目录

插图

机器学习面试笔记-感知机算法

ShenHengheng

1 简介

感知机算法是非常简单的关于二分类的算法, 是**线性分类模型**, 输入是样本的特征, 输出为样本的类别, 取 +1 和-1, 感知机算法针对输入的样本空间将实例划分为正负两类的分离超平面, 属于**判别模型**, 另外由于感知机算法旨在学习将训练数据进行划分的线性超平面, 为此导入基于误分类的损失函数, 利用梯度下降算法进行最小化损失, 求得感知机模型.

感知机算法有两种表示, 具体分为原始形式和对偶形式. 感知机预测是将学习到的感知机模型对新的预测样本进行分类.

感知机算法具有简单, 易用的特点, 也有很多机器学习算法都借鉴了感知机算法的一些思想, 可以说感知机算法是支持向量机算法, 神经网络模型等算法的基础, 感知机算法在 1957 年由 Rosenblatt 提出.

2 感知机模型

定义: 假设输入空间 (特征空间) 是 $\mathcal{X} \subseteq \mathbf{R}^n$, 输出空间是 $\mathcal{Y} = \{+1, -1\}$, 输入 $x \in \mathcal{X}$ 表示输入实例的特征向量, 对应于输入空间的点, 输出 $y \in \mathcal{Y}$ 表示实例的类别, 由输入空间到输出空间的如下函数:

$$f(x) = \text{sign}(w \cdot x + b) \tag{1}$$

称为感知机, 其中 w 和 b 为感知机模型参数, $w \in \mathbf{R}^n$ 叫做权值 (weight) 或者权值向量, $b \in \mathbf{R}$ 叫做偏置 $w \cdot x$ 表示 w 和 b 的内积, sign 为符号函数, 即

$$\text{sign}(x) = \begin{cases} +1 & x = 0 \\ -1 & x < 0 \end{cases}.$$

几何解释: 感知机有以下几何解释, 线性方程

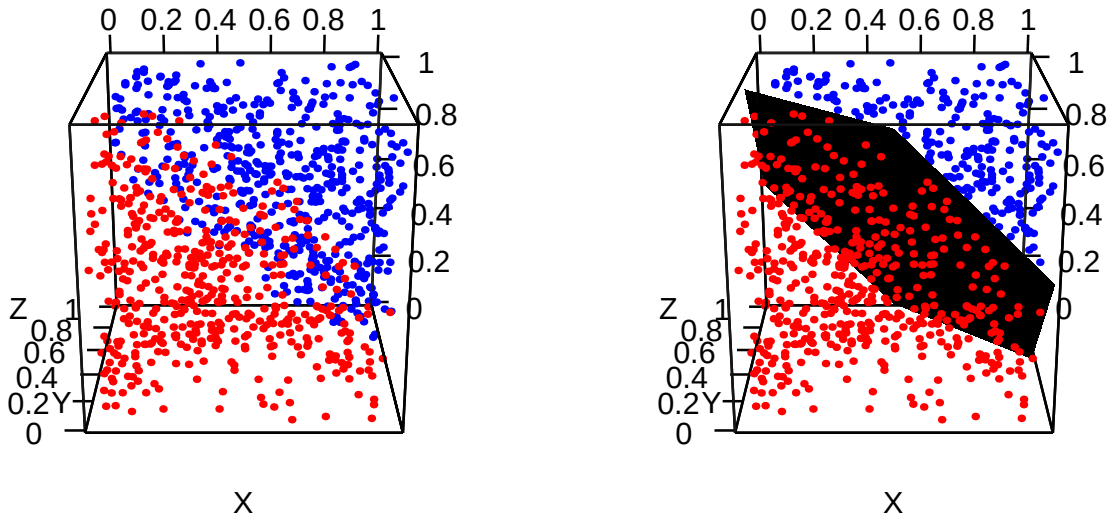
$$w \cdot x + b = 0$$

对应于特征空间 $\mathbf{R}^n$ 中的一个超平面 $\mathcal{S}$, 其中 w 是超平面的法向量, b 是超平面的截距. 这个超平面将特征空间划分为两个部分, 位于两部分的点分别分布在超平面的两侧, 被分为正负两类, 因此, 超平面 $\mathcal{S}$ 又被称作分离超平面.

