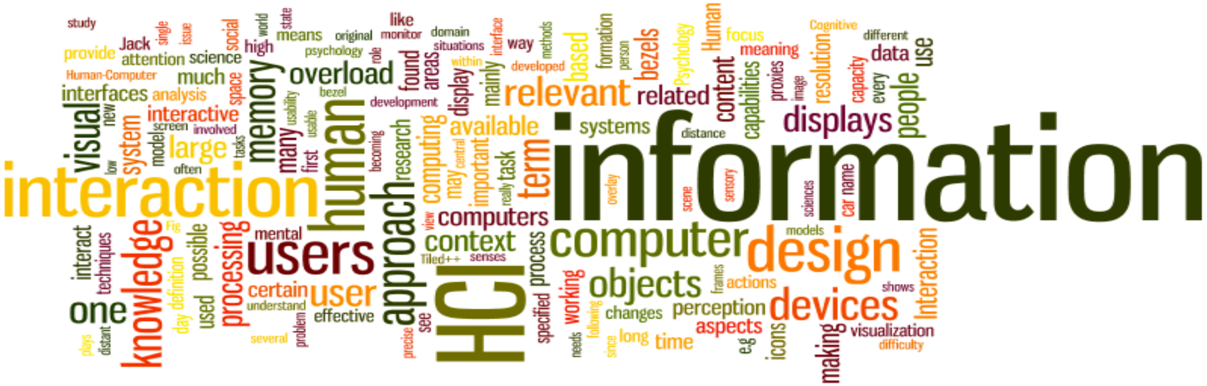
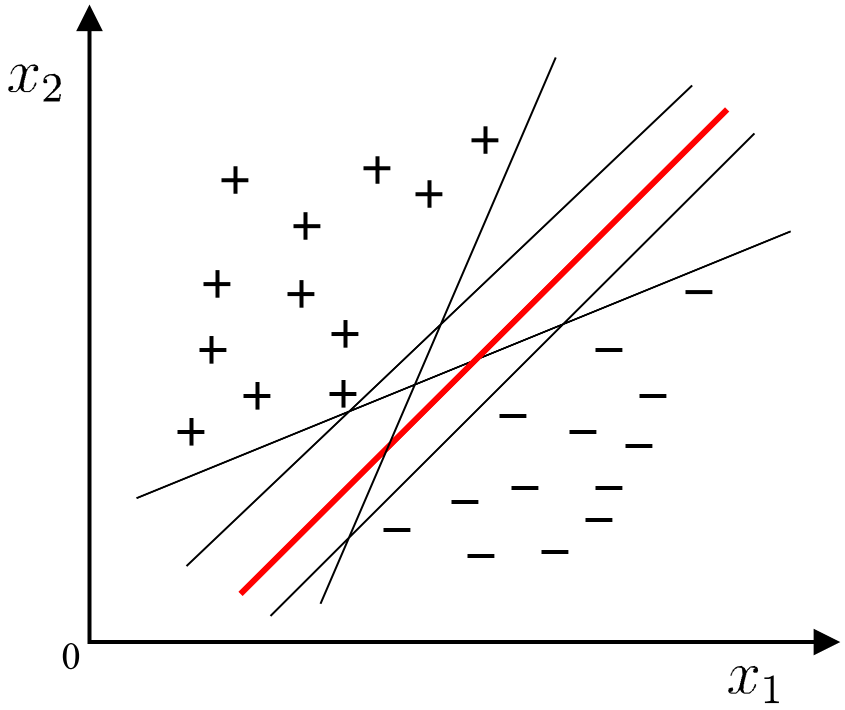
机器学习

——第7章 支持向量机——



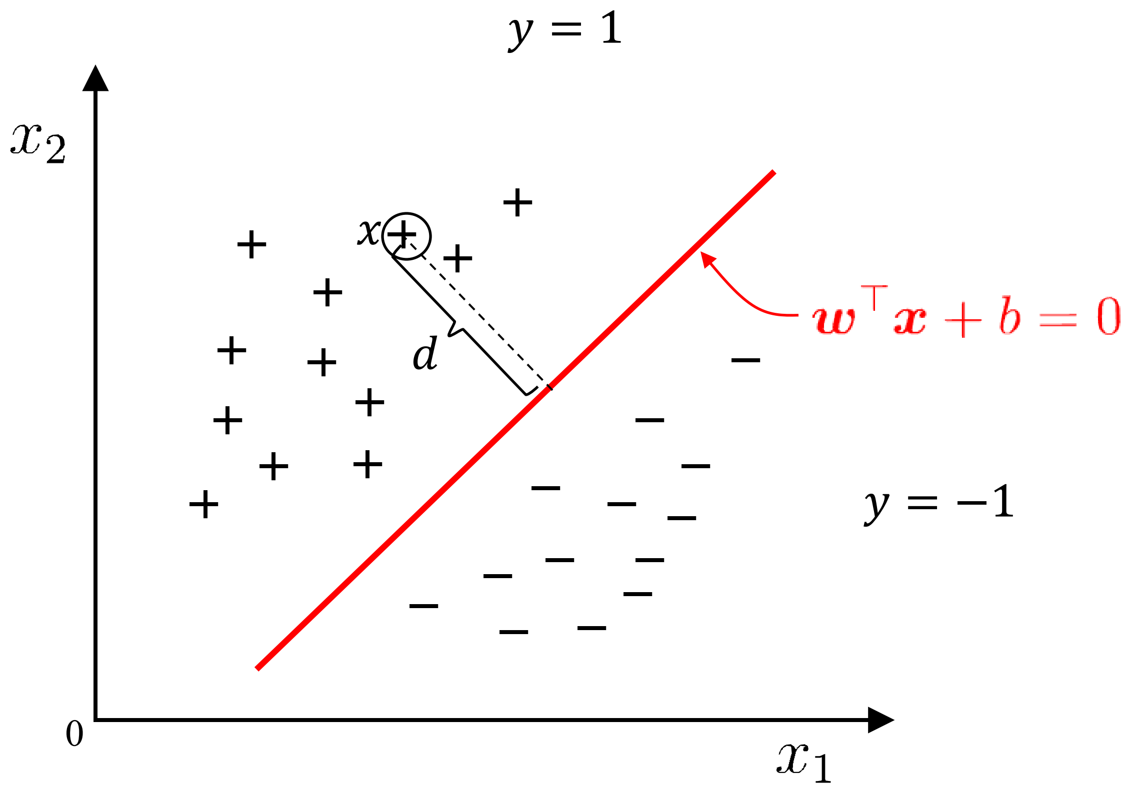
倪张凯
zkni@tongji.edu.cn
https://eezkni.github.io/

二分类问题 Binary Classification



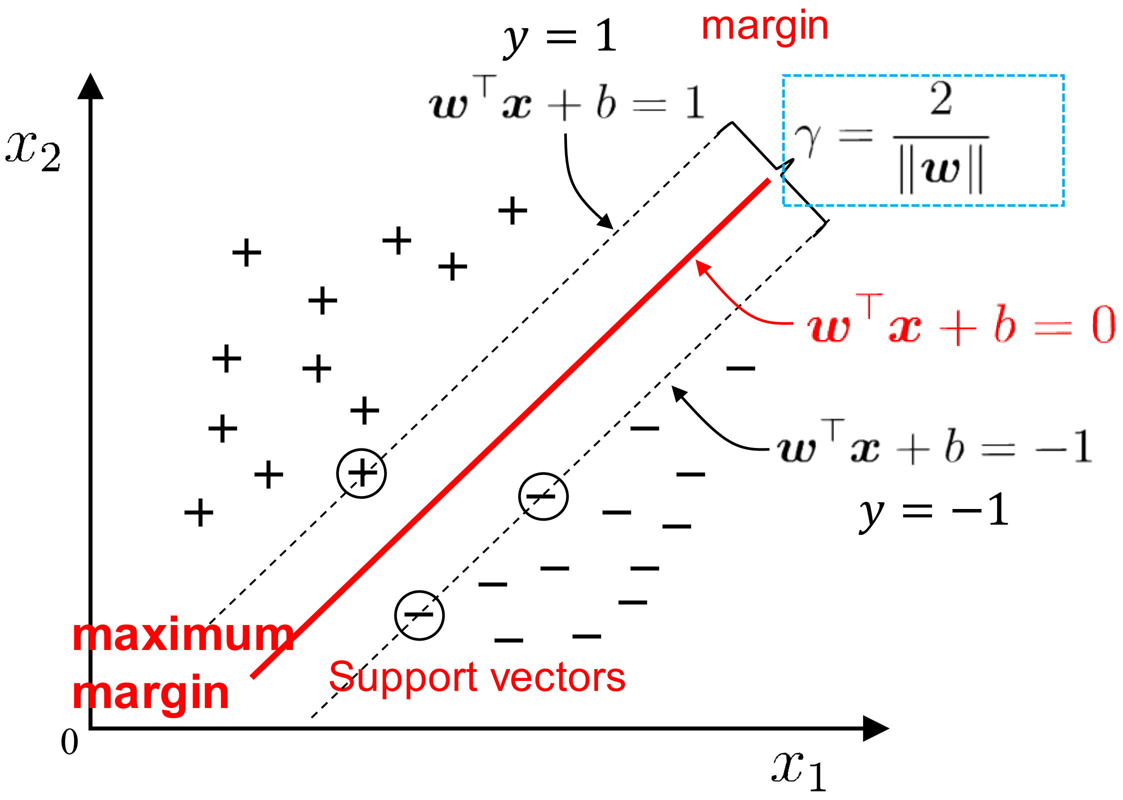
It should choose "right in the middle", with good tolerance, high robustness and the strongest generalization ability

