

- lecture | ① 什么时候能用机器学习. ① 有 pattern 存在, 有可提高的
② 规则 definition 不能编程实现
只能 machine 自学
- ③ 有 data 做为输入.

performance measure.

(2) 机器学习模型: 通过数据 (data) 建立模式 (pattern)

比如电影推荐, 我们希望推荐的正好用户 rating 高, 我们可以致人的个性和电影的特色. 但人的个
性我们不知道, 我们手动通过它之前的观影的 rating 反推喜好 (建模)
然后再比对, 计算距离.

c

① 我们学到的结果为 hypothesis: g . 真实的模式为 f (我们永远不知道) 而我们学习的目标是让 g 与 f 尽量接近, 所以我们可以用一些 指标 $\text{performance measure}$ 来表示这种接近程度 (准确度等)

② 像决策树、信息熵那样, 虽然各个条件我们可以编程实现, 但我们不知道先用哪个条件, 应用哪个条件, 而这样我们必须从数据中才能知道, 这便是机器学习与编程的区别, 机器学习的判断规则来自数据。

③ 机器学习 从数据出发 学习一个 hypothesis g 使得接近模式 f .

A takes D and H to learn g to approximate f .
机器学习算法 数据 假设集 假设 模式

Lecture 2 hypothesis set

perceptron: 感知机模型: $h(s) = \text{sign}(\sum_i w_i x_i - \text{threshold})$
以下是感知机模型的更新方式，目标是对所有样本分类正确。

For $t = 0, 1, \dots$

- ① find a mistake of w_t called $(x_{n(t)}, y_{n(t)})$

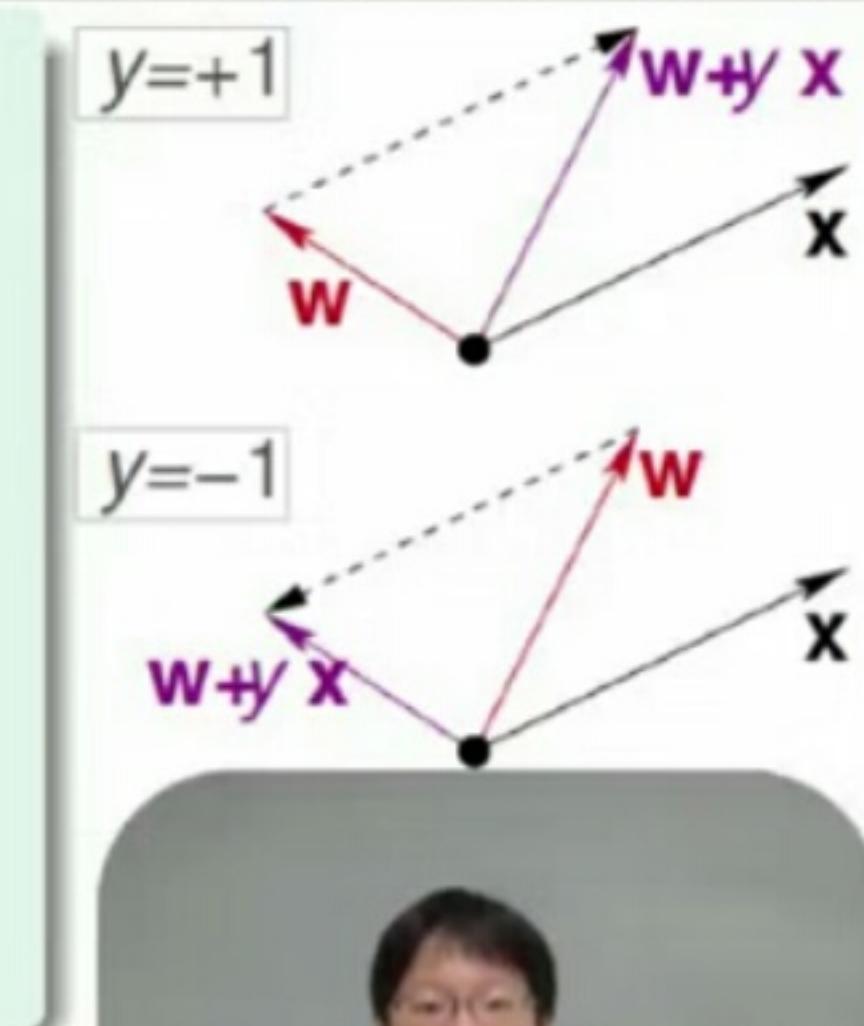
$$\text{sign}(w_t^T x_{n(t)}) \neq y_{n(t)}$$

