台湾大学林轩田机器学习基石课程学习笔记8 -- Noise and Error

> 作者：红色石头
> 微信公众号：AI有道（ID：redstonewill）

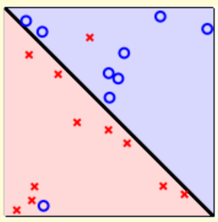
上一节课，我们主要介绍了VC Dimension的概念。如果Hypotheses set的VC Dimension是有限的，且有足够多N的资料，同时能够找到一个hypothesis使它的$E_{in} \approx 0$，那么就能说明机器学习是可行的。本节课主要讲了数据集有Noise的情况下，是否能够进行机器学习，并且介绍了假设空间H下演算法A的Error估计。

一、Noise and Probablistic target

上节课推导VC Dimension的数据集是在没有Noise的情况下，本节课讨论如果数据集本身存在Noise，那VC Dimension的推导是否还成立呢？

首先，Data Sets的Noise一般有三种情况：

- **由于人为因素，正类被误分为负类，或者负类被误分为正类；**
- **同样特征的样本被模型分为不同的类；**
- **样本的特征被错误记录和使用。**



briefly introduced **noise** before **pocket** algorithm

age	23 years
gender	female
annual salary	NTD 1,000,000
year in residence	1 year
year in job	0.5 year
current debt	200,000

credit? $\{\text{no}(-1), \text{yes}(+1)\}$

but more!

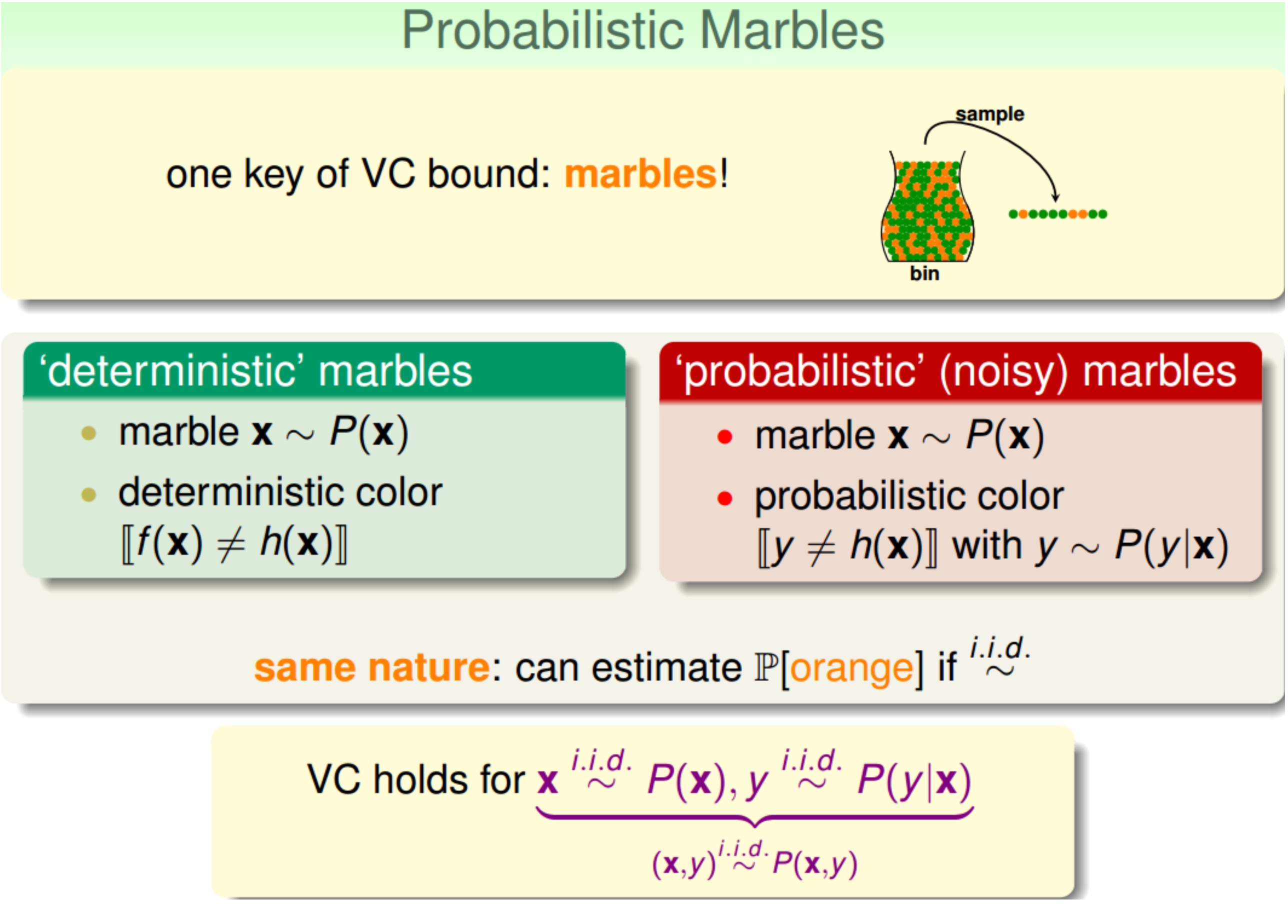
- **noise in** y: good customer, 'mislabeled' as bad?
- **noise in** y: same customers, different labels?
- **noise in x**: inaccurate customer information?