支持向量机 Support Vector Machine

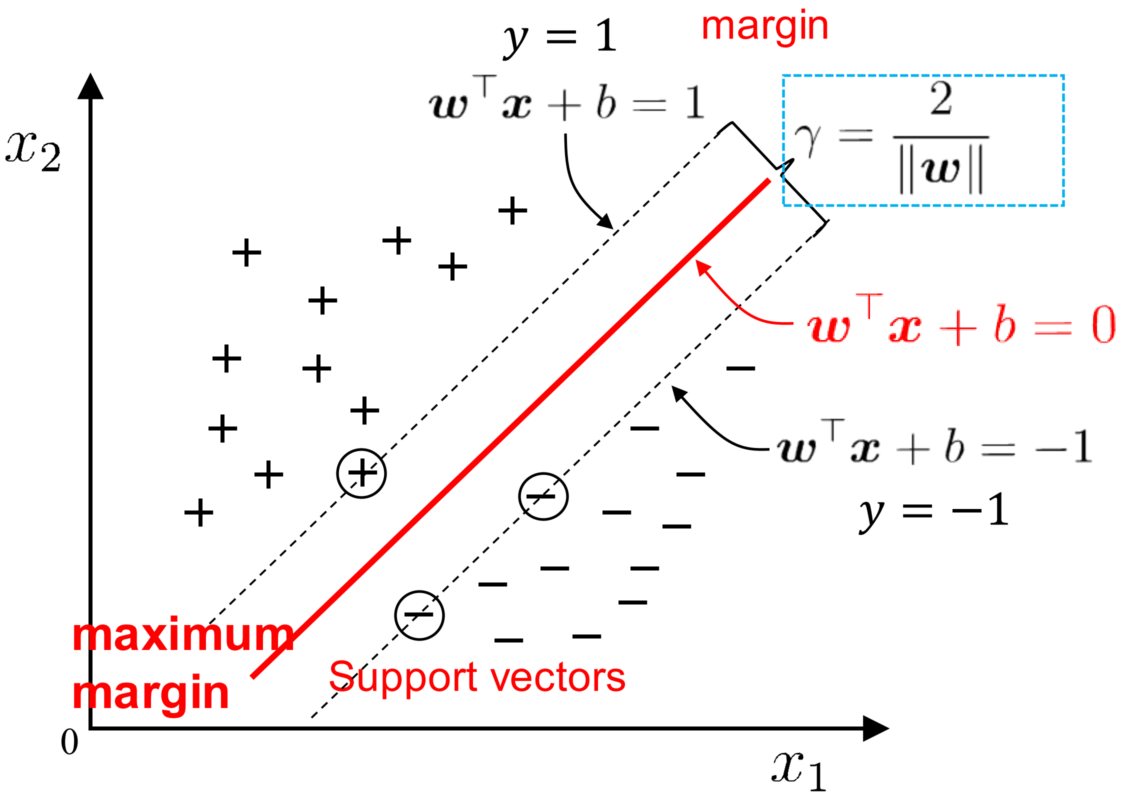


$$d = \frac{|w^T x + b|}{||w||}$$

支持向量机 Support Vector Machine

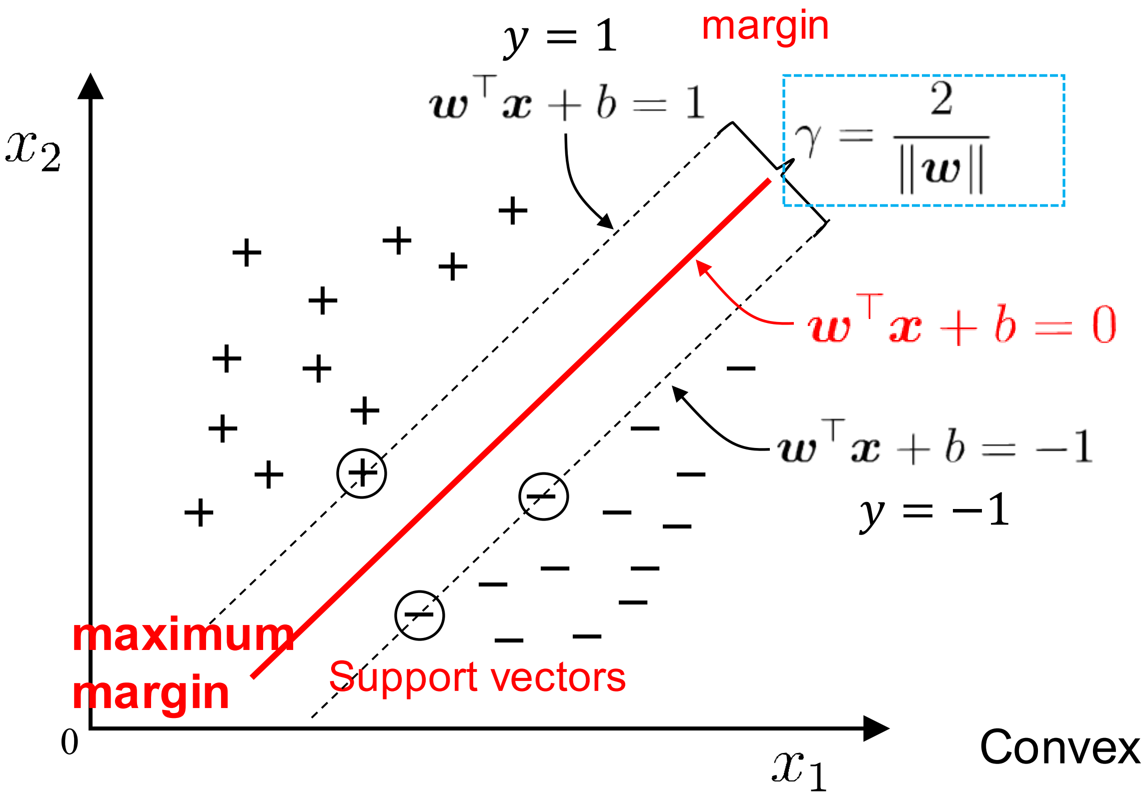


支持向量机 Support Vector Machine



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

支持向量机 Support Vector Machine



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

凸二次规划

Convex Quadratic Programming

$$\arg\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|} \Rightarrow \arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$\text{s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \ i = 1, 2, \ldots, m.$$

对偶问题 Dual problem

在数学和优化领域中，对偶问题是与原始优化问题（原始问题）相对应的一个问题。通常，对偶问题是通过原始问题的一种变换来构建的，它可以帮助我们更好地理解原始问题的性质，提供额外的信息，或者用于求解原始问题。

- 对于最小化问题（Minimization Problem）：对偶问题：max g(λ,ν)其中，g(λ,ν)=infxL(x,λ,ν)，表示对于给定L(x,λ,ν) 的最小值
- 对于最大化问题（Maximization Problem）：对偶问题：min g(λ,ν) 其中，g(λ,ν)=supxL(x,λ,ν)，表示对于给定L(x,λ,ν) 的最大值

在这里：

- λ 和 ν 是对偶变量（Lagrange Multipliers），它们对应于原始问题的约束条件。
- L(x,λ,ν) 是称为拉格朗日函数（Lagrangian Function）的函数，它由原始问题的目标函数和约束条件组成，通常形式为：

L(x,λ,ν)=目标函数−λT·(不等式约束)−νT·(等式约束)

通过求解对偶问题，我们可以获得原始问题的最优值的一个下界或上界。

拉格朗日乘数法的例子

- **问题**：假设你需要在平面上找到距离原点最近的点，但这个点必须位于直线 $y=2x+3$ 上。
- **解决方法**：

拉格朗日乘数法的例子

- **问题**：假设你需要在平面上找到距离原点最近的点，但这个点必须位于直线 $y=2x+3$ 上。
- **解决方法**：

1. **目标函数**：需要最小化的目标函数是到原点的距离的平方，$f(x,y)=x^2+y^2$。
2. **约束**：点必须位于 $y=2x+3$ 上，因此约束条件为 $g(x,y)=y-2x-3=0$。
3. **构建拉格朗日函数**：构建拉格朗日函数 $L(x,y,\lambda)=x^2+y^2+\lambda(y-2x-3)$。
4. **求解**：通过 $L(x,y,\lambda)$ 关于 x,y,λ 的偏导数求解并置为零，可以得到最优解。

KKT（Karush-Kuhn-Tucker）条件的例子

- **问题**：假设你需要在一个盒子中放置最大体积的长方体，但其长、宽、高的和不能超过某个值，例如10
- **解决方法**：

KKT（Karush-Kuhn-Tucker）条件的例子

- **问题**：假设你需要在一个盒子中放置最大体积的长方体，但其长、宽、高的和不能超过某个值，例如10
- **解决方法**：

  1. **目标函数**：**最大化**长方体的体积，即 $f(x,y,z)=xyz$。
  2. **约束**：长、宽、高的和不超过10，**即** $g(x,y,z)=x+y+z-10\leq 0$。这是一个不等式约束。
  3. **构建拉格朗日函数：拉格朗日函数 L(x,y,z,λ)** 结合了目标函数和约束条件，**通过**引入**拉格朗日乘子 λ**：$L(x,y,\lambda)=xyz-\lambda(x+y+z-10)$

KKT（Karush-Kuhn-Tucker）条件的例子

$L(x,y,\lambda)=xyz-\lambda(x+y+z-10)$

4.应用KKT条件

- **梯度为零**：对 $L(x,y,z,\lambda)$ 对 x,y,z,λ 的偏导数应为零。
  - $\frac{\partial L}{\partial x}=yz-\lambda=0$ $\frac{\partial L}{\partial y}=xz-\lambda=0$ $\frac{\partial L}{\partial z}=xy-\lambda=0$
  - $\frac{\partial L}{\partial \lambda}=-x-y-z+10=0$
- 约束条件：$x+y+z\leq 10$。
- **拉格朗日乘子非负**：$\lambda\geq 0$。
- **互补松弛性**：$\lambda(x+y+z-10)=0$。

5.求解：通过解上述方程组，我们可以找到满足约束的最优解。

最优超平面求解 Optimal hyperplane Solution

- Using Lagrangian Multiplier Method and KKT Condition to solve the Optimal Value

$$\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s.t. } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \ i = 1, 2, \ldots, m.$$

- Integrated into: (Where $\alpha_i \geq 0$ is a Lagrangian multiplier)

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

- Let the partial derivative=0

$$\frac{\partial L(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0 \qquad \frac{\partial L(w, b, \alpha)}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0$$

- then

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i, \quad \sum_{i=1}^{m} \alpha_i y_i = 0.$$

最优超平面求解 Optimal hyperplane Solution

同济大学 TONGJI UNIVERSITY

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

$$= \frac{1}{2} w^T w - \sum_{i=1}^{m} \alpha_i y_i w^T x_i - \mathrm{b} \sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \alpha_i$$

$$= \frac{1}{2} w^T \sum_{i=1}^{m} \alpha_i y_i x_i - \sum_{i=1}^{m} \alpha_i y_i w^T x_i + + \sum_{i=1}^{m} \alpha_i$$

$$= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \alpha_i y_i w^T x_i$$

Dual problem:

$$\max_{a} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j = \min_{a} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{m} \alpha_i$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0, \alpha_i \geq 0, i = 1,2,\ldots,m$$

Obtain the optima α (SMO, Sequential Minimal Optimization) algorithm

The optimal solution can be obtained

$$f(x) = w^T x + b = \sum_{i=1}^{m} \alpha_i y_i x_i^T x + b$$

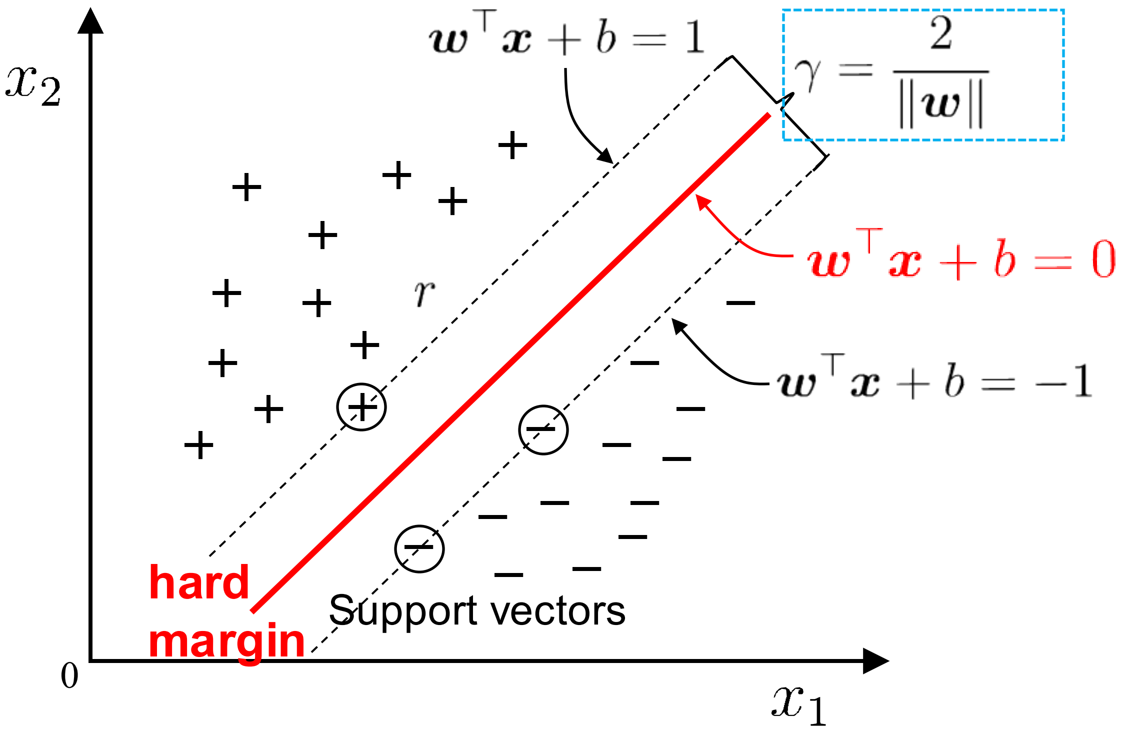
KKT Condition

$$\begin{cases} \alpha_i \geq 0, \\ y_i f(\boldsymbol{x}_i) \geq 1, \\ \alpha_i (y_i f(\boldsymbol{x}_i) - 1) = 0. \end{cases}$$

