第3讲：线性模型

师 佳

化学化工学院

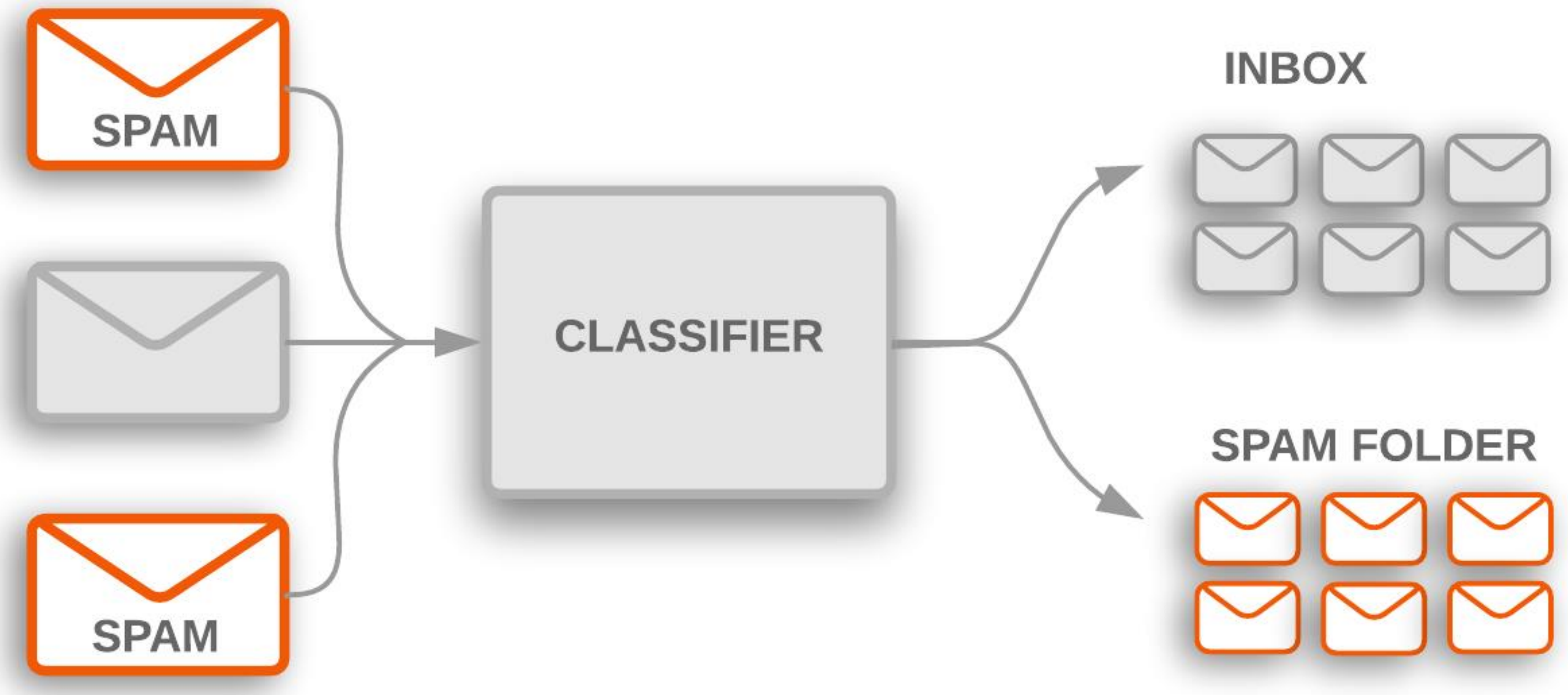
化学工程与生物工程系

纲 要

- 分类问题示例
- 线性分类模型
  - 概述
  - Logistic 回归
  - Softmax 回归
  - 感知器
  - 支持向量机（SVM）
- 总结

分类问题示例

- ➢两分类问题
  - ➢垃圾邮件过滤



分类问题示例

➢两分类问题

➢文本语意分类

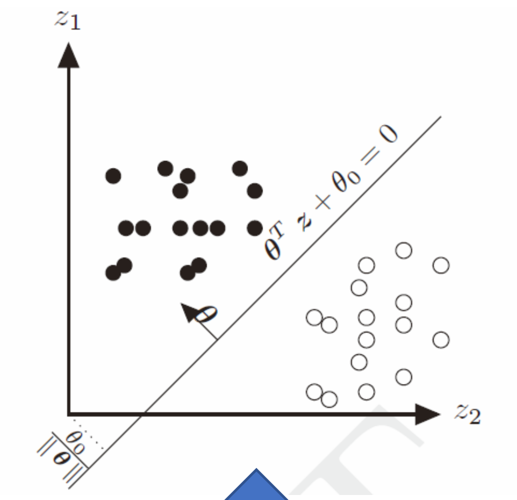
D_1：“我喜欢读书”

D_2：“我讨厌读书”