does VC bound work under **noise**?

之前的数据集是确定的，即没有Noise的，我们称之为Deterministic。现在有Noise了，也就是说在某点处不再是确定分布，而是概率分布了，即对每个(x，y)出现的概

率是$P(y|x)$。

因为Noise的存在，比如在x点，有0.7的概率y=1，有0.3的概率y=0，即y是按照$P(y|x)$分布的。数学上可以证明如果数据集按照$P(y|x)$概率分布且是iid的，那么以前证明机器可以学习的方法依然奏效，VC Dimension有限即可推断Ein和$Eout$是近似的。



$P(y|x)$称之为目标分布（Target Distribution）。它实际上告诉我们最好的选择是什么，同时伴随着多少noise。其实，没有noise的数据仍然可以看成“特殊”的$P(y|x)$概率分布，即概率仅是1和0.对于以前确定的数据集：

$$P(y|x) = 1, for\ y = f(x)$$

$$P(y|x) = 0, for\ y \neq f(x)$$

Target Distribution $P(y|\mathbf{x})$

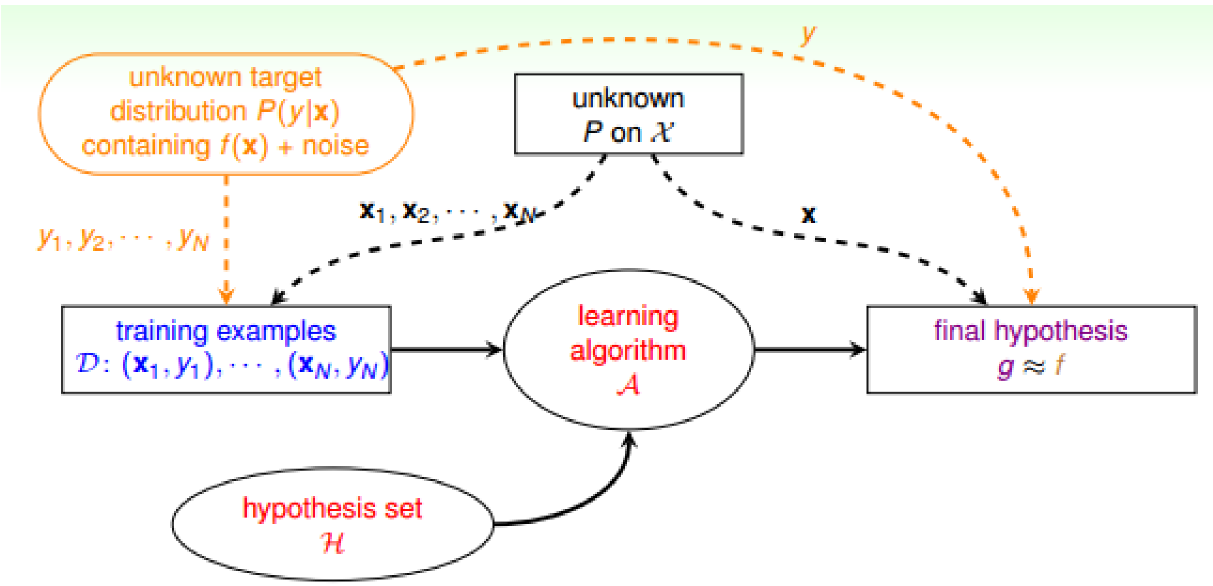
characterizes behavior of **'mini-target'** on one $\mathbf{x}$

