机器学习中的常用损失函数

原创 stephenDC 大数据与人工智能 2019-08-28

点击上方“**大数据与人工智能**”，“星标或置顶公众号”

第一时间获取好内容

作者 | stephenDC

编辑 | zandy

这是作者的第16篇文章

导语

损失函数虽然简单，却相当基础，可以看做是机器学习的一个组件。机器学习的其他组件，还包括激活函数、优化器、模型等。

本文针对机器学习中的回归和分类问题，分别介绍了一些常用的损失函数。除了损失函数的表达式、文中还给出了损失函数的梯度，并对各种损失函数的特性进行了介绍。

回归问题的损失函数

对回归问题而言，衡量模型预测的准确程度，靠的是考察模型预测值与样本实际值之间的差值。因此，回归问题的损失函数应满足两个基本条件，一、应是$|\hat{y}-y|$的函数，二、整体上关于$|\hat{y}-y|$单调递增（其中，y是样本的label，$\hat{y}$代表该样本的预测值）。满足要求且常用的函数如下：

L1 Loss

$$L(\hat{y}, y)=|\hat{y}-y|$$

Gradient

$$\frac{\partial L}{\partial \hat{y}}=\begin{cases}1 & \hat{y}>y \\ -1 & \hat{y}<y\end{cases}$$

Challenge

L1 损失函数最大的问题是其梯度在零点处不连续，这会给基于梯度下降算法的优化算法带来不稳定性。为了应对这个问题，有了以下的 L2 损失函数。

L2 Loss

$$L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$$

Gradient

$$\frac{\partial L}{\partial \hat{y}} = \hat{y} - y$$

Challenge

L2 损失函数通过对$|\hat{y} - y|$取平方，解决了 L1 损失函数在零点处梯度不连续的问题，但是也带来了另外一个问题，就是对异常点不够 robust。对异常点而言，$|\hat{y} - y|$一般会比较大，而因为 L2 损失函数引入的平方的作用，异常点的损失会被进一步放大。因为异常点是一种反常的行为，所以我们并不想让模型去过度学习这种行为。为了应对这一挑战，有了以下的 Huber 损失函数。

Huber Loss

$$L(\hat{y}, y) = \begin{cases} \frac{1}{2}(\hat{y} - y)^2 & |\hat{y} - y| \le \sigma \\ \sigma|\hat{y} - y| - \frac{1}{2}\sigma^2 & |\hat{y} - y| > \sigma \end{cases}$$

Gradient

$$\frac{\partial L}{\partial \hat{y}} = \begin{cases} \hat{y} - y & |\hat{y} - y| \le \sigma \\ +\sigma & \hat{y} - y > \sigma \\ -\sigma & \hat{y} - y < \sigma \end{cases}$$

Challenge

Huber 损失函数几乎完美了有木有，但是有人提出，想要得到一种基于训练集中的少数样本进行预测的模型。没错，这就是 Vapnik 等人提出的 SVM。对分类问题，下文提到的 Hinge Loss 会导出 SVM；而对回归问题，导出 SVM 的是如下的 ε -Insensitive Loss。

ε-Insensitive Loss

$$L(\hat{y}, y) = \begin{cases} 0 & |\hat{y} - y| \le \varepsilon \\ |\hat{y} - y| - \varepsilon & |\hat{y} - y| > \varepsilon \end{cases}$$

Gradient

$$\frac{\partial L}{\partial \hat{y}} = \begin{cases} 0 & |\hat{y} - y| \le \varepsilon \\ +1 & \hat{y} - y > \varepsilon \\ -1 & \hat{y} - y < \varepsilon \end{cases}$$

Challenge

显然，ε-Insensitive Loss 对于那些预测值和实际值差别很小的样本（$|\hat{y} - y| \le \varepsilon$），直接忽略掉了。这样做的结果是，整个样本训练出的模型，在进行预测的时候只有很少部分的样本起作用，这样既增加了模型的鲁棒性，也加快了模型预测时的计算。By the way，这少部分在预测中起作用的样本，被称为支持向量。

分类问题的损失函数

对分类问题，其损失函数取决于样本 label 的编码方式。以二分类为例，我们可以将正类和负类分别编码为{1，0}，也可以分别编码为{1，-1}。

如果样本编码为{1，0}，我们可以设法使模型预测的结果 $\hat{y}$ 介于 0 和 1 之间。这样就可以理解成一种概率，如果 $\hat{y}$ 越大，对应样本属于 1 这一类的概率越大；反之，$\hat{y}$ 越小，对应样本属于 0 这一类的概率越大。既然预测的是概率，当然可以用极大似然的框架求解模型参数，这时候的损失函数就是负的对数似然。

