内容大概分为以下几个部分。

传统的SVM：最大间隔分类器

SVM的优缺点

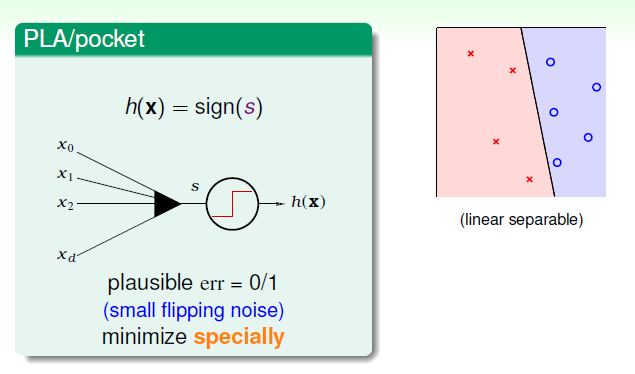
SVM的第一种变形：软间隔分类器

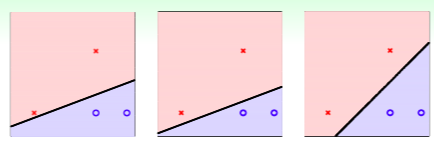
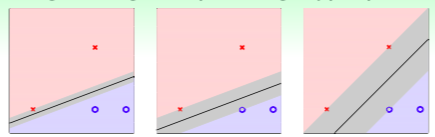
SVM的第二种变形：kernel method

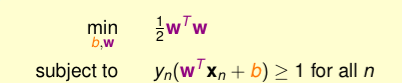
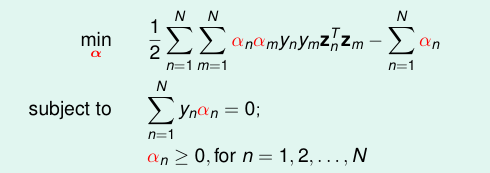
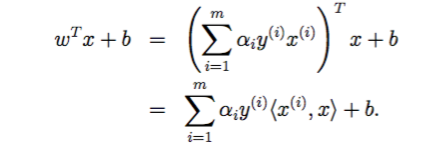
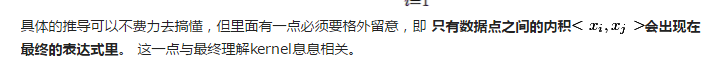
kernel，不止限于SVM

----------我是开始的分割线---------------

## 1. SVM：最大间隔分类器

回想一下所谓感知机：  


感知机算法（PLA）可以找到一条线，把右图中的x和o分开。但是在使用时，往往会存在下面这样尴尬的情况：  
  
这三条线都能成功的分开训练集中的点。但人类的直观感觉，第三种作为分类边界似乎更加‘稳妥一些’。 这种口语化的描述转化为数学语言，就是，我们希望寻找一条线，在能正确分类的同时，与数据点的分隔距离（margin）尽量的大。  
如果真的去测量一下margin，就会得到下图：  
  
最右面的图，确实是具有最大间隔的。而支持向量机，就是可以返回 margin最大的分隔直线 的一种分类算法。

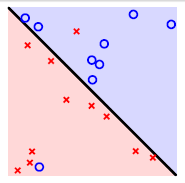
SVM中的优化问题可以写成：，  
求解细节会用到凸优化，这里不再重复，但是我认为 这个过程中有一个 细节是必须要注意到的，就是转化为对偶问题后，待优化函数的形式，和最后表达式的样子：  
上述优化问题的对偶问题长成这样： （z就是x，我也不知道为啥林轩田换符号了）  
而最终的返回结果长成这个样子  


## 2. svm的优缺点

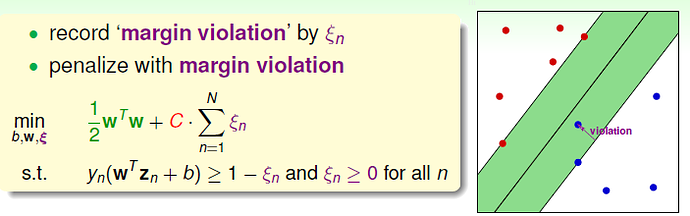
SVM的优点是，返回的分割直线满足margin最大的条件，所以是一个robust的解。而且虽然上面因为没有具体写出推导过程所以这么说比较突兀，****SVM对数据点的依赖 是稀疏的****，只有少量支持向量对最终结果有贡献。

传统的SVM的缺点也很明显。首先SVM的优化问题比较复杂，不仅人难以理解，写起程序来也比较复杂，必须要用到Sequential minimal optimization（SMO）。其次，传统的SVM只能处理线性可分的问题，并且对数据中的噪音也很敏感，因此我们必要对算法进行改造。

