نظریه یادگیری

(*Learning Theory, LT*)

یادگیری احتمالاً تقریباً درست

(*Probably Approximately Correct, PAC*)

(بخش ۲)

دانشگاه فردوسی مشهد
دانشکده مهندسی
رضا منصفی

یادگیری بی‌تعهد و فرضیه‌های ناسازگار

$C \subseteq H$!? (*Agnostic Learning and Inconsistent Hypotheses*)

یادگیر سازگار: **رابطه** $m \geq \frac{1}{\varepsilon}(\ln|H| + \ln(\frac{1}{\delta}))$ با **تعداد نمونه مثال‌های آموزشی** (m) **لازم**، جهت **تضمین یادگیری**

- ✓ با **صحت** ($1 - \varepsilon$) و
- ✓ **اطمینان** ($1 - \delta$)
- ✓ برای هر **فرضیه** در H با مقدار **خطای تجربی صفر** (**سازگار**) بدست آمد.

خطای تجربی صفر: تا به‌حال **فرض** بر این بود که $C \in H$، اگر این **فرض درست نباشد** و داشتن **خطای تجربی صفر** (**فرضیه با خطای صفر**) همیشه **دست‌یافتنی** نیست!

کمینه‌خطای غیرصفر: در این **حالت** بهترین کاری که می‌شود انجام داد این است که از **یادگیر** بخواهیم **فرضیه‌ای** از H ($h \in H$) را **به‌عنوان خروجی** برگرداند که **کمینه خطای غیرصفر** را بر روی **داده‌های آموزشی** داشته باشد.

یادگیر بی‌تعهد: **یادگیری** که **هیچ فرضی مبنی** بر این‌که **مفهوم هدف** ($c \in C$) **قابل نمایش** با H **است** یا **خیر** و فقط **فرضیه‌ای** با **کمینه‌ترین خطای تجربی** را برگرداند، اغلب به آن "**یادگیر بی‌تعهد**" (***Agnostic Learner***) گفته می‌شود.

چرا این نام؟ از **قبل یادگیر هیچ تعهدی** نسبت به این‌که آیا $C \subseteq H$ **است** یا **خیر**، ندارد!

فرضیه خطای صفر: رابطه $m \geq \frac{1}{\varepsilon}(\ln|H| + \ln(\frac{1}{\delta}))$ خروجی یادگیر با فرضیه خطای صفر $error_s(h)=0$ است، می‌توان کران مشابهی را برای حالت عمومی‌تر با خروجی فرضیه با خطای آموزشی غیرصفر $error_s(h)\neq 0$ بدست آورد.

مجموعه داده D: اشاره به مجموعه‌ی خاصی از نمونه‌های آموزشی که در اختیار یادگیر است

در مقابل

مجموعه داده $\check{D}$: توزیع احتمال کل تابع تولیدکننده‌ی داده‌ها (آموزشی و غیرآموزشی) است.

تعریف: خطای کم‌تر-واقعی $(error_D(h))$

در مقابل

خطای واقعی $(error_{\check{D}}(h))$

تعریف: $error_D(h)$ خطای آموزشی از فرضیه h است، به‌طوری‌که این خطا بر روی کسری از مثال‌های آموزشی از D که توسط h اشتباه طبقه‌بندی شده‌اند، تعریف شده است.

توجه مهم: ممکن است خطا بر روی

نمونه‌ی خاصی از داده‌های آموزشی D (خطای کم‌تر-واقعی $error_D(h)$) با

کل نمونه داده‌های آموزشی $\check{D}$ (خطای واقعی $error_{\check{D}}(h)$)

بر روی کل توزیع فرق بکند.

تعریف: h_{best} فرضیه‌ای از H $(h_{best} \in H)$ که کم‌ترین خطا بر روی نمونه‌های آموزشی را دارد.

سؤال: حال برای این‌که مطمئن شویم (با احتمال زیاد) که خطای واقعی $error_{\check{D}}(h_{best})$ بیش‌تر از $\varepsilon + error_D(h_{best})$ نباشد، به چه تعداد نمونه‌های آموزشی نیاز داریم؟

$$error_{\check{D}}(h_{best}) \leq \varepsilon + error_D(h_{best})$$

حالت خاصی: رابطه $m \geq \frac{1}{\varepsilon}(\ln|H| + \ln(\frac{1}{\delta}))$ حالت خاصی از این رابطه است وقتی $error_D(h_{best})$ صفر باشد.

$error_{\check{D}}(h_{best}) \leq \varepsilon +$	0	(خطای واقعی)
$error_{\check{D}}(h_{best}) \leq \varepsilon + error_D(h_{best})$		(خطای کم‌تر-واقعی)

کران‌های هافدینگ عمومی (*General Hoeffding Bounds*)

تعریف: استفاده از **کران‌های *Hoeffding* عمومی** (*General Hoeffding bounds*)،
هم‌چنین بعضاً با نامی دیگر تحت عنوان
کران‌های *Chernoff* افزایشی (*additive Chernoff bounds*)،
انحراف بین **احتمال واقعی رویداد** و
فرکانس مشاهده‌شده آن بر روی ***m*** **آزمایش‌های مستقل**
را **مشخص** می‌نماید.

توجه: به‌طور دقیق‌تر این **کران‌ها** به **آزمون‌هایی** شامل ***m*** **آزمون برنولی متمایز**
(مانند ***m*** **پرتاب مستقل سکه** با **احتمال *p*** **شیرآمدن**) اعمال می‌شود.