Notation: 感知机学习的模型表示如下感知机学习, 由训练数据集 (实例的特征向量及类比)

$$T = \{(x_1, y_1), (x_1, y_1), (x_1, y_1)\}$$

其中, $x_i, x_i \in \mathbf{R}^n$ 表示一个样本实例的特征向量, $y_i, y_i \in \{+1, -1\}$, 其中 $i = 1, 2, 3...$, 即我们的样本只有正负两类, 求得感知机模型 (参考公式 1) 参数 w, b, 感知机预测, 通过学习得到的感知机模型, 对于新的测试样本, 给出对应的类别.



(a) 特征空间表示　　(b) 分离超平面将训练数据集划分为两类

图 1: 分离超平面

3 感知机学习策略

3.1 数据集的线性可分性

定义:(数据集的线性可分性) 给定一个数据集

$$T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_N, y_N)\},$$

其中, $x_i \in \mathbf{R}^n, y_i \in \{+1, -1\}, \quad i = 1, 2, 3, ..., N$, 如果存在某个超平面 $\mathcal{S}$

$$w \cdot x + b = 0$$

能够将数据集的正样本和负样本完全正确地划分到超平面的两侧, 即对所有的 $y_i = +1$ 的实例 i, 有 $w \cdot x_i + b > 0$, 同样地, 对于所有的 $y_i = -1$ 的实例 j, 有 $w \cdot x_j + b < 0$, 则称数据集 T 为线性可分数据集, 否则称为线性不可分数据集.

3.2 感知机的学习策略

假设给定一个数据集, 它是线性可分的, 这时我们需要找到一个分离超平面 $\mathcal{S}$, 使得能够将两类实例分离到超平面的两侧, 即需要确定超平面的模型参数 w, b, 那么怎么求出这些参数呢? 这就需要定义一个学习策略.

什么是学习策略: 学习策略就是要定义一个经验损失函数, 并将 (经验) 损失函数最小化.

往往对于二分类或者说线性可分数据集的分类方法, 有两种损失函数可以考虑:(1) **误分类样本的数量**作为模型的损失函数, 只需要数出误分类的实例数量即可, 但是这种方法不适合优化, 因为它不是关于参数 w, b 的连续可导函数, 对于我们最小化 (优化) 损失函数是很难的.(2) **误分类点到分离超平面的距离作为损失函数**, 这种方法是**感知机**采用的损失函数.

为此, 首先给出样本空间 $\mathbf{R}^n$ 的任一点 x_i 到分离超平面的距离为:

$$\frac{1}{||w||}|w \cdot x_i + b|$$

这里, $||w||$ 是 w 的 L_2 范数.

其次, 对于误分类的数据 (x_i, y_i) 来说,

$$-y_i(w \cdot x_i + b) > 0$$

成立, 因为当 $w \cdot x_i + b > 0$, $y_i = -1$, 而当 $w \cdot x_i + b < 0$, $y_i = +1$, 因此误分类点 x_i 到分离超平面的距离为:

$$-\frac{1}{||w||}|w \cdot x_i + b|$$

这样, 假设超平面 $\mathcal{S}$ 的误分类点的集合为 M, 那么所有误分类点到超平面 $\mathcal{S}$ 的总距离为:

$$-\frac{1}{||w||}\sum_{x_i \in M} y_i(w \cdot x_i + b)$$

不考虑 $-\frac{1}{||w||}$, 就得到了感知机学习的损失函数.

定义: (感知机学习的损失函数) 给定一个训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_N, y_N)\},$$

其中, $x_i \in \mathbf{R}^n, y_i \in \{+1, -1\}, \quad i = 1, 2, 3, ..., N$, 感知机 s 学习的损失函数定义为:

$$L(w, b) = -\frac{1}{||w||}\sum_{x_i \in M} y_i(w \cdot x_i + b) \tag{2}$$

其中, M 为误分类点的集合. 这个函数就是经验损失函数.