## 3. SVM 的第一种变形：软间隔SVM

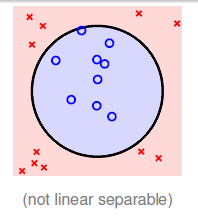
有时数据集大概还是线性可分的，只是存在一些噪音，比如下图：  
  
，对于这种问题，我们的处理方式是 ****允许SVM犯一点错****。 这样的思想就产生了 Soft-Margin SVM 算法。

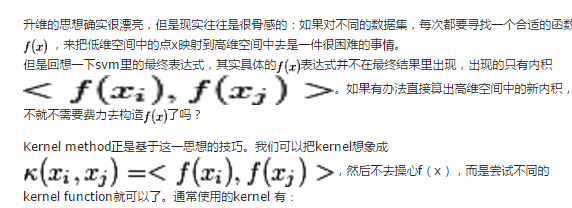
软间隔svm的优化问题大概长成这样：

[](http://cn-discussions.s3.cn-north-1.amazonaws.com.cn/original/3X/5/a/5a46ab08e8457a8f4278dd580a8e9113663388fc.png" \o "软间隔svm.png)

**[软间隔svm.png703x221 23.8 KB](http://cn-discussions.s3.cn-north-1.amazonaws.com.cn/original/3X/5/a/5a46ab08e8457a8f4278dd580a8e9113663388fc.png" \o "软间隔svm.png)**

## 4. SVM的第二种变形： kernel method

有事数据集根本就不是线性可分问题，比如下图：  
  
，这样的问题再搞正则项也没用，这时我们需要一种新的思想： ****升维****。  
升维的思想简单说来就是：在原来feature的基础上人为的构造一些新的feature，在更高维度的空间里，原来不线性可分的问题就会变成线性可分的问题，就又可以用svm了。  
相关的讨论论坛上已经有好多，我再这儿搬运一个帖子，http://www.zhihu.com/question/30371867，对高维空间里的线性可分来个直观感受。



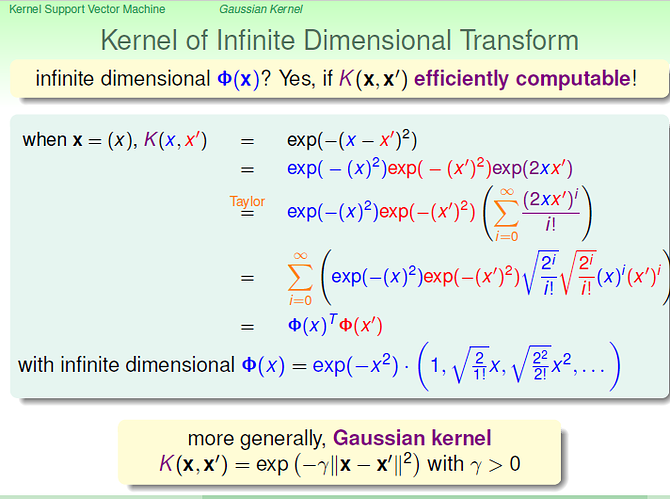
linear kernel（其实就是不用kernel不升维）

Polynomial Kernel

Gaussian Kernel（sklearn里叫rbf kernel）

当你选用kernel时，升维已经自动做好了，所以在调用sklearn中带kernel的svm时才会各种奇形怪状的分类边界。

这里有一种kernel比较特别，高斯核。高斯核对应的映射f（x）是可以反算出来的，结果证明是 ****无穷维****。 有兴趣的同学可以看一下下面的推导。

[](http://cn-discussions.s3.cn-north-1.amazonaws.com.cn/original/3X/0/a/0aa93c764d8174932fed5ffda3da4cc8bc4326be.png" \o "高斯核.png)

**[高斯核.png712x531 59.9 KB](http://cn-discussions.s3.cn-north-1.amazonaws.com.cn/original/3X/0/a/0aa93c764d8174932fed5ffda3da4cc8bc4326be.png" \o "高斯核.png)**

我想看到这里各位同学应该能理解为什么 svm with rbf kernel会这么强大，因为在选用rbf kernel时，数据已经被映射到了无穷维的空间中去，****从而保证数据一定是线性可分的****。

## 5. kernel，不止限于SVM

事实上，SVM这一节里面最重要的思想倒不是SVM本身，而是kernel。 Kernel提供了一种在 ****不写出具体映射表达式的情况下，计算数据内积的方法。**** 原则上说，但凡是需要计算内积的算法，都可以使用kernel trick，而不仅仅局限于svm。

我现在脑子里有的一个例子就是 带kernel的特征筛选算法 Lasso. 附上一篇paper的链接。  
[http://cs.brown.edu/~ls/Publications/nc2014yamada.pdf7](http://cs.brown.edu/~ls/Publications/nc2014yamada.pdf" \t "http://discussions.youdaxue.com/t/topic/_blank)