توجه: **احتمال شیرآمدن سکه متناظر** با **احتمالی** است که **فرضیه**،
نمونه‌ای تصادفی برگرفته را به **اشتباه طبقه‌بندی** (*misclassification*) نماید.

نکته: ***m*** **پرتاب مستقل سکه مرتبط** با
m **نمونه‌ی مستقل برگرفته‌شده** است.

وجه: **فرکانس شیرآمدن** برروی **نمونه‌های** ***m***
مرتبط با **فرکانس اشتباه طبقه‌بندی‌شدن**
برروی ***m*** **نمونه** است.

تعریف: **کران *Hoeffding*** بیان می‌دارد اگر **خطای آموزشی** (***error*$_D$*(h)***)
بر روی **مجموعه** ***D*** شامل ***m*** **نمونه‌های تصادفی** برگرفته،
اندازه‌گیری شده باشد، **آن‌وقت**

$$\Pr[\,(error_{\hat{D}}(h) > error_D(h) + \varepsilon)] \le e^{-2m\varepsilon^2}$$

This gives us a bound on
the probability that
an arbitrarily chosen single hypothesis
has a very misleading training error.

To assure that
the best hypothesis found
by
L
has an error bounded in this way,
we must consider
the probability
that any one of
the |H| hypotheses
could have
a large error.

توجه: کران **Hoeffding** کرانی بر روی **احتمالی** می‌دهد، که
تک‌فرضیه دلخواهِ انتخاب‌شده،
خطای آموزش
بسیار گمراه‌کننده دارد.

نکته: برای اطمینان از این‌که
بهترین (***best***) **فرضیه یافت‌شده**
توسط
L، به این روش
خطای کران‌دارشده
داشته باشد باید
احتمالی
که هر یک از
فرضیه‌های $|H|$
می‌تواند
خطای بزرگی
داشته باشد، بررسی شود.

$$\Pr[\ (\exists h \in H)(error_{\hat{D}}(h) > error_{D}(h) + \varepsilon)] \leq |\ H\ | e^{-2m\varepsilon^2}$$

توجه: برای **حالت یادگیر بدون تعهد** اگر این **احتمال** را δ بنامیم،

$$\delta \le |H| e^{-2m\varepsilon^2}$$

می‌خواهیم بدانیم چند نمونه آموزشی (m) مورد نیاز است تا δ به **مقدار دلخواه** قرار گیرد.

$$m \ge \frac{1}{2\varepsilon^2}\left(\ln(|H|) + \ln\frac{1}{\delta}\right)$$

توجه: **کران خطای فوق**، **تعمیم‌یافته کران قبلی** است $m \ge \frac{1}{\varepsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$

به‌طوری‌که **یادگیر بازهم بهترین فرضیه** $h \in H$ را **برمی‌گزیند**،

ولی ممکن است **بهترین** فرضیه، **خطای آموزشی غیرصفر** داشته باشد.

نکته: **مقایسه دو رابطه** نشان می‌دهد که **تعداد نمونه‌ها** با **مجذور خطا رابطه معکوس** دارد.

$$m \ge \frac{1}{2\varepsilon^2}\left(\ln(|H|) + \ln\frac{1}{\delta}\right)$$

خطای آموزشی غیرصفر

V.s.

$$m \ge \frac{1}{\varepsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

خطای آموزشی صفر

صحت: گران — **اطمینان: ارزان**

$$m \ge \frac{1}{2\varepsilon}\left[\frac{1}{\varepsilon}\left(\ln(|H|) + \ln\frac{1}{\delta}\right)\right]$$

تعداد نمونه‌های متعهد (خطای آموزشی صفر) $\ge \frac{1}{2\varepsilon}$ **تعداد نمونه‌های بی‌تعهد (خطای آموزشی غیرصفر)**

سؤال: چطور دریابیم *'C' PAC-learnable* است؟

Answer: ***Yes/No!***.

Requires two tasks:-

1) *To show that any* ***consistent learner*** *will require only a* ***polynomial number*** *of* ***training examples*** *to* ***learn*** *any* ***c*** *in* ***C***,

and then

2) *To suggest a* ***specific algorithm*** *that uses* ***polynomial time*** *per* ***training example***.

لیترال‌های بولی عطفی PAC–یادگیر است

(*Conjunction of Boolean Literals are PAC-Learnable*)

چند مثال: کرانی جهت یادگیری احتمالاً تقریباً درست مفهوم هدف

برای تعداد نمونه‌های آموزشی کافی

به‌دست آمد، می‌توان

۱) پیچیدگی نمونه (*sample complexity*) و

۲) یادگیری احتمالاً تقریباً درست (*PAC-learnability*)

را برای

چند کلاس‌های مفهوم خاص

مشخص نمود:-

۱) لیترال‌های بولی عطفی (*Conjunction of Boolean Literals*)

✓ الگوریتم *FindS*

۲) یادگیرهای بدون بایاس (*Unbiased Learners*)

۳) مفهوم‌های *k-Term DNF*

۴) مفهوم‌های *k-CNF*

۵) پیچیدگی نمونه برای فضاهای فرضیه نامتناهی

(*Sample Complexity for Infinite Hypothesis Spaces*)

(۱ لیترال‌های بولی عطفی (*Conjunction of Boolean Literals*

Scenario: *Consider any*

consistent learner L

using a

hypothesis space H identical to C.

