面经之凸优化和非凸优化

马东什么
算法工程师

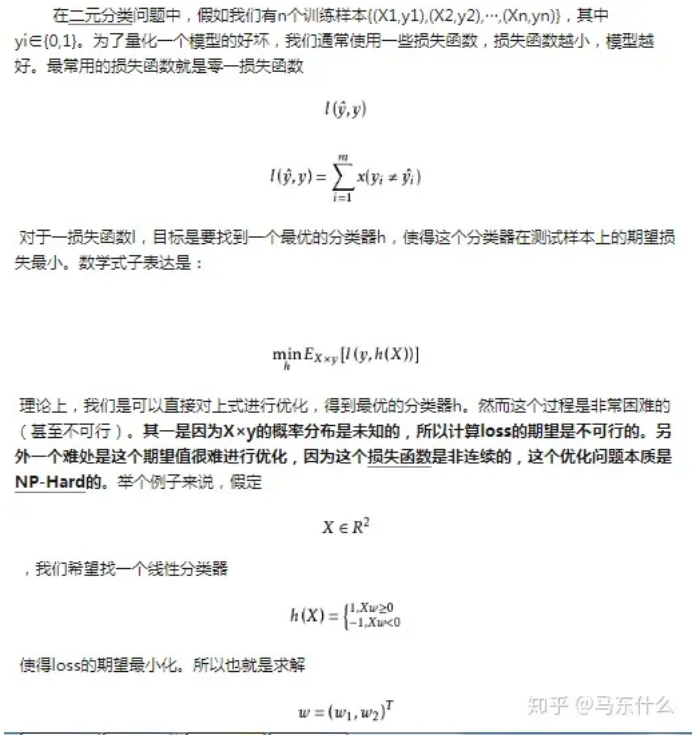
首先还是来大概整理一下凸优化和非凸优化的相关知识，在学习机器学习的过程中貌似经常在不同地方听到这个名词，根据百度百科的定义：

> **凸优化**，或叫做**凸最优化**，**凸最小化**，是数学最优化的一个子领域，研究定义于凸集中的凸函数最小化的问题。凸优化在某种意义上说较一般情形的数学最优化问题要简单，譬如在凸优化中局部最优值必定是全局最优值。凸函数的凸性使得凸分析中的有力工具在最优化问题中得以应用，如次导数等。
> 凸优化应用于很多学科领域，诸如自动控制系统，信号处理，通讯和网络，电子电路设计，数据分析和建模，统计学（最优化设计），以及金融。在近来运算能力提高和最优化理论发展的背景下，一般的凸优化已经接近简单的线性规划一样直捷易行。许多最优化问题都可以转化成凸优化（凸最小化）问题，例如求凹函数*f*最大值的问题就等同于求凸函数 -*f*最小值的问题。

比较重要的地方在于，凸函数的局部最优解必然是全局最优解，并且凸函数的证明方法：

> 判定方法可利用定义法、已知结论法以及函数的二阶导数，对于实数集上的凸函数，一般的判别方法是求它的二阶导数，如果其二阶导数在区间上非负，就称为凸函数。如果其二阶导数在区间上恒大于0，就称为严格凸函数。

自己去对逻辑回归或者线性回归之类的损失函数进行求导可以很容易知道二者的损失函数都是凸函数。那么为什么我们要一直强调凸优化这个东西呢？主要是因为在实际的应用中，我们会碰到很多非凸的问题，这些非凸问题的损失函数很难去求它的最优值，举一个常见的例子，在二分类的过程中，我们最直观的问题的损失函数应该是0-1损失（这部分的具体阐述可见下文第二点）：



