29 PAC-Lernen

29.1 Einleitung

Es gibt eine Vielzahl von formalen und computationellen Lerntheorien; die einzige, die (meines Wissens) wirklich in der Praxis relevant geworden ist, ist das PAC-Lernen, weil man darin auch nach endlich vielen Schritten starke Aussagen über den Lernerfolg machen kann.

Das Problem bei der Induktion von einer gewissen Menge von Beobachtungen ist, dass immer eine Ungewissheit bleibt: ist unsere Generalisierung richtig? Hier sorgen viele Faktoren für Ungewissheit:

- Vielleicht ist die korrekte Hypothese gar nicht im Hypothesenraum;
- vielleicht ist sie darin, aber wir (unser Algorithmus) hat nicht die plausibelste Hypothese (gegeben die Datenlage) ausgewählt, weil er unsere Herangehensweise nicht optimal ist;
- oder aber: wir haben alles bestmöglich gemacht, aber wir hatten einfach Pech mit unseren Beobachtungen: anstatt normaler, repräsentativer Ereignisse haben wir unwahrscheinliche, irreführende Beobachtungen gemacht.

PAC-Lernen konzentriert sich insbesondere auf den letzten Punkt. Das entscheidende ist: natürlich kann es immer sein, dass unsere Beobachtungen nicht repräsentativ sind, aber mit zunehmender Größe unseres Datensatzes wird das immer unwahrscheinlicher.

PAC steht für *probably approximately correct*, und intuitiv gesagt bedeutet PAC-Lernen: wir lernen auf eine Art und Weise, dass es immer unwahrscheinlicher wird, dass wir unsere Hypothese mehr als eine beliebig kleine Distanz von der korrekten Hypothese entfernt ist. Das bedeutet umgekehrt: eine Hypothese, die ernsthaft falsch ist, wird fast mit Sicherheit als falsch erkannt; wenn wir eine Hypothese für richtig halten, dann ist sie mit großer Wahrscheinlichkeit sehr nahe an der korrekten Zielhypothese. Um so etwas sagen zu können, brauchen wir allerdings die passenden Rahmenbedingungen.

29.2 Definitionen

Zunächst müssen wir eine Reihe von Annahmen. Die erste ist die Annahme der **Stationarität**:

> Alle relevanten Beobachtungen, die wir machen, werden von derselben Wahrscheinlichhkeitsverteilung generiert.

Das ist eine sehr wichtige Annahme, und in gewissem Sinn die Voraussetzung induktiven Lernens: wenn die Verteilung im Laufe der Zeit sich (beliebig) ändert, dann erlauben uns die Beobachtungen, die wir gemacht haben, keinerlei Rückschlüsse auf zukünftige Beobachtungen, und jede Form von Induktion ist unmöglich.

Weiterhin haben wir folgendes;

- M ist die Menge aller möglichen Beobachtungen (korrekte Klassifikationen etc., üblicherweise bekannt)
- P ist eine Wahrscheinlichkeitsverteilung über M, die uns sagt wie wahrscheinlich eine Beobachtung ist (üblicherweise unbekannt)
- f ist die Zielfunktion, die wir lernen möchten (unbekannt; wir nehmen wieder an, wir lernen eine Funktion)
- H ist die Menge der Hypothesen, die uns zur Verfügung stehen (bekannt)
- $N = |D|$ ist die Anzahl der Beobachtungen, anhand derer wir unsere Hypothese $h \in H$ auswählen (bekannt bzw. variabel)

Wenn nun f die Zielfunktion ist, h eine Hypothese, dann können wir die Fehlerhaftigkeit von h genau quantifizieren (zumindest abstrakt; konkret kennen wir natürlich die Zahlen nicht):

(418) $error(h) = P(h(x) \neq f(x)|x \in M)$

das bedeutet, etwas genauer,

(419) $error(h) = P(X) : X = \{x \in M : h(x) \neq f(x))$

Natürlich müssen wir voraussetzen können, dass diese Wahrscheinlichkeiten definiert sind.

Wir sagen dass h **annähernd korrekt** ist, falls $error(h) \leq \epsilon$, wobei ϵ eine beliebig kleine Konstante ist.

(ϵ müssen wir natürlich festlegen). Beachten Sie aber dass hierbei 2 unbekannte auftauchen:

1. f, die Zielhypothese die wir nicht kennen, und
2. P, die Verteilung über den Daten, die wir nicht kennen.

Das eigentlich geniale am PAC-Lernen ist dass wir so arbeiten, dass sich die unbekannten "rauskürzen".

