Übung zur Vorlesung

Maschinelles Lernen 2

Sommersemester 2011

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

Blatt 6

Abgabe bis Montag, 30. Mai 2011, 12:00 Uhr, Briefkasten bei Raum FR6061

1. **Kerne für Wörter (9 Punkte)**. Der Bag-of-Words-Kern ist definiert als

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

wobei $\#_w(x)$ die Häufigkeit des Worts w in dem String x ist und L einer natürlichen Sprache entspricht, z.B. Deutsch oder Englisch. Um den Einfluss verschiedener Worte zu gewichten, kann jedem 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

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

Zeige, dass für diese Gewichtung k_1 ein valider Kern ist.

2. **Kerne für N-gramme (12 Punkte).** Der Spektrum-Kern ist ohne Gewichtung definiert als

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

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

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

Implementiere k_2 und k_3. Berechne Kernmatrizen für $n \in 2,3,4$ über die folgende Menge von Strings:

"ananas", "anna", "nato", "natter", "otter", "otto".

Gebe die Matrizen als Zahlen und als Bilder an (z.B. mit `imagesc` in Matlab). Welchen Einfluss hat n auf die Kernmatrizen?

3. **Kerne für Teilstrings (9 Punkte).** Zur Berechnung von komplexen Stringkernen wird häufig ein Suffixbaum verwendet.
   (a) Zeige, dass ein String x genau $\frac{|x|^2+|x|}{2}$ Teilstrings enthält.
   (b) Zeige, dass ein Teilstring w in $O(|w|)$ in einem Suffixbaum gefunden werden kann.
   (c) Zeige, dass ein Suffixbaum nur $O(|x|)$ Speicher benötigt.

Für Fragen zum Übungsblatt bitte in der Google Group http://groups.google.com/group/ml-tu registrieren und die Fragen an die Mailingliste stellen.