Mining the Link Structure of the WWW

(Nutzbarmachung der Hyperlink-Strukturen des WWW)

Seminar Information Retrieval

Georg Kuschk 05.10.2005

Inhalt

1. Motivation / Hintergrund

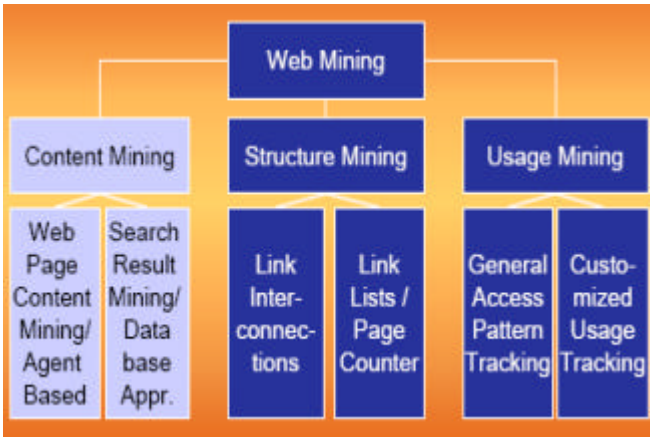
Suchmaschinen : herkömmlicher Ansatz (index-basierend) :

- Durch heutige Speichertechnologien ist es möglich , große Teile des WWW zu speichern und zu indizieren
- Webseiten mit bestimmten Suchwörtern können dadurch mit sehr großer Geschwindigkeit gefunden werden
- aber : Ergebnisumfang meist sehr groß (bis zu Millionen Seiten)

=> Problem : Sortierung nach Relevanz / Informationsgehalt

Naiver Ansatz :
manuell vorgegebene Sortierung / manuelle Kategorisierung

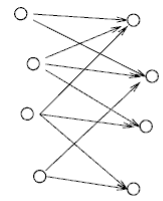
2. Web Structure Mining - Idee (~1999)



2. Web Structure Mining - Idee

- Erforscht die strukturellen Informationen des WWW (hauptsächlich Link-Analyse)
- Idee : Wenn 1.000.000 Authoren auf ihren Webseiten zum Thema “pdf-Datei” einen Link zu “www.adobe.com” angebracht haben , so ist diese Webseite nach mehrheitlicher Meinung die qualitativ hochwertigste zu diesem Thema.
- Betrachtung des WWW als gerichteten Graphen mit den (Webseiten=Knoten und den Hyperlinks=Kanten)
- Einteilung der Webseiten in zwei Kategorien :
  - **hubs** : Übersichtsseiten / Linksammlungen , von denen viele Links ausgehen.
  - **authorities** : Zielseiten , auf welche oft verlinkt wird.

2. Web Structure Mining - Idee



- Die hubs (Linksammlungen) definieren somit durch ihre angebrachten Links die authorities (Zielseiten) , welche die Ergebnis-Webseiten der Suchanfrage darstellen.

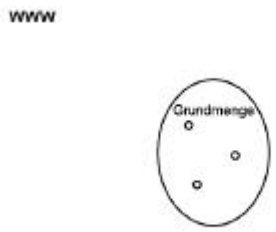
- Die hub-Seiten sind am besten geeignet , um einen Überblick für ein neues Thema zu erlangen , die authority-Seiten stellen das vertiefte Wissen bereit.

2.1 HITS-Algorithmus (Hyperlink Induced Topic Search)

Berechnet Listen von hubs und authorities für WWW- Suchthemen (spezifiziert durch einen oder mehrere Anfrageterme)

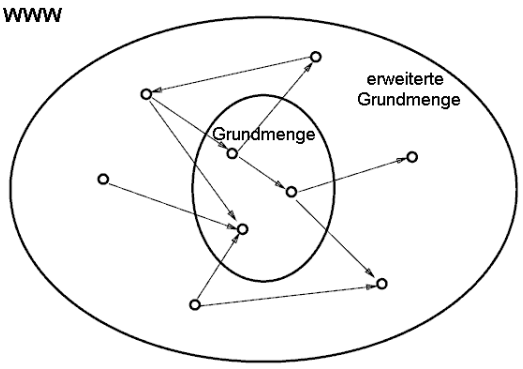
Algorithmus :