$$y_i f(\boldsymbol{x}_i) > 1 \Rightarrow \alpha_i = 0$$

$$\alpha_i > 0 \Rightarrow y_i f(x_i) = 1$$

支持向量机 Support Vector Machine



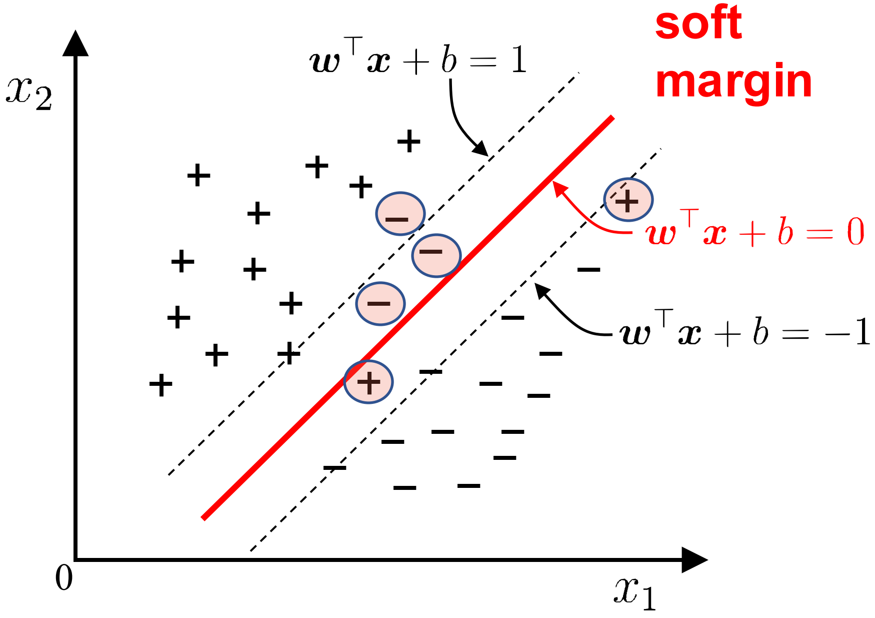
$$\arg\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|} \Rightarrow \arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$\text{s.t.} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, \; i = 1, 2, \ldots, m.$$

软间隔 Soft Margin

-Q: It is difficult to determine a linearly separable hyperplane in the feature space; At the same time, it is difficult to determine whether a linearly separable result is caused by over fitting

-A: The concept of "soft margin" is introduced to allow the support vector machine to not meet the constraints on some samples



软间隔支持向量机 Soft Margin Support Vector Machine

The slack variable ε can be introduced :

$$\min_{w,b}\frac{1}{2}\|w\|_2+C\sum_{i}\varepsilon_i \qquad \text{s.t. } y_i(w^Tx_i+b)\geq 1-\varepsilon_i$$

$$\varepsilon_i \geq 0,\ \text{i=1,2,…m}$$

Integrated into: (Where α_i , ε_i are Lagrangian multipliers)

$$L(w,b,a,\varepsilon,\mu)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{m}\varepsilon_i+\sum_{i=1}^{m}\alpha_i(1-\varepsilon_i-y_i(w^Tx_i+b))-\sum_{i=1}^{m}\mu_i\varepsilon_i$$

Let the partial derivative=0

$$\frac{\partial L(w,b,a,\varepsilon,\mu)}{\partial w}=w-\sum_{i=1}^{m}\alpha_iy_ix_i=0 \qquad \frac{\partial L(w,b,a,\varepsilon,\mu)}{\partial b}=\sum_{i=1}^{m}\alpha_iy_i=0$$

$$\frac{\partial L(w,b,a,\varepsilon,\mu)}{\partial \varepsilon_i}=C-\alpha_i-\mu_i=0$$

软间隔支持向量机 Soft Margin Support Vector Machine

The slack variable ε can be introduced :

$$\min_{w,b} \frac{1}{2}\|w\|_2 + C\sum_i \varepsilon_i \qquad \text{s.t. } y_i(w^T x_i + b) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0, \text{i=1,2,…m}$$

Integrated into: (Where α_i , ε_i are Lagrangian multipliers)

$$L(w,b,a,\varepsilon,\mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \varepsilon_i + \sum_{i=1}^{m} \alpha_i(1 - \varepsilon_i - y_i(w^T x_i + b)) - \sum_{i=1}^{m} \mu_i \varepsilon_i$$

Let the partial derivative=0

then

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i \qquad \sum_{i=1}^{m} \alpha_i y_i = 0$$

$$C = \alpha_i + \mu_i$$

软间隔支持向量机 Soft Margin Support Vector Machine

The slack variable ε can be introduced :

$$\min_{w,b} \frac{1}{2}\|w\|_2 + C\sum_i \varepsilon_i \qquad \text{s.t. } y_i(w^T x_i + b) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0, \text{ i=1,2,…m}$$

Integrated into: (Where α_i , ε_i are Lagrangian multipliers)

$$L(w, b, a, \varepsilon, \mu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \varepsilon_i + \sum_{i=1}^{m} \alpha_i(1 - \varepsilon_i - y_i(w^T x_i + b)) - \sum_{i=1}^{m} \mu_i \varepsilon_i$$

Dual problem:

$$\max_{a} \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad s.t. \sum_{i=1}^{m} \alpha_i y_i = 0, \qquad C \geq \alpha_i \geq 0, i = 1,2, \dots, m$$

Obtain the optima α (SMO, Sequential Minimal Optimization) algorithm

The optimal solution can be obtained

$$f(x) = w^T x + b = \sum_{i=1}^{m} \alpha_i y_i x_i^T x + b$$

KKT Condition

$$\begin{cases} \alpha_i \geq 0, \mu_i \geq 0 \\ y_i f(x_i) - 1 + \varepsilon_i \geq 0 \\ \alpha_i(y_i f(x_i) - 1 + \varepsilon_i) = 0 \\ \varepsilon_i \geq 0, \mu_i \varepsilon_i = 0 \end{cases}$$

软间隔支持向量机 Soft Margin Support Vector Machine

The slack variable ε can be introduced :

$$\min_{w,b} \frac{1}{2}\|w\|_2 + C\sum_{i} \varepsilon_i \quad \text{s.t. } y_i(w^T x_i + b) \geq 1 - \varepsilon_i$$

Dual Problem:

$$\max_{a} \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j = \min_{a} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{m} \alpha_i$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0, C \geq \alpha_i \geq 0, i = 1,2,\dots,m$$

最优超平面求解 Optimal hyperplane Solution

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

$$= \frac{1}{2} w^T w - \sum_{i=1}^{m} \alpha_i y_i w^T x_i - \mathrm{b} \sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \alpha_i$$

$$= \frac{1}{2} w^T \sum_{i=1}^{m} \alpha_i y_i x_i - \sum_{i=1}^{m} \alpha_i y_i w^T x_i + + \sum_{i=1}^{m} \alpha_i$$

$$= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \alpha_i y_i w^T x_i$$

Dual problem:

$$\max_a \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j = \min_a \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{m} \alpha_i$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0, \alpha_i \geq 0, i = 1,2,\ldots,m$$

Obtain the optima α (SMO, Sequential Minimal Optimization) algorithm

The optimal solution can be obtained

$$f(x) = w^T x + b = \sum_{i=1}^{m} \alpha_i y_i x_i^T x + b$$

KKT Condition

$$\begin{cases} \alpha_i \geq 0, \\ y_i f(\boldsymbol{x}_i) \geq 1, \\ \alpha_i(y_i f(\boldsymbol{x}_i) - 1) = 0. \end{cases}$$

$$y_i f(\boldsymbol{x}_i) > 1 \Rightarrow \alpha_i = 0$$

$$\alpha_i > 0 \Rightarrow y_i f(x_i) = 1$$

软间隔支持向量机 Soft Margin Support Vector Machine

The slack variable ε can be introduced :

$$\min_{w,b} \frac{1}{2}\|w\|_2 + C\sum_{i} \varepsilon_i \quad \text{s.t. } y_i(w^T x_i + b) \geq 1 - \varepsilon_i$$

Dual Problem:

$$\max_{a} \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j = \min_{a} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{m} \alpha_i$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0, \boxed{C \geq \alpha_i \geq 0,} i = 1,2,\ldots,m$$

软间隔支持向量机 Soft Margin Support Vector Machine

The slack variable ε can be introduced :

$$\min_{w,b} \frac{1}{2}\|w\|_2 + C\sum_i \varepsilon_i \quad \text{s.t. } y_i(w^T x_i + b) \geq 1 - \varepsilon_i$$

$\varepsilon_i \geq 0$, i=1,2,…m

$$\varepsilon^{\mathrm{i}} = max(0, 1 - y^{(i)} f(x^{(i)}))$$

软间隔支持向量机 Soft Margin Support Vector Machine

The slack variable ε can be introduced :

$$\min_{w,b}\frac{1}{2}\|w\|_2+C\sum_{i}\varepsilon_i \quad \text{s.t. } y_i(w^T x_i+b)\geq 1-\varepsilon_i$$

$$\varepsilon_i \geq 0,\ \text{i=1,2,...m}$$

$$\boxed{\varepsilon^{\mathrm{i}} = max(0,1-\ y^{(i)}f\left(x^{(i)}\right))}$$

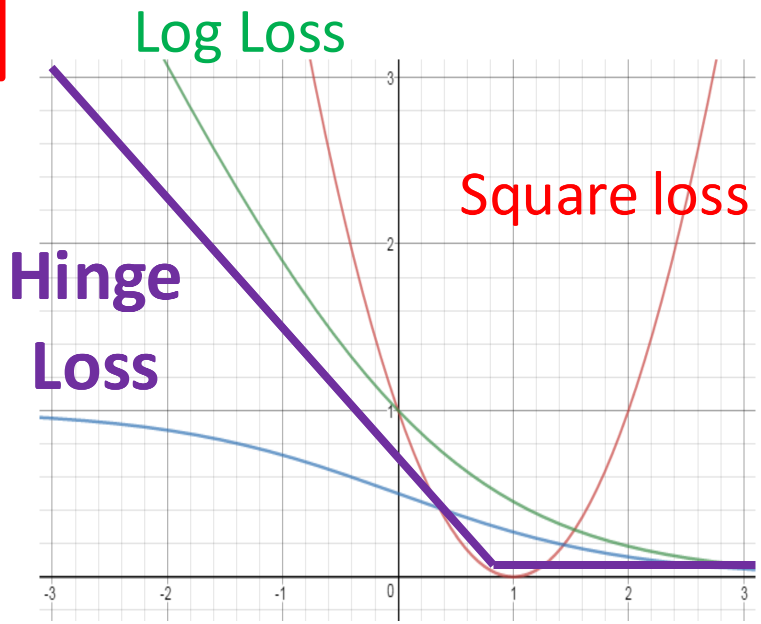
The slack variable ε indicating the extent to which the sample does not meet the constraint.

