7. Übung zur Vorlesung "Internet-Suchmaschinen" im Wintersemester 2007/2008 – mit Musterlösungen –

Dr. Andreas Hotho, Prof. Dr. Gerd Stumme, Dipl.-Wi.Inf. Beate Krause

07. Februar 2008

1 Recommender-Systeme

1. Erklären Sie mit eigenen Worten, was der Pearson-Korrelationskoeffizient aussagt!

   Der Pearson-Korrelationskoeffizient zeigt an, wie sehr zwei Bewertungen in den Abweichungen nach oben und unten vom Mittelwert in den jeweiligen Positionen übereinstimmen.

2. Betrachten Sie die Bewertungen von Filmen durch die Benutzer Alice (A), Bob (B) und Charlie (C) gemäß der folgenden Tabelle:

Film	Alice	Bob	Charlie
Titanic	7	9	5
High Fidelity	5	7	5
American Beauty	5	7	5
Jarhead	4	6	4
Life of Brian	4	6	4
Three Kings	5		
A Fish Called Wanda			4

   Schätzen Sie – gemäß der vorigen Antwort – ab, wie die Größe der Korrelationskoeffizienten $c_{A,B}, c_{B,C}, c_{A,C}$ relativ zueinander aussehen wird!

   Da Alice und Bob in den Abweichungen nach oben und unten übereinstimmen, sind sie maximal korreliert, also $c_{A,B} = 1$. Weniger stark wird die Korrelation von Alice und Charlie sein, da sie sich bei "Titanic" uneinig sind, obwohl die absoluten Werte nahe beieinander liegen. Die Koeffizienten $c_{A,C}$ und $c_{B,C}$ werden gleich sein, da die Bewertungen von A und B nur verschoben sind und somit die gleichen Abweichungen vom jeweiligen Mittelwert aufweisen.

3. Berechnen Sie die Korrelationskoeffizienten $c_{A,B}, c_{B,C}, c_{A,C}$!

   Durch Einsetzen in die Formel ergeben sich $c_{A,B} = 1, c_{B,C} = 0.75, c_{A,C} = 0.75$.

4. Sagen Sie eine Bewertung der Filme "Three Kings" und "A Fish Called Wanda" durch den Anwender Bob voraus! Ziehen sie dazu jeweils den anderen Anwender heran, der den fraglichen Film bewertet hat.

   Three Kings: Alice hat diesen Film mit 5 bewertet. Da wir nur einen weiteren Anwender betrachten, kürzt sich das $w_{Bob,\cdot}$ heraus und man hat

   $p_{Bob,Three\ Kings} = \bar{r}_{Bob} + (r_{Alice,\ Three\ Kings} - \bar{r}_{Alice}) = 7 + (5 - 5) = 7$

   A Fish Called Wanda:

   $p_{Bob,\ Wanda} = \bar{r}_{Bob} + (r_{Charlie,\ Wanda} - \bar{r}_{Charlie}) = 7 + (4 - 4.6) = 6.4$

2 Klassifikation

1. Geben Sie jeweils eine "gute" Klassifikationsfunktion und eine Funktion, die an Overfitting leidet, an für folgende Klassifikationsaufgaben. Gehen Sie davon aus, das für das Lernen der Funktion eine gewisse Menge Trainingsdaten zur Verfügung steht.

   a) Ist ein gegebener Tag X ein Regentag?

      Gut: Am Tag X hat es zu jedem Zeitpunkt geregnet.

      Overfitted: Der Tag ist ein Mittwoch im November, ein Dienstag im Dezember, oder ein Dienstag vor dem 14. Februar. *(Wenn die Trainingsdaten entsprechend aussehen.)*

   b) Ist dieser Zug ein ICE?

      Gut: Der Zug ist weiß, hat eine runde Nase, rote Streifen an der Seite und hat eine Höchstgeschwindigkeit von 300 km/h.

