Lernalgorithmen

SoSe 2012 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Lernalgorithmen Gesamtübersicht

0. Einführung
1. Identifikation (aus positiven Beispielen)
2. Zur Identifikation regulärer Sprachen, mit XML-Anwendung
3. HMM — Hidden Markov Models
4. Lernen mittels Anfragen & zur Roboterorientierung
5. Lernen mit negativen Beispielen
6. PAC-Lernen

PAC-Lernen

Hauptquellen (Überblick):

L. G. Valiant A theory of the learnable. *Communications of the ACM* 27 (1984), pp. 1134–1142.

D. Angluin. Learning regular sets from queries and counterexamples. *Information and Control* 75 (1987), pp. 87–106.

M. J. Kearns und U. V. Vazirani. An Introduction to Computational Learning Theory. 2. Auflage, MIT Press, 1997.

B. K. Natarajan. Machine Learning; a Theoretical Approach. Morgan Kaufmann Publishers, 1991.

K. Morik. Maschinelles Lernen; Skript zur Vorlesung, 12. April 1999; Universität Dortmund.

F. Stephan. Einführung in die Lerntheorie. Skript Uni Heidelberg

P. Fischer Algorithmisches Lernen. Teubner, 1999.

Motivierendes Beispiel:

Wir wollen das Konzept eines “normalgebauten Menschen” von einem Experten lernen.
Wir wissen lediglich, dass der Experte genau die Menschen als “normalgebaut” ansieht, deren Körpergröße und -gewicht in einem bestimmten Intervall liegen.
Unser Zielkonzept ist also ein Kreuzprodukt von Intervallen, m.a.W.:
ein achsenparalleles Rechteck, wobei die Koordinaten sich aus Gewichtsachse und Längenachse ergeben.
Für unsere Lernaufgabe erhalten wir klassifizierte Beispiele.
Nach so einer Trainingsphase formen wir eine Hypothese.
Beispielverfahren:
Wähle kleinstes die positiven Beispiele umschließendes Rechteck.
Wie gut ist unsere Hypothese?

Motivierendes Beispiel: (Forts.)

Wie gut ist unsere Hypothese H für das Zielkonzept C?

Erste Idee: Messe den Fehlerbereich $H \Delta C$, z.B. durch die Euklidische Fläche.

Das ist aber “unfair”, denn sicher kommen nicht alle Größen- / Gewichtskombinationen überhaupt in Frage.

Besser: Wir nehmen an, es gibt eine Wahrscheinlichkeitsverteilung für alle Menschen, und wir messen die “Masse” von $H \Delta C$.

Und: Wie gelangen wir überhaupt zu unserer Stichprobe?

Präzisere Begriffe

X: das *Lernuniversum* mit Wahrscheinlichkeitsverteilung D
$\mathcal{C} \subseteq 2^X$: *(Ziel-)Konzeptklasse*; $\mathcal{H} \subseteq 2^X$ Hypothesenklasse
(im "Hintergrund" meist syntaktische Beschreibung (*Repräsentation*) $R : \Sigma^* \to \mathcal{C}$ oder $\mathcal{H}$; damit kann man die Größe von Konzepten messen)
Stichprobe: Folge klassifizierter Beispiele; entspr. Stichprobenraum:

$$\mathcal{S}(X, \mathcal{C}) = \{(\langle x_1, C(x_1)\rangle, \ldots, \langle x_m, C(x_m)\rangle) \mid x_i \in X, m \in \mathbb{N}, C \in \mathcal{C}\}$$

$EX_{D,C}$: Prozedur (Orakel), die gemäß der Verteilung D mittels C klassifizierte Beispiele erzeugt; ein Aufruf liefert Paar $\langle x, C(x)\rangle \in X \times \{0, 1\}$ zurück.
Wir gehen meist davon aus, dass Stichprobe durch unabhängige Aufrufe von $EX_{D,C}$ erzeugt wurde.
Ein *Lernalgorithmus* ist nun eine Abb. $A : \mathcal{S}(X, \mathcal{C}) \to \mathcal{H}$.
Der *Fehler* von Hypothese H bzgl. Konzept C ist: $\varepsilon(H) := D(H \Delta C)$.

Präzisere Begriffe (Forts.)

Ziel: Finden von Hypothese H, die ϵ*-gut* ist, d.h., $\varepsilon(\mathrm{H}) < \epsilon$.

Unrealistische Forderung: Lernalgorithmus findet stets gute Hypothese.
Besser: Nur auf einem kleinen Anteil $\delta \in [0, 1]$ möglicher Hypothesen (der *Unzuverlässigkeit*) liefert der Algorithmus schlechte Hypothesen.

