感知机学习算法与逻辑回归
Perceptron Learning Algorithm and Logistic Regression

陈昱夫

Lab1报告存在的一些问题

- 1. 实验原理很重要，写的是模型原理与公式推导等，而不是实现的流程。
- 2. 伪代码与流程图的规范。
- 3. 代码截图分模块，每一模块加一句话描述作用，在截图内补充必要的注释。
- 4. 结果分析着重在对比分析不同参数，不同创新点下的不同结果，不要直接截取大段代码运行结果，尽量通过图表可视化。
- 5. 思考题算送分题，不要忘记写。
- 6. 所有图片均需要注意排版，不要连续一大段截图，也不要太小以至于内容都看不清。

感知机学习算法（PLA）

- PLA针对二元分类问题：$y = \{+1, -1\}$

- 通过一个共享的权重向量$w = (w_1, w_2, \dots, w_{\mathrm{d}})$ 和某个样例的特征向量$x = (x_1, x_2, \dots, x_{\mathrm{d}})$，来计算该样例的分数，通过与某个阈值$\theta$比较大小，来判断样例的类别：

$$sign\left(\left(\sum_{\mathrm{i}=1}^{\mathrm{d}} \mathrm{w}_i x_i\right) - \theta\right)$$

感知机学习算法（PLA）

- 将阈值θ转化成模型待学习的参数：

$$
\begin{aligned}
& sign\left(\left(\sum_{\mathrm{i}=1}^{\mathrm{d}} \mathrm{w}_i x_i\right)-\theta\right) \\
&= sign\left(\left(\sum_{\mathrm{i}=1}^{\mathrm{d}} \mathrm{w}_i x_i\right)+(-\theta)\times(+1)\right) \\
&= sign\left(\sum_{\mathrm{i}=0}^{\mathrm{d}} \mathrm{w}_i x_i\right) \\
&= sign(W^T X)
\end{aligned}
$$

- 其中：

$$
W=(w_0, w_1, w_2, \ldots, w_{\mathrm{d}})
$$
$$
X=(+1, x_1, x_2, \ldots, x_{\mathrm{d}})
$$

感知机学习算法（PLA）

- PLA算法步骤：
  - 1. 给每一个样本的特征向量前加一维常数项1
  - 2. 随机初始化$(\mathrm{d}+1)$维的权重向量W_0
  - 3. 遍历训练样本，每当找到一个预测错误的样本$\boldsymbol{x}_i$，则更新权重向量，直到所有的训练样本都预测正确。

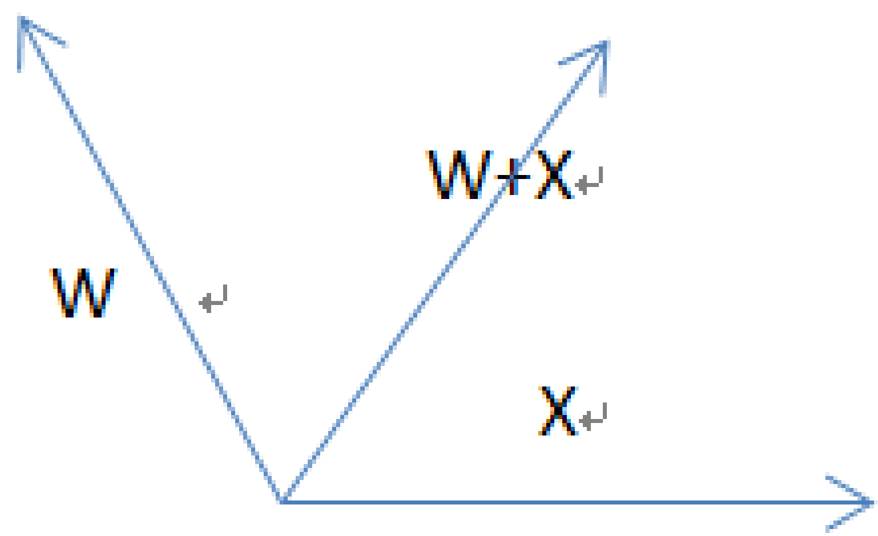
$$W_{t+1} \leftarrow W_t + \mathrm{y}_i \boldsymbol{x}_i$$

  - 4. 通过$sign\left(W_{final}\boldsymbol{x}\right)$来预测一个样本$\boldsymbol{x}$的标签

感知机学习算法（PLA）

- $W^{t+1} \leftarrow W^{t} + [\![\mathrm{y_i} \neq sign(W_t^T \boldsymbol{x}_i)]\!] \mathrm{y}_i \boldsymbol{x}_i$