	我	喜欢	讨厌	读书
D_1	1	1	0	1
D_2	1	0	1	1

$$\boldsymbol{x}_1 = [1\ 1\ 0\ 1\ ]^{\mathrm{T}},$$

$$\boldsymbol{x}_2 = [1\ 0\ 1\ 1\ ]^{\mathrm{T}}.$$

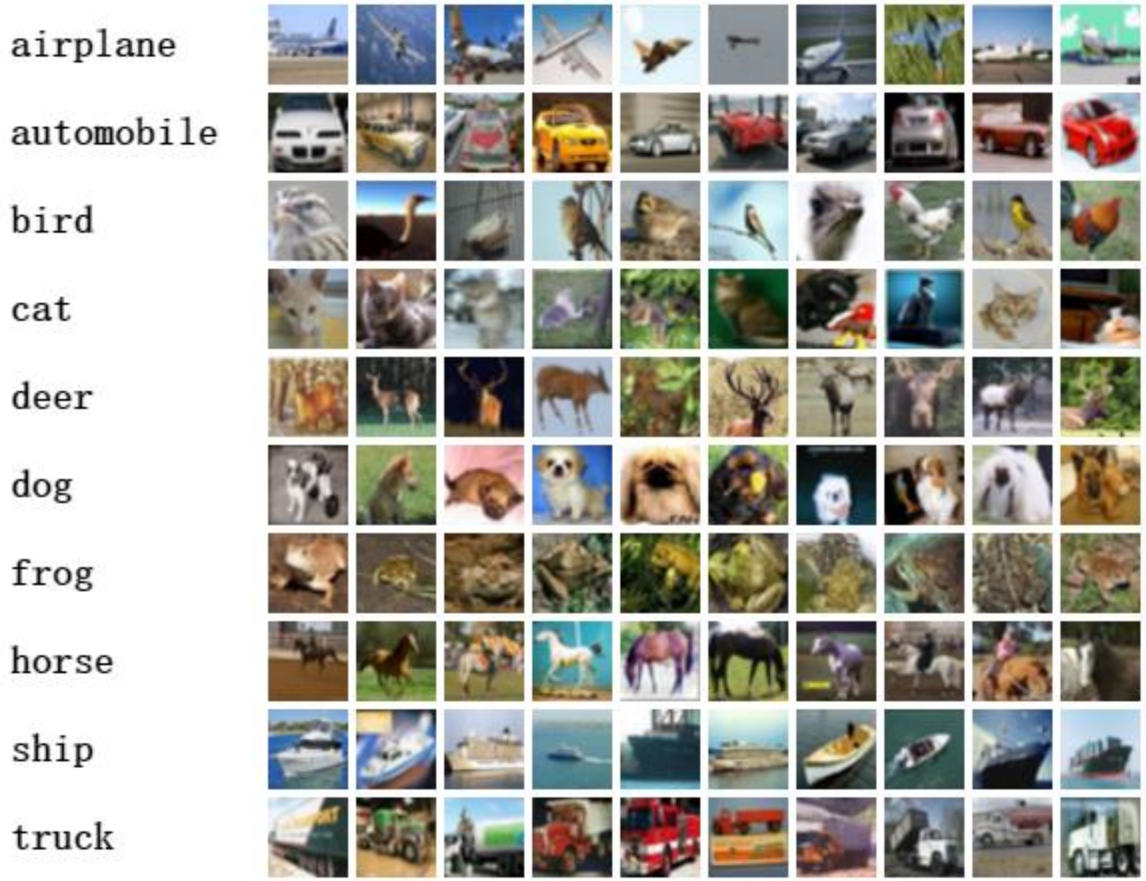


爱读书

不爱读书

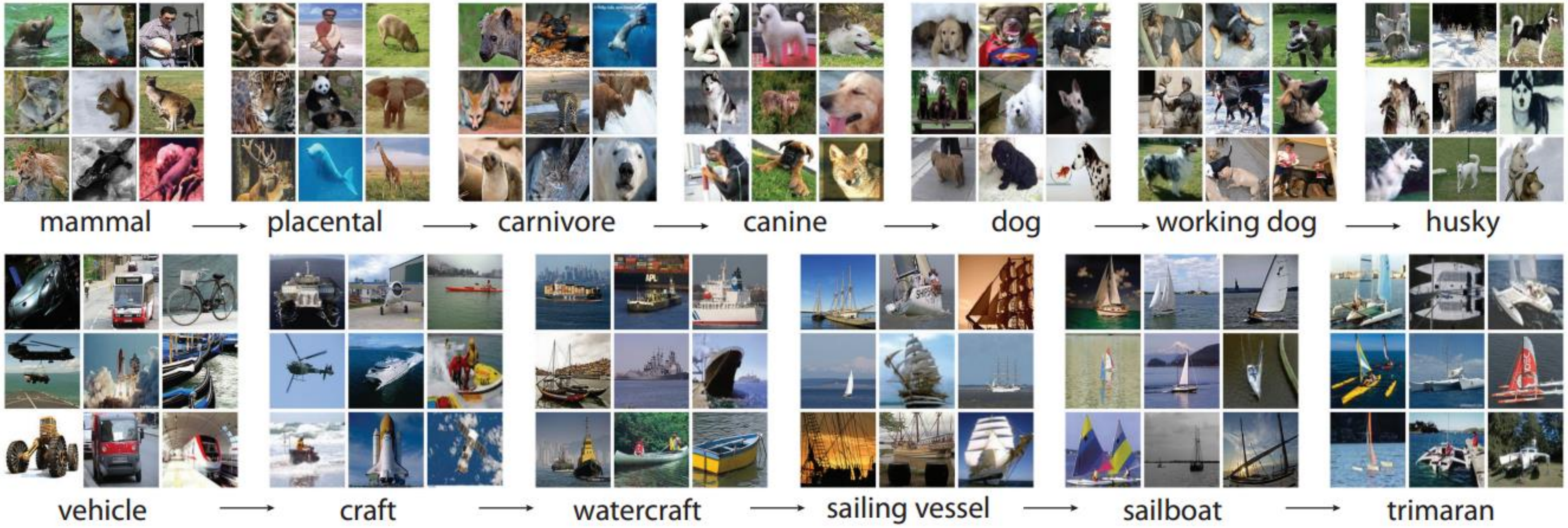
分类问题示例

- 多分类问题
  - 数据集：CIFAR-10
  - 60000张32x32色彩图像，共10类
  - 每类6000张图像



分类问题示例

- 多分类问题
  - 数据集：ImageNet
  - 14,197,122 images, 21841 synsets



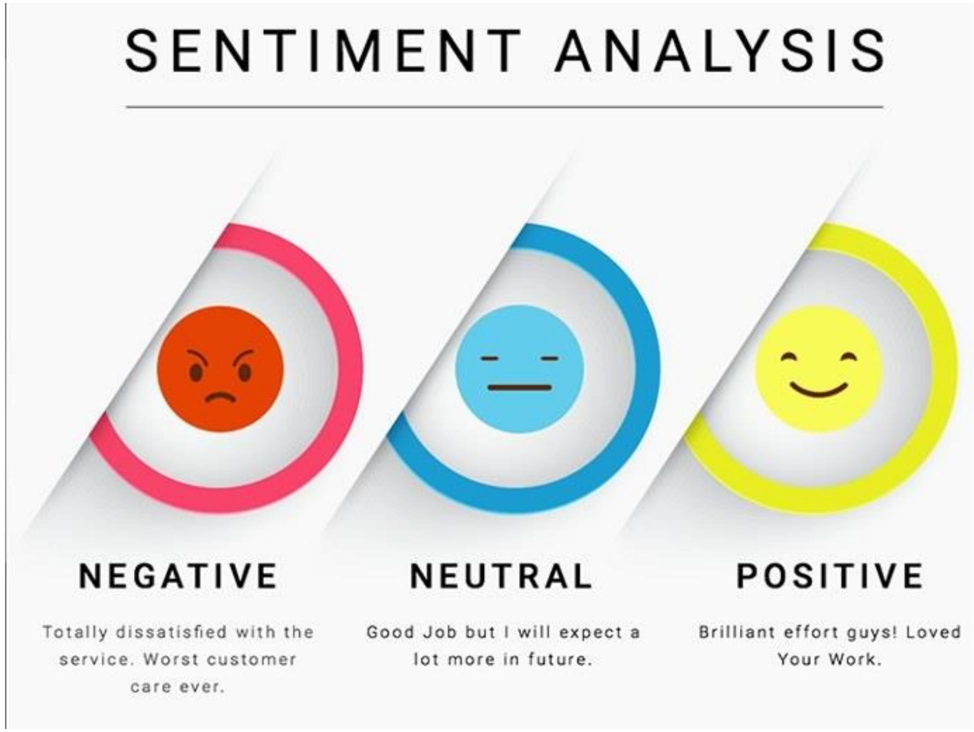
分类问题示例

➢多分类问题

➢文档归类

分类问题示例

- **多分类问题**
  - **文本情感分类**

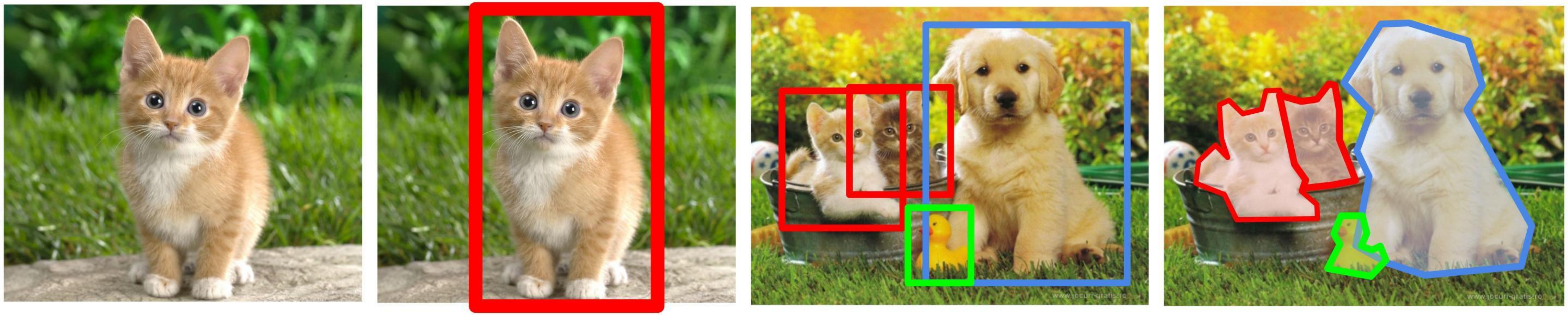


Review (X)	Rating (Y)
"This movie is fantastic! I really like it because it is so good!"	★★★★☆
"Not to my taste, will skip and watch another movie"	★★☆☆☆
"This movie really sucks! Can I get my money back please?"	★☆☆☆☆

