



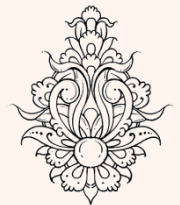
ماشین‌های بردار پشتیبان



دانشگاه شهید بهشتی
پژوهشکده‌ی فضای مجازی
پاییز ۱۳۹۹
احمد محمودی ازناوه

فهرست مطالب

- ماشین بردار پشتیبان (SVM)
 - تاریخچه
 - معرفی
 - داده‌های جدایی‌پذیر خطی (Hard Margin)
- Soft Margin
- مجموعه‌های جدایی‌ناپذیر خطی
 - نگاهت به فضای با ابعاد بالا
 - Inner product kernel
- مثال XOR
- Matlab در SVM

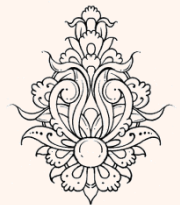


تاریخچه

- 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.

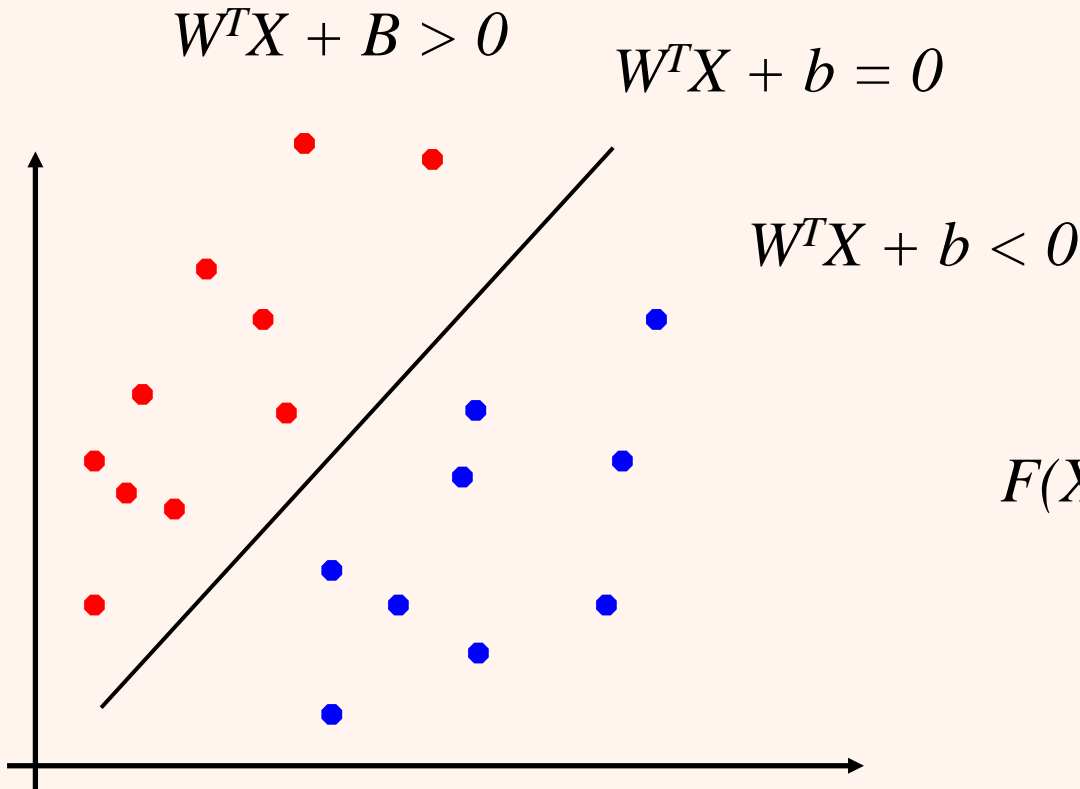


Cortes, C. and V. Vapnik (1995). "Support-vector networks." Machine Learning 20(3): 273-297.

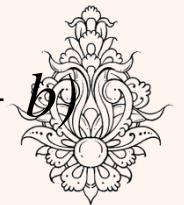


معرفی

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



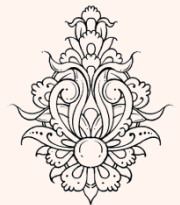
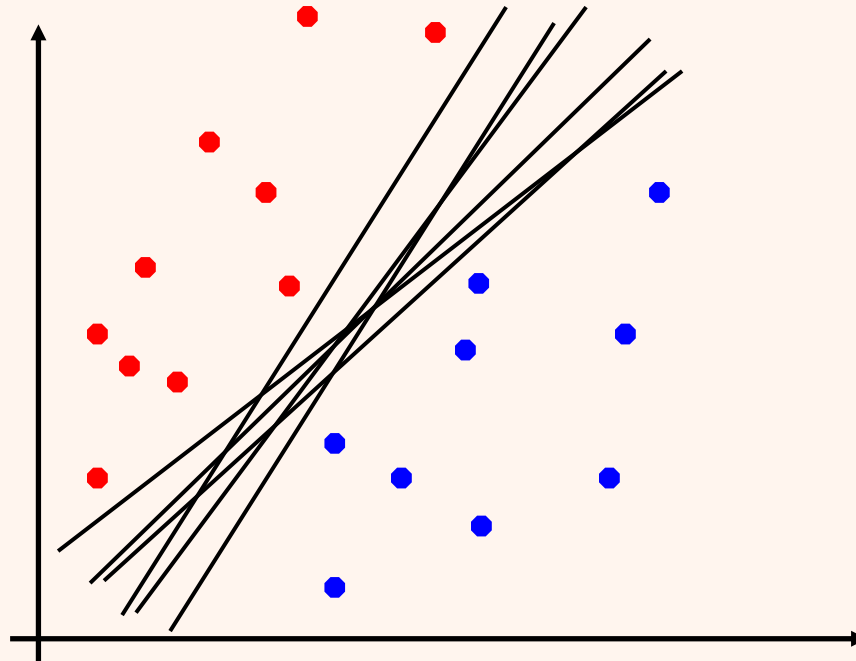
$$F(X) = \text{SIGN}(W^T X + b)$$



مرز بهینه

• سوال

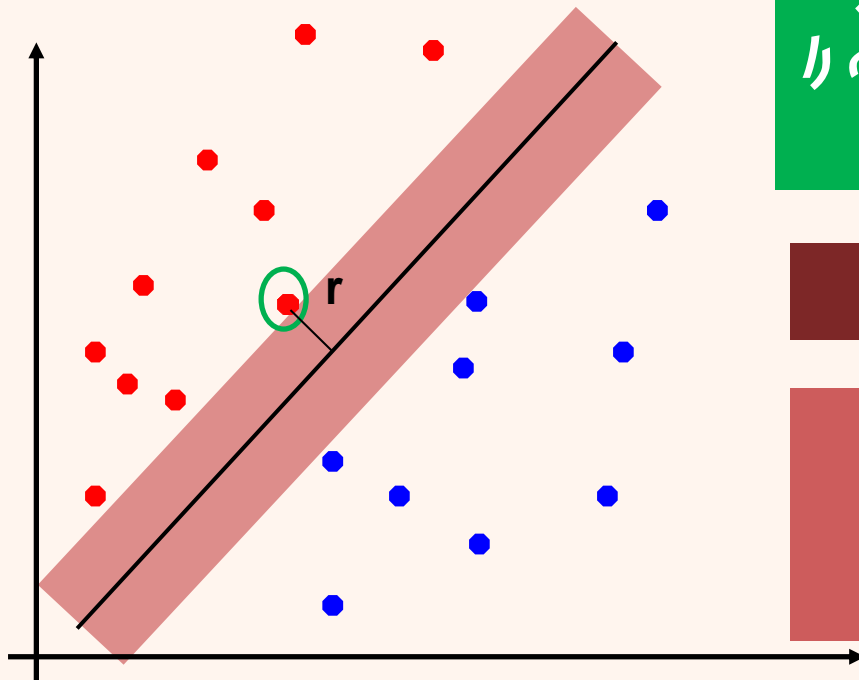
– کدام یک از مرزها، مرزی بهینه برای جداسازی است؟



مرز جداسازی

- می‌خواهیم به گونه‌ای بهترین مرز جداسازی را به دست آوریم.

Margin of separation



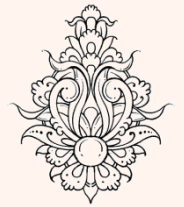
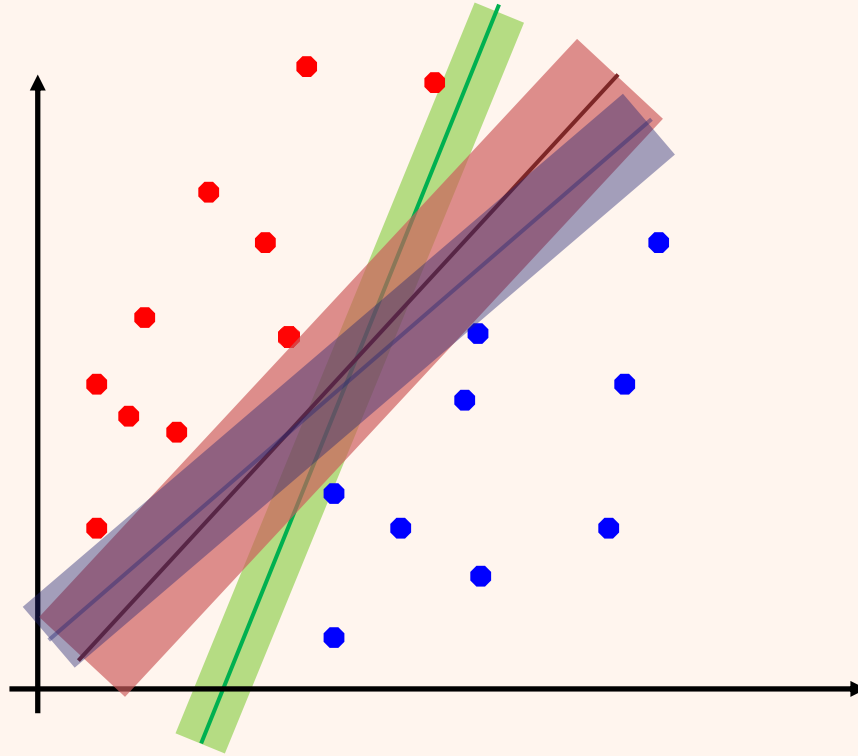
فرض کنیم نزدیک‌ترین نقطه به مرز جداسازی در نظر گرفته شده و فاصله را r بنامیم.

هدف ماکزیم نمودن r است.

یک ماشیه مشخص می‌کنیم هر مرزی که ماشیهی پهن‌تری را نتیجه دهد، بهتر است.

