个人资料

zouxy09

访问： 6072104次
积分： 26193
等级： BLOG 7
排名： 第166名

原创： 116篇 转载： 11篇
译文： 1篇 评论： 3463条

个人简介

关注：机器学习、计算机视觉、人机交互和人工智能等领域。
邮箱：zouxy09@qq.com
微博：Erik-zou
交流请发邮件，不怎么看博客私信^-^

机器学习中的范数规则化之（一）L0、L1与L2范数

2014-05-04 12:32 164907人阅读 评论(73) 收藏 举报

分类： 机器学习（45）

机器学习中的范数规则化之（一）**L0**、**L1**与**L2**范数

zouxy09@qq.com

http://blog.csdn.net/zouxy09

今天我们聊聊机器学习中出现的非常频繁的问题：过拟合与规则化。我们先简单的来理解下常用的L0、L1、L2和核范数规则化。最后聊下规则化项参数的选择问题。这里因为篇幅比较庞大，为了不吓到大家，我将这个五个部分分成两篇博文。知识有限，以下都是我一些浅显的看法，如果理解存在错误，希望大家不吝指正。谢谢。

监督机器学习问题无非就是“minimizeyour error while regularizing your parameters”，也就是在规则化参数的同时最小化误差。最小化误差是为了让我们的模型拟合我们的训练数据，而规则化参数是防止我们的模型过分拟合我们的训练数据。多么简约的哲学啊！因为参数太多，会导致我们的模型复杂度上升，容易过拟合，也就是我们的训练误差会很小。但训练误差小并不是我们的最终目标，我们的目标是希望模型的测试误差小，也就是能准确的预测新的样本。所以，我们需要保证模型“简单”的基础上最小化训练误差，这样得到的参数才具有好的泛化性能（也就是测试误差也小），而模型“简单”就是通过规则函数来实现的。另外，规则项的使用还可以约束我们的模型的特性。这样就可以将人对这个模型的先验知识融入到模型的学习当中，强行地让学习到的模型具有人想要的特性，例如稀疏、低秩、平滑等等。要知道，有时候人的先验是非常重要的。前人的经验会让你少走很多弯路，这就是为什么我们平时学习最好找个大牛带带的原因。一句点拨可以为我们拨开眼前乌云，还我们一片晴空万里，醍醐灌顶。对机器学习也是一样，如果被我们人稍微点拨一下，它肯定能更快的学习相应的任务。只是由于人和机器的交流目前还没有那么直接的方法，目前这个媒介只能由规则项来担当了。

还有几种角度来看待规则化的。规则化符合奥卡姆剃刀(Occam's razor)原理。这名字好霸气，razor！不过它的思想很平易近人：在所有可能选择的模型中，我们应该选择能够很好地解释已知数据并且十分简单的模型。从贝叶斯估计的角度来看，规则化项对应于模型的先验概率。民间还有个说法就是，规则化是结构风险最小化策略的实现，是在经验风险上加一个正则化项(regularizer)或惩罚项(penalty term)。

一般来说，监督学习可以看做最小化下面的目标函数：

OpenCV (29)
机器学习 (46)
计算机视觉 (73)
Deep Learning (18)
语音识别与TTS (13)
图像处理 (55)
Linux (15)
Linux驱动 (4)
嵌入式 (18)
OpenAL (3)
Android (1)
C/C++编程 (18)
摄像头相关 (5)
数学 (5)
Kinect (9)
神经网络 (8)
随谈 (2)

文章存档

2015年10月 (4)
2015年04月 (2)
2014年12月 (1)
2014年08月 (1)
2014年05月 (2)
展开

阅读排行

Deep Learning（深度学 (475395)
Deep Learning（深度学 (354241)
Deep Learning（深度学 (332106)
Deep Learning论文笔记 (211145)
Deep Learning（深度学 (211056)
Deep Learning（深度学 (190419)
Deep Learning（深度学 (182513)
从最大似然到EM算法浅 (169145)
机器学习中的范数规则化 (164844)
Deep Learning（深度学 (155221)

评论排行

Deep Learning论文笔记 (259)
从最大似然到EM算法浅 (165)
基于Qt的P2P局域网聊天 (164)
Deep Learning（深度学 (159)
时空上下文视觉跟踪（S (130)
计算机视觉、机器学习相 (120)
机器学习算法与Python实 (94)
Deep Learning（深度学 (77)
机器学习中的范数规则化 (73)
压缩跟踪Compressive T (70)

最新评论

目标检测的图像特征提取之（一）
xing_jl: 2（2）那里梯度算子是不是错了？如果以右和上为正方向那么算子应该分别为，'吧？

$$w^* = \arg\min_w \sum_i L(y_i, f(x_i; w)) + \lambda\Omega(w)$$

其中，第一项$L(y_i,f(x_i;w))$ 衡量我们的模型（分类或者回归）对第i个样本的预测值$f(x_i;w)$和真实的标签y_i之前的误差。因为我们的模型是要拟合我们的训练样本的嘛，所以我们要求这一项最小，也就是要求我们的模型尽量的拟合我们的训练数据。但正如上面说言，我们不仅要保证训练误差最小，我们更希望我们的模型测试误差小，所以我们需要加上第二项，也就是对参数w的规则化函数Ω(w)去约束我们的模型尽量的简单。

OK，到这里，如果你在机器学习浴血奋战多年，你会发现，哎哟哟，机器学习的大部分带参模型都和这个不但形似，而且神似。是的，其实大部分无非就是变换这两项而已。对于第一项Loss函数，如果是Square loss，那就是最小二乘了；如果是Hinge Loss，那就是著名的SVM了；如果是exp-Loss，那就是牛逼的 Boosting了；如果是log-Loss，那就是Logistic Regression了；还有等等。不同的loss函数，具有不同的拟合特性，这个也得就具体问题具体分析的。但这里，我们先不究loss函数的问题，我们把目光转向“规则项Ω(w)”。

规则化函数Ω(w)也有很多种选择，一般是模型复杂度的单调递增函数，模型规则化值就越大。比如，规则化项可以是模型参数向量的范数。然而，不同的选择对参数w的约束不同，取得的效果也不同，但我们在论文中常见的都聚集在：零范数、一范数、二范数、迹范数、Frobenius范数和核范数等等。这么多范数，到底它们表达啥意思？具有啥能力？什么时候才能用？什么时候需要用呢？不急不急，下面我们挑几个常见的娓娓道来。