How: *Can use*

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

to compute the

number ***m*** *of* ***random training examples sufficient***

to ***ensure*** *that* ***L*** *will,*

with ***probability*** ***(1 - δ),***

output *a* ***hypothesis***

with ***maximum error ε.***

Accomplish: *Need only to determine the*

size |H| *of the* ***hypothesis space.***

Hypothesis space H: *Defined by* ***conjunctions*** *of* ***literals based*** *on*

n Boolean variables.

Size: *The* ***size |H|*** *of this* ***hypothesis space*** *is*

3^n.

Sample Complexity of Conjunction Learning

Scenario: A class **C** of target concepts described over **n** Boolean features by conjunctions of Boolean literals.

n=1, $|H| = 3^1 = 3$

old
¬old
-

n=2 $|H| = 3^2 = 9$

old	tall
old	¬tall
old	-
¬old	tall
¬old	¬tall
¬old	-
-	tall
-	¬tall
-	-

Note: Hypothesis Space **H** defined by Conjunctions of Literals based on **n** Boolean variables.

Definition: A Boolean literal is any

- Boolean variable (e.g., Old), or its
- Negation (e.g., -Old).

Hint: Conjunctions of Boolean literals include target concepts such as "Old ∧ ¬Tall".

A fact: There are only **3** **possibilities** for each **variable** in any given **hypothesis**:

Note:

1) The **variable** as a **literal** in the **hypothesis** (appear **positively**),

2) Its **negation** as a **literal** in the **hypothesis** (appear **negatively**), or

3) **Ignore** it (not **appear** in a given **conjunction**).

A **sufficient** number of **examples** to **learn** a **PAC concept**

Substituting $|H| = 3^n$ into

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

Note:

Gives **bound** for the **sample complexity** of **learning conjunctions** of up to **n Boolean** literals.

$$m \geq \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + \ln 3^n\right) = \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + n\ln 3\right)$$

مجموعه توانی (Power Set)

If **S** is the set {**x**, **y**, **z**}, then the subsets of **S** are:

{} the **empty set**)

{**x**} , {**y**} , {**z**}

{**x**, **y**} , {**x**, **z**}

{**y**, **z**} , {**x**, **y**, **z**}

Properties:

If **S** is a finite set with **|S| = n** elements, then the number of subsets of **S** is $|\mathcal{P}(S)| = 2^n$

Example: *if a* ***consistent learner*** *attempts to learn a* ***target concept*** *described by* ***conjunctions*** *of up to* **$n = 10$** ***Boolean literals***, *with a* **$\delta = 0.05$** **$(1 - \delta)\% = 95\%$** ***probability*** *that it will learn a* ***hypothesis*** *with* ***error*** *less than* **$\varepsilon = 0.1$**, ***confidence*** **$= 1 - \varepsilon = 0.9$** *then it* ***suffices*** *to present* ***m randomly*** *drawn* ***training examples***, *where*

$$m \geq \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + n\ln 3\right) = \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + \ln 3^{n}\right)$$

$$m \geq \frac{1}{\varepsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

$m = (1 \div 0.1) * (\ln(1 \div 0.05) + (10 * \ln(3))) =$ **$\underline{140}$** ***samples***

تحلیل مثال قبل

Note: ***m grows***

1) ***linearly*** *in the* ***number*** *of* ***literals*** ***n***,
2) ***linearly*** *in* **1/ε**, *and*
3) ***logarithmically*** *in* **1/δ**.

Question: *What about the* ***overall computational effort***?

Answer: *Of course, depends*
on the ***specific learning algorithm***.
However,

as long as our ***learning algorithm***
requires no more than
1) ***polynomial computation***
per ***training example***,

and

no more than a ***polynomial number*** *of*
2) ***training examples***,

then

the ***total computation***
required will be
polynomial as well.

$$m \geq \frac{1}{\varepsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$$

$$m \geq \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + \ln 3^n\right) = \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + n\ln 3\right)$$

Find-S
مثال: لیترال بولی عطفی

Note: *In the case of* ***learning conjunctions*** *of* ***Boolean literals****, one* ***algorithm*** *that meets* ***Conjunction Boolean Literals*** *has already been presented as* ***FIND-S****.*

Note: *It is the* ***FIND-S algorithm****, which* ***incrementally computes*** *the* ***most specific hypothesis consistent*** *with the* ***training examples****.*

Note: *For each new* ***positive training example****, this* ***algorithm computes*** *the* ***intersection*** *of the* ***literals shared*** *by the* ***current hypothesis*** *and the* ***new training example****, using* ***time linear*** *in* ***n****.*

Note: *Therefore, the* ***FIND-S algorithm PAC-learns*** *the* ***concept class*** *of* ***conjunctions*** *of* ***n Boolean literals*** *with* ***negations****.*

Theorem: *PAC-learnability of Boolean conjunctions.*

The class C of conjunctions of Boolean literals is PAC-learnable by the FIND-S algorithm using H = C.

Proof: *Equation*

$$m \geq \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + \ln 3^n\right) = \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + n\ln 3\right)$$

shows that the sample complexity for this concept class is polynomial in n, 1/δ, and 1/ε, and independent of size(c).

