



Machine Learning Foundations

learning and practice



Zhang xuesen

ML

2016

目录

1	When Can machines Learn?	5
1.1	What is Maching Learning	6
1.1.1	Machine learning	6
1.2	Learning to Answer Yes/No	7
1.2.1	Perceptron	7
1.2.2	Pocket	8
1.3	Types of Learning	10
1.4	Feasibility of Learning	12
1.4.1	No Free Lunch	12
1.4.2	Inferring something	12
1.4.3	Connection to Learning	12
1.4.4	Multiple h	13
2	Why Can machines Learn?	14
2.1	Training versus Testing	15
2.1.1	Effective Number lines	15
2.1.2	Effective Number of Hypotheses	15
2.1.3	Break Point	16
2.2	Theory of Generalization	17
2.3	VC demension	18
2.4	Noise and Error	19
2.4.1	Error Measure	19
2.4.2	Weighted Classification	19

目录

3	How Can machines Learn?	20
3.1	Linear Regression	21
3.1.1	Linear Regression hypothesis	21
3.1.2	Linear Regression Algorithm	21
3.1.3	Generalization issue	22
3.1.4	Linear Classification vs. Linear Regression	22
3.2	Logistic Regression	24
3.2.1	Logistic Regression Problem	24
3.2.2	Logistic Hypothesis	24
3.2.3	Three Linear Models	24
3.2.4	Minimizing $E_{in}(W)$	25
3.2.5	Gradient Descent	25
3.3	Linear Models for Classification	27
3.3.1	Linear Models Revisited	27
3.3.2	Stochastic Gradient Decent	28
3.3.3	Multiclass via Binary	28
3.4	Nonlinear Transformation	30
4	How Can machines Learn Better?	31
4.1	Hazard of Overfitting	32
4.1.1	What is Overfitting	32
4.1.2	Dealing with Overfitting	33
4.2	Regularization	34
4.2.1	Regularized Hypothesis Set	34
4.2.2	Weight Decay Regularization	34
4.2.3	Regularization and VC Theory	35
4.2.4	General Regularizers	35
4.3	Validation	37
4.3.1	Model Selection Problem	37
4.3.2	Validation Set	37

4.3.3	Leave-One-Out Cross Validation	37
4.3.4	V-fold Cross validation	37
4.4	Three Learning Principles	39
4.4.1	Occam's Razor	39
4.4.2	Sampling Bias	39
4.4.3	Data Snooping	39
4.4.4	Three Related Fields	40

Chapter 1

When Can machines Learn?

1.1 What is Machine Learning

让机器去学习。有时候规则很难定义，比如怎么定义什么是一颗树。而让机器通过数据学习就会使问题变简单了。可以让机器自动挖掘一些模式。

1.1.1 Machine learning

improving some Performance measure with experience computed from Data.

use **data** to compute hypothesis g that **approximates** target f .

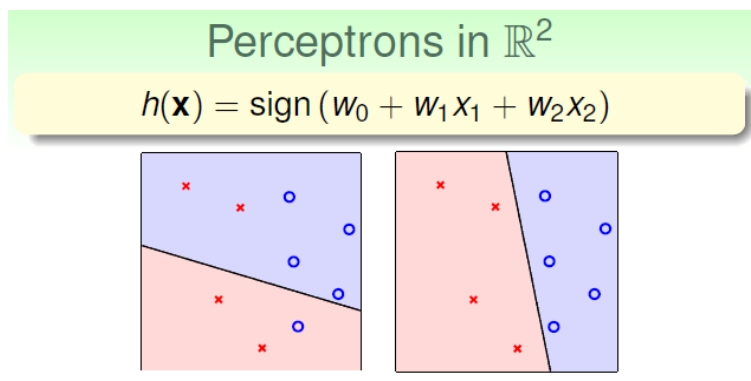
Lecture 1: The Learning Problem

- Course Introduction
foundation oriented and story-like
 - What is Machine Learning
use data to approximate target
 - Applications of Machine Learning
almost everywhere
 - Components of Machine Learning
 A takes \mathcal{D} and \mathcal{H} to get g
 - Machine Learning and Other Fields
related to DM, AI and Stats
-

1.2 Learning to Answer Yes/No

1.2.1 Perceptron

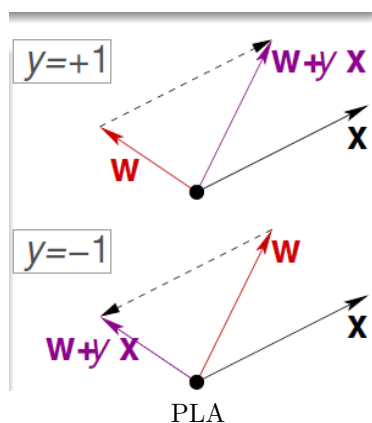
$$\begin{aligned}h(x) &= \text{sign}\left(\sum_{i=1} W_i x_i - \text{threshold}\right) \\&= \text{sign}\left(\sum_{i=0} W_i x_i\right) \\&= \text{sign}(W^T X)\end{aligned}$$



Algorithm 1 Perceptron Learning Algorithm

```
for  $t \leftarrow 1$  to  $m$  do
  Find a mistake of  $W_t$ 
  eg.  $\text{sign}(W_t^T x_{n(t)}) \neq y_{n(t)}$ 
   $W_{t+1} \leftarrow W_t + y_{n(t)} x_{n(t)}$ 
end for
```

