第一章：

第一章的主题是机器学习。我今天是准备用通俗的语言梳理机器学习的内容。

在开始进行机器学习之前，我们需要明白为什么要学习机器学习，以及机器学习包括哪些内容。机器学习可以帮助我们更好的应对未知的问题环境、领域，更简便地处理大数据，甚至还可以帮我们理解我们人类自己的学习和器官。

那么机器学习又包括什么呢？学习意味着改变，而机器学习主要指电脑通过经验自动地进行提升，也就是***Improve on Task with respect to Performance metric based on Experience***。

机器学习主要包括监督学习、无监督学习、半监督学习和强化学习，我们在这一章主要讲的是监督学习和无监督学习。

对于监督学习，其实就是根据给出的带有标签的数据，训练出一个函数。监督学习的关键在于函数的形式和如何优化。要注意不能过拟合，这样虽然训练集上效果可以很好，但是预测其他的数据的时候表现很差，当然同时也要防止欠拟合。

监督学习我们从 rate-learning入手，介绍了max-min search，max-min search，其实就是把对方相像的最强，对方一定会选择最优的策略，选择最大值。而我们在不同的状况下，考虑自己的棋子放一个选定了的位置上，对方又放在了他的最优地位后，我们会取得这种下法的最小值，在所有的最小值里取得最大值的方法进行操作。

离散的数据点，我们有一种方法叫做决策树，我们希望得到的决策树是最短的。对此我们有两种方法。

第一种是利用剩余熵，我们选择属性A，将训练集分成子集，子集的比例乘以他的熵，然后进行求和这是信息的增益，选择最大的信息增益就是我们每次选择的根节点。

MDL，最小描述长度方法，描述长度由学习得到的决策树以及其他的那些例外的，不能够被决策树描述的数据 组成。一点点增加树的复杂性，当DL不在变小的时候停止增加。或者，可以将树完全生成，然后进行自下而上的剪枝。

之后根据叶贝斯公式，



其实就是我们学过的条件概率。

在此基础上又有了朴素叶贝斯NBC，

叶贝斯训练网**BBN这两个其实是类似的。**



除了以上我们学习的内容以外呢，监督学习还有梯度下降、支持向量机svm等等。

监督学习的内容就到这里，无监督学习是**本质上是一个统计手段，在没有标签的数据里可以发现潜在的一些结构的一种训练方式，大多数的时候其实就是**聚类、降维**。**

**聚类的重点是如何去度量相似性，如何让簇内的相似度最高，簇间的相似度最小。**

**相似度的度量有很多种，比如，闵可夫斯基距离，分别令p=1，2，无穷大，得到欧氏距离、曼哈顿距离、切比雪夫距离，也有其他的，比如，余弦距离、马氏距离、海明距离、杰卡德距离。**

**聚类的方法有很多但是还没有特别一致的分类方法，现在采用的大多数是韩家伟提出的，partition 划分方法，hierarchical，基于层次的方法，density，基于密度的方法，grid，基于网个的方法，基于模型的方法。**

**partition 划分方法，的例子有k-means，k-medoids，k-means是先选k个点作为每个簇的中心，然后把其他的划分到这k个簇里面，计算均值重新作为中心，重复重复如此。**

**不过k-means特别容易受到噪声的影响。所以我们提出了k-medoids，每次尝试用其他的点替代当前的中心，如果效果更好就进行替换。这中算法比k-means的鲁棒性好，但是它本身依旧是基本没有鲁棒性的。而如何处理数据特别多的问题呢，有两个优化，一个是clara，进行抽样，另一种是依靠爬山算法的clarans。**

**而且这两种方法k-meansk-medoids必须要知道到底划分为多少簇。**

**而hierarchical，基于层次的方法能够很好的解决这个问题。自顶而下 diana 或者自底而上 agnes的建立树，便可以自如地决定多少簇了。**

**不再采用数据点之间进行反复的计算，而是引入了新的相似性的度量，cf，只需要进行一遍扫描。就得到了birch算法。**

**采用多个数据点描述形状，代表点不断向均值靠近，这样就得到了cure算法。**

**把点连成图，每个点只考虑前k个最近的点连边。然后进行图割，切边后进行聚类，这就是chameleon算法。**

**以上的这些呢，好像对于簇的形状都不是任意的。于是我们就提出了density，基于密度的方法。主要分为两种一个是DBSCADN，一个是DENCLUE。DBSCADN的簇定义为了密度连接点的最大的结合，自行定义 密度到达多少可能形成簇，比如1cm2至少要有5个点，连接在一起的簇就形成了更大的簇。DENCLUE是考虑每一个点对旁边影响，影响叠加在一起，然后可以当最高点超过了阈值的时候就当做一个簇。**

**DENCLUE的速度很快，但是能不能再快一些呢？我们介绍了grid，基于密度的算法。就是把数据划分到网格里面，然后可以借用基于密度的方法，再聚类。速度很快，但是会牺牲一些准确率并且形状只能是水平和垂直的。**