《机器学习基础》

线性模型

https://github.com/styluck/mech_learn

大纲

- 线性回归模型
  - 多元回归模型
- 线性分类模型
  - Logistic Regression
  - Softmax Regression
  - Perceptron
  - SVM

线性模型

- 线性模型（Linear Model）
  - 是机器学习中应用最广泛的模型，指通过样本特征的线性组合来进行预测的模型.
  - 线性判别函数：

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

  - 线性的决策函数：

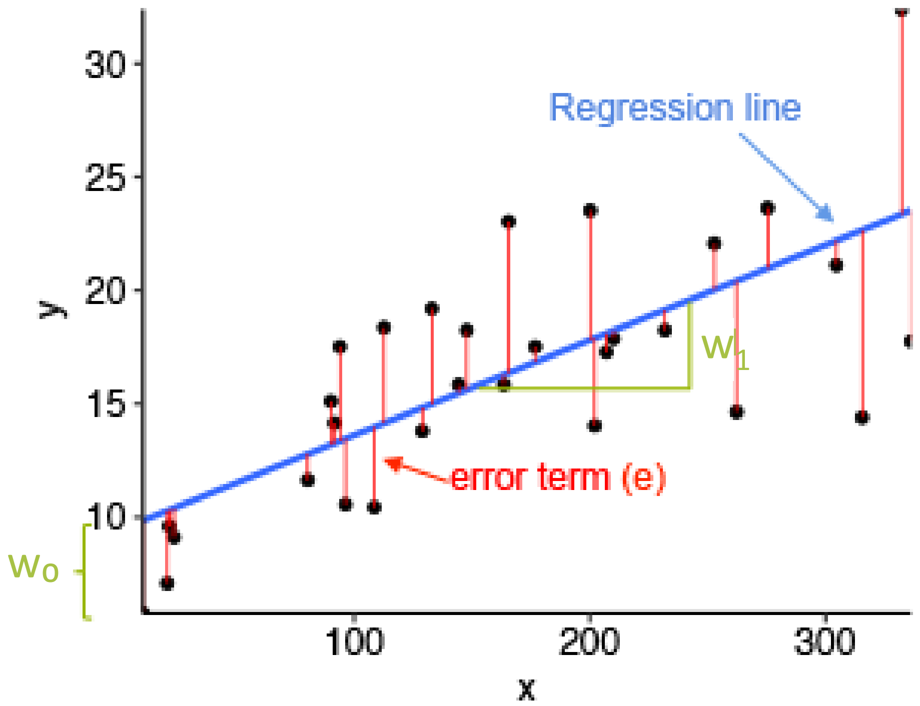
$$y = f(\boldsymbol{x};\boldsymbol{w})$$

线性回归和线性分类

一元线性回归

- **一元线性回归模型的基本形式是：**

$$y = w_0 + w_1 x + \varepsilon$$

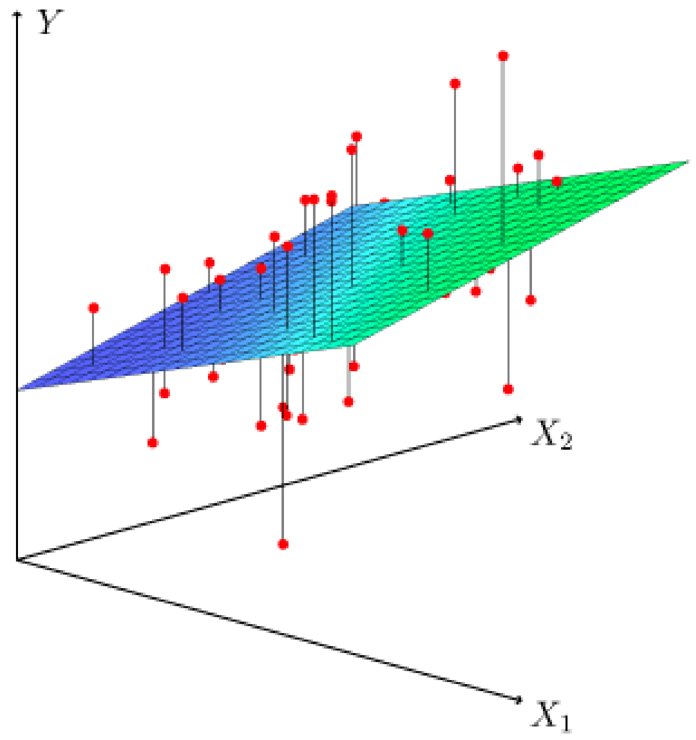


其中：
y是因变量，也就是我们想要预测的变量。
x是自变量，也就是用来预测因变量的变量。
w_0是截距项，表示当自变量x为0时因变量y的预期值。
w_1是斜率系数，表示自变量x每变化一个单位，因变量y预期会变化的量。
ε是误差项，表示模型未能解释的随机变异。