软间隔支持向量机 Soft Margin Support Vector Machine

The slack variable ε indicating the extent to which the sample does not meet the constraint.

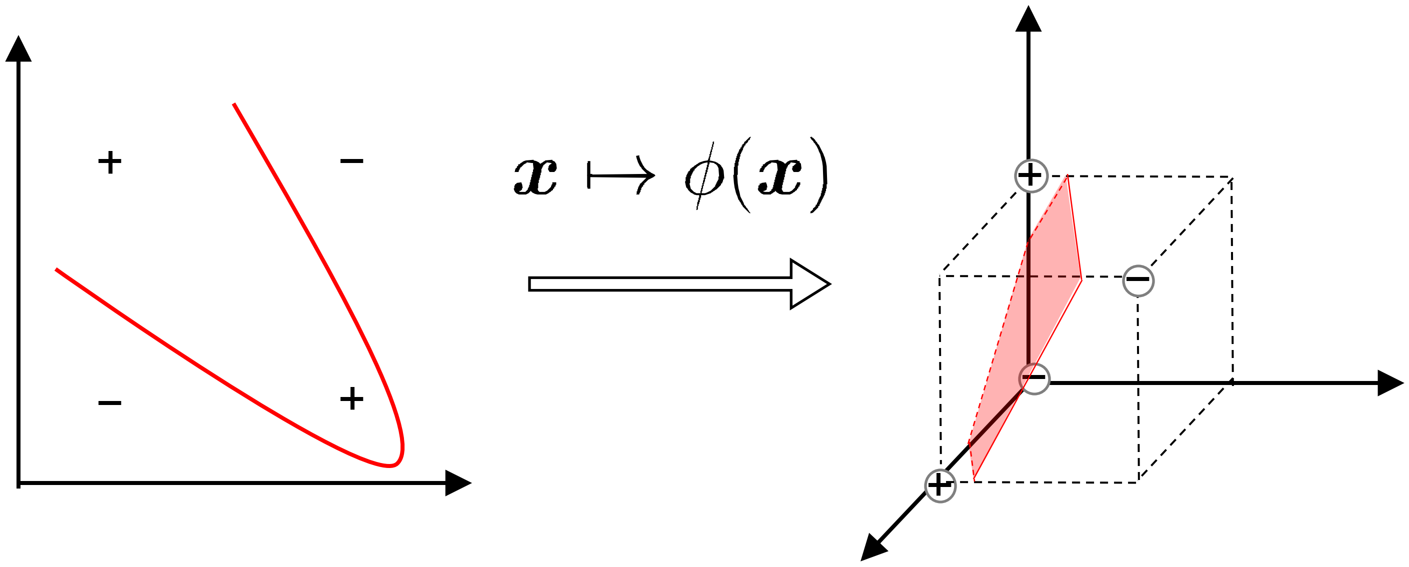
$$\boxed{\varepsilon^{\mathrm{i}} = max(0, 1 - y^{(i)} f(x^{(i)}))}$$

Hinge Loss function



线性不可分 linearly Inseparable Problem

-Q: What if there is no hyperplane that can correctly divide two types of samples?
-A: The samples are mapped from the original space to a higher dimensional feature space, making the samples linearly separable in this feature space



最优超平面求解 Optimal hyperplane Solution

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

$$= \frac{1}{2} w^T w - \sum_{i=1}^{m} \alpha_i y_i w^T x_i - \mathrm{b} \sum_{i=1}^{m} \alpha_i y_i + \sum_{i=1}^{m} \alpha_i$$

$$= \frac{1}{2} w^T \sum_{i=1}^{m} \alpha_i y_i x_i - \sum_{i=1}^{m} \alpha_i y_i w^T x_i + + \sum_{i=1}^{m} \alpha_i$$

$$= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \alpha_i y_i w^T x_i$$

Dual problem:

$$\max_{a} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j = \min_{a} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{m} \alpha_i$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0, \alpha_i \geq 0, i = 1,2,\ldots,m$$

KKT Condition

$$\begin{cases} \alpha_i \geq 0, \\ y_i f(\boldsymbol{x}_i) \geq 1, \\ \alpha_i (y_i f(\boldsymbol{x}_i) - 1) = 0. \end{cases}$$

$$y_i f(\boldsymbol{x}_i) > 1 \Rightarrow \alpha_i = 0$$

$$\alpha_i > 0 \Rightarrow y_i f(x_i) = 1$$

Obtain the optima α (SMO, Sequential Minimal Optimization) algorithm

The optimal solution can be obtained

$$f(x) = w^T x + b = \sum_{i=1}^{m} \alpha_i y_i x_i^T x + b$$

核支持向量机 Kernel SVM

sample $\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$, then the hyperplane $f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b$

Original question:

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s.t.} \quad y_i(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b) \geq 1, \ \ i = 1, 2, \ldots, m.$$

Dual problem:

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j) - \sum_{i=1}^{m} \alpha_i$$
$$\text{s.t.} \quad \sum_{i=1}^{m} \alpha_i y_i = 0, \ \ \alpha_i \geq 0, \ \ i = 1, 2, \ldots, m.$$

Optimal solution:

$$\boxed{f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b = \sum_{i=1}^{m} \alpha_i y_i \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}) + b}$$

核函数 Kernel Function

sample $\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$, then the hyperplane $f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b$

Original question:

Kernel Function

$$\boxed{\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)}$$

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

$$\text{s.t.} \quad y_i(\boldsymbol{w}^\top \phi(\boldsymbol{x}_i) + b) \geq 1, \ i = 1, 2, \ldots, m.$$

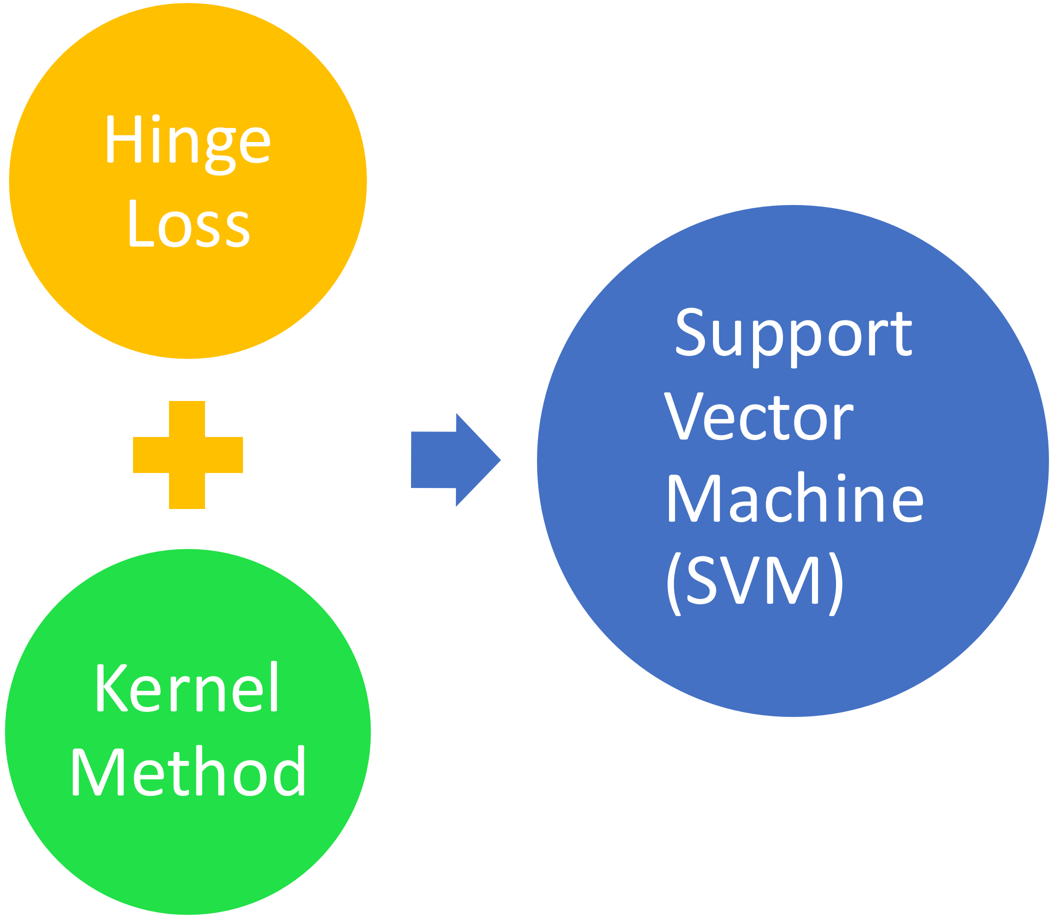
Dual problem:

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boxed{\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)} - \sum_{i=1}^{m} \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{m} \alpha_i y_i = 0, \ \alpha_i \geq 0, \ i = 1, 2, \ldots, m.$$

Optimal solution:

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}) + b = \sum_{i=1}^{m} \alpha_i y_i \boxed{\phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x})} + b$$

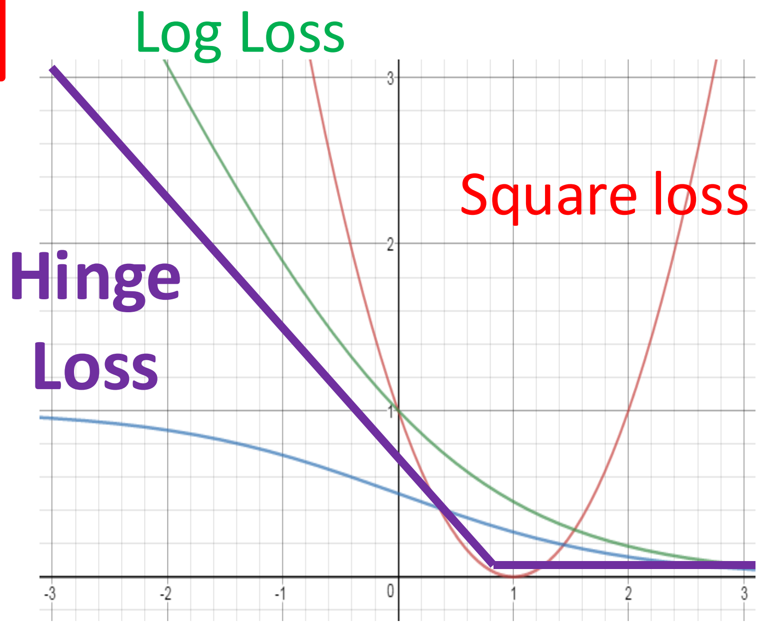

Hinge Loss
Kernel Method
Support Vector Machine (SVM)