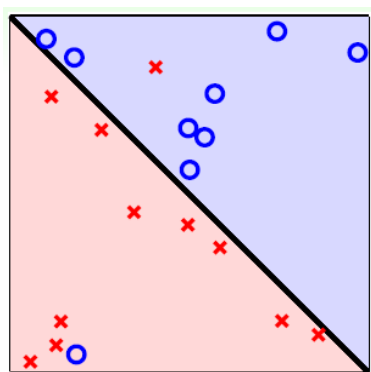
缺点就是必须线性可分 Linear Separability



PLA Fact: W_t Gets more Aligned with W_f , 学习是能够保证 W 逐渐趋近于理想的 W_f , 内积越大越相似。

$$\begin{aligned}w_f^T w_{t+1} &= w_f^T (w_t + y_{n(t)} x_{n(t)}) \\&\geq w_f^T w_t + \min_n y_n w_f^T x_n \\&> w_f^T w_t + 0.\end{aligned}$$

因此整个更新过程是 $w_f \leftarrow w_t$ 的，但有时候数据是有噪音的或者本身就不可分。



1.2.2 Pocket

Algorithm 2 Pocket Algorithm

```
initialize pocket weights  $\hat{w}$ 
for  $t \leftarrow 1$  to  $m$  do
  1. Find a mistake of  $W_t$ 
  eg.  $\text{sign}(W_t^T x_{n(t)}) \neq y_{n(t)}$ 
  2.  $W_{t+1} \leftarrow W_t + y_{n(t)} x_{n(t)}$ 
  3.  $W_{t+1} \leftarrow \arg \min_{\text{mistakes}} (\hat{w}, w_t)$ 
end for
```



Important!

对PLA算法一个简单的修改 (Pocket里永远是最好的)，能处理有少许噪音的数据，但是速度要慢一点因为有比较的过程。



Importance

对PLA算法一个简单的修改 (Pocket里永远是最好的)，能处理有少许噪音的数据，但是速度要慢一点因为有比较的过程。

Lecture 2: Learning to Answer Yes/No

- Perceptron Hypothesis Set
hyperplanes/linear classifiers in \mathbb{R}^d
 - Perceptron Learning Algorithm (PLA)
correct mistakes and improve iteratively
 - Guarantee of PLA
no mistake eventually if linear separable
 - Non-Separable Data
hold somewhat 'best' weights in pocket
-

1.3 Types of Learning

- **Binary classification:**

patient features \Rightarrow sick or not

$$\mathcal{Y} = \{-1, +1\} \quad (1.1)$$

- **Multiclass Classification:**

patient features \Rightarrow which type of cancer

$$\mathcal{Y} = \{1, 2, \dots, K\} \quad (1.2)$$

- **regression:**

patient features \Rightarrow how many days before recovery

$$\mathcal{Y} = \mathbb{R} \quad (1.3)$$

- **Sequence Learning:**

NLP

Learning with Different Output Space \mathcal{Y}

- **binary classification:** $\mathcal{Y} = \{-1, +1\}$
- **multiclass classification:** $\mathcal{Y} = \{1, 2, \dots, K\}$
- **regression:** $\mathcal{Y} = \mathbb{R}$
- **structured learning:** $\mathcal{Y} = \text{structures}$
- ... and a lot more!!

-
- Supervised
 - Unsupervised
 - Semi-supervised
 - Reinforcement

Learning with Different Data Label y_n

- **supervised**: all y_n
- unsupervised: no y_n
- semi-supervised: some y_n
- reinforcement: implicit y_n by goodness(\tilde{y}_n)
- ... and more!!

输入，输出，训练模式，算法类型。

Lecture 3: Types of Learning

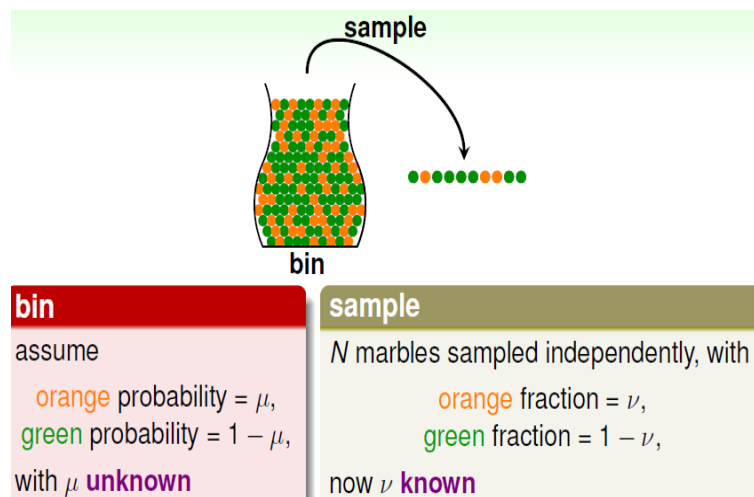
- Learning with Different Output Space \mathcal{Y}
[classification], [regression], structured
 - Learning with Different Data Label y_n
[supervised], un/semi-supervised, reinforcement
 - Learning with Different Protocol $f \Rightarrow (\mathbf{x}_n, y_n)$
[batch], online, active
 - Learning with Different Input Space \mathcal{X}
[concrete], raw, abstract
-

