机器学习-正则化技术深度总结

无关部分

自我介绍：

本人，姓名：**黄海安**，网名：**深度眸**，南京航空航天大学硕士，业余工程师

各位朋友可以通过以下方式获取前面的视频和文档

(1) 关注：**机器学习算法全栈工程师** 微信公众号，后续我的文章和视频会首先发布在上面，其二维码为：

(2) 加我个人的 QQ 群，ML 和 DL 视频分享群(**678455658**)。在群里大家可以对本视频提出任何意见

(3) 腾讯视频网站、其他公众号和其他一些网站等

欢迎各位提出宝贵意见！

摘要

正则化是一种有效的防止过拟合、提高模型泛化能力方法，在机器学习和深度学习算法中应用非常广泛，本文从机器学习正则化着手，首先阐述了正则化技术的一般作用和概念，然后针对 L1 和 L2 范数正则从 4 个方面深入理解，最后对常用的典型算法应用进行了分析和总结，后续文章将分析深度学习中的正则化技术。注意：本文有对应的视频讲解，如果对文中哪里不理解的可以观看对应的视频。

一、正则化作用及其常见术语

正则化技术广泛应用在机器学习和深度学习算法中，其本质作用是**防止过拟合、提高模型泛化能力**。过拟合简单理解就是训练的算法模型太过复杂了，过分考虑了当前样本结构。其是防止过拟合的其中一种技术手段。在早期的机器学习领域一般只是将范数惩罚叫做正则化技术，而在深度学习领域认为：能够显著减少方差，而不过度增加偏差的策略都可以认为是正则化技术，故推广的正则化技术还有：扩增样本集、早停止、Dropout、集成学习、多任务学习、对抗训练、参数共享等(具体见“花书”，会在下一次文章中重点分析)。对于机器学习领域正则化技术可以从以下几个不同角度进行理解：

(1) 正则化等价于结构风险最小化，其是通过在经验风险项后加上表示模型复杂度的正则化项或惩罚项，达到选择经验风险和模型复杂度都较小的模型目的。

经验风险：机器学习中的风险是指模型与真实解之间的误差的积累，经验风险是指使用训练出来的模型进行预测或者分类，存在多大的误差，可以简单理解为训练误差，经验风险最小化即为训练误差最小。

结构风险：结构风险定义为经验风险与置信风险(置信是指可信程度)的和，置信风险越大，模型推广能力越差。可以简单认为结构风险是经验风险后面多加了一项表示模型复杂度的函数项，从而可以同时控制模型训练误差和测试误差，结构风险最小化即为在保证模型分类精度(经验风险)的同时，降低模型复杂度，提高泛化能力。

$$R(f)=\frac{1}{n}\sum_{i=1}^{n}L\left(y_i,f\left(x_i\right)\right)+\lambda\Omega(f) \tag{1}$$

其中，$R(f)$ 表示结构风险，$L\left(y_i,f\left(x_i\right)\right)$ 表示第 i 个样本的经验风险，$\Omega(f)$ 是表征模型复杂度的正则项，λ 是正则化参数。根据奥卡姆剃刀定律，“如无必要，勿增实体”，即认为相对简单的模型泛化能力更好。而模型泛化能力强、泛化误差小，即表示模型推广能力强，通俗理解就是在训练集中训练得到的优秀模型能够很好的适用于实际测试数据，而不仅仅是减少训练误差或者测试误差。泛化误差定义如下：

$$E=Bias^2\left(X\right)+Var\left(X\right)+Noise \tag{2}$$

其中，E 表示泛化误差，$Bias$ 代表偏差，Var 代表方差，$Noise$ 代表噪声

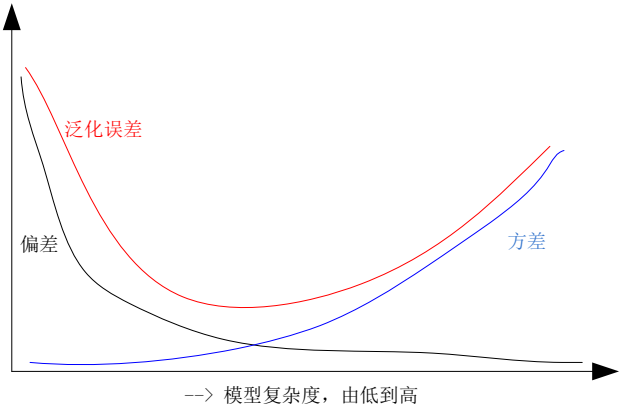

图 1 泛化误差与偏差和方差的关系

从上图可以看出，随着训练程度加深，模型复杂度会增加，偏差减少，方差增大，而泛化误差呈现 U 型变化，对于一个“好的系统”通常要求误差小，正则化的作用即为适当的控制模型复杂度，从而使得泛化误差曲线取最小值。