Im Folgenden:
ϵ (obere Schranke) für zulässigen Fehler und δ zulässige Unzuverlässigkeit.
Frage: Wie groß muss Stichprobe sein: damit ϵ und δ eingehalten werden?
$\rightsquigarrow$ *Stichprobenkomplexität* $\mathrm{m}(\epsilon, \delta)$

Definition [PAC-Lernbarkeit]

Eine Begriffsklasse $\mathcal{C} \subseteq 2^X$ ist *PAC-lernbar* durch einen Hypothesenraum $\mathcal{H} \subseteq 2^X$, wenn es einen Algorithmus $A(\cdot,\cdot)$ mit Stichproblemkomplexität $m(\cdot,\cdot)$ gibt, sodass

—für jede Wahl der Genauigkeits- und Zuverlässigkeitsparameter $\epsilon, \delta \in (0,1)$ und

—bei allen beliebigen aber festen Wahrscheinlichkeitsverteilungen D über X und

—für alle $C \in \mathcal{C}$ gilt:

—A erhält eine Stichprobe S_C der Größe $m(\epsilon,\delta)$,

—A antwortet (in einer Zeit, die durch ein Polynom über $1/\epsilon, 1/\delta, |C|, |X|$ begrenzt ist)

—mit einer Wahrscheinlichkeit von mindestens $1-\delta$

mit einer Hypothese $H \in \mathcal{H}$, deren Fehler nicht größer ist als ϵ.

$A(\delta,\epsilon)$ hat Zugriff auf Orakel $EX_{D,C}$.

Mehr zu PAC

Gilt $\mathcal{H} = \mathcal{C}$, so heißt $\mathcal{C}$ auch *streng PAC-lernbar*.

Das eigentliche Lernkriterium lässt sich mit $m = m(\epsilon, \delta)$ auch wie folgt schreiben:

$$Pr_{S_C \sim D^m}[D(A(S_C)\Delta C) \geq \epsilon] < \delta.$$

PAC-Kriterium

Um Effizienz sinnvoll einfordern zu können, betrachtet man zumeist *strukturierte Universen* und demzufolge strukturierte Konzeptklassen: $X = \bigcup_{n\in\mathbb{N}} X_n$, $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_n$, wobei $\mathcal{C}_n \subseteq 2^{X_n}$.

Dann antwortet A bei Stichprobenkomplexität $m(\epsilon, \delta, n)$ in Zeit $p(1/\epsilon, 1/\delta, n)$ für ein Polynom p. $\rightsquigarrow$ *polynomielle PAC-Lernbarkeit*.

Zwei Sichten auf PAC-Algorithmen

1) Entwirf Algorithmus, der wiederholt Anfragen an das $\mathrm{EX}_{\mathrm{D},\mathrm{C}}$-Orakel stellen darf. Die Laufzeit wird wesentlich von der Anzahl der "nötigen" Orakel-Anfragen abhängen.

2) Entwirf Algorithmus, der als Eingabe eine Stichprobe vom Umfang m erhält, d.h. eine Folge von gemäß C klassifizierten Beispielen, die selbst durch m unabhängige Aufrufe von $\mathrm{EX}_{\mathrm{D},\mathrm{C}}$ erzeugt wurde.

Beide Darstellungen lassen sich ineinander überführen und werden im Folgenden unterschiedslos gebraucht.

Die Eingabe im Fall 1) besteht (nur) aus den Parametern ϵ und δ sowie evtl. n. Im Fall 2) gehen diese Parameter oft nur zur Bestimmung von $\mathrm{m} = \mathrm{m}(\epsilon, \delta(, \mathrm{n}))$ ein.

Ein Beispiel

$\mathcal{APR}_n$: Klasse der achsenparallelen n-dimensionalen Rechtecke.
$\mathcal{APR}$: Klasse der achsenparallelen endlich-dimensionalen Rechtecke.
Zugehöriges Universum: $X = \bigcup_{n \in \mathbb{N}} \mathbb{R}^n$.

Der vorher schon beschriebene Lerner genügt, um zu zeigen:

Satz: Die Klasse $\mathcal{APR}_2$ ist (polynomiell) streng PAC-lernbar mit Stichprobenkomplexität $m = \lceil (4/\epsilon) \ln(4/\delta) \rceil$.

Der Algorithmus SER formaler...

Eingabe: Stichprobe $(\langle (x_1, y_1), \ell_1 \rangle, \ldots, \langle (x_m, y_m), \ell_m \rangle)$
für ein Konzept $C \in \mathcal{APR}_2$, mit $\ell_i = C(x_i, y_i)$.

