## Problem Set 2

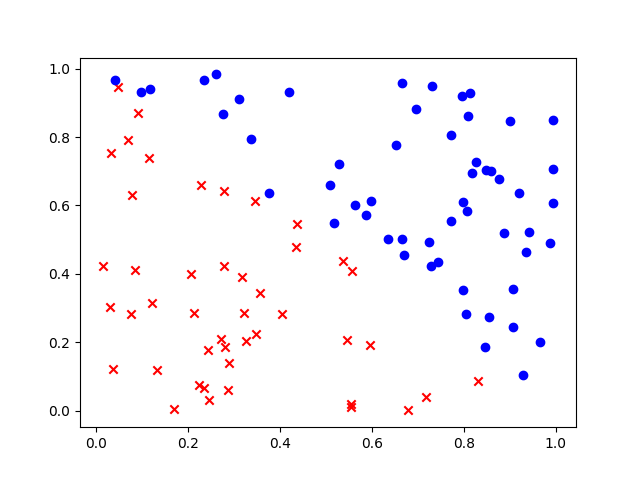
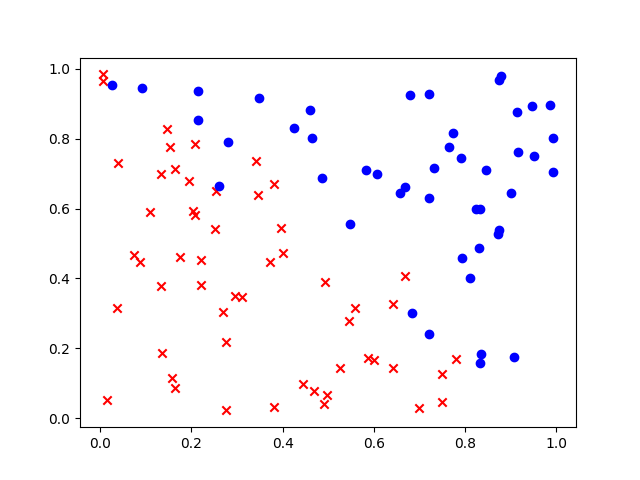
## SVM分类器中的取值问题

而对于最后一个等式，我们希望尽可能大，当y和x固定后，由于对不加以任何限制，导致可以通过改变theta的取值来使得这个等式取值变大。

由于预测的label是，我们自然希望预测出来的数越远离0的边界越好，那么某个下得到了某个预测值 ，很自然地，当的时候,由于是线性回归， ，实际上这两个值在预测上没有区别。

因为回归学习到的是该分界线的法向量，最后均需要进行单位化，和没有任何区别。

对于线性可分的两组数据点，如下图右，总是可以通过更改的单位来不断降低损失；对于下图左而言，并不是完全显性可分的，就必然存在一定的误分，这一部分误分导致一定是负数，随意改动的单位可能也会使得这部分误分的损失加大，从而限制了这种随意性。



一些可能的改进方法：

1. 调整学习速率为其他常数

不会影响上述对方向向量改变的影响

1. 随着迭代次数减小学习速率

当学习速率更新变慢后，相当于限制了的变化，当数量级足够小，则可以使原函数收敛。

1. 对原始特征进行线性放缩

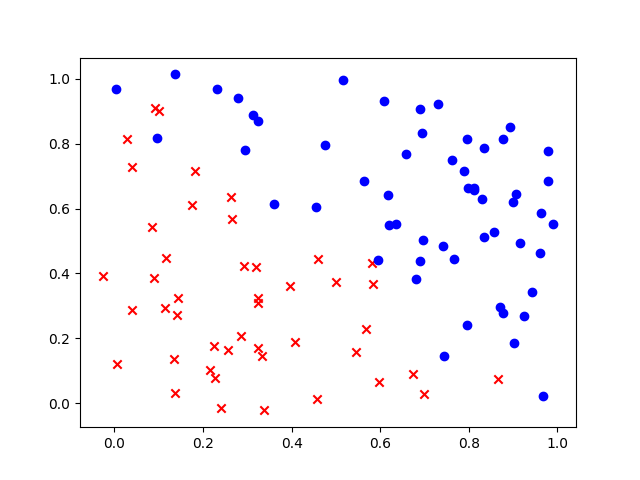
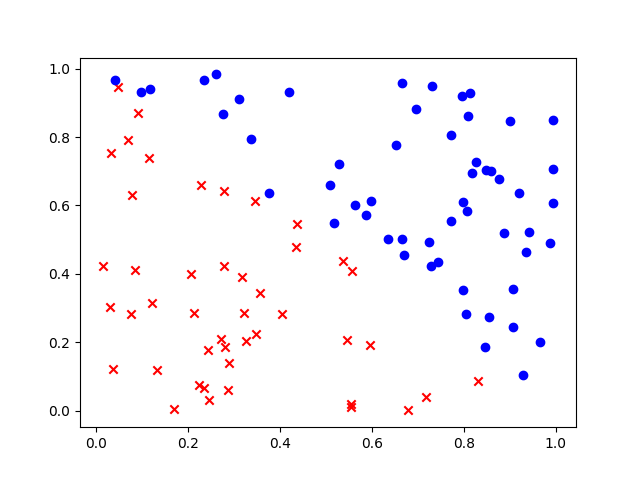
不会影响