- ② (try to) correct the mistake by

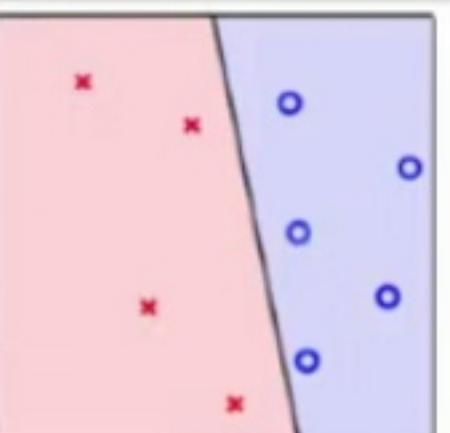
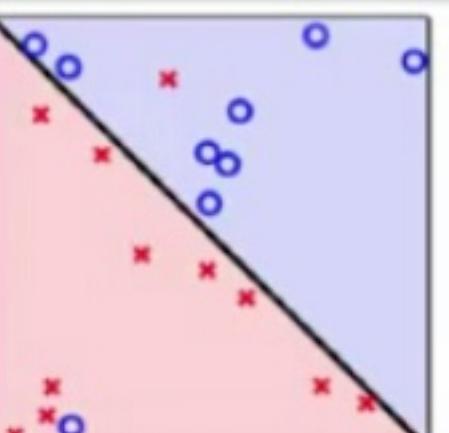
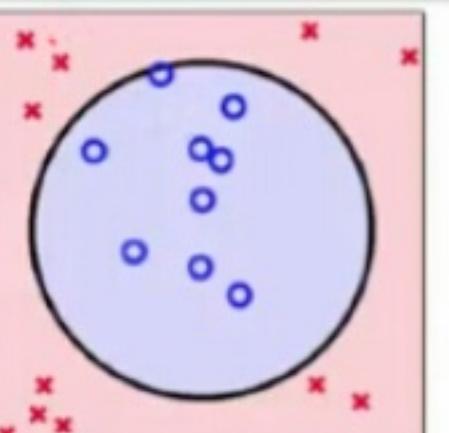
$$w_{t+1} \leftarrow w_t + y_{n(t)} x_{n(t)}$$

... until no more mistakes

return last w (called w_{PLA}) as g



Lecture 2.3

Learning to Answer Yes/No	Guarantee of PLA
<h2>Linear Separability</h2> <p>线性可分时 PLA 才会停下来</p>	
<ul style="list-style-type: none">if PLA halts (i.e. no more mistakes), (necessary condition) \mathcal{D} allows some w to make no mistakecall such \mathcal{D} linear separable	
	(linear separable)
	(not linear separable)
	(not linear separable)
<p>assume linear separable \mathcal{D}, does PLA always halt?</p>	
Learning to Answer Yes/No	modify PLA algor
Hsuan-Tien Lin (NTU CSIE)	initialize pocket

在数据非线性不可分的时候，
或者不知道是否线性可分时，我们可以
可以用 pocket 方法

PLA需要在data线性
可分离才会停下来

Learning to Answer Yes/No

Non-Separable Data

Pocket Algorithm

modify PLA algorithm (black lines) by **keeping best weights in pocket**

initialize pocket weights \hat{w}

For $t = 0, 1, \dots$

- ① find a (random) mistake of w_t called $(x_{n(t)}, y_{n(t)})$
- ② (try to) correct the mistake by

$$w_{t+1} \leftarrow w_t + y_{n(t)} x_{n(t)}$$

- ③ if w_{t+1} makes fewer mistakes than \hat{w} , replace \hat{w} by w_{t+1}

...until enough iterations

return \hat{w} (called w_{POCKET}) as g

有囊體遺傳算法

a simple modification of PLA to find
(somewhat) 'best' weights

Lecture 4, 我们不能准确地进行学习 hypothesis, 但我们可以用一些推论. (概率学论). 我们可以通过采样数据里面的信息推测全局的信息.

