SVM

ماشین‌های بردار پشتیبان

دانشگاه شهید بهشتی
پژوهشکده‌ی فضای مجازی
پاییز ۱۴۰۱
احمد محمودی ازناوه

فهرست مطالب

- ماشین بردار پشتیبان(SVM)
  - تاریخچه
  - معرفی
  - داده‌های جدایی‌پذیر خطی(Hard Margin)
- Soft Margin
- مجموعه‌های جدایی‌ناپذیر خطی
  - نگاشت به فضایی با ابعاد بالا
  - Inner product kernel
- SVR
- مواجهه با مجموعه‌داده‌های imbalanced

تاریخچه

- نسخه‌ی اولیه‌ی SVM توسط آقای Vladimir Vapnik ارائه شد.
- Vapnik با همکاری خانم Corinna Cortes استاندارد کنونی SVM را در سال ۱۹۹۳ پایه‌ریزی کرده و در سال ۱۹۹۵ منتشر نمودند.

Cortes, C. and V. Vapnik (1995). "Support-vector networks." Machine Learning 20(3): 273-297.

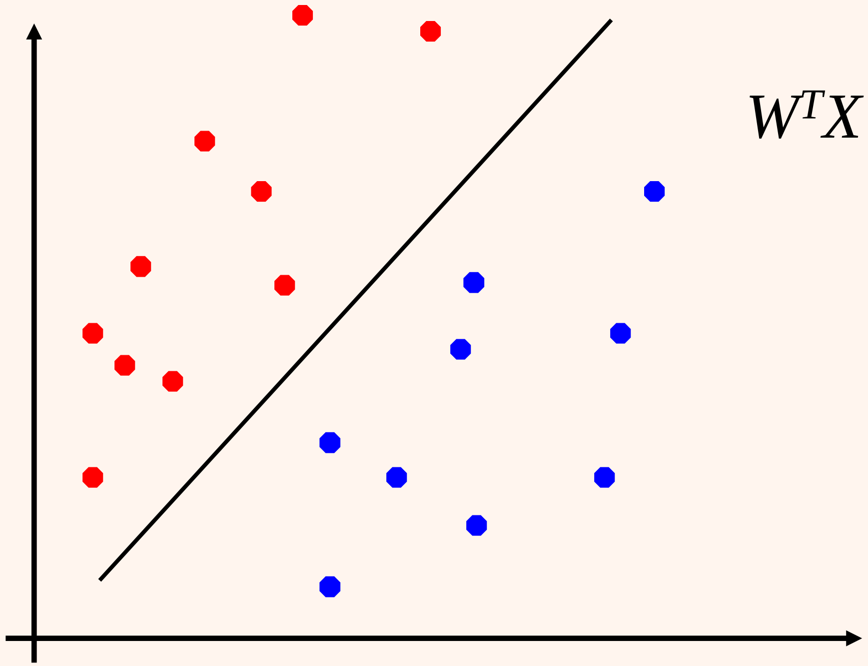
معرفی

- یک جداکننده‌ی خطی را می‌توان همانند شکل زیر در نظر گرفت.

$$W^T X + B > 0$$

$$W^T X + b = 0$$

$$W^T X + b < 0$$

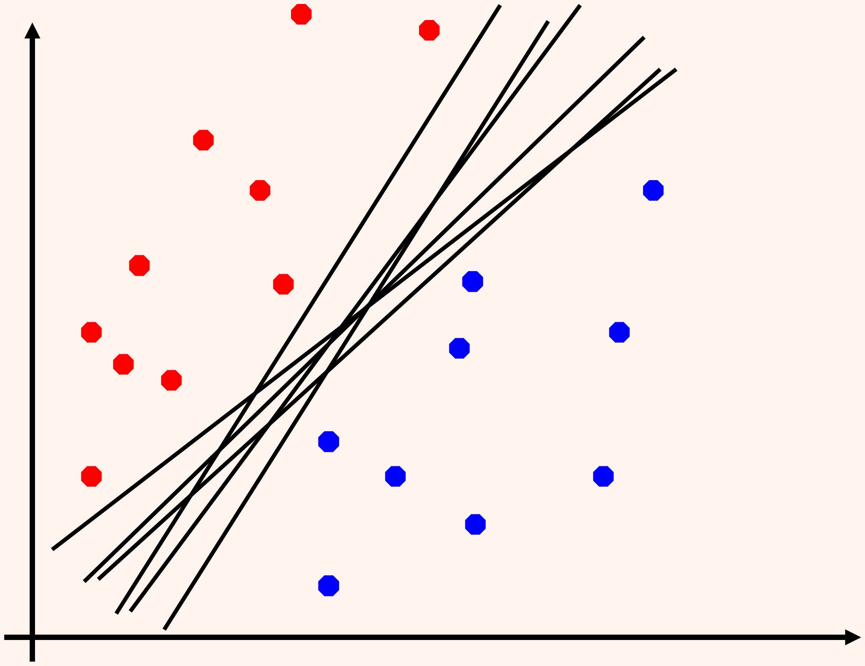


$$F(X) = SIGN(W^T X + b)$$

www.cs.utexas.edu/~mooney/cs391L/slides/svm.ppt

مرز بهینه

- سوال
  - کدام یک از مرزها، مرزی بهینه برای جداسازی است؟

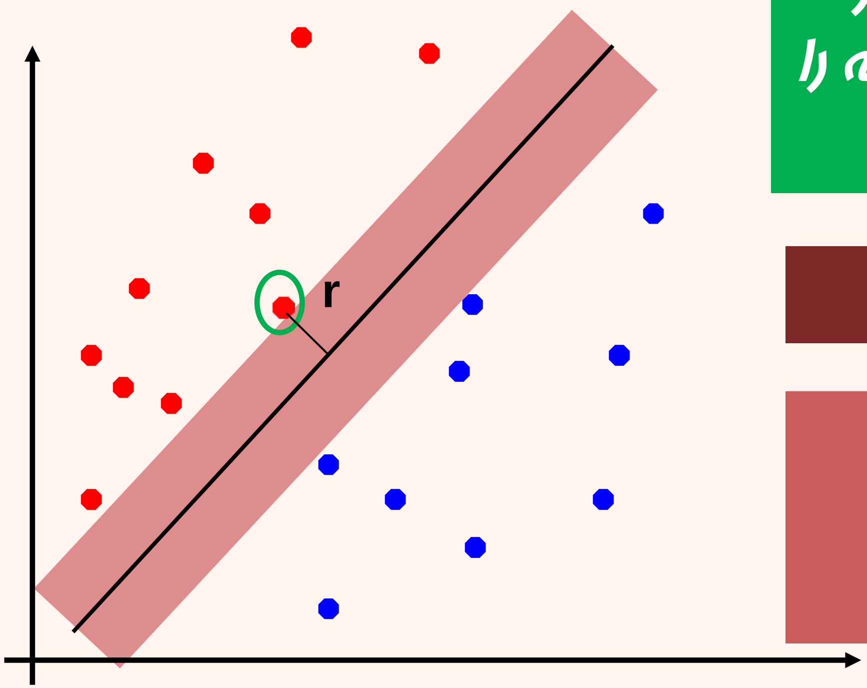


www.cs.utexas.edu/~mooney/cs391L/slides/svm.ppt

مرز جداسازی

- می‌خواهیم به گونه‌ای بهترین مرز جداسازی را به‌دست آوریم.

Margin of separation

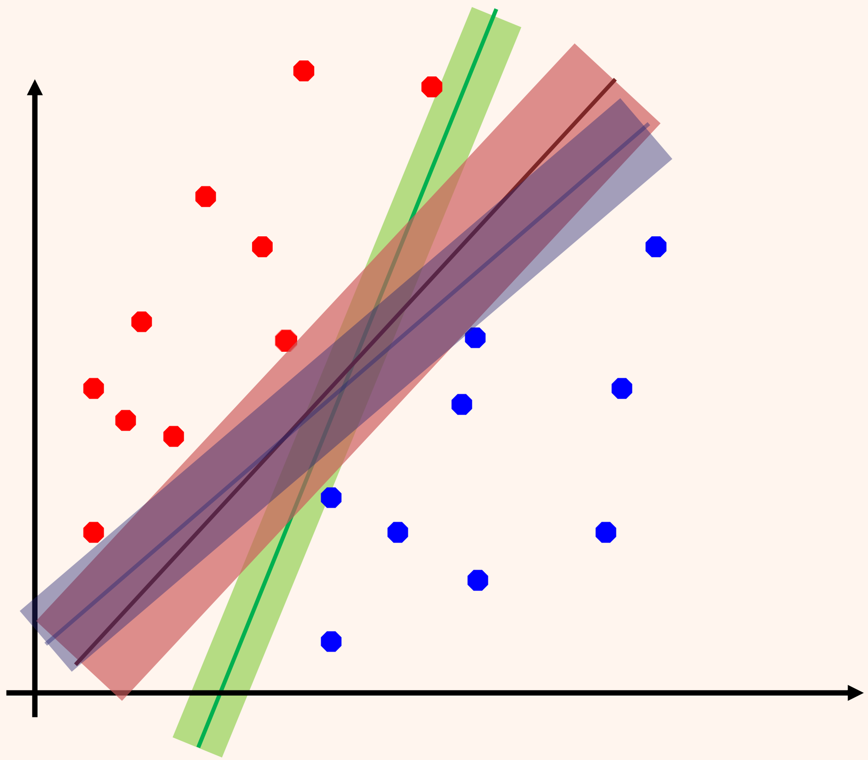


فرض کنیم نزدیک‌ترین نقطه به مرز جداسازی در نظر گرفته شده و فاصله را r بنامیم.

هدف ماکزیمم نمودن r است.

یک حاشیه مشخص می‌کنیم هر مرزی که حاشیه‌ی پهن‌تری را نتیجه دهد، بهتر است.