分类问题示例

- ➢分类问题的扩展
  - ➢图像分类、目标检测、实例分割

Classification | Classification + Localization | Object Detection | Instance Segmentation



CAT | CAT | CAT, DOG, DUCK | CAT, DOG, DUCK

Single object | Multiple objects

线性分类模型

➢概述

➢**线性分类模型**：给定一个 D 维样本 $\boldsymbol{x} = [x^1, \cdots, x^D]^\mathrm{T}$， 其线性组合函数为

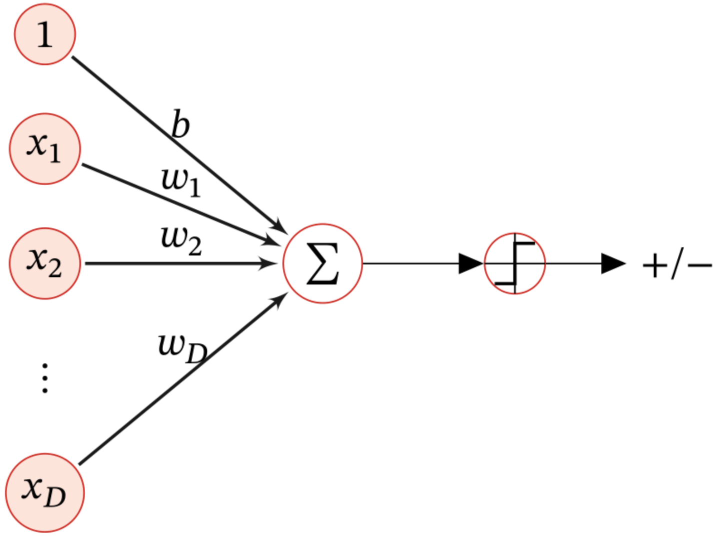
$$
\begin{aligned}
f(\boldsymbol{x};\boldsymbol{w}) &= w_1x_1 + w_2x_2 + \cdots + w_Dx_D + b \\
&= \boldsymbol{w}^\top\boldsymbol{x} + b,
\end{aligned}
$$

引入一个**非线性的决策函数**（Decision Function） $g(\cdot)$来预测输出目标

$$y = g(f(\boldsymbol{x};\boldsymbol{w}))$$

如符号函数

$$
\begin{aligned}
g(f(\boldsymbol{x};\boldsymbol{w})) &= \mathrm{sgn}(f(\boldsymbol{x};\boldsymbol{w})) \\
&\triangleq \begin{cases} +1 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) > 0, \\ -1 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) < 0. \end{cases}
\end{aligned}
$$



线性分类模型

➢概述

- ➢二分类问题所有满足$f(x;w)=0$的点组成一个**分割超平面**将特征空间一分为二，划分成两个区域，每个区域对应一个类别。
- ➢每个样本点到决策平面的**有向距离**（Signed Distance）为

$$\gamma = \frac{f(\boldsymbol{x};\boldsymbol{w})}{\|\boldsymbol{w}\|}.$$

- ➢给定N 个样本的训练集$\mathcal{D}=\{(\boldsymbol{x}^{(n)},y^{(n)})\}_{n=1,\dots,N}$，其中$y^{(n)}\in\{+1,-1\}$，线性模型试图学习到参数$\boldsymbol{w}^*$，使得对于每个样本$(\boldsymbol{x}^{(n)},y^{(n)})$尽量满足

$$y^{(n)}f(\boldsymbol{x}^{(n)};\boldsymbol{w}^*)>0, \qquad \forall n\in[1,N]$$

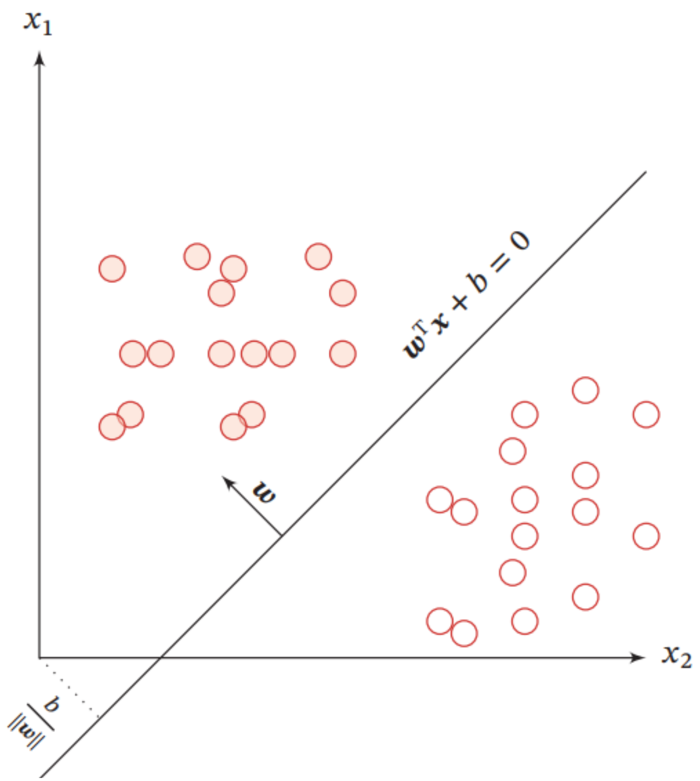


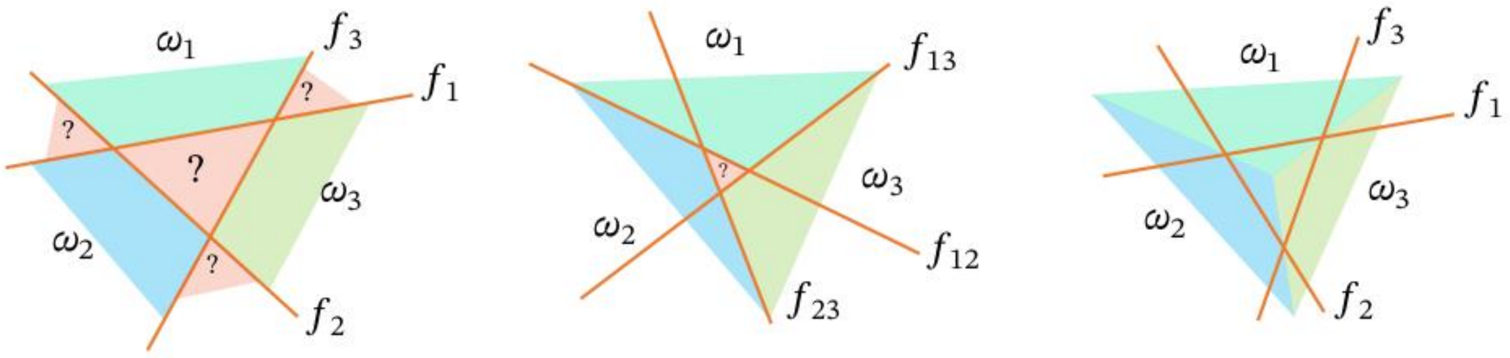
图 3.2　二分类的决策边界示例

定义 3.1 – 两类线性可分：对于训练集 $\mathcal{D}=\{(\boldsymbol{x}^{(n)},y^{(n)})\}_{n=1}^{N}$，如果存在权重向量 $\boldsymbol{w}^*$，对所有样本都满足 $yf(\boldsymbol{x};\boldsymbol{w}^*)>0$，那么训练集 $\mathcal{D}$ 是线性可分的.

