学习算法基础

蔡少伟

中国科学院大学

2016

Outline

Outline

关于本章教学

课程目的

- 机器学习本身是一个很大的领域，本章只介绍机器学习基本概念和重要理论以及典型的学习算法，并不深入。
- 希望通过本章学习，培养机器学习算法的思维和直觉。

教学内容

- 基本概念
- 实验方法（模型评估与选择）
- 基本学习理论（PAC学习，VC维，稳定性）
- 典型学习算法（支持向量机，神经网络）

主要参考教材

- 《机器学习》by 周志华，清华大学出版社
- Machine Learning, by Tom M. Mitchell
- Introduction to Machine Learning (3rd edition), by Ethem Alpaydin
- Deep Learning, Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press.
- Online resources, https://github.com/ty4z2008/Qix/blob/master/dl.md (links to many online books and papers)

什么是机器学习

我们经常会利用经验做出预判，比如

- 感到和风，看到晚霞，我们会认为明天是个晴天；
- 某个西瓜色泽青绿，根蒂蜷缩，敲声浊响，我们就判断是个好瓜；
- 看到邮件包含“免费致富”“商业推广”等词，我们就判断是垃圾邮件。

计算机能帮忙吗？
机器学习正是这样一门学科，它致力于研究如何通过计算的手段，利用经验来改善系统自身的性能。

- 在计算机系统中，“经验”通常以数据形式存在。
- 机器学习研究的主要内容，是关于在计算机上从数据中产生模型的算法，即“学习算法”。我们把经验数据提供给学习算法，它就能基于这些数据产生模型。
- 面对新的情况也即新的数据时，把新数据输入给模型，模型就会提供相应的判断。

基本术语

假定我们收集了一批关于西瓜的数据

编号	色泽	根蒂	敲声
1	青绿	蜷缩	浊响
2	乌黑	蜷缩	浊响
3	青绿	笔直	清脆
4	乌黑	微蜷	沉闷

- 这组记录的集合称为一个数据集(data set)。
- 每条记录是关于一个对象或事件的描述，称为一个示例(instance)或样本(sample)。
- 反映对象或时间在某方面的表现或性质的事项，例如“色泽”，“敲声”，称为属性(attribute)或特征(feature)，属性上的取值称为属性值(attribute value)。属性张成的空间称为属性空间(attribute space).
- 每个样本拥有的(属性,属性值）个数称为它的维度(dimensionality).

从数据中学得模型的过程称为“学习”或“训练”。

- 训练过程使用的数据称为训练数据(training data)，其中每个样本称为一个训练样本。训练样本组成的集合称为训练集(training set)。
- 一个模型对应了关于数据的某种潜在的规律，也称为假设(hypothesis)；这种潜在规律的自身，称为真相或真实(ground-truth)。模型也称为学习器(learner)。
- 学习的过程就是为了找出或逼近真相。

基本术语

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	笔直	清脆	否
4	乌黑	微蜷	沉闷	否

- 要建立预测模型，我们还需要训练样本的结果信息，称为标记(label)。
- 拥有了标记信息的示例，称为样例(example)。一般的，用$(\mathbf{x}_i, y_i)$表示第i个样例，其中$y_i \in \mathcal{Y}$是示例$\mathbf{x_i}$的标记，$\mathcal{Y}$是所有标记的集合，称为标记空间或输出空间。

学习的任务

- 若我们要预测的是离散值，例如“好瓜”“坏瓜”，此类学习任务称为分类(classification)。
- 若要预测的是连续值，例如西瓜成熟度0.95, 0.37，此类学习任务称为回归(regression)。
- 我们还可以对数据进行分簇，也即将训练集分为若干组，每组称为一个簇（cluster)，此类学习任务称为聚类(clustering)。

根据训练数据是否拥有标记信息，学习任务可大致划分为两大类：监督学习和无监督学习。分类和回归是前者的代表，而聚类是后者的代表。

假设空间

从样例中学习是一个归纳（从特殊到一般）的过程，因此也称为归纳学习。

我们可以把学习过程看成一个在所有假设组成的空间中进行搜索的过程，搜索目标就是找到与训练集匹配的假设。

一旦假设的表示形式确定，那么也就隐含地为学习算法确定了假设空间。

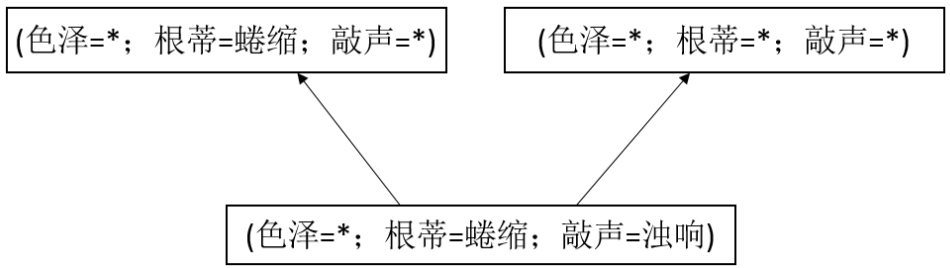
- 对西瓜的例子，我们的假设表示形式为“(色泽=？)∧(根蒂=？)∧(敲声=？)”，于是我们的假设空间就包含了所有这种形式的可能取值所形成的假设。
- 如果某个属性无论取什么值都合适，我们用通配符“*”表示。
- 假设之间的“一般特殊关系”。
  h_1 = (色泽=青绿，根蒂=*，敲声=浊响)
  h_2 = (色泽=青绿，根蒂=*，敲声=*)，
  则h_1比h_2特殊，而h_2比h_1一般。