支持向量机 Support Vector Machine

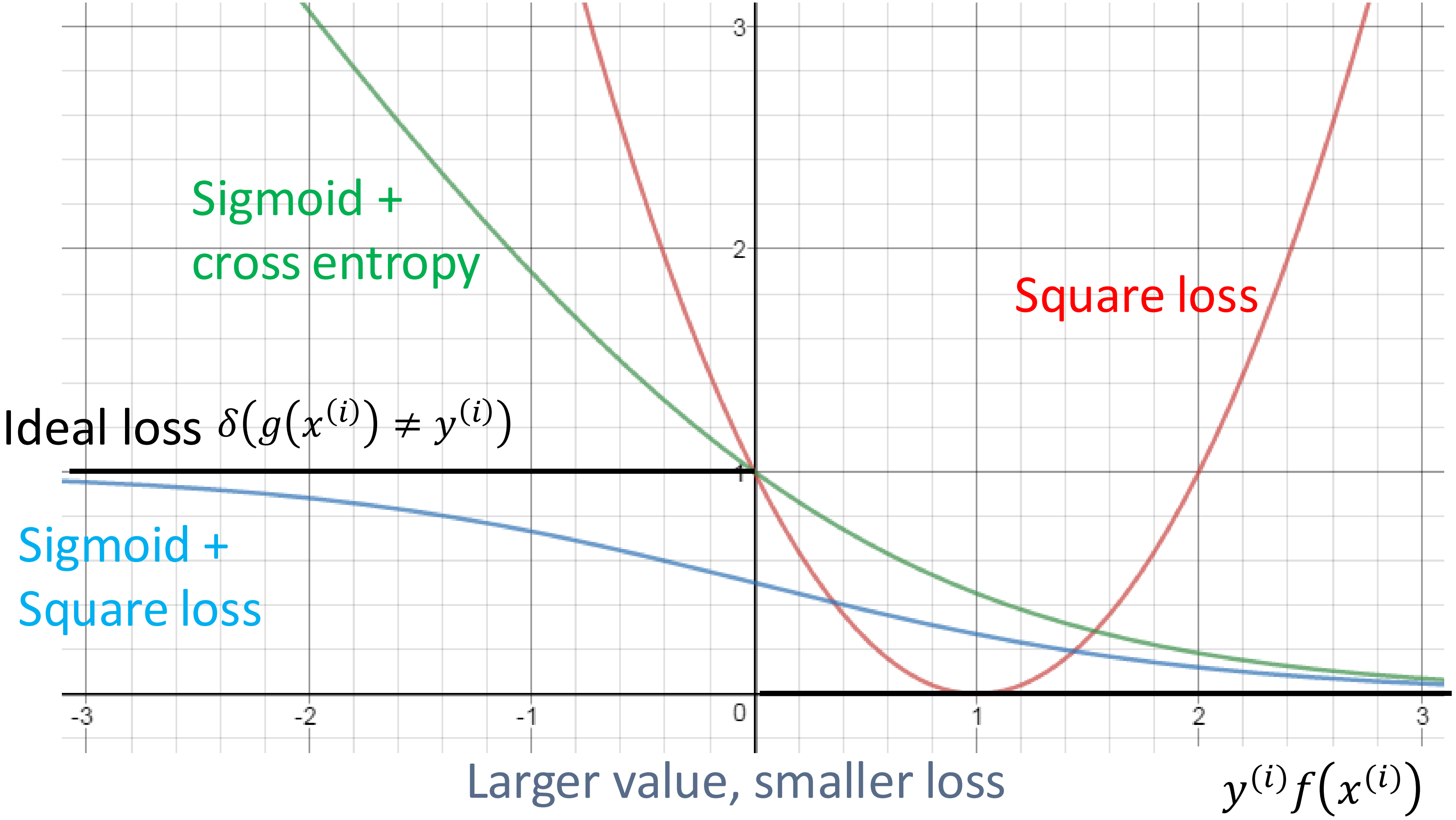
The slack variable ε indicating the extent to which the sample does not meet the constraint.

$$\boxed{\varepsilon^{\mathrm{i}} = max(0, 1 - y^{(i)} f(x^{(i)}))}$$

Hinge Loss function

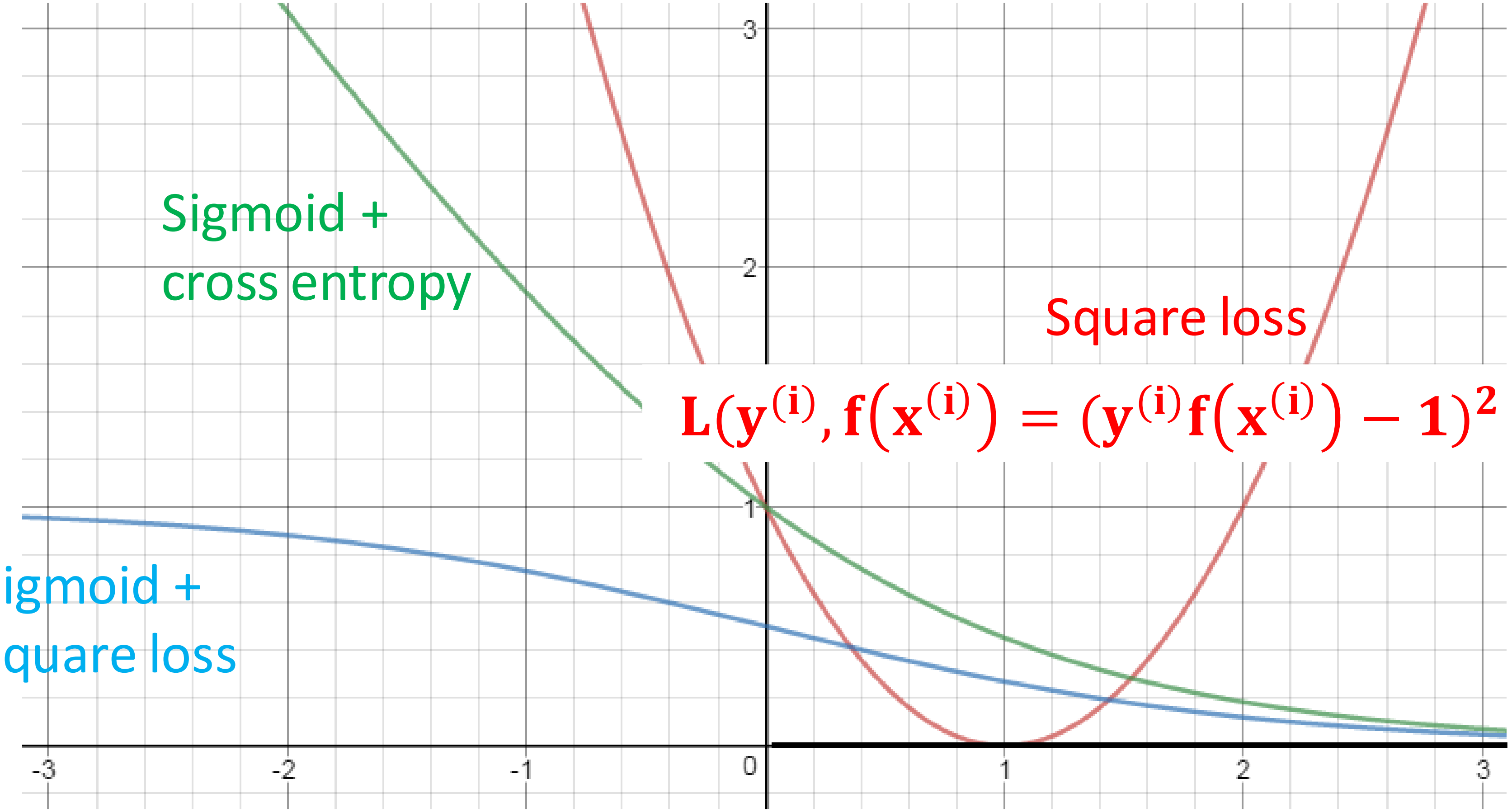


损失函数 Loss function



损失函数 Loss function

Square Loss:



$$\mathbf{L}(\mathbf{y}^{(\mathbf{i})},\mathbf{f}(\mathbf{x}^{(\mathbf{i})}) = (\mathbf{y}^{(\mathbf{i})}\mathbf{f}(\mathbf{x}^{(\mathbf{i})}) - \mathbf{1})^{\mathbf{2}}$$

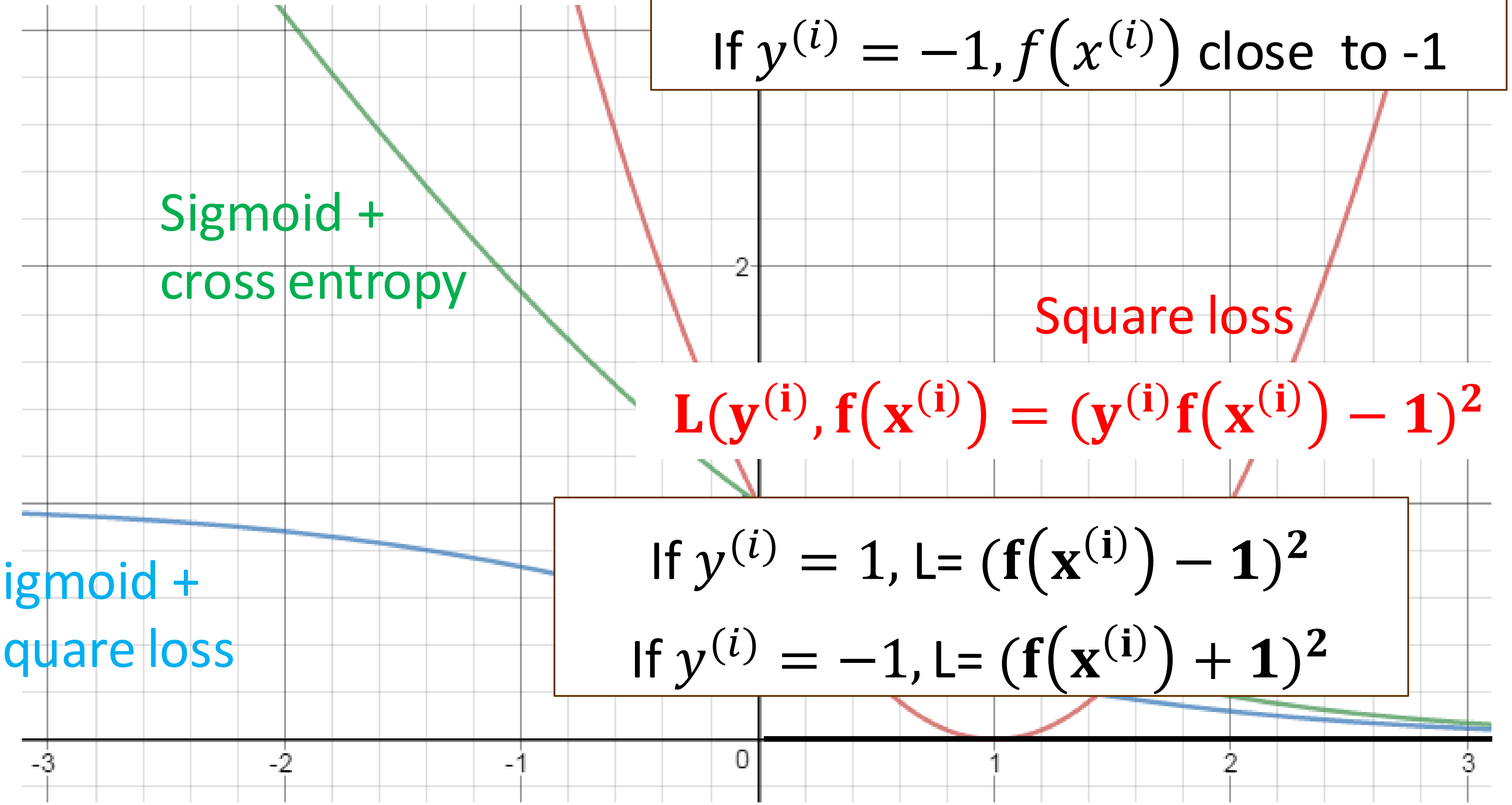
Larger value, smaller loss $y^{(i)}f(x^{(i)})$

损失函数 Loss function

Square Loss:

If $y^{(i)} = 1, f\left(x^{(i)}\right)$ close to 1

If $y^{(i)} = -1, f\left(x^{(i)}\right)$ close to -1

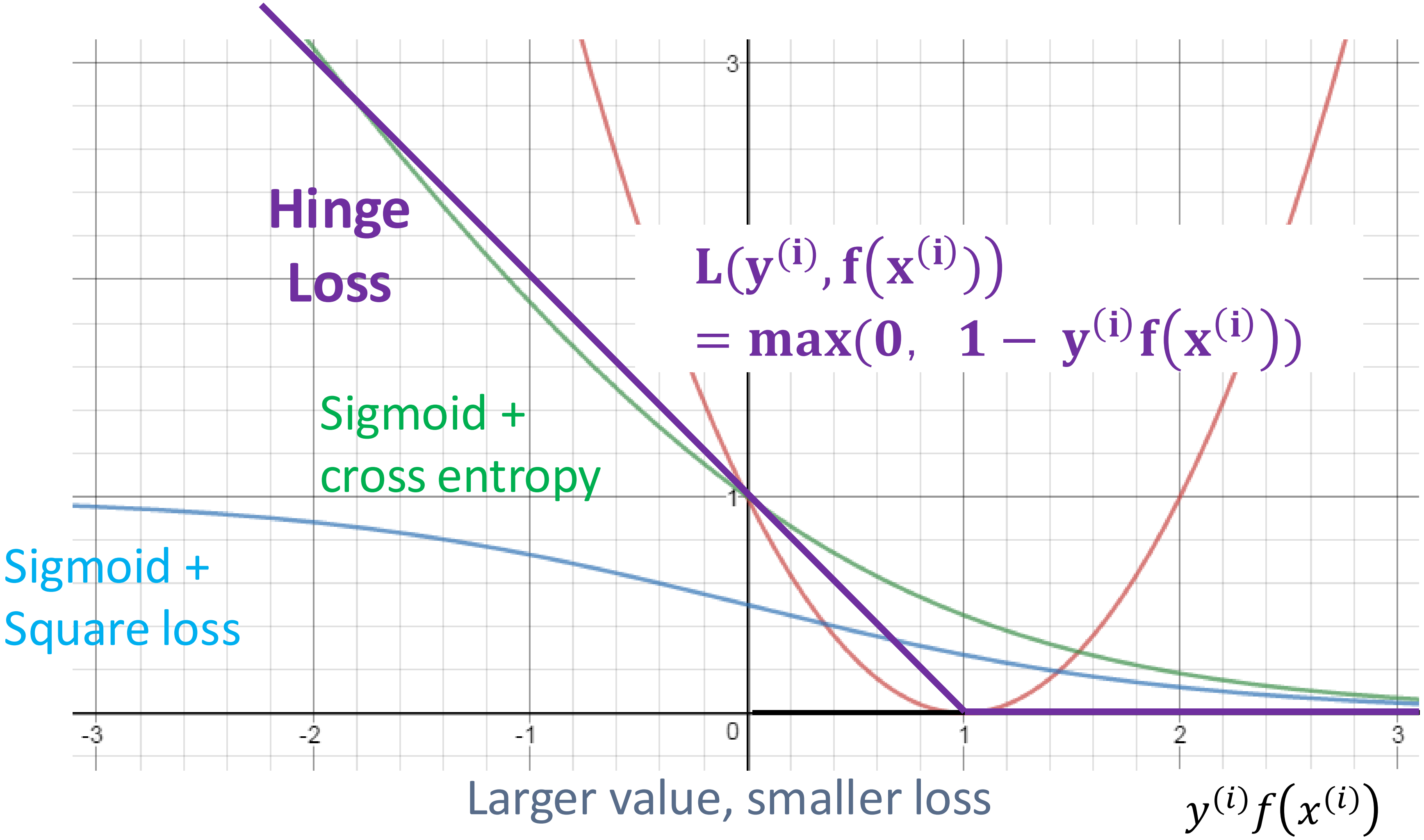


$$\mathbf{L}\left(\mathbf{y}^{(\mathbf{i})}, \mathbf{f}\left(\mathbf{x}^{(\mathbf{i})}\right)\right) = \left(\mathbf{y}^{(\mathbf{i})} \mathbf{f}\left(\mathbf{x}^{(\mathbf{i})}\right) - \mathbf{1}\right)^{\mathbf{2}}$$

If $y^{(i)} = 1$, L= $\left(\mathbf{f}\left(\mathbf{x}^{(\mathbf{i})}\right) - \mathbf{1}\right)^{\mathbf{2}}$

If $y^{(i)} = -1$, L= $\left(\mathbf{f}\left(\mathbf{x}^{(\mathbf{i})}\right) + \mathbf{1}\right)^{\mathbf{2}}$

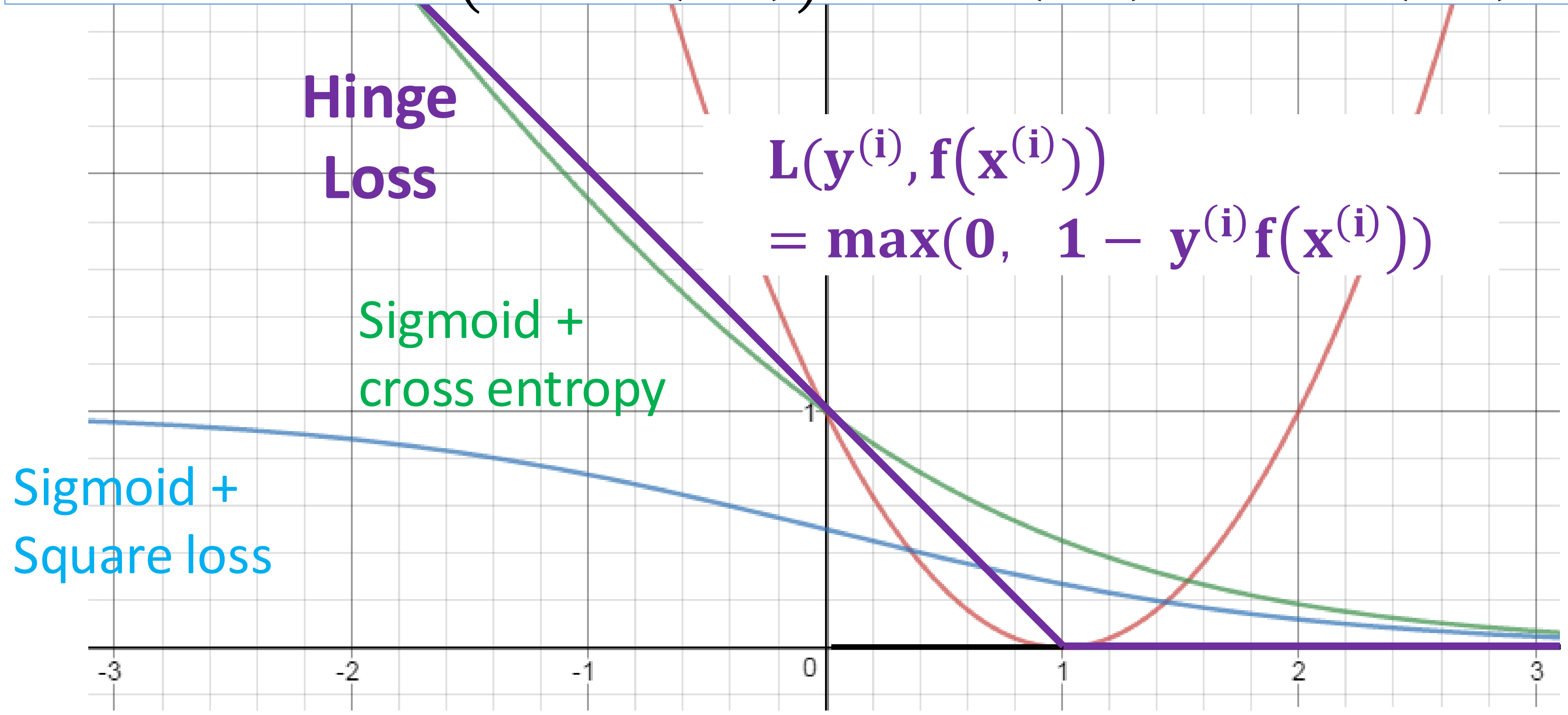
损失函数 Loss function



损失函数 Loss function

If $y^{(i)} = 1,\ \max\left(0, 1 - f\left(x^{(i)}\right)\right)$ $\quad 1 - f\left(x^{(i)}\right) < 0 \quad f\left(x^{(i)}\right) > 1$

If $y^{(i)} = -1, \max\left(0, 1 + f\left(x^{(i)}\right)\right)$ $\quad 1 + f\left(x^{(i)}\right) < 0 \quad f\left(x^{(i)}\right) < -1$



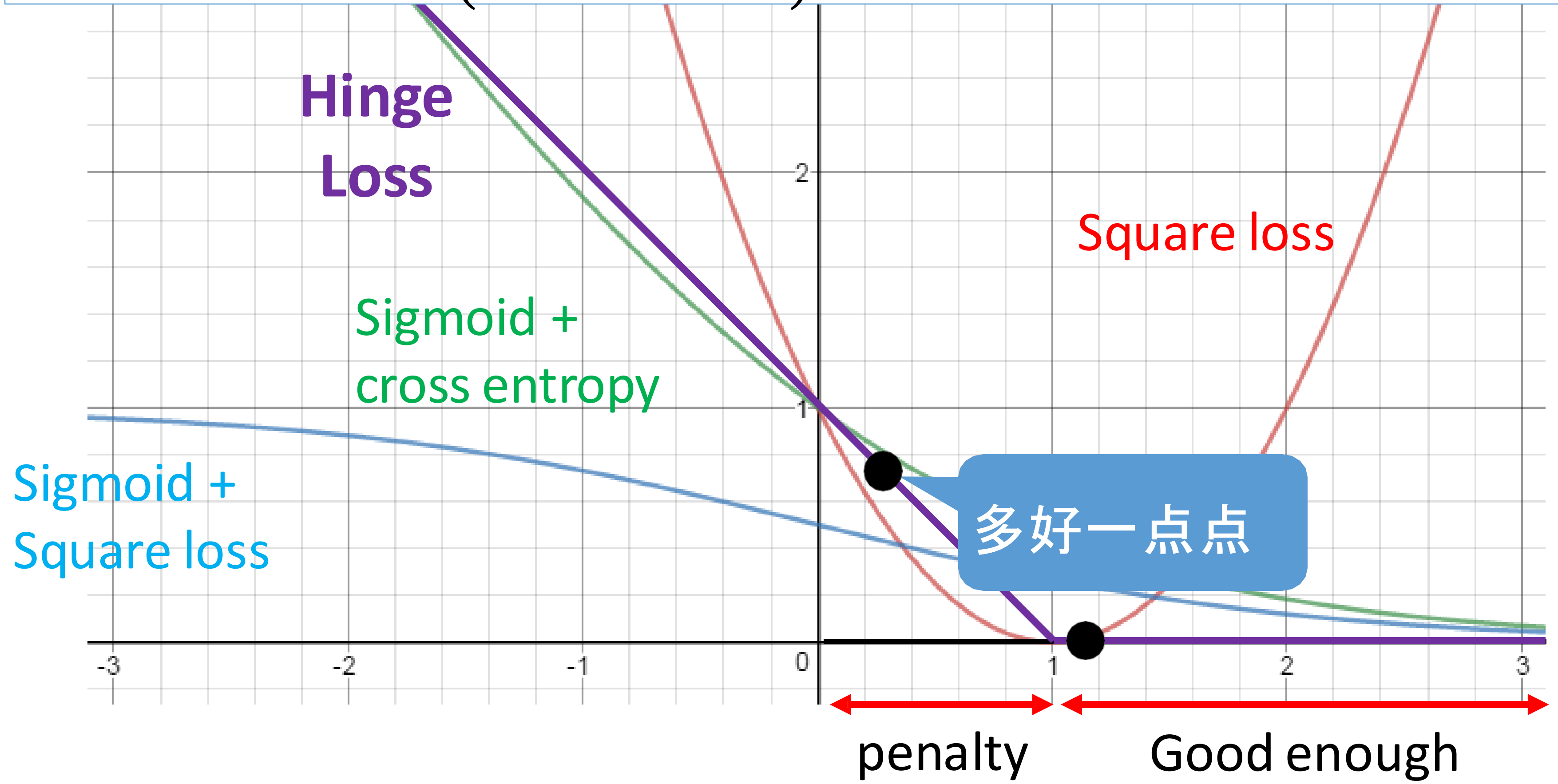
$$\mathbf{L}\left(\mathbf{y}^{(\mathbf{i})}, \mathbf{f}\left(\mathbf{x}^{(\mathbf{i})}\right)\right) = \mathbf{max}\left(\mathbf{0},\ \ \mathbf{1} - \mathbf{y}^{(\mathbf{i})}\mathbf{f}\left(\mathbf{x}^{(\mathbf{i})}\right)\right)$$

