深入理解L1、L2正则化


MrLi 关注他

正则化（Regularization）是机器学习中一种常用的技术，其主要目的是控制模型复杂度，减小过拟合。最基本的正则化方法是在原目标（代价）函数 中添加惩罚项，对复杂度高的模型进行“惩罚”。其数学表达形式为：

$$\tilde{J}(w;X,y)=J(w;X,y)+\alpha\Omega(w)$$

式中 X 、 y 为训练样本和相应标签， w 为权重系数向量； $J()$ 为目标函数， $\Omega(w)$ 即为惩罚项，可理解为模型“规模”的某种度量；参数 α 控制控制正则化强弱。不同的 Ω 函数对权重 w 的最优解有不同的偏好，因而会产生不同的正则化效果。最常用的 Ω 函数有两种，即 l_1 范数和 l_2 范数，相应称之为 l_1 正则化和 l_2 正则化。此时有：

$$l_1:\Omega(w)=||w||_1=\sum_i|w_i|$$

$$l_2:\Omega(w)=||w||_2^2=\sum_i w_i^2$$

本文将从不同角度详细说明 l_1 、 l_2 正则化的推导、求解过程，并对 l_1 范数产生稀疏性效果的本质予以解释。

一、 l_1 、 l_2 正则化来源推导

可从带约束条件的优化求解和最大后验概率两种思路来推导 l_1 、 l_2 正则化，下面将予以详细分析。

1、正则化理解之基于约束条件的最优化

对于模型权重系数 w 求解是通过最小化目标函数实现的，即求解：

$$\min_w J(w;X,y)$$

我们知道，模型的复杂度可用VC维来衡量。通常情况下，模型VC维与系数 w 的个数成线性关系：即 w 数量越多，VC维越大，模型越复杂。因此，为了限制模型的复杂度，很自然的思路是减少系数 w 的个数，即让 w 向量中一些元素为0或者说限制 w 中非零元素的个数。为此，我们可在原优化问题中加入一个约束条件：

$$\min_w J(w;X,y)$$
$$s.t.\ ||w||_0\le C$$

$||\cdot||_0$ 范数表示向量中非零元素的个数。但由于该问题是一个NP问题，不易求解，为此我们需要稍微“放松”一下约束条件。为了达到近似效果，我们不严格要求某些权重 w 为0，而是要求权重 w 应接近于0，即尽量小。从而可用 l_1 、 l_2 范数来近似 l_0 范数，即：

$$\min_w J(w;X,y)\ s.t.\ ||w||_1\le C \quad 或 \quad \min_w J(w;X,y)\ s.t.\ ||w||_2\le C$$

使用 l_2 范数时，为方便后续处理，可对 $||w||_2$ 进行平方，此时只需调整 C 的取值即可。利用拉格朗日算子法，我们可将上述带约束条件的最优化问题转换为不带约束项的优化问题，构造拉格朗日函数：

$$L(w,\alpha)=J(w;X,y)+\alpha(||w||_1-C) \quad 或$$

$$L(w,\alpha)=J(w;X,y)+\alpha\left(||w||_2^2-C\right)$$

其中 $\alpha > 0$ ，我们假设 α 的最优解为 α^* ，则对拉格朗日函数求最小化等价于：

$\min_w J(w;X,y) + \alpha^* ||w||_1$ 或

$\min_w J(w;X,y) + \alpha^* ||w||_2^2$

可以看出，上式与 $\min_w \tilde{J}(w;X,y)$ 等价。

故此，我们得到对 l_1 、 l_2 正则化的第一种理解：

- l_1 正则化等价于在原优化目标函数中增加约束条件 $||w||_1 \leq C$
- l_2 正则化等价于在原优化目标函数中增加约束条件 $||w||_2^2 \leq C$

2、正则化理解之最大后验概率估计

在最大似然估计中，是假设权重 w 是未知的参数，从而求得对数似然函数：

$$l(w) = \log[P(y|X;w)] = \log\left[\prod_i P(y^i|x^i;w)\right]$$

通过假设 y^i 的不同概率分布，即可得到不同的模型。例如若假设 $y^i \sim N\left(w^T x^i,\ \sigma^2\right)$ 的高斯分布，则有：

$$l(w) = \log\left[\prod_i \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(y^i - w^T x^i\right)^2}{2\sigma^2}}\right] = -\frac{1}{2\sigma^2}\sum_i \left(y^i - w^T x^i\right)^2 + C$$