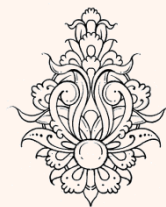
دانشگاه
تهران
پهشتی

مردز بهینه

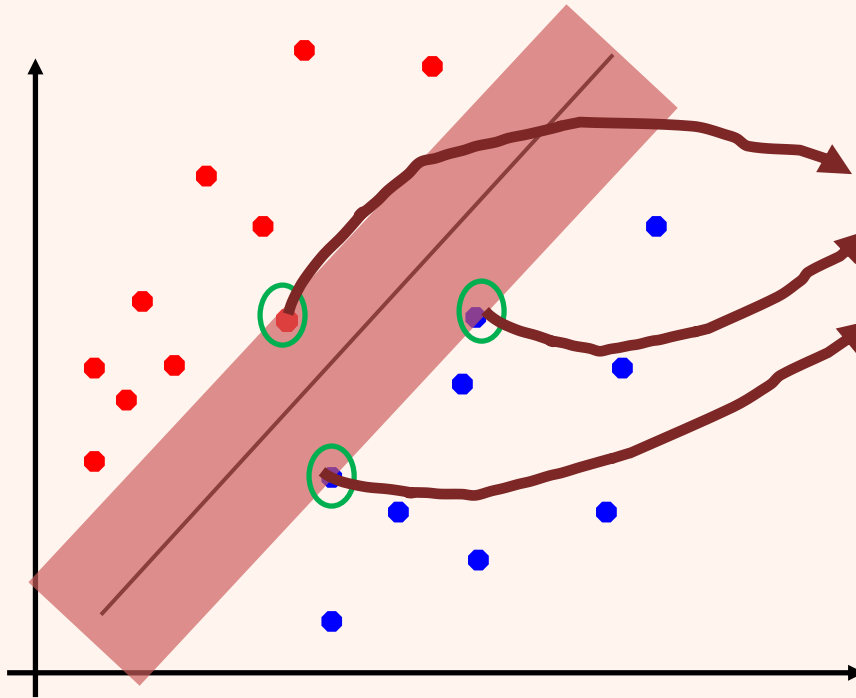


ماشیهی ماکزیمه

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

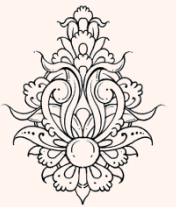


- به نمونه‌های روی مرز ماشیه «بردار پشتیبان» می‌گویند.



بردارهای پشتیبان

Optimal hyperplane



مرز جداسازی

- برای معادله‌ی مرز جداسازی داشته‌یم:

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

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

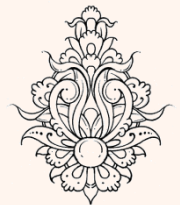
$$(X_i, d_i = -1) \quad W^T X_i + b < 0$$

- فرض کنیم مرز بهینه توسط W_{op} و b_{op} مشخص شود.

- اگر نزدیک‌ترین نقطه به مرز جداسازی را در نظر گرفته، فاصله را « r » بنامیم.

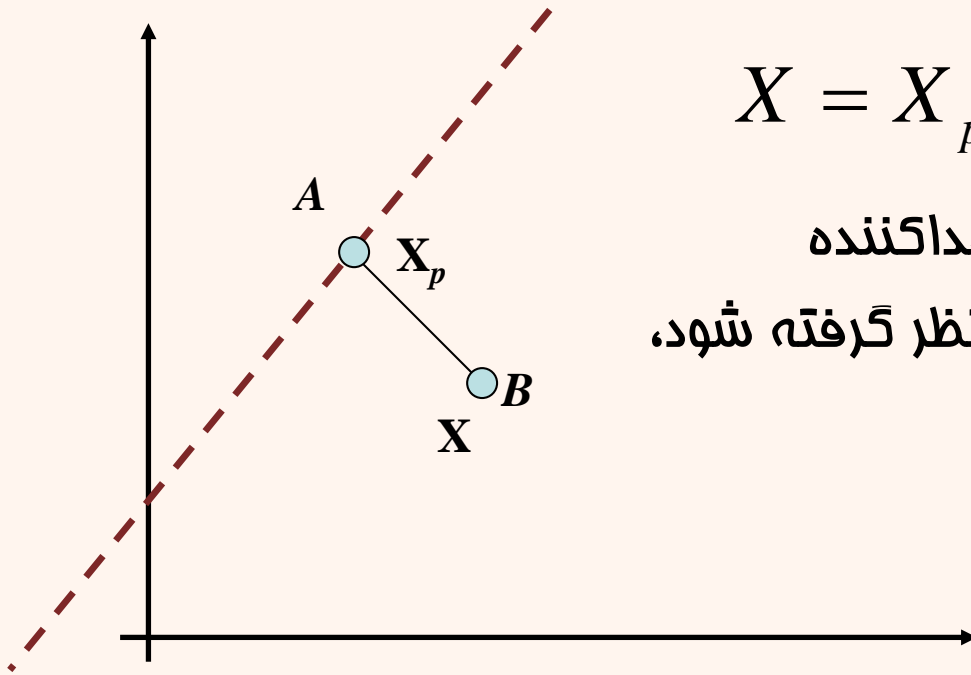
• هدف

- ماکزیمم نمودن فاصله یا همان $\rho = 2r$ است.



مرز جداسازی (ادامه...)

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

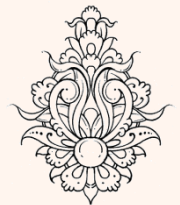


$$X = X_p + \overrightarrow{AB}$$

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

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

$$\overrightarrow{AB} = r \frac{W_{op}}{\|W_{op}\|}$$



مرز جداسازی (ادامه...)

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

• داشتهیم:

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

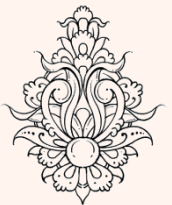
$$g(X) = W_{op}^T \left[X_p + r \frac{W_{op}}{\|W_{op}\|} \right] + b_{op}$$

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

روی مرز پس برابر با صفر

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

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



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

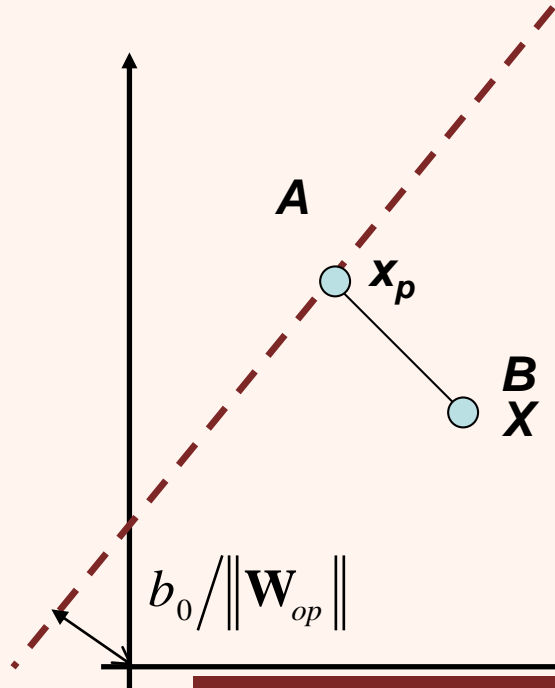
مرز جداسازی (ادامه...)

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

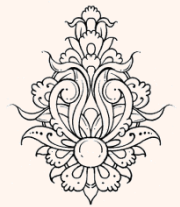
$$r = \frac{g(X)}{\|W_{op}\|}$$

هدف ماکزیمم نمودن r است.

در این حالت تمت شرایطی می‌باید W کمینه گردد.



$$r = \frac{g(X)}{\|W_{op}\|} = \frac{b_{op}}{\|W_{op}\|} \quad X=0$$



مثبت یا منفی بودن b_{op} نشان دهندهی این است که مبدأ در کدام سمت قط مرزی است.

فاصله از مبدأ مفاصمات

مرز جداسازی (ادامه...)

مکان یافتن W_{op} و b_{op} است

• جداساز خطی را به صورت زیر در نظر می‌گیریم:

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

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

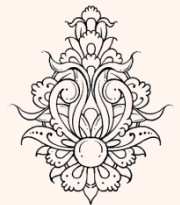
به صورت کلی داریم:

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

• رابطه‌ی بالا برای تمامی الگوهای آموزشی برقرار است.

• و در نتیجه برای بردارهای پشتیبان

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



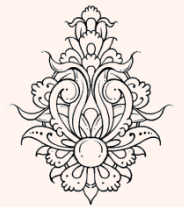
مرز جداسازی (ادامه...)

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

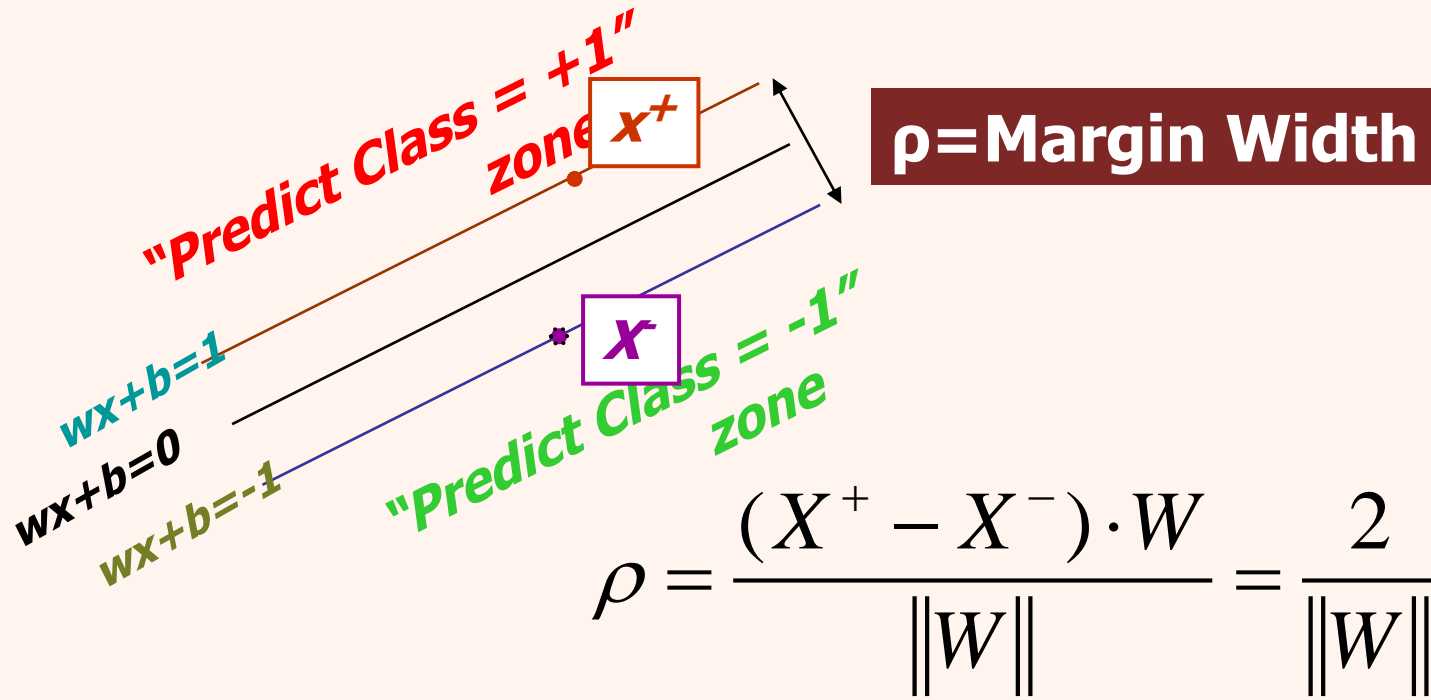
$$r = \frac{g(X^s)}{\|W_{op}\|} = \begin{cases} \frac{1}{\|W_{op}\|} \\ -\frac{1}{\|W_{op}\|} \end{cases}$$

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

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

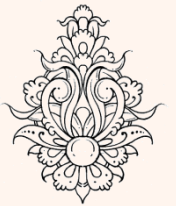


جدایی پذیر خطی



می‌دانیم:

- $W \cdot X^+ + b = +1$
- $W \cdot X^- + b = -1$
- $W \cdot (X^+ - X^-) = 2$



جدایی پذیر خطی

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

- با توجه به دو رابطه‌ی
- $d_i(W_{op}^T X + b_{op}) \geq 1$ به این نتیجه می‌رسیم که W_{op} می‌باید مینیمم گردد.