- can be viewed as 'ideal mini-target' + noise, e.g.
  - $P(\circ|\mathbf{x}) = 0.7$, $P(\times|\mathbf{x}) = 0.3$
  - ideal mini-target $f(\mathbf{x}) = \circ$
  - 'flipping' noise level $= 0.3$
- deterministic target f: **special case of target distribution**
  - $P(y|\mathbf{x}) = 1$ for $y = f(\mathbf{x})$
  - $P(y|\mathbf{x}) = 0$ for $y \neq f(\mathbf{x})$

goal of learning:

predict **ideal mini-target (w.r.t. $P(y|\mathbf{x})$)**
on **often-seen inputs (w.r.t. $P(\mathbf{x})$)**

在引入noise的情况下，新的学习流程图如下所示：



VC still works, **pocket algorithm explained :-)**

二、ERROR Measure

机器学习需要考虑的问题是找出的矩g与目标函数f有多相近，我们一直使用E_{out}进行误差的估计，那一般的错误测量有哪些形式呢？

我们介绍的矩g对错误的衡量有三个特性：

- **out-of-sample：样本外的未知数据**

- **pointwise：对每个数据点x进行测试**
- **classification：看prediction与target是否一致，classification error通常称为0/1 error**

- how well? previously, considered out-of-sample measure

$$E_{\text{out}}(g) = \mathop{\mathcal{E}}_{\mathbf{x}\sim P} [\![g(\mathbf{x}) \neq f(\mathbf{x})]\!]$$

- more generally, **error measure** $E(g, f)$
- naturally considered
  - out-of-sample: averaged over unknown **x**
  - pointwise: evaluated on one **x**
  - classification: $[\![$prediction $\neq$ target$]\!]$

PointWise error实际上就是对数据集的每个点计算错误并计算平均，Ein和$Eout$的pointwise error的表达式为：

in-sample

$$E_{\text{in}}(g) = \frac{1}{N}\sum_{n=1}^{N} \text{err}(g(\mathbf{x}_n), f(\mathbf{x}_n))$$

out-of-sample

$$E_{\text{out}}(g) = \mathop{\mathcal{E}}_{\mathbf{x}\sim P} \text{err}(g(\mathbf{x}), f(\mathbf{x}))$$

pointwise error是机器学习中最常用也是最简单的一种错误衡量方式，未来课程中，我们主要考虑这种方式。pointwise error一般可以分成两类：0/1 error和squared error。0/1 error通常用在分类（classification）问题上，而squared error通常用在回归（regression）问题上。

