

## یادگیری ماشین ماشین‌های بردار پشتیبان

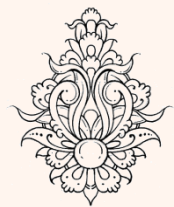


دانشگاه شهید بهشتی  
پژوهشکده‌ی فضای مجازی  
پاییز ۱۴۰۲  
احمد محمودی ازناوه



# 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
- مواجهه با مجموعه داده‌های imbalanced
- SVR



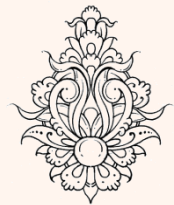


# تاریخچه

- 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.

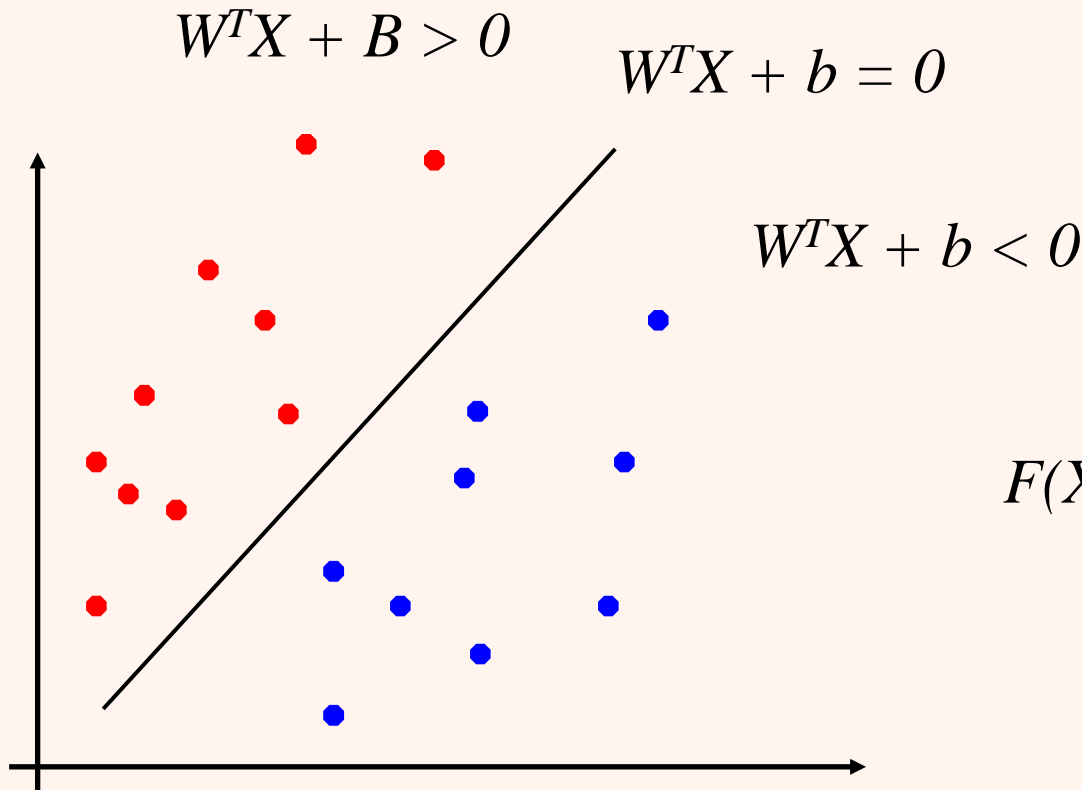


**Cortes, C. and V. Vapnik (1995). "Support-vector networks." Machine Learning 20(3): 273-297.**





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



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

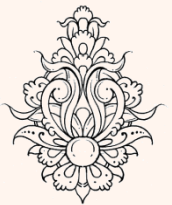
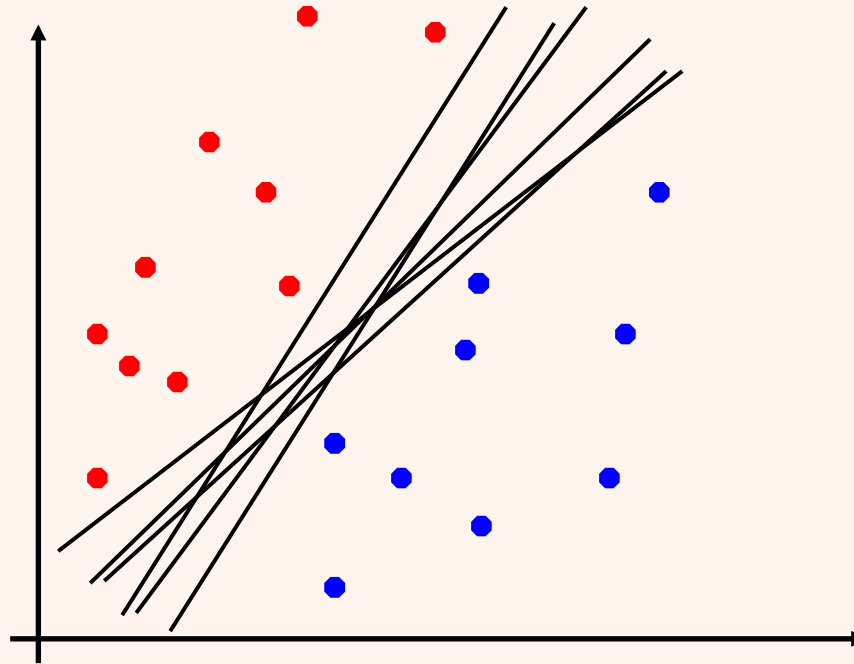




# مرز بهینه

## • سوال

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

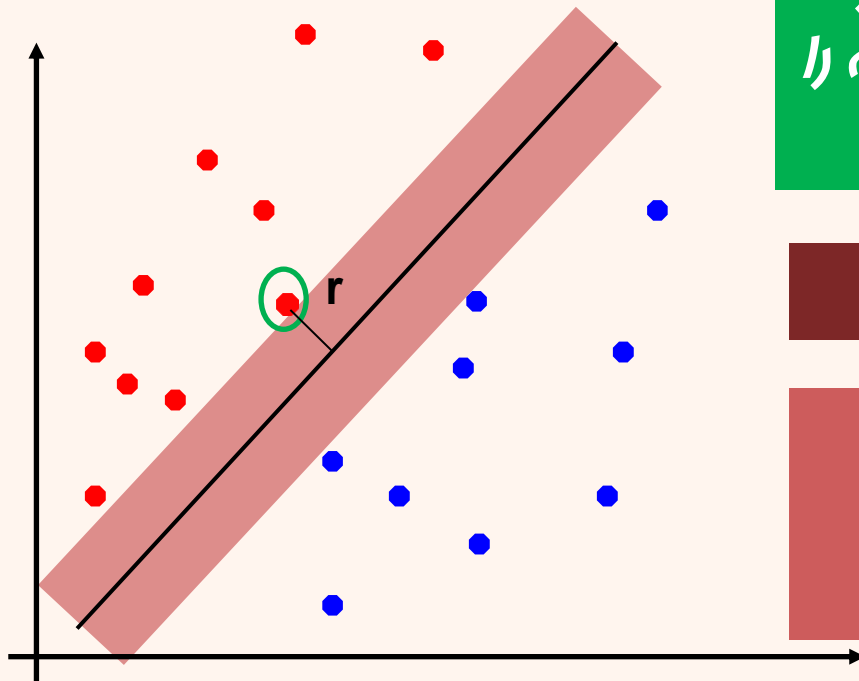




# مرز جداسازی

- 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.

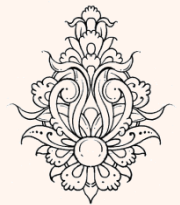
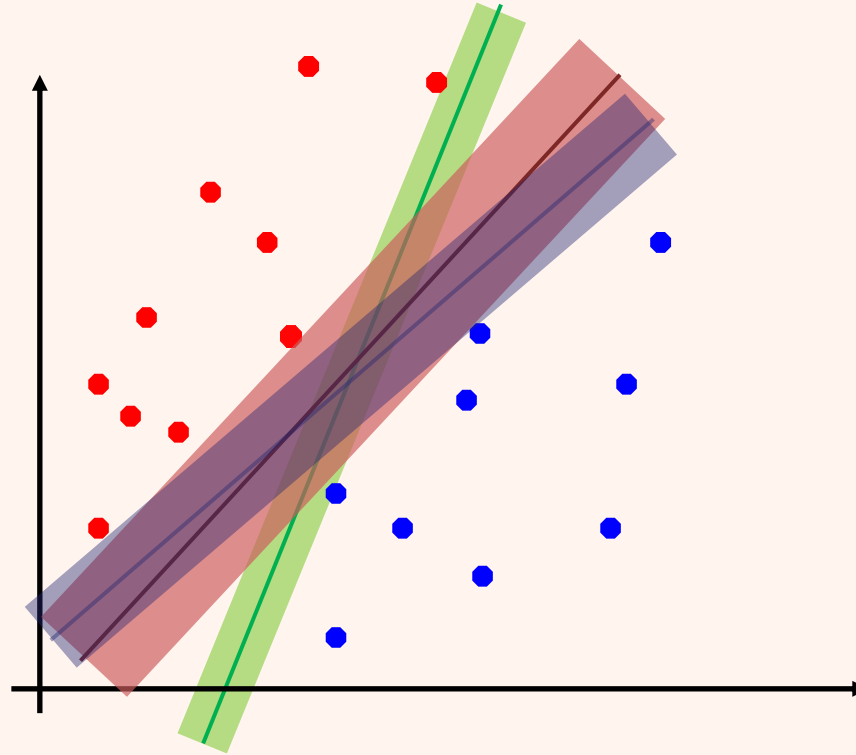
هدف ما کنیم نمودن  $r$  است.

یک ماشین به مشخص می‌کنیم هر مرزی که ماشین به پهن‌تری را نتیجه دهد، بهتر است.