مرز بهینه



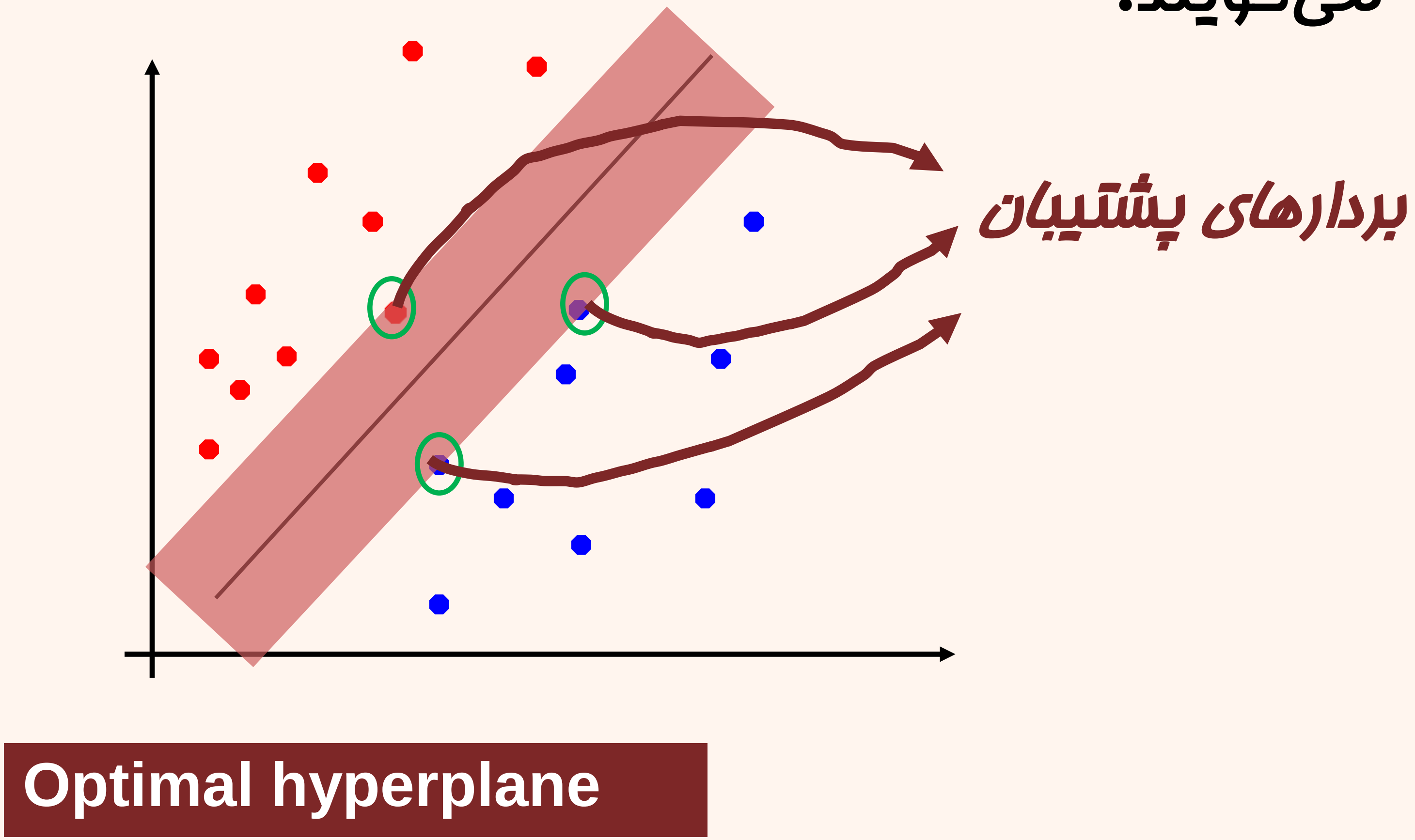
حاشیه‌ی ماکزیمم

- ماکزیمم نمودن حاشیه (Margin) ایده‌ی خوبی است جهت جداسازی خطی، این شیوه را LSVM یا Linear SVM می‌نامند.
- در این حالت نمونه‌هایی که به روی مرز حاشیه هستند، از اهمیت ویژه‌ای برخوردارند.
- بدین‌وسیله می‌توان از نمونه‌های دیگر صرفنظر کرد و تنها به نمونه‌های مهم روی مرز حاشیه پرداخت.

Support Vector

بردار پشتیبان

- به نمونه‌های روی مرز حاشیه «بردار پشتیبان» می‌گویند.



مرز جداسازی

- برای معادله‌ی مرز جداسازی داشتیم:

$$W^T X + b = 0$$

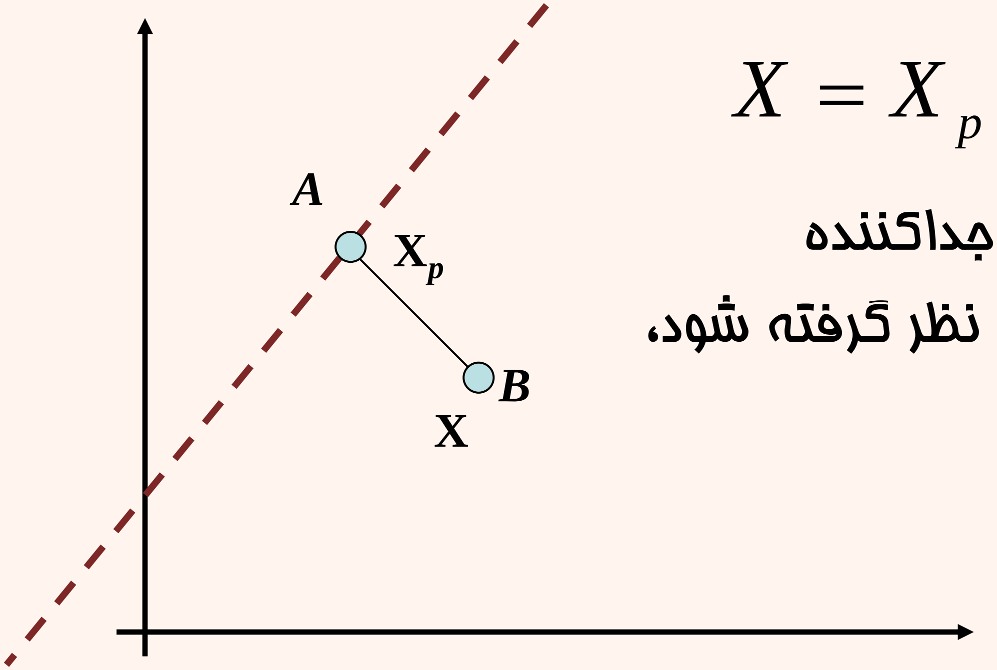
$$(X_i, d_i = +1) \qquad W^T X_i + b > 0$$

$$(X_i, d_i = -1) \qquad W^T X_i + b < 0$$

- فرض کنیم مرز بهینه توسط $\boldsymbol{W}_{op}$ و b_{op} مشخص شود.
- اگر نزدیک‌ترین نقطه به مرز جداسازی را در نظر گرفته، فاصله را «**r**» بنامیم.
- هدف
  – ماکزیمم نمودن فاصله یا همان $\rho=2\boldsymbol{r}$ است.

مرز جداسازی (ادامه...)

- در صورتی که X بردار پشتیبان باشد، طبق شکل زیر خواهیم داشت:



$$X = X_p + \overrightarrow{\mathbf{AB}}$$

- AB در جهت عمود بر مرز جداکننده
- اگر اندازه‌ی بردار AB=r در نظر گرفته شود، خواهیم داشت:

$$X = X_p + r\frac{W_{op}}{\|W_{op}\|} \qquad \overrightarrow{AB} = r\frac{W_{op}}{\|W_{op}\|}$$

مرز جداسازی (ادامه...)

- داشتیم:

$$g(X) = W_{op}^{\ T} X + b_{op}$$

$$X = X_p + r\frac{W_{op}}{\|W_{op}\|}$$

$$g(X) = W_{op}^T [X_p + r\frac{W_{op}}{\|W_{op}\|}] + b_{op}$$

$$g(X) = \underbrace{W_{op}^T X_p + b_{op}}_{\text{روی مرز پس برابر با صفر}} + r\frac{W_{op}}{\|W_{op}\|} W_{op}^T$$

$$g(X) = r\frac{\|W_{op}\|^2}{\|W_{op}\|}$$

$$\boxed{g(X) = r\|W_{op}\|}$$

مرز جداسازی (ادامه...)

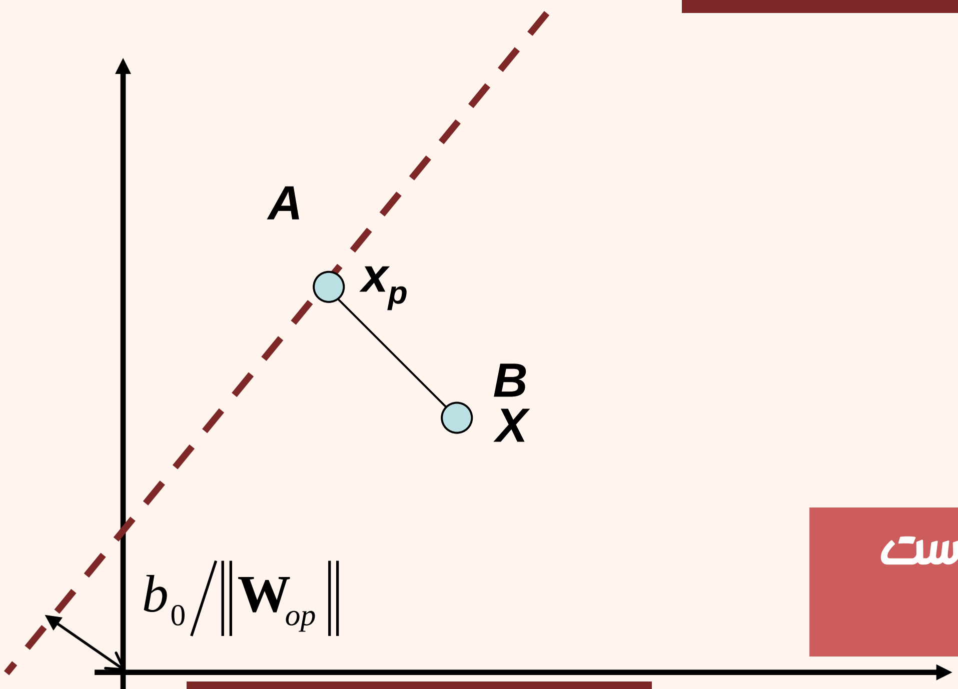
$$g(X) = W_{op}^{\ T} X + b_{op}$$

$$\boxed{g(X) = r\left\|W_{op}\right\|}$$

$$r = \frac{g(X)}{\left\|W_{op}\right\|}$$

هدف ماکزیمم نمودن r است.

در این حالت تحت شرایطی می‌باید W کمینه گردد.



فاصله از مبدا مختصات

$X=0$

$$r = \frac{g(X)}{\left\|W_{op}\right\|} = \frac{b_{op}}{\left\|W_{op}\right\|}$$

مثبت یا منفی بودن bop نشان دهنده‌ی این است
که مبدأ در کدام سمت خط مرزی است.

مرز جداسازی (ادامه...)

مساله یافتن W_{op} و b_{op} است

- جداساز خطی را به صورت زیر در نظر می‌گیریم:

$$(X_i,+1) \quad W_{op}^T X_i + b_{op} \geq 1 \quad for\ d_i = +1$$

$$(X_i,-1) \quad W_{op}^T X_i + b_{op} \leq -1 \quad for\ d_i = -1$$

به صورت کلی داریم:

$$d_i(W_{op}^T X_i + b_{op}) \geq 1$$

- رابطه‌ی بالا برای تمامی الگوهای آموزشی برقرار است.
- و در نتیجه برای بردارهای پشتیبان

$$g(X^s) = W_{op}^T X^s + b_{op} = \pm 1$$

مرز جداسازی (ادامه...)