Hoeffding 的不等式说明抽样得到的估计与真实情况很接近.

$$P[|V - \mu| > \varepsilon] \leq 2 \exp(-2\varepsilon^2 N)$$

其中 V 是样本中某个参数, μ 是全局中这个参数, N 表示样本中的个数.
我们把这句话 ' $V - \mu$ ' 叫 *probably approximately correct (PAC)*

$$P[|E_{in}(h) - E_{out}| > \varepsilon] \leq 2 \exp(-2\varepsilon^2 N)$$

E_{in} 表示你手中的资料,
 E_{out} 表示全局资料

for fixed h , 我们可以通过我们手上的数据建立的模型 \bar{E}_{in} 估计全局情形: unknown $E_{out}(h) = \mathbb{E}_{x \in P} [P(h(x) \neq f(x))]$

$$\bar{E}_{in}(h) = \mathbb{E}_{x \in P} [P(h(x_n) \neq f(x_n))].$$

\bar{E}_{in} 表示 Data in sample error.

$$P[|\bar{E}_{in}(h) - E_{out}(h)| > \varepsilon] \leq 2e^{-\varepsilon^2 N}$$

if ' $\bar{E}_{in}(h) \approx E_{out}(h)$ ' and ' $\bar{E}_{in}(h)$ small'

$\Rightarrow E_{out}(h)$ small $\Rightarrow h \approx f$ with respect to P .

所以我们可以用这个规则来验证某个 h 是否是好的.

Bad sample: E_{in} and E_{out} far away - can get worse when involving choice
 原因只是: 本来铜板正反概率一样, 但你可以说这是正面的铜板.

PAC 保证 在已有样本学习的模型是可以学习推广的。

$$P_D[\text{Bad } D] = 2M \exp(-2\epsilon^2 N)$$

Bad D 表示这个数据让决策器很容易学到不好的 g .

M 是 hypothesis set 里面的个数, N 是数据里面的个数.

Theory of Generalization Bounding Function: Inductive Cases

Putting It All Together

$$\begin{aligned} B(N, k) &= 2\alpha + \beta \\ \alpha + \beta &\leq B(N-1, k) \\ \alpha &\leq B(N-1, k-1) \\ \Rightarrow B(N, k) &\leq B(N-1, k) + B(N-1, k-1) \end{aligned}$$

		k					
	B(N, k)	1	2	3	4	5	6
N	1	1	2	2	2	2	2
	2	1	3	4	4	4	4
	3	1	4	7	8	8	8
	4	1	≤ 5	11	16	16	16
	5	1	≤ 6	≤ 16	≤ 26	31	32
	6	1	≤ 7	≤ 22	≤ 42	≤ 57	63

growth function 与具体的分界函数类型有关，表示的是这种函数能划分的 set 的最大个数的函数。它有个 break point 表示 growth function 取值等于 2^N 的位置。

Bounding function 表示所有 break point 为上的点，在数据量为 N 时能取的上界（即分 set 数也就是 bounding 不定式里的值）。

为上的点，在数据量为 N 时能取的上界（即分 set 数也就是 bounding 不定式里的值）。

PAC 和 VC 维度从理论上说明了机器是可以学习的。

PAC 说明机器学习里面样本和真实情况差距附合一个不等式

VC 维度从量化上说明这个差距只与模式复杂度有关。

选取 hypothesis 时应 VC 维度在中间的部分。

2D perceptron 的 $VC = d+1$.

如果我们的 hypothesis set 的 VC 维比数据高，数据量比较大

同时演算法能够学到一个合适的 g 那就对了。

The theory of generalization

~~增长函数最大值~~ 多少个 dichotomizes)、~~增长函数~~ $M_H(n)$ 表示 H 在数据集 D 上取 n 个元素能够实现不同取值的最大种类数
break point (绽放开一线曙光的点, $dichotomizes < 2^N$ 的点, 分类器对于这 k 个点产生的合格的 dichotomizes 不能 shattered 掉所有的情况。

所以可以说 break point k 可以把成长函数限制到多项式

bounding function: 描述的是当 break point 为 k 时, 增长函数的最大可能取值函数 (与具体的 hypothesis 无关). $B(N, k)$.

irrelevant of the details of H eg. $B(N, 3)$ 为下面两种情形都确定了上界.

- ① possible interleaving ($k=3$)
- ② 1D perceptron ($k=3$).

↑
不公道的

现在我们希望证明 $B(N, k)$ 小于一个多项式.

Table of Boundary bcs.

