C1-9

# Machine Learning
# by Andrew Ng, Stanford Engineering

Xiaojie Zhou

szxjzhou@163.com

2016.9.13

# 第十集：特征选择

- Vapnik-Chervonenkis维(Vapnik-Chervonenkis (VC) Dimension)
- 模型选择(Model Selection)
- 交叉验证(Cross Volidation)
- 特征选择(Feature Selection)

# 第十集：特征选择

- Vapnik-Chervonenkis维
  - 上一节中考虑的是假设类H为有限假设类的情况，现在来探讨假设类H为无限假设类的情况
    - 有限假设类指其中只包含有限个假设，每种假设都对应某种模型和参数
  - 但实际上在计算机世界里绝对的无限并不存在
    - 对于一个由d个实数作为参数的假设类来说，从数学的角度看是无限的，但是由于计算机由64位表示一个double类型，因此总体的特征也就64d位，对应的假设个数为$2^{64d}$个
    - 由此可得到对应的在给定训练误差和泛化误差的距离γ，要求训练误差得到的最优假设与泛化误差最优的假设的泛化误差间距离小于等于2γ的概率至少为1-δ的情况下至少需要的样本数

$$\begin{aligned} m &\geq \frac{1}{2\gamma^2}\log\frac{2k}{\delta} \\ &= O\left(\frac{1}{\gamma^2}\log\frac{k}{\delta}\right) \end{aligned} \longrightarrow m \geq O\left(\frac{1}{\gamma^2}\log\frac{2^{64d}}{\delta}\right) = O\left(\frac{d}{\gamma^2}\log\frac{1}{\delta}\right) = O_{\gamma,\delta}(d)$$

# 第十集：特征选择

- Vapnik-Chervonenkis维
  - 由此可以得到一个大致的结论，对于一个由d个实数作为参数的假设类来说所需的样本复杂度与d的关系为线性
    - 即样本数目要随着d的增加而线性增加
  - 但是这个结论并不完全准确，因为这一结论依赖于假设类中参数的个数

