

SZEGEDI TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI ÉS INFORMATIKAI KAR

BÓLYAI INTÉZET
SZTOCHASZTIKA TANSZÉK

A PageRank algoritmus
SZAKDOLGOZAT

Készítette: Gáspár Tamás
Matematika BSc hallgató

Témavezető: Dr. Kevei Péter
Egyetemi docens
Sztochasztika tanszék

SZEGED, 2019

Tartalomjegyzék

1. Bevezető	1
1.1. A PageRank algoritmus története	1
1.2. PageRank program	1
2. Alapfogalmak és definíciók	2
2.1. Linkek, web és a lógó oldalak	2
2.2. A PageRank definíciója	3
3. A sajátértékprobléma	4
3.1. Linkmátrix és kapcsolata az egyenletrendszerrel	4
3.2. Az 1 sajátérték	5
3.3. A legnagyobb sajátérték	6
4. Konvergencia a PageRank vektorhoz	7

1. Bevezető

Dolgozatom témája a PageRank algoritmus, melynek legfőbb alkalmazási területe az internetes weboldalak bizonyos szempontok szerinti rangsorolása.

Ebben a fejezetben röviden ismertetem a PageRank algoritmus történetét és megemlítek egy PageRank számolásához használható, általam készített programot. A második fejezetben a PageRank algoritmus tárgyalásához szükséges alapvető fogalmakat és definíciókat vezetek be.

1.1. A PageRank algoritmus története

A PageRank algoritmust Larry Page és Sergei Brin alkották meg 1998-ban, a Stanford Egyetem hallgatóiként. Arra kerestek megoldást, hogy miként lehet az akkoriban robbanásszerű növekedésnek induló internetet weboldalait továbbra is rangsorolni, mivel látszott, hogy az akkor alkalmazott keresőmotorok erre már hamarosan nem lesznek képesek.

Úgy gondolták, hogy a rangsorolás alapja az internet hiperlink struktúrája kell hogy legyen. Feltették, hogy ha egy oldal linkel egy másikra, az kifejezi azt, hogy az oldal készítői megbíznak a linkelt oldalban, ezért az algoritmusukat úgy építették fel, hogy egy oldal fontossága (ezt szintén PageRanknak nevezik) attól függ, hogy mennyi és milyen fontos oldalak linkelnek rá.

Larry Page és Sergei Brin találmánya olyan jól alkalmazhatónak bizonyult, hogy Google néven saját vállalkozást alapítottak. A cég keresőmotorjának alapja máig a PageRank algoritmus, skálázhatóságát mutatja, hogy a Google ma már több mint 130 billió weboldalt indexel.

1.2. PageRank program

Dolgozatomhoz grafikus felhasználói felülettel ellátott asztali alkalmazást is készítettem, Java nyelven. Ez a program képes előre megadott oldalszámú webet különböző paraméterek alapján véletlenszerűen generálni, majd ehhez PageRankot számolni. A 150 oldalnál kisebb méretű webekhez tartozó mátrixokat meg is tudja jeleníteni.

A program és annak forráskódja is letölthető a következő oldalról:

<https://github.com/Gtomika/PageRank/releases/tag/v4.1>

A futtatáshoz a számítógépen telepítve kell hogy legyen a Java.

2. Alapfogalmak és definíciók

2.1. Linkek, web és a lógó oldalak

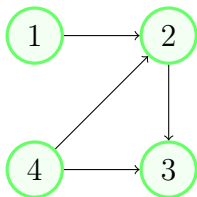
Ahhoz hogy PageRankről beszélhessünk, először a web fogalmát kell bevezetni.

1. Definíció: Linkhalmaz

Legyen V a weboldalak halmaza. Ekkor $L \subset V \times V$ **linkhalmazban** (v_1, v_2) $(v_1, v_2 \in V \text{ és } v_1 \neq v_2)$ pontosan akkor van benne ha v_1 linkel v_2 -re.

2. Definíció: Web

Legyen V a weboldalak halmaza, L pedig az ehhez tartozó linkhalmaz. Ekkor a **web** egy irányított gráf, melynek csúcsai V elemei, élei pedig L elemei, ahol ha $v_1, v_2 \in V$ és $(v_1, v_2) \in L$, akkor az él v_1 -ből v_2 -be mutat.



