AGML, Sommersemester 2020 Vorlesung 1 (4 Seiten)

PAC Learning

Thomas Kesselheim Letzte Aktualisierung: 22. April 2020

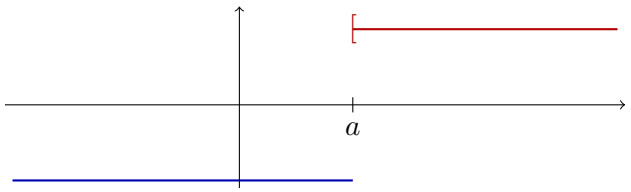
Zum Einstieg betrachten wir *binäre Klassifikation*. Das heißt, wir müssen Datenpunkte klassifizieren nach „positiv“ oder „negativ“. Einige korrekt klassifizierte Punkte sind uns gegeben. Es könnte sich also beispielsweise um E-Mails handeln, bei denen wir automatisch „Spam“ und „Nicht-Spam“ unterscheiden wollen. Diese Beschriftung ist ein *Label*.

1 Schwellenwertfunktionen

Unser erstes Beispiel nimmt stark vereinfachend an, dass jeder Datenpunkt nur ein einziges Merkmal $x \in X := \mathbb{R}$ hat, das eine reelle Zahl ist. Die jeweilige Ausprägung des Merkmals charakterisiert einen Datenpunkt perfekt: Wann immer $x \geq a$ ist, handelt es sich um einen positiv zu klassifizierenden Punkt, ansonsten um einen, der negativ zu klassifizieren ist.

Das heißt, wir können eine Funktion $f\colon X \to \{-1, +1\}$ angeben, die die korrekten Labels beschreibt. Sie lautet

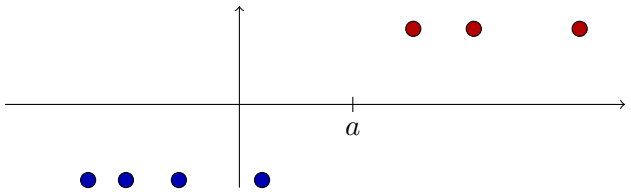
$$f(x) = \begin{cases} +1 & \text{falls } x \geq a \\ -1 & \text{sonst} \end{cases}$$



Diese Funktion nennen wir *Grundwahrheit* (*ground truth*).

Obwohl die Struktur der korrekten Labels sehr einfach ist, ist die Aufgabe nicht trivial. Wir kennen nämlich a nicht. Uns werden lediglich m Datenpunkte mit korrekten Labels gegeben. Die zentrale Frage ist: Wie groß muss m sein, damit wir neue Datenpunkte einigermaßen zuverlässig klassifizieren können?

Ein Beispiel mit $m = 7$ könnte also so aussehen:



2 Hypothesen und Fehler

Konkreter nehmen wir an, dass die Datenpunkte x aus irgendeiner Wahrscheinlichkeitsverteilung $\mathcal{D}$ gezogen werden. Unseren *Hypothesenraum* bezeichnen wir mit $\mathcal{H}$. In diesem Fall ist $\mathcal{H}$ die Menge aller Funktionen der Form $h_{a'}\colon \mathbb{R} \to \{-1, +1\}$ mit

$$h_{a'}(x) = \begin{cases} +1 & \text{falls } x \geq a' \\ -1 & \text{sonst} \end{cases}$$

Unser Ziel ist es, eine Hypothese h mit möglichst kleinem Fehler $\mathrm{err}_{\mathcal{D},f}(h)$ zu finden. Dieser ist wie folgt definiert.

