模式识别与机器学习

4. 16、 Softmax判据的学习

待学习的参数

学什么

$$\text{Softmax:} \quad \max_i \frac{\exp(\boldsymbol{w}_i^T \boldsymbol{x} + w_{0i})}{\sum_{j=1}^{K} \exp(\boldsymbol{w}_j^T \boldsymbol{x} + w_{0j})}$$

- 给定训练样本，学习K组参数$\{\boldsymbol{w}_i，w_{0i}\}_{i=1,2,\ldots K}$。

训练样本

训练样本

- 给定K个类别，共N个标定过的训练样本：

$$\mathcal{X} = \{(\boldsymbol{x}_1, \boldsymbol{t}_1), (\boldsymbol{x}_2, \boldsymbol{t}_2) \ldots, (\boldsymbol{x}_N, \boldsymbol{t}_N)\}$$

- 输出真值采用one-hot方式：
  - ✓ 假设$\boldsymbol{x}_n \in C_i$，则
  - ✓ 为了每一项概率分布表述方便，$\boldsymbol{t}_n$也表达为集合形式：

$$\boldsymbol{t}_n : \{t_n^i\}_{i=1,2,\ldots,K}$$

$$\boldsymbol{t}_n = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \longleftarrow i\text{-}th$$

目标函数

Softmax模型输出的概率分布

- 给定单个样本$\boldsymbol{x}$，one-hot形式的输出标签$\{l_i\}_{i=1,\ldots,K}$符合多项分布（试验次数$N=1$）。
  - ✓ 分布参数p_i：模型输出属于C_i类的后验概率z_i，其为待学习参数。

$$p(\{l_i\}|\boldsymbol{x}) = \prod_{i=1}^{K} z_i^{l_i}, \quad \text{where } z_i = p(C_i|\boldsymbol{x}) = \frac{\exp(\boldsymbol{w}_i^T\boldsymbol{x}+w_{0i})}{\sum_{j=1}^{K}\exp(\boldsymbol{w}_j^T\boldsymbol{x}+w_{0j})}$$

多项分布（Multinomial）

- 随机变量x有K个互斥的取值，每个取值对应的概率为p_i，$\sum_{i=1}^{K} p_i = 1$。
- 该随机变量x试验N次，第i个值的次数为N_i（$\sum_{i=1}^{K} N_i = N$），则试验次数的概率联合分布为多项分布：

$$\mathrm{Multi}(N_1, N_2, \ldots, N_K; \{p_i\}) = N! \prod_{i=1}^{K} \frac{p_i^{N_i}}{N_i!}$$

- 若$N=1$，则多项分布表达为：

$$\prod_{i=1}^{K} p_i^{N_i}$$

目标函数：最大似然估计

似然函数

- 如果参数是最优的，意味着对大部分样本$(\boldsymbol{x}_n, \boldsymbol{t}_n)$而言，输出概率$p(\{t_n^i\}|\boldsymbol{x}_n)$ 应该是较大的（无论t_n^i取值是1还是0）。
- 因此，使用最大似然估计：针对所有训练样本$\mathcal{X}$，最大化输出标签分布的似然函数，以此求得参数$\{\boldsymbol{w}_i，w_{0i}\}_{i=1,2,\ldots K}$的最优值。
- 似然函数为所有训练样本输出概率的乘积。
- 所以，目标函数表达为：

$$\max L\left(\{\boldsymbol{w}_i, w_{0i}\}|\mathcal{X}\right) = \max \prod_{n=1}^{N} p(\{t_n^i\}|\boldsymbol{x}_n)$$

目标函数：最大似然估计

目标函数

$$\max L\left(\{\boldsymbol{w}_i, w_{0i}\}|\mathcal{X}\right) = \max \prod_{n=1}^{N} p(\{t_n^i\}|\boldsymbol{x}_n)$$

$$p\big(\{t_i\}|\boldsymbol{x}\big) = \prod_{i=1}^{K} z_i^{t_i}$$

- 对似然函数求取log再取反，得到目标函数：

$$\begin{aligned}\min J &= -\log L \\ &= -\log \prod_{n=1}^{N} \prod_{i=1}^{K} (z_n^i)^{t_n^i} = -\sum_{n=1}^{N} \sum_{i=1}^{K} t_n^i \log z_n^i\end{aligned}$$