1. ábra. Egy 4 oldalból álló web

A linkhalmaz definíciójában lévő egyszerűsítő feltétel amely nem engedi, hogy egy oldal saját magára linkel, azért tehető meg, mert bár egy valós web esetén ez lehetséges, de a PageRank algoritmusba az ilyen típusú linkek nem számítanak bele, ez ugyanis lehetővé tenné, hogy egy oldal egyszerűen növelje a saját értékelését azzal, hogy sokszor linkel önmagára. Az ilyen linkeket ezért egyszerűen elhagyhatjuk.

Előfordulhat olyan web, ahol bizonyos oldalakról nincsenek kimenő linkek. Az ilyen oldalakat lógó oldalnak nevezzük (azokat a linkeket amelyek rájuk mutatnak pedig lógó linkeknek). A későbbiekben az ilyen oldalak problémákat okoznak, ezért megadok egy módszert, mellyel a ezeket az oldalakat el lehet tüntetni: minden lógó oldalt helyettesítünk egy az összes másik oldalra linkelővel.



Egy lógó oldallal (4) rendelkező web, és ugyanez a web, helyettesített oldallal.

A helyettesítés mögötti heurisztika az, hogy ha egy böngésző egy olyan oldalra érkezik, ahonnan a linkeken keresztül nem tud továbbmenni, akkor véletlenszerűen, egyenletes eloszlás szerint választ az összes oldal közül.

2.2. A PageRank definíciója

Egy oldal fontossága azon múlik, hogy mennyi oldal linkel rá, és hogy ezek milyen fontosak. A linkelő oldalak fontosságára azért van szükség, mert enélkül egy oldaltulajdonos tudná úgy növelni a weboldalának fontosságát, hogy rengeteg oldalt hoz létre, melyek mind linkelnek egymásra és a saját oldalára (ezt link farmnak nevezik [2]).

Az egy oldal fontosságát leíró pozitív valós számot is szokás az algoritmushoz hasonlóan PageRanknak nevezni. Egy webet el lehet úgy is képzelni, mint az oldalak demokráciáját ahol minden oldalnak szavazata van és ezt a szavazatot (és még a kapott szavazatokat is) továbbosztja úgy, hogy linkel a többi oldalra.

3. Definíció: Weboldal PageRankja

Legyen adott egy web, V az oldalak, L a linkek halmaza. Legyen $v_i \in V$ oldal PageRankja $r(v_i)$, az oldalról kimenő linkek száma pedig $|v_i|$.

Jelölje $B_i \subset V$ azon oldalak halmazát amelyen linkelnek v_i -re, azaz

$$B_i = \{v_j \in V : \exists l \in L, \quad l = (v_j, v_i)\}$$

Ekkor bármely v_i oldal PageRankját megkapjuk a következő módon:

$$r(v_i) = \sum_{v_j \in B_i} \frac{r(v_j)}{|v_j|}$$

A definícióban megjelenik az is, hogy egy oldalnak mennyi kimenő linkje van. Minél több oldalra linkel, annál kevesebbet fog számítani az ő linkjének értéke. Ez ellensúlyozza a már említett link farmokat.

A szummában szereplő oldalak egyikénél sem lehet a kimenő linkek száma nulla, mert mindegyik oldal eleme a B_i halmaznak, azaz legalább v_i -re linkelnek. Előfordulhat, hogy egy oldalra nincs egyetlen link sem. A definícióban ilyenkor egy üres összeg szerepel, azaz az oldal rangja 0.

A PageRank definíciója tehát rekurzív. Egy oldal rangjának meghatározásához minden rá linkelő oldal rangját ismernünk kell. Ha a webünk n db oldalt tartalmaz, akkor a definíció meghatároz n db lineáris egyenletet.

A cél az, hogy olyan algoritmust adjunk meg, amely egyrészt pontosan meghatározza minden oldal PageRankját, másrészt nagyon nagy n -re is hatékonyan működik, mind idő-, mind tárigény szempontjából.