دانشگاه  
شهید  
بهشتی



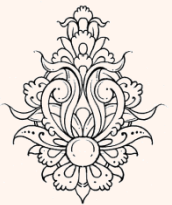
# مرز بهینه





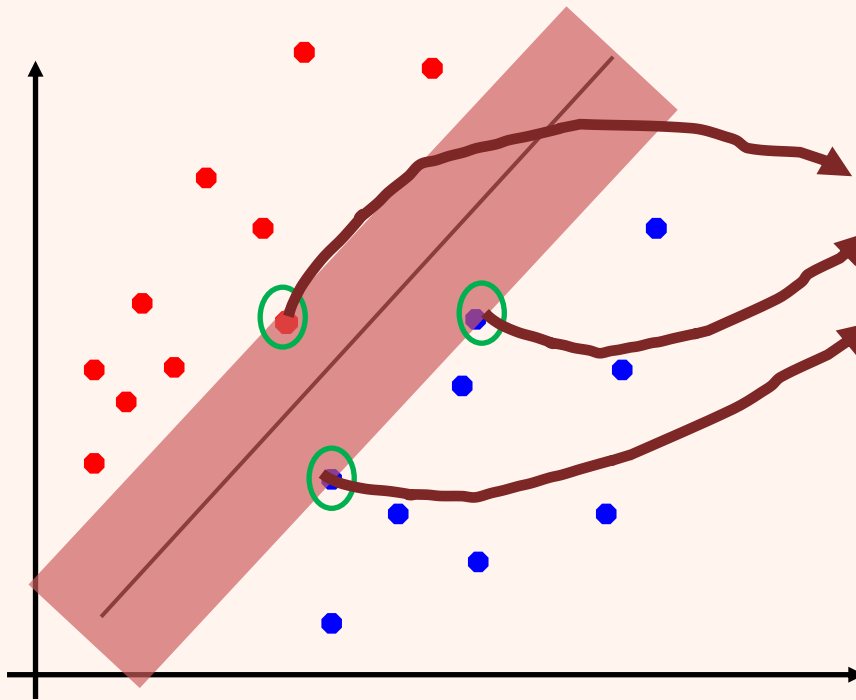
# ماشیهی ماکزیمم

- ماکزیمم 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.



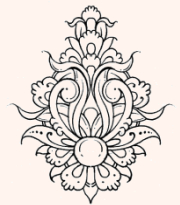


- به نمونه‌های روی مرز ماشیه «بردار پشتیبان» می‌گویند.



بردارهای پشتیبان

Optimal hyperplane





# مرز جداسازی

- برای معادله‌ی مرز جداسازی داشتیم:

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

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

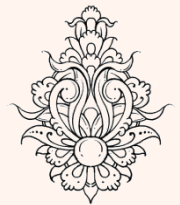
$$(X_i, d_i = -1) \quad W^T X_i + b < 0$$

- 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.



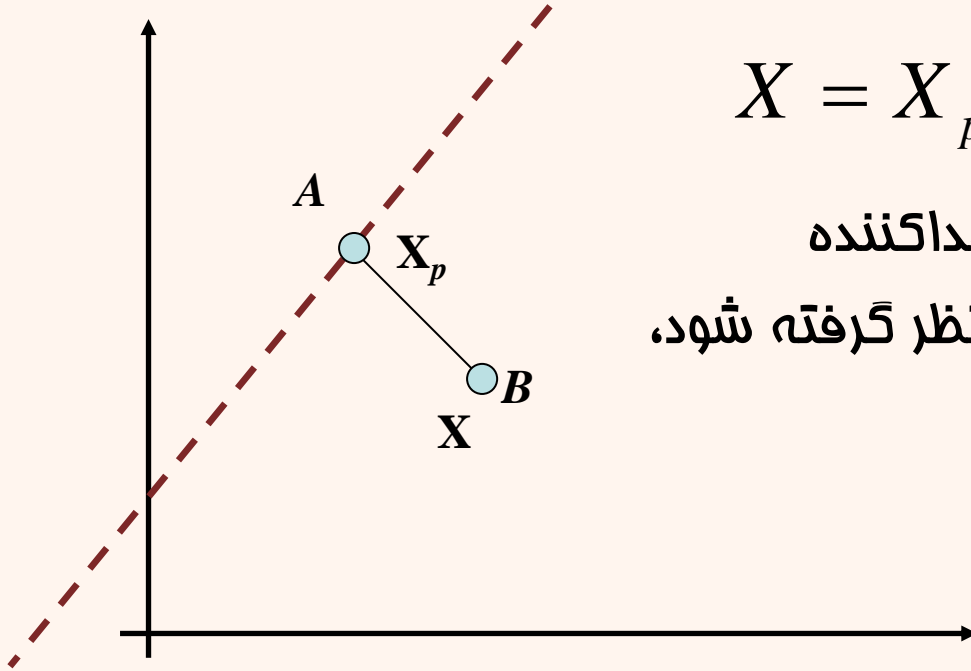


# مرز جداسازی (ادامه...)

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

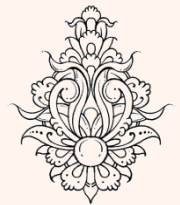
$$X = X_p + \overrightarrow{AB}$$

- $AB$  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...)

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

• داشتیم:

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

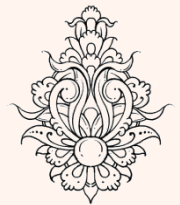
$$g(X) = W_{op}^T \left[ X_p + r \frac{W_{op}}{\|W_{op}\|} \right] + b_{op}$$

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

روی مرز پس برابر با صفر

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

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





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

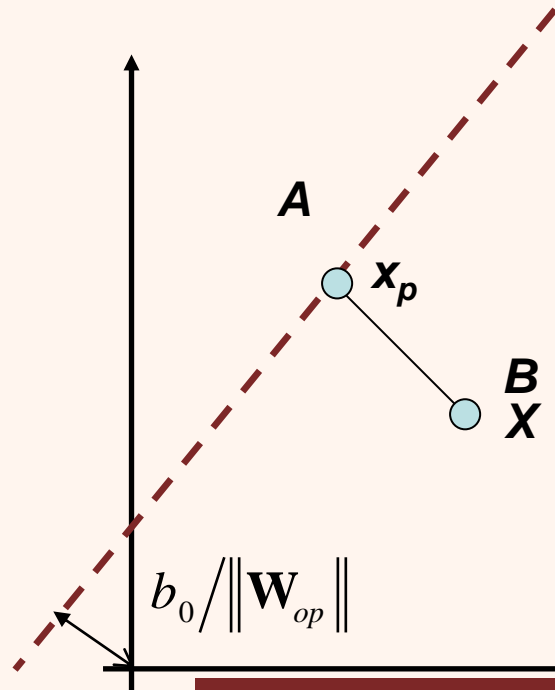
$$g(X) = r \|W_{op}\|$$

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

# مرز جداسازی (ادامه...)

هدف ماکزیمم نمودن  $r$  است.

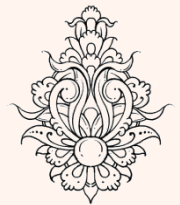
در این حالت تمت شرایطی می‌باید  $W$  کمینه گردد.



فاصله از مبدا مختصات

$$X=0$$

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



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



# مرز جداسازی (ادامه...)

ماتریس یافتن  $W_{op}$  و  $b_{op}$  است

• جداساز خطی را به صورت زیر در نظر می‌گیریم:

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

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

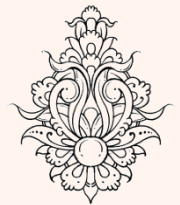
به صورت خطی داریم:

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

• رابطه‌ی بالا برای تمامی الگوهای آموزشی برقرار است.

• و در نتیجه برای بردارهای پشتیبان

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





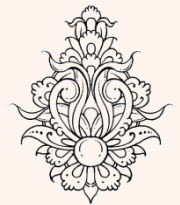
# مرز جداسازی (ادامه...)

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

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

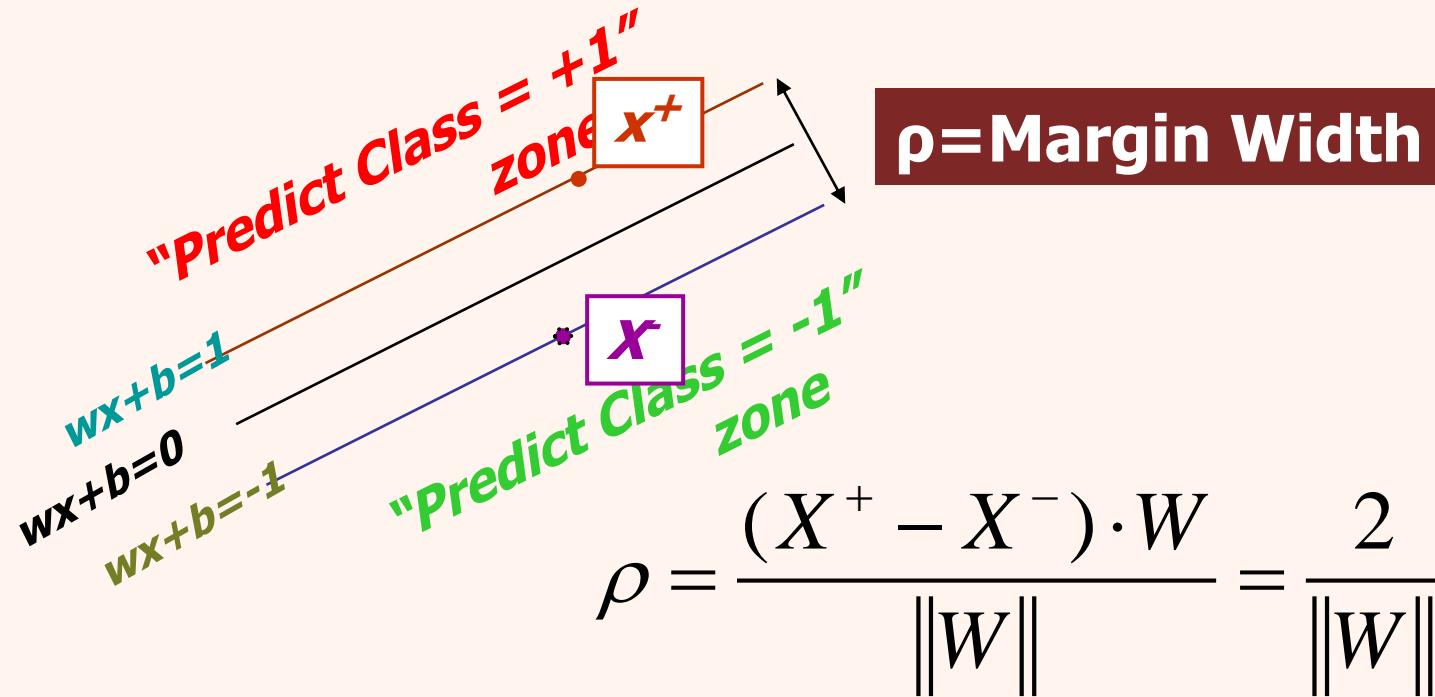
• در نتیجه فاصله‌ی دو بردار پشتیبان در دو طرف مرز:

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



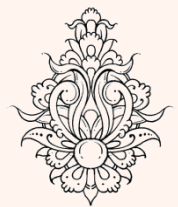


# جدایی‌پذیر خطی



می‌دانیم:

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





# جدایی‌پذیر خطی

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

- با توجه به دو رابطه‌ی
- به این نتیجه می‌رسیم که  $W_{op}$  می‌باید مینیمم گردد.