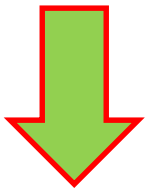
To incrementally process each training example, the FIND-S algorithm requires effort linear in n and independent of 1/δ, 1/ε, and size(c).

Therefore, this concept class is PAC-learnable by the FIND-S algorithm.

Result holds for any consistent learner, including Find-S.

Concrete examples:

δ = ε = 0.05,	**n=10**	*gives*	**280**	*examples*
δ = 0.01, ε = 0.05,	**n=10**	*gives*	**312**	*examples*
δ = ε = 0.01,	**n=10**	*gives*	**1,560**	*examples*
δ = ε = 0.01,	**n=50**	*gives*	**5,954**	*examples*



۲) یادگیرهای بدون بایاس (*Unbiased learners*)

Unbiased Learners: *Not all* ***concept classes*** *have* ***polynomially bounded sample complexity*** *according to the* ***bound*** *of* $m \geq \frac{1}{\varepsilon}(\ln |H| + \ln (\frac{1}{\delta}))$

Example: *Consider the* ***unbiased concept class C*** *that contains every* ***teachable concept*** *relative to* ***X***.

The set ***C*** *of all definable* ***target concepts*** *corresponds to the* ***power set*** *of* ***X***, *the* ***set*** *of all* ***subsets*** *of* ***X,*** *which contains* $|C| = 2^{|X|}$ ***concepts***.

Suppose that instances in ***X*** *are defined by* ***n Boolean features***.

In this case, there will be $|X| = 2^n$ ***distinct instances***, *and therefore* $|C| = 2^{|X|} = 2^{2\wedge n}$ ***distinct concepts***.

Of course to learn such an ***unbiased concept class***, *the* ***learner*** *must itself use an* ***unbiased hypothesis*** *space* $H = C$.

Substituting $|H| = 2^{2\wedge n}$ *into* $m \geq \frac{1}{\varepsilon}(\ln |H| + \ln (\frac{1}{\delta}))$ *gives the sample* ***complexity*** *for* ***learning*** *the* ***unbiased concept class relative*** *to* ***X***. $m \geq 1/\varepsilon(2^n \ln(2) + \ln((1/\delta))$

Note: *Consider any* ***Boolean function*** *over* ***n Boolean features*** *such as the* ***hypothesis*** *space of* ***DNF*** *or* ***decision trees***.

Note: *There are* 2^{2^n} *of these, so a* ***number*** *of* ***examples*** *to learn a* ***PAC concept*** *is:*

$$m \geq \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + \ln 2^{2^n}\right) = \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + 2^n \ln 2\right)$$

Concrete examples:

$\delta = \varepsilon = 0.05,$	$n = 2$	*gives*	**115**	*examples*
$\delta = \varepsilon = 0.05,$	$n = 10$	*gives*	**14,256**	*examples*
$\delta = \varepsilon = 0.05,$	$n = 20$	*gives*	**14,536,410**	*examples*
$\delta = \varepsilon = 0.05,$	$n = 50$	*gives*	1.561×10^{16}	*examples*

Non-Polynomial

K-Term DNF (۳

A fact: *It is also possible to find*

concept classes

that have

polynomial sample complexity.

Surprise: *But* ***nevertheless***

cannot be learned

in ***polynomial time*****!!!!**

Example: *One interesting* ***example*** *is the*

concept class C *of*

k-term Disjunctive Normal Form (k-term DNF) expressions.

Form: ***k-term DNF expressions***

are of the ***form***

$$T_1 \vee T_2 \vee \ldots \vee T_k,$$

where each ***term*** T_i

is a ***conjunction*** *of*

n Boolean attributes

and their ***negations*** $T_i = (C_1 \wedge C_2 \wedge \ldots \wedge C_n)$

How: Assuming $H = C$,

it is easy to **show** that $|H|$ is at **most** 3^{nk}

(because there are **k terms**, each of which

may take on 3^n **possible values**).

Note: 3^{nk} is an **overestimate** of H,

because it is **double counting**

the **cases** where $T_i = T_j$

and where T_i is

more-general-than T_j

Upper Bound: Can use this **upper bound** on $|H|$

to obtain an **upper bound**

on the **sample complexity**,

Substituting this into $m \geq \frac{1}{\varepsilon}(\ln |H| + \ln (\frac{1}{\delta}))$

gives $$m \geq \frac{1}{\varepsilon}\left(nk \ln 3 + \ln \frac{1}{\delta} \right)$$

which indicates that the

sample complexity of

k-term DNF is **polynomial** in

1/δ , **1/ε** , **n** , and **k**.

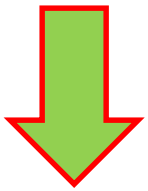
Attention: *Despite having*
polynomial sample complexity,
the computational complexity
is not polynomial,

Why: *Because this* ***learning problem*** *can be shown to be equivalent to other* ***problems*** *that are known to be* ***unsolvable*** *in* ***polynomial time*** *(****unless RP = NP****).*

Hint: *Thus,* ***although k-term DNF*** *has*
polynomial sample complexity,
it does not have
polynomial computational complexity
for a learner using H = C.

A Fact: *The* ***surprising fact*** *about* ***k-term DNF***
is that ***although*** *it is not* ***PAC-learnable***,
there is a ***strictly larger concept class*** *that is!*

Note: *This is* ***possible because*** *the* ***larger concept class*** *has* ***polynomial computation complexity***
per ***example*** *and* ***still*** *has* ***polynomial sample complexity***.



