（1）《机器学习实战》训练营进度问题。

参照训练营教学大纲（会有微调，以每周发布的教学计划为准），模式是采用2+1+1，即2个月的核心学习期、1个月的辅助总结期、1场天池o2o比赛。基本上一周两章（《机器学习实战》书籍）。

书籍+项目代码，合理安排时间。

<https://github.com/RedstoneWill/MachineLearningInAction-Camp>

（2）o2o比赛什么时候开始？

本训练营将进行两次带打比赛：初级赛和复杂赛，难度分成简单和复杂。初级赛第四周（2018/10/22-2018/10/28）；复杂赛暂定第九周。

（3）想复习一下数学，请推荐一下复习的方向。

机器学习中的数学知识主要包括：

**线性代数**：标量、向量、矩阵、张量、范数、特征分解、奇异值分解（SVD）、伪逆矩阵、几种常用的距离：曼哈顿距离、欧式距离、闵可夫斯基距离、切比雪夫距离、余弦夹角、汉明距离、杰卡德相似系数、杰卡德距离。

**概率**：为什么使用概率、随机变量、概率分布、条件概率、贝叶斯公式、期望、方差、协方差、常见的分布函数：0-1分布、几何分布、二项分布、高斯分布、指数分布、泊松分布，拉格朗日乘子法、最大似然估计、

**信息论**：熵、联合熵、条件熵、相对熵、互信息、最大熵模型。

**数值计算**：上溢和下溢、计算复杂性和NP问题、数值计算。

**最优化**：最优化理论、最优化问题的数学描述、凸集和凸集分离定理、梯度下降、牛顿法。

麻省理工公开课：线性代数

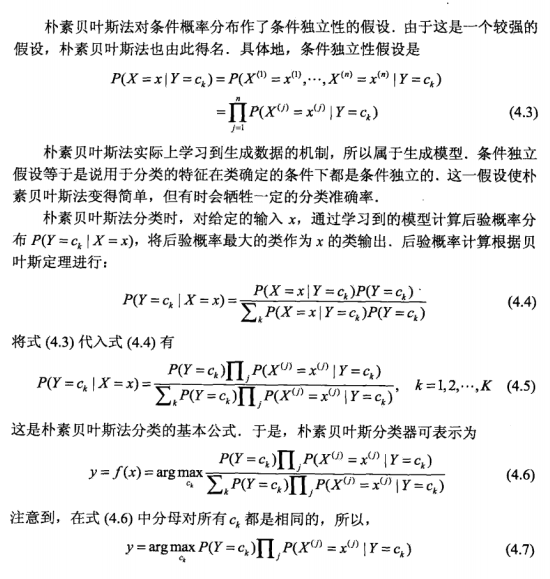
（4）决策树，KNN和随机森林做预测时，如果预测结果是概率值，这个概率值是如何计算的？

KNN中无论是回归还是分类问题，都是找到与样本最近的k个实例点。对于分类问题，计算这k个实例点所属类别最多的那一类即为预测类别；对于回归问题，计算这k个实例点的y值，求平均。

决策树和随机森林做预测时，计算叶子节点所有例点所属类别最多的那一类即为预测类别；对于回归问题，计算这k个实例点的y值，求平均。

（5）朴素贝叶斯，如果不“朴素”的时候，怎么处理？

朴素贝叶斯对条件概率分布做了条件独立性的假设。



如果特征条件不是相互独立，例如身高、体重，则需要使用一般的贝叶斯估计。

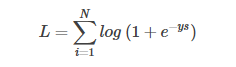
（6）感觉选择哪种分类器，好像和标签有关系，比如二分类的0,1 或者 -1,1 ，或者多分类，不知道针对这个问题有没有什么规律和心得，求分享~

选择哪种分类器与标签label并无关系，只是为了得到简单易懂的损失函数Loss，选择0/1标签或者-1/1标签。

例如逻辑回归中，交叉熵损失函数一般设定y标签为0/1：



其实逻辑回归也可以使用-1/+1标签：



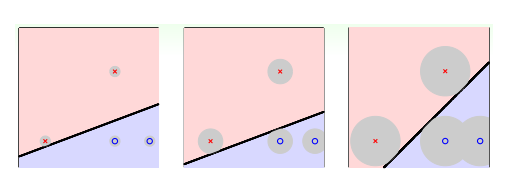
其中，s=wx，表示得分函数。

在SVM中，为了推导方便，也使用-1/+1标签，使用y(wx+b)表示间隔，若y(wx+b)>0，则正间隔，正确的一侧；若y(wx+b)<0，则负间隔，错误的一侧。

