

交叉熵、相对熵（KL散度）、JS散度和Wasserstein距离（推土机距离）



KevinCK

百度 计算机视觉工程师

1,241 人赞同了该文章

目录：

- 信息量
- 熵
- 相对熵（KL散度）
- 交叉熵
- JS散度
- 推土机理论
- Wasserstein距离
- WGAN中对JS散度，KL散度和推土机距离的描述

写在前面的总结：

1、目前分类损失函数为何多用交叉熵，而不是KL散度。

首先损失函数的功能是通过样本来计算模型分布与目标分布间的差异，在分布差异计算中，KL散度是最合适的。但在实际中，某一事件的标签是已知不变的（例如我们设置猫的label为1，那么所有关于猫的样本都要标记为1），即目标分布的熵为常数。而根据下面KL公式可以看到， $KL散度 - 目标分布熵 = 交叉熵$ （这里的“-”表示裁剪）。所以我们不用计算KL散度，只需要计算交叉熵就可以得到模型分布与目标分布的损失值。

从上面介绍，知道了模型分布与目标分布差异可用交叉熵代替KL散度的条件是目标分布为常数。如果目标分布是有变化的（如同为猫的样本，不同的样本，其值也会有差异），那么就不能使用交叉熵，例如蒸馏模型的损失函数就是KL散度，因为蒸馏模型的目标分布也是一个模型，该模型针对同类别的不同样本，会给出不同的预测值（如两张猫的图片a和b，目标模型对a预测为猫的值是0.6，对b预测为猫的值是0.8）。

注：交叉熵和KL散度应用方式不同的另一种解释（我更倾向于上面我自己的解释，更具公式解释性）：

交叉熵：其用来衡量在给定的真实分布下，使用非真实分布所指定的策略消除系统的不确定性所需要付出的努力的大小。这也是为什么在机器学习中的分类算法中，我们总是最小化交叉熵，因为交叉熵越低，就证明由算法所产生的策略最接近最优策略，也间接证明我们算法所算出的非真实分布越接近真实分布。

KL散度（相对熵）：衡量不同策略之间的差异呢，所以我们使用KL散度来做模型分布的拟合损失。

开始详细内容：

信息量：



任何事件都会承载着一定的信息量，包括已经发生的事件和未发生的事件，只是它们承载的信息量会有所不同。如昨天下雨这个已知事件，因为已经发生，既定事实，那么它的信息量就为0。如明天会下雨这个事件，因为未有发生，那么这个事件的信息量就大。

从上面例子可以看出信息量是一个与事件发生概率相关的概念，而且可以得出，事件发生的概率越小，其信息量越大。这也很好理解，狗咬人不算信息，人咬狗才叫信息嘛。

我们已知某个事件的信息量是与它发生的概率有关，那我们可以通过如下公式计算信息量：

假设 X 是一个离散型随机变量，其取值集合为 \mathcal{X} ，概率分布函数 $p(x) = \Pr(X = x), x \in \mathcal{X}$ ，则定义事件 $X = x_0$ 的信息量为： $I(x_0) = -\log(p(x_0))$

熵：

我们知道：当一个事件发生的概率为 $p(x)$ ，那么它的信息量是 $-\log(p(x))$ 。

那么如果我们将这个事件的所有可能性罗列出来，就可以求得该事件信息量的期望，

信息量的期望就是熵，所以熵的公式为：

假设事件 X 共有 n 种可能，发生 x_i 的概率为 $p(x_i)$ ，那么该事件的熵 $H(X)$ 为：

$$H(X) = -\sum_{i=1}^n p(x_i) \log(p(x_i))$$

然而有一类比较特殊的问题，比如投掷硬币只有两种可能，字朝上或花朝上。买彩票只有两种可能，中奖或不中奖。我们称之为0-1分布问题（二项分布的特例），对于这类问题，熵的计算方法可以简化为如下算式：

$$H(X) = -\sum_{i=1}^n p(x_i) \log(p(x_i)) = -p(x) \log(p(x)) - (1 - p(x)) \log(1 - p(x))$$

