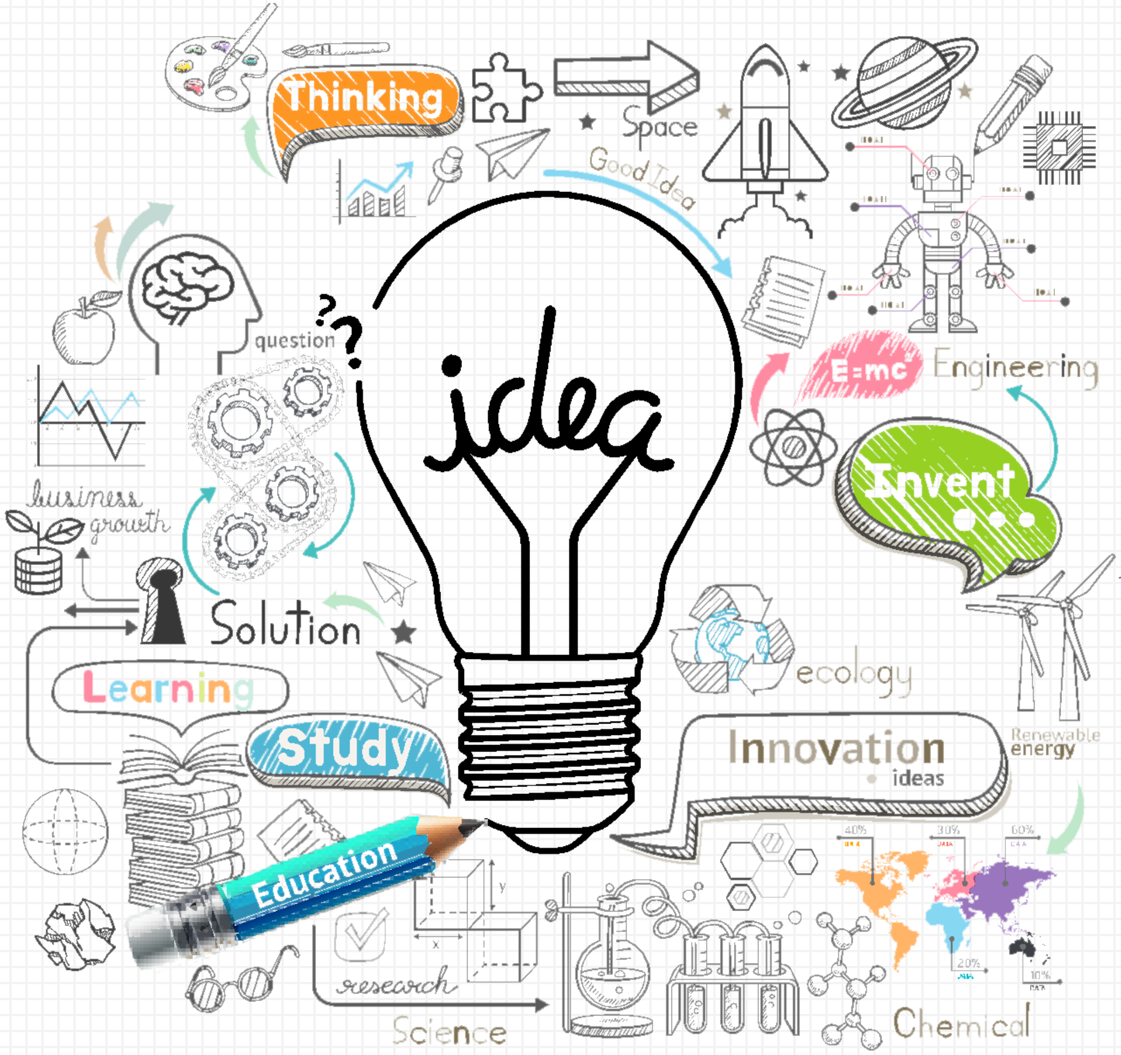


数据挖掘算法

支持向量机(SVM)

0 课程安排及相关知识

- 推导SVM待优化的基本形式（☆）
  - 向量基本知识：向量内积、内积几何意义，向量范数计算方法等。
- 推导SVM待优化的进阶形式（☆☆☆☆）
  - 高数基本知识：函数求导，矩阵乘法。
  - 拉格朗日乘数法和KKT条件：基本原理，表现形式，求解方法。
    - 梯度知识：什么是梯度，梯度方向是什么方向?
- 使用SMO算法进行求解（☆☆☆☆）

基本原理

- 深度学习版本的SVM（☆☆☆）
  - 机器学习知识：损失函数，分类任务、回归任务，神经网络知识。
- SVM的程序调用（☆☆☆）
  - https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
  - 参数讲解，其他应用中的相关知识。

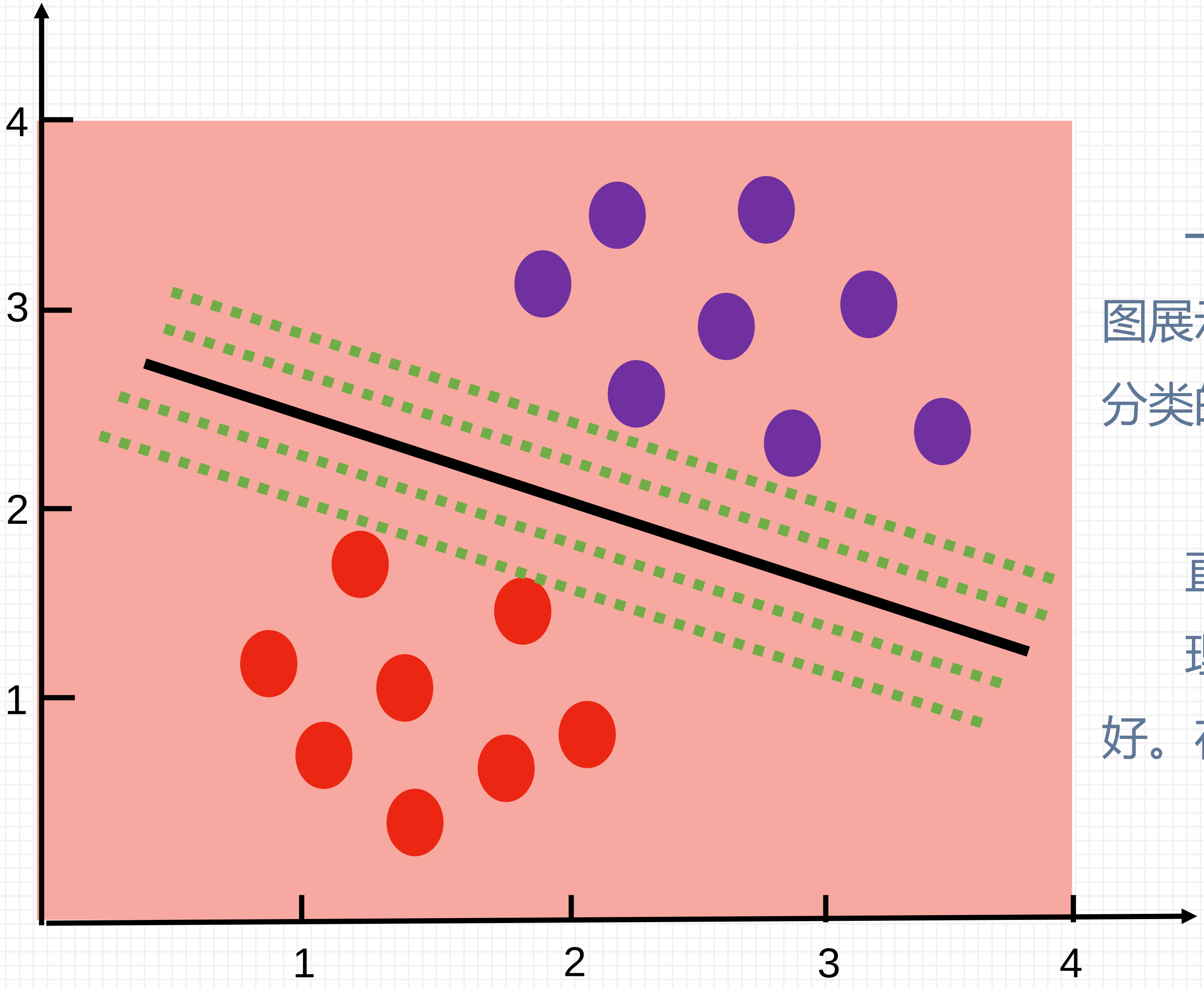
拓展知识

di yi zhang jie

第一章节

推导SVM待优化的基本形式

（☆☆）

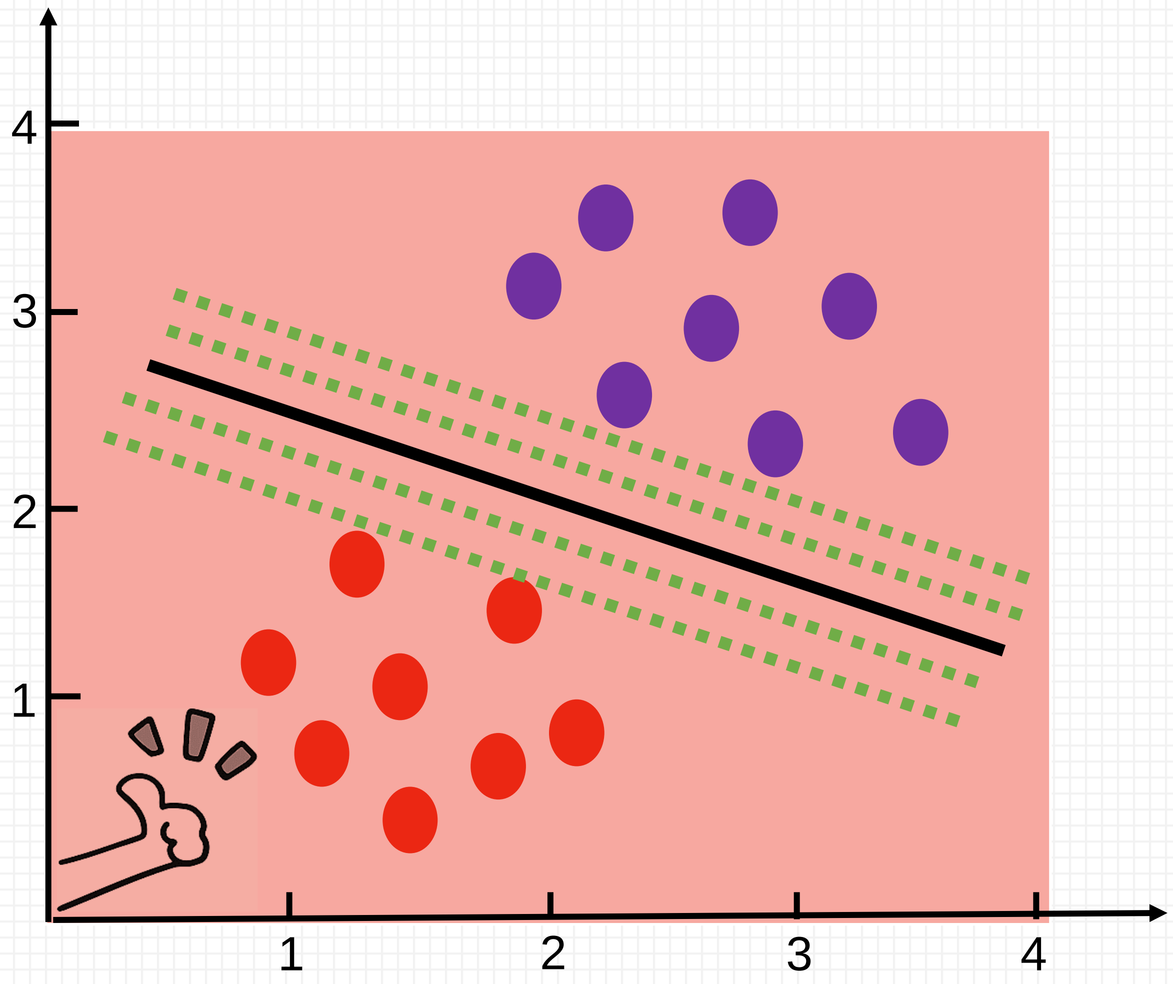


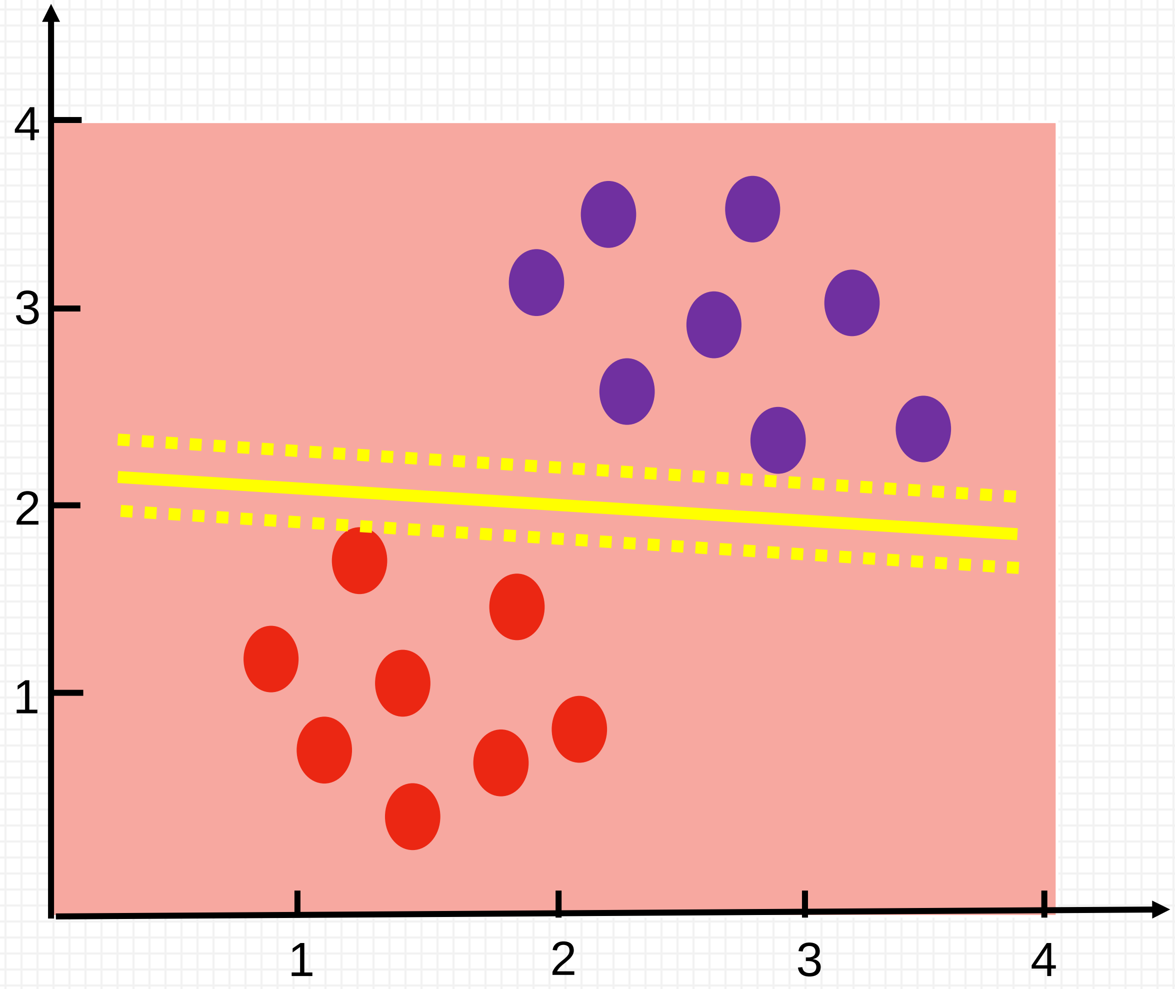
一组**线性可分**的数据，考虑一个问题，如右图展示的几条直线，都是能够将数据进行正确二分类的，问：哪个划分是最好的?

