CH02 感知机

前言

章节目录

1. 感知机模型
2. 感知机学习策略
    a. 数据集的线性可分性
    b. 感知机学习策略
    c. 感知机学习算法
3. 感知机学习算法
    a. 感知机学习算法的原始形式
    b. 算法的收敛性
    c. 感知机学习算法的对偶形式

导读

感知机是二类分类的**线性分类模型**。

- 损失函数$L(w,b)$的经验风险最小化
- 本章中涉及到向量内积，有超平面的概念，也有线性可分数据集的说明，在策略部分有说明损关于失函数的选择的考虑，可以和CH07一起看。另外， 感知机和SVM的更多联系源自margin的思想， 实际上在

本章的介绍中并没有体现margin的思想，参考文献中有给出对应的文献。
- 本章涉及的两个例子，思考一下为什么$\eta=1$，进而思考一下参数空间，这两个例子设计了相应的测试案例实现， 在后面的内容中也有展示。
- 在**收敛性证明**那部分提到了偏置合并到权重向量的技巧，合并后的权重向量叫做扩充权重向量，这点在LR和SVM中都有应用，但是这种技巧在书中的表示方式是不一样的，采用的不是统一的符号体系，或者说不是统一的。本书三个章节讨论过算法的收敛性，感知机，AdaBoost，EM算法。
- 第一次涉及Gram Matrix $G=[x_i\cdot x_j]_{N\times N}$
- 感知机的激活函数是符号函数。
- 感知机是神经网络和支持向量机的基础。
- 当我们讨论**决策边界**的时候, 实际上是在考虑算法的**几何解释**。
- 关于感知机为什么不能处理异或问题, 可以借助下图理解。

上面紫色和橙色为两类点, 线性的分割超平面应该要垂直于那些红粉和紫色的线.

- 提出感知机算法的大参考文献是本文第一篇文献， 这个文章发表在Psycological Review上。不过这个文章，真的不咋好看。
- 书中有提到函数间隔，几何间隔，这里间隔就是margin
- 在CH07中有说明， 分离超平面将特征空间划分为两个部分，一部分是正类， 一部分是负类。法向量指向的一侧为正类，另一侧为负类。
- 感知机损失函数$L=\max(0,-y_i(w\cdot x_i+b))$，这个在CH07中将hinge loss的时候有说明。

三要素

模型

输入空间：$\mathcal{X}\subseteq\mathbf{R^n}$

输出空间：$\mathcal{Y}=+1,-1$

> *决策函数：* $f(x)=sign(w\cdot x+b)$

策略

确定学习策略就是定义**(经验)**损失函数并将损失函数最小化。

注意这里提到了**经验**，所以学习是base在**训练数据集**上的操作

损失函数选择

> *损失函数的一个自然选择是误分类点的总数，但是，这样的损失函数不是参数w,b的连续可导函数，不易优化*
>
> *损失函数的另一个选择是误分类点到超平面S的总距离，这是感知机所采用的*

感知机学习的经验风险函数(损失函数)

$$L(w,b)=-\sum_{x_i\in M}y_i(w\cdot x_i+b)$$

其中M是误分类点的集合

给定训练数据集T，损失函数$L(w,b)$是w和b的连续可导函数

算法

原始形式

> *输入：* $T=\{(x_1,y_1),(x_2,y_2),\dots,(x_N,y_N)\}$
> $x_i\in\mathcal{X}=\mathbf{R^n},y_i\in\mathcal{Y}=\{-1,+1\},i=1,2,\dots,N;\ \ 0<\eta\leqslant 1$
>
> *输出：* $w,b;f(x)=sign(w\cdot x+b)$
>
> 1. *选取初值*w_0,b_0
> 2. *训练集中选取数据*(x_i,y_i)
> 3. *如果*$y_i(w\cdot x_i+b)\leqslant 0$
>
> $$w\leftarrow w+\eta y_ix_i$$
> $$b\leftarrow b+\eta y_i$$
>
> 4. *转至(2)，直至训练集中没有误分类点*

注意这个原始形式中的迭代公式，可以对x补1，将w和b合并在一起，合在一起的这个叫做扩充权重向量，书上有提到。

对偶形式

对偶形式的基本思想是将w和b表示为实例x_i和标记y_i的线性组合的形式，通过求解其系数而求得w和b。

> *输入：* $T=\{(x_1,y_1),(x_2,y_2),\dots,(x_N,y_N)\}$
> $x_i\in\mathcal{X}=\mathbf{R^n},\mathbf{y_i}\in\mathcal{Y}=\{-1,+1\},i=1,2,\dots,\mathcal{N};0<\eta\leqslant 1$
>
> *输出：*
>
> $$\alpha,b;f(x)=sign\left(\sum_{j=1}^{N}\alpha_jy_jx_j\cdot x+b\right)$$
> $$\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_N)^T$$
>
> 1. $\alpha\leftarrow 0,b\leftarrow 0$
> 2. *训练集中选取数据*(x_i,y_i)
> 3. *如果*$y_i\left(\sum_{j=1}^{N}\alpha_jy_jx_j\cdot x+b\right)\leqslant 0$