1.4 Feasibility of Learning

1.4.1 No Free Lunch

Learning from D (to infer something outside D) is doomed if any 'unknown' f can happen. :(

1.4.2 Inferring something



Hoeffding's Inequality

坏事情发生的几率有多大

$$\mathbb{P}[|\nu - \mu| > \epsilon] \leq 2 \exp(-2\epsilon^2 N)$$

跟 μ 无关, 跟 N 有关, 如果样本够多, 大概可以去近似。

1.4.3 Connection to Learning

$$E_{out}(h) = E_{x \sim P} [h(x) \neq f(x)]$$
$$E_{in}(h) = \frac{1}{N} \sum_{n=1}^N [h(x) \neq f(x)]$$

E_{in} in-of-sampling (Known)

E_{out} out-of-sampling (UnKnown)

就像刚才一样我们不需要知道 E_{out} , 只需要 N 足够大即可。

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

这样从理论保证, 对于固定的 h , 如果数据足够多的话。

$$E_{in} \approx E_{out}$$

1.4.4 Multiple h

Bound of Bad data

$$\begin{aligned} \mathbb{P}_D[\text{bad } \mathcal{D}] &= \mathbb{P}_D[\text{bad } \mathcal{D} \text{ for all } h] \\ &\leq 2M \exp -2\epsilon^2 N \end{aligned}$$

如果算法 A 找到一个 g 保证 $E_{in} \approx 0$, 那么理论保证 $E_{out} \approx 0$

M 应该代表了复杂度, N 代表了数据多少, 理解这两个数值对ML的优化很有帮助。

Lecture 4: Feasibility of Learning

- Learning is Impossible?
absolutely no free lunch outside \mathcal{D}
- Probability to the Rescue
probably approximately correct outside \mathcal{D}
- Connection to Learning
verification possible if $E_{in}(h)$ small for fixed h
- Connection to Real Learning
learning possible if $|\mathcal{H}|$ finite and $E_{in}(g)$ small

Chapter 2

Why Can machines Learn?

2.1 Training versus Testing

Importance

$$\mathbb{E}_{out}(g) \underbrace{\approx}_{\text{test}} \mathbb{E}_{in}(g) \underbrace{\approx}_{\text{train}} 0$$

2.1.1 Effective Number lines

lines in 2D	
N	effective(N)
1	2
2	4
3	8
4	$14 < 2^N$

- effective(N) can replace M and
- effective(N) $\ll 2^N$

2.1.2 Effective Number of Hypotheses

dichotomies 二分

	hypotheses \mathcal{H}	dichotomies $\mathcal{H}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$
e.g.	all lines in \mathbb{R}^2	$\{0000, 000\text{X}, 00\text{XX}, \dots\}$
size	possibly infinite	upper bounded by 2^N

$$m_{\mathcal{H}}(N) = 2^N \Leftrightarrow$$

there exists N inputs that can be shattered

2.1.3 Break Point

if no K inputs can be shattered by H ,

call k a break point for H .

$$m_{\mathcal{H}}(k) < 2^k \quad (2.1)$$

$$k+1, k+2, \dots \text{also break points} \quad (2.2)$$

$$(2.3)$$

break points 跟 成长函数 是有关的。

Lecture 5: Training versus Testing

- Recap and Preview
 - two questions:** $E_{\text{out}}(g) \approx E_{\text{in}}(g)$, and $E_{\text{in}}(g) \approx 0$
- Effective Number of Lines
 - at most 14 through the eye of 4 inputs**
- Effective Number of Hypotheses
 - at most $m_{\mathcal{H}}(N)$ through the eye of N inputs**
- Break Point
 - when $m_{\mathcal{H}}(N)$ becomes 'non-exponential'**

2.2 Theory of Generalization

Theory of Generalization.

2.3 VC demension

VC demension.

2.4 Noise and Error

Target Distributiion but not Target Function

VC still works.

2.4.1 Error Measure

用来判别 f 是否有效。

0/1 error

$$E(\hat{y}, y) = \mathbb{I}[\hat{y} \neq y]$$

squared error

$$E(\hat{y}, y) = (\hat{y} - y)^2$$

不同的error 'Guide' Learning

2.4.2 Weighted Classification

CIA cost vs. market cost

Copy一些样本就相当于在这些样本上增加了weight。

Lecture 8: Noise and Error

- Noise and Probabilistic Target
can replace $f(x)$ by $P(y|x)$
- Error Measure
affect 'ideal' target
- Algorithmic Error Measure
user-dependent \implies plausible or friendly
- Weighted Classification
easily done by virtual 'example copying'

Chapter 3

How Can machines Learn?

3.1 Linear Regression

3.1.1 Linear Regression hypothesis

$$h(x) = W^T x$$



Importance

Find **lines/hyperplanes** with small **residuals**.

The Error Measure

$$E_{in}(w) = \frac{1}{N} \sum_{n=1}^N (w^T x_n - y_n)^2$$
$$E_{out}(h) = \mathbb{E}_{x \sim P} (w^T x_n - y_n)^2$$

Min E_{in} ?

