该文档是极速PDF编辑器生成,
如果想去掉该提示,请访问并下载:
http://www.jisupdfeditor.com/

Lecture9 LearningTheory

Aims

This lecture will introduce you to some foundational results that apply in machine learning irrespective of any particular algorithm, and will enable you to define and reproduce some of the fundamental approaches and results from the computational and statistical theory. Following it you should be able to:

- describe a basic theoretical framework for sample complexity of learning
- describe the Probably Approximately Correct (PAC) learning framework
- describe the Vapnik-Chervonenkis (VC) dimension framework
- describe the Mistake Bounds framework and apply the WINNOW algorithm within this framework
- outline the "No Free Lunch" Theorem

Some questions to ask, without focusing on any particular algorithm:

- Sample complexity
  - How many training examples required for learner to converge (with high probability) to a *successful* hypothesis ?
- Computational complexity
  - How much computational effort required for learner to converge (with high probability) to a successful hypothesis ?
- Hypothesis complexity
  - How do we measure the complexity of a hypothesis ?
  - How large is a hypothesis space ?
- Mistake bounds
  - How many training examples will the learner *misclassify* before converging to a successful hypothesis ?

该文档是极速PDF编辑器生成,
如果想去掉该提示,请访问并下载:
http://www.jisupdfeditor.com/

We start to look at PAC learning using Concept Learning.

Given:

Instances X: Possible days, each described by the attributes
Sky, AirTemp, Humidity, Wind, Water, Forecast

Target function c: $EnjoySport : X \rightarrow \{0, 1\}$

Hypotheses H: Conjunctions of literals. E.g.
$\langle ?, Cold, High, ?, ?, ? \rangle$

Training examples D: Positive and negative examples of target function
$\langle x_1, c(x_1) \rangle, \ldots \langle x_m, c(x_m) \rangle$

Determine:

A hypothesis h in H such that $h(x) = c(x)$ for all x in D?

A hypothesis h in H such that $h(x) = c(x)$ for all x in X?

Given: set of instances X
set of hypotheses H
set of possible target concepts C
training instances generated by a fixed, unknown probability distribution $\mathcal{D}$ over X

Learner observes a sequence D of training examples of form $\langle x, c(x) \rangle$, for some target concept $c \in C$

instances x are drawn from distribution $\mathcal{D}$
teacher provides target value $c(x)$ for each

Learner must output a hypothesis h estimating c

h is evaluated by its performance on subsequent instances drawn according to $\mathcal{D}$

Note: randomly drawn instances, noise-free classifications

该文档是极速PDF编辑器生成,
如果想去掉该提示,请访问并下载:
http://www.jisupdfeditor.com/

Training error of hypothesis h with respect to target concept c

- How often $h(x) \neq c(x)$ over training instances

True error of hypothesis h with respect to c

- How often $h(x) \neq c(x)$ over future random instances

How big is the hypothesis space for *EnjoySport* ?

Instance space

$$\begin{aligned} \textit{Sky} \times \textit{AirTemp} \times \ldots \times \textit{Forecast} &= 3 \times 2 \times 2 \times 2 \times 2 \times 2 \\ &= 96 \end{aligned}$$