- این مسأله معادل مینیمم کردن

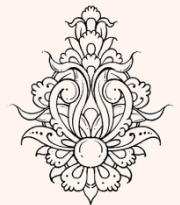
$$\Phi(W) = \frac{1}{2} W^T W$$

– Φ یک تابع محدب (Convex Function) است.

– طبق رابطه‌ی $d_i(W_{op}^T X_i + b_{op}) \geq 1$ برای N الگوی آموزشی شروط زیر می‌باید برقرار باشد:

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

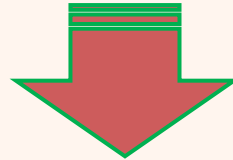
این میزان بزرگ‌تر یا مساوی صفر است



وزن‌ها و بایاس را به گونه‌ای بیابید که:

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

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



وزن‌ها و بایاس را به گونه‌ای بیابید که:

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

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



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.

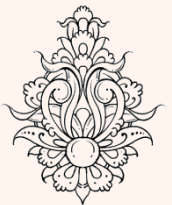
– قید عبارتی است که همواره بزرگتر یا مساوی صفر است.

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

به ازای هر نمونه یک قید داریم

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

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

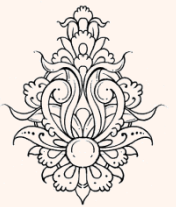


Duality theorem

- با توجه به تابع اولیه و قید می‌توان نشان داد که حل این مساله (primal) معادل حل مساله‌ی بهینه‌سازی زیر (dual) است:

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

بدین ترتیب ابتدا نسبت به W_{op} و b_{op} مشتق می‌گیریم.



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

Lagrange multiplier (nonnegative)

$$\frac{\partial J}{\partial W} = 0 \Rightarrow W - \sum_{i=1}^N \alpha_i d_i X_i = 0 \Rightarrow W_{op} = \sum_{i=1}^N \alpha_i d_i X_i$$

$$\frac{\partial J}{\partial b} = 0 \Rightarrow -\sum_{i=1}^N \alpha_i d_i = 0$$

b_{op} به دست نمی آید

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

ولی یک قید می دهد

به ازای یک مقدار ثابت α ، مقدار W و یک قید به دست می آید. با جایگزینی و استفاده از این روابط نسبت به α با توجه به قیدها نقطه‌ی بهینه را به دست می آوریم.



یافتن بهینه

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

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

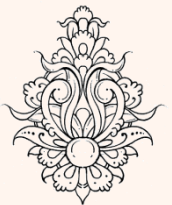
• برای مقادیر بهینه داشتیم:

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

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

• پس خواهیم داشت:

$$J(W_{op}, b_{op}, \alpha) = \frac{1}{2} W_{op}^T W_{op} - W_{op}^T W_{op} + 0 + \sum_{i=1}^N \alpha_i$$



یافتن بهینه

$$\begin{aligned} Q(\alpha) &= \sum_{i=1}^N \alpha_i - \frac{1}{2} W_{op}^T W_{op} \\ &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \left[\sum_{i=1}^N \alpha_i d_i X_i \right]^T \left[\sum_{j=1}^N \alpha_j d_j X_j \right] \\ &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i d_i \alpha_j d_j X_i^T X_j \end{aligned}$$

$$\left\{ \begin{aligned} &= \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i d_i \alpha_j d_j X_i^T X_j \\ &\sum_{i=1}^N \alpha_i d_i = 0 \\ &\alpha_i \geq 0 \text{ for } i = 0, 1, \dots, N \end{aligned} \right.$$



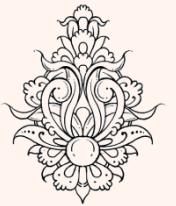
Karush–Kuhn–Tucker(KKT) condition of optimization theory

• 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.




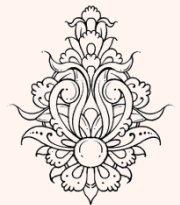
یافتن رویه بهینه

• پس از به دست آوردن α خواهیم داشت:

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

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

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



یافتن رویه‌ی بهینه

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

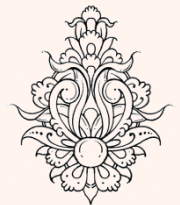
هر α_i مخالف صفر، نشان‌دهنده‌ی این است که X_i متناظرش یک بردار پشتیبان است.
در این حالت تابع جداکننده همانند زیر است:

$$g(X) = \sum \alpha_i d_i \underbrace{X_i^T X}_b + b$$

ضرب داخلی دو بردار

توجه:

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



Hard Margin

خلاصه

برای یک مجموعه جدایی پذیر قطعی

$$\min \frac{1}{2} W^T W$$

$$\text{s.t. } d_i (W^T X_i + b) - 1 \geq 0$$

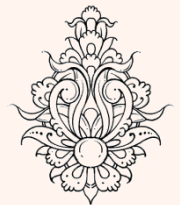
که معادل عبارت زیر است:

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

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

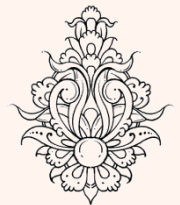
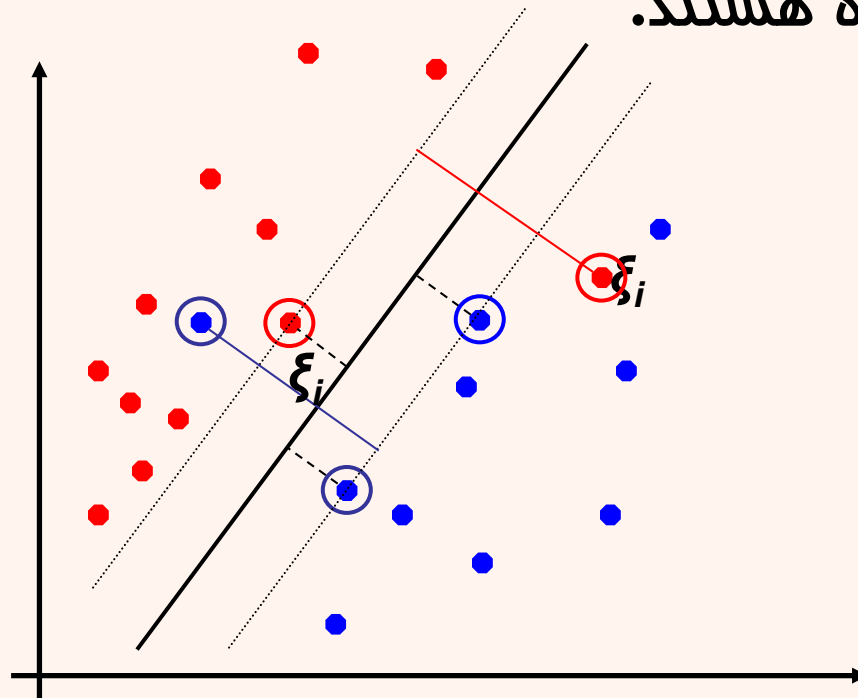
که با توجه به قضیه دوگانی معادل حل مسئله زیر خواهد بود

$$\max_{\alpha \geq 0} \left[\min_{W, b} J(W, b, \alpha) \right]$$



Soft Margin

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

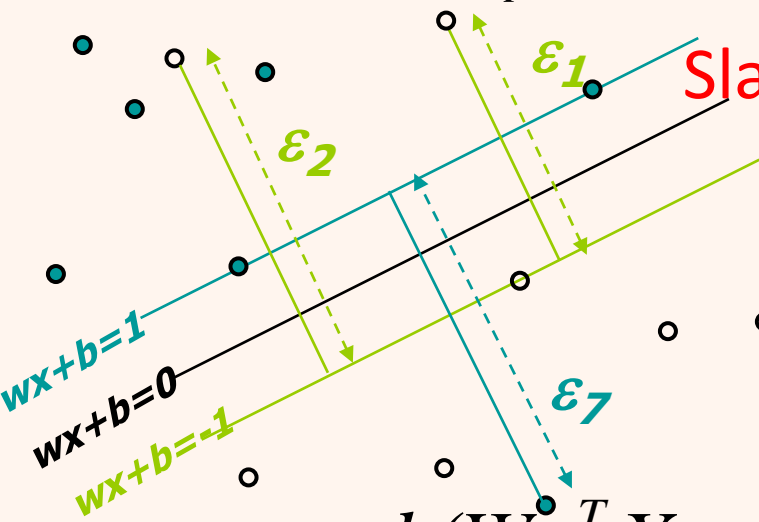


Soft Margin

- مسأله‌ی **Hard Margin** را تبدیل به حل مسأله‌ی **Soft Margin** می‌شود.

- ماشیهی جداسازی soft گفته می‌شود، در صورتی که برای برخی داده‌ها شرط زیر نقض شود:

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



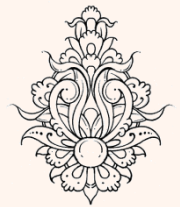
- با اضافه کردن یک **Slack Variable**

- مسأله را بار دیگر بررسی می‌کنیم.