1. 增加对的限制，如

可以！

1. 想原始数据增加一个均值为0的高斯噪声。

可以的，但是要看原始图像的的margin如何，如下图，如果加完高斯噪声后适当的让其变得线性不可分，则能够抵消掉对方向向量的线性变换的影响。



## SVM中的Hinge Loss问题

Intuition: 对于二分类问题：，显然，当预测值的情况，我们有很大的把握认为预测准确，但是当的时候，把握没有这么大，可能存在一定的误分类，hinge loss就是为了刻画这种损失。

即有：

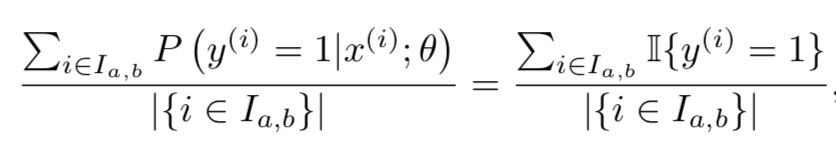
此时，对于线性可分的数据集，是否还存在上述的对方向向量进行线性变更导致损失函数减小的情形呢？应当是不存在了。如果不断的调整，则会导致后项不断变小，但是当小到一定程度（为负数）后，该项损失将收到0的约束，故而不会继续影响损失。

## 模型标定

well-calibrated(reliable)当一个模型的概率预测和经验观测相吻合，这个模型就被称为well-calibrated

如：那么该样本中70%的这样的数据点应当是正标签的。

RL总是倾向于well-calibrated，也就是：

**

1. 当（a,b）= (0,1)时，上式对于Logistic Regression是正确的：

对于LR，有：

其中：

所以：

即该分类器对于[0,1]上是well-calibrated的

1. 对于一个well-calibrated的二元分类器，这意味着该分类器会有100%准确率吗？

对于100%准确的分类器，有：

假设b<1/2,则上式不成立

1. L-2的regularization对lr的well-calibrated有什么影响？

Regularization：

这会导致模型不是well-calibrated的了。

结论：

1. （0，1）对与lr成立
2. 对古GLM，所有的都成立

## 关于贝叶斯的解释性以及Regularization

1. 证明:若

因为，

有：

1. 对有先验的高斯分布假设，,这和L\_2规范化是一样的。

证明：

由假设：

故而：

1. 假设有一个线性模型：, ,

求解.

令,

1. 假设参数满足拉普拉斯分布，有：

Linear regression with L2 regularization is also commonly called Ridge regression, and when L1 regularization is employed, is commonly called Lasso regression. 对于任何一个GLM都可以这样做

Lasso regression (i.e L1 regularization) is known to result in sparse parameters, where most of the parameter values are zero, with only some of them non-zero.