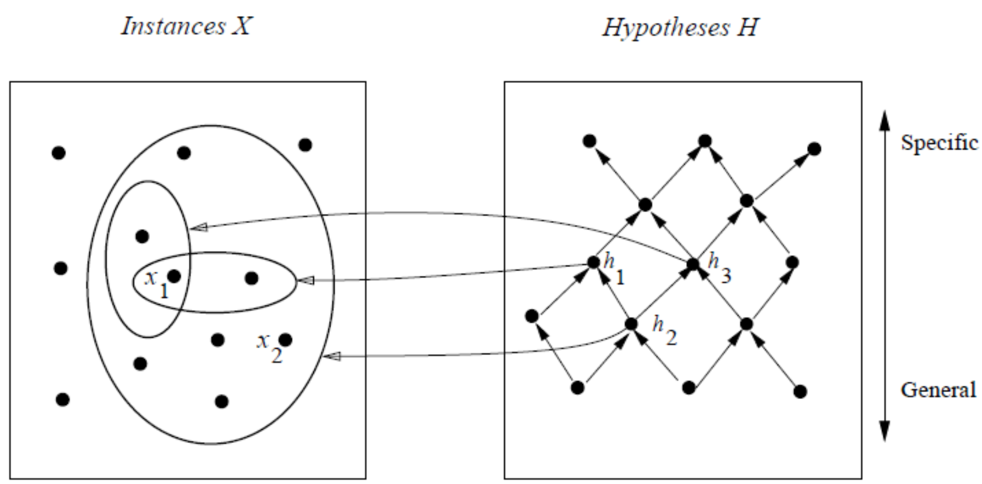
Hypothesis space

$$\begin{aligned} \textit{Sky} \times \textit{AirTemp} \times \ldots \times \textit{Forecast} &= 5 \times 4 \times 4 \times 4 \times 4 \times 4 \\ &= 5120 \\ \text{(semantically distinct}^1 \text{ only)} &= 1 + (4 \times 3 \times 3 \times 3 \times 3 \times 3) \\ &= 973 \end{aligned}$$

The learning problem $\equiv$ searching a hypothesis space. How ?

留学生IT一站式辅导平台 课业·实习·科研·竞赛·求职 DueApe 懂due 懂IT更懂你

该文档是极速PDF编辑器生成,
如果想去掉该提示,请访问并下载:
http://www.jisupdfeditor.com/



x_1 = <*Sunny, Warm, High, Strong, Cool, Same*>
x_2 = <*Sunny, Warm, High, Light, Warm, Same*>

h_1 = <*Sunny, ?, ?, Strong, ?, ?*>
h_2 = <*Sunny, ?, ?, ?, ?, ?*>
h_3 = <*Sunny, ?, ?, ?, Cool, ?*>

Definition: Let h_j and h_k be Boolean-valued functions defined over instances X. Then h_j is **more_general_than_or_equal_to** h_k (written $h_j \geq_g h_k$) if and only if

$$(\forall x \in X)[(h_k(x) = 1) \rightarrow (h_j(x) = 1)]$$

Intuitively, h_j is **more_general_than_or_equal_to** h_k if any instance satisfying h_k also satisfies h_j.

h_j is (strictly) *more_general_than* h_k (written $h_j >_g h_k$) if and only if $(h_j \geq_g h_k) \wedge (h_k \not\geq_g h_j)$.

h_j is *more_specific_than* h_k when h_k is *more_general_than* h_j.

该文档是极速PDF编辑器生成，
如果想去掉该提示，请访问并下载：
http://www.jisupdfeditor.com/

A hypothesis h is **consistent** *with a set of training examples D of target concept c if and only if $h(x) = c(x)$ for each training example $\langle x, c(x) \rangle$ in D.*

$$Consistent(h, D) \equiv (\forall \langle x, c(x) \rangle \in D)\ h(x) = c(x)$$

The **version space***, $VS_{H,D}$, with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with all training examples in D.*

$$VS_{H,D} \equiv \{h \in H | Consistent(h, D)\}$$

Note: in the diagram

(r = training error, $error$ = true error)

Definition: *The version space $VS_{H,D}$ is said to be ϵ-***exhausted** *with respect to c and $\mathcal{D}$, if every hypothesis h in $VS_{H,D}$ has error less than ϵ with respect to c and $\mathcal{D}$.*

$$(\forall h \in VS_{H,D})\ error_{\mathcal{D}}(h) < \epsilon$$

So $VS_{H,D}$ is *not* ϵ-exhausted if it contains at least one h with $error_{\mathcal{D}}(h) \geq \epsilon$.

该文档是极速PDF编辑器生成,
如果想去掉该提示,请访问并下载:
http://www.jisupdfeditor.com/

[Haussler, 1988].

> *If the hypothesis space H is finite, and D is a sequence of $m \geq 1$ independent random examples of some target concept c, then for any $0 \leq \epsilon \leq 1$, the probability that the version space with respect to H and D is not ϵ-exhausted (with respect to c) is less than*
>
> $$|H|e^{-\epsilon m}$$

Interesting! This bounds the probability that any consistent learner will output a hypothesis h with $error(h) \geq \epsilon$

If we want this probability to be below δ

$$|H|e^{-\epsilon m} \leq \delta$$

then

$$m \geq \frac{1}{\epsilon}(\ln |H| + \ln(1/\delta))$$

... if want to assure that with probability 95%, VS contains only hypotheses with $error_{\mathcal{D}}(h) \leq .1$, then it is sufficient to have m examples, where

$$m \geq \frac{1}{0.1}(\ln 973 + \ln(1/0.05))$$

$$m \geq 10(\ln 973 + \ln 20)$$

$$m \geq 10(6.88 + 3.00)$$

$$m \geq 98.8$$

DueApe 懂due 懂IT更懂你

该文档是极速PDF编辑器生成,
如果想去掉该提示,请访问并下载:
http://www.jisupdfeditor.com/

Definition: C *is* **PAC-learnable** *by* L *using* H *if for all* $c \in C$, *distributions* $\mathcal{D}$ *over* X, ϵ *such that* $0 < \epsilon < 1/2$, *and* δ *such that* $0 < \delta < 1/2$, *learner* L *will with probability at least* $(1 - \delta)$ *output a hypothesis* $h \in H$ *such that* $error_{\mathcal{D}}(h) \leq \epsilon$, *in time that is polynomial in* $1/\epsilon$, $1/\delta$, n *and* $size(c)$.

