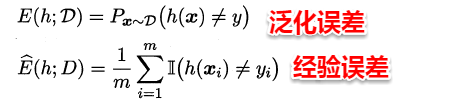
上篇主要介绍了常用的特征选择方法及稀疏学习。首先从相关/无关特征出发引出了特征选择的基本概念，接着分别介绍了子集搜索与评价、过滤式、包裹式以及嵌入式四种类型的特征选择方法。子集搜索与评价使用的是一种优中生优的贪婪算法，即每次从候选特征子集中选出最优子集；过滤式方法计算一个相关统计量来评判特征的重要程度；包裹式方法将学习器作为特征选择的评价准则；嵌入式方法则是通过L1正则项将特征选择融入到学习器参数优化的过程中。最后介绍了稀疏表示与压缩感知的核心思想：稀疏表示利用稀疏矩阵的优良性质，试图通过某种方法找到原始稠密矩阵的合适稀疏表示；压缩感知则试图利用可稀疏表示的欠采样信息来恢复全部信息。本篇将讨论一种为机器学习提供理论保证的学习方法--计算学习理论。

#**13、计算学习理论**

计算学习理论（computational learning theory）是通过“计算”来研究机器学习的理论，简而言之，其目的是分析学习任务的本质，例如：**在什么条件下可进行有效的学习，需要多少训练样本能获得较好的精度等，从而为机器学习算法提供理论保证**。

首先我们回归初心，再来谈谈经验误差和泛化误差。假设给定训练集D，其中所有的训练样本都服从一个未知的分布T，且它们都是在总体分布T中独立采样得到，即**独立同分布**（independent and identically distributed，i.i.d.），在《贝叶斯分类器》中我们已经提到：独立同分布是很多统计学习算法的基础假设，例如最大似然法，贝叶斯分类器，高斯混合聚类等，简单来理解独立同分布：每个样本都是从总体分布中独立采样得到，而没有拖泥带水。例如现在要进行问卷调查，要从总体人群中随机采样，看到一个美女你高兴地走过去，结果她男票突然冒了出来，说道：you jump，i jump，于是你本来只想调查一个人结果被强行撒了一把狗粮得到两份问卷，这样这两份问卷就不能称为独立同分布了，因为它们的出现具有强相关性。

回归正题，**泛化误差指的是学习器在总体上的预测误差，经验误差则是学习器在某个特定数据集D上的预测误差**。在实际问题中，往往我们并不能得到总体且数据集D是通过独立同分布采样得到的，因此我们常常使用经验误差作为泛化误差的近似。

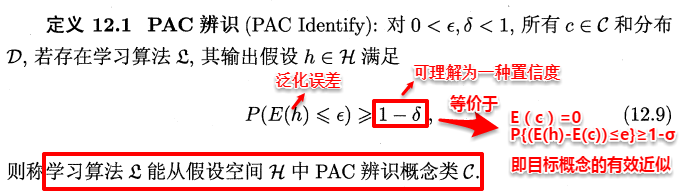
[](https://camo.githubusercontent.com/8a45b5402d5682415cf7c38caf5e80af17d6f90e/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383534663338643466652e706e67)

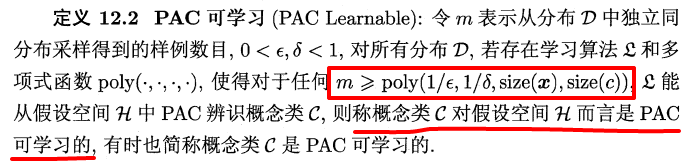
##**13.1 PAC学习**

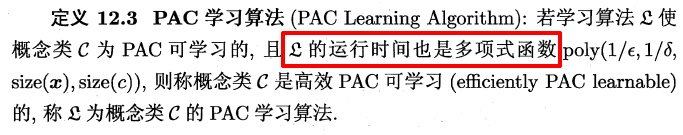
在高中课本中，我们将**函数定义为：从自变量到因变量的一种映射；对于机器学习算法，学习器也正是为了寻找合适的映射规则**，即如何从条件属性得到目标属性。从样本空间到标记空间存在着很多的映射，我们将每个映射称之为**概念**（concept），定义：