一、L0范数与L1范数

L0范数是指向量中非0的元素的个数。如果我们用L0范数来规则化一个参数矩阵W的话，就是希望W的大部分元素都是0。这太直观了，太露骨了吧，换句话说，让参数W是稀疏的。OK，看到了“稀疏”二字，大家都应该从当下风风火火的“压缩感知”和“稀疏编码”中醒悟过来，原来用的漫山遍野的“稀疏”就是通过这玩意来实现的。但你又开始怀疑了，是这样吗？看到的papers世界中，稀疏不是都通过L1范数来实现吗？脑海里是不是到处都是$||W||_1$影子呀！几乎是抬头不见低头见。没错，这就是这节的题目把L0和L1放在一起的原因，因为他们有着某种不寻常的关系。那我们再来看看L1范数是什么？它为什么可以实现稀疏？为什么大家都用L1范数去实现稀疏，而不是L0范数呢？

L1范数是指向量中各个元素绝对值之和，也有个美称叫“稀疏规则算子”（Lasso regularization）。现在我们来分析下这个价值一个亿的问题：为什么L1范数会使权值稀疏？有人可能会这样给你回答“它是L0范数的最优凸近似”。实际上，还存在一个更美的回答：任何的规则化算子，如果他在$W_i=0$的地方不可微，并且可以分解为一个“求和”的形式，那么这个规则化算子就可以实现稀疏。这说是这么说，W的L1范数是绝对值，|w|在w=0处是不可微，但这还是不够直观。这里因为我们需要和L2范数进行对比分析。所以关于L1范数的直观理解，请待会看看第二节。

对了，上面还有一个问题：既然L0可以实现稀疏，为什么不用L0，而要用L1呢？个人理解一是因为L0范数很难优化求解（NP难问题），二是L1范数是L0范数的最优凸近似，而且它比L0范数要容易优化求解。所以大家才把目光和万千宠爱转于L1范数。

$$\min \|x\|_0 \quad \text{s.t. } Ax = b$$

在一定条件下，以概率1意义下等价

$$\min \|x\|_1 \quad \text{s.t. } Ax = b$$

OK，来个一句话总结：L1范数和L0范数可以实现稀疏，L1因具有比L0更好的优化求解特性而被广泛应用。

好，到这里，我们大概知道了L1可以实现稀疏，但我们会想呀，为什么要稀疏？让我们的参数稀疏有什么好处呢？这里扯两点：

1）特征选择**(Feature Selection)**：

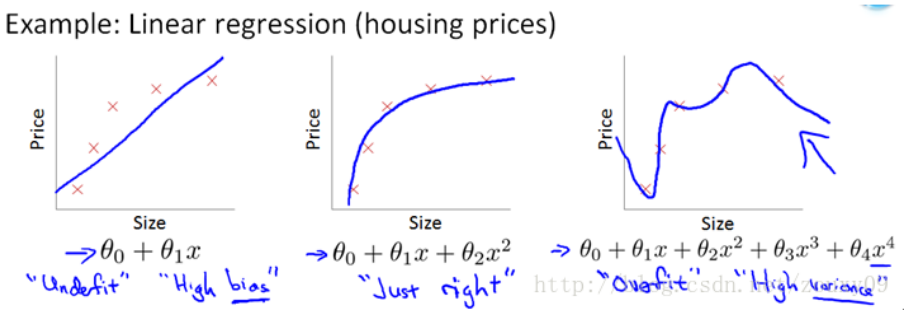
大家对稀疏规则化趋之若鹜的一个关键原因在于它能实现特征的自动选择。一般来说，x_i的大部分元素（也就是特征）都是和最终的输出y_i没有关系或者不提供任何信息的，在最小化目标函数的时候考虑x_i这些额外的特征，虽然可以获得更小的训练误差，但在预测新的样本时，这些没用的信息反而会被考虑，从而干扰了对正确y_i的预测。稀疏规则化算子的引入就是为了完成特征自动选择的光荣使命，它会学习地去掉这些没有信息的特征，也就是把这些特征对应的权重置为0。

2）可解释性**(Interpretability)**：

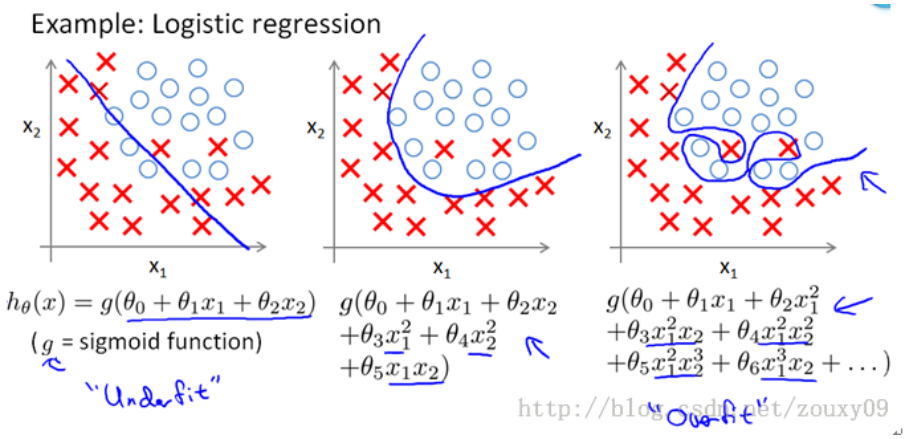
另一个青睐于稀疏的理由是，模型更容易解释。例如患某种病的概率是y，然后我们收集到的数据x是1000维的，也就是我们需要寻找这1000种因素到底是怎么影响患上这种病的概率的。假设我们这个是个回归模型：$y=w_1*x_1+w_2*x_2+...+w_{1000}*x_{1000}+b$（当然了，为了让y限定在[0,1]的范围，一般还得加个Logistic函数）。通过学习，如果最后学习到的w*就只有很少的非零元素，例如只有5个非零的w_i，那么我们就有理由相信，这些对应的特征在患病分析上面提供的信息是巨大的，决策性的。也就是说，患不患这种个因素有关，那医生就好分析多了。但如果1000个w_i都非0，医生面对这1000种因素，累觉不爱。