`Bestimme` $x_{min} = \min\{x_i \mid i \in \{1, \ldots, m\}, \ell_i = 1\}$.
`Bestimme` $y_{min} = \min\{y_i \mid i \in \{1, \ldots, m\}, \ell_i = 1\}$.
`Bestimme` $x_{max} = \max\{x_i \mid i \in \{1, \ldots, m\}, \ell_i = 1\}$.
`Bestimme` $y_{max} = \max\{y_i \mid i \in \{1, \ldots, m\}, \ell_i = 1\}$.
`Liefere Hypothese` $H = [x_{min}, x_{max}] \times [y_{min}, y_{max}]$.

Beweis

Sei $C = [l, r] \times [b, t]$ das Zielkonzept C, D die zugrunde liegende Verteilung auf $\mathbb{R}^2$ und $S_{C,D}$ eine Stichprobe für C. Sei $H = \text{SER}(S_{C,D})$.
Klar: $H \subseteq C$. $\rightsquigarrow$ Fehlerbereich $F = C \Delta H = C \setminus H$ bildet "Rand".
Angenommen, $D(C) > \epsilon/2$. (Sonst hätte selbst das leere Rechteck einen Fehler $< \epsilon$.)
Bilde vier achsenparallele Randrechtecke R_i mit $D(R_i) = \epsilon/4$.
Z.B.: $R_1 = [l, x^*] \times [b, t]$ mit $x^* = \inf\{x \geq l \mid D([l, x] \times [b, t]) \geq (\epsilon/4)\}$.
Enthält $S_{C,D}$ von jedem R_i ein (positives) Beispiel, so gilt: $F \subset \bigcup_{i=1}^{4} R_i$.

$$D(F) < D\left(\bigcup_{i=1}^{4} R_i\right) \leq \sum_{i=1}^{4} D(R_i) \leq \sum_{i=1}^{4} \frac{\epsilon}{4} = \epsilon.$$

Wie groß muss $S_{C,D}$ sein, damit dies mit Wahrsch. $\geq (1 - \delta)$ auftritt?

Wahrsch., dass ein zufälliges Beispiel nicht in R_i liegt, ist $\leq 1 - (\epsilon/4)$.
m Versuche, die stets R_i verfehlen, haben Wahrsch. $\leq (1 - (\epsilon/4))^m$.
Wahrsch., mind. eines der R_i bei m-maligem Ziehen zu verfehlen, ist $\leq 4(1 - (\epsilon/4))^m$.
Ist dieser Ausdruck $< \delta$, so ist $Pr[D(F) < \epsilon] > (1 - \delta)$. Wir müssen

$$4(1 - (\epsilon/4))^m < \delta$$

nach m auflösen. Das ist schwierig, daher Abschätzung:

$$\begin{aligned} 4\left(1 - \left(\frac{\epsilon}{4}\right)\right)^m &= 4\left(1 - \left(\frac{\epsilon}{4}\right)\right)^{m \cdot \frac{4}{\epsilon}\frac{\epsilon}{4}} \\ &= 4\left(\left(1 - \left(\frac{\epsilon}{4}\right)\right)^{\frac{4}{\epsilon}}\right)^{m \cdot \frac{\epsilon}{4}} \leq 4\left(\frac{1}{e}\right)^{m \cdot \frac{\epsilon}{4}} = 4 \cdot e^{m \cdot \frac{\epsilon}{4}} \end{aligned}$$

Nützliche Formeln

$$\forall x > 0 : (1 + x^{-1})^x < e$$

$$\forall x > 0 : (1 - x^{-1})^x < e^{-1}$$

Daraus folgt insbesondere:

$$\forall x > 0 : (1 - x) < e^{-x}$$

Nach Valiant (1984) konnte Tschernoff (Chernoff) zeigen:
Nach m unabhängigen Bernoulli-Experimenten mit Erfolgswahrscheinlichkeit p ist die Wahrsch. dafür, dass höchstens $k < mp$ dieser Experimente erfolgreich waren, höchstens

$$\left(\frac{m - mp}{m - k}\right)^{m-k} \left(\frac{mp}{k}\right)^{k} < e^{-mp+k} \left(\frac{mp}{k}\right)^{k}.$$

Eine Verallgemeinerung

Der Beweis lässt sich nicht nur für $n = 2$ führen, sodass wir festhalten können:

Satz: Die Klasse $\mathcal{APR}$ ist polynomiell streng PAC-lernbar mit Stichprobenkomplexität $m = \lceil (2n/\epsilon) \ln(2n/\delta) \rceil$.

Ein Lemma von Valiant, leicht angepasst

Es sei $L(p, q, S)$ für $p, q \in (0, 1)$ und $S \in \mathbb{N}$ die kleinste ganze Zahl, sodass nach $L(p, q, S)$ unabhängigen Bernoulli-Experimenten mit Erfolgswahrscheinlichkeit wenigstens p die Wahrsch., weniger als S Erfolge zu beobachten, kleiner als q ist.