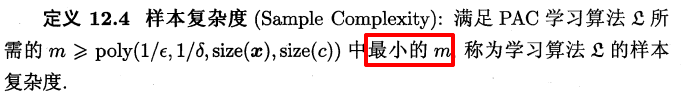
若概念c对任何样本x满足c(x)=y，则称c为**目标概念**，即最理想的映射，所有的目标概念构成的集合称为\*\*“概念类”**； 给定学习算法，它所有可能映射/概念的集合称为**“假设空间”**，其中单个的概念称为**“假设”**（hypothesis）； 若一个算法的假设空间包含目标概念，则称该数据集对该算法是**可分\*\*（separable）的，亦称**一致**（consistent）的； 若一个算法的假设空间不包含目标概念，则称该数据集对该算法是**不可分**（non-separable）的，或称**不一致**（non-consistent）的。

举个简单的例子：对于非线性分布的数据集，若使用一个线性分类器，则该线性分类器对应的假设空间就是空间中所有可能的超平面，显然假设空间不包含该数据集的目标概念，所以称数据集对该学习器是不可分的。给定一个数据集D，我们希望模型学得的假设h尽可能地与目标概念一致，这便是**概率近似正确** (Probably Approximately Correct，简称PAC)的来源，即以较大的概率学得模型满足误差的预设上限。

[](https://camo.githubusercontent.com/8743c6a56cd623803d13408a1eb52a503e0cd61a/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383534663434366639372e706e67)

[](https://camo.githubusercontent.com/dcc005dfd11ad3b3b1dad0fdb71e69a52b3469bc/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383534663438326430622e706e67)

[](https://camo.githubusercontent.com/a272d9e0a787349dae028dff913c8242dd5e17d0/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383534663437643030362e706e67)

[](https://camo.githubusercontent.com/0d17478b73c26d09857e7b524db5a7a65b1c8abd/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383534663436616439312e706e67)

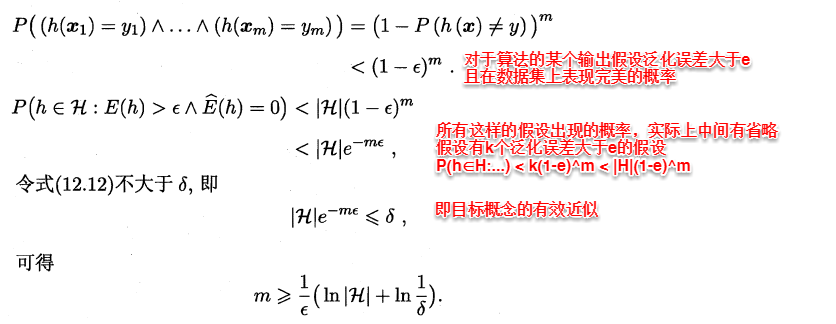
上述关于PAC的几个定义层层相扣：定义12.1表达的是对于某种学习算法，如果能以一个置信度学得假设满足泛化误差的预设上限，则称该算法能PAC辨识概念类，即该算法的输出假设已经十分地逼近目标概念。定义12.2则将样本数量考虑进来，当样本超过一定数量时，学习算法总是能PAC辨识概念类，则称概念类为PAC可学习的。定义12.3将学习器运行时间也考虑进来，若运行时间为多项式时间，则称PAC学习算法。

显然，PAC学习中的一个关键因素就是**假设空间的复杂度**，对于某个学习算法，**若假设空间越大，则其中包含目标概念的可能性也越大，但同时找到某个具体概念的难度也越大**，一般假设空间分为有限假设空间与无限假设空间。

##**13.2 有限假设空间**

###**13.2.1 可分情形**

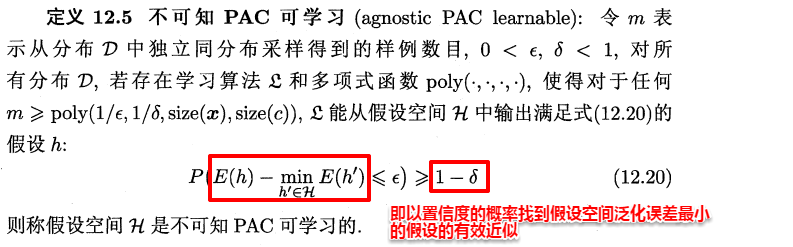
可分或一致的情形指的是：**目标概念包含在算法的假设空间中**。对于目标概念，在训练集D中的经验误差一定为0，因此首先我们可以想到的是：不断地剔除那些出现预测错误的假设，直到找到经验误差为0的假设即为目标概念。但**由于样本集有限，可能会出现多个假设在D上的经验误差都为0，因此问题转化为：需要多大规模的数据集D才能让学习算法以置信度的概率从这些经验误差都为0的假设中找到目标概念的有效近似**。

[](https://camo.githubusercontent.com/759c12770968037c9d0f19b57ec1666fc23e5724/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383534663438346633302e706e67)