Definition 1.1. *Der* tatsächliche Fehler *(oder* tatsächliches Risiko*)* $\mathrm{err}_{\mathcal{D},f}(h)$ *einer Hypothese* h *hinsichtlich einer Wahrscheinlichkeitsverteilung* $\mathcal{D}$ *über Datenpunkte und Grundwahrheit* f *ist*

$$\mathrm{err}_{\mathcal{D},f}(h) := \mathbf{Pr}_{x\sim\mathcal{D}}\left[h(x) \neq f(x)\right] \ .$$

Beispiel 1.2. *Sei* $\mathcal{D}$ *die uniforme Verteilung auf* $[0,1]$*. Der tatsächliche Fehler einer Hypothese* $h_{a'}$ *ist* $\mathrm{err}_{\mathcal{D},f}(h_{a'}) = |a - a'|$*, wenn* $a, a', \in [0,1]$*. Da* a *jedoch im Allgemeinen nicht bekannt ist, kann dieser jedoch von einem Lernalgorithmus nicht berechnet werden.*

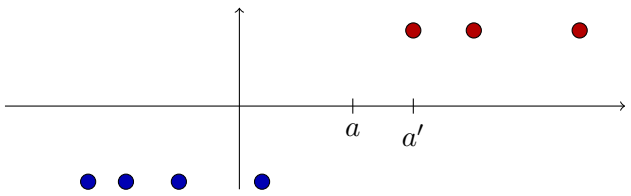
In diesem Beispiel und auch im Folgenden vereinfachen wir uns das Leben durch die Annahme, dass die Grundwahrheit *realisierbar* ist. Das heißt, dass $f \in \mathcal{H}$. Dies ist in unserem Beispiel natürlich erfüllt. In der Realität hingegen sind die Hypothesenklassen meist nicht mächtig genug, um alle Datenpunkt richtig zu klassifizieren.

3 Lernen mit Samples

Wie finden wir also eine Hypothese h, die den tatsächlichen Fehler $\mathrm{err}_{\mathcal{D},f}(h)$ möglichst klein hält? Wir nehmen an, dass uns m Datenpunkte mit korrekten Labels gegeben sind. Seien also $x_1, \ldots, x_m$ Datenpunkten, die unabhängig und identisch verteilt aus $\mathcal{D}$ gezogen sind. Außerdem seien $y_1 = f(x_1), \ldots, y_m = f(x_m)$ die zugehörigen korrekten Labels. Die Menge aller Samples bezeichnen wir mit $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$.

Ein einfacher Lernalgorithmus wählt nun das größte a', sodass $h_{a'}$ die Menge S korrekt klassifiziert. Das heißt, wir setzen a' auf den Wert des kleinsten x_i mit $y_i = 1$, wenn es ein solches gibt. Anderenfalls $a' = \infty$.

In unserem Beispiel sieht das so aus:

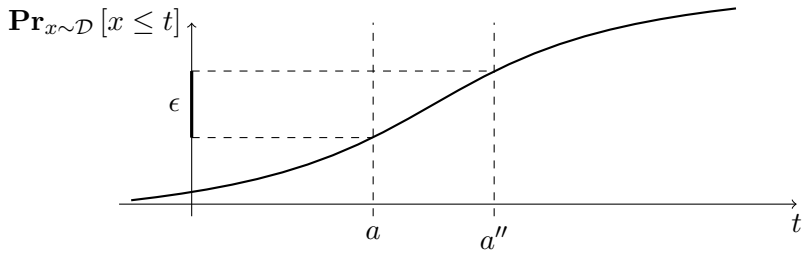


Satz 1.3. *Sei* $h_{a'}$ *die vom einfachen Lernalgorithmus berechnete Hypothese, der als Eingabe ein Sample von* m *Datenpunkten mit korrekten Labels gemäß* f *erhält, die unabhängig und identisch verteilt aus* $\mathcal{D}$ *gezogen werden. Dann gilt für alle* $\epsilon > 0$*, dass* $\mathbf{Pr}\left[\mathrm{err}_{\mathcal{D},f}(h_{a'}) \geq \epsilon\right] \leq \mathrm{e}^{-\epsilon m}$ *.*

Beweis. Weil unser Lernalgorithmus das größte a' wählt, wird auf jeden Fall gelten, dass $a' \geq a$. Falsch klassifiziert werden alle Punkte im Bereich $[a, a')$. Somit gilt nun für jede Verteilung $\mathcal{D}$

$$\mathrm{err}_{\mathcal{D},f}(h_{a'}) = \mathbf{Pr}_{x\sim\mathcal{D}}\left[x \in [a, a')\right] \ .$$

Sei nun a'' die kleinste Zahl, sodass $\mathbf{Pr}_{x\sim\mathcal{D}}\left[x \in [a, a'']\right] \geq \epsilon$. Der Fehler von $h_{a'}$ wird also höchstens ϵ sein, falls $a' \leq a''$. Dies geschieht, wenn es mindestens ein i gibt, sodass $x_i \in [a, a'']$. Sei $\mathcal{E}_i$ das Ereignis, dass $x_i \in [a, a'']$. Es gilt $\mathbf{Pr}\left[\mathcal{E}_i\right] \geq \epsilon$. (Für den Fall einer stetigen Verteilung gilt hier Gleichheit.)