- این مسئله معادل مینیمم کردن

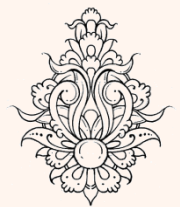
$$\Phi(W) = \frac{1}{2} W^T W$$

–  $\Phi$  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

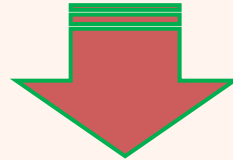




وزن‌ها و بایاس را به گونه‌ای بیابید که:

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

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.

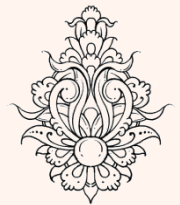
– قید عبارتی است که همواره بزرگتر یا مساوی صفر است.

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

به ازای هر نمونه یک قید داریم

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

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





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

$$J(W, \alpha) = f(W) - \alpha g(W)$$

قضیه:

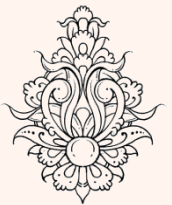
مساله بهینه‌سازی معادل یافتن پارامترها به صورت زیر است:

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

ابتدا عبارت داخلی در نظر گرفته می‌شود:

$$\max_{\alpha \geq 0} J(W, b, \alpha) = \begin{cases} f(W) & g(W) \geq 0 \\ \infty & g(W) < 0 \end{cases}$$

با کمینه کردن در حلقه خارجی دیده می‌شود که  $\min f(W)$  در شرایطی که  $g(W) \geq 0$  به دست می‌آید.



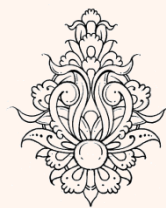


## Duality theorem

- با توجه به تابع اولیه و قید می‌توان نشان داد که حل این مساله (primal) معادل حل مساله 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)

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

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

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

$b_{op}$  به دست نمی آید

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

ولی یک قید می دهد

به ازای یک مقدار ثابت  $\alpha$  ، مقدار  $W$  و یک قید به دست می آید. با جایگزینی و استفاده از این روابط نسبت به  $\alpha$  با توجه به قیدها نقطه ی بهینه را به دست می آوریم.





# یافتن بهینه

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

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

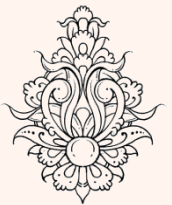
• برای مقادیر بهینه داشتیم:

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

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

• پس خواهیم داشت:

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

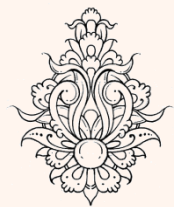




# یافتن بهینه

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

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



$\alpha_i$  ها وابسته به ضرب داخلی نمونه‌های آموزشی خروجی‌های مرتبط است



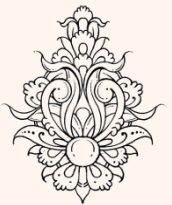
## Karush–Kuhn–Tucker(KKT) condition of optimization theory

- با توجه به شرط KKT خواهیم داشت:

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

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

- در نتیجه تنها  $\alpha_i$  متناظر با بردارهای پشتیبان غیرصفر خواهد بود.





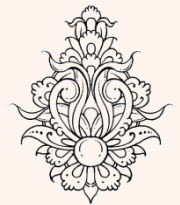
# یافتن رویه ی بهینه

- پس از به دست آوردن  $\alpha$  خواهیم داشت:

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

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

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





# 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  $\alpha_i$  مخالف صفر، نشان‌دهنده‌ی این است که  $X_i$  متناظرش یک بردار پشتیبان است.

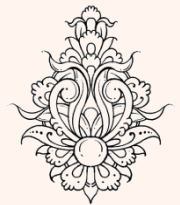
در این حالت تابع جداکننده همانند زیر است:

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

ضرب داخلی دو بردار

توجه:

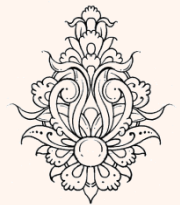
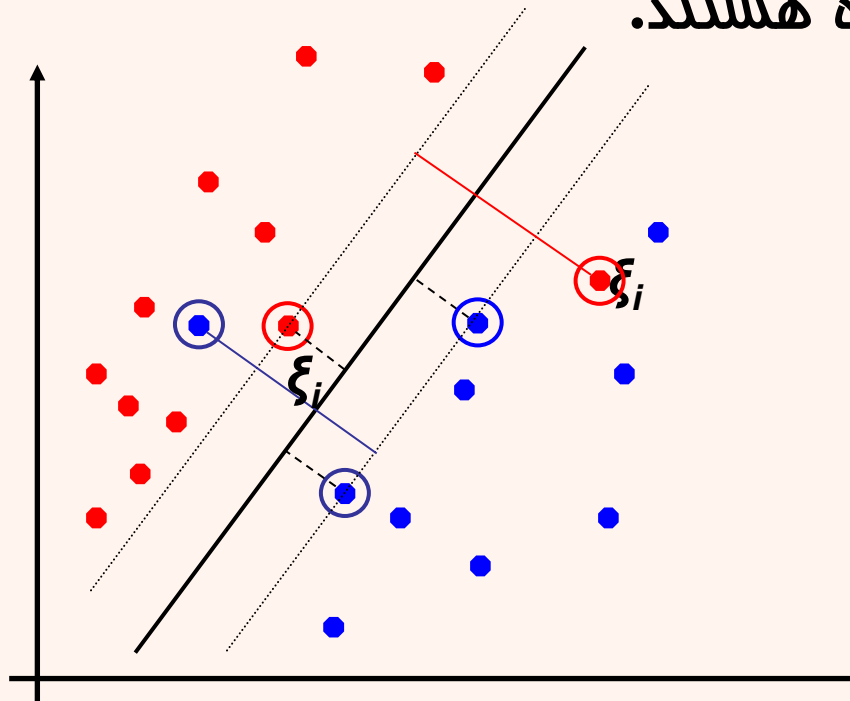
حل مساله بهینه‌سازی وابسته به محاسبه ضرب داخلی بین تمامی نمونه‌های آموزشی است.





# Soft Margin

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



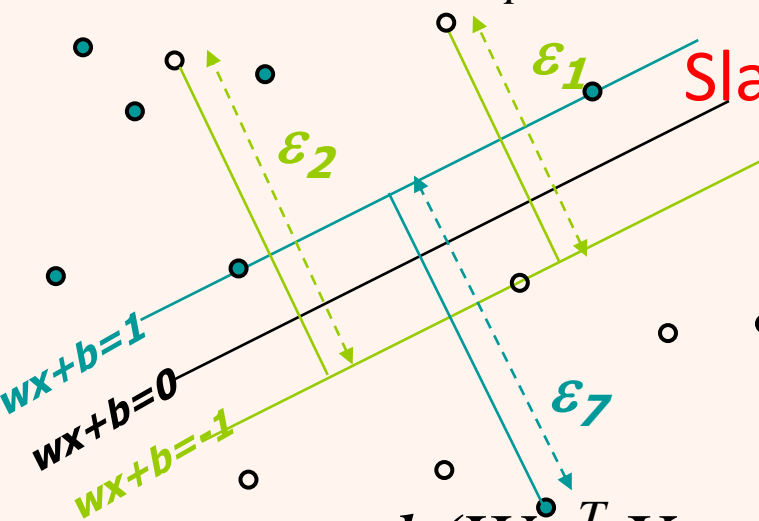


# Soft Margin

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

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

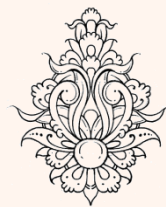
$$d_i(W_{op}^T X + b_{op}) \geq 1$$



- با اضافه کردن یک **Slack Variable** مسأله را بار دیگر بررسی می‌کنیم.

- این متغیر میزان انحراف از شرط فوق را نشان می‌دهد.