Larger value, smaller loss $\quad y^{(i)} f\left(x^{(i)}\right)$

损失函数 Loss function

If $y^{(i)} = 1, \ \max\left(0, 1 - f\left(x^{(i)}\right)\right) \quad 1 - f\left(x^{(i)}\right) < 0 \quad f\left(x^{(i)}\right) > 1$

If $y^{(i)} = -1, \max\left(0, 1 + f\left(x^{(i)}\right)\right) \quad 1 + f\left(x^{(i)}\right) < 0 \quad f\left(x^{(i)}\right) < -1$



二分类 Binary Classification

- Step 1: Function (Model)

$$f(x) = \sum_{j} w_j x_j + b$$

- Step 2: Cost function

$$C(f) = \sum_{n} L\left(f\left(x^{(i)}\right), y^{(i)}\right)$$

线性SVM Linear SVM

- Step 1: Function (Model)

New w

$$f(x) = \sum_j w_j x_j + b = \boxed{\begin{bmatrix} w \\ b \end{bmatrix}} \cdot \boxed{\begin{bmatrix} x \\ 1 \end{bmatrix}} = w^T x$$

New x

- Step 2: Cost function

$$= \sum_i L\left(f\left(x^{(i)}\right), y^{(i)}\right) + \lambda \|w\|_2$$

$$L(y^{(i)}, f\left(x^{(i)}\right) = max(0, 1 - \; y^{(i)} f\left(x^{(i)}\right))$$

Hinge loss

线性SVM Linear SVM

- Step 1: Function (Model)

$$f(x) = \sum_j w_j x_j + b = \boxed{\begin{bmatrix} w \\ b \end{bmatrix}} \cdot \boxed{\begin{bmatrix} x \\ 1 \end{bmatrix}}$$

- Step 2: Cost function

$$= \sum_i L\left(f\left(x^{(i)}\right), y^{(i)}\right) + \lambda \|w\|_2$$

convex

$$L(y^{(i)}, f\left(x^{(i)}\right) = max(0,1 - y^{(i)} f\left(x^{(i)}\right))$$

- Step 3: gradient descent?

线性SVM Linear SVM

- Step 1: Function (Model)

$$f(x) = \sum_j w_j x_j + b = \boxed{\begin{bmatrix} w \\ b \end{bmatrix}} \cdot \boxed{\begin{bmatrix} x \\ 1 \end{bmatrix}}$$

- Step 2: Cost function

convex

$$= \sum_i L\left(f\left(x^{(i)}\right), y^{(i)}\right) + \lambda \|w\|_2$$

$$L(y^{(i)}, f\left(x^{(i)}\right) = max(0, 1 - y^{(i)} f\left(x^{(i)}\right))$$

Compared with logistic regression,
linear SVM has different _(__________)_?

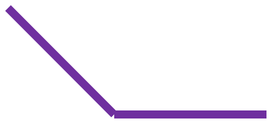
线性SVM Linear SVM

Ignore regularization for simplicity

$$\sum_i L\left(f\left(x^{(i)}\right), y^{(i)}\right) \qquad L\left(y^{(i)}, f\left(x^{(i)}\right)\right) = max\left(0, 1 - y^{(i)} f\left(x^{(i)}\right)\right)$$

$$\frac{\partial L\left(f\left(x^{(i)}\right), y^{(i)}\right)}{\partial w_j} = \frac{\partial L\left(f\left(x^{(i)}\right), y^{(i)}\right)}{\partial f\left(x^{(i)}\right)} \underbrace{\frac{\partial f\left(x^{(i)}\right)}{\partial w_j}}_{x_j^i} \qquad \boxed{f(x^n) = w^T \cdot x^n}$$

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



$$\frac{\partial \sum_i L\left(y^{(i)}, f\left(x^{(i)}\right)\right)}{\partial w_j} = \sum_i -\delta\left(y^{(i)} f\left(x^{(i)}\right) < 1\right) y^{(i)} x_j^i$$

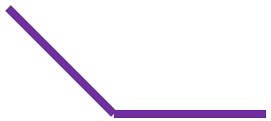
线性SVM Linear SVM

Ignore regularization for simplicity

$$\sum_i L(f(x^{(i)}), y^{(i)}) \qquad L(y^{(i)}, f(x^{(i)})) = max(0, 1 - \ y^{(i)} f(x^{(i)}))$$

$$\frac{\partial L(f(x^{(i)}), y^{(i)})}{\partial w_j} = \frac{\partial L(f(x^{(i)}), y^{(i)})}{\partial f(x^{(i)})} \underbrace{\frac{\partial f(x^{(i)})}{\partial w_j}}_{x_j^i} \qquad \boxed{f(x^n) = w^T \cdot x^n}$$

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



$$\frac{\partial \sum_i L(y^{(i)}, f(x^{(i)}))}{\partial w_j} = \sum_i \underbrace{-\delta(\ y^{(i)} f(x^{(i)}) < 1)\ y^{(i)}}_{c^n(W)}\ x_j^i$$

$$w_j \leftarrow w_j - \eta \sum_i c^n(W)\, x_j^i$$

线性SVM Linear SVM

Minimizing total loss function L:

$$min \sum_{i} \varepsilon^{\mathrm{i}} + \lambda \|w\|_2$$

$$\boxed{\varepsilon^{\mathrm{i}} = max(0, 1 - y^{(i)} f(x^{(i)}))}$$

$$y^{(i)}(w^T x_j + b) \geq 1 - \varepsilon^{\mathrm{i}}$$

ε^{i}: slack variable

$$\boxed{\begin{aligned} \varepsilon^{\mathrm{i}} &\geq 0 \\ \varepsilon^{\mathrm{i}} &\geq 1 - y^{(i)} f(x^{(i)}) \end{aligned}}$$

$$\Rightarrow y^{(i)} f(x^{(i)}) \geq 1 - \varepsilon^{\mathrm{i}}$$

线性SVM Linear SVM

Minimizing total loss function L:

$$min \sum_{i} \varepsilon^{\mathrm{i}} + \lambda \|w\|_2$$

$$s.t.\ y^{(i)}(w^T x_j + b) \geq 1 - \varepsilon^{\mathrm{i}}$$

ε^{i}: slack variable

线性SVM Linear SVM

Minimizing total loss function L:

$$min \sum_i \varepsilon^{\mathrm{i}} + \lambda \|w\|_2$$

$$s.t.\ y^{(i)}(w^T x_j + b) \geq 1 - \varepsilon^{\mathrm{i}}$$

ε^{i}: slack variable

$$min \frac{1}{2}\|w\|_2 + C \sum_i \varepsilon^{\mathrm{i}}$$

$$s.t.\ y^{(i)}(w^T x_j + b) \geq 1 - \varepsilon^{\mathrm{i}}$$

支持向量 Support vectors

$$w^{(*)} = \sum_{\mathrm{n}} a_n^* x^n$$

$a^{(*)}$ may be sparse

Linear combination of data points

$x^{(i)}$ with non-zero $a^{(*)}$ are support vectors

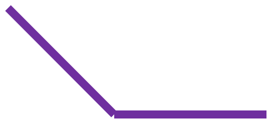
线性SVM Linear SVM

Ignore regularization for simplicity

$$\sum_i L\left(f\left(x^{(i)}\right), y^{(i)}\right) \qquad L\left(y^{(i)}, f\left(x^{(i)}\right)\right) = max\left(0, 1 - y^{(i)} f\left(x^{(i)}\right)\right)$$

$$\frac{\partial L\left(f\left(x^{(i)}\right), y^{(i)}\right)}{\partial w_j} = \frac{\partial L\left(f\left(x^{(i)}\right), y^{(i)}\right)}{\partial f\left(x^{(i)}\right)} \underbrace{\frac{\partial f\left(x^{(i)}\right)}{\partial w_j}}_{x_j^i} \qquad \boxed{f(x^n) = w^T \cdot x^n}$$

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



$$\frac{\partial \sum_{\mathrm{i}} L\left(y^{(i)}, f\left(x^{(i)}\right)\right)}{\partial w_j} = \sum_{\mathrm{i}} \underbrace{-\delta\left(y^{(i)} f\left(x^{(i)}\right) < 1\right) y^{(i)}}_{c^n(W)} x_j^i$$

$$w_j \leftarrow w_j - \eta \sum_{\mathrm{i}} c^n(W)\, x_j^i$$

支持向量 Support vectors

$$w^{(*)} = \sum_{\mathrm{n}} a_n^* x^n$$

$a^{(*)}$ may be sparse

Linear combination of data points

$x^{(i)}$ with non-zero $a^{(*)}$ are support vectors

$$w_1 = w_1 - \textstyle\sum_i c^n(w)\, x_1^n$$

$$w_i = w_i - \textstyle\sum_i c^n(w)\, x_i^n$$

$$w_k = w_k - \sum_i c^n(w)\, x_k^n$$

线性SVM Linear SVM

$$w^{(*)} = \sum_{\mathrm{n}} a_n^* x^n$$

$a^{(*)}$ may be sparse

Linear combination of data points

$x^{(i)}$ with non-zero $a^{(*)}$ are support vectors

$$w_1 = w_1 - \sum_i c^n(w)\, x_1^n$$

$$w_i = w_i - \sum_i c^n(w)\, x_i^n$$

$$w_k = w_k - \sum_i c^n(w)\, x_k^n$$

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

Hinge loss: usually zero

线性SVM Linear SVM

$$w^{(*)} = \sum_{\text{n}} a_n^* x^n$$

$a^{(*)}$ may be sparse

Linear combination of data points

$x^{(i)}$ with non-zero $a^{(*)}$ are support vectors

$$w_1 = w_1 - \sum_i c^n(w)\, x_1^n$$

$$w_i = w_i - \sum_i c^n(w)\, x_i^n$$

$$w_k = w_k - \sum_i c^n(w)\, x_k^n$$

If w initialized as 0

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

Hinge loss: usually zero

线性SVM Linear SVM

$$w^{(*)} = \sum_{\text{n}} a_n^* x^n$$

$a^{(*)}$ may be sparse

Linear combination of data points

$x^{(i)}$ with non-zero $a^{(*)}$ are support vectors

$$w_1 = w_1 - \sum_i c^n(w)\, x_1^n$$

$$w_i = w_i - \sum_i c^n(w)\, x_i^n$$

$$w_k = w_k - \sum_i c^n(w)\, x_k^n$$

If w initialized as 0

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

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

Hinge loss: usually zero

逻辑斯蒂回归求解 Logistic regression solution

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[\begin{array}{c} -y^{(i)} log\left((f_\theta(x^{(i)})\right) \\ -(1-y^{(i)})\, log\left(1-(f_\theta(x^{(i)})\right) \end{array}\right]$$

