1. 稀疏矩阵
2. 鲁棒性
3. L1和l2正则化的区别

简单来说也就是范数其实在 [0,∞)范围内的值，是向量的投影大小，在机器学习中一般会用于衡量向量的距离。范数有很多种，我们常见的有L1-norm和L2-norm，其实还有L3-norm、L4-norm等等，所以抽象来表示，我们会写作Lp-norm，一般表示为

$$||x||_p = (\sum_i |x_i|^p)^{1/p}$$

:

对于上面这个抽象的公式，如果我们代入p值，

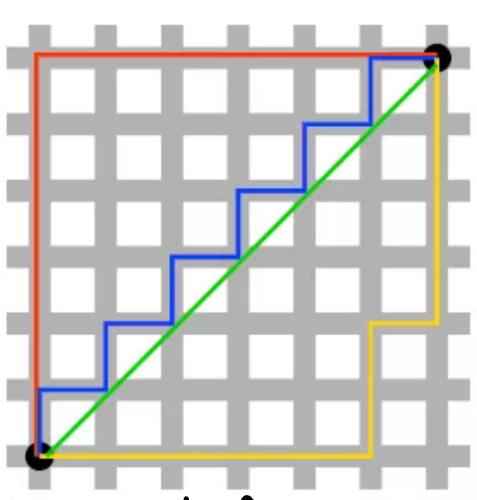
若p为1，则就是我们常说的L1-norm：

$$||x||_1 = \sum_i |x_i| = |x_1| + |x_2| + \ldots + |x_i|$$

若p为2，则是我们常说的L2-norm：

$$||x||_2 = \sqrt{(\sum_i |x_i|^2)} = \sqrt{x1^2 + x2^2 + \ldots + x_i^2}$$

L2-norm的距离就是两个黑点之间的绿线，而另外的3条线，都是L1-norm的大小。



(x_2, y_2)

(x_1, y_1)

绿线 $\sqrt{(y_2-y_1)^2+(x_2-x_1)^2} = \sqrt{z_1^2+z_2^2}$

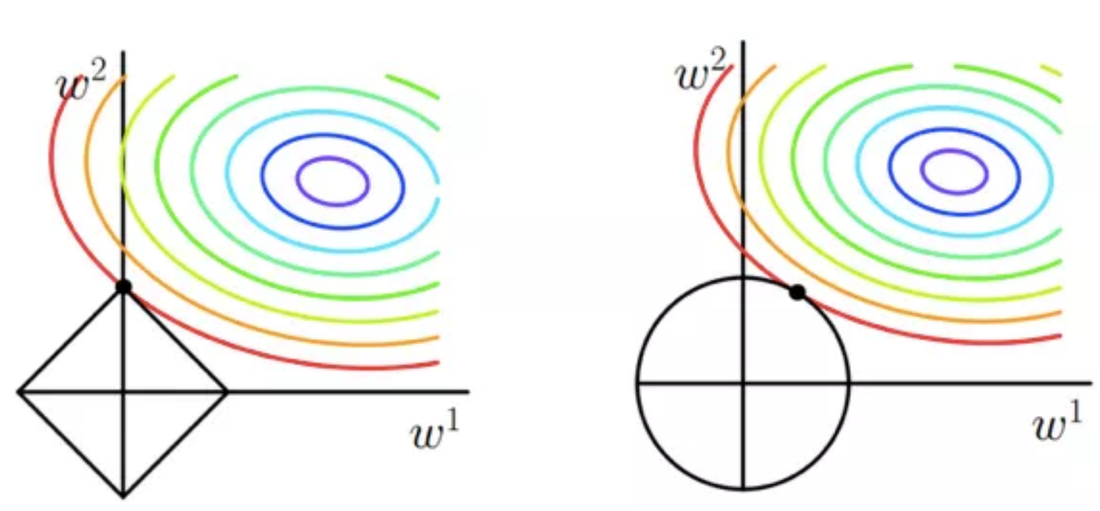
红线 $|(x_2-x_1)|+|(y_2-y_1)| = |z_1|+|z_2|$

绿线与黄线只是红线的变形，总的距离不变。

Q1：L1和L2正则化项的区别？

首先，我们从上面那张二维的图可以看出，对于L2-norm，其解是唯一的，也就是绿色的那条；而对于L1-norm，其解不唯一，因此L1正则化项，其计算难度通常会高于L2的。

其次，**L1通常是比L2更容易得到稀疏输出的**，会把一些不重要的特征直接置零，至于为什么L1正则化为什么更容易得到稀疏解，可以看下图：



上图代表的意思就是目标函数-平方误差项的等值线和L1、L2范数等值线（左边是L1），我们正则化后的代价函数需要求解的目标就是在经验风险和模型复杂度之间的平衡取舍，在图中形象地表示就是黑色线与彩色线的交叉点。

对于L1范数，其图形为菱形，二维属性的等值线有4个角（高维的会有更多），"突出来的角"更容易与平方误差项进行交叉，而这些"突出来的角"都是在坐标轴上，即W1或则W2为0；