直觉告诉我：**黑色的是最好的**。

理由：黑色的看起来“容错性、健壮性”最好。在机器学习中也称“泛化性”更好 。

分割平面距离边界向量越远越好



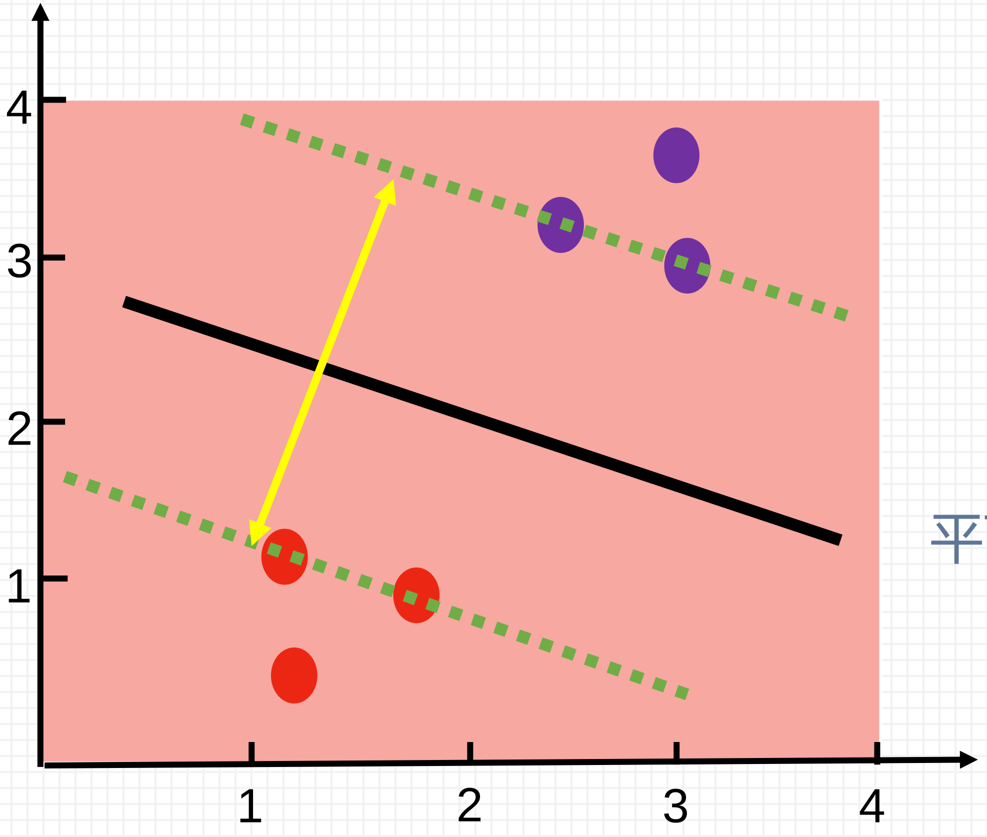


分割平面对应的几何间隔越大越好

N维空间的超平面

$$\boldsymbol{w^T x + b = 0}$$

其中w, x, b均为N维列向量。



绿色虚线：$w^T x + b = \pm k$

分类时：$w^T x + b \geq k$, 归为+1类,

$w^T x + b \leq -k$, 归为−1类。

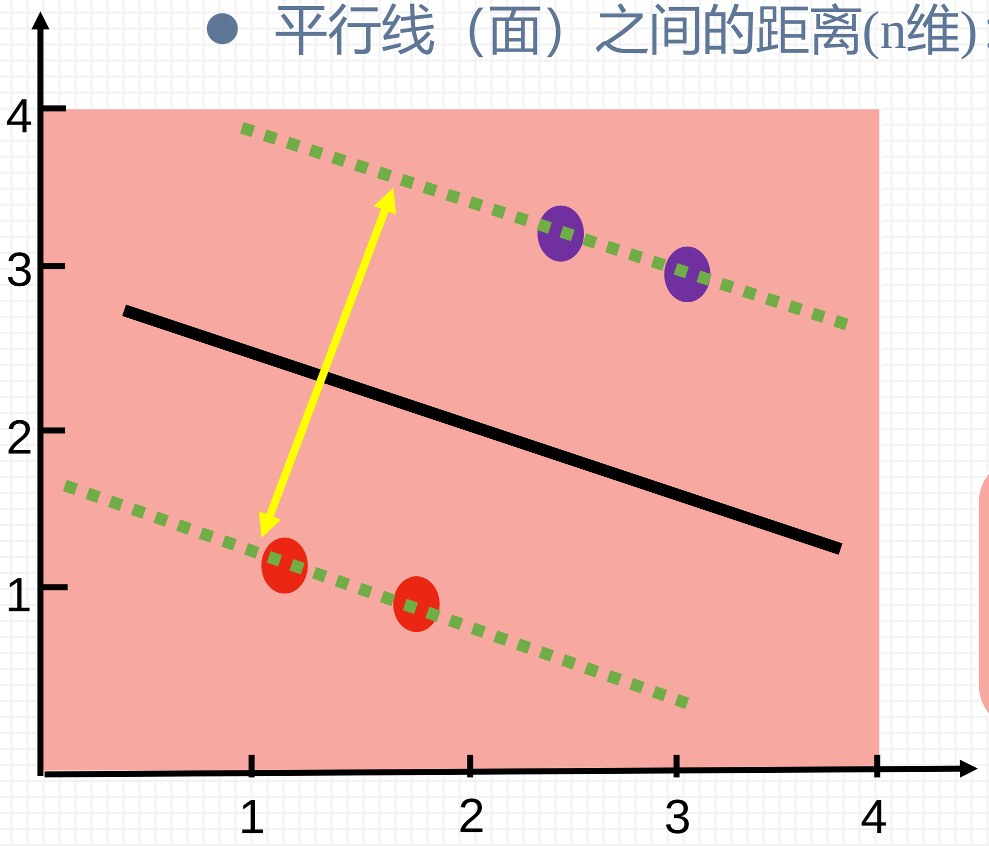
实际中：由于对超平面的权重系数做等规模放缩不影响超平面的性质。因此

$$W^T x + b = \pm k \Rightarrow \boldsymbol{w^T x + b = \pm 1}$$

找一组超平面的 w, b 使得两组绿色虚线间隔最大。

间隔的计算

- 2维距离：$ax + by + c = 0$，$d = \frac{|c_1 - c_2|}{\sqrt{a^2+b^2}}$
- 平行线（面）之间的距离(n维)：$w^T x + b = \pm 1$，$d = \frac{2}{\sqrt{w^T w}}$

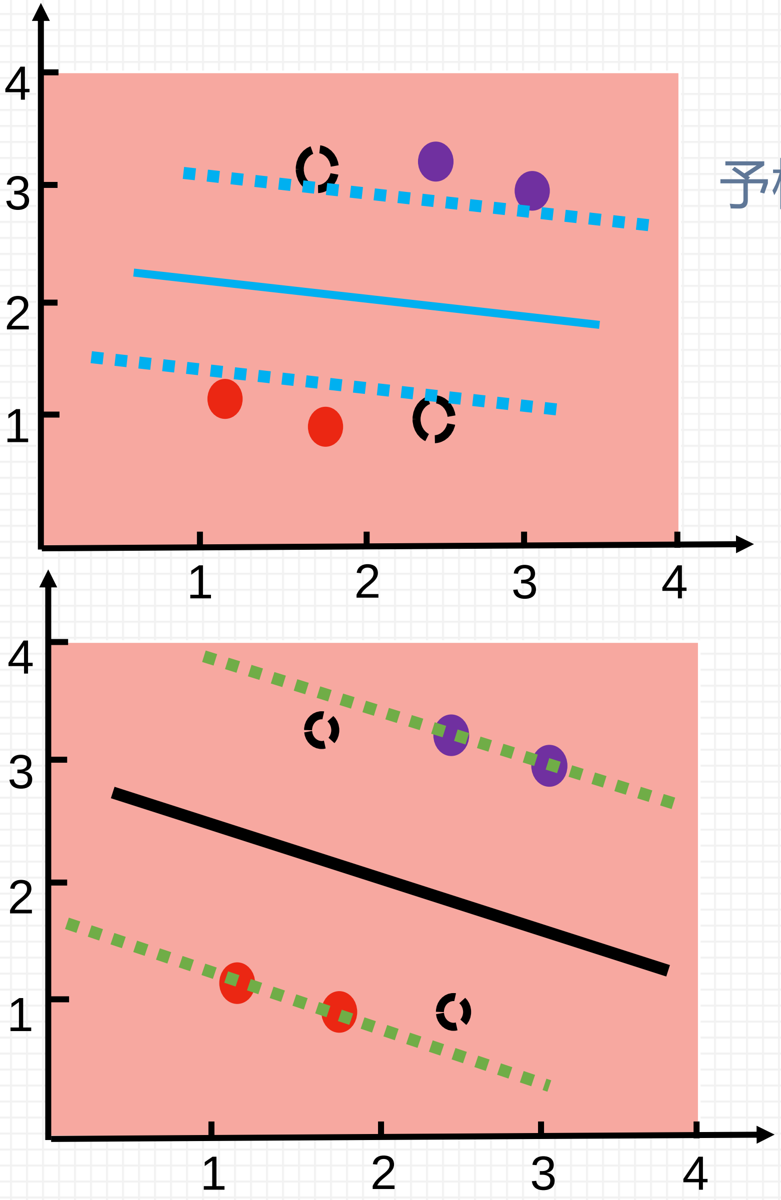


找一组超平面的 ***w b*** 使得两组绿色虚线间隔最大。

$$max \frac{2}{\sqrt{w^T w}} \rightarrow max \frac{2}{w^T w} \rightarrow min \frac{1}{2} w^T w$$

SVM基本形式1，n个样本(x_i, y_i)，$x_i \in R^N, y_i \in \{+1, -1\}$：

$$\boldsymbol{min \frac{1}{2} w^T w, \ s.t. \ y_i(w^T x_i + b) \geq 1}$$



也许一味地坚持硬性条件：$y_i(w^T x_i + b) \geq 1$，并不是最好的，也许给予模型一点点宽容，反而能带来模型的泛化能力的提高。所以约束条件变为：

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

但是，ξ_i 整体又不能太大。体现为：

$$min \quad \frac{1}{2} w^T w + C \sum_i \xi_i$$

C成为惩罚系数，平衡模型的“软硬程度”。

SVM基本形式2，n个样本(x_i, y_i)，$x_i \in R^N, y_i \in \{+1, -1\}$：

$$\boldsymbol{min\ \frac{1}{2} w^T w} + \mathrm{C} \sum_i \xi_i$$

$$\boldsymbol{s.t.\ \ y_i\left(w^T x_i + b\right) \geq 1 - \xi_i \quad , \quad \xi_i \geq 0}$$

di er zhang jie

第二章节

推导SVM待优化的进阶形式

（☆☆☆☆）

硬间隔SVM目标： $min\ \frac{1}{2}w^Tw,\ \ s.t.\ y_i(w^Tx_i+b)\geq 1$

构造拉格朗日函数:

$$L(w,b,\alpha)=\frac{1}{2}w^Tw+\sum_i\alpha_i[1-y_i(w^Tx_i+b)]\ \ s.t.\ \ \alpha_i\geq 0$$

$$\because \max_{\alpha} L(w,b,\alpha)=\frac{1}{2}w^Tw$$

$$\therefore \min_{w,b}\frac{1}{2}w^Tw=\min_{w,b}\max_{\alpha}L(w,b,\alpha)$$

对偶： $\min_{w,b}\max_{\alpha}L(w,b,\alpha)=\max_{\alpha}\min_{w,b}L(w,b,\alpha)$

我们不加证明的指出上式在SVM问题中是成立的，并且相应的解满足KKT条件。

[1] 李航. 统计学习方法[M]. 清华大学出版社, 2012. 附录C。

[2] https://zhuanlan.zhihu.com/p/219284970

目标函数: $\max_{\alpha}\min_{w,b} L(w,b,\alpha)$, $s.t.\ \alpha_i \geq 0$

$$L(w,b,\alpha)=\frac{1}{2}w^Tw+\sum_i \alpha_i[1-y_i(w^Tx_i+b)]$$

先求L关于w,b的极小值:

$$\frac{\partial L}{\partial w}=0 \Rightarrow w-\sum_i \alpha_i y_i x_i=0 \Rightarrow \boldsymbol{w=\sum_i \alpha_i y_i x_i}$$

$$\frac{\partial L}{\partial b}=0 \Rightarrow -\sum_i \alpha_i y_i=0 \Rightarrow \boldsymbol{\sum_i \alpha_i y_i=0}$$

将L拆开，并将相关结果带入:

$$L(w,b,\alpha)=\frac{1}{2}w^Tw+\sum_i \alpha_i-\sum_i \alpha_i y_i w^T x_i-b\sum_i \alpha_i y_i$$

$$L(w,b,\alpha)=\frac{1}{2}w^Tw+\sum_i\alpha_i-\sum_i\alpha_iy_iw^Tx_i-b\sum_i\alpha_i\,y_i$$

$$=\frac{1}{2}w^Tw+\sum_i\alpha_i-\sum_i\alpha_iy_iw^Tx_i-0$$

$$=\frac{1}{2}\left(\sum_i\alpha_iy_ix_i\right)^T\left(\sum_i\boldsymbol{\alpha_i\,y_ix_i}\right)+\sum_i\alpha_i-\sum_i\alpha_iy_i\left(\sum_i\boldsymbol{\alpha_iy_ix_i}\right)^Tx_i$$

$$=\frac{1}{2}\left(\sum_i\alpha_iy_ix_i\right)^T\left(\sum_j\boldsymbol{\alpha_j\,y_jx_j}\right)+\sum_i\alpha_i-\sum_i\alpha_iy_i\left(\sum_j\boldsymbol{\alpha_jy_jx_j}\right)^Tx_i$$

为防止混淆，将w写为$\sum_j\alpha_j\,y_jx_j$

$$=\frac{1}{2}\sum_i\sum_j\alpha_i\alpha_jy_iy_j({x_i}^Tx_j)+\sum_i\alpha_i-\sum_i\sum_j\alpha_i\alpha_jy_iy_j({x_i}^Tx_j)$$

$$=\sum_i\alpha_i-\frac{1}{2}\sum_i\sum_j\alpha_i\alpha_jy_i\,y_j({x_i}^Tx_j)\,,s.t.\;\;\alpha_i\geq 0\,,\sum_i\alpha_iy_i=0$$

$$\boldsymbol{w}=\sum_i\boldsymbol{\alpha_iy_ix_i}$$

$$\sum_i\boldsymbol{\alpha_iy_i}=\boldsymbol{0}$$

SVM进阶形式 1，n个样本$(x_i\,,y_i)$，$x_i\in R^N,y_i\in\{+1,-1\}$：

$$max\ \sum_i\alpha_i-\frac{1}{2}\sum_i\sum_j\alpha_i\alpha_jy_iy_j({x_i}^Tx_j)$$

s.t. $\alpha_i\geq 0\,,\sum_i\alpha_i\,y_i=0$

软间隔SVM目标：$\min \frac{1}{2}w^Tw + C\sum_i \xi_i$， $s.t.\ \ y_i(w^Tx_i + b) \geq 1 - \xi_i$ ， $\xi_i \geq 0$

构造拉格朗日函数：

$$L(w,b,\alpha,\mu,\xi) = \frac{1}{2}w^Tw + C\sum_i \xi_i + \sum_i \alpha_i[1-\xi_i - y_i(w^Tx_i + b)] + \sum_i \mu_i(-\xi_i)$$

$$s.t.\ \ \alpha_i \geq 0\ ,\ \ \mu_i \geq 0\ ,\ \ \xi_i \geq 0$$

$$\because \max_{\alpha,\mu} L(w,b,\alpha,\mu,\xi) = \frac{1}{2}w^Tw + C\sum_i \xi_i$$

$$\therefore \min_{w,b,\xi} \frac{1}{2}w^Tw + C\sum_i \xi_i = \min_{w,b,\xi}\max_{\alpha,\mu} L(w,b,\alpha,\mu,\xi)$$

