مثبت، مقدار بهینه اسکالر b را به صورت زیر تعیین می کنیم :

$$b^*=1-\mathrm{w}^{*T}\mathrm{x}_s \qquad for \qquad y_s=+1$$

فرمول لاگرانژ..

- توجه شود که برچسب های مختلف لاگرانژ تاکید بر این نکته دارد که (($L(w,b,a)$ برای اصلی و D برای دوگان) از تابع هدف یکسان ولی با شرایط متفاوت؛ دو راه-حل،از دو مسیر زیر بدست می آید:
  - مینیمم کردن $L(w,b,a)$
  - ماکزیمم کردن L_D
- محاسبه a_i ، به حل معادله بالا، موسوم به quadratic problem منجر شود.

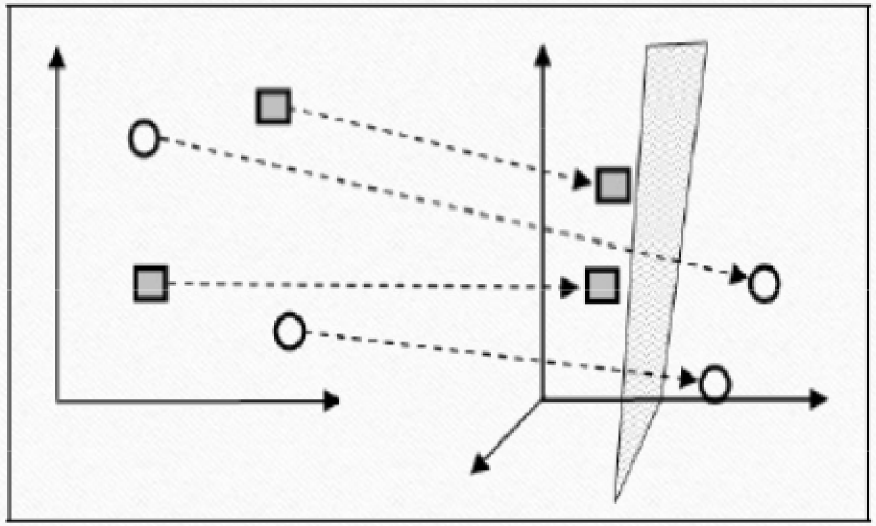
- **مسئله جداسازی غیر خطی:**
  - ✓ SVC با حاشیه ماکسیمال، نقطه آغازین الگوریتم های SVM قلمداد می شود.
  - ✓ زمانی که تفکیک خطی نمونه ها بطور کامل شدنی نیست این امکان وجود دارد که حاشیه ها منفی باشند. در چنین مواردی فضای شدنی مساله اولیه تهی و حل مساله بهینه سازی غیر ممکن می باشد.

- **دو رویکرد برای برای حل مسائل تفکیک ناپذیر(غیرخطی):**
  - ✓ بهینه سازی با حاشیه های انعطاف پذیر (نرم)
  - ✓ بکارگیری kernel trick به منظور خطی سازی مسائل غیر خطی مورد نظر

جدائی پذیری خطی و غیر خطی، مفهوم کرنل

- در حالتی که جدا پذیری بصورت خطی نباشد (جدا پذیری غیر خطی) ایده اصلی این است که نمونه ها را به یک فضای با بعد بالا (feature space) فضای مشخصه نگاشت دهیم که در فضای جدید مشخصه ها، نمونه ها می توانند به صورت خطی از هم جدا شوند .
- این امر نیاز به اعمال یک تابع هسته (کرنل) را به همراه خواهد آورد.

نمایش دستیابی به جدایی پذیری خطی، برای مسئله ای که دارای این خاصیت نیست، به کمک نگاشت



Kernel Mapping from input to feature space.

SVC با حاشیه انعطاف پذیر(نرم) و بهینه سازی:

- ✓ مواردی را در نظر بگیرید که در آن ها تعدادی از نقاط متعلق به کلاس های مختلف با یکدیگر مخلوط شده اند . این نقاط بیانگر خطای آموزش می باشند
- ✓ هدف از بکارگیری حاشیه انعطاف پذیر توسعه دادن الگوریتم SVC است به نحوی که در تعیین ابرصفحه امکان وجود داده های حاوی اختلال نیز فراهم شود .