可以有许多策略对这个假设空间进行搜索，比如从一般到特殊，从特殊到一般，搜索过程可以不断删除与正例不一致的假设还有和反例一致的假设。最终会获得与训练集一致的假设。

版本空间

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	笔直	清脆	否
4	乌黑	微蜷	沉闷	否

所有与训练集一致的假设构成的集合，我们称为版本空间(version space)。
比如，对应于上表训练集的版本空间如下：



这三个假设对应了三个选好西瓜的模型，那么我们应该选择哪个模型呢？

归纳偏好

仅根据训练集，我们无法断定版本空间中哪一个假设更好。
然而，对一个具体的学习算法而言，它必须产生一个模型。这时，学习算法本身的“偏好”就起到关键作用。对于选西瓜的例子

- 如果我们的算法喜欢“尽可能特殊”的模型，则它会选择
  “好瓜$\longleftrightarrow$(色泽=*)$\wedge$(根蒂=蜷缩)$\wedge$(敲声=浊响)”
- 如果我们的算法喜欢“尽可能一般”的模型，并且由于某种原因它更“相信”根蒂属性，则它会选择
  “好瓜$\longleftrightarrow$(色泽=*)$\wedge$(根蒂=蜷缩)$\wedge$(敲声=*)”

归纳偏好：机器学习算法在学习过程中对某种类型假设的偏好，称为归纳偏好(inductive bias)或简称偏好(bias)。

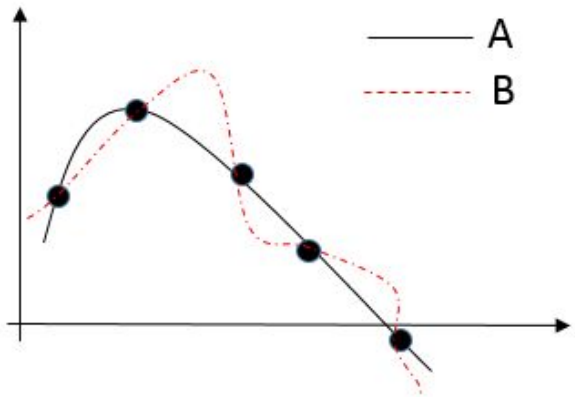
Remark 1
任何一个有效的机器学习算法必有其归纳偏好，否则无法产生确定的学习结果。

归纳偏好

有没有一般性的原则来引导算法确立“正确的”偏好呢？

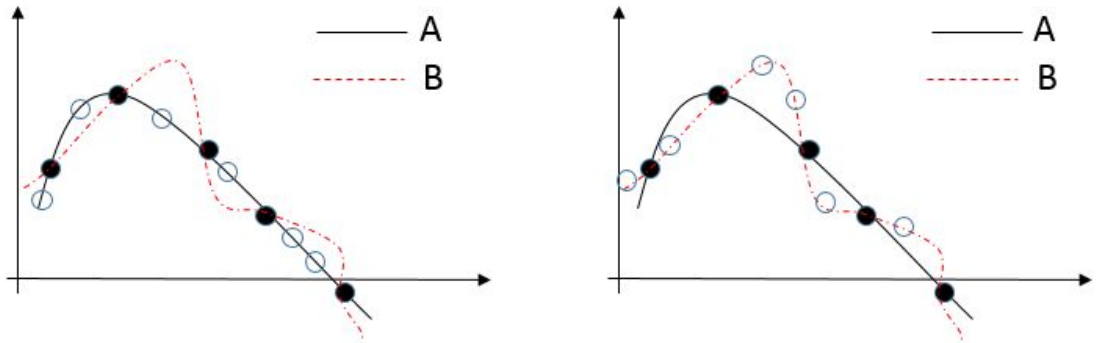
- “奥卡姆剃刀”是一种常见的，自然科学中最基本的原则，即“如果有多个假设与观察一致，则选最简单的那个”。

比如，在下面的例子中，A曲线和B曲线都和样本训练集一致。如果我们用“奥卡姆剃刀”原则，则会选择A曲线，因为它描述起来更简单。



No Free Lunch Theorem

我们满怀信心地期待A曲线模型会比B曲线模型好；也就是说，A的泛化能力比B强（与训练集外的样本更一致）。然而，事实可能如我们期望，也可能是相反的。



没有免费的午餐（黑点：训练样本；白点：测试样本）

Remark 2

对于一个学习算法$\mathcal{A}_a$，若它在某些问题上比学习算法$\mathcal{A}_b$好，则必然存在另一些问题，在那里$\mathcal{A}_b$比$\mathcal{A}_a$好。这个结论对任何算法都成立。

Proof of No Free Lunch Theorem

假设样本空间$\mathcal{X}$和假设空间$\mathcal{H}$都是离散的。
$P(h|X,\mathcal{A}_a)$: 算法$\mathcal{A}_a$基于训练集X产生假设h的概率。
f: 我们希望学习的真实目标函数。
那么，$\mathcal{A}_a$在训练集外的所有样本的误差为

$$E_{ote}(\mathcal{A}_a|X,f)=\sum_h\sum_{\mathbf{x}\in\mathcal{X}-X}P(\mathbf{x})\mathbb{I}(h(\mathbf{x})\neq f(\mathbf{x}))P(h|X,\mathcal{A}_a)$$