在二元分类问题中，假如我们有n个训练样本{(X1,y1),(X2,y2),⋯,(Xn,yn)}，其中yi∈{0,1}。为了量化一个模型的好坏，我们通常使用一些损失函数，损失函数越小，模型越好。最常用的损失函数就是零一损失函数

$$l(\hat{y},y)$$

$$l(\hat{y},y)=\sum_{i=1}^{m}x(y_i\neq\hat{y}_i)$$

对于一损失函数l，目标是要找到一个最优的分类器h，使得这个分类器在测试样本上的期望损失最小。数学式子表达是：

$$\min_h E_{X\times y}[l(y,h(X))]$$

理论上，我们是可以直接对上式进行优化，得到最优的分类器h。然而这个过程是非常困难的（甚至不可行）。**其一是因为X×y的概率分布是未知的，所以计算loss的期望是不可行的。另外一个难处是这个期望值很难进行优化，因为这个损失函数是非连续的，这个优化问题本质是NP-Hard的。**举个例子来说，假定

$$X\in R^2$$

，我们希望找一个线性分类器

$$h(X)=\begin{cases}1, Xw\geq 0\\ -1, Xw<0\end{cases}$$

使得loss的期望最小化。所以也就是求解

$$w=(w_1,w_2)^T$$

知乎 @马东什么

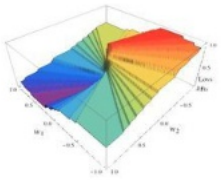
，我们希望找一个线性分类器

$$h(X)=\begin{cases}1, Xw\geq 0\\ -1, Xw<0\end{cases}$$

使得loss的期望最小化。所以也就是求解

$$w=(w_1,w_2)^T$$

关于 w1，w2以及loss的图像大致如下：

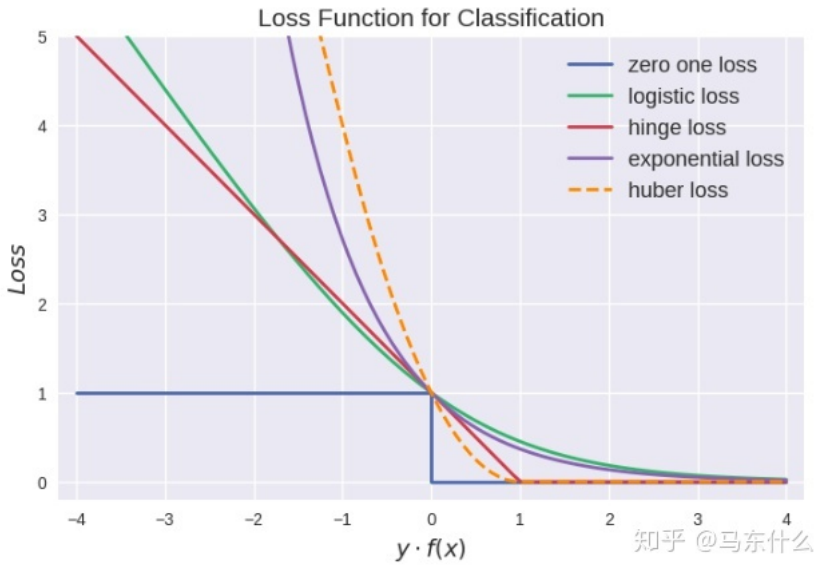


w1,w2与loss的关系图

这个函数显然是非连续的。常用的优化方法，比如梯度下降，对此都失效了。**正因此，可以考虑一个与零一损失相接近的函数，作为零一损失的替身。这个替身就称作代理损失函数（surrogate loss function）。**

然而0-1损失函数不便于求解，当然如果要硬算也是能算的，比如逻辑回归的权重w，你完全可以用暴力的网格搜索来不断搜索不同权重的备选空间，只要算力和时间足够，等到宇宙毁灭那天，你一定能算出来，或者是一些启发式的优化方法，比如随机搜索、模拟退火、贝叶斯优化、遗传算法等等，如果去深入研究这些优化方法可以发现，他们有一个共同的问题就是都不太稳定，可能每一次得到的结果都不尽相同，比如遗传算法涉及到交叉和变异，实际上存在着比较强的随机性，很难保证能得到最优解。但是如果能够把非凸函数转化为凸函数，然后使用梯度下降法这类最优化的方法来求解，就能在“凸函数的局部最优解就是全局最优解”的背景下得到最优结果。（简单说就是凸优化问题有严谨的数据理论证明作为支持，用起来比较放心）。