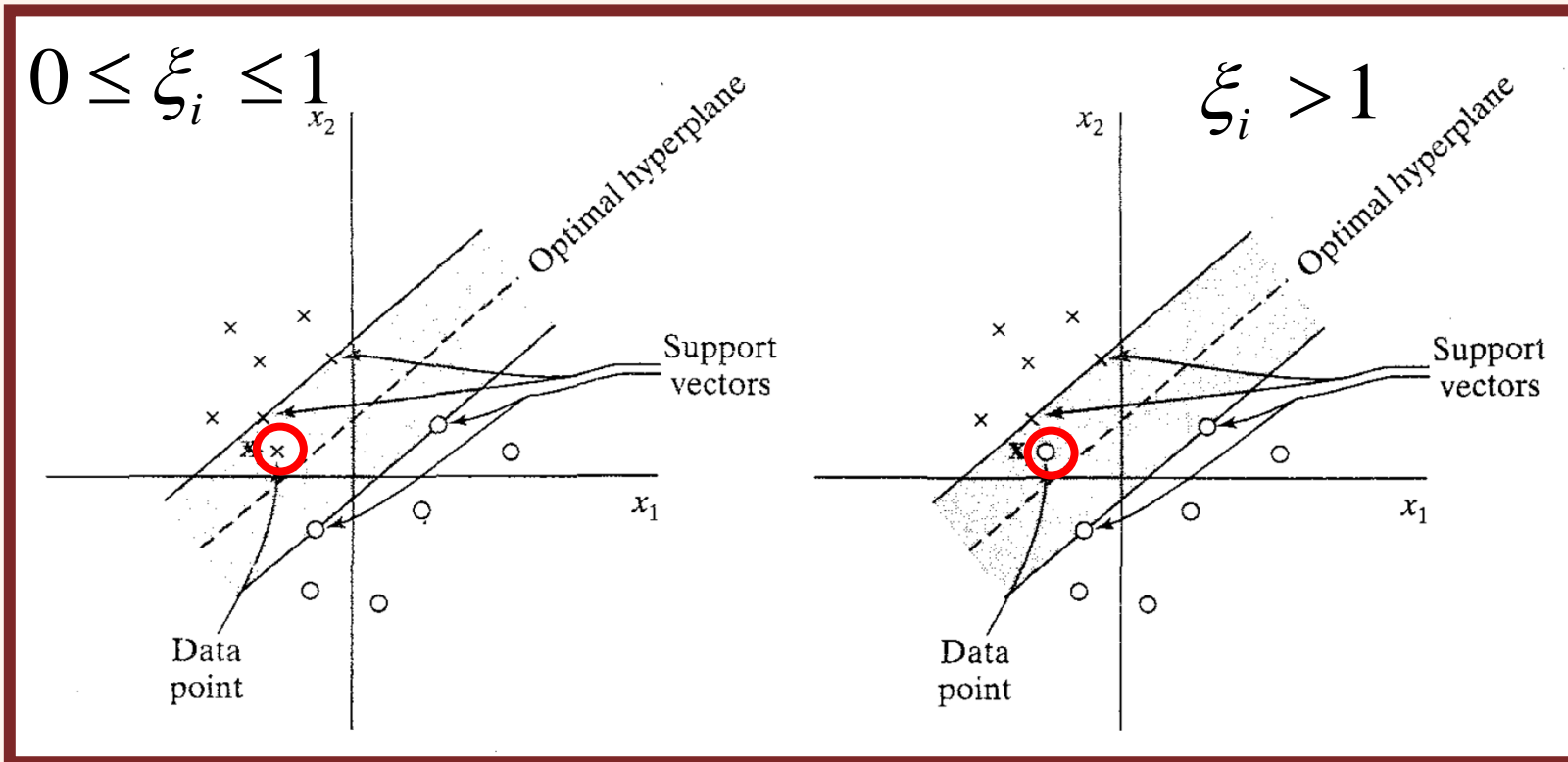
- این متغیر میزان انحراف از شرط

- فوق را نشان می‌دهد.

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



Soft Margin Classification



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

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



Soft Margin Classification

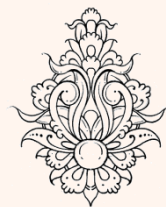
$$d_i(\mathbf{W}_{op}^T \mathbf{X} + b_{op}) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N$$

- در این حالت بردارهای پشتیبان آن‌هایی هستند که در رابطه‌ی تساوی در عبارت بالا صدق می‌کنند، حتی با وجود $\xi > 0$

– در صورتی که داده‌های نویزی از مجموعه خارج شود، رویه‌ی جداکننده تخییر خواهد کرد.

- هدف یافتن «رویه‌ای جداکننده با بیشترین ماشیه» است که فضای دسته‌بندی نادرست در آن مینیمم شود:

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



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 می‌نیمیم کردن عبارت زیر است:

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

regularization parameter

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



Soft Margin

- برای Hard Margin داشتیم:

Find W and b such that
 $\Phi(W) = \frac{1}{2} W^T W$ is minimized and for all $\{(X_i, d_i)\}$
 $d_i (W^T X_i + b) \geq 1$

- با اضافه کردن Slack Variable داریم:

Find W and b such that
 $\Phi(W) = \frac{1}{2} W^T W + C \sum \xi_i$ 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

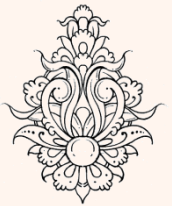
- رابطه‌ی لاگرانژ زیر تعریف می‌شود به گونه‌ای که همه‌ی نیازمندی‌ها را پوشش دهد:

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

- 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 لاگراثر از عبارت زیر محاسبه خواهند شد:

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

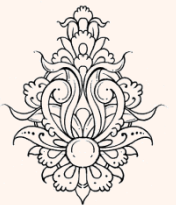
با در نظر گرفتن قیود زیر

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

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

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:

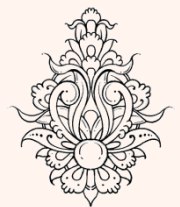
$$A \mathbf{x} \leq b$$

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

$$A_{eq} \mathbf{x} = b_{eq}$$

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

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



```

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

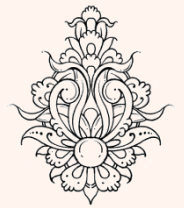
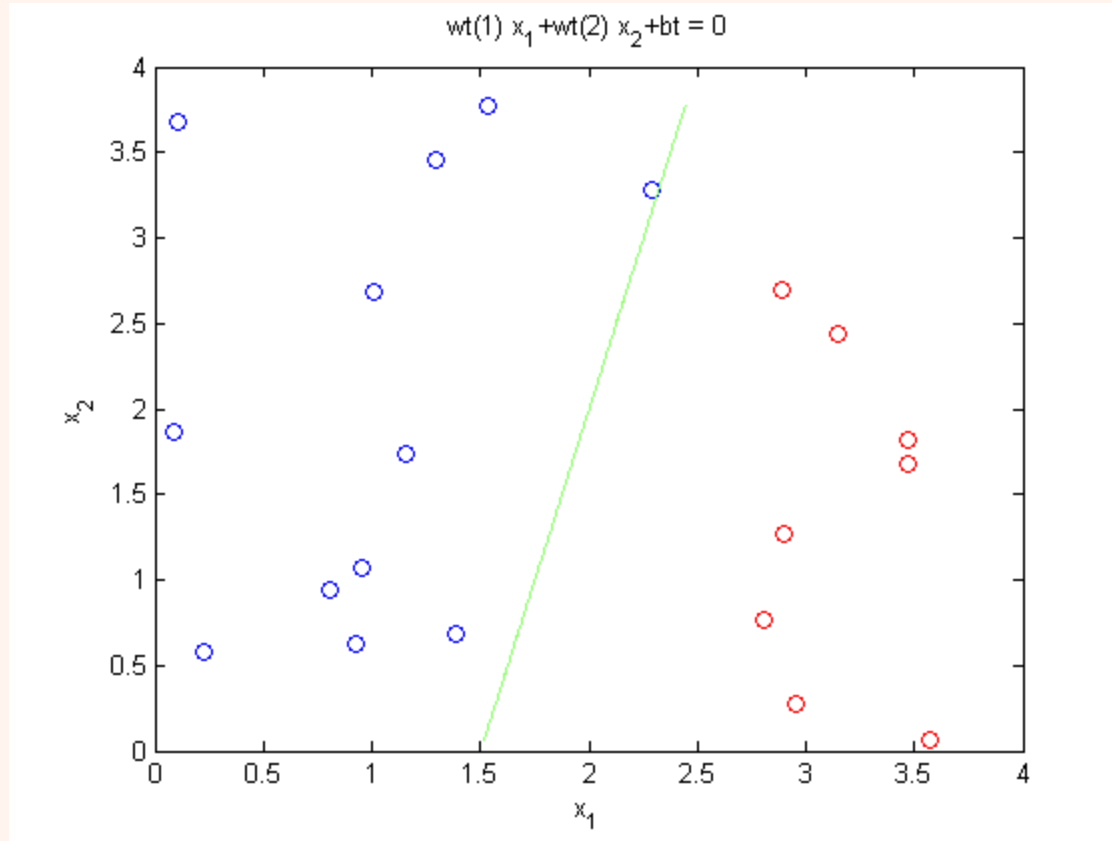
d = sign (wt (1) * X (1 ,:) + wt (2) * X (2 ,:) + bt) ;
x1min=min(X(1,:));
x1max=max(X(1,:));
x2min=min(X(2,:));
x2max=max(X(2,:));

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]);

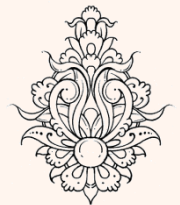
```

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...)

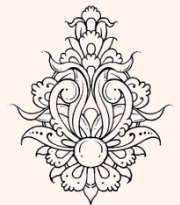
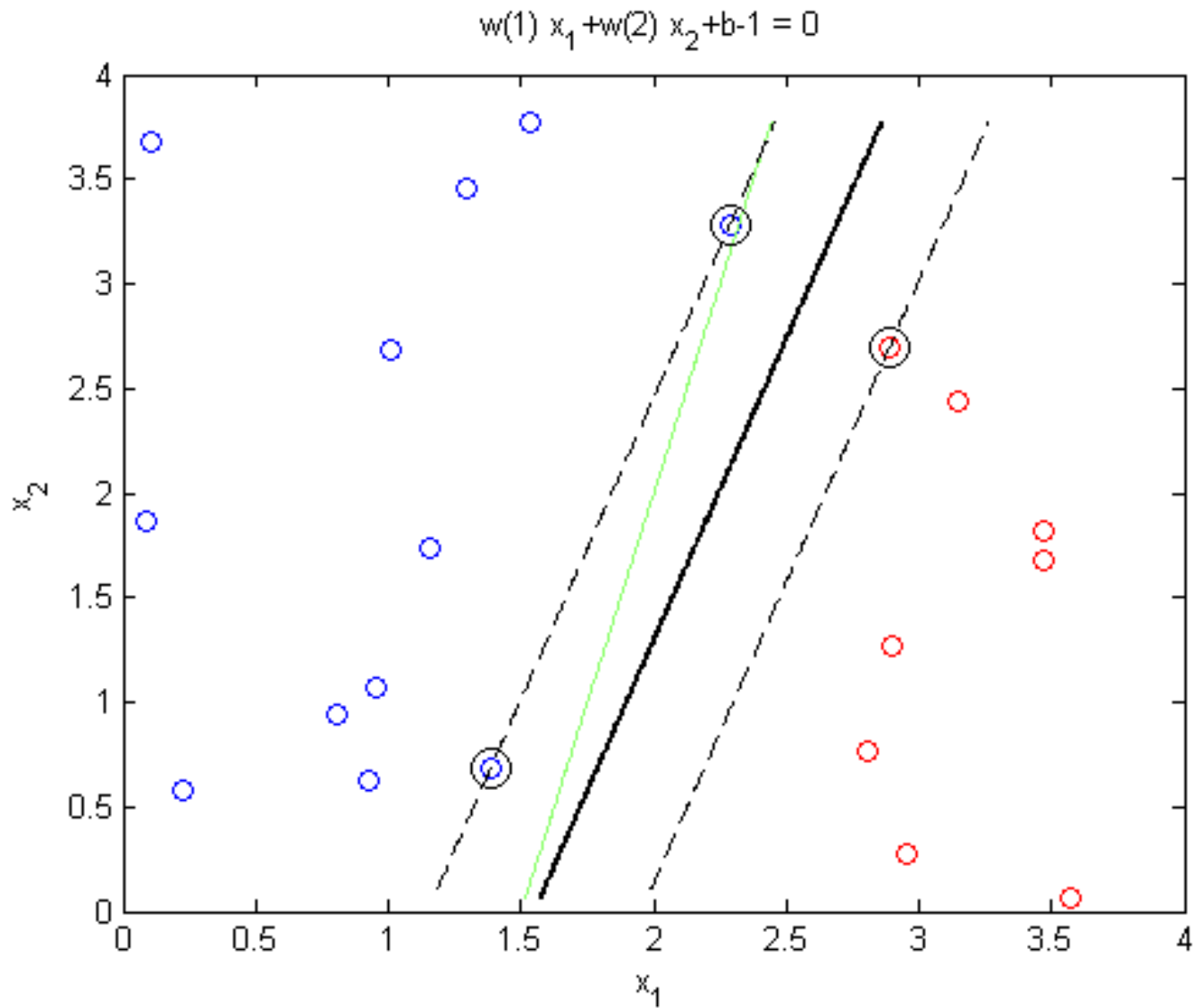
```
plot(X(1,Svs),X(2,Svs),'ko','MarkerSize',12);  
Line=@(x1,x2) w(1)*x1+w(2)*x2+b;  
LineA=@(x1,x2) w(1)*x1+w(2)*x2+b+1;  
LineB=@(x1,x2) w(1)*x1+w(2)*x2+b-1;
```

```
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','--');
```