3.1.2 Linear Regression Algorithm

$$E_{in}(w) = \frac{1}{N} \sum_{n=1}^N (w^T x_n - y_n)^2$$
$$= \frac{1}{N} \|XW - Y\|^2$$

$$\begin{aligned}
& \text{let } \nabla_w E = 0 \\
& E_{in}(W) = \frac{1}{N} \|XW - Y\|^2 \\
& = \frac{1}{N} (W^T \underbrace{X^T X}_A W - W^T \underbrace{X^T y}_b + \underbrace{y^T y}_c) \\
& = \frac{1}{N} (W^T A W - 2W^T b + c) \\
& \nabla_w E = \frac{1}{N} (2AW - 2b) \\
& = \frac{2}{N} (AW - b) \\
& = 0 \\
& W^* = \underbrace{(X^T X)^{-1} X^T y}_{\text{pseudo-inverse}}
\end{aligned}$$

3.1.3 Generalization issue

不像是学习的过程，而是一步登天。

不是特别懂呀。。

3.1.4 Linear Classification vs. Linear Regression

Linear Classification	Linear Regression
$\mathcal{Y} = \{-1, +1\}$ $h(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$ $\text{err}(\hat{y}, y) = \mathbb{I}[\hat{y} \neq y]$	$\mathcal{Y} = \mathbb{R}$ $h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ $\text{err}(\hat{y}, y) = (\hat{y} - y)^2$
NP-hard to solve in general	efficient analytic solution

可以用LR来优化linear Classification。VC

Lecture 9: Linear Regression

- Linear Regression Problem
use hyperplanes to approximate real values
- Linear Regression Algorithm
analytic solution with pseudo-inverse
- Generalization Issue
 $E_{\text{out}} - E_{\text{in}} \approx \frac{2(d+1)}{N}$ on average
- Linear Regression for Binary Classification
0/1 error \leq squared error

3.2 Logistic Regression

3.2.1 Logistic Regression Problem

$$f(x) = \mathcal{P}(+1|x) \in [0, 1]$$

3.2.2 Logistic Hypothesis

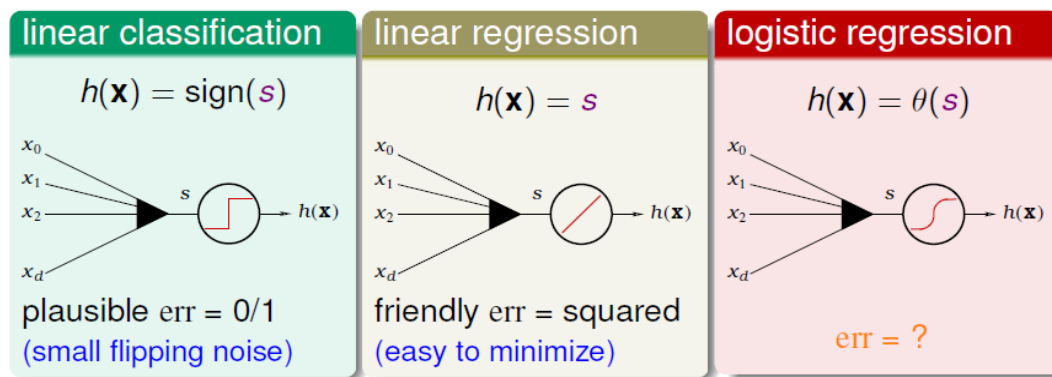
$$\begin{aligned} h(x) &= \theta(W^T X) \\ &= \frac{1}{1 + \exp(-W^T X)} \\ \theta(s) &= \frac{1}{1 + \exp(-s)} \end{aligned}$$

Sigmoid: smooth, monotonic

3.2.3 Three Linear Models

linear scoring function:

$$S = W^T X$$



Likelihood

$$\mathcal{P}(y|x) = \begin{cases} f(x), & \text{for } y = +1 \\ 1 - f(x), & \text{for } y = -1 \end{cases}$$

$$g^* = \arg \max_h \text{likelihood}(h)$$

$$g^* \propto \prod_{n=1} h(y_n x_n)$$

Cross-Entropy

$$\begin{aligned} \text{likelihood}(w) &\propto \max_w \prod \theta(y_n W^T x_n) \\ &\propto \max_w \log \prod \theta(y_n W^T x_n) \\ &\propto \min_w \sum -\log \theta(y_n W^T x_n) \\ &\propto \min_w \sum \log(1 + \exp(-y_n W^T x_n)) \\ &\propto \min_w \sum \text{err}(W, x_n, y_n) \end{aligned}$$

3.2.4 Minimizing $E_{in}(W)$

用梯度链法求梯度就行了。

求完梯度不像LR一样是close-form, 一步登天。

PLA思想iterative Optimization

$$w_{t+1} \leftarrow w_t + \underbrace{\frac{1}{\eta}}_{\eta} \underbrace{\llbracket \text{sign}(w^T x) \neq y_n \rrbracket * y_n x_n}_{\nu}$$

3.2.5 Gradient Descent

E_{in} , 可以用Greedy approach

$$\begin{aligned} W_{t+1} &\leftarrow W_t + \eta V \\ \min_{||v||=1} E_{in}(\underbrace{W_t + \eta V}_{W_{t+1}}) \end{aligned}$$