به منظور دستیابی به این هدف یک متغیر کمبود ξ_i معرفی می شود تا از این طریق بتوان مقدار خطای دسته بندی را در نظر گرفت :

$$\min_{w,b}\left(\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i\right)$$

$$Subject\ \ to:$$

$$y_i\left(w^T x_i+b\right)\geq 1-\xi_i \qquad \xi_i\geq 0 \quad i=1,2,\cdots,n$$

نقش C در رابطه فوق ایجاد تعادل میان پیچیدگی ماشین و تعداد نقاط تفکیک ناپذیر است . این پارامتر توسط کاربر و بر اساس تجربه یا تحلیل تعیین می شود .
متغیر کمبود ξ_i بیانگر فاصله موجود میان ابرصفحه و داده ای است که دسته بندی آن نادرست می باشد .در واقع این فاصله میزان انحراف یک نمونه را از وضعیت ایده ال تفکیک پذیری می سنجد.

با استفاده از روش ضرایب لاگرانژ می توان مساله دوگان متناظر با حاشیه انعطاف پذیر را به صورت زیر فرموله کرد :

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{Subject to:}$$

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

$$0 \le \alpha_i \le C \qquad i = 1, 2, \cdots, n$$

در حالت تفکیک ناپذیر شرط مکمل کروش - کان - تاکر به صورت زیر است

$$\alpha_i \left[y_i \left(\mathbf{w}^T \mathbf{x}_i + b \right) - 1 + \xi_i \right] = 0 \qquad i = 1, 2, \cdots, n$$

و

$$\gamma_i \xi_i = 0 \qquad i = 1, 2, \cdots, n$$

در نقطه زینی $\alpha_i + \gamma_i = C$ است. بنابراین

$$\xi_i = 0 \quad if \quad \alpha_i \prec C$$

در نتیجه وزن بهینه برابر است با :

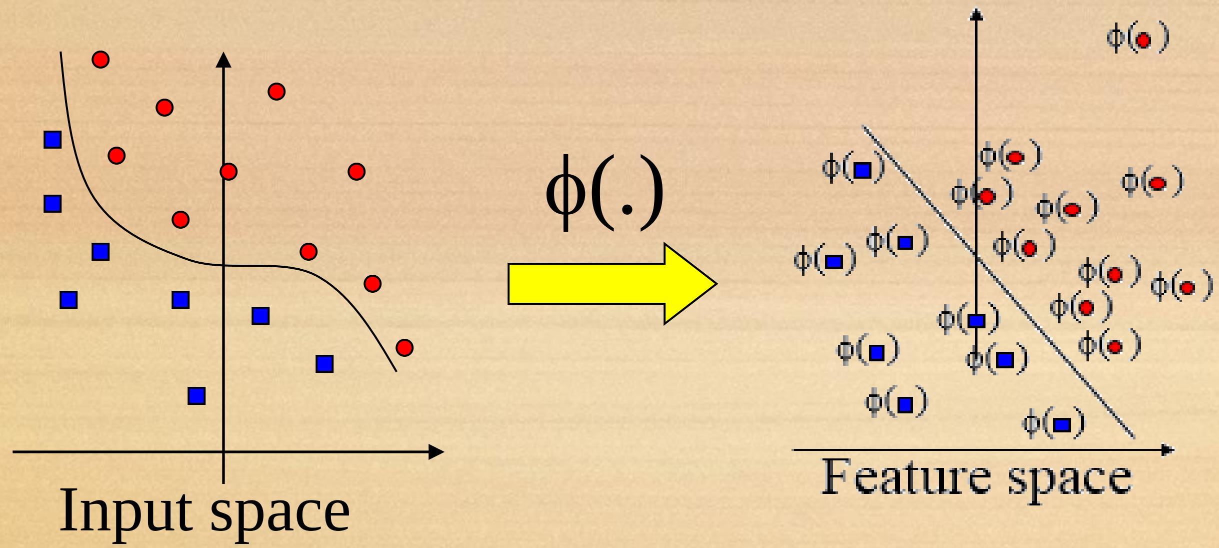
$$\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i$$