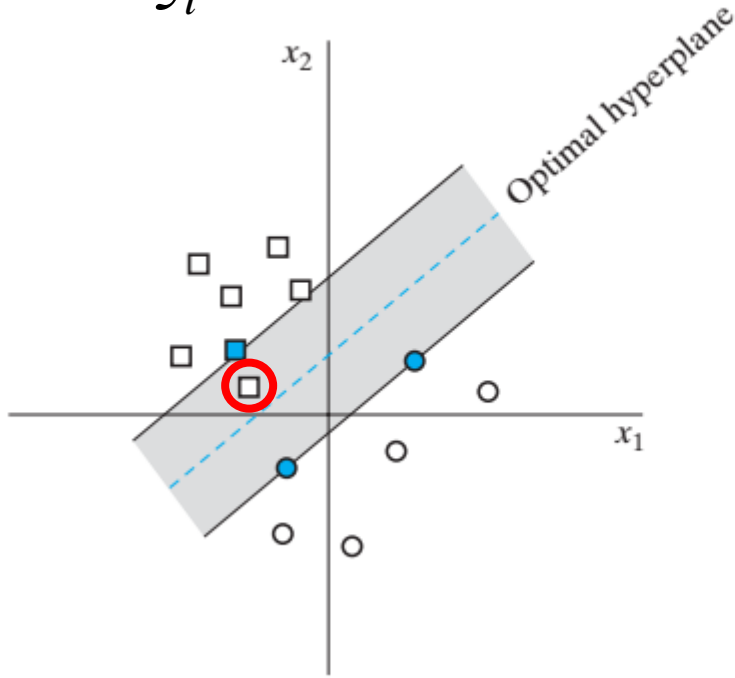
$$d_i(W_{op}^T X_i + b_{op}) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N$$



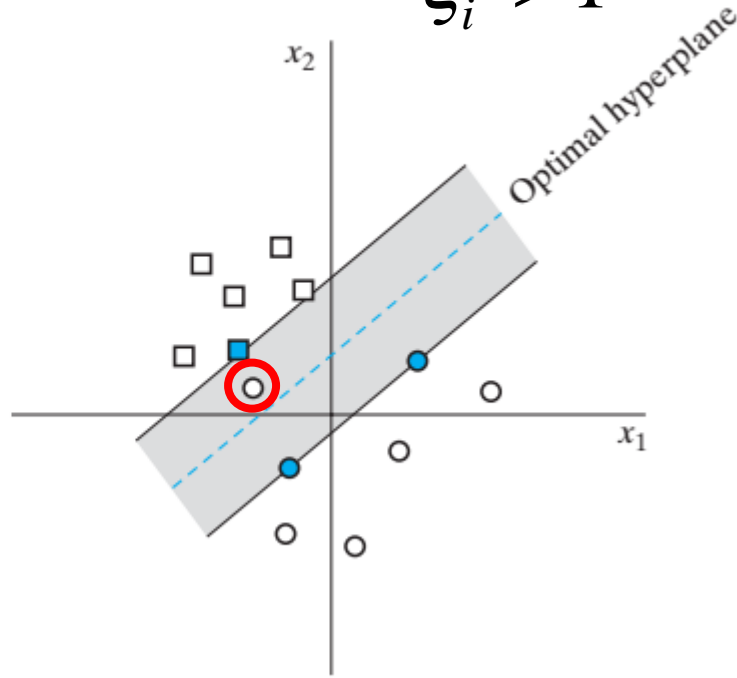


# Soft Margin Classification

$$0 \leq \xi_i \leq 1$$

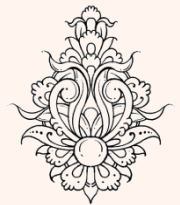


$$\xi_i > 1$$



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

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





# Soft Margin Classification

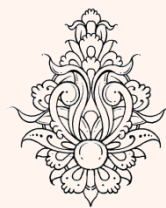
$$d_i(\mathbf{W}_{op}^T \mathbf{X} + b_{op}) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N$$

- 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 خطای 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:

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

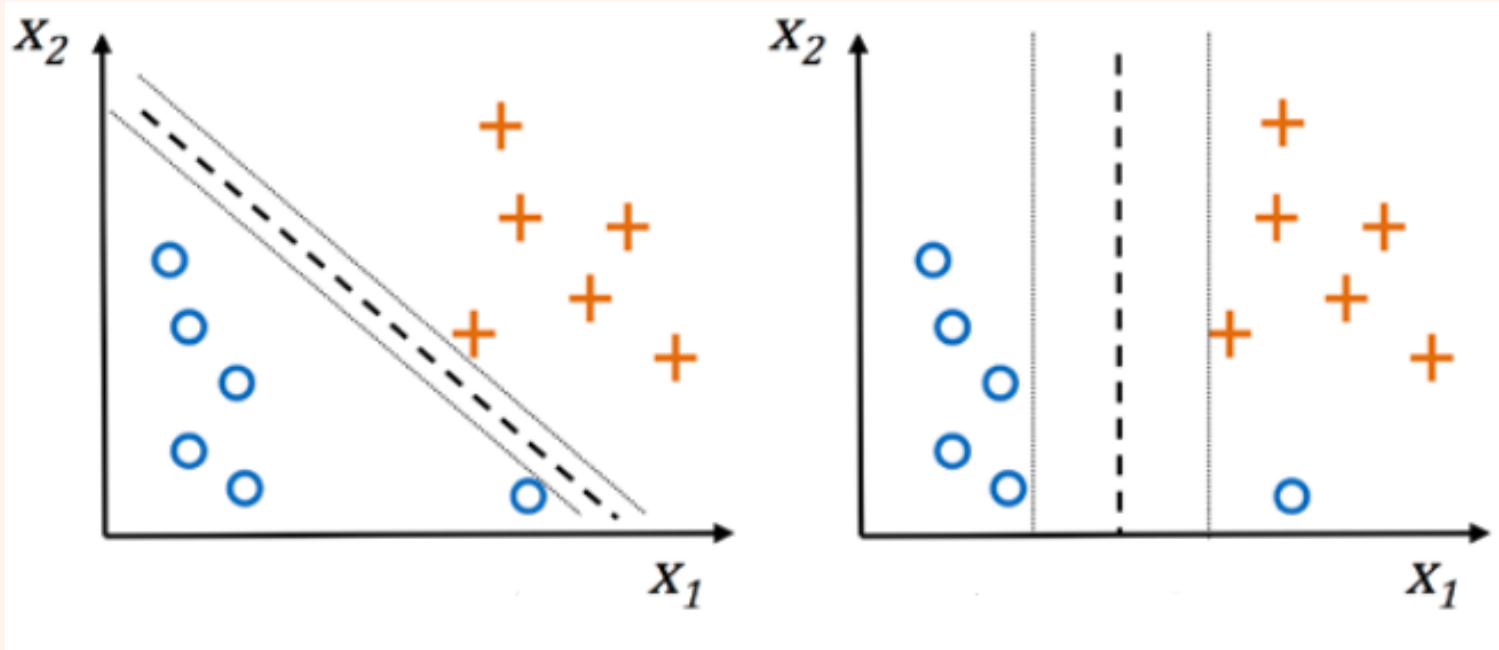
regularization parameter

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



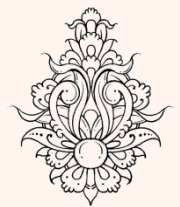


# مثال



**Large value for  $C$**

**Small value for  $C$**





# 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

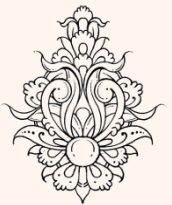
- رابطه‌ی لاگرانژ زیر تعریف می‌شود به گونه‌ای که همه‌ی نیازمندی‌ها را پوشش دهد:

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

– بخش آخر از این رو اضافه شده است که تا نامنفی بودن  $\xi$  را تضمین کند.

- حل این بهینه‌سازی معادل حل مساله‌ی زیر است:

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





• که دوگان این مساله به صورت زیر است:

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

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

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

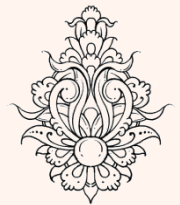
$$W = \sum_{i=1}^N \alpha_i d_i X_i$$

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

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

$$\frac{\partial J}{\partial \xi_i} = 0 \Rightarrow C - \alpha_i - \mu_i = 0 \quad \Rightarrow$$

$$\alpha_i = C - \mu_i$$





# Soft Margin Classification

در نهایت ضرایب لاگراثر از عبارت زیر محاسبه خواهند شد:

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

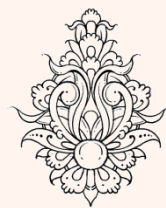
با در نظر گرفتن قیود زیر

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

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

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  $\mathbf{x}$  است به گونه‌ای که تابع درجه ۲ به شکل زیر کمینه شود:

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

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

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

- و شروط زیر برقرار باشد:

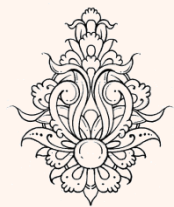
$$A\mathbf{x} \leq \mathbf{b}$$

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

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

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

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





```

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,:));

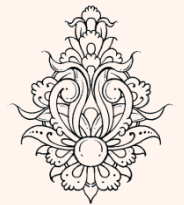
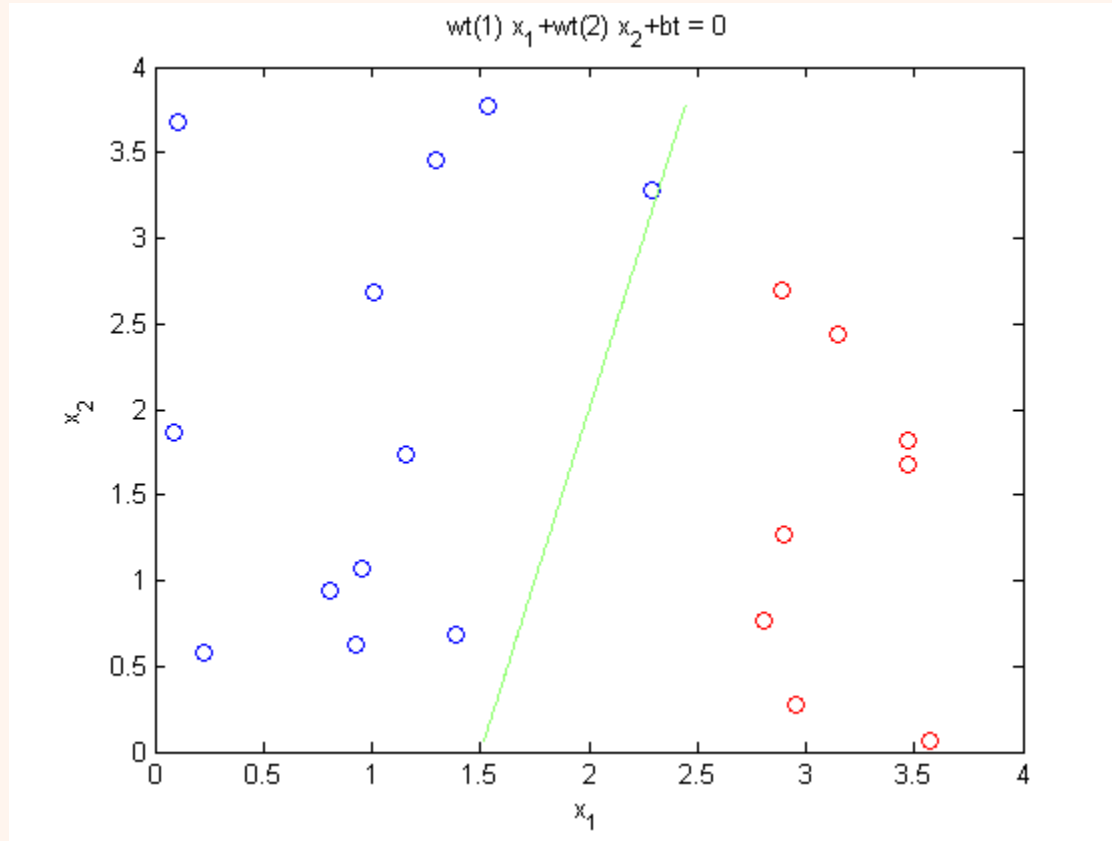
figure;
axis([x1min x1max x2min x2max]);
plot (X (1 ,find (d ==1)) ,X (2 ,find (d ==1)) , ' or ' ) ;
hold on
plot (X( 1,find (d ==-1)) ,X(2, find (d ==-1)) , ' ob ' ) ;

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

```



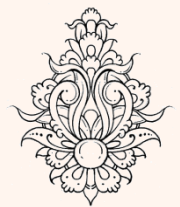
# مثال (ادامه...)