对偶： $\min_{w,b,\xi}\max_{\alpha,\mu} L(w,b,\alpha,\mu,\xi) = \max_{\alpha,\mu}\min_{w,b,\xi} L(w,b,\alpha,\mu,\xi)$

我们不加证明的指出上式在SVM问题中是成立的，并且相应的解满足KKT条件。

目标函数:
$$\max_{\alpha,\mu} \min_{w,b,\xi} L(w,b,\alpha,\mu,\xi), \;\; s.t. \;\; \alpha_i \geq 0 \;, \;\; \mu_i \geq 0 \;, \;\; \xi_i \geq 0$$

$$L(w,b,\alpha,\mu,\xi) = \frac{1}{2} w^T w + C \sum_i \xi_i + \sum_i \alpha_i [1 - \xi_i - y_i (w^T x_i + b)] + \sum_i \mu_i (-\xi_i)$$

先求L关于w, b, ξ的极小值:

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w - \sum_i \alpha_i y_i x_i = 0 \Rightarrow \boldsymbol{w = \sum_i \alpha_i y_i x_i}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow -\sum_i \alpha_i y_i = 0 \Rightarrow \boldsymbol{\sum_i \alpha_i y_i = 0}$$

$$\frac{\partial L}{\partial \boldsymbol{\xi_i}} = 0 \Rightarrow \boldsymbol{C - \alpha_i - \mu_i = 0}$$

QQ：1366420642

将L拆开，并将相关结果带入：

$$L(w,b,\alpha)=\frac{1}{2}w^Tw+C\sum_i\xi_i+\sum_i\alpha_i[1-\xi_i-y_i(w^Tx_i+b)]+\sum_i\mu_i(-\xi_i)$$

$\sum_i(C-\alpha_i-\mu_i)\xi_i=0$

$$=\frac{1}{2}w^Tw\boxed{+C\sum_i\xi_i}+\sum_i\alpha_i\boxed{-\sum_i\alpha_i\xi_i}-\sum_i\alpha_iy_iw^Tx_i-b\sum_i\alpha_iy_i\boxed{-\sum_i\mu_i\xi_i}$$

$$=\frac{1}{2}(\sum_i\alpha_iy_ix_i)^T(\sum_i\alpha_iy_ix_i)+\sum_i\alpha_i-\sum_i\alpha_iy_i(\sum_i\alpha_iy_ix_i)^Tx_i$$

$$=\frac{1}{2}(\sum_i\alpha_iy_ix_i)^T(\sum_j\alpha_jy_jx_j)+\sum_i\alpha_i-\sum_i\alpha_iy_i(\sum_j\alpha_jy_jx_j)^Tx_i$$

为防止混淆，将w写为$\sum_j\alpha_j\,y_jx_j$

$$=\frac{1}{2}\sum_i\sum_j\alpha_i\alpha_jy_iy_j({x_i}^Tx_j)+\sum_i\alpha_i-\sum_i\sum_j\alpha_i\alpha_jy_iy_j({x_i}^Tx_j)$$

$\ldots\ldots,\ s.t.\ \ \alpha_i\geq 0\,,\sum_i\alpha_iy_i=0$

$$=\sum_i\alpha_i-\frac{1}{2}\sum_i\sum_j\alpha_i\alpha_jy_iy_j({x_i}^Tx_j)\,,\boxed{s.t.\ \ \alpha_i\geq 0\ ,\ \ \mu_i\geq 0\ ,\ \ \xi_i\geq 0,\ \ \sum_i\alpha_iy_i=0\ ,\ \ C-\alpha_i-\mu_i=0}$$

SVM进阶形式 2，n个样本$(x_i\,,y_i)$，$x_i\in R^N,y_i\in\{+1,-1\}$：

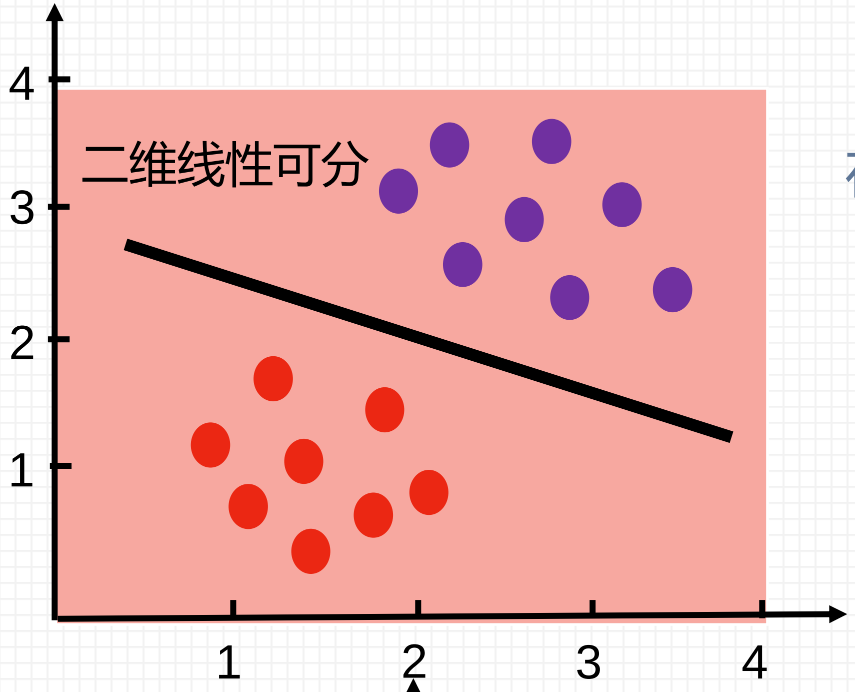
$$max\ \sum_i\alpha_i-\frac{1}{2}\sum_i\sum_j\alpha_i\alpha_jy_iy_j({x_i}^Tx_j)$$

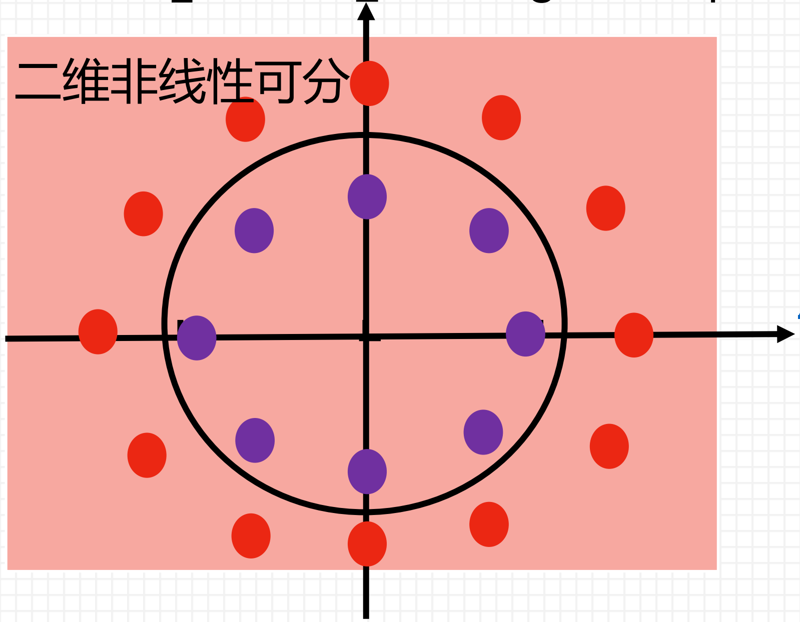
$s.t.\ \alpha_i\geq 0\ ,\ \ \mu_i\geq 0\ ,\ \ \xi_i\geq 0,\ \ C-\alpha_i-\mu_i=0\ ,\ \ \sum_i\alpha_i\,y_i=0$

$s.t.\ C\geq\alpha_i\geq 0\ ,\ \ \mu_i\geq 0\ ,\ \ \xi_i\geq 0,\ \ \sum_i\alpha_i\,y_i=0$

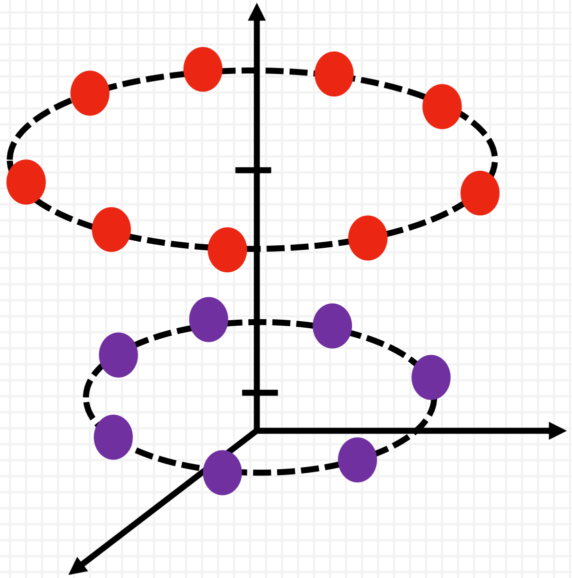
第 2.5 章 节

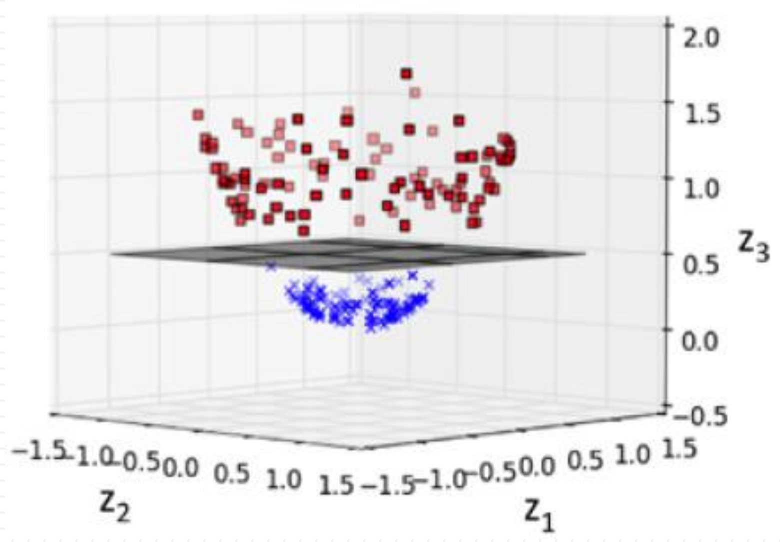
非线性可分与核技巧





在 2 维平面，红色点坐标(a,b)在半径为 1 的圆内部，紫色坐标在圆外部，不是线性可分的。





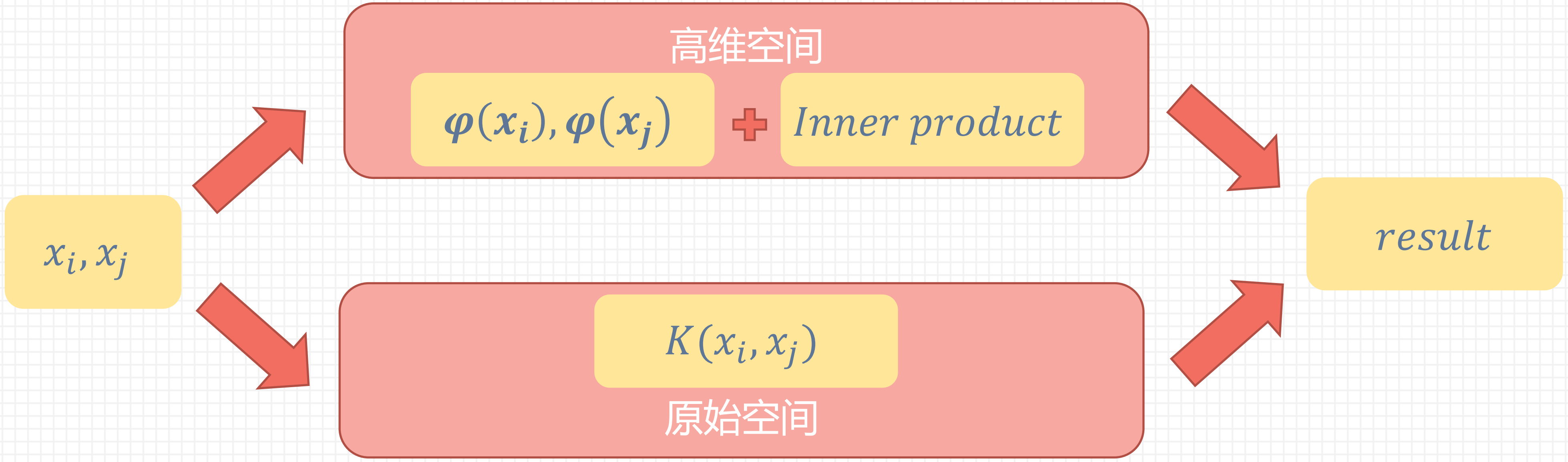
可以构造 z 纬度：$z=ab$，得到升纬后的新坐标(a,b,ab)。如图所示，可以使用$Z=1$平面进行线性划分。

因此，对于线性不可分的情况，我们可以先对样本x升纬度，然后进行后续推导。记升维后的样本为 $\tilde{x}=\varphi(x)$。

$$max\sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j (x_i{}^T x_j) \qquad\qquad max\sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j \tilde{x}_i{}^T \tilde{x}_j$$

1. **使用$\varphi(\cdot)$对变量升维$\Rightarrow$ $\tilde{x}_i = \varphi(x_i)$**
2. **计算升维后的$\varphi(x_i)$与$\varphi(x_j)$的内积**

有没有一种可能，我们可以直接通过一个$K(x_i, x_j)$直接得到x_i, x_j升维后的内积呢?



$$K(x,z)=\phi(x)\cdot\phi(z)=\begin{bmatrix}{x_1}^2\\ \sqrt{2}x_1x_2\\ {x_2}^2\end{bmatrix}\cdot\begin{bmatrix}{z_1}^2\\ \sqrt{2}z_1z_2\\ {z_2}^2\end{bmatrix}$$

$$x=\begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

$$={x_1}^2{z_1}^2+2x_1x_2z_1z_2+{x_2}^2{z_2}^2$$

$$\phi(x)=\begin{bmatrix}{x_1}^2\\ \sqrt{2}x_1x_2\\ {x_2}^2\end{bmatrix}$$

$$=(x_1z_1+x_2z_2)^2=\left(\begin{bmatrix}x_1\\ x_2\end{bmatrix}\cdot\begin{bmatrix}z_1\\ z_2\end{bmatrix}\right)^2$$

$$=(x\cdot z)^2$$

$$K(x,z)=(x\cdot z)^2 \qquad x=\begin{bmatrix}x_1\\ \vdots\\ x_k\end{bmatrix}\quad z=\begin{bmatrix}z_1\\ \vdots\\ z_k\end{bmatrix}$$

$$=(x_1z_1+x_2z_2+\cdots+x_kz_k)^2$$

$$=\underline{{x_1}^2}\,\underline{{z_1}^2}+\underline{{x_2}^2}\,\underline{{z_2}^2}+\cdots+\underline{{x_k}^2}\,\underline{{z_k}^2}$$

$$+2\underline{x_1x_2}\,\underline{z_1z_2}+2\underline{x_1x_3}\,\underline{z_1z_3}+\cdots$$

$$+2\underline{x_2x_3}\,\underline{z_2z_3}+2\underline{x_2x_4}\,\underline{z_2z_4}+\cdots$$

$$=\phi(x)\cdot\phi(z)$$

$$\phi(x)=\begin{bmatrix}{x_1}^2\\ \vdots\\ {x_k}^2\\ \sqrt{2}x_1x_2\\ \sqrt{2}x_1x_3\\ \vdots\\ \sqrt{2}x_2x_3\\ \vdots\end{bmatrix}$$

