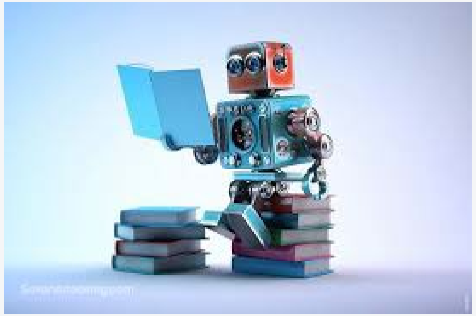
به نام خدا

یادگیری ماشین



یادگیری ماشین

آرش عبدی هجراندوست

arash.abdi.hejrandoost@gmail.com

دانشگاه علم و صنعت

دانشکده مهندسی کامپیوتر

نیم سال اول ۱۴۰۱-۱۴۰۲

ماشین بردار پشتیبان
Support Vector Machine - SVM

- ابزاری برای یادگیری با نظارت
- خیلی وقت‌ها اولین انتخاب می‌تواند باشد، اگر دانش زمینه‌ای خاصی از مساله که ما را به سمت روش خاصی سوق دهد، نداشته باشیم.
- ابزار دسته‌بندی است، نه هر یادگیری‌ای (برخلاف شبکه عصبی)

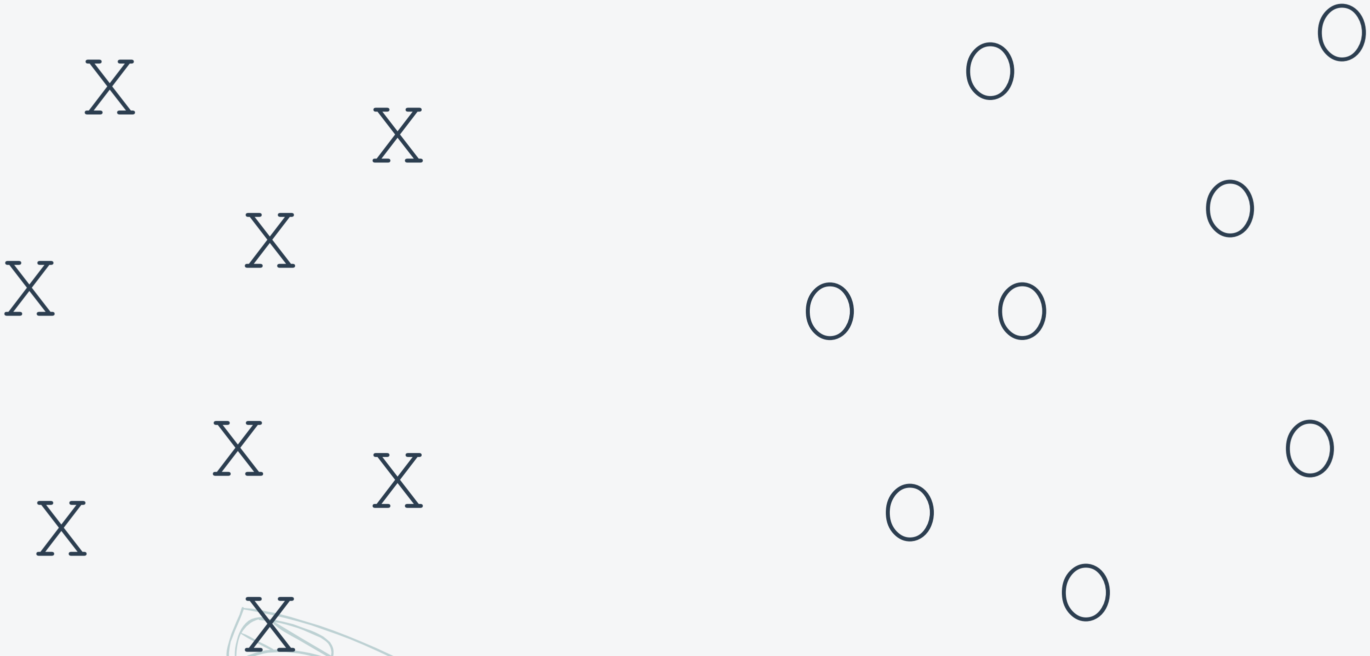
ویژگی‌های مشخصه

- خط جداکننده‌ای بر اساس ایجاد بیشترین حاشیه پیدا می‌کند
  - تعمیم بالا
- ابرصفحه‌ای خطی ایجاد می‌کند (برای جداکردن داده‌ها).
  - هرچند با استفاده از «ترفند هسته» داده‌ها به فضای دیگری (احتمالا با ابعاد بزرگتر) نگاشت می‌شوند.
  - با این ترفند، اگر داده‌ها در فضای اولیه به صورت خطی، جداپذیر نباشند، احتمالا در فضای نگاشت (فضای هسته) به صورت خطی جداپذیر خواهند بود.

ویژگی‌های مشخصه

- روشی (تقریبا) بدون پارامتر است.
  - درد پارامتر کشیده‌ها معنای جمله فوق را می‌فهمند!
    - پارامترهای آزاد یک الگوریتم، میخ‌های تابوتش هستند.
  - SVM در عوض، بخشی از خود نمونه‌ها را ذخیره می‌کند.
    - در بدترین حالت، تمام نمونه‌ها را
    - ولی معمولا درصد بسیار پایینی از آنها را
    - مثلا نمونه داده‌هایی به تعداد چند برابر ابعاد داده‌ها، نگهداری می‌شود.

