شبکه‌های عصبی مصنوعی
۱۳-۱۱-۷۱۳-۰۱
بخش چهارم

SVM

دانشگاه شهید بهشتی
دانشکده‌ی مهندسی برق و کامپیوتر
زمستان ۱۳۹۳
احمد محمودی ازناوه

http://faculties.sbu.ac.ir/~a_mahmoudi/

فهرست مطالب

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.

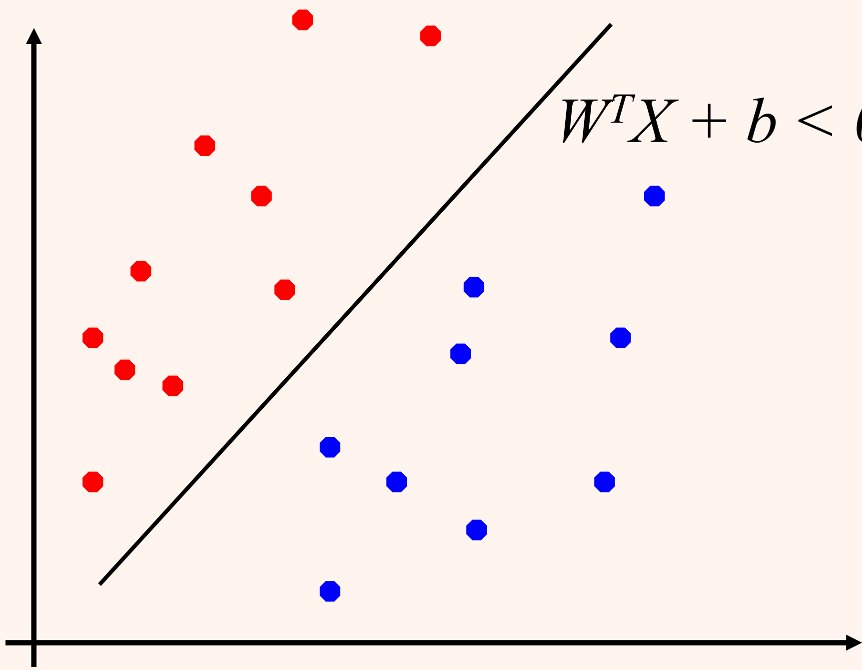
معرفی

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

$$W^TX + B > 0$$

$$W^TX + b = 0$$

$$W^TX + b < 0$$

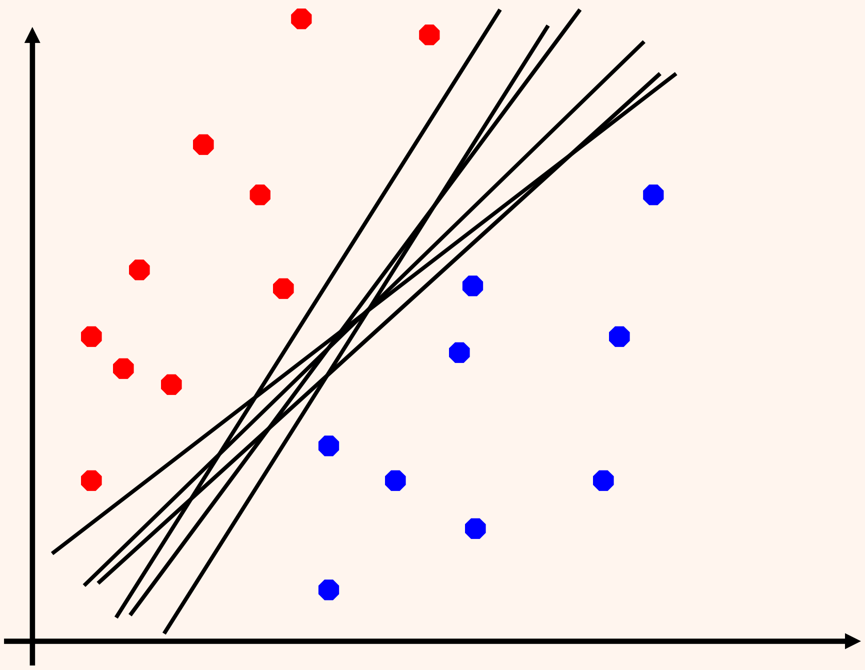


$$F(X) = SIGN(W^TX + b)$$

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

مرز بهینه

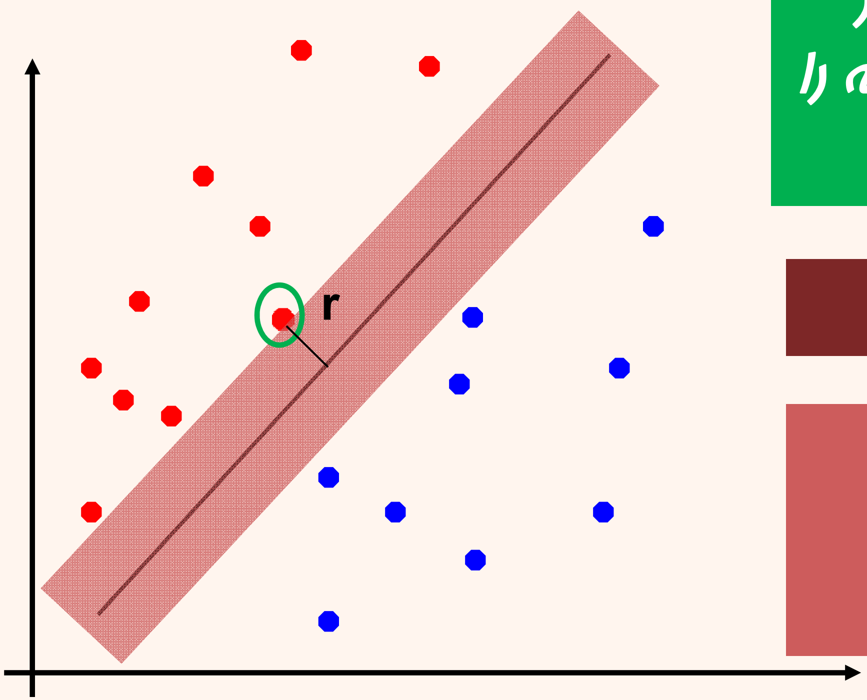
- سوال
  - کدام یک از مرزها، مرزی بهینه برای جداسازی است؟



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.

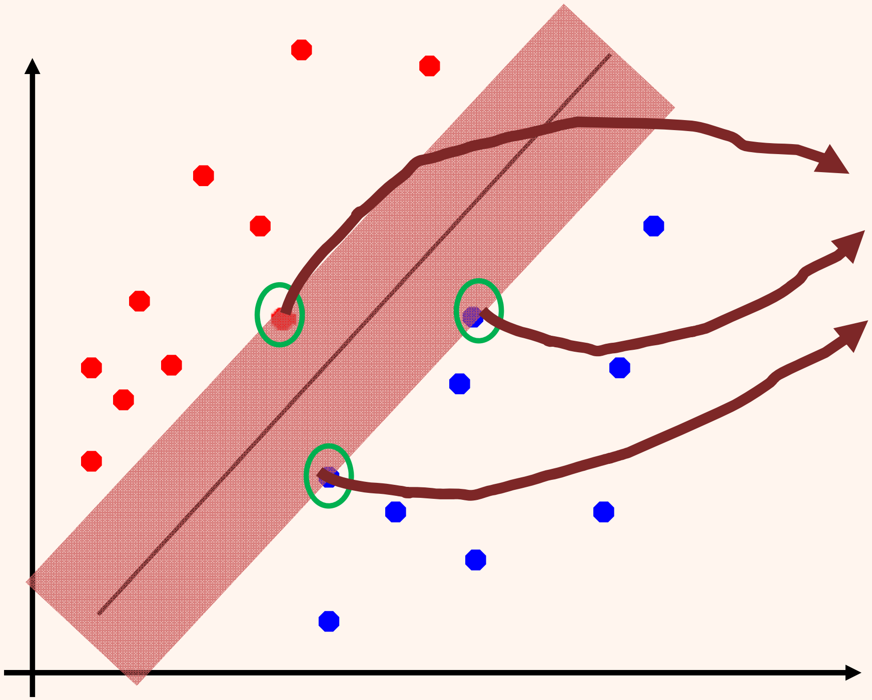
حاشیه‌ی ماکزیمم

- 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.