# 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;
```

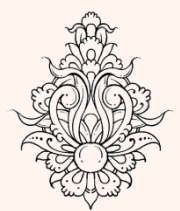
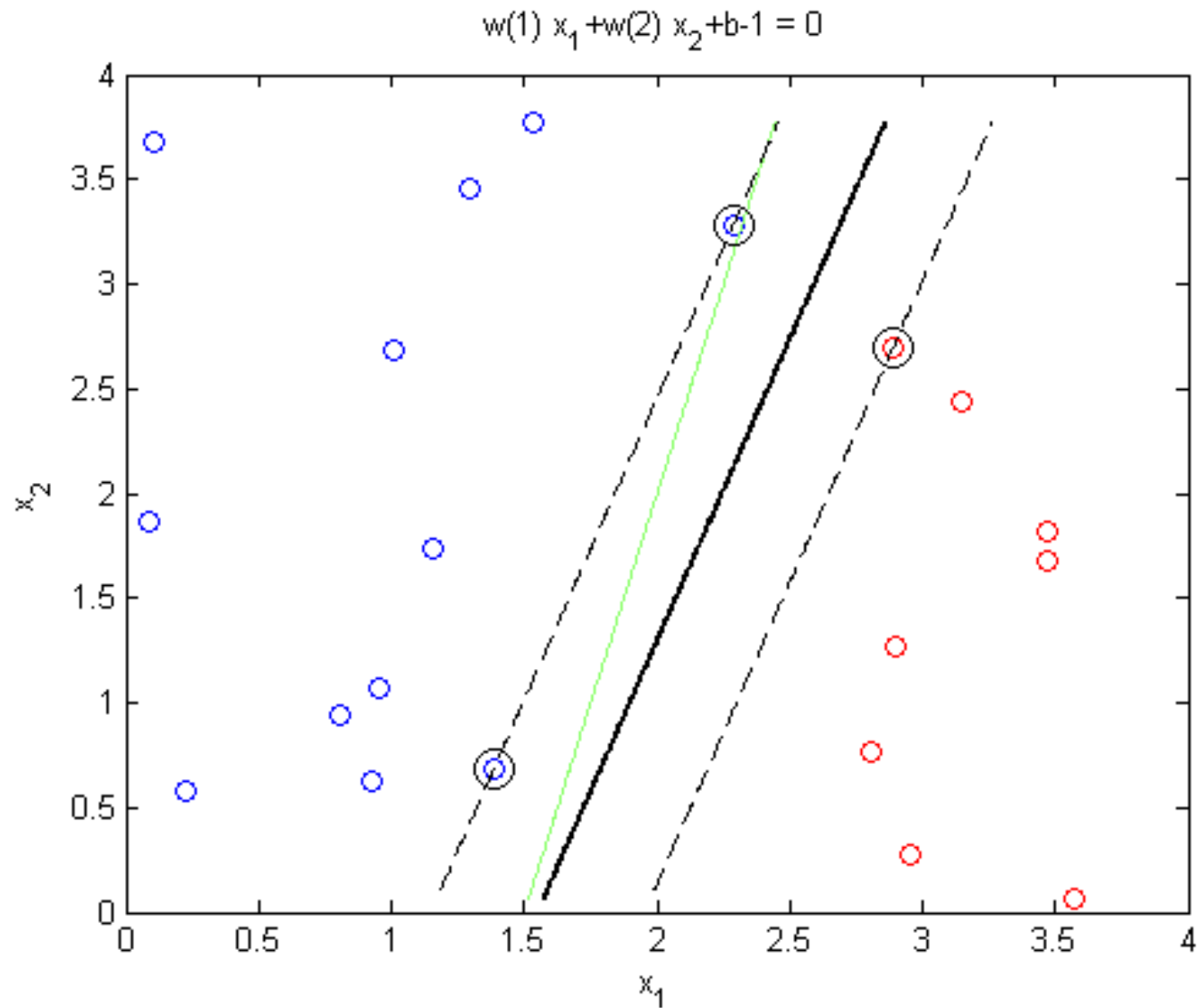
```
handle=ezplot(Line,[x1min x1max x2min x2max]);  
set(handle,'Color','k','LineWidth',2);
```

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



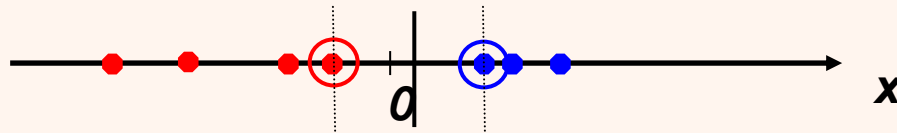




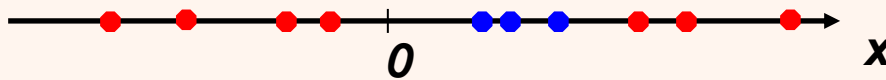


# SVM غیرخطی

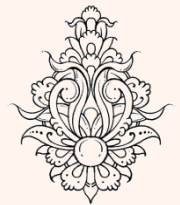
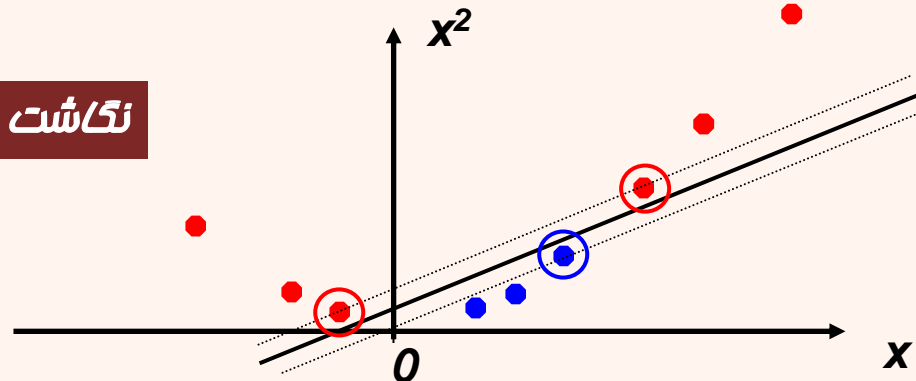
- برای داده‌هایی که قابلیت جداسازی خطی دارند، عملکرد سیستم بسیار خوب است.



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



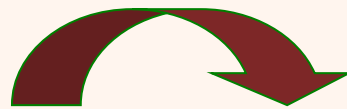
نگاشت به یک فضای *High Dimension*



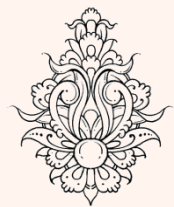
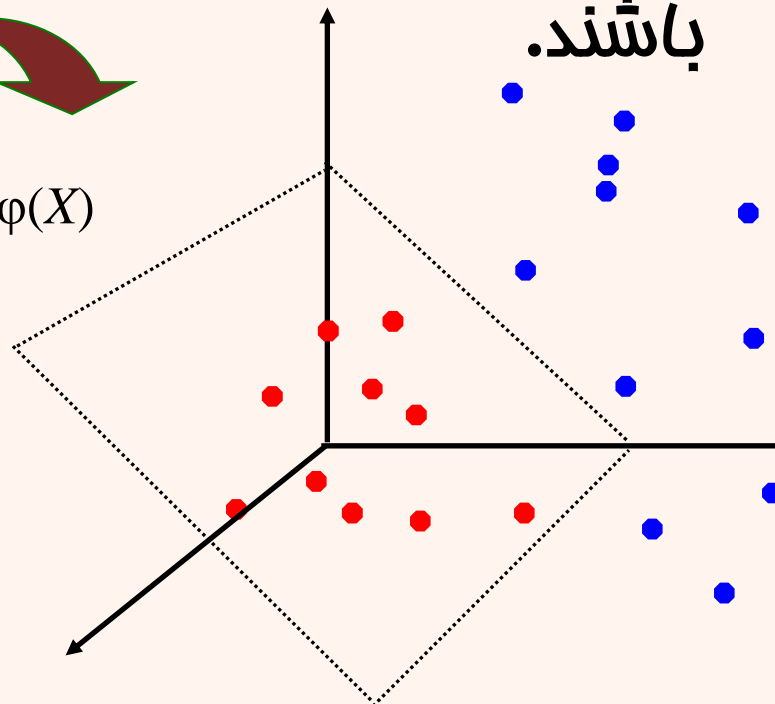
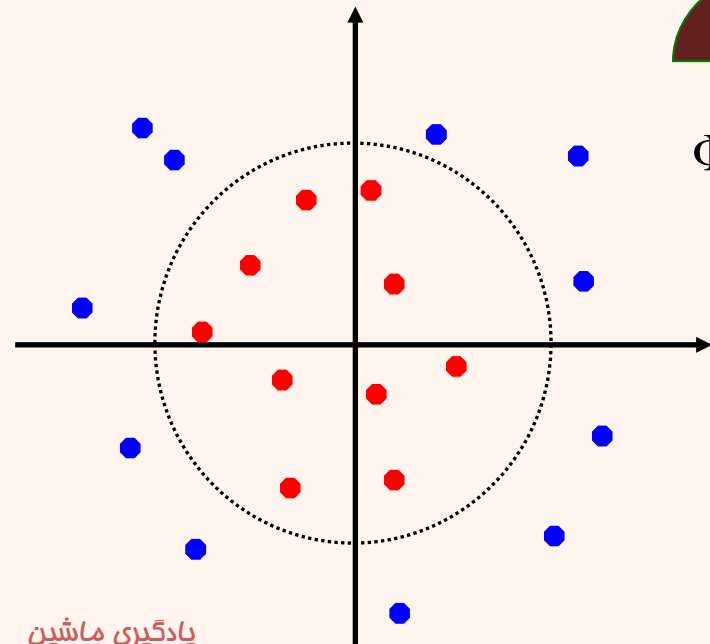


# نگاشت به فضای بالاتر

- 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.



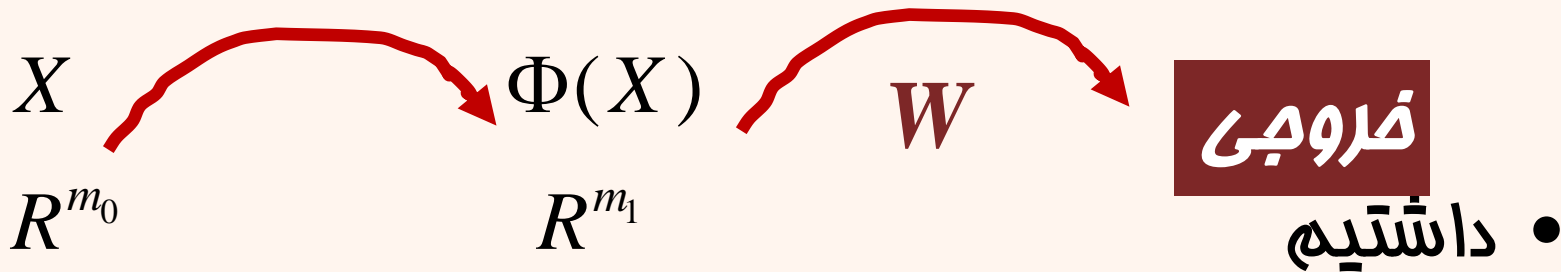
$$\Phi: X \rightarrow \phi(X)$$



دانشگاه  
تهران



# نگاشت به فضای بالاتر



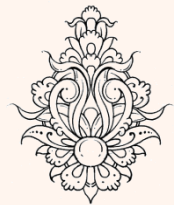
$$W^T X + b = 0$$

- 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:

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





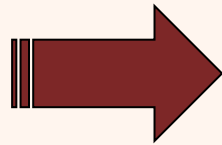
# نگاشت به فضای بالاتر

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

• با فرض  $\varphi_0(\mathbf{X}) = 1$

• خواهیم داشت:

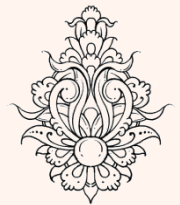
$$\sum_{j=0}^{m1} w_j \varphi_j(X) = 0$$



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

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

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






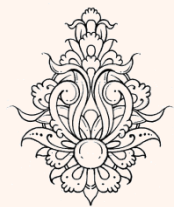
# نگاشت به فضای بالاتر

- 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:

$$d_i \left( \sum_{j=0}^{m_1} w_j \varphi_j(X_i) - 1 \right) \geq 0$$