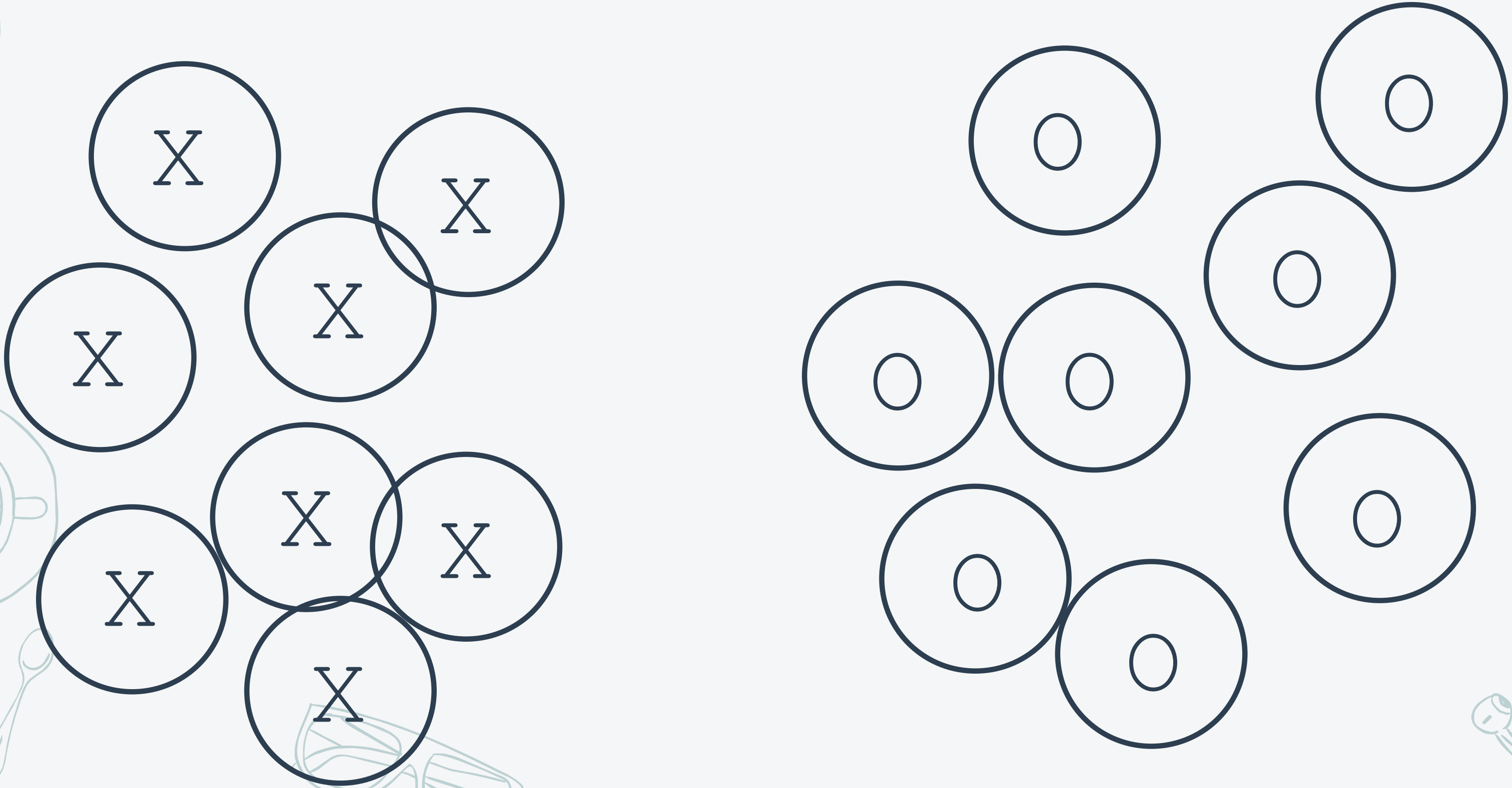
کدام خط جدا کننده؟



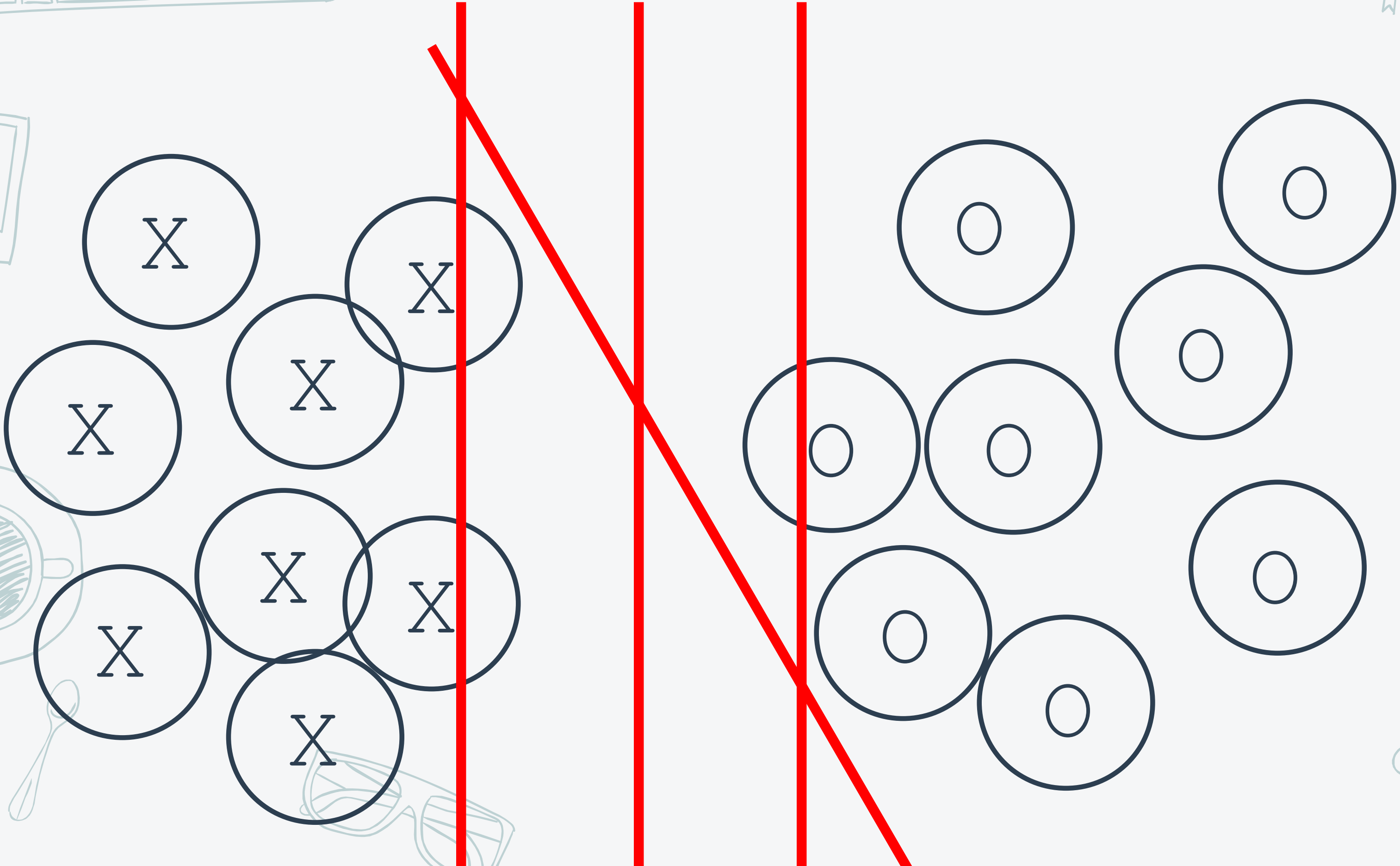
دقت بیشینه ؟

بیشتر از دقت بیشینه؟

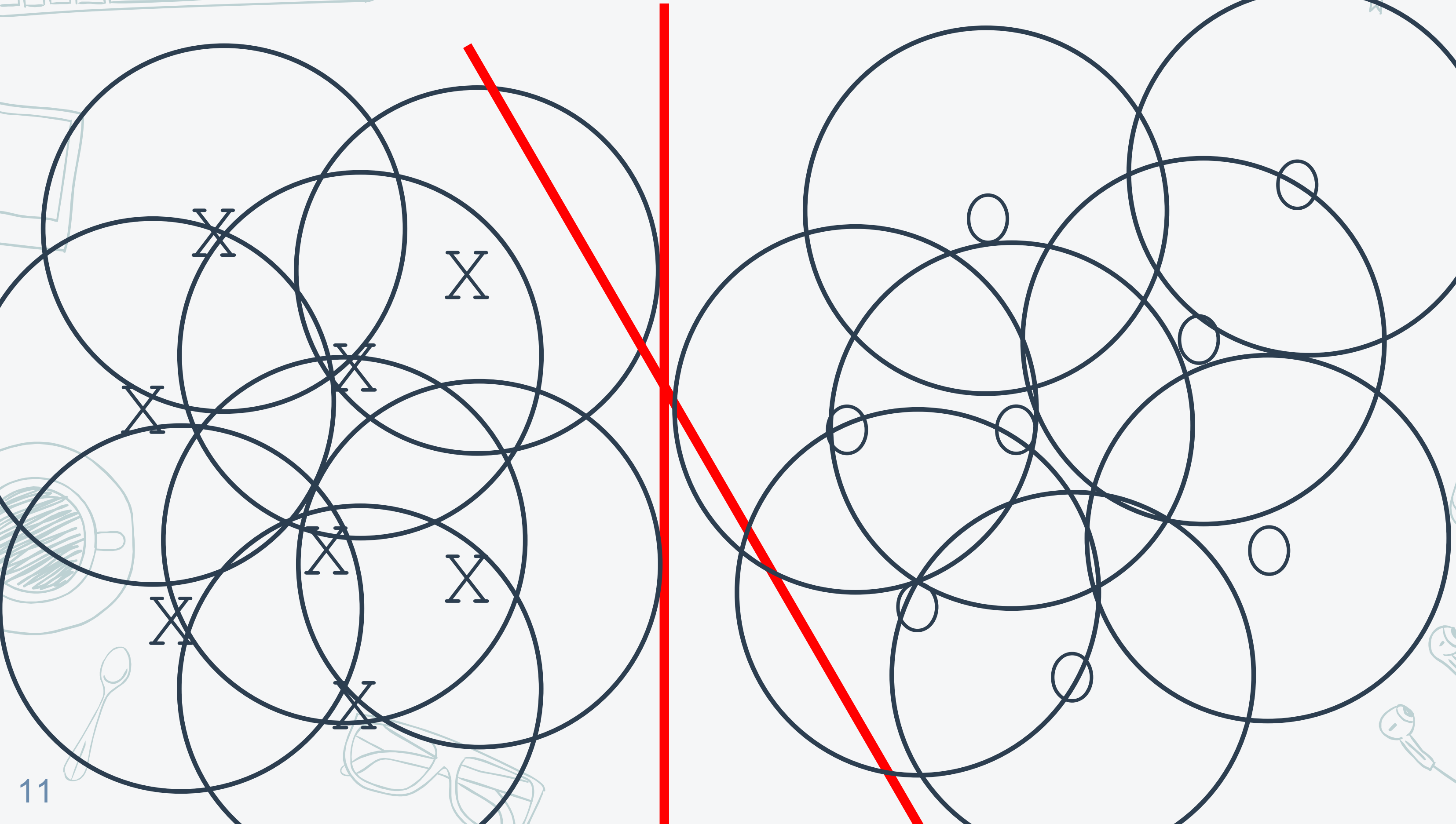
نویز و تعمیم



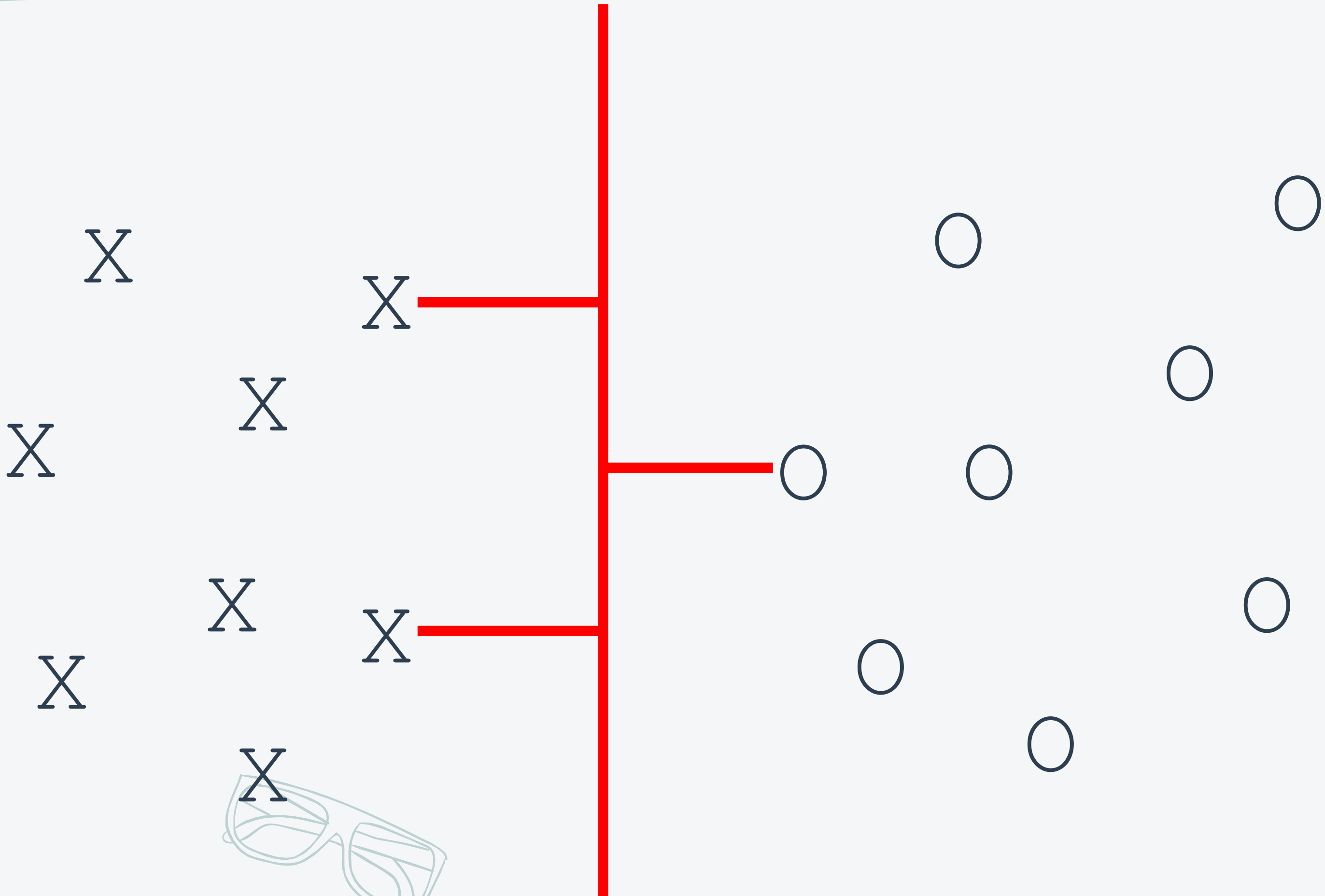
آیا خطوط با بیشترین دقت مناسب هستند؟