مقدار بهینه برای اسکالر b با توجه به رابطه اول و هر نقطه داده ای دلخواه درمجموعه آموزشی که برای آن داشته باشیم $0 \le \alpha_i \le C$ به دست می آید

➢ Kernel trick:

- ✓ بر اساس ضرب داخلی داده های مفروض یک تابع کرنل مناسب تعریف می شود
- ✓ با یک تبدیل غیر خطی از فضای ورودی به فضای خصیصه با ابعاد بیش تر (حتی نامتناهی) مواجه هستیم تا از این طریق بتوان مسائل را به صورت خطی تفکیک پذیر ساخت
- ✓ این رویکرد را می توان با توجه به **قضیه پوشش** در مورد تفکیک پذیری الگوها توجیه کرد : احتمال آنکه یک مساله پیچیده دسته بندی الگو در فضایی با ابعاد بیش تر، به صورت خطی تفکیک پذیر باشد بیش تر از احتمال تفکیک پذیری خطی این مساله در فضایی با ابعاد کم تر است .

فرض کنید که $\Phi: X \rightarrow H$ تبدیلی غیر خطی از فضای ورودی X به فضای خصیصه H باشد که در آن مساله به صورت خطی تفکیک پذیر است.



- انجام محاسبات در فضای ویژگی میتواند پرهزینه باشد برای اینکه ابعاد بیشتری دارد.
- در حالت کلی ابعاد این فضا بی نهایت است.
- برای غلبه بر این مشکل از kernel trick استفاده میشود.

کرنل ضرب داخلی: کرنل، برای تمامی $x,x'\epsilon X$ یک تابع $K(x,x')$ بوده و به صورت زیر تعریف می شود:

$$K(x,x')=\Phi^T(x)\Phi(x')$$

- ✓ علاوه بر توابع کرنل خطی می توان از توابع کرنل چند جمله ای یا سیگموئید نیز استفاده کرد . در سال های اخیر تحقیقات زیادی درباره توابع کرنل مختلف انجام شده است .
- ✓ به دلیل پیچیدگی مسائل Kernel trick همیشه تضمین نمی کند که آن ها بطور کامل تفکیک پذیر خطی باشند، بنابراین در اغلب اوقات ایده های مربوط به حاشیه انعطاف پذیر و Kernel trick را یکپارچه ساخته تا از مزایای هر دو رویکرد بهره مند شویم.
- ✓ در این حالت فرم دوگان مساله بهینه سازی به صورت زیر خواهد بود :

$$\max_\alpha W(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i,x_j)$$

$$\text{Subject to:}$$

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

$$0\le\alpha_i\le C \qquad i=1,2,\cdots,n$$

با استفاده از روش ضرایب لاگرانژ دسته بندی کننده بهینه را به دست می آوریم:

$$f(x)=\sum_{i=1}^{n}\alpha_i^* y_i K(x_i,x)+b^*$$

درنتیجه

$$.b^*=1-\sum_{i=1}^{n}\alpha_i^* y_i K(x_i,x_s)$$

➢ مثال:

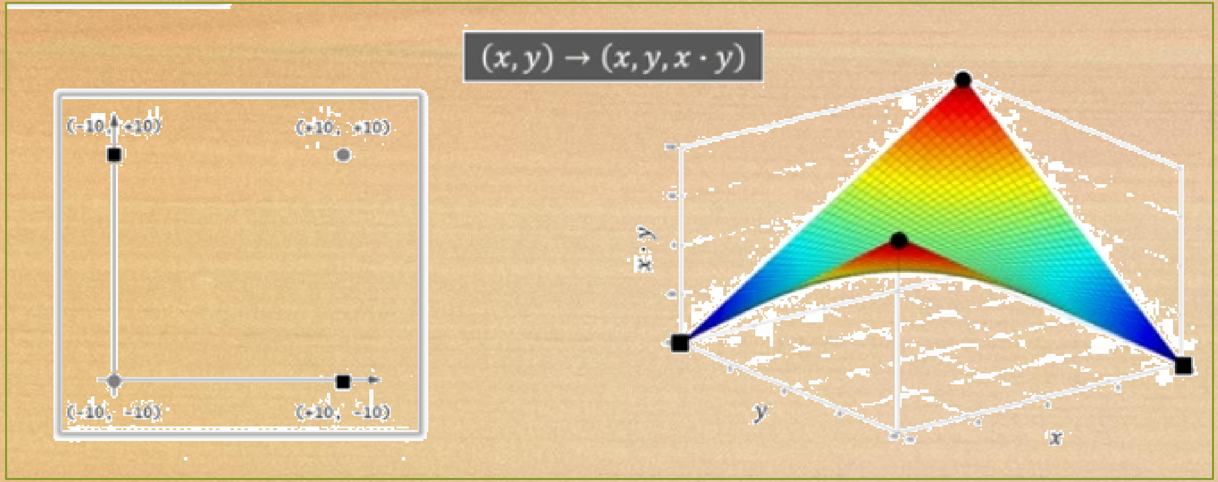
مساله XOR یکی از مسائل دسته بندی بوده که به صورت خطی تفکیک پذیر نمی باشد . به منظور نمایش اهمیت یکپارچه سازی SVC با حاشیه انعطاف پذیر و Kernel trick در حل مسائل پیچیده دسته بندی از این مساله استفاده می کنیم. یک مجموعه داده XOR دو بعدی را می توان به تصادف و توسط چهار توزیع گوسی متفاوت تولید کرد . " * " و "•" نمونه های متعلق به دو کلاس را نمایش می دهند .

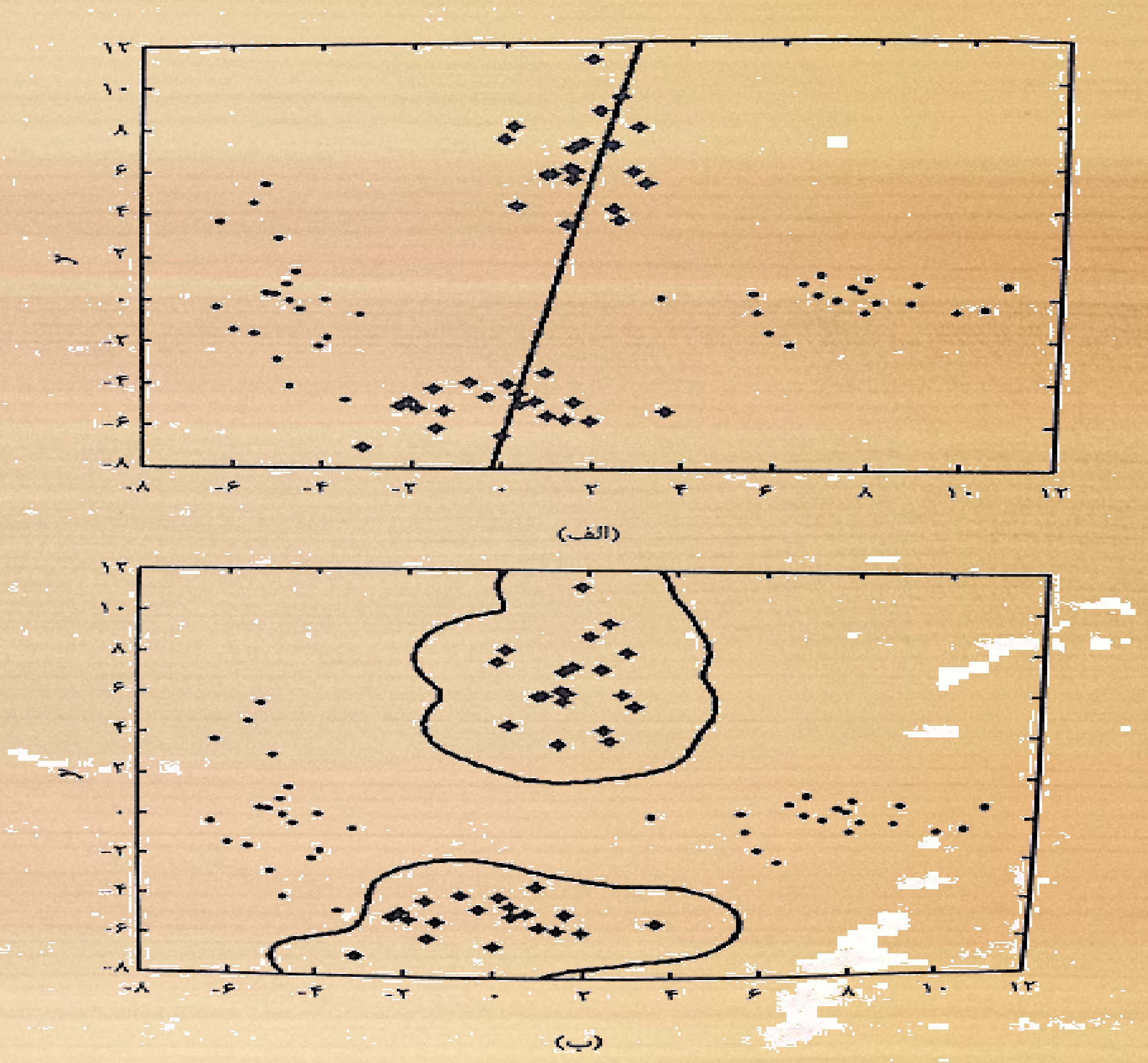
همانگونه که در شکل الف نشان داده شده است SVC مقدماتی و کرنل خطی در حل مساله XOR با شکست کامل مواجه می شوند. یک مرز خطی توانایی تفکیک دو کلاس مورد نظر را ندارد . همانگونه که ملاحظه می کنید این مرز خطی تمامی نمونه ها را به دو قسمت تقسیم کرده است . بنابراین این رویکرد نمی تواند هدف مورد نظر از دسته بندی را برآورده سازد .