结论: 通过上面的公式, 可以发现, 损失函数 $L(w, b)$ 是非负的, **如果没有误分类点, 损失函数值为 0**, 而且误分类点越少, 误分类点离平面越近 (即 $-\frac{1}{||w||}y_i(w \cdot x_i + b)$ 与与越小), 损失函数值就越小. 因为对于特定的数据集的损失函数, 在误分类是从参数的 w, b 的損失函数; 在正确分类时是 0, 因此给定训练数据集, 损失函数 $L(w, b)$ 是关于 w, b 的连续可导函数.

4 感知机学习算法

感知机学习问题转化为感知机学习的损失函数 (参考 2) 最优化问题. 最优化算法这里选择梯度下降算法. 本节主要讲解感知机学习的具体算法 (包含原始形式和对偶形式), 并证明算法在线性可分数据集下的收敛性.

4.1 感知机学习算法的原始形式

给定一个训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_N, y_N)\},$$

其中, $x_i \in \mathbf{R}^n, y_i \in \{+1, -1\}, \quad i = 1, 2, 3, ..., N$, 求 w, b, 使其为以下损失函数极小化问题的解:

$$L(w, b) = -\frac{1}{||w||}\sum_{x_i \in M} y_i(w \cdot x_i + b) \tag{3}$$

其中, M 为误分类点的集合.

由于感知机算法是误分类点驱动的, 并且在梯度下降过程中, 不是误分类点集合 M 中的所有误分类点全部梯度下降, 而是每个误分类点梯度下降.

假设误分类点的集合 M 是固定的, 那么损失函数 (参考 2) 的梯度由:

$$\nabla_w L(w, b) = \sum_{x_i \in M} y_i \cdot x_i \tag{4a}$$

$$\nabla_b L(w, b) = \sum_{x_i \in M} y_i \tag{4b}$$

给出.

随机选择一个误分类点 (x_i, y_i), 那么对它的 (w, b) 进行更新如下:

$$w \leftarrow w + \eta y_i x_i \tag{5a}$$

$$b \leftarrow b + \eta y_i \tag{5b}$$

式中 $\eta, 0 < \eta \leq 1$ 表示学习率, 通过不断的更新 w, b, 使得损失函数 $L(w, b)$ 不断减小, 直至为 0.

算法 1: 感知机学习算法的原始形式

输入: 训练数据集 $T = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_N, y_N)\}$, 其中, $x_i \in \mathbf{R}^n, y_i \in \{+1, -1\}, \quad i = 1, 2, 3, ..., N$, , 学习率 $\eta, 0 < \eta \leq 1$.

输出: w, b, 感知机模型 $f(x) = \text{sign}(w \cdot x + b)$

(1) 选取初始值 w_0, b_0

(2) 在训练集中选取数据 (x_i, y_i)

(3) 如果 $y_i(w \cdot x_i + b) \leq 0$

$$w \leftarrow w + \eta y_i x_i \tag{6a}$$

$$b \leftarrow b + \eta y_i \tag{6b}$$

(4) 转至 (2), 直至训练集中没有误分类.

4.2 使用 R 实现感知机算法

感知机模型

```
Classify <- function(x, weights) {
  return(sign(x %*% weights))
}
```

感知机学习算法

```
Perceptron <- function(data, threshold) {
  w <- c(-threshold, runif(ncol(data) - 2))
  n <- nrow(data)
  label <- data[ , 1]
  obs <- data[ , 2:ncol(data)]
  misclassfied <- TRUE
  while (misclassfied) {
    misclassfied <- FALSE
    for (i in 1:n) {
      if(label[i]*Classify(obs[i,], w)<=0){
        w <- w + label[i] * obs[i , ]
        misclassfied <- TRUE
      }
    }
  }
  return(w)
}
```

4.3 算法的收敛性证明

4.4 感知机算法的对偶形式