Theorie des maschinellen Lernens

Hans U. Simon

25. April 2017

3 Ein formales Lernmodell

Das Basismodell in Abschnitt 3.1 beschäftigt sich mit dem PAC-Lernen binärer Hypothesen unter der Realisierbarkeitsannahme (Realizability Assumption). Abschnitt 3.2 behandelt Verallgemeinerungen des Basismodells. Abschnitt 3.3 erweitert das PAC-Lernmodell um Effizienzkriterien. Zu diesem Zweck werden parametrisierte Lernprobleme eingeführt. Ein wichtiges Markenzeichen des im Folgenden vorgestellten Lernmodells ist seine „Verteilungsunabhängigkeit“. Darauf gehen wir in Abschnitt 3.4 kurz ein.

3.1 PAC-Lernen unter der Realisierbarkeitsannahme

Wir verwenden die Standardnotationen $\mathcal{X}, \mathcal{Y}, Z, \mathcal{D}, f$, die aus dem vorangegangenen Kapitel bekannt sind. In diesem Abschnitt beschäftigen wir uns mit binären Klassifikationsproblemen und setzen daher $Z = \mathcal{X} \times \mathcal{Y}$ und $\mathcal{Y} = \{0,1\}$. $f : \mathcal{X} \to \{0,1\}$ ist dann die Funktion, welche für die Markierung der Beispielinstanzen aus $\mathcal{X}$ zuständig ist. Es sei an die Definition

$$L_{\mathcal{D},f}(h) = \Pr_{x \sim \mathcal{D}}[h(x) \neq f(x)]$$

erinnert, welche die erwartete Fehlklassifikationsrate (kurz: Fehlerrate) einer Hypothese $h : \mathcal{X} \to \{0,1\}$ festlegt. Ebenso erinnern wir an die Realisierbarkeitsannahme für binäre Hypothesenklassen, die von $(\mathcal{H}, \mathcal{D}, f)$ erfüllt wird, wenn $\mathcal{H}$ eine Hypothese h mit $L_{\mathcal{D},f}(h) = 0$ enthält. Eine bezüglich $(\mathcal{D}, f)$ zufällige Trainingsmenge S der Größe m hat dann die Form $S = ((x_1, y_1), \ldots, (x_m, y_m))$ mit $S|_{\mathcal{X}} \sim \mathcal{D}^m$ und $y_i = f(x_i)$ für $i = 1, \ldots, m$. Folgende Definition legt ein erstes Basismodell des PAC-Lernens fest:

Definition 3.1 (PAC-Lernen unter der Realisierbarkeitsannahme) *Eine binäre Hypothesenklasse $\mathcal{H}$ heißt* PAC-lernbar unter der Realisierbarkeitsannahme*, wenn eine Funktion $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ und ein Lernalgorithmus A existieren, so dass für alle $0 < \varepsilon, \delta < 1$ sowie für jede Wahl von $\mathcal{D}$ und f, welche die Realisierbarkeitsannahme nicht verletzt, folgende Bedingung erfüllt ist: A, gestartet auf einer bezüglich $(\mathcal{D}, f)$ zufälligen Trainingsmenge S der Größe $m \geq m(\varepsilon, \delta)$, ermittelt eine Hypothese $h : \mathcal{X} \to \{0,1\}$ und es gilt*

$$\Pr_{S|_{\mathcal{X}} \sim D^m}[L_{\mathcal{D},f}(h) \leq \varepsilon] \geq 1 - \delta \ . \tag{1}$$

In Worten: mit einer Wahrscheinlichkeit von mindestens $1-\delta$ wird eine zufällige Trainingsmenge generiert, aus der A eine Hypothese mit einer Fehlerrate von maximal ε extrahiert.