$$W_{opt} = \sum_{i=1}^N \alpha_i \cdot d_i(\Phi(X_i))$$

  
اسکالر  $m_1 \times 1$

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





# نگاشت به فضای بالاتر

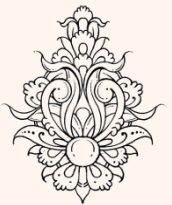
$$\sum_{i=1}^N \alpha_i \cdot d_i \Phi^T(X_i) \Phi(X) = 0$$

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

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

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

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





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

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

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

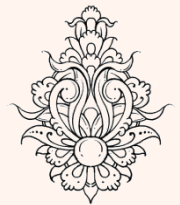
$$\Rightarrow K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}(\mathbf{x}_i)^T \boldsymbol{\phi}(\mathbf{x}_j):$$

$$\text{where } \boldsymbol{\phi}(X) = [1 \ x_1^2 \ \sqrt{2} x_1 x_2 \ x_2^2 \ \sqrt{2} x_1 \ \sqrt{2} x_2]$$

## 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 مقارن

هدف یافتن ضرایب لاگراش 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 قیود زیر

$$0 \leq \alpha \leq C$$

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.



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



# کرنل‌های معمول

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

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

***polynomial***

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

***RBF***

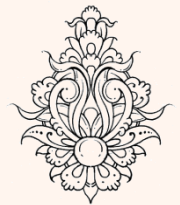
$$\exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_i\|^2\right)$$

***tanh***

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

***chi-squared kernel***

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

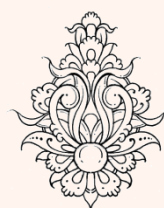
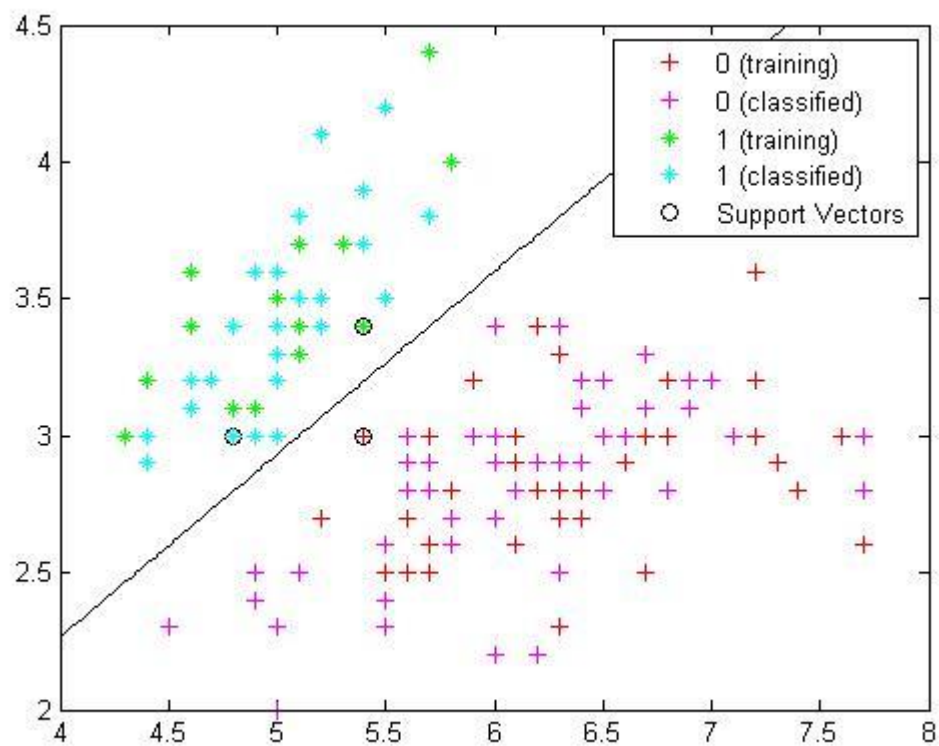