اگر نویز بیشتر شود ؟ (نه فقط نویز)



حاشیه امن !



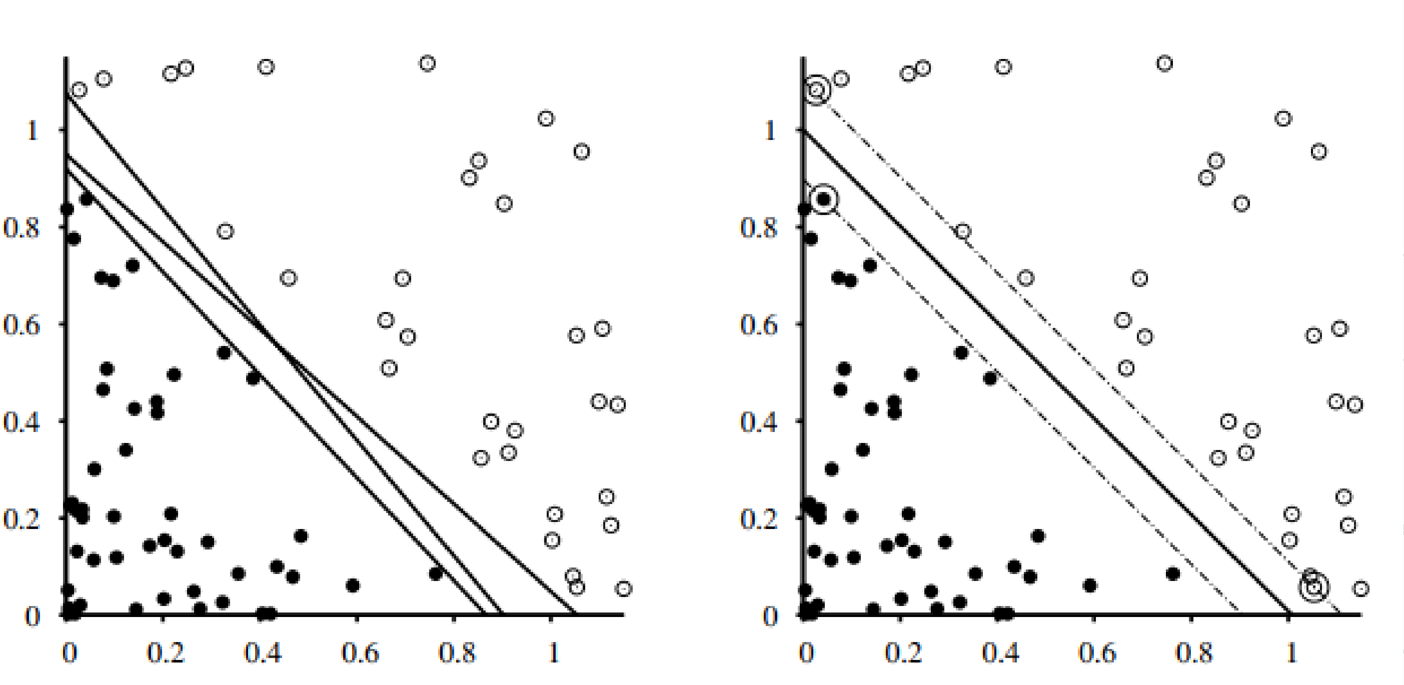
کدام خط جداکننده؟

SVM

- برخی از نمونه‌ها در تعیین خط جداگننده مهم تر هستند (باید باشند)
- تعمیم بیشتر 🡐

رگرسیون Logistic

- یکی از خطوط را پیدا می‌کند
- تمام نمونه داده‌ها (نقاط) در یافتن آن تاثیر دارند.