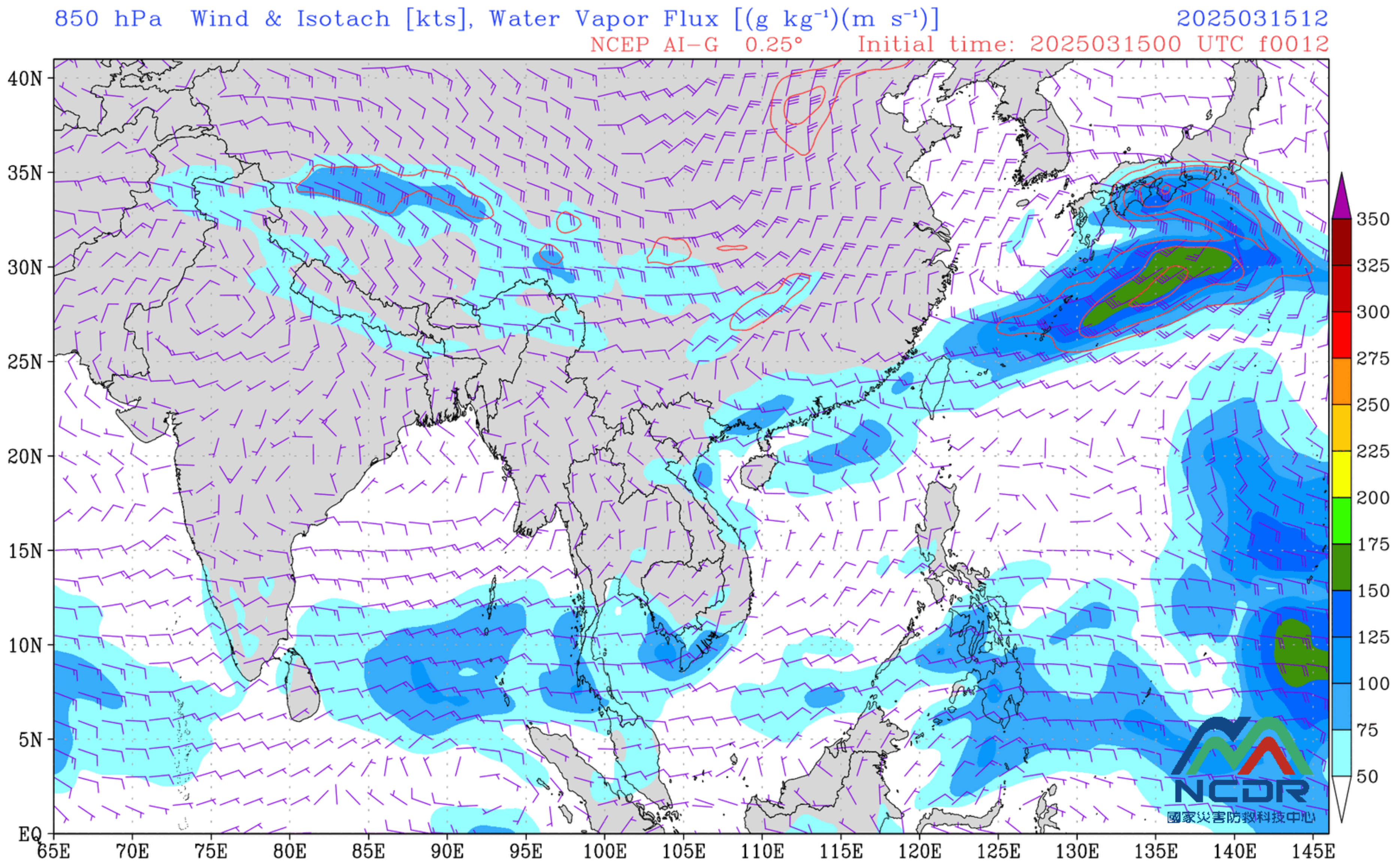
多元线性回归

- 多元线性回归模型的一般形式是

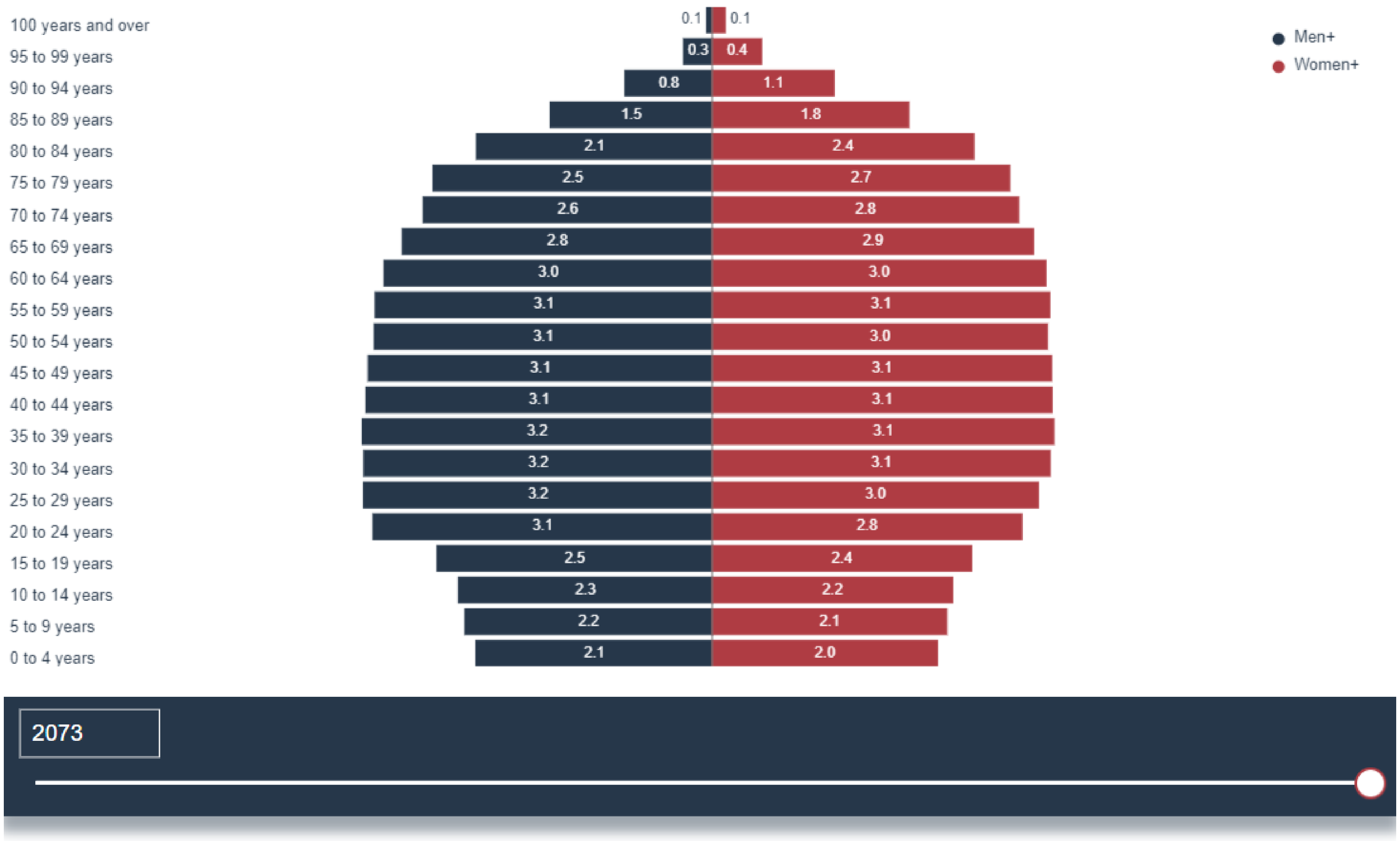
- $y = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n + \varepsilon$



气候研究


850 hPa Wind & Isotach [kts], Water Vapor Flux [(g kg⁻¹)(m s⁻¹)]
2025031512
NCEP AI-G 0.25°
Initial time: 2025031500 UTC f0012
40N
35N
30N
25N
20N
15N
10N
5N
EQ
65E
70E
75E
80E
85E
90E
95E
100E
105E
110E
115E
120E
125E
130E
135E
140E
145E
350
325
300
275
250
225
200
175
150
125
100
75
50
NCDR
國家災害防救科技中心