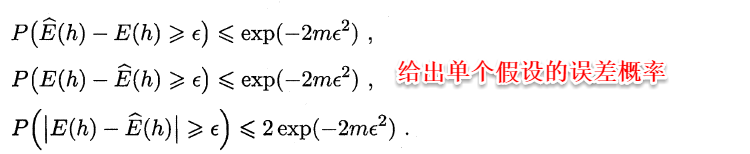
通过上式可以得知：**对于可分情形的有限假设空间，目标概念都是PAC可学习的，即当样本数量满足上述条件之后，在与训练集一致的假设中总是可以在1-σ概率下找到目标概念的有效近似。**

###**13.2.2 不可分情形**

不可分或不一致的情形指的是：**目标概念不存在于假设空间中**，这时我们就不能像可分情形时那样从假设空间中寻找目标概念的近似。但**当假设空间给定时，必然存一个假设的泛化误差最小，若能找出此假设的有效近似也不失为一个好的目标，这便是不可知学习(agnostic learning)的来源。**

[](https://camo.githubusercontent.com/91aba394acf84c256e1026735079ad9900291b8d/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383534663438356632652e706e67)

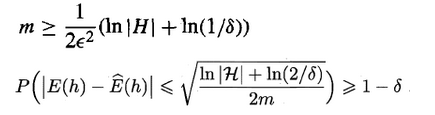
这时候便要用到**Hoeffding不等式**：

[](https://camo.githubusercontent.com/d6bbdee59a569460cadfac2a9dc9a162af7c379c/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383534663436393730612e706e67)

对于假设空间中的所有假设，出现泛化误差与经验误差之差大于e的概率和为：

[9.png](https://camo.githubusercontent.com/49321ecff9e3ae380e1ac137e24b9ff6a6a08167/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383534663431313466642e706e67)

因此，可令不等式的右边小于（等于）σ，便可以求出满足泛化误差与经验误差相差小于e所需的最少样本数，同时也可以求出泛化误差界。

[](https://camo.githubusercontent.com/e7c6e8530795a9580ddaebc42b2d698ed39839ea/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383534663434306130322e706e67)

##**13.3 VC维**

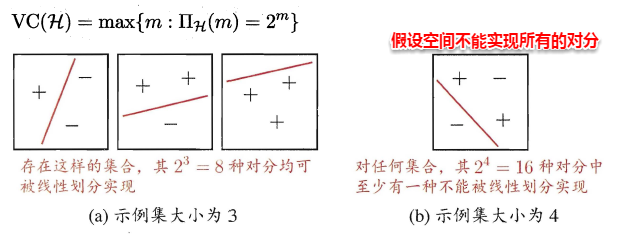
现实中的学习任务通常都是无限假设空间，例如d维实数域空间中所有的超平面等，因此要对此种情形进行可学习研究，需要度量**假设空间的复杂度**。这便是**VC维**（Vapnik-Chervonenkis dimension）的来源。在介绍VC维之前，需要引入两个概念：

**增长函数**：对于给定数据集D，假设空间中的每个假设都能对数据集的样本赋予标记，因此一个假设对应着一种打标结果，不同假设对D的打标结果可能是相同的，也可能是不同的。随着样本数量m的增大，假设空间对样本集D的打标结果也会增多，增长函数则表示假设空间对m个样本的数据集D打标的最大可能结果数，因此**增长函数描述了假设空间的表示能力与复杂度。**

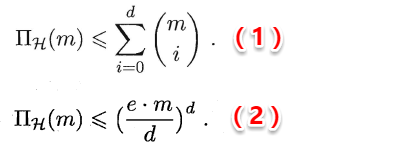
[11.png](https://camo.githubusercontent.com/9dc514c3fdfb87705d39607603c20899f9d55a85/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383535626139373063642e706e67)

**打散**：例如对二分类问题来说，m个样本最多有2^m个可能结果，每种可能结果称为一种\*\*“对分”\*\*，若假设空间能实现数据集D的所有对分，则称数据集能被该假设空间打散。

**因此尽管假设空间是无限的，但它对特定数据集打标的不同结果数是有限的，假设空间的VC维正是它能打散的最大数据集大小**。通常这样来计算假设空间的VC维：若存在大小为d的数据集能被假设空间打散，但不存在任何大小为d+1的数据集能被假设空间打散，则其VC维为d。