یافتن خط جداکننده

- SVM تلاش می‌کند قدرت تعمیم را افزایش دهد
- جدا کنندگی بر اساس ایجاد بیشترین حاشیه (Margin)
  - حاشیه: دوبرابر فاصله خط تا نزدیکترین نقطه
- مساله دو کلاسه (۱+ و ۱−)
- خط جدا کننده: $\{x : w.x + b = 0\}$
- می‌توان با کمک روش نزول در راستای گرادیان پارامترهای خط (w, b) را به گونه‌ای یافت که حاشیه بیشینه گردد، و تمام داده‌ها هم (در صورت امکان) به درستی جدا شوند.
- اما روش محاسباتی دیگری برای حل مساله فوق وجود دارد که در ادامه خلاصه آن بیان می‌شود

حل مسئله برای حالت دو بعدی

- نمونه های آموزشی

$$x \in \Re^n$$
$$y \in \{-1, 1\}$$

- تابع تصمیم گیری

$$\mathrm{sign}(\langle w,x\rangle + b)$$
$$w \in \Re^n$$
$$b \in \Re$$

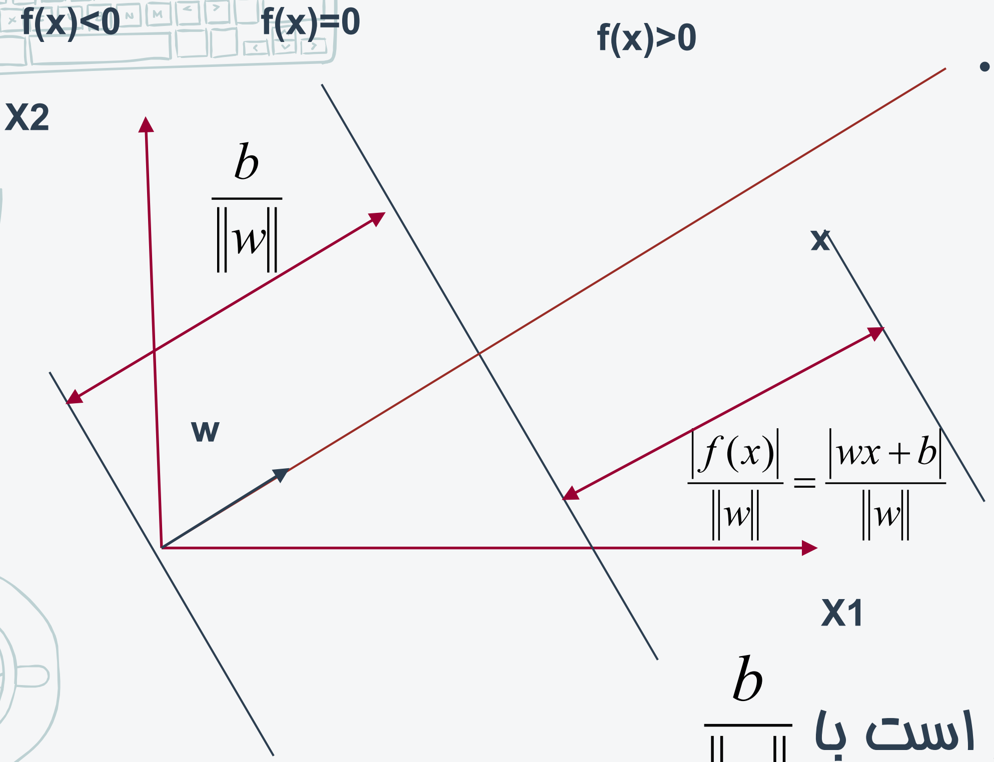
- ابر صفحه

$$\langle w, x\rangle + b = 0$$

$$w_1x_1 + w_2x_2 \dots + w_nx_n + b = 0$$

- میخواهیم مقادیر W, b را به گونه‌ای پیدا کنیم که:
  - نمونه های آموزشی را بدقت دسته بندی کند
    - با این فرض که داده ها بصورت خطی جدا پذیر باشند
  - حاشیه را حداکثر نماید

- بردار w بر هر دو صفحه مثبت و منفی عمود خواهد بود.



- فاصله خط جداکننده از مبدا برابر است با $\frac{b}{\|w\|}$

- فاصله نمونه ای مثل x از خط جدا کننده برابر است با $\frac{|f(x)|}{\|w\|}=\frac{|wx+b|}{\|w\|}$

What is the distance of a point $\mathbf{x}$ to the hyperplane $\mathcal{H}$?

Consider some point $\mathbf{x}$. Let $\mathbf{d}$ be the vector from $\mathcal{H}$ to $\mathbf{x}$ of minimum length. Let $\mathbf{x}^P$ be the projection of $\mathbf{x}$ onto $\mathcal{H}$. It follows then that:

$\mathbf{x}^P = \mathbf{x} - \mathbf{d}$.

$\mathbf{d}$ is parallel to $\mathbf{w}$, so $\mathbf{d} = \alpha\mathbf{w}$ for some $\alpha \in \mathbb{R}$.

$\mathbf{x}^P \in \mathcal{H}$ which implies $\mathbf{w}^T\mathbf{x}^P + b = 0$

therefore $\mathbf{w}^T\mathbf{x}^P + b = \mathbf{w}^T(\mathbf{x} - \mathbf{d}) + b = \mathbf{w}^T(\mathbf{x} - \alpha\mathbf{w}) + b = 0$

which implies $\alpha = \frac{\mathbf{w}^T\mathbf{x}+b}{\mathbf{w}^T\mathbf{w}}$

The length of $\mathbf{d}$:

$$\|\mathbf{d}\|_2 = \sqrt{\mathbf{d}^T\mathbf{d}} = \sqrt{\alpha^2\mathbf{w}^T\mathbf{w}} = \frac{|\mathbf{w}^T\mathbf{x}+b|}{\sqrt{\mathbf{w}^T\mathbf{w}}} = \frac{|\mathbf{w}^T\mathbf{x}+b|}{\|\mathbf{w}\|_2}$$

- اگر قدر مطلق $f(x)$ برای نزدیکترین نقطه به خط (SV) را با $\rho/2$ نشان دهیم، برای هر نمونه داده i داریم:

$$\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b \leq -\rho/2 \quad \text{if } y_i = -1$$
$$\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b \geq \rho/2 \quad \text{if } y_i = 1$$
$$\Leftrightarrow \quad y_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) \geq \rho/2$$

- برای بردارهای پشتیبان x (نزدیک‌ترین نمونه‌ها به خط جداکننده)، معادله بالا به صورت تساوی حاکم است.
- پارامترهای w و b مقیاس پذیر هستند.
  - اندازه آنها را طوری تغییر می‌دهیم که برای نزدیکترین نقطه به خط داشته باشیم: $|f(x)| = 1$
- با تغییر مقیاس پارامترهای w و b با $\rho/2$ (تقسیم دو طرف نامعادله فوق بر $\rho/2$)، فاصله هر بردار پشتیبان x با خط (ابرصفحه) جداکننده خواهد بود:

$$r = \frac{y_s(\mathbf{w}^T\mathbf{x}_s + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

- بدین صورت اندازه حاشیه نسبت عکس با اندازه w خواهد داشت:

$$\rho = 2r = \frac{2}{\|\mathbf{w}\|}$$

راه حل یافتن خط جدا کننده

Minimise $||\mathbf{w}||^2$

Subject to : $y_i(<\mathbf{w},x_i> + \mathbf{b}) \geq 1$ for all I

Where $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w}$

- این یک مسئله quadratic programming با محدودیت هائی بصورت نامعادلات خطی است.
- روش‌های شناخته شده‌ای برای چنین مسئله‌ای وجود دارد.
  - حل معادله زیر متناظر با حل معادله بالا است و میتواند مقدار w را نتیجه دهد:

$$\text{maximise}: W(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\mathbf{x}_j + \sum_{i=1}^{N}\alpha_i$$

$$\text{subject to}: \sum_{i=1}^{N}\alpha_i y_i = 0, \alpha_i \geq 0, i = 1,\ldots N$$

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

  - در این صورت w خواهد بود:
  - b به صورت مجزا محاسبه می‌شود (از معادله اولیه)

$$\text{maximise}: W(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j + \sum_{i=1}^{N} \alpha_i \qquad w = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

$$\text{subject to}: \sum_{i=1}^{N} \alpha_i y_i = 0, \alpha_i \ge 0, i = 1, \ldots N$$

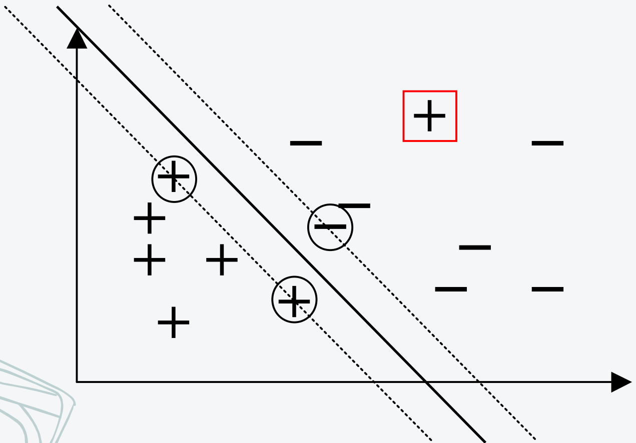
- معادله فوق محدب است ← یک جواب سرتاسری یکتا دارد که به راحتی پیدا می‌شود.
- داده‌ها در معادله فوق صرفا به صورت ضرب داخلی بین جفت نمونه‌ها حاضر شده‌اند.
  - خب؟
    - خب که خب! یه دیقه صب کن!
- حتی خود ابرصفحه جداکننده هم به صورت ضرب داخلی جفت نمونه‌ها قابل بیان است:

$$f(\mathbf{x}, \alpha^*, b^*) = \mathbf{w}^* \mathbf{x} + b^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i \mathbf{x} + b^* = \sum_{i \in SV} \alpha_i^* y_i \mathbf{x}_i \mathbf{x} + b^*$$

- α_i ها برای همه نمونه‌ها غیر از بردارهای پشتیبان صفر هستند.
  - تعداد بردارهای پشتیبان (SV) نسبت به کل داده‌ها بسیار کمتر است ← کوچک بودن مدل
  - برای داده جدید، ضرب داخلیش با تمام SV ها باید محاسبه شود و علامت تابع f فوق، تعیین کننده دسته داده است.