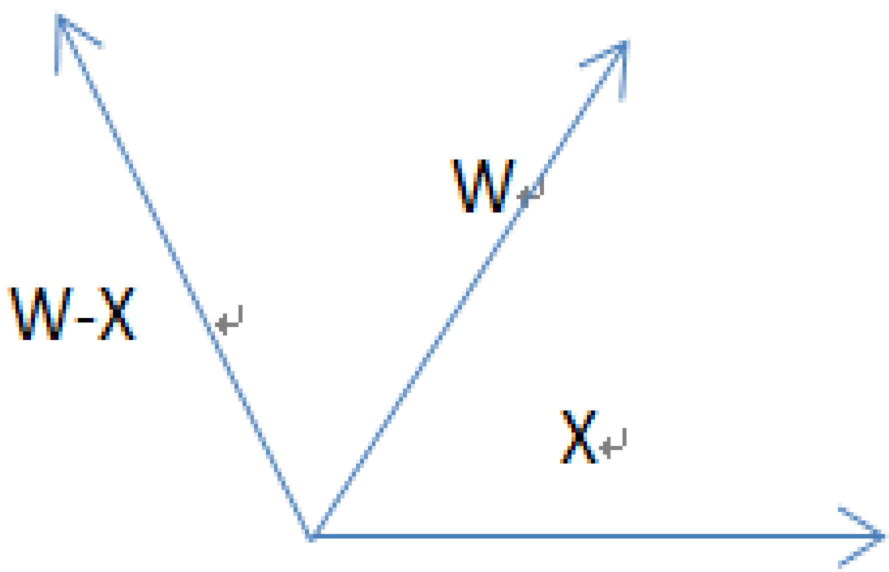
- 正样例被预测为负的情况下：



感知机学习算法（PLA）

- $W^{t+1} \leftarrow W^{t} + [\![\mathrm{y_i} \neq sign(W_t^T \boldsymbol{x}_i)]\!] \mathrm{y}_i \boldsymbol{x}_i$

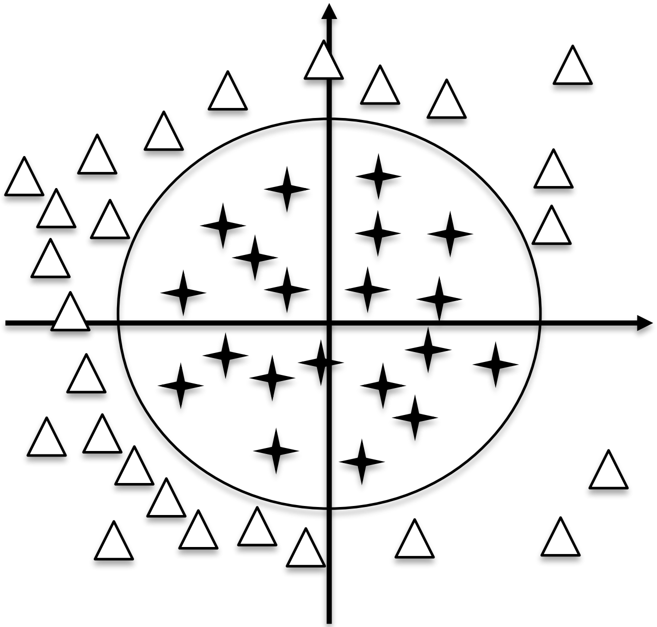
- 负样例被预测为正的情况下：



感知机学习算法（PLA）

- 原始的PLA不适用于非线性可分的数据集

思考题：有什么手段可以使PLA适用于非线性可分的数据集？



感知机学习算法（PLA）

- 原始的PLA不适用于非线性可分的数据集，两种解决思路：
  - 1. 设置迭代次数，到一定程度就返回此时的w，不管它到底满不满足所有训练集。
  - 2. 找一个w，使得在训练集里以此w来划分后，分类错误的样本最少。即相当于有一个口袋放着一个w，把算到的w跟口袋里的w比对，放入比较好的一个w，这种算法又被称为口袋（pocket）算法。

逻辑回归算法（LR）

- 硬分类模型：非概率模型，通常有一个决策函数来直接判断样例的类别，例如PLA，决策树等。

- 软分类模型：概率模型，通常先输出每个分类的概率，再根据概率大小来判断样例的类别。

逻辑回归算法（LR）

- 对于一个软分类模型，我们假设对某个样例x来说，属于某个类别y的条件概率为：$f(x) = P(y|x) \in [0,1]$

- 与PLA类似，我们通过一系列权重来计算某个样例的分数：$s = \sum_{t=0}^{d} w_i x_i = \boldsymbol{w}^T \boldsymbol{x}$