Radial Basis Function Kernel

$$x=\begin{bmatrix}x_1\\ x_2\\ \vdots\end{bmatrix}\quad z=\begin{bmatrix}z_1\\ z_2\\ \vdots\end{bmatrix}$$

$$K(x,z)=exp\left(-\frac{1}{2}\|x-z\|_2^2\right)=\phi(x)\cdot\phi(z)?$$

$\phi(*)$ has inf dim!!!

$$=exp\left(-\frac{1}{2}\|x\|_2^2-\frac{1}{2}\|z\|_2^2+x\cdot z\right)$$

$$=exp\left(-\frac{1}{2}\|x\|_2^2\right)exp\left(-\frac{1}{2}\|z\|_2^2\right)exp(x\cdot z)\quad=C_xC_zexp(x\cdot z)$$

$$=C_xC_z\sum_{i=0}^{\infty}\frac{(x\cdot z)^i}{i!}=C_xC_z+C_xC_z(x\cdot z)+C_xC_z\frac{1}{2}(x\cdot z)^2\cdots$$

$$[C_x]\cdot[C_z]\qquad\begin{bmatrix}C_xx_1\\ C_xx_2\\ \vdots\end{bmatrix}\cdot\begin{bmatrix}C_zz_1\\ C_zz_2\\ \vdots\end{bmatrix}\qquad\frac{1}{\sqrt{2}}\begin{bmatrix}C_x{x_1}^2\\ \vdots\\ \sqrt{2}C_xx_1x_2\\ \vdots\end{bmatrix}\cdot\frac{1}{\sqrt{2}}\begin{bmatrix}C_z{z_1}^2\\ \vdots\\ \sqrt{2}C_zz_1z_2\\ \vdots\end{bmatrix}$$

$$[C_x,C_xx_1,C_xx_2,\ldots C_xx_1^2,..\sqrt{2}C_xx_1x_2\ldots C_xx_1^{99}\ldots]$$
$$[C_z,C_zz_1,C_zz_2,\ldots C_zz_1^2,..\sqrt{2}C_zz_1z_2\ldots C_zz_1^{99}\ldots]$$

$$\boldsymbol{max\sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j ({x_i}^T x_j)}$$

升维

$$\boldsymbol{max\sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j (\varphi(x_i)^T \varphi(x_j))}$$

核技巧

SVM进阶形式 3，n个样本$(x_i\ ,y_i)$，$x_i \in R^N, y_i \in \{+1,-1\}$：

$$\boldsymbol{max \sum_i \alpha_i - \tfrac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j K(x_i, x_j)}$$

$$s.t.\ C \geq \boldsymbol{\alpha_i \geq 0}\ ,\ \boldsymbol{\mu_i \geq 0}\ ,\ \boldsymbol{\xi_i \geq 0},\ \boldsymbol{\sum_i \alpha_i\, y_i = 0}$$

到此，我们学习了SVM的基本形式，该模型属于凸优化问题，可以使用凸优化包对上述问题进行求解。下面我们将学习针对SVM的更加高效的优化方法：SMO(Sequential minimal optimization).

di san zhang jie

第三章节

使用SMO算法进行求解

（☆☆☆☆）

$$\boldsymbol{max\sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i\alpha_j y_i y_j K(x_i, x_j)}$$

$$\boldsymbol{s.t.\ \ C \geq \alpha_i \geq 0\ ,\ \mu_i \geq 0\ ,\ \xi_i \geq 0,\ \ \sum_i \alpha_i\, y_i = 0}$$

SMO算法的思想：同时调整n个变量太复杂，可以固定一些变量，调整少量的变量，反正能改善模型性能就好。每次调整少的变量，简单还快。

我们每次调整1个变量，固定n-1个变量。

不行，因为$\sum_i \alpha_i y_i = 0$，确定其余n-1个，那么最后1个变量也就固定了。

那就每次调整2个，固定n-2个吧。

鼠鼠你啊，真滴是太棒辣！

不妨设$\boldsymbol{\alpha_1\ y_1 + \alpha_2\ y_2 = D}$，$\boldsymbol{\alpha_1 = \frac{(D-\alpha_2\, y_2)}{y_1} = y_1(D - \alpha_2 y_2)}$，$\boldsymbol{\alpha_i \in [0, C]}$。

$$\max \sum_i \alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j K(x_i, x_j)$$

$$s.t.\ \ C \geq \alpha_i \geq 0\ ,\ \mu_i \geq 0\ ,\ \xi_i \geq 0,\ \sum_i \alpha_i\, y_i = 0$$

$$\xrightarrow{\text{忽略条件}} \min \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j k_{ij} - \sum_i \alpha_i$$

$$\min \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j k_{ij} - \sum_i \alpha_i$$

$$= \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j k_{ij} - (\alpha_1+\alpha_2) - \sum_{i=3}\alpha_i$$

$$= \frac{1}{2}\left(\sum_{i=1}^{2}\sum_j \alpha_i\alpha_j y_i y_j k_{ij} + \sum_{i=3}\sum_j \alpha_i\alpha_j y_i y_j k_{ij}\right) - (\alpha_1+\alpha_2) - \sum_{i=3}\alpha_i$$

$$= \frac{1}{2}\left[\sum_{i=1}^{2}\left(\sum_{j=1}^{2}\alpha_i\alpha_j y_i y_j k_{ij} + \sum_{j=3}\alpha_i\alpha_j y_i y_j k_{ij}\right) + \sum_{i=3}\sum_j \alpha_i\alpha_j y_i y_j k_{ij}\right] - (\alpha_1+\alpha_2) - \sum_{i=3}\alpha_i$$

$$= \frac{1}{2}\left[\sum_{i=1}^{2}\left(\sum_{j=1}^{2}\alpha_i\alpha_j y_i y_j k_{ij} + \sum_{j=3}\alpha_i\alpha_j y_i y_j k_{ij}\right) + \sum_{i=3}\left(\sum_{j=1}^{2}\alpha_i\alpha_j y_i y_j k_{ij} + \sum_{j=3}\alpha_i\alpha_j y_i y_j k_{ij}\right)\right] - (\alpha_1+\alpha_2) - \sum_{i=3}\alpha_i$$

$$= \frac{1}{2}k_{11}\alpha_1^2 + \frac{1}{2}k_{22}\alpha_2^2 + \alpha_1\alpha_2 y_1 y_2 k_{12} + \sum_{i=1}^{2}\sum_{j=3}\alpha_i\alpha_j y_i y_j k_{ij} - (\alpha_1+\alpha_2) \boxed{+ \frac{1}{2}\sum_{i=3}\sum_{j=3}\alpha_i\alpha_j y_i y_j k_{ij} - \sum_{i=3}\alpha_i}$$

$$= \frac{1}{2}k_{11}\alpha_1^2 + \frac{1}{2}k_{22}\alpha_2^2 + \alpha_1\alpha_2 y_1 y_2 k_{12} + \alpha_1 y_1 \sum_{j=3}\alpha_j y_j k_{1j} + \alpha_2 y_2 \sum_{j=3}\alpha_j y_j k_{2j} - (\alpha_1+\alpha_2)$$

$$\boldsymbol{min\ \ \frac{1}{2}k_{11}\alpha_1^2+\frac{1}{2}k_{22}\alpha_2^2+\alpha_1\alpha_2y_1y_2k_{12}+\alpha_1y_1\sum_{j=3}\alpha_jy_jk_{1j}+\alpha_2y_2\sum_{j=3}\alpha_jy_jk_{2j}-(\alpha_1+\alpha_2)}$$

记： $\boldsymbol{v_i=\sum_{j=3}\alpha_jy_jk_{ij}}$， 又前边推得$w=\sum_j\alpha_j\,y_jx_j$，所以

$$f(x_i)=w^Tx_i+b=\left(\sum_j\alpha_j\,y_jx_j\right)^Tx_i+b=\sum_j\alpha_j\,y_jx_j^T\,x_i+b=\sum_j\alpha_j\,y_jk_{ij}+b$$

$$v_i=f(x_i)-\sum_{j=1}^{2}\alpha_j\,y_jk_{ij}-b$$

$$\alpha_1=y_1(D-\alpha_2y_2)$$

$$min\ \ \frac{1}{2}k_{11}\big(y_1(D-\alpha_2y_2)\big)^2+\frac{1}{2}k_{22}\alpha_2^2+y_1(D-\alpha_2y_2)\alpha_2y_1y_2k_{12}+y_1(D-\alpha_2y_2)y_1v_1+\alpha_2y_2v_2-\big(y_1(D-\alpha_2y_2)+\alpha_2\big)$$

$$\frac{1}{2}k_{11}(D-\alpha_2y_2)^2+\frac{1}{2}k_{22}\alpha_2^2+(D-\alpha_2y_2)\alpha_2y_2k_{12}+(D-\alpha_2y_2)v_1+\alpha_2y_2v_2-y_1(D-\alpha_2y_2)-\alpha_2$$

令 $\frac{\partial obj}{\partial a_2}=0\Rightarrow k_{11}\alpha_2-k_{11}Dy_2+k_{22}\alpha_2+Dy_2k_{12}-2k_{12}\alpha_2-v_1y_2+v_2y_2+y_1y_2-1=0$

$$\boldsymbol{(k_{11}+k_{22}-2k_{12})\alpha_2=y_2(k_{11}D-k_{12}D+v_1-v_2+y_2-y_1)}$$

$$(\boldsymbol{k_{11}+k_{22}-2k_{12})\alpha_2 = y_2(k_{11}D - k_{12}D + v_1 - v_2 + y_2 - y_1)}$$

只看等式右端

将条件带入：$v_i = f(x_i) - \sum_{j=1}^{2} \alpha_j\, y_j k_{ij} - b$，$\alpha_1\, y_1 + \alpha_2\, y_2 = D$

$$\Rightarrow y_2 \begin{bmatrix} k_{11}(\alpha_1\, y_1 + \alpha_2\, y_2) - k_{12}(\alpha_1\, y_1 + \alpha_2\, y_2) \\ +(f_1 - \alpha_1\, y_1 k_{11} - \alpha_2\, y_2 k_{12} - b) - (f_2 - \alpha_1\, y_1 k_{21} - \alpha_2\, y_2 k_{22} - b) + y_2 - y_1 \end{bmatrix}$$

$$\Rightarrow y_2[(y_2 - y_1) + (f_1 - f_2) + k_{11}\alpha_2 y_2 - 2k_{12}\alpha_2 y_2 + \alpha_2\, y_2 k_{22}]$$

$$\Rightarrow y_2[(f_1 - y_1) - (f_1 - y_2) + \alpha_2 y_2(k_{11} - 2k_{12} + k_{22})]$$

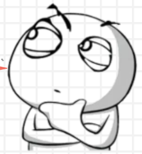
记：$E_i = f_i - y_i$，$\eta = k_{11} - 2k_{12} + k_{22}$

$$\boldsymbol{\rightarrow \eta\alpha_2 = y_2[(E_1 - E_2) + \alpha_2 y_2 \eta]}$$

$$\boldsymbol{\rightarrow \eta\alpha_2 = \alpha_2\eta + y_2(E_1 - E_2)}$$

$$\boldsymbol{\rightarrow \alpha_2 = \alpha_2 + \frac{y_2(E_1 - E_2)}{\eta}}$$

意义?



类别迭代方法：比如求解$f(x) = 0$，将其在x_n进行一阶泰勒展开，$f(x) = f(x_n) + f'(x_n)(x - x_n) = 0$

$x = x_n - \frac{f(x_n)}{f'(x_n)}$，将得到的$x$看做新的$x_n$(其实就是$x_{n+1}$)进行迭代: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

$$\boldsymbol{\alpha_2^{new} = \alpha_2^{old} + \frac{y_2\left(E_1^{old} - E_2^{old}\right)}{\eta}}$$

结合之前的条件: $\alpha_1 y_1 + \alpha_2 y_2 = D$， $\alpha_1 = \frac{(D-\alpha_2 y_2)}{y_1} = y_1(D - \alpha_2 y_2)$

改写为： $\alpha_1^{old} y_1 + \alpha_2^{old} y_2 = D, \alpha_1^{new} = y1(D - \alpha_2^{new} y_2)$

$$\Rightarrow \alpha_1^{new} = y1\left(\alpha_1^{old} y_1 + \alpha_2^{old} y_2 - \alpha_2^{new} y_2\right)$$

$$\Rightarrow \boldsymbol{\alpha_1^{new} = \alpha_1^{old} + y_1 y_2(\alpha_2^{old} - \alpha_2^{new})}$$

上边的迭代是没有考虑边界条件的，回顾我们要优化的问题以及相应的条件：

$$max \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$s.t.\ \ C \geq \alpha_i \geq 0\ ,\ \ \mu_i \geq 0\ ,\ \ \xi_i \geq 0,\ \ \sum_i \alpha_i\, y_i = 0$$

$$\boldsymbol{C \geq \alpha_1 \geq 0, C \geq \alpha_2 \geq 0}$$

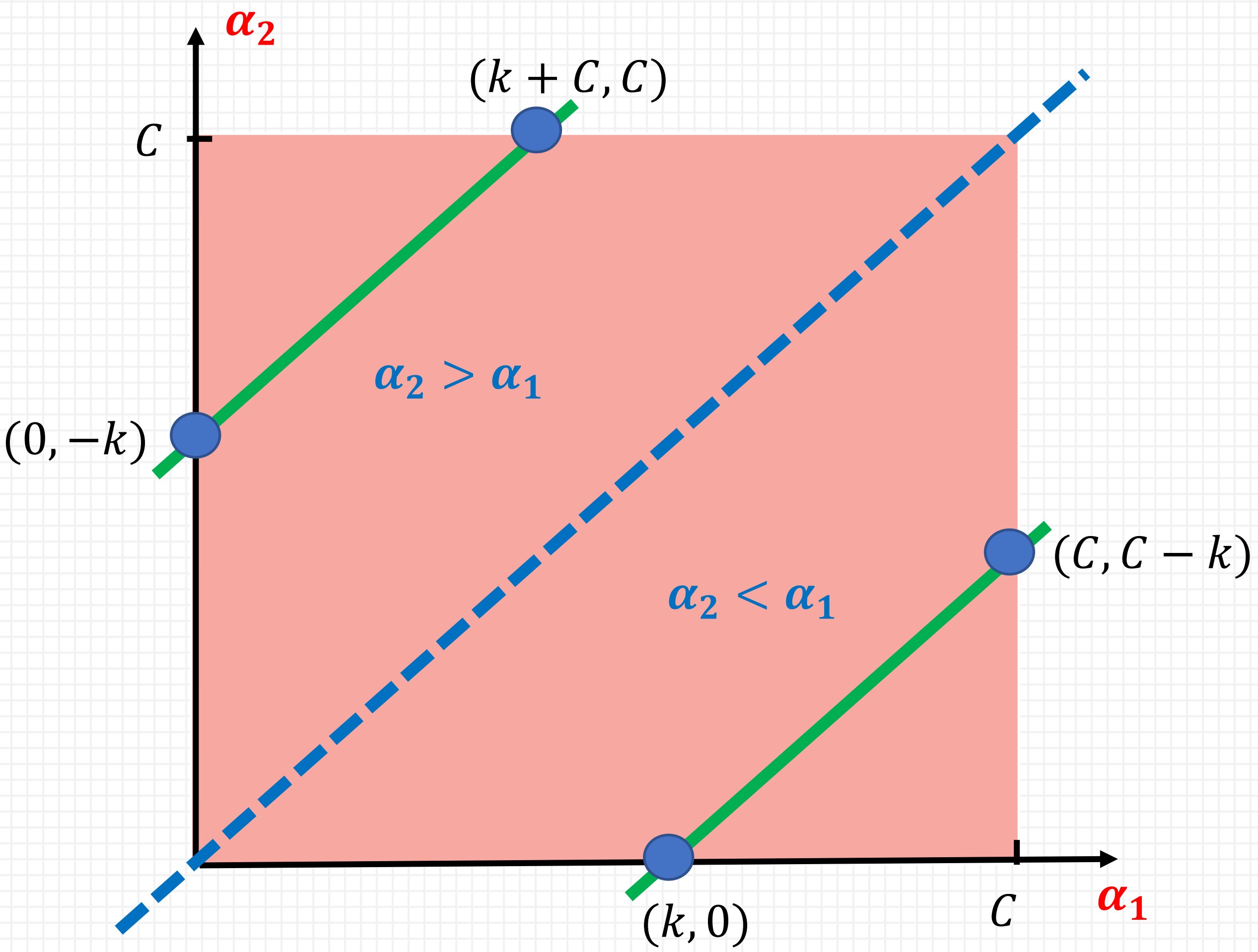
$$\boldsymbol{\alpha_1 y_1 + \alpha_2 y_2 = D}$$

