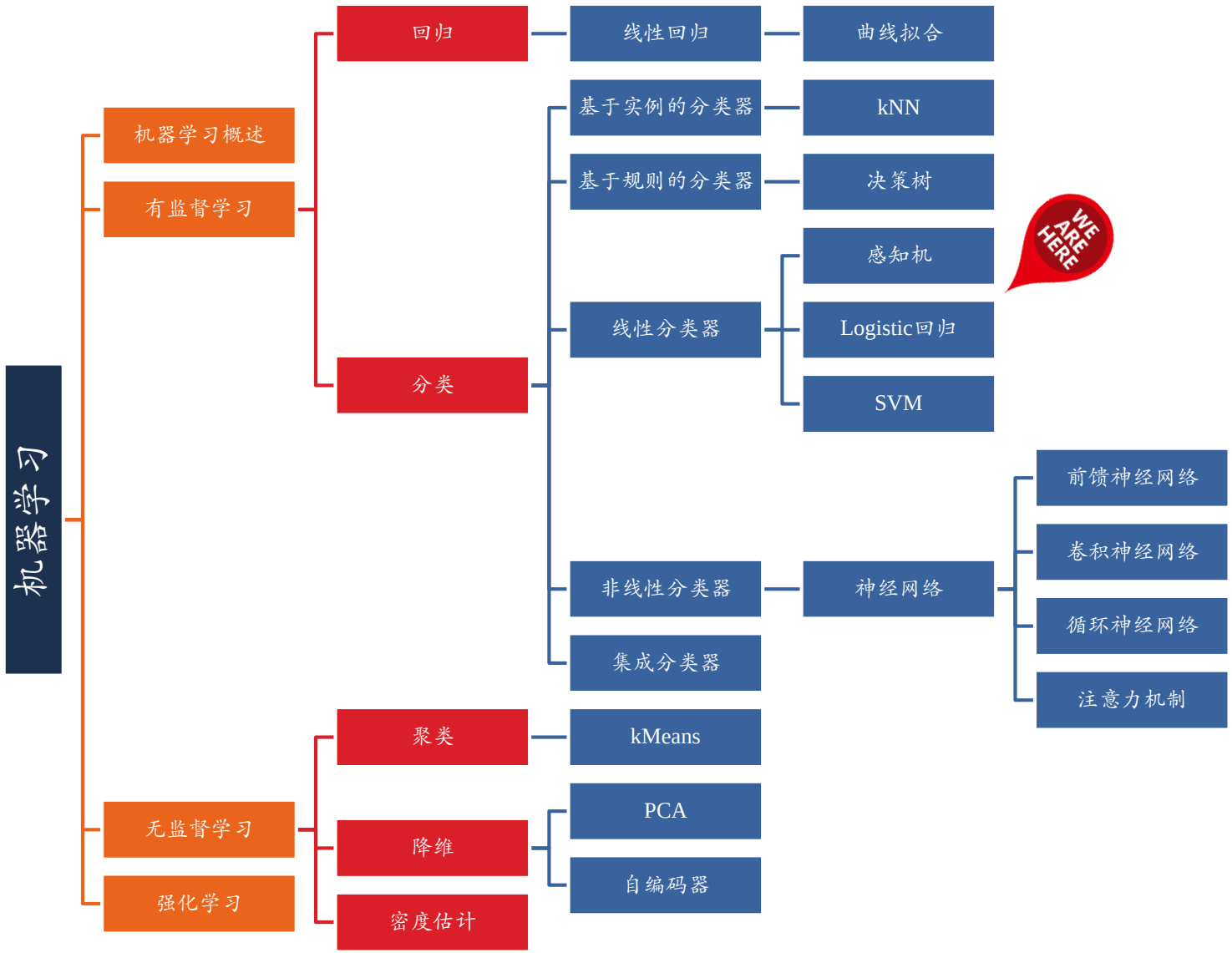
“AI+X微专业”基础类课程

华五教学协同中心
Consortium for the Advancement of Teaching and Learning at E5

《模式识别与机器学习》
第四讲：线性分类

邱锡鹏
复旦大学 计算机学院
xpqiu@fudan.edu.cn

路线图


机器学习
机器学习概述
有监督学习
回归
线性回归
曲线拟合
分类
基于实例的分类器
kNN
基于规则的分类器
决策树
线性分类器
感知机
Logistic回归
SVM
WE ARE HERE
非线性分类器
神经网络
前馈神经网络
卷积神经网络
循环神经网络
注意力机制
集成分类器
无监督学习
聚类
kMeans
降维
PCA
自编码器
密度估计
强化学习

内容

- 判别函数（Discriminant Functions）
- 模型
  - 生成式模型
  - 判别式模型

本讲内容主要来自

- “Pattern Recognition and Machine Learning” by Bishop
  - Ch. 4.1.1-4.1.4, 4.1.7, 4.2.1-4.2.2, 4.3.1-4.3.3

- 《神经网络与深度学习》邱锡鹏
  - 第3章 线性模型

分类
Classification

分类（Classification）

- 给定训练集：$\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$
  - $\mathbf{x}^{(n)} \in \mathbb{R}^{D}$
  - $y^{(n)} \in \mathcal{Y} = \{1, 2, \cdots, C\}$ 为离散变量，C为类别总数

