支持向量机
Support vector machine

数据应用学院

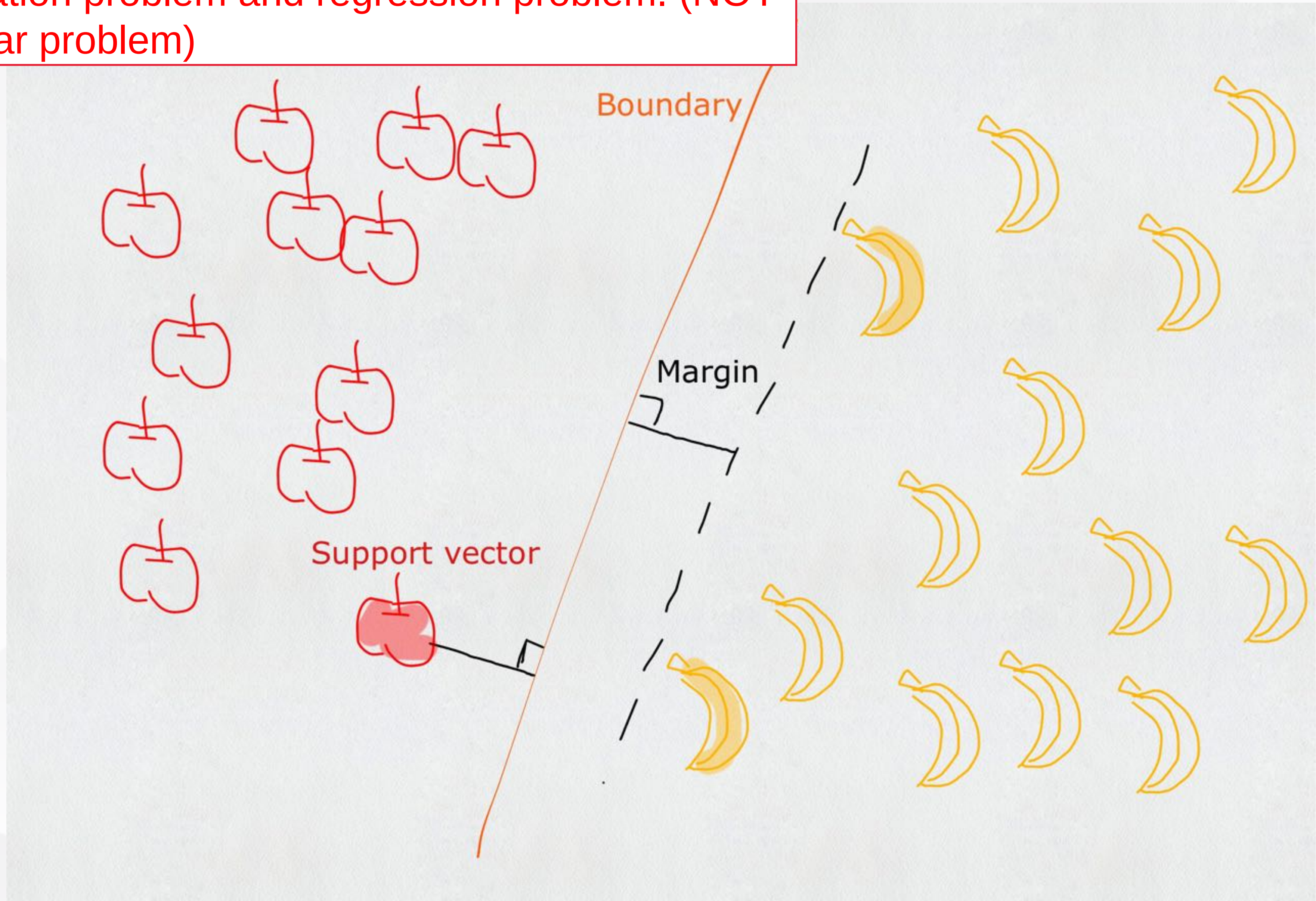
万门大学

Henry Tang

概要

- 支持向量机简介与历史
- 支持向量机分类与回归
  - 间隔
  - 支持向量
  - 对偶问题
  - KKT 条件
  - 核函数
  - 软间隔与正则化
- 代码实战

SVM=SVC(classification)+SVR(regression)
SVM - a strong algorithm, can solve both classification problem and regression problem. (NOT only linear problem)



机器学习算法

- 回归（监督学习）：
  - 线性回归 （linear regression）
  - 决策树回归 （decision tree regression）
  - 支持向量回归 （support vector regression）
  - 神经网络（neural network）

Supervised learning:
There is y (label) (dependent variable)
- Regression: continuous;
- Categorize: uncontinious

Unsupervised learning:
No y (no label)

机器学习算法

- 分类 （监督学习）：
  - 逻辑回归 （logistic regression）
  - 决策树分类（decision tree classification）
  - 支持向量机 （support vector machine）
  - 朴素贝叶斯 （naive Bayes）
- 聚类（半监督学习）：
  - K 均值 （K-means）

NOTE:
From these two slides, we can see:
Decision Tree and SVM can handle both Classification and Regression problems.

支持向量机简介

- 支持向量机与上世纪90年代正式由Vapnik和Cortes正式发表，在文本分类任务中显示出卓越性能，很快成为机器学习主流技术，直接掀起了2000年前后统计学习的高潮。
- 支持向量机思想直观，但细节复杂，涵盖凸优化，核函数，拉格朗日算子等理论
  kernel
- 支持向量机 （support vector machine） ：
  - 二分类模型，也可用于回归分析
  - 学习策略： 最大间隔化
  - 凸优化，对偶，KKT条件

