1 Informationssuche im Internet

Das Internet enthält eine große Menge unterschiedlicher und sehr heterogener Daten, die in unterschiedlicher Art und Weise aufbereitet sind. Der interessante Teil des Internet für uns hier ist das WWW.

Die Struktur des WWW besteht aus Seiten, die untereinander verlinkt sind, d. h. es gibt kein zentrales Verzeichnis. Um Informationen im Netz zu finden, muß ich mich also beginnend von einer mir bekannten Startseite weiterklicken, bis ich etwas interessantes gefunden habe.

Alternative: Suchmaschinen.

Eine Suchmaschine muß dabei zwei Sachen leisten:

- Beginnend von ihr bekannten Startseiten die Seiteninhalte katalogisieren und allen Verweisen folgen, die sie findet. Dies tun sogenannte Robots, das sind kleine Programme, die automatisch Webseiten laden und den in ihnen enthaltenen Links folgen.

  Damit baut die Suchmaschine einen Index auf, der all die Seiten umfaßt, die von den der Suchmaschine schon bekannten Seiten erreichbar sind (die transitive Hülle). Das bedeutet, daß nicht alle überhaupt existierenden Seiten auch durch Suchmaschinen gefunden werden können:

  - Seiten, die nicht von der Suchmaschine bekannten Seiten verlinkt sind, sind nicht erreichbar.
  - Der Anbieter einer Website kann angeben, daß bestimmte Seiten nicht indexiert werden sollen (normalerweise respektieren Suchmaschinen diese Hinweise).
  - Seiten, die nur durch Benutzereingaben erreichbar sind (wo man z. B. in ein Formular Begriffe eingeben muß, um den Inhalt der entsprechenden Seiten zu bekommen, wie dies bei öffentlichen Datenbanken und Katalogen häufig der Fall ist), können nicht automatisch gefunden werden.
  - Das Netz ist viel zu groß, um es schnell genug regelmäßig zu durchsuchen, die Suche hängt dem aktuellen Zustand immer hinterher.

- Eine Möglichkeit bieten, auf die so indizierten Seiten auch geeignet zuzugreifen, z. B. durch das Einsortieren in Verzeichnisse oder die Auswahl durch Suchbegriffe, d. h. zu *selektieren*. Ein wichtiger Faktor hierbei ist der *Überfluß* an Information, wodurch Selektion nach Stichwörtern etc. nicht mehr ausreicht. Gesucht sind die *passensten* Seiten zu einem Thema.

  Wenn ich also Informationen aufgrund von Stichworten suche, bekomme ich viel zu viele Seiten, die dieses Stichwort enthalten, diese muß ich also nach Relevanz sortieren. Hier ist also die Frage zu beantworten, welche Seiten die *wichtigsten* sind. Ein gebildeter Mensch könnte nun alle Seiten lesen und z. B. nach Qualität und Gehalt sortieren, das kann der Computer leider nicht, er muß das an äußeren Merkmalen festmachen:

- Seiten, in denen der Begriff in Überschriften vorkommt, sind wichtiger
- Seiten, auf denen der Begriff als Keyword angegeben ist, sind wichtiger
- Seiten, auf denen der Begriff häufig vorkommt, sind wichtiger
- Seiten, auf denen viel Text ist, sind wichtiger
- Seiten, die viele Links setzen, sind wichtiger
- Seiten, auf die viele andere Seiten verweisen, sind wichtiger
- Seiten, auf die andere Seiten von vielen unterschiedlichen Domains verweisen, sind wichtiger
- Seiten, auf die viele andere *wichtige* Seiten verweisen, sind wichtiger (das ist die PageRank Idee)
- ...

Die Frage bleibt, welche von diesen Maßen sinnvoll sind. Die ersten fünf Maße haben außerdem das Problem, daß der Seitenanbieter sie steuern kann, d. h. er kann eine hohe Plazierung erreichen, indem er die Seite entsprechend gestaltet. Die anderen Maße sind schwerer zu manipulieren (aber auch nicht unmöglich).

In der Praxis wird eine Kombination obiger Maße zusammen mit vielen weiteren benutzt.

Die Kernfrage hier ist also: „Wie finde ich in einer Überfülle an Informationen die für mich relevanten. Welche techniken kann ich da einsetzen?“

2 Repräsentation von vernetzten Dokumenten

Eine Menge vernetzter Dokumente ist technisch gesehen nichts anderes als ein Graph, d. h. eine Menge von Knoten (das sind die Dokumente), die durch gerichtete Kanten (das sind die Links) verbunden sind (Abb. 1). Möchte man nun selbst also die Wichtigkeit bestimmen, muß man das Dokumentennetzwerk natürlich im Programm irgendwie repräsentieren. Die einzelnen Seiten sind vielleicht Dateien auf der Festplatte oder Einträge in einer Datenbank, und in den Seiten stehen auch die Links.

Um auf diesem Dokumentennetzwerk arbeiten zu können, muß ich es in meinem Computerprogramm modellieren, d. h. durch irgendwelche Datenstrukturen repräsentieren und mit irgendwelchen Algorithmen darauf arbeiten:

- Die einfachste Idee besteht darin, einfach alle Seiten in einer Liste abzulegen und dann der Reihe nach durchzugehen. Jedes Dokument ist ein Datensatz (objektorientiert gesprochen ein Objekt), das aus den Dokumentinhalt, den Verweisen auf andere Dokumente und ggf. aus einer Bewertung des Dokumentes besteht.[1]

[1] Das Parsen einer HTML-Seite und das Extrahieren der interessanten Informationen ist übrigens nicht trivial, vor allem da viele Seiten kein korrektes HTML enthalten.