$B(N, k)$	1	2	3	4	5	6	...
1	1	2	2	2	2	2	
2	1	3	4	4	4	4	
3	1	4	7	8	8	7	
4	1	$\leq B_{3,1} + B_{3,2}$	$\leq B_{3,2} + B_{3,3}$	15	16	16	
5	1	$\leq B_{4,1} + B_{4,2}$	$\leq B_{4,2} + B_{4,3}$	$\leq B_{4,3} + B_{4,4}$	31	32	
6	1				63		
...							

对于 $k=1$ 来说, 所有的排列只能有一种

$$\begin{aligned}
 & B(N, k) = 2^N + \beta \\
 & (\alpha + \beta) \leq B(N-1, k) \\
 & \alpha \leq B(N-1, k-1) \\
 \Rightarrow B(N, k) & \leq \underbrace{B(N-1, k-1)}_{增长出极上P出数的上限} + B(N-1, k)
 \end{aligned}$$

对于 $N < k$ 来说

$$B(N, k) = 2^N$$

对于 $N=k$ 来说.

$$B(N, k) = 2^N - 1$$

Bounding Function: The Theorem. $B(N, k) \leq \sum_{i=0}^{k-1} \binom{N}{i}$. (最高阶是 $k-1$)
 $\therefore B(N, k) \leq O(n^{k-1})$

VC bounding: $P[|\bar{E}_{in}(h) - \bar{E}_{out}(h)| > \varepsilon] \leq 4m_h(2N) \exp(-\frac{1}{3}\varepsilon^2 N)$

probably & loosely, for $N \geq 2, k \geq 3$

$$m_H(N) \leq B(N, k) = \sum_{i=0}^{k-1} \binom{N}{i} \leq N^{k-1}$$

两个条件可使得 $\bar{E}_{out} \approx \bar{E}_{in}$

- ① $m_H(N)$ breaks at k (good H)
- ② N large enough (good D)

VC dimension.

VC维描述的是 hypothesis 的性质. $d_{VC} = n$ 表示这个 hypothesis 可以最多 shatter 某 n 个点. 在数值上是 $d_{VC} = \min(k) - 1$.

→ 数学描述为它的成长函数 $d_{VC} = \max(N): m_H(N) = 2^N$.

计 $N \geq 2, d_{VC} \geq 2, m_H(N) \leq N^{d_{VC}}$

VC维度的作用是, 如果 d_{VC} 是有限的, 那么 hypothesis 就能保证 $E_{out}(g) \approx E_m(g)$
以下摘抄自博文(《VC维含义的个人理解》)

shatter(分散)的根本概念: 对于一个给定集合 $S = \{x_1, \dots, x_d\}$, 如果一个 H 能够实现集合 S 中所有元素的任意一种方式, 则称 H 能分散 S .

VC维的定义: H 的 VC 维表示能被 H 分散的最大集合的大小. 即只要能存在 d 个元素, H 能使得这 d 个元素任意取值(在 2 分类中为 0 或 1), 则 $d_{VC} \geq d$.

<

问题：数据集线性可分，PLA会停下来。我猜是因为，设 $d_{VC} = d$ ，当把几个数据中的所有 d 个的权向量组合都尝试一遍之后，选一个效果最好的，这便是该分类器能达到的最好效果了，就停下来了。

问题：为什么 VC 维有限 $E_{out} \approx \bar{E}_{in}$ 数字上解释得通。

对于 2D PLA 来说。（这时候 hypothesis 的复杂度 d_{VC} 已确定）

对于线性可分的数据集

\downarrow
PLA（线性感知机）可以收敛

$\left| \begin{array}{l} \text{当进行足够多次尝试后} \\ \text{↓} \end{array} \right.$

$E_{in}(g) = 0$. \rightarrow 对于 2D PLA $\left| \begin{array}{l} E_{out}(g) \approx \bar{E}_{in}(g) \\ \text{来说, } E_{out}(g) \approx 0 \end{array} \right.$

x_n 服从 P , $y_n = f(x_n)$.

\downarrow
 $P[(E_{in}(g) - E_{out}(g)) > \varepsilon] \leq \dots$ by $d_{VC} = 3$

$\left| \begin{array}{l} N \text{ 足够大} \\ \text{↓} \end{array} \right.$

感知机的 VC 维.

1D 感知机: $d_{VC} = 2$; 2D 感知机: $d_{VC} = 3$; d-D 感知机: $d_{VC} = \underline{d+1}$.