```
clear all;
close all;
load fisheriris
data = [meas(:,1), meas(:,2)];
groups = ismember(species,'setosa');
[train, test] = crossvalind('holdOut',groups);
cp = classperf(groups);
svmStruct =
svmtrain(data(train,:),groups(train),'showplot',true,'boxconstr
aint',1e6);
title(sprintf('Kernel Function: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





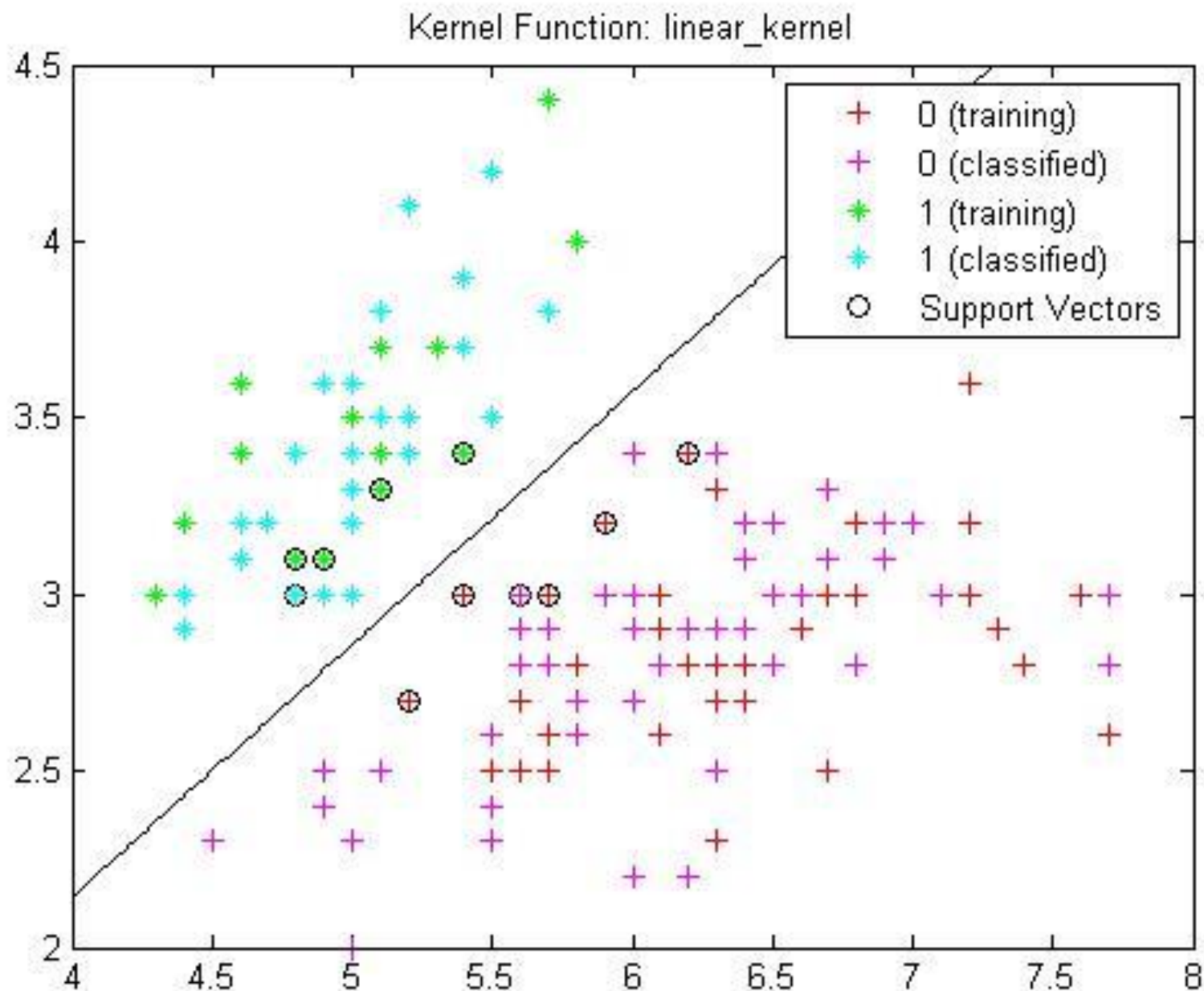




```
clear all;
close all;
load fisheriris
data = [meas(:,1), meas(:,2)];
groups = ismember(species,'setosa');
[train, test] = crossvalind('holdOut',groups);
cp = classperf(groups);
svmStruct = svmtrain(data(train,:),groups(train),'showplot',true);
title(sprintf('Kernel Function: %s',...
    func2str(svmStruct.KernelFunction)),...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







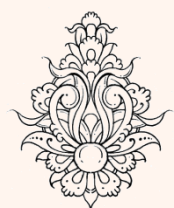
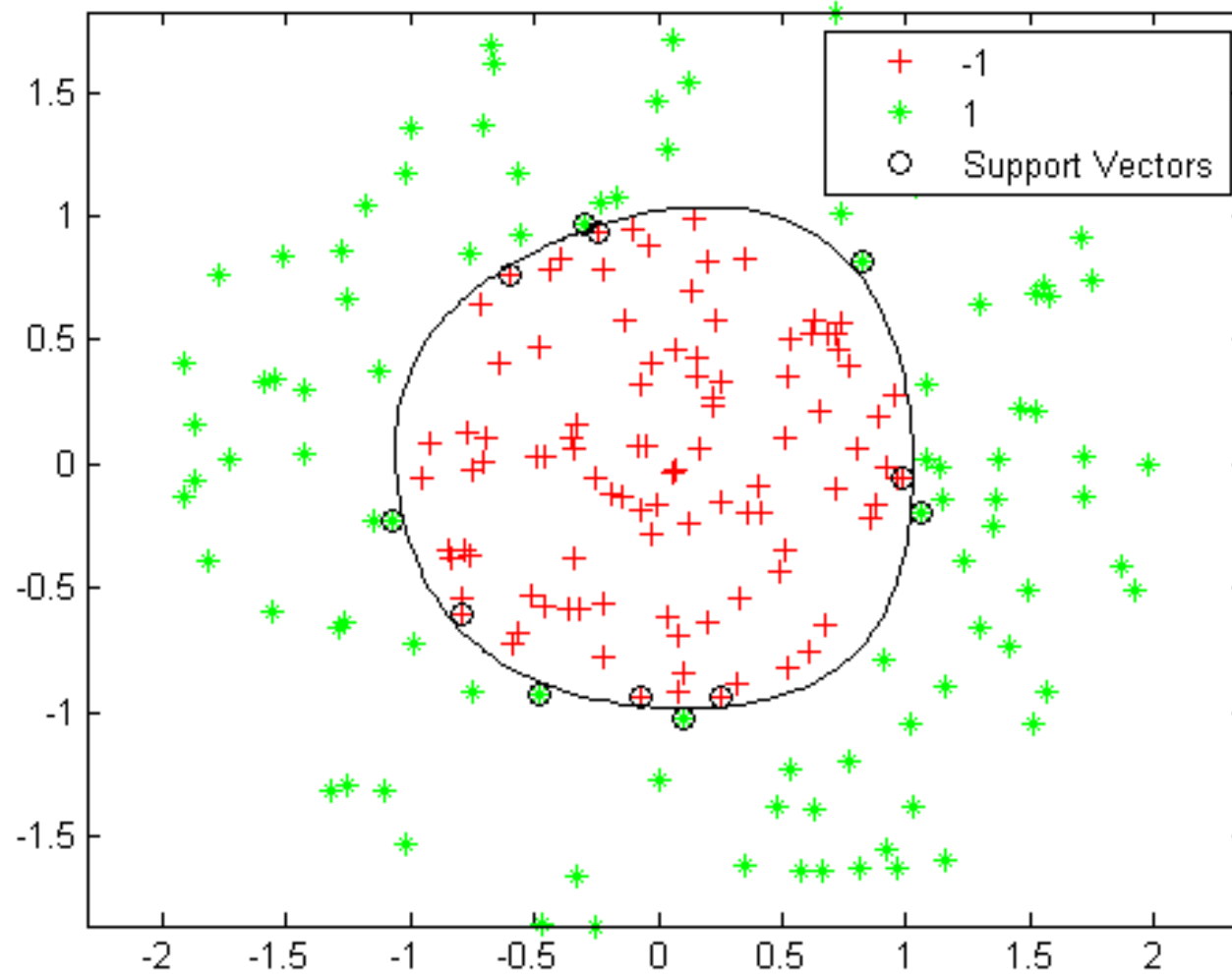