(2) 正则化等价于带约束的目标函数中的约束项

以平方误差损失函数和 L2 范数为例，优化问题的数学模型如下：

$$J(\theta)=\sum_{i=1}^{n}\left(y_i-\theta^T x_i\right)^2 \tag{3}$$

$$s.t. \quad \|\theta\|_2^2 \le C \tag{4}$$

针对上述带约束条件的优化问题，采用拉格朗日乘积算子法可以转化为无约束优化问题，即

$$J(\theta)=\sum_{i=1}^{n}\left(y_i-w^T x_i\right)^2+\lambda\left(\|\theta\|_2^2-C\right) \tag{5}$$

由于参数 C 为常数，可以忽略，故上述公式和标准的正则化公式完全一致。

(3) 从贝叶斯角度考虑，正则项等价于引入参数的模型先验概率，可以简单理解为对最大似然估计引入先验概率，从而转化为最大后验估计，其中的先验概率即对于正则项

这部分内容后面详细讲解。

二、机器学习正则化技术基本概念

正则化也可以称为规则化、权重衰减技术，不同的领域叫法不一样，数学上常称为范数，例如 L1 和 L2 范数，统计学领域叫做惩罚项、罚因子，以信号降噪为例：

$$x(i)^*=\arg\min_{x(i)}\{F(x(i))=\frac{1}{2}\|y(i)-x(i)\|_2^2+\lambda R(x(i))\} \tag{6}$$

其中，$x(i)$ 既可以是原始信号，也可以是小波或者傅立叶变换等的系数，$R(x(i))$ 是罚函数(范数罚)，λ 是正则项(惩罚项)，$y(i)$ 是传感器采集到的含噪信号，$I=\{0,...,N-1\}$，N 为信号点数，$x(i)^*$ 为降噪后输出，上述公式中的正则化技术作用和机器学习中的完全一样。

下面给出范数的数学公式，方便后面分析：

(1) P 范数：

$$Lp=(\sum_{i=1}^{n}|x_i|^p)^{\frac{1}{p}} \tag{7}$$

(2) L0 范数：0 范数表示向量中非零元素的个数（即为其稀疏度）

(3) L1 范数：即向量元素绝对值之和，p 范数取 1 则为 1 范数

$$\|x\|_1=\sum_{i=1}^{n}|x_i| \tag{8}$$

(4) L2 范数：即向量元素绝对值的平方和再开方，也称为欧几里得距离，p 范数取 2 则为 2 范数

$$\|x\|_2=\sqrt{\sum_{i=1}^{n}|x_i|^2} \tag{9}$$

(5) ∞ 范数: 即所有向量元素绝对值中的最大值，p 范数取 ∞ 则为 ∞ 范数

$$\|x\|_\infty=\max_i|x_i| \tag{10}$$

(6) $-\infty$ 范数：即所有向量元素绝对值中的最小值，p 范数取-∞ 则为-∞ 范数

$$\|x\|_{-\infty}=\min_i|x_i| \tag{11}$$

假设向量长度为 2 维，则有下列图形：

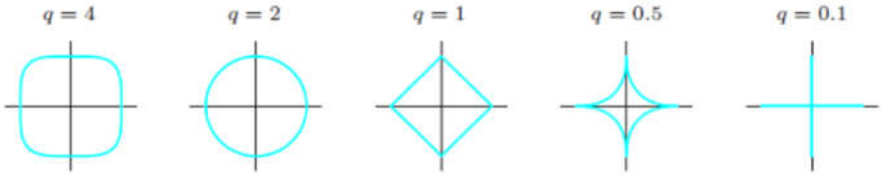

图 2 向量长度为 2 情况下的范数图形

假设向量长度为 3 维，则有下列图形：

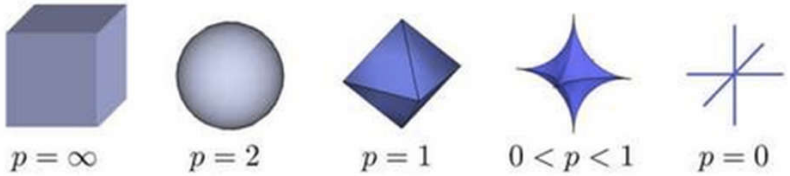

图 3 向量长度为 3 情况下的范数图形

从上述各图可以看出：$q(p)$越小，曲线越贴近坐标轴，$q(p)$越大，曲线越远离坐标轴，并且棱角越明显，当$q(p)$取 0 时候，是完全和坐标轴贴合，当$q(p)$取∞时候，呈现正方体形状。同时也可以看出，采用不同的范数作为正则项，会得到完全不同的算法模型结果，故而对于不同要求的模型，应该采用不同的范数作为正则项。