二、L2范数

除了L1范数，还有一种更受宠幸的规则化范数是L2范数: $||W||_2$。它也不逊于L1范数，它有两个美称，在回归里面，有人把有它的回归叫“岭回归”（Ridge Regression），有人也叫它“权值衰减weight decay”。这用的很多吧，因为它的强大功效是改善机器学习里面一个非常重要的问题：过拟合。至于过拟合是什么，上面也解释了，就是模型训练时候的误差很小，但在测试的时候误差很大，也就是我们的模型复杂到可以拟合到我们的所有训练样本了，但在实际预测新的样本的时候，糟糕的一塌糊涂。通俗的讲就是应试能力很强，实际应用能力很差。擅长背诵知识，却不懂得灵活利用知识。例如下图所示（来自Ng的course）：



上面的图是线性回归，下面的图是Logistic回归，也可以说是分类的情况。从左到右分别是欠拟合（underfitting，也称High-bias）、合适的拟合和过拟合（overfitting，也称High variance）三种情况。可以看到，如果模型复杂（可以拟合任意的复杂函数），它可以让我们的模型拟合所有的数据点，也就是基本上没有误差。对于回归来说，就是我们的函数曲线通过了所有的数据点，如上图右。对分类来说，就是我们的函数曲线要把所有的数据点都分类正确，如下图右。这两种情况很明显过拟合了。



OK，那现在到我们非常关键的问题了，为什么L2范数可以防止过拟合？回答这个问题之前，我们得先看看L2范数是个什么东西。

L2范数是指向量各元素的平方和然后求平方根。我们让L2范数的规则项$||W||_2$最小，可以使得W的每个元素都很小，都接近于0，但与L1范数不同，它不会让它等于0，而是接近于0，这里是有很大的区别的哦。而越小的参数说明模型越简单，越简单的模型则越不容易产生过拟合现象。为什么越小的参数说明模型越简单？我也不懂，我的理解是，限制了参数很小，实际上就限制了多项式某些分量的影响很小（看上面线性回归的模型的那个拟合的图），这样就相当于减少参数个数。其实我也不太懂，希望大家可以指点下。

这里也一句话总结下：通过L2范数，我们可以实现了对模型空间的限制，从而在一定程度上避免了过拟合。

L2范数的好处是什么呢？这里也扯上两点：

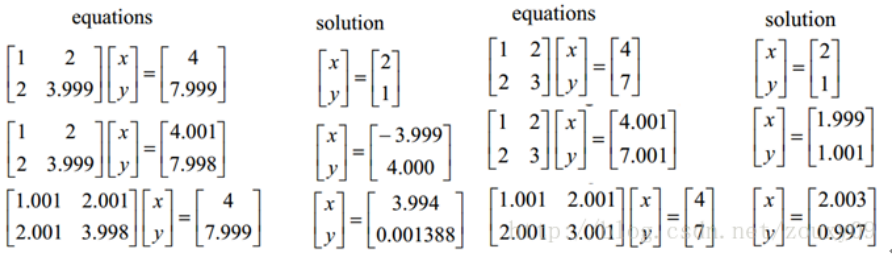
1）学习理论的角度：

从学习理论的角度来说，L2范数可以防止过拟合，提升模型的泛化能力。

2）优化计算的角度：

从优化或者数值计算的角度来说，L2范数有助于处理 condition number不好的情况下矩阵求逆很困难的问题。哎，等等，这condition number是啥？我先google一下哈。

这里我们也故作高雅的来聊聊优化问题。优化有两大难题，一是：局部最小值，二是：ill-condition病态问题。前者俺就不说了，大家都懂吧，我们要找的是全局最小值，如果局部最小值太多，那我们的优化算法就很容易陷入局部最小而不能自拔，这很明显不是观众愿意看到的剧情。那下面我们来聊聊ill-condition。ill-condition对应的是well-condition。那他们分别代表什么？假设我们有个方程组AX=b，我们需要求解X。如果A或者b稍微的改变，会使得X的解发生很大的改变，那么这个方程组系统就是ill-condition的，反之就是well-condition的。我们具体举个例子吧：



咱们先看左边的那个。第一行假设是我们的AX=b，第二行我们稍微改变下b，得到的x和没改变前的差别很大，看到吧。第三行我们稍微改变下系数矩阵A，可以看到结果的变化也很大。换句话来说，这个系统的解对系数矩阵A或者b太敏感了。又因为一般我们的系数矩阵A和b是从实验数据里面估计得到的，所以它是存在误差的，如果我们的系统对这个误差是可以容忍的就还好，但系统对这个误差太敏感了，以至于我们的解的误差更大，那这个解就太不靠谱了。所以这个方程组系统就是ill-conditioned病态的，不正常的，不稳定的，有问题的，哈哈。这清楚了吧。右边那个就叫well-condition的系统了。

还是再啰嗦一下吧，对于一个ill-condition的系统，我的输入稍微改变下，输出就发生很大的改变，这不好啊，这表明我们的系统不能实用啊。你想想看，例如对于一个回归问题y=f(x)，我们是用训练样本x去训练模型f，使得y尽量输出我们期待的值，例如0。那假如我们遇到一个样本x'，这个样本和训练样本x差别很小，面对他，系统本应该输出和上面的y差不多的值的，例如0.00001，最后却给我输出了一个0.9999，这很明显不对呀。就好像，你很熟悉的一个人脸上长了个青春痘，你就不认识他了，那你大脑就太差劲了，哈哈。所以如果一个系统是ill-conditioned病态的，我们就会对它的结果产生怀疑。那到底要相信它多少呢？我们得找个标准来衡量吧，因为有些系统的病没那么重，它的结果还是可以相信的，不能一刀切吧。终于回来了，上面的condition number就是拿来衡量ill-condition系统的可信度的。condition number衡量的是输入发生微小变化的时候，输出会发生多大的变化。也就是系统对微小变化的敏感度。condition number值小的就是well-conditioned的，大的就是ill-conditioned的。