线性分类

$$f(x) = w^T x + b$$

w: 权重

b: 截距（位移）

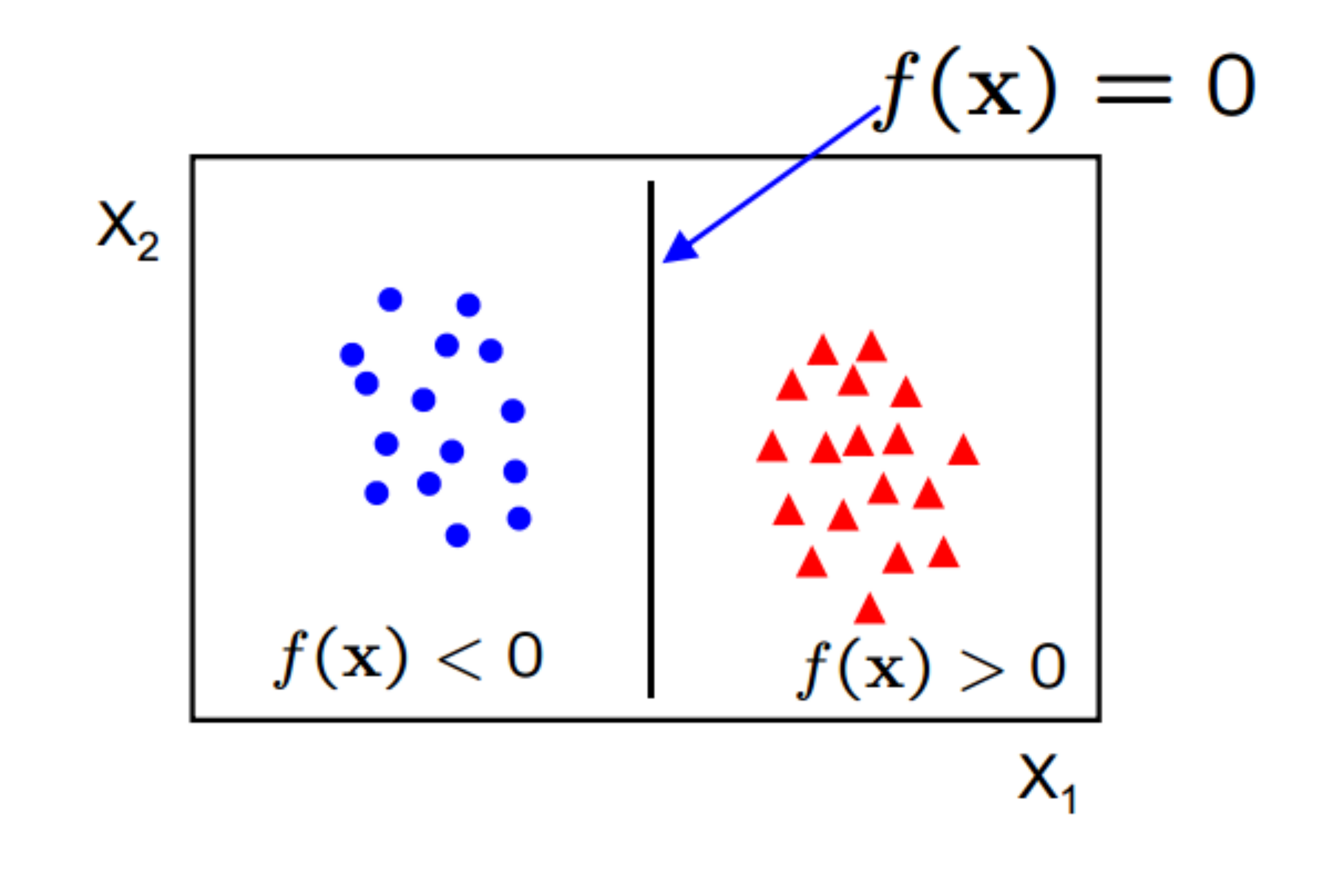
训练数据用来训练得到权重和截距

分类算法的起源： 逻辑回归

$$h_\theta(x) = \frac{1}{1 + e^{-f(x)}}$$

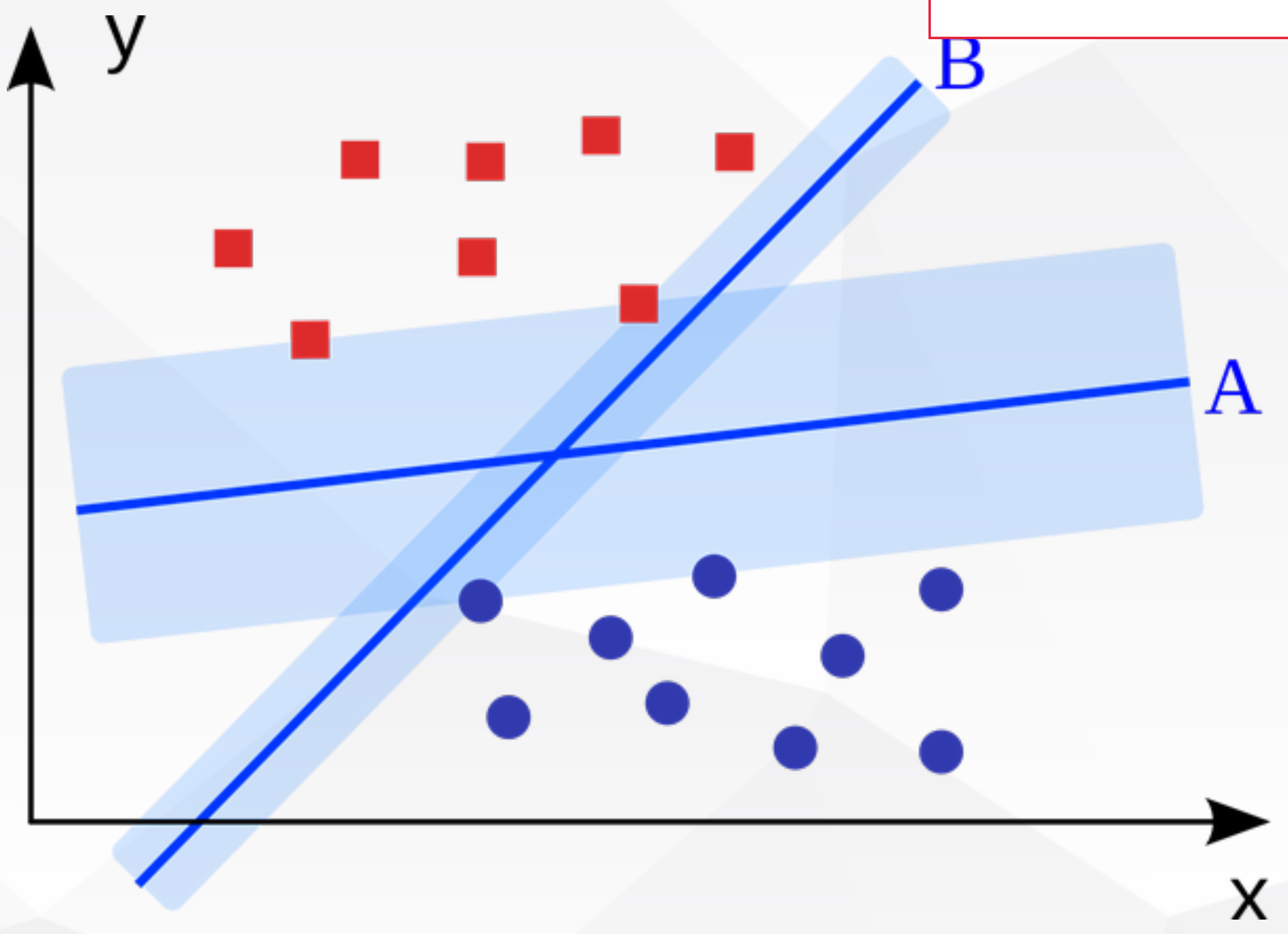
$$P(y = 1|x;\theta) = h_\theta(x)$$

$$P(y = 0|x;\theta) = 1 - h_\theta(x)$$



如何决定最好的参数：w & b

A is better than B;
Because A can separate two classes better-- larger margin, less effected by noise



支持向量机的核心思想： 最大间隔化， 最不受到噪声的干扰

支持向量机

Please refer to the word document called "SVM intro from MIT" for detailed explanation of SVM from MIT professor

- SVM划分的超平面：

$$w^T x + b = 0$$

$\boldsymbol{w} = (w_1, w_2, \dots, w_d)$ 为法向量，决定超平面方向

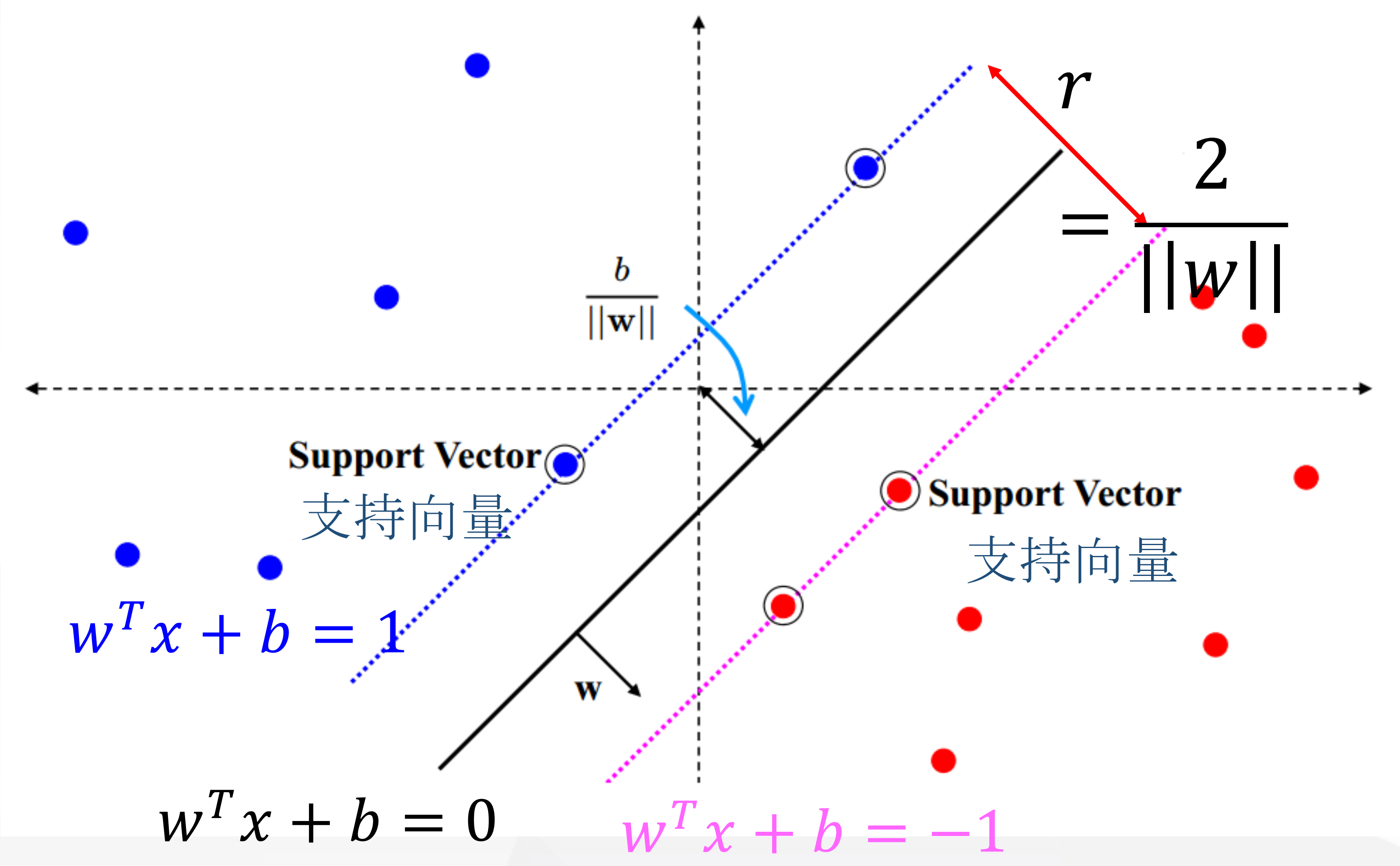
- 假设超平面将样本正确划分

$$\begin{cases} w^T x + b \geq 1, y = +1 \\ w^T x + b \leq -1, y = -1 \end{cases}$$

- 距离超平面最近的几个点叫做支持向量

- 间隔（margin）：

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



支持向量机

- 支持向量机： 最大间隔化

- 约束条件:

$$\max_{w,b} \frac{2}{||w||}$$

$$s.t.\ y_i(w^T x_i + b) \geq 1$$

$(x_i, y_\mathrm{i}) \in D$ 样本空间

等价于

$$\min_{w,b} \frac{1}{2} ||w||^2$$

$$s.t.\ y_i(w^T x_i + b) \geq 1$$

$(x_i, y_\mathrm{i}) \in D$ 样本空间

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, y_i \in \{-1, +1\}$$

支持向量机的最基本表达形式， 是一个线性约束的凸二次规划问题， 有最优解

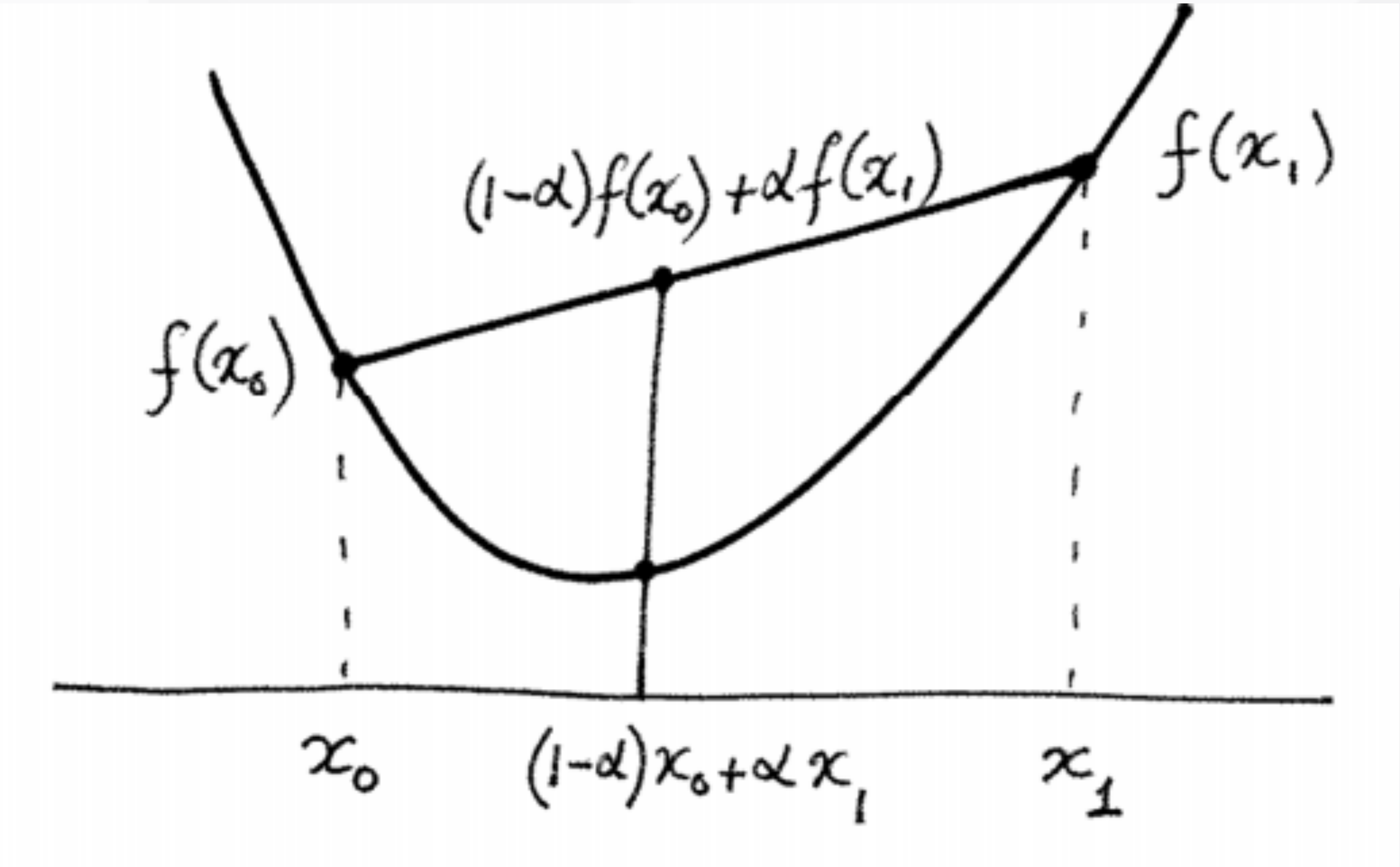
凸优化

凸函数：

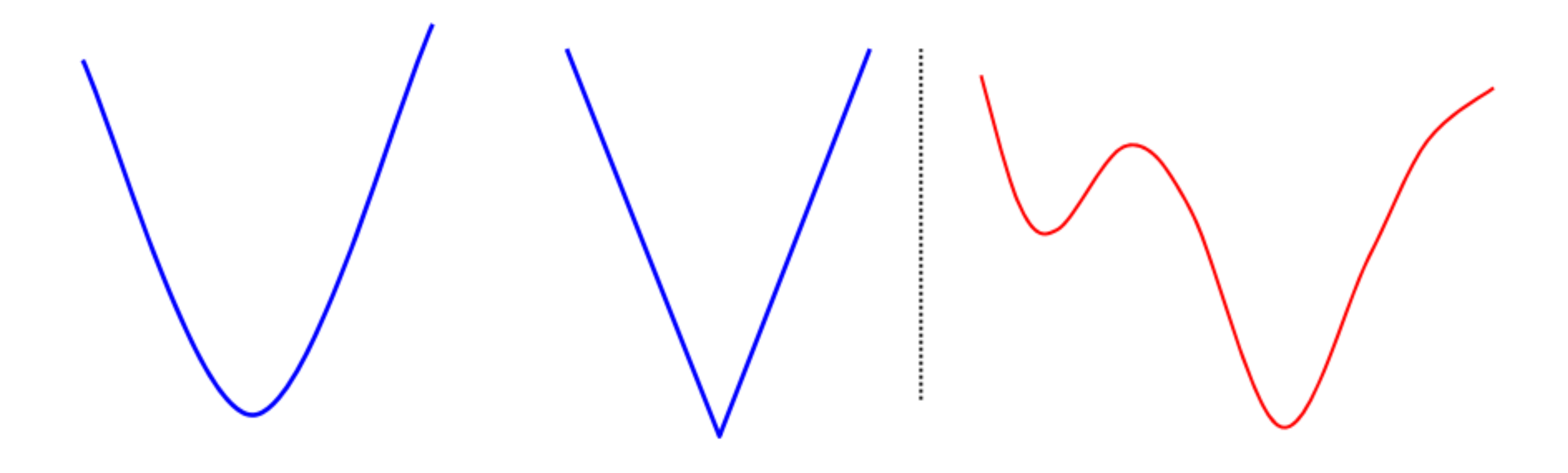
设 *f(x)* 为定义在n维欧式空间中某个凸集 *S* 上的函数，若对于任何实数α $(0 < \alpha < 1)$ 以及 *S* 中的不同两点 *x*, *y*, 均有：

$$f\big((1-\alpha)x_0 + \alpha x_1\big) \leq (1-\alpha)f(x_0) + \alpha f(x_1)$$

那么， $f(x)$ 为定义在凸集 S 上的凸函数



凸函数



凸函数　　　　　　非凸函数

凸优化

常见的凸函数：

- 仿射函数（affine function）：$w^T x + b$
- 二次函数（quadratic function）：$x^T A x + bx + c$ （A 为半正定矩阵）
- 最小平方差函数（least square）：$||y - Ax||_2^2$ （$A^T A$ 总是半正定矩阵）
- max 函数： $\max(x_1, x_2, \ldots, x_n)$

凸优化

- 有约束条件的凸优化问题：

$$\min_x \ f(x)$$
$$s.t. \ g_i(x) \leq 0, i = 1, 2, \dots, m$$
$$h_j(x) \ = 0, j = 1, 2, \dots, n$$

如果 f(x), g(x) 为凸函数，h(x) 为仿射函数时， 这是一个凸优化的问题。

对于支持向量机：

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

SVM 是一个凸二次规划问题， 有最优解

拉格朗日对偶

VIP: (see this page and after) - Lecture 15.5 !!
SVM's boundary conditions satisfy KKT conditions, then original lagrange equation has "duality" property due to KKT. Thus we can get extreme of lagrange (prime problem) = extreme of its dual problem. Its dual problem is easier to solve in math. (Dual problem's equation can be found in P 21 of PDF, it is the same as MIT's notes-eq'n 5(word doc)).

- 拉格朗日乘子法（Lagrange multipliers）
  - 一种寻找多元函数在一组约束条件下的极值的方法。通过引入拉格朗日乘子，可将有d个变量和k个约束条件的最优化问题转化为d+k个变量的无约束优化问题求解。
  - 将原问题的约束规划问题转化为对偶问题，易于求解

拉格朗日乘子：等式约束

对于等式约束问题：

$$\min f(x)$$
$$s.t.\ g_i(x) = 0$$

那么必然有：

1) 约束曲面上的任一点x, 该点的梯度$\nabla g(x)$ 正交于约束曲面

2) 在最优点x^*,$\nabla f(x^*)$ 也正交于曲面

则存在$\lambda \neq 0$:

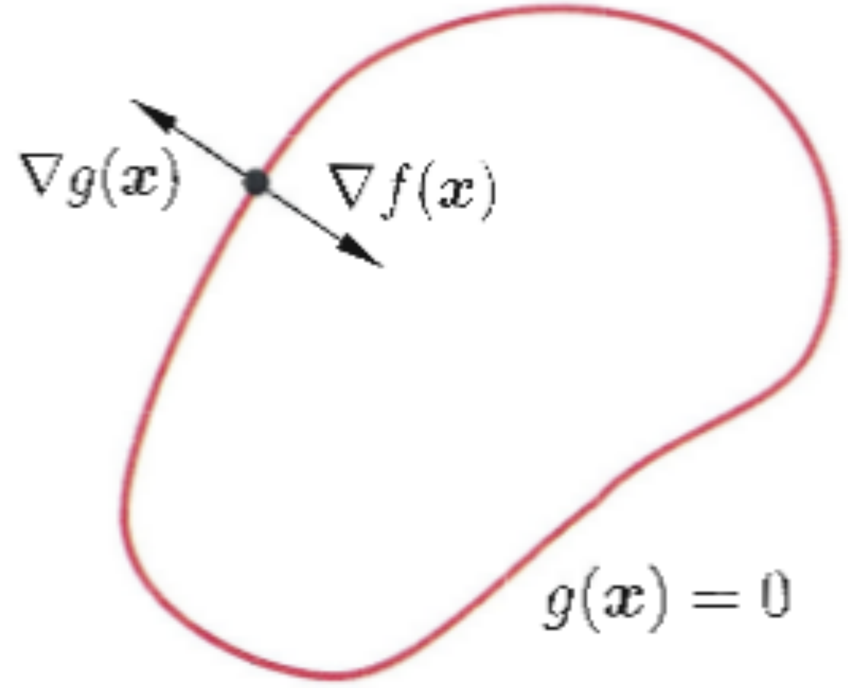
$$\nabla f(x^*) + \lambda \nabla g(x^*) = 0$$