（7）逻辑回归与线性svm进行二元分类时，分别适用哪些场景？

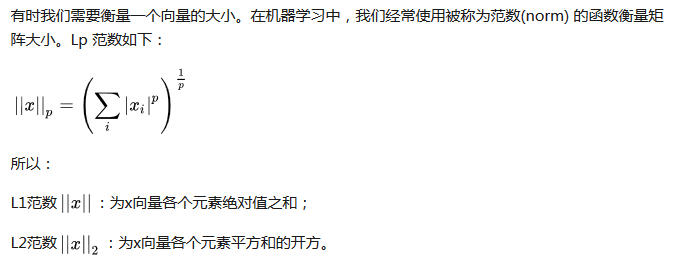
逻辑回归是一种最简单最基本的机器学习分类算法，利用梯度下降算法，不断修正参数w，使其损失函数越来越小。一般能够得到正确的分类面，但不能保证分类线是最健壮的分类面。

SVM是功能强大的分类器，能够在得到正确分类面的基础上使得分类面足够健壮。



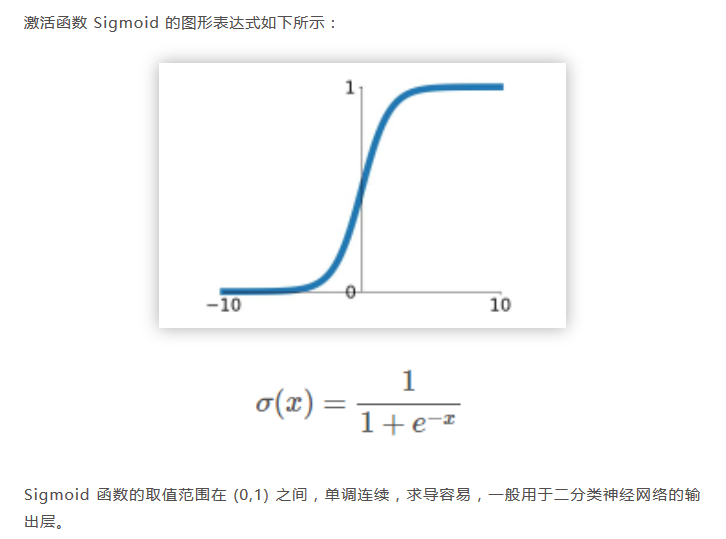
两种方法都是常见的分类算法，从[目标函数](https://www.baidu.com/s?wd=%E7%9B%AE%E6%A0%87%E5%87%BD%E6%95%B0&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)来看，区别在于逻辑回归采用的是logistical loss，svm采用的是hinge loss。这两个[损失函数](https://www.baidu.com/s?wd=%E6%8D%9F%E5%A4%B1%E5%87%BD%E6%95%B0&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "_blank)的目的都是增加对分类影响较大的数据点的权重，减少与分类关系较小的数据点的权重。SVM的处理方法是只考虑support vectors，也就是和分类最相关的少数点，去学习分类器。而逻辑回归通过非线性映射，大大减小了离分类平面较远的点的权重，相对提升了与分类最相关的数据点的权重。两者的根本目的都是一样的。此外，根据需要，两个方法都可以增加不同的正则化项，如l1,l2等等。所以在很多实验中，两种算法的结果是很接近的。  
  
但是逻辑回归相对来说模型更简单，好理解，实现起来，特别是大规模线性分类时比较方便。而SVM的理解和优化相对来说复杂一些。但是SVM的理论基础更加牢固，有一套结构化风险最小化的理论基础，虽然一般使用的人不太会去关注。还有很重要的一点，SVM转化为对偶问题后，分类只需要计算与少数几个支持向量的距离，这个在进行复杂核函数计算时优势很明显，能够大大简化模型和计算  
  
svm 更多的属于非参数模型，而logistic regression 是参数模型，本质不同。其区别就可以参考参数模型和非参模型的区别就好了。  
logic 能做的 svm能做，但可能在准确率上有问题，svm能做的logic有的做不了。