Im Folgenden bezeichnen wir mit $m_{\mathcal{H}}$ die Funktion $m_{\mathcal{H}}$, die der Definition 3.1 genügt und unter dieser Nebenbedingung die kleinstmöglichen Werte $m_{\mathcal{H}}(\varepsilon, \delta)$ hat. Sie wird als *Stichprobenkomplexität (sample complexity)* oder auch *Informationskomplexität* von $\mathcal{H}$ bezeichnet. Für endliche binäre Hypothesenklassen $\mathcal{H}$ gilt (wie wir aus dem vorangegangenen Kapitel wissen):

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq \left\lceil \frac{\ln(|\mathcal{H}|/\delta)}{\varepsilon} \right\rceil \ .$$

3.2 Verallgemeinerungen des Basismodells

Die Realisierbarkeitsannahme ist bei Anwendungen fast nie erfüllt und daher unrealistisch. Realistischer ist es anzunehmen, dass markierte Beispiele (x, y) bezüglich einer (dem Lerner unbekannten) Verteilung $\mathcal{D}$ auf $\mathcal{X} \times \mathcal{Y}$ generiert werden. Wir setzen dann

$$L_{\mathcal{D}}(h) = \Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y] \ .$$

Für eine gegebene Instanz x werden i.A. sogar beide Klassifikationslabels mit einer echt positiven Wahrscheinlichkeit vorkommen. Anstatt eine Fehlerrate nahe bei Null anzustreben, müssen wir uns dann mit einer Fehlerrate begnügen, die nah an der besten mit Hypothesen aus $\mathcal{H}$ erzielbaren Fehlerrate liegt. Diese Überlegungen führen zu folgender

Definition 3.2 (Agnostisches PAC-Lernen binärer Hypothesen) *Eine binäre Hypothesenklasse $\mathcal{H}$ heißt* agnostisch PAC-lernbar*, wenn eine Funktion $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ und ein Lernalgorithmus A existieren, so dass für alle $0 < \varepsilon, \delta < 1$ sowie für jede Verteilung $\mathcal{D}$ auf $\mathcal{X} \times \mathcal{Y}$ folgende Bedingung erfüllt ist: A, gestartet auf einer zufälligen Trainingsmenge $S \sim D^m$ mit $m \geq m(\varepsilon, \delta)$, ermittelt eine Hypothese $h : \mathcal{X} \to \{0, 1\}$ und es gilt*

$$\Pr_{S\sim D^m}[L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \varepsilon] \geq 1 - \delta \ . \tag{2}$$

In Worten: mit einer Wahrscheinlichkeit von mindestens $1 - \delta$ wird eine zufällige Trainingsmenge generiert, aus der A eine Hypothese extrahiert, deren Fehlerrate die bezüglich $\mathcal{H}$ bestmögliche Fehlerrrate um maximal ε überschreitet.

Unter der Realisierbarkeitsannahme ergibt sich $\min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') = 0$ und das Erfolgskriterium (2) kollabiert zu dem aus Definition 3.1 bekannten Kriterium (1).

Die Definitionen 3.1 und 3.2 sind auf binäre Klassifikationsprobleme zugeschnitten. Implizit haben wir eine sogenannte *Null-Eins-Verlustfunktion* der Form

$$\ell_{0-1}(h, (x, y)) = \begin{cases} 0 & \text{falls } h(x) = y \\ 1 & \text{falls } h(x) \neq y \end{cases} \tag{3}$$

verwendet. Beachte, dass

$$L_{\mathcal{D}}(h) = \Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y] = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell_{0-1}(h,(x,y))] \ ,$$

d.h., die Fehlerrate von h deckt sich mit dem für h zu erwartenden Null-Eins-Verlust.
Es ist wünschenswert eine breitere Palette von Problemen zu erfassen. Zum Beispiel:

Multiklassen-Klassifikationsprobleme Es gibt $k \geq 2$ mögliche Klassifikationslabels. Wir setzen dann $\mathcal{Y} = [k]$. Ein Beispiel $(x, i) \in \mathcal{X} \times \mathcal{Y}$ zeigt an, dass x das i-te Klassifikationslabel erhalten hat. Wir können weiterhin mit der Null-Eins-Verlustfunktion agieren.