Damit *nicht* $a' \le a''$ gilt, darf keines der Ereignisse $\mathcal{E}_i$ eintreten. Wir interessieren uns also für $\bigcap_i \bar{\mathcal{E}}_i$. Weil $x_1, \dots, x_m$ unabhängige Züge aus $\mathcal{D}$ sind, gilt nun auch

$$\mathbf{Pr}\left[\bigcap_i \bar{\mathcal{E}}_i\right] = \prod_i \mathbf{Pr}\left[\bar{\mathcal{E}}_i\right] \le (1-\epsilon)^m \quad .$$

Wir können nun die Abschätzung $1 - x \le \mathrm{e}^{-x}$ für alle $x \in \mathbb{R}$ verwenden. Somit erhalten wir insgesamt die Behauptung. □

Satz 1.3 sagt uns nun insbesondere, dass wenn wir $m = \frac{1}{\epsilon}\ln\left(\frac{1}{\delta}\right)$ wählen, die Wahrscheinlichkeit, dass der tatsächliche Fehler unserer gefundenen Hypothese größer als ϵ ist, kleiner als δ wird.

4 PAC-Lernbarkeit

Diese Aussage gilt für alle $\epsilon > 0$ und alle $\delta > 0$. Wenn die Anzahl der Samples also nur groß genug ist, werden wir mit großer Wahrscheinlichkeit nur einen sehr kleinen Fehler haben. Die Hypothesenklassen, für die dies gilt, heißen PAC-lernbar.

Definition 1.4. *Eine Hypothesenklasse $\mathcal{H}$ heißt* PAC-lernbar *(im realisierbaren Sinn), wenn es eine Funktion $m_{\mathcal{H}}$ und einen Lernalgorithmus $\mathcal{A}$ gibt, sodass der Algorithmus für alle $\epsilon, \delta > 0$, jede Verteilung $\mathcal{D}$ und alle $f \in \mathcal{H}$, gegeben ein Sample S von Größe mindestens $m_{\mathcal{H}}(\epsilon, \delta)$ von Datenpunkten mit korrekten Labels, eine Hypothese $h_S \in \mathcal{H}$ berechnet, sodass $\mathbf{Pr}\left[\mathrm{err}_{\mathcal{D},f}(h_S) < \epsilon\right] \ge 1 - \delta$.*

Ferner heißt sie effizient PAC-lernbar*, wenn es einen Polynomialzeitalgorithmus $\mathcal{A}$ mit obiger Eigenschaft gibt.*

PAC steht für „probably approximately correct“. „Probably“ bedeutet in diesem Fall, dass die Wahrscheinlichkeit mindestens $1 - \delta$ ist, „approximately correct“ bezieht sich darauf, dass $\mathrm{err}_{\mathcal{D},f}(h_S) < \epsilon$.

Nicht jede Hypothesenklasse ist PAC-lernbar. Zum Beispiel ist die Klasse aller Hypothesen $\mathbb{N} \to \{-1, +1\}$ nicht PAC-lernbar. Dies werden wir im Laufe der Vorlesung beweisen.