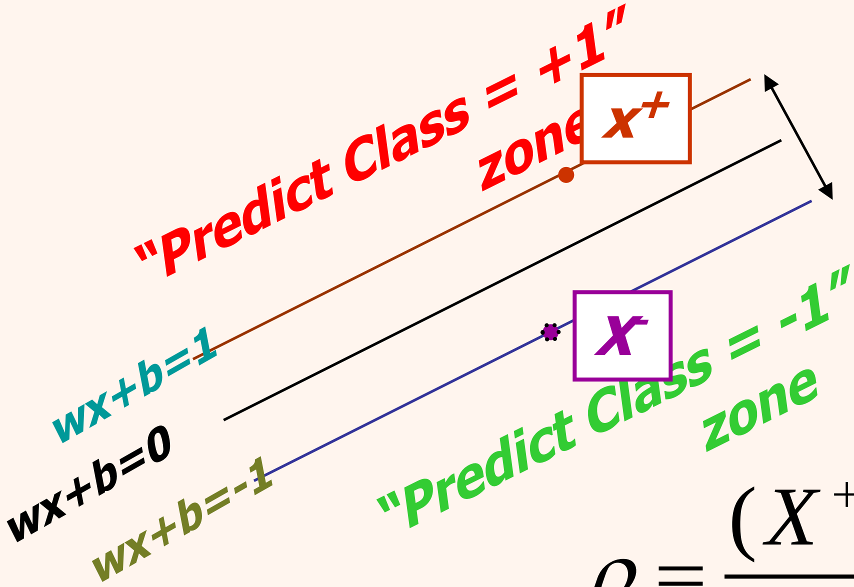
$$g(X^s) = W_{op}{}^T X^s + b_{op} = \pm 1$$

$$r = \frac{g(X^s)}{\|W_{op}\|} = \begin{cases} \dfrac{1}{\|W_{op}\|} \\ -\dfrac{1}{\|W_{op}\|} \end{cases}$$

- در نتیجه فاصله‌ی دو بردار پشتیبان در دو طرف مرز:

$$\rho = 2r = \frac{2}{\|W_{op}\|}$$

جدایی‌پذیر خطی



ρ=Margin Width

$$\rho = \frac{(X^+ - X^-)\cdot W}{\|W\|} = \frac{2}{\|W\|}$$

می‌دانیم:

- **W** . **X**$^+$ *+ b = +1*
- **W** . **X**$^-$ *+ b = -1*
- **W** . *(***X**$^+$**-X**$^-$*) = 2*

جدایی‌پذیر خطی

$$\rho = 2r = \frac{\|\pm 2\|}{\|W_{op}\|}$$

- با توجه به دو رابطه‌ی

$$d_i(W_{op}{}^T X + b_{op}) \ge 1$$

- به این نتیجه می‌رسیم که $\mathbf{W}_{\mathrm{op}}$ می‌باید مینیمم گردد.
- این مسأله معادل مینیمم کردن

$$\Phi(W) = \frac{1}{2} W^T W$$

  - Φ یک تابع محدب (Convex Function) است.
  - طبق رابطه‌ی $d_i(W_{op}{}^T X_i + b_{op}) \ge 1$ برای N الگوی آموزشی شروط زیر می‌باید برقرار باشد:

$$[d_i(W_{op}{}^T X_i + b_{op}) - 1] \ge 0$$

این میزان بزرگ‌تر یا مساوی صفر است

خلاصه

وزن‌ها و بایاس را به گونه‌ای بیابید که:

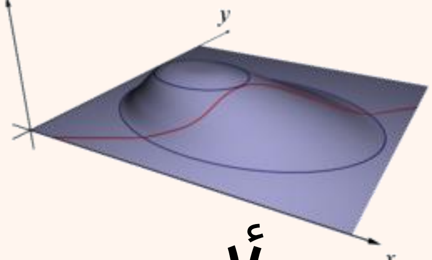
$$\rho = \frac{2}{\|W\|} \text{ is maximized}$$

and for all (X_i, d_i), i=1..n : $\quad d_i(W^T X_i + b) \geq 1$

وزن‌ها و بایاس را به گونه‌ای بیابید که:

$\Phi(\boldsymbol{W}) = 1/2\ \|W\|^2 = 1/2 W^T W$ is minimized

and for all (X_i, d_i), i=1..n : $\quad d_i\ (\boldsymbol{W}^T \boldsymbol{X}_i + b) \geq 1$



یافتن بهینه

Lagrange multiplier

- از روش «ضرایب لاگرانژ» برای حل این مسأله‌ی بهینه‌سازی استفاده می‌شود:
  - در این شیوه به تابع هدف اولیه قید مورد نظر را اضافه می‌کنیم.
  - قید عبارتی است که همواره بزرگتر یا مساوی صفر است.

$$J(W,b,\boldsymbol{\alpha}) = \frac{1}{2}W^T W - \sum_{i=1}^{N} \alpha_i [d_i (W^T X_i + b) - 1]$$

به ازای هر نمونه یک قید داریم

- به سادگی می‌توان نشان داد که نقطه بهینه نقطه زینی این رابطه است:

$$\min_{W,b} \max_{\alpha \geq 0} \; J(W,b,\alpha)$$

$$
\begin{cases}
= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i d_i \alpha_j d_j X_i^T X_j \\
\sum_{i=1}^{N} \alpha_i d_i = 0 \\
\alpha_i \geq 0 \ for \ 1 = 0, 1, \ldots.., N
\end{cases}
$$

α_i ها وابسته به ضرب داخلی نمونه‌های آموزشی خروجی های مرتبط است

یافتن بهینه

Karush–Kuhn–Tucker(KKT) condition of optimization theory

- با توجه به شرط KKT خواهیم داشت:

$$\alpha_i = 0 \quad \Rightarrow \quad [d_i(W^T X_i + b) - 1] > 0$$

$$\alpha_i > 0 \quad \Rightarrow \quad [d_i(W^T X_i + b) - 1] = 0$$

- در نتیجه تنها α_i متناظر با بردارهای پشتیبان غیرصفر خواهد بود.

یافتن رویه‌ی بهینه

- پس از به دست آوردن α خواهیم داشت:

$$W_{op} = \sum_{i=1}^{N} \alpha_i d_i X_i$$

$$X^s = X^{\text{support vector}} \Rightarrow W_{op}^T X^s + b_{op} = \pm 1$$

$$\Rightarrow b_{op} = \pm 1 - W_{op}^T X^s$$

یافتن رویه‌ی بهینه

$W = \sum \alpha_i d_i X_i$ $\qquad$ $b = d_k - W^T X_k$ for any X_k such that $\alpha_k \neq 0$

هر α_i مخالف صفر، نشان‌دهنده‌ی این است که X_i متناظرش یک بردار پشتیبان است.

در این حالت تابع جداکننده همانند زیر است:

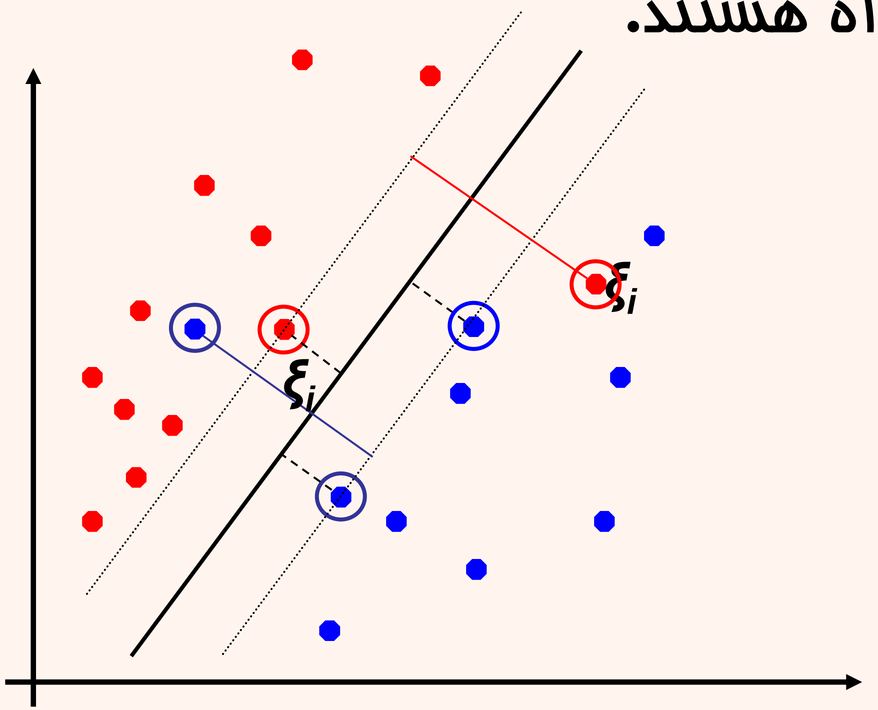
$$g(X) = \sum \alpha_i d_i \underbrace{X_i^T X}_{\text{ضرب داخلی دو بردار}} + b$$

توجه:

حل مساله بهینه‌سازی وابسته به محاسبه ضرب داخلی بین تمامی نمونه‌های آموزشی است.

Soft Margin

- SVM برای داده‌های جدایی‌پذیر خطی مورد بررسی قرار گرفت.
- حال اگر مجموعه‌ی داده‌های آموزش قابلیت جداسازی خطی بدا خطای صفر را نداشته باشند، چه خواهد شد؟ به بیان بهتر صحبت در مورد مسائل جدایی‌پذیر است که با نویز همراه هستند.