Regressionsprobleme: Es geht um die Vorhersage reeller Werte. Wir setzen dann $\mathcal{Y} = \mathbb{R}$. Als Verlustfunktionen bieten sich an der *quadratische Fehler*

$$\ell_{sq}(h,(x,y)) = (h(x) - y)^2$$

oder auch der absolute Fehler

$$\ell_{abs}(h,(x,y)) = |h(x) - y| \ ,$$

den h auf dem Beispiel (x, y) produziert.

Unüberwachtes Lernen Während beim überwachten Lernen Beispiele die Form $z = (x, y)$ haben und aus der Menge $Z = \mathcal{X} \times \mathcal{Y}$ stammen, sind beim unüberwachten Lernen die Beispiele $z \in Z$ unmarkiert. Was geeignete Verlustfunktionen betrifft, sei auf die einschlägige Literatur zu „Clustering“ verwiesen (zum Beispiel Kapitel 22 des Lehrbuches von Shalev-Shwartz und Ben-David).

Diese Überlegungen führen schließlich zu folgender recht allgemein gefassten

Definition 3.3 (Agnostisches PAC-Lernen (allgemeiner Fall)) *Eine Hypothesenklasse $\mathcal{H}$ heißt* agnostisch PAC-lernbar bezüglich einer Verlustfunktion $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$, *wenn eine Funktion $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ und ein Lernalgorithmus A existieren, so dass für alle $0 < \varepsilon, \delta < 1$ sowie für jede Verteilung $\mathcal{D}$ auf Z folgende Bedingung erfüllt ist: A, gestartet auf einer zufälligen Trainingsmenge $S \sim D^m$ mit $m \geq m(\varepsilon, \delta)$, ermittelt eine Hypothese $h \in \mathcal{H}$ und es gilt*

$$\Pr_{S\sim D^m}[L_{\mathcal{D}}(h) \leq \min_{h'\in\mathcal{H}} L_{\mathcal{D}}(h') + \varepsilon] \geq 1 - \delta \ , \tag{4}$$

wobei $L_{\mathcal{D}}(h) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$. In Worten: mit einer Wahrscheinlichkeit von mindestens $1-\delta$ wird eine zufällige Trainingsmenge generiert, aus der A eine Hypothese aus $\mathcal{H}$ extrahiert, deren Verlustwert den bezüglich $\mathcal{H}$ kleinstmöglichen Verlustwert um maximal ε überschreitet.

Eine Variante des agnostischen PAC-Lernmodells ist „Darstellungs-unabhängiges agnostisches Lernen“. Hier wird dem Algorithmus erlaubt, seine Hypothesen aus einer evtl. größeren Klasse $\mathcal{H}' \supset \mathcal{H}$ zu wählen. Die Verlustfunktion hat dann die Form $\ell : \mathcal{H}' \times Z \to \mathbb{R}_+$, d.h., sie

ist auch auf Hypothesen der erweiterten Klasse $\mathcal{H}'$ definiert. Wie wir später sehen werden, ist diese Variante insbesondere unter Effizienzkriterien interessant.

Die uns aus dem Abschnitt 3.1 bekannte Definition der Informationskomplexität können wir auf die allgemeine Definition 3.3 übertragen: die Funktion $m_{\mathcal{H}}$, die der Definition 3.3 genügt und unter dieser Nebenbedingung die kleinstmöglichen Werte $m_{\mathcal{H}}(\varepsilon, \delta)$ hat, wird als *Informationskomplexität* von $\mathcal{H}$ (bezüglich der Verlustfunktion ℓ) bezeichnet.