Optimal hyperplane

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 W_{op} و b_{op} مشخص شود.
- فرض: فرض کنیم نزدیک‌ترین نقطه به مرز جداسازی را در نظر گرفته، فاصله را «**r**» بنامیم.

مرز جداسازی (ادامه...)

- هدف
  - ماکزیمم نمودن فاصله یا همان $\rho=2\mathbf{r}$ است.
  - برای نقاط روی مرز جداسازی بهینه داریم:

$$W_{op}^{T}X + b_{op} = 0$$

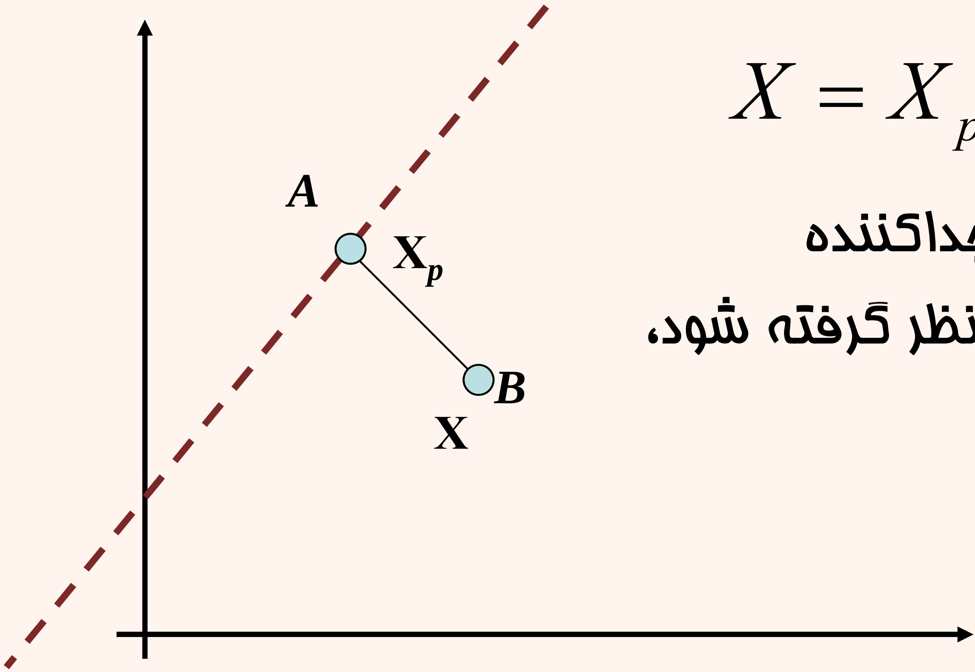
  - برای نقاط خارج از مرز داریم:

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

  - $g(X)$ می‌تواند مثبت یا منفی باشد.

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:

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

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

مرز جداسازی (ادامه...)

- داشتیم:

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

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

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

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

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

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

مرز جداسازی (ادامه...)

هدف ماکزیمم نمودن r است.

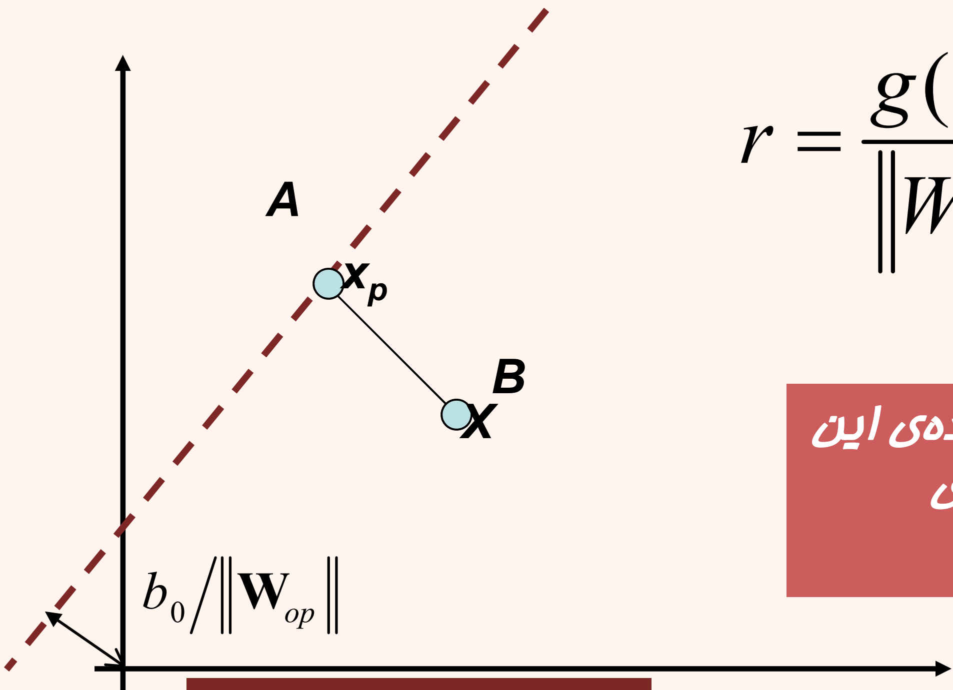
$$\boxed{g(X) = r\left\|W_{op}\right\|}$$

در این حالت تحت شرایطی می‌باید W کمینه گردد.

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

X=0

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



فاصله از مبدا مختصات

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

مرز جداسازی (ادامه...)

مسأله 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:

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

- 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...)

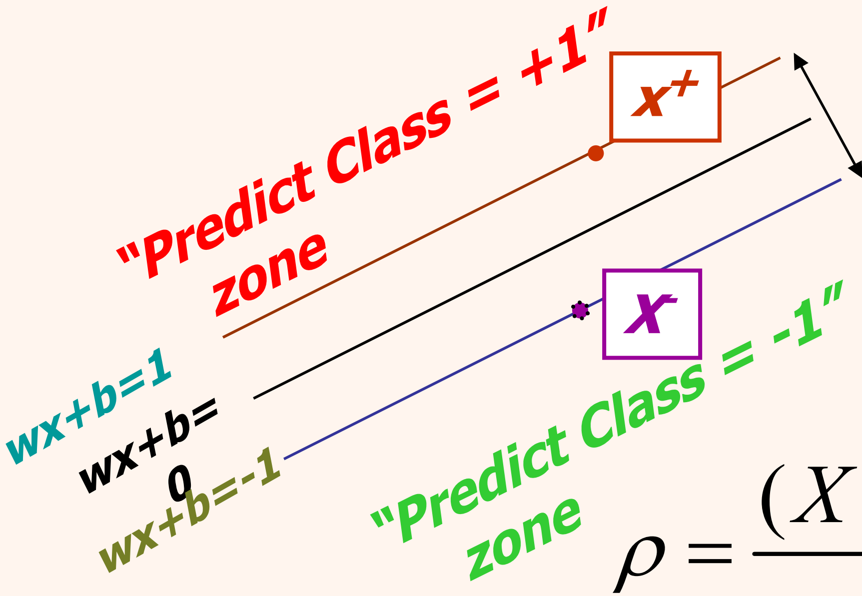
$$g(X_s) = W_{op}{}^T X^s + b_{op} = \pm 1$$