- 分类任务：
  - 学习$\boldsymbol{x}$和y之间的映射函数f，使之能够对于一条输入$\boldsymbol{x}$正确输出其对应的类别y

线性模型

- 使用“线性模型”进行分类

g是非线性函数

$$y = g(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b)$$

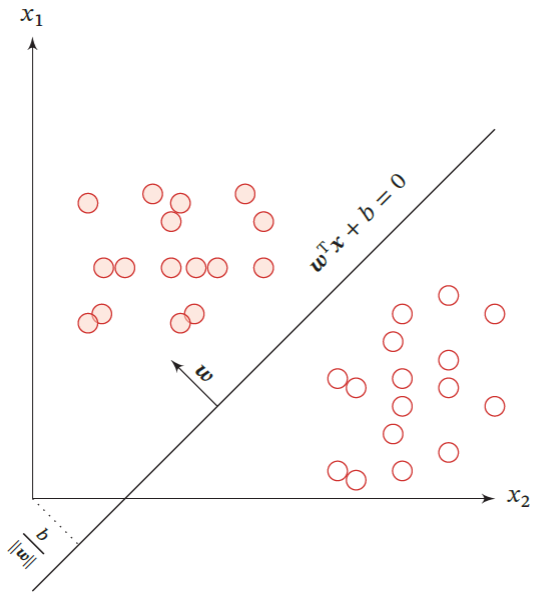
- 上式称为广义线性模型（Generalized Linear Model）

$$y = \mathrm{sgn}(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b)$$

$$\triangleq \begin{cases} +1 & \text{if} \quad \mathbf{w}^{\mathrm{T}}\mathbf{x} + b > 0, \\ -1 & \text{if} \quad \mathbf{w}^{\mathrm{T}}\mathbf{x} + b < 0. \end{cases}$$

- 决策边界（Decision Boundary）
  - 关于x的线性函数

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



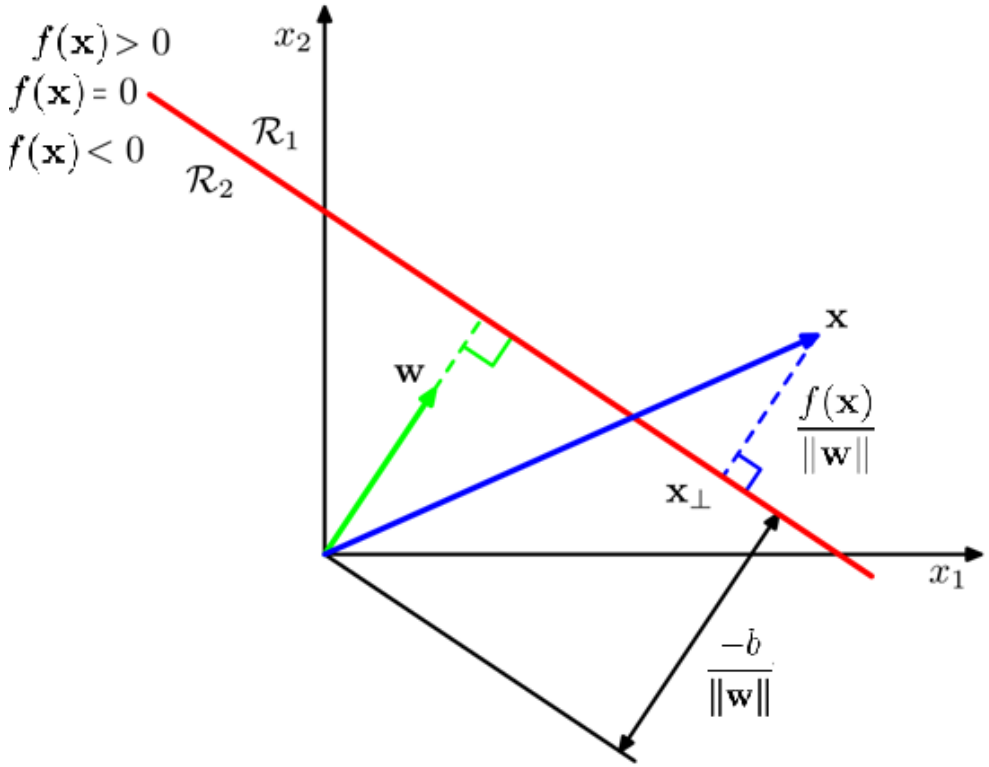
判别函数（Discriminant Functions）

- 线性判别函数（以二分类为例）

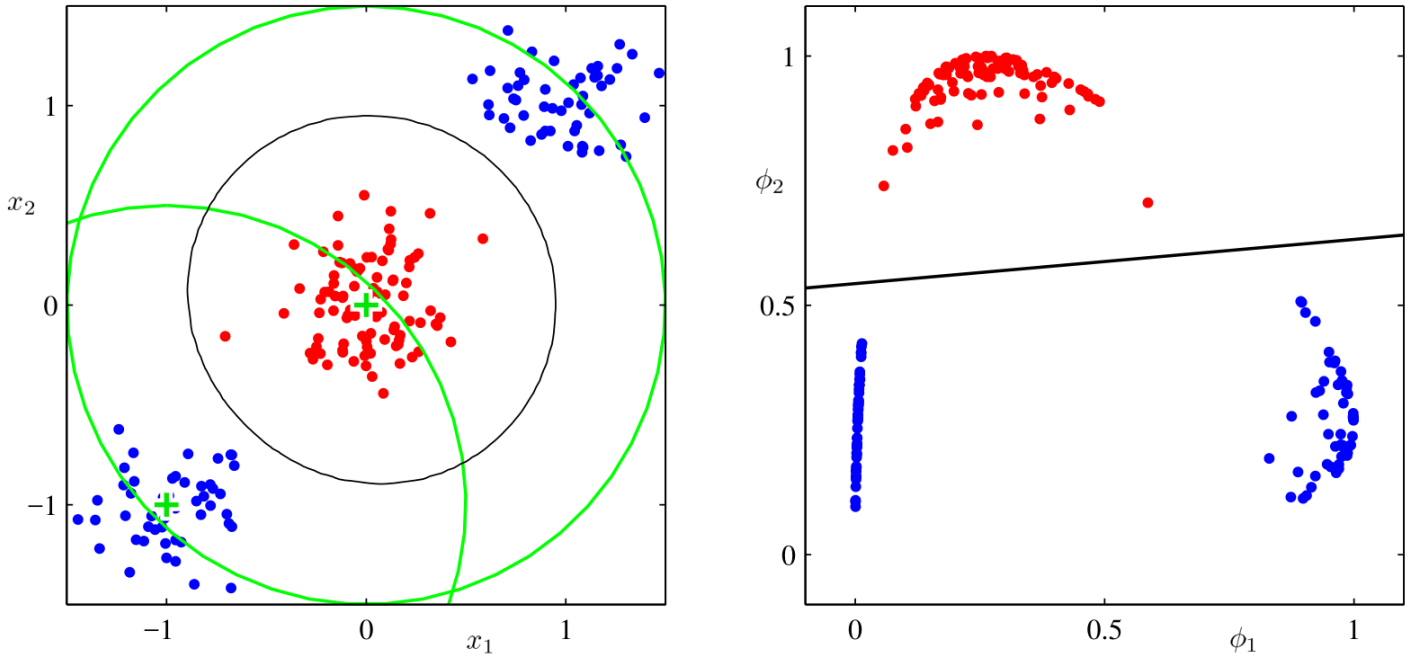
$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$$