η 可以使用泰勒展开

$$E_{in}(W_t + \eta V) \approx E_{in}(W_t) + \eta v^T \nabla E_{in}(W_t)$$

要使 $E_{in}(W_{t+1})$ 最小化, 我们只需要保证 V 的方向和 ∇ 的 **方向相反** 并且大小相同即可。(V即是W要走的方向)

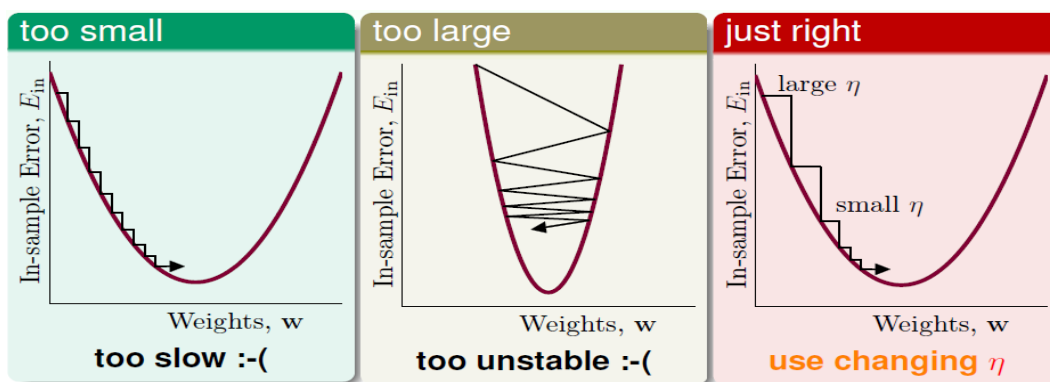
$$V = -\frac{\nabla}{||\nabla||}$$



SGD

$$W_{t+1} \leftarrow W_t - \eta \frac{\nabla}{\|\nabla\|}$$

不同的 η 影响效果也不同，一个启发式的做法是根据梯度的模重新定义 η



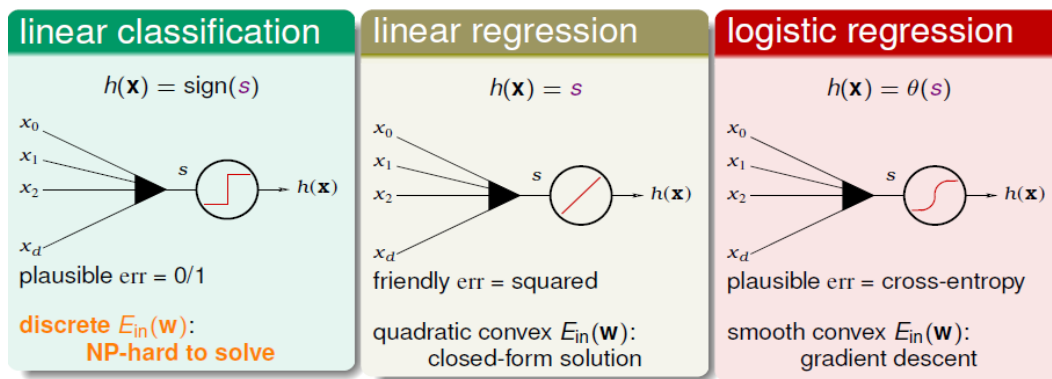
其实就是GD，算法不写了。

Lecture 10: Logistic Regression

- Logistic Regression Problem
 $P(+1|\mathbf{x})$ as target and $\theta(\mathbf{w}^T \mathbf{x})$ as hypotheses
- Logistic Regression Error
cross-entropy (negative log likelihood)
- Gradient of Logistic Regression Error
 θ -weighted sum of data vectors
- Gradient Descent
roll downhill by $-\nabla E_{in}(\mathbf{w})$

3.3 Linear Models for Classification

3.3.1 Linear Models Revisited



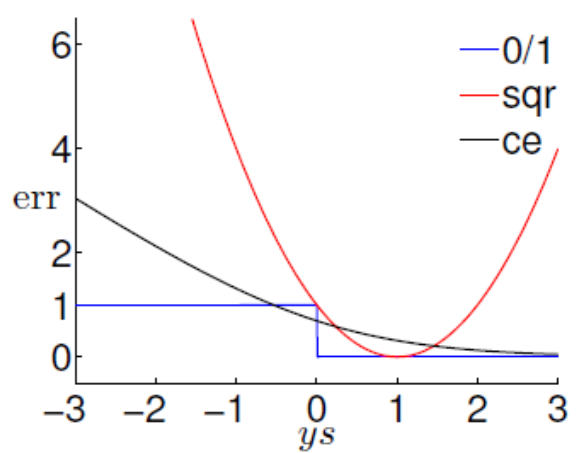
Error Functions Revisited

(ys) 表示了一定的物理意义。

linear classification	linear regression	logistic regression
$h(\mathbf{x}) = \text{sign}(s)$ $\text{err}(h, \mathbf{x}, y) = \llbracket h(\mathbf{x}) \neq y \rrbracket$	$h(\mathbf{x}) = s$ $\text{err}(h, \mathbf{x}, y) = (h(\mathbf{x}) - y)^2$	$h(\mathbf{x}) = \theta(s)$ $\text{err}(h, \mathbf{x}, y) = -\ln h(y\mathbf{x})$
$\text{err}_{0/1}(s, y)$ $= \llbracket \text{sign}(s) \neq y \rrbracket$ $= \llbracket \text{sign}(ys) \neq 1 \rrbracket$	$\text{err}_{\text{SQR}}(s, y)$ $= (s - y)^2$ $= (ys - 1)^2$	$\text{err}_{\text{CE}}(s, y)$ $= \ln(1 + \exp(-ys))$