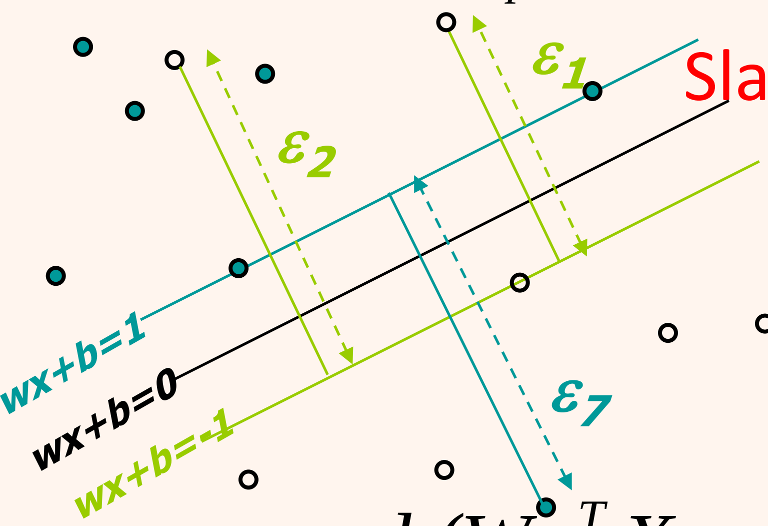


Soft Margin

- مسأله‌ی Hard Margin را تبدیل به حل مسأله‌ی Soft Margin می‌شود.
- حاشیه‌ی جداسازی soft گفته می‌شود، در صورتی که برای برخی داده‌ها شرط زیر نقض شود:

$$d_i(W_{op}{}^T X + b_{op}) \geq 1$$

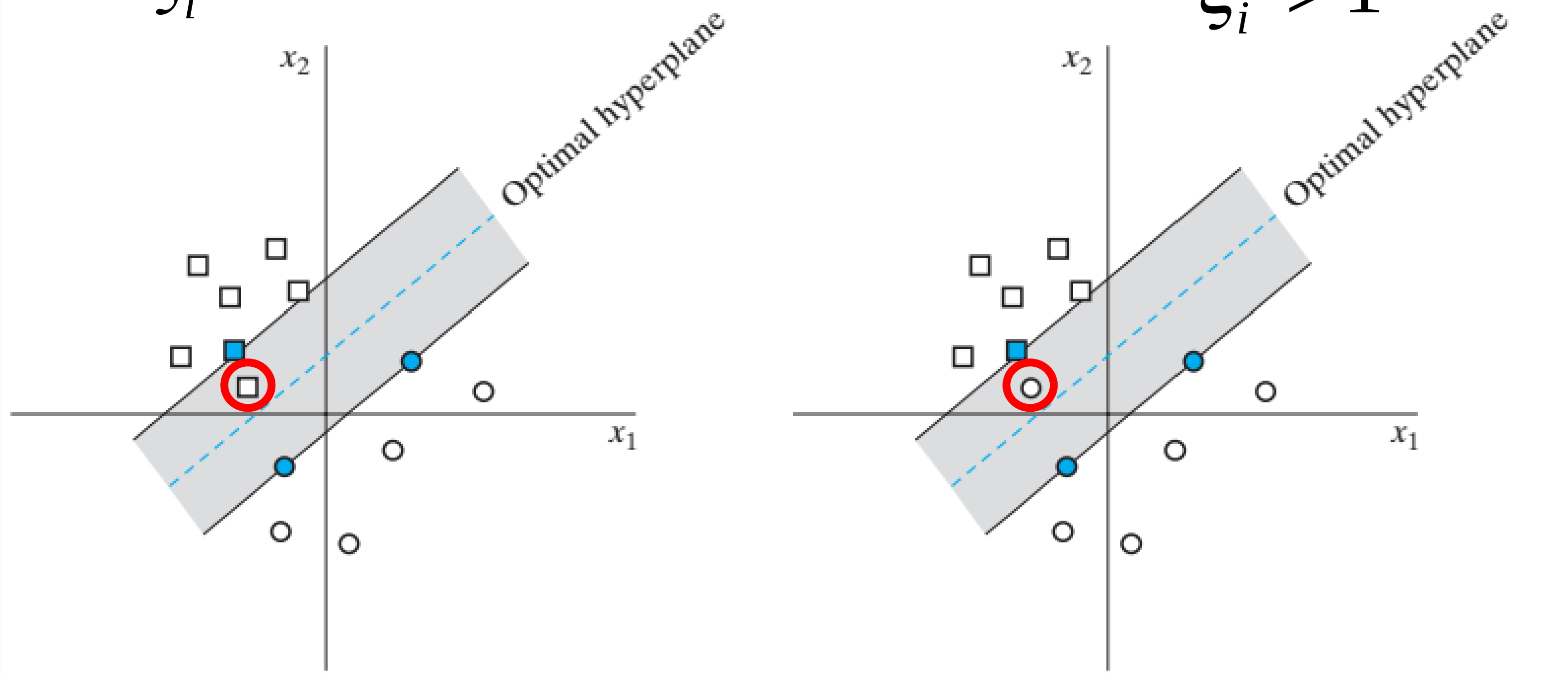
- با اضافه کردن یک Slack Variable مسأله را بار دیگر بررسی می‌کنیم.
- این متغیر میزان انحراف از شرط فوق را نشان می‌دهد.



$$d_i(W_{op}{}^T X_i + b_{op}) \geq 1 - \xi_i, \qquad i = 1, 2, \cdots, N$$

Soft Margin Classification

$0 \le \xi_i \le 1$ $\xi_i > 1$



- دو حالت ممکن است رخ دهد:
  - داده‌ی در کلاس درست ولی در حاشیه قرار گیرد.
  - داده‌ی آموزشی به اشتباه دسته‌بندی شود.

$$d_i(W_{op}{}^T X_i + b_{op}) \ge 1 - \xi_i, \qquad i = 1, 2, \cdots, N$$

Soft Margin Classification

$$d_i(\mathbf{W}_{op}^T\mathbf{X}+b_{op})\geq 1-\xi_i,\qquad i=1,2,\cdots,N$$

- در این حالت بردارهای پشتیبان آن‌هایی هستند که در رابطه‌ی تساوی در عبارت بالا صدق می‌کنند، حتی با وجود ξ>0
  - در صورتی که داده‌های نویزی از مجموعه خارج شود، رویه‌ی جداکننده تغییر خواهد کرد.
- هدف یافتن «رویه‌ای جداکننده با بیشترین حاشیه» است که خطای دسته‌بندی نادرست در آن مینیمم شود:

$$\Phi(\xi)=\sum_{i=1}^{N}I(\xi_i-1)\qquad I(\xi)=\begin{cases}0 & \text{if } \xi\leq 0\\ 1 & \text{if } \xi>0\end{cases}$$

Soft Margin Classification

- با توجه به این که کمینه کردن چنین تابعی یک مسأله‌ی بهینه‌سازی nonconvex است و در رده‌ی NP-complete قرار می‌گیرد، آن را با تابع زیر تقریب می‌زنیم:

$$\Phi(\xi) = \sum_{i=1}^{N} \xi_i$$

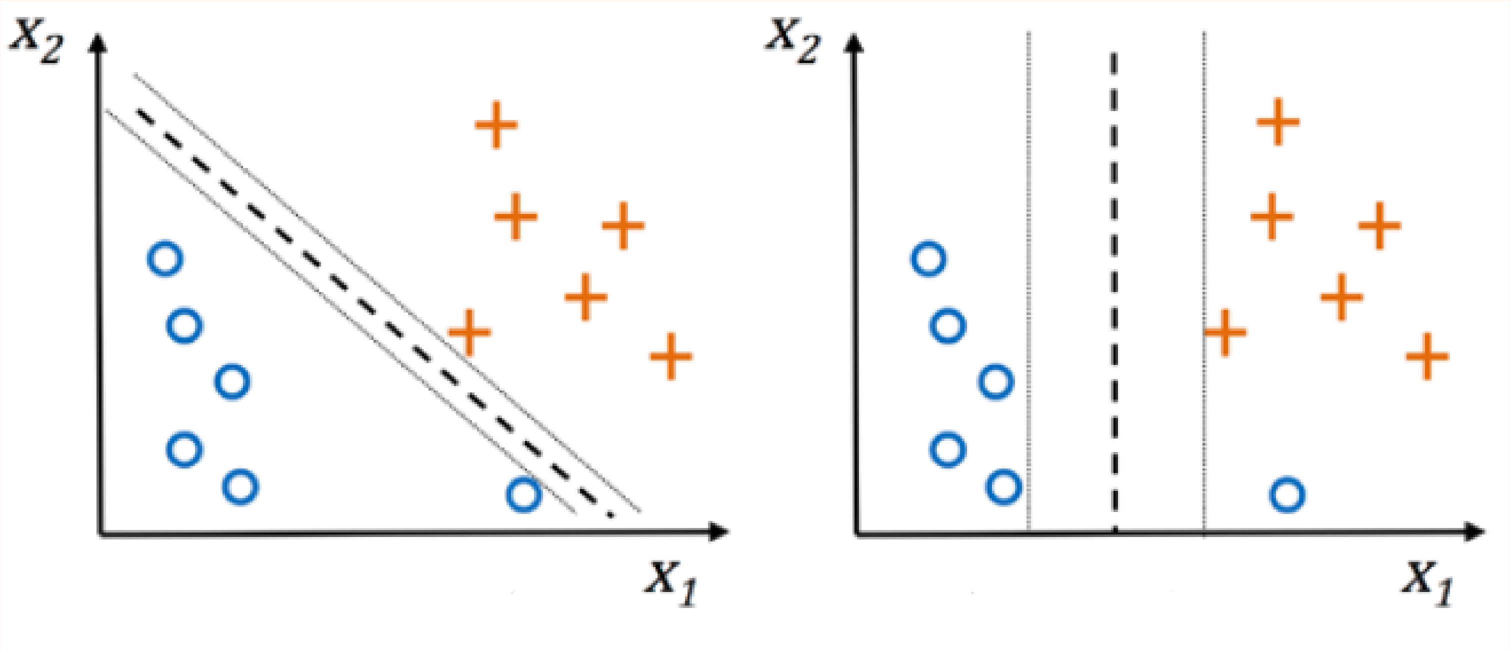
- و در کل هدف مینیمم کردن عبارت زیر است:

$$\Phi(W, \xi) = \frac{1}{2} W^T W + C \sum_{k=1}^{R} \varepsilon_k$$

regularization parameter

این پارامتر نوعی مصالحه بین پیچیدگی ماشین و خطا برقرار می‌کند. هرچه C به صفر نزدیک‌تر باشد به این معناست که خطا اهمیت کم‌تری دارد و در نتیجه حاشیه بزرگ‌تر می‌شود. و هرچه بزرگ‌تر باشد، مسأله به حالت **hard margin** نزدیک‌تر می‌شود.

مثال



Large value for C

Small value for C

Raschka, S., and V. Mirjalili. Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and Tensorflow 2, 3rd Edition. Packt Publishing, 2019.

Soft Margin

- برای Hard Margin داشتیم:

Find W and b such that
$\Phi(\boldsymbol{W}) = ½\, W^T W$ is minimized and for all $\{(X_i, d_i)\}$
$d_i\,(\boldsymbol{W}^T\boldsymbol{X}_i + b) \geq 1$

- با اضافه کردن Slack Variable داریم:

Find W and b such that
$\Phi(W) = ½\, W^T W + C\Sigma\xi_i$ is minimized and for all $\{(X_i, d_i)\}$
$d_i\,(W^T X_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

یافتن رویه‌ی بهینه

- رابطه‌ی لاگرانژ زیر تعریف می‌شود به گونه‌ای که همه‌ی نیازمندی‌ها را پوشش دهد:

$$J(W,b,\xi,\alpha,\mu)=\frac{1}{2}W^TW+C\sum_i \xi_i-\sum_{i=1}^{N}\alpha_i[d_i(W^TX_i+b)-1+\xi_i]-\sum_i \mu_i\xi_i$$

  - بخش آخر از این رو اضافه شده است که تا نامنفی بودن ξ را تضمین کند.

