Computational Learning Theory

Sommersemester 2017

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

GOETHE UNIVERSITÄT FRANKFURT AM MAIN
Institut für Informatik
AG Theoretische Informatik

Übungsblatt 7

Ausgabe: 06.06.2017
Abgabe: 12.06.2017

Hinweis: Beachten Sie beim Bearbeiten der Aufgaben, dass der Perzeptron-Algorithmus aus der Vorlesung und die dazugehörigen Analysen (Satz 8.23 und Satz 8.25) mit Hypothesen der Form $\text{sign}(\langle w, x\rangle)$ statt $\text{sign}(\langle w, x\rangle + t)$ arbeiten. Wie lernt der Perzeptron-Algorithmus dennoch den Schwellenwert t?

Aufgabe 7.1 *Threshold-Funktionen mit ganzzahligen Koeffizienten* $(6 + 6 = 12$ Punkte)

a) Die Threshold-Funktion $f : \{0,1\}^n \to \{-1,1\}$ ist zu lernen.

   i) Zeigen Sie, dass der Perzeptron-Algorithmus höchstens $\mathcal{O}(n^2 \cdot W^2)$ Aktualisierungen durchführt, falls es für f eine Implementierung $f(x) = \text{sign}(\langle w, x\rangle - t)$ mit ganzzahligen Gewichten $w_1, \ldots, w_n \in \mathbb{Z}$ und einem ganzzahligen Schwellenwert $t \in \mathbb{Z}$ mit $|w_1|, \ldots, |w_n|$, $|t| \leq W$ gibt.
   Hinweis: Für alle $x \in \{0,1\}^n$ gilt: $\langle w, x\rangle \geq t \iff \langle w, x\rangle - t + \frac{1}{2} \geq 0$.

   ii) Wie viele Gegenbeispiele reichen für den Halbierungsalgorithmus aus?

b) Betrachten Sie die Funktion $\text{COMP}_n : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ zum Vergleich von Binärzahlen, also

$$\text{COMP}_n(x, y) = \begin{cases} +1 & \text{falls } \sum_{i=1}^n x_i 2^{i-1} \geq \sum_{i=1}^n y_i 2^{i-1}, \\ -1 & \text{sonst.} \end{cases}$$

   i) Es gibt viele Implementierungen von COMP_n als Thresholdfunktion $\text{sign}(\langle w, x\rangle - \langle u, y\rangle)$ mit ganzzahligen Gewichten und Schwellenwert 0. Geben Sie eine für jede Implementierung gültige (möglichst gute) untere Schranke für den Absolutbetrag des größten Gewichts von w an.

   ii) Wie viele Gegenbeispiele benötigt der Perzeptron-Algorithmus für COMP_n im schlimmsten Fall? Wie viele Gegenbeispiele reichen stets aus?

Aufgabe 7.2 *Perzeptron terminiert nicht immer* $(5 + 5 = 10$ Punkte)

In der Vorlesung wurde die Frage aufgeworfen, ob der Perzeptron-Algorithmus immer hält. Konstruieren Sie für die beiden folgenden Zielfunktionen eine unendliche Folge $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ von Gegenbeispielen. Begründen Sie jeweils, ob die Beispielmenge linear trennbar ist.

a) Die Zielfunktion sei die Threshold-Funktion $f : \mathbb{R}^n \to \{-1,1\}$ mit $f(x) = \text{sign}(x_1)$.

b) Die Zielfunktion sei die XOR-Funktion $g : \{0,1\}^2 \to \{-1,1\}$ mit $g(x_1, x_2) = \text{sign}\big((x_1 \oplus x_2) - \frac{1}{2}\big)$.

Bitte wenden!

Aufgabe 7.3 *Perzeptron für* VERTEXCOVER *und* HITTINGSET (5 + 5 = 10 Punkte)

a) In Aufgabe 4.3 b) haben wir gezeigt, dass das schwache ERM-Problem für HALBRAUM NP-hart ist, indem wir VERTEXCOVER auf das ERM-Problem reduziert haben.

Für eine Eingabe $G = (V, E)$ und $k \leq |V|$ haben wir die Beispielfolge $S \subseteq \mathbb{R}^{|V|}$ wie folgt konstruiert:

- Für jeden Knoten $v \in V$ klassifiziere den v-ten kanonischen Einheitsvektor e_v negativ.
- Für jede Kante $\{a, b\} \in E$ klassifiziere den Vektor $v_{\{a,b\}} := \frac{e_a + e_b}{2}$ positiv und füge ihn $(n + 1)$-mal zu S hinzu.

Dann gilt für jedes VertexCover $H \subseteq V$ der Größe k, dass die Threshold-Funktion $\operatorname{sign}(\langle w, x\rangle - \frac{1}{4})$ mit $w = \sum_{h \in H} e_h$ nur die Knoten des VertexCovers falsch klassifiziert.

Begründen Sie kurz, weshalb S nicht linear trennbar ist. Zeigen Sie, dass der Perzeptron-Algorithmus für S höchstens $\mathcal{O}(k)$ Aktualisierungen durchführt, unabhängig davon, in welcher Reihenfolge die Beispiele vorgelegt werden.

b) Betrachten Sie nun das HITTINGSET-Problem, welches eine Verallgemeinerung von VERTEX-COVER darstellt. Eine HITTINGSET-Instanz besteht aus nicht-leeren Mengen $V_1, \ldots, V_m \subseteq V$ und einer Zahl $k \leq |V|$, sodass ein HittingSet $H \subseteq V$ existiert mit $|H| \leq k$ und $H \cap V_i \neq \emptyset$ für alle $i \in \{1, \ldots, m\}$.

Geben Sie auch hier eine möglichst gute obere Schranke für die Anzahl der Aktualisierungen des Perzeptron-Algorithmus an.

Hinweis: Gehen Sie vor wie in Teilaufgabe a). Konstruieren Sie eine Beispielfolge $S \subseteq \mathbb{R}^{|V|}$ und geben Sie eine Threshold-Funktion $\operatorname{sign}(\langle w, x\rangle - t)$ an, sodass für jedes HittingSet $H \subseteq V$ der Größe k nur die Knoten des HittingSets falsch klassifiziert werden.