  - 通过阈值函数来进行分类

- **x在w方向的投影为** $\frac{f(\mathbf{x})}{\|\mathbf{w}\|}$



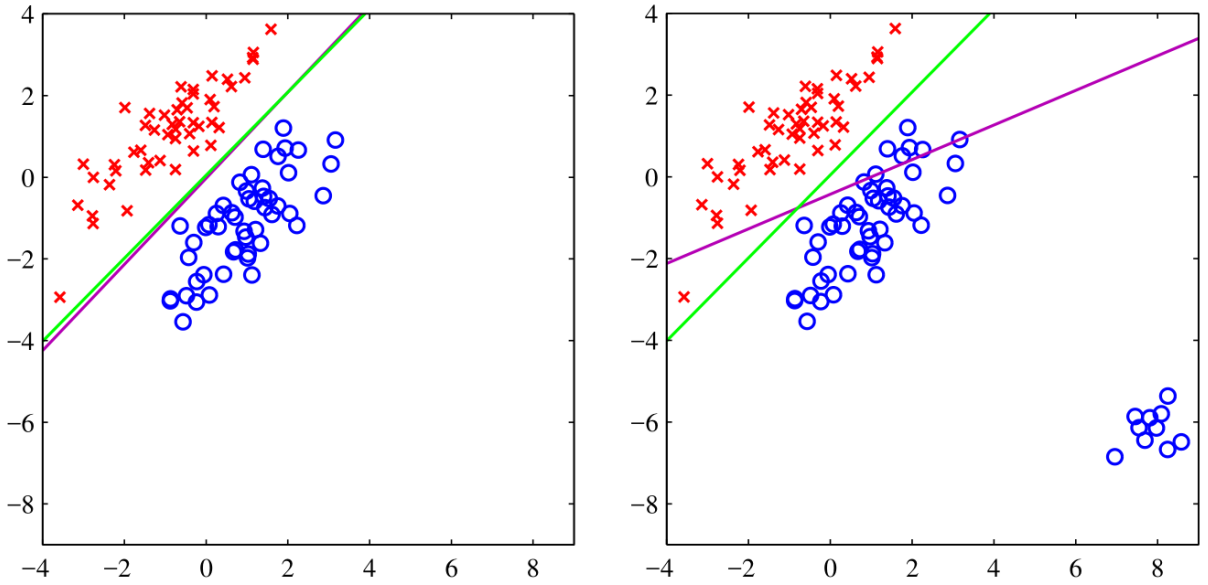
使用“基函数”的广义线性模型



损失函数

- 如何学习决策边界?
- 最小平方误差?

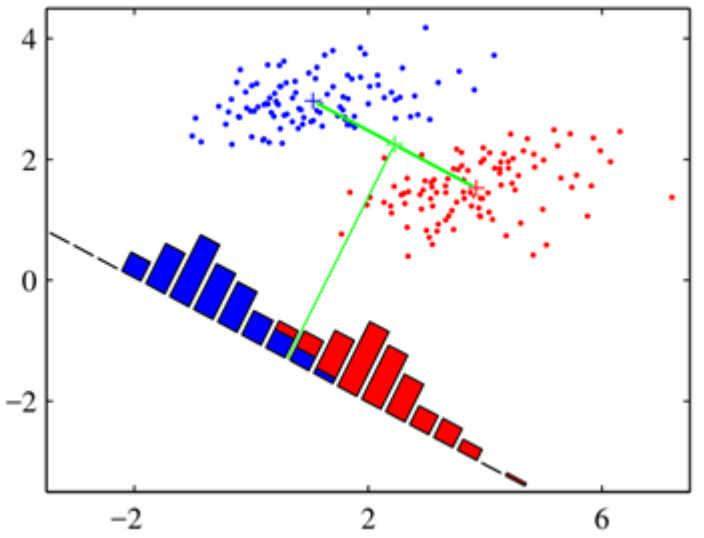
$$\mathcal{R}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y^{(n)} - \mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)}\right)^2$$