换一个更加直观的角度来说，我们平常在对于分类问题的评价指标是准确率，但是准确率本身无法直接进行最优解的求解，所以退而求其次，我们选择了logit loss作为代理损失函数（代理损失函数或者称为替代损失函数，一般是指当目标函数非凸、不连续时，数学性质不好，优化起来比较复杂，这时候需要使用其他的性能较好的函数进行替换。这是应用过程中很常见的一种做法，将非凸问题转化为凸问题便于直接求解），logit loss可以直接使用凸优化的一套理论到工具的组合拳进行求解，并且**如果最优化代理损失函数的同时也最优化了原本的损失函数，就称校对性(calibration)或者一致性(consistency)。这个性质与所选择的代理损失函数相关。一个重要的定理是，如果代理损失函数是凸函数，并且在0点可导，其导数小于0，那么它一定是具有一致性的。**



祭出这张经典的图

至于这些损失函数是如何找到的以及是如何证明他们具有一致性的，这个不讨论了，太麻烦太难了，对我来说暂时不是那么重要了。

数，f和g代表了两层的隐层带来的效果，当f，g都是线性的时候，h是线性的。但在深度学习里用到的函数，Logistic, ReLU等等，都是非线性，并且非常多。把他们复合起来形成的函数h,便是非凸的。求解这个非凸函数的最优解，类似于求凸优化中某点的gradient，然后按照梯度最陡的方向搜索。不同的是，**复合函数无法求gradient，于是这里使用Back Propagation求解一个类似梯度的东西**，通过这种间接的方式进行更新。所以很多时候神经网络的设计不合适比较容易卡在不同的局部最优，并且每次局部最优的结果都可能差别不小，针对这类问题，有很多的研究，不过我也没怎么看过，这里就不废话了。

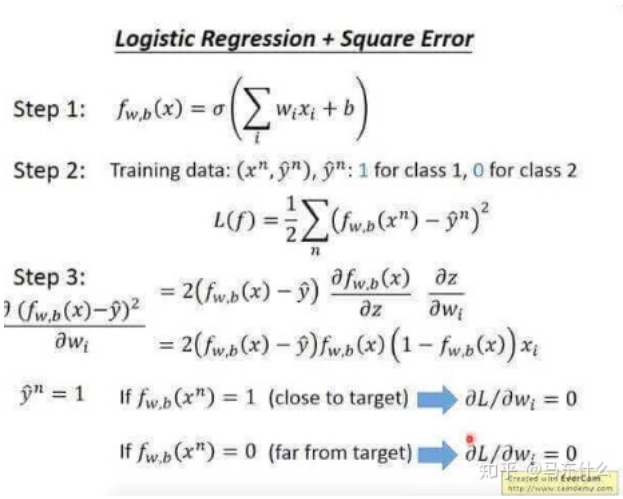
不得不说，这些题是真的毒。确实很考察功底，shit。

为什么逻辑回归不用mse,为什么分类就不能有mse?

这个问题深入研究的话还挺复杂，还好有大佬已经证明了：

角度1—梯度角度解释：

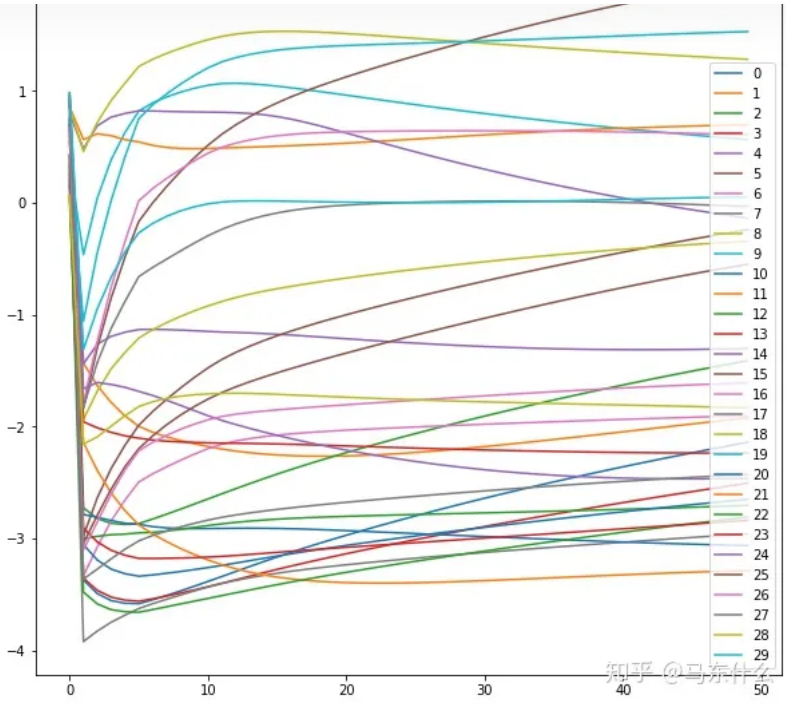
二元分类为什么不能用MSE做为损失函数？-SofaSofa
sofasofa.io/forum_main_post.php?postid…