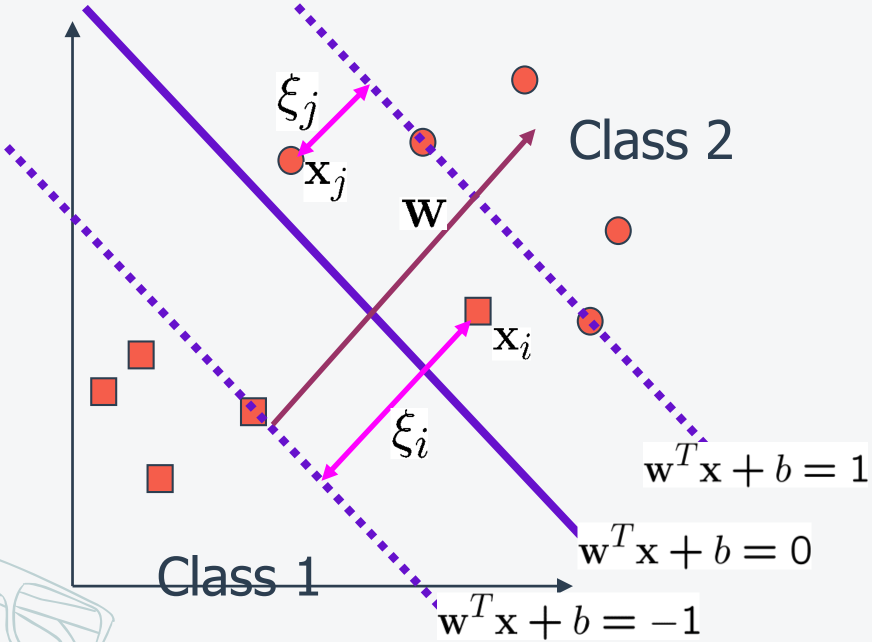
داده هائی که بصورت خطی جدا پذیر نیستند (به دلیل نویز)

- یک فرض بسیار قوی در SVM این بود که داده‌ها بصورت خطی جداپذیر باشند. در حالیکه در عمل در بسیاری مواقع این فرض صحیح نیست.
  - مثلا به دلیل وجود نویز و تناقض در داده‌های آموزشی



افزودن متغیرهای slack

- یک راه حل این است که اندکی کوتاه آمده و مقداری خطا در دسته بندی را بپذیریم!
- این کار با معرفی متغیر ξ_i انجام می‌شود که نشانگر میزان خطا در ارزیابی برخی از نمونه‌ها توسط تابع $w^T x+b$ می‌باشد.



- با معرفی متغیر ξ_i, $i=1, 2, \ldots, N$, محدودیت های قبلی ساده‌تر شده و رابطه
$y_i\ (\langle w,x_i\rangle + b) \geq 1$
- بصورت زیر تغییر می‌کند:

$$y_i\ (\langle w,x_i\rangle + b) \geq 1 - \xi_i\ , \quad \xi_i \geq 0$$

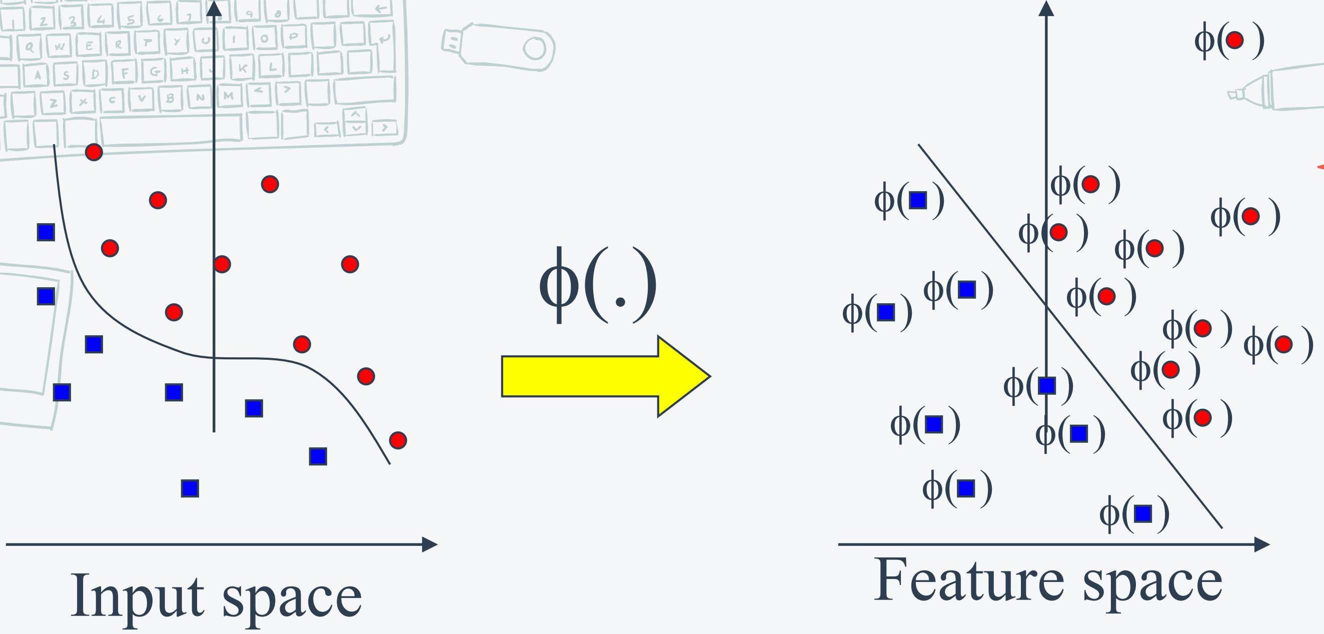
- در حالت ایده آل همه این متغیرهای ξ_i باید صفر باشند.
- در این‌صورت مسئله بهینه‌سازی تبدیل می‌شود به یافتن w به نحوی که معادله زیر کمینه گردد:

$$\|\mathbf{w}\|^2 + C\sum_i \xi_i^{\,2}$$

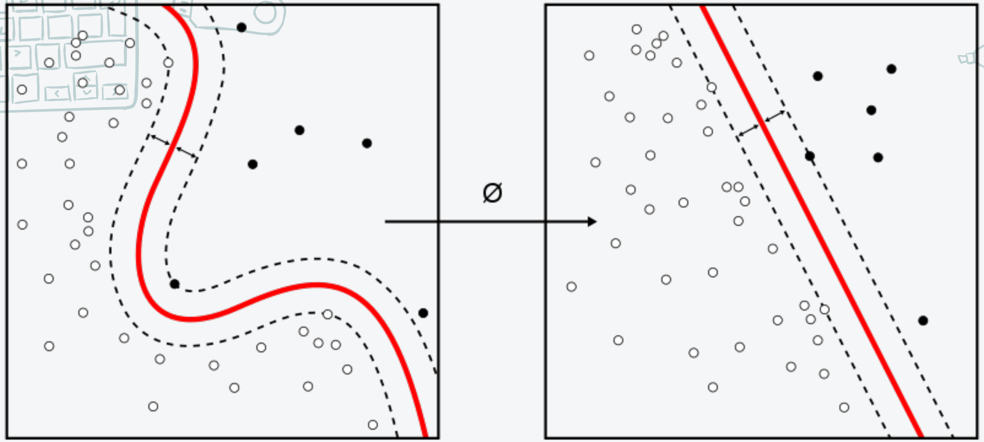
$$\text{subject to:} \qquad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i$$

- که در آن $C > 0$ می‌باشد.
- جمله اضافه شده در تابع هدف سعی دارد تا حد امکان همه متغیرهای slack را کوچک نماید.
- مقدار مناسب پارامتر C، بر اساس داده‌ها و میزان نویزی بودن آن‌ها و با آزمون و خطا باید پیدا شود (مهم)

داده هائی که بصورت خطی جدا پذیر نیستند (ذاتی)

