Übung zur Vorlesung

Maschinelles Lernen 2

Sommersemester 2009

Abteilung Maschinelles Lernen
Institut für Softwaretechnik und
theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: krm@cs.tu-berlin.de

Blatt 5

Abgabe bis Montag, 25. Mai 2009, 13:00 Uhr
per Email an mikio@cs.tu-berlin.de, Ausarbeitung im Sekretariat FR6052, oder bei Mikio Braun, FR6058.

Aufgaben

1. **Kerne für Wörter (10 Punkte)** Der "Bag-of-Words"-Kern ist definiert als

$$k_1(x,z) = \sum_{w \in L} \#_w(x)\#_w(z) \cdot N_w,$$

wobei $\#_w(x)$ die Häufigkeit von w in der Sequenz x ist und die Einbettungssprache L einer natürlichen Sprache entspricht, z.B. Deutsch. Um den Einfluß verschiedener Worte zu Gewichten kann jedem Wort w ein Gewicht N_w zugewiesen werden.

Häufig wird die inverse Dokumentfrequenz (IDF) als Gewichtung verwendet. Für einen Datensatz D aus Dokumenten bestimmt man für jedes Wort w in D die Teilmenge D_w von Dokumenten, die es enthalten. Die Gewichtung ist dann definiert als

$$N_w = \log|D| - \log|D_w|.$$

Beweise, dass für diese Gewichtung k_1 tatsächlich ein Kernel ist.

2. **Kerne für N-gramme (10 Punkte)** Der Spektrum-Kern ist ohne Gewichtung definiert als

$$k_2(x,z) = \sum_{w \in L} \#_w(x)\#_w(z) \quad \text{mit} \quad L = \mathcal{A}^n,$$

wobei $\mathcal{A}^n$ die Menge aller Sequenzen der Länge n (N-gramme) ist. Definiert man die Einbettungssprache L als Vereinigung von N-grammen verschiedener Längen, erhält man einen "geblendeten" Spektrum-Kern

$$k_3(x,z) = \sum_{w \in L} \#_w(x)\#_w(z) \quad \text{mit} \quad L = \bigcup_{i=1}^{n} \mathcal{A}^i.$$

Berechne für beide Kerne k_2 und k_3 Kernmatrizen für $n = 3$ über die folgende Menge von Sequenzen:

"ananas", "anna", "natter", "otter", "otto"

3. **Kerne für Teilsequenzen (10 Punkte)** Zur Berechnung des Subsequenz-Kerns wird ein Suffixbaum verwendet.

Zeige, dass ...

(a) eine Sequenz x genau $\frac{|x|^2+|x|}{2}$ Teilsequenzen (Substrings) enthält,

(b) jede Teilsequenz w in $\mathcal{O}(|w|)$ in einem Suffixbaum zu erreichen ist und

(c) ein Suffixbaum nur $\mathcal{O}(|x|)$ Speicher benötigt.