线性分类模型

➢概述

➢多分类问题



(a) “一对其余”方式　(b) “一对一”方式　(c) “argmax”方式

➢ **“argmax” 方式：共需要C个判别函数** $f_c(\boldsymbol{x};\boldsymbol{w}_c)=\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}+b_c, \qquad c\in\{1,\cdots,C\}$

“argmax”方式的预测函数定义为：

$$y=\arg\max_{c=1}^{C} f_c(\boldsymbol{x};\boldsymbol{w}_c)$$

定义 3.2 – 多类线性可分： 对于训练集 $\mathcal{D}=\{(\boldsymbol{x}^{(n)},y^{(n)})\}_{n=1}^{N}$，如果存在$C$个权重向量 $\boldsymbol{w}_1^*,\cdots,\boldsymbol{w}_C^*$，使得第$c(1\le c\le C)$类的所有样本都满足 $f_c(\boldsymbol{x};\boldsymbol{w}_c^*)>f_{\tilde{c}}(\boldsymbol{x},\boldsymbol{w}_{\tilde{c}}^*),\forall\tilde{c}\neq c$，那么训练集 $\mathcal{D}$ 是线性可分的.

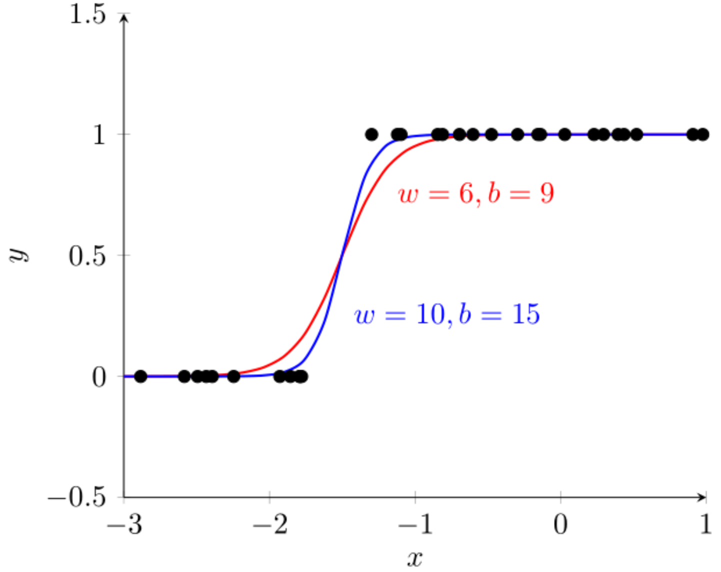
线性分类模型

➢Logistic回归

- ➢用途:两分类问题
- ➢模型：引入非线性函数$g:\mathbb{R}^D \to (0,1)$来预测类别标签的后验概率$p(y=1|\boldsymbol{x})$

$$p(y=1|\boldsymbol{x}) = g(f(\boldsymbol{x};\boldsymbol{w})) = \sigma(\boldsymbol{w}^\top \boldsymbol{x})$$

$$\triangleq \frac{1}{1+\exp(-\boldsymbol{w}^\top \boldsymbol{x})}$$



- ➢学习准则：交叉熵损失函数
  - ➢给定N 个训练样本$\{(\boldsymbol{x}^{(n)},y^{(n)})\}_{n=1,\ldots,N}$，用Logistic回归模型对每个样本$\boldsymbol{x}^{(n)}$进行预测，输出其标签为1的后验概率，记为$\hat{y}^{(n)}$，经验风险函数定义为：

$$\mathcal{R}(\boldsymbol{w}) = -\frac{1}{N}\sum_{n=1}^{N}\left(p_r(y^{(n)}=1|\boldsymbol{x}^{(n)})\log \hat{y}^{(n)} + p_r(y^{(n)}=0|\boldsymbol{x}^{(n)})\log(1-\hat{y}^{(n)})\right)$$

$$= -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\log \hat{y}^{(n)} + (1-y^{(n)})\log(1-\hat{y}^{(n)})\right)$$

线性分类模型

➢Logistic回归

➢优化算法：梯度下降法（牛顿法）

➢ 经验风险函数$\mathcal{R}(w)$关于参数w的偏导数（梯度）为：

$$
\begin{aligned}
\frac{\partial \mathcal{R}(\boldsymbol{w})}{\partial \boldsymbol{w}} &= -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\frac{\hat{y}^{(n)}(1-\hat{y}^{(n)})}{\hat{y}^{(n)}}\boldsymbol{x}^{(n)} - (1-y^{(n)})\frac{\hat{y}^{(n)}(1-\hat{y}^{(n)})}{1-\hat{y}^{(n)}}\boldsymbol{x}^{(n)}\right) \\
&= -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}(1-\hat{y}^{(n)})\boldsymbol{x}^{(n)} - (1-y^{(n)})\hat{y}^{(n)}\boldsymbol{x}^{(n)}\right) \\
&= -\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}\left(y^{(n)}-\hat{y}^{(n)}\right).
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial}{\partial \boldsymbol{w}}\log \hat{y}^{(n)} &= \frac{\partial}{\partial \boldsymbol{w}}\log \frac{1}{1+\exp\left(-\boldsymbol{w}^T\boldsymbol{x}^{(n)}\right)} \\
&= \frac{1}{\hat{y}^{(n)}}\frac{\partial}{\partial \boldsymbol{w}}\left(\frac{1}{1+\exp\left(-\boldsymbol{w}^T\boldsymbol{x}^{(n)}\right)}\right) \\
&= \frac{1}{\hat{y}^{(n)}}\frac{\exp\left(-\boldsymbol{w}^T\boldsymbol{x}^{(n)}\right)}{\left(1+\exp\left(-\boldsymbol{w}^T\boldsymbol{x}^{(n)}\right)\right)^2}\boldsymbol{x}^{(n)} \\
&= \frac{\hat{y}^{(n)}\left(1-\hat{y}^{(n)}\right)}{\hat{y}^{(n)}}\boldsymbol{x}^{(n)}
\end{aligned}
$$

➢ Logistic 回归的训练过程为：初始化 $w_0 \leftarrow 0$，然后通过下式来迭代更新参数

$$
\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \alpha\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}\left(y^{(n)}-\hat{y}_{\boldsymbol{w}_t}^{(n)}\right)
$$

