

Sommersemester 2018

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

## Übungsblatt 6

Ausgabe: 21.05.2018

Abgabe: 28.05.2018

- Wir haben zwei Varianten des Winnow-Algorithmus kennengelernt: *Winnow-1* „bestraft“ falschliegende Experten drakonisch, indem ihr Gewicht auf 0 gesetzt wird. *Winnow-2* hingegen multipliziert Gewichte mit einem konstanten Faktor.
- In der Vorlesung wird später für den Perzeptron-Algorithmus eine obere Schranke für die hinreichende Gegenbeispielzahl hergeleitet. Wie auch schon für den Winnow-Algorithmus spielt hier der Margin eine wichtige Rolle, allerdings nicht für bzgl. der 1-Norm, sondern der 2-Norm. Die Threshold-Funktion  $f$  sei zu lernen.
  - Sei  $x^{(0)}, \dots, x^{(T-1)} \in \mathbb{R}^n$  eine Folge von Gegenbeispielen für den Perzeptron-Algorithmus mit  $\|x^{(i)}\|_2 \leq R$  für  $0 \leq i < T$ .
  - Sei  $\rho > 0$  der Margin von  $f$ , d. h. es gibt ein  $w \in \mathbb{R}^n$  mit  $\|w\|_2 = 1$  und für alle  $i$  gilt

$$0 < \rho \leq f(x^{(i)}) \cdot \langle w, x^{(i)} \rangle.$$

Dann wird gezeigt, dass der Perzeptron-Algorithmus höchstens  $T \leq \frac{R^2}{\rho^2}$  Gegenbeispiele benötigt.

Benutzen Sie dieses Resultat in den Aufgaben 6.2 und 6.3.

### Aufgabe 6.1 *Disjunktionen auf $k$ Literalen*

(8 Punkte)

In der Vorlesung wurde gezeigt, dass Winnow-1 beim Lernen von Disjunktionen bestehend aus  $k$  Literalen auf  $n$  Variablen stets mit  $\mathcal{O}(k \cdot \log n)$  Gegenbeispielen auskommt. In dieser Aufgabe zeigen wir, dass Winnow-1 diesbezüglich (asymptotisch) optimal ist, d. h. dass im Worst-Case  $\Omega(k \cdot \log n)$  Gegenbeispiele notwendig sind. Es gelte  $0 < \mu \leq 1$  und  $k \leq n^{1-\mu}$ .

Sei  $\text{MONOTONE-DISJUNKTION}_{n,k}$  die Klasse der monotonen Disjunktionen mit höchstens  $k$  Literalen auf  $n$  Variablen. Zeigen Sie:

$$\text{VC}(\text{MONOTONE-DISJUNKTION}_{n,k}) = \Omega(k \cdot \log n).$$

*Hinweis:* Betrachten Sie zunächst den Fall  $k = 1$ .

**Bitte wenden!**

**Aufgabe 6.2** *Perzeptron vs. Winnow*

(8 + 4 = 12 Punkte)

Sei im Folgenden  $k = o(n)$  und der Beispierraum sei  $X = \{0, 1\}^n \times \{1\}$ .

- a) Wie viele Gegenbeispiele benötigt der Perzeptron-Algorithmus mindestens, um die Funktion

$$f(x_1, \dots, x_n) = x_1 \vee \dots \vee x_k$$

zu lernen, wie viele reichen aus? Bestimmen Sie jeweils eine asymptotische untere bzw. obere Schranke.

*Hinweis:* Konstruieren Sie zum Nachweis der unteren Schranke eine Folge von Gegenbeispielen, die den Perzeptron-Algorithmus zu möglichst vielen Aktualisierungen zwingt. Betrachten Sie zunächst den Fall  $k = 1$ . Nehmen Sie an, dass der Perzeptron-Algorithmus mit dem Nullvektor beginnt.

Wie viele Gegenbeispiele reichen für Winnow-1 bzw. Winnow-2 aus?

- b) Betrachten Sie die Funktion  $f_k : \{0, 1\}^n \rightarrow \{-1, +1\}$  mit

$$f_k(x_1, \dots, x_n) = 1 \iff \sum_{i=1}^n x_i \geq k$$

Bestimmen Sie die hinreichenden Gegenbeispielzahlen von Winnow-2 und Perzeptron für  $f_k$  asymptotisch. Bestimmen Sie dazu zunächst den jeweiligen Margin.

Wie ändern sich jeweils die Gegenbeispielzahlen, wenn wir nur Beispiele mit höchstens  $m$  Einsen (für ein  $k \leq m < n$ ) erhalten?

**Aufgabe 6.3** *Threshold-Funktionen mit ganzzahligen Koeffizienten* (2+5+5 = 12 Punkte)

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

- i) Zeigen Sie, dass der Perzeptron-Algorithmus höchstens  $\mathcal{O}(n^2 \cdot W^2)$  Aktualisierungen durchführt, falls  $f$  eine Implementierung  $f(x) = \text{sign}(\langle w, x \rangle - t)$  mit ganzzahligen Gewichten  $w_1, \dots, w_n \in \mathbb{Z}$  und ganzzahligem Schwellenwert  $t \in \mathbb{Z}$  mit  $|w_1|, \dots, |w_n|, |t| \leq W$  besitzt.

*Hinweis:* Aufgabe 5.3 c)

- ii) Wie viele Gegenbeispiele reichen für den Halbierungsalgorithmus aus?

- b) Der Perzeptron-Algorithmus erhalte die Beispiele

$$x^{(i)} = \underbrace{((-1)^i, \dots, (-1)^i, (-1)^{i+1}}_{i \text{ Komponenten}}, 0, \dots, 0) \in \mathbb{R}^n$$

mit  $f(x^{(i)}) = (-1)^{i+1}$  und  $1 \leq i \leq n$  in beliebiger, auch wiederholter Reihenfolge. Zeigen Sie, dass der Perzeptron-Algorithmus mindestens  $\Omega(2^n)$  Aktualisierungen benötigt, bis eine trennende Hyperebene gefunden wird.

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

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

- i) Es gibt viele Implementierungen von  $\text{COMP}_n$  als Thresholdfunktion  $\text{sign}(\langle w, x \rangle - \langle u, y \rangle)$  mit ganzzahligen Gewichten und Schwellenwert 0. Geben Sie eine für jede solche Implementierung gültige (möglichst gute) untere Schranke für den Absolutbetrag des größten Gewichts von  $(w, u)$  an.
- ii) Wie viele Gegenbeispiele benötigt der Perzeptron-Algorithmus für  $\text{COMP}_n$  im schlimmsten Fall? Wie viele Gegenbeispiele reichen stets aus?

*Hinweis:* In Teil b) und c) können Sie den Schwellenwert ignorieren.