K-CNF Concepts (۴

Note: This **larger class** is the **class** of **k-CNF expressions**:

Conjunctions of **arbitrary length** of the form

$T_1 \wedge T_2 \wedge . \; . \; . \wedge T_j$, . ,

where each is a **disjunction** of up to

k Boolean attributes.

Note: It is straightforward to show that

k-CNF

subsumes

k-DNF,

because any

k-term DNF

expression can easily be rewritten as a

k-CNF expression (but not vice versa).

Note: Although

k-CNF is more expressive than

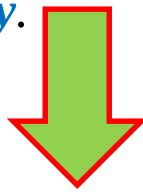
k-term DNF,

it has both **polynomial sample complexity** and **polynomial** time **complexity**.

Note: Hence, the **concept class k-term DNF** is

PAC learnable by an **efficient algorithm** using

H = k-CNF.



See Kearns and **Vazirani (**1994**) for a more detailed discussion.

Other Concept Classes

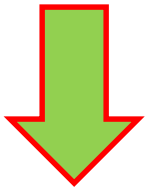
- ***k-term DNF:*** — ***Disjunctions*** *of at most* ***k unbounded conjunctive*** *terms:*
  - $\ln(|H|)=O(kn)$
  - $T_1 \vee T_2 \vee \cdots \vee T_k$

- ***k-DNF:*** — ***Disjunctions*** *of any number of terms each limited to at most k* ***literals****:*
  - $\ln(|H|)=O(n^k)$
  - $((L_1 \wedge L_2 \wedge \cdots \wedge L_k) \vee (M_1 \wedge M_2 \wedge \cdots \wedge M_k) \vee \cdots$

- ***k-clause CNF:*** — ***Conjunctions*** *of at most* ***k unbounded disjunctive*** *clauses:*
  - $\ln(|H|)=O(kn)$
  - $C_1 \wedge C_2 \wedge \cdots \wedge C_k$

- ***k-CNF:*** — ***Conjunctions*** *of any number of clauses each limited to at most k* ***literals****:*
  - $\ln(|H|)=O(n^k)$
  - $((L_1 \vee L_2 \vee \cdots \vee L_k) \wedge (M_1 \vee M_2 \vee \cdots \vee M_k) \wedge \cdots$

Note: *All of these classes have* ***polynomial sample complexity*** *given a fixed value of* ***k****.*



۵) پیچیدگی نمونه برای فضای فرضیه نامتناهی

(*Sample Complexity for Infinite Hypothesis Spaces*)

Hint: *Showed* ***that sample complexity*** *for* ***PAC learning grows*** *as the* ***logarithm*** *of the* ***size*** *of the* ***hypothesis space***.

Note: *While* ***Equation*** $m \geq \frac{1}{\varepsilon}(\ln |H| + \ln (\frac{1}{\delta}))$ *is* ***quite useful***, *there are* **2** ***drawbacks*** *to* ***characterizing sample complexity*** *in* ***terms*** *of* ***|H|***.

1) *It can* ***lead*** *to* ***quite weak bounds*** *(recall that the* ***bound*** *on* δ *can be* ***significantly greater*** *than* **1** *for* ***large |H|***).

2) *In the case of* ***infinite hypothesis spaces*** *we* ***cannot apply*** *this* ***Equation*** *at all!*

How: *Here we consider a* ***second measure*** *of the* ***complexity*** *of* ***H***, *called the* ***Vapnik-Chervonenkis dimension*** *of* ***H*** (***VC dimension***, *or* ***VC(H)***, *for* ***short***).

VC: *Can* ***state bounds*** *on* ***sample complexity*** *that use* ***VC(H) rather than |H|***.

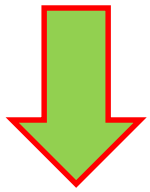
Note: *In many cases, the* ***sample complexity bounds based*** *on* ***VC(H)*** *will be* ***tighter*** *than those from* ***Equation*** $m \geq \frac{1}{\varepsilon}(\ln |H| + \ln (\frac{1}{\delta}))$

Note: *These* ***bounds*** *allow us to* ***characterize*** *the* ***sample complexity*** *of many* ***infinite hypothesis spaces***, *and can be shown to be* ***fairly tight***.

Infinite Hypothesis Spaces

Note: *The preceding analysis was restricted to*
finite hypothesis spaces.
Some ***infinite hypothesis spaces***
(such as those including ***real-valued thresholds*** *or parameters)*
are more expressive than others.
Compare a ***rule allowing*** *one*
threshold *on a* ***continuous feature*** (***length < 3cm***)
Vs. one allowing two
thresholds *on a* ***continuous feature*** (**1cm < *length* < 3cm**).

Note: *Need some* ***measure*** *of the*
expressiveness of infinite hypothesis spaces.
The ***Vapnik-Chervonenkis*** *(**VC**)* ***dimension***
provides just such a ***measure***,
denoted ***VC(H)***.
Analagous to
ln|H|, *there are* ***bounds*** *for* ***sample complexity*** *using*
VC(H).



Conjunction of Boolean literals

Example: Consider **learning** the **concept class** C_n of **conjunctions** of at most **n Boolean literals** $x_1, \ldots, x_n$.

Note: A **Boolean literal** is **either** a **variable** x_i ; $i \in [1, n]$, or its **negation** x_i.