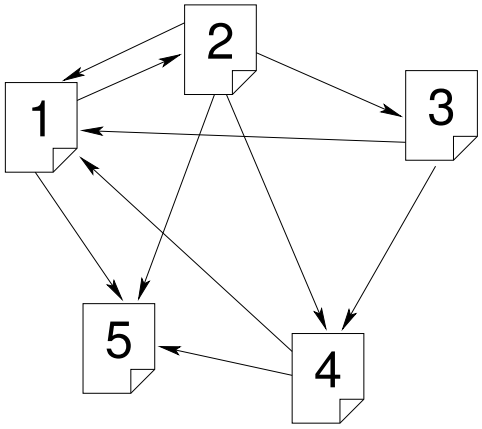

Abbildung 1: Ein Dokumentennetzwerk mit 5 Dokumenten

- Der „mathematischere“ Ansatz ist die Modellierung eines Graphen in form einer Matrix, wobei sich sowohl die Adjazenzmatrix als auch die Inzidenzmatrix anbieten:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 1 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \end{pmatrix}$$

  Dabei bedeutet eine 1 in Spalte x und Zeile y der Adjazenzmatrix, daß eine Kante vom Knoten Nr. x zum Knoten Nr. y existiert.[2] In der Inzidenzmatrix hingegen beschreibt jede Spalte genau eine Kante, die von dem knoten, der mit 1 gekennzeichnet ist zum Knoten, der mit -1 gekennzeichnet ist, verläuft.

Matrixdarstellungen besitzen rechnerisch große Vorteile, da man mit Matrizen sehr einfach rechnen kann.[3] Für unsere Aufgabenstellung ist wahrscheinlich die Adjazenzmatrix die einfachste Modellierung.[4]

Allerdings stehen dem Vorteil der mathematicshen Einfachheit auch Nachteile gegenüber: eine Adjazenzmatrix für n Dokumente hat n^2 Einträge, ihre Größe wächst also quadratisch, obwohl die meisten Einträge 0 sind.[5] Für kleine Beispielnetzwerke ist dies allerdings kein Problem.

Technisch können Matrizen je nach Programmiersprache entweder als mehrdimensionale Arrays oder als verschachtelte Arrals (Array in Array) realisiert werden.

[2] bitte anhand der Graphik selbst überprüfen!

[3] zu vielen Programmiersprachen existieren auch große Mathematikbibliotheken zur Matrizenrechnung.

[4] Grundsätzlich kann man natürlich auch völlig ohne Arrays auskommen und den einfachen Vorschlag der Liste realisieren (die, wenn man genauer hinsieht, eine Art versteckte Inzidenzmatrix darstellt), allerdings macht man sich damit das Leben unnötig schwer.

[5] auch hier gibt es spezielle Bibliotheken für die effiziente Arbeit mit dünnbesetzten Matrizen.

Die Aufgabenstellung hier ist also: „Wie modelliere ich ein Dokumentnetzwerk (einen Graphen) und wie programmiere ich auf Matrizen“.

3 Google's Erfolgsgeheimnis

Intensivere Beschäftigung mit dem Pagerank. Informationen zu Google habe ich auf `http://www.kinf.wiai.uni-bamberg.de/COM/Google.html` zusammengestellt. Ziel hier ist: „Wie funktioniert die Pagerankformel?“

4 PageRank Calculator

Mit obigen Fähigkeiten (Matrizenrechnung und PageRank-Papers) sollte es möglich sein, ein Programm zu schreiben, das ein *kleines* Dokumentennetzwerk einliest (entweder von Hand einzugeben oder aus einer Datenbank), und den Pagerank der Seiten berechnet. Das eigentliche Projekt.

Hier sind schöne Erweiterungen möglich, z. B. könnte man versuchen, die Webseiten der Schule oder der Stadtverwaltung zu nehmen und auf ihnen den Pagerank zu berechnen.[6]

5 Experimente zum Pagerank

- Wie ändert eine Änderung der Linkstruktur den Pagerank einer Seite?
- Kann ich Linknetze aufbauen, die sich gegenseitig den Pagerank erhöhen?
- Was hängt hier wie zusammen?

Hier sind auch Erweiterungen möglich:[7]

- Was passiert, wenn ich *gewichtete* Kanten zulasse, d. h. die Kante bekommt eine Bewertung (Seite 1 verweist stärker oder schwächer auf Seite 2, d. h. in Spalte 2 Zeile 1 der Adjazenzmatrix steht nicht mehr einfach eine 1 sondern eine 3 (stärker) oder $\frac{1}{4}$ (schwächer).
- Was passiert, wenn auch *negative* Bewertungen möglich sind (negative Kantengewichte)?

[6] Achtung! Das ist anspruchsvoller, da hier die Seiten aus dem Netz geholt werden und die Links aus ihnen extrahiert werden müssen. Hier ist geschickte Programmierung (oder Kenntnis der richtigen Tools) nötig.

[7] Achtung, das ist dann nicht mehr der Original PageRank! Frage: können Suchmaschinen sowas einsetzen? Wieso/wieso nicht? (Anmerkung: woher könnte die Suchmaschine die Gewichte bekommen?)

6 Präsentation

Ein Projektbericht, der die einzelnen Schritte des Softwareprojektes von der Problemanalyse bis zur lauffähigen Software und vor allem natürlich die Software selbst hier an der Uni Bamberg am Lehrstuhl in Form eines Besuchs der Gruppe vorstellt. hierzu gehören sowohl die Demonstration der Software als auch die Diskussion über die angewandten techniken, die aufgetretenen Probleme, die gefundenen Lösungen, die Erfahrungen aus den Experimenten etc.

Von unserer Seite gibt es dazu neben hoffentlich hilfreichen Kommentaren auch einen Bericht aus der Forschung mit reichlich Gelegenheit zu Fragen und Diskussion.