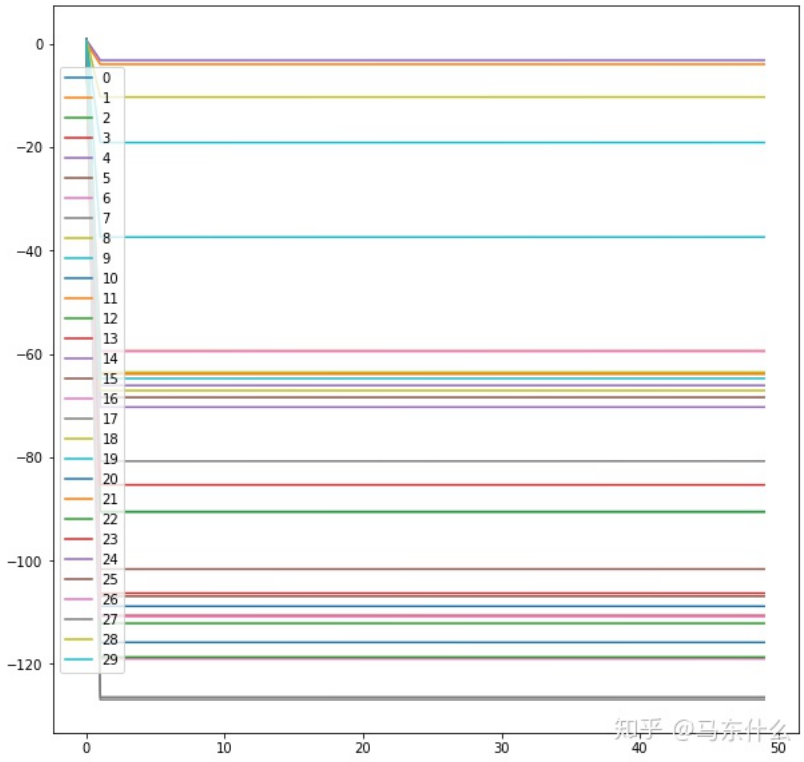
如果我们使用mse的话，经过推导之后梯度的更新公式为：

$$\cdots 2\left(f_{w,b}(x)-\hat{y}\right)f_{w,b}(x)\left(1-f_{w,b}(x)\right)x_i$$

其中fw，b代表的是我们的训练模型，根据上面的梯度更新公式可知，当预测值趋于0或者趋于1，也就是预测越来越准的情况下，梯度会趋于消失，即梯度消失问题，这样模型很难收敛。下面举个列子详细说明：



这个是损失函数为log损失并且使用bgd来求解的时候，权重系数的变化过程，迭代50轮，可以看到大部分权重系数是在不断更新的，少部分贡献度很低的权重系数几乎不动。然后我们使用mse作为损失函数来看看：



很明显的梯度消失，模型经过少量的迭代之后就梯度消失了从而没办法继续更新权重系数了

角度二——凸or不凸：

这里涉及到代理损失函数的概念，一旦理解了代理损失函数的概念，去理解贝叶斯优化这类调参方式就不难了，贝叶斯优化本质上也是把调参这么一个无法用常规的最优化方法求解的问题转化为用代理损失函数去近似替代，搜了一下，百度百科居然也有解释，直接搬运：