因为有 threshold, 所以证明的时候多加了一列。

对于线性感知机 d_{VC} 表示有多少个自由度, 表示这个 hypothesis 的复杂度。

hypothesis 的选取对结果的影响。

我们关注两个点: ① E_{out} 与 E_{in} 足够接近 $P[Bad] \leq 4 \cdot (2N)^{d_{VC}} \exp(\dots)$

② E_{in} 是多么小。要保证有许多选择? 由里有个问题。 d_{VC} 与 E_{in} 有什么关系?

我们如果选择一个 d_{VC} 比较小的 g.

第一个条件附合, 因为 $P[Bad] \leq 4 \cdot (2N)^{d_{VC}}$ 小。

第二个条件不附合 因为 d_{VC} 较小, 模型太简单。

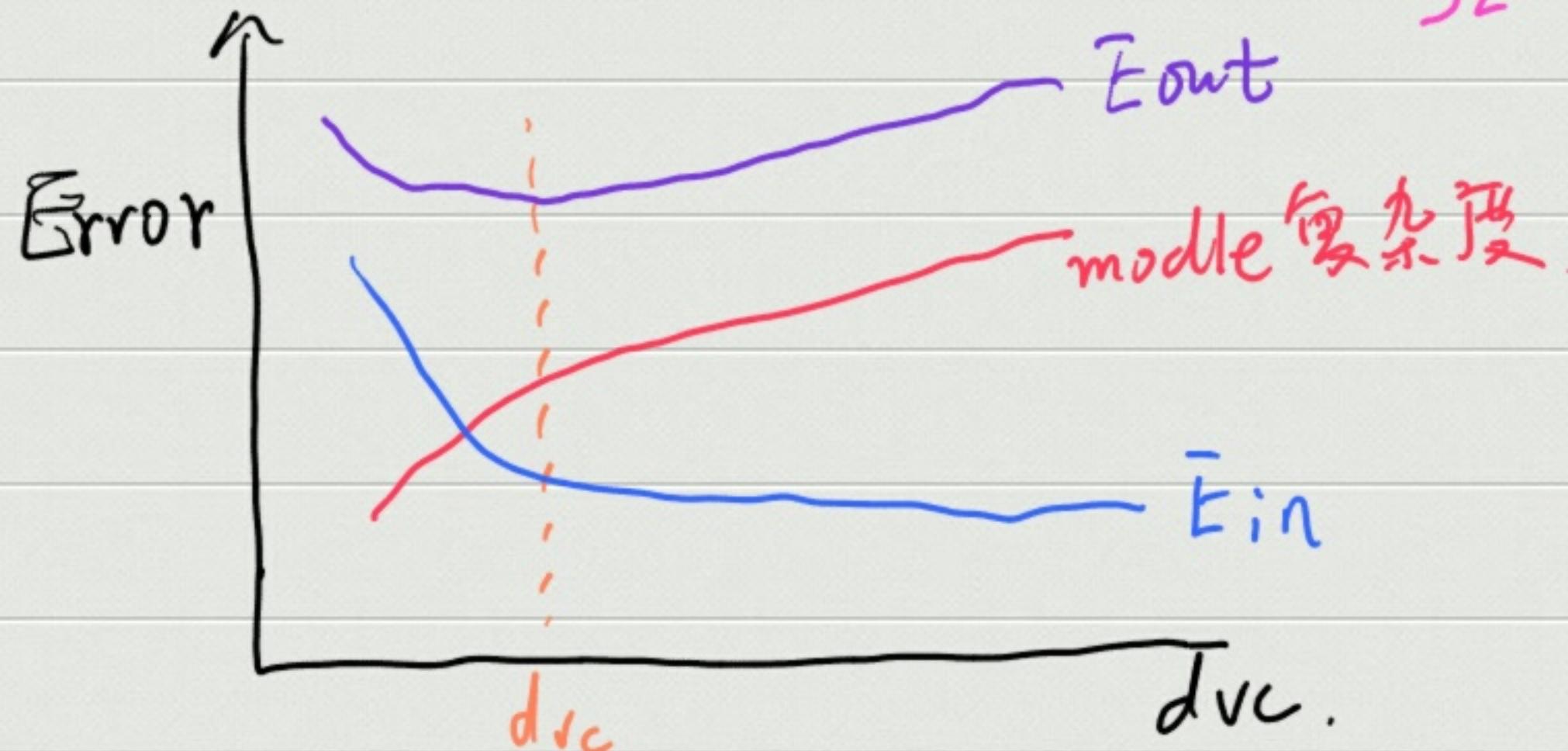
$m(H)$ 表示的就是 hypothesis set 里对于确定的 n, 有多少个不同的 g

而 $M(H)$ 小, 则说明 g 的选择比较少, 不容易选到好 g. 比如用 1D PLA 来对 2 维点分类虽然 $E_{in} \neq 0$.