1.) Mit einer index-basierenden Suchmaschine wird mittels der Anfrageterme eine Grundmenge von Webseiten zusammengetragen



2.1 HITS-Algorithmus

2.) Erweiterung dieser Grundmenge durch Hinzufügen jener Seiten , auf die von der Grundmenge aus verwiesen wird und solcher , die auf Seiten aus der Grundmenge verweisen. (bis zu einer festgelegten Obergrenze)



2.1 HITS-Algorithmus

3.) Alle Links/Kanten zwischen Seiten derselben WWW-Domain entfernen (da Homepage-interne Navigation)

4.) Jeder Seite $p \in V$ wird nun eine nicht-negatives authority-Gewicht x_p , sowie ein nicht-negatives hub-Gewicht y_p zugewiesen.

- nur relative Werte interessant
- Werte normalisieren (z.B. Summe der Quadrate = 1)
- jede Seite gleiches Start-Gewicht

D.h. auf eine gute authority-Seite (hohes authority-Gewicht) wird von vielen anderen Seiten aus verlinkt , eine gute hub-Seite (hohes hub-Gewicht) verlinkt auf viele andere (themenrelevante) Seiten.

2.1 HITS-Algorithmus

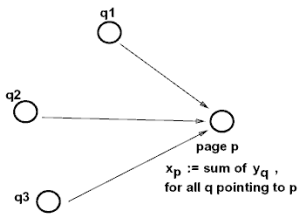
5.) Die authority- und hub-Gewichte werden nun für jede Seite p iterativ mit folgenden Updatefunktionen berechnet :

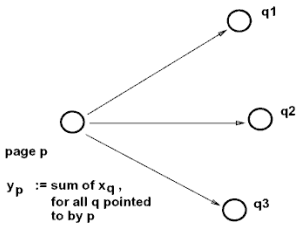
Operation I: $x_p = \sum_{q \to p} y_q$

und

Operation O: $y_p = \sum_{p \to q} x_q$

$(p \to q$ bedeutet : Seite p verlinkt auf Seite $q)$





2.1 HITS-Algorithmus

Fasse alle authority-Gewichte der Seiten in einem Vektor $x = (x_{p_1}, x_{p_2}, ..., x_{p_n})$ zusammen ; hub-Gewichte analog $y = (y_{p_1}, y_{p_2}, ..., y_{p_n})$

Iterationsfunktion :

Iterate(G, k)
{
  G : Menge von n Webseiten (erweiterte Grundmenge)
  k : Anzahl der Iterationen
  Initialisiere die Vektoren x, y mit dem Einheitsvektor
  $x, y = (1,1,...,1) \in R^n$
  for i = 1,2,...,k do
    Wende die Operation I auf (x_{i-1}, y_{i-1}) an: $x_i' = I(x_{i-1}, y_{i-1})$
    Wende die Operation O auf (x_i', y_{i-1}) an: $y_i' = O(x_i', y_{i-1})$
    Normalisiere $x_i' \Rightarrow x_i$
    Normalisiere $y_i' \Rightarrow y_i$
  endfor
  return(x_k, y_k)
}

k=20 ausreichend

2.1 HITS-Algorithmus

Alternative Berechnung der Gewichte :

- Seien $p_1, ..., p_n \in V$ die Seiten der erweiterten Grundmenge.
- Definiere die $n \times n$ Matrix A als Adjazenzmatrix der Seiten.

  D.h. $a_{ij} = 1$ gdw. Seite p_i verlinkt auf Seite p_j. (sonst $a_{ij} = 0$)

- neue Updatefunktionen :

  Operation I: $x = A^T y$
  Operation O: $y = Ax$

$x = A^T y = A^T Ax = (A^T A)x = (A^T A)A^T y = (A^T A)(A^T A)x$ ...
$y = Ax = AA^T y = (AA^T)y = (AA^T)Ax = (AA^T)(AA^T)y$ ...