0/1 error

$$\text{err}(\tilde{y}, y) = [\![\tilde{y} \neq y]\!]$$

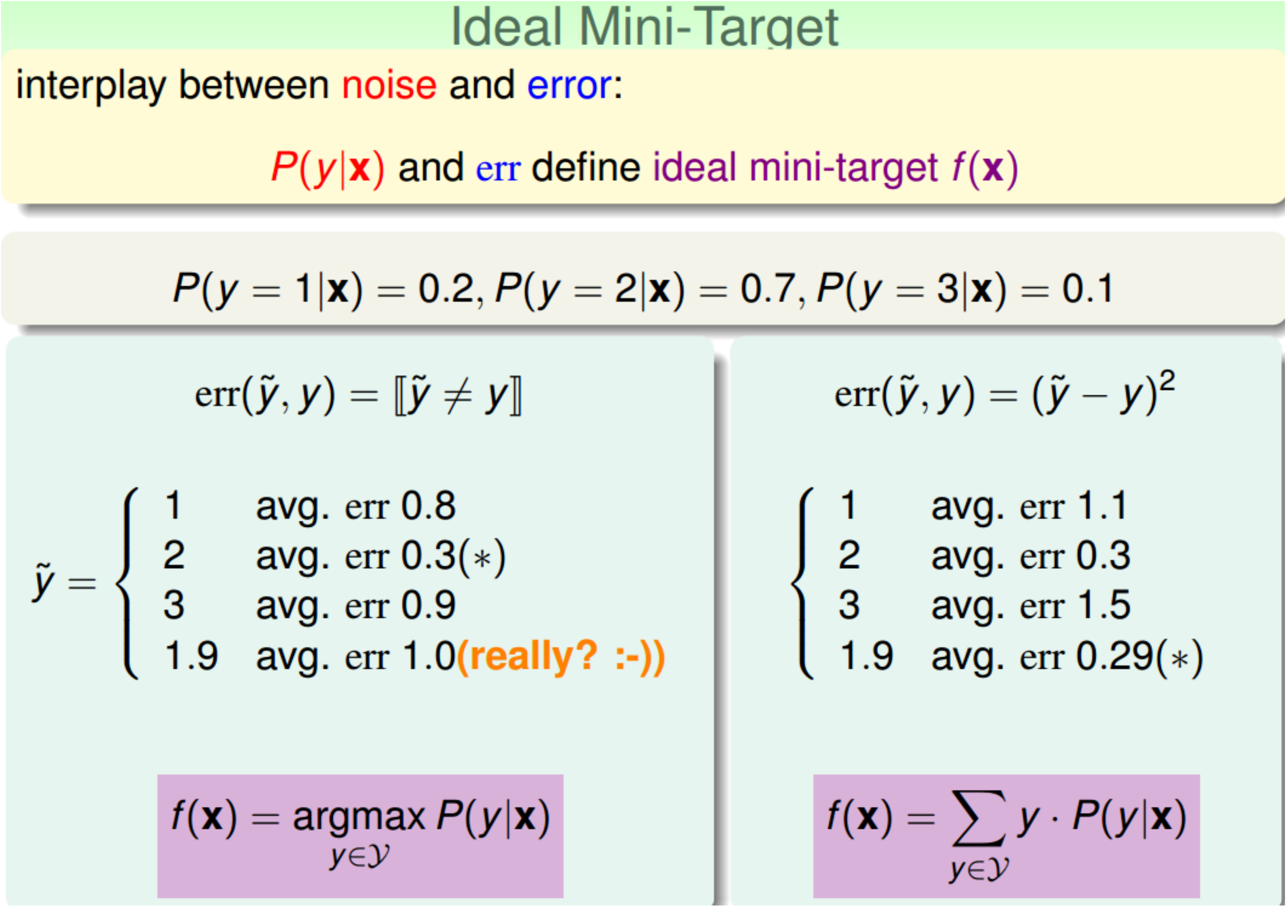
- correct or incorrect?
- often for classification