三、机器学习正则化技术的深度理解

为了更好的理解正则化技术原理，下面从 4 个方面进行深度分析，希望对大家理解有帮助。

3.1 简单数值假设分析法

此处以 L2 范数讲解，下面的各图形来自吴恩达的机器学习课程。

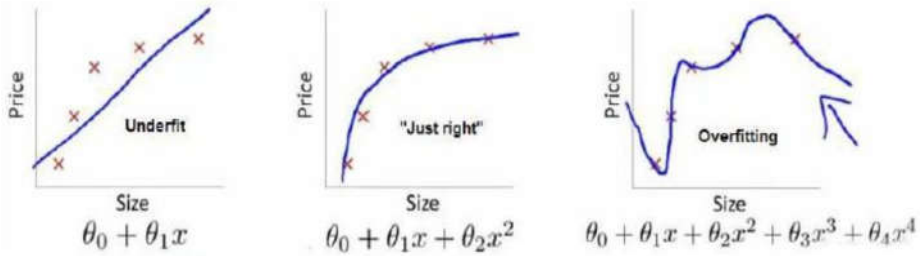

图 4 不同参数下的曲线拟合结果

首先需要明确：左边的曲线拟合是欠拟合，中间的曲线拟合是刚好合适，右边的曲线拟合是过拟合。对于右边的拟合曲线，有

$$h_\theta(x)=\theta_0+\theta_1x_1+\theta_2x_2^2+\theta_3x_3^3+\theta_4x_4^4 \tag{12}$$

从上式可以看出，由于θ_3和θ_4对应了高阶，导致拟合曲线是 4 阶曲线，出现了过拟合。正则化的目的为适当缩减θ_3和θ_4的值，例如都为 0.0001，则上述曲线本质上等价于

$$h_\theta(x)=\theta_0+\theta_1x_1+\theta_2x_2^2 \tag{13}$$

也就是变成了中间的刚好合适的拟合曲线。对θ_3和θ_4增加 L2 正则项后的代价函数表达式为：

$$J(\theta)=\min_\theta\frac{1}{n}\sum_{i=1}^{n}\left(\left(h_\theta(x^i)-y^i\right)^2+1000\theta_3^2++1000\theta_4^2\right) \tag{14}$$

从上式可以看出，θ_3^2和θ_4^2均大于 0，其乘上了 1000，要是$J(\theta)$最小，则会迫使模型学习到的θ_3和θ_4会非常小，因为只有在θ_3和θ_4会非常小的情况下整个代价函数值才会取的较小

指。在实际开发中，是对所有参数进行正则化，为了使代价函数尽可能的小，所有的参数θ的值（不包括θ_0）都会在一定程度上减小，但是减少程度会不一样，从而实现了权重衰减、简化模型复杂度的作用。

3.2 图形分析法

此处采用 L1 和 L2 范数讲解，

(1) L2 范数正则

$$J(\beta)=\sum_{i=1}^{n}\left(y_i-\beta^T x_i\right)^2+\lambda\|\beta\|_2^2 \tag{15}$$

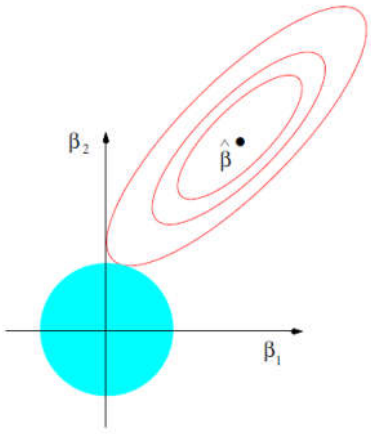

图 5 L2 范数与代价函数的寻优图示

蓝色的圆形空间表示 L2 范数空间，设为$\beta_1^2+\beta_2^2=r^2$，可以看出，当r从 0 逐渐增大时候，该圆形也逐渐增大，红色的线表示原始代价函数解空间即$\sum_{i=1}^{n}\left(y_i-\beta^T x_i\right)^2$，此处为了方便绘图，设参数只有 2 维。红色圆环上的任何一点都表示一个可行解即代表一组β_1,β_2，其中任何一个红色圆环上面的β_1,β_2对应的代价函数值一样(可以简单理解为等值线)，$\hat{\beta}$代表最佳解空间。

由于初中数学知识可知，当正则项$\beta_1^2+\beta_2^2=r^2$和原代价函数项$\sum_{i=1}^{n}\left(y_i-\beta^T x_i\right)^2$这两个空间有交集时候，即代表了一个$J(\beta)$的解，当不存在正则项时候，$\lambda$为 0，$J(\beta)$的解即为$\sum_{i=1}^{n}\left(y_i-\beta^T x_i\right)^2$的解，表示没有解空间没有受到任何约束，通过样本集训练，不容易直接收敛到最优值$\hat{\beta}$，出现过拟合，然而在增加了正则项后，随着不断增加r取值，原始解空间会被不断压缩，如果选择的λ合适，则可以将最优点压缩到$\hat{\beta}$处，从而得到合适的模型。上面就是 L2 范数正则可以避免过拟合的图示。

(2) L1 范数正则

$$J(\beta)=\sum_{i=1}^{n}\left(y_i-\beta^T x_i\right)^2+\lambda\sum_{j=1}^{d}\left|\beta_j\right| \tag{16}$$