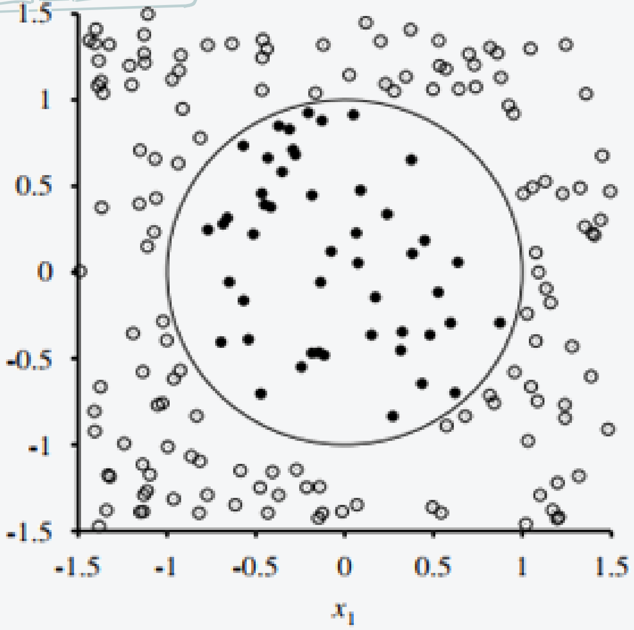


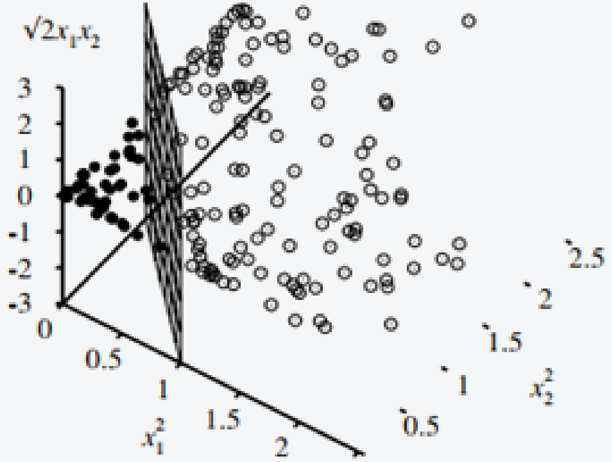
- نگاشت داده‌ها به فضایی دیگر (فضای ویژگی) با کمک تابع $\phi(x)$ که در آن فضا داده‌ها به صورت خطی جداپذیرند
  - اگر ابعاد فضای جدید به اندازه کافی بزرگ باشد، داده‌ها اغلب به صورت خطی جداپذیر خواهند بود.
- انجام محاسبات در فضای ویژگی میتواند پرهزینه باشد، زیرا ابعاد بیشتری دارد.
- در حالت کلی ابعاد این فضا بی‌نهایت است.
- برای غلبه بر این مشکل از ترفند هسته (kernel trick) استفاده می‌شود.



- نگاشت داده‌ها به فضایی دیگر (فضای ویژگی) با کمک تابع $\phi(x)$ که در آن فضا داده‌ها به صورت خطی جداپذیرند
  - اگر ابعاد فضای جدید به اندازه کافی بزرگ باشد، داده‌ها اغلب به صورت خطی جداپذیر خواهند بود.
- انجام محاسبات در فضای ویژگی میتواند پرهزینه باشد، زیرا ابعاد بیشتری دارد.
- در حالت کلی ابعاد این فضا بی‌نهایت است.
- برای غلبه بر این مشکل از ترفند هسته (kernel trick) استفاده می‌شود.

آیا نگاشت می‌تواند سودمند باشد؟





$$x = (x_1, x_2) \Rightarrow \phi(x) = (x_1{}^2, x_2{}^2, \sqrt{2}x_1x_2)$$

آیا در مثال فوق، با فضای نگاشت دارای ابعاد کمتر هم می‌توان دسته‌بندی خطی داشت؟

تعمیم SVM به حالت غیر خطی

- با داشتن $\phi(x)$، کافی است در مساله بهینه‌سازی SVM (مدل زیر)، به جای x ها، $\phi(x)$ قرار داد.

$$\text{maximise}: W(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\mathbf{x}_j + \sum_{i=1}^{N}\alpha_i$$

$$\text{subject to}: \sum_{i=1}^{N}\alpha_i y_i = 0, \alpha_i \geq 0, i = 1,\ldots N$$

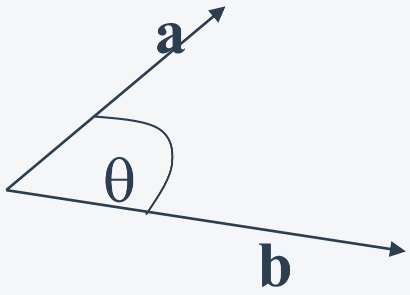
- اما تابع نگاشت $\phi(x)$ را چگونه می‌توان یافت؟
  - یافتن آن برای خطی کردن فضای داده‌ها می‌تواند بسیار دشوار و یا غیر ممکن باشد.
- اما جایگذاری $\phi(x_i).\phi(x_j)$ به جای $x_i.x_j$ در مساله فوق، دست‌آورد مهمی در پی دارد!
- برای محاسبه $\phi(x_i).\phi(x_j)$ لازم نیست $\phi(x_j)$ و $\phi(x_i)$ ابتدا به صورت جداگانه محاسبه شوند.
- مثلا در مثال اسلاید قبل، می‌توان به سادگی نشان داد:

$$\phi(x_i).\phi(x_j) = (x_i.x_j)^2$$

- یعنی در مثال فوق، بدون داشتن تابع $\phi(x)$ (با خروجی ۳ بعدی مشخص) و صرفا با جایگذاری $(x_i.x_j)^2$ به جای $x_i.x_j$ در تابع هدف SVM، داده‌های ذاتا غیرخطی، به صورت خطی جداپذیر می‌شوند.
  - آیا هنوز هم ایمان نمی‌آورید؟ پس همانا قومی ستم‌پیشه هستید!

تابع هسته

- در مثال فوق، تابع $(x_i.x_j)^2$، تابع هسته (Kernel Function) نام دارد.
- تابع هسته به صورت $K(x_i.x_j)$ نوشته می‌شود.
  - دو بردار را می‌گیرد و عددی حقیقی (که می‌تواند بیانگر نوعی شباهت بین جفت نمونه‌های ورودی باشد) را برمی‌گرداند.
- تابع هسته روی جفت نمونه‌ها اعمال می‌شود و بیانگر **ضرب داخلی دو نقطه در فضای نگاشت** است.
- **ضرب داخلی**، معیاری است که بیانگر **میزان شباهت** دو نقطه (دو نمونه داده) است.



$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}| \cos\theta$$

ترفند هسته

- با جایگذاری تابع هسته به جای ضرب داخلی جفت نمونه‌ها در محاسبات SVM، از تعریف صریح فضای نگاشت بی‌نیاز می‌شویم!
- در ریاضیات SVM، هم در یافتن خط جداکننده (فاز آموزش) و هم در مرحله آزمایش، از ضرب داخلی بین جفت نمونه‌ها استفاده شده است و هیچ x_i ای به صورت مستقل (بدون ضرب داخلی) وجود ندارد.
- بنابراین جایگذاری ضرب‌های داخلی با تابع هسته، کافی است.
- فاز آموزش:

$$\text{maximise}: W(\alpha) = -\frac{1}{2}\sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j + \sum_{i=1}^{N} \alpha_i$$

$$\text{subject to}: \sum_{i=1}^{N} \alpha_i y_i = 0, \alpha_i \geq 0, i = 1, \ldots N$$

- فاز آزمایش(استفاده از علامت خروجی تابع f):

$$f(\mathbf{x}, \alpha^*, b^*) = \mathbf{w}^* \mathbf{x} + b^* = \sum_{i=1}^{N} \alpha_i^* y_i \mathbf{x}_i \mathbf{x} + b^* = \sum_{i \in SV} \alpha_i^* y_i \mathbf{x}_i \mathbf{x} + b^*$$

تعریف تابع هسته

- طبق تئوری Mercer در ریاضیات، هر تابع هسته‌ای که شرایط mercer را ارضا کند، متناظر با یک فضای ویژگی خواهد بود.
  - وجود فضای ویژگی برای آن تابع هسته تضمین می‌شود.
  - فضای ویژگی می‌تواند دارای ابعاد بسیار زیاد باشد، حتی اگر تابع هسته در ظاهر ساده باشد.
- تابع $R \longrightarrow X \times X : k$ را هسته نامیده می‌شود اگر:
  - متقارن باشد: $k(x,y) = k(y,x)$
  - ماتریس $\boldsymbol{K}$ (موسوم به Gram matrix) حاصل از تابع k، positive semi-definite باشد.