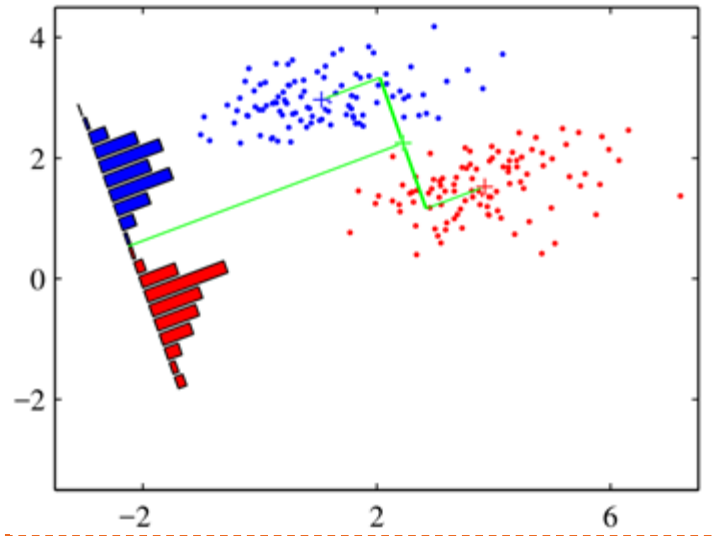
需要寻找更合适的准则

Fisher线性判别（Fisher Linear Discriminant）

- 什么是好的投影方向?
  - 一个自然的方法：类中心的连线方向
  - 方差问题



- Fisher准则：最大化类间距离，最小化类内方差



Fisher线性判别

Fisher准则

$$m_k = \frac{1}{N_k}\sum_{n\in\mathcal{C}_k}\mathbf{w}^{\mathrm{T}}\mathbf{x}^{(n)} = \mathbf{w}^{\mathrm{T}}\mathbf{m}_k$$

类间离散度 Inter-class separation

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

类内离散度 Intra-class variance

$$s_k^2 = \sum_{n\in\mathcal{C}_k}(y^{(n)} - m_k)^2 = \mathbf{w}^{\mathrm{T}}\sum_{n\in\mathcal{C}_k}(\mathbf{x}^{(n)} - \mathbf{m}_k)^2$$

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}S_b\mathbf{w}}{\mathbf{w}^{\mathrm{T}}S_w\mathbf{w}} \xrightarrow{\text{argmax}} \mathbf{w} \propto S_w^{-1}(m_1 - m_1)$$

Fisher线性判别总结

- 当每个类为高斯分布时比较有效
- 对噪声（异常点）比较敏感

- Fisher线性判别是一种降维（dimensionality reduction）方法，用来提取特征
  - 也称为Linear Discriminant Analysis

感知器
Perceptron

感知器

Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN[1]

F. ROSENBLATT

Cornell Aeronautical Laboratory

HAVING told you about the giant digital computer known as I.B.M. 704 and how it has been taught to play a fairly creditable game of chess, we'd like to tell you about an even more remarkable machine, the perceptron, which, as its name implies, is capable of what amounts to original thought. The first perceptron has yet to be built,

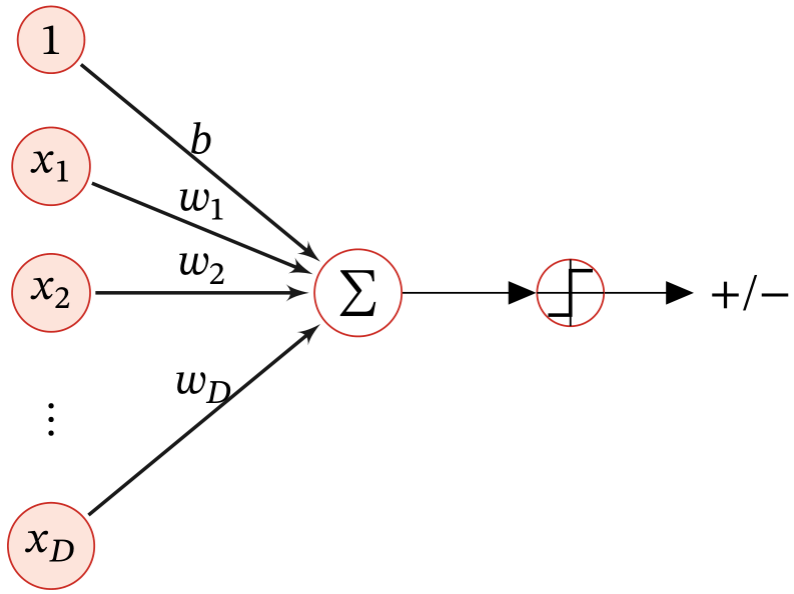
The New Yorker, December 6, 1958 P. 44

The IBM 704 computer

感知器

- 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。

$$\hat{y} = \begin{cases} +1 & \text{当} \mathbf{w}^{\mathrm{T}}\mathbf{x} > 0 \\ -1 & \text{当} \mathbf{w}^{\mathrm{T}}\mathbf{x} \leq 0 \end{cases},$$