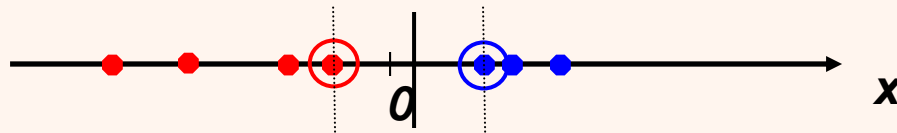
```
handleB=ezplot(LineB,[x1min x1max x2min x2max]);  
set(handleB,'Color','k','LineWidth',1,'LineStyle','--');
```



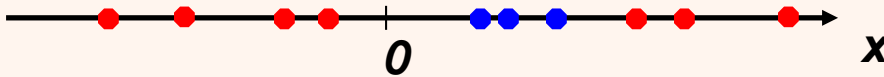


SVM غیرخطی

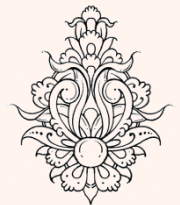
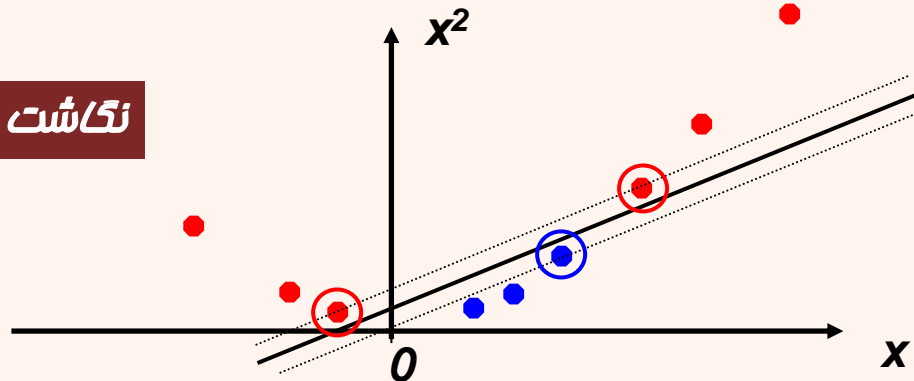
- برای داده‌هایی که قابلیت جداسازی خطی دارند، عملکرد سیستم بسیار خوب است.



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

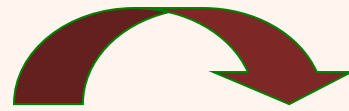


نگاشت به یک فضای *High Dimension*

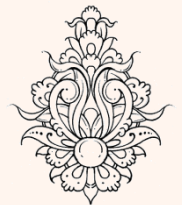
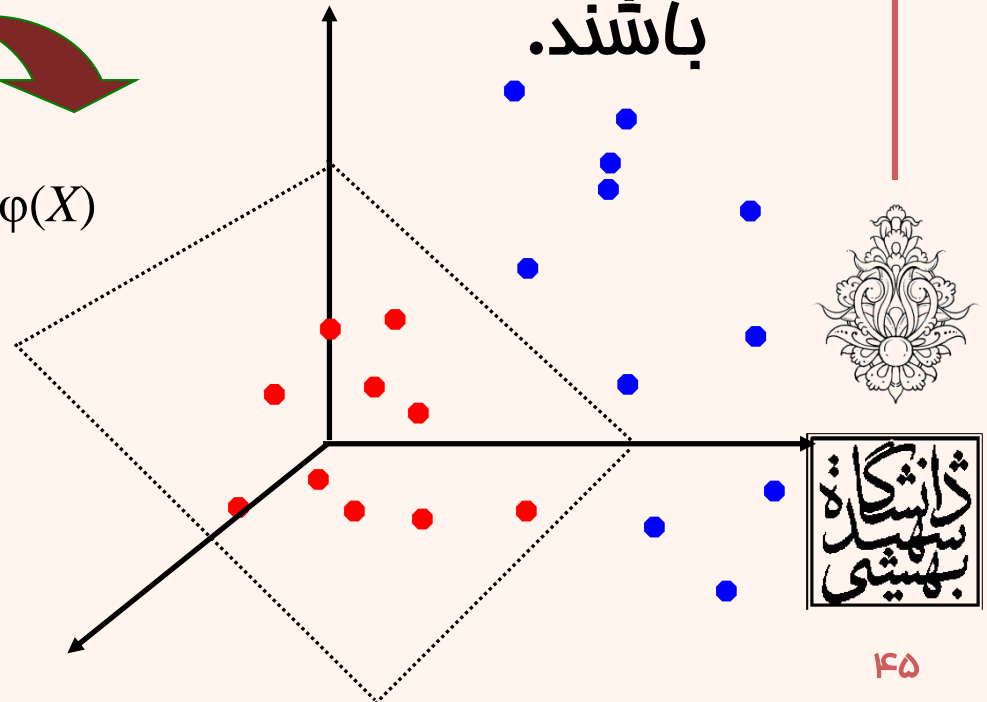
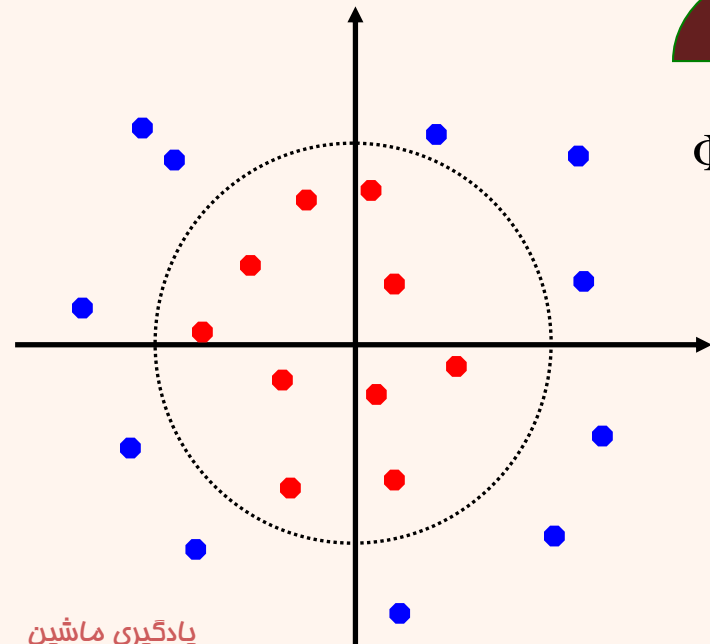


نگاشت به فضای بالاتر

- 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.

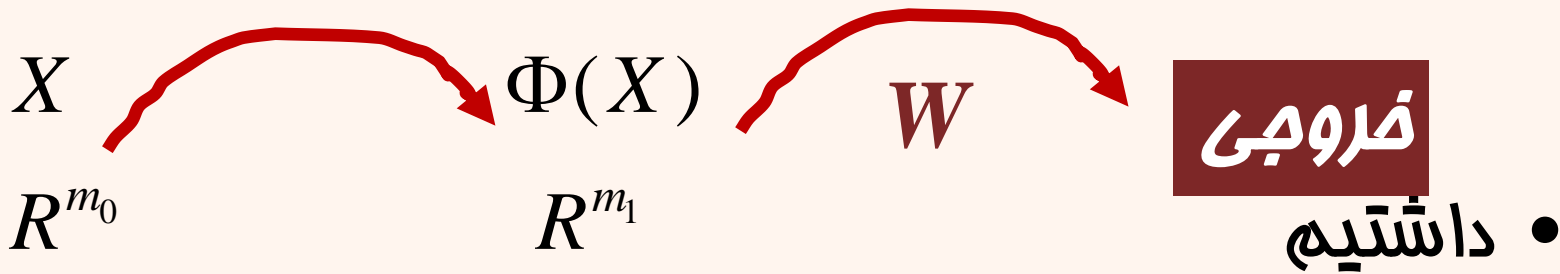


$$\Phi: X \rightarrow \phi(X)$$



ژانسه
سپیدی
بهشتی

نگاشت به فضای بالاتر



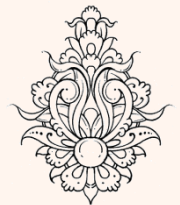
$$W^T X + b = 0$$

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

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

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

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



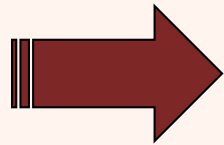
نگاشت به فضای بالاتر

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

• با فرض $\varphi_0(\mathbf{X}) = 1$

• خواهیم داشت:

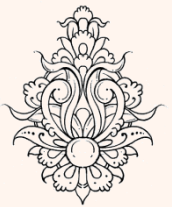
$$\sum_{j=0}^{m1} w_j \varphi_j(X) = 0$$



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

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

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



نگاشت به فضای بالاتر

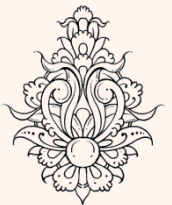
- در این مرحله تمامی شروط و قیودی که برای جداسازی خطی در نظر گرفتیم وجود دارد تنها به جای X_i ها $\Phi(X_i)$ در نظر گرفته می‌شود:

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

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

اسکالر \searrow $m_1 \times 1$

$$W_{opt}^T \Phi(\mathbf{X}) = 0 \quad \Rightarrow \quad \sum_{i=1}^N \alpha_i \cdot d_i \Phi^T(\mathbf{X}_i) \Phi(\mathbf{X}) = 0$$



نگاشت به فضای بالاتر

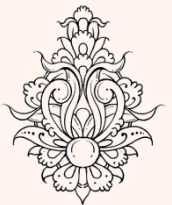
$$\sum_{i=1}^N \alpha_i \cdot d_i \Phi^T(X_i) \Phi(X) = 0$$