(ys): classification correctness score

$$\begin{aligned}
 0/1 \quad \text{err}_{0/1}(s, y) &= \llbracket \text{sign}(ys) \neq 1 \rrbracket \\
 \text{sqr} \quad \text{err}_{\text{SQR}}(s, y) &= (ys - 1)^2 \\
 \text{ce} \quad \text{err}_{\text{CE}}(s, y) &= \ln(1 + \exp(-ys)) \\
 \text{scaled ce} \quad \text{err}_{\text{sCE}}(s, y) &= \log_2(1 + \exp(-ys))
 \end{aligned}$$



3.3.2 Stochastic Gradient Decent

$$w_{t+1} \leftarrow w_t + \eta \frac{1}{N} \sum_{n=1} \nabla_i$$

N的数据量实在太大，我们可以使用随机抽样的方式进行。SGD = GD + zero-mean 'noise' step后，平均随机梯度趋近于平均真是梯度。这样做简单，计算量小，适合大数据，online learning，SGD全靠抖，不是特别稳定比较GD而言。

$$w_{t+1} \leftarrow w_t + \eta \nabla_i$$

SGD算法很像PLA

SGD logistic regression:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \eta \cdot \theta \left(-y_n \mathbf{w}_t^T \mathbf{x}_n \right) (y_n \mathbf{x}_n)$$

PLA:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + 1 \cdot \left[y_n \neq \text{sign}(\mathbf{w}_t^T \mathbf{x}_n) \right] (y_n \mathbf{x}_n)$$

3.3.3 Multiclass via Binary

- One vs. All [One class at a Time]
- One vs. One []

Lecture 11: Linear Models for Classification

- Linear Models for Binary Classification
three models useful in different ways
 - Stochastic Gradient Descent
follow negative stochastic gradient
 - Multiclass via Logistic Regression
predict with maximum estimated $P(k|\mathbf{x})$
 - Multiclass via Binary Classification
predict the tournament champion
-

3.4 Nonlinear Transformation

$$x \in \mathcal{X} \xrightarrow{\Phi} z \in \mathcal{Z}$$

perceptrons in Z -space = quadratic hypotheses in X -space

Linear/simpler model first.

Lecture 12: Nonlinear Transformation

- Quadratic Hypotheses

linear hypotheses on quadratic-transformed data

- Nonlinear Transform

happy linear modeling after $\mathcal{Z} = \Phi(\mathcal{X})$

- Price of Nonlinear Transform

computation/storage/[model complexity]

- Structured Hypothesis Sets

linear/simpler model first

Chapter 4

How Can machines Learn Better?

4.1 Hazard of Overfitting

4.1.1 What is Overfitting

Hazard :

Bad Generalization

low E_{in} , high E_{out} .

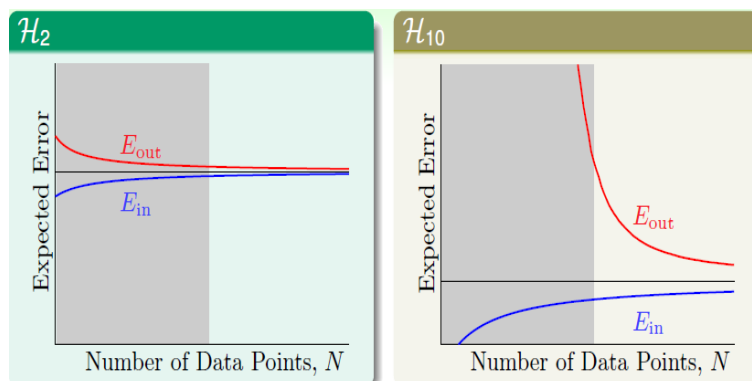


不管模型多复杂 M ，或者数据量多少 N ，其实就以下两个考虑。

$$\mathbb{E}_{out} \approx \mathbb{E}_{in} \approx 0$$

即使 E_{in} 非常小有什么用？作为预测使用我们需要的是 E_{out} 。

下图揭示了模型大小，与数据量大小的关系。



4.1.2 Dealing with Overfitting

Driving Analogy Revisited	
learning	driving
overfit use excessive d_{VC} noise limited data size N	commit a car accident 'drive too fast' bumpy road limited observations about road condition
start from simple model data cleaning/pruning data hinting regularization validation	drive slowly use more accurate road information exploit more road information put the brakes monitor the dashboard

Lecture 13: Hazard of Overfitting

- What is Overfitting?
lower E_{in} but higher E_{out}
 - The Role of Noise and Data Size
overfitting 'easily' happens!
 - Deterministic Noise
what \mathcal{H} cannot capture acts like noise
 - Dealing with Overfitting
data cleaning/pruning/hinting, and more
-