- حل این بهینه‌سازی معادل حل مساله‌ی زیر است:

$$\min_{W,b,\varepsilon}\max_{\alpha,\mu\geq 0} J(W,b,\xi,\alpha,\mu)$$

Soft Margin Classification

در نهایت ضرایب لاگرانژ از عبارت زیر محاسبه خواهند شد:

$$\max_{\alpha\geq 0} Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j X_i^T X_j$$

با در نظر گرفتن قیود زیر

$$\sum_{i=1}^{N}\alpha_i d_i = 0$$

$$0 \leq \alpha_i \leq C$$

بقیه مراحل مانند حالت قبل خواهد بود:

$$W_{op} = \sum_{i=1}^{N_s}\alpha_i d_i X_i$$

Quadratic programming

- QP فرآیند یافتن بردار x است به گونه‌ای که تابع درجه ۲ به شکل زیر کمینه شود:

$$\min_x \frac{1}{2}\mathbf{x}^T H\mathbf{x} + c^T\mathbf{x}$$

```
H(k1,k2)=d(k1)*d(k2)*X(:,k1)'*X(:,k2);
```

```
c=-ones(n,1);
```

- و شروط زیر برقرار باشد:

$$A\mathbf{x} \le b$$

$$l \le \mathbf{x} \le u$$

$$Aeq.\mathbf{x} = beq$$

```
Aeq=d;
beq=0;
lb=zeros(n,1);
ub=C*ones(n,1);
```

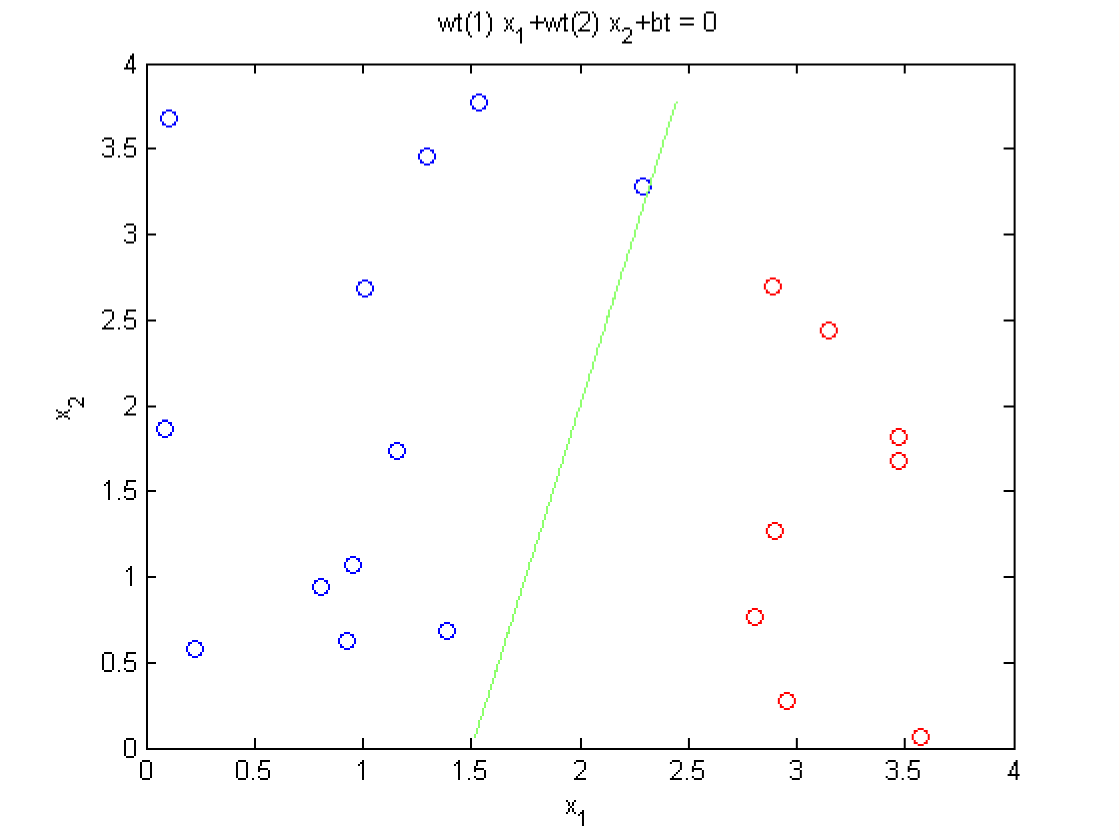
```
alpha=quadprog(H,f,[],[],Aeq,beq,lb,ub)';
```

مثال

```
n = 20;
rand('seed',2);
X = 4* rand (2 ,n) ;
bt = -6;
wt = [4 ; -1];

d = sign (wt (1) * X (1 ,:) + wt (2) * X (2 ,:) + bt) ;
x1min=min(X(1,:));
x1max=max(X(1,:));
x2min=min(X(2,:));
x2max=max(X(2,:));

figure;
axis([x1min x1max x2min x2max]);
plot (X (1 ,find (d ==1)) ,X (2 ,find (d ==1)) , ' or ') ;
hold on
plot (X( 1,find (d ==-1)) ,X(2, find (d ==-1)) , ' ob ') ;

Linet=@(x1,x2) wt(1)*x1+wt(2)*x2+bt;
ezplot(Linet,[x1min x1max x2min x2max]);
```


http://asi.insa-rouen.fr/enseignants/~scanu/Cours_SVM/TP1_Linear_SVM_Primal.pdf

مثال(ادامه...)

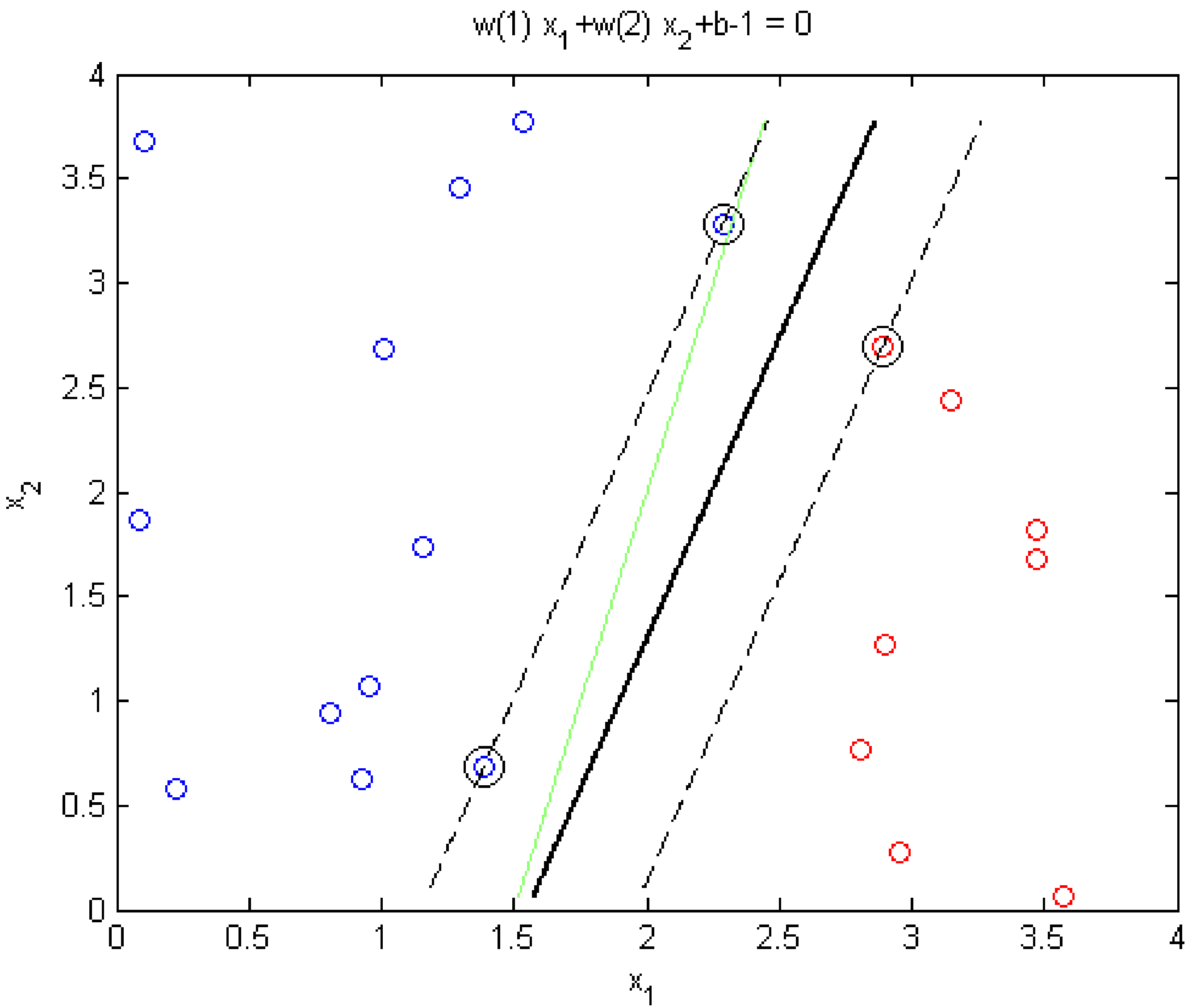


مثال(ادامه...)

```
C=10;
H=zeros(n,n);
for k1=1:n
    for k2=1:n
        H(k1,k2)=d(k1)*d(k2)*X(:,k1)'*X(:,k2);
    end
end
f=-ones(n,1);
Aeq=d;
beq=0;
 lb=zeros(n,1);
 ub=C*ones(n,1);
 alpha=quadprog(H,f,[],[],Aeq,beq,lb,ub)';
 Svs=find(alpha> 1e-5);
 w=0;
for k1=Svs
    w=w+alpha(k1)*d(k1)*X(:,k1);
end
b=mean(d(Svs)-w'*X(:,Svs));
```

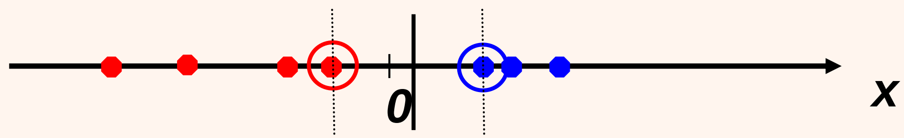
مثال(ادامه...)

```
plot(X(1,Svs),X(2,Svs),'ko','MarkerSize',12);
Line=@(x1,x2) w(1)*x1+w(2)*x2+b;
LineA=@(x1,x2) w(1)*x1+w(2)*x2+b+1;
LineB=@(x1,x2) w(1)*x1+w(2)*x2+b-1;


handle=ezplot(Line,[x1min x1max x2min x2max]);
set(handle,'Color','k','LineWidth',2);

handleA=ezplot(LineA,[x1min x1max x2min x2max]);
set(handleA,'Color','k','LineWidth',1,'LineStyle','--');

handleB=ezplot(LineB,[x1min x1max x2min x2max]);
set(handleB,'Color','k','LineWidth',1,'LineStyle','--');
```