      Overfitted: Der Zug fährt an einem Wochentag zu einer vollen Stunde von Hannover nach Karlsruhe.

   c) Kann ich jetzt an dieser Ampel losfahren?

      Gut: die Ampel zeigt grün und die zugehörigen Ampeln für die anderen Richtungen zeigen Rot.

      Overfitted: Es ist 13:19:28 Uhr oder 13:19:29 Uhr oder 13:19:30 Uhr oder ...

   d) Ist diese Mail Spam?

      Overfitted: Der Text lautet "Three Steps to the Software You Need at the Prices You Want" oder "Astounding Mortgages for the USA!" oder ...

      Besser: Die Mail enthält die Worte "Software, OEM, Bargain" oder "Viagra, Cialis" oder "Stock, Opportunity" oder ...

Gut: z. B. Bayes-Klassifizierer trainieren. Dieser lernt Wahrscheinlichkeiten der Art "Wenn die Worte X, Y, Z vorkommen, dann ist die Wahrscheinlichkeit P, daß die Mail Spam ist.". Diese werden aus den umgekehrten bedingten Wahrscheinlichkeiten der Trainingsdaten ermittelt und in der Regel mit einfachen Heuristiken wie oben kombiniert.

2. Wie könnten (in einem kleinen Beispiel, etwa 10 Dimensionen) Prototyp-Vektoren aussehen für die Klassifikationsaufgabe "Behandelt diese Nachrichtenmeldung das Thema Sport/Basketball/Politik?"

	Prototyp für		
Term	Sport	Basketball	Politik
spiel	0.3	0.7	0.1
rennen	0.3	0	0.1
entscheidung	0.6	0.3	0.7
korb	0	0.8	0
wahl	0	0	0.8
wurf	0.7	0.8	0
minister	0	0	0.9
medaille	0.4	0	0
gewinner	0.8	0.8	0.3
team	0.5	0.8	0.2

3. Konstruieren Sie einen Fall, wo Nearest-Neighbor bei der Aufgabe, einen Artikel in "Sport" und "Politik" zu klassifzieren, versagt!

   Ein Politiker könnte in einer Rede Sport-Metaphern benutzt haben, in der Art "In jedem Team muß es einen Mannschaftsführer geben, der es anführt. Nur so kann das Spiel gewonnen werden! *(bla bla bla)*". Wenn ein solcher Politik-Artikel bei der Nearest-Neighbor-Berechnung vorkommt, könnte z. B. ein Sport-Artikel fälschlicherweise als Politik klassifiziert werden.

4. Sie haben einen Trainingsdatensatz mit 20 Dokumenten. 8 von diesen klassifizieren Sie als Spam ($c = 1$), 12 als Nicht-Spam ($c = 0$). Folgende Wahrscheinlichkeiten konnten für die ersten Wörter in allen Dokumenten erstellt werden. Sie möchten mit Hilfe des Naive Bayes Klassifizierers entscheiden, ob das folgende Dokument Spam oder Nichtspam darstellt: o(web, marketing, seo). Der Naive Bayes Klassifizier ist definiert durch:

$$\arg\max_{c_i \in C} P(c_i|o) = (P(c_i) * P(o|c_i))/P(o) = P(c_i)P(o|c_i)$$

   Die Entscheidungsregel für den naiven Bayes Klassifikator ist also

$$\arg\max_{c_i \in C} P(c_i) \prod_{j=1}^{d} P(o_j|c_i)$$

d bezeichnet die Anzahl an Termen mit $o = (o_1, ..., o_d)$. Für diese Terme wurden die folgenden bedingten Wahrscheinlichkeiten im Trainingsdatensatz errechnet:

$P(web|c = 1) = 0.2$
$P(web|c = 0) = 0.8$
$P(marketing|c = 1) = 0.5$
$P(marketing|c = 0) = 0.5$
$P(seo|c = 1) = 0.9$
$P(seo|c = 0) = 0.1$

Es gilt:

$$P(S = 1|o) = (0.4 * 0.2 * 0.5 * 0.9)/0.4 = 0.036$$
$$P(S = 0|o) = (0.6 * 0.8 * 0.5 * 0.1)/0.4 = 0.024$$

Das Dokument ist mit höherer Wahrscheinlichkeit Spam.