如果样本编码为{1，-1}，则通过模型预测的结果$\hat{y}$的正负进行分类。$\hat{y}>0$，分到+1 对应的类，$\hat{y}$越大确信度越高；$\hat{y}<0$，分到-1 对应的类，$\hat{y}$越小确信度越高。在这种情况下，$-\hat{y}\cdot y$表明了误分类的程度，损失函数自然应当是$-\hat{y}\cdot y$的函数，而且整体上应是$-\hat{y}\cdot y$的单调递增函数。

在下面列举的损失函数中，我们不再一一说明对应的类别编码方式。请大家自行对应。

Cross Entropy Loss

$$L(\hat{y}, y) = -y\log(\hat{y})$$

Gradient

$$\frac{\partial L}{\partial \hat{y}} = -\frac{y}{\hat{y}}$$

Note

1、如果两个类分别编码为{0,1}，$y=1$可以看做样本属于 1 类的概率为 1，属于 0 类的概率为 0，这是样本的真实概率分布。$\hat{y}$则代表了模型预测的样本属于 1 类的概率。交叉熵表征了这两个概率分布之间的接近程度。

Soft-Max Loss

$$L(\hat{y}, y) = -\log(\hat{y}_k)$$

其中，y 和 $\hat{y}$ 都是向量，$\hat{y}_k$ 是 $\hat{y}$ 的第 k 个分量（y 的第 k 个分量为 1，其余均为 0）。

Gradient

$$\frac{\partial L}{\partial \hat{y}_i} = \begin{cases} 0 & i \neq k \\ -\dfrac{1}{\hat{y}_i} & i = k \end{cases}$$

Note

1、对比可以发现，Soft-Max 损失和上面提到的 Cross Entropy 损失是一回事。Soft-Max 损失可以看做 Cross Entropy 在多分类问题上扩展。

Log Loss

$$L(\hat{y}, y) = \log(1 + \exp(-\hat{y} \cdot y))$$

Gradient

$$\frac{\partial L}{\partial \hat{y}} = \frac{-y\exp(-\hat{y} \cdot y)}{1 + \exp(-\hat{y} \cdot y)}$$

Note

1、当$-\hat{y} \cdot y > 0$，且值很大时，指数运算$\exp(-\hat{y} \cdot y)$在计算时有溢出的风险，可进行截断。

2、Log Loss 的梯度计算公式里，分子分母都有指数项，$-\hat{y} \cdot y > 0$时可以对分子分母同时放缩，然后计算（Soft-Max 也适用此技巧）。

Hinge Loss

$$L(\hat{y}, y) = \begin{cases} 0 & \hat{y} \cdot y > 1 \\ 1 - \hat{y} \cdot y & \hat{y} \cdot y \le 1 \end{cases}$$

Gradient

$$\frac{\partial L}{\partial \hat{y}} = \begin{cases} 0 & \hat{y} \cdot y > 1 \\ -y & \hat{y} \cdot y \le 1 \end{cases}$$

Note

1、对 Hinge 损失来说，如果样本被正确分类，且距离分类边界的距离超过了 margin，则损失记为 0。因此，Hinge 损失函数最小化的目标，是让样本尽量都被正确分类，且距离分类边界足够远。所以，Hinge 损失导出了分类的 SVM。

2、第 1 点的更具体内容，可参看《稀疏核机（上）—SVM 回顾》。

Exponential Loss

$$L(\hat{y}, y) = \exp(-\hat{y} \cdot y)$$

Gradient

$$\frac{\partial L}{\partial \hat{y}} = -y \exp(-\hat{y} \cdot y)$$

Note

1、指数损失函数加上前向分步算法，可以导出大名鼎鼎的 AdaBoost。

2、前面说过类别编码为{+1，-1}时，$-\hat{y} \cdot y$ 表明了样本误分类的程度。而指数函数又对这一程度进行了缩放，因此 AdaBoost 在训练中将重点放在了上一次被误分类的样本上。

3、指数函数比平方函数增长更快，因此指数损失相比与 L2 损失，对异常点更加的不 robust。

小结

本文对机器学习中的常用损失函数进行了梳理总结，有不尽及错误之处，恳请各位不吝指出，感激不尽。

-end-

相关内容阅读

1.特征工程（上）—特征选择

2.特征工程（中）-特征表达