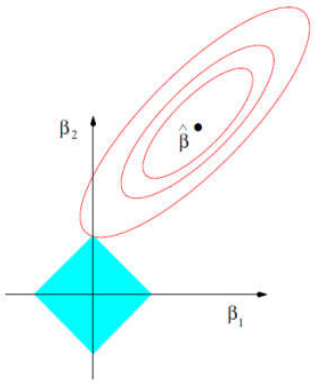

图 6 L1 范数与代价函数的寻优图示

同上述 L2 分析一致，L1 范数对应的解空间图形为菱形，作用和 L2 一致。

需要注意：L2 范数与原代价函数的交点处所得到的参数 β 可以无限缩小，但是一定不会为 0，然而 L1 范数与原代价函数的交点一般在坐标轴上，从而使得某些 β_i=0，得到稀疏解(当然，并没有绝对保证一定交于坐标轴，但是通过实验发现，大部分都可以得到稀疏解)。同时观察上一节的 L0 范数的解空间图形发现：如果使用 L0 范数正则，则可以保证一定得到稀疏解，但是由于 L0 范数的实际求解是 NP 问题，难以计算，故在实际应用中一般都是采用 L1 范数代替 L0 范数得到稀疏解，可以简单认为 L1 范数是 L0 范数的凸近似。

3.3 公式推导分析法

此处采用损失函数为误差平方和、正则项为 L1 和 L2 范数的线性回归为例讲解。增加 L2 正则项后其代价函数为：

$$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}\left(y^i-h_\theta\left(x^i\right)\right)^2+\lambda\sum_{j=1}^{n}\theta_j^2\right] \tag{17}$$

其中 m 为样本个数，n 为特征个数，$\sum_{i=1}^{m}\left(y_i-h_\theta\left(x^i\right)\right)^2$ 为原代价函数，$\sum_{j=1}^{n}\theta_j^2$ 为 L2 范数。

为了最小化代价函数，直接对各 θ_j 进行求导然后等于 0 即可求得估计值(具体推导请查阅其他文献)，可得：

$$\theta_j:=\theta_j\left(1-\alpha\frac{\lambda}{m}\right)-\alpha\frac{1}{m}\sum_{i=1}^{m}\left(y^i-h_\theta\left(x^i\right)\right)x_j^i \tag{18}$$

从上式可以看出：α 为步长，$0<1-\alpha\frac{\lambda}{m}<1$，很明显 L2 范数的作用就是对每一个 θ_j 进行了一定程度的缩减，但是一定不会缩减为 0，从公式也可以看出 L2 范数的作用。

对于 L1 正则项后其代价函数为：

$$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}\left(y^i-h_\theta\left(x^i\right)\right)^2+\lambda\sum_{j=1}^{n}\left|\theta_j\right|\right] \tag{19}$$

直接对各 θ_j 进行求导然后等于 0 即可求得估计值(具体推导请查阅其他文献)，可得：

$$\theta_j:=\theta_j-\alpha\frac{\lambda}{m}\mathrm{sgn}(\theta_j)-\alpha\frac{1}{m}\sum_{i=1}^{m}\left(y^i-h_\theta\left(x^i\right)\right)x_j^i \tag{20}$$

$$\mathrm{sgn}(\theta_j)=\begin{cases}1 & \theta_j>0\\ 0 & \theta_j=0\\ -1 & \theta_j<0\end{cases} \tag{21}$$

从上式可以看出：当上一轮 θ_j 大于 0 时，下一次更新 θ_j 一定减少，当上一轮 θ_j 小于 0 时，

下一次更新θ_j一定增加，也就是说每一轮训练，θ_j都是一定往 0 方向靠近，最终可得近似的稀疏解，同样从公式也可以看出 L1 范数的作用。

同时从上述公式可以看出，在$|\theta_j|<1$情况下，由于 L2 范数正则作用，每次θ_j都是减少$\alpha\frac{\lambda}{m}\theta_j$，而 L1 范数正则作用下，每次$\theta_j$都是减少$\alpha\frac{\lambda}{m}\mathrm{sgn}(\theta_j)$，很明显参数优化速度 L1 快于 L2。

3.4 贝叶斯推断分析法

以 L1 和 L2 范数为例，所得结论可以推广到 P 范数中，首先需要知道：整个最优化问题从贝叶斯观点来看是一种贝叶斯最大后验估计，其中正则化项对应后验估计中的先验信息，损失函数对应后验估计中的似然函数，两者的乘积即对应贝叶斯最大后验估计的形式。针对 L1 和 L2 范数还有结论：**L2 范数相当于给模型参数θ设置一个协方差为 1/alpha 的零均值高斯先验分布，L1 范数相当于给模型参数θ设置一个参数为 1/alpha 的拉普拉斯先验分布**。

为了讲清楚上述结论，需要具备几点前置知识点：(1) 高斯分布和拉普拉斯分布的定义和形状；(2) 贝叶斯定理；(3) 最大似然估计；(4) 最大后验估计。下面我对这 4 个知识点进行解释。