线性分类模型

➢Softmax回归（多分类Logistic回归）

- ➢用途:**多分类问题**
- ➢模型：对于多类问题，类别标签$y \in \{1,2,\cdots,C\}$可以有C个取值．给定一个样本$\boldsymbol{x}$，Softmax回归预测的属于类别c的**条件概率**为

$$p(y=c|\boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{w}_c^{\top}\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_c^{\top}\boldsymbol{x})}{\sum_{c'=1}^{C}\exp(\boldsymbol{w}_{c'}^{\top}\boldsymbol{x})}$$

$$\begin{bmatrix}1.2\\0.9\\0.4\end{bmatrix} \longrightarrow \boxed{\text{Softmax}} \longrightarrow \begin{bmatrix}0.46\\0.34\\0.20\end{bmatrix}$$

向量表示

$$\hat{\boldsymbol{y}} = \mathrm{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x}) = \frac{\exp(\boldsymbol{W}^{\top}\boldsymbol{x})}{\boldsymbol{1}_C^{\top}\exp(\boldsymbol{W}^{\top}\boldsymbol{x})},$$

其中$\boldsymbol{W}=[\boldsymbol{w}^1,\cdots,\boldsymbol{w}^C]$是由$C$个类的权重向量组成的矩阵，$\boldsymbol{1}^C$ 为C 维的全1向量，$\boldsymbol{y} \in \mathbb{R}^C$为所有类别的预测条件概率组成的向量， 取预测函数为：

$$\hat{y} = \arg\max_{c=1}^{C} p(y=c|\boldsymbol{x}) = \arg\max_{c=1}^{C} \boldsymbol{w}_c^{\top}\boldsymbol{x}.$$

线性分类模型

➢Softmax回归（多分类Logistic回归）

➢学习准则：**交叉熵损失函数**

➢用C维的one-hot向量$\boldsymbol{y}\in\{0,1\}^C$来表示类别标签

$$\boldsymbol{y}=[I(1=c),I(2=c),\cdots,I(C=c)]^{\top}$$

➢ Softmax回归模型的经验风险损失函数为

$$\mathcal{R}(\boldsymbol{W})=-\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}\boldsymbol{y}_c^{(n)}\log\hat{\boldsymbol{y}}_c^{(n)}=-\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{y}^{(n)})^{\top}\log\hat{\boldsymbol{y}}^{(n)}$$

➢优化算法：**梯度下降法**

➢经验风险函数$\mathcal{R}(w)$关于参数w的偏导数（梯度）为：

$$\frac{\partial\mathcal{R}(\boldsymbol{W})}{\partial\boldsymbol{W}}=-\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}\left(\boldsymbol{y}^{(n)}-\hat{\boldsymbol{y}}^{(n)}\right)^{\top}$$

➢Softmax回归的训练过程为：初始化$W_0\leftarrow0$，然后通过下式进行迭代更新

$$\boldsymbol{W}_{t+1}\leftarrow\boldsymbol{W}_t+\alpha\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}^{(n)}\left(\boldsymbol{y}^{(n)}-\hat{\boldsymbol{y}}_{\boldsymbol{W}_t}^{(n)}\right)^{\top}\right)$$

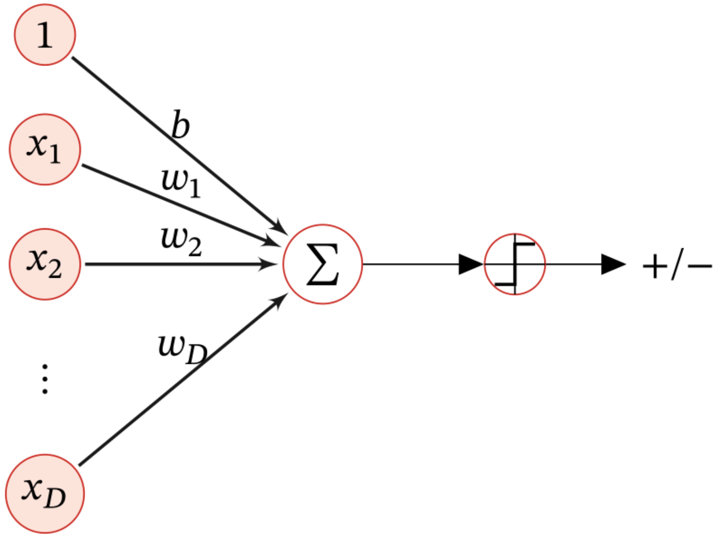
线性分类模型

➢感知器

- ➢用途：**二分类问题**
- ➢模型：**符号函数**

$$g(f(\boldsymbol{x};\boldsymbol{w})) = \operatorname{sgn}(f(\boldsymbol{x};\boldsymbol{w}))$$

$$\triangleq \begin{cases} +1 & \text{if } f(\boldsymbol{x};\boldsymbol{w}) > 0, \\ -1 & \text{if } f(\boldsymbol{x};\boldsymbol{w}) < 0. \end{cases}$$



- ➢给定N个样本的训练集：$\{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1,\dots,N}$，其中$y^{(n)} \in \{+1,-1\}$，感知器学习算法试图找到一组参数$\boldsymbol{w}*$，使得对于每个样本$(\boldsymbol{x}^{(n)}, y^{(n)})$有

$$y^{(n)}\boldsymbol{w}^{*\top}\boldsymbol{x}^{(n)} > 0, \qquad \forall n \in \{1,\cdots,N\}$$

- ➢学习准则：感知器的损失函数为

$$\mathcal{L}(\boldsymbol{w};\boldsymbol{x},y) = \max(0, -y\boldsymbol{w}^{\top}\boldsymbol{x})$$

线性分类模型

➢感知器

➢优化算法：梯度下降法

➢ 每一个样本的梯度

$$\frac{\partial \mathcal{L}(\boldsymbol{w};\boldsymbol{x},y)}{\partial \boldsymbol{w}}=\begin{cases}0 & \text{if} \quad y\boldsymbol{w}^{\top}\boldsymbol{x}>0,\\ -y\boldsymbol{x} & \text{if} \quad y\boldsymbol{w}^{\top}\boldsymbol{x}<0.\end{cases}$$

算法 3.1　两类感知器的参数学习算法

输入: 训练集 $\mathcal{D}=\{(\boldsymbol{x}^{(n)},y^{(n)})\}_{n=1}^{N}$，最大迭代次数 T

初始化：$\boldsymbol{w}_0 \leftarrow 0, k \leftarrow 0, t \leftarrow 0$；
repeat
　对训练集 $\mathcal{D}$ 中的样本随机排序；
　for $n=1\cdots N$ **do**
　　选取一个样本 $(\boldsymbol{x}^{(n)},y^{(n)})$；
　　if $\boldsymbol{w}_k^{\top}(y^{(n)}\boldsymbol{x}^{(n)})\leq 0$ **then**
　　　$\boldsymbol{w}_{k+1}\leftarrow \boldsymbol{w}_k+y^{(n)}\boldsymbol{x}^{(n)}$；
　　　$k\leftarrow k+1$；
　　end
　　$t\leftarrow t+1$；
　　if $t=T$ **then** break;　　// 达到最大迭代次数
　end
until $t=T$；