4.2 Regularization

4.2.1 Regularized Hypothesis Set

模型复杂度大，不是容易overfitting么，那么限制一下权值大小（多少）。

$\mathcal{H}_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right.$ $\quad \text{while } w_3 = \dots = w_{10} = 0 \left. \right\}$ regression with \mathcal{H}_2 : $\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w})$ $\text{s.t. } w_3 = \dots = w_{10} = 0$	$\mathcal{H}'_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right.$ $\quad \text{while } \geq 8 \text{ of } w_q = 0 \left. \right\}$ regression with \mathcal{H}'_2 : $\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w})$ $\text{s.t. } \sum_{q=0}^{10} \mathbb{I}[w_q \neq 0] \leq 3$
--	---

直接限制 W 个数，优化是 NP , bad news for sparse hypothesis Set. 因此可以Softer下。

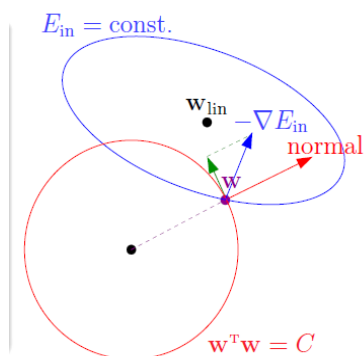
$\mathcal{H}'_2 \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right.$ $\quad \text{while } \geq 8 \text{ of } w_q = 0 \left. \right\}$ regression with \mathcal{H}'_2 : $\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w}) \text{ s.t. } \sum_{q=0}^{10} \mathbb{I}[w_q \neq 0] \leq 3$	$\mathcal{H}(C) \equiv \left\{ \mathbf{w} \in \mathbb{R}^{10+1} \right.$ $\quad \text{while } \ \mathbf{w}\ ^2 \leq C \left. \right\}$ regression with $\mathcal{H}(C)$: $\min_{\mathbf{w} \in \mathbb{R}^{10+1}} E_{\text{in}}(\mathbf{w}) \text{ s.t. } \sum_{q=0}^{10} w_q^2 \leq C$
---	---

4.2.2 Weight Decay Regularization

这样的意思就是把 W 现在在一个 \sqrt{C} 的圆里边。

$$\begin{aligned} \min_{w \in \mathbb{R}^{Q+1}} E_{\text{in}}(w) &= \frac{1}{N} \sum_{n=1} (w^T z - y)^2 \\ \text{s.t. } \underbrace{\sum_{q=0} w_q^2}_{w^T w} &\leq C \end{aligned}$$

负的梯度方向可以垂直分解为圆的切线和法线方向。但是 C 限制了法线方向因此只能沿着圆的切线方向移动。因此 W_{reg} 的最优解是：负的梯度方向和法线方向是平行的。



$$-\nabla E_{in} W_{reg} \propto W_{reg}$$

使用有条件最优化工具Lagrange multiplier，可以模仿线性回归解正规方程了，也叫ridge regression。另外可以根据梯度公式推导出新的优化公式，这也是常见的L2正规化公式。

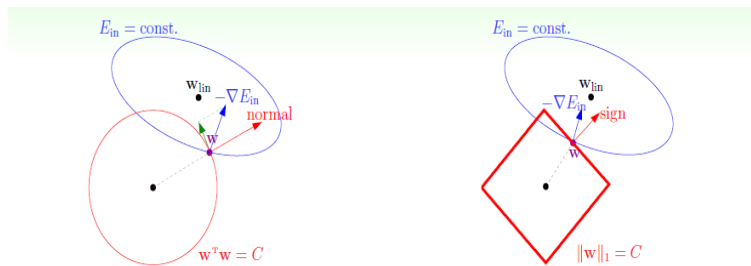
$$\begin{aligned} \nabla E_{in}(W) + \frac{2\lambda}{N} W &= 0 \\ W &\leftarrow (Z^T Z + \lambda I)^{-1} Z^T y \end{aligned}$$

$$\begin{aligned} \text{solve } \nabla E_{in}(W) + \frac{2\lambda}{N} W &= 0 \\ \min_W E_{in}(W) + \frac{\lambda}{N} \overbrace{W^T W}^{\text{regular}} \end{aligned}$$

4.2.3 Regularization and VC Theory

4.2.4 General Regularizers

L2 and L1



L2 Regularizer

$$\Omega(\mathbf{w}) = \sum_{q=0}^Q w_q^2 = \|\mathbf{w}\|_2^2$$

- convex, differentiable everywhere
- easy to optimize

L1 Regularizer

$$\Omega(\mathbf{w}) = \sum_{q=0}^Q |w_q| = \|\mathbf{w}\|_1$$

- convex, **not** differentiable everywhere
- **sparsity** in solution

Lecture 14: Regularization

- Regularized Hypothesis Set
original \mathcal{H} + constraint
- Weight Decay Regularization
add $\frac{\lambda}{N} \mathbf{w}^T \mathbf{w}$ in E_{aug}
- Regularization and VC Theory
regularization decreases d_{EFF}
- General Regularizers
target-dependent, [plausible], or [friendly]