Want $\{ \min_\theta J(\theta)$:

Repeat

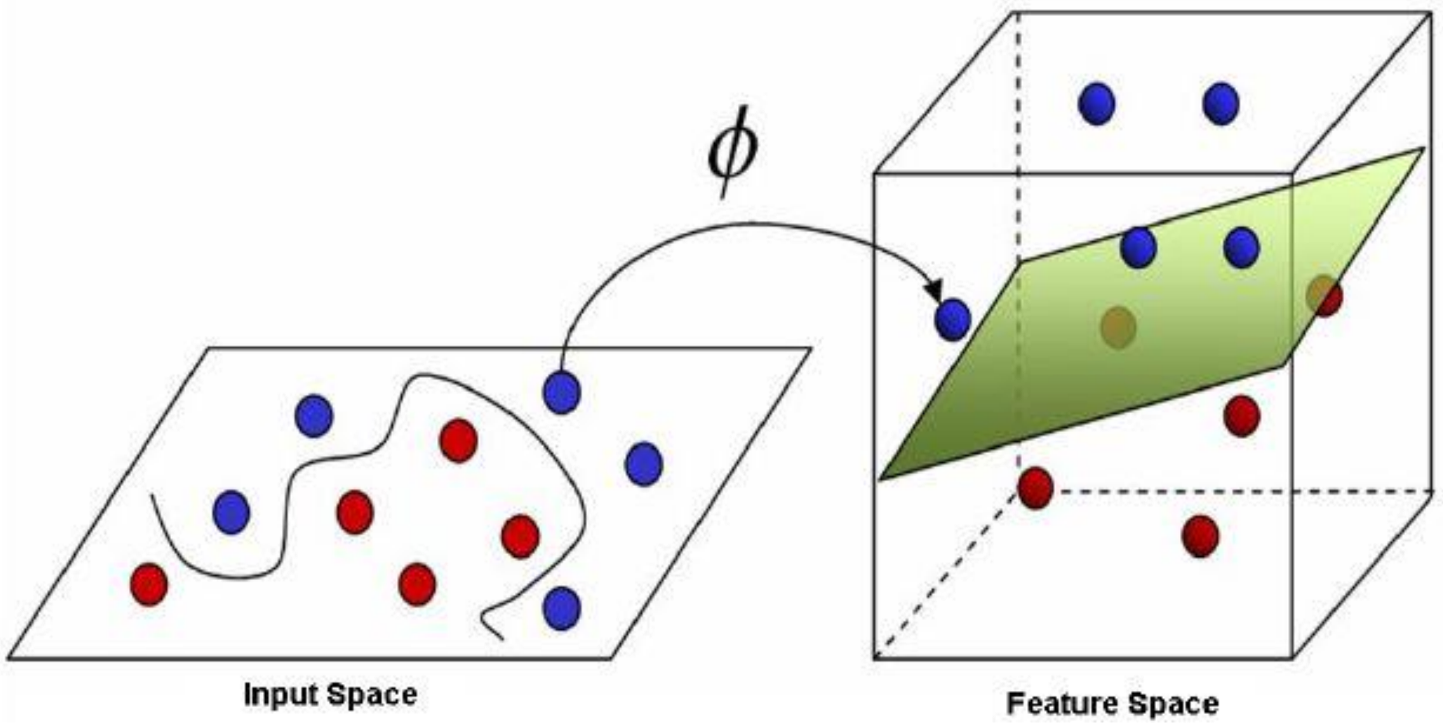
$$\theta_j := \theta_j - a\frac{1}{N}\sum_{i=1}^{N}(f_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

(simultaneously update all θ_j)

$\}$

Algorithm looks identical to linear regression!

核技巧 Kernel Trick



核技巧 Kernel Trick

$$w = \sum_n a_n x^n = Xa \qquad X = \begin{bmatrix} x^1 & x^2 & \cdots\cdots & x^N \end{bmatrix} \qquad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}$$

$$w = X\boldsymbol{\alpha}$$

Step 1: $f(x) = w^T x \Rightarrow f(x) = \boldsymbol{\alpha}^T X^T x$

$$[\alpha_1 \quad \cdots \quad \alpha_N] \qquad \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^N \end{bmatrix} \qquad x$$

$$f(x) = \sum_n a_n (x^n . x) \qquad \begin{bmatrix} x^1 \cdot x \\ x^2 \cdot x \\ \vdots \\ x^N \cdot x \end{bmatrix}$$

$$= \sum_n a_n K(x^n . x)$$

核技巧 Kernel Trick

Step 1: $f(x) \;\; = \sum_n a_n K(x^n . x)$ Find $a_1^*, a_2^*, \ldots a_n^*,$

Step 2, 3: Find $a_1^*, \ldots, a_n^*, \ldots, a_N^*$, minimizing loss function L

$$L(f) = \sum_i L\left(f\left(x^{(i)}\right), y^{(i)}\right)$$

$$= \sum_i L\left(\sum_n a_n K(x^n . x) \;, y^{(i)}\right)$$

We only need to know the inner project between a pair of vectors x and z

Kernel Trick

常用核函数 Kernel Function

- Liner kernel

$$K(x,z) = x^T z$$

- Polynomial kernel

$$K(x,z) = (x^T z)^d$$

- Gaussian kernel / Radial Basis Function Kernel

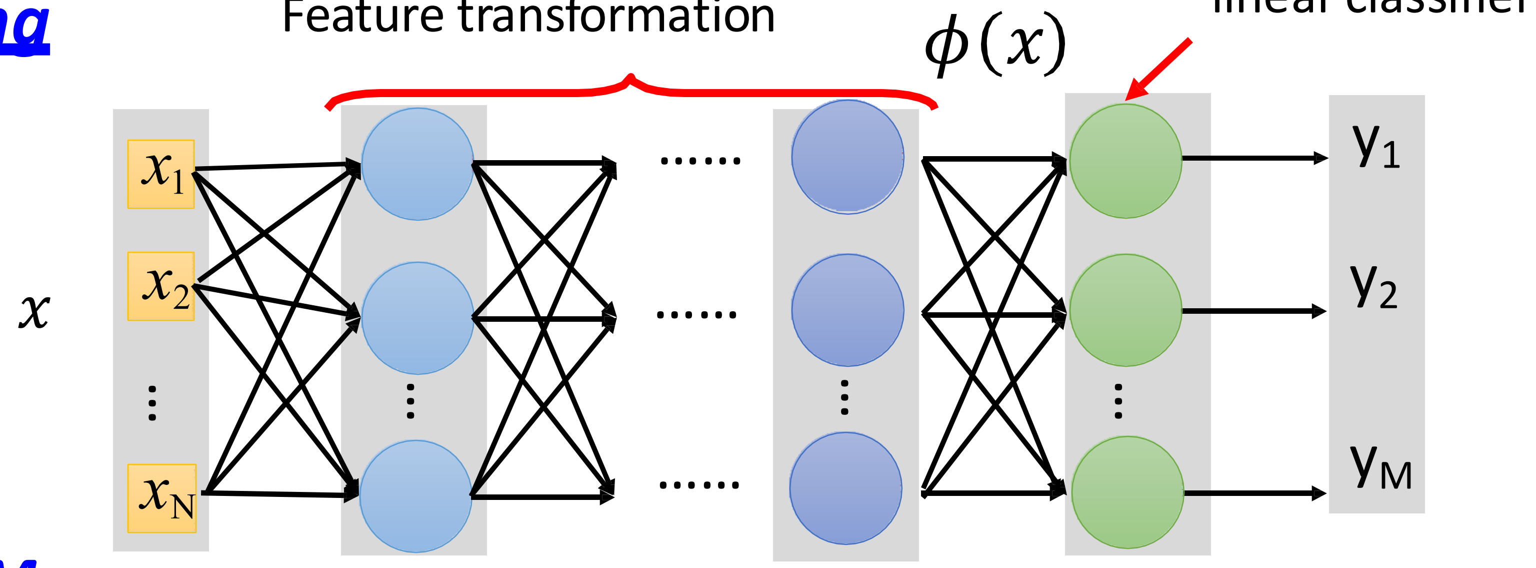
$$K(x,z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)$$

- Sigmoid kernel

$$K(x,z) = \tanh(\beta x^T z + \theta)$$

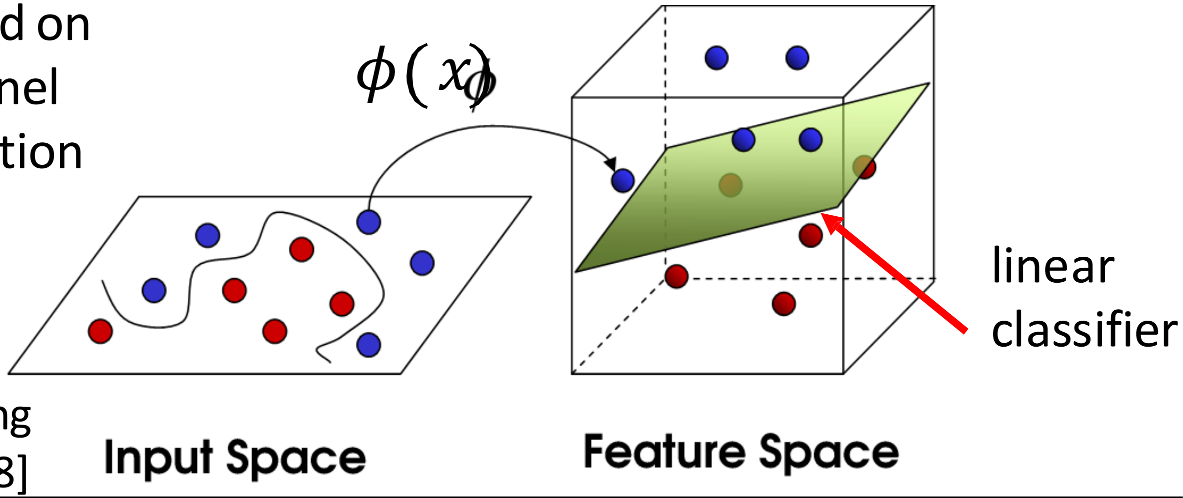
深度学习和支持向量机 Deep learning VS SVM

Deep Learning



SVM

Based on kernel function



Multiple Kernel learning
[Alpaydin，Chapter 13.8]

SVM 软件包

- LIBSVM
  http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- LIBLINEAR
  http://www.csie.ntu.edu.tw/~cjlin/liblinear/
- SVM^{light}、SVM^{perf}、SVM^{struct}
  http://svmlight.joachims.org/svm_struct.html
- Pegasos
  http://www.cs.huji.ac.il/~shais/code/index.html

- Demo

- Support Vector Machine (dash.gallery)

SVM 方法 SVM related methods

- Support Vector Regression (SVR)
  - [Bishop chapter 7.1.4]
- Ranking SVM
  - [Alpaydin, Chapter 13.11]
- One-class SVM
  - [Alpaydin, Chapter 13.11]

- Support Vector Machine (dash.gallery)