3. A sajátértékprobléma

3.1. Linkmátrix és kapcsolata az egyenletrendszerrel

Egy web linkmátrixa az oldalak közötti kapcsolatokat reprezentálja mátrixos formában.

4. Definíció: Linkmátrix

Legyen adott egy web ahol az oldalak halmaza V . Ennek linkmátrixa legyen $A = (a_{i,j}) \quad i, j = 0, 1, \dots, |V|$, ahol

$$a_{i,j} = \begin{cases} \frac{1}{|v_i|}, & \text{ha } (v_i, v_j) \in L. \\ 0, & \text{egyébként.} \end{cases} \quad (1)$$

Tehát a linkmátrix sorai kifejezik azt, hogy egy oldal mennyi és melyik másik oldalakra linkel.



Példa: egy 4 oldalból álló web és linkmátrixa

Kapcsolat van az oldalak PageRankjának definíciójából kapott egyenletrendszer és a linkmátrix között. Ennek segítségével az egyenletrendszer felírható a következő mátrixegyenletként:

$$x = xA$$

ahol x az oldalak rangjait tartalmazó vektor. Ezt átalakítva a következőt kapjuk:

$$x = A^T x$$

Ebből mátrixegyenletből látszik, hogy a keresett vektor az A^T mátrix 1 sajátértékéhez tartozó sajátvektor. Természetesen egy sajátvektor bármilyen nem nulla skalárszorosa is sajátvektor ezért az egyértelműség kedvéért tegyük fel, hogy az általunk keresett vektor az, ahol a komponensek összege 1.

3.2. Az 1 sajátérték

Felmerül a kérdés, hogy minden A^T mátrixnak sajátértéke-e az 1. Ehhez elég belátni, hogy az A linkmátrixnak az 1 mindig sajátértéke, mivel egy mátrixnak és transzponáltjának a sajátértékei megegyeznek.

1. Tétel: Az 1 sajátértéke minden lógó oldalt nem tartalmazó web linkmátrixának

Bizonyítás. Legyen A egy tetszőleges, lógó oldalt nem tartalmazó, n oldalból álló web linkmátrixa. Ekkor az i . sorhoz ($i = 1, 2, \dots, n$) tartozó sorösszeg $|v_i| \times \frac{1}{|v_i|} = 1$, azaz minden ilyen mátrix sztochasztikus.

Legyen e az az n komponensű oszlopvektor, melynek minden komponense 1. Ekkor az

$$Ae = e$$

egyenlet teljesül, mert az eredményvektor i . komponense ($i = 1, 2, \dots, n$)

$$\sum_{j=1}^n 1a_{i,j} = 1$$

a mátrix sztochasztikus tulajdonsága miatt. Tehát bármely linkmátrixnak ez az e vektor sajátvektora, 1 sajátértékkel.

□

Ugyan bármely mátrixnak és transzponáltjának sajátértékei megegyeznek, de az nem igaz, hogy ezen sajátértékekhez tartozó sajátvektorok is azonosak, ezért a fenti tételben bevezetett e vektor nem biztos, hogy megoldása lesz a PageRank definíciója által meghatározott egyenletrendszernek. Ezzel a megoldás egyébként is használhatatlan az oldalak rangsorolásában, mivel minden oldalnak azonos rangot ad.

A lógó oldalakat tartalmazó webek esetén az 1 nem lesz mindig sajátértéke az A^T mátrixnak (lehetnek 0 összegű sorok, ami a fenti bizonyítást elrontja), azonban figyeljük meg, hogy ha elvégezzük a 2.1 alfejezetben megadott helyettesítést akkor a kapott mátrix már sztochasztikus lesz, és erre már érvényes az előbbi tétel.

Tehát beláttuk, hogy minden ilyen mátrixnak vagy sajátértéke az 1, vagy egy egyszerű helyettesítéssel el lehet érni, hogy az legyen.

3.3. A legnagyobb sajátérték

Ha sok oldalból álló webek esetén akarjuk meghatározni a rangokat, akkor a pontos megoldást adó módszerek nagyon lassúak, ezért közelítő, iteratív módszerre van szükség. Ideális választás a hatványiteráció, mert ebben az esetben elég tárolni mindössze a linkmátrix transzponáltját, és a PageRank vektor éppen aktuális iterációját a számoláshoz.