当$y_1 \neq y_2$，即二者异号:

$=> \alpha_1 - \alpha_2 = k, k = \pm D$，并考虑$\alpha_1$和$\alpha_2$的大小关系

$C \geq \alpha_1 \geq 0, C \geq \alpha_2 \geq 0$

$\alpha_1 y_1 + \alpha_2 \ y_2 = D$



当$\alpha_1 > \alpha_2, \max \alpha_2 = C - k$，反之$\max \alpha_2 = C$，最大值，取交集：$H_2 = \min(C, C - \alpha_1 + \alpha_2)$

当$\alpha_1 > \alpha_2, \min \alpha_2 = 0$，反之$\min \alpha_2 = -k$，最小值，取交集：$L_2 = \max(\alpha_2 - \alpha_1, 0)$

当$y_1 \neq y_2$，即二者异号:

$$L_2 \leq \ \alpha_i \leq H_2$$

其中:

$$H_2 = \min(C, C - \alpha_1 + \alpha_2)$$

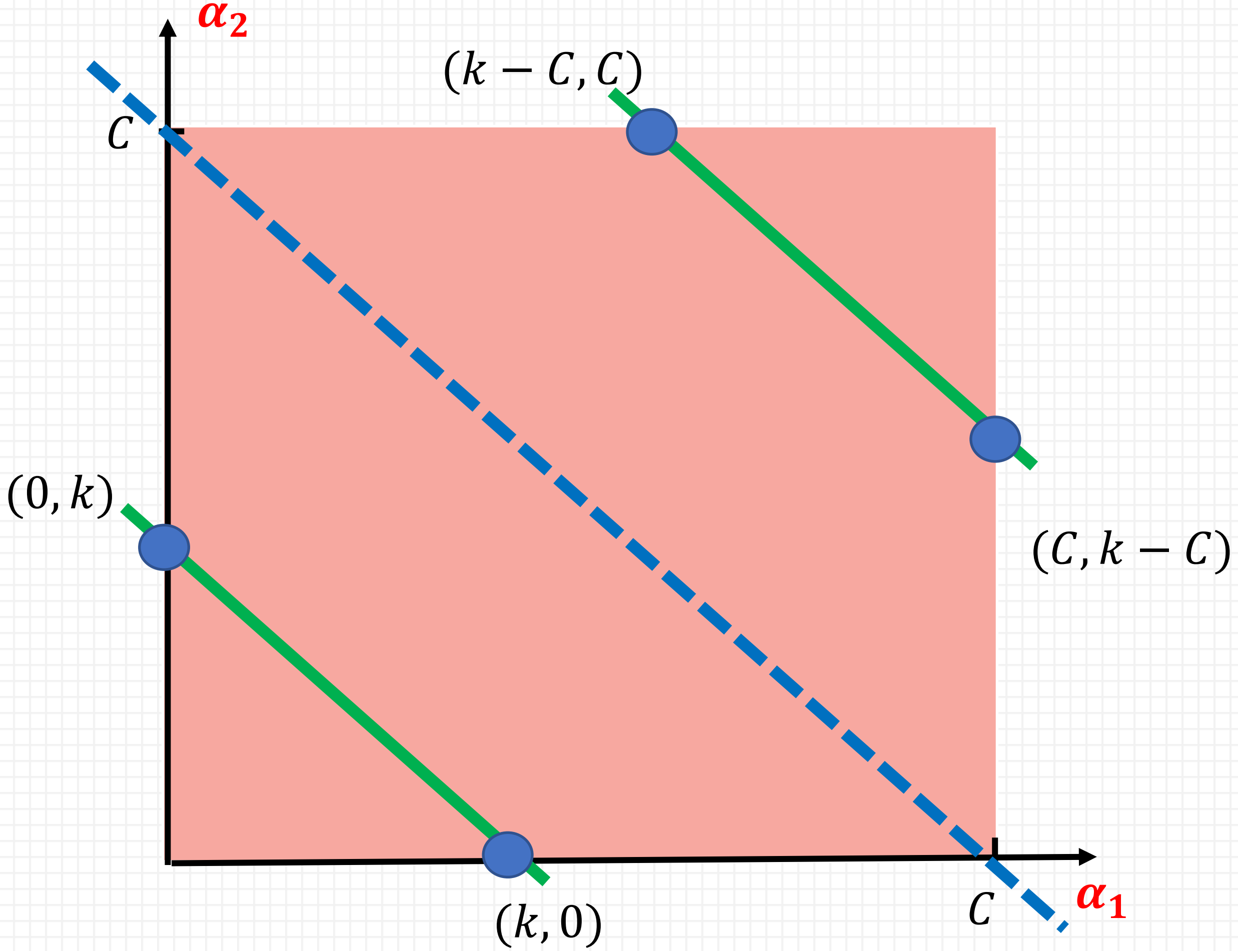
$$L_2 = \max(\alpha_2 - \alpha_1, 0)$$

当$y_1 = y_2$，即二者同号:

$=> \alpha_1 + \alpha_2 = k, k = \pm D$，亦考虑2种情况。

$C \geq \alpha_1 \geq 0, C \geq \alpha_2 \geq 0$

$\alpha_1 y_1 + \alpha_2\ y_2 = D$



情况1，$\max \alpha_2 = C$，反之 $\max \alpha_2 = k$，

最大值，取交集：$H_2 = \min(C, \alpha_1 + \alpha_2)$

情况2, $\min \alpha_2 = k - C$，反之 $\min \alpha_2 = 0$，

最小值，取交集：$L_2 = \max(\alpha_1 + \alpha_2 - C, 0)$

当$y_1 = y_2$，即二者同号:

$$L_2 \leq \alpha_i \leq H_2$$

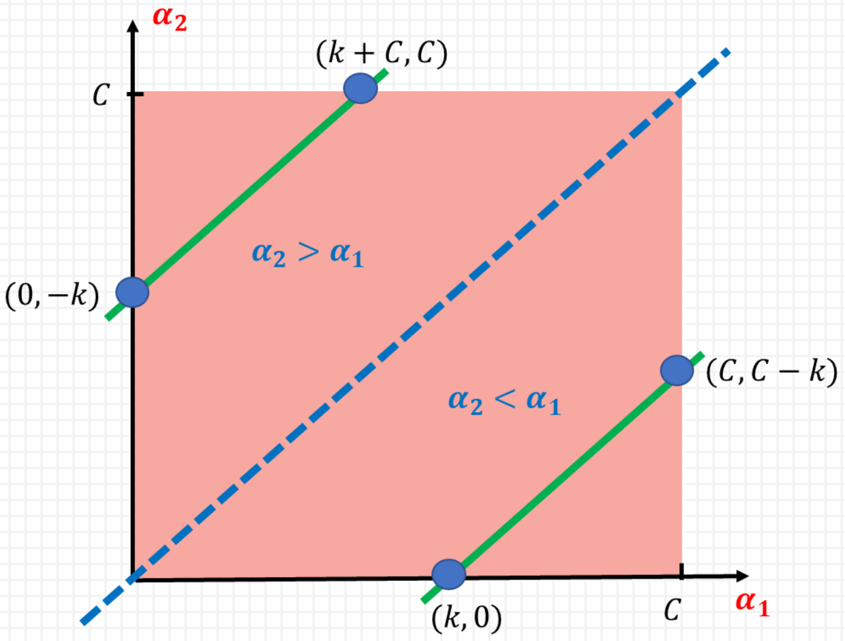
其中:

$$H_2 = \max(\alpha_1 + \alpha_2 - C, 0)$$

$$\alpha_2^{new} = \alpha_2^{old} + \frac{y_2\left(E_1^{old} - E_2^{old}\right)}{\eta}, \quad \alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new})$$

$$\boldsymbol{\alpha_2^{new} = \begin{cases} H_2, & \alpha_2 \geq H_2 \\ \alpha_2^{old} + \frac{y_2\left(E_1^{old} - E_2^{old}\right)}{\eta}, & else \\ L_2, & \alpha_2 \leq L_2 \end{cases}}$$

$$\boldsymbol{\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new})}$$



为什么α_1不需要裁剪?

答：在$\alpha_1 y_1 + \alpha_2 \ y_2 = D$的约束下，$\alpha_1$和$\alpha_2$被约束到了一条直线上。此时要求的就是，目标函数在一条平行于对角线的线段上的最优值。这使得两个变量的最优化问题成为实质上的单变量的最优化问题。不妨考虑为α_2的最优化问题，把握住了α_2，那么α_1自然就定了。

第 3.5 章节

α_1、α_2的选取与b、E_i的推导

KKT for multiple equality & inequality constraints

Given the constrained optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^2} f(\mathbf{x})$$

subject to

$$h_i(\mathbf{x})=0 \text{ for } i=1,\dots,l \text{ and } g_j(\mathbf{x})\le 0 \text{ for } j=1,\dots,m$$

Define the **Lagrangian** as

$$\mathcal{L}(\mathbf{x},\boldsymbol{\mu},\boldsymbol{\lambda})=f(\mathbf{x})+\boldsymbol{\mu}^t\,\mathbf{h}(\mathbf{x})+\boldsymbol{\lambda}^t\,\mathbf{g}(\mathbf{x})$$

Then $\mathbf{x}^*$ a local minimum $\iff$ there exists a unique $\boldsymbol{\lambda}^*$ s.t.

1. $\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*)=\mathbf{0}$
2. $\lambda_j^*\ge 0$ for $j=1,\dots,m$
3. $\lambda_j^*\,g_j(\mathbf{x}^*)=0$ for $j=1,\dots,m$
4. $g_j(\mathbf{x}^*)\le 0$ for $j=1,\dots,m$
5. $\mathbf{h}(\mathbf{x}^*)=\mathbf{0}$
6. Plus positive definite constraints on $\nabla_{\mathbf{xx}}\mathcal{L}(\mathbf{x}^*,\boldsymbol{\lambda}^*)$.

① $\frac{\partial L}{\partial w}=0\Rightarrow w^*-\sum_i\alpha_i^*y_ix_i=0\Rightarrow w^*=\sum_i\alpha_i^*y_ix_i$

$\frac{\partial L}{\partial b}=0\Rightarrow-\sum_i\alpha_i^*y_i=0\Rightarrow\sum_i\alpha_i^*y_i=0$

$\frac{\partial L}{\partial \xi_i}=0\Rightarrow C-\alpha_i^*-\mu_i^*=0$

② $\alpha_i^*\ge 0$，$\mu_i^*\ge 0$

③ $\alpha_i^*\left[1-\xi_i^*-y_i\left(w^{*T}x_i+b^*\right)\right]=0$，$-\mu_i^*\xi_i^*=0$

④ $1-\xi_i^*-y_i\left(w^{*T}x_i+b^*\right)\le 0$，$-\xi_i^*\le 0$

KKT条件在SVM中的表现：指出正确的解应该满足何种形式。

根据前边的推导，SVM的解应满足KKT条件，且满足原始约束，如$\alpha_i\ge 0$，$\xi_i\ge 0$ 等。从而得到 $C\ge\alpha_i^*\ge 0$，由此我们有：

①当 $\boldsymbol{\alpha_i^*=0}$，$C-\alpha_i^*-\mu_i^*=0\Rightarrow\mu_i^*=C\ne 0$，又 $y_i(w^Tx_i+b)-1+\xi_i\ge 0$，$\mu_i^*\xi_i^*=0$，得 $\xi_i^*=0$，从而 $y_i(w^Tx_i+b)\ge 1$，**即 $\boxed{y_if(x_i)\ge 1。}$** **(对应能够被完美分类的样本)**

在实际中是严格大于（小于）

②当 $\boldsymbol{0<\alpha_i^*<C}$，$C-\alpha_i^*-\mu_i^*=0\Rightarrow\mu_i^*=C-\alpha_i^*\ne 0$，又$\alpha_i^*\left[1-\xi_i^*-y_i\left(w^{*T}x_i+b^*\right)\right]=0$，得 $1-\xi_i^*-y_i\left(w^{*T}x_i+b^*\right)=0$，又 $\mu_i^*\xi_i^*=0$，所以 $\xi_i^*=0$，从而 $y_i(w^Tx_i+b)=1$，**即 $y_if(x_i)=1$。(对应卡在边界的样本，即支持向量)**

③当 $\boldsymbol{\alpha_i^*=C}$，$C-\alpha_i^*-\mu_i^*=0\Rightarrow\mu_i^*=0$，又 $1-\xi_i^*-y_i\left(w^{*T}x_i+b^*\right)=0\Rightarrow y_i\left(w^{*T}x_i+b^*\right)=1-\xi_i^*$，又 $\mu_i^*\xi_i^*=0$，得 $\xi_i^*\ge 0$，从而 $y_i(w^Tx_i+b)\le 1$，**即 $\boxed{y_if(x_i)\le 1。}$** **(对应超平面和边界之间的样本)**

等价KKT条件：① $\alpha_i^* = 0 \Leftrightarrow y_i f(x_i) \geq 1$。② $0 < \alpha_i^* < C \Leftrightarrow y_i f(x_i) = 1$。③ $\alpha_i^* = C \Leftrightarrow y_i f(x_i) \leq 1$。

α_i的更新公式： $\alpha_1^{new} = \alpha_1^{old} + y_1 y_2(\alpha_2^{old} - \alpha_2^{new})$ ， $\alpha_2^{new} = \begin{cases} H_2, & \alpha_2 \geq H_2 \\ \alpha_2^{old} + \frac{y_2(E_1^{old} - E_2^{old})}{\eta}, & else \\ L_2, & \alpha_2 \leq L_2 \end{cases}$

- α_1、 α_2的选取
  - **外循环先选α_1**： 目标是尽量选取违反KKT条件的变量，并且优先判断支撑边界上的样本对应的α。具体操作为，先检验$0 < \alpha_i < C$，对应的$y_i f(x_i) = 1$是否成立？选择不成立、结果差的最离谱的 α_i 作为此次的 α_1。如果都满足，则再遍历其余样本。
  - **内循环选择α_2**：思路是，在选定α_1的基础上，选一个α_2能够有足够大的变动：具体为，因为α_2^{new}的变动量是与$|E_1 - E_2|$成正比的，而α_1选定之后，E_1就定了。所以，如果E_1是正的，那么选择最小的E_i的对应的α_i作为α_2，反之选择最大的E_i的对应的α_i作为α_2。
    - 如果上述的选择方式不能使α_2能够有足够大的变动（变动量小于某个阈值），那么启发式的遍历支持向量对应的α_i作为α_2，判断是否能够满足条件。再不行，那就遍历其余样本，最后，实在不行，那就换个α_1。