至于林轩没有说的我也不知道

VC 维传递的信息,

$$E_{\text{out}} \leq E_{\text{in}}(g) + \sqrt{\frac{8}{N} \ln \left(\frac{4(d_{\text{VC}})^{d_{\text{VC}}}}{\delta} \right)} \quad (\text{这个上周我们更关心})$$



随着 d_{VC} 的增大，模型越来越复杂。
由于 g 的选择越来越多，所以 E_{in} 越来
越小，但因为 δ 变大，使得 E_{out} 后面
越来越大。

关于 N 的选取：在理论上 $P[|E_{\text{in}} - E_{\text{out}}| > \epsilon] \leq \delta$ ，要使 ϵ 和 δ 很小需要
的数据量大约是 $N \approx 10000 d_{\text{VC}}$ ，但在实际上 $N \approx 10 d_{\text{VC}}$ 就足够了，
说明这个 VC bound 是很宽松的。

当 d_{VC} 有限的时候 如果数据量够大，就会得到很低的 E_{in} 。

对于有 noise 的数据集

只要保证 $P[x]$ 比较大的样本的取值为 $P[y|x]$ 比较大的情况

Error Measure.

一般有以下考虑的方法：

① out-of-sample 在未知的上取平均.

② pointwise 逐点的评估.

③ classification. 预测 \neq target.

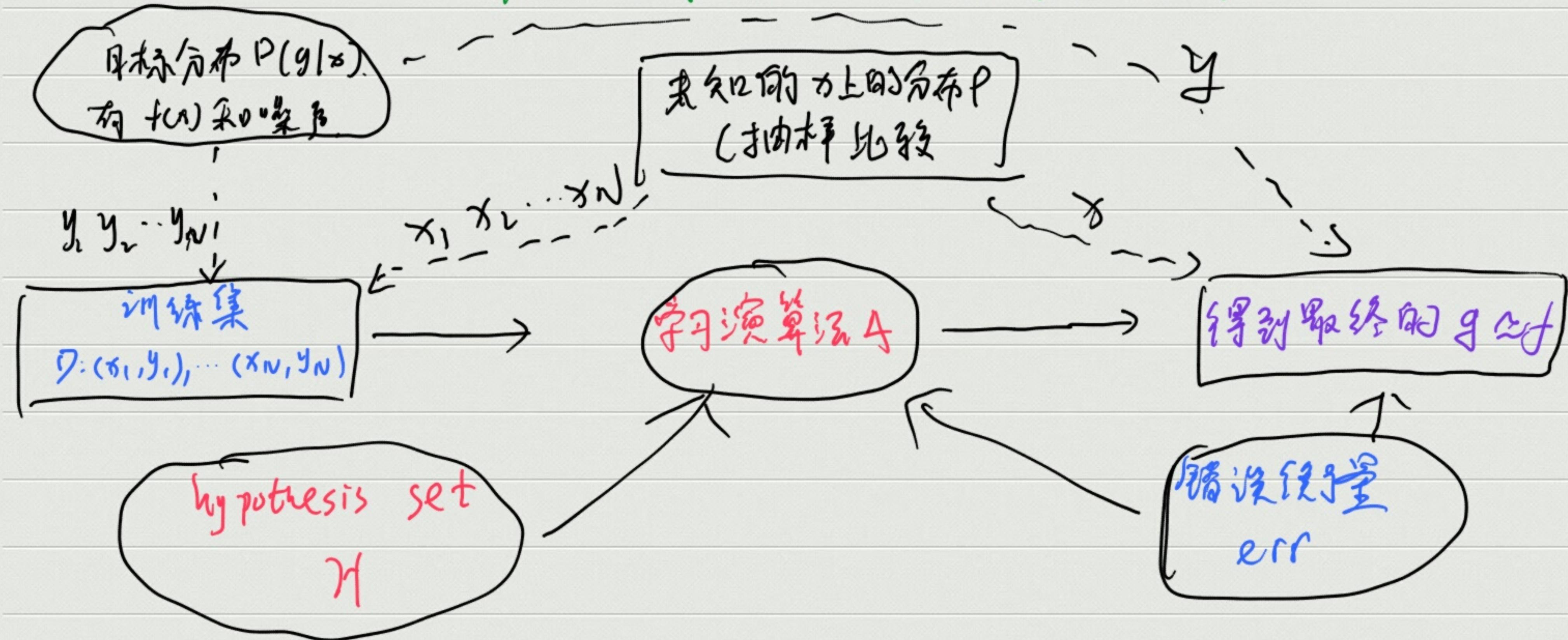
0/1 Error (适合分类) $\text{err}(\tilde{y}, y) = [\tilde{y} \neq y]$

