

Computational Learning Theory

Wintersemester 2011/2012

Prof. Dr. Georg Schnitger

Arbeitsgruppe Theoretische Informatik, Institut für Informatik



Blatt 1

Ausgabe: 24.10.2011

Abgabe: 31.10.2011

1.1. Aufgabe (8)

Symmetrische Funktionen

Eine symmetrische Funktion $f : \{0, 1\}^n \rightarrow \{0, 1\}$ errechnet dieselbe Ausgabe unter allen Permutationen einer Eingabe. Um eine unbekannte solche Funktion f zu erlernen erhalten wir klassifizierte Beispiele $(x, b) \in \{0, 1\}^n \times \{0, 1\}$ mit $f(x) = b$.

Gib die hinreichende Anzahl von Beispielen für einen PAC-Algorithmus mit Fehler ϵ und Vertrauensparameter δ an und beschreibe eine effiziente Implementierung.

1.2. Aufgabe (8)

Totale Ordnungen

Wir suchen nach einem effizienten PAC-Algorithmus für das Lernen einer unbekannt totalen Ordnung \leq auf den n Elementen der Menge M . Für ein klassifiziertes Beispiel $(x, y, b) \in M \times M \times \{0, 1\}$ gilt $x \leq y \Leftrightarrow b = 1$.

Gib die hinreichende Beispiellanzahl an und beschreibe den Algorithmus.

1.3. Aufgabe (8)

Entscheidungsbäume

Sei $X = \{x_1, \dots, x_n\}$ eine Menge von booleschen Variablen. Ein Entscheidungsbaum T ist ein binärer Baum, dessen innere Knoten eine Markierung aus X tragen und zwei Kinder haben. Die beiden entsprechenden Kanten sind mit 0 und 1 markiert. Blätter tragen die Markierung 0 oder 1. Eine Belegung der Variablen aus X definiert also einen Weg von T 's Wurzel zu einem Blatt, wenn im Knoten mit Markierung x_i die Kante mit x_i 's Wert verfolgt wird. T wird genau dann von einer Belegung erfüllt, wenn dieser Weg in einem mit 1 markierten Blatt endet. $\mathcal{T}_{n,p}$ sei die Klasse der Entscheidungsbäume auf n Variablen und mit höchstens p inneren Knoten.

Die Konzeptklasse $\text{Decision}_{n,p}$ besitzt für jeden Entscheidungsbaum $T \in \mathcal{T}_{n,p}$ das Konzept $\{x : x \text{ erfüllt } T\}$.

Bestimme mit Hilfe einer oberen Schranke für die Mächtigkeit von $\mathcal{T}_{n,p}$ eine ausreichende Anzahl von Beispielen für einen PAC-Algorithmus für $\text{Decision}_{n,p}$. Die Angabe eines Algorithmus ist nicht notwendig.