Lemma: $L(p, q, S) \leq 2p(S + \ln(q))$.

Setzt man die "Lösung" $m = 2p(S + \ln(q^{-1}))$ sowie $k = S$ in die Chernoff-Ungleichung ein, ergibt sich als Wahrsch.schranke:

$$e^{-2S-2\ln(q^{-1})+S} \cdot [2(1 + \ln(q^{-1})/S)]^{(S/\ln(q^{-1}))\ln(q^{-1})} \leq e^{-S-2\ln(q^{-1})} \cdot 2^S \cdot e^{\ln(q^{-1})} = (2/e)^S \cdot e^{-\ln(q^{-1})} \leq q.$$

Bei der ersten Ungl. benutze $(1 + x^{-1})^x < e$ für $x = \ln(q^{-1})/S$. □

Logische Erinnerungen

Definition [k-KNF$_n$] Eine Formel $C_1 \wedge \ldots \wedge C_\ell$, bei der jedes C_i (Klausel) eine Disjunktion von höchstens k Literalen über n Booleschen Variablen ist, ist in k-*konjunktiver Normalform* (k-KNF).

Definition [k-DNF$_n$] Eine Formel $M_1 \vee \ldots \vee M_\ell$, bei der jedes M_i (Monom) eine Konjunktion von höchstens k Literalen über n Booleschen Variablen ist, ist in k-*disjunktiver Normalform* (k-DNF).

Lemma: Eine Formel aus k-DNF$_n$ enthält höchstens $O((2n)^k)$ viele Monome.

Sei $M(n,k)$ die genaue Zahl der Monome, die eine Formel aus k-DNF$_n$ höchstens enthält.

n strukturiert DNF-Formeln. So lässt sich formulieren:

Satz: Die Klasse k-DNF ist polynomiell streng PAC-lernbar mit Stichprobengröße

$$m = \mathcal{O}((2/\epsilon)((2n)^k + \ln(1/\delta))).$$

Grundidee des Algorithmus DEL-MON:
Starte mit einer immer erfüllten, maximalen k-DNF$_n$-Formel

$$H_0^n = x_1 \vee \bar{x}_1 \vee \cdots \vee x_n \vee \bar{x}_n \vee (x_1 \wedge x_2) \vee (\bar{x}_1 \wedge x_2) \vee \cdots \vee (\bar{x}_{n-k+1} \vee \cdots \bar{x}_n).$$

Lösche dann nach und nach Monome, die von negativen Beispielbelegungen wahr gemacht würden.
Dazu werden in einer Schleife nacheinander alle Beispiele $(\vec{a}_i, \ell_i)$ mit $\ell_i = 0$ betrachtet, wobei $\vec{a}_i \in \{0,1\}^n$ Belegung und $\ell_i \in \{0,1\}$ Ergebnis.
Sei H_i Hypothese nach dem i-ten Schleifendurchlauf.

$$F_i = C \Delta H_i = \{\vec{a} \in \{0,1\}^n \mid C(\vec{a}) \neq H_i(\vec{a})\}$$
$$= \{\vec{a} \in \{0,1\}^n \mid C(\vec{a}) = 0 \wedge H_i(\vec{a}) = 1\}.$$

Klar: $F_i \subseteq F_{i-1}$, also $D(F_i) \leq D(F_{i-1})$.

Angenommen, $\varepsilon(H_m) = D(F_m) \geq \epsilon$.

Wir haben also nicht genügend viele Monome entfernt.

Da die Stichprobe mithilfe von D ermittelt wurde, hat jedes gezogene Beispiel eine Wahrsch. $\geq \epsilon$, aus F_m zu stammen.

DEL-MON versagt (möglicherweise), wenn das Ereignis "Beispiel liegt in F_m" bei m (unabh.) Versuchen weniger als $M(n,k)$-mal auftritt.

Betrachte also (wiederholtes) Bernoulli-Experiment mit Erfolgswahrsch. ϵ.

Ziel: Bestimme $m = m(\epsilon, \delta, n)$ so, dass die Wahrsch., weniger als $M(n,k)$ viele "Erfolge" zu haben, kleiner als δ ist.

Das Lemma von Valiant liefert die Behauptung. $\square$

Analog kann man zeigen:

Satz: Die Klasse k-KNF ist polynomiell streng PAC-lernbar mit Stichprobengröße

$$m = \mathcal{O}((2/\epsilon)((2n)^k + \ln(1/\delta))).$$

Philosophische Notiz:

Unsere Lerner für $\mathcal{APR}$ und für k-KNF benutzen nur die positiven Beispiele, der Lerner für k-DNF nur die negativen Beispiele!