Note: For $n = 4$, an example is the **conjunction**: $x_1 \wedge -x_2 \wedge x_4$, where $-x_2$ denotes the **negation** of the **Boolean literal** x_2.

Note: $(1, 0, 0, 1)$ is a **positive example** for this **concept** while $(1, 0, 0, 0)$ is a **negative example**.

Note: Observe that for $n = 4$, a **positive example** $(1, 0, 1, 0)$ implies that the target **concept cannot contain** the **literals** x_1 and x_3 and that it **cannot** contain the **literals** x_2 and x_4.

Note: In contrast, a **negative example** is not as **informative** since it is not known which of its n bits are **incorrect**.

Note: A **simple algorithm** for **finding** a **consistent hypothesis** is thus **based** on **positive examples** and consists of the following:-

for each **positive example** $(b_1, \ldots, b_n)$ and $i \in [1, n]$, if $b_i = 1$ then x_i is ruled out as a **possible literal** in the **concept class** and if $b_i = 0$ then x_i is **ruled** out.

Note: The **conjunction** of all the **literals** not **ruled** out is thus a **hypothesis consistent** with the **target**.

Note: An **example training sample** as well as a **consistent hypothesis** for the case $n = 6$.

Note: We have $|H| = |Cn| = 3n$, **since** each **literal** can be **included positively**, with **negation**, or **not included**. **Plugging** this into the **sample complexity bound** for **consistent** hypotheses yields the following **sample complexity bound** for any $m \geq \frac{1}{\epsilon}\left((\log 3)n + \log \frac{1}{\delta}\right)$ $\varepsilon > 0$ and $\delta > 0$:

Note: Thus, the **class of conjunctions** of at most **n Boolean literals** is **PAC-learnable**.

Note: **That** the **computational complexity** is also **polynomial**, since the **training** cost per **example** is in $O(n)$.

Note: For $\delta = 0.02$, $= 0.1$, and $n = 10$, the **bound** becomes $m \geq 149$.

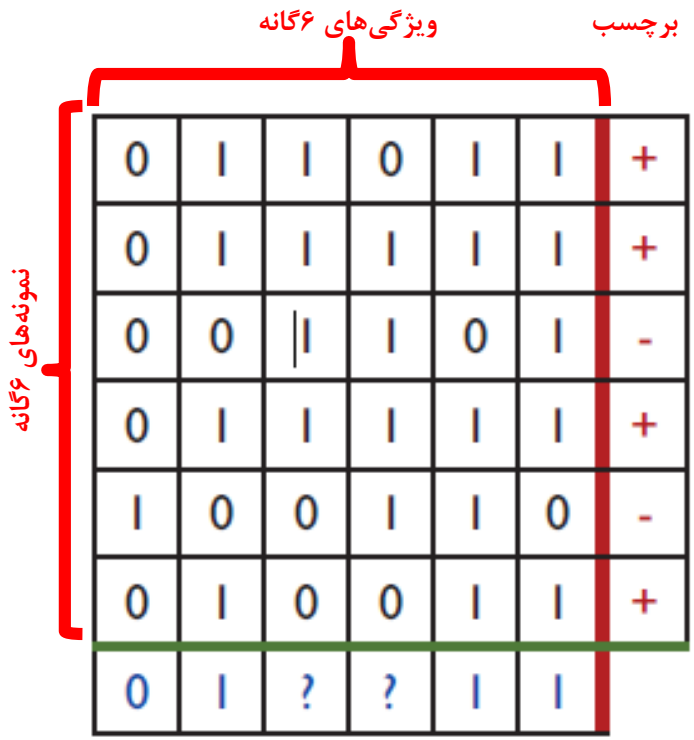
Note: Thus, for a **labeled** sample of at least **149 examples**, the **bound guarantees 99% accuracy** with a **confidence** of **at least 98%**.

Note: *Each of the **first six rows** of the **table***
*a **training example** with its **label**, + or −,*
__indicated__ in the __last column__.

Note: *The **last row contains 0** (**respectively 1**)*
*in **column** $i \in [1, 6]$*
*__if the i^{th} entry__ is **0** (**respectively 1**)*
*for all the **positive examples**.*

Note: *It **contains** "**?**" if **both 0** and **1***
*__appear__ as an i^{th} **entry***
*for some **positive example**.*

Note: *Thus, for this **training sample**,*
*the **hypothesis returned** by*
*the **consistent algorithm***
*__described__ in the **text** is $x_1 \wedge x_2 \wedge x_5 \wedge x_6$.*



Computational Complexity of Learning

Note: *Determining whether or not there exists a*

k-term DNF *or* ***k-clause CNF*** ***formula***

consistent *with a given* ***training*** *set is* ***NP-hard***.

Note: *Therefore, these* ***classes*** *are* ***not PAC-learnable***

due to ***computational complexity***.

Note: *There are* ***polynomial time algorithms*** *for* ***learning*** ***k-CNF*** *and* ***k-DNF***.