2.1 HITS-Algorithmus

- Diese Iterationen des Vektors x, d.h. große Potenzen von A^TA auf x angewendet und normalisiert , konvergieren gegen den Haupt-Eigenvektor von A^TA

  Analog konvergieren die Werte für den normalisierten Vektor y gegen den Haupt-Eigenvektor von AA^T

  (Beweis siehe [2] Theorem 3.1)

- Die Konvergenz erfolgt für "gut gewählte" Initialisierungsvektoren (alle Elemente positiv).

=> die Berechnung des Haupt-Eigenvektors von A^TA bzw. von AA^T liefert ebenfalls das gewünschte Ergebnis.

2.1 HITS-Algorithmus

Was ist das Ergebnis ?

- Dem Nutzer werden die gefundenen Webseiten sortiert nach den authority-Gewichten präsentiert.
  D.h. die relevantesten (am meisten empfohlenen) Seiten zuerst.

- Desweiteren wird (als quasi Nebenprodukt) eine Liste der besten Übersichtsseiten / Linksammlungen geliefert.

2.1 HITS-Algorithmus

Nachteile von HITS :

1.) Ignorieren des Textinhaltes der Seiten.
(reine Link-basierende Berechnung)

=> Auf eng-fokussierten Themen liefert HITS Resultate
für allgemeinere Oberthemen.

(Da die Suchwörter alle in einer Oberthemen-Seite
zusammenhangslos stehen können und damit ebenfalls von
einer index-basierenden Suchmaschine gefunden werden,
und diese Oberthemen-Seiten einen größeren
Verlinkungsgrad haben.)

Bsp. : Eine Suche bzgl. “sächsische schweiz höhe falkenstein”
liefert hauptsächlich Apartment- und Ferienwohnungs-Informationen

2.1 HITS-Algorithmus

Nachteile von HITS :

2.) Aufgrund der Gleichgewichtung der Links , driftet HITS
manchmal ab , wenn die hubs verschiedene Themen abdecken.

=> Bsp. : Die Homepage eines Chemikers mag gute Links
zu anderen Chemie-relevanten Seiten haben , jedoch sind
dann auch meist Links bzgl. seiner Hobbies oder seines
Wohnortes enthalten , welche dann ebenfalls als
Chemie-relevante Seiten erachtet werden.

Verbesserungsansatz : „Clever“

2.2. Clever - Algorithmus

- Kombination des Inhaltes einer Homepage mit den Link-Informationen (“Clever system”):

- Ersetzung der Updatefunktionen durch gewichtete Summen , welche jedem Link ein nicht-negatives Gewicht zuordnen.

Gewichtung abhängig von folgenden Faktoren :

- Aufspaltung von umfangreichen hub-Seiten in kleinere Einheiten, da benachbarte Links einer Seite auf ein bestimmtes Thema stärker fokussiert sind als die ganze Seite.

- Es ist zu erwarten , dass der umgebende Text einer Hyperlink Definition (anchor text) Schlagwörter wie z.B. “chemistry” oder Wortstämme davon enthält. Solche Links bekommen ein größeres Gewicht zugewiesen.

2.2. Clever - Algorithmus

weitere Änderungen :

bisher : Links zwischen zwei Seiten derselben Domain nicht betrachten

jetzt : Diesen Links ein sehr kleines Gewicht zuordnen.
(Da im allgemeinen die Seiten interessant sind , auf die “global” am meisten verwiesen wird – nicht lokal.)

mathematischen Änderungen :

- Die Matrix enthält nun nicht mehr nur 0 und 1 , sondern nicht-negative , reellwertige Zahlen.