(1) 高斯分布和拉普拉斯分布

a) 高斯分布的概率密度函数定义为：

$$f\left(x\,|\,\mu,\sigma^2\right)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{22}$$

$$\text{记为: } X\sim\left(\mu,\sigma^2\right) \tag{23}$$

其中，μ为数学期望，尺度参数σ为标准差，x为随机变量，其图形为：

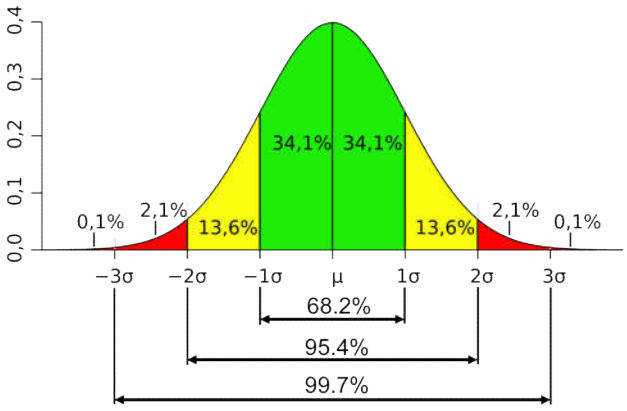

图 7 高斯概率密度函数图示

其中，μ控制曲线的左右移动，σ控制曲线的衰减快慢，σ越大，曲线越平缓，衰减越慢。

b) 拉普拉斯分布的概率密度函数定义为：

$$f\left(x\,|\,\mu,b\right)=\frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right) \tag{24}$$

其中，μ为数学期望，b为尺度参数，x为随机变量，其图形为：

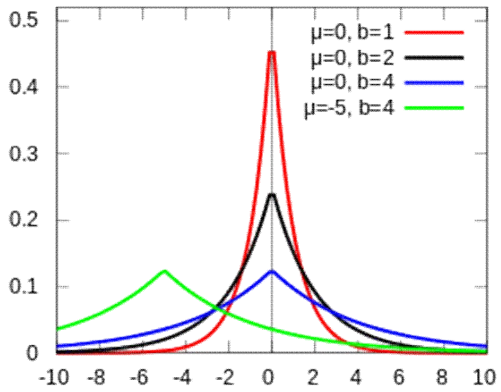

图 8 拉普拉斯概率密度函数图示

(2) 最大似然估计

定义为：在已知试验结果（即是样本）的情况下，用来估计满足这些样本分布的参数，把可能性最大的那个参数 $\hat{\theta}$ 作为真实 θ 的参数估计。

通俗理解就是，该算法作用是找到一组参数 $\hat{\theta}$ 来最大化复现或者拟合当前样本空间。可以发现如果样本空间不一样，则通过最大似然估计得到的 $\hat{\theta}$ 也不一样，即最大似然估计永远都是基于当前样本，所以可以想象出很容易出现过拟合即求得的参数 $\hat{\theta}$ 只是能很好的拟合当前样本，然而推广和泛化能力很弱，是不是有点像没有加正则项的损失函数。

若总体 X 属于离散型，其分布律 $P\left(X=x\right)=p\left(x;\theta\right)$，$\theta$ 形式已知，是待估参数，设 $X_1,X_2,\cdots X_n$ 为来自总体 X 的样本，其 $X_1,X_2,\cdots X_n$ 的联合分布律为：

$$\prod_{i=1}^{n}p\left(x_i;\theta\right) \tag{25}$$

又设 $x_1,x_2,\cdots x_n$ 是来自 $X_1,X_2,\cdots X_n$ 对应的一个样本，易知样本 $X_1,X_2,\cdots X_n$ 观察到 $x_1,x_2,\cdots x_n$ 的概率，亦即事件 $\left\{X_1=x_1,X_2=x_2,\cdots X_n=x_n\right\}$ 发生的概率为：

$$L(\theta)=L\left(x_1,x_2,\cdots x_n;\theta\right)=\prod_{i=1}^{n}p\left(x_i;\theta\right) \tag{26}$$

$L(\theta)$ 为似然函数，最大化 $L(\theta)$ 所得到的参数 θ 即为最大似然估计法。由于上述式子不好计算，而因为 $L(\theta)$ 和 $\ln L(\theta)$ 在同一处取极值，故通常取对数即所谓的对数似然函数，其中最大似然函数如下所示：

$$\theta_{MLE}=\arg\max_{\theta}\prod_{i=1}^{n}p\left(x_i;\theta\right) \tag{27}$$

如果上述公式不能理解，请各位读者去复习一下大学课程《概率论与数理统计》中的参数估计章节，为了更方便理解，下面举一个例子：假设我要统计出整个大学内所有同学的身高分布情况，设全校一共 20000 人，数量庞大，所有人都去问一遍不太靠谱，所以我打算采用抽样方法来估计，假设我已经知道身高分布服从高斯分布，但是我不知道高斯分布中的均值和方差参数，现在我打算采用最大似然估计方法来确定这两个参数。首先需要明确，全校 20000 即为总体 X，我随机从各个班抽取 10 名同学，假设一共抽了 2000 个同学，那么 2000 同学就构成了样本空间，则抽取的 2000 个同学就构成了 $x_1,x_2,\cdots x_{2000}$，由于每个样本的概率密度函数已知，则很容易写出似然函数，对数求导即可求解参数。