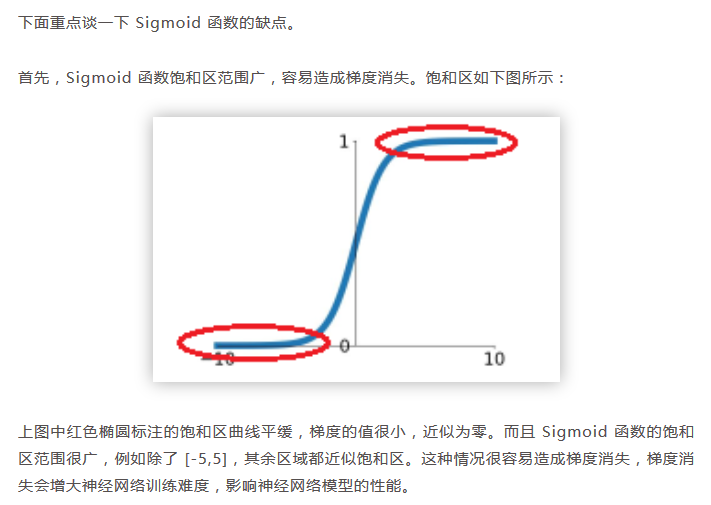
（8）想让老师对范数的概念做个介绍，自己总是不是很清楚。



（9）sigmoid函数里面那个变量x，可以变为ax + b，这个变化在实际中有用得到的时候么？我没查资料，先抛出来再说吧。另外我记着以前看梯度消失和梯度爆炸，就是因为sigmoid在x=0附近梯度特别大，在x距离原点很大的时候，梯度特别小。所以我猜测是不是可以通过调整sigmoid函数中x的系数，避免这个问题。

Sigmoid函数里的值应该是机器学习模型的线性输出wx。





（10）1，绘制决策树有没有好的工具包可以用？

2，优化问题里面，参数迭代的运动轨迹和等高线，这种图是怎么画出来的。

Matplotlib.annotations

（11）讲下算法工程师和数据挖掘工程师在算法掌握程度以及其他技能上有啥区别，以及从数据挖掘工程师转算法工程师需要在储备些什么知识。

算法工程师，通常负责数据清洗和过滤、特征工程、算法效果提升，以及算法的调研、实现，并将其应用于业务，注意，算法工程师通常不局限于机器学习算法，这种对知识背景要求也比较高一些。

数据挖掘工程师，主要是应用机器学习和数据挖掘算法，进行用户行为分析和用户属性挖掘，为算法和业务场景提供支持，Java、Python、Scala 、R，会一门就差不多了，当然这是大数据时代，熟悉 Hadoop/Spark 更好。