Satz: Die Klasse $S_{mon} = \{f_a : \{0,1\}^n \to \{0,1\} \mid (\forall x)[f_a(x) = 0 \Leftrightarrow x \leq_{lex} a]\}$ aller monotonen Funktionen ist PAC-lernbar, aufgefasst als "Argumentmengen".

Beweis Der Lernalgorithmus M bestimmt unter den Beispielen aus $\{0,1\}^n$ das lexikographisch größte b mit $f(b) = 0$ und wählt die Funktion f_b.
Falls kein solches Beispiel existiert, wird $b = 0^n$ gewählt.
Zu bestimmen bleibt die notwendige Anzahl m der Beispiele, damit der Algorithmus seinen Spezifikationen genügt.

Zu gegebenem n, δ und ϵ liest M dann $m = \lceil \frac{2}{\delta\epsilon} \rceil$ Beispiele $(x, f(x))$ ein und bestimmt unter ihnen wie angegeben eine Hypothese $g : \{0,1\}^n \to \{0,1\}$, also $g = f_b$. Wir bestimmen m so, dass gilt: Mit Wahrscheinlichkeit $1-\delta$ ist der Fehler von g im Vergleich mit der Zielfunktion $f = f_a$ bezüglich der Verteilung D durch ϵ begrenzt, d.h.: $D(\{x \mid g(x) \neq f(x)\}) \leq \epsilon$.

Es sei c der lexikographisch erste Vektor mit $D(\{x \mid c <_{lex} x \leq_{lex} a\}) \leq \epsilon$.
Dies liefert eine Abschätzung für den möglichen Fehlerbereich.
Es gibt zwei Fälle:

- $c = 0^n$: Dann ist stets $c \leq_{lex} b \leq_{lex} a$ und mit Wahrscheinlichkeit 1 wird eine Formel gelernt, die der Bedingung $D(\{x \mid f(x) \neq g(x)\}) \leq \epsilon$ genügt.

- $c \neq 0^n$: Dann ist $D(\{x \mid c \leq_{lex} x \leq_{lex} a\}) > \epsilon$. $\rightsquigarrow$
  Mit Wahrscheinlichkeit $1-(1-\epsilon)^m$ ist 1 aus m Beisp. in diesem Intervall.
  m muss so groß gewählt werden, dass $1-(1-\epsilon)^m \geq 1-\delta$ ist.

Also ist die Bedingung $1-(1-\epsilon)^m \geq 1-\delta$ hinreichend. Forme um:

$$\begin{aligned} 1-\delta &\leq 1-(1-\epsilon)^m \\ (1-\epsilon)^m &\leq \delta \\ m \cdot \log(1-\epsilon) &\leq \log(\delta) \\ m &\geq \frac{\log(\delta)}{\log(1-\epsilon)} \end{aligned}$$

Nun benötigt man am Ende nur ein hinreichend großes m, dieses muss aber nicht optimal sein. Man kann den Term $|\log(1-\epsilon)| = \epsilon + \epsilon^2 + \epsilon^3 + \ldots$ nach unten durch ϵ abschätzen und $|\log(\delta)|$ nach oben durch $\frac{1}{\delta}$ abschätzen.
Dadurch erhält man die einfachere Bedingung $m \geq \frac{1}{\delta\epsilon}$, ein Polynom in n, $\frac{1}{\delta}$ und $\frac{1}{\epsilon}$.
Die Rechenzeit ist ebenfalls polynomial in n und der Beispielanzahl m, also polynomial in den drei Parametern von M. □
Um mit 99% Wahrscheinlichkeit einen Fehler von 1% zu haben, benötigt man 10000 Beispiele.

Das Lemma von Haussler, eine Alternative zum Lemma von Valiant

Die *Stichprobenkomplexität* eines Algorithmus für gegebenes $\mathcal{H}$, $\mathcal{C}$, ϵ und δ ist die Anzahl von Beispielen $m(\epsilon, \delta, \mathcal{H})$, (hier: Abh. von $\mathcal{H}$) nach denen der Algorithmus spätestens ein beliebiges $C \in \mathcal{C}$ mit Wahrscheinlichkeit δ bis auf einen Fehler von ϵ approximiert (PAC-gelernt) hat.

Zunächst benutzen wir einen Hypothesenraum mit endlicher Größe $|\mathcal{H}|$.

Satz: [Stichprobenkomplexität] Die Stichprobenkomplexität m eines endlichen Hypothesenraums $\mathcal{H}$ mit $\mathcal{H} = \mathcal{C}$ ist begrenzt durch

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

das kleinste m, für das die Ungleichung gilt, gibt die Anzahl von Beispielen an, nach denen spätestens jedes $C \in \mathcal{C}$ PAC-gelernt werden kann.