    یعنی برای هر m نقطه دلخواه در فضا، ماتریس $\boldsymbol{K}$ با درایه‌های $k_{i,j} = k(x_i, x_j)$ در رابطه زیر صدق کند:

$$\forall c \in R^m, c^T \boldsymbol{K} c \geq 0$$

مثال‌هایی از توابع هسته

- تابع هسته خطی:

$$k(x,y) = x^T y + c$$

- تابع هسته چند جمله‌ای:

$$K(\mathbf{x}_j, \mathbf{x}_k) = (1 + \mathbf{x}_j \cdot \mathbf{x}_k)^d$$

  - متناظر با فضای ویژگی که ابعادش برحسب d نمایی است!
  - و در حالت کلی‌تر:

$$k(x,y) = (\alpha x^T y + c)^d$$

- تابع هسته گاوسی یا RBF (Radial Basis Function):

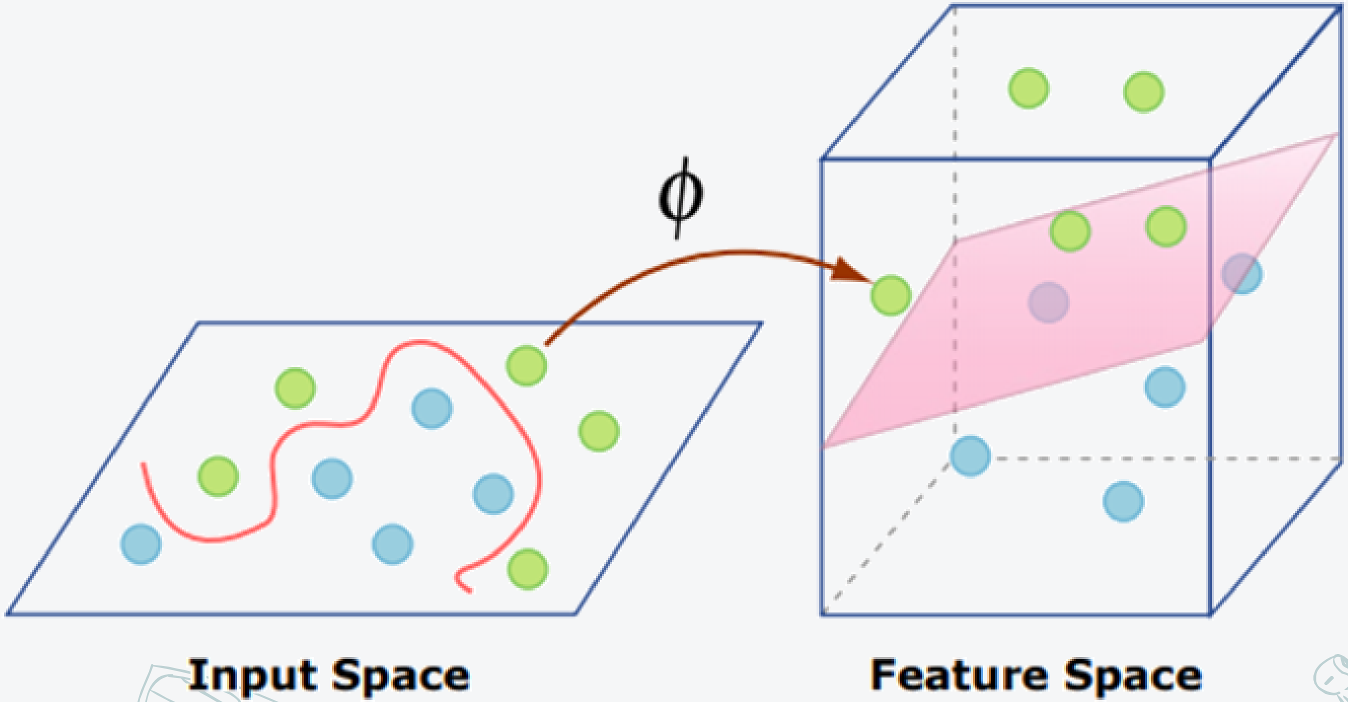
$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$