可定义拉格朗日函数, λ 为拉格朗日乘子：

$$L(x,\lambda) = f(x) + \lambda g(x)$$

同时：

$$\nabla_x L(x,\lambda) = 0\ ;\quad \nabla_\lambda L(x,\lambda) = 0$$



(a) 等式约束

拉格朗日乘子：不等式约束

对于不等式约束问题：

$$\min f(x)$$

$$s.t.\ g_i(x) \leq 0$$

1) $g(x) < 0$, 约束不起作用，直接求解 $\nabla f(x) = 0, \lambda = 0$
2) $g(x) = 0, \nabla f(x^*) + \lambda \nabla g(x^*) = 0,\ \lambda > 0$

那么拉格朗日函数为：

$$L(x,\lambda) = f(x) + \lambda g(x)$$

同时需要满足的条件为：

$$\nabla_x L(x,\lambda) = 0$$

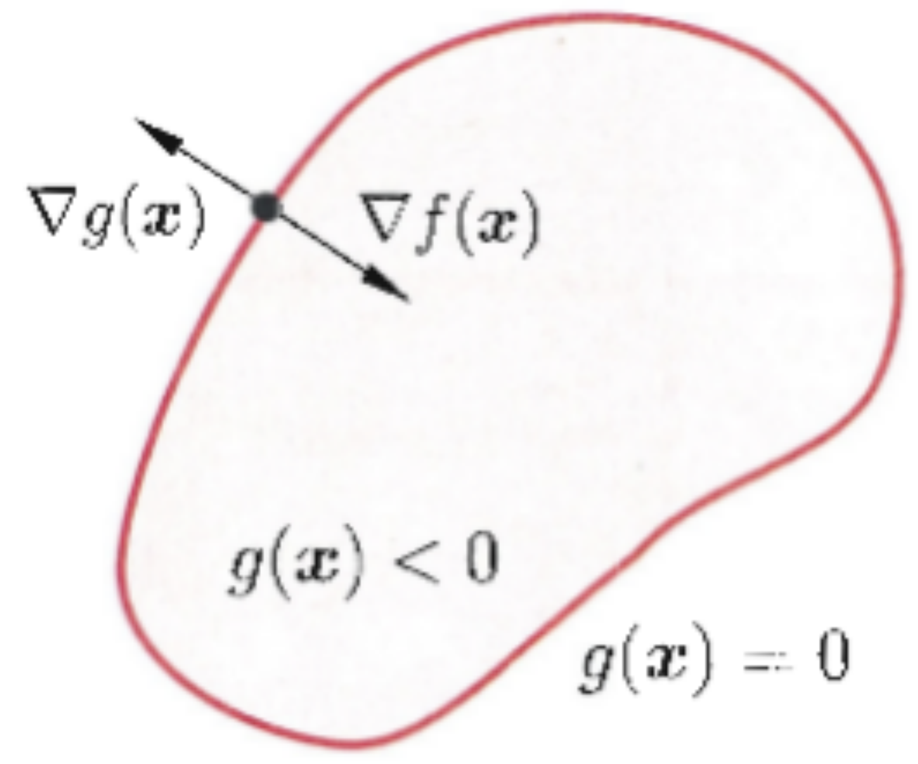
$$\nabla_\lambda L(x,\lambda) = 0$$

$$\lambda g(x) = 0$$

$$\lambda \geq 0$$

即Karush-Kuhn-Tucker (KKT) 条件

$$g(x) \leq 0$$



(b) 不等式约束

拉格朗日对偶

- 广义拉格朗日函数：

$$\min_{x} \ f(x)$$

$$s.t.\ g_i(x) \leq 0, i = 1, 2, \dots, n$$

$$h_j(x) \ = 0, j = 1, 2, \dots, m$$

$$L(x, u, \lambda) = f(x) + \sum_{i=1}^{n} \mu_i g_i(x) + \sum_{i=1}^{m} \lambda_j h_j(x)$$

满足**KKT** 条件：

$$\begin{cases} \nabla_{x,\mu,\lambda} L(x, \mu, \lambda) = 0 \\ g_i(x) \leq 0 \\ \mu_i \geq 0 \\ \mu_i g_i(x) = 0 \end{cases}$$

拉格朗日对偶

构造拉格朗日乘子：

$$\min_{x} \ f(x)$$

$$s.t. \ g_i(x) \leq 0, i = 1, 2, \dots, n$$

$$h_j(x) \ = 0, j = 1, 2, \dots, m$$

$$L(x, u, \lambda) = f(x) + \sum_{i=1}^{n} \mu_i g_i(x) + \sum_{i=1}^{m} \lambda_j h_j(x) \ \ (\mu_i \geq 0)$$

主问题（primal problem）：$p^* = \min_{x} \max_{\mu,\lambda} L\ (x, \mu, \lambda)$

其对偶问题（dual problem）：$d^* = \max_{\mu,\lambda} min_{x} L\ (x, \mu, \lambda)$

拉格朗日对偶

通常： $d^* \le p^*$ 即对偶问题的最优解是原始主问题最优解的下限（弱对偶性）

但若满足KKT 条件的方程组的解：

$$\nabla_{x,\mu,\lambda} L(x,\mu,\lambda) = 0$$
$$g_i(x) \le 0$$
$$\mu_i \ge 0$$
$$\mu_i g_i(x) = 0$$

此时 $d^* = p^*$ 强对偶性，原始问题和对偶问题最优解严格相等

支持向量机：利用拉格朗日对偶和KKT条件求解的经典对偶问题

支持向量机的对偶问题

支持向量机最大间隔化下的损失函数：

$$\min_{w,b} \frac{1}{2}||w||^2$$

$$s.t.\, y_i(\boldsymbol{w}^T\boldsymbol{x_i}+b) \geq 1, i=1,2,3,\ldots,m$$

那么约束可以写为：

$$g_i(\boldsymbol{w}) = 1 - y_i(\boldsymbol{w}^T x_i + b) \leq 0$$

构造拉格朗日函数：

$$L(\boldsymbol{w},b,\boldsymbol{\lambda}) = \frac{1}{2}||\boldsymbol{w}||^2 + \sum_{i=1}^{m} \lambda_i(1 - y_i(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x_i} + b))$$

找到对偶问题的形式：

$$\nabla_{\boldsymbol{w}} L(\boldsymbol{w},b,\boldsymbol{\lambda}) = 0 \quad \Rightarrow \quad \boldsymbol{w} = \sum_{i=1}^{m} \lambda_i y_i \boldsymbol{x_i}$$

$$\nabla_b L(\boldsymbol{w},b,\boldsymbol{\lambda}) = 0 \quad \Rightarrow \quad 0 = \sum_{i=1}^{m} \lambda_i y_i$$

支持向量机的对偶问题

那么代入拉格朗日函数可以得到其对偶问题：

$$\max_{\lambda} W(\lambda) = \max_{\lambda}\min_{w,b} L(w,b,\lambda) = \sum_{i=1}^{m}\lambda_i - \frac{1}{2}\sum_{i=1}^{m} y_i y_j \lambda_i \lambda_j \boldsymbol{x}_i^T \boldsymbol{x}_j$$

Same as Eq'n 5 from MIT notes

$$s.t. \sum_{i=1}^{m} \lambda_i y_i = 0$$

$$\lambda_i \geq 0, i = 1, 2, \dots, m$$

上述过程满足KKT条件：

$$\lambda_i \geq 0$$

$$1 - y_i f(\boldsymbol{x}_i) \leq 0$$

$$\lambda_i\left(1 - y_i f(\boldsymbol{x}_i)\right) = 0$$

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b = \sum_{i=1}^{m} \lambda_i y_i \boldsymbol{x}_i^T \boldsymbol{x} + b$$

Same as Eq'n 6 from MIT notes

支持向量

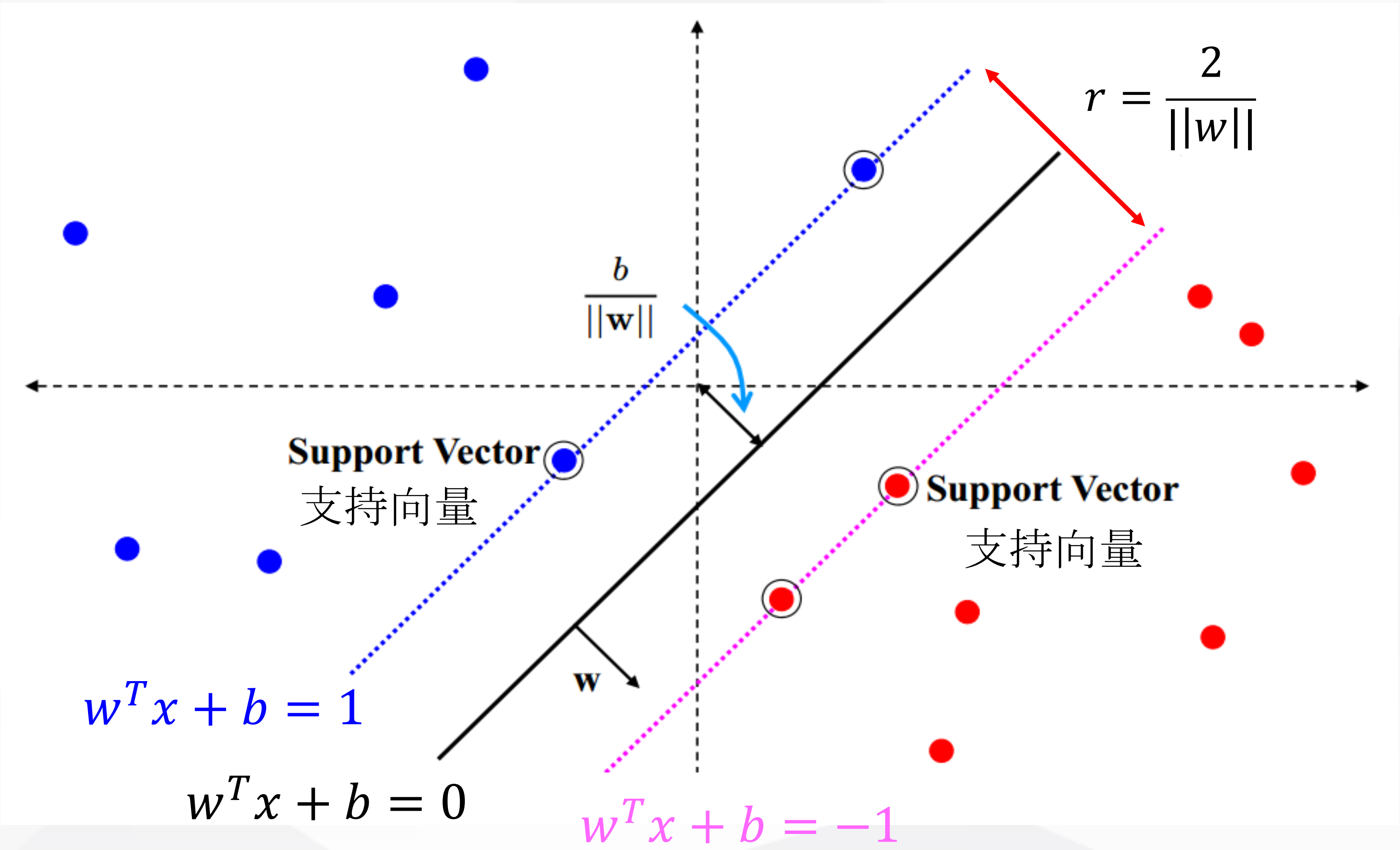
$$f(x) = w^T x + b = \sum_{i=1}^{m} \lambda_i y_i \boldsymbol{x}_i^T \boldsymbol{x} + b$$