相对熵（KL散度）：

相对熵又称KL散度，如果我们对于同一个随机变量 x 有两个单独的概率分布 $P(x)$ 和 $Q(x)$ ，我们可以使用 KL 散度（Kullback-Leibler (KL) divergence）来衡量这两个分布的差异。

在机器学习中， P 往往用来表示样本的真实分布， Q 用来表示模型所预测的分布，那么KL散度就可以计算两个分布的差异，也就是Loss损失值。

$$D_{KL}(p||q) = \sum_{i=1}^n p(x_i) \log\left(\frac{p(x_i)}{q(x_i)}\right)$$

从KL散度公式中可以看到 Q 的分布越接近 P （ Q 分布越拟合 P ），那么散度值越小，即损失值越小。

因为对数函数是凸函数，所以KL散度的值为非负数。

有时会将KL散度称为KL距离，但它并不满足距离的性质：

1. KL散度不是对称的；
2. KL散度不满足三角不等式。

我们将KL散度公式进行变形：

$$D_{KL}(p||q) = \sum_{i=1}^n p(x_i) \log\left(\frac{p(x_i)}{q(x_i)}\right) = \sum_{i=1}^n p(x_i) \log(p(x_i)) - \sum_{i=1}^n p(x_i) \log(q(x_i)) = -H(p(x)) + [-\sum_{i=1}^n p(x_i) \log(q(x_i))]$$

等式的前一部分恰巧就是p的熵，等式的后一部分，就是交叉熵：

$$H(p, q) = -\sum_{i=1}^n p(x_i) \log(q(x_i))$$

在机器学习中，我们需要评估label和predicts之间的差距，使用KL散度刚刚好，即 $D_{KL}(y||\hat{y})$ ，由于KL散度中的前一部分 $-H(y)$ 不变，故在优化过程中，只需要关注交叉熵就可以了。所以一般在机器学习中直接用交叉熵做loss，评估模型。

JS散度：

JS散度度量了两个概率分布的相似度，基于KL散度的变体，解决了KL散度非对称的问题。一般地，JS散度是对称的，其取值是0到1之间。定义如下：

$$JS(P_1||P_2) = \frac{1}{2} KL(P_1||\frac{P_1+P_2}{2}) + \frac{1}{2} KL(P_2||\frac{P_1+P_2}{2})$$

Wasserstein距离（该部分摘自KL散度、JS散度、Wasserstein距离）：

KL散度和JS散度度量的问题：

如果两个分配P,Q离得很远，完全没有重叠的时候，那么KL散度值是没有意义的，而JS散度值是一个常数。这在学习算法中是比较致命的，这就意味这这一点的梯度为0。梯度消失了。

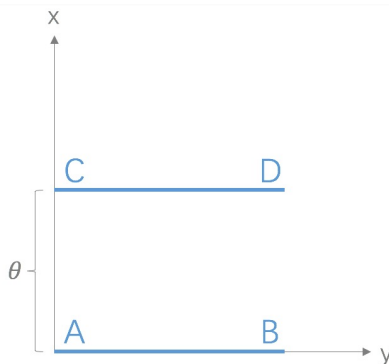
Wasserstein距离度量两个概率分布之间的距离，定义如下

$$W(P_1, P_2) = \inf_{\gamma \in \Pi(P_1, P_2)} \mathbb{E}_{(x,y) \sim \gamma} [|x - y|]$$

$\Pi(P_1, P_2)$ 是P1和P2分布组合起来的所有可能的联合分布的集合。对于每一个可能的联合分布 γ ，可以从中采样 $(x,y) \sim \gamma$ 得到一个样本x和y，并计算出这对样本的距离 $|x-y|$ ，所以可以计算该联合分布 γ 下，样本对距离的期望值 $\mathbb{E}_{(x,y) \sim \gamma} [|x-y|]$ 。在所有可能的联合分布中能够对这个期望值取到的下界 $\inf_{\gamma \in \Pi(P_1, P_2)} \mathbb{E}_{(x,y) \sim \gamma} [|x-y|]$ 就是Wasserstein距离。

直观上可以把 $\mathbb{E}_{(x,y) \sim \gamma} [|x-y|]$ 理解为在 γ 这个路径规划下把土堆P1挪到土堆P2所需要的消耗。而Wasserstein距离就是在最优路径规划下的最小消耗。所以Wasserstein距离又叫Earth-Mover距离。