感知器

- 学习算法
  - 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

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

  - 用这个样本来更新权重

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

感知器的学习过程

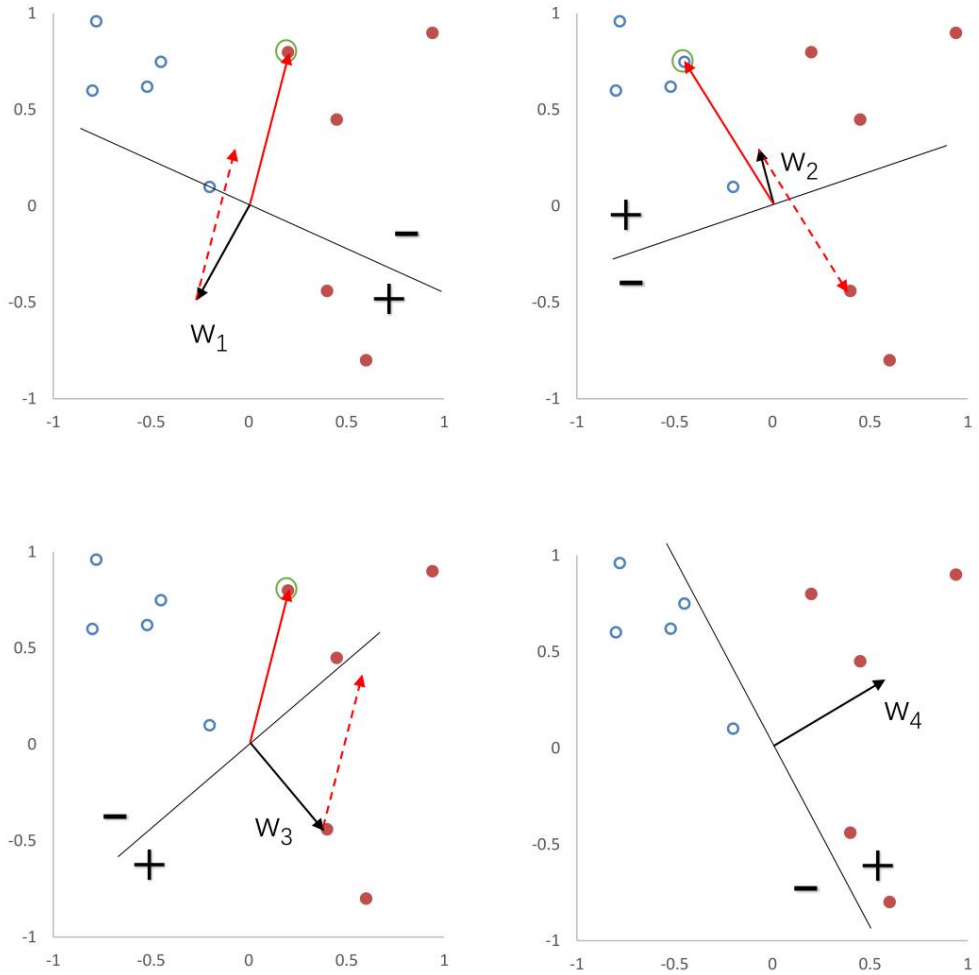
算法 3.1　两类感知器的参数学习算法

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

感知器参数学习的更新过程



感知器总结

- 一种简单有效的线性分类器

- 对于两类问题，如果训练集是线性可分的，那么感知器算法可以在有限次迭代后收敛。
  - 《神经网络与深度学习》 3.4.2节

- 损失函数?
  - 根据感知器的学习策略，可以反推出感知器的损失函数为

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

概率模型

一种概率方法

- 以二分类为例

$$p(\omega_1|\mathbf{x}) = \frac{p(\mathbf{x},\omega_1)}{p(\mathbf{x})}$$

$$= \frac{p(\mathbf{x}|\omega_1)p(\omega_1)}{p(\mathbf{x}|\omega_1)p(\omega_1) + p(\mathbf{x}|\omega_2)p(\omega_2)}$$

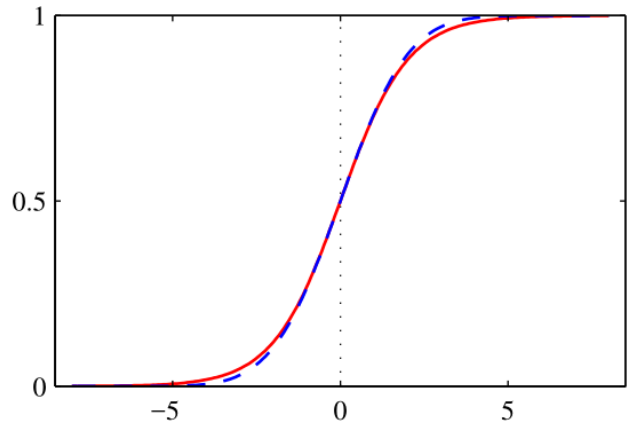
- 分类问题转换为概率密度估计（Probabilistic Density Estimation）问题
  - 概率生成模型（Probabilistic Generative Models）
    - 可以根据$p(x|\omega_c)$生成每个类的样本

概率生成模型

$$p(\omega_1|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_1)p(\omega_1)}{p(\mathbf{x}|\omega_1)p(\omega_1) + p(\mathbf{x}|\omega_2)p(\omega_2)}$$

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

$$a = \log \frac{p(\mathbf{x}|\omega_1)p(\omega_1)}{p(\mathbf{x}|\omega_2)p(\omega_2)}$$



Logistic Sigmoid

当假设类条件密度为高斯分布

- 假设类条件密度为高斯分布，并有相同的方差

$$\mathcal{N}(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu_k)^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\mu_k)\right)$$

- μ_k, Σ为高斯分布的均值和方差，D为特征维数。

$$a = \log \frac{p(\mathbf{x}|\omega_1)p(\omega_1)}{p(\mathbf{x}|\omega_2)p(\omega_2)}$$

- a的形式为 $a = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$
  - 二次项 $\mathbf{x}^{\mathrm{T}}\Sigma^{-1}\mathbf{x}$ 相互抵消