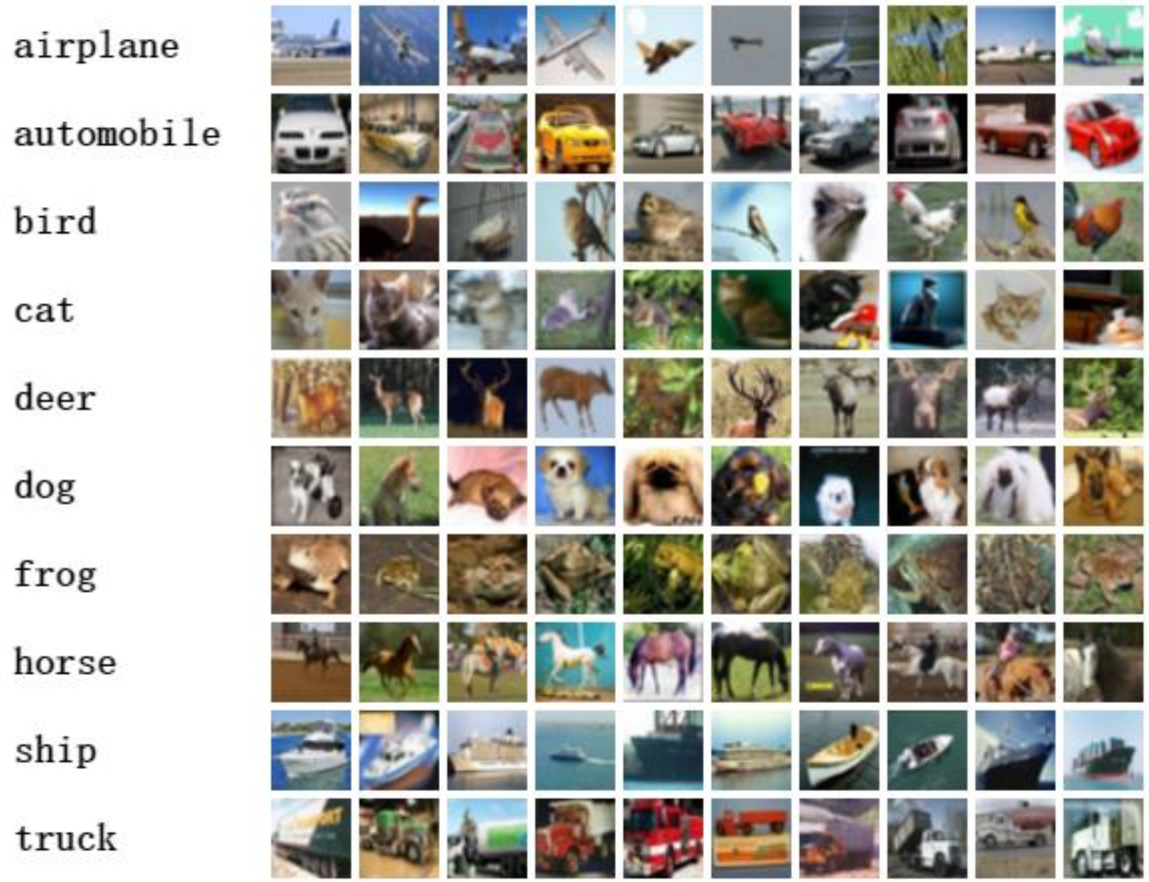
人口增长预测



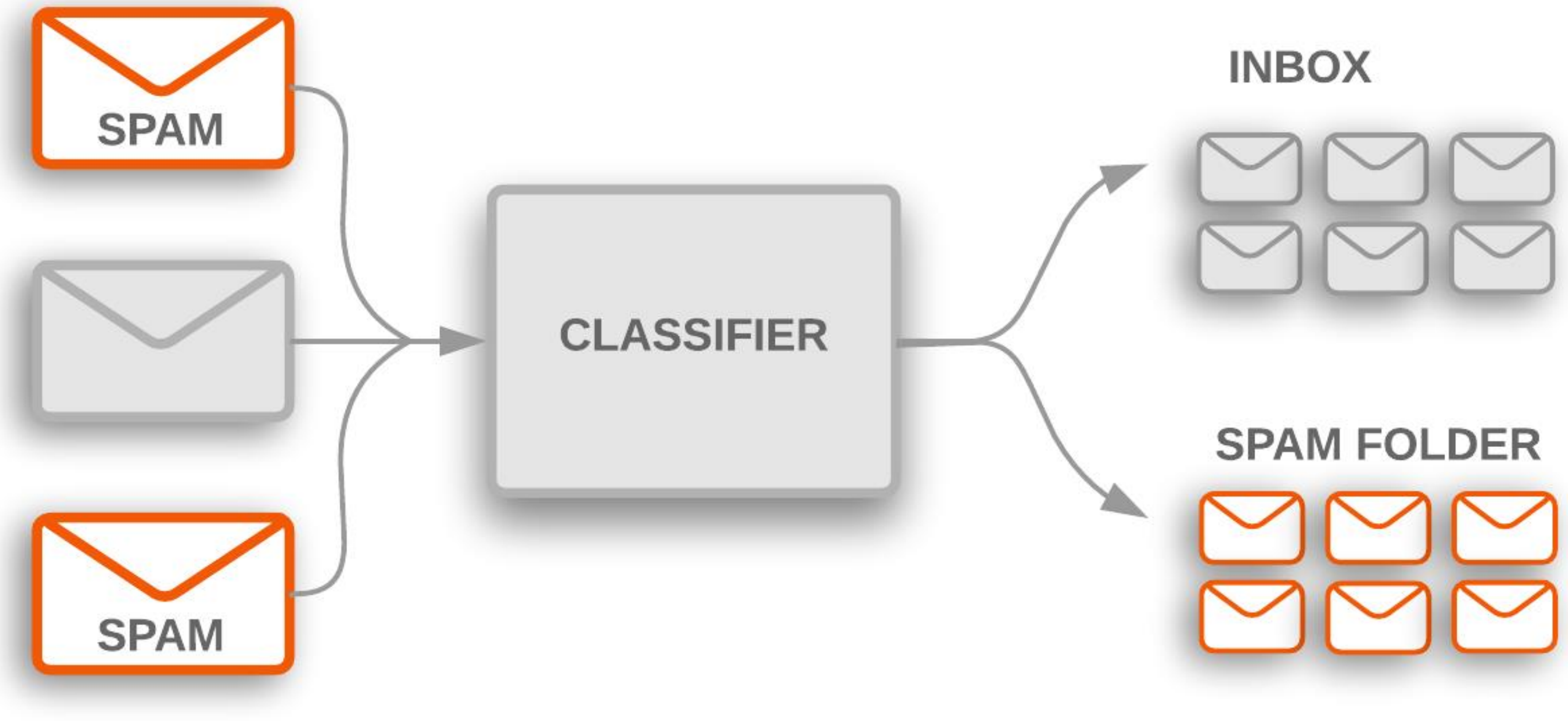
100 years and over
95 to 99 years
90 to 94 years
85 to 89 years
80 to 84 years
75 to 79 years
70 to 74 years
65 to 69 years
60 to 64 years
55 to 59 years
50 to 54 years
45 to 49 years
40 to 44 years
35 to 39 years
30 to 34 years
25 to 29 years
20 to 24 years
15 to 19 years
10 to 14 years
5 to 9 years
0 to 4 years
0.1 0.1
0.3 0.4
0.8 1.1
1.5 1.8
2.1 2.4
2.5 2.7
2.6 2.8
2.8 2.9
3.0 3.0
3.1 3.1
3.1 3.0
3.1 3.1
3.1 3.1
3.2 3.1
3.2 3.1
3.2 3.0
3.1 2.8
2.5 2.4
2.3 2.2
2.2 2.1
2.1 2.0
Men+
Women+
2073

示例：图像分类

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



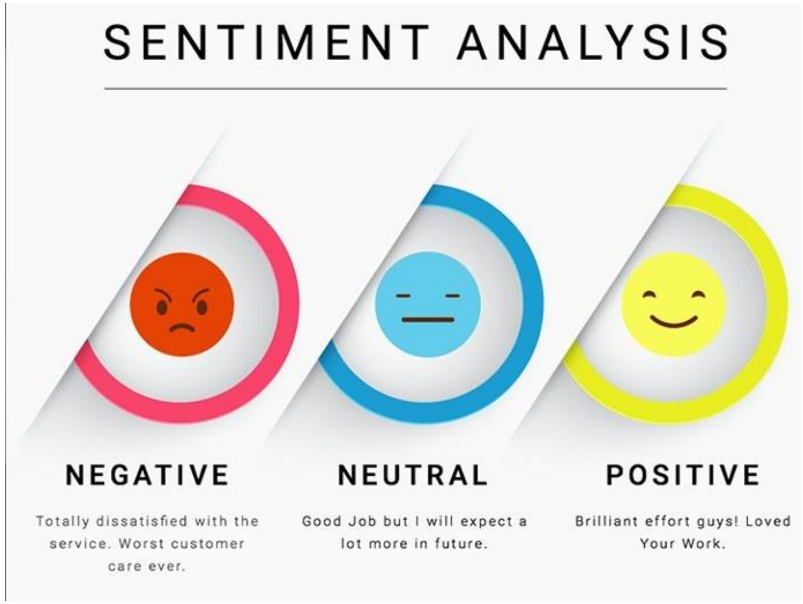
示例：垃圾邮件过滤



示例：文档归类

https://towardsdatascience.com/automated-text-classification-using-machine-learning-3df4f4f9570b

示例：情感分类

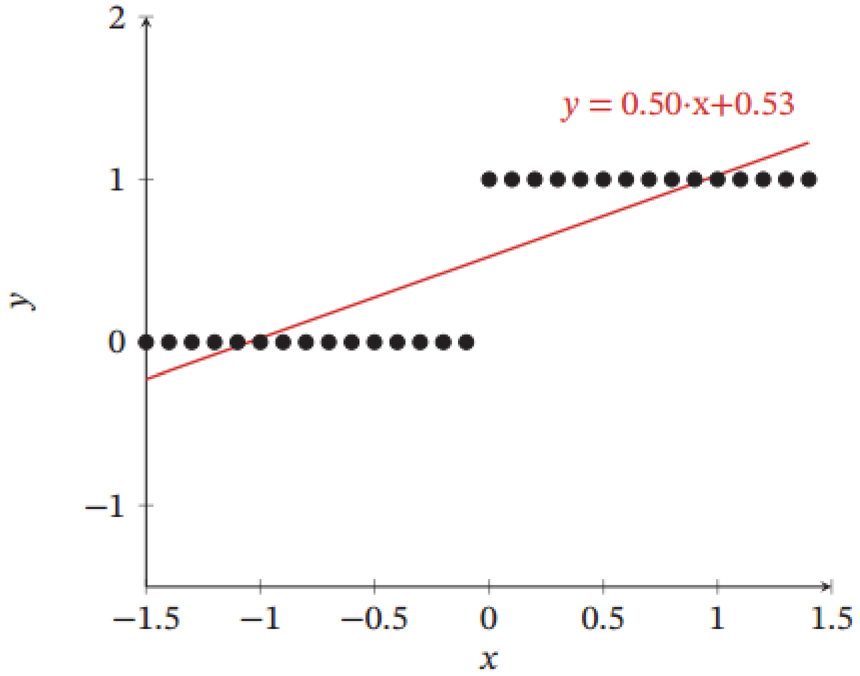


Review (X)	Rating (Y)
"This movie is fantastic! I really like it because it is so good!"	★★★★☆
"Not to my taste, will skip and watch another movie"	★★☆☆☆
"This movie really sucks! Can I get my money back please?"	★☆☆☆☆