输出: $\boldsymbol{w}_k$

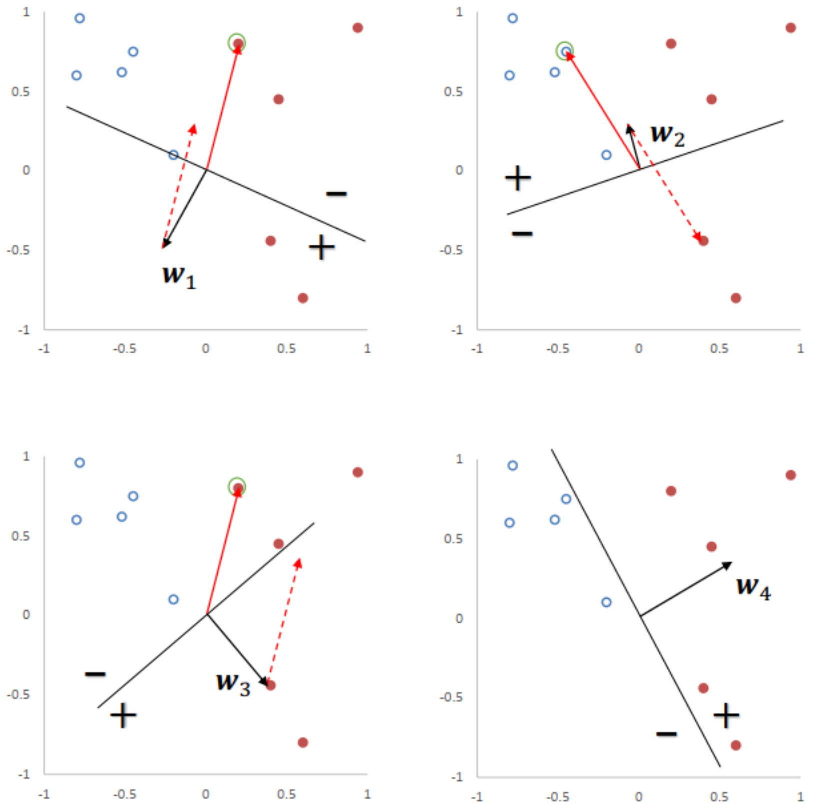


图 3.5　感知器参数学习的更新过程

线性分类模型

➢感知器

➢ 感知器收敛性

- 当数据集是两类线性可分时，对于训练集$\mathcal{D}=\{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1,\ldots,N}$， 其中$\boldsymbol{x}^{(n)}$为样本的增广特征向量，$y^{(n)}\in\{-1,1\}$，那么存在一个正的常数$\gamma(\gamma>0)$ 和权重向量$\boldsymbol{w}^*$，并且$||\boldsymbol{w}^*||=1$，对所有n都满足$(\boldsymbol{w}^*)^{\mathsf{T}}(y^{(n)}\boldsymbol{x}^{(n)})\geq\gamma$．可以证明如下定理．

定理 3.1 – 感知器收敛性：给定训练集 $\mathcal{D}=\{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$，令 R 是训练集中最大的特征向量的模，即

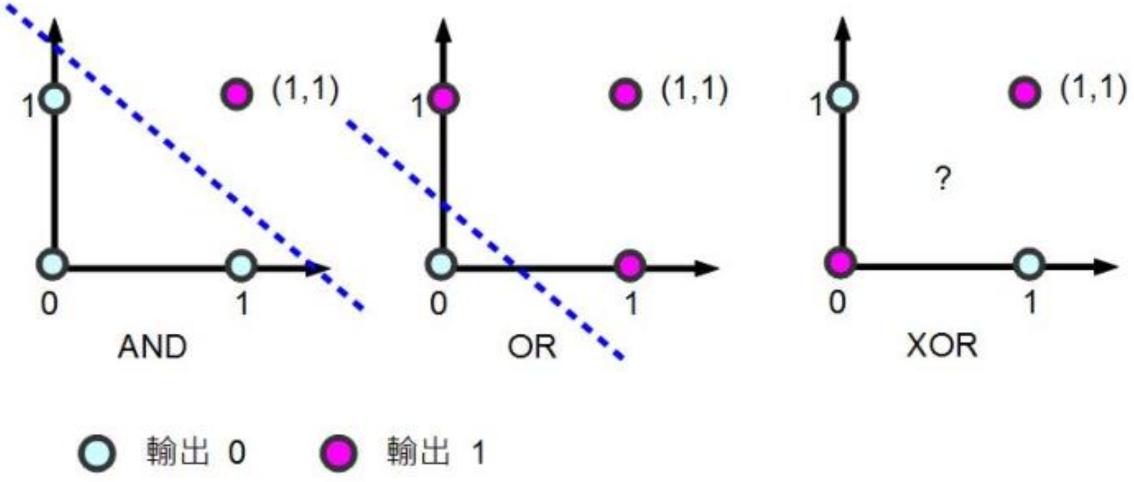
$$R=\max_{n}||x^{(n)}||.$$

如果训练集 $\mathcal{D}$ 线性可分，两类感知器的参数学习算法3.1的权重更新次数不超过 $\frac{R^2}{\gamma^2}$.

线性分类模型

➢感知器

- ➢感知器的缺陷：
  - ➢在数据集线性可分时，感知器虽然可以找到一个超平面把两类数据分开，但并不能保证其泛化能力.
  - ➢感知器对样本顺序比较敏感．每次迭代的顺序不一致时，找到的分割超平面也往往不一致.
  - ➢如果训练集不是线性可分的，就永远不会收敛.
- ➢感知器无法解决XOR问题



线性分类模型

➢支持向量机（Support Vector Machine：SVM）

- ➢用途：**二分类问题**
- ➢模型：符号函数 $g(f(\boldsymbol{x};\boldsymbol{w})) = \mathrm{sgn}(f(\boldsymbol{x};\boldsymbol{w}))$

$$\triangleq \begin{cases} +1 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) > 0, \\ -1 & \text{if} \quad f(\boldsymbol{x};\boldsymbol{w}) < 0. \end{cases}$$

定义**间隔**（Margin）γ为整个数据集D中所有样本到分割超平面的最短距离

$$\gamma = \min_n \gamma^{(n)} = \min_n \frac{|\boldsymbol{w}^\top \boldsymbol{x}^{(n)} + b|}{\|\boldsymbol{w}\|} = \min_n \frac{y^{(n)}(\boldsymbol{w}^\top \boldsymbol{x}^{(n)} + b)}{\|\boldsymbol{w}\|}$$