而对于L2范数，交叉点一般都是在某个象限中，很少有直接在坐标轴上交叉的。

因此L1范数正则化项比L2的更容易得到稀疏解。

Q2：各有什么优势，如何作选择？

直接上结论：

1）因为L1范数正则化项的"稀疏解"特性，L1更适合用于特征选择，找出较为"关键"的特征，而把一些不那么重要的特征置为零。

2）L2范数正则化项可以产生很多参数值很小的模型，也就是说这类的模型抗干扰的能力很强，可以适应不同的数据集，适应不同的"极端条件"。

🤔 **如何作为Loss Function**

讲完了作为正则化项的内容了，那么讲讲L1、L2范数作为损失函数的情况。假设我们有一个线性回归模型，我们需要评估模型的效果，很常规的，我们会用"距离"来衡量误差！

若使用L1-norm来衡量距离，那就是我们的LAD（Least Absolute Deviation，最小绝对偏差），其优化的目标函数如下：

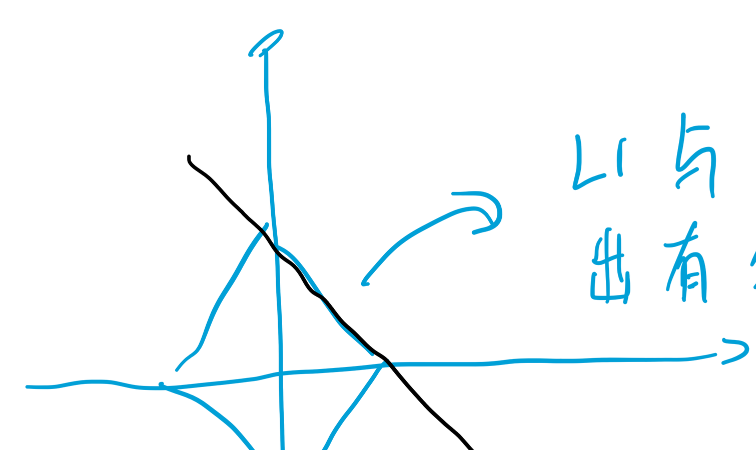
实际意义上的解释就是预测值与真实值之间的绝对值。

若使用L2-norm，那就是我们的LSE（Least Squares Error，最小二乘误差），其优化的目标函数如下：

针对两者的差异，可以看下表：

L1损失函数的结果更具鲁棒性，也就是说对于异常值更加不敏感。而根据其范数"天性"，L2的求解更为简单与"唯一"。

1. L1正则化并不是总是输稀疏解：

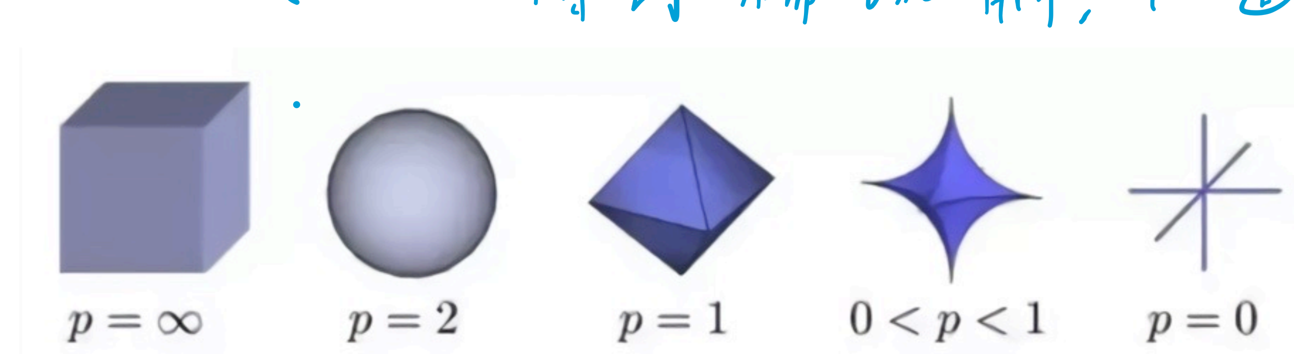


L1与线的共同部份即是解，可以看出有多个解，且不是稀疏解。

2. L1更容易得到稀疏解：

因为L1有"刺"，高维度则刺更多，碰到等值线的概率更高，L_2碰到等高线的点很少是稀疏解，因为是在像限中。

3. $0<P<1$ 更容易得到稀疏解，但是计算复杂。



4. sparse solution：

Once a variable has a 0 coefficient, it has no impact on the model anymore, the model uses fewer and fewer variables. This is what we mean by a sparse solution - **it only uses a few variables in the dataset**.

Other methods may produce a solution where many variables have small, but non-zero coefficients. These models are not sparse, since you still need all the variables to produce the solution. A sparse solution is generally preferred since you can explain your model in terms of just a few variables.