考虑二分类问题，真实目标函数f可以是任何$\mathcal{X}\mapsto\{0,1\}$的函数。对所有可能的$f$按均匀分布对误差求和，有

$$\begin{aligned}\sum_f E_{ote}(\mathcal{A}_a|X,f)&=\sum_f\sum_h\sum_{\mathbf{x}\in\mathcal{X}-X}P(\mathbf{x})\mathbb{I}(h(\mathbf{x})\neq f(\mathbf{x}))P(h|X,\mathcal{A}_a)\\&=\sum_{\mathbf{x}\in\mathcal{X}-X}P(\mathbf{x})\sum_h P(h|X,\mathcal{A}_a)\sum_f\mathbb{I}(h(\mathbf{x})\neq f(\mathbf{x}))\\&=\sum_{\mathbf{x}\in\mathcal{X}-X}P(\mathbf{x})\sum_h P(h|X,\mathcal{A}_a)\frac{1}{2}2^{|\mathcal{X}|}\\&=\frac{1}{2}2^{|\mathcal{X}|}\sum_{\mathbf{x}\in\mathcal{X}-X}P(\mathbf{x})\sum_h P(h|X,\mathcal{A}_a)\\&=2^{|\mathcal{X}|-1}\sum P(\mathbf{x})\cdot 1\qquad(\textit{this is not dependent on }\mathcal{A}_a)\end{aligned}$$

On No Free Lunch Theorem

既然所有学习算法的期望性都一样，那我们为什么还研究学习算法？

NFL(No Free Lunch)定理有一个重要前提：所有问题出现的机会相同，或者说所有问题同等重要。但实际情况并非如此。

一般来说，我们只关注自己正在试图解决的问题，希望为它找一个解决方案，至于这个解决方案中其他问题甚至类似问题上是否为好方案，我们并不关心。

NFL定理的重要寓意

- 脱离具体问题去谈论“什么学习算法更好”毫无意义。
- 要谈论算法的相对优劣，必须针对具体的学习问题。
- 学习算法自身的归纳偏好与问题是否匹配，往往起到决定性作用。

Outline

模型评估与选择

在现实任务中，我们往往有多种学习算法可供选择，甚至对同一个学习算法，当使用不同的参数配置时，也会产生不同模型。
那么，我们应该选择哪一个学习算法，使用哪一种参数配置呢？这就是机器学习中的“模型选择”问题。

基本概念

- 错误率(error rate)：分类错误的样本个数占样本总数的比例
- 精度(accuracy)=1-错误率
- 误差：学习器的实际预测输出与样本的真实输出之间的差异
- 训练误差：学习器在训练集上的误差，也称经验误差
- 泛化误差：学习器在新样本上的误差

理想的解决方案是对候选模型的泛化误差进行评估，然后选择泛化误差最小的那个模型。
然而，我们事先并不知道新样本是什么样，所以这个办法行不通。

过拟合

我们实际能做的是努力使经验误差最小化。在很多情况下，我们可以学得一个经验误差很小，甚至对所有训练样本都正确分类，即分类错误率为零。
但是，这样的学习器在多数情况下对新样本并没有很好的表现。这是因为过拟合现象。

过拟合(overfitting)

- 为了使得泛化误差较小，我们应该从训练样本中尽可能学出适用于所有潜在样本的普遍规律。
- 当学习器把训练样本的一些自身特点当作了所有潜在样本都有的一般性质时候，就会导致泛化性能下降。

与“过拟合”相对的是欠拟合(underfitting)现象，是指对训练样本的一般性质尚未学好。欠拟合比较容易克服，可以通过加强训练强度。

Remark 3
过拟合是机器学习面临的关键障碍，各类学习算法都必然带有一些针对过拟合的措施；然而，过拟合是无法彻底避免的，我们能做的只是缓解。

评估方法

通常，我们可以通过实验测试来对学习器的泛化误差进行评估。

为此，需要使用一个测试集来测试学习器对新样本的判别能力。
以测试集上的测试误差(testing error)作为泛化误差的近似 。

- 假设测试样本是从样本真实分布中独立同分布采样得到的
- 测试集应该尽可能与训练集互斥，也即测试样本尽可能不出现在训练集中。

我们只有一个包含m个样例的数据集D，既要训练，又要测试，怎么做到呢？

通过对D进行适当处理，从中产生训练集S和测试集T。

- 留出法
- 交叉验证法
- 自助法

留出法

留出法(hold-out)：直接将数据集D划分为两个互斥的集合，其中一个集合作为训练集S，另一个作为测试集T。

需要注意的几点

- 训练/测试集的划分要尽可能保持数据分布的一致性。（比如分类任务中至少要保持样本类别的比例相似）。
  - 保留类别比例的采样方式通常称为分层采样(stratified sampling)。
- 即便给定训练/测试集的样本比例，仍存在多种划分方式对数据集D进行分割。单次使用留出法得到的估计结果往往不可靠。
  - 一般要采用若干次随机划分，重复进行实验评估后取平均值作留出法的评估结果。
- 如何确定训练/测试集的样本比例
  - 没有完美解决方案，常见做法是将大约2/3～4/5的样本用于训练，剩余样本用于测试。
  - 测试集至少应该包含30个样例[book "Machine Learning", Mitchell, 1997]。

交叉验证法

交叉验证法(cross validation)：

- 将数据集D划分为k个大小相似的互斥子集，即$D = D_1 \cup D_2 \cup \cdots \cup D_k$，$D_i \cap D_j = \emptyset$。每个子集$D_i$都尽可能保持数据分布一致性。
- 每次用k-1个子集的并集作为训练集，余下的那个子集作为测试集；这样就可以进行k次训练和测试，最终返回这k个测试结果的均值。
- 通常把交叉验证法称为“k折交叉验证法”。

注意点

- 存在多种划分方式对数据集D进行分割。
  - 一般要采用若干次不同的随机划分重复p次，这样的评估结果是这p次k折交叉验证结果的均值。常见的有“10次10折交叉验证法”