Note: *Construct all* ***possible disjunctive clauses*** *(****conjunctive terms****) of*

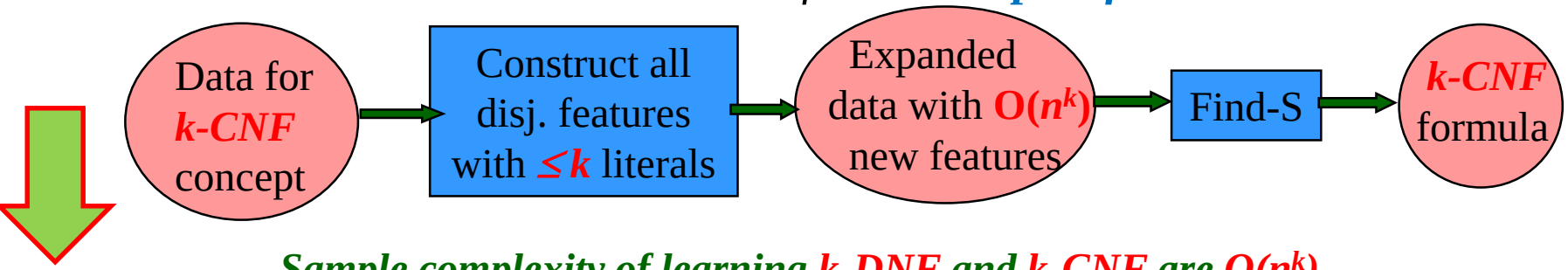
at ***most k literals*** *(there are* $O(n^k)$ *of these),*

add each as a new ***constructed feature****,*

and then use ***FIND-S*** *(****FIND-G****) to* ***find*** *a*

purely conjunctive *(****disjunctive****)* ***concept***

in ***terms*** *of these* ***complex features***.



Sample complexity of learning k-DNF and k-CNF are $O(n^k)$.

Training on $O(n^k)$ ***examples with*** $O(n^k)$ ***features takes*** $O(n^{2k})$ ***time***

Probabilistic Algorithms

Note: *Since **PAC** learnability only*
requires an approximate answer with
***high probability**, a probabilistic algorithm that only halts and returns*
*a **consistent hypothesis** in*
***polynomial time** with a*
***high-probability** is **sufficient**.*

Note: *However, it is generally assumed that*
***NP** complete problems*
***cannot** be **solved** even with high probability by a*
***probabilistic polynomial-time algorithm**,*
*i.e. **RP ≠ NP**.*

Note: *Therefore, given this assumption, classes like*
***k-term DNF** and*
***k-clause CNF** are*
*not **PAC** learnable in that form.*

(CO)LT Conclusions

Learning Algorithms: *The **PAC framework** provides a **theoretical framework** for **analyzing** the **effectiveness** of **learning algorithms**.*

Sample Complexity: *The **sample complexity** for any **consistent learner** using some **hypothesis space**, **H**, can be determined from a **measure** of its **expressiveness** **|H|** or **VC(H)**, **quantifying bias** and relating it to **generalization**.*

Computational Complexity: *If sample **complexity** is **tractable**, then the **computational complexity** of finding a **consistent hypothesis** in **H** governs its **PAC learnability**.*

Constant Factors: *Constant factors are more important in **sample complexity** than in **computational complexity**, since our ability to gather data is generally not growing **exponentially**.*

Worst-case Upper Bounds: *Experimental results suggest that **theoretical sample complexity bounds over-estimate** the **number** of **training instances** needed in practice since they are **worst-case upper bounds**.*

(CO)LT Conclusions (cont)

Note: *Additional results produced for analyzing:*

1) ***Learning*** *with* ***queries***

2) ***Learning*** *with* ***noisy data***

3) *Average case* ***sample complexity*** *given assumptions about the* ***data distribution***.

4) ***Learning finite automata***

5) ***Learning neural networks***

Good News: *Analyzing* ***practical algorithms*** *that use a* ***preference bias*** *is* ***difficult***.

Some effective ***practical algorithms motivated*** *by* ***theoretical*** *results:*

1) ***Boosting***

2) ***Support Vector Machines*** *(****SVM****)*

Basic Combinatorics Counting

	dup[s] allowed	dup[s] not allowed
order relevant	**samples**	**permutations**
order irrelevant	**selections**	**combinations**

Pick 2 from {a , b}

samples	**selections**	**permutations**	**combinations**
aa	**aa**	**ab**	**ab**
ab	**ab**	**ba**	
ba	**bb**		
bb			

k - samples : $n^k = 4$

k - permutations : $\frac{n!}{(n-k)!} = 2$

k - selections: $\binom{n+k-1}{k} = \frac{(n+k-1)!}{k!(n-1)!} = 3$

k - combinations : $\binom{n}{k} = \frac{n!}{k!(n-k)!} = 1$

All $O(n^k)$

Exercises

Q1 Consider training a two-input perceptron. Give an upper bound on the number of training examples sufficient to assure with 90% confidence that the learned perceptron will have true error of at most 5%. Does this bound seem realistic?

Q2 Consider the class C of concepts of the form $(a \leq x \leq b) \wedge (c \leq y \leq d)$, where a, b, c, and d are integers in the interval (0, 99). Note each concept in this class corresponds to a rectangle with integer-valued boundaries on a portion of the x, y plane. Hint: Given a region in the plane bounded by the points $(0, 0)$ and $(n-1, n-1)$, the number of distinct rectangles with integer-valued boundaries within this region is $(\frac{n(n+1)}{2})^2$.

(*a*) Give an upper bound on the number of randomly drawn training examples sufficient to assure that for any target concept c in C, any consistent learner using $H = C$ will, with probability 95%, output a hypothesis with error at most .15.