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

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

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

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

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

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

- با توجه به دو رابطه‌ی
- 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 $d_i(W_{op}{}^T X + b_{op}) \geq 1$ برای N الگوی آموزشی شرط زیر می‌باید برقرار باشد:

$$\sum_{i=1}^{N} [d_i(W_{op}{}^T X + b_{op}) - 1]$$

این میزان بزرگ‌تر یا مساوی صفر است

شبکه عصبی

خلاصه

وزن‌ها و بایاس را به گونه‌ای بیابید که:

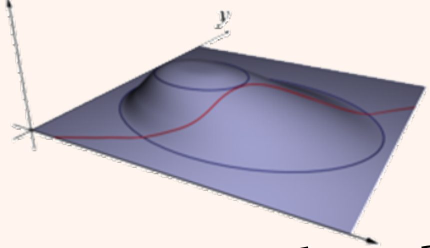
$$\rho = \frac{2}{\|W\|} \text{ is maximized}$$

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:

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

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



یافتن رویه‌ی بهینه

- رابطه‌ی لاگرانژ زیر تعریف می‌شود به گونه‌ای که هر دو قید ذکر شده را پوشش دهد:

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

Lagrange multiplier(nonnegative)

- برای به‌دست آوردن W_{op} و b_{op} نسبت به هر دو مشتق می‌گیریم.

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

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

b_{op} به‌دست نمی‌آید

ولی یک قید می‌دهد

یافتن رویه‌ی بهینه

- دو شرط برای α خواهیم داشت:

$$\begin{cases} \sum_{i=1}^{N} \alpha_i d_i = 0 \\ \alpha_i [d_i (W^T X_i + b) - 1] = 0 \end{cases}$$

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

صفر

غیر صفر

- به ازای هر α_i برای الگوهای آموزشی متناظر با SVها رابطه‌ی زیر برقرار است.

$$d_i (W_{op}^T X_i + b_{op}) - 1 = 0$$

یافتن رویه‌ی بهینه

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

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

- برای مقادیر بهینه داشتیم:

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

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

Duality theorem

- پس خواهیم داشت:

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

Dual Problem

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

$$\begin{cases} = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} W_{op}^T W_{op} \quad = Q(\alpha) \\ \sum_{i=1}^{N} \alpha_i d_i = 0 \end{cases}$$

مینیمم نمودن W همانند ماکزیمم نمودن Q است زیرا در W_{op} مقدار W کم‌ترین میزان است و در این صورت است که کل عبارت ماکزیمم می‌شود.

یافتن رویه‌ی بهینه

$$Q(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}W_{op}^T W_{op}$$

$$= \sum_{i=1}^{N}\alpha_i - \frac{1}{2}[\sum_{i=1}^{N}\alpha_i d_i X_i]^T[\sum_{j=1}^{N}\alpha_j d_j X_j]$$

$$= \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i d_i \alpha_j d_j X_i^T X_j$$

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

α_i ها وابسته به الگوهای X_i , X_j و خروجی های مرتبط است

یافتن رویه‌ی بهینه

$$= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} [\sum_{i=1}^{N} \alpha_i d_i X_i]^T [\sum_{j=1}^{N} \alpha_j d_j X_j]$$

جهت محاسبه‌ی αها می‌توان نسبت به α_k مشتق گرفته برابر با صفر قرار داد.

$$\frac{\partial Q(\alpha)}{\partial \alpha_k} = 1 - \sum_{\substack{i=1 \\ i \neq k}}^{N} \alpha_i d_i d_k X_i^T X_k - \alpha_k d_k^2 X_k^T X_k = 0$$

N معادله و N مجهول

$$M_{i,j} = X_i^T X_j$$

ضرب داخلی

$$\frac{\partial Q(\alpha)}{\partial \alpha_k} = 1 - d_k \sum_{i=1}^{N} \alpha_i d_i M_{i,k} = 0$$

یافتن رویه‌ی بهینه

$$\frac{\partial Q(\alpha)}{\partial \alpha_k} = 1 - d_k \sum_{i=1}^{N} \alpha_i d_i M_{i,k} = 0$$

- پس از به دست آوردن α خواهیم داشت:

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

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

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

یافتن رویه‌ی بهینه

$W = \Sigma \alpha_i d_i X_i$ $b = d_k - W^T X_k$ 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:

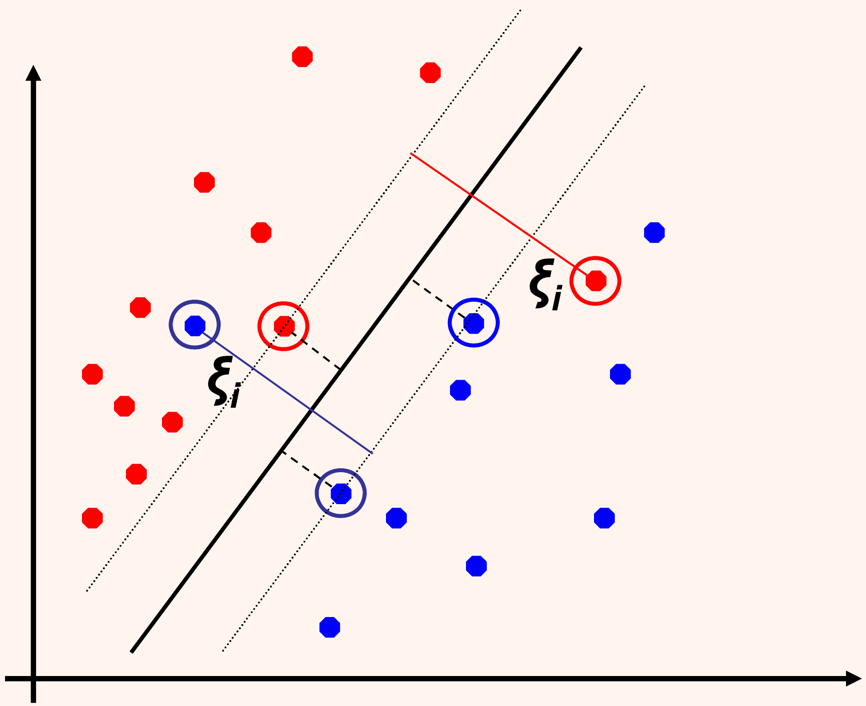
$$g(X) = \Sigma \alpha_i d_i \underbrace{X_i^T X}_{\text{ضرب داخلی دو بردار}} + b$$

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 را نداشته باشند، چه خواهد شد؟ به بیان بهتر صحبت در مورد مسائل جدایی‌پذیر است که با نویز همراه هستند.