线性决策函数

- 线性的决策函数预测分类问题的输出效果差，因为输出目标y是一些离散的标签

$$y = f(x; w)$$



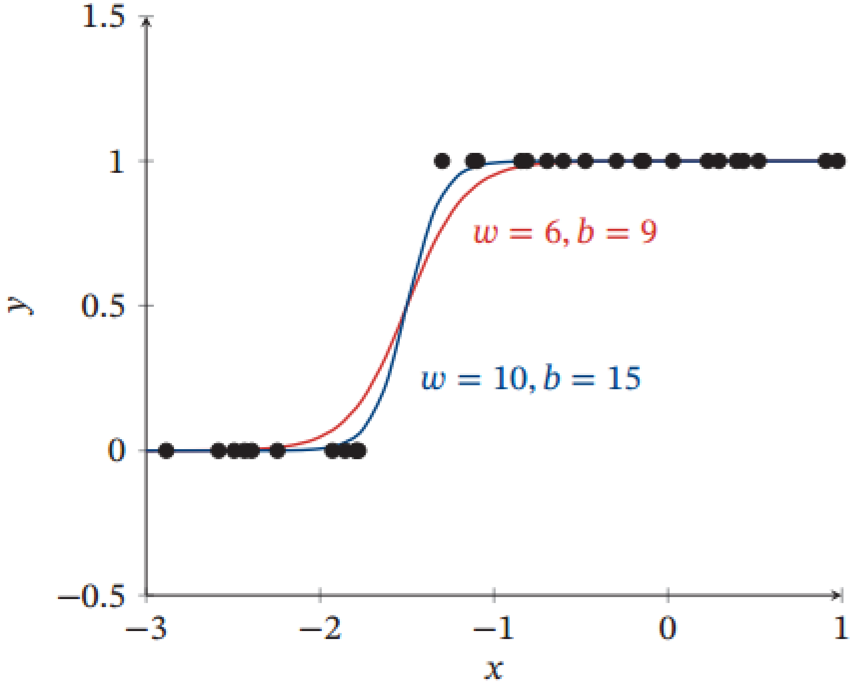
(a) 线性回归

非线性决策函数

- 引入一个非线性的决策函数g来更好的进行拟合

$$y = g(f(x; w))$$

- 把线性函数的值域从实数区间“挤压”到了(0,1)之间，可以用来表示概率。



线性模型+非线性决策函数

- 感知器
- Logistic回归
- Softmax回归
- 支持向量机

解决分类问题

感知器

二分类

- 二分类问题的类别标签只有两种取值
- 通常可以设为 $\{+1, -1\}$ 或 $\{0, 1\}$.
- 决策函数

$$g(f(\boldsymbol{x}; \boldsymbol{w})) = \begin{cases} 1 & \text{if} \quad f(\boldsymbol{x}; \boldsymbol{w}) > 0 \\ 0 & \text{if} \quad f(\boldsymbol{x}; \boldsymbol{w}) < 0 \end{cases}$$

- 或

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

二分类模型的基本框架

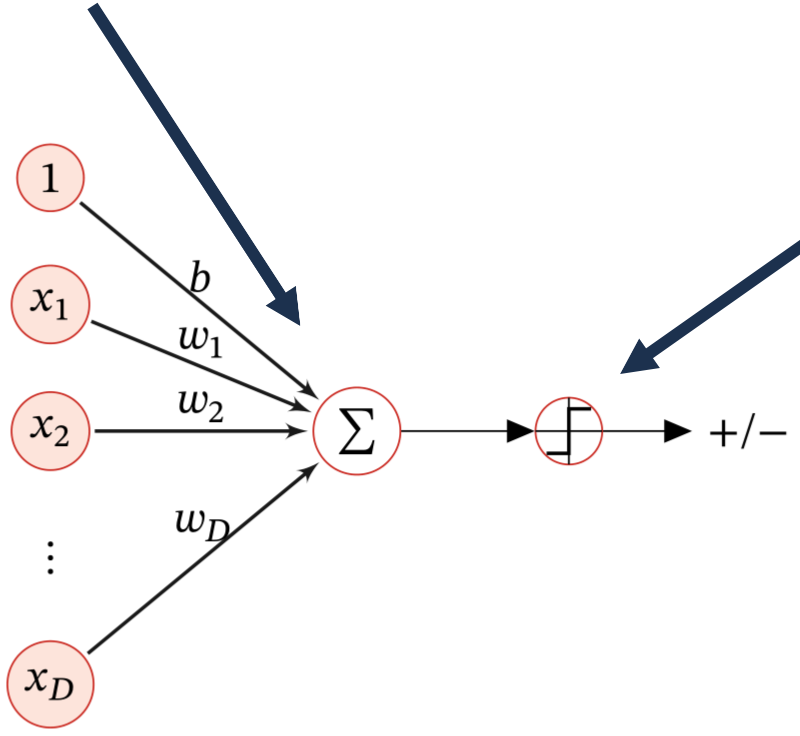
- 线性判别函数 + 二分类决策函数

回归函数：

$$f(\boldsymbol{x};\boldsymbol{w}) = \boldsymbol{w}^\top \boldsymbol{x} + b$$

符号函数：

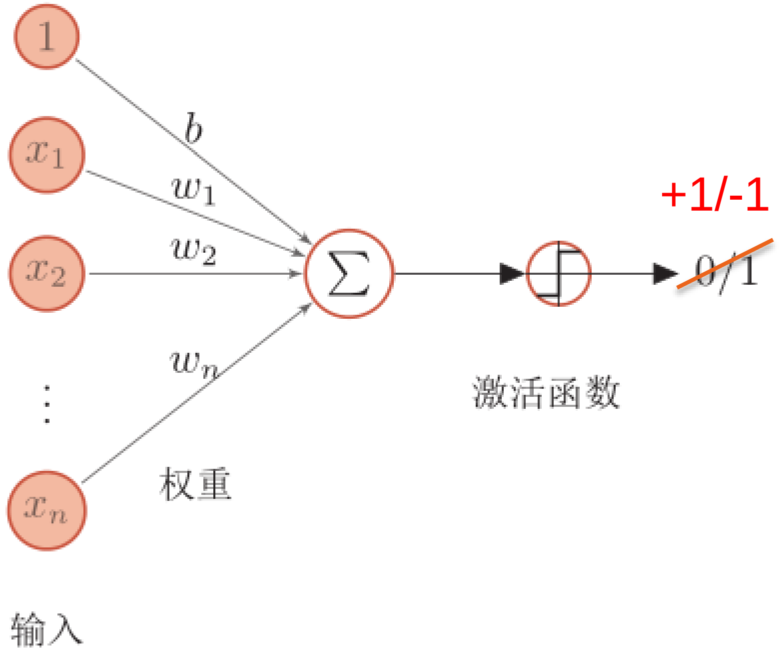
$$g(f(\boldsymbol{x};\boldsymbol{w})) = sgn(f(\boldsymbol{x};\boldsymbol{w}))$$