$$1\{\theta_0 + \theta_1 x_1 + \cdots \theta_n x_n \geq 0\}$$

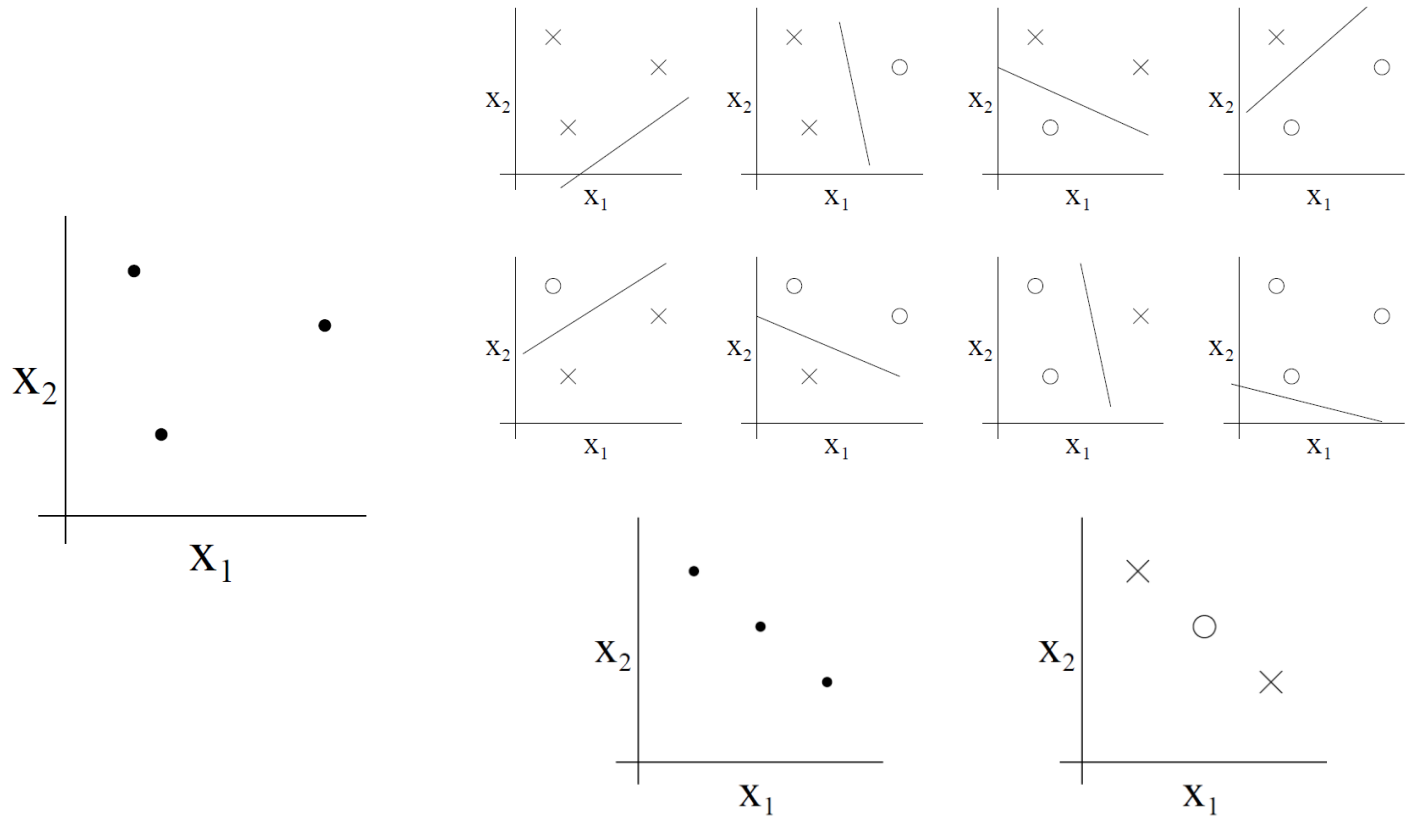
$$1\{(u_0^2 - v_0^2) + (u_1^2 - v_1^2)x_1 + \cdots (u_n^2 - v_n^2)x_n \geq 0\}$$

即使是定义相同事物的假设，参数个数都有可能不同。如左图，表示的都是n阶曲线，但一个拥有n+1个参数，另一个则拥有2n+2个参数

  - 由此进行一个更为广泛的定义：对于定义域X上任选的样本组成的集合S，当假设类H中总存在某一假设能识别出集合S的任意标记方法，称假设类H分散(shatter)了集合S
  - 而对于一个假设类H来说，它能分散的最大集合S的模称为该假设类H的Vapnik-Chervonenkis维（简称VC维，记为VC(H)）

# 第十集：特征选择

- Vapnik-Chervonenkis维
  - 下面来看一个分散和Vapnik-Chervonenkis维的例子




从中可以看出一阶直线假设h(x)=1{θ0+θ1x≥0}所构造的假设类H可以识别出如图所示的三个样本组成的集合S的任意标记方法，因此称由一阶直线假设h(x)=1{θ0+θ1x≥0}所构造的假设类H分散了三个样本组成的集合S.与此同时由于找不出一个由四个样本组成的集合S能被一阶直线假设所构造的假设类H分散，因此假设类H来说能分散的最大集合S的模为3，即该假设类H的VC维VC(H)=3

# 第十集：特征选择

- Vapnik-Chervonenkis维
  - 对于Vapnik-Chervonenkis维来说满足下面的定理
    - 定理：给定假设类H，令d表示假设类H的VC维VC(H)，在概率至少为1-δ的情况下对于假设类中任意假设满足：

$$|\varepsilon(h) - \hat{\varepsilon}(h)| \leq O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} + \frac{1}{m}\log\frac{1}{\delta}}\right)$$

$$\varepsilon(\hat{h}) \leq \varepsilon(h^*) + O\left(\sqrt{\frac{d}{m}\log\frac{m}{d} + \frac{1}{m}\log\frac{1}{\delta}}\right)$$