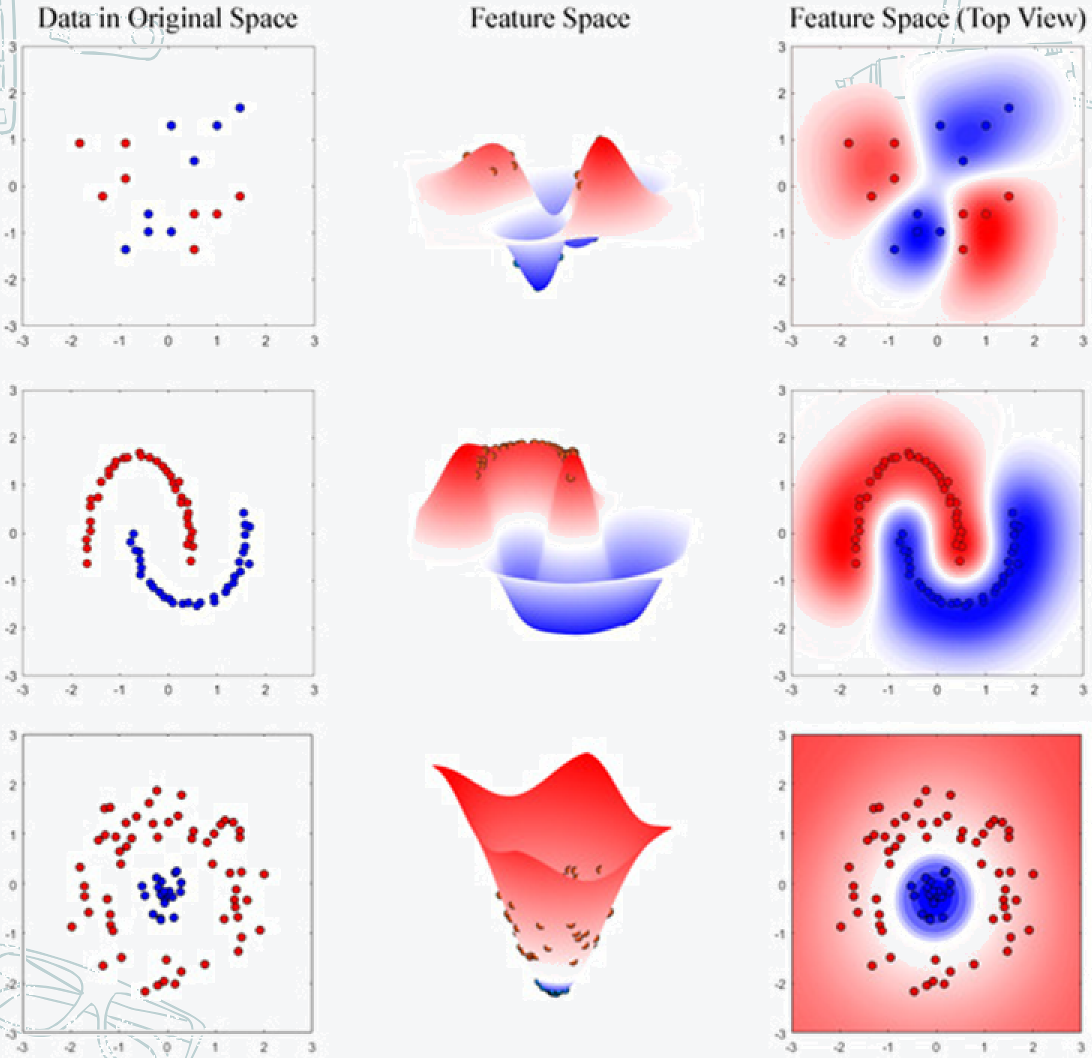
  - کاربردی و مشهورترین

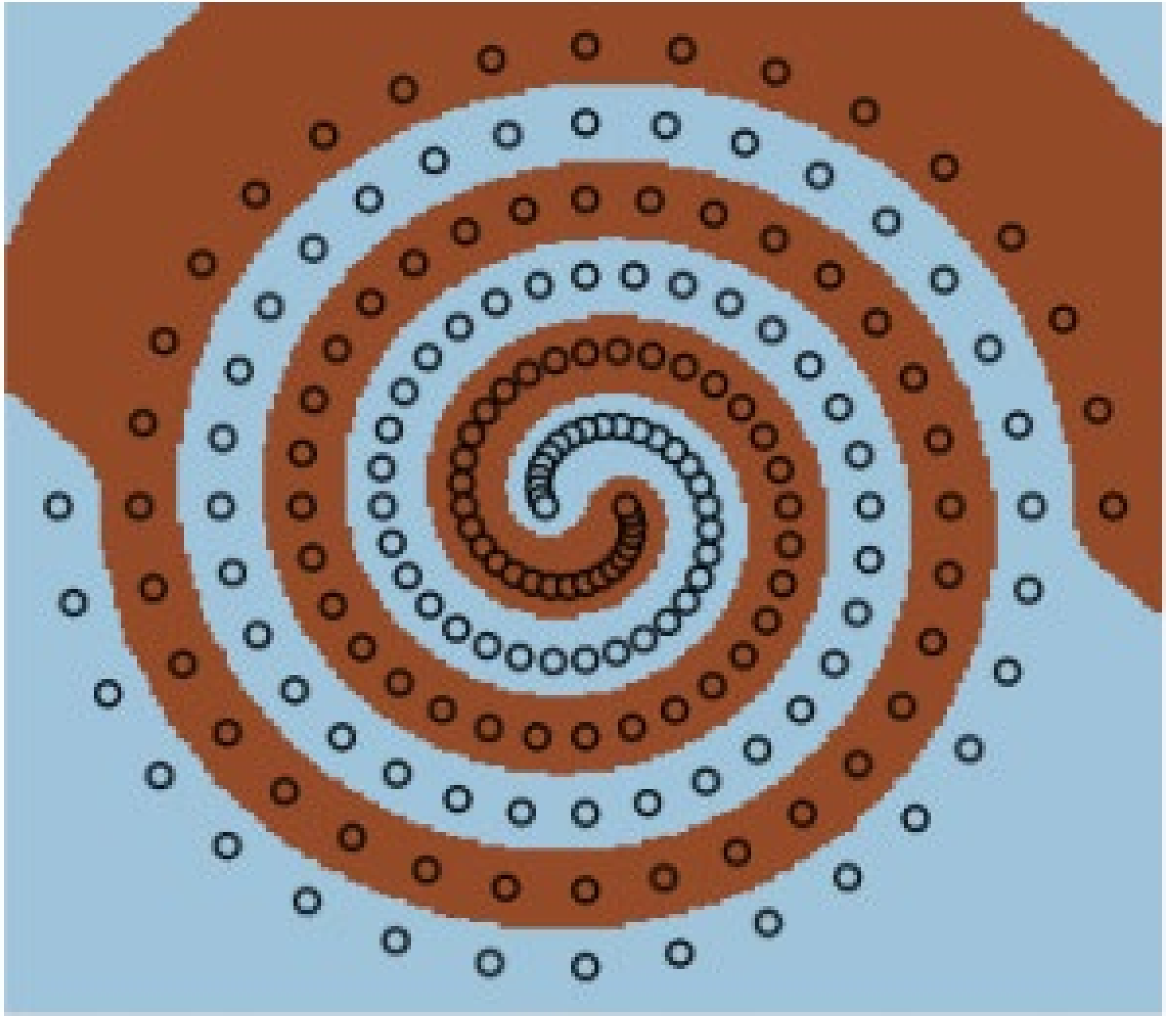
ساختن توابع هسته پیچیده‌تر

- اگر k و k' یک تابع هسته باشد:
  - $k + k'$ نیز یک تابع هسته است
  - ck برای $c > 0$ یک تابع هسته است
  - $ak + bk'$ برای $a,b > 0$ یک تابع هسته است.
  - ...
- بدین ترتیب می‌توان توابع پیچیده‌تر را با ترکیب توابع ساده‌تر ایجاد کرد.

خط جدا کننده در فضای ویژگی
منحنی متناظر در فضای اولیه




Data in Original Space
Feature Space
Feature Space (Top View)



چند نکته تستی :)!

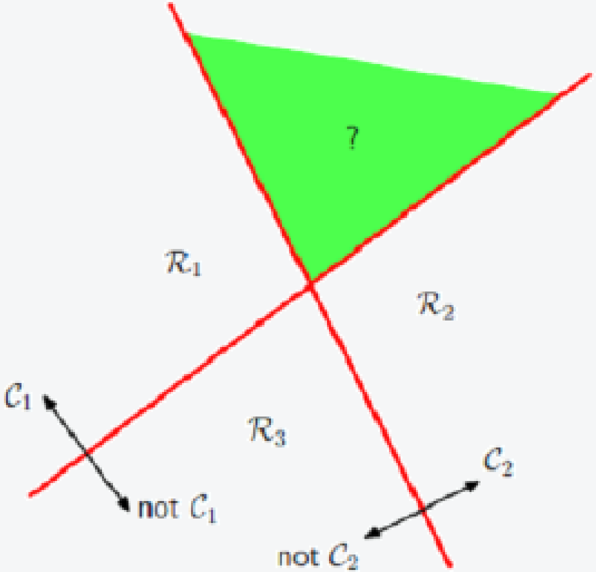
- توجه به پارامتر c (اهمیت متغیرهای slack)
- توجه به پارامترهای تابع هسته انتخاب شده
- نرمال سازی داده‌ها قبل از آموزش SVM
  - مثلا میانگین و واریانس (انحراف معیار) هر ویژگی در داده‌ها محاسبه شود و کلیه ویژگی‌ها منهای میانگین و تقسیم بر انحراف معیار شوند.
- استفاده از توابع هسته مختلف

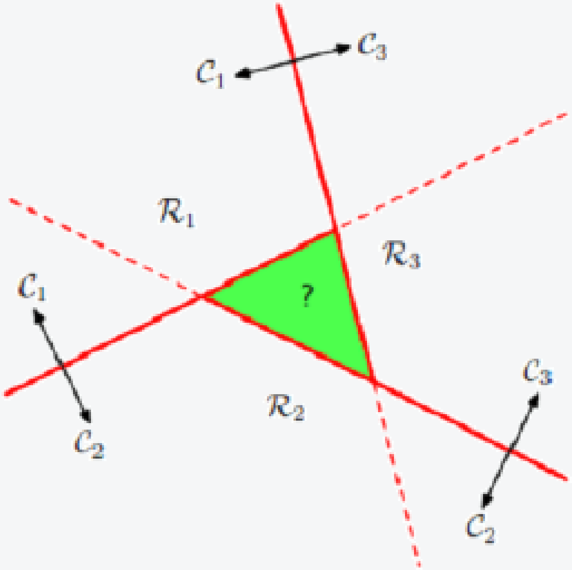
ترفند هسته در غیر SVM

- هم کاربرد دارد (!)
- بسیاری از روش‌ها و الگوریتم‌های مطرح در یادگیری ماشین، از جمله دسته‌بندی کننده‌های مختلف، با استفاده از ترفند هسته، به حالت غیرخطی تعمیم داده می‌شوند.
- شرط اعمال ترفند هسته در هر مساله و روش ریاضیاتی، وجود ضرب داخلی بین جفت نمونه‌ها در روابط مربوطه و عدم استفاده از نمونه‌های ورودی خارج از ضرب داخلی در ریاضیات مساله است.
- برای اعمال ترفند هسته، نوعا لازم است ریاضیات روش‌ها به گونه‌ای بازنویسی/ بازتولید شود که ویژگی فوق در آن پدید آید.
  - ← نسخه هسته‌ای روش‌های موجود

SVM چند کلاسه

- داشتن k تا SVM مجزا (هر یک جدا کننده یک کلاس از بقیه)
  - تناقض در پاسخ ها
  - بیشینه گیری بین خروجی ها
    - مشکل scale های متفاوت
  - داده های آموزشی نامتوازن (یک کلاس داده های کم: کلاس دوم داده زیاد)
    - در نظر گرفتن برچسب خروجی +۱ برای کلاس اصلی و (k-1)/1- برای کلاس مقابل (بقیه کلاس ها)
- در نظر گرفتن یک تابع هدف یگانه برای همه SVM های k گانه
  - هزنیه محاسباتی
- داشتن svm برای تمام جفت کلاس های ممکن و رای گیری بین svm ها برای هر کلاس برای تصمیم گیری نهایی (اکثریت)
  - تناقض در تقسیم فضاها


?
$\mathcal{R}_1$
$\mathcal{R}_2$
$\mathcal{R}_3$
$\mathcal{C}_1$
not $\mathcal{C}_1$
$\mathcal{C}_2$
not $\mathcal{C}_2$


$\mathcal{C}_1$
$\mathcal{C}_3$
$\mathcal{R}_1$
$\mathcal{R}_3$
?
$\mathcal{C}_1$
$\mathcal{C}_2$
$\mathcal{R}_2$
$\mathcal{C}_3$
$\mathcal{C}_2$

با تشکر