广义线性模型 $p(\mathcal{C}_1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b)$

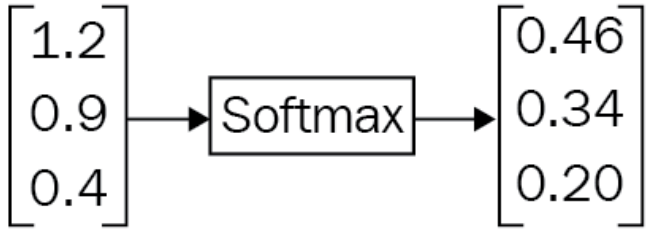
扩展到多分类

- 当类别数>2

$$p(\omega_k|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_k)p(\omega_k)}{\sum_k p(\mathbf{x}|\omega_k)p(\omega_k)} \qquad a = \log p(\mathbf{x}|\omega_k)p(\omega_k)$$

$$= \frac{\exp(a_k)}{\sum_{k'} \exp(a_{k'})} \triangleq \text{softmax}(a_k)$$

- Softmax函数
  - 当 $a_k \gg a_j \Rightarrow p(\omega_k|\mathbf{x}) \to 1$



生成模型 vs. 判别模型

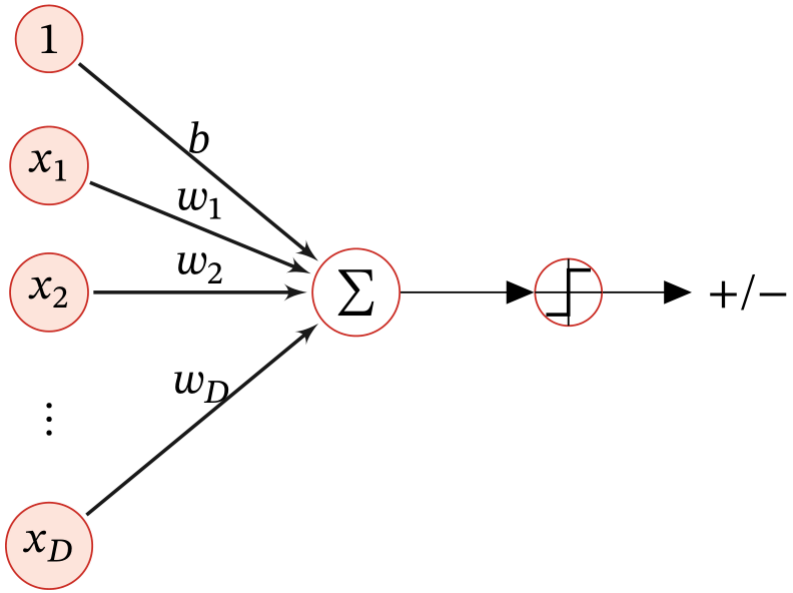
- 生成模型
  - 可以生成样本
  - 生成能力和分类能力等价
  - 参数通常会更多
  - 需要好的类条件密度建模
- 判别模型
  - 只能用作分类
  - 不将简单问题转换为复杂问题
    - 分类往往比密度估计更容易
  - 参数通常会更少
  - 需要好的决策边界

Logistic回归

Logistic回归

- Logistic Regression
- 模型

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



损失函数?

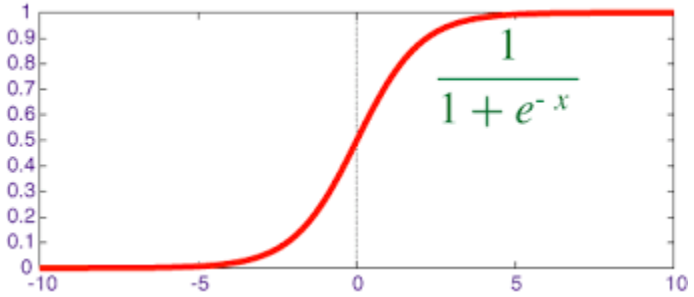
函数f：线性函数
函数g：把线性函数的值域从实数区间“挤压”到了(0,1)之间，可以用来表示概率。

Logistic函数与回归

Logistic函数

除了logisitc，还有什么函数$f: R \to (0,1)$?

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



Logistic回归

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

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

学习准则

- 模型预测条件概率$p_\theta(y|\boldsymbol{x})$

$$p_\theta(y=1|\boldsymbol{x})=\sigma(\boldsymbol{w}^T\boldsymbol{x})$$

- 真实条件概率$p_r(y|\boldsymbol{x})$
  - 对于一个样本(x,y^*)，其真实条件概率为

$$p_r(y=1|\boldsymbol{x})=y^*$$
$$p_r(y=0|\boldsymbol{x})=1-y^*$$

如何衡量两个条件分布的差异?

熵（Entropy）

- 在信息论中，熵用来衡量一个随机事件的不确定性。

$$H(p) = -\sum_{x} p(x) \log p(x)$$

  - 熵越高，则随机变量的信息越多
  - 熵越低，则随机变量的信息越少

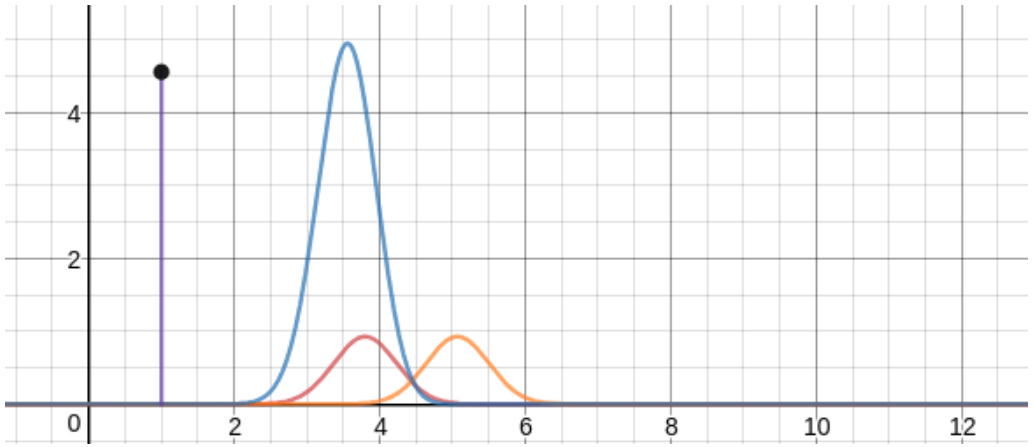
- 在对分布 p 的符号进行编码时，熵也是理论上最优的平均编码长度，这种编码方式称为熵编码（Entropy Encoding）