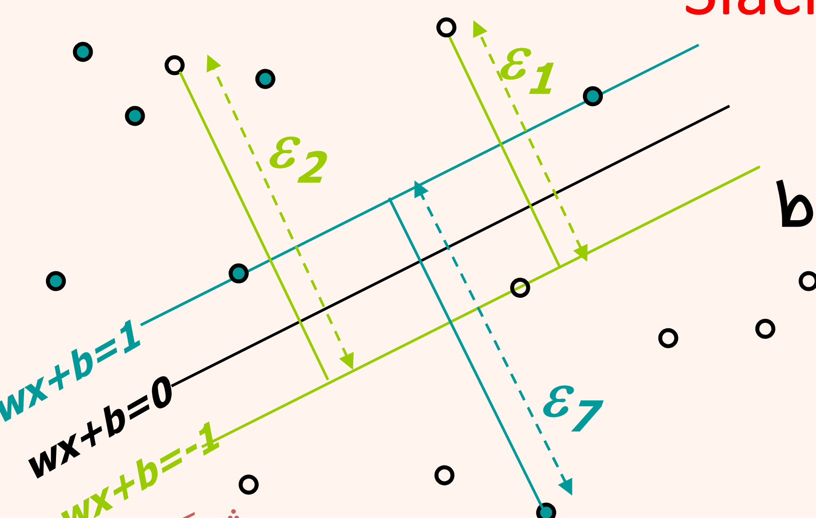


Soft Margin

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

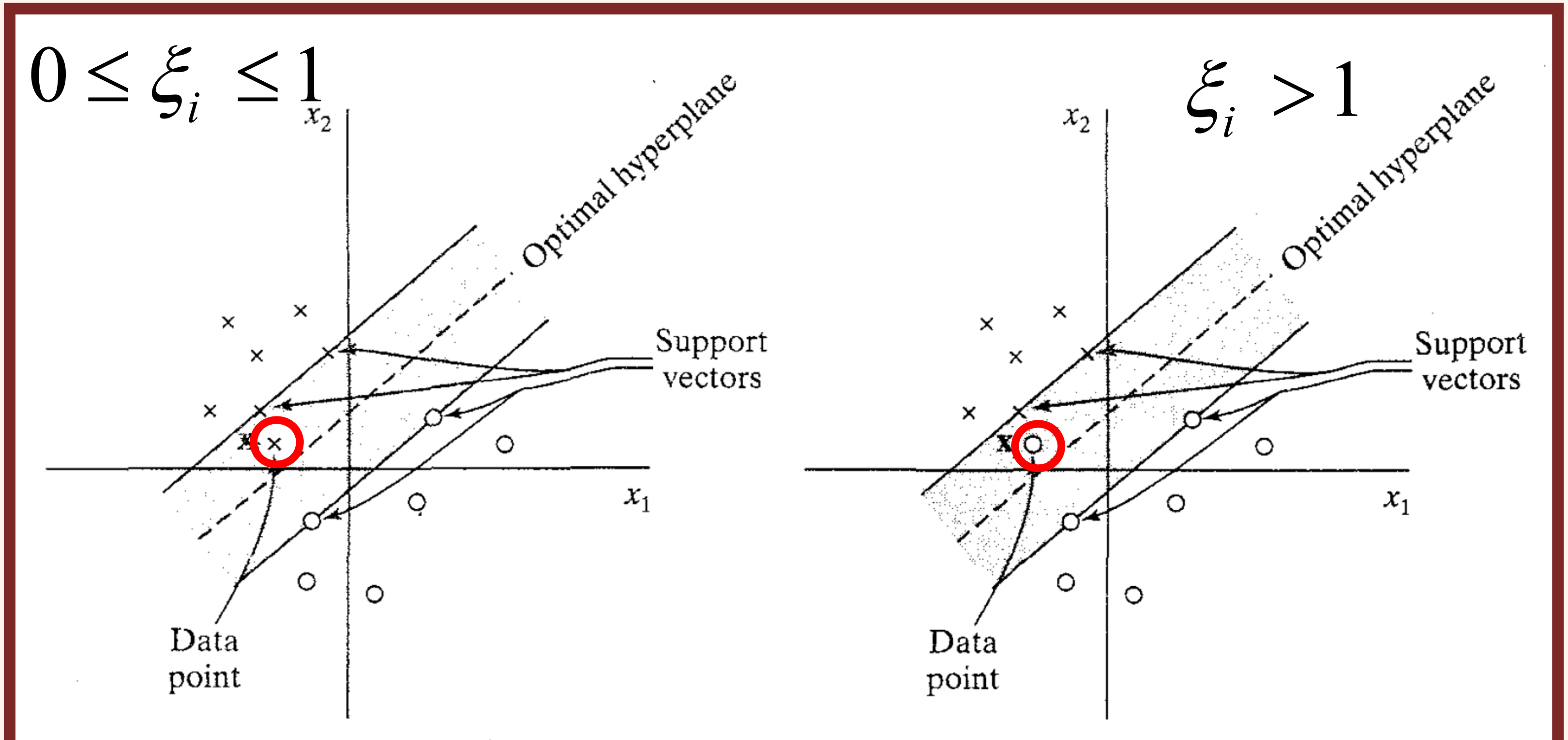
$$d_i(W_{op}{}^T X + b_{op}) \geq 1$$

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



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

Soft Margin Classification



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

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

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 $\xi>0$
  - در صورتی که داده‌های نویزی از مجموعه خارج شود، رویه‌ی جداکننده تغییر خواهد کرد.
- هدف یافتن «رویه‌ای جداکننده» است که در آن خطای طبقه‌بندی 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 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.

Soft Margin

- برای Hard Margin داشتیم:

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

- 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.

Soft Margin Classification

در نهایت ضرایب لاگرانژ از عبارت زیر محاسبه خواهند شد:

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

با در نظر گرفتن قیود زیر

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

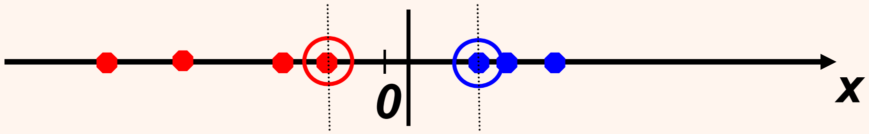
$$0\le\alpha\le C$$

بقیه مراحل مانند حالت قبل خواهد بود:

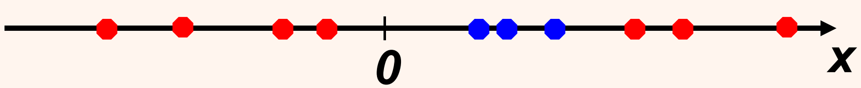
$$W_{op}=\sum_{i=1}^{N_s}\alpha_i d_i X_i$$

SVM غیرخطی

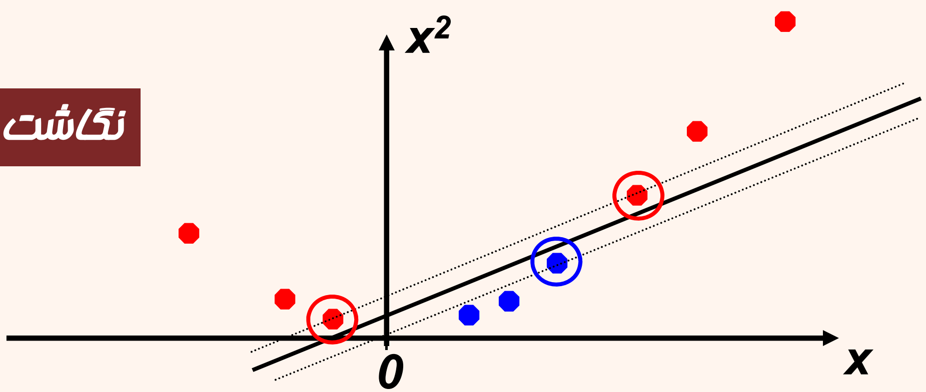
- برای داده‌هایی که قابلیت جداسازی خطی دارند، عملکرد سیستم بسیار خوب است.