逻辑回归算法（LR）

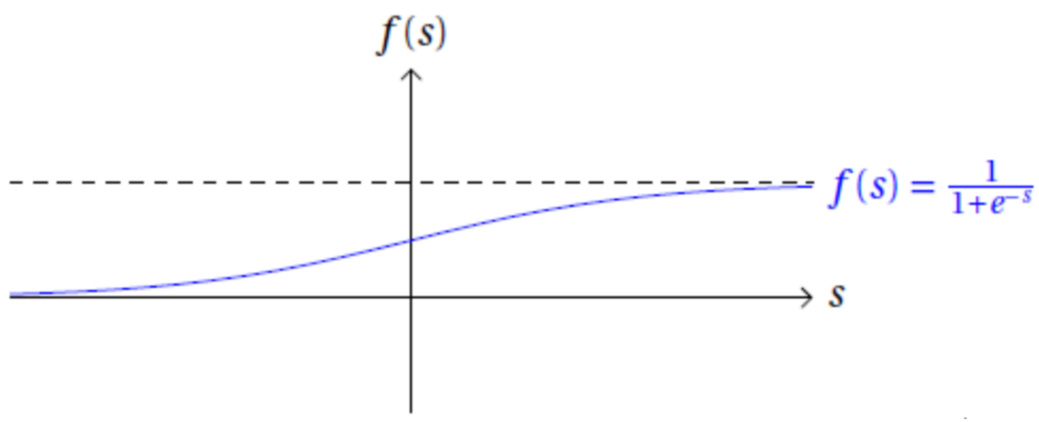
- 计算出来的分数s的范围是$(-\infty, +\infty)$，我们需要某种决策函数将加权分数映射到另一个更合理的数据空间，使加权分数的大小能够反映概率的大小

- 在逻辑回归里使用的是： *logistic function*

$$\theta(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}}$$

逻辑回归算法（LR）

- 在逻辑回归里使用的是： *logistic function*
  - $\theta(-\infty)=0$，当加权分数无穷小，该数据属于正类别的概率为 0
  - $\theta(0)=0.5$，当加权分数为 0，该数据属于任一类别的概率相同均为0.5
  - $\theta(+\infty)=1$，当加权分数无穷大，该数据属于正类别的概率为 1



逻辑回归算法（LR）

- 利用逻辑回归函数构建预测函数$h(x)$， $h(x)$相当于样本属于正类的概率，属于负类的概率为$1-h(x)=h(-x)$

$$h(x)=\theta(x)=\frac{1}{1+e^{-\boldsymbol{w}^T\boldsymbol{x}}}$$

- 那么某个样例x属于某个类别y的条件概率可以表示为

$$f(x)=P(y|x)=h(\mathrm{x})^y\big(1-h(x)\big)^{1-y}$$

  - 当$y=1$，$f(x)=P(\mathrm{y}=1|x)=h(x)$
  - 当$y=0$，$f(x)=P(\mathrm{y}=0|x)=1-h(x)$
  - 在某种模型下利用给定数据 x 得到给定标签 y 的概率，称之为该问题的似然（likelihood）

逻辑回归算法（LR）

- 考虑整个训练集的似然函数

$$likelihood = \prod_{i=1}^{N} P(\mathrm{y}|x_i) = \prod_{i=1}^{N} h(x_i)^{y_i}\left(1 - h(x_i)\right)^{1-y_i}$$

- 根据最大似然估计，我们需要找到一组参数使得似然最大。对似然函数取负对数得到目标函数$L(\mathrm{w})$

$$L(\mathrm{w}) = -log(likelihood)$$
$$= -\sum_{i=1}^{N}(y_i log(h(x_i)) + (1 - y_i)log(1 - h(x_i)))$$

- 我们的目的是取$L(\mathrm{w})$最小时的w作为模型最后的参数

逻辑回归算法（LR）

- 使用梯度下降法来优化问题，首先另目标函数对参数进行求导，
  设$u = 1 + e^{-w^T x_n}$， $\mathrm{v} = -w^T x_n$

$$
\begin{aligned}
\frac{\partial \mathrm{L}(w_i)}{\partial w_i} &= -\sum_{n=1}^{N}\left[(y_n)\left(\frac{\partial \log(h(x_n))}{\partial h(x_n)}\right)\left(\frac{\partial h(x_n)}{\partial u}\right)\left(\frac{\partial u}{\partial v}\right)\left(\frac{\partial v}{\partial w_i}\right) + (1-y_n)\left(\frac{\partial \log(1-h(x_n))}{\partial h(x_n)}\right)\left(\frac{\partial h(x_n)}{\partial u}\right)\left(\frac{\partial u}{\partial v}\right)\left(\frac{\partial v}{\partial w_i}\right)\right] \\
&= -\sum_{n=1}^{N}\left[(y_n)\left(\frac{1}{h(x_n)}\right) + (1-y_n)\left(\frac{-1}{1-h(x_n)}\right)\right]\left[\left(\frac{-1}{u^2}\right)(e^v)(-x_{n,i})\right] \\
&= -\sum_{n=1}^{N}\left[(y_n)\left(\frac{1}{h(x_n)}\right) - (1-y_n)\left(\frac{1}{1-h(x_n)}\right)\right][h(x_n)(1-h(x_n))](x_{n,i}) \\
&= -\sum_{n=1}^{N}[(y_n)(1-h(x_n)) - (1-y_n)h(x_n)](x_{n,i}) \\
&= -\sum_{n=1}^{N}(y_n - h(x_n))(x_{n,i})
\end{aligned}
$$