Beweis (nach Haussler) Der Versionenraum zu $\mathcal{H}$ enthält zu jedem Zeitpunkt nur mit den bisher gesehenen Beispielen konsistente Hypothesen.
Der Versionenraum ist also endlich.
Wir betrachten den schlimmsten Fall: Bezüglich der bisher gesehenen Beispiele ist eine Hypothese im Versionenraum zwar korrekt, tatsächlich ist ihr Fehler aber größer als ϵ.
Die Wahrscheinlichkeit, dass eine beliebige Hypothese mit Fehler größer als ϵ ein zufällig gezogenes Beispiel richtig klassifiziert, ist also höchstens $(1-\epsilon)^m$.
Bei k Hypothesen ist die Wahrscheinlichkeit, dass eine von ihnen schlecht ist (d.h. einen Fehler größer als ϵ hat), höchstens $k(1-\epsilon)^m$.
Daher ist die Wahrscheinlichkeit, dass eine der Hypothesen im Versionenraum einen Fehler größer als ϵ hat, höchstens $|\mathcal{H}|(1-\epsilon)^m$.
Dies beschränkt die Wahrscheinlichkeit, dass m Beispiele nicht ausreichen, um die schlechten Hypothesen aus dem Versionenraum zu tilgen.
Da $\epsilon \in (0,1)$, können wir sie ganz grob abschätzen als $|\mathcal{H}|e^{-\epsilon m}$.
Damit ein Begriff PAC-gelernt wird, muss diese Wahrscheinlichkeit kleiner sein als δ:

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

Eine Umformung dieser Gleichung ergibt das Gewünschte. □

Hinweis: Hieraus ergibt sich auch die Lernbarkeit von k-DNF wegen $|\mathcal{H}_{n,k}| = c^{(2n)^k}$.

Vapnik-Chervonenkis-Dimension (VC-Dimension)

Sei $\mathcal{H}$ der Hypothesenraum über X und S eine m-elementige Teilmenge von X. S wird von $\mathcal{H}$ *zerschmettert* (shattered), falls es für alle $S' \subseteq S$ eine Hypothese $h_{S'} \in \mathcal{H}$ gibt, die S' abdeckt, d.h. $S \cap h_{S'} = S'$. M.a.W.: $2^S = \{h \cap S \mid h \in \mathcal{H}\}$. Alle Teilmengen von S werden also durch Hypothesen in $\mathcal{H}$ erkannt.

Die *Vapnik-Chervonenkis-Dimension* von $\mathcal{H}$, $\mathrm{VCdim}(\mathcal{H})$, ist die Anzahl der Elemente von der größten Menge S, wobei S von $\mathcal{H}$ zerschmettert wird.

$$\mathrm{VCdim}(\mathcal{H}) = \max\{m : \exists S \subseteq X, |S| = m, \mathcal{H} \text{ zerschmettert } S\}.$$

Sie gibt also an, wieviele Unterschiede $\mathcal{H}$ machen kann. Wenn es kein Maximum der Kardinalität von S gibt, ist VCdim unendlich.

Mitteilung [Blumer et al.] Unter gewissen maßtheoretischen Annahmen gilt:
$\mathcal{C}$ ist streng PAC-lernbar gdw. $\mathrm{VCdim}(\mathcal{C}) < \infty$.

Satz: [VC-Dimension endlicher Hypothesenräume]
Wenn der Hypothesenraum $\mathcal{H}$ endlich ist, dann ist $\mathrm{VCdim}(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$.

Beweis: Um eine Menge der Größe m zu zerschmettern, sind mindestens 2^m verschiedene Hypothesen nötig, weil es ja 2^m verschiedene Teilmengen gibt. □

Umgekehrt kann man für $\mathcal{H} \subseteq 2^X$ per Induktion zeigen:

$$|\mathcal{H}| \leq (|X| + 1)^{\mathrm{VCdim}(\mathcal{H})}.$$

Mitteilung
Die Ermittlung der genauen Vapnik-Chervonenkis-Dimension ist NP-hart.

Ein konkretes Beispiel

Lemma: $\mathrm{VCdim}(\mathcal{APR}_2) = 4$.

$\mathrm{VCdim}(\mathcal{APR}_2) \geq 4$ bedeutet: Es gibt mindestens eine vierelementige Teilmenge des $\mathbb{R}^2$, von der jede Teilmenge durch achsenparallele Rechtecke abgedeckt werden kann.
Betrachte dazu die Punkte $(2,2)$, $(1,1)$, $(3,1)$, $(2,0)$ an der Tafel.

$\mathrm{VCdim}(\mathcal{APR}_2) < 5$ bedeutet: Es gibt keine fünfelementige Teilmenge des $\mathbb{R}^2$, von der jede Teilmenge durch achsenparallele Rechtecke abgedeckt werden kann.