squared error

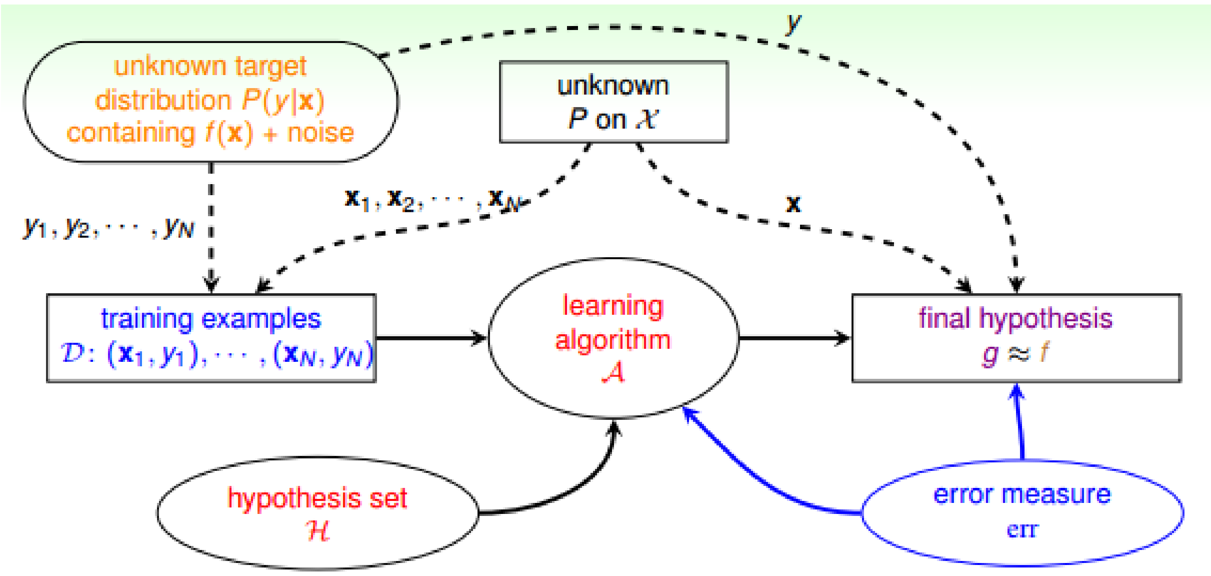
$$\text{err}(\tilde{y}, y) = (\tilde{y} - y)^2$$

- how far is $\tilde{y}$ from y?
- often for regression

Ideal Mini-Target由$P(y|x)$和err共同决定，0/1 error和squared error的Ideal Mini-Target计算方法不一样。例如下面这个例子，分别用0/1 error和squared error来估计最理想的mini-target是多少。0/1 error中的mini-target是取P(y|x)最大的那个类，而squared error中的mini-target是取所有类的加权平方和。



有了错误衡量，就会知道当前的矩g是好还是不好，并会让演算法不断修正，得到更好的矩g，从而使得g与目标函数更接近。所以，引入error measure后，学习流程图如下所示：



三、Algorithmic Error Measure

Error有两种：false accept和false reject。false accept意思是误把负类当成正类，false reject是误把正类当成负类。 根据不同的机器学习问题，false accept和false reject应该有不同的权重，这根实际情况是符合的，比如是超市优惠，那么false reject

应该设的大一些；如果是安保系统，那么false accept应该设的大一些。

two types of error: false accept and false reject

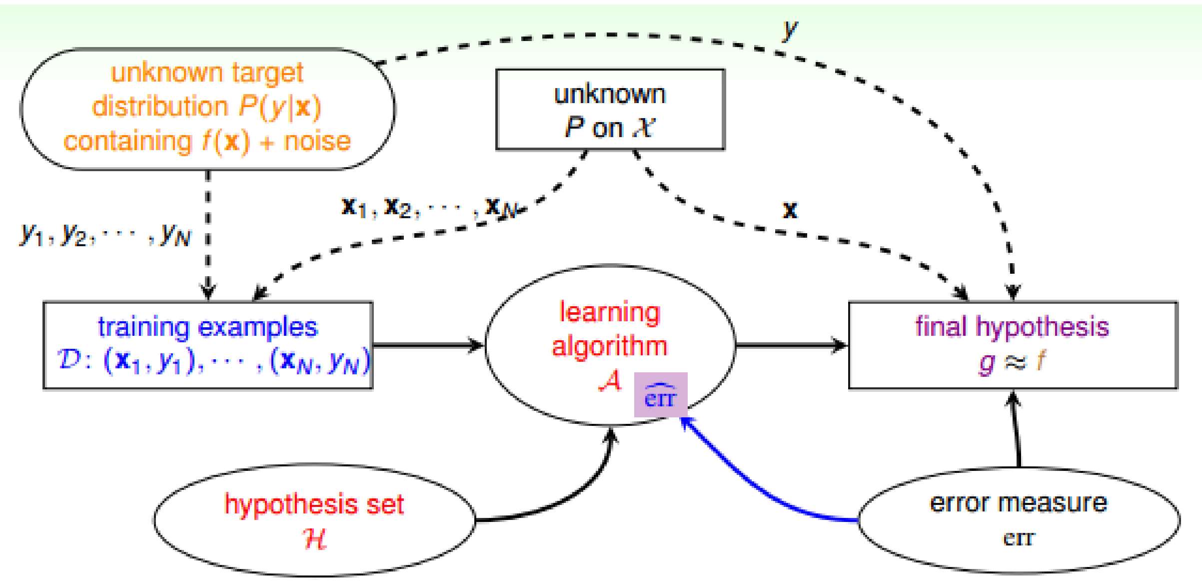
		g: +1	g: -1
f	+1	no error	false reject
	-1	false accept	no error