如果方阵A是非奇异的，那么A的conditionnumber定义为：

$$\kappa(A) = \| A \| \ \| A^{-1} \|$$

也就是矩阵A的norm乘以它的逆的norm。所以具体的值是多少，就要看你的norm是什么了。如果方阵A是奇异的，那么A的condition number就是正无穷大了。实际上，每一个可逆方阵都存在一个condition number。但如果要计算它，我们需要一个方阵的norm（范数）和Machine Epsilon（机器的精度）。为什么要范数？范数就相当于衡量一个矩阵的大小，我们知道矩阵是没有大小的，当上面不是要衡量一个矩阵A或者向量b变化的时候，我们的解x变化的大小吗？所以肯定得要有一个东西来度量矩阵和向量的大小吧？对了，他就是范数，表示矩阵大小或者向量长度。OK，经过比较简单的证明，对于AX=b，我们可以得到以下的结论：

$$\frac{\|\Delta x\|}{\|x\|} \le \|A\| \cdot \|A^{-1}\| \cdot \frac{\|\Delta b\|}{\|b\|} \qquad \frac{\|\Delta x\|}{\|x\|} \le K(A) \cdot \frac{\|\Delta b\|}{\|b\|} \qquad \frac{\|\Delta x\|}{\|x + \Delta x\|} \le K(A) \frac{\|\Delta A\|}{\|A\|}$$

也就是我们的解x的相对变化和A或者b的相对变化是有像上面那样的关系的，其中k(A)的值就相当于倍率，看到了吗？相当于x变化的界。

对condition number来个一句话总结：conditionnumber是一个矩阵（或者它所描述的线性系统）的稳定性或者敏感度的度量，如果一个矩阵的condition number在1附近，那么它就是well-conditioned的，如果远大于1，那么它就是ill-conditioned的，如果一个系统是ill-conditioned的，它的输出结果就不要太相信了。

好了，对这么一个东西，已经说了好多了。对了，我们为什么聊到这个的了？回到第一句话：从优化或者数值计算的角度来说，L2范数有助于处理 condition number不好的情况下矩阵求逆很困难的问题。因为目标函数如果是二次的，对于线性回归来说，那实际上是有解析解的，求导并令导数等于零即可得到最优解为：

$$\hat{\mathbf{w}} = \left(X^T X\right)^{-1} X^T \mathbf{y}$$

然而，如果当我们的样本X的数目比每个样本的维度还要小的时候，矩阵X^TX将会不是满秩的，也就是X^TX会变得不可逆，所以w*就没办法直接计算出来了。或者更确切地说，将会有无穷多个解（因为我们方程组的个数小于未知数的个数）。也就是说，我们的数据不足以确定一个解，如果我们从所有可行解里随机选一个的话，很可能并不是真正好的解，总而言之，我们过拟合了。

但如果加上L2规则项，就变成了下面这种情况，就可以直接求逆了：

$$w^* = \left(X^T X + \lambda I\right)^{-1} X^T y$$

这里面，专业点的描述是：要得到这个解，我们通常并不直接求矩阵的逆，而是通过解线性方程组的方式（例如高斯消元法）来计算。考虑没有规则项的时候，也就是λ=0的情况，如果矩阵X^TX的 condition number 很大的话，解线性方程组就会在数值上相当不稳定，而这个规则项的引入则可以改善condition number。

另外，如果使用迭代优化的算法，condition number 太大仍然会导致问题：它会拖慢迭代的收敛速度，而规则项从优化的角度来看，实际上是将目标函数变成λ-strongly convex（λ强凸）的了。哎哟哟，这里又出现个λ强凸，啥叫λ强凸呢？

当f满足：

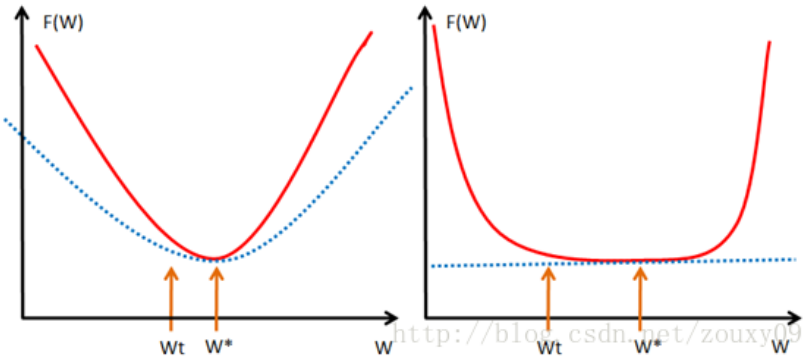
$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\lambda}{2} \| y - x \|^2$$

时，我们称f为λ-stronglyconvex函数，其中参数λ>0。当λ=0时退回到普通convex 函数的定义。

在直观的说明强凸之前，我们先看看普通的凸是怎样的。假设我们让f在x的地方做一阶泰勒近似（一阶泰勒展开忘了吗？f(x)=f(a)+f'(a)(x-a)+o(||x-a||).）：

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + o(\| y - x \|)$$

直观来讲，convex 性质是指函数曲线位于该点处的切线，也就是线性近似之上，而 strongly convex 则进一步要求位于该处的一个二次函数上方，也就是说要求函数不要太“平坦”而是可以保证有一定的“向上弯曲”的趋势。专业点说，就是convex 可以保证函数在任意一点都处于它的一阶泰勒函数之上，而strongly convex可以保证函数在任意一点都存在一个非常漂亮的二次下界quadratic lower bound。当然这是一个很强的假设，但是同时也是非常重要的假设。可能还不好理解，那我们画个图来形象的理解下。



大家一看到上面这个图就全明白了吧。不用我啰嗦了吧。还是啰嗦一下吧。我们取我们的最优解w*的地方。如果我们的函数f(w)，见左图，也就是红色那个函数，都会位于蓝色虚线的那根二次函数之上，这样就算w_t和w*离的比较近的时候，$f(w_t)$和f(w*)的值差别还是挺大的，也就是会保证在我们的最优解w*附近的时候，还存在较大的梯度值，这样我们才可以在比较少的迭代次数内达到w*。但对于右图，红色的函数f(w)只约束在一个线性的蓝色虚线之上，假设是如右图的很不幸的情况（非常平坦），那在w_t还离我们的最优点w*很远的时候，我们的近似梯度$(f(w_t)-f(w^*))/(w_t-w^*)$就已经非常小了，在$w_t$处的近似梯度$\partial f/\partial w$就更小了，这样通过梯度下降$w_{t+1}=w_t-\alpha^*(\partial f/\partial w)$，我们得到的结果就是w的变化非常缓慢，像蜗牛一样，非常缓慢的向我们的最优点w*爬动，那在有限的迭代时间内，它离我们的最优点还是很远。