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

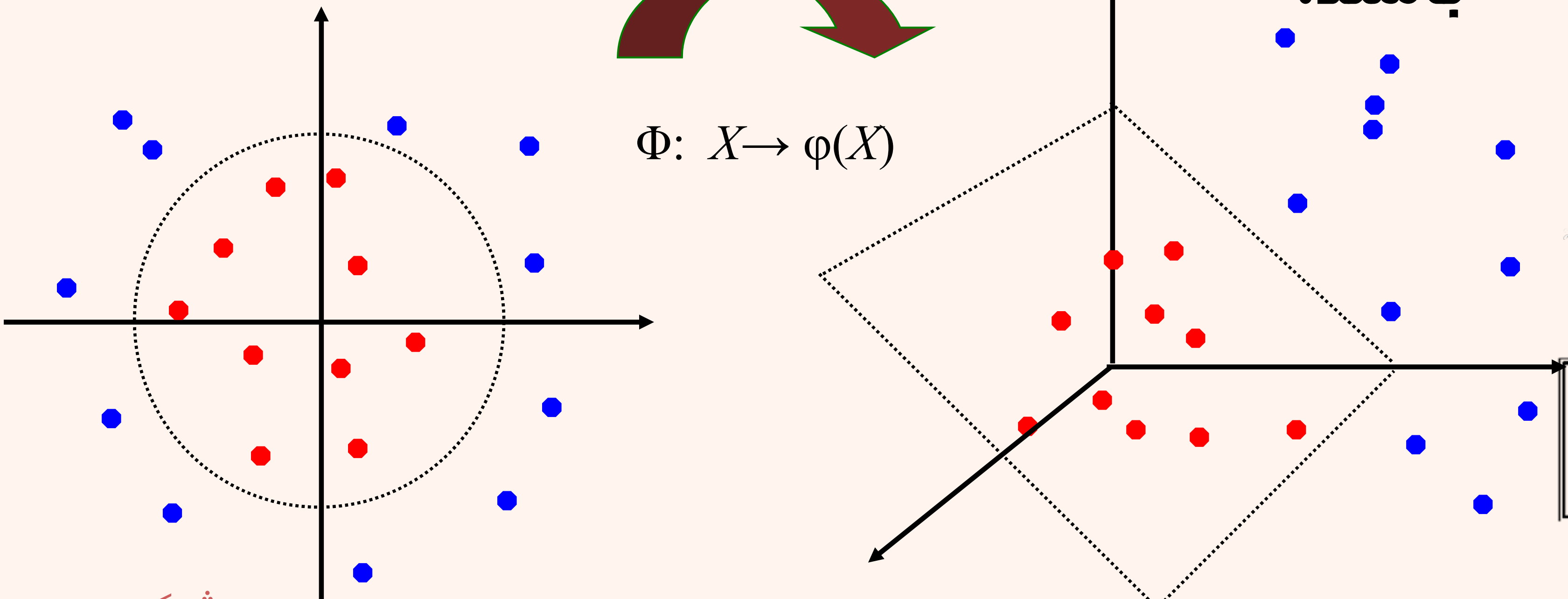


نگاشت به یک فضای High Dimension



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

$X_i \rightarrow \Phi(X_i) \xrightarrow{W}$ **خروجی**

R^{m0} R^{m1}

- داشتیم

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

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

$$\Phi(X) = [\varphi_1(X), \varphi_2(X), \ldots\ldots\ldots, \varphi_{m1}(X)]^T$$

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

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

نگاشت به فضای بالاتر

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

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

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

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

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

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

نگاشت به فضای بالاتر

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

$$d_i[\sum_{i=1}^{m1} w_j \varphi_j(X_i) - 1] \geq 0$$

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

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.

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

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

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

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

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

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

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

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

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

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.

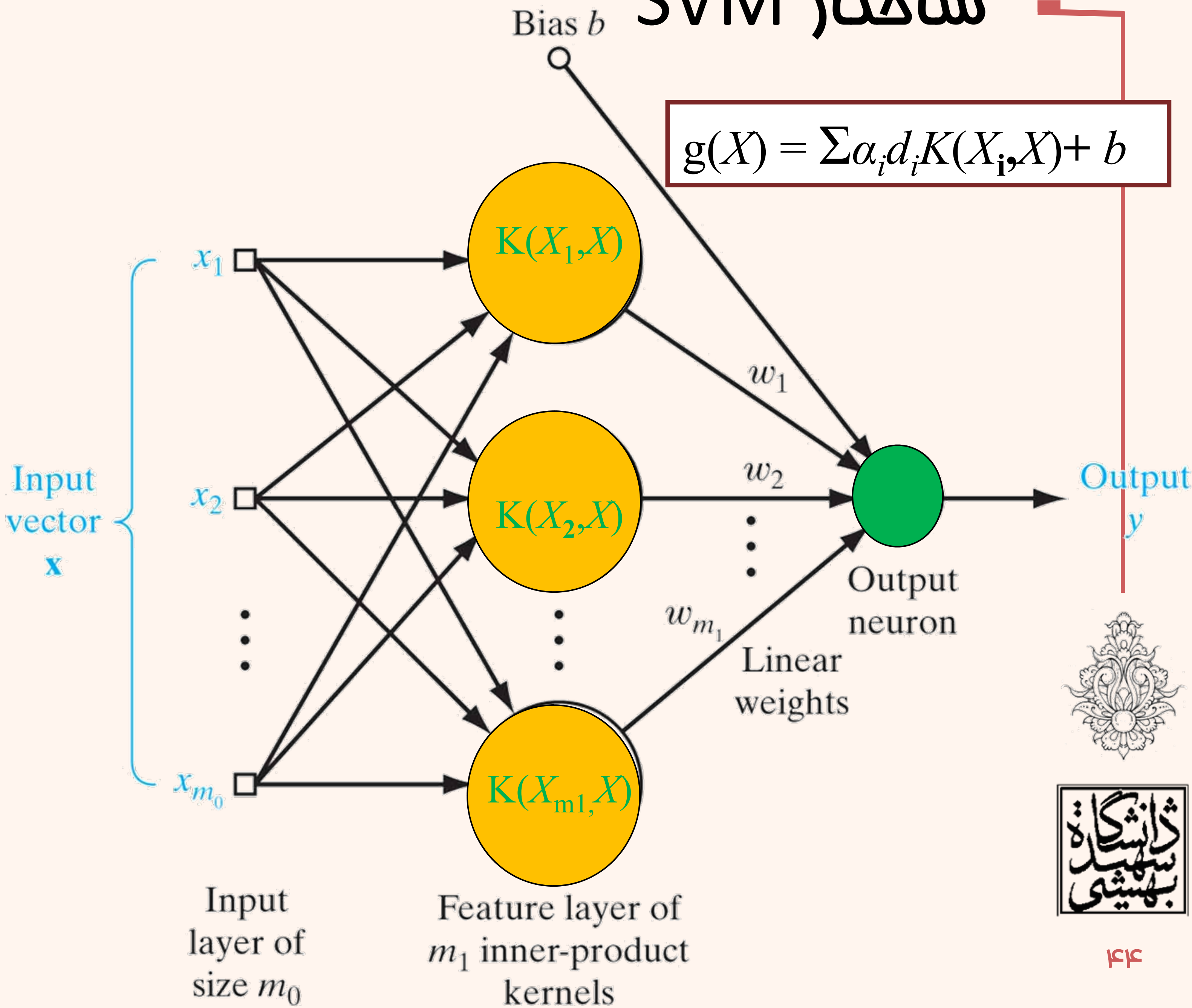
$$\boxed{g(X)=\sum\alpha_i d_i K(X_i,X)+b}$$

توابع نگاشت

TABLE 6.1 Summary of Mercer Kernels		
Type of support vector machine	Mercer kernel $k(\mathbf{x}, \mathbf{x}_i), i = 1, 2, ..., N$	Comments
Polynomial learning machine	$(\mathbf{x}^T\mathbf{x}_i + 1)^p$	Power p is specified *a priori* by the user
Radial-basis-function network	$\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x} - \mathbf{x}_i\|^2\right)$	The width σ^2, common to all the kernels, is specified *a priori* by the user
Two-layer perceptron	$\tanh(\beta_0\mathbf{x}^T\mathbf{x}_i + \beta_1)$	Mercer's theorem is satisfied only for some values of β_0 and β_1

ساختار SVM

$$g(X) = \Sigma \alpha_i d_i K(X_{\mathbf{i}},X)+ b$$



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 مسأله به ابعاد داده‌ها بستگی ندارد.