Wasserstein距离相比KL散度、JS散度的优越性在于，即便两个分布没有重叠，Wasserstein距离仍然能够反映它们的远近；而JS散度在此情况下是常量，KL散度可能无意义。WGAN本作通过简单的例子展示了这一点。考虑如下二维空间中的两个分布 P_1 和 P_2 ， P_1 在线段AB上均匀分布， P_2 在线段CD上均匀分布，通过控制参数 θ 可以控制着两个分布的距离远近。



此时容易得到（读者可自行验证）

$$KL(P_1||P_2) = KL(P_1||P_2) = \begin{cases} +\infty & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases} \quad (\text{突变})$$

$$JS(P_1||P_2) = \begin{cases} \log 2 & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases} \quad (\text{突变})$$

$$W(P_0, P_1) = |\theta| \quad (\text{平滑})$$

KL散度和JS散度是突变的，要么最大要么最小，Wasserstein距离却是平滑的，如果我们要用梯度下降法优化 θ 这个参数，前两者根本提供不了梯度，Wasserstein距离却可以。类似地，在高维空间中如果两个分布不重叠或者重叠部分可忽略，则KL和JS既反映不了远近，也提供不了梯度，但是Wasserstein却可以提供有意义的梯度。

WGAN中对KL散度和JS散度的描述（摘自：郑华滨：令人拍案叫绝的Wasserstein GAN）

假设 P_r 表示真实样本分布， P_g 是由生成器产生的样本分布。原始GAN中：

判别器损失函数：

$$-\mathbb{E}_{x \sim P_r} [\log D(x)] - \mathbb{E}_{x \sim P_g} [\log(1 - D(x))] \quad (\text{公式1})$$

生成器损失函数：

$$\mathbb{E}_{x \sim P_g} [\log(1 - D(x))] \quad (\text{公式2})$$

$$\mathbb{E}_{x \sim P_g} [-\log D(x)] \quad (\text{公式3})$$

最优判别器：

首先根据公式1，当生成器固定时，确定最优的判别器。对判别器进行求导，并令导数为0，则：

$$-\frac{P_r(x)}{D(x)} + \frac{P_g(x)}{1 - D(x)} = 0 \Rightarrow (1 - D(x))P_r(x) = D(x)P_g(x) \Rightarrow D(x)(P_g(x) + P_r(x)) = P_r(x)$$

化简得：

$$D^*(x) = \frac{P_r(x)}{P_r(x) + P_g(x)} \quad (\text{公式4})$$

从公式4也可以很容易看出最优判别器的特征：当 $P_r(x) = 0$ 且 $P_g(x) \neq 0$ ，最优判别器可以给出概



生成器损失：

普通的GAN在训练时会有一明显问题，即如果判别器训练的太好，生成器就会完全学不动；那么当判别器为最优时，我们可以通过公式推导生成器的损失函数是怎样。

生成器损失函数（1）

$$\mathbb{E}_{x \sim P_g} [\log(1 - D(x))] \quad (\text{公式2})$$

首先给公式2添加一个不依赖生成器的项（真实分布损失： $\mathbb{E}_{x \sim P_r} [\log D(x)]$ ）：

$$\mathbb{E}_{x \sim P_r} [\log D(x)] + \mathbb{E}_{x \sim P_g} [\log(1 - D(x))]$$

添加该项后，上式变成了公式1的反，即最小化生成器损失变为了最大化判别器损失。代入最优判别器即公式4，再进行简单的变换可以得到：

$$\mathbb{E}_{x \sim P_r} \log \frac{P_r(x)}{\frac{1}{2}[P_r(x) + P_g(x)]} + \mathbb{E}_{x \sim P_g} \log \frac{P_g(x)}{\frac{1}{2}[P_r(x) + P_g(x)]} - 2 \log 2 \quad (\text{公式5})$$

根据JS散度的公式：

$$JS(P_1 || P_2) = \frac{1}{2} KL(P_1 || \frac{P_1 + P_2}{2}) + \frac{1}{2} KL(P_2 || \frac{P_1 + P_2}{2}) \quad (\text{公式6})$$