式中 C 为常数项，由于常数项和系数项不影响 $\max l(w)$ 的解，因而可令 $J(w;X,y) = -l(w)$ 即可得到线性回归的代价函数。

在最大后验概率估计中，则将权重 w 看作随机变量，也具有某种分布，从而有：

$$P(w|X,y) = \frac{P(w,X,y)}{P(X,y)} = \frac{P(X,y|w)P(w)}{P(X,y)} \propto P(y|X,w)P(w)$$

同样取对数有：

$$\text{MAP} = \log P(y|X,w)P(w) = \log P(y|X,w) + \log P(w)$$

可以看出后验概率函数为在似然函数的基础上增加了一项 $\log P(w)$ 。 $P(w)$ 的意义是对权重系数 w 的概率分布的先验假设，在收集到训练样本 $\{X,y\}$ 后，则可根据 w 在 $\{X,y\}$ 下的后验概率对 w 进行修正，从而做出对 w 更好地估计。

若假设 w_j 的先验分布为0均值的高斯分布，即 $w_j \sim N\left(0,\sigma^2\right)$ ，则有：

$$\log P(w) = \log\prod_j P(w_j) = \log\prod_j \left[\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\left(w_j\right)^2}{2\sigma^2}}\right] = -\frac{1}{2\sigma^2}\sum_j w_j^2 + C'$$

可以看到，在高斯分布下 $\log P(w)$ 的效果等价于在代价函数中增加 l_2 正则项。

若假设 w_j 服从均值为0、参数为 a 的拉普拉斯分布，即：

$$P(w_j) = \frac{1}{\sqrt{2a}} e^{\frac{-|w_j|}{a}}$$

则有：

$$\log P(w) = \log\prod_j \frac{1}{\sqrt{2a}} e^{\frac{-|w_j|}{a}} = -\frac{1}{a}\sum_j |w_j| + C'$$

可以看到，在拉普拉斯分布下 $\log P(w)$ 的效果等价于在代价函数中增加 l_1 正则项。

故此，我们得到对于 l_1 、 l_2 正则化的第二种理解：

- l_1 正则化可通过假设权重 w 的先验分布为拉普拉斯分布，由最大后验概率估计导出；
- l_2 正则化可通过假设权重 w 的先验分布为高斯分布，由最大后验概率估计导出。