(*b*) Now suppose the rectangle boundaries a, b, c, and d take on *real* values instead of integer values. Update your answer to the first part of this question.

Q3

In this chapter we derived an expression for the number of training examples sufficient to ensure that every hypothesis will have true error no worse than ϵ plus its observed training error $error_D(h)$. In particular, we used Hoeffding bounds to derive Equation (7.3). Derive an alternative expression for the number of training examples sufficient to ensure that every hypothesis will have true error no worse than $(1+\gamma)error_D(h)$. You can use the general Chernoff bounds to derive such a result.

Chernoff bounds: Suppose $X_1, \ldots, X_m$ are the outcomes of m independent coin flips (Bernoulli trials), where the probability of heads on any single trial is $\Pr[X_i = 1] = p$ and the probability of tails is $\Pr[X_i = 0] = 1 - p$. Define $S = X_1 + X_2 + \cdots + X_m$ to be the sum of the outcomes of these m trials. The expected value of S/m is $E[S/m] = p$. The Chernoff bounds govern the probability that S/m will differ from p by some factor $0 \leq \gamma \leq 1$.

$$\Pr[S/m > (1+\gamma)p] \leq e^{-mp\gamma^2/3}$$

$$\Pr[S/m < (1-\gamma)p] \leq e^{-mp\gamma^2/2}$$

Q4

Consider a learning problem in which $X = \Re$ is the set of real numbers, and $C = H$ is the set of intervals over the reals, $H = \{(a < x < b) \mid a, b \in \Re\}$. What is the probability that a hypothesis consistent with m examples of this target concept will have error at least ϵ? Solve this using the VC dimension. Can you find a second way to solve this, based on first principles and ignoring the VC dimension?

Q5

Consider the space of instances X corresponding to all points in the x, y plane. Give the VC dimension of the following hypothesis spaces:

(a) H_r = the set of all rectangles in the x, y plane. That is, $H = \{((a < x < b) \wedge (c < y < d)) | a, b, c, d \in \Re\}$.

(b) $H_c =$ circles in the x, y plane. Points inside the circle are classified as positive examples

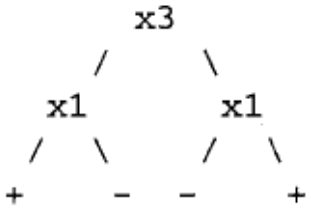
(c) $H_t =$triangles in the x, y plane. Points inside the triangle are classified as positive examples

Q6

Write a consistent learner for H_r from Exercise 7.5. Generate a variety of target concept rectangles at random, corresponding to different rectangles in the plane. Generate random examples of each of these target concepts, based on a uniform distribution of instances within the rectangle from $\langle 0, 0 \rangle$ to $\langle 100, 100 \rangle$. Plot the generalization error as a function of the number of training examples, m. On the same graph, plot the theoretical relationship between ϵ and m, for $\delta = .95$. Does theory fit experiment?

Q7

Consider the hypothesis class H_{rd2} of "regular, depth-2 decision trees" over n Boolean variables. A "regular, depth-2 decision tree" is a depth-2 decision tree (a tree with four leaves, all distance 2 from the root) in which the left and right child of the root are *required to contain the same variable*. For instance, the following tree is in H_{rd2}.



(*a*) As a function of n, how many syntactically distinct trees are there in H_{rd2}?

(*b*) Give an upper bound for the number of examples needed in the PAC model to learn H_{rd2} with error ϵ and confidence δ.

(*c*) Consider the following WEIGHTED-MAJORITY algorithm, for the class H_{rd2}. You begin with all hypotheses in H_{rd2} assigned an initial weight equal to 1. Every time you see a new example, you predict based on a weighted majority vote over all hypotheses in H_{rd2}. Then, instead of eliminating the inconsistent trees, you cut down their weight by a factor of 2. How many mistakes will this procedure make at most, as a function of n and the number of mistakes of the best tree in H_{rd2}?

Q 8 This question considers the relationship between the PAC analysis considered in this chapter and the evaluation of hypotheses discussed in Chapter 5. Consider a learning task in which instances are described by n boolean variables (e.g., $x_1 \wedge \bar{x_2} \wedge x_3 \ldots \bar{x_n}$) and are drawn according to a fixed but unknown probability distribution $\mathcal{D}$. The target concept is known to be describable by a conjunction of boolean attributes and their negations (e.g., $x_2 \wedge \bar{x_5}$), and the learning algorithm uses this concept class as its hypothesis space H. A consistent learner is provided a set of 100 training examples drawn according to $\mathcal{D}$. It outputs a hypothesis h from H that is consistent with all 100 examples (i.e., the error of h over these training examples is zero).

(a) We are interested in the true error of h, that is, the probability that it will misclassify future instances drawn randomly according to $\mathcal{D}$. Based on the above information, can you give an interval into which this true error will fall with at least 95% probability? If so, state it and justify it briefly. If not, explain the difficulty.

(b) You now draw a new set of 100 instances, drawn independently according to the same distribution $\mathcal{D}$. You find that h misclassifies 30 of these 100 new examples. Can you give an interval into which this true error will fall with approximately 95% probability? (Ignore the performance over the earlier training data for this part.) If so, state it and justify it briefly. If not, explain the difficulty.

(c) It may seem a bit odd that h misclassifies 30% of the new examples even though it perfectly classified the training examples. Is this event more likely for large n or small n? Justify your answer in a sentence.