所以仅仅靠convex 性质并不能保证在梯度下降和有限的迭代次数的情况下得到的点w会是一个比较好的全局最小点w*的近似点（插个话，有地方说，实际上让迭代在接近最优的地方停止，也是一种规则化或者提高泛化性能的方法）。正如上面分析的那样，如果f(w)在全局最小点w*周围是非常平坦的情况的话，我们有可能会找到一个很远的点。但如果我们有“强凸”的话，就能对情况做一些控制，我们就可以得到一个更好的近似解。至于有多好嘛，这里面有一个bound，这个 bound 的好坏也要取决于strongly convex性质中的

常数α的大小。看到这里，不知道大家学聪明了没有。如果要获得strongly convex怎么做？最简单的就是往里面加入一项(α/2)*||w||^2。

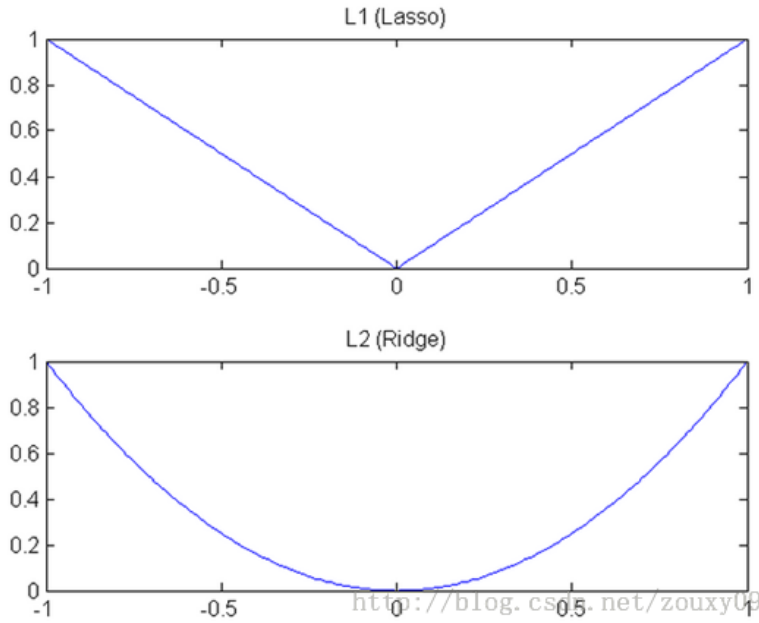
呃，讲个strongly convex花了那么多的篇幅。实际上，在梯度下降中，目标函数收敛速率的上界实际上是和矩阵X^TX的 condition number有关，X^TX的 condition number 越小，上界就越小，也就是收敛速度会越快。

这一个优化说了那么多的东西。还是来个一句话总结吧：L2范数不但可以防止过拟合，还可以让我们的优化求解变得稳定和快速。

好了，这里兑现上面的承诺，来直观的聊聊L1和L2的差别，为什么一个让绝对值最小，一个让平方最小，会有那么大的差别呢？我看到的有两种几何上直观的解析：

1）下降速度：

我们知道，L1和L2都是规则化的方式，我们将权值参数以L1或者L2的方式放到代价函数里面去。然后模型就会尝试去最小化这些权值参数。而这个最小化就像一个下坡的过程，L1和L2的差别就在于这个“坡”不同，如下图：L1就是按绝对值函数的“坡”下降的，而L2是按二次函数的“坡”下降。所以实际上在0附近，L1的下降速度比L2的下降速度要快。所以会非常快得降到0。不过我觉得这里解释的不太中肯，当然了也不知道是不是自己理解的问题。



L1在江湖上人称Lasso，L2人称Ridge。不过这两个名字还挺让人迷糊的，看上面的图片，Lasso的图看起来就像ridge，而ridge的图看起来就像lasso。

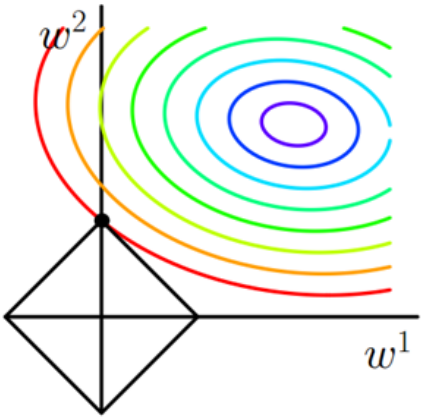
2）模型空间的限制：

实际上，对于L1和L2规则化的代价函数来说，我们可以写成以下形式：

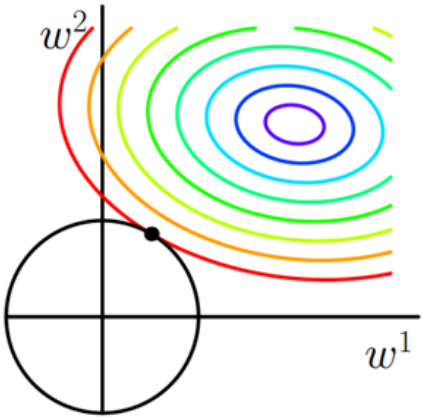
$$Lasso: \quad \min_w \frac{1}{n} \| y - Xw \|^2, \quad s.t. \| w \|_1 \le C$$

$$Ridge: \quad \min_w \frac{1}{n} \| y - Xw \|^2, \quad s.t. \| w \|_2 \le C$$

也就是说，我们将模型空间限制在w的一个L1-ball 中。为了便于可视化，我们考虑两维的情况，在(w1, w2)平面上可以画出目标函数的等高线，而约束条件则成为平面上半径为C的一个 norm ball 。等高线与 norm ball 首次相交的地方就是最优解：



(a) ℓ_1-ball meets quadratic function. ℓ_1-ball has corners. It's very likely that the meet-point is at one of the corners.