- 该目标函数其实就是交叉熵的表达式。

目标函数：交叉熵解释

Softmax模型输出的概率分布

- 针对单个样本$\boldsymbol{x}_n$，可以得到softmax预测输出的概率分布$p\left(\{l_n^i\}_{i=1,\ldots,K}|\boldsymbol{x}_n\right)$、输出真值的概率分布$q\left(\{l_n^i\}_{i=1,\ldots,K}|\boldsymbol{x}_n\right)$。
- p和q都是多项分布（试验次数$N=1$），分布情况如下表：

给定样本$\boldsymbol{x}_n$	模型预测输出的概率分布 $p\left(\{l_n^i\}_{i=1,\ldots,K}\vert\boldsymbol{x}_n\right)$	输出真值的概率分布 $q\left(\{l_n^i\}_{i=1,\ldots,K}\vert\boldsymbol{x}_n\right)$
属于每个类C_i的概率	z_n^i	t_n^i

目标函数：交叉熵解释

交叉熵

- 给定单个样本$\boldsymbol{x}_n$，希望softmax模型预测输出的概率分布$p\ (\boldsymbol{l}_n|\boldsymbol{x}_n)$符合输出真值的概率分布$q(\boldsymbol{l}_n|\boldsymbol{x}_n)$。
  - ✓ 使用交叉熵来估计这两种分布的差异程度：

$$H(p,q) = -\sum_{i=1}^{K} t_n^i \log z_n^i$$

- 给定N个训练样本，把每个样本的交叉熵求和，得到目标函数：

$$\min_{\boldsymbol{w},w_0} -\sum_{n=1}^{N}\sum_{i=1}^{K} t_n^i \log z_n^i$$

目标函数优化

对参数$\boldsymbol{w}_k$求偏导

$$J(\{\boldsymbol{w}_i, w_{0i}\}) = -\sum_{n=1}^{N}\sum_{i=1}^{K} t_n^i \log z_n^i$$

$$\text{where} \quad z_n^i = \frac{\exp(y_n^i)}{\sum_{j=1}^{K}\exp(y_n^j)}, \quad y_n^i = \boldsymbol{w}_i^T \boldsymbol{x}_n + w_{0i}$$

- 对于任意参数（为区分求和索引i，记做$\boldsymbol{w}_k$），其只与对应的y_n^k有关，但所有的$\{y_n^k\}_{k=1,\ldots,K}$决定了一个z_n^i，所有的$\{z_n^i\}_{i=1,\ldots,K,n=1,\ldots,N}$决定了$J$。
- 因此，基于链式法则，目标函数对任意参数$\boldsymbol{w}_k$的偏导表达为：

$$\frac{\partial J}{\partial \boldsymbol{w}_k} = -\sum_{n=1}^{N}\sum_{i=1}^{K} \frac{\partial J}{\partial z_n^i}\frac{\partial z_n^i}{\partial y_n^k}\frac{\partial y_n^k}{\partial \boldsymbol{w}_k}$$

目标函数优化

对参数$\boldsymbol{w}_k$求偏导

$$J(\{\boldsymbol{w}_i, w_{0i}\}) = -\sum_{n=1}^{N}\sum_{i=1}^{K} t_n^i \log z_n^i$$

$$\text{where}\quad z_n^i = \frac{\exp(y_n^i)}{\sum_{j=1}^{K}\exp(y_n^j)},\quad y_n^i = \boldsymbol{w}_i^T\boldsymbol{x}_n + w_{0i}$$

$$\frac{\partial J}{\partial \boldsymbol{w}_k} = -\sum_{n=1}^{N}\sum_{i=1}^{K}\frac{\partial J}{\partial z_n^i}\frac{\partial z_n^i}{\partial y_n^k}\frac{\partial y_n^k}{\partial \boldsymbol{w}_k}$$

$$= -\sum_{n=1}^{N}\sum_{i=1}^{K} t_n^i(\delta_{ik} - z_n^k)\boldsymbol{x}_n$$

$$= \sum_{n=1}^{N}(z_n^k - t_n^k)\boldsymbol{x}_n$$

$$\sum_{i=1}^{K} t_n^i\delta_{ik} = t_n^i$$
$$\sum_{i=1}^{K} t_n^i z_n^k = z_n^k$$