- Konvergenz weiterhin gegeben

Yahoo! vs. Clever

Studie :

37 gewöhnliche User wurden in 26 Themen , in denen sie keine Experten sind , befragt , wie sie die resultierenden Listen bewerten (wieviel sie über das Thema gelernt haben)

Ergebnis :

50% Clever besser
19% Yahoo! besser
31% Yahoo! und Clever gleichauf

Aber :

- Da hier die Suchtechniken untersucht werden sollten , wurde der große Vorteil von Yahoo! - die manuelle Klassifizierung - (Authoren Anmerkungen, online Zusammenfassungen) ausser Acht gelassen.
- Studie nicht umfangreich
- Stand 1999

3.1 Anwendung : Web Trawling

- Automatisiertes Auffinden von expliziten und impliziten Web-Communities.

Warum ?

- Erforschung der intellektuellen und soziologischen Entwicklung des WWW
- Sammeln von detailierten Informationen über Personengruppen mit bestimmten Interessen.
  (z.B. automatische Suche nach illegalen Tauschbörsen)

3.1 Anwendung : Web Trawling

Idee :

- Thematisch zusammenhängende Web-Communities im Kern dicht verlinkt (hubs/Linkseiten zu authorities/Zielseiten)
- enthalten somit gerichtete , bipartite Graphen
- aufgrund der dichten Vernetzung im Kern sehr wahrscheinlich auch kleinere , vollständige bipartite Graphen. (5-10 Kern-Webseiten)
- Durchsuchung des WWW-Graphen nach solchen Strukturen

3.2 Anwendung : Halbautomatische Klassifizierung

- Klassifizierung von Webseiten
- Yahoo! : Themen-Baum mit Unterthemen , Knoten enthalten die relevante Seiten
- Umfangreiche **automatische** Klassifizierungen mittels "Clever" :
  - Vorteil der Aktualität
  - Einmalige manuelle Konstruktion des "Themen"-Baumes.

Verfahrensweise :

- Jeder Knoten soll nun mit den besten hubs (Linksammlungen) und authorities (Zielseiten) bevölkert werden.

Einfachste Form :	Name des Knotens (Thema) als Anfrageterm
Praktisch :	Manuelle Überarbeitung der "Clever"-Ergebnisse

3.2 Anwendung : Halbautomatische Klassifizierung

- Liefert “Clever” einen Mix an relevanten und irrelevanten Seiten , so muss ein Administrator die qualitativ hochwertigen Seiten hervorheben. (relevance feedback)
- Zusätzlich werden einige Beispielseiten/prominente Vertreter der Themen manuell in die Grundmenge eingefügt.
- Bestimmte Themen werden von überwältigenden Web-Präsenzen dominiert => Nutzer-spezifizierte “stop-sites” sowie ihre verlinkte Umgebung werden aus der Grundmenge der Webseiten gelöscht.

  Bsp. : Rubrik “Building and Construction Supplies / Doors and Windows” sehr schwierig nicht ständig Microsoft Seiten geliefert zu bekommen =>Hinzufügen einer “stop-site” www.microsoft.com
- Ein Knoten wird “Clever” nun somit als Kombination von Anfragetermen , Beispiel-hub/-authority Seiten und möglicher “stop-sites” beschrieben.

4. Zusammenfassung

- Effizienzsteigerung von Suchmaschinen durch Ausnutzen der , in Hyperlinks enthaltenen, strukturellen Zusatzinformationen :
- Sortieren der Suchergebnisse nach Informationsgehalt
- Automatische und wiederholte Generierung eines Themen-Baums mit entsprechenden

ABER :

Es wird immer mehr versucht , mit Linkfarmen und ähnlichen Massnahmen die Linkpopularität zu beeinflussen.

Sie verliert damit immer mehr ihre ursprüngliche Bedeutung, nämlich dass ein Link von einer anderen Seite als Empfehlung anzusehen ist.

5. Literatur

[1] Soumen Chakrabarti, Byron E. Dom, David Gibson, Jon Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins — „Mining the Link Structure of the World Wide Web“

[2] Jon M. Kleinberg — „Authoritative Sources in a Hyperlinked Environment“