(b) ℓ_2-ball meets quadratic function. ℓ_2-ball has no corner. It is very unlikely that the meet-point is on any of axes."

可以看到，L1-ball 与L2-ball 的不同就在于L1在和每个坐标轴相交的地方都有“角”出现，而目标函数的测地线除非位置摆得非常好，大部分时候都会在角的地方相交。注意到在角的位置就会产生稀疏性，例如图中的相交点就有w1=0，而更高维的时候（三维的L1-ball 是什么样的？）除了角点以外，还有很多边的轮廓也是既有很大的概率成为第一次相交的地方，又会产生稀疏性。

相比之下，L2-ball 就没有这样的性质，因为没有角，所以第一次相交的地方出现在具有稀疏性的位置的概率就变得非常小了。这就从直观上来解释了为什么L1-regularization能产生稀疏性，而L2-regularization 不行的原因了。

因此，一句话总结就是：L1会趋向于产生少量的特征，而其他的特征都是0，而L2会选择更多的特征，这些特征都会接近于0。Lasso在特征选择时候非常有用，而Ridge就只是一种规则化而已。

OK，就聊到这里。下一篇博文我们聊聊核范数和规则化项参数选择的问题。全篇的参考资料也请见下一篇博文，这里不重复列出。谢谢。

参考知识库

机器学习知识库

12967 关注 | 1997 收录

.NET知识库

1617 关注 | 800 收录

软件测试知识库

2155 关注 | 290 收录

算法与数据结构知识库

9631 关注 | 2263 收录

猜你在找

统计机器学习入门——线性模型选择与正则化2

统计机器学习入门——线性模型选择与正则化1

Python自动化测试MonkeyRunner

Android自动化测试第二季（提高篇）

Android自动化测试第三季

机器学习中的范数规则化之一L0L1与L2范数

机器学习中的范数规则化之一L0L1与L2范数

读机器学习中的范数规则化之一L0L1与L2范数笔记

机器学习中的范数规则化之一L0L1与L2范数

机器学习中的范数规则化之一L0L1与L2范数

查看评论

57楼 ShengkeXue 前天 23:23发表

lz 分析得很详细，内容也不死板，读了之后受益颇多！

56楼 lyjcyyz123 2016-10-23 09:23发表

奔波在英文文献的数学证明中不能自拔，每次看到让人豁然开朗的文章都是出自楼主的，哎哎哎哎，点个赞

55楼 春夏秋冬又一年 2016-10-09 09:56发表

楼主，模型空间限制这里的 Lasso的公式 是不是应该是一次方而不是二次方？

54楼 skyeagles123 2016-09-29 16:47发表

文风幽默而不失严谨，数学苦手的福音

53楼 wgd852372 2016-09-14 11:27发表

小伙子，不错啊

52楼 everyday-new 2016-08-25 23:21发表

楼主，写的太棒了，能否转载一下，以备以后经常回顾

51楼 alifuyou 2016-08-23 11:27发表

讲的通俗易懂，谢谢。

50楼 gladys132013 2016-08-22 22:12发表

讲得太棒啦！我翻了几篇论文都快整疯了，楼主一会儿就讲明白了，特别顺畅！感觉您言语间特别能理解初学者的心情！特地在这注册了个账号给您点赞，棒棒哒！

49楼 qq_821878768 2016-08-04 10:53发表

写的实在是太好了，解决我很大的问题，，万分感谢分享

48楼 Lui_madfrog 2016-06-13 15:45发表

个人的浅显理解，正规项的加入限制了模型参数空间的选择，从而降低了模型的VC维，进而可以认为是降低了模型的复杂度。最终达到降低由于数据过少或者噪声影响引起的的过拟合。

47楼 wanglinhua615 2016-06-10 22:28发表

谢谢楼主，挺好的~

46楼 sinat_35178418 2016-06-07 23:18发表

您好我想请问一下有没有用来参考的原文什么的？我看有很多英文的地方，所以想问一下是否有英文原文？

45楼 追寻梦土 2016-06-02 16:05发表

谢谢楼主，好博文!!!!!

44楼 qq_25297013 2016-06-01 21:19发表

方阵的condition number应该是这个方阵的最大特征值和最小特征值的比，跟博主说的其实是一样的（矩阵的逆矩阵的特征值编程之前的倒数），不过这样好理解一点吧。

43楼 载道2011 2016-03-30 16:27发表

对于初学者的我感觉也不是很难,通俗易懂!感谢博主!!!

42楼 IT_Shero 2016-03-21 17:17发表

您的讲解真幽默，寓教于乐，赞

41楼 修行中的麻雀 2016-03-10 22:17发表

楼主可否再具体讲一下L0范数和L1范数呢，最好可以给个例子解释一下啊，在下理解力尚浅，还请多指教啊

40楼 hustlx 2016-02-29 22:55发表

好厉害

39楼 wa2003 2016-02-22 10:41发表

牛文啊，猴赛雷！

38楼 stcdamkw 2016-02-06 17:48发表

讲得很棒，特地登录一下给你个赞。

37楼 qq_33845637 2016-01-25 14:18发表

太棒了，我终于懂=老师讲的是什么了。以后还是看你的博客好了

36楼 这一刻就出发 2016-01-21 11:48发表

楼主您好，我想问一下你在最后用可视化方式解释L1 能解决稀疏性，而L2不擅长解决稀疏性这一块儿，所说的L1-ball与损失函数在角点相交的概率大，且这个位置会产生稀疏性，这里的稀疏性指的是W1=0的意思吗？ 如果是的话，那么L1是高维的时候，“很多边的轮廓也可能成为第一次相交的地方，也会产生稀疏性”，怎么就敢确定这些轮廓对应的w1坐标是0呢？ 我对这里相交产生稀疏性还不是很理解。

35楼 u010395786 2015-12-09 20:05发表

楼主把l0、l1、l2规则约束解释的非常好。我遇到一个函数拟合的问题：目标函数是明确的，基函数形式也是固定的，现在想用尽可能少的基函数拟合目标函数。我使用的是OMP挑选权重大的基函数。使用OMP算法得到的解有时候会不稳定。请问我的这种做法是否正确，如果转化为l1约束的优化问题能否得到更好的解。请指点，感激不尽！

