Computational Learning Theory

Sommersemester 2018

Mario Holldack, M. Sc.
Prof. Dr. Georg Schnitger
Hannes Seiwert, M. Sc.

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institut für Informatik
AG Theoretische Informatik

Übungsblatt 4

Ausgabe: 07.05.2018
Abgabe: 14.05.2018

Um nachzuweisen, dass ein Entscheidungsproblem A NP-hart ist, genügt es, ein NP-hartes Problem B polynomiell auf A zu reduzieren (kurz $B \leq_p A$):
Ein Problem B ist genau dann *polynomiell auf A reduzierbar*, wenn es eine in Polynomialzeit berechenbare Transformation T gibt, sodass für jede Eingabe x von Problem B gilt:

$$x \in B \iff T(x) \in A$$

Aufgabe 4.1 *Schnitte von Halbräumen* (10 Punkte)

Zeigen Sie, dass das schwache Konsistenzproblem für die Konzeptklasse aller Schnitte $H_1 \cap H_2$ zweier Halbräume NP-hart ist.

Hinweis: Reduzieren Sie SetSplitting auf das schwache Konsistenzproblem. Im SetSplitting-Problem sind ein endliches Universum $U = \{1, \ldots, n\}$ und Teilmengen $T_1, \ldots, T_m \subseteq U$ gegeben. Zu entscheiden ist, ob eine Partition $U = U_1 \,\dot\cup\, U_2$ existiert, sodass $T_i \not\subseteq U_j$ für alle $i = 1, \ldots, m$ und $j = 1, 2$ gilt.

Für eine Eingabe $U, T_1, \ldots, T_m$ konstruieren Sie die transformierte Eingabe, d. h. eine klassifizierte Beispielmenge $S \subseteq \mathbb{R}^n$ wie folgt:

- Für jedes $u \in U$ klassifiziere den u-ten kanonischen Einheitsvektor e_u negativ.
- Für jedes T_i klassifiziere den Vektor $v_{T_i} := \frac{1}{|T_i|} \sum_{t \in T_i} e_t$ positiv.

Aufgabe 4.2 *k-Klausel-KNF und k-DNF* (5 + 5 Punkte)

In der Vorlesung wurde gezeigt, dass k-Klausel-KNFs[1] mit der Hypothesenklasse k-KLAUSEL-KNF vermutlich[2] nicht effizient PAC-lernbar sind. Abhilfe schafft hier die Wahl einer anderen (größeren) Hypothesenklasse.

a) Zeigen Sie: Für jede k-Klausel-KNF ϕ existiert eine äquivalente k-DNF ψ.

b) Sei $k \in \mathbb{N}$ fest. Beschreiben Sie einen effizienten Algorithmus, der das starke Konsistenzproblem für die Hypothesenklasse k-DNF löst, und bestimmen Sie seine Laufzeit.

Bitte wenden!

[1] d. h. KNF-Formeln mit höchstens k Disjunktionstermen: $\phi = \bigwedge_{i=1}^{k} \bigvee_j l_{i,j}$

[2] Sofern NP $\not\subseteq$ RP gilt, d. h. sofern nicht jedes Problem in NP einen effizienten probabilistischen Algorithmus mit einseitig beschränktem Fehler besitzt.

Aufgabe 4.3 *Schwierigkeit des ERM-Problems im agnostischen Fall* (8 + 4 Punkte)

Wir wissen bereits, wie man das Konsistenzproblem für Monome und Halbräume effizient lösen kann. Nun betrachten wir das Lernen im agnostischen Fall für den 0-1-Loss: Eine endliche Multimenge S von klassifizierten Beispielen ist gegeben. Eine Hypothese mit der geringsten Anzahl von Fehlklassifikationen – unter allen Hypothesen in der Hypothesenklasse – ist zu bestimmen.

a) Betrachten Sie die Hypothesenklasse MONOTON-MONOM_n aller monotoner Monome über n Variablen. (Ein monotones Monom ist eine Konjunktion positiver Literale.)
Zeigen Sie: Das schwache ERM-Problem[3] für $(\text{MONOTON-MONOM}_n)_{n\in\mathbb{N}}$ ist NP-hart.

b) Betrachten Sie die Hypothesenklasse RECHTECK_n aller n-dimensionaler Rechtecke.
Zeigen Sie: Das schwache ERM-Problem für $(\text{RECHTECK}_n)_{n\in\mathbb{N}}$ ist NP-hart.

Hinweis: Sie können dieselbe Transformation wie in Teil a) verweden.

Hinweis: Sie können jeweils VERTEXCOVER auf das ERM-Problem reduzieren. Wählen Sie jeweils n als die Anzahl der Knoten $|V|$.

Folgende Vorgehensweise ist zu empfehlen: Stellen Sie Knoten durch positive Beispiele und Kanten durch negative Beispiele dar – oder umgekehrt. Sie können Beispiele mehrfach erstellen und dadurch eine Gewichtung der Beispiele simulieren. Wählen Sie die Gewichte der „Kanten-Beispiele“ hinreichend groß, um zu gewährleisten, dass jede Lösung des ERM-Problems Fehlklassifizierungen nur auf „Knoten-Beispielen“ machen kann.

Beschreiben Sie Ihre Transformation und führen Sie den Nachweis der Äquivalenz. Eine Begründung der Laufzeit ist nicht nötig.

Bonusaufgabe 4.4. *Rechtecke lernen mit verrauschten Beispielen* (3 + 5 Extrapunkte)

Wir betrachten das ERM-Problem für die die Konzeptklasse RECHTECK_d aller d-dimensionalen Rechtecke:

Gegeben ist eine Menge $S \subseteq \mathbb{R}^d \times \{0,1\}$ von s klassifizierten Beispielen.
Gesucht ist ein Rechteck $R_{a,b}$, welches den 0-1-Loss bzgl. S minimiert.

a) Beschreiben Sie einen Algorithmus mit möglichst geringer Laufzeit, der das Problem für $d=1$ löst. Analysieren Sie auch die Laufzeit Ihres Algorithmus (in Abhängigkeit von s).

In Aufgabe 4.3 b) haben wir gezeigt, dass das ERM-Problem für allgemeines $d\in\mathbb{N}$ NP-hart ist. Dabei sind wir allerdings davon ausgegangen, dass die Beispielmenge S „bösartig“ gewählt werden darf. Wir betrachten nun den Fall einer Beispielmenge S, die von einer „gutartigen“ Verteilung zufällig erzeugt wird.

b) Gegeben sei ein (unbekanntes) Zielkonzept $c \in \text{RECHTECK}_d$. Die Beispiele x_i werden gemäß der Gleichverteilung auf $[0,1]^d$ zufällig gezogen. Beispiele x_i innerhalb von c werden mit Wahrscheinlichkeit $1-\eta$ positiv klassifiziert, Beispiele x_i außerhalb c werden mit Wahrscheinlichkeit $1-\eta$ negativ klassifiziert. Dabei sei $0 < \eta \leq 1/4$.

Beschreiben Sie eine *effiziente* Heuristik, die für diesen Fall mit guter Wahrscheinlichkeit „vernünftige“ Ergebnisse liefert. Eine „hand-waving“ Argumentation, warum Ihr Ansatz sinnvoll ist, genügt. Eine mathematische Analyse ist nicht verlangt.

[3] siehe Definition 6.11 im Skript