4.3 Validation

4.3.1 Model Selection Problem

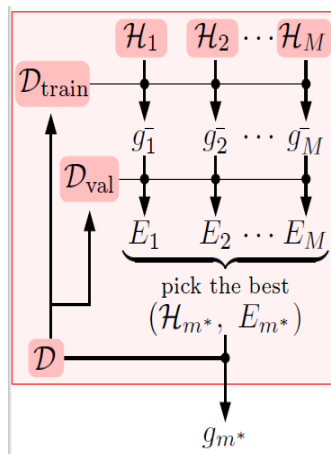
$$E_{out} \approx E_{in} \approx 0$$

selecting by E_{in} is dangerous, 因为模型在训练集训练过了, 会overfitting

Model selection by Best E_{test} split出验证集。

4.3.2 Validation Set

E_{val} 用于选择模型, 桥接 E_{in} 和 E_{out} . 如果能保证 D_{val}, D_{train} 以及 D_{test} 都是iid的来自于同一个分布 P 的话, 效果是有保证的。



4.3.3 Leave-One-Out Cross Validation

4.3.4 V-fold Cross validation

交叉验证。

Lecture 15: Validation

- Model Selection Problem

dangerous by E_{in} and dishonest by E_{test}

- Validation

select with $E_{\text{val}}(\mathcal{D}_{\text{train}})$ while returning $\mathcal{A}_{m^*}(\mathcal{D})$

- Leave-One-Out Cross Validation

huge computation for almost unbiased estimate

- V-Fold Cross Validation

reasonable computation and performance

4.4 Three Learning Principles

4.4.1 Occam's Razor



Simple is Better !!

The simplest model that fits the data is also the most plausible.

4.4.2 Sampling Bias



iid is importance !!

If the data is sampled in biased way, learning will produce a similarly biased outcome.

Match test scenario(distribution) as much as possible.

4.4.3 Data Snooping

偷看数据很严重。

Dealing with Data Snooping

- truth—**very hard to avoid**, unless being extremely honest
 - extremely honest: **lock your test data in safe**
 - less honest: **reserve validation and use cautiously**
-
- be blind: avoid **making modeling decision by data**
 - be suspicious: interpret research results (including your own) by proper **feeling of contamination**

4.4.4 Three Related Fields

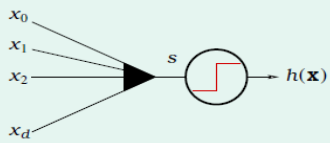
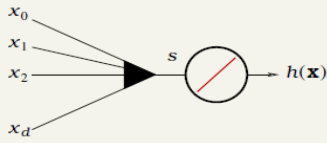
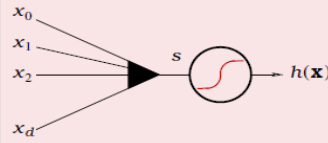
Three Related Fields

Data Mining	Artificial Intelligence	Statistics
<ul style="list-style-type: none"> • use (huge) data to find property that is interesting • difficult to distinguish ML and DM in reality 	<ul style="list-style-type: none"> • compute something that shows intelligent behavior • ML is one possible route to realize AI 	<ul style="list-style-type: none"> • use data to make inference about an unknown process • statistics contains many useful tools for ML

Three Theoretical Bounds

Hoeffding	Multi-Bin Hoeffding	VC
$P[\text{BAD}] \leq 2 \exp(-2\epsilon^2 N)$ <ul style="list-style-type: none"> • one hypothesis • useful for verifying/testing 	$P[\text{BAD}] \leq 2M \exp(-2\epsilon^2 N)$ <ul style="list-style-type: none"> • M hypotheses • useful for validation 	$P[\text{BAD}] \leq 4m_{\mathcal{H}}(2N) \exp(\dots)$ <ul style="list-style-type: none"> • all \mathcal{H} • useful for training

Three Linear Models

PLA/pocket	linear regression	logistic regression
$h(\mathbf{x}) = \text{sign}(s)$  <p>plausible err = 0/1 (small flipping noise) minimize specially</p>	$h(\mathbf{x}) = s$  <p>friendly err = squared (easy to minimize) minimize analytically</p>	$h(\mathbf{x}) = \theta(s)$  <p>plausible err = CE (maximum likelihood) minimize iteratively</p>

Three Key Tools

Feature Transform	Regularization	Validation
$E_{\text{in}}(\mathbf{w}) \rightarrow E_{\text{in}}(\tilde{\mathbf{w}})$ $d_{\text{VC}}(\mathcal{H}) \rightarrow d_{\text{VC}}(\mathcal{H}_{\Phi})$	$E_{\text{in}}(\mathbf{w}) \rightarrow E_{\text{in}}(\mathbf{w}_{\text{REG}})$ $d_{\text{VC}}(\mathcal{H}) \rightarrow d_{\text{EFF}}(\mathcal{H}, \mathcal{A})$	$E_{\text{in}}(h) \rightarrow E_{\text{val}}(h)$ $\mathcal{H} \rightarrow \{g_1^-, \dots, g_M^-\}$
<ul style="list-style-type: none"> by using more complicated Φ lower E_{in} higher d_{VC} 	<ul style="list-style-type: none"> by augmenting regularizer Ω lower d_{EFF} higher E_{in} 	<ul style="list-style-type: none"> by reserving K examples as \mathcal{D}_{val} fewer choices fewer examples

Three Learning Principles

Occam's Razor	Sampling Bias	Data Snooping
simple is good	class matches exam	honesty is best policy

Lecture 15: Validation

Lecture 16: Three Learning Principles

- Occam's Razor
simple, simple, simple!
- Sampling Bias
match test scenario as much as possible
- Data Snooping
any use of data is 'contamination'
- Power of Three
relatives, bounds, models, tools, principles