3.3 Effizientes PAC-Lernen und Parametrisierte Lernprobleme

Eine (mit einem Komplexitätsparameter n) parametrisierte Hypothesenklasse hat die Form $\mathcal{H} = (\mathcal{H}_n)_{n\geq 1}$. Entsprechend ist die Beispielmenge $Z = (Z_n)_{n\geq 1}$ und die Verlustfunktion $\ell = (\ell_n)_{n\geq 1}$ parametrisiert. Die Verlustfunktion ℓ_n hat die Form $\ell_n : \mathcal{H}_n \times Z_n \to \mathbb{R}_+$. $\mathcal{H}$ gilt dann bezüglich ℓ als agnostisch PAC-lernbar, wenn jede Unterklasse $\mathcal{H}_n$ bezüglich ℓ_n agnostisch PAC-lernbar ist. $\mathcal{H}$ heißt bezüglich ℓ *effizient agnostisch PAC-lernbar*, falls zusätzlich gilt:

1. Es gibt einen allgemeinen Algorithmus[1] A, der für alle $n \geq 1$ und gestartet auf einer Trainingsmenge mit Beipielen aus Z_n, ein Algorithmus zum agnostischen PAC-Lernen von $\mathcal{H}_n$ bezüglich ℓ_n ist.

2. Wenn $m_{\mathcal{H}_n}(\varepsilon, \delta)$ die Informationskomplexität von $\mathcal{H}_n$ bezüglich ℓ_n bezeichnet, dann ist die Funktion $m_{\mathcal{H}}(n, \varepsilon, \delta) = m_{\mathcal{H}_n}(\varepsilon, \delta)$ polynomiell in $n, 1/\varepsilon, 1/\delta$ beschränkt (beherrschbare Informationskomplexität).

3. Die Laufzeit von A auf Trainingsmengen $S \in Z_n^m$ ist polynomiell in m und n beschränkt (beherrschbare Berechnungskomplexität).

In Anwendungen wie Booleschen bzw. geometrischen Klassifikationsproblemen, gilt häufig $Z_n = \mathcal{X}_n \times \{0, 1\}$ mit $\mathcal{X}_n = \{0, 1\}^n$ bzw. $\mathcal{X}_n = \mathbb{R}^n$. Die Hypothesen aus $\mathcal{H}_n$ sind dann Boolesche oder geometrische Klassifikatoren $h : \mathcal{X}_n \to \{0, 1\}$. Für jede Wahl von n wird $\ell_n : \mathcal{H}_n \times Z_n \to \mathbb{R}_+$ als Null-Eins-Verlustfunktion gewählt.

3.4 Verteilungsunabhängigkeit des PAC-Lernmodells

Das PAC-Lernmodell legt fest, dass ein Lernalgorithmus A sein Erfolgskriterium für jede Wahl der Verteilung $\mathcal{D}$ erreichen muss. Er hat dabei keinerlei Vorwissen über $\mathcal{D}$ und kann lediglich über die zufällige Trainingsmenge S Rückschlüsse auf $\mathcal{D}$ tätigen. Zum Beispiel könnte der empirische Verlustwert (bei Klassifikationsproblemen die empirische Fehlerrate)

$$L_S(h) = \frac{1}{m} \cdot \sum_{i=1}^{m} \ell(h, z_i)$$

für eine Trainingsmenge $S = (z_1, \ldots, z_m) \sim \mathcal{D}^m$ ein guter Schätzwert für $L_D(h)$ sein.

[1] im Unterschied zu einem separaten Algorithmus A_n für jede Wahl von n

Lernmodelle, bei denen der Lernerfolg unabhängig von der Wahl der Verteilung $\mathcal{D}$ erreicht werden muss, heißen „verteilungsunabhängige Modelle“. Im Gegensatz dazu stehen „verteilungsabhängige Modelle“, bei denen die Algorithmen auf bestimmte Verteilungen (die evtl. auch noch schöne mathematische Eigenschaften haben), spezialisiert sein dürfen. Verteilungsunabhängige Modelle stellen wesentlich höhere Anforderungen an den Lerner.