交叉熵（Cross Entropy）

- 交叉熵是按照概率分布q的最优编码对真实分布为p的信息进行编码的平均编码长度。

$$H(p,q) = -\sum_{x} p(x)\log q(x)$$

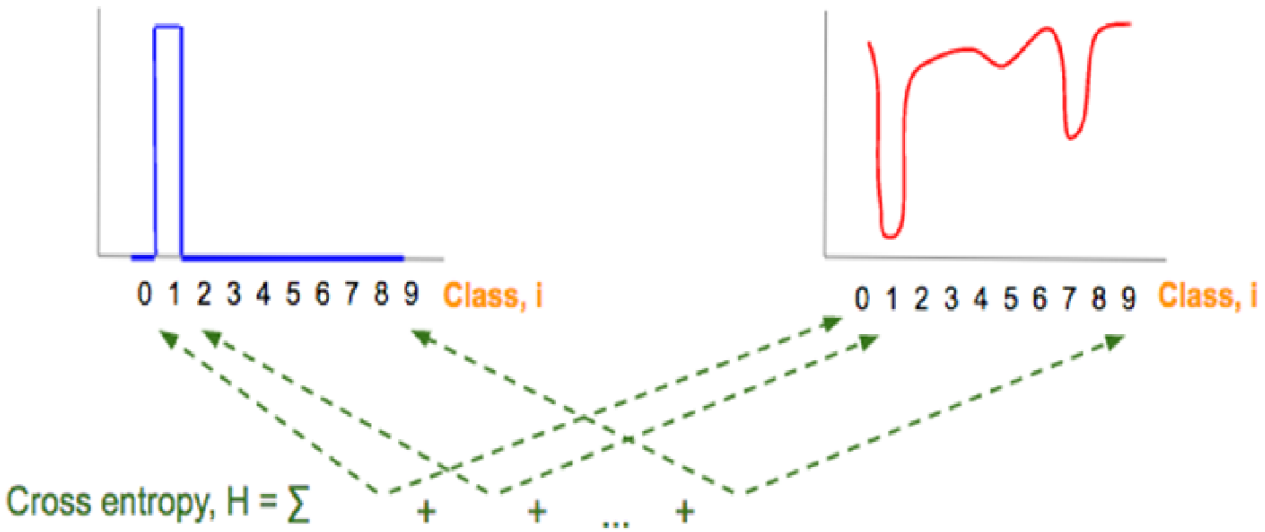
- 在给定p的情况下，如果 p和 q越接近，交叉熵越小；
  - $p = q$时，交叉熵等于熵
- 如果p和 q越远，交叉熵就越大 .



交叉熵

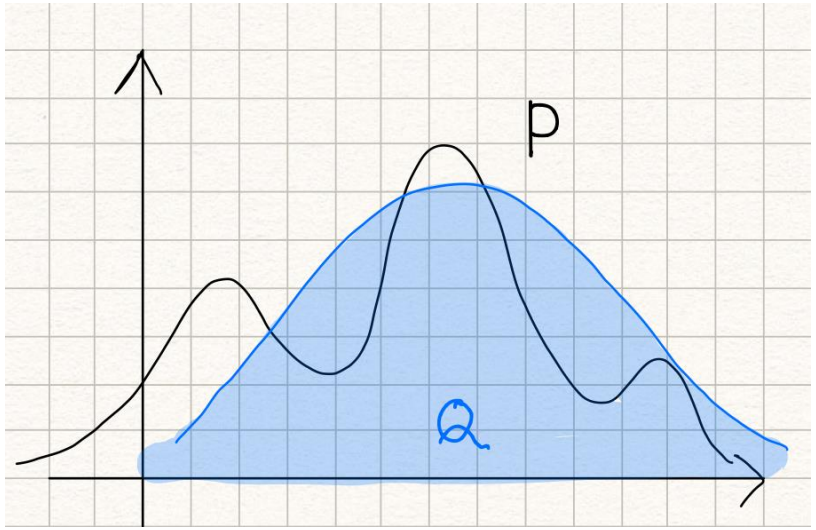
概率p

概率q的负对数$-\log q$



KL散度（Kullback-Leibler Divergence）

- KL散度是用概率分布q来近似p时所造成的信息损失量。
  - KL散度是按照概率分布q的最优编码对真实分布为p的信息进行编码，其平均编码长度（即交叉熵）H(p,q)和p的最优平均编码长度（即熵）H(p)之间的差异。



$$\mathrm{KL}(p,q) = H(p,q) - H(p)$$

$$= \sum_x p(x) \log \frac{p(x)}{q(x)}$$

交叉熵损失

$$\mathrm{KL}(p_r, p_\theta) = \sum_{y=0}^{1} p_r(y|\mathbf{x}) \log \frac{p_r(y|\mathbf{x})}{p_\theta(y|\mathbf{x})}$$