Zunächst unterscheiden wir zwei Arten von Hypothesen, nämlich solche, die **ernsthaft falsch**, und solche, die **annähernd korrekt** sind, auf in

$$H_{\downarrow} = \{h : error(h) > \epsilon\} \tag{420}$$

$$H_{\uparrow} = \{error(h) \leq \epsilon\} \tag{421}$$

Wir können uns $H_{\uparrow}$ vorstellen als eine Kugel, die einen gewissen Radius um die korrekte Hypothese hat.

Nun nehmen wir eine Hypothese h, die wir erstellt haben. Nach unserer Konstruktion gilt: h ist konsistent mit den N Beobachtungen, die wir gemacht haben. Uns interessiert die Wahrscheinlichkeit

$$P(h \in H_{\downarrow}),$$

also die Wahrscheinlichkeit, dass unsere Hypothese "ernsthaft falsch" ist.

- ▷ Nun können wir sagen, dass die Wahrscheinlichkeit, dass unsere Hypothese falsch ist, und dennoch ein Beispiel richtig klassifiziert, allerhöchstens $1 - \epsilon$ ist –
- ▷ denn wir haben eine Wahrscheinlichkeitsmasse von $\geq \epsilon$ auf die falsch klassifizierten Beispiele gesetzt:

$$P(h(x) = f(x) | h(x) \in H_{\downarrow}) \leq 1 - \epsilon \tag{422}$$

Diese Tatsache allein scheint nicht sonderlich interessant, denn ϵ ist üblicherweise ziemlich klein, also ist $1 - \epsilon \approx 1$. Wir haben aber

$$\epsilon > 0,$$

und deswegen gilt: für alle $\delta > 0$ gibt es ein $n \in \mathbb{N}$, so dass

(423) $(1-\epsilon)^n < \delta$

Diese Beobachtung ist entscheidend, denn wir haben:

(424) $P(h(x) = f(x) : \forall x \in D | h \in H_{\downarrow}) \leq (1-\epsilon)^N$

Denn wir gehen davon aus, dass alle unsere Beispiele in D korrekt sind; es gibt also keine Störungen in unseren Daten. Dementsprechend sinkt die Wahrscheinlichkeit, dass wir eine Hypothese in $H_{\downarrow}$ haben, die konsistent mit unseren Daten ist, mit der Zahl der Beobachtungen die wir machen. Als nächstes sehen wir, dass wir die Wahrscheinlichkeit beachten müssen, dass *irgendeine* Hypothese in $H_{\downarrow}$ konsistent ist mit unseren Daten. Das ist natürlich

(425) $P(\exists h \in H_{\downarrow}.\forall x \in D : h(x) = f(x)) \leq |H_{\downarrow}|(1-\epsilon)^N \leq |H|(1-\epsilon)^N$

Das setzt natürlich voraus, dass $|H|$ endlich ist, sonst ist der Term undefiniert. Wenn wir also

- ein beliebes ϵ auswählen, dass eine Abweichung als "ernsthaft falsch" definiert,
- ein beliebiges δ, dass die maximale Wahrscheinlichkeit festlegt, dass die Hypothese ernsthaft falsch ist,
- dann müssen wir nur ein N finden so dass

(426) $|H|(1-\epsilon)^N \leq \delta$

Um diesen Term nach N aufzulösen, muss man etwas tricksen. Man kann zeigen dass

(427) $1 - \epsilon \leq e^{-\epsilon} = \frac{1}{e^{\epsilon}}$

(denn je kleiner ϵ, desto kleiner e^{ϵ}, desto größer $1/e^{\epsilon}$). Also reicht es, N so zu wählen dass

$$
\begin{aligned}
(428) \qquad |H|(e^{-\epsilon})^N &\leq \delta \\
(429) \qquad \Leftrightarrow ln(|H|(e^{-\epsilon})^N) &\leq ln(\delta) \\
(430) \qquad \Leftrightarrow ln(|H|) + ln(e^{-\epsilon}) \cdot N &\leq ln(\delta) \\
(431) \qquad \Leftrightarrow ln(|H|) - \epsilon \cdot N &\leq ln(\delta) \\
(432) \qquad \Leftrightarrow -ln(|H|) + \epsilon \cdot N &\geq ln(\frac{1}{\delta}) \\
(433) \qquad \Leftrightarrow \epsilon \cdot N &\geq ln(\frac{1}{\delta}) + ln(|H|) \\
(434) \qquad \Leftrightarrow N &\geq \frac{1}{\epsilon} \cdot (ln(\frac{1}{\delta}) + ln(|H|))
\end{aligned}
$$

Das bedeutet: wenn wir N entsprechend wählen, dann gilt für eine Hypothese h, die mit N Beispielen konsistent ist:

> Mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ hat h eine Fehlerrate von höchstens ϵ.