3 Clustern

1. Formulieren Sie den k-Means-Algorithmus in Pseudocode!

Algorithm 1 k-Means

Input: Menge X von Objekten aus U, Distanzfunktion d, Clusteranzahl k, maximale Anzahl von Iterationen *maxIter*
Output: Cluster $C_i, i = 1, \ldots, k$

$iter \leftarrow 1$
Wähle Zentroide $c_1, \ldots, c_k \in U$ zufällig
repeat
  for $x \in X$ **do**
    Ordne x dem Cluster $C_i, i := \text{argmin}_{i=1\ldots k}\, d(x, c_i)$ zu
  end for
  for $i = 1 \ldots k$ **do**
    Berechne Zentroid c_i neu als $c_i \leftarrow \mu(C_i) = \frac{1}{C_i} \sum_{x \in C_i} x$
  end for
  $iter \leftarrow iter + 1$
until Zuordnung ändert sich nicht mehr oder $iter \geq maxIter$

2. Beschreiben Sie kurz das prinzipielle Vorgehen beim divisiven Clustern.

    1. Starte mit allen Objekten in einem Cluster
    2. Wähle zu teilenden Cluster, beispielsweise den größten oder denjenigen mit der größten Varianz.

3. Teile diesen Cluster, z. B. so daß die Varianz in den beiden Teilen minimiert wird.
4. Falls noch teilbare Cluster vorhanden: weiter bei 2.

3. Erläutern Sie warum diese Verfahren typischerweise schlechter arbeiten als die agglomerativen.

   Da bei divisien Verfahren zwingend immer ein Cluster geteilt werden muß, kann es dazu kommen, daß die Clusterstruktur der zugrundeliegenden Daten nicht wiedergegeben wird. So könnte z. B. ein Cluster, der auf natürliche Weise in drei Cluster zerfallen würde, durch teilen in zwei Cluster einen der drei Cluster "verlieren".

4 Informationsextraktion

Skizzieren Sie die Informationsextraktions-Aufgaben, die beim Bau eines Webangebots folgender Machart anstehen: auf `http://www.rottentomatoes.com` gibt es zu Filmen eine aggregierte Sicht der Bewertungen verschiedener Kritiker, insbesondere eine numerische Angabe der durchschnittlichen Bewertungen, sowie Textschnipsel aus den Reviews, Spielpläne nahegelegener Kinos und Daten zum Film selbst (Startdatum, Regisseur, usw.).

Extrahieren der Bewertungen: An einer bestimmten Stelle jeder Kritik (→ DOM-Baum!) steht die Bewertung (z. B. drei von vier Sternen); diese wird extrahiert und numerisch repräsentiert ($3/4 = 75\%$).

Extrahieren des Fazits: Viele Review-Seiten haben eine ausgezeichnete Stelle (z. B. der letzte HTML-Absatz `<p>`), an der das Fazit der Kritik zu finden ist.

Extrahieren von Kino-Spielplänen: Viele Kinos haben eine feste Seite, auf der der aktuelle Spielplan steht. Die Informationen dieser Seiten müssen mit dem Titel des jeweiligen Films abgeglichen und die Spielzeiten z. B. per Regex extrahiert werden.

Daten zum Film: Die Daten zum Film selbst könnten bspw. aus der Internet Movie Database extrahiert werden. Dort gibt es zu jedem Film eine Seite, die wiederum in einer festen HTML-Struktur die Details jedes Films wiedergibt. Dort kann aus dem DOM-Baum die gewünschte Information extrahiert werden.