- 截距 b 的计算:
  - 若 $0 < \alpha_1^{new} < C$，$y_i(w^Tx_i + b) = 1 \Rightarrow w^Tx_i + b = y_i \Rightarrow y_i = b - w^Tx_i$

$$b_1^{new} = y_1 - \sum_{i=3} \alpha_i y_i k_{i1} - \alpha_1^{new} y_1 k_{11} - \alpha_2^{new} y_2 k_{21}$$

$$E_1^{old} = \sum_{i=3} \alpha_i y_i k_{i1} + \alpha_1^{old} y_1 k_{11} + \alpha_2^{nold} y_2 k_{21} + b^{old} - y_1$$

$$\Rightarrow b_1^{new} = \left(-E_1^{old} + \sum_{i=3} \alpha_i y_i k_{i1} + \alpha_1^{old} y_1 k_{11} + \alpha_2^{nold} y_2 k_{21} + b^{old}\right) - \sum_{i=3} \alpha_i y_i k_{i1} - \alpha_1^{new} y_1 k_{11} - \alpha_2^{new} y_2 k_{21}$$

$$\Rightarrow \boldsymbol{b_1^{new} = -E_1^{old} + (\alpha_1^{old} - \alpha_1^{new})y_1 k_{11} + (\alpha_2^{old} - \alpha_2^{new})y_2 k_{21} + b^{old}}$$

  - 若$0 < \alpha_2^{new} < C$，同理

$$\Rightarrow \boldsymbol{b_2^{new} = -E_2^{old} + (\alpha_1^{old} - \alpha_1^{new})y_1 k_{12} + (\alpha_2^{old} - \alpha_2^{new})y_2 k_{22} + b^{old}}$$

哪个α满足条件，就$b^{new} = b_i^{new}$，否则就取二者均值作为新的b。

- $\boldsymbol{E_i}$ 的计算:

$$\boldsymbol{E_i^{new} = \sum \alpha_j^{new} y_j k_{ji} + b^{new} - y_i}$$

● 算法流程

- **输入：训练集n个样本(x_i, y_i)，$x_i \in R^N$,$y_i \in \{+1,-1\}$，精度 ϵ，输出：α**

① **初始化迭代次数 $k=0$，设置 $\alpha_i^k=0$**

② **选择 α_1^k、 α_2^k，并计算**

$$E_i^k = \alpha_1^k y_1 k_{1i} + \alpha_2^k y_2 k_{2i} + \sum_{j=3} \alpha_j y_j k_{ji} - y_i，\ \eta = k_{11} - 2k_{12} + k_{22}$$

③ **更新计算 α_1^{k+1}、 α_2^{k+1}**

$$\alpha_2^{k+1} = \alpha_2^k + \frac{y_2(E_1^k - E_2^k)}{\eta},clip,\ \alpha_1^{k+1} = \alpha_1^k + y_1 y_2(\alpha_2^k - \alpha_2^{k+1})$$

④ **更新计算 b^{k+1} 、E_i^{k+1}**

$$b_1^{new} = -E_1^{old} + \left(\alpha_1^{old} - \alpha_1^{new}\right) y_1 k_{11} + \left(\alpha_2^{old} - \alpha_2^{new}\right) y_2 k_{21} + b^{old}$$

$$b_2^{new} = -E_2^{old} + \left(\alpha_1^{old} - \alpha_1^{new}\right) y_1 k_{12} + \left(\alpha_2^{old} - \alpha_2^{new}\right) y_2 k_{22} + b^{old}$$

$$E_i^{new} = \sum \alpha_j^{new} y_j k_{ji} + b^{new} - y_i$$

⑤ **是否在ϵ范围内满足KKT条件**

$$\sum_i \alpha_i y_i = 0, y_i f(x_i): \begin{cases} \geq 1，\ \{x_i \mid \alpha_i = 0\} \\ = 1，\ \{x_i \mid 0 < \alpha_i < C\} \\ \leq 1，\ \{x_i \mid \alpha_i = C\} \end{cases}$$