逻辑回归算法（LR）

- 求得目标函数的梯度为

$$\nabla L(\mathrm{w}) = -\sum_{\mathrm{n}=1}^{N}(y_{\mathrm{n}} - h(x_{\mathrm{n}}))(x_{\mathrm{n}})$$

- 根据梯度下降法，权重的更新公式为

$$w_{\mathrm{t+1}} \leftarrow w_t - \eta \nabla L(w_t)$$

思考题：不同的学习率η对模型收敛有何影响？从收敛速度和是否收敛两方面来回答。

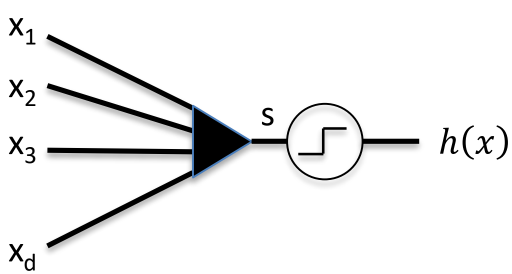
逻辑回归算法（LR）

- LR算法步骤：
  - 1. 给每一个样本的特征向量前加一维常数项1
  - 2. 随机初始化$(\mathrm{d}+1)$维的权重向量W_0
  - 3. 计算当前梯度$\nabla L(\mathrm{w}_t) = -\sum_{\mathrm{n}=1}^{N}\left(y_{\mathrm{n}} - h\left(x_{\mathrm{n}}\right)\right)\left(x_{\mathrm{n}}\right)$
  - 4. 根据梯度更新权重$w_{\mathrm{t}+1} \leftarrow w_t - \eta\nabla L(w_t)$
  - 5. 重复步骤3步骤4直到满足一定的收敛条件
  - 6. 根据模型最后的权重，通过$h(x)$的概率取值来预测某个样例x

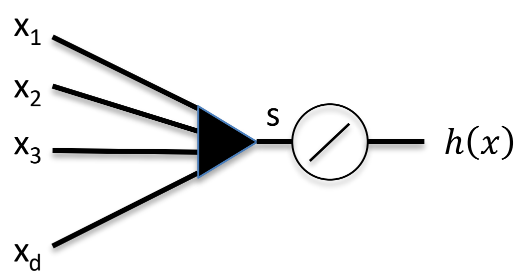
思考题：使用梯度的模长是否为零作为梯度下降的收敛终止条件是否合适，为什么？一般如何判断模型收敛？

总结与对比

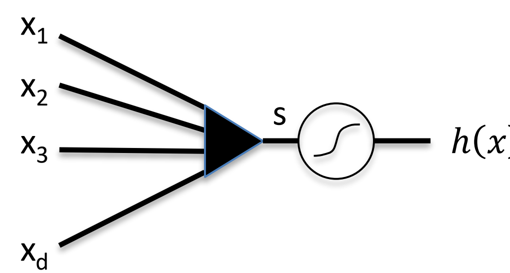
$h(x) = \text{sign}(s)$



$h(x) = s$



$h(x) = \theta(s)$



PLA: 0/1误差

线性回归: 均方误差

逻辑回归: 交叉熵

思考题

- 有什么手段可以使PLA适用于非线性可分的数据集？

- 不同的学习率η 对模型收敛有何影响？从收敛速度和是否收敛两方面来回答。

- 使用梯度的模长是否为零作为梯度下降的收敛终止条件是否合适，为什么？一般如何判断模型收敛？

注意事项

- 实现固定迭代次数的PLA与基于批梯度下降的逻辑回归，分别提交一份代码。
- 本次数据为train.csv，前40列为特征，最后一列是标签（$0\ or\ 1$）。
- 请自行分好训练验证集（在报告里说明怎么分的），评测指标为验证集上的准确率
- 验收时使用的基准模型如下，学习率均设为1：
  - PLA：固定迭代次数，权重初始化为零向量，每次迭代按顺序从第一个样例开始找下一个错误的样例
  - LR：固定迭代次数，权重初始化为零向量，使用批梯度下降优化
- 提交文件
  - 实验报告：17******_wangxiaoming.pdf。
  - 代码：17******_wangxiaoming.zip。如果代码分成多个文件，最好写份readme.txt。
- DDL: **2019年9月26日23：00：00**