مثال XOR Problem

$$X_1 = [-1\ \text{-}1] \rightarrow \mathrm{d}_1 = -1$$

$$X_2 = [-1\ 1] \rightarrow \mathrm{d}_2 = +1$$

$$X_3 = [1\ \text{-}1] \rightarrow \mathrm{d}_3 = +1$$

$$X_4 = [1\ 1] \rightarrow \mathrm{d}_4 = -1$$

$$N = 4$$

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

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

- نمونه‌های آموزشی دو بعدی هستند.

XOR Problem

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

$$X = [x_1 \;\; x_2]$$

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

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

$$= 1 + x_1^2 x_{i1}^2 + 2x_1 x_2 x_{i1} x_{i2} + x_{i2}^2 x_2^2 + 2x_1 x_{i1} + 2x_2 x_{i2}$$

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

XOR Problem

$$= 1 + x_1^2 x_{i1}^2 + 2x_1 x_2 x_{i1} x_{i2} + x_{i2}^2 x_2^2 + 2x_1 x_{i1} + 2x_2 x_{i2}$$

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

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

XOR Problem

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

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

$$\begin{matrix} X_1 = [-1\ \text{-}1] \\ X_2 = [-1\ 1] \\ X_3 = [1\ \text{-}1] \\ X_4 = [1\ 1] \end{matrix} \begin{bmatrix} 1 & 1 & \sqrt{2} & 1 & -\sqrt{2} & -\sqrt{2} \\ 1 & 1 & -\sqrt{2} & 1 & -\sqrt{2} & \sqrt{2} \\ 1 & 1 & -\sqrt{2} & 1 & \sqrt{2} & -\sqrt{2} \\ 1 & 1 & \sqrt{2} & 1 & \sqrt{2} & \sqrt{2} \end{bmatrix}$$

XOR Problem

$$K_{4\times4}=\begin{bmatrix}1&1&\sqrt{2}&1&-\sqrt{2}&-\sqrt{2}\\1&1&-\sqrt{2}&1&-\sqrt{2}&\sqrt{2}\\1&1&-\sqrt{2}&1&\sqrt{2}&-\sqrt{2}\\1&1&\sqrt{2}&1&\sqrt{2}&\sqrt{2}\end{bmatrix}\begin{bmatrix}1&1&1&1\\1&1&1&1\\\sqrt{2}&-\sqrt{2}&-\sqrt{2}&\sqrt{2}\\1&1&1&1\\-\sqrt{2}&-\sqrt{2}&\sqrt{2}&\sqrt{2}\\-\sqrt{2}&\sqrt{2}&-\sqrt{2}&\sqrt{2}\end{bmatrix}$$

$$K_{4\times4}=\begin{bmatrix}9&1&1&1\\1&9&1&1\\1&1&9&1\\1&1&1&9\end{bmatrix}$$

XOR Problem

$$N = 4$$

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

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

- برای به دست آوردن α ها مشتق گرفته برابر با صفر قرار می‌دهیم:

$$\frac{\partial \mathrm{Q}(\alpha)}{\partial \alpha_1} = 0 \Rightarrow 1\text{-}9\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 0$$

XOR Problem

$$\frac{\partial Q(\alpha)}{\partial \alpha_1}=0 \Rightarrow 1\text{-}9\alpha_1+\alpha_2+\alpha_3+\alpha_4=0$$

$$\frac{\partial Q(\alpha)}{\partial \alpha_2}=0 \Rightarrow 1+\alpha_1-9\alpha_2-\alpha_3+\alpha_4=0$$

$$\frac{\partial Q(\alpha)}{\partial \alpha_1}=0 \Rightarrow 1+\alpha_1-\alpha_2-9\alpha_3-\alpha_4=0$$

$$\frac{\partial Q(\alpha)}{\partial \alpha_4}=0 \Rightarrow 1\text{-}\alpha_1+\alpha_2+\alpha_3-9\alpha_4=0$$

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

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

- بنابراین هر چهار ورودی، بردار پشتیبان هستند.
- پس از محاسبه‌ی α ها $\mathbf{W}_{opt}$ را محاسبه می‌کنیم:

XOR Problem

- جهت محاسبه‌ی اندازه‌ی وزن بهینه داریم:

$$\frac{1}{2}\left\|\mathbf{W}_{\text{opt}}\right\|^2 = \frac{1}{4} \Rightarrow \left\|\mathbf{W}_{\text{opt}}\right\| = \frac{1}{\sqrt{2}}$$

- داشتیم:

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

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

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

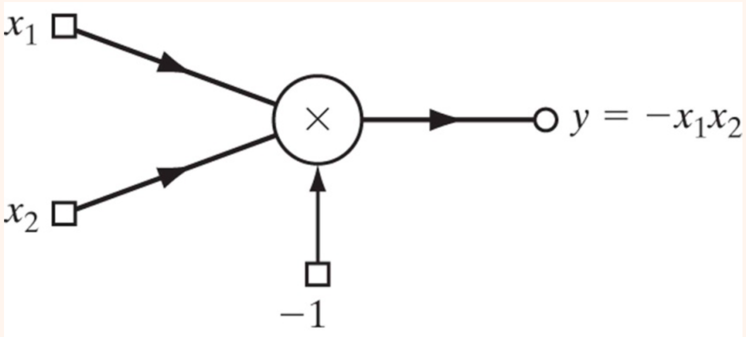
XOR Problem

- رویه‌ی بهینه به وسیله‌ی رابطه‌ی زیر محاسبه می‌شود:

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

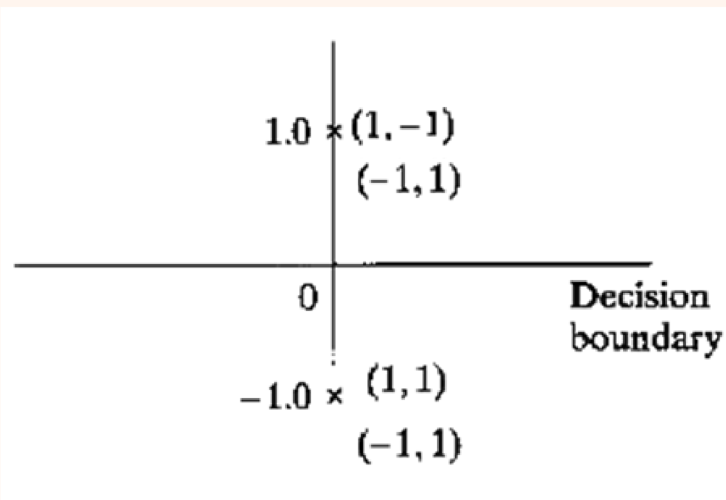
$$\left[0, 0, \frac{-1}{\sqrt{2}}, 0, 0, 0\right]\begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix} = 0 \quad \Rightarrow \quad -x_1x_2 = 0$$

XOR Problem



(a)

(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.