数据研发工程师，一般就是负责数据仓库创建或者数据分析平台的开发、设计、运维，要对 Hadoop、Hive、Spark、Storm 等框架原理比较熟悉。

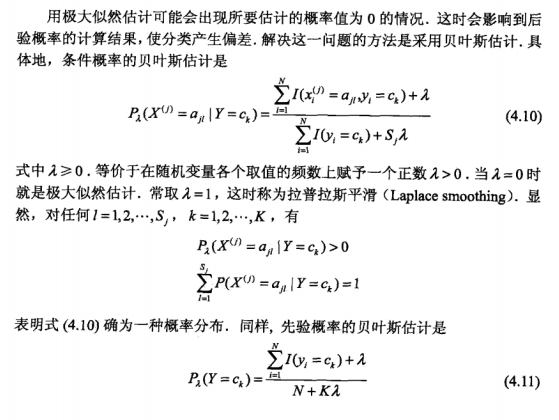
（12）knn，决策树和朴素贝叶斯分类算法怎么优化？

总体上讲，增加训练样本数量。

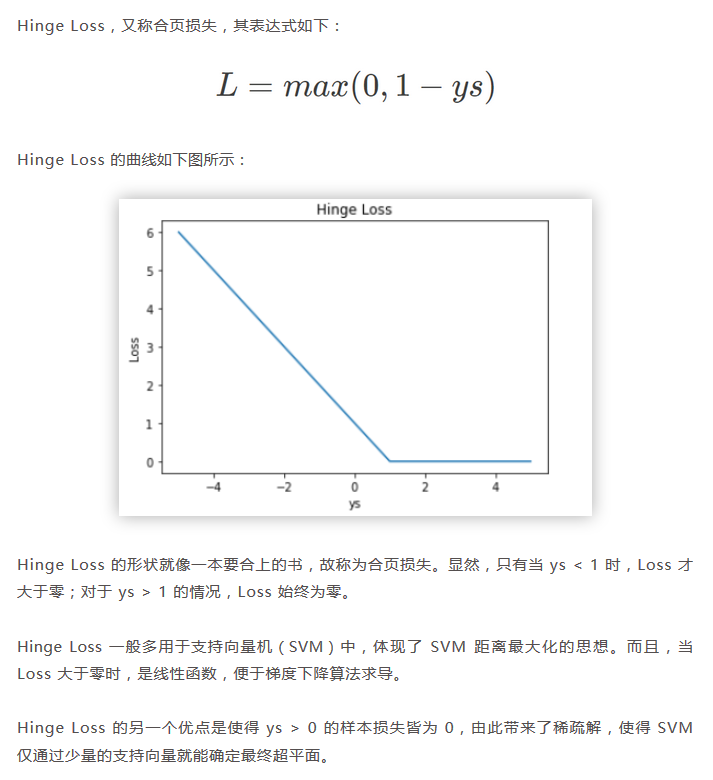
knn：使用交叉验证，选择合适的k值。

决策树：后减枝。后剪枝就是先把整颗决策树构造完毕，然后自底向上的对非叶结点进行考察，若将该结点对应的子树换为叶结点能够带来泛华性能的提升，则把该子树替换为叶结点。

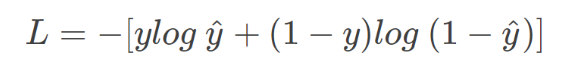
朴素贝叶斯：使用贝叶斯估计代替极大似然估计。



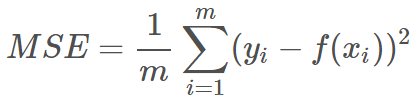
（13）hinge loss与svm的关系，怎样优化。逻辑回归和线性回归输出是什么分布。



逻辑回归loss使用的是交叉熵损失：



线性回归loss使用的是均方误差损失：



（14）我想问下在使用SVM的时候要判断数据是否线性可分，二维可以直接通过绘图，如果是三维甚至是更高维的数据，该如何判断数据是否线性可分呢？

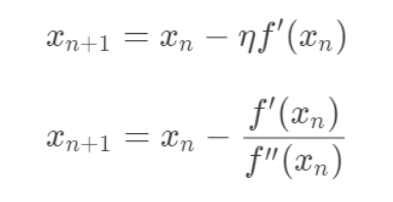
一般可以先使用最简单的线性SVM模型进行分类，看下预测准确率。

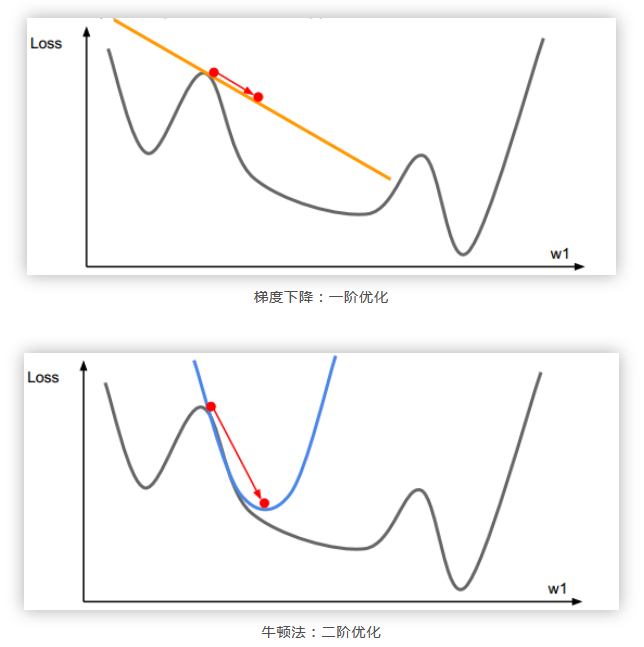
（15）不同场景下选择损失函数有什么不同，原理是什么

<https://mp.weixin.qq.com/s/AdpO4xxTi0G7YiTfjEz_ig>

<https://mp.weixin.qq.com/s/Xbi5iOh3xoBIK5kVmqbKYA>

（16）老师好，请先看贴图，截自周志华机器学习page60；从图上可以看出，学习率已经算出来了，是个二阶导数的倒数，但是平时训练时学习率都是我们自定义的，比如0.1,0.01之类，这两者没有联系么？





首先，我们来看一下牛顿法的优点。第一，牛顿法的迭代更新公式中没有参数学习因子，也就不需要通过交叉验证选择合适的学习因子了。第二，牛顿法被认为可以利用到曲线本身的信息, 比梯度下降法更容易收敛（迭代更少次数）。如下图是一个最小化一个目标方程的例子, 红色曲线是利用牛顿法迭代求解, 绿色曲线是利用梯度下降法求解。显然，牛顿法最优化速度更快一些。

<https://mp.weixin.qq.com/s/eXomhbRU5vc78DJcJlSigA>