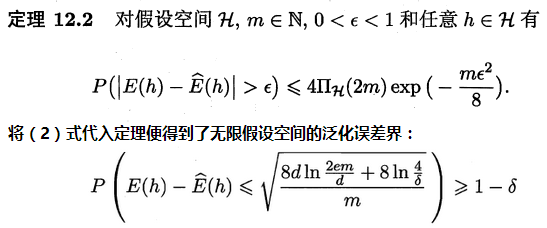
[](https://camo.githubusercontent.com/21016db144e2155d42057af4454e400a9ab49a4b/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383535626232306331652e706e67)

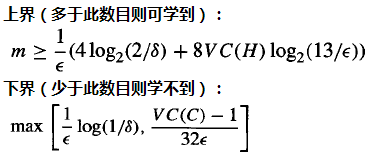
同时书中给出了假设空间VC维与增长函数的两个关系：

[](https://camo.githubusercontent.com/47c9897be33914becd89fd83928634b237632d70/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383535626138336562382e706e67)

直观来理解（1）式也十分容易： 首先假设空间的VC维是d，说明当m<=d时，增长函数与2^m相等，例如：当m=d时，右边的组合数求和刚好等于2^d；而当m=d+1时，右边等于2^(d+1)-1，十分符合VC维的定义，同时也可以使用数学归纳法证明；（2）式则是由（1）式直接推导得出。

在有限假设空间中，根据Hoeffding不等式便可以推导得出学习算法的泛化误差界；但在无限假设空间中，由于假设空间的大小无法计算，只能通过增长函数来描述其复杂度，因此无限假设空间中的泛化误差界需要引入增长函数。

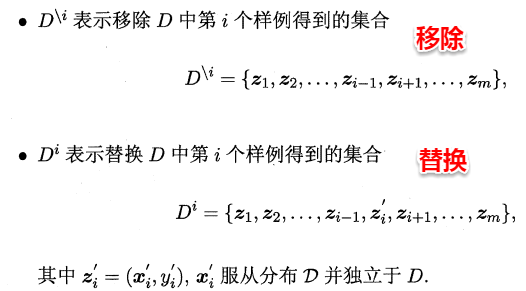
[](https://camo.githubusercontent.com/a702b2157f87ce12e98fd62a451cbf7139169066/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383535626162633839302e706e67)

[](https://camo.githubusercontent.com/bdc73a553a0dc904aef565d66c3141565d44c3c1/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383535626135623263332e706e67)

上式给出了基于VC维的泛化误差界，同时也可以计算出满足条件需要的样本数（样本复杂度）。若学习算法满足**经验风险最小化原则（ERM）**，即学习算法的输出假设h在数据集D上的经验误差最小，可证明：**任何VC维有限的假设空间都是（不可知）PAC可学习的，换而言之：若假设空间的最小泛化误差为0即目标概念包含在假设空间中，则是PAC可学习，若最小泛化误差不为0，则称为不可知PAC可学习。**

##**13.4 稳定性**

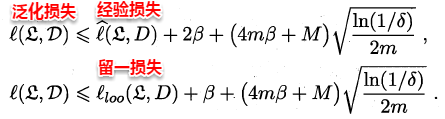
稳定性考察的是当算法的输入发生变化时，输出是否会随之发生较大的变化，输入的数据集D有以下两种变化：

[](https://camo.githubusercontent.com/4eecca8d2a8ae10382f1a54428737ead8317aa84/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383535626164633561382e706e67)

若对数据集中的任何样本z，满足：

[17.png](https://camo.githubusercontent.com/9807e47c928e5e8a8881109229602a9395a568d9/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383535626135396230362e706e67)

即原学习器和剔除一个样本后生成的学习器对z的损失之差保持β稳定，称学习器关于损失函数满足**β-均匀稳定性**。同时若损失函数有上界，即原学习器对任何样本的损失函数不超过M，则有如下定理：

[](https://camo.githubusercontent.com/de3b89f8908b98c238ded500637c6c97bbc3f759/68747470733a2f2f692e6c6f6c692e6e65742f323031382f31302f31382f356263383535626162653763332e706e67)

事实上，**若学习算法符合经验风险最小化原则（ERM）且满足β-均匀稳定性，则假设空间是可学习的**。稳定性通过损失函数与假设空间的可学习联系在了一起，区别在于：假设空间关注的是经验误差与泛化误差，需要考虑到所有可能的假设；而稳定性只关注当前的输出假设。

在此，计算学习理论就介绍完毕，一看这个名字就知道这一章比较偏底层理论了，最终还是咬着牙看完了它，这里引用一段小文字来梳理一下现在的心情：“孤岂欲卿治经为博士邪？但当涉猎，见往事耳”，就当扩充知识体系吧~