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

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

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

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



مثال

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

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

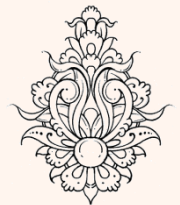
$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= (1 + \mathbf{x}_i^T \mathbf{x}_j)^2 = 1 + x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 + 2 x_{i1} x_{j1} + 2 x_{i2} x_{j2} \\ &= [1 \ x_{i1}^2 \ \sqrt{2} x_{i1} x_{i2} \ x_{i2}^2 \ \sqrt{2} x_{i1} \ \sqrt{2} x_{i2}]^T [1 \ x_{j1}^2 \ \sqrt{2} x_{j1} x_{j2} \ x_{j2}^2 \ \sqrt{2} x_{j1} \ \sqrt{2} x_{j2}] \\ &= \boldsymbol{\varphi}(\mathbf{x}_i)^T \boldsymbol{\varphi}(\mathbf{x}_j), \end{aligned}$$

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

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 لاگراثر بیشینه در عبارت زیر است:

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

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

با در نظر گرفتن قيود زیر

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

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.

$$g(X) = \sum \alpha_i d_i K(X_i, X) + b$$



کرنل‌های معمول

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

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

polynomial

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

RBF

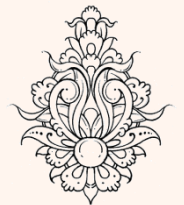
$$\exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_i\|^2\right)$$

tanh

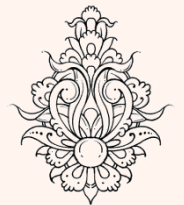
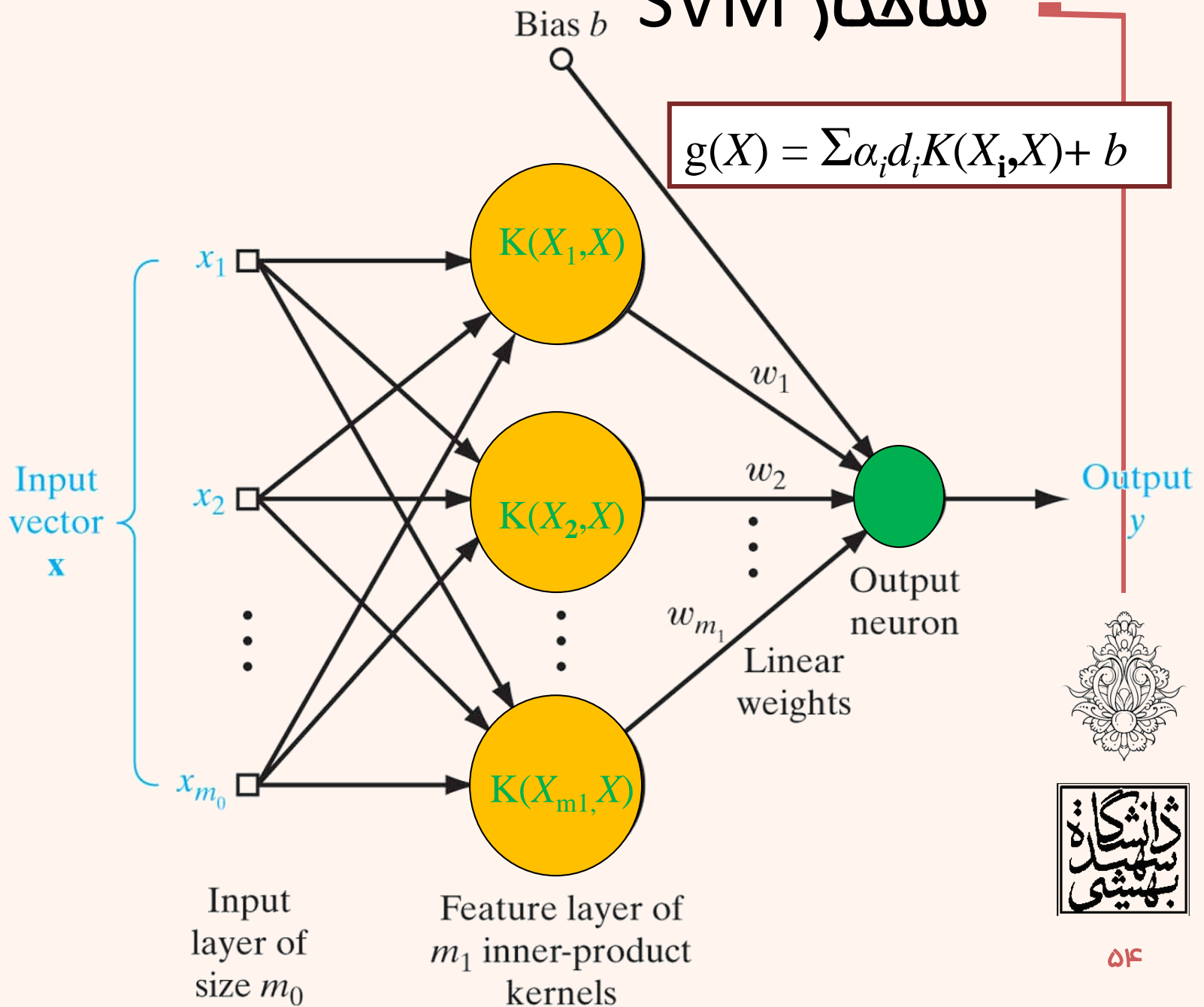
$$\tanh\left(\beta_0 \mathbf{x}^T \mathbf{x}_i + \beta_1\right)$$

chi-squared kernel

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

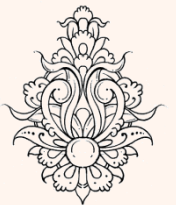


ساختار SVM



چند نکته

- SVM به گونه‌ای است که به شیوه‌های رایج برای طراحی شبکه‌های RBF و MLP نیازی ندارد.
 - در SVM، ابعاد فضای خصیصه توسط بردارهای پشتیبان مشخص می‌شود.
 - تعداد توابع شعاعی مورد استفاده و مراکز آن به صورت خودکار مشخص می‌گردد (RBF network).
 - تعداد لایه‌های مخفی و وزن‌ها به صورت خودکار مشخص می‌شود. (two-layer perceptron)
- پیچیدگی مسئله به ابعاد داده‌ها بستگی ندارد.



XOR Problem مثال

$$X_1 = [-1 \ -1] \rightarrow d_1 = -1$$

$$X_2 = [-1 \ 1] \rightarrow d_2 = +1$$

$$X_3 = [1 \ -1] \rightarrow d_3 = +1$$

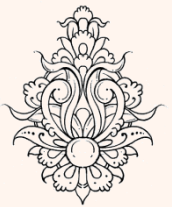
$$X_4 = [1 \ 1] \rightarrow d_4 = -1$$

$$N = 4$$

$$K(X, X_i) = \Phi^T(X) \cdot \Phi(X_i)$$

$$K(X, X_i) = (1 + X^T X_i)^2$$

• نمونه‌های آموزشی دو بعدی هستند.



XOR Problem

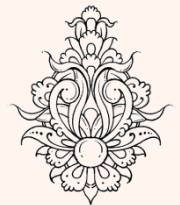
$$X_i = [x_{i1} \ x_{i2}]$$

$$X = [x_1 \ x_2]$$

$$K(X, X_i) = (1 + X^T X_i)^2$$

$$\begin{aligned} &= (1 + [x_{i1} \ x_{i2}] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix})^2 = (1 + x_{i1}x_1 + x_{i2}x_2)^2 \\ &= 1 + x_1^2 x_{i1}^2 + 2x_1 x_2 x_{i1} x_{i2} + x_{i2}^2 x_2^2 + 2x_1 x_{i1} + 2x_2 x_{i2} \end{aligned}$$

- حال اگر بخواهیم پاسخ به دست آمده را با ضرب داخلی دو بردار $\phi(X)$ و $\phi(X_i)$ نشان دهیم خواهیم داشت:

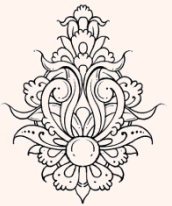


XOR Problem

$$= 1 + x_1^2 x_{i_1}^2 + 2x_1 x_2 x_{i_1} x_{i_2} + x_{i_2}^2 x_2^2 + 2x_1 x_{i_1} + 2x_2 x_{i_2}$$

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

$$\boldsymbol{\varphi}(x_i) = [1 + x_{i_1}^2, \sqrt{2}x_{i_1} x_{i_2}, x_{i_2}^2, \sqrt{2}x_{i_1}, \sqrt{2}x_{i_2}]^T \quad i=1,2,3,4$$

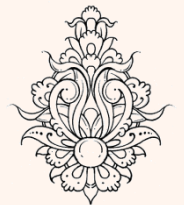


XOR Problem

$$\boldsymbol{\varphi}(\mathbf{x}) = [1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]^T$$

$$\boldsymbol{\varphi}(x_i) = [1 + x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}]^T \quad i=1,2,3,4$$

$$\begin{array}{l} X_1 = [-1 \ -1] \\ X_2 = [-1 \ 1] \\ X_3 = [1 \ -1] \\ X_4 = [1 \ 1] \end{array} \left[\begin{array}{cccccc} 1 & 1 & \sqrt{2} & 1 & -\sqrt{2} & -\sqrt{2} \\ 1 & 1 & -\sqrt{2} & 1 & -\sqrt{2} & \sqrt{2} \\ 1 & 1 & -\sqrt{2} & 1 & \sqrt{2} & -\sqrt{2} \\ 1 & 1 & \sqrt{2} & 1 & \sqrt{2} & \sqrt{2} \end{array} \right]$$

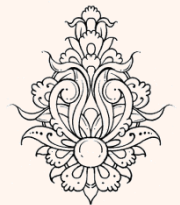


XOR Problem

$$K_{4 \times 4} = \begin{bmatrix} 1 & 1 & \sqrt{2} & 1 & -\sqrt{2} & -\sqrt{2} \\ 1 & 1 & -\sqrt{2} & 1 & -\sqrt{2} & \sqrt{2} \\ 1 & 1 & -\sqrt{2} & 1 & \sqrt{2} & -\sqrt{2} \\ 1 & 1 & \sqrt{2} & 1 & \sqrt{2} & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ \sqrt{2} & -\sqrt{2} & -\sqrt{2} & \sqrt{2} \\ 1 & 1 & 1 & 1 \\ -\sqrt{2} & -\sqrt{2} & \sqrt{2} & \sqrt{2} \\ -\sqrt{2} & \sqrt{2} & -\sqrt{2} & \sqrt{2} \end{bmatrix}$$

$$K(X, X_i) = (1 + X^T X_i)^2$$

$$K_{4 \times 4} = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$



XOR Problem

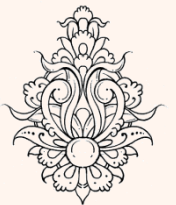
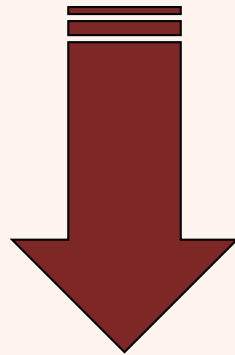
$$N = 4$$

$$Q(\alpha) = \sum_{i=1}^4 \alpha_i - \frac{1}{2} \sum_{i=1}^4 \sum_{j=1}^4 \alpha_i \alpha_j d_i d_j K(X_i, X_j)$$

$$Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4$$

$$- \frac{1}{2} (9\alpha_1^2 + 9\alpha_2^2 + 9\alpha_3^2 + 9\alpha_4^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 - 2\alpha_3\alpha_4)$$