感知器

- 由 Frank Roseblatt 于 1957 年提出， 是一种广泛使用的线性分类器． 感知器可谓是最简单的人工神经网络，只有一个神经元
- 模拟生物神经元行为的机器，有与生物神经元相对应的部件，如权重（突触）、偏置（阈值）及激活函数（细胞体），输出为+1或-1。

$$\hat{y} = \mathrm{sgn}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) = \begin{cases} +1 & 当\mathbf{w}^{\mathrm{T}}\mathbf{x} > 0 \\ -1 & 当\mathbf{w}^{\mathrm{T}}\mathbf{x} \le 0 \end{cases},$$



感知器

- 学习算法
  - 感知器的学习算法是一种错误驱动的在线学习算法:
  - 先初始化一个权重向量$\boldsymbol{w} \leftarrow 0$（通常是全零向量）；
  - 每次分错一个样本$(\boldsymbol{x}, y)$时，即

$$y\boldsymbol{w}^T\boldsymbol{x} < \boldsymbol{0}$$

  - 用这个样本来更新权重

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + y\boldsymbol{x}$$

  - 根据感知器的学习策略，可以反推出感知器的损失函数为

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

感知器

- 采用随机梯度下降，其每次更新的梯度为

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

感知器的学习过程

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

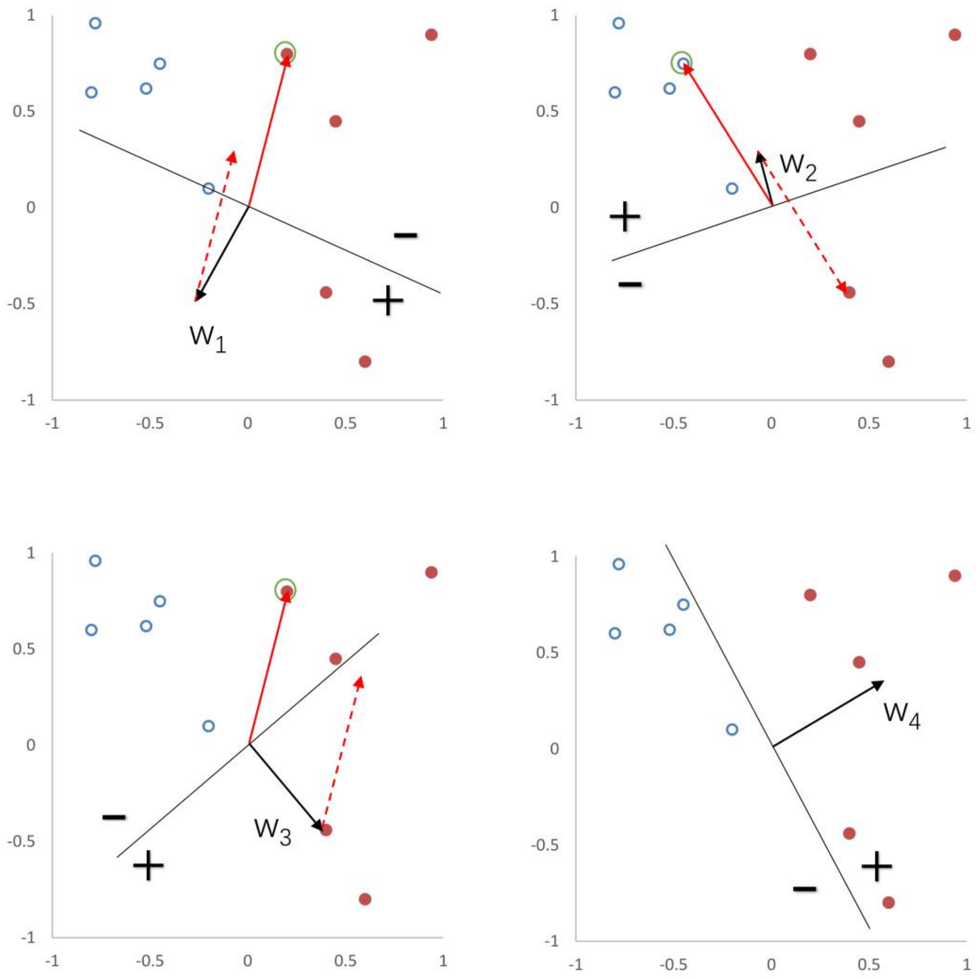
初始化：$\boldsymbol{w}_0 \leftarrow 0, k \leftarrow 0, t \leftarrow 0$；
repeat
　对训练集 $\mathcal{D}$ 中的样本随机排序;
　for $n = 1 \cdots N$ **do**
　　选取一个样本 $(\boldsymbol{x}^{(n)}, y^{(n)})$;
　　if $\boldsymbol{w}_k^{\mathrm{T}}(y^{(n)}\boldsymbol{x}^{(n)}) \leq 0$ **then**
　　　$\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k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

表示分错

对比Logistic回归的更新方式:

$$w^{t+1} \leftarrow w^{t} + \alpha \frac{1}{N}\sum_{i=1}^{N} x_i(y_i - \hat{y}_i)$$

感知器参数学习的更新过程

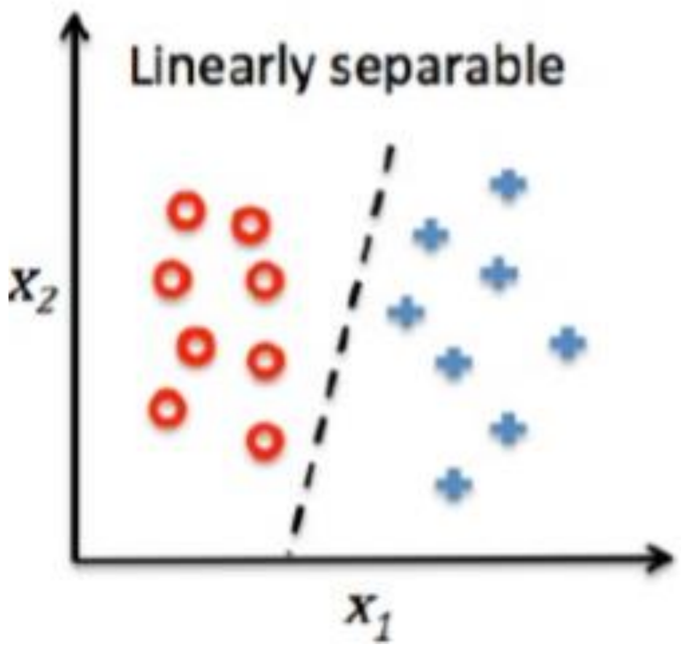


感知器模型的基本框架

- 感知器模型的总体思路就是，先用逻辑函数把线性回归的结果 (-∞,∞) 映射到 (0,1)，再通过决策边界建立与分类的概率联系。

- 存在问题
  - 函数g非连续，无法求导$g\big(f(x;w)\big) = sgn\big(f(x;w)\big)$
  - 学习速率较慢

感知器遇到线性不可分的情况



支持向量机

样本不平衡带来的困难

- 模型偏向多数类
  - 多数类样本主导损失函数，模型更关注拟合多数类，导致少数类识别能力差。例如医疗诊断中，“健康”样本远多于“患病”样本，模型可能直接将所有样本预测为“健康”。
- 评估指标失效
  - 常用的准确率无法真实反映模型对少数类的分类能力。如少数类占比仅 1% 时，即使模型完全忽略少数类，仍可能获得 99% 的“高准确率”。
- 特征学习不充分
  - 少数类样本数量少，模型难以学习到其独特特征，泛化能力下降。

支持向量机

- 支持向量机的分类方法，是在一组分布中找出一个超平面作为决策边界，使模型在数据上的分类误差尽量接近于零
  - SVM 的决策边界由“支持向量”决定，而非全部样本。在样本不平衡场景中，若少数类样本恰好是支持向量（即位于间隔边界或误分类的关键样本），模型会天然重视它们，因为这些样本直接影响超平面的位置和间隔大小。
  - 即使少数类样本数量少，只要成为支持向量，就会主导分类决策。
- 支持向量机在线性和非线性分类中，效果都非常好。