در نتیجه از ترکیب SVC با حاشیه انعطاف پذیر و کرنل با پایه شعاعی گوسی به منظور حل این مساله استفاده می کنیم.

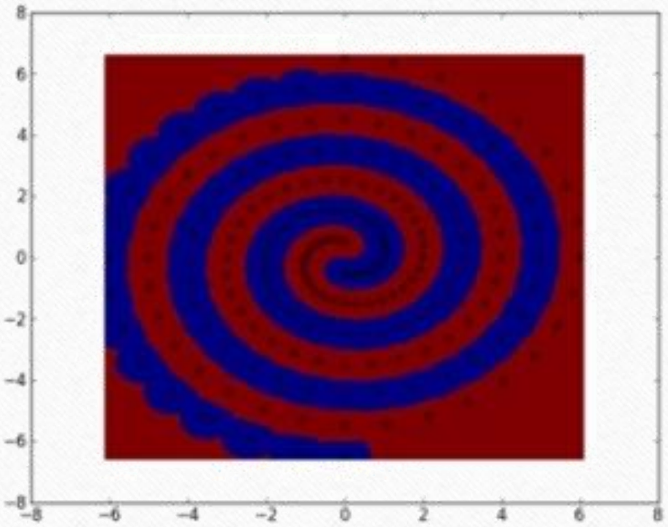
$$K(x_i, x) = \exp\left(-\frac{\|x-x_i\|^{۲}}{۲\sigma^{۲}}\right)$$