square Error. (适合于回归)
 $\text{err}(\tilde{y}, y) = (\tilde{y} - y)^2$

总结得到：用 0/1 error 选出来的 $f(x)$ 为 $f(x) = \underset{y \in Y}{\operatorname{argmax}} P(y|x)$.

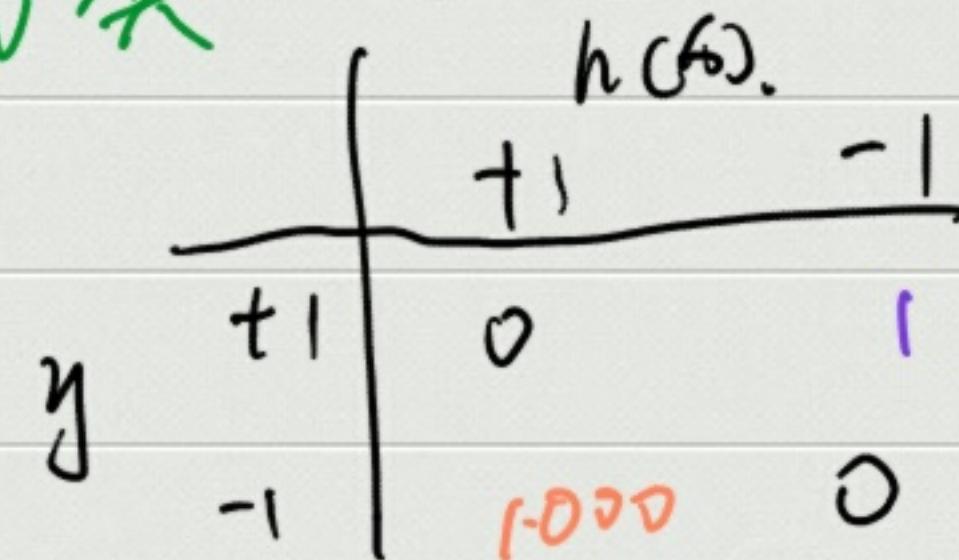
如果用平方 error 采样 $f(x) = \sum_{y \in Y} y \cdot P(y|x)$. (加权平均值)

带错误评估和 Noise 的学习流程



加权分类

C2A 损失 (Error, Loss, ...) 处理.



$$\bar{E}_{out}(h) = \sum_{(x,y) \sim P} \left\{ \begin{array}{l} 1 \\ 0 \\ 0 \end{array} \right\} \cdot \left\{ \begin{array}{l} y = +1 \\ y = -1 \end{array} \right\} \cdot [|y - h(x)|]$$

$$\bar{E}_{in}(h) = \frac{1}{N} \sum_{n=1}^N \left\{ \begin{array}{l} 1 \\ 0 \\ 0 \end{array} \right\} \cdot \left\{ \begin{array}{l} y_n = +1 \\ y_n = -1 \end{array} \right\} \cdot [|y_n - h(x_n)|].$$

(对 $y = -1$ 复制 1000 份, 可保证 pocket 依然可以用 0/1 error)