Re: 摇扇子的诸葛亮 2016-01-12 17:13发表

回复u010395786：omp算法结果与迭代次数有关，同时与信号稀疏度也有关。想获得稳定的结果，需要对字典的选择和目标函数做改进

34楼 有来有去-CV 2015-11-15 17:30发表

赞一个，写的很好很详细，大部分懂了，有些细节没搞明白。

33楼 SanShuZhiChuMen 2015-11-14 02:56发表

> 引用“anwenxixi”的评论：
>
> 你好，弱弱的问一下||w||式子中2作为下标时表示的是L2范式，那2作为上标时表示的是什么意思啊？

平方

32楼 fairytale12 2015-08-06 17:26发表

"而越小的参数说明模型越简单，越简单的模型则越不容易产生过拟合现象。"，请问出自哪里呢？

31楼 凝香沁雪 2015-07-20 15:17发表

您好，我研究方向也是模式识别，请问如果目标函数中存在矩阵2范数求和应该怎么解？如何将双求和的向量2范数化成矩阵的形式？

30楼 四 2015-06-06 09:48发表

讲得很棒！但有一个问题不太明白，希望大家能帮忙指点一下。这句话：
“而目标函数的测地线除非位置摆得非常好，大部分时候都会在角的地方相交。”是为什么呢？
谢谢！

Re: Sunshine_in_Moon 2015-09-09 10:30发表

回复四：除非相切

Re: 景语 2015-08-05 09:35发表

回复四：从图上可以直观理解，矩形的顶点总是四个方向上绝对值最高的，因此最可能与等高线相交的就是这些顶点

29楼 u010716993 2015-05-27 14:22发表

前辈 麻烦问一下SVM分类器如何读取训练集和测试集 测试集的格式是什么 您有代码么 谢谢！

28楼 初入Cplusplus 2015-05-22 15:56发表

很好 正好学习中

27楼 相门码农 2015-04-26 19:36发表

赞楼主~

26楼 _KDH 2015-04-25 16:05发表

后面很直观。！！！
为什么那个加了正则化项之后就会变成强凸的。

25楼 liufeng_cp 2015-04-22 13:59发表

参数越小 曲面越平滑 越平滑的曲面越“大条”（即丢失细部，与L1去除一些细部维度有点殊途同归） 所以说L2也能简化模型

24楼 liufeng_cp 2015-04-22 13:48发表

L1代表空间的维度小；L2代表model在空间中曲面的平滑度

23楼 mazhiran-persistence 2015-03-31 18:48发表

请问，模型空间限制那个地方，'目标函数的等高线'如何理解？

Re: guet_qhr 2015-04-03 10:45发表

回复mazhiran-persistence：我的理解是不带约束目标函数值。

22楼 jiafeimao1991 2015-03-30 21:20发表

楼主你好。我刚接触范数，上述大部分的内容都觉得讲的非常的精彩，只是有点不明白的是在范数的模型函数与价值函数的等高线平面内相交的坐标系中W1和W2是什么意思？还有。范数的下降模型的坐标系也不是很懂，为什么收敛点的坐标为0呢？

21楼 西西大爷 2015-03-24 10:31发表

你好，弱弱的问一下||w||式子中2作为下标时表示的是L2范式，那2作为上标时表示的是什么意思啊？

20楼 晦光 2015-02-03 00:00发表

土问lz l1、l2-ball图是哪里引用的，想去看看原著加强下理解，谢谢

Re: xuweimdm 2015-02-26 14:04发表

回复晦光：PRML中P146有类似的图

19楼 小商1989 2015-01-25 11:23发表

L2范数最开始的意思应该不是让参数小，而是把参数w限制在某个范围内：min（loss function L），约束条件 ||w||2 < M;用数学转成等价形式为min(L+入||w||2)。从新的形势看就成了参数小则model简单。 从回归理解，加入L2范数使得可逆有解；从线性分类器理解，w越小，y=wx+b，x变化很大，y才变化一点点，更stable，更general。

18楼 heshuangping 2015-01-12 10:45发表

“XTX的 condition number 越小，上界就越小，也就是收敛速度会越快”
这里应该是条件数越小，收敛速率上界就越大，收敛速度就越快

另外 那张表示L1和L2的规则化作用的图片是哪本书里面的呢

17楼 王道的博客 2014-11-23 13:54发表

多谢楼主的解析。。。对L1 L2 等相关知识有了初步的认识！

16楼 chuminnan2010 2014-11-20 21:01发表

实际上，对于L1和L2规则化的代价函数来说，我们可以写成以下形式：{是不是有问题，L1是不是没有平方啊？？？？}

15楼 CJZ_WORK 2014-11-06 11:18发表

请问：为什么在norm ball中，第一次相交在角的位置就会出现稀疏性？是因为相交在角的位置，会使得其他维上的映射为0吗？轮廓位置相交又为什么产生稀疏性。

Re: 景语 2015-08-05 09:42发表

回复CJZ_WORK：w=0，相当于这一维的特征就没用了，因此用于特征选择时可以产生稀疏

14楼 CJZ_WORK 2014-11-06 11:18发表

请问：为什么在norm ball中，第一次相交在角的位置就会出现稀疏性？是因为相交在角的位置，会使得其他维上的映射为0吗？轮廓位置相交又为什么产生稀疏性。

13楼 didiwai1990 2014-10-24 21:38发表

请问博主能不能把一些参考资料列出来，多谢了！

12楼 huagong_adu 2014-10-24 21:10发表

讲的挺好的，很通俗，赞楼主

11楼 迷雾forest 2014-10-17 10:29发表

“上面的图是线性回归，下面的图是Logistic回归，也可以说是分类的情况。从左到右分别是欠拟合（underfitting，也称High-bias）、合适的拟合和过拟合（overfitting，也称High variance）三种情况。可以看到，如果模型复杂（可以拟合任意的复杂函数），它可以让我们的模型拟合所有的数据点，也就是基本上没有误差。对于回归来说，就是我们的函数曲线通过了所有的数据点，如上图右。对分类来说，就是我们的函数曲线要把所有的数据点都分类正确，如下图右。这两种情况很明显过拟合了”
这里面上面的图不是线性回归问题，被称为“多项式回归”更合适一些

10楼 vencevc 2014-10-08 10:27发表

问个问题：L1范数的w参数要如何求解？

9楼 xyy19920105 2014-08-12 10:32发表

博主大牛啊，这个内容是挺充分的，只是有些东西看的感觉有些不对啊.....望博主回头没事多看看，改掉些小瑕疵.....