一个特例：留一法(Leave-One-Out)，令$k = m = |D|$，即每个子集只含一个样本

- 绝大多数情况下，留一法中被实际评估的模型与期望评估的用D训练出来的模型很相似。因此，评估结果往往被认为比较准备。
- 计算开销较大，对大数据集难以忍受。

自助法

我们希望评估的是用D训练出来的模型，但是在留出法和交叉验证法中，由于保留了一部分样本用于测试，实际评估的模型所使用的训练集比D小。

自助法(bootstrapping)：给定包含m个样本的数据集D，对它进行m次带放回的随机采样产生数据集D'作为训练集（$|D'| = |D|$）；$D \setminus D'$作为测试集。

D中有一部分样本会在D'中多次出现，而有些样本则不出现。

一个样本在m次采样中始终不被采到的概率是 $\lim_{n\to\infty}(1-\frac{1}{m})^m = \frac{1}{e} \approx 0.368$

注意点

- 自助法在数据集较小，难以有效划分训练/测试集时很有用。
- 自助法对初始数据产生多个不同的训练集，这对集成学习等方法有很大好处。
- 自助法产生的数据集改变了初始数据集的分布，会引入估计偏差。
  - 在初始数据量足够时，留出法和交叉验证法更常用一些。

算法调参

大多数学习算法都有些参数需要设定，参数配置不同，学得模型的性能往往有显著差别。
因此，在进行模型评估与选择时，除了要对适用学习算法进行选择，还需要对算法参数进行设定。

注意点

- 学习算法的很多参数是在实数范围内取值，不可能对每种配置训练模型
  - 对每个参数选择一个范围和变化步长，例如在[0,0.2]范围内以0.05为步长，则实际要评估的候选参数值只有5个。
  - 这种折中使得学习过程变得可行。
- 因为算法一般有多个参数，调参需要极大计算资源和工程量。

Remark 4
机器学习涉及两类参数：一类是算法的参数，也称为“超参数”，数目一般在*10*以内。另一类是模型的参数，数目可能很多，例如大型深度学习模型甚至有百亿个参数。

在模型选择完成之后，学习算法和参数配置已经选定，此时应该用数据集D重新训练模型。这个模型使用所有m个样本训练，这才是我们最终提交给用户的模型。

性能度量

对模型的泛化性能进行评估，不仅需要有效可行的实验评估方法，还需要有衡量模型泛化能力的评价标准，也就是性能度量(performance measure)。
在对比不同模型的能力时，使用不同的性能度量往往会导致不同的批判结果。

常用的性能度量

- 错误率与精度
- 查准率，查全率与F1
- 代价敏感错误率

错误率与精度

对于真实目标函数f，样例集D（$|D| = m$）
错误率是分类错误的样本数占样本总数的比例

$$E(f; D) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\mathbf{x_i}) \neq y_i).$$

精度是分类正确的样本数占样本总数的比例

$$acc(f; D) = \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\mathbf{x_i}) = y_i) = 1 - E(f; D).$$

更一般地，对于数据分布D和概率密度函数$p(\cdot)$，错误率与精度可分别描述为

$$E(f; D) = \int_{\mathbf{x}\sim D}\sum_{i=1}^{m}\mathbb{I}(f(\mathbf{x}) \neq y)p(\mathbf{x})d\mathbf{x}.$$

$$acc(f; D) = \int_{\mathbf{x}\sim D}\sum_{i=1}^{m}\mathbb{I}(f(\mathbf{x}) = y)p(\mathbf{x})d\mathbf{x} = 1 - E(f; D).$$

查准率，查全率与*F1*

以西瓜问题为例，有时候我们关心的是“挑出来的西瓜中有多少比例是好瓜”，或者“所有好瓜中有多少比例被挑了出来”，这就是查准率和查全率。

真实情况	预测结果	
	正例	反例
正例	TP（真正例）	FN（假反例）
反例	FP（假正例）	TN（真反例）

查准率P与查全率R分别定义为

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

F1是基于查准率与查全率的调和平均定义的：

$$\frac{1}{F1} = \frac{1}{2} \cdot (\frac{1}{P} + \frac{1}{R})$$

可得

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

代价敏感错误率

在现实人物中常常会遇到这样的情况：不同类型的错误所造成的后果不同。例如，在医疗诊断中，错误把患者诊断为健康人与错误把健康人诊断为患者，前者的后果可能是丧失了拯救生命的时机，后者则是增加了进一步检查的麻烦。为了权衡不同类型错误所造成的不同损失，可以为错误赋予非均等代价。

Table: 二分类代价矩阵

真实类别	预测类别 第0类	第1类
第0类	0	$cost_{01}$
第1类	$cost_{10}$	0

令D^+与D^-分别代表样例集D的正例子集和反例子集，则代价敏感错误率为

$$E(f;D;cost)=\frac{1}{m}(\sum_{\mathbf{x_i}\in D^+}\mathbb{I}(f(\mathbf{x_i})\neq y_i)\cdot cost_{01}+\sum_{\mathbf{x_i}\in D^-}\mathbb{I}(f(\mathbf{x_i})\neq y_i)\cdot cost_{10}).$$

代价敏感错误率

在现实人物中常常会遇到这样的情况：不同类型的错误所造成的后果不同。例如，在医疗诊断中，错误把患者诊断为健康人与错误把健康人诊断为患者，前者的后果可能是丧失了拯救生命的时机，后者则是增加了进一步检查的麻烦。为了权衡不同类型错误所造成的不同损失，可以为错误赋予非均等代价。