Mit anderen Worten: sie ist wahrscheinlich annähernd korrekt. Diese Nummer N – gegeben ϵ und δ – nennt man die Stichprobenkomplexität des Hypothesenraumes H (denn sie hängt natürlich von H ab).

Betrachten wir das Kriterium

$$(435) \quad N \geq \frac{1}{\epsilon} \cdot (ln(\frac{1}{\delta}) + ln(|H|))$$

Dann fällt uns auf:

- δ (unsere verbleibende Unsicherheit) spielt logarithmisch eine Rolle (also mit schrumpfenden δ wächst N eher langsam);
- $|H|$ – unser Hypothesenraum – spielt ebenfalls logarithmisch eine Rolle (also mit wachsendem $|H|$ wächst N eher langsam);
- ϵ ist ein linearer Faktor, wenn wir ϵ verringern, wächst N proportional.

PAC-Lernen ist insofern sehr vorteilhaft, als dass wir nach einer endlichen Anzahl von Datenpunkten starke Aussagen über die Qualität unserer Hypothesen machen können. Man beachte insbesondere, dass P hierbei keine

Rolle spielt, PAC-Lernen ist also unabhängig von der zugrundeliegenden Verteilung! Auf der anderen Seite haben wir aber eine Vielzahl von Voraussetzungen, die oft nicht erfüllt sind.

Als Summe und Zusammenfassung kann man aber festhalten: überall wo PAC-Lernen möglich ist, da kann man äußerst zufriedenstellende Ergebnisse erzielen. Das Problem ist – dass wir meistens nicht die Voraussetzungen erfüllen.

29.3 PAC-lernbare Probleme I

Wir wenden uns jetzt einigen einfachen Beispielen zu, in denen man PAC-Lernen verwenden kann. Wir nehmen einmal an, wir haben folgendes Szenario. Wir suchen eine Zahl $x \in \mathbb{R}$ "lernen", so dass gilt:

falls $y \geq x$, dann hat y label 1; ansonsten hat y label 0.

Auch das ist natürlich eine Funktion $f : \mathbb{R} \to \{0, 1\}$, definiert durch

$$f_x(y) = \begin{cases} 1, \text{ falls } y \geq x \\ 0 \text{ andernfalls} \end{cases} \tag{436}$$

Wir können z.B. y als eine Temperatur interpretieren, und den Wert $f(y)$ als eine Beobachtung, z.B. die Apfelblüte. Dementsprechend haben wir

- eine unbekannte Zielfunktion f_x
- einen Hypothesenraum $H = \{f_x : x \in \mathbb{R}\}$
- einen Datensatz $D \subseteq \mathbb{R} \times \{0, 1\}$

Was uns fehlt ist eine Wahrscheinlichkeitsverteilung $P : \mathbb{R} \to [0, 1]$ – wie wir gesehen haben, ist PAC-Lernen letzten Endes unabhängig von dieser Verteilung. ; Wir müssen also nur annehmen, dass es eine solche Verteilung gibt.

Der Algorithmus Der Algorithmus wird mit immer neuen Daten konfrontiert. Wir haben nach jedem Datensatz $D = \{(y_1, f(y_1), ..., (y_i, f(y_i))\}$ den wir beobachten zwei wichtige Beispiele:

$$\underline{x} = max\{x : (x, 0) \in D\} \tag{437}$$

$$\overline{x} = min\{x : (x, 1) \in D\} \tag{438}$$

(439)

Wir wissen dass $\underline{x} < \overline{x}$, weil sonst die Beobachtung mit unseren Wissen inkonsistent ist.

Der Algorithmus kann nun ein beliebiges $x \in (\underline{x}, \overline{x})$ wählen – das Problem ist PAC-lernbar, d.h. die Wahrscheinlichkeit, von der richtigen Hypothese signifikant entfernt zu liegen, konvergiert gegen 0.

29.4 PAC-lernbare Probleme II

Eben haben wir eine einzelne Zahl gesucht, die den reellen Zahlenstrang in zwei Teile spaltet. Wir betrachtet nun ein etwas komplexeres Problem, nämlich die Suche nach einem Intervall $[x_1, x_2]$:

Falls $y \in [x_1, x_2]$, dann hat y label 1; ansonsten hat y label 0.