8楼 chenzhong2006 2014-07-23 10:25发表

请问一下：强凸与严格凸是一样的吗？如果不是一样，那它们的区别是？我理解的是一样的。。。

7楼 chenzhong2006 2014-07-23 09:43发表

讲得特别好！还是有一个疑问哈：您说让L2范数的规则项||W||2最小，可以使得W的每个元素都很小，都接近于0，但与L1范数不同，它不会让它等于0，而是接近于0，这里是有很大的区别。其实L2范数和L1范数都应该是使得W的每个元素都很小，都接近于0，但是L0范数应该是等于0吧。这是他们的区别？

6楼 tianhan4 2014-07-12 22:54发表

我也是学习人工智能和机器学习这块的，本科的时候做的是kinect相关开发，现在正在学楼主看过的python机器学习实战，正在广州准备读研。感觉看楼主的文章受益匪浅又有很多共鸣，佩服楼主学习的毅力和效率，共勉

5楼 CanaanShen 2014-07-06 14:38发表

受教了，非常感谢

4楼 motoyule 2014-07-06 08:43发表

楼主怎么没有提到方程的不适定性呢？从数学角度来看，正则化为解不适定问题的手段。我个人觉得，太多的“个人理解”不如好好找找数学角度的解释。

3楼 Joey_Tang 2014-05-20 10:48发表

博主，我想问问L2的规则项 lamta * I，这个I矩阵是单位矩阵吗？！

2楼 salan668 2014-05-05 17:12发表

“越小的参数说明模型越简单”，我觉得您之后的理解是对的，但简单的模型应该是参数少，而不是参数小。我的理解是参数小说明它的“抗干扰”能力强，也就能够较好的去预测样本的准确度。

Re: zouxy09 2014-05-06 22:33发表

回复salan668：您好，其实这一点我也不太明白的。不过您这么一说似乎也蛮有道理的。想请问下您的这个解释有来源吗？或者您知道哪里有相关的数学证明吗？

Re: super_vision 2015-03-16 13:33发表

回复zouxy09： 参数太大的一个问题就是一个很小的扰动, 就会产生一个较大的函数值的变化, 抗干扰能力较差. 可以参考一下PRML中第一章关于线性回归过拟合问题的讲解, 好像书中给出了不同的参数值下的拟合结果, 过拟合的时候参数值非常大了就

Re: Uncle_Joke 2015-01-07 19:39发表

回复zouxy09：关于“越小的参数模型越简单”，我的理解是，过拟合时，曲线表现为大致经过每个样本点，对于曲线上的很小的区间往往弯曲度十分大（没法上图，不直观呀），这间接的说明了在该区间的导数很大，对线性回归来说，权值就是导数，所以越小的参数模型越简单。不知道这样理解对不对

Re: nyyznyyz1991 2016-04-27 21:38发表

回复Uncle_Joke："越小的参数越简单“ 这句话我有点疑惑，w1*x1+w2*x2+b=0我如果同时扩大2倍，2w1*x1+2w2*x2+2b=0很明显复杂度是一样的，但是参数都扩大了两倍；另外w1，w2在优化问题内更多的是构成了梯度的大小和方向

Re: xuweimdm 2015-02-25 23:27发表

回复Uncle_Joke：我觉得你这个解释有道理，并且同时也可以解释抗干扰能力强这个特点。但我的另一个观点是：其实对于模型简单这个特点来说，L2不应该和L1比较，或者说这两种选择下得到的模型都简单，L1倾向于选择使得特征数目较少，对于多项式函数多项式次数比较低，这当然是一个简单的模型。而对于L2来说，多项式次数虽然高了，但是它的每一项系数非常小，所以在每一个样本点附近的导数（该导数代表了模型也就是方差）同样可以小，因此对于过拟合问题也有很好的预防能力。所以，我的结论是：在过拟合这点上，L1和L2无法进行比较，各自在其他方面有不同的侧重点而已。

Re: liufeng_cp 2015-04-22 13:48发表

回复xuweimdm：赞同你的观点

1楼 silence1214 2014-05-04 14:59发表

讲的挺好的，只是有些理论挺复杂的还是难懂，不过看了后有了个直观的感觉。正则化本来就是个很复杂的事情。其实现在对于大数据的情况下，正则项可以忽略的

Re: cuilipengpeng 2015-05-25 16:19发表

回复silence1214：在大数据下正则化项可以忽略。。。？为啥。。。？你是指的大数据环境下样本观测值很多，不会产生病态方程组问题。。。。？

Re: wx20101005060708 2015-05-18 15:42发表

回复silence1214：您好，我想向您请教一下，为什么大数据的情况下，正则项可以忽略的？？麻烦回答，感激不尽。

Re: Coastchb 2016-09-26 14:40发表

回复wx20101005060708：参数更新的真正目标是最小化期望损失函数，即让每一种输入输出对应的损失函数的期望值达到最小，这样得到的模型是最佳的；但是实际上我们只是以最小化经验损失函数（所有训练样本的平均损失函数）来更新参数了。如果训练数据很小，那经验损失函数和期望损失函数相差很多，偏离了真正目标，学出来的模型只是能够很好拟合训练数据那个范围内的样本而不能应用到其他样本（即过拟合）；如果训练数据够大，（极端情况下，可以覆盖所有可能的输入输出），那经验损失函数和期望损失函数就很接近了，学出来的模型就能应用到更多的样本。

Re: whxtbest 2015-09-13 16:36发表

回复wx20101005060708：我个人的理解是在小数据集上，样本对概率空间的描述可能不完整（样本缺失），所以容易导致在训练集上的过拟合。而大数据集下，全部的样本可以很好的描述概率空间，只要保证能合理的抽样得到训练集，可以最大的避免过拟合。简单的说，大数据集下可以从训练集的选择上去避免过拟合，而非从模型上。所以正则项变的没那么重要。