$$\lambda_i \geq 0$$

$$1 - y_i f(x_i) \leq 0$$

$$\lambda_i\big(1 - y_i f(x_i)\big) = 0$$

如果 $y_i f(x_i) > 1$, 那么$\lambda_i = 0$, 对于$f(x)$没有贡献



SVM 模型与支持向量紧密相关！

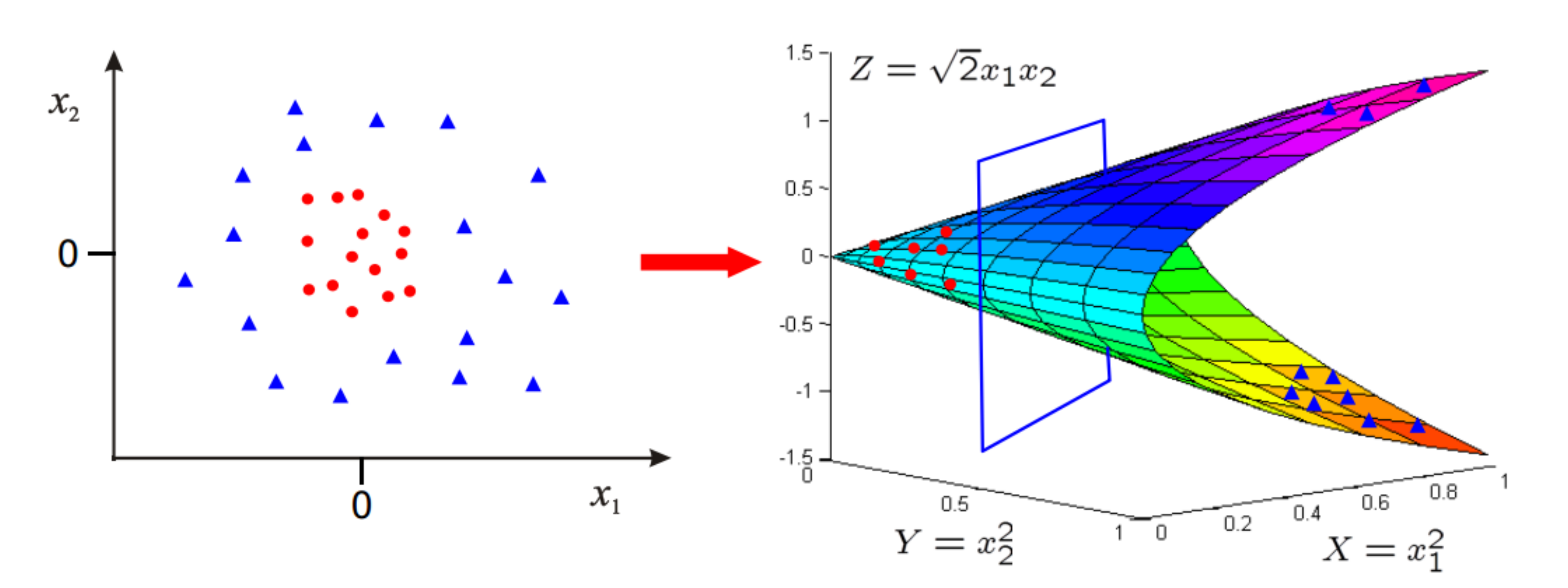
2. ALL vectors that have contributions to SVM model can be named "SV"-- support vectors.
For cases where all of the points can be 100% separated, SV = points that falls on margin(boundaries) (i.e. no points fall into upper and lower boundaries)
But not all of the points can be separated 100%, or maybe we don't want to overfit, in this case, we introduce "soft margin" definition (P28), then SV=points inside and on soft boundaries AND points that are miss

1. From KKT condition and f(x) function above we can see:
For points (vectors) that is outside of the boundary(...>1), then to satisfy KKT, namta = 0, sub to equation f(x), found that vector xi*vector xj has no contribute to f(x) --> So the conclusion is that: f(x) (SVM model) ONLY depend on vectors that is on boundaries(margins) in this case(see pt2 for other cases)

核函数（Kernel）

假设有个映射函数从二维空间映射到三维空间：

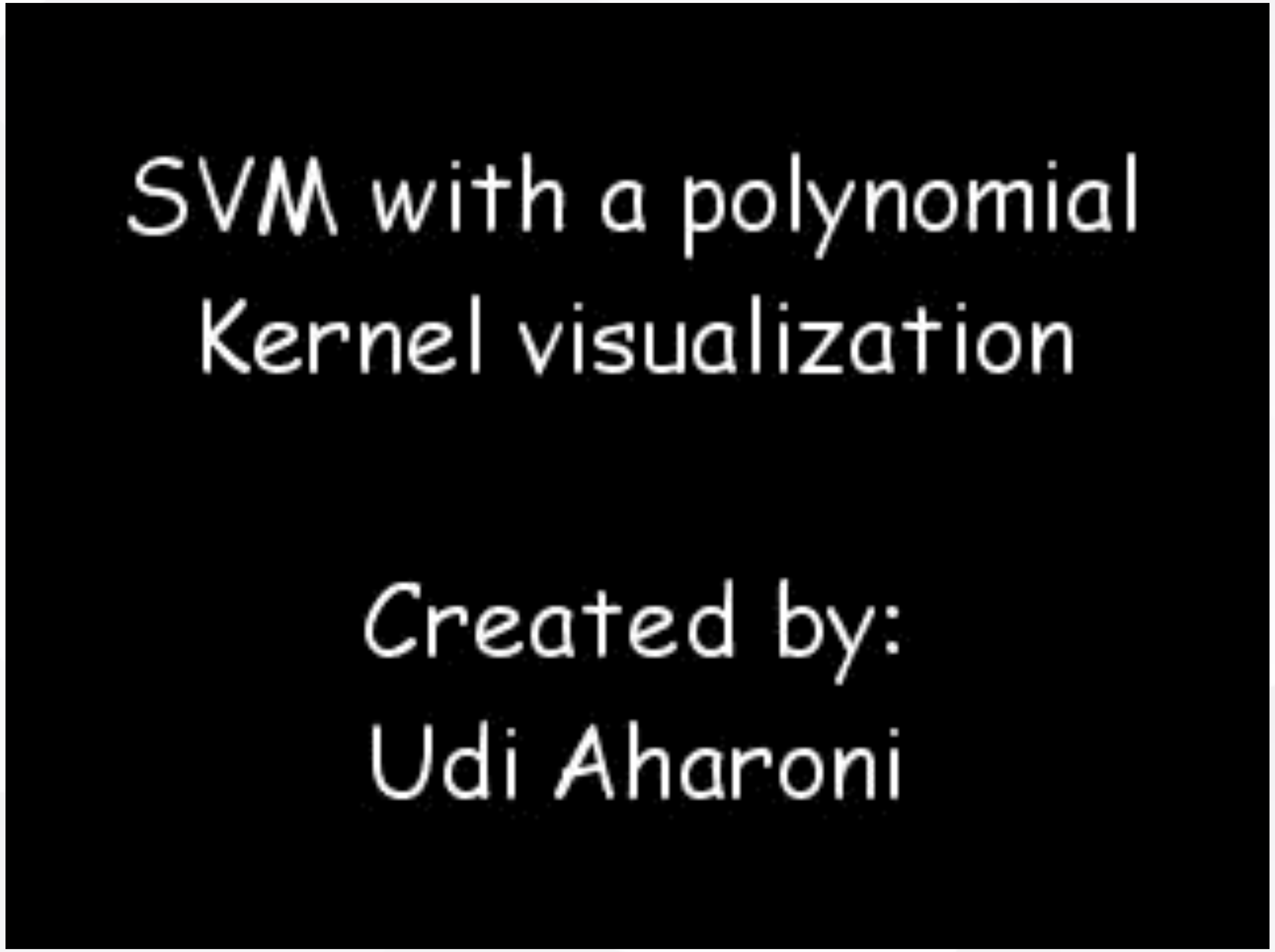
$$\Phi:\begin{pmatrix}x_1\\x_2\end{pmatrix}\rightarrow\begin{pmatrix}{x_1}^2\\{x_2}^2\\\sqrt{2}\,x_1x_2\end{pmatrix}\qquad \mathbb{R}^2\rightarrow\mathbb{R}^3$$