于是公式5就可以继续写成：

$$2JS(P_r || P_g) - 2 \log 2 \quad (\text{公式7})$$

公式7即为生产器损失函数1在判别器最优条件下的值。

根据原始GAN定义的判别器loss，我们可以得到最优判别器的形式；而在最优判别器的下，我们可以把原始GAN定义的生成器loss等价变换为最小化真实分布 P_r 与生成分布 P_g 之间的JS散度。我们越训练判别器，它就越接近最优，最小化生成器的loss也就会越近似于最小化 P_r 和 P_g 之间的JS散度。

问题就出在这个JS散度上。我们会希望如果两个分布之间越接近它们的JS散度越小，我们通过优化JS散度就能将 P_g “拉向” P_r ，最终以假乱真。这个希望在两个分布有所重叠的时候是成立的，但是如果两个分布完全没有重叠的部分，或者它们重叠的部分可忽略，那它们的JS散度就变成了 $\log 2$ 。

换句话说，无论 P_r 跟 P_g 是远在天边，还是近在眼前，只要它们俩没有一点重叠或者重叠部分可忽略，JS散度就固定是常数 $\log 2$ ，而这对于梯度下降方法意味着——梯度为0！此时对于最优判别器来说，生成器肯定是得不到一丁点梯度信息的；即使对于接近最优的判别器来说，生成器也有很大机会面临梯度消失的问题。

生成器损失函数（2）

$$\mathbb{E}_{x \sim P_g} [-\log D(x)] \quad (\text{公式3})$$

上文推导已经得到在最优判别器 D^* 下

$$\mathbb{E}_{x \sim P_r} [\log D^*(x)] + \mathbb{E}_{x \sim P_g} [\log(1 - D^*(x))] = 2JS(P_r || P_g) - 2 \log 2 \quad (\text{公式9})$$

我们可以把KL散度（注意下面是先g后r）变换成含 D^* 的形式：



$$\begin{aligned}
KL(P_g||P_r) &= \mathbb{E}_{x \sim P_g} [\log \frac{P_g(x)}{P_r(x)}] \\
&= \mathbb{E}_{x \sim P_g} [\log \frac{P_g(x)/(P_r(x) + P_g(x))}{P_r(x)/(P_r(x) + P_g(x))}] \\
&= \mathbb{E}_{x \sim P_g} [\log \frac{1 - D^*(x)}{D^*(x)}] \\
&= \mathbb{E}_{x \sim P_g} \log[1 - D^*(x)] - \mathbb{E}_{x \sim P_g} \log D^*(x)
\end{aligned}
\tag{公式10}$$

由公式3, 9, 10可得最小化目标的等价变形

$$\begin{aligned}
\mathbb{E}_{x \sim P_g} [-\log D^*(x)] &= KL(P_g||P_r) - \mathbb{E}_{x \sim P_g} \log[1 - D^*(x)] \\
&= KL(P_g||P_r) - 2JS(P_r||P_g) + 2\log 2 + \mathbb{E}_{x \sim P_r} [\log D^*(x)]
\end{aligned}$$

注意上式最后两项不依赖于生成器G, 最终得到最小化公式3等价于最小化

$$KL(P_g||P_r) - 2JS(P_r||P_g) \tag{公式11}$$

这个等价最小化目标存在两个严重的问题。第一是它同时要最小化生成分布与真实分布的KL散度, 却又要最大化两者的JS散度, 一个要拉近, 一个却要推远! 这在直观上非常荒谬, 在数值上则会导致梯度不稳定, 这是后面那个JS散度项的毛病。

(未完待续.....)

参考:

blog.csdn.net/tsyccnh/a...

豆浆机: [论文笔记] 损失函数整理

blog.csdn.net/weixin_33...

blog.csdn.net/zhangping...

郑华滨: 令人拍案叫绝的Wasserstein GAN

KL散度、JS散度、Wasserstein距离

如何通俗的解释交叉熵与相对熵?