支持向量机

- 给定一个二分类器数据集

$$\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^{N} \qquad y_n \in \{+1, -1\}$$

- 如果两类样本是线性可分的，即存在一个超平面

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

- 将两类样本分开，那么对于每个样本都有

$$y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) > 0$$

- 每个样本 $x^{(n)}$ 到分割超平面的距离为

$$\gamma_n = \frac{|\boldsymbol{w}^\top \boldsymbol{x}_n + b|}{||w||} = \frac{y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b)}{||w||}$$

支持向量机

- 支持向量机的目标是寻找一个超平面$(\boldsymbol{w}^*, b^*)$使得γ最大，即

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

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

- 等价于

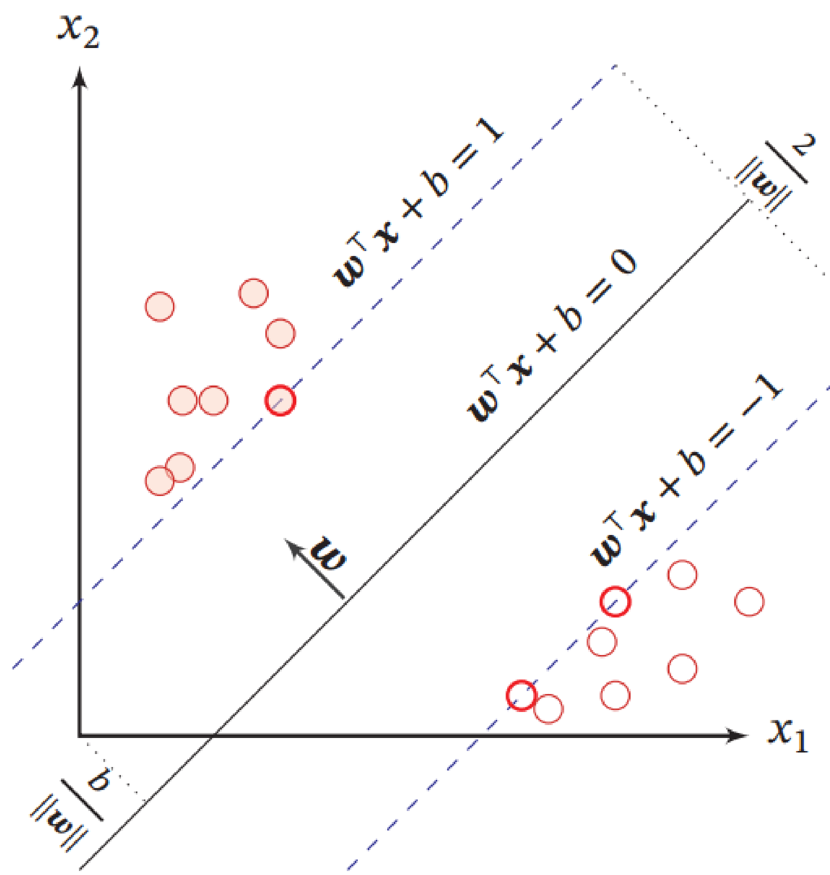
$$\max_{\boldsymbol{w},b} \quad \frac{1}{||\boldsymbol{w}||^2}$$

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

- 数据集中所有满足$y_n(\boldsymbol{w}^\top \boldsymbol{x}_n + b) = 1$的样本点，都称为**支持向量**

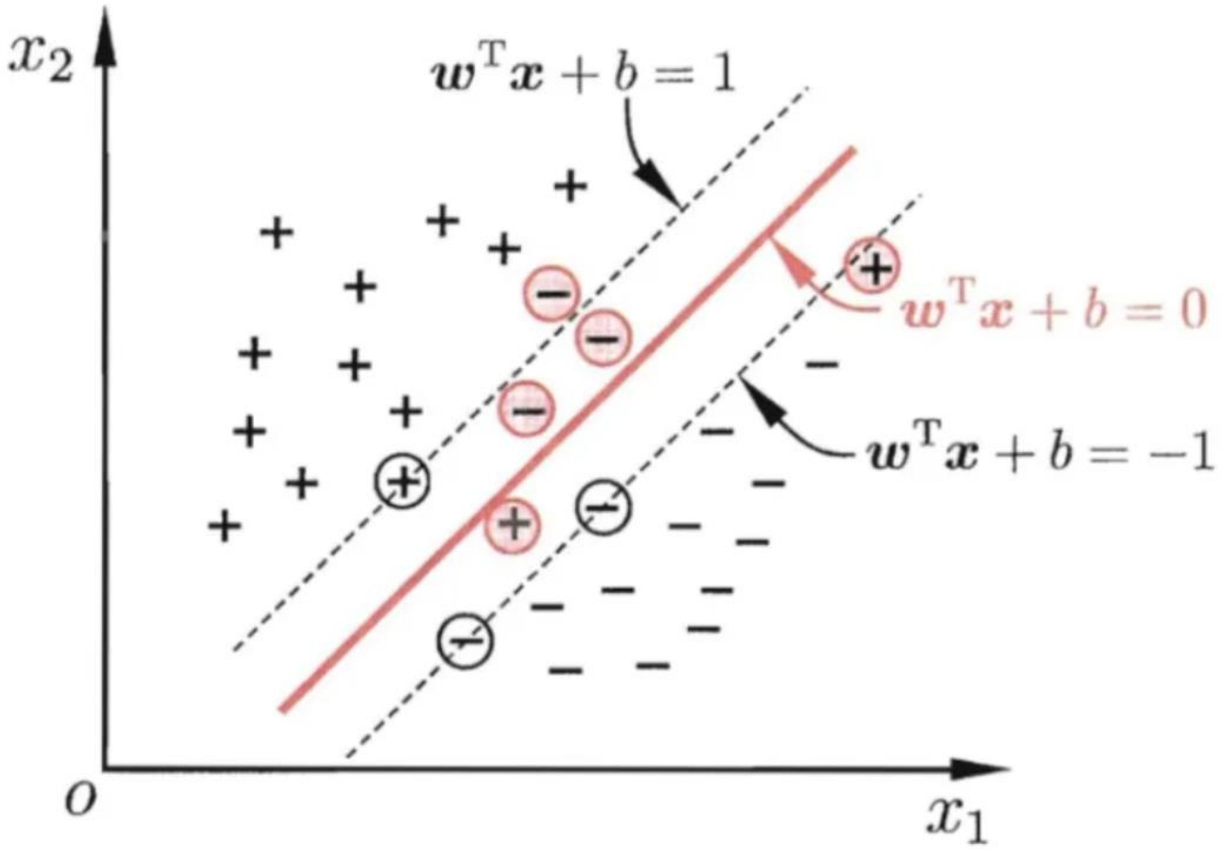
支持向量机

- 其分割超平面有很多个，但是间隔最大的超平面是唯一的



支持向量机

- 在最大化间隔的同时，不满足约束的样本应尽可能地少

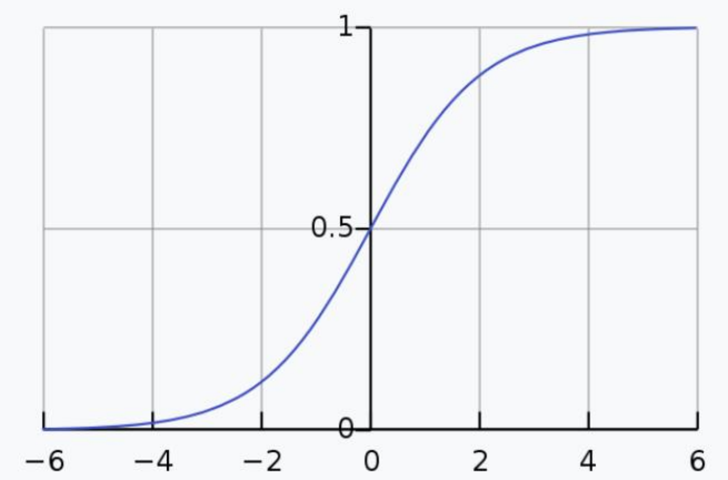


Logistic 回归

Logistic函数与回归

- **Logistic函数**

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$



- **优点**
  - 将任意实数映射到 [0,1] 范围内
  - 任意阶可导
  - 在0 附近几乎是线性的，但在两端趋于平缓
  - 将异常值压向 0 或 1