低维映射到高维，非线性核函数

核函数

支持向量机核函数的可视化

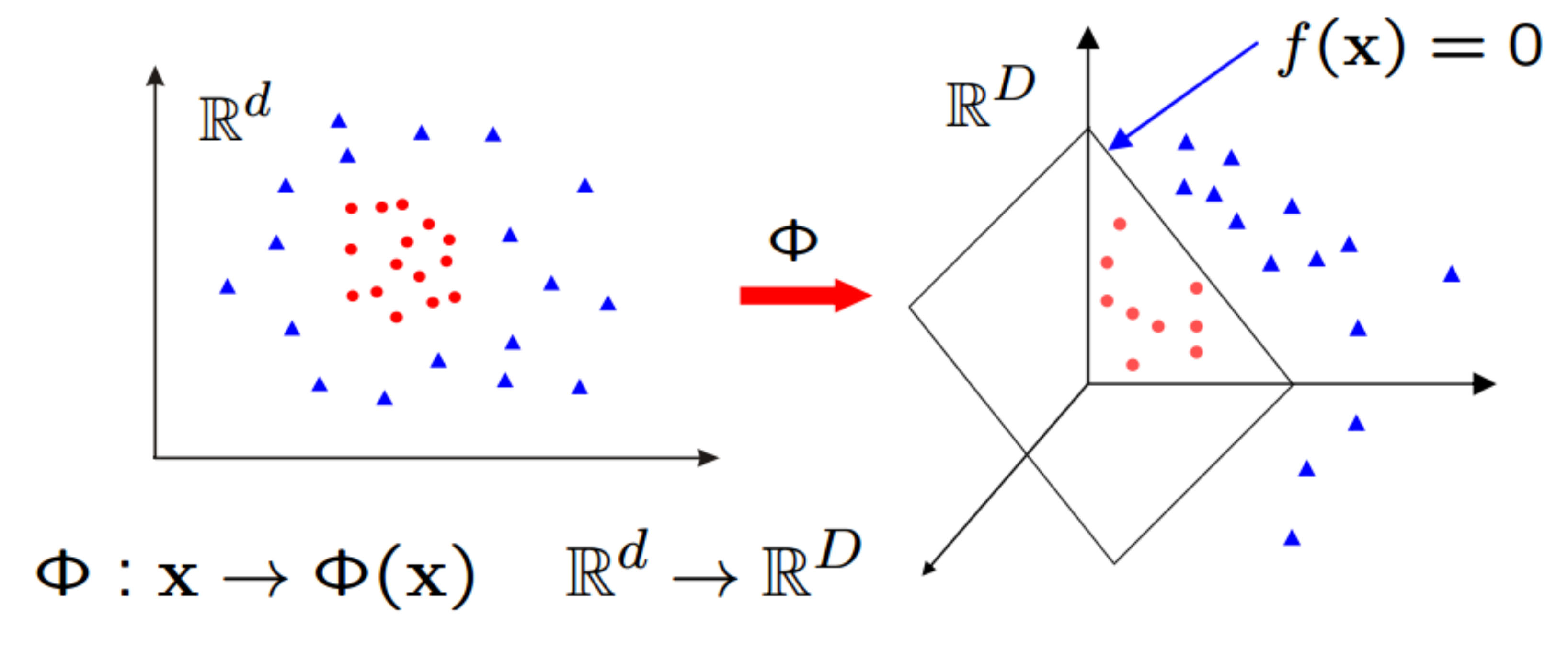


https://www.youtube.com/watch?v=3liCbRZPrZA

核函数

- 映射函数：
- $f(x) = \boldsymbol{w}^{\boldsymbol{T}}\phi(x) + b$
- 优化问题变为:

$$\min_{w,b} \frac{1}{2}\left||w|\right|^2$$
$$s.t. y_i(w^T\phi(x_i) + b) \geq 1$$



对偶问题为:

$$\max_{\lambda} W(\lambda) = \sum_{i=1}^{m} \lambda_i - \frac{1}{2}\sum_{i=1}^{m} y_i y_j \lambda_i \lambda_j \phi(\boldsymbol{x_i})^{\boldsymbol{T}}\phi(\boldsymbol{x_j})$$
$$s.t. \sum_{i=1}^{m} \lambda_i y_i = 0$$
$$\lambda_i \geq 0, i = 1, 2, \ldots, m$$

核函数

单独计算映射函数$\boldsymbol{\phi(x)}$以及$\boldsymbol{\phi(x_i)^T\phi(x_j)}$十分困难，现设想构造核函数

比如：

$$\Phi: \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} {x_1}^2 \\ {x_2}^2 \\ \sqrt{2}\, x_1 x_2 \end{pmatrix} \qquad \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$\begin{aligned} \Phi(x)^T\Phi(z) &= (x_1^2, x_2^2, \sqrt{2}\, x_1 x_2)\begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2}\, z_1 z_2 \end{pmatrix} \\ &= x_1^2 x_1^2 + x_1^2 x_1^2 + 2x_1 x_1 x_1 x_1 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= (\boldsymbol{x}^T \boldsymbol{z})^2 \end{aligned}$$

The beauty of Kernel function is that we don't need to know the transformation function from a dimension to another dim. We only need to know the final form of transformed dot product (i.e. phy xi * phy xj

$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

- 训练器不用学习和计算映射函数的显性表达式
- 寻找到核函数，然后在原始样本空间做内积，比如 $\boldsymbol{k(x,z) = \left(x^T z\right)^2}$

核函数

- 线性核函数： $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$

- 多项式核函数： $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^T \boldsymbol{x}_j)^d$ ， 当 d > 0

- 高斯核函数： $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{(-\frac{|x_i - x_j|^2}{2\sigma^2})}$ σ > 0

- 拉普拉斯核函数： $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{(-\frac{|x_i - x_j|}{2\sigma^2})}$ σ > 0

- Sigmoid 核函数： $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\beta \boldsymbol{x}_i^T \boldsymbol{x}_j + \theta)$ $\beta > 0, \theta < 0$

正则化与软间隔

针对情况： 如果样本不是完全能够划分开

解决方法： 允许支持向量机在一些样本出错，定义软间隔

引入正则化强度~~参数c~~，损失函数重新定义为：

Regularization parameter=lambda= 1/C; C increase(Lambda decrease), tolerance to mistake decrease, easier to overfitting (see notes in Jupyter NB code)

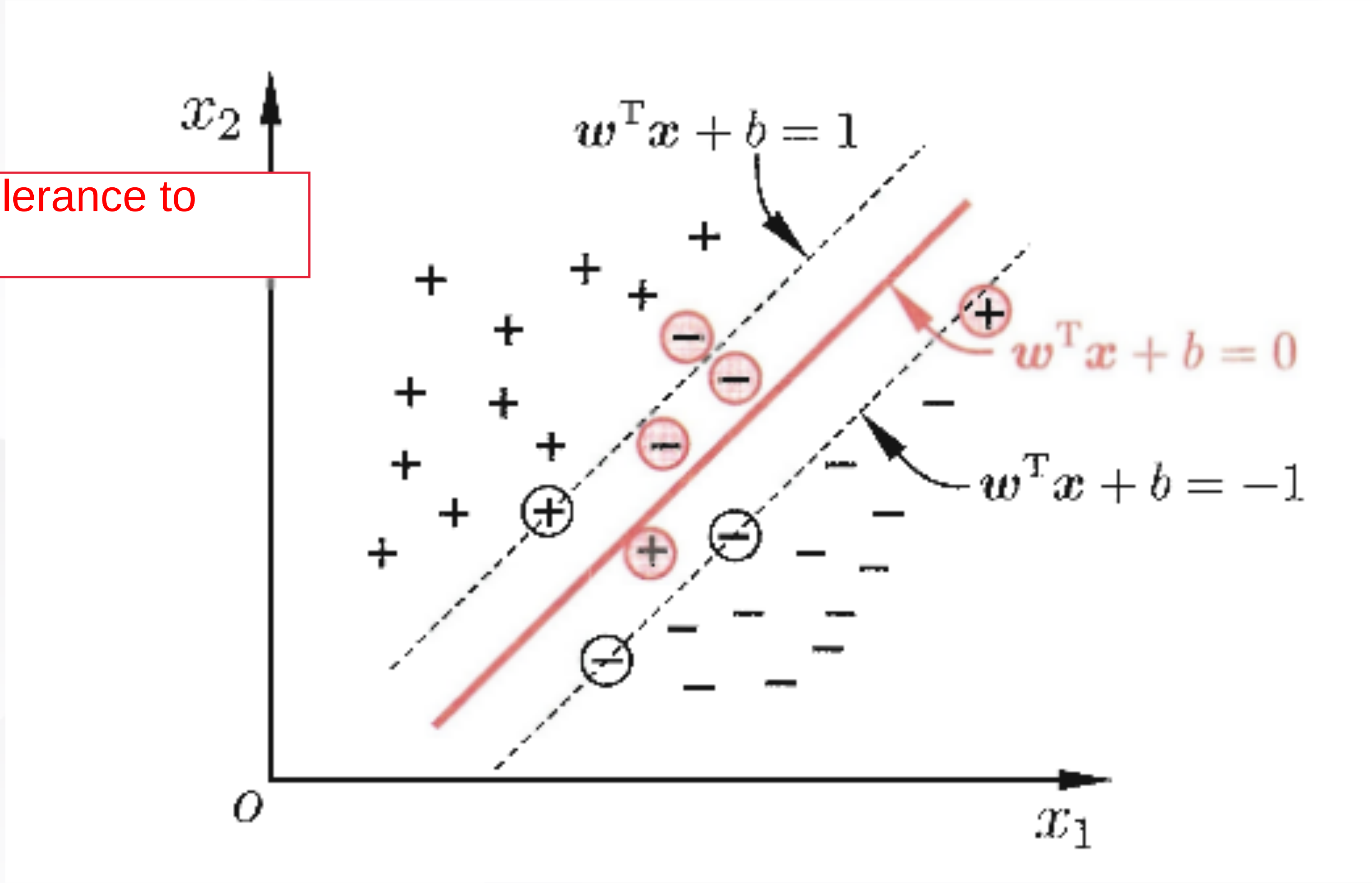
$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} \max(0, 1 - y_i f(x_i))$$