Table: 二分类代价矩阵

真实类别	预测类别 第0类	第1类
第0类	0	$cost_{01}$
第1类	$cost_{10}$	0

令D^+与D^-分别代表样例集D的正例子集和反例子集，则代价敏感错误率为

$$E(f;D;cost)=\frac{1}{m}(\sum_{\mathbf{x_i}\in D^+}\mathbb{I}(f(\mathbf{x_i})\neq y_i)\cdot cost_{01}+\sum_{\mathbf{x_i}\in D^-}\mathbb{I}(f(\mathbf{x_i})\neq y_i)\cdot cost_{10}).$$

比较检验

有了实验评估方法和性能度量，看起来就能对学习器的性能进行评估比较了。但具体怎么做比较呢？是直接取性能度量的值然后比大小吗？

几个因素

- 我们希望比较的是泛化性能，而实验评估只能获得测试集上的性能
- 测试集上的性能与测试集本身的选择有很大关系
- 很多学习算法本身有一定的随机性

统计假设验证为我们进行学习器性能比较提供了重要依据。
基于假设验证结果我们可以推断出，如果在测试集上观察到学习器A比学习器B好，则A的泛化性能是否在统计意义上优于B，以及这个结论的把握有多大。

常见的假设验证

- 二项检验
- t检验
- McNemar检验
- Friedman检验与Nemenyi检验

Outline

基础知识

- 给定训练样本集$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_m, y_m)\}$, $\mathbf{x}_i \in \mathcal{X}$ 我们主要讨论二分类问题，即$y_i \in \mathcal{Y} = \{-1, +1\}$。→概念学习。
- 假设$\mathcal{X}$中所有样本服从一个隐含未知的分布$\mathcal{D}$, D上所有样本都是独立地从这个分布上采样得到，即独立同分布（简称iid）样本。
- 令h为从$\mathcal{X}$到$\mathcal{Y}$的一个映射，其泛化误差为

$$E(h; \mathcal{D}) = P_{x \sim \mathcal{D}}(h(\mathbf{x}) \neq y)$$

- h在D上的经验误差为

$$\widehat{E}(h; D) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}(h(\mathbf{x}_i) \neq y_i)$$

- 由于D是$\mathcal{D}$的独立同分布采样，因此h的经验误差的期望等于其泛化误差的期望。
- 在上下文明确时，我们将$E(h; \mathcal{D})$ 和$\widehat{E}(h)$分别简记为$E(h)$ 和$\widehat{E}(h)$。
- 令ϵ为$E(h)$的上限，即$E(h) \leq \epsilon$；我们通常用ϵ表示预先设定的学得模型所应该满足的误差要求，亦称为“误差参数”。
- 若h在数据集D上的经验误差为0，则称h与D一致，否则称其与D不一致。

基础知识

- 令c表示“概念”，这是从样本空间$\mathcal{X}$到标记空间$\mathcal{Y}$的一个映射，它决定了示例$\mathbf{x}$的标记y。
- 若对任何样例$(\mathbf{x}, y)$都有$c(\mathbf{x}) = y$，则称c为目标概念；所有我们希望学得的目标概念所构成的集合称为“概念类”，用符号$\mathcal{C}$表示。
- 给定一个算法$\mathcal{A}$，它所考虑到所有可能概念的集合称为算法$\mathcal{A}$的“假设空间”，用符号$\mathcal{H}$表示。每个$h \in \mathcal{H}$称为“假设”。
- 由于学习算法事先并不知道概念类的真实存在，因此$\mathcal{H}$和$\mathcal{C}$通常是不同的。
- 若目标函数$c \in \mathcal{H}$，则$\mathcal{H}$中存在能将所有示例与真实标记一致的方式完全分开，这时称该问题对学习算法$\mathcal{A}$而言是“可分的”，也称为“一致的”；若$c \notin \mathcal{H}$，则称该问题对学习算法$\mathcal{A}$而言是“不可分的”，也称为“不一致的”。

PAC学习-基础概念

一个最基本的计算学习理论是概率近似正确(Probably Approximately Correct, PAC)理论[Valiant, 1984]

- 给定训练集D，我们希望学习算法$\mathcal{A}$学的的模型所对应的假设h尽可能接近目标概念c。
- 精确地学到目标概念c？不现实。各种因素：比如因为训练集D上包含的有限数量样例，通常存在一些在D上等价的假设，学习算法无法对他们进行区别。
- 我们希望以较大的概率学得误差满足预设上限的模型，这就是“概率”“近似正确”的含义。

Definition 1

PAC辨识：对于$0 < \epsilon, \delta < 1$，所有$c \in \mathcal{C}$和分布$\mathcal{D}$，若学习算法$\mathcal{A}$的输出$h \in \mathcal{H}$满足$P\big(E(h) \leq \epsilon\big) \geq 1 - \delta$则称学习算法$\mathcal{A}$从假设空间$\mathcal{H}$中PAC辨识概念类$\mathcal{C}$。

PAC学习-基础概念

Definition 2

PAC可学习：令m表示从分布$\mathcal{D}$中独立同分布采样得到的样例数目，$0 < \epsilon, \delta < 1$，对于所有分布$\mathcal{D}$，如果存在算法$\mathcal{A}$和多项式函数poly，使得对于任何$m \geq poly(1/\epsilon, 1/\delta, size(\mathbf{x}), size(c))$，算法$\mathcal{A}$能从其假设空间$\mathcal{H}$中PAC辨识概念类$\mathcal{C}$，则称概念类$\mathcal{C}$ 对假设空间$\mathcal{H}$而言是PAC可学习的，有时间称概念类$\mathcal{C}$是PAC可学习的。

Note: PAC可学习需要的样例数目m与误差ϵ，置信度δ，数据（示例）本身的复杂度$size(\mathbf{x})$，目标概念的复杂度$size(c)$有关。条件是限定在多项式函数内。