二、 l_1 、 l_2 正则化效果分析

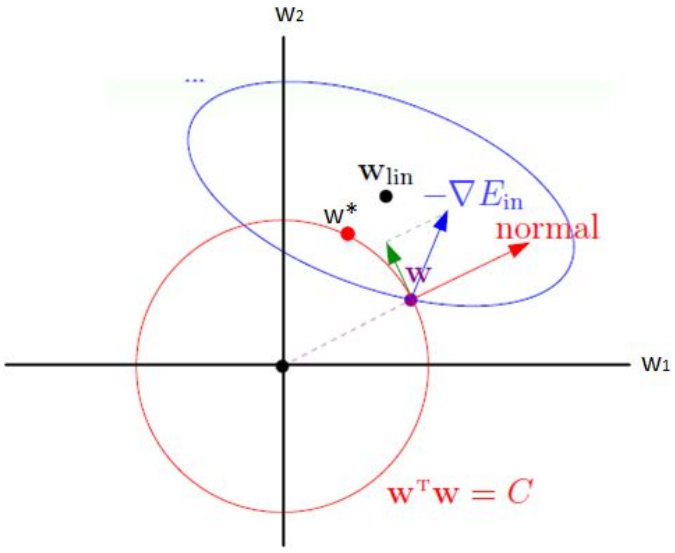
本文将从直观分析和理论推导两个角度来说明 l_1 、 l_2 正则化的效果。

1、直观理解

考虑带约束条件的优化解释，对 l_2 正则化为：

$$\min_w J(w;X,y)$$
$$s.t. ||w||_2 \leq C$$

该问题的求解示意图如下所示：

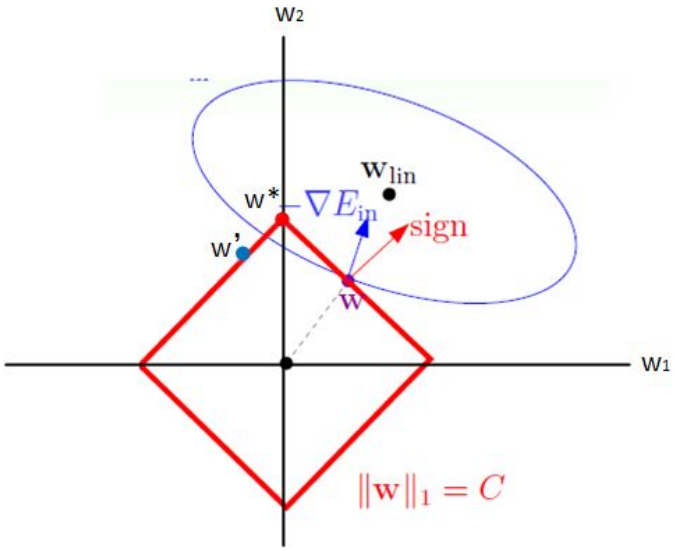


图中椭圆为原目标函数 $J(w)$ 的一条等高线，圆为半径 $\sqrt{C}$ 的 l_2 范数球。由于约束条件的限制， w 必须位于 l_2 范数球内。考虑边界上的一点 w ，图中蓝色箭头为 $J(w)$ 在该处的梯度方向 $\nabla J(w)$ ，红色箭头为 l_2 范数球在该处的法线方向。由于 w 不能离开边界（否则违反约束条件），因而在使用梯度下降法更新 w 时，只能朝 $\nabla J(w)$ 在 范数球上 w 处的切线方向更新，即图中绿色箭头的方向。如此 w 将沿着边界移动，当 $\nabla J(w)$ 与范数球上 w 处的法线平行时，此时 $\nabla J(w)$ 在切线方向的分量为0， w 将无法继续移动，从而达到最优解 w^*（图中红色点所示）。

对于 l_1 正则化：

$$\min_w J(w;X,y)$$
$$s.t. ||w||_1 \leq C$$

同理，其求解示意图如下所示：



其主要差别在于 l_1 、 l_2 范数球的形状差异。由于此时每条边界上 w 的切线和法线方向保持不变，在图中 w 将一直朝着 $\nabla J(w)$ 在切线方向的分量沿着边界向左上移动。当 w 跨过顶点到达 w' 时， $\nabla J(w)$ 在切线方向的分量变为右上方，因而 w 将朝右上方移动。最终， w 将稳定在顶点处，达到最优解 w^* 。此时，可以看到 $w_1 = 0$ ，这也就是采用 l_1 范数会使 w 产生稀疏性的原因。

以上分析虽是基于二维的情况，但不难将其推广到多维情况，其主要目的是为了直观地说明 l_1 、 l_2 正则化最优解的差异，以及 l_1 范数为什么为产生稀疏性。

2、理论分析

假设原目标函数 $J(w)$ 的最优解为 w^* ，并假设其为二阶可导，将 $J(w)$ 在 w^* 处进行二阶泰勒展开有：

$$\hat{J}(w) = J(w^*) + \frac{1}{2}(w - w^*)^T H(w - w^*)$$

式中 H 为 $J(w)$ 在 w^* 处的Hessian矩阵，注意 w^* 为 $J(w)$ 的最优解，其一阶导数为0，因而式中无一阶导数项。 $\hat{J}(w)$ 取得最小值时有：

$$\nabla_w \hat{J}(w) = H(w - w^*) = 0$$

由于 l_2 正则化的目标函数为在 $J(w)$ 中添加 $\Omega(w) = \frac{1}{2}\alpha||w||_2^2 = \frac{1}{2}\alpha w^T w$ ，因而有：

$$\nabla_w \tilde{J}(w) = \nabla_w \hat{J}(w) + \nabla_w \Omega(w) = H(w - w^*) + \alpha w$$

设其最优解为 $\tilde{w}$ ，则有：

$$H(\tilde{w} - w^*) + \alpha\tilde{w} = 0$$

$$\tilde{w} = (H + \alpha I)^{-1} H w^*$$

由于 H 是对称矩阵，可对其作奇异值分解，即 $H = Q\Lambda Q^T$ ，代入上式有：

$$\tilde{w} = Q(\Lambda + \alpha I)^{-1}\Lambda Q^T w^*$$

Λ 为对角矩阵，且对角线元素为 H 的特征值 λ_j 。

由上式可知 $\tilde{w}$ 为 w^* 的的每个元素以 $\frac{\lambda_j}{\lambda_j + \alpha}$ 比例放缩得到。若 $\lambda_j \gg \alpha$ ，则 w_j^* 受正则化的影响较小；若 $\lambda_j \ll \alpha$ ，则 w_j^* 受正则化的影响较大，将收缩到接近于0的值。同时，若 $w_j^* \neq 0$ ，则 $\tilde{w}_j \neq 0$ ，因而 l_2 正则化不会产生稀疏性的效果。