(3) 最大后验估计

首先需要明确：**最大后验估计和最大似然估计联系非常密切，对最大似然估计引入先验概率估计即转化为最大后验估计，最大后验概率估计可以看作是规则化的最大似然估计**。最大似然估计属于频率派的观点，其认为参数 θ 是一个固定不变的常量，只是我们现在还不知道它的值，可以通过随机产生的样本去估计这个参数。最大后验估计属于贝叶斯学派推导而来，其认为一切皆变量，服从某一个分布，认为参数 θ 是一个未知的随机变量，我们可以

给出参数θ分布情况的先验概率，然后基于贝叶斯定理估计模型。

根据贝叶斯公式可得，后验概率计算公式为：

$$p(\theta \mid X)=\frac{p(X \mid \theta) p(\theta)}{p(X)} \tag{28}$$

$$=\frac{(\prod_{i=1}^{n} p\left(x_i ; \theta\right)) p(\theta)}{\int(\prod_{i=1}^{n} p\left(x_i ; \theta\right)) p(\theta) d \theta} \tag{29}$$

由于分母计算非常困难，而我们的目的是求最大化后验概率，故分母不进行计算(我们在朴素贝叶斯算法中也是这样处理的)，只考虑分子：

$$\theta_{MAP}=\arg \max _{\theta}(\prod_{i=1}^{n} p\left(x_i ; \theta\right)) p(\theta) \tag{30}$$

仔细观察θ_{MLE}和θ_{MAP}的形式，可以很容易看出：**最大后验估计就是在最大似然估计函数上面乘上了一项先验分布而已**。可不能小看了这个先验分布，其作用非常大，后面会详述。

下面开始解释：**L2 范数相当于给模型参数θ设置一个零均值高斯先验分布。**以线性回归模型为例，结论可以推广到任意模型，线性模型方程可以表示为：

$$Y=\theta^T X+\varepsilon \tag{31}$$

其中，ε表示误差，假设$\varepsilon_i \sim N\left(0, \sigma^2\right)$，$\theta_i \sim N\left(0, \tau^2\right)$则有：

$$p\left(\varepsilon_i\right)=\frac{1}{\sqrt{2 \pi \sigma^2}} \exp (-\frac{\left(\varepsilon_i\right)^2}{2 \sigma^2}) \tag{32}$$

$$p(y_i \mid x_i ; \theta)=\frac{1}{\sqrt{2 \pi \sigma^2}} \exp (-\frac{\left(y_i-\theta^T x_i\right)^2}{2 \sigma^2}) \tag{33}$$

通过上面的θ_{MAP}可知最大后验估计方程形式，对其取对数：

$$\arg \max _{\theta} \ln L(\theta)=\arg \max _{\theta}(\ln \prod_{i=1}^{n} p\left(y_i \mid x_i ; \theta\right)+\ln p(\theta)) \tag{34}$$

$$=\ln \prod_{i=1}^{n} \frac{1}{\sqrt{2 \pi \sigma^2}} \exp (-\frac{\left(y_i-\theta^T x_i\right)^2}{2 \sigma^2})+\ln \prod_{j=1}^{d} \frac{1}{\sqrt{2 \pi \tau^2}} \exp (-\frac{\left(\theta_j\right)^2}{2 \tau^2})$$

$$=-\frac{1}{2 \sigma^2} \sum_{i=1}^{n}\left(y_i-\theta^T x_i\right)^2-\frac{1}{2 \tau^2} \sum_{j=1}^{d} \theta_j{}^2-n \ln \sigma \sqrt{2 \pi}-d \ln \tau \sqrt{2 \pi}$$

要求上式最大，去掉负号和统一处理前面的参数，故而可以转化为

$$\arg \min _{\theta} \ln L(\theta)=\sum_{i=1}^{n}\left(y_i-\theta^T x_i\right)^2+\lambda \sum_{j=1}^{d} \theta_j{}^2 \tag{35}$$

上式正好是线性回归问题在 L2 范数正则下的代价函数，故验证了结论。

下面开始解释：**L1 范数相当于给模型参数θ设置一个拉普拉斯先验分布。**以线性回归模型为例，结论可以推广到任意模型，同样假设$\varepsilon_i \sim N\left(0, \sigma^2\right)$，而$\theta_i \sim Laplace\left(0, b\right)$，

$$\arg \max _{\theta} \ln L(\theta)=\ln \prod_{i=1}^{n} p\left(y_i \mid x_i ; \theta\right)+\ln p(\theta) \tag{36}$$

$$=\ln \prod_{i=1}^{n} \frac{1}{\sqrt{2 \pi \sigma^2}} \exp (-\frac{\left(y_i-\theta^T x_i\right)^2}{2 \sigma^2})+\ln \prod_{j=1}^{d} \frac{1}{2 b} \exp (-\frac{\left|\theta_j\right|}{b})$$

$$=-\frac{1}{2 \sigma^2} \sum_{i=1}^{n}\left(y_i-\theta^T x_i\right)^2-\frac{1}{b} \sum_{i=1}^{n}\left|\theta_j\right|-n \ln \sigma \sqrt{2 \pi}-d \ln 2 b$$