> $$\alpha_i \leftarrow \alpha_i + \eta$$
> $$b \leftarrow b + \eta y_i$$
>
> *4. 转至(2)，直至训练集中没有误分类点*

Gram matrix

对偶形式中，训练实例仅以内积的形式出现。

为了方便可预先将训练集中的实例间的内积计算出来并以矩阵的形式存储，这个矩阵就是所谓的Gram矩阵

$$G = [x_i \cdot x_j]_{N \times N}$$

例子

例2.1

这个例子里面$\eta = 1$

> *感知机学习算法由于采用不同的初值或选取不同的误分类点，解可以不同。*

另外，在这个例子之后，证明算法收敛性的部分，有一段**为了便于叙述与推导**的描述，提到了将偏置并入权重向量的方法，这个在涉及到内积计算的时候可能都可以用到，可以扩展阅读CH06，CH07部分的内容描述。

例2.2

这个例子也简单，注意两点

1. $\eta = 1$
2. $\alpha_i \leftarrow \alpha_i + 1, b \leftarrow b + y_i$

以上：

1. 为什么η选了1，这样得到的值数量级是1
2. 这个表达式中用到了上面的$\eta = 1$这个结果，已经做了简化

所以，这里可以体会下，调整学习率η的作用。学习率决定了参数空间。

Logic_01

经常被举例子的**异或**问题[1]，用感知机不能实现，因为对应的数据非线性可分。但是可以用感知机实现其他逻辑运算，也就是提供对应的逻辑运算的数据，然后学习模型。

这个例子的数据是二元的，其中NOT运算只针对输入向量的第一个维度

Logic_02

这个例子的数据是三元的

MNIST_01

这个选择两类数据进行区分，不同的选择应该得到的结果会有一定差异，数据不上传了，在sklearn里面有相应的数据，直接引用了，注意测试案例里面用的是01，相对来讲好区分一些。

问题

损失函数

知乎上有个问题

```
感知机中的损失函数中的分母为什么可以不考虑？
有些人解释是正数，不影响，但是分母中含有 w，而其也是未知数，在考虑损失函数的最值时候会不影响么？想不通
```

这个对应了书中P_{27}中`不考虑1/||w||，就得到感知机学习的损失函数`

题中问考虑损失函数最值的时候，不会有影响么？

1. 感知机处理线性可分数据集，二分类，$\mathcal{Y}=\{+1,-1\}$，所以涉及到的乘以 $\boldsymbol{y_i}$ 的操作实际贡献的是符号；
2. 损失函数 $L(w,b)=-\sum_{x_i\in M}y_i(w\cdot x_i+b)$,其中 M 是错分的点集合，线性可分的数据集肯定能找到超平面 S，所以这个损失函数最值是0。
3. 如果正确分类，$y_i(w\cdot x_i+b)=|w\cdot x_i+b|$ ，错误分类的话，为了保证正数就加个负号，这就是损失函数里面那个负号，这个就是函数间隔；
4. $\frac{1}{||w||}$ 用来归一化超平面法向量，得到几何间隔，也就是点到超平面的距离， 函数间隔和几何间隔的差异在于同一个超平面 (w,b) 参数等比例放大成 (kw,kb) 之后，虽然表示的同一个超平面，但是点到超平面的函数间隔也放大了，但是几何间隔是不变的；
5. 具体算法实现的时候， w要初始化，然后每次迭代针对错分点进行调整，既然要初始化，那如果初始化个 $||w||=1$ 的情况也就不用纠结了，和不考虑 $\frac{1}{||w||}$ 是一样的了；
6. 针对错分点是这么调整的

   $w\leftarrow w+\eta y_i x_i$
   $b\leftarrow b+\eta y_i$

   前面说了 y_i 就是个符号，那么感知机就可以解释为针对误分类点，通过调整 w,b 使得超平面向该误分类点一侧移动，迭代这个过程最后全分类正确；
7. 感知机的解不唯一，和初值有关系，和误分类点调整顺序也有关系；
8. 这么调整就能找到感知机的解？能，Novikoff还证明了，通过有限次搜索能找到将训练数据完全正确分开的分离超平面。

所以，

如果只考虑损失函数的最值，那没啥影响，线性可分数据集，最后这个损失就是0； 那个分母用来归一化法向量，不归一化也一样用，感知机的解不唯一；说正数不影响的应该考虑的是不影响超平面调整方向吧

参考

1.

↑ top

---

1. XOR↩