其他常用的激活函数

Hyperbolic tangent

$$f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Arctangent function

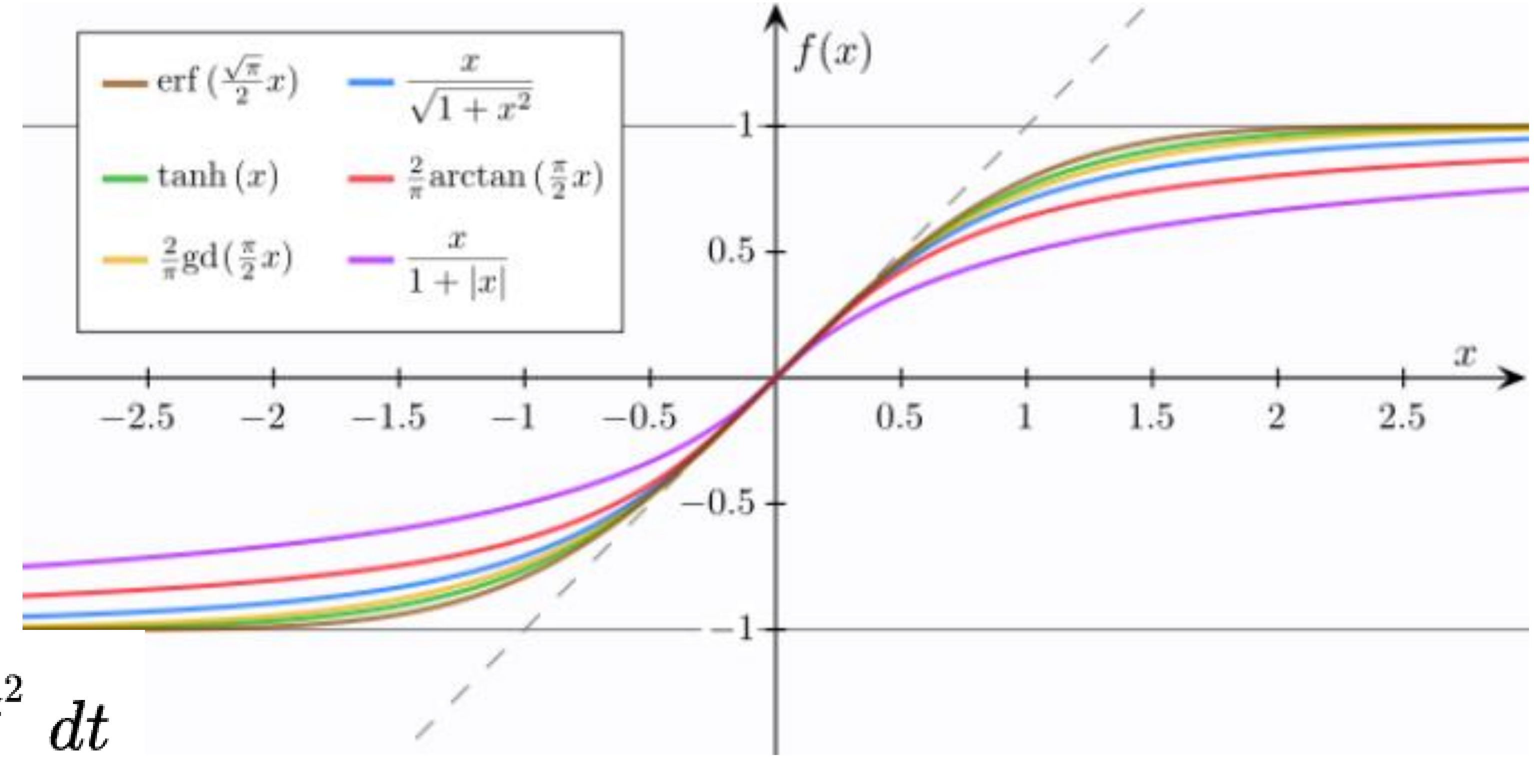
$$f(x) = \arctan x$$

Gudermannian function

$$f(x) = \mathrm{gd}(x) = \int_0^x \frac{dt}{\cosh t}$$

Error function

$$f(x) = \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$$



Logistic函数与回归

- 对于一个二元逻辑回归，我们只需要确保

$$P(y=1)+P(y=0)=1$$

$$P(y=1)=\sigma(x)=\frac{1}{1+\exp(-x)}$$

- 根据 Logistic 函数的性质

$$1-\sigma(x)=\sigma(-x)$$

- 有：

$$P(y=0)=1-\sigma(x)=1-\frac{1}{1+\exp(-x)}=\frac{\exp(-x)}{1+\exp(-x)}$$

推导过程

- 将分类问题看作条件概率估计问题

  - 基于已有数据以及学习权重类别标签的条件概率 $p(y=c|x)$ 最大化。
  - 以二分类为例(0, 1),

$$p(y = 1|x) = g(f(x;\ w))$$

  - 标签 $y = 1$的后验概率为

$$p(y=1|x)=\sigma(w^{\top}x):=\frac{1}{1+\exp(-w^{\top}x)}$$

推导过程

- 标签 $y = 0$ 的后验概率为

$$p(y=0|x) = 1 - p(y=1|x) = \frac{\exp(-w^{\top}x)}{1+\exp(-w^{\top}x)}$$

- 在二分类模型中，事件发生与不发生的概率之比$p/(1-p)$ 称为事件的几率. 我们有

$$w^{\top}x = \log\frac{p(y=1|x)}{1-p(y=1|x)}$$

风险函数

- 我们使用极大似然估计法来求解. 我们把单个样本看做一个事件，那么这个事件发生的概率就是:

$$P(y|x) = \begin{cases} p, y = 1 \\ 1 - p, y = 0 \end{cases}$$

- 它等价于:

$$P(y|x) = p^{y}(1-p)^{1-y}$$

- 对于每个样本: $P(y_i|x_i) = p^{y_i}(1-p)^{1-y_i}$
- 则组样本的概率:

$$P = P(y_1|x_1)P(y_2|x_2)P(y_3|x_3)\cdots P(y_m|x_m) = \prod_{i=1}^{m} p^{y_i}(1-p)^{1-y_i}$$

风险函数

- 组样本的概率：

$P = P(y_1|x_1)P(y_2|x_2)P(y_3|x_3)\cdots P(y_m|x_m) = \prod_{i=1}^{m} p^{y_i}(1-p)^{1-y_i}$；

- 则似然函数为

$$L(w) = \prod_{i=1}^{m} P(y_i|x_i) = \prod_{i=1}^{m} \left(\sigma(w^\top x_i)\right)^{y_i} \left(1-\sigma(w^\top x_i)\right)^{1-y_i}$$

梯度下降

- 等式两边同时取对数：

$$L(w) = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log\left(\sigma(w^\top x_i)\right) + (1-y_i)\log\left(1-\sigma(w^\top x_i)\right)\right).$$

- 交叉熵损失函数，模型在训练集的风险函数为

$$l(w) = \log L(w) = \sum_{i=1}^{m}\left(y_i \log\left(\sigma(w^\top x_i)\right) + (1-y_i)\log\left(1-\sigma(w^\top x_i)\right)\right)$$

$$J(w) = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log \hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\right)$$

$$\hat{y}_i = \sigma(w^\top x_i)$$

推导过程

- 梯度为：

$$\frac{\partial J(w)}{\partial w} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i\frac{\hat{y}_i}{\hat{y}_i}(1-\hat{y}_i)x_i-(1-y_i)\frac{\hat{y}_i(1-\hat{y}_i)}{1-\hat{y}_i}x_i\right)$$

$$= -\frac{1}{N}\sum_{i=1}^{N}(y_i(1-\hat{y}_i)x_i-(1-y_i)\hat{y}_i x_i)$$

$$= -\frac{1}{N}\sum_{i=1}^{N}x_i(y_i-\hat{y}_i).$$

$$\sigma'(z) = \left(\frac{1}{1+e^{-z}}\right)' = \frac{e^{-z}}{(1+e^{-z})^2} = \frac{1+e^{-z}-1}{(1+e^{-z})^2} = \frac{1}{(1+e^{-z})}\left(1-\frac{1}{(1+e^{-z})}\right) = \sigma(z)(1-\sigma(z))$$

$$\frac{\partial J(w)}{\partial w} = \frac{1}{N}\sum_{i=1}^{N}\left(x_i\cdot\left(\sigma(w^\top x_i)-y_i\right)\right)$$