对于 l_1 正则化，只需将 $\Omega(w)$ 替换为 w 的 l_1 范数，同理可以得到：

$$\nabla_w \tilde{J}(w) = \nabla_w \hat{J}(w) + \nabla_w \Omega(w) = H(w - w^*) + \alpha \mathrm{sign}(w)$$

其最优解满足：

$$H\left(\tilde{w}-w^{*}\right)+\alpha \operatorname{sign}\left(\tilde{w}\right)=0$$

为了简化讨论，我们假设 H 为对角阵，即 $H=\operatorname{diag}\left[H_{11}, H_{22}, \ldots, H_{nn}\right]$ ， $H_{jj}>0$ 。此时 w 的不同分量之间没有相关性，该假设可通过对输入特征进行预处理（例如使用PCA）得到，此时 $\tilde{w}$ 的解为：

$$\tilde{w}_{j}=\operatorname{sign}\left(w_{j}^{*}\right) \max \left\{\left|w_{j}^{*}\right|-\frac{\alpha}{H_{jj}}, 0\right\}$$

当 $\left|w_{j}^{*}\right| \leq \frac{\alpha}{H_{jj}}$ 时，可知 $\tilde{w}_{j}=0$ ，因而 l_1 正则化会使最优解的某些元素为0，从而产生稀疏性； $\left|w_{j}^{*}\right|>\frac{\alpha}{H_{jj}}$ 时， $\tilde{w}_{j}$ 会在原有最优解上偏移一个常数值。

综上， l_2 正则化的效果是对原最优解的每个元素进行不同比例的放缩； l_1 正则化则会使原最优解的元素产生不同量的偏移，并使某些元素为0，从而产生稀疏性。

参考文献：

1. Ian Goodfellow, Yoshua Bengio and Aaron Courville. DeepLearning.
2. Hsuan-Tien Lin. Machine Learning Foundations Lecture 14.

编辑于 2017-09-16

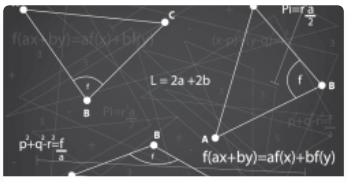
机器学习

推荐阅读

机器学习中正则化项L1和L2的直观理解

正则化（Regularization）机器学习中几乎都可以看到损失函数后面会添加一个额外项，常用的额外项一般有两种，一般英文称作ℓ1-norm和ℓ2-norm，中文称作L1正则化和L2正则化，或者L1范数和L2...

lijia... 发表于计算机视觉...



机器学习------令人头疼的正则化项

EdisonGzq

机器学

1. 损失函
学习中最
前馈神经
看成最大
有数据
(x^\left(

xiaoyuy

9 条评论 切换为时间排序

写下你的评论...

何志 5 个月前

后面的理论分析太赞了

赞

何志 5 个月前

要是VC维分析那里更加详细一点（比如增加正则化如何降低模型复杂度的数学描述），或者增加方差-偏差分解的角度的解释，那就更加完美了。

1

MrLi (作者) 回复 何志 5 个月前

这是在Deep Learning书中看到的，你可以去原书看看更详细的分析。

赞 查看对话

MrLi (作者) 回复 何志 5 个月前

谢谢建议，我再查查资料思考一下。

赞 查看对话

dragonfly 2 个月前

好文 l1正则的图容易产生一个误解 如果梯度的方向跟某条边垂直 是不是可以说此时的解落在了边上呢？而其实二维平面下解只能落在顶点上。

赞

MrLi (作者) 回复 dragonfly 2 个月前

如果梯度和边界垂直，此时就是最优解了，这里考虑的是一般情况下解的更新方向。

赞 查看对话

dragonfly 1 个月前

为什么l1正则会到达w'呢？感觉如图的形式 可能在到达上面的顶点前就能有一个最优解 就是梯度下降方向与法向量平行

赞

MrLi (作者) 1 个月前

我的理解是这种情况是可能会出现的，但可能性比较小，而且实际数据维度比较大，不太容易出现。

赞

meng hu 20 天前

这个解释好清晰啊

赞