• دست آوردن α_i ها بهینه منجر به روابط زیر می شود:



XOR Problem

$$1 - 9\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 0$$

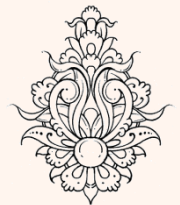
$$1 + \alpha_1 - 9\alpha_2 - \alpha_3 + \alpha_4 = 0$$

$$1 + \alpha_1 - \alpha_2 - 9\alpha_3 - \alpha_4 = 0$$

$$1 - \alpha_1 + \alpha_2 + \alpha_3 - 9\alpha_4 = 0$$

$$\alpha_i = \frac{1}{8}$$

$$Q(\alpha) = \frac{1}{4}$$



- بنابراین هر چهار ورودی، بردار پشتیبان هستند.
- پس از محاسبه α ها W_{opt} را محاسبه می‌کنیم:



XOR Problem

• جهت محاسبه‌ی اندازه‌ی وزن بهینه داریم:

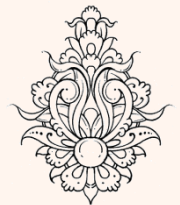
$$\frac{1}{2} \|\mathbf{w}_{opt}\|^2 = \frac{1}{4} \quad \Rightarrow \quad \|\mathbf{w}_{opt}\| = \frac{1}{\sqrt{2}}$$

$$\mathbf{w}_{opt} = \sum_{i=1}^N \alpha_i \cdot d_i(\Phi(\mathbf{x}_i))$$

• داشتیم:

$$\mathbf{w}_o = \frac{1}{8} [-\varphi(\mathbf{x}_1) + \varphi(\mathbf{x}_2) + \varphi(\mathbf{x}_3) - \varphi(\mathbf{x}_4)]$$

$$= \frac{1}{8} \left[- \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \\ -\sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ -\sqrt{2} \\ \sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \\ -\sqrt{2} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix} \right] = \begin{bmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

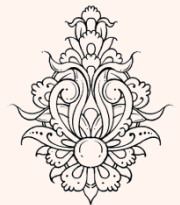


XOR Problem

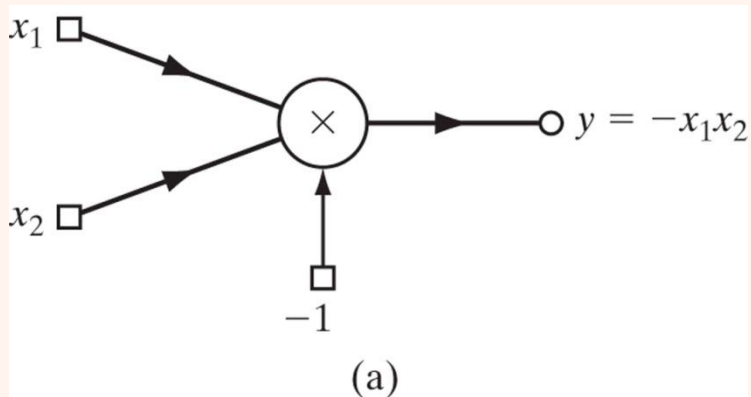
- رویه‌ی بهینه به وسیله‌ی رابطه‌ی زیر محاسبه می‌شود:

$$W_{opt}^T \varphi(X) = 0$$

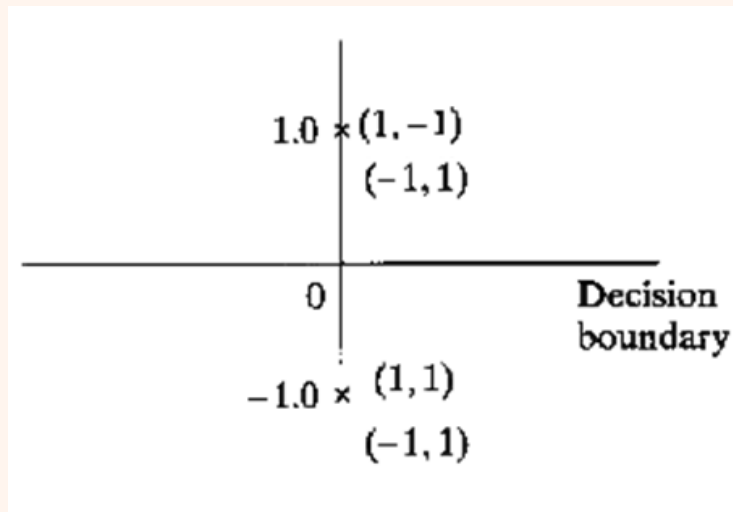
$$\begin{bmatrix} 0 & 0 & \frac{-1}{\sqrt{2}} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix} = 0 \quad \Rightarrow \quad -x_1x_2 = 0$$

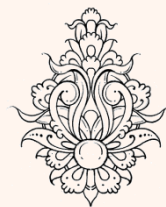
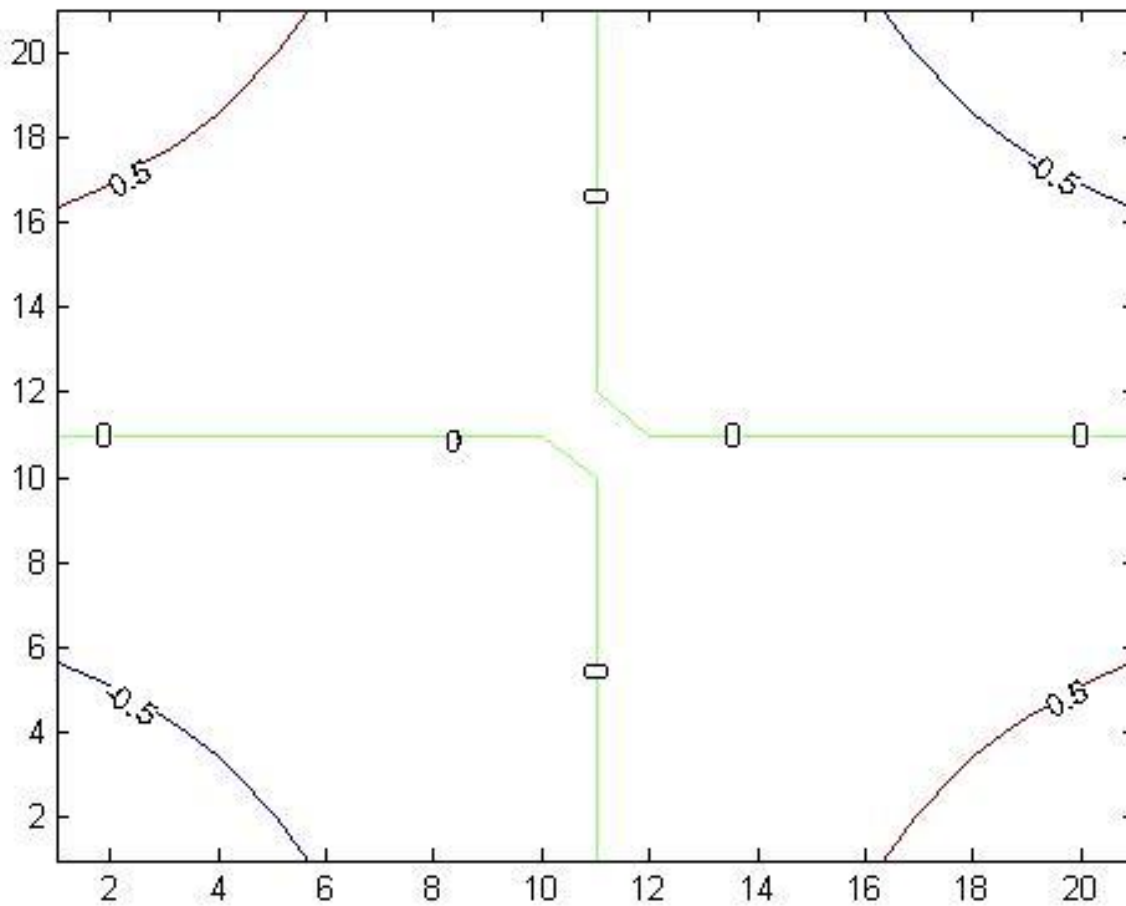


XOR Problem



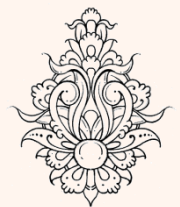
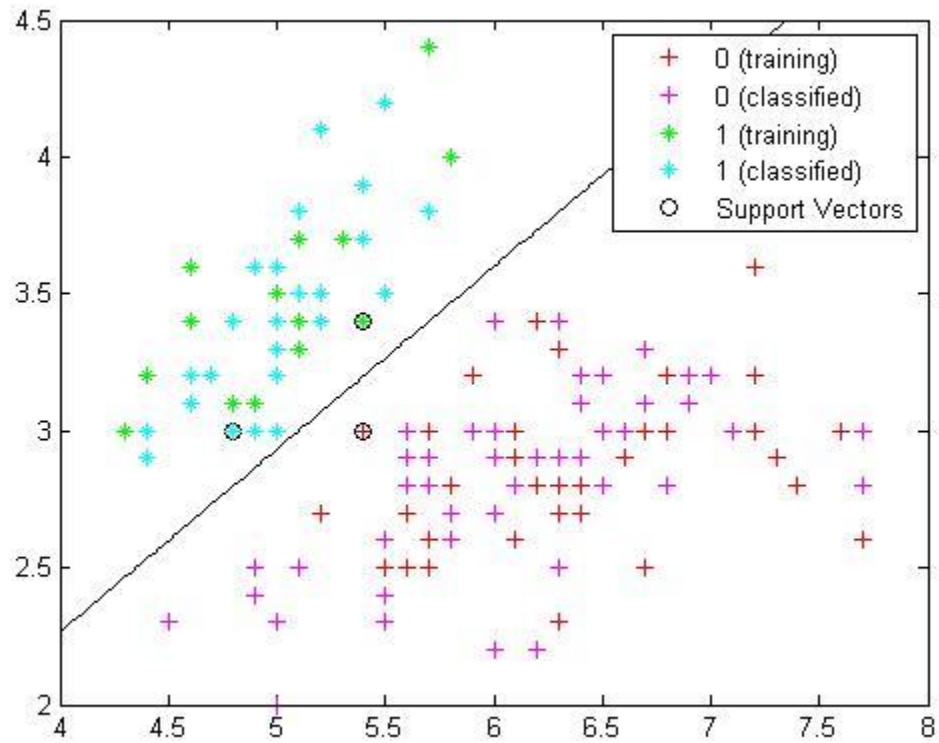
(a) Polynomial machine for solving the XOR problem. (b) Induced images in the feature space due to the four data points of the XOR problem.





```
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,'boxconstraint',1e6);
title(sprintf('Kernel Function: %s',...
    func2str(svmStruct.KernelFunction),...
    'interpreter','none'));
classes = svmclassify(svmStruct,data(test,:), 'showplot',true);
classperf(cp,classes,test);
cp.CorrectRate
```