推导过程

- 采用梯度下降法，Logistic 回归的训练过程为:
  - 初始化 $\mathrm{w}^0 \leftarrow 0$，然后通过下式来迭代更新参数:

$$w^{t+1} \leftarrow w^t + \alpha \frac{1}{N} \sum_{i=1}^{N} x_i (y_i - \hat{y}_i)$$

多分类与 Softmax 回归

多分类（Multi-class Classification）

- 分类的类别数C大于2，一般需要多个线性判别函数
  - “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

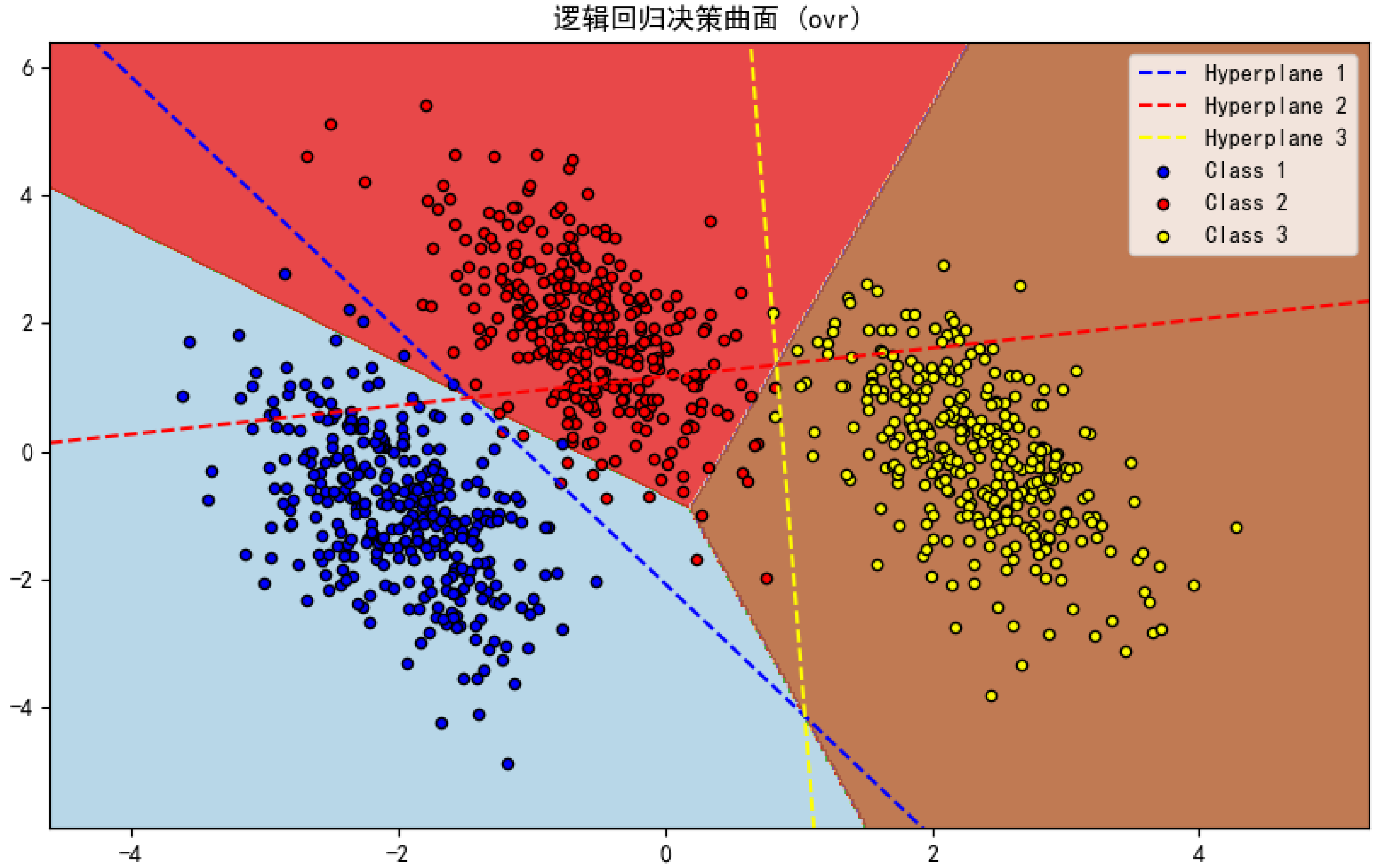
$$y = \arg\max_{c=1}^{C} f_c(\boldsymbol{x};\boldsymbol{w}_c)$$

多分类（Multi-class Classification）

- 相邻两类i和j的决策边界实际上是由

$$f_i(x; w_i) - f_j(x; w_j) = 0$$

- 决定.



Softmax回归

- Softmax函数

$$\mathrm{softmax}(z_c) = \frac{\exp(z_c)}{\sum_{i=1}^{N} \exp(z_i)}$$

- 给定一个样本x，Softmax回归预测的属于类别c的条件概率为

$$\begin{aligned} p(y = c|\boldsymbol{x}) &= \mathrm{softmax}(\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x}) \\ &= \frac{\exp(\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{x})}{\sum_{c'=1}^{C} \exp(\boldsymbol{w}_{c'}^{\mathrm{T}}\boldsymbol{x})}, \end{aligned}$$

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

Softmax回归

- 多分类

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

- Softmax回归的决策函数可以表示为

$$\begin{aligned}\hat{y} &= \arg\max_{c=1}^{C} p(y = c|\boldsymbol{x}) \\ &= \arg\max_{c=1}^{C} \frac{\exp(\boldsymbol{w}_c^{\mathsf{T}}\boldsymbol{x})}{\sum_{c'=1}^{C} \exp(\boldsymbol{w}_{c'}^{\mathsf{T}}\boldsymbol{x})}\end{aligned}$$

Softmax回归

- 向量形式可以写为

$$\hat{\boldsymbol{y}} = \mathrm{softmax}(\boldsymbol{W}^{\top}\boldsymbol{x})$$

$$= \frac{\exp(\boldsymbol{W}^{\top}\boldsymbol{x})}{\mathbf{1}_C^{\top}\exp(\boldsymbol{W}^{\top}\boldsymbol{x})},$$

- 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,
- $\hat{\boldsymbol{y}} \in \mathbb{R}^C$ 为所有类别的预测条件概率组成的向量

交叉熵损失

- KL散度

$$\mathrm{D_{kl}}(p_r(y|x)||p_\theta(y|x)) = \sum_{y=1}^{C} p_r(y|x)\log\frac{p_r(y|x)}{p_\theta(y|x)}$$

$$\propto -\sum_{y=1}^{C} p_r(y|x)\log p_\theta(y|x)$$

交叉熵损失

y^*为x的真实标签

$$= -\log p_\theta(y^*|x)$$

负对数似然

参数学习

- 采用交叉熵损失函数，Softmax回归模型的风险函数为

$$J(W) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_i^c \log \hat{y}_i^c$$
$$= -\frac{1}{N}\sum_{n=1}^{N} (\boldsymbol{y}_i)^\top \log \hat{\boldsymbol{y}}_i$$

$$\hat{\boldsymbol{y}}_i = \mathrm{softmax}(W^\top \boldsymbol{x}_i)$$

- 风险函数 $J(W)$关于W的梯度为

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

总结

线性分类模型小结

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