## 构造kernel

假设都是kernel，组合后东东是不是kernel？

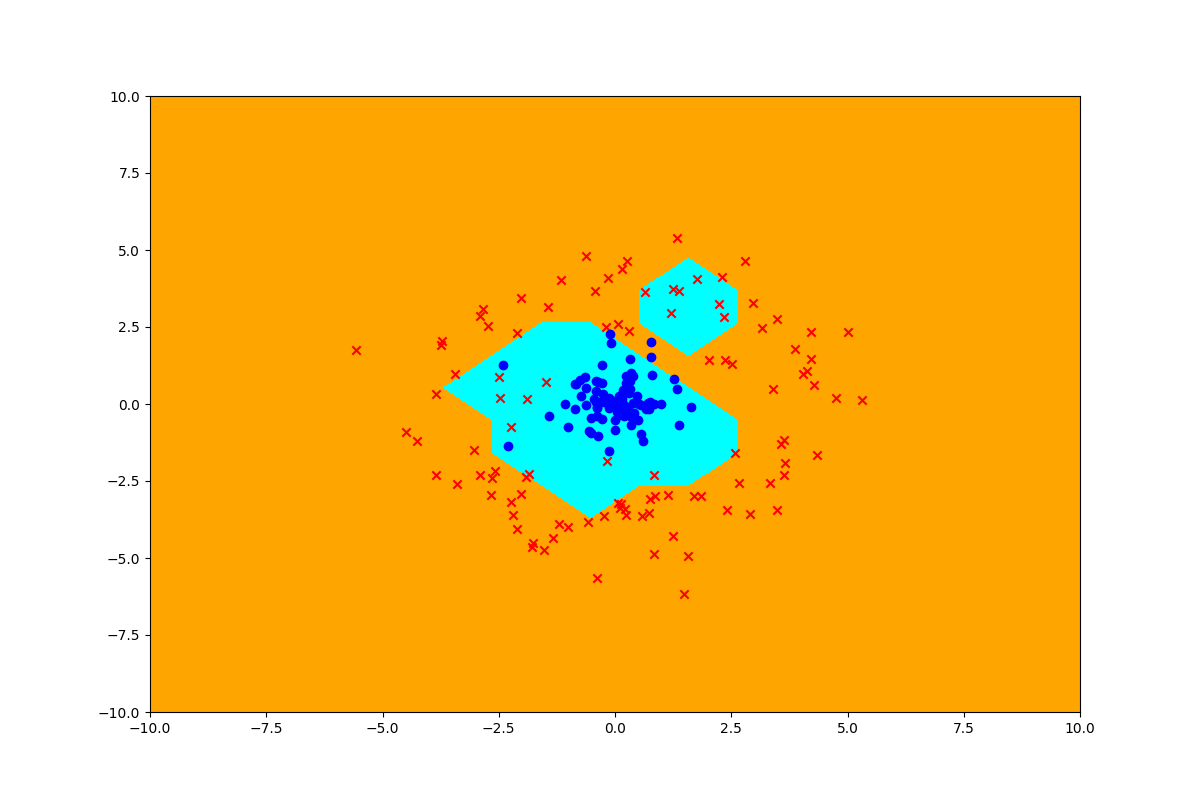
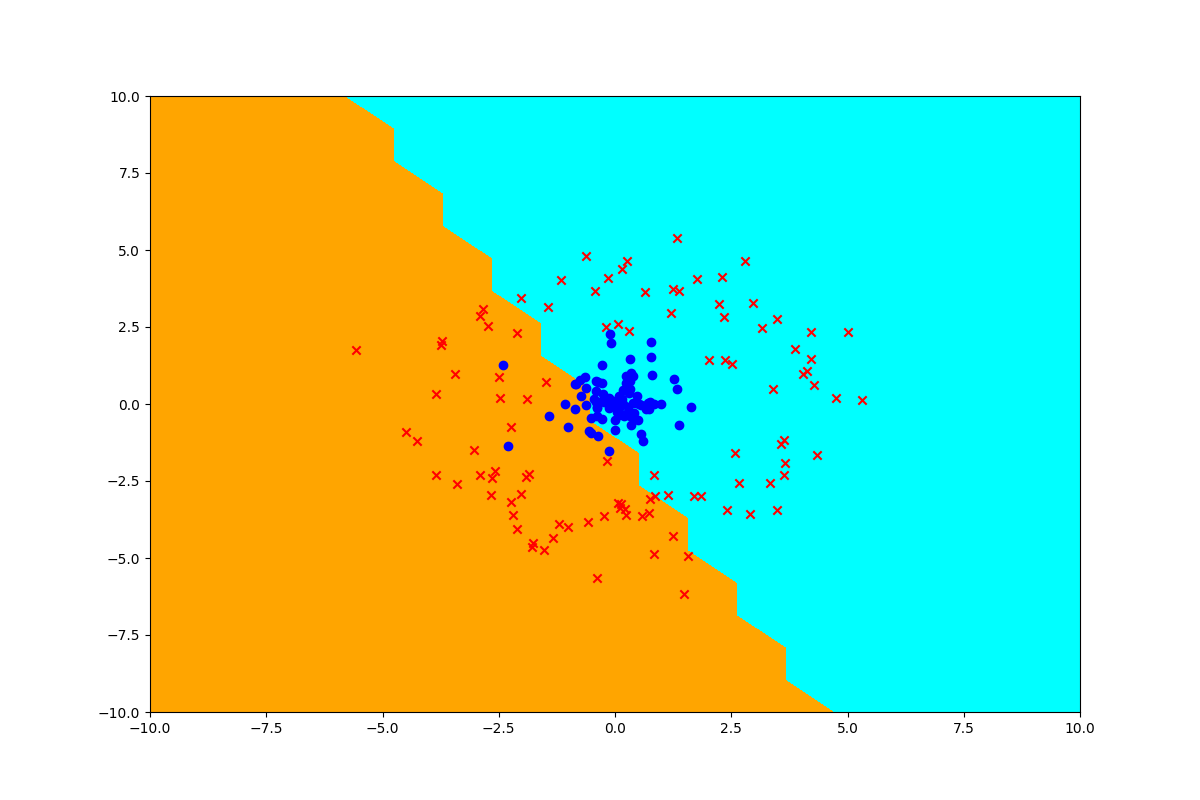
Kernel 矩阵K需要是对阵半正定阵

，假设K2=2K1，则不对

1. , p(x) a polynomial over x with positive coefficients.

## perceptron & kernel

图左为dot kernel，图右为rbf kernel; dot kernel 分类的不好



dot操作实际上没有feature，map，K(x,z)=，实际上没有map，而原来的数据并不是线性可分的。