引入松弛变量（slack variables） $\xi_i \geq 0$

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} \xi_i$$

$$s.t. y_i f(x_i) \geq 1 - \xi_i$$

$$\xi_i \geq 0, i = 1,2,3 \dots m$$



软间隔支持向量机

- 拉格朗日函数：

$$L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\lambda},\boldsymbol{r})=\frac{1}{2}\left\|\boldsymbol{w}\right\|^2+C\sum_{i=1}^{m}\xi_i+\sum_{i=1}^{m}\lambda_i(1-\xi_i\ -\ y_i(\boldsymbol{w}^T\boldsymbol{x_i}+b))-\sum_{i=1}^{m}r_i\xi_i\quad(\lambda_i\geq 0,r_i\geq 0)$$

对w,b,ξ求偏导可得：

$$0=\sum_{i=1}^{m}\lambda_i y_i\qquad \boldsymbol{w}=\sum_{i=1}^{m}\lambda_i y_i\boldsymbol{x_i}\qquad C=\lambda_i+r_i$$

相应的对偶问题为：

$$\max_{\lambda} W(\lambda)=\sum_{i=1}^{m}\lambda_i-\frac{1}{2}\sum_{i=1}^{m}y_i y_j\,\lambda_i\lambda_j\boldsymbol{x_i^T}\boldsymbol{x_j}$$

$$s.t.\sum_{i=1}^{m}\lambda_i y_i=0$$

$$C\geq\lambda_i\geq 0, i=1,2,\ldots,m$$

软间隔支持向量机

- 软间隔的KKT条件为：

$$\lambda_i \geq 0, r_i \geq 0$$

$$1 - y_i f(\boldsymbol{x_i}) - \xi_i \leq 0$$

$$\lambda_i(1 - y_i f(\boldsymbol{x_i}) - \xi_i) = 0$$

$$\xi_i \geq 0, r_i \xi_i = 0$$

支持向量：

$\lambda_i = 0 \quad \Rightarrow \quad y_i f(\boldsymbol{x_i}) \geq 1,$ 样本落在间隔外，不影响模型

$0 < \lambda_i < C \quad \Rightarrow \quad y_i f(\boldsymbol{x_i}) = 1,$ 样本落在最大间隔边界上，为支持向量

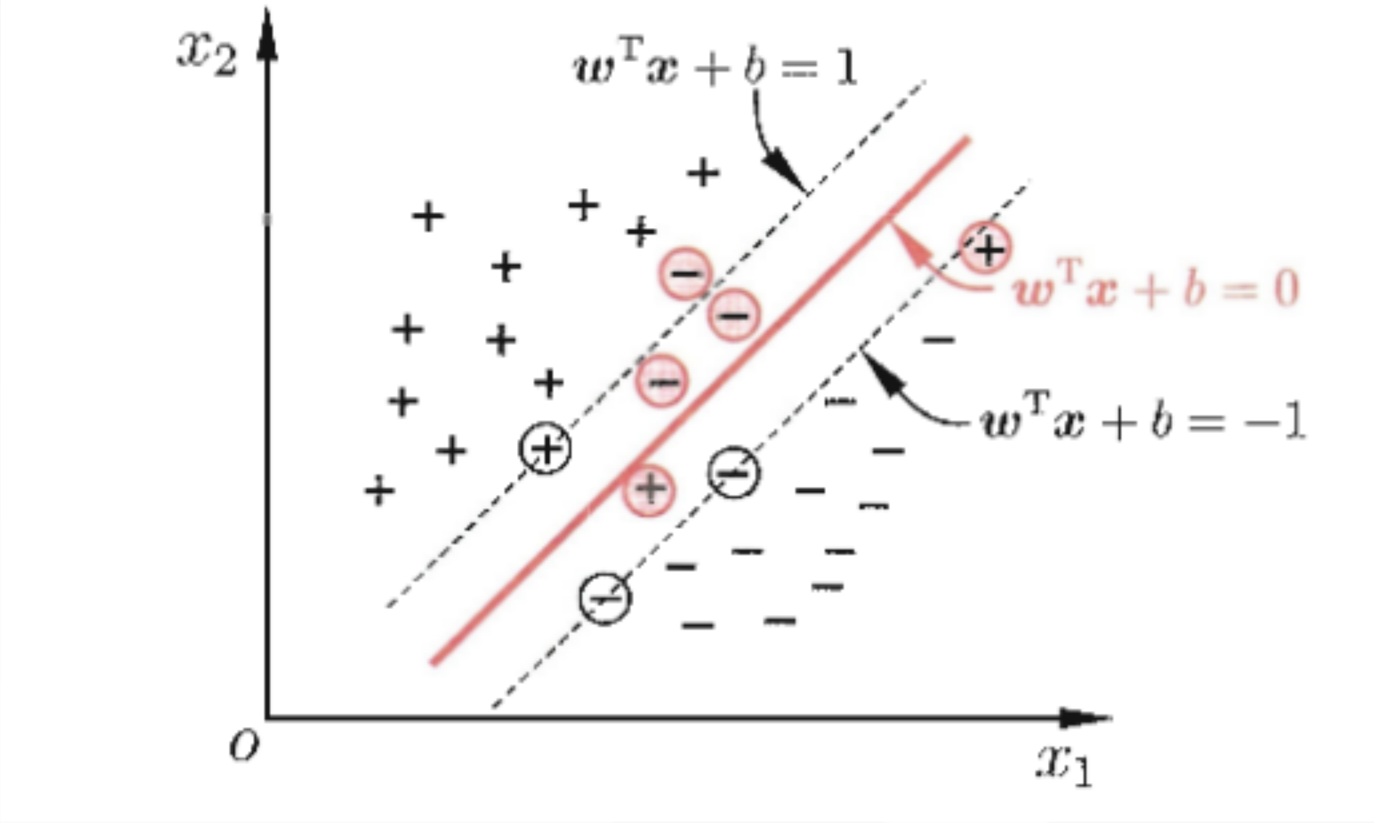
$\lambda_i = C, \xi_i \leq 1 \quad \Rightarrow \quad y_i f(\boldsymbol{x_i}) = 1 - \xi_i \leq 1,$ 样本落在间隔内部为支持向量

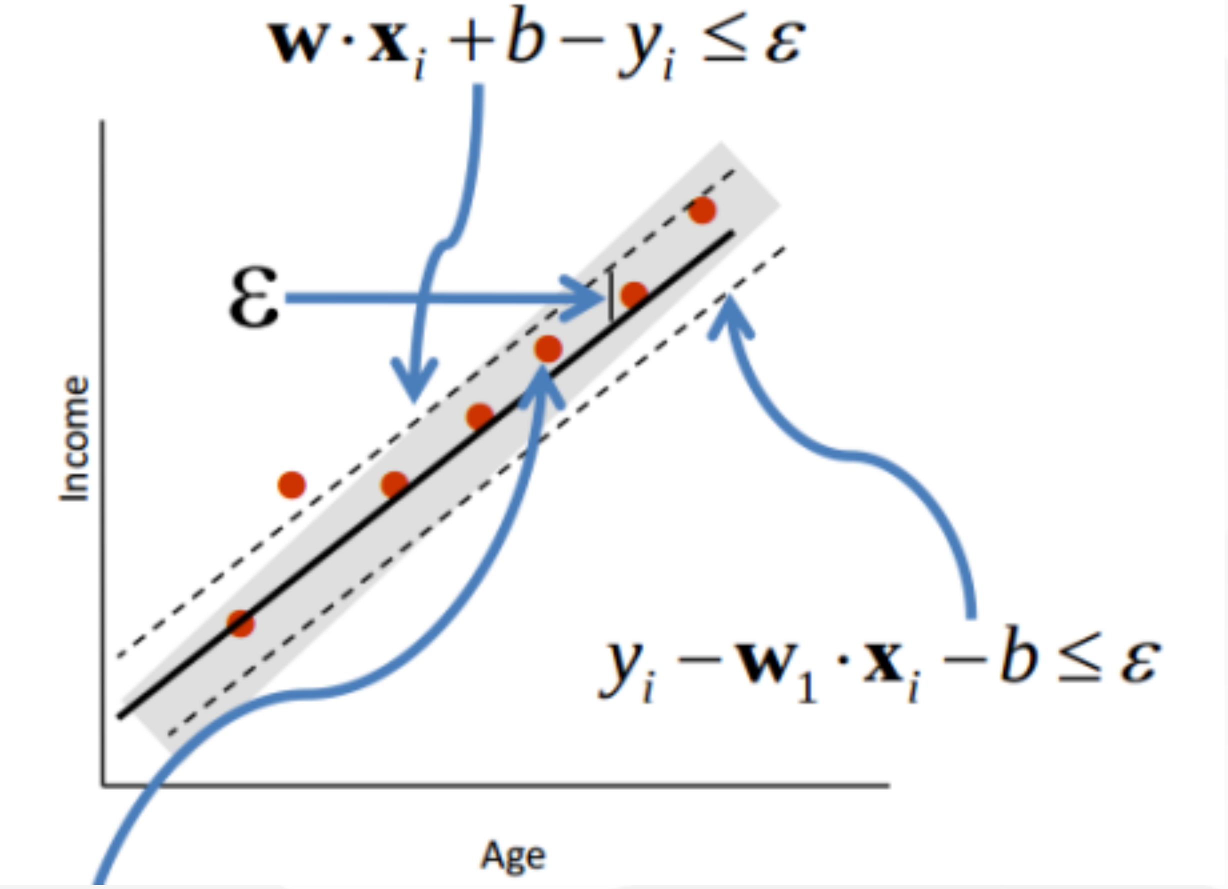
$\lambda_i = C, \xi_i > 1 \quad \Rightarrow \quad y_i f(\boldsymbol{x_i}) = 1 - \xi_i < 0,$ 样本被错误分类，为支持向量

支持向量机分类与回归

SVM regression = SVR, support vectors = all vectors that are outside of the margin.