要求上式最大，去掉负号和统一处理前面的参数，故而可以转化为

$$\arg\min_{\theta} \ln L(\theta) = \sum_{i=1}^{n}\left(y_i - \theta^T x_i\right)^2 + \lambda\sum_{j=1}^{d}\left|\theta_j\right| \tag{37}$$

上式正好是线性回归问题在 L1 范数正则下的代价函数，故验证了结论。

附加一句：如果误差符合 0 均值的高斯分布，那么最大似然估计法的结果就是最小二乘法，这也是为何误差定义经常使用 $\sum_{i=1}^{n}\left(y_i - \theta^T x_i\right)^2$ 的原因，因为这个公式是基于概率推导出来的，有比较强的科学依据。

四、机器学习正则化技术的典型算法应用

4.1 逻辑回归

二分类逻辑回归使用 Sigmoid 作为决策函数进行分类，该函数可以将任意的输入映射到[0,1]区间，当预测结果小于 0.5，则表示负类，当预测结果大于 0.5.则表示正类，其模型本质是求最大似然估计，具体求解似然函数通常使用梯度下降法，而前面说过：最大似然估计法没有考虑训练集以外的因素，很容易造成过拟合，故而逻辑回归一般采用 L2 范数进行正则化操作，Sigmoid 函数定义和图形如下：

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T X}} \tag{38}$$

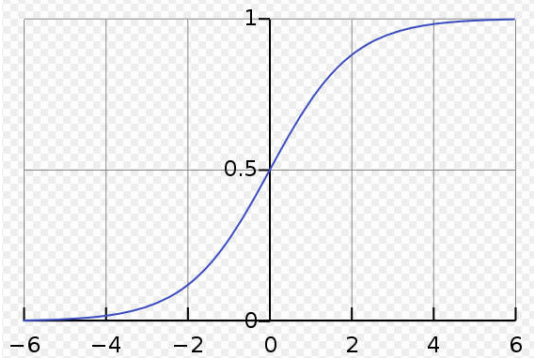

图 9 Sigmoid 函数图示

其中似然方程和对数似然方程为：

$$L(\theta) = \prod_{i=1}^{n} P(y_i \mid x_i;\theta) = \prod_{i=1}^{n}\left(h_\theta(x_i)\right)^{y_i}\left(1-h_\theta(x_i)\right)^{1-y_i} \tag{39}$$

$$l(\theta) = \log L(\theta) = \sum_{i=1}^{n}\left(y_i \log h_\theta(x_i) + (1-y_i)\log(1-h_\theta(x_i))\right) \tag{40}$$

正则化后的代价函数为：

$$J(\theta) = -\frac{1}{n}[\sum_{i=1}^{n}\left(y_i \log h_\theta(x_i) + (1-y_i)\log(1-h_\theta(x_i))\right)] + \frac{\lambda}{2n}\sum_{j=1}^{d}\theta_j^2 \tag{41}$$

注意：正则化是针对损失函数，而不是似然函数，故需要将似然函数取负号转换为损失函数，然后再加上正则项。

4.2 岭回归(Ridge Regression)

岭回归本质上是针对线性回归问题引入了 L2 范数正则，通过缩减回归系数避免过拟合问题，最先用来处理特征数多于样本数的情况(高维小样本问题)，现在也用于在估计中加人偏差，从而得到更好的估计，加了正则化后的代价函数如下：

$$\hat{\beta}^{ridge}=\arg\min_{\beta}\left\{\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{d}x_{ij}\beta_j)^2+\lambda\sum_{j=1}^{d}\beta_j^2\right\} \tag{42}$$

其中，$\hat{\beta}$ 表示估计的回归系数，n 表示样本个数，d 表示回归系数个数，y_i 表示第 i 个样本实际输出值，β_j 表示第 j 个回归系数，λ 为正则化参数。当 λ=0，表示不添加正则，则很容易导致原代价函数为 0，预测值与实际值完全贴合即出现了所谓的过拟合问题，当 λ 过大，会导致 β_j 系数变小，但不会为 0，减少了模型复杂度，原代价函数值较大，出现欠拟合。在实际开发中，通常使用交叉验证集多次循环迭代确定最佳 λ 值。

$$\hat{\beta}^{ridge}=\left(X^TX+\lambda I\right)^{-1}X^Ty \tag{43}$$

带正则化的代价函数采用最小二乘法或者正规方程可以得到上述回归系数结果，可以发现：经过 L2 范数罚后，不仅仅压缩了系数，而且可以使得原先可能不可逆的矩阵一定可逆（$X^TX+\lambda I$ 一定可逆），这也是 L2 正则的好处之一。