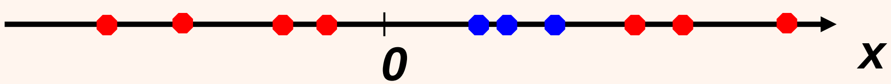

w(1) x1 +w(2) x2+b-1 = 0
x2
x1
0
0.5
1
1.5
2
2.5
3
3.5
4

SVM غیرخطی

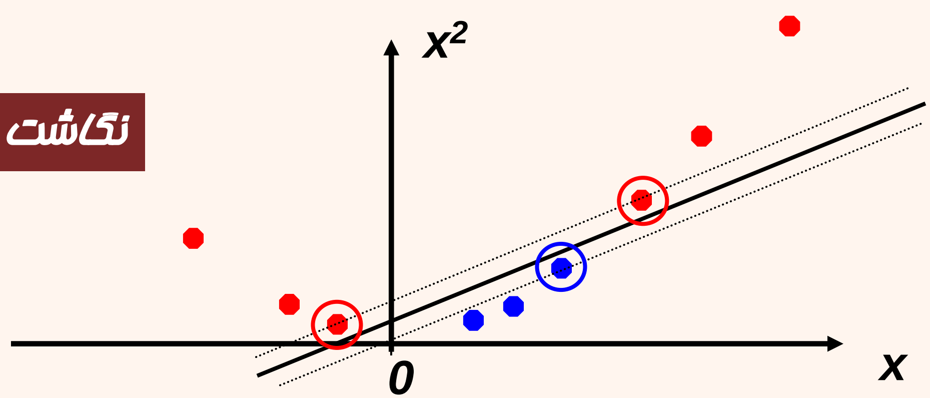
- برای داده‌هایی که قابلیت جداسازی خطی دارند، عملکرد سیستم بسیار خوب است.



- اگر داده‌ها به صورت‌های زیر باشد، مسأله چگونه حل می‌شود؟



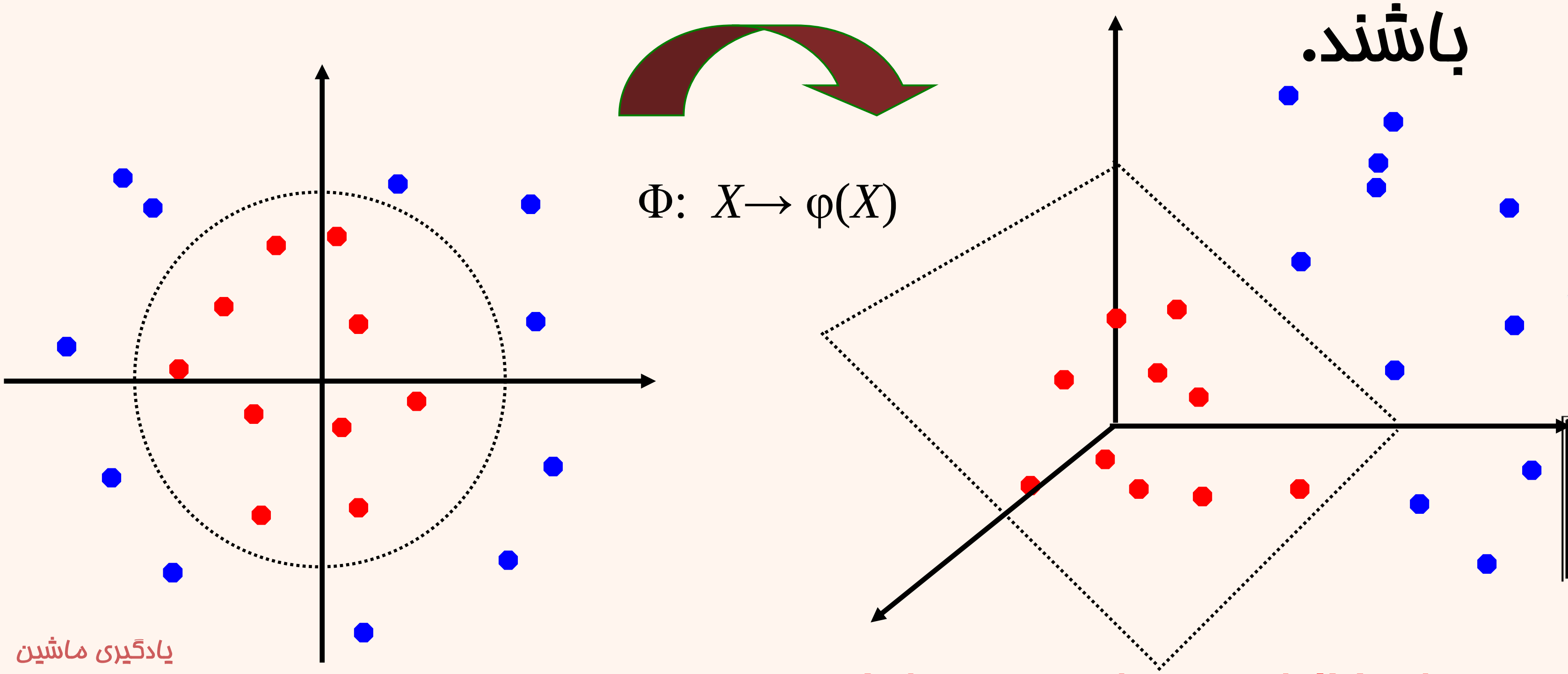
نگاشت به یک فضای High Dimension



www.cs.utexas.edu/~mooney/cs391L/slides/svm.ppt

نگاشت به فضای بالاتر

- همواره فضای ورودی می‌تواند به فضایی با ابعاد بالاتر نگاشت گردد.
- این نگاشت می‌تواند به صورتی باشد که در این فضای جدید ورودی‌ها قابلیت جداسازی داشته باشند.



www.cs.utexas.edu/~mooney/cs391L/slides/svm.ppt

نگاشت به فضای بالاتر

$$\underset{R^{m_0}}{X} \longrightarrow \underset{R^{m_1}}{\Phi(X)} \xrightarrow{W} \text{خروجی}$$

- داشتیم

$$W^T X + b = 0$$

- هنگامی‌که ورودی‌ها به فضای دیگری نگاشت شوند، برای نگاشت جدید خواهیم داشت:

$$\Phi(X) = [\varphi_1(X), \varphi_2(X), \ldots\ldots\ldots, \varphi_{m_1}(X)]^T$$

- در این حالت هدف یافتن رویه‌ی جداسازی است به‌گونه‌ای که:

$$\sum_{j=1}^{m_1} w_j \varphi_j(X) + b = 0$$

نگاشت به فضای بالاتر

$$\sum_{j=1}^{m1} w_j \varphi_j(X) + b = 0$$

- با فرض $\varphi_0(\mathbf{X}) = 1$
- خواهیم داشت:

$$\sum_{j=0}^{m1} w_j \varphi_j(X) = 0$$

$$\Rightarrow \quad W^T \Phi(X) = 0$$

$$\Phi(X) = [1, \Phi(X)]^T$$

$$W = [b = w_0, w_1, w_2, \ldots\ldots w_{m1}]^T$$

نگاشت به فضای بالاتر

- در این مرحله تمامی شروط و قیودی که برای جداسازی خطی در نظر گرفتیم وجود دارد تنها به جای X_i ها $\Phi(X_i)$ در نظر گرفته می‌شود:

$$d_i\left(\sum_{j=0}^{m_1} w_j \varphi_j(X_i) - 1\right) \geq 0$$

$$W_{opt} = \sum_{i=1}^{N} \alpha_i . d_i (\Phi(X_i))$$

اسکالر

$m_1 \times 1$

$$\mathbf{W}_{opt}^T \Phi(\mathbf{X}) = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i . d_i \Phi^T(\mathbf{X}_i)\Phi(\mathbf{X}) = 0$$

نگاشت به فضای بالاتر

$$\sum_{i=1}^{N} \alpha_i . d_i \Phi^T(X_i)\Phi(X) = 0$$

$$K(X_i, X_j) = \varphi(X_i)^T \varphi(X_j)$$

$$\sum_{i=1}^{N} \alpha_i . d_i K(X_i, X) = 0$$

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j K(X_i, X_j)$$

تابع kernel، تابعی است که معادل ضرب داخلی دو بردار خصیصه است.

مثال

$\mathbf{x}=[x_1 \ \ x_2]^T;$

$K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2,$

$K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2 = 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2} + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$

$= [1 \ \ x_{i1}^2 \ \ \sqrt{2}\, x_{i1}x_{i2} \ \ x_{i2}^2 \ \ \sqrt{2}x_{i1} \ \ \sqrt{2}x_{i2}]^T [1 \ \ x_{j1}^2 \ \ \sqrt{2}\, x_{j1}x_{j2} \ \ x_{j2}^2 \ \ \sqrt{2}x_{j1} \ \ \sqrt{2}x_{j2}]$

$= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j),$

$\Rightarrow K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j):$

where $\varphi(X) = [1 \ \ x_1^2 \ \ \sqrt{2}\, x_1x_2 \ \ x_2^2 \ \ \sqrt{2}x_1 \ \ \sqrt{2}x_2]$

Mercer's theorem:

Every semi-positive definite symmetric function is a kernel

$K(X_1, X_1)$	$K(X_1, X_2)$	$K(X_1, X_3)$	…	$K(X_1, X_n)$
$K(X_2, X_1)$	$K(X_2, X_2)$	$K(X_2, X_3)$		$K(X_2, X_n)$
…	…	…	…	…
$K(X_n, X_1)$	$K(X_n, X_2)$	$K(X_n, X_3)$	…	$K(X_n, X_n)$

نگاشت به فضای بالاتر

$$Q(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j \underbrace{\Phi(X_i)\Phi(X_j)}_{K(X_i,X_j)}$$

$$K_{N\times N}=\left\{K(X_i,X_j)\right\}_{i,j=1}^{N}$$

ماتریس متقارن

هدف یافتن ضرایب لاگرانژ بیشینه در عبارت زیر است:

$$Q(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j K(X_i,X_j)$$

با در نظر گرفتن قیود زیر

$$\sum_{i=1}^{N}\alpha_i d_i=0$$

$$0\le\alpha_i\le C$$

kernel trick

در صورت یافتن تابع **kernel** مناسب بدون این که درگیر مشکلات فضای با ابعاد بالا (نکبت ابعاد) شویم، تنها از نتیجه این نگاشت بهره می‌بریم.