KL散度

$$\propto -\sum_{y=0}^{1} p_r(y|\mathbf{x}) \log p_\theta(y|\mathbf{x})$$

交叉熵损失

y^*为x的真实标签

$$= -p_r(y=1|\mathbf{x}) \log p_\theta(y=1|\mathbf{x}) - p_r(y=0|\mathbf{x}) \log(1 - p_\theta(y=1|\mathbf{x}))$$

$$= -y^* \log p_\theta(y=1|\mathbf{x}) - (1-y^*) \log(1 - p_\theta(y=1|\mathbf{x}))$$

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

负对数似然

梯度下降

- 交叉熵损失函数，模型在训练集的风险函数为

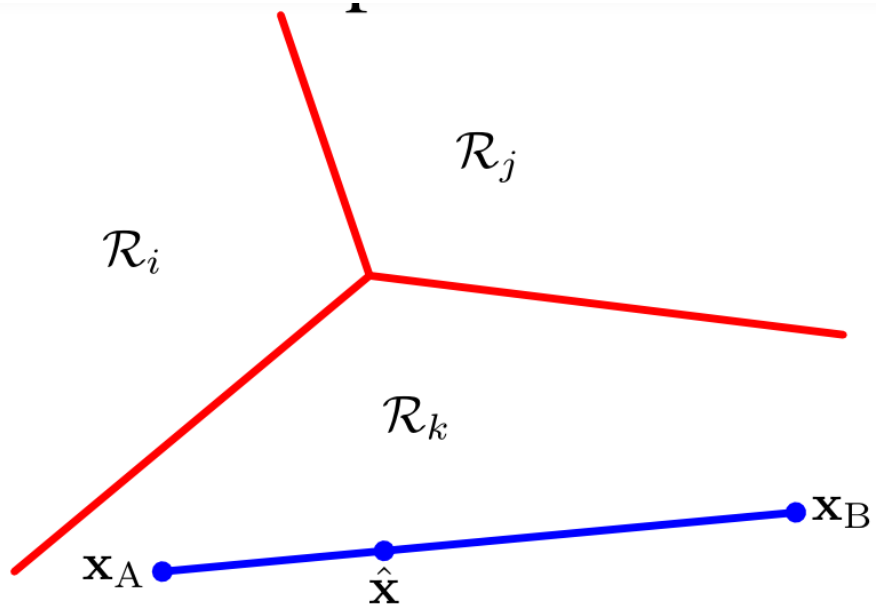
$$\mathcal{R}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\log\hat{y}^{(n)} + (1-y^{(n)})\log(1-\hat{y}^{(n)})\right)$$

- 梯度为

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

Softmax回归

多分类问题的决策边界



- A solution is to build K linear functions:

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}$$

  assign $\boldsymbol{x}$ to class $\arg\max_k y_k(\boldsymbol{x})$
- Gives connected, convex decision regions

$$\begin{aligned} \hat{\boldsymbol{x}} &= \lambda \boldsymbol{x}_A + (1-\lambda)\boldsymbol{x}_B \\ y_k(\hat{\boldsymbol{x}}) &= \lambda y_k(\boldsymbol{x}_A) + (1-\lambda) y_k(\boldsymbol{x}_B) \\ \Rightarrow y_k(\hat{\boldsymbol{x}}) &> y_j(\hat{\boldsymbol{x}}), \forall j \neq k \end{aligned}$$

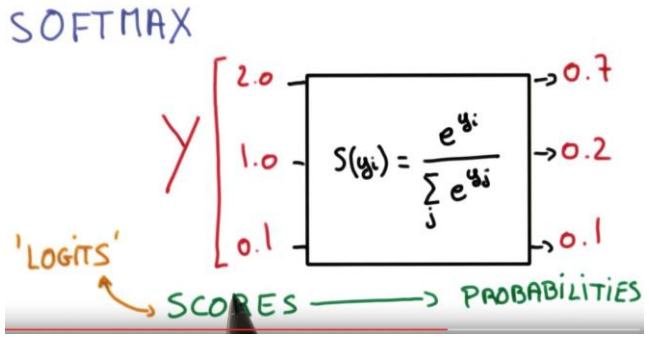
Softmax回归

- 多分类问题

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

- 利用softmax函数，目标类别y = c的条件概率为：

$$P(y = c|\mathbf{x}) = \text{softmax}(\mathbf{w}_c^{\top}\mathbf{x})$$
$$= \frac{\exp(\mathbf{w}_c^{\top}\mathbf{x})}{\sum_{i=1}^{C}\exp(\mathbf{w}_i^{\top}\mathbf{x})}.$$



参数学习

- 模型：Softmax回归
- 学习准则：交叉熵

$$\mathcal{R}(W) = -\frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}^{(n)})^{\mathrm{T}}\log\hat{\mathbf{y}}^{(n)}$$

- 优化：梯度下降

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

交叉熵损失函数

- 负对数似然损失函数

$$\mathcal{L}(\mathbf{y}, f(\mathbf{x}, \theta)) = -\sum_{c=1}^{C} y_c \log f_c(\mathbf{x}, \theta)$$

- 对于一个三类分类问题，类别为[0,0,1]，预测类别概率为[0.3,0.3,0.4]，则

$$\begin{aligned}\mathcal{L}(\theta) &= -\big(0 \times \log(0.3) + 0 \times \log(0.3) + 1 \times \log(0.4)\big) \\ &= -\log(0.4).\end{aligned}$$

总结

- 广义线性模型
  - 线性决策边界
- 学习准则
  - 最小平方误差
  - Fisher准则
  - 感知器准则
- 概率模型
  - 生成模型 VS 判别模型
  - Logistic回归、Softmax回归
  - 学习准则
    - 交叉熵=KL散度=负对数似然