PAC学习-基础概念

Definition 3

PAC学习算法：若学习算法$\mathcal{A}$使得概念类$\mathcal{C}$为PAC可学习的，且$\mathcal{A}$的运行时间也是多项式函数$poly(1/\epsilon, 1/\delta, size(\mathbf{x}), size(c))$，则称概念类$\mathcal{C}$是高效PAC可学习的，称算法$\mathcal{A}$为概念类$\mathcal{C}$的PAC学习算法

Definition 4

样本复杂度：

满足PAC学习算法$\mathcal{A}$所需要的$m \geq poly(1/\epsilon, 1/\delta, size(\mathbf{x}), size(c))$中的最小m，称为学习算法$\mathcal{A}$ 的样本复杂度。

PAC学习-假设空间

PAC学习中的一个关键因素是假设空间$\mathcal{H}$的复杂度，因为$\mathcal{H}$包含了学习算法$\mathcal{A}$所有可能输出的假设。

- 理想情况：$\mathcal{H} = \mathcal{C}$，称为“恰PAC学习”，意味着学习算法的能力恰好与学习任务匹配。
- 现实中，我们对概念类$\mathcal{C}$通常一无所知，理想情况基本不可能。
- 考虑$\mathcal{H} \neq \mathcal{C}$的情况
  - $\mathcal{H}$越大，其包含任意目标概念的可能性就越大；
  - $\mathcal{H}$越大，从中找到目标概念的难度也越大。
  - $\mathcal{H}$有限时，我们称$\mathcal{H}$为“有限假设空间”；否则称为“无限假设空间”。

PAC学习-可分情形

可分情形意味着目标概念c属于假设空间$\mathcal{H}$，即$c \in \mathcal{H}$。
对于这种情形，我们关心的是：给定一个训练集D，如何找出满足误差参数的假设呢？

一个简单的策略：只保留与D一致的假设，剔除与D不一致的假设。

- 若D足够大，则可不断借助D的样例剔除不一致的假设，直到$\mathcal{H}$中仅剩下一个假设为止，这个假设就是目标概念c。
- 然而，通常情况下，训练集规模有限，假设空间$\mathcal{H}$中可能存在多个与D一致的假设，对这些假设，无法根据D对它们进行区分。

Question: 需要多少样例才能学得目标概念c的有效近似呢？

对于PAC学习而言，问题就是，需要多少样例，才能使学习算法能以概率$1-\delta$找到目标假设的ϵ近似？

PAC学习-可分情形

Question: 需要多少样例，才能使学习算法能以概率$1-\delta$找到目标假设的ϵ近似（误差小于ϵ）？
Recall: 对于算法的假设空间$\mathcal{H}$，只保留与D一致的假设。
Idea：我们只要保证，泛化误差大于ϵ但与训练集D一致的所有假设出现的概率之和不大于δ。

对于一个误差大于ϵ的假设$h \in \mathcal{H}$，对分布$\mathcal{D}$上随机采样点任何样例$(\mathbf{x}, y)$,

$$P\big(h(\mathbf{x}) = y\big) = 1 - P\big(h(\mathbf{x}) \neq y\big) = 1 - E(h) < 1 - \epsilon$$

所以，任一个$h \in \mathcal{H}$，h误差大于ϵ且h与训练集D(包含m个样例)一致的概率为

$$P\big((h(\mathbf{x}_1) = y_1) \wedge \cdots \wedge (h(\mathbf{x}_m) = y_m)\big) = \big(1 - P(h(\mathbf{x}) \neq y)\big)^m < (1-\epsilon)^m$$

我们关心的，泛化误差大于ϵ但与训练集D一致的所有假设出现的概率之和

$$P\big(h \in \mathcal{H} : E(h) > \epsilon \wedge \widehat{E}(h) = 0\big) < |\mathcal{H}|(1-\epsilon)^m < |\mathcal{H}|e^{-m\epsilon}$$

令$|\mathcal{H}|e^{-m\epsilon} \leq \delta$, 可得

$$m \geq \frac{1}{\epsilon}\ln\frac{|\mathcal{H}|}{\delta}$$

所以，有限假设空间都是PAC可学习的，所需样本数目如上所示。

PAC学习-不可分情形

对于较为困难的学习问题，目标概念c往往不存在于假设空间$\mathcal{H}$中。
这种情况下，$\mathcal{H}$的任意一个假设都会在训练集上出现或多或少的错误。
由Hoeffding不等式，可得

Lemma 5
若训练集D包含m个从分布$\mathcal{D}$独立同分布采样而得的样例，$0 < \epsilon < 1$，则对于任意$h \in \mathcal{H}$，有

$$P\left(\widehat{E}(h) - E(h) \geq \epsilon\right) \leq exp(-2m\epsilon^2)$$

$$P\left(E(h) - \widehat{E}(h) \geq \epsilon\right) \leq exp(-2m\epsilon^2)$$

$$P\left(\left|E(h) - \widehat{E}(h)\right| \geq \epsilon\right) \leq 2exp(-2m\epsilon^2)$$

Corollary 6
若训练集D包含m个从分布$\mathcal{D}$独立同分布采样而得的样例，$0 < \epsilon < 1$，则对于任意$h \in \mathcal{H}$，下面式子至少以$1 - \delta$的概率成立：

$$\widehat{E}(h) - \sqrt{\frac{ln(2/\delta)}{2m}} \leq E(h) \leq \widehat{E}(h) + \sqrt{\frac{ln(2/\delta)}{2m}}$$