$$\frac{\partial J}{\partial z_n^i} = \frac{t_n^i}{z_n^i}$$

$$\delta_{ik} = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{otherwise}\end{cases}$$

$$\frac{\partial z_n^i}{\partial y_n^k} = z_n^i(\delta_{ik} - z_n^k)$$

$$\frac{\partial y_n^k}{\partial \boldsymbol{w}_k} = \boldsymbol{x}_n$$

目标函数优化

对参数w_{0k}求偏导

$$J(\{\boldsymbol{w}_i, w_{0i}\}) = -\sum_{n=1}^{N}\sum_{i=1}^{K} t_n^i \log z_n^i$$

$$\text{where} \quad z_n^i = \frac{\exp(y_n^i)}{\sum_{j=1}^{K}\exp(y_n^j)}, \quad y_n^i = \boldsymbol{w}_i^T \boldsymbol{x}_n + w_{0i}$$

- 目标函数对任意w_{0k}的偏导为：

$$\frac{\partial J}{\partial w_{0k}} = -\sum_{n=1}^{N}\sum_{i=1}^{K} \frac{\partial J}{\partial z_n^i}\frac{\partial z_n^i}{\partial y_n^k}\frac{\partial y_n^k}{\partial w_{0k}}$$

$$= -\sum_{n=1}^{N}\sum_{i=1}^{K} t_n^i(\delta_{ik} - z_n^k) = \sum_{n=1}^{N}(z_n^k - t_n^k)$$

梯度分析

梯度分析

- 从梯度公式可以看到，第i个线性方程的参数$\boldsymbol{w}_i$和 w_{0i}的更新不仅依赖于第i类的样本$\boldsymbol{x}_n \in C_i$，而且还依赖于所有剩余类的样本$\boldsymbol{x}_n \in C_j, i \neq j$。
- 可见，判别式学习是依赖所有类的训练样本来学习参数。

$$\frac{\partial J}{\partial \boldsymbol{w}_i} = \sum_{n=1}^{N} (z_n^i - t_n^i)\boldsymbol{x}_n$$

$$\frac{\partial J}{\partial w_{0i}} = \sum_{n=1}^{N} (z_n^i - t_n^i)$$

梯度对比

Softmax vs. Logistic

$$\frac{\partial J}{\partial \boldsymbol{w}_i} = \sum_{n=1}^{N} (z_n^i - t_n^i)\boldsymbol{x}_n$$

$$\frac{\partial J(\boldsymbol{w}, w_0)}{\partial \boldsymbol{w}} = \sum_{n=1}^{N} (z_n - t_n)\boldsymbol{x}_n$$

$$\frac{\partial J}{\partial w_{0i}} = \sum_{n=1}^{N} (z_n^i - t_n^i)$$

$$\frac{\partial J}{\partial w_0} = \sum_{n=1}^{N} (z_n - t_n)$$

- Softmax：针对每个输出类别分别计算梯度值，但每个参数的梯度值与所有类别样本都相关。

目标函数优化

参数更新

- 采用梯度下降法更新所有$\{\boldsymbol{w}_i, w_{0i}\}$：
  - ✓ 设当前时刻为k，下一个时刻为$k+1$
  - ✓ η为更新步长。