XOR Problem

```
X=[-1 -1 1 1;
    -1 1 -1 1];
d=[-1 1 1 -1];
n=4;
K=zeros(n,n);
for i=1:n
    for j=1:n
        xi1=X(1,i);
        xi2=X(2,i);
        xj1=X(1,j);
        xj2=X(2,j);
        fi=[1 xi1^2   xi2^2
sqrt(2)*xi1*xi2  sqrt(2)*xi1
sqrt(2)*xi2 ];
        fj=[1 xj1^2 xj2^2
sqrt(2)*xj1*xj2  sqrt(2)*xj1
sqrt(2)*xj2 ];

K(i,j)=fi*fj';%Kernel
    end
end
```

```
KD=zeros(n,n);
for i=1:n
    for j=1:n

KD(i,j)=K(i,j)*d(i)*d(j);
    end
end

alfa=inv(KD)*ones(n,1);
```

- می‌توان گفت برای به دست آوردن α

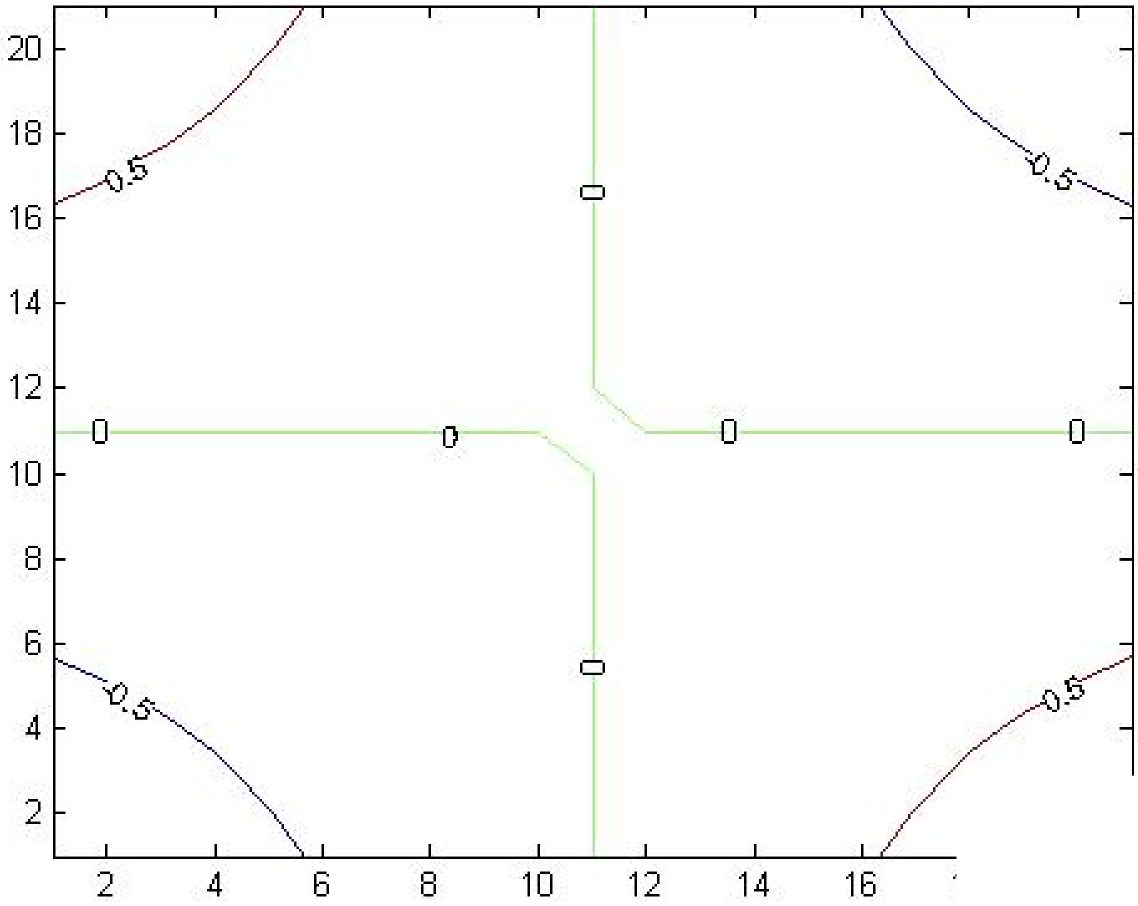
$$Q(\alpha)=\sum_{i=1}^{m1}\alpha_i-\frac{1}{2}\sum_{i=1}^{m1}\sum_{j=1}^{m1}\alpha_i\alpha_j d_i d_j K(X_i,X_j)$$

$$\frac{\partial Q(\alpha)}{\partial \alpha_j}=1-\sum_{i=1}^{m1}\alpha_i \underbrace{d_i d_j K(X_i,X_j)}_{K'(i,j)}=0 \quad j=1,2,\ldots,m_1$$

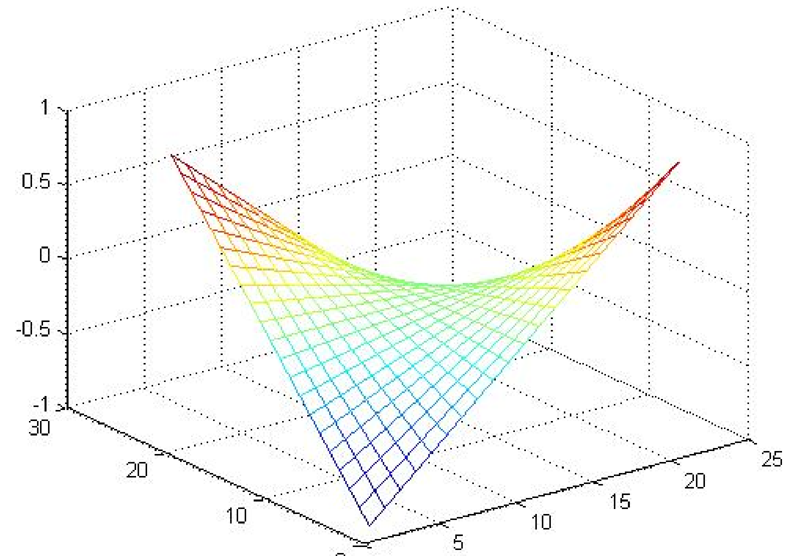
$$\boldsymbol{\alpha}=K'^{-1}.\begin{bmatrix}1\\ \cdot\\ \cdot\\ 1\end{bmatrix}$$

```
f=zeros(4,6);
for i=1:n
    xi1=X(1,i);
    xi2=X(2,i);
    f(i,:)=alfa(i)*d(i)*[1 xi1^2 xi2^2
sqrt(2)*xi1*xi2 sqrt(2)*xi1 sqrt(2)*xi2 ];
end
W=sum(f);

O=zeros(1,4);
for i=1:n
    xi1=X(1,i);
    xi2=X(2,i);
    O(i)=W*[1 xi1^2 xi2^2 sqrt(2)*xi1*xi2
sqrt(2)*xi1 sqrt(2)*xi2 ]';
end
```

```
x1=-1:0.1:1;
x2=-1:0.1:1;
for i=1:21
    for j=1:21
        xi1=x1(i);
        xi2=x2(j);
        M(i,j)=W*[1 xi1^2 xi2^2
sqrt(2)*xi1*xi2 sqrt(2)*xi1 sqrt(2)*xi2 ]';
    end
end

mesh(M);
figure;
[c,h]=contour(M,[-1 -0.5 0 0.5 1]);
clabel(c,h);
```