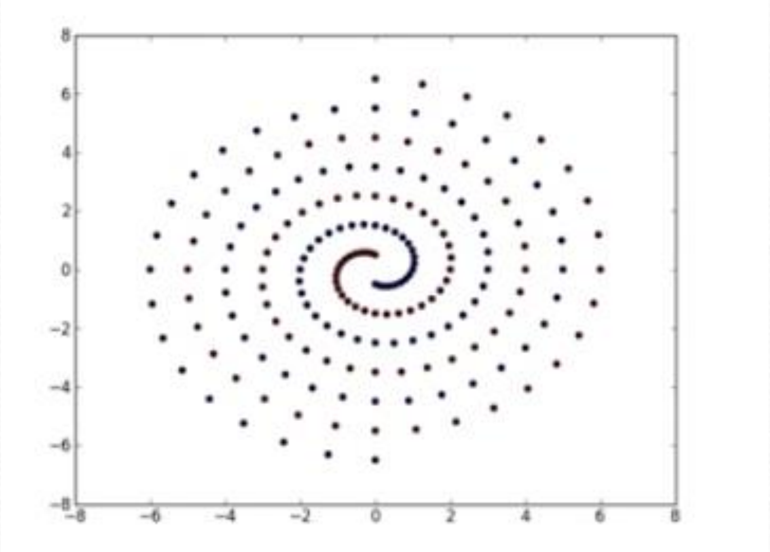
پارامتر C را برابر با 1 و پارامتر کرنل را مساوی با 1 قرار می دهیم $(\sigma = 1)$. مرز تفکیک کننده متناظر را در شکل ب ملاحظه می فرمایید با استفاده از Kernel trick مرز دیگر خطی نبوده و تنها یک کلاس را شامل می شود. با بررسی نمونه هایی که در داخل و خارج از مرز قرار دارند در می یابیم که دسته بندی کننده نمونه ها را به درستی دسته بندی کرده است.


$(x, y) \rightarrow (x, y, x \cdot y)$
(-10, +10)
(+10, +10)
(-10, -10)
(+10, -10)
x · y
y
x


(الف)
(ب)

مثال تابع کرنل گوسی:





مقدار مناسب برای پارامترهای تابع کرنل چگونه تعیین می‌شود؟

- ❑ پارامتر C
  - مقادیر کوچکتر: بایاس بیشتر، واریانس کمتر
  - مقادیر بزرگتر: بایاس کمتر، واریانس بیشتر
- ❑ پارامتر σ
  - مقادیرکوچکتر: بایاس کمتر، واریانس بیشتر
  - مقادیر بزرگتر: بایاس بیشتر، واریانس کمتر

➢ ماشین‌های بردارپشتیبانی و شبکه‌های عصبی:

- توسعه شبکه های عصبی قبل از آنکه براساس اصول نظری انجام گرفته باشد، براساس روش های ابتکاری و تعداد زیادی از کاربردها و آزمایش ها صورت گرفته است ولی ماشین های بردار پشتیبان برعکس.
- از مزیت های SVMها آن است که جواب یک SVM سراسری و یکتا است، درحالیکه شبکه های عصبی عمدتا از چندین کمینه محلی رنج می برند.
- برخلاف شبکه های عصبی، پیچیدگی محاسباتی مربوط به ماشین های بردار پشتیبانی به ابعاد فضای ورودی بستگی ندارد.
- یکی از دلایلی که به موجب آن عملکرد ماشین های بردار پشتیبانی در اغلب موارد از عملکرد شبکه های عصبی بهتر است، کمتر رخ دادن مشکل برازش بیش از اندازه در SVMها است.
- در اغلب موارد از کرنل های گوسی استفاده می شود.

➢ نقاط ضعف ماشین های بردار پشتیبانی:

با وجود آنکه SVMها از دیدگاه عملی و کاربردی دارای مزیت های قابل توجهی هستند، اما محدودیت هایی نیز دارند.

- یکی از پرسش هایی که هنوز پاسخ کاملی برای آن نیست، نحوه انتخاب پارامترهای تابع کرنل است.
- محدویت دوم، در رابطه با اندازه و سرعت است (هم در آموزش و هم آزمایش).
- پردازش داده های گسسته یکی دیگر از مشکلات است.

با این وجود، SVM ها دارای یک شالوده نظری منسجم بوده و جواب های ایجاد شده توسط آنها، سراسری و یکتا است. امروزه ماشین های بردار پشتیبانی به متداول ترین تکنیک های پیش بینی در داده کاوی تبدیل و در بیشتر ابزارهای تجاری داده کاوی بکار گرفته شده اند.