    - 从中可以进一步得到样本复杂度与d的关系为线性，即样本数目要随着假设类H的VC维的增加而线性增加
      - 而在实际情况下VC维一般与模型的参数相关，n个参数对应n+1的VC维

# 第十集：特征选择

- 模型选择
  - 通过经验风险最小化的研究得到了对于训练误差和泛化误差的基本关系，并从这个关系中通过VC维找出了所需的样本个数
  - 而对于机器学习问题来说更加本质的问题在于我们应该选择一个怎样的模型，特别是在样本数量达不到要求时
    - 比如说在广义线性模型中选取一条几阶的曲线进行拟合；在局部加权回归中如何控制样本的权值；在SVM中如何控制软间隔的参数C
    - 在开始这个问题前先对这个问题做一个更加具体的定义：假设一个由有限个模型组成的集合M
      - 对于广义线性模型M中各模型可以代表不同阶的曲线；对于局部加权回归M中各模型可以代表不同样本权值；对于SVM中M中各模型可以代表不同软间隔的参数C；当然M中也可以是多种模型的集合
      - 但这里面假设的是有限的集合M，但实际上也可以类似推广到无限的情况

# 第十集：特征选择

- 交叉验证
  - 对于模型选择来说一种最容易想到的办法是通过训练集将所有的模型都训练一遍，得到各自最优的参数，然后再通过比较训练误差来选择模型
    - 但这种方法非常容易选到高阶过拟合的模型（因为过拟合的模型训练误差往往都比较小）
  - 那么既然通过完整的训练集直接比较最终的训练误差不可行，那么就会想能不能构造出已知标准答案的测试集对训练效果进行测试
    - 这种方法也就是hold-out交叉验证(hold-out cross validation，也被称为“简易交叉验证”(simple cross validation))，方法如下：
      - 1、将训练集分成两个子集，一部分作为训练集，另一部分作为交叉验证集
        - 交叉验证集中数据一般占总体的1/3~1/4之间
      - 2、用训练集的数据对全部的模型进行训练
      - 3、用交叉验证集的数据检验每个模型的训练效果，选取最优的模型
        - 最终可以直接输出结果，也可以选取最优模型后再通过完整的数据集进行重新训练后再输出结果

# 第十集：特征选择

- 交叉验证
  - hold-out交叉验证具有下列优劣
    - hold-out交叉验证的好处在于简单好理解，但问题在于只使用了原样本集中一部分的样本进行训练，而这部分样本训练出来的最优模型并不一定是对于全部样本进行训练的最优模型（尤其是对于样本集中样本数本来就不多的情况）
  - 由此提出k-fold交叉验证(k-fold cross validation)的方法
    - 1、将训练集S均分成k个子集，记为S1, …, Sk
      - 常见的选择有：k=5, k=10
    - 2、对于每个模型m，依次选定某一个子集Si，用其他子集S1, …, Si-1, Si+1, …, Sk对模型m进行训练，训练结束后再用子集Si进行验证，从而得到该模型在该子集下的误差εi.该模型m的误差为所有子集下的误差的均值(ε1+…+εk)/k
    - 3、检验每个模型的误差，选取最优模型

# 第十集：特征选择

- 交叉验证
  - 而对于k-fold交叉验证具有下列优劣
    - 其好处在于训练的样本数比起hold-out交叉验证更大，不过算法复杂度很高
  - 而对于k-fold交叉验证来说还有一种极端情况，将子集个数k设置为训练集S的大小
    - 在这种情况下每个子集中只有一个元素，因此对于某一模型在某一子集下的误差进行验证时只验证一个样本，因此这种方法称为leave-one-out交叉验证(leave-one-out cross validation)
      - 这样做训练的样本数和算法复杂度都进一步的提升，比较适合样本数目很少的情况