Unbiased concept class C contains all target concepts definable on instance space X.

$$|C| = 2^{|X|}$$

Say X is defined using n Boolean features, then $|X| = 2^n$.

$$|C| = 2^{2^n}$$

An unbiased learner has a hypothesis space able to represent *all* possible target concepts, i.e., $H = C$.

$$m \geq \frac{1}{\epsilon}(2^n \ln 2 + \ln(1/\delta))$$

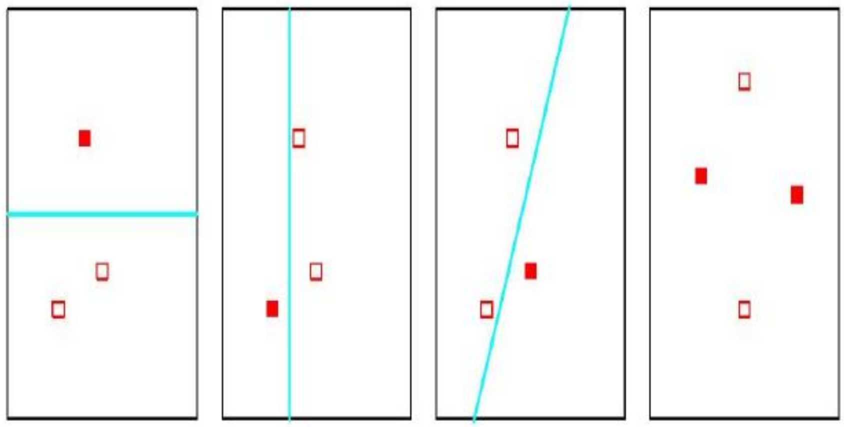
i.e., exponential (in the number of features) sample complexity !

Suppose we have a dataset described by d Boolean features, and a hypothesis space of conjunctions of up to d Boolean literals. Then the largest subset of instances that can be shattered is at least d.

Definition: *The* **Vapnik-Chervonenkis dimension**, $VC(H)$, *of hypothesis space* H *defined over instance space* X *is the size of the largest finite subset of* X *shattered by* H. *If arbitrarily large finite sets of* X *can be shattered by* H, *then* $VC(H) \equiv \infty$.

From the earlier slide on shattering a set of instances by a conjunctive hypothesis, if we have an instance space X where each instance is described by d Boolean features, and a hypothesis space H of conjunctions of up to d Boolean literals, then the VC Dimension $VC(H) = d$.

该文档是极速PDF编辑器生成，
如果想去掉该提示，请访问并下载：
http://www.jisupdfeditor.com/



In general, for linear classifiers in d dimensions the VC dimension is $d + 1$.

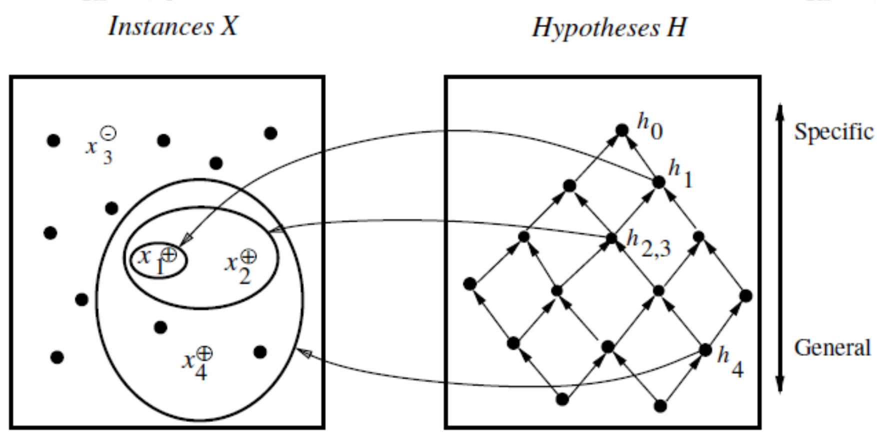
The FIND-S Algorithm

An online, specific-to-general, concept learning algorithm:

- Initialize h to the most specific hypothesis in H
- For each positive training instance x
  - For each attribute constraint a_i in h
    - If the constraint a_i in h is satisfied by x
    - Then do nothing
    - Else replace a_i in h by the next more general constraint satisfied by x

留学生IT一站式辅导平台 课业·实习·科研·竞赛·求职 DueApe 懂due 懂IT 更懂你

该文档是极速PDF编辑器生成，
如果想去掉该提示，请访问并下载：
http://www.jisupdfeditor.com/



h_0 = <∅, ∅, ∅, ∅, ∅, ∅>

x_1 = <*Sunny Warm Normal Strong Warm Same*>, +
x_2 = <*Sunny Warm High Strong Warm Same*>, +
x_3 = <*Rainy Cold High Strong Warm Change*>, -
x_4 = <*Sunny Warm High Strong Cool Change*>, +

h_1 = <*Sunny Warm Normal Strong Warm Same*>
h_2 = <*Sunny Warm ? Strong Warm Same*>
h_3 = <*Sunny Warm ? Strong Warm Same*>
h_4 = <*Sunny Warm ? Strong ? ?* >

- $2n$ terms in initial hypothesis
- first mistake, remove half of these terms, leaving n
- each further mistake, remove at least 1 term
- in worst case, will have to remove all n remaining terms
  - would be most general concept - everything is positive
- worst case number of mistakes would be $n + 1$
- worst case sequence of learning steps, removing only one literal per step

留学生IT一站式辅导平台 课业·实习·科研·竞赛·求职
DueApe 懂due 懂IT更懂你

该文档是极速PDF编辑器生成，
如果想去掉该提示，请访问并下载：
http://www.jisupdfeditor.com/

FIND-S returns the single most-specific consistent hypothesis.

An extension of the FIND-S concept learning algorithm is the CANDIDATE-ELIMINATION algorithm which returns *all* consistent hypotheses, i.e., it finds the Version Space.

Now consider the HALVING ALGORITHM:

- Learns concept using CANDIDATE-ELIMINATION algorithm
- Classifies new instances by majority vote of Version Space hypotheses

How many mistakes will the HALVING ALGORITHM make before converging to correct h?

- ... in worst case?
- ... in best case?

- how many mistakes worst case ?
  - on every step, mistake because majority vote is incorrect
  - each mistake, number of hypotheses reduced by at least half
  - hypothesis space size $|H|$, worst-case mistake bound $\lfloor \log_2 |H| \rfloor$
- how many mistakes best case ?
  - on every step, no mistake because majority vote is correct
  - still remove all incorrect hypotheses, up to half
  - best case, no mistakes in converging to correct h

```
While some instances are misclassified
  For each instance x
    classify x using current weights w
    If predicted class is incorrect
      If x has class 1
        For each x_i = 1, w_i ← αw_i          # Promotion
        (if x_i = 0, leave w_i unchanged)
      Otherwise
        For each x_i = 1, w_i ← w_i/α          # Demotion
        (if x_i = 0, leave w_i unchanged)
```

Here x and w are vectors of features and weights, respectively.

Winnow2

- user-supplied threshold θ
  - class is 1 if $\sum w_i a_i > \theta$
  - typically, the worst-case mistake-bound is something like $\mathcal{O}(r \log n)$

该文档是极速PDF编辑器生成,
如果想去掉该提示,请访问并下载:
http://www.jisupdfeditor.com/

该文档是极速PDF编辑器生成，
如果想去掉该提示，请访问并下载：
http://www.jisupdfeditor.com/

No Free Lunch Theorem

Two main results are:

- Uniformly averaged over all target functions, the expected off-training-set error for all learning algorithms is the same.
- Assuming that the training set $\mathcal{D}$ can be learned correctly by all algorithms, averaged over all target functions no learning algorithm gives an off-training set error superior to any other:

$$\Sigma_F[\mathbb{E}_1(E|F,\mathcal{D}) - \mathbb{E}_2(E|F,\mathcal{D})] = 0$$

where F is the set of possible target functions, E is the off-training set error, and $\mathbb{E}_1$, $\mathbb{E}_2$ are expectations for two learning algorithms.

留学生IT一站式辅导平台 课业·实习·科研·竞赛·求职 DueApe 懂due 懂IT 更懂你