中文名	代理损失函数	外文名	surrogate loss function

定义

在二元分类问题中，假如我们有n个训练样本{(X1,y1),(X2,y2),⋯,(Xn,yn)}，其中yi∈{0,1}。为了量化一个模型的好坏，我们通常使用一些损失函数，损失函数越小，模型越好。最常用的损失函数就是零一损失函数

$$l(\hat{y}, y)$$

$$l(\hat{y}, y) = \sum_{i=1}^{m} x(y_i \neq \hat{y}_i)$$

对于一损失函数l，目标是要找到一个最优的分类器h，使得这个分类器在测试样本上的期望损失最小。数学式子表达是：

$$\min_h E_{X \times y}[l(y, h(X))]$$

理论上，我们是可以直接对上式进行优化，得到最优的分类器h。然而这个过程是非常困难的（甚至不可行）。**其一是因为X×y的概率分布是未知的，所以计算loss的期望是不可行的。另外一个难处是这个期望值很难进行优化，因为这个损失函数是非连续的，这个优化问题本质是NP-Hard的**。举个例子来说，假定

$$X \in R^2$$

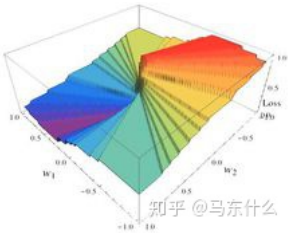
，我们希望找一个线性分类器

$$h(X) = \begin{cases} 1, Xw \geq 0 \\ -1, Xw < 0 \end{cases}$$

使得loss的期望最小化。所以也就是求解

$$w = (w_1, w_2)^T$$

关于 w1，w2以及loss的图像大致如下：

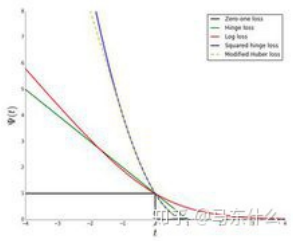


w1,w2与loss的关系图

这个函数显然是非连续的。常用的优化方法，比如梯度下降，对此都失效了。**正因此，可以考虑一个与零一损失相接近的函数，作为零一损失的替身。这个替身就称作代理损失函数（surrogate loss function）**。

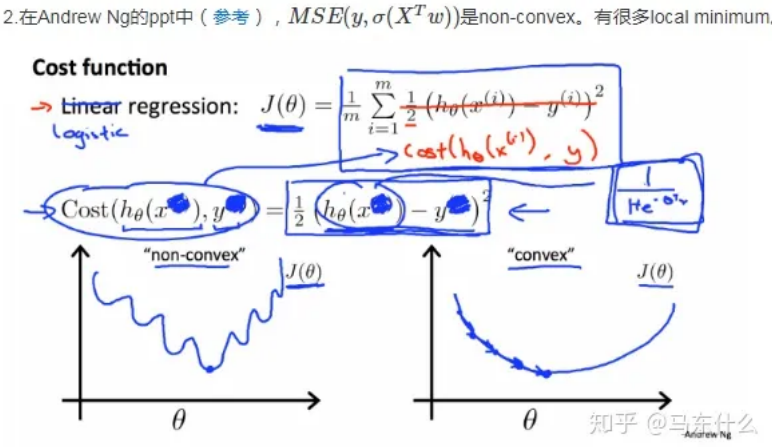
性质

如果最优化代理损失函数的同时也最优化了原本的损失函数，就称校对性(calibration)或者一致性(consistency)。这个性质与所选择的代理损失函数相关。一个重要的定理是，如果代理损失函数是凸函数，并且在0点可导，其导数小于0，那么它一定是具有一致性的（有了这条基本的性质我们就可以放心的优化与原始的损失函数具有一致性的损失函数了，这也解释了为什么我们使用目前的这些损失函数来解决问题）。



零一损失函数与logloss，hinge loss等的联系

那么，根据吴恩达大佬的ppt，当我们使用mse作为逻辑回归的损失函数的时候推导出来的损失函数是一个非凸函数，我们无法使用梯度下降法这类凸优化的方法来求解一个非凸函数的最优值（至于怎么证明是非凸的百度搜索即可，我也比较少去亲手推导证明非凸函数的问题），如果一定要使用凸优化的方法去解决非凸函数优化问题则会很容易陷入局部最优值。



编辑于 2019-07-12 16:26

机器学习　算法工程师