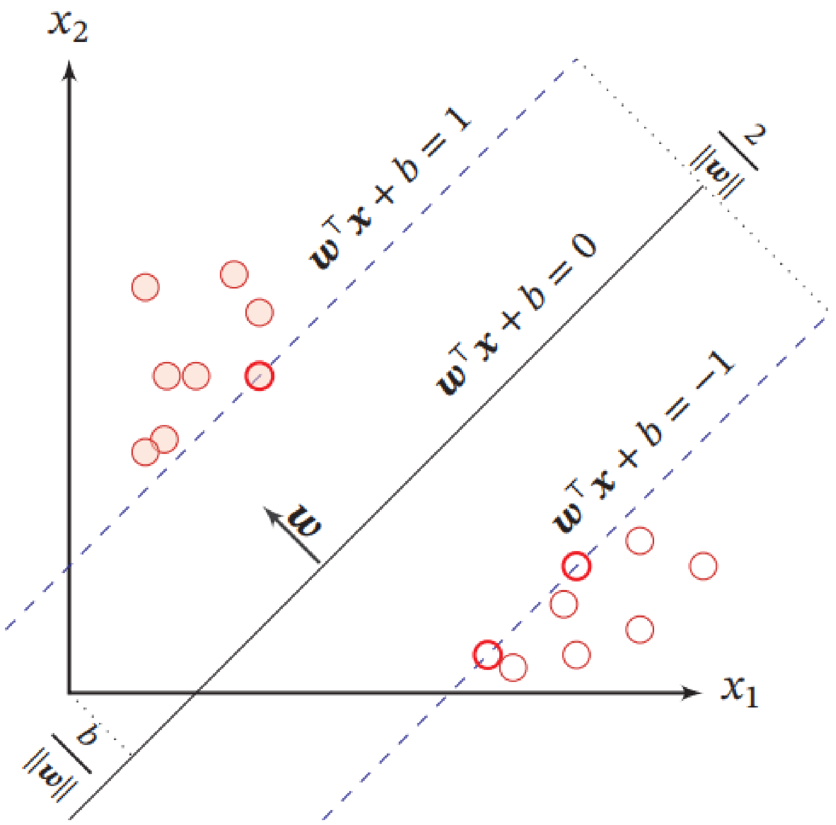


图 3.6　支持向量机示例

- ➢学习准则： 支持向量机的目标是**寻找一个超平面(w^*, b^*)使得γ最大**， 即

$$\max_{\boldsymbol{w},b} \quad \gamma$$

$$\text{s.t.} \quad \frac{y^{(n)}(\boldsymbol{w}^\top \boldsymbol{x}^{(n)} + b)}{\|\boldsymbol{w}\|} \geq \gamma, \forall n \in \{1, \cdots, N\}$$

线性分类模型

➢补充：约束优化问题

- ➢**约束优化(Constrained Optimization)问题**中变量x需要满足一些等式或不等式的约束.
- ➢**线性规划(Linear Programming)问题**:目标函数和所有的约束函数都为线性函数.
- ➢**非线性规划(Nonlinear Programming)问题**:目标函数或任何一个约束函数为非线性函数.
- ➢约束优化问题可以表示为:

$$
\begin{aligned}
\min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\
\text{s.t.} \quad & h_m(\boldsymbol{x}) = 0, \quad m = 1, \dots, M \\
& g_n(\boldsymbol{x}) \leq 0, \quad n = 1, \dots, N
\end{aligned}
$$

可行域

$$
\mathcal{D} = \mathrm{dom}(f) \cap \bigcap_{m=1}^{M} \mathrm{dom}(h_m) \cap \bigcap_{n=1}^{N} \mathrm{dom}(g_n) \subseteq \mathbb{R}^D,
$$

- ➢在非线性优化问题中，有一类比较特殊的问题是凸优化(Convex Optimization)问题
  - ➢变量 x 的可行域为凸集(Convex Set)，即对于集合中任意两点，它们的连线全部位于集合内部
  - ➢目标函数 f 也必须为凸函数，即满足

$$
f\big(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}\big) \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}),\ \forall \alpha \in [0,1].
$$

线性分类模型

➢补充：约束优化问题

- ➢约束优化问题通常使用**拉格朗日乘数法**来进行求解问题.
  - ➢ 等式约束优化问题

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & h_m(x) = 0, \quad m = 1, \ldots, M \end{aligned}$$

拉格朗日函数

$$\Lambda(x, \lambda) = f(x) + \sum_{m=1}^{M} \lambda_m h_m(x)$$

  - ➢ 如果 $f(x^*)$ 是原始约束优化问题的局部最优值，那么存在一个λ^* 使得(x^*, λ^*)为拉格朗日函数$\Lambda(x, \lambda)$的驻点，满足

$$\frac{\partial \Lambda(x,\lambda)}{\partial x} = 0, \qquad \frac{\partial \Lambda(x,\lambda)}{\partial \lambda} = 0$$

$$\Rightarrow \quad \nabla f(x) + \sum_{m=1}^{M} \lambda_m \nabla h_m(x) = 0, \qquad h_m(x) = 0, \quad \forall m = 1, \cdots, M$$

  - ➢ 拉格朗日乘数法是将一个有D个变量和M 个等式约束条件的最优化问题转换为一个有 $D + M$ 个变量的函数求驻点的问题
  - ➢ 因为驻点不一定是最小解，所以在实际应用中需根据具体问题来验证是否为最小解

线性分类模型

➢补充：约束优化问题

➢约束优化问题通常使用拉格朗日乘数法来进行求解问题.

➢ 不等式约束优化问题

$$\boxed{\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & h_m(\boldsymbol{x}) = 0, \quad m = 1, \dots, M \\ & g_n(\boldsymbol{x}) \le 0, \quad n = 1, \dots, N \end{aligned}}$$

拉格朗日函数 ⟶

$$\Lambda(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) = f(\boldsymbol{x}) + \sum_{m=1}^{M} a_m h_m(\boldsymbol{x}) + \sum_{n=1}^{N} b_n g_n(\boldsymbol{x})$$

➢ 等价于如下优化问题（主问题）

$$\boxed{\begin{aligned} \min_{\boldsymbol{x}} \max_{\boldsymbol{a},\boldsymbol{b}} \quad & \Lambda(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}), \\ \text{s.t.} \quad & \boldsymbol{b} \ge 0, \end{aligned}}$$

对偶函数 ⟶

$$\Gamma(\boldsymbol{a}, \boldsymbol{b}) = \inf_{\boldsymbol{x} \in \mathcal{D}} \Lambda(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{b}) \le \Lambda(\tilde{\boldsymbol{x}}, \boldsymbol{a}, \boldsymbol{b}) \le f(\tilde{\boldsymbol{x}})$$

令p^*是原问题的最优值

$$\Gamma(\boldsymbol{a}, \boldsymbol{b}) \le \boldsymbol{p}^*$$

拉格朗日对偶问题 ⟵

$$\boxed{\begin{aligned} \max_{\boldsymbol{a},\boldsymbol{b}} \quad & \Gamma(\boldsymbol{a}, \boldsymbol{b}), \\ \text{s.t.} \quad & \boldsymbol{b} \ge 0. \end{aligned}}$$

令d^*表示拉格朗日对偶问题的最优值

弱对偶性 ⟶ $\boldsymbol{d}^* \le \boldsymbol{p}^*$

强对偶性 ⟶ $\boxed{\boldsymbol{d}^* = \boldsymbol{p}^*}$

KKT条件 ⟶

$$\boxed{\begin{aligned} \nabla f(\boldsymbol{x}^*) + \sum_{m=1}^{M} a_m^* \nabla h_m(\boldsymbol{x}^*) + \sum_{n=1}^{N} b_n^* \nabla g_n(\boldsymbol{x}^*) = 0 \\ h_m(\boldsymbol{x}^*) = 0, \qquad m = 1, \cdots, M \\ g_n(\boldsymbol{x}^*) \le 0, \qquad n = 1, \cdots, N \\ b_n^* g_n(\boldsymbol{x}^*) = 0, \qquad n = 1, \cdots, N \\ b_n^* \ge 0, \qquad n = 1, \cdots, N \end{aligned}}$$

线性分类模型

➢支持向量机（Support Vector Machine：SVM）

➢优化算法：

$$\max_{\boldsymbol{w},b} \quad \gamma \qquad \text{s.t.} \quad \frac{y^{(n)}(\boldsymbol{w}^{\top}\boldsymbol{x}^{(n)}+b)}{\|\boldsymbol{w}\|} \geq \gamma, \forall n \in \{1,\cdots,N\}$$

限制$\|\boldsymbol{w}\|\cdot\gamma=1$

$$\max_{\boldsymbol{w},b} \quad \frac{1}{\|\boldsymbol{w}\|^2} \qquad \text{s.t.} \quad y^{(n)}(\boldsymbol{w}^{\top}\boldsymbol{x}^{(n)}+b) \geq 1, \forall n \in \{1,\cdots,N\}.$$