表明，样例数目m较大时，h的经验误差是对其泛化误差的很好近似。

PAC学习-不可分情形

- 当$c \notin \mathcal{H}$时，学习算法$\mathcal{A}$无法学得目标概念c的ϵ近似。
- 但是，给定假设空间$\mathcal{H}$，必存在一个泛化误差最小的假设，找出此假设的ϵ近似也不失为一个较好的目标。
- 以此为目标可将PAC学习推广到$c \notin \mathcal{H}$的情况，这中学习称为“不可知学习”(agnostic learning)。

Definition 7

不可知PAC可学习(agnostic PAC learnable)：令m表示从分布$\mathcal{D}$中独立同分布采样得到的样例数目，$0 < \epsilon, \delta < 1$，对于所有分布$\mathcal{D}$，如果存在算法$\mathcal{A}$和多项式函数poly，使得对于任何$m \geq poly(1/\epsilon, 1/\delta, size(\mathbf{x}), size(c))$，算法$\mathcal{A}$能从其假设空间$\mathcal{H}$中输出满足下式的假设$h$:

$$P(E(h) - \min_{h' \in \mathcal{H}} E(h') \leq \epsilon) \geq 1 - \delta$$

则称假设空间$\mathcal{H}$是不可知PAC可学习的。

类似地，我们可以定义不可知PAC学习算法，而且满足上述要求的最小m称为学习算法的样本复杂度。

VC维

- 现实学习任务中所面临的通常是无限假设空间，例如实数域的所有区间，$\mathbb{R}^d$空间中的所有线性超平面。
- 欲对此种情形的可学习性进行研究，需要度量假设空间的复杂度。
- 最常见的是考虑假设空间的“VC维”[Vapnik and Chervonenkis, ”On the uniform convergence of relative frequencies of events to their probabilities”, Theory of Probability and Its Applications 1971]

先看几个概念:增长函数(growth function),对分(dichotomy)和打散(shattering)

给定假设空间$\mathcal{H}$和示例集$D = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$，$\mathcal{H}$中每个假设都能对$D$中示例赋予标记，对$D$的标记结果记为

$$h|_D = \{\big(h(\mathbf{x}_1), h(\mathbf{x}_2), \ldots, h(\mathbf{x}_m)\big)\}$$

随着m的增大，$\mathcal{H}$中所有假设对D中示例所能赋予标记的可能结果数也增大。

VC维

Definition 8

对所有$m \in \mathbb{N}$，假设空间$\mathcal{H}$的增长函数为

$$\Pi_{\mathcal{H}}(m) = \max_{\{\mathbf{x}_1,\cdots,\mathbf{x}_m\}\subseteq\mathcal{X}} |\{(h(\mathbf{x}_1),\cdots,h(\mathbf{x}_m))|h \in \mathcal{H}\}|$$

增长函数表示假设空间$\mathcal{H}$对m个示例所能赋予标记的最大可能结果数。显然，这个最大可能结果数越大，$\mathcal{H}$的表示能力越强，对学习任务的适应能力也越强。增长函数描述了假设空间的表示能力，由此反映了假设空间的复杂度。

可以利用增长函数来估计经验误差与泛化误差之间的关系

Theorem 9

对假设空间$\mathcal{H}$，$m \in \mathbb{N}$，$0 < \epsilon < 1$和任意$h \in \mathcal{H}$，有

$$P(|E(h) - \hat{E}(h)| > \epsilon) \leq 4\Pi_{\mathcal{H}}(2m)exp(-\frac{m\epsilon^2}{8})$$

Note：假设空间$\mathcal{H}$中不同假设对D中示例赋予标记的结果可能相同，也可能不同。尽管可能包含无穷多个假设，但其对D中示例赋予标记的可能结果是有限的。比如二分类问题的m个示例，最多有2^m个可能结果。

VC维

- 对二分类问题来说，对D中示例赋予标记的每种可能结果称为对D的一种“对分”。（把D中示例分为两类，因此称为对分。）
- 若假设空间$\mathcal{H}$能实现实例集D上的所有对分，即$\Pi_{\mathcal{H}}(m) = 2^m$，则称实例集$D$能被假设空间$\mathcal{H}$“打散”。（也就是说，D上的标记结果，在假设空间中总有一个h能产生）

Definition 10

假设空间$\mathcal{H}$的VC维是能被$\mathcal{H}$打散的最大实例集的大小

$$\mathrm{VC}(\mathcal{H}) = \max\{m : \Pi_{\mathcal{H}}(m) = 2^m\}$$

- $VC(\mathcal{H}) = d$表明存在大小为d的实例集能被假设空间$\mathcal{H}$打散，并不是所有大小为d的实例集都能被假设空间$\mathcal{H}$打散。
- VC维的物理意义：VC维可以反映假设$\mathcal{H}$的强大程度(powerfulness)，VC维越大，$\mathcal{H}$也越强，因为它可以打散更多的点。
- VC维与数据分布$\mathcal{D}$无关。

VC维-两个例子

Example 11

实数域中的区间$[a,b]$：

令$\mathcal{H}$表示实数域中所有闭区间构成的集合$\{h_{[a,b]} : a,b \in \mathbb{R}, a \leq b\}$, $\mathcal{X} = \mathbb{R}$。对于$x \in \mathcal{X}$，若$x \in [a,b]$, 则$h_{[a,b]} = +1$，否则$h_{[a,b]} = -1$。

令$x_1 = 0.5, x_2 = 1.5$，则假设空间$\mathcal{H}$可以将实例集$\{x_1, x_2\}$打散，由这些假设可以实现：$h_{[0,1]}$，$h_{[0,2]}$，$h_{[1,2]}$和$h_{[2,3]}$。所以假设空间$\mathcal{H}$的VC维至少为2；