A hatványiteráció azonban a domináns sajátértékhez tartozó sajátvektorhoz konvergál. Ahhoz, hogy alkalmazhassuk az 1-hez tartozó sajátvektor megtalálására, be kell látni, hogy az 1 domináns sajátérték. A sajátértékek egyezése miatt itt is elég ezt a linkmátrixra belátni.

2. Tétel: Minden lógó oldalt nem tartalmazó linkmátrix bármely λ sajátértékre $|\lambda| \leq 1$.

Bizonyítás. A sajátértékek abszolút értékének felülről becsléséhez alkalmazzuk a Gershgorin körtételt. Eszerint a mátrix összes sajátértékének benne kell lennie legalább egy Gershgorin körben. A Gershgorin körtétel ugyan komplex mátrixokra is érvényes, de a linkmátrix csak valós elemeket tartalmazhat. A köröket meghatározó halmazok:

$$K_i = \{r \in \mathbb{R} : |r - a_{i,i}| \leq R_i\} \quad i = 1, 2, \dots, n$$

ahol R_i az i . sorösszeg és $a_{i,i}$ az i sorban lévő főátlóbeli elem.

Már korábban beláttuk, hogy a feltételeknek eleget tévő linkmátrixok sztochasztikusak, azaz a sorösszeg minden sornál egy. A linkhalmaz 2.1 alfejezetben szereplő definíciójában nem engedjük meg azt, hogy egy oldal önmagára linkeljen, ezért $a_{i,i}$ minden sorra 0.

Ezen állítások miatt a köröket meghatározó halmazok a következőre egyszerűsödnek minden sor esetén:

$$K_i = \{r \in \mathbb{R} : |r| \leq 1\} \quad i = 1, 2, \dots, n$$

Ebből következik, hogy bármely λ sajátértékre $|\lambda| \leq 1$ -nek teljesülnie kell. \square

Beláttuk, hogy az egy minden esetben sajátérték, és azt is hogy ennél (abszolút értékben) nem is lehet nagyobb sajátérték. Ebből azonban nem következik az, hogy domináns is lenne. Előfordulhat olyan eset is, hogy a -1 is sajátérték, vagy az 1 többszörös multiplicitással szerepel.

4. Konvergencia a PageRank vektorhoz

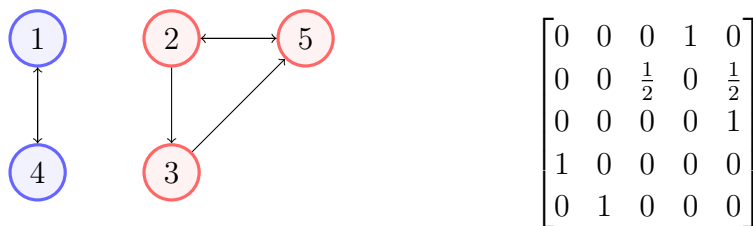
Amikor egy nagy webhez tartozó PageRank vektort szeretnénk közelíteni, fontos tudni, hogy az iteráció úgy fog viselkedni ahogyan várjuk.

A következő lehetőségeket kell ellenőrizni:

1. Biztosan konvergálni fog-e a hatványiteráció?
2. Ha igen, akkor az indítási helytől függetlenül mindig ugyanabba a vektorba?

Egyértelmű, hogy az indulási helytől független, biztos konvergenciát a 3.3 alfejezet végén említett szélsőséges esetek ronthatják el [1].

Ezek az eseten olyan webek esetén fordulnak elő, amelyek valójában több egymástól független (azaz nincsenek közöttük linkek) webből állnak össze [3].



Két egymással nem érintkező "szubwebből" álló web és linkmátrixa.

A fenti ábrán látható web linkmátrixának sajátértékei között szerepel a -1 és az 1 kétszeres multiplicitással.

Hivatkozások

- [1] ??? How google works: Markov chains and eigenvalues, May 2015.
- [2] Carl D. Meyer Amy N. Langville. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2012.
- [3] Tanya Leise Kurt Bryan. *The linear algebra behind Google*. 2016.