机器学习演算法A的cost function error估计有多种方法，真实的err一般难以计算，常用的方法可以采用plausible或者friendly，根据具体情况而定。

Algorithmic Error Measures $\widehat{\mathrm{err}}$

- true: just err
- plausible:
  - 0/1: minimum 'flipping noise'—NP-hard to optimize, **remember? :-)**
  - squared: minimum Gaussian noise
- friendly: easy to optimize for $\mathcal{A}$
  - closed-form solution
  - convex objective function

引入algorithm error measure之后，学习流程图如下：



四、Weighted Classification

实际上，机器学习的Cost Function即来自于这些error，也就是算法里面的迭代的目标函数，通过优化使得Error（Ein）不断变小。
cost function中，false accept和false reject赋予不同的权重，在演算法中体现。对不同权重的错误惩罚，可以选用virtual copying的方法。

Systematic Route: Connect E_{in}^{w} and $E_{\text{in}}^{0/1}$

original problem

		$h(\mathbf{x})$ +1	$h(\mathbf{x})$ -1
y	+1	0	1
	-1	1000	0

$$\mathcal{D}: \begin{array}{c} (\mathbf{x}_1, +1) \\ (\mathbf{x}_2, -1) \\ (\mathbf{x}_3, -1) \\ \ldots \\ (\mathbf{x}_{N-1}, +1) \\ (\mathbf{x}_N, +1) \end{array}$$

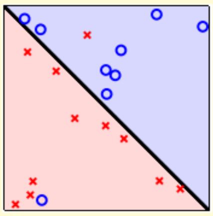
equivalent problem

		$h(\mathbf{x})$ +1	$h(\mathbf{x})$ -1
y	+1	0	1
	-1	1	0

$$\begin{array}{c} (\mathbf{x}_1, +1) \\ (\mathbf{x}_2, -1), (\mathbf{x}_2, -1), \ldots, (\mathbf{x}_2, -1) \\ (\mathbf{x}_3, -1), (\mathbf{x}_3, -1), \ldots, (\mathbf{x}_3, -1) \\ \ldots \\ (\mathbf{x}_{N-1}, +1) \\ (\mathbf{x}_N, +1) \end{array}$$

after **copying -1 examples 1000 times**,
E_{in}^{w} for LHS $\equiv E_{\text{in}}^{0/1}$ for RHS!

Weighted Pocket Algorithm



		$h(\mathbf{x})$ +1	$h(\mathbf{x})$ -1
y	+1	0	1
	-1	1000	0

using 'virtual copying', weighted pocket algorithm include:

- weighted PLA:
  randomly check -1 example mistakes with 1000 times more probability
- weighted pocket replacement:
  if $\mathbf{w}_{t+1}$ reaches smaller E_{in}^{w} than $\hat{\mathbf{w}}$, replace $\hat{\mathbf{w}}$ by $\mathbf{w}_{t+1}$

systematic route (called 'reduction'):
can be applied to many other algorithms!

五、总结

本节课主要讲了在有Noise的情况下，即数据集按照$P(y|x)$概率分布，那么VC

Dimension仍然成立，机器学习算法推导仍然有效。机器学习cost function常用的Error有0/1 error和squared error两类。实际问题中，对false accept和false reject应该选择不同的权重。

注明：

文章中所有的图片均来自台湾大学林轩田《机器学习基石》课程。