```
r = sqrt(rand(100,1)); % radius
t = 2*pi*rand(100,1); % angle
data1 = [r.*cos(t), r.*sin(t)]; % points
r2 = sqrt(3*rand(100,1)+1); % radius
t2 = 2*pi*rand(100,1); % angle
data2 = [r2.*cos(t2), r2.*sin(t2)]; % points
plot(data1(:,1),data1(:,2),'r.')
plot(data2(:,1),data2(:,2),'b.')
axis equal
data3 = [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
```









# ساخت کرنل‌های جدید

- براساس کرنل‌های موجود به سادگی می‌توان کرنل‌های جدید ساخت:

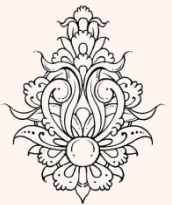
– هر تابع  $K(.,.)$  یک کرنل است چنانچه معادل ضرب داخلی بردارهای حاصل از نگاشت باشد.

جمع دو کرنل

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

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

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

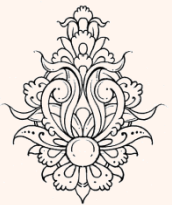




# مواجهه با مجموعه داده‌های نامتوازن

## imbalanced Data

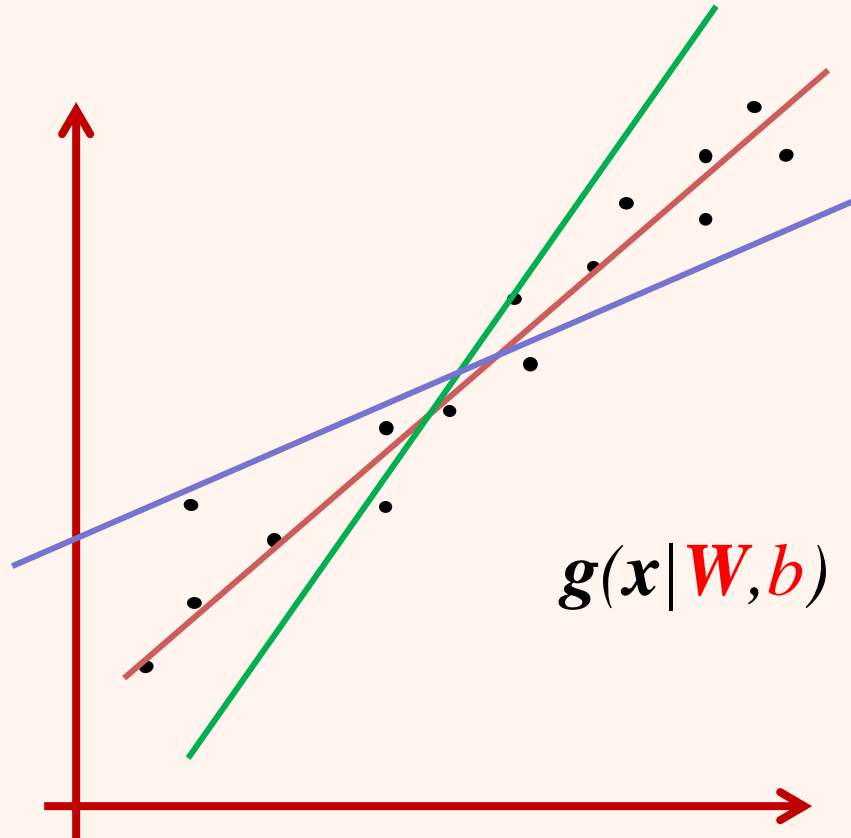
- در اکثر شیوه‌های یادگیری دسته، هنگامی که کلاس‌ها **نامتوازن** باشند، سوگیری به سمت دسته با داده بیشتر خواهیم داشت.
- ماشین بردار پشتیبان نیز از این قاعده مستثنی نیست.
- برای این که ماشین بردار پشتیبان در این شرایط خوب عمل کند، چه پیشنهادی دارید؟



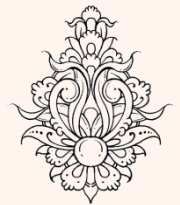


# رگرسیون خطی

کدامیک از این خطوط داده‌ها را بهتر  
مدل می‌کنند؟



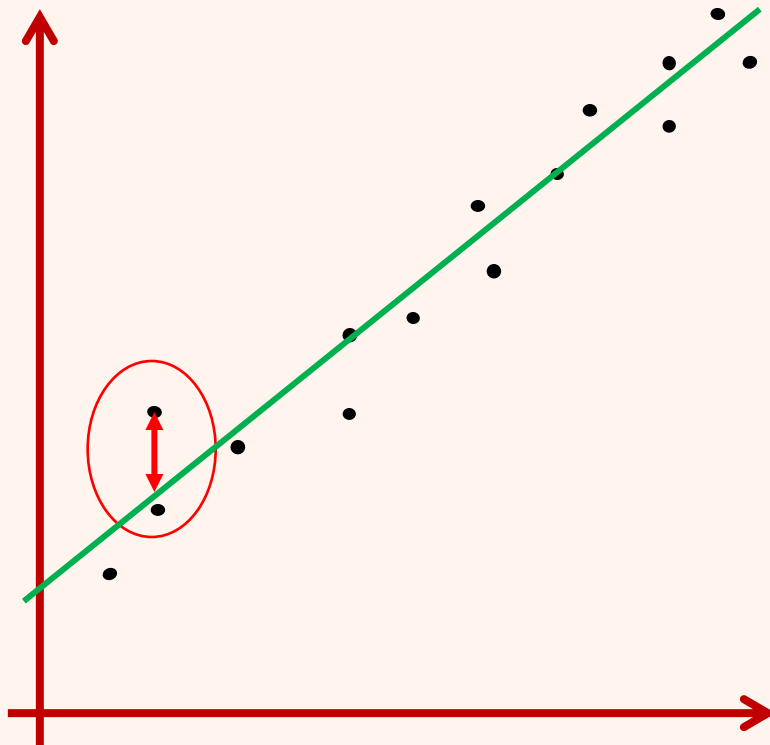
$$g(x|W, b) = W^T x + b$$





# تابع خطا

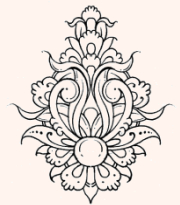
چه تابعی برای محاسبه  
خطا مناسب است؟



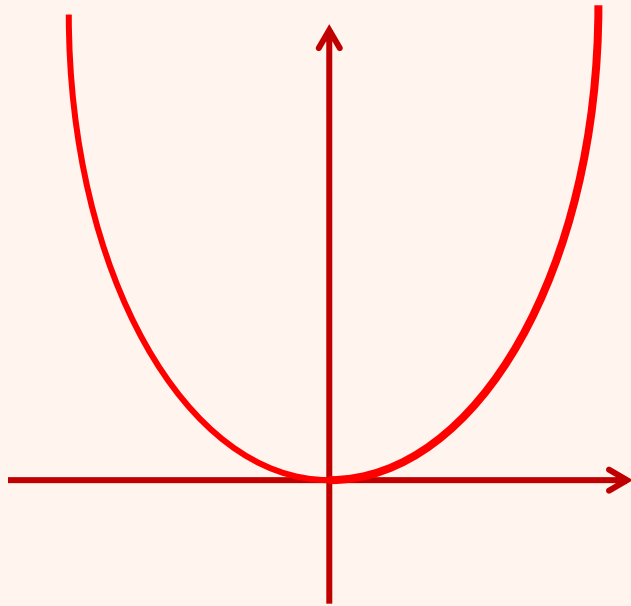
$$err_i = (W^T X_i + b - r_i)^2$$

$$err_i = |W^T X_i + b - r_i|$$

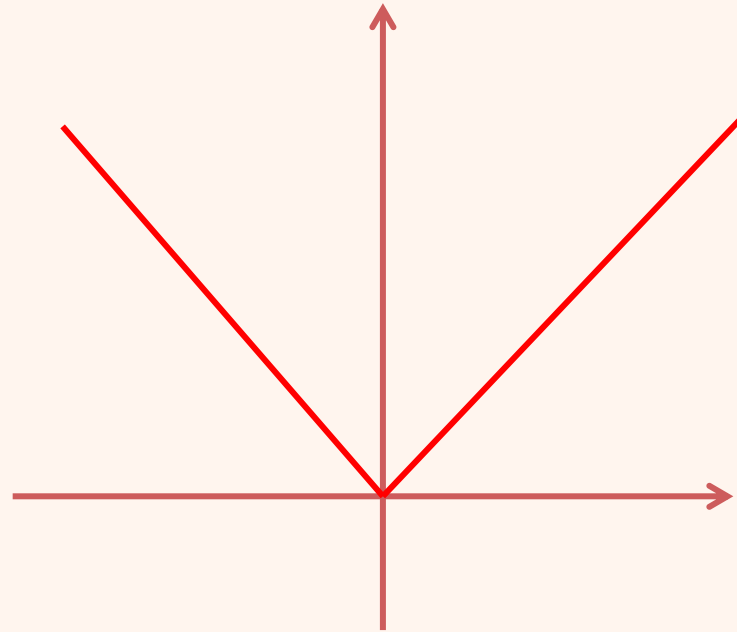
کدامیک نسبت به  
داده‌های دورافتاده  
مقاوم‌تر هستند؟





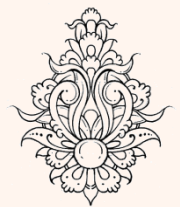


$$err_i = (W^T X_i + b - r_i)^2$$



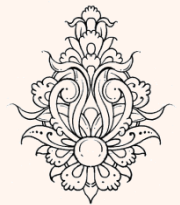
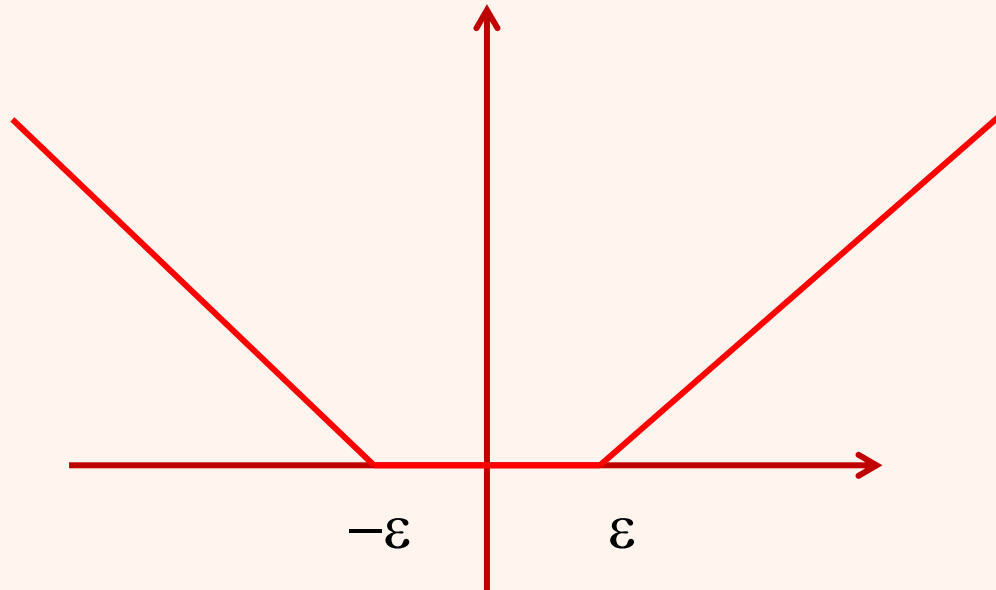
$$err_i = |W^T X_i + b - r_i|$$

برای این که نسبت به بیش‌برازش حساسیت کم‌تری داشته باشیم، پیشنهادی دارید؟





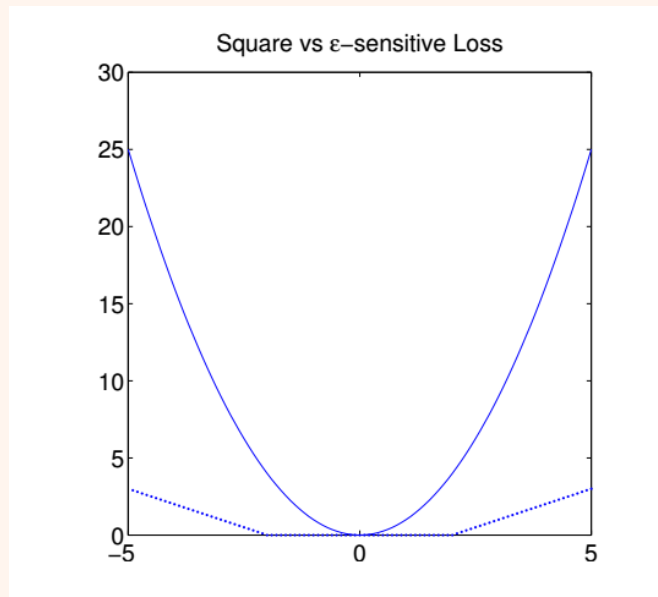
# $\epsilon$ -insensitive loss function



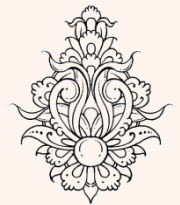


## $\epsilon$ -sensitive(hing) Loss

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



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

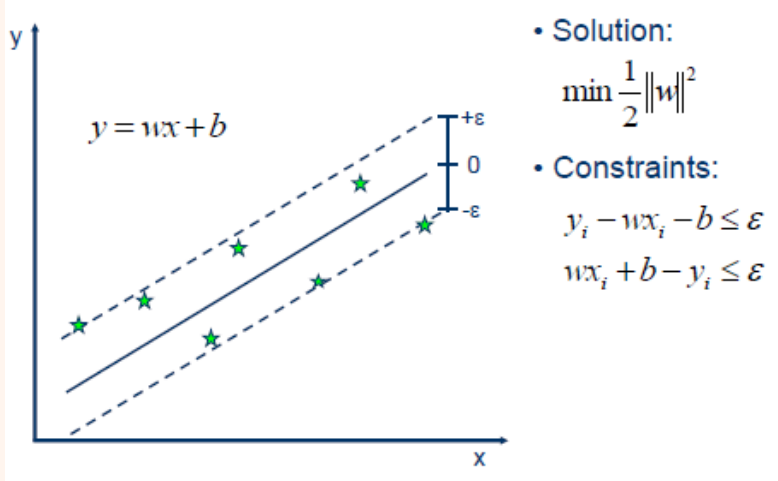




# Support Vector regression(SVR)

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

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



[http://www.saedsayad.com/support\\_vector\\_machine\\_reg.htm](http://www.saedsayad.com/support_vector_machine_reg.htm)