Diskutiere anhand des bisherigen Tafelbildes:

der fünfte Punkt liegt innerhalb der konvexen Hülle der anderen vier oder “auf dem Rand”.

Genauer: Betrachte APR, das fünf Punkte umschließt, Dieses wird bestimmt durch höchstens vier dieser Punkte. Ein Punkt lässt sich also von diesen höchstens vieren nicht trennen.

Lemma: (ohne Beweis) $\mathrm{VCdim}(k - \mathrm{DNF}_n) = \Theta(n^k)$.

VC Dimension und PAC-Lernen

Begriffserweiterung für $\mathcal{H} \subseteq \Sigma^*$:
$\mathrm{VCdim}(\mathcal{H}) : \mathbb{N} \to \mathbb{N}$, $\mathrm{VCdim}(\mathcal{H})(n) = \mathrm{VCdim}(\{H \cap \Sigma^{\leq n} \mid H \in \mathcal{H}\})$.

Mitteilung: Dann gibt es einen Lernalgorithmus für $\mathcal{H}$ mit Stichprobenkomplexität

$$m(\epsilon, \delta, n) = \frac{1}{\epsilon}\left((n+1)\,\mathrm{VCdim}(\mathcal{H})(n)\ln 2 - \ln \delta\right).$$

Dabei wird einfach irgendeine mit der gezogenen Beispielmenge konsistente Hypothese ausgegeben.

Folgerung: Eine Konzeptmenge ist mit polynomieller Stichprobenkomplexität lernbar gdw. ihre VC-Dimension ist polynomiell.

Grenzen der PAC-Lernbarkeit

Das *Konsistenzproblem* zu einer Konzeptklasse $\mathcal{C}$ ist wie folgt definiert:
Gegeben eine Stichprobe

$$(\langle x_1, \ell_1 \rangle, \ldots, \langle x_m, \ell_m \rangle),$$

gibt es ein Konzept $C \in \mathcal{C}$, sodass $C(x_i) = \ell_i$ für alle i?

Die randomisierte Komplexitätsklasse RP wollen wir hier nicht genau einführen.
Wäre RP=NP, so würde das bedeuten, man könnte mit Wahrsch. $> .5$ bestimmen, ob eine bel. Formel erfüllbar ist.
Allgemein wird daher geglaubt, dass RP von NP verschieden ist.

Mitteilung: Falls RP von NP verschieden ist, so gibt ist für $\mathcal{C}$ keinen polynomiellen strengen PAC-Lerner, sobald das Konsistenzproblem für $\mathcal{C}$ NP-hart ist.

Grenzen strenger PAC-Lernbarkeit

Definition [k-Term-DNF] Eine Formel $T_1 \vee \ldots \vee T_k$, bei der jeder Term T_i eine Konjunktion über beliebig vielen (aber natürlich höchstens n, der Anzahl Boolescher Variablen) Literalen ist, ist in k-*Term disjunktiver Normalform* (k-Term-DNF).

Mitteilung: Das Konsistenzproblem für k-Term-DNF ist NP-hart;
somit ist diese Klasse nicht polynomiell streng PAC-lernbar, sofern RP von NP verschieden ist.

ABER: Zu jeder Formel in k-Term DNF gibt es (durch Ausmultiplizieren) eine äquivalente in k-KNF.
Daher gibt es einen polynomiellen PAC-Lerner für k-Term-DNF.

Grenzen “schwacher” PAC-Lernbarkeit

Mitteilung: Es ist möglich, für einen (hier nicht näher angegebenen) abgeschwächten PAC-Lernbarkeitsbegriff eine Konzeptklasse zu konstruieren, aus deren PAC-Lernbarkeit folgen würde, dass das RSA-Verschlüsselungssystem gebrochen werden könnnte.
Genauer könnte man dann das erste Bit eines Textes korrekt klassifizieren, und damit eben (gemäß bekannter Resultate zur RSA-Verschlüsselung) den gesamten Text.

Genaueres findet man im angeführten Buch von Fischer.

PAC-artiges Lernen regulärer Sprachen

Es gebe eine beliebige aber fixierte Wahrscheinlichkeitsverteilung $P(\cdot)$ auf der Menge aller Wörter über dem Alphabet Σ. Dem Lerner sei diese Verteilung unbekannt.

$R \subseteq \Sigma^*$ sei die zu lernende Sprache. Der Lerner kann Informationen durch Aufruf zweier Orakel gewinnen:

- IN(x) liefert 1 gdw. $x \in R$ (sonst 0);
- EX liefert gemäß der Wahrscheinlichkeitsverteilung P ein Paar $(x, d) \in \Sigma^* \times \{0, 1\}$ mit $x \in R$ gdw. $d = 1$.
  Aufeinanderfolgende Aufrufe von EX seien statistisch unabhängig.