20
18
16
14
12
10
8
6
4
2
2
4
6
8
10
12
14
16
-0.5
-0.5
0.5
0.5
0
0
0
0
0
0

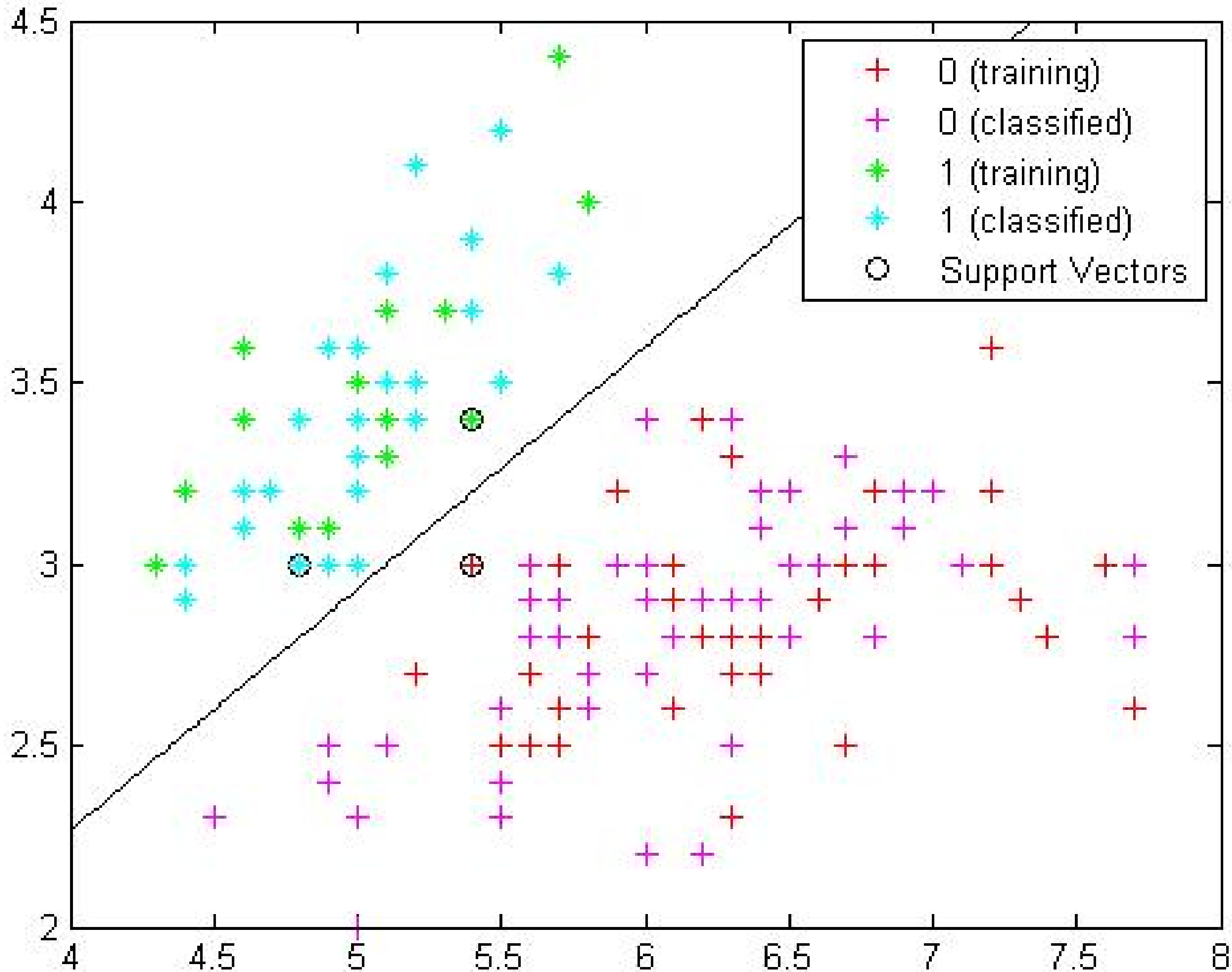


1
0.5
0
-0.5
-1
30
20
10
5
10
15
20
25

شبکه عصبی

مثال

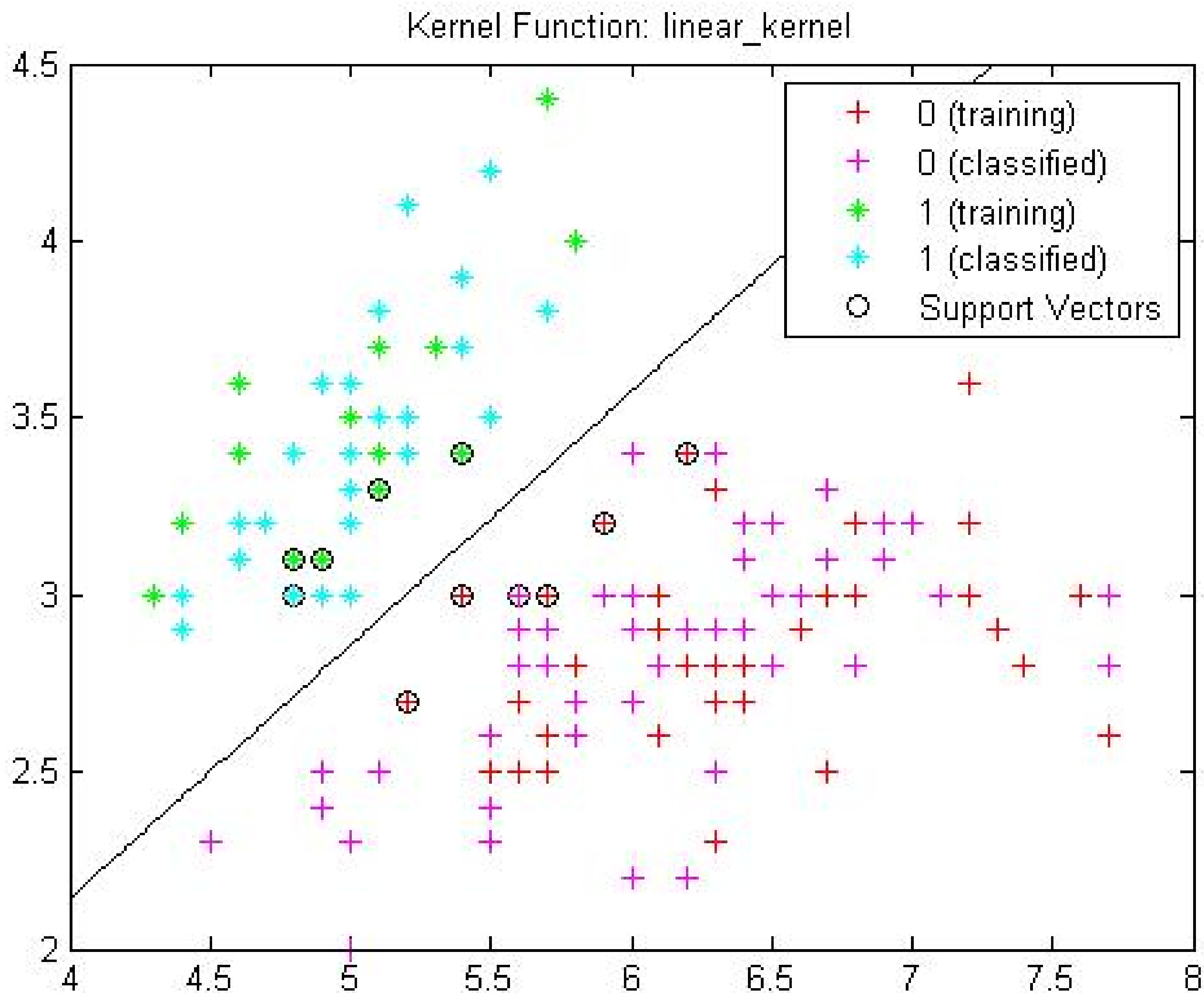
```
clear all;
close all;
load fisheriris
data = [meas(:,1), meas(:,2)];
groups = ismember(species,'setosa');
[train, test] = crossvalind('holdOut',groups);
cp = classperf(groups);
svmStruct =
svmtrain(data(train,:),groups(train),'showplot',true,'boxconstr
aint',1e6);
title(sprintf('Kernel Function: %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