⑥ **$\mathrm{k} = \mathrm{k} + 1$**

di si zhang jie

第四章节

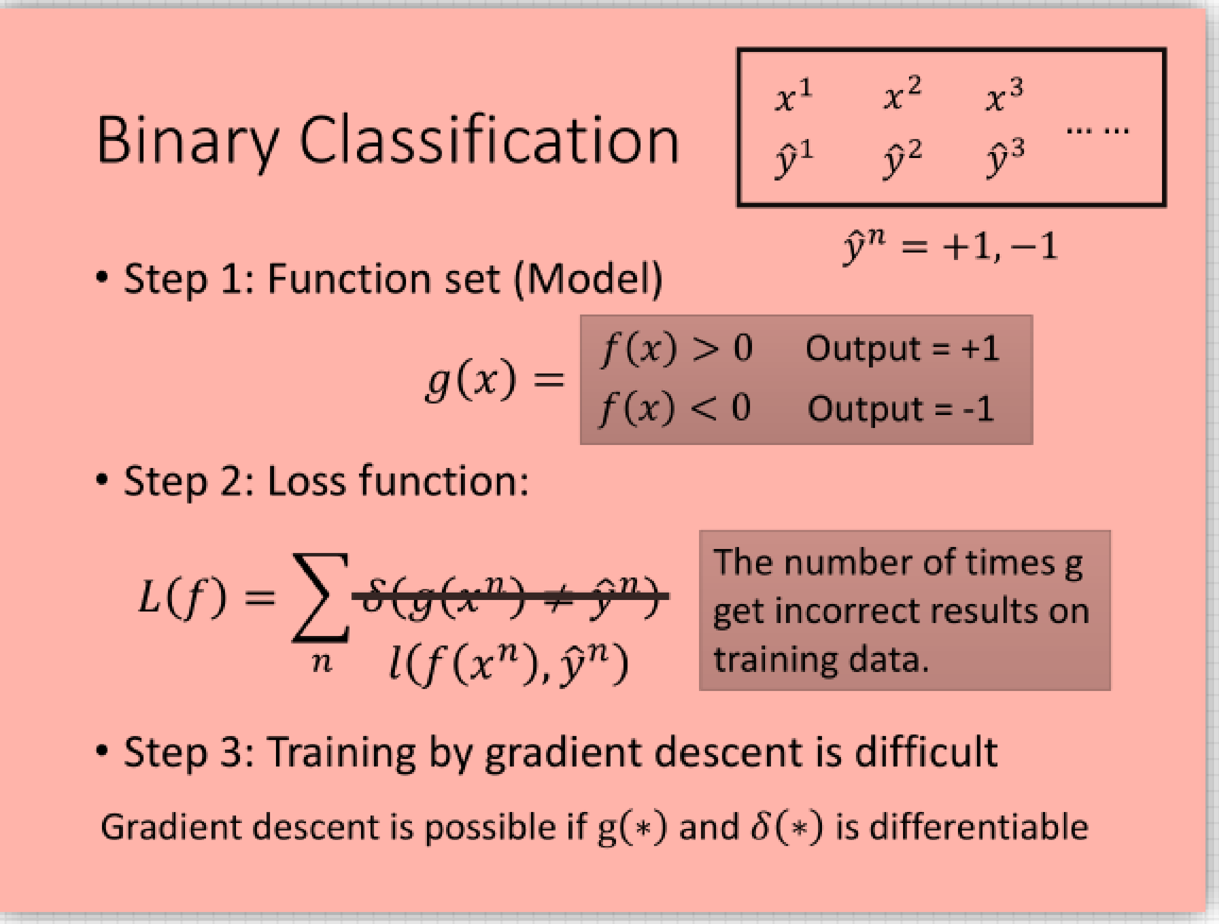
深度学习版本的SVM

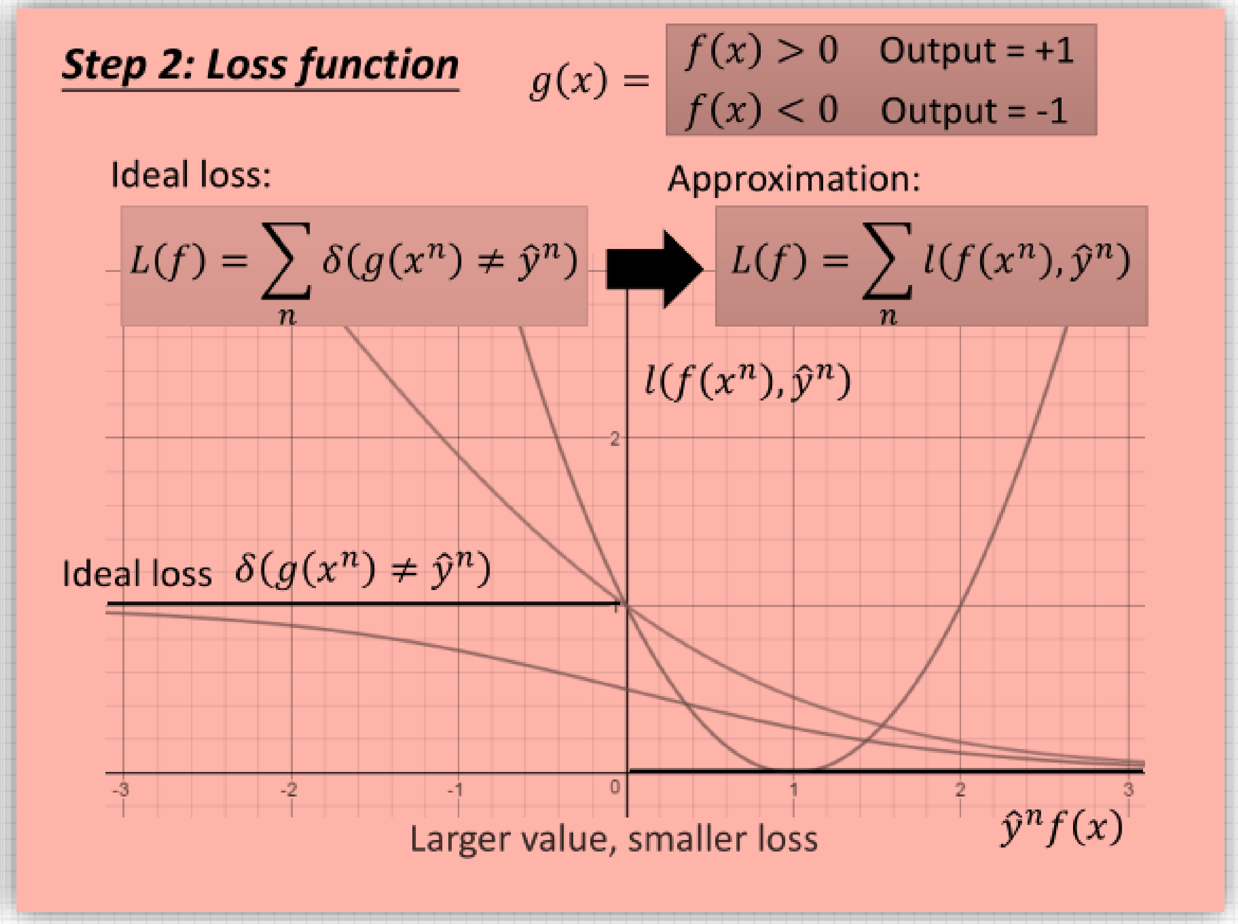
（☆☆☆）

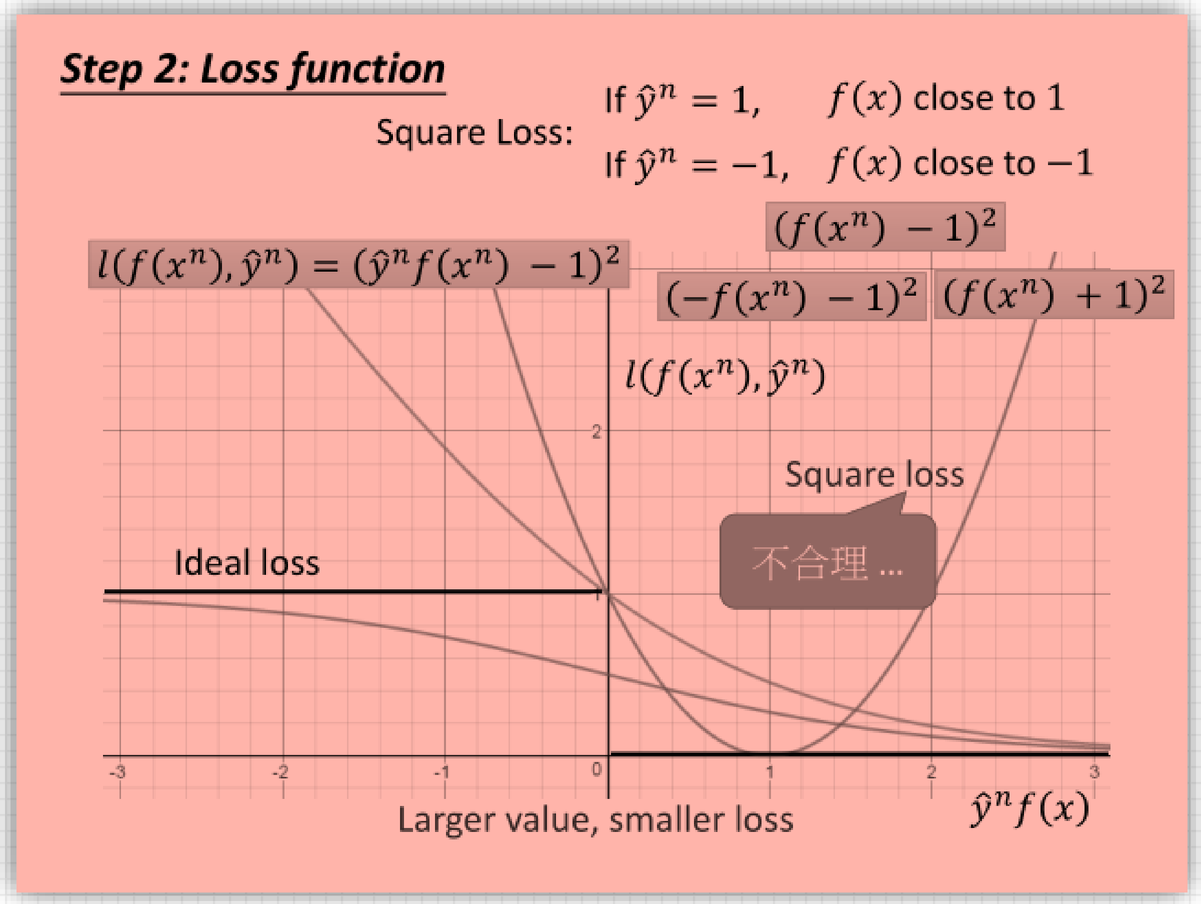
QQ：1366420642

深度学习版本的SVM，指的是使用DL中构造损失函数，利用误差反向传播，梯度下降的思路进行求解。

本节相关内容、图像来自李宏毅老师16年ML课程课件。

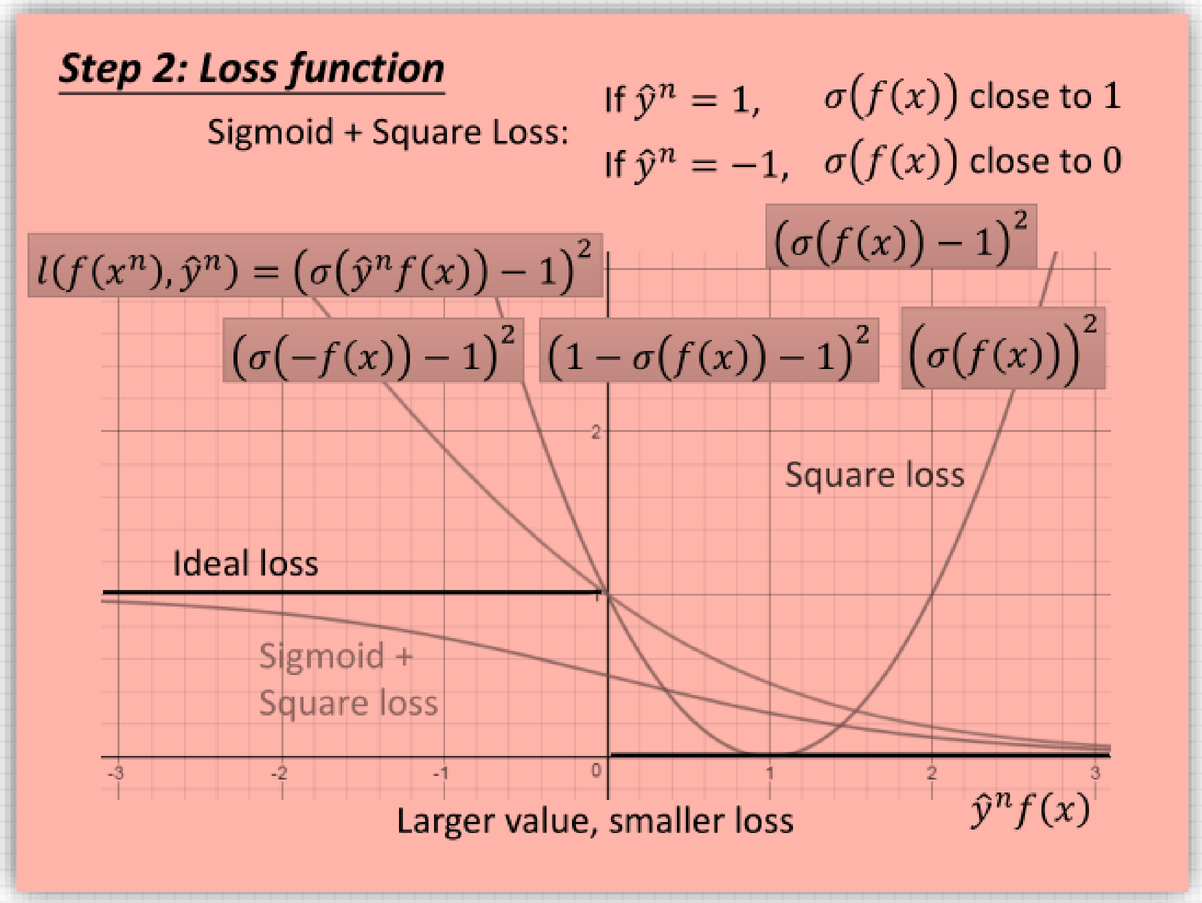




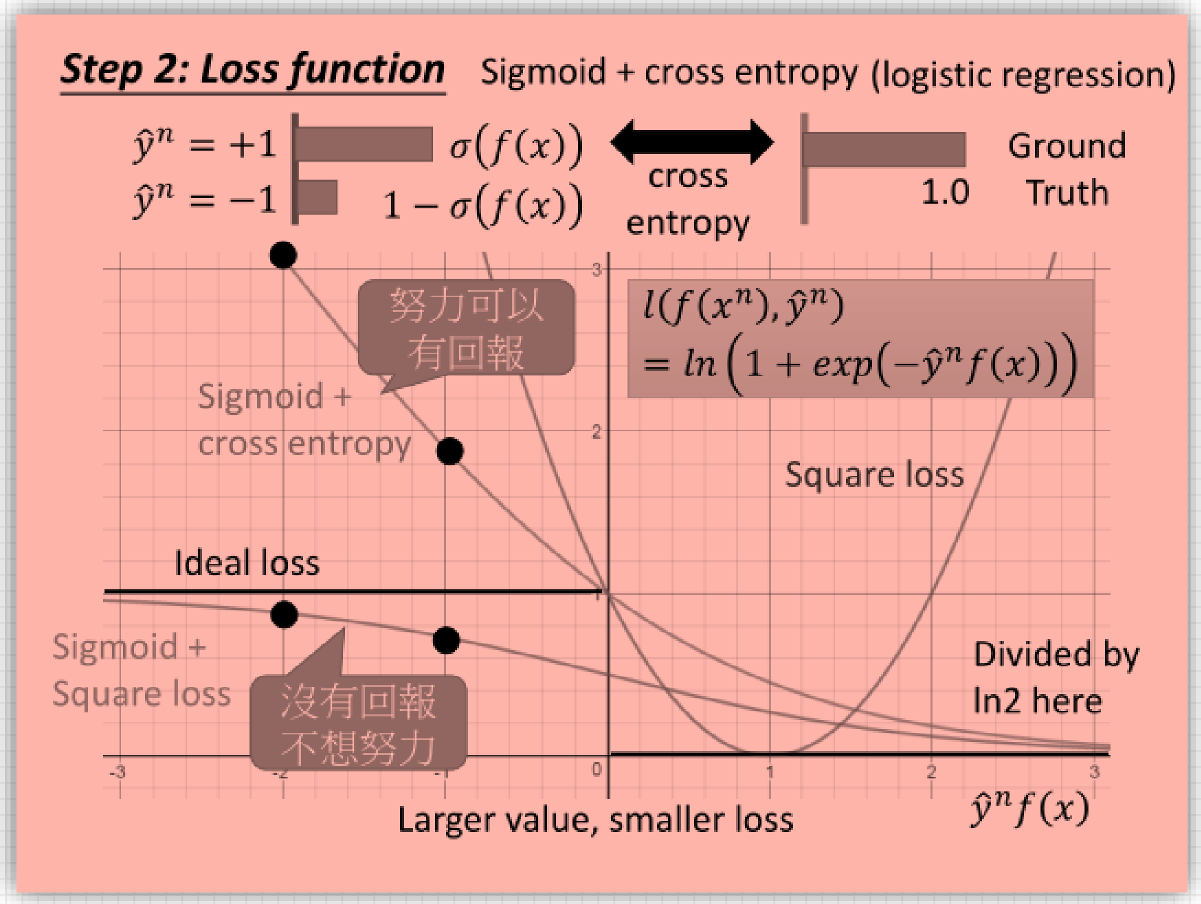


Square Loss

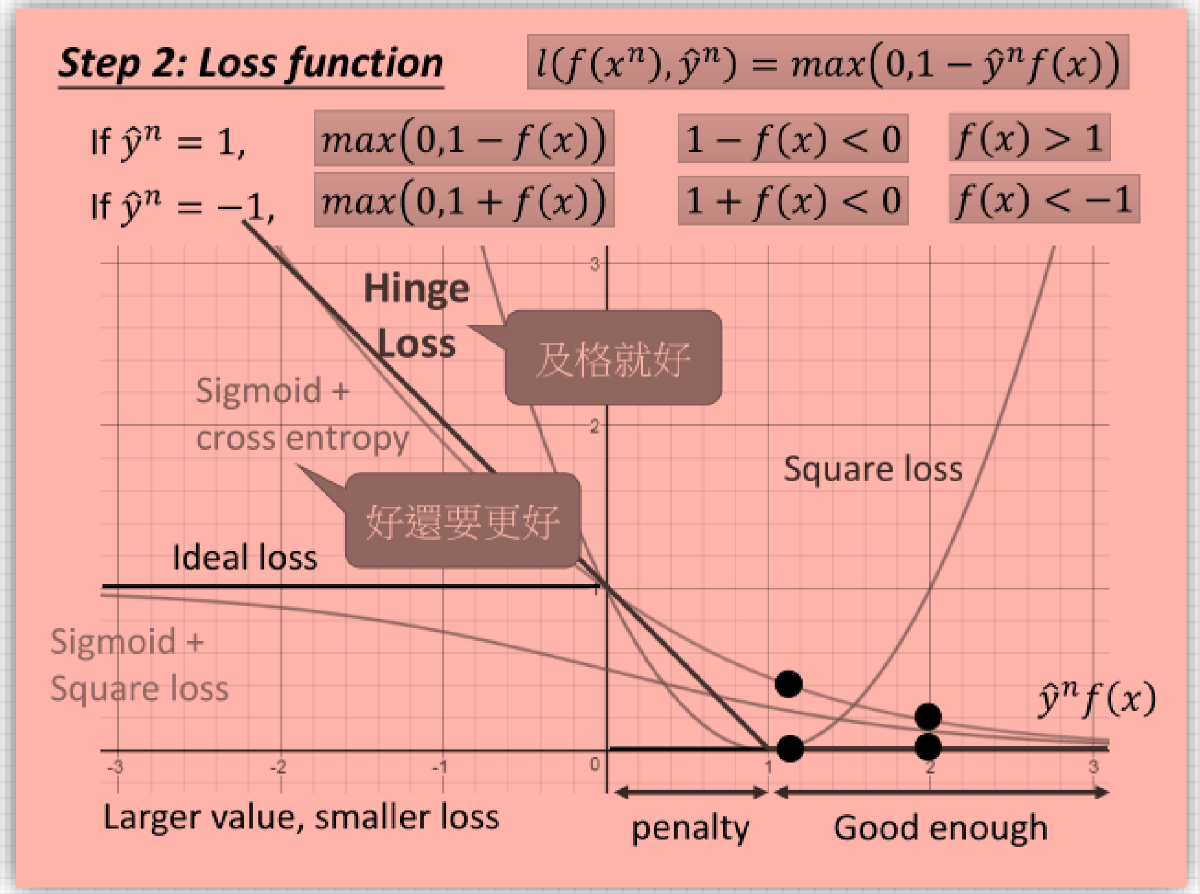
Sigmoid + Square Loss



Hinge Loss



Sigmoid + Cross Entropy



接下来证明，对于n个训练样本，如果对如下函数进行最小化，那么你得到的就是一个SVM。

$$Minimize\ \ L(f)=\sum l(f(x^n),\hat{y}^n)+\lambda\|w\|_2,\ \ l(f(x^n),\hat{y}^n)=max\left(0,1-\hat{y}^n f(x^n)\right)$$

Proof:

$$Minimize\ \ L(f)=\sum \boldsymbol{\epsilon^n}+\lambda\|w\|_2,\ \ \epsilon^n=max\left(0,1-\hat{y}^n f(x^n)\right)$$

$\epsilon^n=max\left(0,1-\hat{y}^n f(x^n)\right)$ $\nLeftrightarrow$ $\epsilon^n\geq 0,\ \ \epsilon^n\geq 1-\hat{y}^n f(x^n)$

But!!!

$\epsilon^n=max\left(0,1-\hat{y}^n f(x^n)\right)$ + ***Minimize*** $\Leftrightarrow$ $\epsilon^n\geq 0,\ \ \epsilon^n\geq 1-\hat{y}^n f(x^n)$ + ***Minimize***

$$\boldsymbol{Minimize\ \ L(f)=\sum \epsilon^n+\lambda\|w\|_2,\ \ \epsilon^n\geq 0,\ \ \hat{y}^n f(x^n)\geq 1-\epsilon^n}$$

PPT13页，软间隔SVM目标：$min\ \frac{1}{2}w^Tw+C\sum_i\xi_i,\ \ s.t.\ \ y_i(w^Tx_i+b)\geq 1-\xi_i\ \ ,\ \ \xi_i\geq 0$

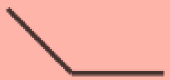
Linear SVM – gradient descent

Ignore regularization for simplicity

$$L(f)=\sum_n l(f(x^n),\hat{y}^n) \qquad l(f(x^n),\hat{y}^n)=max\left(0,1-\hat{y}^n f(x^n)\right)$$

$$\frac{\partial l(f(x^n),\hat{y}^n)}{\partial w_i}=\frac{\partial l(f(x^n),\hat{y}^n)}{\partial f(x^n)}\boxed{\frac{\partial f(x^n)}{\partial w_i}}x_i^n \qquad \boxed{\begin{array}{l}f(x^n)\\=w^T\cdot x^n\end{array}}$$

$$\frac{\partial max\left(0,1-\hat{y}^n f(x^n)\right)}{\partial f(x^n)}=\begin{cases}-\hat{y}^n & \text{If } \hat{y}^n f(x^n)<1\\ 0 & \text{otherwise}\end{cases}$$



$$\frac{\partial L(f)}{\partial w_i}=\sum_n \underbrace{-\delta(\hat{y}^n f(x^n)<1)\hat{y}^n}_{c^n(w)} x_i \qquad w_i\leftarrow w_i-\eta\sum_n c^n(w)x_i^n$$

亦可以说明SVM的权重w，是训练集样本的线性组合

Dual Representation

$$w^*=\sum_n \alpha_n^* x^n \qquad \text{Linear combination of data points}$$

α_n^* may be sparse ➡ x^n with non-zero α_n^* are support vectors

$$\left.\begin{array}{l} w_1\leftarrow w_1-\eta\sum_n c^n(w)x_1^n\\ \vdots\\ w_i\leftarrow w_i-\eta\sum_n c^n(w)x_i^n\\ \vdots\\ w_k\leftarrow w_k-\eta\sum_n c^n(w)x_k^n\end{array}\right\}$$

If w initialized as **0**

$$w\leftarrow w-\eta\sum_n c^n(w)x^n$$

$$c^n(w)=\frac{\partial l(f(x^n),\hat{y}^n)}{\partial f(x^n)}$$

Hinge loss: usually zero

c.f. for logistic regression, it is always non-zero

第 4.5 章 节

SVM与逻辑回归、回归任务、NN的关系

$$Minimize\ \ L(f)=\sum l(f(x^n),\hat{y}^n)+\lambda\|w\|_2$$

使用不同的损失函数对上式进行优化，得到的就是不同的模型。

① 当 $l(f(x^n),\hat{y}^n)=max\left(0,1-\hat{y}^n f(x^n)\right)$ 时，得到就是一个SVM。

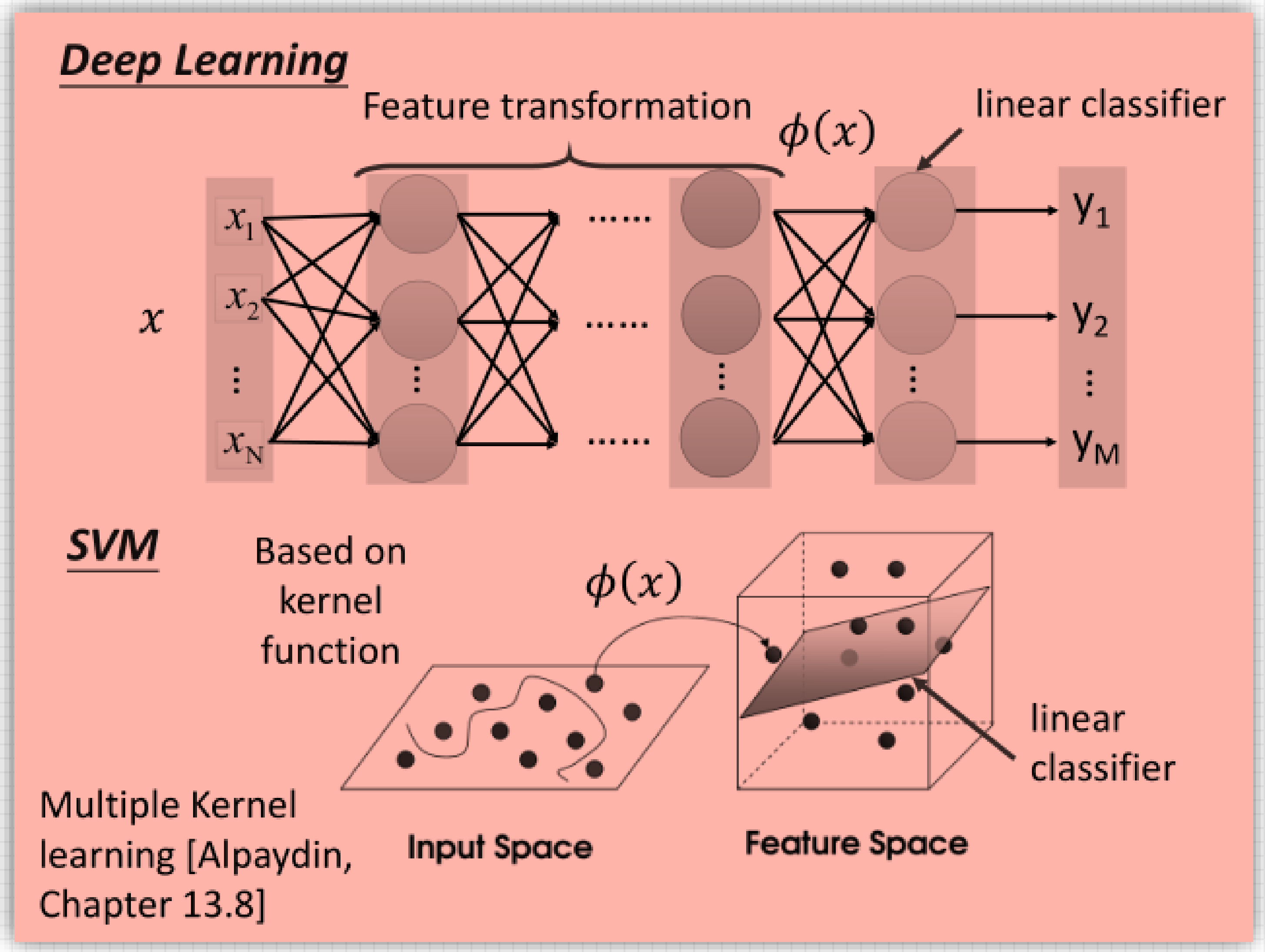
② 当 $l(f(x^n),\hat{y}^n)=\ln\left(1+\exp\left(-\hat{y}^n f(x^n)\right)\right)$ 时，得到就是一个 Logistics Regression 。