Der (zu konstruierende) Lernalgorithmus L_a^* bekommt bei seinem Aufruf zwei Parameter übergeben, die *Genauigkeit* ϵ und die *Unzuverlässigkeit* δ (beide in $(0,1)$ gelegen).

Ziel ist es, eine *ϵ-Näherung* von R zu inferieren, d.i., einen Automaten A zu finden, so dass die Wahrscheinlichkeit für das Ereignis "symmetrische Differenz von R und $L(A)$" höchstens gleich ϵ ist, also

$$\sum_{x \in R \triangle L(A)} P(x) \leq \epsilon$$

gilt.

Ist A eine ϵ-Näherung von R, so ist die Wahrscheinlichkeit dafür, durch einen Aufruf von EX einen Unterschied von R und $L(A)$ zu finden, höchstens gleich ϵ.

Frage: Wieviele Iterationen eines noch zu beschreibenen Grundalgorithmus sind nötig, um die Genauigkeit ϵ mit Wahrsch. mind. $(1-\delta)$ zu erzielen?

L_a^*: eine Modifikation von L^*.

Eine Elementanfrage wird durch einen Aufruf von IN simuliert.
Jede Hypothese wird durch eine Anzahl von Aufrufen von EX getestet, was in diesem stochastischen Modell die Äquivalenzanfragen an den Lehrer ersetzt.
Liefert nämlich irgendein Aufruf von EX ein Paar (x, d), so dass $d = 1$, aber $M(S, E, T)$ lehnt x ab (oder umgekehrt), so fungiert x als ein Gegenbeispiel für die Hypothese $M(S, E, T)$ von L_a^*, und L_a^* modifiziert seine Hypothese wie vordem L^*.
Liefert keiner der Aufrufe von EX ein Gegenbeispiel, so hält L_a^* mit der Ausgabe $M(S, E, T)$.

Frage: Wieviele Aufrufe —in Abhängigkeit von δ und ϵ— muss L_a^* machen, um eine Hypothese “genügend” zu testen?

Wir lassen eine Abhängigkeit von der Zahl der bisher getesteten Hypothesen zu. Sei

$$r_i = \frac{1}{\epsilon}\left(\ln\frac{1}{\delta} + (\ln 2)(i+1)\right).$$

Sind bislang i Hypothesen getestet worden, so macht L_a^* $\lceil r_i \rceil$ Aufrufe von EX.

Satz: Es sei n die Anzahl der Zustände des minimalen DEAs für die unbekannte, zu lernende reguläre Sprache $R \subseteq \Sigma^*$.
Dann terminiert L_a^* nach

$$\mathcal{O}\left(n + \frac{1}{\epsilon}\left(n \ln \frac{1}{\delta} + n^2\right)\right)$$

vielen Aufrufen des EX Orakels. Die Wahrscheinlichkeit dafür, dass der dabei von L_a^* ausgegebene Automat eine ϵ-Näherung von R ist, beträgt mindestens $1 - \delta$.

Die Laufzeit von L_a^* ist polynomiell in n und der Länge der durch die EX-Aufrufe erhaltenen Beispiele.

Beweis: Wie wir schon bei der Analyse des Algorithmus L^* gesehen hatten, werden höchstens $n-1$ Gegenbeispiele verarbeitet, bis der Algorithmus terminiert. Daher beträgt die Zahl der Aufrufe von EX bis zur Terminierung höchstens:

$$\begin{aligned}\sum_{i=1}^{n-2}(r_i+1) &= (n-1)+\frac{1}{\epsilon}\left((n-1)\ln\frac{1}{\delta}+(\ln 2)\sum_{i=0}^{n-2}(i+1)\right)\\ &\in \mathcal{O}\left(n+\frac{1}{\epsilon}\left(n\ln\frac{1}{\delta}+n^2\right)\right).\end{aligned}$$

Wie groß ist die Wahrscheinlichkeit dafür, dass L_a^* mit einer Hypothese terminiert, die keine ϵ-Näherung von R darstellt, nachdem bislang i Hypothesen getestet wurden?
Diese beträgt (sozusagen in "Runde i") höchstens $(1-\epsilon)^{r_i}$.
Daher ist die Wahrscheinlichkeit dafür, dass der dabei von L_a^* ausgegebene Automat keine ϵ-Näherung von R ist, höchstens:

$$\begin{aligned}\sum_{i=0}^{n-2}(1-\epsilon)^{r_i} &\leq \sum_{i=0}^{n-2}e^{-\epsilon r_i}\\ &\leq \sum_{i=0}^{n-2}\frac{\delta}{2^{i+1}}\leq\delta.\end{aligned}$$