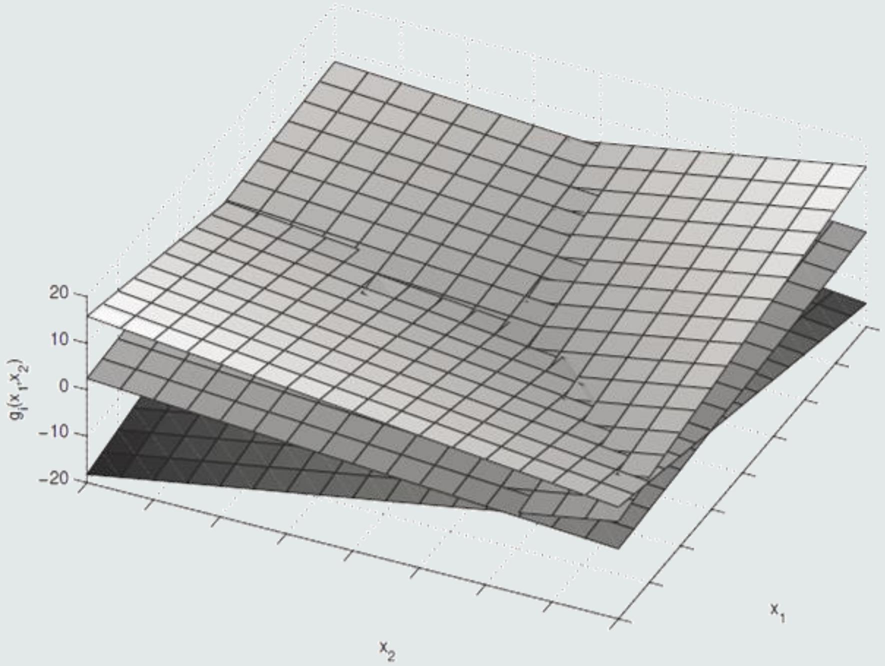
$$\boldsymbol{w}_i^{(k+1)} = \boldsymbol{w}_i^{(k)} - \eta^{(k)} \sum_{n=1}^{N} \left(z_n^i \big|_{\{\boldsymbol{w}_i^{(k)}, w_{0i}^{(k)}\}} - t_n^i \right) \boldsymbol{x}_n$$

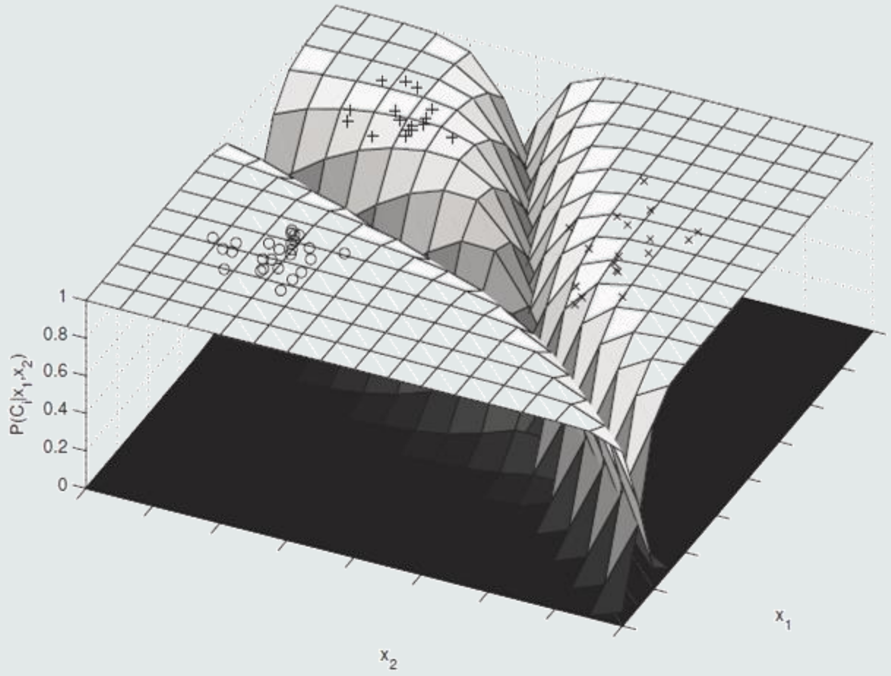
$$w_{0i}^{(k+1)} = w_{0i}^{(k)} - \eta^{(k)} \sum_{n=1}^{N} \left(z_n^i \big|_{\{\boldsymbol{w}_i^{(k)}, w_{0i}^{(k)}\}} - t_n^i \right)$$

Softmax判据学习示例

- $K=3$
- 左图：3条线性判据$\boldsymbol{w}_i^T\boldsymbol{x}+w_{0i}=0$
- 右图：3条softmax$(\boldsymbol{w}_i^T\boldsymbol{x}+w_{0i})$曲线

✓ Softmax输出的非线性形式就是exp函数的非线性形式。





Softmax判据虽然输出非线性，但仍然只能刻画线性分类边界，如何实现非线性分类边界?