③ 当 $l(f(x^n),\hat{y}^n)=(f(x^n)-\hat{y}^n)^2$ 时，得到的就是一个带 L2 Regularization 的回归模型，也叫 岭回归。

④ 当 $l(f(x^n),\hat{y}^n)=(f(x^n)-\hat{y}^n)^2$ ，且 $+\lambda\|w\|_1$时，得到是 L1 Regularization 的回归模型，也叫 LASSO回归。

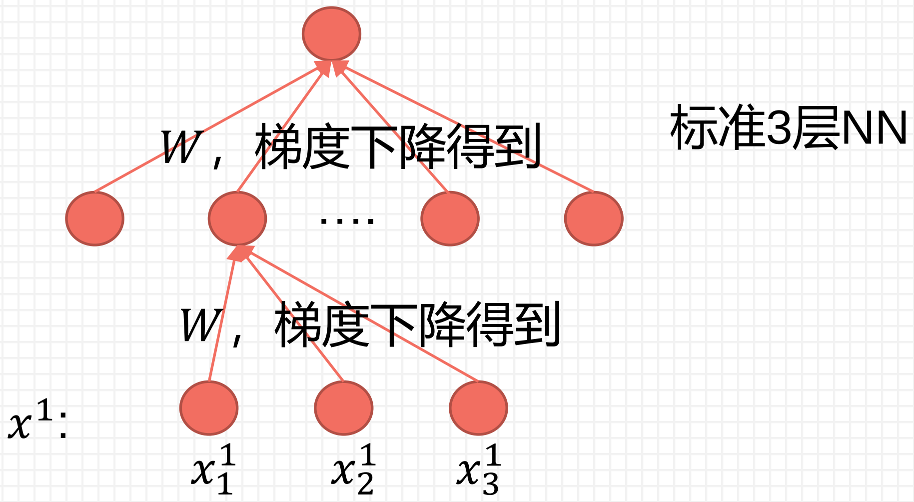
因此，从这个角度，SVM、逻辑回归、带正则项的回归，他们是一样的。区别仅在于选择什么优化函数。

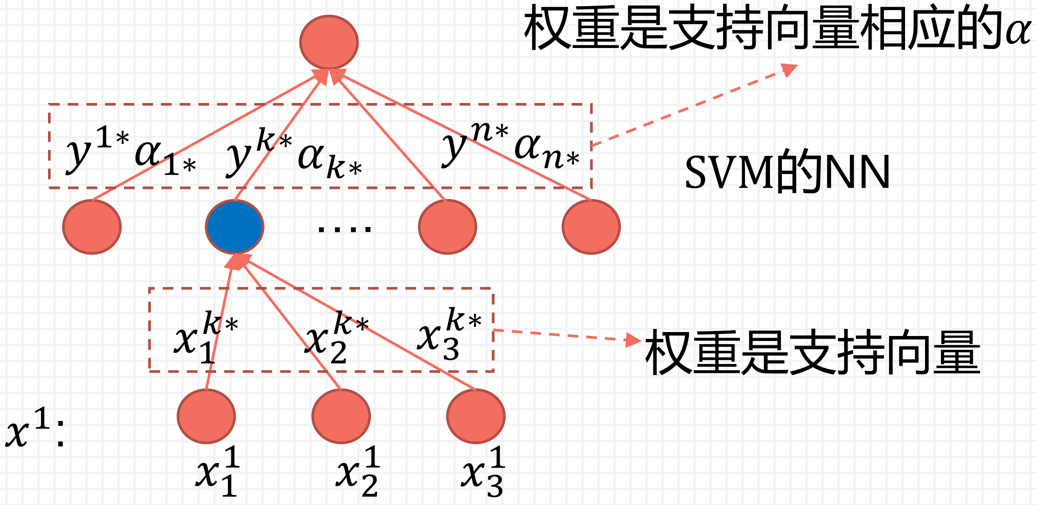
具体L1，L2正则化的原理、理解，读者自行学习。推荐一个B站UP 王木头学科学 的视频：



训练的时候：

NN的隐藏层一定程度上要做的事情和 SVM 中 $\phi(\cdot)$要做的事情类似。





$$\boldsymbol{f}(\boldsymbol{x}^{\boldsymbol{i}}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}^{\boldsymbol{i}}, \mathrm{w} = \sum \alpha_{n*} y^{n*} x^{n*}$$,忽略偏置、$\varphi(\cdot)$

$$f^i = \sum \alpha_{n*} y^{n*} (x^{n*} \cdot x^i)$$ ，x^{n*}为支持向量

预测的时候：

SVM也是可以写成单个隐含层的神经网络，其中权重是支持向量和其相应的α

di wu zhang jie

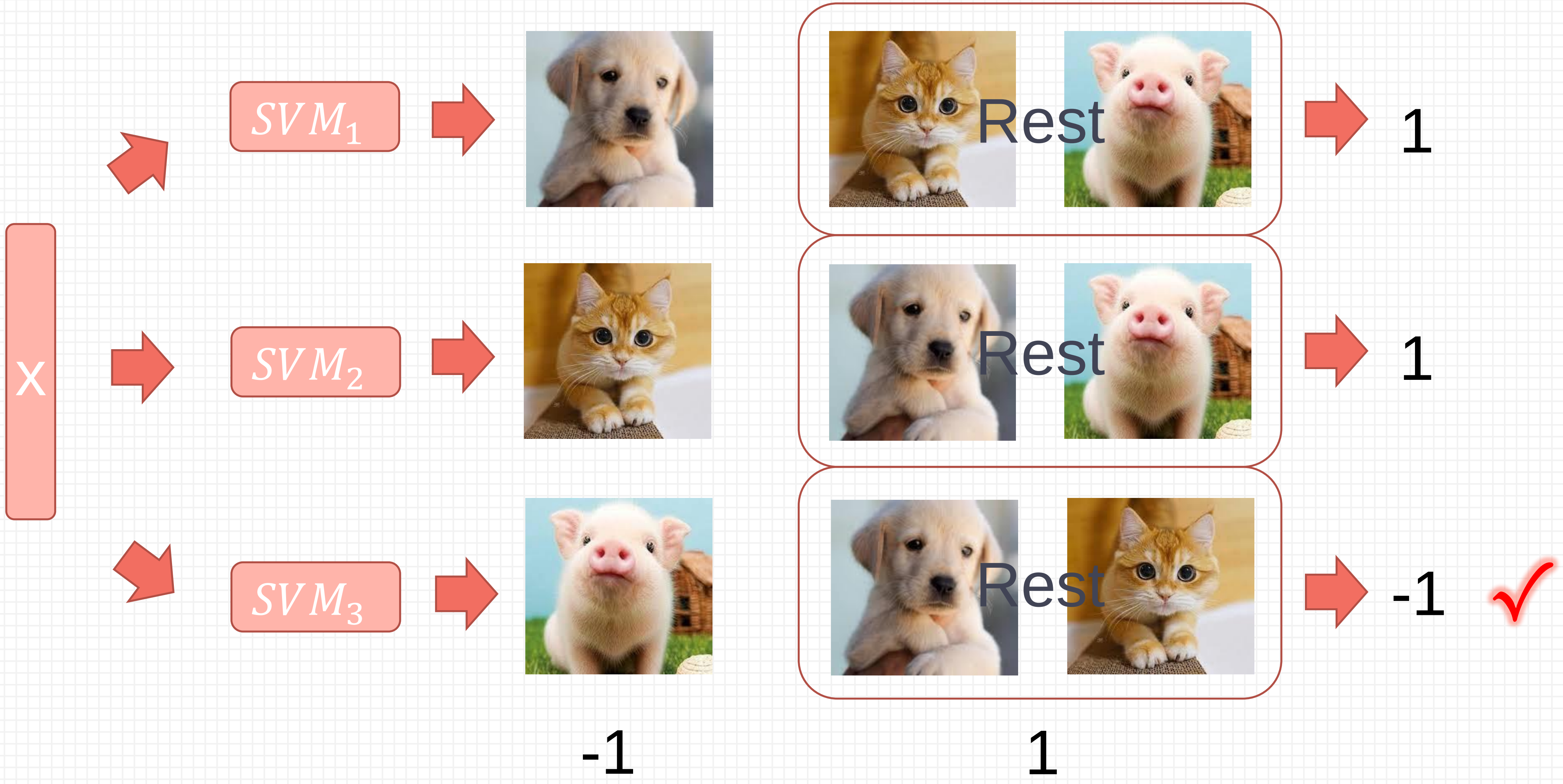
第五章节

SVM的程序调用

（☆☆☆）

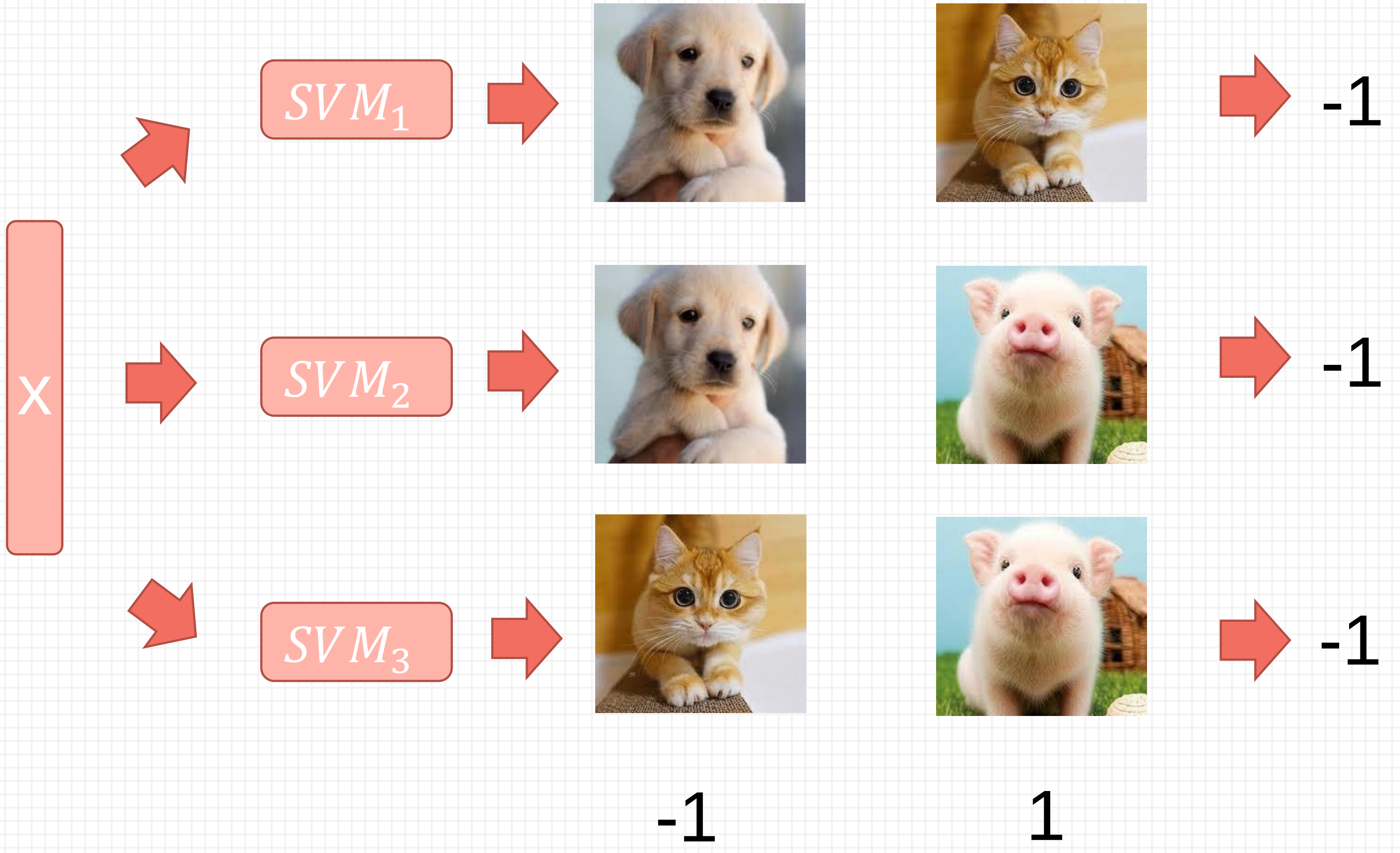
SVM怎么做多分类?

① one-vs-rest（OVR）：思路简单，高效。需要比k次，即训练k个模型，k为类别数目。



SVM怎么做多分类?

② one-vs-one（OVO）：准确，慢。需要比 $k(k-1)/2$次， 即训练个 $k(k-1)/2$ 模型，k为类别数目。



- 顺序要对应：

  比如狗、猫、猪

  最后输出才能对上：

  (-1，-1，-1) 对应

  (狗，猫)、(狗、猪)、

  (猫，猪)。

- **输出的-1含义不一样，需要额外的信息标识。**
- **最后采用投票法输出最终结果**

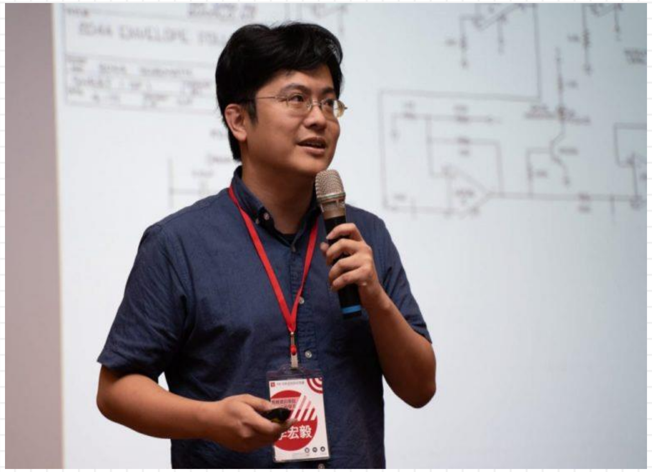
Standard SVM:

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

For Large Dataset :

https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html#sklearn.linear_model.SGDClassifier



本次课程部分图片内容来自李宏毅老师ML课程，
感谢老师的精彩课程讲解及相关开源内容。

所有课程PPT以及code下载链接（或者评论区、视频简介均可下载）：

https://github.com/CHENHUI-X/My-lecture-slides-and-code ， 如果对你有帮助，可以 

支持我。

UP知识水平有限，且PPT内容公式较多，纯手工敲写，相关内容如有错误及不足，请务必指出，感谢。

THANKS

谢谢观看