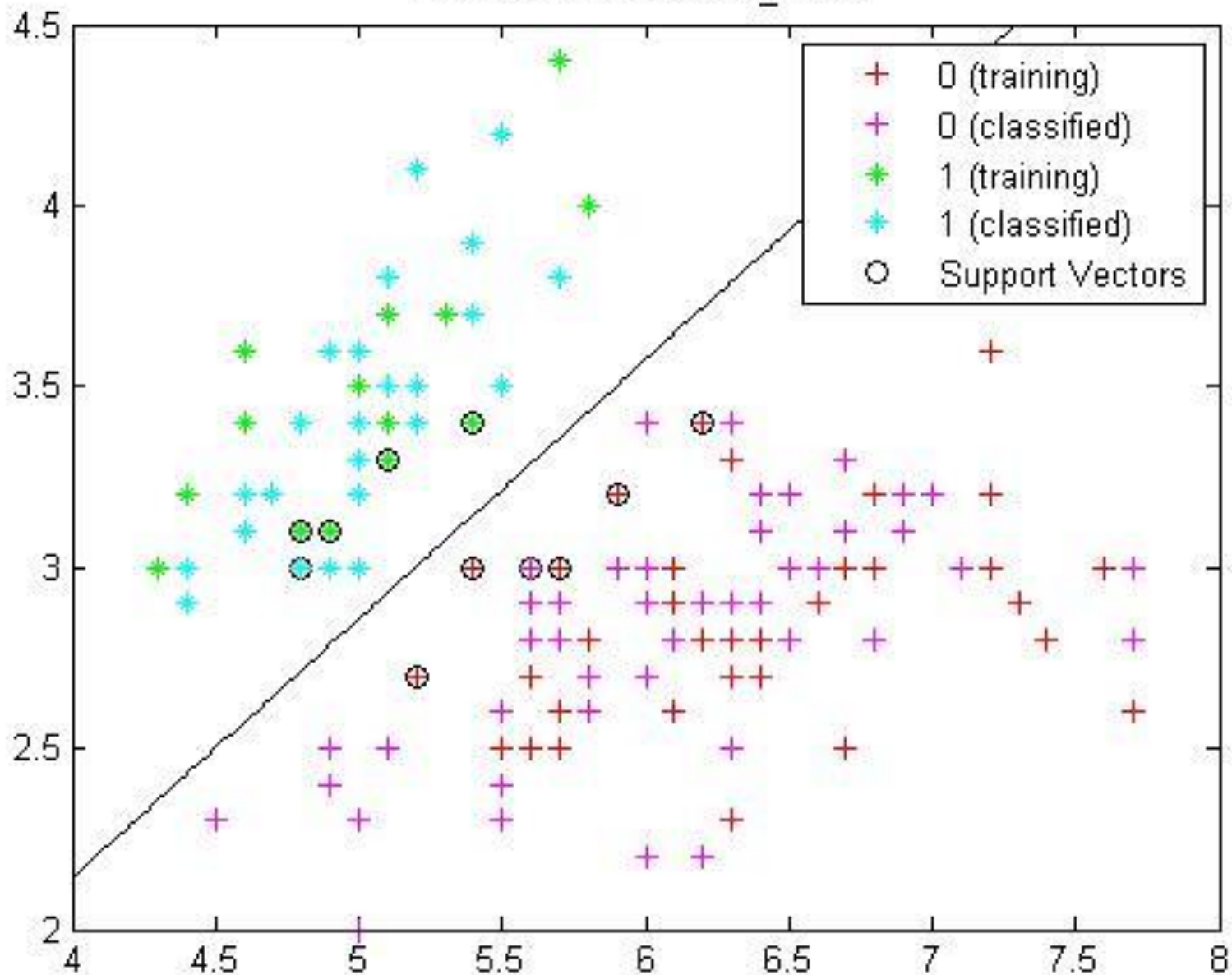




```
clear all;
close all;
load fisheriris
data = [meas(:,1), meas(:,2)];
groups = ismember(species,'setosa');
[train, test] = crossvalind('holdOut',groups);
cp = classperf(groups);
svmStruct = svmtrain(data(train,:),groups(train),'showplot',true);
title(sprintf('Kernel Function: %s',...
    func2str(svmStruct.KernelFunction),...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



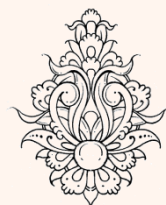
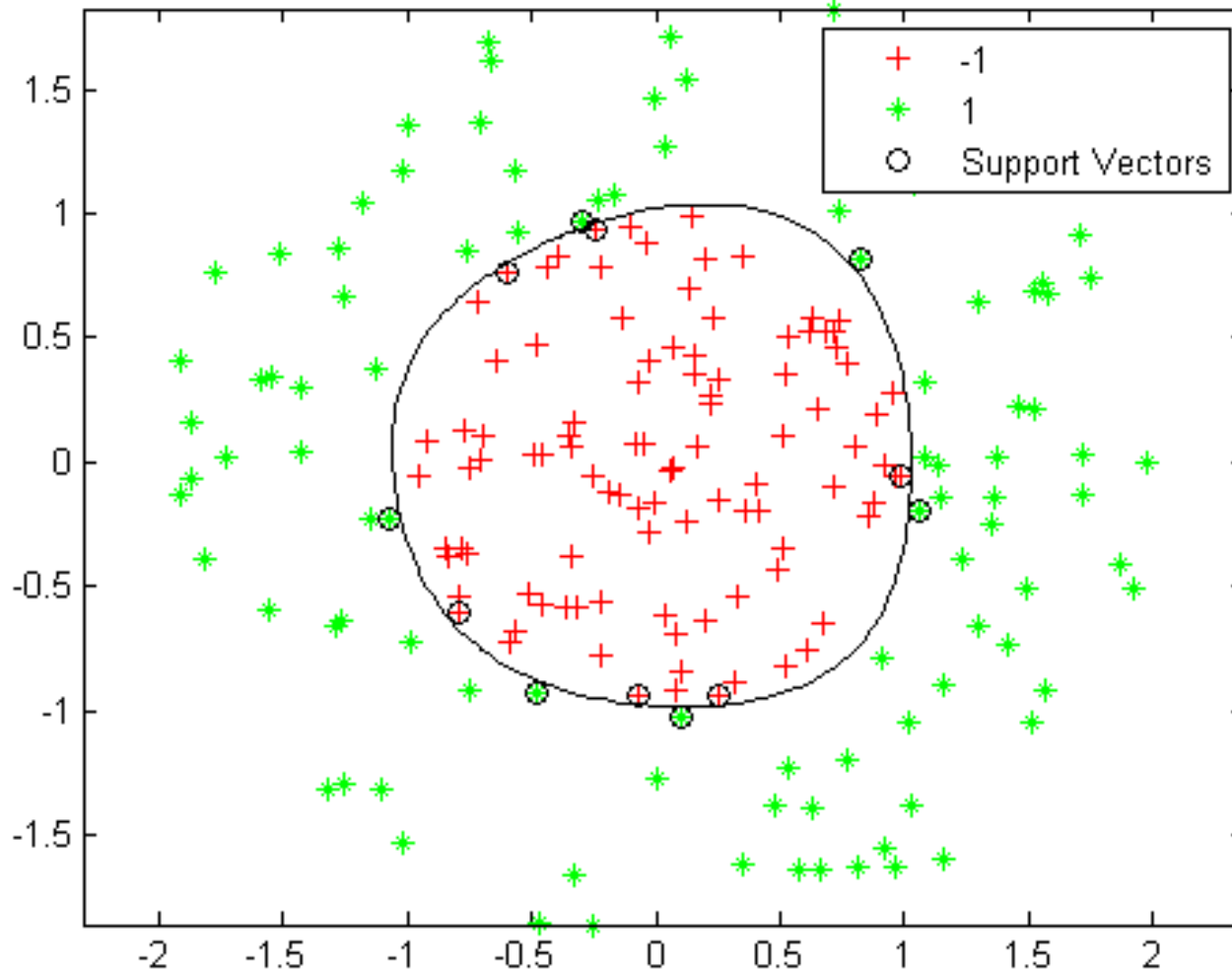
Kernel Function: linear_kernel





```
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;
c1 = svmtrain(data3,theclass,'Kernel_Function','rbf',...
    'boxconstraint',Inf,'showplot',true);
hold on
axis equal
```





ساخت کرنل‌های جدید

- براساس کرنل‌های موجود به سادگی می‌توان کرنل‌های جدید ساخت:

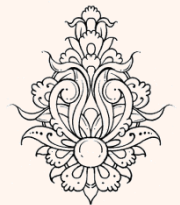
- هر تابع $K(\dots)$ یک کرنل است چنانچه معادل ضرب داخلی بردارهای حاصل از نگاشت باشد.

جمع دو کرنل

$$K(x, y) = c_1 K_1(x, y) + c_2 K_2(x, y) \quad c_1, c_2 \geq 0$$

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

$$\phi(x) \cdot \phi(y) = c_1 \phi_1(x) \cdot \phi_1(y) + c_2 \phi_2(x) \cdot \phi_2(y)$$



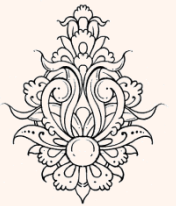
ساخت کرنل‌های جدید

ضرب دو کرنل

$$K(x, y) = K_1(x, y) \cdot K_2(x, y)$$

$$\phi(x) = \left(\phi_{1,i}(x) \phi_{2,j}(x) \right)_{i \in \{1, \dots, n\}, j \in \{1, \dots, m\}}$$

$$\begin{aligned} \phi(x) \cdot \phi(y) &= \sum_{i,j} \phi_{1,i}(x) \phi_{2,j}(x) \phi_{1,i}(y) \phi_{2,j}(y) \\ &= \sum_i \phi_{1,i}(x) \phi_{1,i}(y) \sum_j \phi_{2,j}(x) \phi_{2,j}(y) \\ &= \sum_i \phi_{1,i}(x) \phi_{1,i}(y) K_2(x, y) \\ &= K_1(x, y) K_2(x, y) \end{aligned}$$



ساخت کرنل‌های جدید (ادامه...)

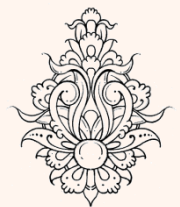
$$k(x, z) = k_1(f(x), f(z)), \text{ where } f : X \rightarrow X.$$

$$k(x, z) = g(x)g(z), \text{ for } g : X \rightarrow R.$$

$$k(x, z) = f(k_1(x, z)),$$

$$k(x, z) = \exp(\tilde{k}(x, z))$$

$$= K_1(x, y) K_2(x, y)$$



$$k(x, z) = \exp\left(\frac{-\|x-z\|^2}{\sigma^2}\right)$$

$$\begin{aligned} k(x, z) &= \exp\left(\frac{-\|x-z\|^2}{\sigma^2}\right) = \exp\left(\frac{-\|x\|^2 - \|z\|^2 + 2x^T z}{\sigma^2}\right) = \\ &= \exp\left(\frac{-\|x\|^2}{\sigma^2}\right) \exp\left(\frac{-\|z\|^2}{\sigma^2}\right) \exp\left(\frac{2x^T z}{\sigma^2}\right) = \\ &= (g(x)g(z)) \exp(k_1(x, z)) \end{aligned}$$