$$\boxed{g(X)=\sum\alpha_i d_i K(X_i,X)}$$

کرنل‌های معمول

$k(\mathbf{x}, \mathbf{x}_i)$

$i = 1, 2, \ldots, N$

polynomial

$$\left(\mathbf{x}^T \mathbf{x}_i + 1\right)^p$$

RBF

$$\exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{x} - \mathbf{x}_i\right\|^2\right)$$

tanh

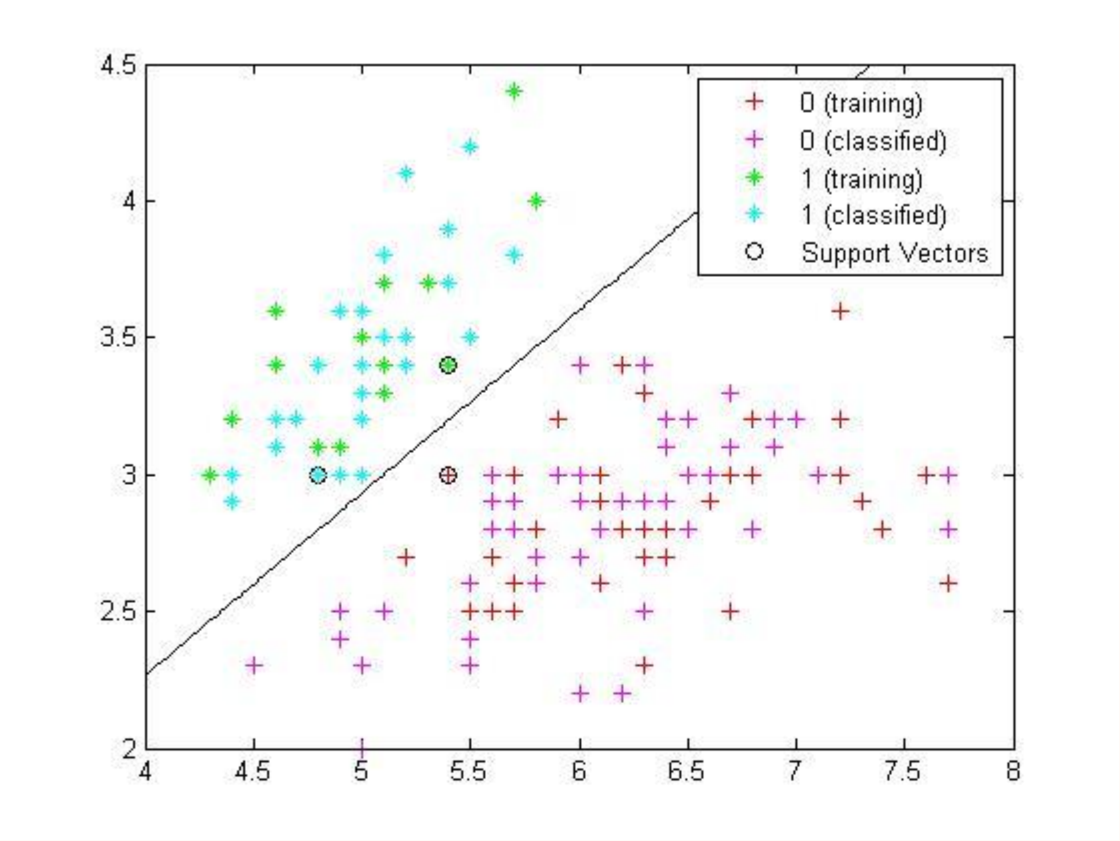
$$\tanh\left(\beta_0 \mathbf{x}^T \mathbf{x}_i + \beta_1\right)$$

chi-squared kernel

$$1 - \sum_{i=1}^{n} \frac{\left(\mathbf{x} - \mathbf{x}_i\right)}{\frac{1}{2}\left(\mathbf{x} + \mathbf{x}_i\right)^2}$$

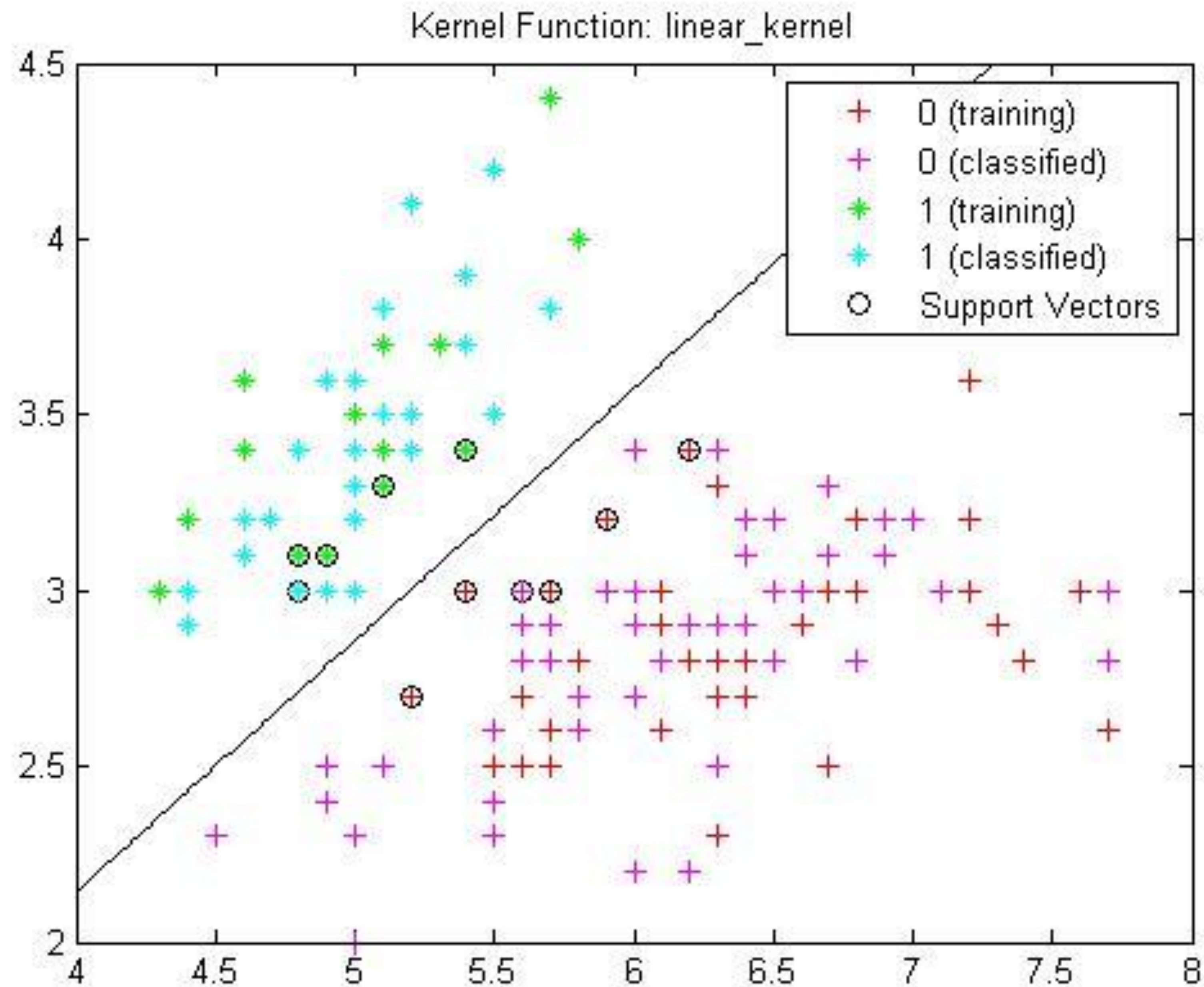
مثال

```
clear all;
close all;
load fisheriris
data = [meas(:,1), meas(:,2)];
groups = ismember(species,'setosa');
[train, test] = crossvalind('holdOut',groups);
cp = classperf(groups);
svmStruct =
svmtrain(data(train,:),groups(train),'showplot',true,'boxconstr
aint',1e6);
title(sprintf('Kernel Function: %s',...
    func2str(svmStruct.KernelFunction)),...
    'interpreter','none');
classes = svmclassify(svmStruct,data(test,:),'showplot',true);
classperf(cp,classes,test);
cp.CorrectRate
```


0 (training)
0 (classified)
1 (training)
1 (classified)
Support Vectors
4.5
4
3.5
3
2.5
2
4
4.5
5
5.5
6
6.5
7
7.5
8

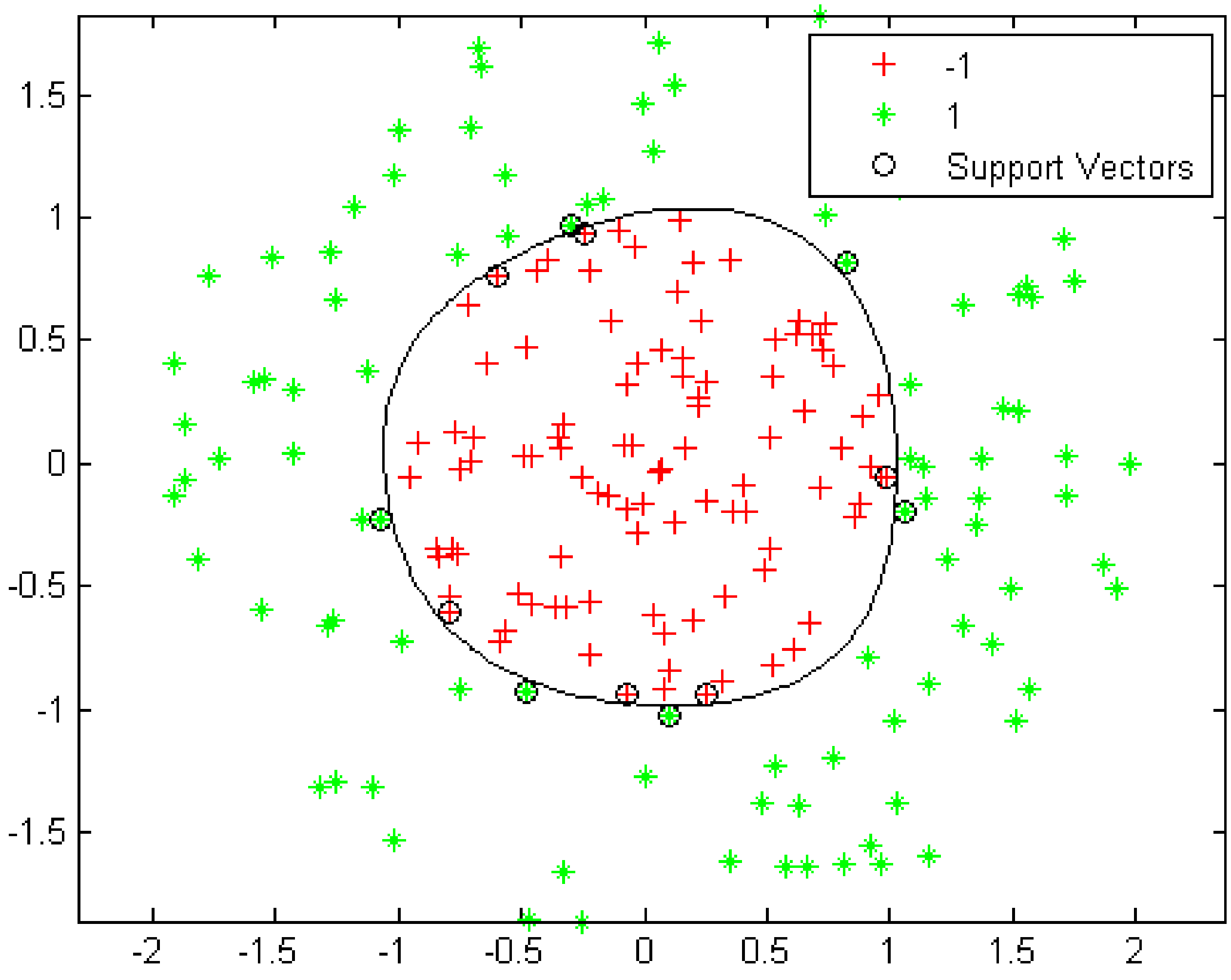
مثال

```
clear all;
close all;
load fisheriris
data = [meas(:,1), meas(:,2)];
groups = ismember(species,'setosa');
[train, test] = crossvalind('holdOut',groups);
cp = classperf(groups);
svmStruct = svmtrain(data(train,:),groups(train),'showplot',true);
title(sprintf('Kernel Function: %s',...
    func2str(svmStruct.KernelFunction)),...
    'interpreter','none');
classes = svmclassify(svmStruct,data(test,:),'showplot',true);
classperf(cp,classes,test);
cp.CorrectRate
```