# 第十集：特征选择

- 特征选择
  - 交叉验证可以用来选择合适的模型类型和参数对数据特征进行展示，然而这建立在数据反映了真实事物特征的基础上
    - 有些情况下数据中有些特征和我们希望反映的数据特征可能关系不大，如果对于这些无关的特征也进行拟合，一方面花了无谓的时间，另一方面可能出现过拟合的问题。因此这就需要在进行具体的特征学习前对数据进行特征选择
      - 比如在垃圾邮件分类问题中，特征为字典中某词是否出现，高达50000.但里面很多词（比如"a", "an", "of"）并不能提供任何有效的信息
    - 对于特征选择问题来说，给定n个特征，可能的特征子集数为$2^n$
      - 对此一种最简单的方法是将所有的特征子集都进行训练，选出最优的特征子集，但这样做时间复杂度非常高，由此将会用到启发式的方法进行解决

# 第十集：特征选择

- 特征选择
  - 而我们最容易想到的启发式方法就是每次找当前最优的特征加入到特征集合中，我们可以通过设定阈值来控制特征集中特征个数，这种方法也被称为前向搜索法(forward search)
    - 1、初始化特征集F为空集
    - 2、尝试将不在特征集中的某一特征fi放入特征集中，运用任意一种交叉验证方法得到当前特征集合下的误差ε(fi)
      - 交叉验证时只能用到在当前特征集合中的特征进行训练
    - 3、将第2步中放入的特征fi拿出特征集合，尝试再将下一个不在特征集中的某一特征放入特征集中，运用交叉验证方法得到当前特征集合下的误差
    - 4、得到所有误差后选择最优的特征永久加入到特征集合中，如果特征数目已经达到阈值即可结束，否则回到第2步继续挑选下一个特征

# 第十集：特征选择

- 特征选择
  - 事实上，前向搜索法是封装模型特征选择方法(wrapper model feature selection)的一个特例
    - 封装模型特征选择方法中交叉验证将被放在循环中，需要循环地进行特征的学习以选取到最优的或者淘汰掉最劣的特征（这个对应后向选择法(backward search)）。这种算法的准确但是具有很高的运算量
      - 这里特别注明，这种算法的准确性是相对的。封装模型特征选择方法作为一种启发式的算法，并不能保证这样计算后能得到最优的特征子集
    - 由此为了降低运算量，提出了过滤特征选择法(filter feature selection)，这种方法相较封装模型方法来说没那么准确，但运算量也少得多
      - 在过滤特征选择法中直接估计每个特征对于最终结果y的影响程度，然后选取影响程度大的特征
      - 其中需要选择多少特征也可以通过类似于封装模型特征选择方法进行判断（依次计算选取一个特征的误差、两个特征的误差…），但由于采取该方法的特征个数本来就很大，因此一般不做这一步，直接手动指定

# 第十集：特征选择

- 特征选择
  - 由此可见特征选择法的关键在于选取怎样的指标用于反映某一特征对于最终结果y的影响程度
    - 一种简单的方法是直接计算该特征与最终结果y的相关系数
    - 而更为常用且准确的方法为计算该特征与最终结果y的互信息(mutual information)

$$\mathrm{MI}(x_i, y) = \sum_{x_i \in \{0,1\}} \sum_{y \in \{0,1\}} p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)}$$

$$\mathrm{MI}(x_i, y) = \mathrm{KL}\left(p(x_i, y) || p(x_i)p(y)\right)$$

这里面假设了特征和结果均为二元的情况，但实际上我们很容易将其推广到多元的情况。其计算也非常简单，直接从训练集中进行统计即可。实际上互信息是Kullback-Leibler散度(Kullback-Leibler (KL) divergence)的一种形式。对于互信息来说，如果xi与y的相关程度低，互信息的值也低（特别地，当xi与y独立时p(xi,y)=p(xi)p(y)，互信息为0）；如果xi与y的相关程度高，互信息的值也高