线性回归

1. g 的形式: $g = w^T x$ 而 PLA 的形式为: $g = \text{sign}(w^T x)$

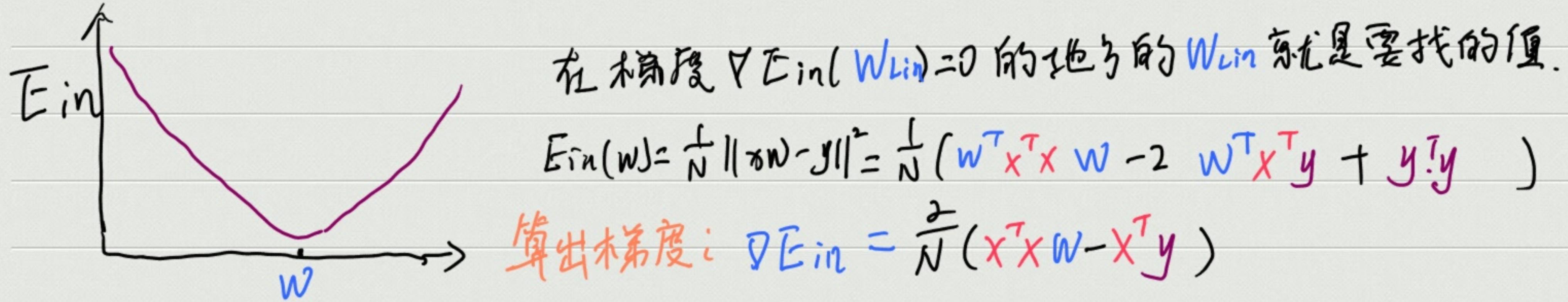
在 error measure 方面: squared error $\text{err}(\hat{y}, y) = (\hat{y} - y)^2$

$$E_{in}(w) = \frac{1}{N} \sum_{n=1}^N \frac{(h(x_n) - y_n)^2}{w^T x_n}. \quad E_{out}(w) = \sum_{(x, y) \in P} (w^T x - y)^2.$$

矩阵形式表示 E_{in} .

$$\begin{aligned} E_{in}(w) &= \frac{1}{N} \sum_{n=1}^N (w^T x_n - y_n)^2 = \frac{1}{N} \sum_{n=1}^N (x_n^T w - y_n)^2 \\ &= \frac{1}{N} \left\| \begin{bmatrix} x_1^T w - y_1 \\ x_2^T w - y_2 \\ \vdots \\ x_N^T w - y_N \end{bmatrix} \right\|^2 \quad \begin{array}{l} \text{(向量的} \\ \text{L}_2 \text{范数)} \end{array} = \frac{1}{N} \left\| \left[\begin{array}{c} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{array} \right] w - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \right\|^2 = \frac{1}{N} \|xw - y\|^2 \end{aligned}$$

$$\min_w \bar{E}_{in}(w) = \frac{1}{N} \|xw - y\|^2 \quad \bar{E}_{in}(w) : \text{连续可微凸函数}$$



$$\text{令 } \nabla \bar{E}_{in}(w) = 0, \quad w = (x^T x)^{-1} \cdot x^T y.$$

证明线性回归可以 work.

考察 \bar{E}_{in} 的平均， $\boxed{\bar{\bar{E}}_{in} = \mathbb{E}_{D \sim P^N} \{ \bar{E}_{in}(w_{lin} \text{ w.r.t. } D) \} = \text{noise level} \cdot \left(1 - \frac{d+1}{N}\right)}$

$$\text{证明: } \bar{E}_{in}(w_{lin}) = \frac{1}{N} \|y - \hat{y}\|^2 = \frac{1}{N} \|\underbrace{(y - x x^T y)}_{x \cdot w_{lin}}\|^2 = \frac{1}{N} \|(I - x x^T) \cdot y\|^2.$$

把 $x x^T$ 看作中心矩阵.

$\mathbb{I} - H$: 表示把 y 转换到 $y - \hat{y}$ 上 span 的层面上.

$\text{trace}(\mathbb{I} - H) = N - (d+1)$. N 个自由度的向量 (y) 投影到 $d+1$ 的空间里,
则 $\mathbb{I} - H$ 的自由度为 $N - (d+1)$.

$$\begin{aligned} ? \quad \overline{E}_{in}(w_{lin}) &= \frac{1}{N} \|y - \hat{y}\|^2 = \frac{1}{N} \|(I - H)\text{noise}\|^2 = \frac{1}{N} (N - (d+1)) \|\text{noise}\|^2 \\ &= \text{noise level} \cdot \left(1 - \frac{d+1}{N}\right), \\ \overline{E}_{out} &= \text{noise level} \cdot \left(1 + \frac{d+1}{N}\right), \quad ? \end{aligned}$$

既然线性回归比线性分类计算方便多了，那可不可以用线性回归替换线性分类
经过计算，我们发现对于线性分类而言 $\text{err}_{\text{of}} \leq \text{err}_{\text{sqr}}$.
所以我们可以用回归的 E_{in} 做分类的 E_{in} 的更宽松的 bound. 这样计算比较
efficient. 我们一般先用线性回归求出一个大概的值，作为 PLA 和 pocket 的初值.