编辑于 2019-07-18

机器学习 损失函数 深度学习 (Deep Learning)

文章被以下专栏收录

▲ 赞同 1241 ▼ 21 条评论 分享 喜欢 收藏 ...

知乎

推荐阅读



简单的交叉熵，你真的懂了吗？

蔡杰 发表于AI部落

信息量、信息熵、交叉熵、KL散度、JS散度、Wasserstein...

前两篇介绍了目标检测中的回归损失函数，本来这篇打算介绍目标检测中的分类损失函数。但是介绍 classification loss function 自然绕不过交叉熵，所以还是简单的把信息论中的一些概念在这里普...

陈伟

机器学习理论—损失函数（一）：交叉熵与KL散度

1. Introduction 信息论其实给我一种很厉害，但又很玄妙的感觉。一个大佬希望定义信息这个概念，然后先写了写他觉得如果要定义信息，那么它应该有哪些数学性质...

苗思奇 发表于机器学习理...

Wasserstein距

本文参考 Lilian V 《From GAN to \GAN to WGAN主的Wasserstein距记与大家分享。机器学习问题，尤...

炸带鱼

21 条评论 切换为时间排序

写下你的评论...

Jian wang

2019-07-24

写的很好，受教了。问个问题，如果两个分配P,Q离得很远，完全没有重叠的时候，那么KL散度值是没有意义的，这句话应该怎么理解。

赞

Jack Stark 回复 Jian wang

2019-07-25

我们的目的是通过最小化损失函数来最小化两个分布的距离，由于GAN中真实分布P和生成器定义的分布Q是低维空间的低维流形，即完全没有重叠或重叠可忽略不计的情况，这个时候生成器的分布变化后，两者的KL散度都没有变化（等于0），损失函数不变的话就没有梯度了，没梯度模型自然学不动了😂。所以说这种情况下KL散度没有意义。

赞 10

Jian wang 回复 Jack Stark

2019-07-25

了解了，谢了

赞

展开其他 1 条回复

圈圈虫

2019-07-27


总结得很好，非常感谢

赞


Taiyue Chen

2019-09-07

不用kl原因，我理解为，通常一个标签都是设置为one hot模式，即我们常说的硬分布，log1=0所以，一般都是只用第二项，也就是交叉熵。

candywisdom 回复 Taiyue Chen2019-10-12

就算不是硬分布，标签固定的情况下，前面那项也是常数，不影响训练结果

 4


知乎用户2019-09-14


请教 在distillation里用KL和cross entropy有啥区别吗

 1


candywisdom2019-10-12

感觉按照这个公式求wass距离计算量有点大啊，需要排列组合。

 赞


知乎用户 回复 candywisdom2019-10-12


w距离用对偶形式转成另一个形式（带Lipschitz约束），然后用采样的方式得到一个估计。至于怎么加Lipschitz约束，就有原文章提的gradient clip，后续提出的gradient penalty等方法

 2


lyn wes2020-02-14

AB、CD上的KL散度跟JS散度计算应该按照什么思路呢，\theta不等于0时

 赞

我没看到你看到我2020-03-14

请问为啥kl散度公式左边两个p之间是双竖线分隔，交叉熵之间是逗号分隔？这个是有怎样的含义啊？

 1

我是个好人 回复 我没看到你看到我2020-12-22


同求

 赞


阿灿啊2020-06-13


写得非常好👍

 赞


wert2020-06-20


文中kl散度非负说的有点小问题～
负对数函数是凸函数，根据凸性，函数值期望大于等于期望的函数值，导出KL散度非负～

 9


SkippingStone2020-10-04


请问两个分配没有重叠是指？（如果两个分配P,Q离得很远，完全没有重叠的时候）

 赞


dou无知无畏 回复 SkippingStone2020-10-13

请问您理解这个的意思了吗？我也有点不太懂什么叫没有重叠

 赞

不落之霜华 回复 dou无知无畏2020-11-27


意思就是在高维的空间上，这些分布其实是像流形曲面一样的东西，如果这两个流形曲去优化的。

- 

Danny小猿

2020-10-19


写得很好，学习了

👍 赞
- 

Jarvis

2020-11-04

牛的

👍 赞
- 

知乎用户

02-01

写的不错

👍 赞