5 Weiteres Beispiel: Lernen von Intervallen

Nun betrachten wir als Hypothesenklasse $\mathcal{H}$ die Menge aller Funktionen der Form $h_{a',b'} \colon \mathbb{R} \to \{-1, +1\}$ mit

$$h_{a',b'}(x) = \begin{cases} +1 & \text{falls } x \in [a', b'] \\ -1 & \text{sonst} \end{cases}$$

Wir sind wieder im realisierbaren Fall. Das heißt, es gibt eine Grundwahrheit $f \in \mathcal{H}$, die alle Datenpunkte richtig klassifiziert. Nun also

$$f(x) = \begin{cases} +1 & \text{falls } x \in [a,b] \\ -1 & \text{sonst} \end{cases}$$

Gegeben ist wieder eine Menge $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ von Datenpunkten mit korrekten Labels. Unser Lernalgorithmus wählt das kleinste Intervall $[a', b']$, das S korrekt klassifiziert. Das heißt, wir setzen a' auf den Wert des kleinsten x_i mit $y_i = 1$ und b' auf den Wert des größten x_i mit $y_i = 1$. Den Fall, dass es kein i mit $y_i = 1$ gibt, ignorieren wir.

Satz 1.5. *Für alle $\epsilon > 0$ gilt* $\mathbf{Pr}\left[\mathrm{err}_{\mathcal{D},f}(h_{a',b'}) \geq \epsilon\right] \leq 2\mathrm{e}^{-\frac{\epsilon m}{2}}$.

Beweis. Weil unser Lernalgorithmus das kleinste Intervall wählt, wird auf jeden Fall gelten, dass $a' \geq a$ und $b' \leq b$. Falsch klassifiziert werden alle Punkte im Bereich $[a, a') \cup (b', b]$. Somit gilt nun für jede Verteilung $\mathcal{D}$

$$\mathrm{err}_{\mathcal{D},f}(h_{a',b'}) = \mathbf{Pr}_{x\sim\mathcal{D}}\left[x \in [a,a') \cup (b',b]\right] \ .$$

Sei außerdem ähnlich wie oben a'' die kleinste Zahl, sodass $\mathbf{Pr}_{x\sim\mathcal{D}}\left[x \in [a,a'']\right] \geq \frac{\epsilon}{2}$. Analog sei b'' die größte Zahl, sodass $\mathbf{Pr}_{x\sim\mathcal{D}}\left[x \in [b'',b]\right] \geq \frac{\epsilon}{2}$. Damit $\mathrm{err}_{\mathcal{D},f}(h_{a',b'}) \leq \epsilon$ ist es nun hinreichend, dass $a' \leq a''$ und $b' \geq b''$. Dies geschieht, wenn es je mindestens ein i gibt, sodass $x_i \in [a, a'']$ bzw. $x_i \in [b'', b]$.

Für jedes i gilt

$$\mathbf{Pr}\left[x_i \in [a,a'']\right] \geq \frac{\epsilon}{2} \quad \text{und} \quad \mathbf{Pr}\left[x_i \in [b'',b]\right] \geq \frac{\epsilon}{2} \ .$$

Weil $x_1, \ldots, x_m$ unabhängige Züge aus $\mathcal{D}$ sind, gilt nun auch

$$\mathbf{Pr}\left[x_1, \ldots, x_m \notin [a,a'']\right] \leq \left(1 - \frac{\epsilon}{2}\right)^m \quad \text{und} \quad \mathbf{Pr}\left[x_1, \ldots, x_m \notin [b'',b]\right] \leq \left(1 - \frac{\epsilon}{2}\right)^m \ .$$

Damit gilt auch

$$\mathbf{Pr}\left[x_1, \ldots, x_m \notin [a,a''] \text{ oder } x_1, \ldots, x_m \notin [b'',b]\right] \leq 2\left(1 - \frac{\epsilon}{2}\right)^m \ ,$$

wobei wir die Abschätzung $\mathbf{Pr}\left[\mathcal{E} \cup \mathcal{F}\right] \leq \mathbf{Pr}\left[\mathcal{E}\right] + \mathbf{Pr}\left[\mathcal{F}\right]$ für zwei Ereignisse $\mathcal{E}$ und $\mathcal{F}$ verwendet haben.

Die Behauptung folgt nun wieder mit der Abschätzung $1 - x \leq \mathrm{e}^{-x}$ für alle $x \in \mathbb{R}$. □

Referenzen

- Foundations of Machine Learning, Kapitel 2.1
- Siehe auch die Vorlesungsskripte von Anna Karlin `https://courses.cs.washington.edu/courses/cse522/17sp/` und Avrim Blum `http://www.cs.cmu.edu/~avrim/ML14/`. Diese enthalten weitere Referenzen.