4.3 Lasso 回归

拉索回归(lasso 回归)本质上是针对线性回归问题引入了 L1 范数正则，通过缩减回归系数避免过拟合问题，其不同于 L2 范数，其可以将某些系数缩减为 0 即所谓的具备稀疏性(稀疏性的好处是简化计算、容易理解模型、减少存储空间、不容易出现过拟合等等)，加了正则化后的代价函数如下：

$$\hat{\beta}^{lasso}=\arg\min_{\beta}\left\{\sum_{i=1}^{n}(y_i-\beta_0-\sum_{j=1}^{d}x_{ij}\beta_j)^2+\lambda\sum_{j=1}^{d}\left|\beta_j\right|\right\} \tag{44}$$

其中，参数函数和岭回归中相同。L1 范数罚有一个问题：由于 $|x|$ 函数在 0 处不可导，故而直接使用最小二乘法、梯度下降法等方法均失效，但是由于其为第一类间断点中的可去间断点，可以通过补充该点的定义解决，通常，对于线性回归中的 lasso 回归可以采用近似的前向逐步回归替代。

4.4 SVM

支持向量机 SVM 优化目的为寻找一个超平面，使得正负样本能够以最大间隔分离开，从而得到更好的泛化性能，其通过引入核函数来将低维线性不可分的样本映射到高维空间从而线性可分，通过引入惩罚参数 C(类似于正则化参数)来对错分样本进行惩罚，从而减少模型复杂度，提高泛化能力，其优化目标如下：

$$\min_{\theta,b}\frac{1}{n}\sum_{i=1}^{n}\max(1-y_i\left(\theta^Tx_i+b\right),0)+\frac{1}{2CN}\theta^T\theta \tag{45}$$

$$\lambda=\frac{1}{2C} \tag{46}$$

大家如果不知道上面公式的推导，不用紧张，对于本次内容不是重点，只需要关注后面正则项部分，惩罚参数 C 作用和正则化参数 λ 作用一致，只是反相关而已。需要明白以下结论：

(1) C 越大，λ 越小，表示对分错样本的惩罚程度越大，正则化作用越小，偏差越小，方差越大，越容易出现过拟合(通俗理解，原本将低维空间映射到 5 维空间正好线性可分，但是由于惩罚过于严重，任何一个样本分错了都不可原谅，结果系统只能不断提高维数来拟合样本，假设为 10 维，最终导致映射维数过高，出现过拟合样本现象，数学上称为 VC 维较大)；

(2) C 越小，λ 越大，表示对分错样本的惩罚程度越小，正则化作用越大，偏差越大，方差越小，越容易出现欠拟合(通俗理解，原本将低维空间映射到 5 维空间正好线性可分，但是由于惩罚过小，分错了好多样本都可以理解，比较随意，结果系统也采用简化版来拟合

样本，假设为 3 维，最终导致映射维数过低，出现欠拟合样本现象，数学上称为 VC 维较小)。

五、正则化技术总结

本文从各个角度深入的分析了机器学习算法中使用到的正则化技术，正则化技术是机器学习和深度学习中非常重要的内容，不管是在面试、笔试还是实际应用中都至关重要，通过本文您应该知道以下重要结论：

(1) 正则化的作用是防止过拟合、提高模型泛化能力

(2) 正则化等价于结构风险最小化

(3) 正则化等价于带约束的目标函数中的约束项

(4) 正则项等价于引入参数的模型先验概率

(5) 在误差符合均值为 0 的高斯分布，则最大似然估计和最小二乘法等价

(6) 最大后验估计就是在最大似然估计函数上面乘上了一项先验分布而已

(7) L2 范数相当于给模型参数 θ 设置一个零均值高斯先验分布，L1 范数相当于给模型参数 θ 设置一个拉普拉斯先验分布

(8) L0 和 L1 正则可以得到稀疏解，而 L2 不能，并且参数优化速度 L1 快于 L2，但是 L2 更容易理解，计算更方便。

有一个需要注意的地方：正则化方法一般都是不对偏移项进行正则的，原因是它们也没有和数据直接有乘法等交互，其不会影响到最后结果中某个数据维度的作用，如果你执意要对其进行正则化，也是可以的，对结果没有多少影响。

前面讨论了，正则化是一种可以有效防止过拟合的方法，然而如何判断模型是否或者容易出现过拟合？常用的办法有：(1) 比较模型对验证集和训练集的识别精度，如果验证集识别精度大幅低于训练集，则可以判断模型存在过拟合；(2) 训练集的代价函数快速下降至 0 附近，也可以怀疑出现了过拟合；(3) 样本个数比特征个数少，也很容易出现过拟合。

由于本人水平有限，如果哪里有写错的或者理解不到位的，欢迎提出建议！

六、参考内容

[1] 概率论与数理统计 浙大版（第四版）教材
[2] 吴恩达 CS229 课程
[3] Deep Learning
[4] pattern recognition and machine learning
[5] http://www.jianshu.com/p/a47c46153326
[6] http://www.jianshu.com/p/f71848c7aaf3
[7] https://www.zhihu.com/question/23536142/answer/90135994
[8] 其他优秀的博客