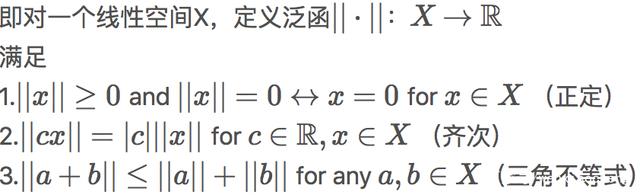
**向量的范数：**

向量范数是定义了向量的类似于长度的性质，满足正定，齐次，三角不等式的关系就称作范数。

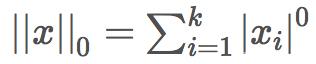


向量的范数一般有L0, L1, L2与L\_infinity范数,



# L0范数:

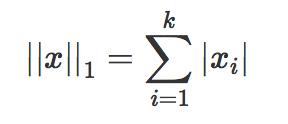
定义为



即非0元素个数。

L0范数表示向量中非零元素的个数。如果我们使用L0来规则化参数向量w，就是希望w的元素大部分都为零。L0范数的这个属性，使其非常适用于机器学习中的稀疏编码。在特征选择中，通过最小化L0范数来寻找最少最优的稀疏特征项。但是，L0范数的最小化问题是NP难问题。而L1范数是L0范数的最优凸近似，它比L0范数要更容易求解。因此，优化过程将会被转换为更高维的范数（例如L1范数）问题。

# L1范数:



L1范数是向量中各个元素绝对值之和，也被称作“Lasso regularization”（稀疏规则算子）。

在机器学习特征选择中，稀疏规则化能够实现特征的自动选择。一般来说，输入向量X的大部分元素（也就是特征）都是和最终的输出Y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，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。

L0范数与L1范数都可以实现稀疏，而L1范数比L0具有更好的优化求解特性而被广泛使用。 L0范数本身是特征选择的最直接的方案，但因为之前说到的理由，其不可分，且很难优化，因此实际应用中我们使用L1来得到L0的最优凸近似。

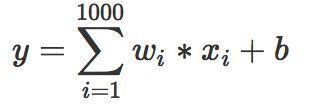
总结一下上两段的结论就是：L1范数和L0范数可以实现稀疏，L1因为拥有比L0更好的优化求解特性而被广泛应用。这样我们大概知道了可以实现稀疏，但是为什么我们希望稀疏？让参数稀疏有什么好处呢？这里有两个理由：

1）特征选择(Feature Selection)：

大家希望稀疏规则化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的大部分元素（也就是特征）都是和最终的输出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。

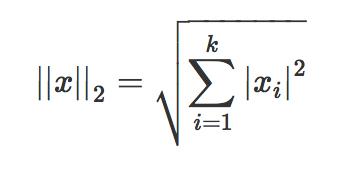
2）可解释性(Interpretability)：

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。假设这是个回归模型：



当然了，为了让y限定在的范围，一般还得加个Logistic函数。 通过学习，如果最后学习到的就只有很少的非零元素，例如只有5个非零的，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，患不患这种病只和这5个因素有关，那医生就好分析多了。但如果1000个都非0，医生面对这1000种因素只能一脸懵逼不知如何是好。

# L2范数



L2范数是最常用的范数了，我们用的最多的度量距离欧氏距离就是一种L2范数。在回归里面，有人把加了L2范数项的回归c称为“岭回归”（Ridge Regression），有人也叫它“权值衰减weight decay”。它被广泛的应用在解决机器学习里面的过拟问题合。

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范数实际上是什么。

L2范数是指向量各元素的平方和然后求平方根。我们让L2范数的规则项最小，可以使得的每个元素都很小，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，这是有很大的区别的。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？因为当限制了参数很小，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。

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。

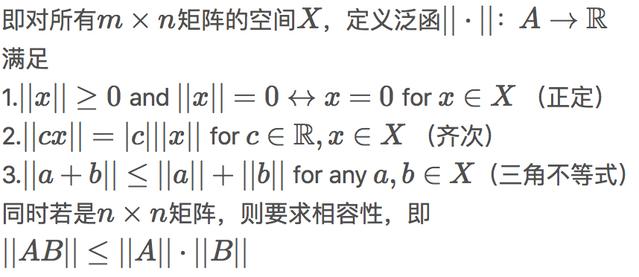
L2范数的好处是什么呢？

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。

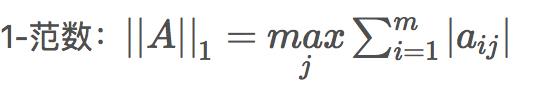
2）优化计算的角度： 从优化或者数值计算的角度来说，L2范数有助于处理 condition number不好的情况下矩阵求逆很困难的问题。关于condition number的知识参见这篇文章：https://blog.csdn.net/zouxy09/article/details/24972869

# 矩阵的范数：

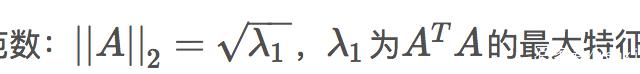
矩阵范数是定义了矩阵的类似于长度的性质，与向量范数不同，他要满足额外的性质：



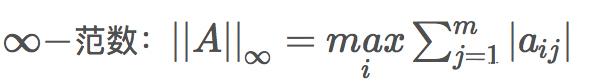
矩阵范数一般有1-, 2-, infinity-, F-范数



列和范数，即所有矩阵列向量绝对值之和的最大值

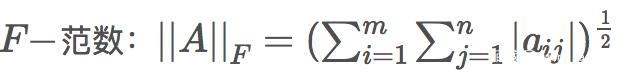


谱范数，即ATA矩阵的最大特征值的开平方。

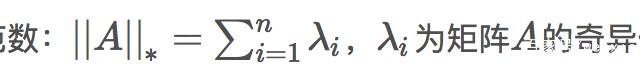


行和范数，即所有矩阵行向量绝对值之和的最大值。以上这些都是诱导范数，由向量的Lp范数诱导而来。非诱导范数常见的为F-范数，即Frobenius范数以及核范数。

行和范数，即所有矩阵行向量绝对值之和的最大值，matlab调用函数norm(A, inf)。 以上这些都是诱导范数，由向量的范数诱导而来。



Frobenius范数，即矩阵元素绝对值的平方和再开平方



核范数英文称呼叫Nuclear Norm。这个相对于上面火热的L1和L2来说，可能大家就会陌生点。那它是干嘛用的呢？约束Low-Rank（低秩）

秩可以度量相关性，而矩阵的相关性实际上有带有了矩阵的结构信息。如果矩阵之间各行的相关性很强，那么就表示这个矩阵实际可以投影到更低维的线性子空间，也就是用几个向量就可以完全表达了，它就是低秩的。所以我们总结的一点就是：如果矩阵表达的是结构性信息，例如图像、用户-推荐表等等，那么这个矩阵各行之间存在着一定的相关性，那这个矩阵一般就是低秩的。

如果A是一个m行n列的数值矩阵，rank(A)是X的秩，假如rank(A)远小于m和n，则我们称X是低秩矩阵。低秩矩阵每行或每列都可以用其他的行或列线性表出，可见它包含大量的冗余信息。利用这种冗余信息，可以对缺失数据进行恢复，也可以对数据进行特征提取。

好了，低秩有了，那约束低秩只是约束rank(A)呀，和我们这节的核范数有什么关系呢？他们的关系和L0与L1的关系一样。因为rank()是非凸的函数，在优化问题里面很难求解，那么就需要寻找它的凸近似来近似它了。对,rank(A)的凸近似就是核范数。