Kernel Function: linear_kernel
0 (training)
0 (classified)
1 (training)
1 (classified)
Support Vectors
4.5
4
3.5
3
2.5
2
4
4.5
5
5.5
6
6.5
7
7.5
8

مثال

```
r = sqrt(rand(100,1)); % radius
t = 2*pi*rand(100,1); % angle
data1 = [r.*cos(t), r.*sin(t)]; % points
r2 = sqrt(3*rand(100,1)+1); % radius
t2 = 2*pi*rand(100,1); % angle
data2 = [r2.*cos(t2), r2.*sin(t2)]; % points
plot(data1(:,1),data1(:,2),'r.')
plot(data2(:,1),data2(:,2),'b.')
axis equal
data3 = [data1;data2];
theclass = ones(200,1);
theclass(1:100) = -1;
cl = svmtrain(data3,theclass,'Kernel_Function','rbf',...
    'boxconstraint',Inf,'showplot',true);
hold on
axis equal
```


-1
1
Support Vectors
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

ساخت کرنل‌های جدید

- براساس کرنل‌های موجود به سادگی می‌توان کرنل‌های جدید ساخت:
  - هر تابع K(.,.) یک کرنل است چنانچه معادل ضرب داخلی بردارهای حاصل از نگاشت باشد.

جمع دو کرنل

$$K(x,y)=c_1\boxed{K_1(x,y)}+c_2\boxed{K_2(x,y)} \qquad c_1,c_2\geq 0$$

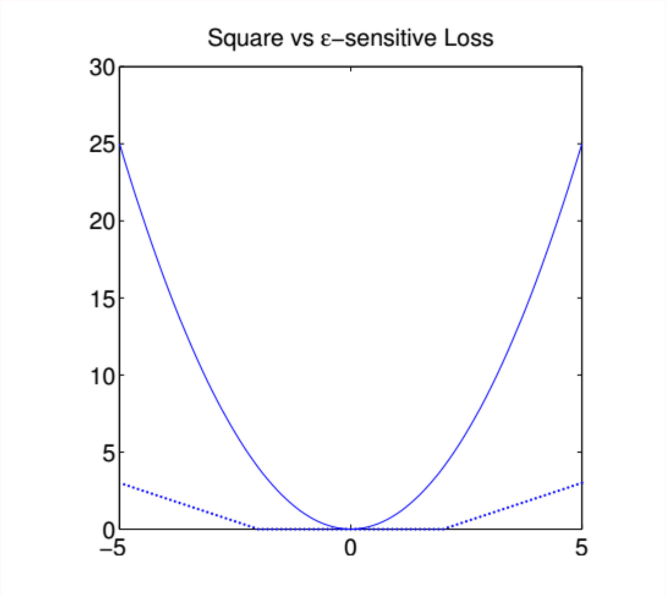
$$\phi(x)=\left(\sqrt{c_1}\phi_1(x),\sqrt{c_2}\phi_2(x)\right)$$

$$\phi(x).\phi(y)=c_1\boxed{\phi_1(x).\phi_1(y)}+c_2\boxed{\phi_2(x).\phi_2(y)}$$

رگرسیون

ε-sensitive(hing) Loss

$$E(\theta \mid \mathcal{X}) = \frac{1}{2}\sum_{t=1}^{N} 1\left(\left|r^t - g\left(x^t \mid \theta\right)\right| > \varepsilon\right)\left(\left|r^t - g\left(x^t \mid \theta\right)\right| - \varepsilon\right)$$

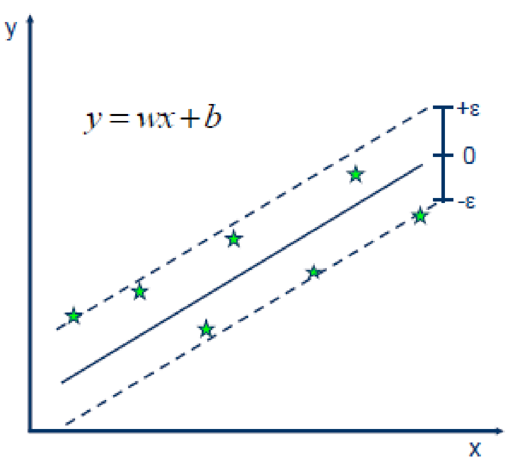


***Yıldız, Olcay Taner, and Ethem Alpaydın. "Statistical Tests Using Hinge/E-Sensitive Loss." In Computer and Information Sciences Iii*, 153-60: Springer, 2013.**

Support Vector regression(SVR)

$$(W^T X_i + b) - r_i \leq \varepsilon$$

$$\frac{1}{2} W^T W \rightarrow \text{minimize}$$



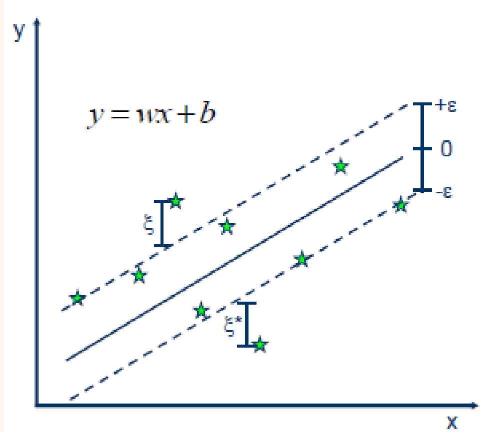
- Solution:

$$\min \frac{1}{2} \|w\|^2$$

- Constraints:

$$y_i - wx_i - b \leq \varepsilon$$
$$wx_i + b - y_i \leq \varepsilon$$

http://www.saedsayad.com/support_vector_machine_reg.htm

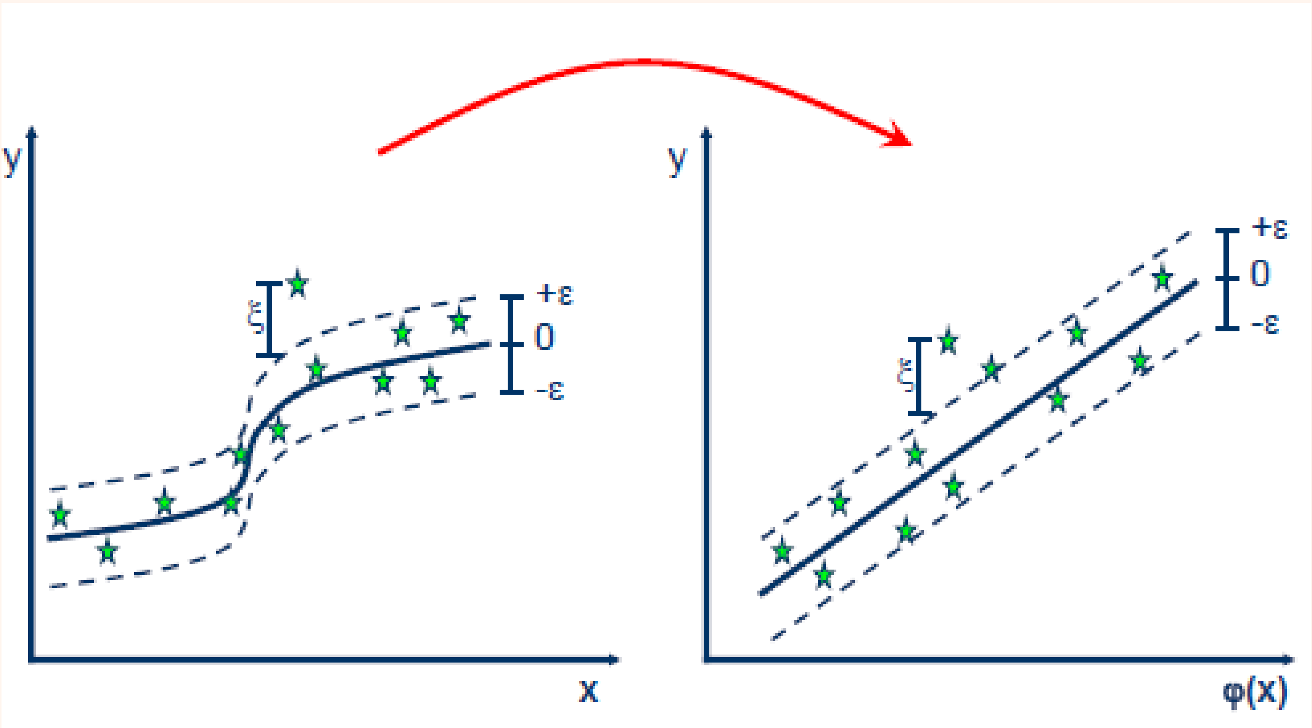


- Minimize:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*)$$

- Constraints:

$$y_i - wx_i - b \leq \varepsilon + \xi_i$$
$$wx_i + b - y_i \leq \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0$$


y
x
+ε
0
-ε
ξ
y
φ(x)
+ε
0
-ε
ξ