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 \qquad \text{s.t.} \quad 1-y^{(n)}(\boldsymbol{w}^{\top}\boldsymbol{x}^{(n)}+b) \leq 0, \qquad \forall n \in \{1,\cdots,N\}.$$

$$\Lambda(\boldsymbol{w},b,\lambda) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{n=1}^{N}\lambda_n\left(1-y^{(n)}(\boldsymbol{w}^{\top}\boldsymbol{x}^{(n)}+b)\right),$$

$$\boldsymbol{w} = \sum_{n=1}^{N}\lambda_n y^{(n)}\boldsymbol{x}^{(n)} \qquad 0 = \sum_{n=1}^{N}\lambda_n y^{(n)}$$

$$\max_{\lambda\geq 0}\Gamma(\lambda)$$

$$\Gamma(\lambda) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\lambda_m\lambda_n y^{(m)}y^{(n)}(\boldsymbol{x}^{(m)})^{\top}\boldsymbol{x}^{(n)} + \sum_{n=1}^{N}\lambda_n.$$

对偶问题

$$f(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^{*\top}\boldsymbol{x}+b^*) = \operatorname{sgn}\left(\sum_{n=1}^{N}\lambda_n^* y^{(n)}(\boldsymbol{x}^{(n)})^{\top}\boldsymbol{x}+b^*\right)$$

最优预测函数

线性分类模型

➢支持向量机（Support Vector Machine：SVM）

➢扩展：非线性可分样本

➢ 容忍部分不满足约束的样本，可以引入松弛变量（Slack Variable）ξ，将优化问题变为

$$\begin{aligned}
\min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\
\text{s.t.} \quad & 1 - y^{(n)}(\boldsymbol{w}^\top\boldsymbol{x}^{(n)} + b) - \xi_n \leq 0, \qquad \forall n \in \{1,\cdots,N\} \\
& \xi_n \geq 0, \qquad \forall n \in \{1,\cdots,N\}
\end{aligned}$$

$$\downarrow$$

$$\min_{\boldsymbol{w},b} \quad \underbrace{\sum_{n=1}^{N}\max\left(0, 1 - y^{(n)}(\boldsymbol{w}^\top\boldsymbol{x}^{(n)} + b)\right)}_{\text{Hinge 损失函数}} + \underbrace{\frac{1}{2C}\|\boldsymbol{w}\|^2}_{\text{正则化项}},$$

线性分类模型

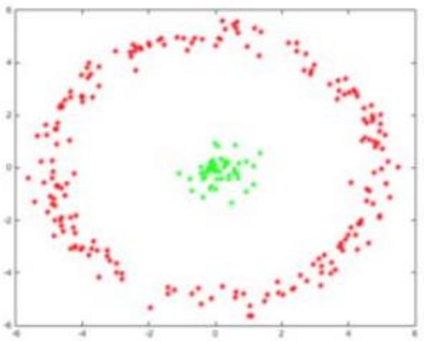
➢支持向量机（Support Vector Machine：SVM）

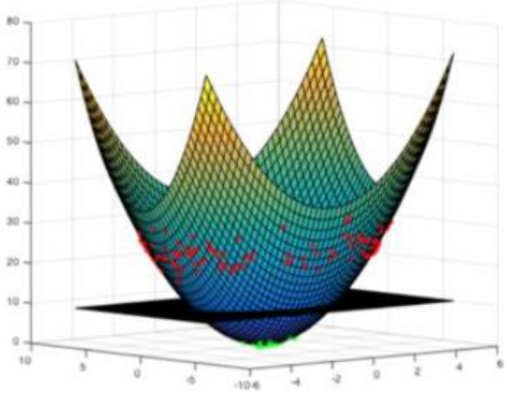
➢扩展：非线性可分样本

➢ 可以使用**核函数**（Kernel Function）隐式地将样本从原始特征空间映射到更高维的空间，并解决原始特征空间中的线性不可分问题

$$f(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^{*\top}\phi(\boldsymbol{x}) + b^*)$$

$$= \operatorname{sgn}\left(\sum_{n=1}^{N} \lambda_n^* y^{(n)} k(\boldsymbol{x}^{(n)}, \boldsymbol{x}) + b^*\right)$$





其中 $k(\boldsymbol{x}, \boldsymbol{z}) = \phi(\boldsymbol{x})^\top\phi(\boldsymbol{z})$ 称为核函数

例：

$$k(\boldsymbol{x}, \boldsymbol{z}) = (1 + \boldsymbol{x}^\top\boldsymbol{z})^2 = \phi(\boldsymbol{x})^\top\phi(\boldsymbol{z}),$$

$$\phi(\boldsymbol{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2]^\top$$

$$\Phi : \mathbf{R}^2 \to \mathbf{R}^3$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \to \Phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 + x_2^2 \end{bmatrix} \in \mathbf{R}^3$$

总结

表 3.1　几种常见的线性模型对比

线性模型	激活函数	损失函数	优化方法
线性回归	-	$(y-\boldsymbol{w}^{\top}\boldsymbol{x})^2$	最小二乘、梯度下降
Logistic 回归	$\sigma(\boldsymbol{w}^{\top}\boldsymbol{x})$	$\boldsymbol{y}\log\sigma(\boldsymbol{w}^{\top}\boldsymbol{x})$	梯度下降
Softmax 回归	$\mathrm{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x})$	$\boldsymbol{y}\log\mathrm{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x})$	梯度下降
感知器	$\mathrm{sgn}(\boldsymbol{w}^{\top}\boldsymbol{x})$	$\max(0,-y\boldsymbol{w}^{\top}\boldsymbol{x})$	随机梯度下降
支持向量机	$\mathrm{sgn}(\boldsymbol{w}^{\top}\boldsymbol{x})$	$\max(0,1-y\boldsymbol{w}^{\top}\boldsymbol{x})$	二次规划、SMO 等

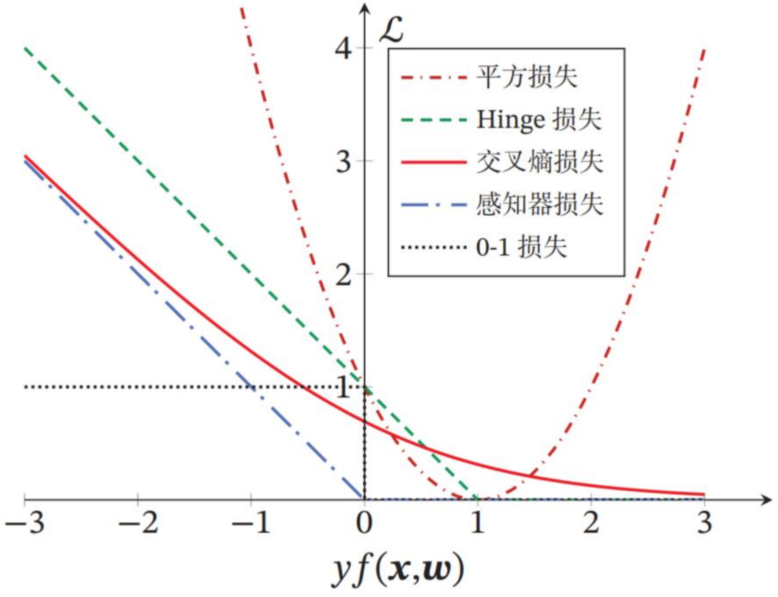


图 3.7　不同损失函数的对比

课后作业

- 完成
  - 习题3-1
  - 习题3-2
  - 习题3-6

谢 谢