SVM categorization = SVC, support vectors = all vectors fall into and on the boundaries AND vectors that are miss categorized(see page before)





对于SVM分类，容忍一定数目的错分类

对于SVM回归，容忍ε 以内的偏差，margin以外的计算损失

Summary of support vectors:
For both SVC and SVR, support vectors are defined as the vectors that can affect SVM model. For Categorization problem, these are points that CANNOT be corrected separated (on boundary, inside boundary, miss categorized); For Regression problem, these are points that CANNOT fit into regression boundaries. Since for general cases, only small numbers of points cannot be categories/fit into regression, so SVM only need to consider small numbers of points to calculation, which makes it efficient (see P35 - summary of SVM property "稀疏性")

支持向量机回归分析（SVR）

- SVR的主凸优化问题：

$$\min_{w} \frac{1}{2}||w||^2$$

$$s.t. y_i - \boldsymbol{w^T x_i} - b \le \epsilon$$

$$\boldsymbol{w^T x_i} + b - y_i \le \epsilon$$

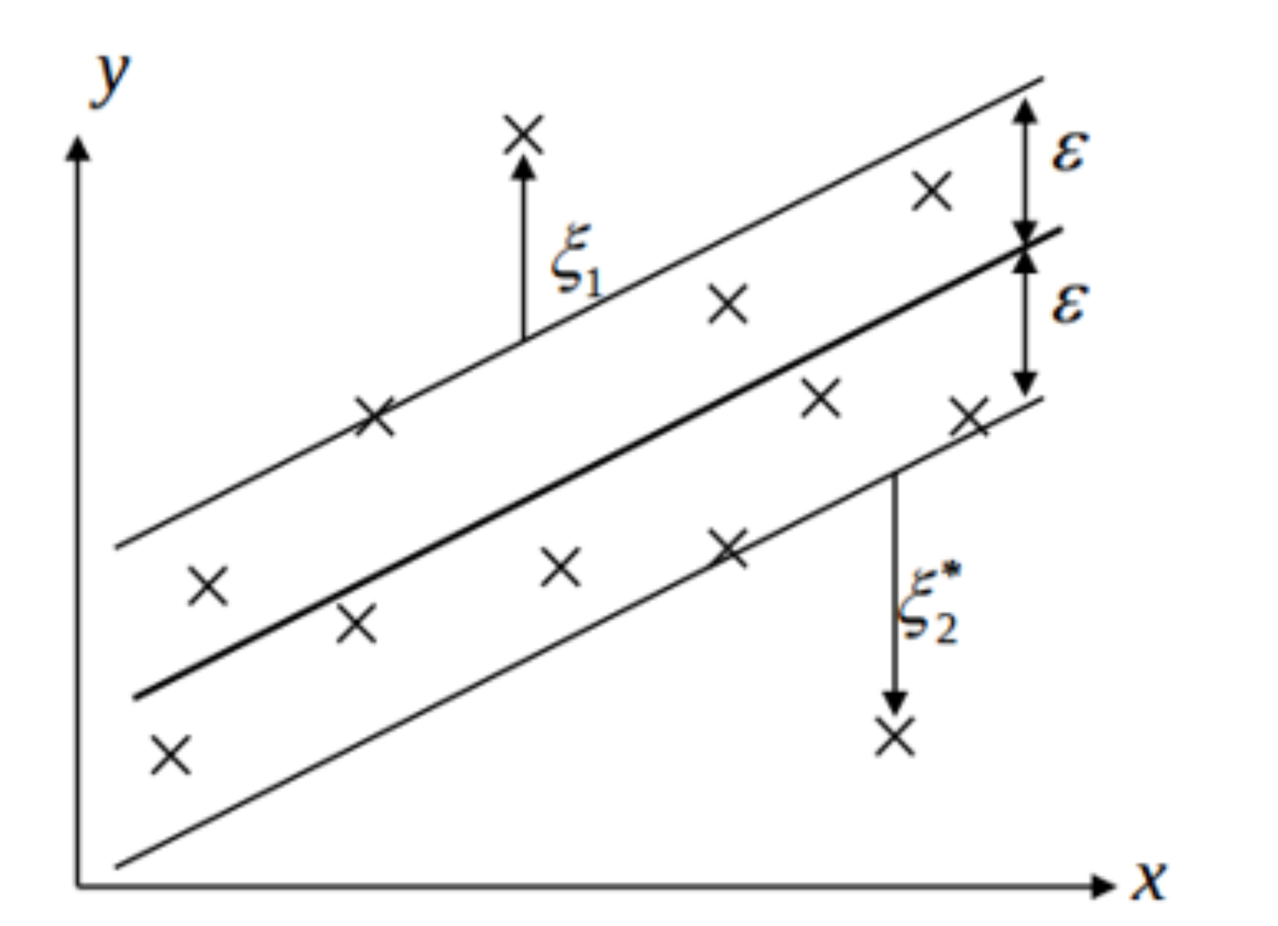
软间隔，对于落在margin以外的点计入损失：

$$\min_{w} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i')$$

$$s.t. \boldsymbol{w^T x_i} + b - y_i \le \epsilon + \xi_i$$

$$y_i - \boldsymbol{w^T x_i} - b \le \epsilon + \xi_i'$$

$$\xi_i \ge 0, \xi_i' \ge 0, i = 1,2,3 \dots m$$



SVR 对偶问题

构造拉格朗日函数, 引入拉格朗日乘子 $\lambda_i \geq 0, \lambda_i' \geq 0, r_i \geq 0, r_i' \geq 0$

$$L(\boldsymbol{w}, b, \boldsymbol{\lambda}, \boldsymbol{\lambda}', \boldsymbol{\xi}, \boldsymbol{\xi}', \boldsymbol{r}, \boldsymbol{r}') = \frac{1}{2}\left||\boldsymbol{w}|\right|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i') + \sum_{i=1}^{m}\lambda_i(f(x_i) - y_i - \epsilon - \xi_i)$$

$$+\sum_{i=1}^{m}\lambda_i'(y_i - f(x_i) - \epsilon - \xi_i') - \sum_{i=1}^{m} r_i\xi_i - \sum_{i=1}^{m} r_i'\xi_i'$$

对$\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}'$求偏导则有：

$$\boldsymbol{w} = \sum_{i=1}^{m}(\lambda_i' - \lambda_i)\boldsymbol{x_i} \qquad 0 = \sum_{i=1}^{m}(\lambda_i - \lambda_i') \qquad C = \lambda_i + r_i \qquad C = \lambda_i' + r_i'$$

SVR 对偶形式为：

$$\max_{\lambda,\lambda'} W(\lambda, \lambda') = \sum_{i=1}^{m}[y_i(\lambda_i' - \lambda_i) - \epsilon(\lambda_i' + \lambda_i)] - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\lambda_i' - \lambda_i)(\lambda_j' - \lambda_j)\boldsymbol{x_i^T}\boldsymbol{x_j}$$

$$s.t. \sum_{i=1}^{m}(\lambda_i' - \lambda_i) = 0$$

$$C \geq \lambda_i, \lambda_i' \geq 0, i = 1, 2, \dots, m$$

SVR 的对偶问题

满足KKT条件，则要求：

$$\lambda_i \lambda_i' = 0, \xi_i \xi_i' = 0$$

$$\lambda_i(f(\boldsymbol{x_i}) - y_i - \epsilon - \xi_i) = 0$$

$$\lambda_i'(y_i - f(\boldsymbol{x_i}) - \epsilon - \xi_i') = 0$$

$$(C - \lambda_i)\xi_i = 0, (C - \lambda_i')\xi_i' = 0$$

当且仅当：

$$f(\boldsymbol{x_i}) - y_i - \epsilon - \xi_i = 0\text{时，}\lambda_i \neq 0$$

$$y_i - f(\boldsymbol{x_i}) - \epsilon - \xi_i' = 0\text{时，}\lambda_i' \neq 0$$

也就是当样本落在ϵ −间隔带以外，为支持向量，相应的λ_i，λ_i' 取非零值

$$f(\boldsymbol{x}) = \sum_{i=1}^{m}(\lambda_i' - \lambda_i)\boldsymbol{x_i^T x} + b$$

对于$0 < \lambda_i < C, \xi_i = 0$，取 $b = y_i + \epsilon - \sum_{i=1}^{m}(\lambda_i' - \lambda_i)\boldsymbol{x_i^T x}$ 的平均值

支持向量机算法总结

- 优点：
  - 解决高维特征的分类问题和回归问题很有效,在特征维度大于样本数时依然有很好的效果
  - 稀疏性：仅仅使用支持向量来做超平面的决策，无需依赖全部数据
  - 核函数可以很灵活的来解决各种非线性的分类回归问题
  - 样本量不是海量数据的时候，分类准确率高，泛化能力强
- 缺点：
  - 如果特征维度远远大于样本数，则SVM表现一般
  - SVM在样本量非常大，核函数映射维度非常高时，计算量过大，不太适合使用
  - SVM对缺失数据敏感