对任意大小为3的实例集$\{x_3, x_4, x_5\}$，不妨设$x_3 < x_4 < x_5$，则$\mathcal{H}$中不存在任何假设$h_{[a,b]}$能实现对分结果$\{(x_3,+),(x_4,-),(x_5,+)\}$。

于是$\mathcal{H}$的VC维为2。

Example 12

二维实平面上的线性划分：

令$\mathcal{H}$表示二维实平面上所有线性划分构成的集合，$\mathcal{X} = \mathbb{R}^2$。

画图可以知道，存在大小为3的实例集可以被$\mathcal{H}$打散。

但不存在大小为4的实例集可以被$\mathcal{H}$打散。

于是$\mathcal{H}$的VC维为3。

实际上，可以严格证明，$\mathbb{R}^d$空间中线性超平面构成的假设空间的VC维是d+1。

基于VC维的泛化误差

VC维和增长函数有密切联系: $\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} \leq \left(\frac{em}{d}\right)^d$

由此，我们可以从基于增长函数的误差估计来得到基于VC维的误差估计

Theorem 13
若假设空间$\mathcal{H}$的*VC*维为d，则对任意$m > d, 0 < \delta < 1$和$h \in \mathcal{H}$，有

$$P\left(|E(h) - \widehat{E}(h)| \leq \sqrt{\frac{8d \ln \frac{2em}{d} + 8 \ln \frac{4}{\delta}}{m}}\right) \geq 1 - \delta$$

证明很简单，我们已经知道

$$P(|E(h) - \widehat{E}(h)| > \epsilon) \leq 4\Pi_{\mathcal{H}}(2m)\exp\left(-\frac{m\epsilon^2}{8}\right) \leq 4\left(\frac{2em}{d}\right)^d \exp\left(-\frac{m\epsilon^2}{8}\right)$$

令$4\left(\frac{2em}{d}\right)^d \exp\left(-\frac{m\epsilon^2}{8}\right) = \delta$，解得$\epsilon = \sqrt{\frac{8d \ln \frac{2em}{d} + 8 \ln \frac{4}{\delta}}{m}}$。
以上定理表明

- 其他条件相同的情况下，VC维大（模型越复杂），泛化误差越大。Powerful $\mathcal{H}$ not always good!
- 模型较复杂时(VC维较大)，需要更多的训练数据。

稳定性分析

- 基于VC维的分析得到的结果和具体学习算法无关。这使得人们可以脱离具体算法的设计来考虑学习方法和问题本身的性质。
- 另一方面，如果希望获得与具体算法有关的结果，则需要其他分析。这方面的一个值得关注的方向就是稳定性(stability)分析。
- 稳定性考虑的是算法在输入发生变化时，输出是否会发生较大的变化。

学习算法的输入是训练集，我们先定义训练集的两种变化。
给定$D = \{\mathbf{z}_1 = (\mathbf{x}_1, y_1), \mathbf{z}_2 = (\mathbf{x}_2, y_2), ..., \mathbf{z}_m = (\mathbf{x}_m, y_m)\}$，$\mathbf{x}_i \in \mathcal{X}$是来自分布$\mathcal{D}$的独立同分布示例，$y_i = \{+1, -1\}$。
对假设空间$\mathcal{H}$和学习算法$\mathcal{A}$，令$\mathcal{A}_D \in \mathcal{H}$表示算法$\mathcal{A}$基于训练集$D$从假设空间$\mathcal{H}$学得到假设。

训练集的两种变化

- $D^{\setminus i}$表示从D中移除第i个样例得到的集合。
- D^i表示用分布$\mathcal{D}$中随机采样的一个样例替换D中第i个样例得到的集合。

稳定性分析

损失函数$\ell(\mathcal{A}_D(\mathbf{x}), y)$: $\mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$刻画了假设$\mathcal{A}_D$的预测标记$\mathcal{A}_D(\mathbf{x})$与真实标记$y$之间的差别，简记为$\ell(\mathcal{A}_D, \mathbf{z})$。在这个函数的基础上，我们可以定义下面几种损失。

- 泛化损失：$\ell(\mathcal{A}, D) = \mathbb{E}_{\mathbf{x}\in\mathcal{X}, \mathbf{z}=(\mathbf{x},y)}[\ell(\mathcal{A}_D, \mathbf{z})]$
- 经验损失：$\hat{\ell}(\mathcal{A}, D) = \frac{1}{m}\sum_{i=1}^{m} \ell(\mathcal{A}_D, \mathbf{z_i})$
- 留一损失：$\ell_{loo}(\mathcal{A}, D) = \frac{1}{m}\sum_{i=1}^{m} \ell(\mathcal{A}_{D\setminus i}, \mathbf{z_i})$

Definition 14

对任何$\mathbf{x} \in \mathcal{X}, \mathbf{z} = (\mathbf{x}, y)$，若学习算法$\mathcal{A}$满足

$$|\ell(\mathcal{A}_D, \mathbf{z}) - \ell(\mathcal{A}_{D\setminus i}, \mathbf{z})| \leq \beta, \quad i = 1, 2, ..., m$$

则称算法$\mathcal{A}$关于损失函数ℓ满足β-均匀稳定性。

Note：移除示例的稳定性包含了替换实例的稳定性。

关于计算学习理论

- 1984年，Valiant提出PAC学习，由此产生了“计算学习理论”这个机器学习的分支领域。”A theory of the learnable”, Communication of the ACM.
- 一本很好的入门教材: Kearns, M.J. and U. V. Vazirani, An Introduction to Computational Learning Theory. MIT Press. (1994)
- 计算学习理论最重要的学术会议是诡计计算学习理论会议(COLT)。