Auch das ist natürlich eine Funktion $f : \mathbb{R} \to \{0, 1\}$, definiert durch

$$f_{x_1,x_2}(y) = \begin{cases} 1, \text{ falls } y \in [x_1, x_2] \\ 0 \text{ andernfalls} \end{cases} \tag{440}$$

Wir können z.B. y als eine Temperatur interpretieren, und den Wert $f(y)$ als eine Beobachtung, z.B. die Apfelblüte. Dementsprechend haben wir

- eine unbekannte Zielfunktion f_x
- einen Hypothesenraum $H = \{f_{x_1,x_2} : x_1, x_2 \in \mathbb{R}\}$
- einen Datensatz $D \subseteq \mathbb{R} \times \{0, 1\}$

Die Wahrscheinlichkeitsverteilung $P : \mathbb{R} \to [0, 1]$ wird wieder beliebig gewählt und spielt letzten Endes keine Rolle.

Der Algorithmus Der Algorithmus wird mit immer neuen Daten konfrontiert. Wir haben nach jedem Datensatz $D = \{(y_1, f(y_1), ..., (y_i, f(y_i))\}$, den wir beobachten, zwei wichtige Beispiele:

$$\underline{x} = max\{x : (x, 1) \in D\} \tag{441}$$

$$\overline{x} = min\{x : (x, 1) \in D\} \tag{442}$$

(443)

Die Strategie, die wir wählen, ist nun vergleichbar. Auch das ist PAC-Lernbar. Der Algorithmus kann nun ein beliebiges $x \in (\underline{x}, \overline{x})$ wählen – das Problem ist PAC-lernbar, d.h. die Wahrscheinlichkeit, von der richtigen Hypothese signifikant entfernt zu liegen, konvergiert gegen 0.

29.5 PAC-lernbare Probleme III

Es gibt auch Sprachen die sind PAC-lernbar. Nimm einen deterministischen endlichen Automaten $(Q, \delta, \Sigma, q_0, F)$, mit $\delta : Q \times \Sigma \to Q$ einer partiellen Übergangsfunktion. Wir definieren deren Inversion $\delta : Q \times \Sigma \to Q$

(444) $\delta^{-1}(q, a) = \{q' : \delta(q', a) = q\}$

Wir sagen, $(Q, \delta, \Sigma, q_0, F)$ ist **reversibel** (reversible), wenn gilt: die Inversion δ^{-1} kann als eine partielle Funktion aufgefasst werden, also:

(445) f.a. $q \in Q : |\delta^{-1}(q)| \leq 1$

Anders gesagt:

> wenn wir alle Übergänge umdrehen sowie Start und Endzustände vertauschen, dann ist der resultierende Automat wiederum deterministisch.

Eine reguläre Sprache ist **reversibel**, wenn es einen reversiblen Automaten gibt, der sie erkennt.

Beispiel 1 $(ab)^*$ ist reversibel: ein einfacher Automat mit 2 Zuständen.

Beispiel 2 $(aa)^*$ ist reversibel.

Beispiel 3 $b^*(ab^*ab*)^*$ ist reversibel.

Beispiel 4 aa^* ist **nicht** reversibel. Warum? Weil wir immer in einem Endzustand landen q_f müssen mit $\delta(q_f, a) = q_f$. Allerdings müssen wir ja in diesen Zustand reingekommen sein: $\delta(q_0, a) = q_f$ (aber $\delta(q_f, a) \neq q_0$). Das Argument ist etwas komplex auszubuchstabieren, sollte aber klar sein.

Beispiel 5 $b(ab)^*$ ist nicht reversibel – siehe das Argument oben!

Beispiel 6 Jede endliche Sprache ist reversibel, denn ein Automat ohne Schleife ist immer reversibel. Man kann ihn einfach als einen Baum schreiben!

Beispiel 7 $(aa)^* \cup (ab)^*$ ist reversibel.

Beispiel 8 $(abc)^* \cup (bac)^*$ ist **nicht** reversibel: a bringt uns in einen anderen Zustand als b; aber wenn ein c kommt, werden sie wieder äquivalent!

Wir sehen daran, dass die reversiblen Sprachen einigermassen mächtig sind. Ein wichtiges Theorem der Lerntheorie besagt nun:

R-Lernbarkeit Die reversiblen Sprachen sind PAC-lernbar.

Das bedeutet: wenn wir immer neue Worte sehen, die mit unserer Hypothese konsistent sind, dann geht die Wahrscheinlichkeit, dass wir die falsche Kandidatensprache haben, gegen 0. Da PAC-Lernbarkeit sehr mächtig ist insbesondere in Hinblick auf viele Anwendungen, ist das ein sehr wichtiges Ergebnis!

(Den Beweis schauen wir uns aber nicht an, der ist lang und kompliziert).