Hírportálok rovatainak modell alapú minősítése

Schlotter Ildikó

Tudományos diákköri dolgozat

2004.

Konzulensek: Gáspár Csaba, Távközlési és Médiainformatikai Tanszék
Lukács András, MTA SZTAKI Informatikai Kutatólaboratórium

Tartalomjegyzék

1. Absztrakt

A világháló egyre bővülő, nehezen átlátható rendszerében különböző témájú és minőségű dokumentumok, dokumentumcsoportok széles skáláját találhatjuk meg. Éppen ezért sokszor felmerül az igény egy adott oldal vagy oldalcsoport, rovat jellemzésére, minőségének vizsgálatára. Jó példa erre az elektronikus médiában részt vevő tartalomszolgáltatók internetes rovatainak minősége, melynek ismerete alapvető fontossággal bír az adott szolgáltató cég számára.

Az eddig megjelent publikációk zöme a felhasználók szempontjából vizsgálta a böngészés folyamatát, azaz az egyes felhasználói csoportok tipikus viselkedését próbálta modellezni, igen kis hangsúlyt helyezve a meglátogatott oldalak tulajdonságaira. Ezzel szemben az általunk e dolgozatban alkalmazott rovatalapú megközelítés erősen figyelembe veszi az egyes oldalcsoportok tulajdonságait. Ezen tulajdonságok közül a *minőség* nyilvánvalóan kulcsfontosságú. A dolgozat központi kérdése tehát, hogy lehetséges-e a rovatok modell alapú jellemzése úgy, hogy az mentes legyen a szubjektív minősítés hátrányaitól. Egy ilyen objektív jellegű minősítéstől elvárjuk például, hogy mutasson időbeli stabilitást, és ne függjön olyan jellemzőktől, mint az adott rovatban található dokumentumok száma.

Célunk egy olyan sztochasztikus modell megalkotása, mely a rovatok tulajdonságainak függvényében képes leírni a böngészés folyamatát. A modell felépítéséhez, majd teszteléséhez egy magyar hírportál internetes forgalmát rögzítő naplóállomány (weblog adatbázis) szolgált alapul. A megfelelően megalkotott modell lehetőséget ad arra is, hogy szimuláljuk egy adott jellemzőkkel bíró portálon történő böngészést. Elvárásaink szerint a szimuláció során előállított weblognak – a paraméterek megfelelő hangolása esetén – hasonlítania kell a valódi adatokra.

A dolgozatban áttekintjük a modellalkotás főbb kérdéseit és lehetőségeit, és megvizsgáljuk a felállított modell előnyeit és hátrányait. Szimulációt végzünk, és segítségével egy konkrét hírportál adatainak elemzésére alkalmazzuk a modellt; végül értékeljük a kapott eredményeket.

2. Bevezetés

A XXI. század információs társadalmában központi szerepet játszik a számítástechnika talán legváratlanabb vívmánya, a világháló. Mára a weben keresztül elérhető információk és szolgáltatások mindennapi életünk szerves részét képezik. Az internethasználat az egész világon és Magyarországon is folyamatosan terjed, az általa nyújtott lehetőségek kihasználása legtöbbünk számára természetessé válik.

A világháló megjelenésének egyik legfontosabb következménye a személyi és tömegkommunikációs eszközök skálájának kibővülése. A sokféle szolgáltatás közül a legfontosabbak között találhatók meg a különböző aktuális híreket, tudományos vagy szórakoztató információkat közlő internetes újságok, hírportálok. Ezek megjelenése alapvetően átformálta az emberek tájékozódási, újságolvasási szokásait.

Ebben a dolgozatban ilyen hírportálok oldalait, illetve az oldalakat közös téma alapján egy csoportba foglaló rovatokat, azok böngészésre tett hatásait vizsgáljuk meg. A vizsgálat célja az, hogy valamilyen módon jellemezni tudjuk ezeket a rovatokat, pontosabban, hogy meg tudjunk határozni számukra valamiféle objektív minőségi mércét. Egy ilyen mérce felállításához ismernünk kell a felhasználók viselkedését is, ehhez nyújtott segítséget az a különlegesen nagyméretű adattömeg, melyet egy nagyobb magyarországi híroldalt üzemeltető vállalat bocsájtott rendelkezésünkre. Ezek az adatok az egyes felhasználók böngészési adatait tartalmazzák egy elektronikus naplóállomány, ún. *weblog* formájában.

2.1. A témaválasztás indoklása

Az információs társadalomban a kommunikáció eszközévé váló internet egy komplex rendszer, mely nagy mennyiségű számítógép összekapcsolásával jött létre. Egyik legfontosabb tulajdonsága, hogy segítségével a legkülönfélébb adatok válnak a korábbi lehetőségekhez képest nagyságrendekkel megnövekedett mennyiségben elérhetővé. A világháló gigantikus mérete, a rajta fellelhető információk szinte végtelennek tűnő tárháza ugyanakkor nem csupán előnyt jelent. A weben megtalálható dokumentumok sokszor hibásak, hiányosak, legtöbbször pedig egyszerűen csak rossz minőségűek. Ebben a helyzetben ígéretesnek és hasznosnak tűnik egy olyan mérce felállítására, melynek segítségével lehetségessé válik az egyes dokumentumok, oldalcsoportok minőségének meghatározására, mérése.

Ebben a tanulmányban az internetes tömegkommunikációban jelentős szerepet játszó hírportálokkal foglalkozunk. Ennek oka, hogy egy internetes újság esetén nem csupán a felhasználók, azaz az olvasók kíváncsiak egy-egy oldal, vagy az

azonos témájú oldalakat összefogó rovatok minőségére, hanem maga az üzemeltető is. Mivel a magasabb minőség jobban kielégíti a felhasználók igényeit, ezért minél színvonalasabb egy híroldal által olvasásra felkínált rovat, annál többen fogják rendszeresen látogatni azt, így adva lehetőséget a vállalati profit növelésére a hirdetéseken keresztül. Így az adott vállalat számára mindenképpen fontos lehetőséget jelentene egy ilyen minőségi mérték felállítását segítő módszer megalkotása.

Fontos volt számunkra, hogy eddig még senki nem foglalkozott a két különböző, felhasználó- illetve tartalomorientált megközelítés összekapcsolásával. Úgy gondoljuk, hogy új, komplexebb szempontok megfogalmazásával lehetőségünk nyílik az eddigeknél összetettebb és valósághűbb modellek megalkotására. A bemutatásra kerülő modellezés célja tehát, hogy megfelelő modellillesztés esetén olyan szimulációkra adjon lehetőséget, melyek végül – a szimulált böngészés naplóállományán kívül – eredményül adják majd az egyes rovatok minősítését is.

2.2. Alapvető célok

A munka során elsődleges cél volt, hogy megalkossuk a hírportálokon található rovatok minősítésének módszerét.

Ha egy oldal vagy egy rovat minőségét irodalmi fejtegetések és esztétikai elemzések nélkül szeretnénk megállapítani, a következő lehetőségek merülnek fel:

- A dokumentumban található szövegek elemzése

  A természetes nyelvű szövegek analízisét segítő technikák egyelőre nem állnak olyan szinten, mely az egyes szövegek jellemzését lehetővé tennék. Ezen kívül egy oldal minőségét minden valószínűség szerint nem csak a rajta megtalálható szöveges tartalom jellemzi, így ez a módszer eleve nem vezethet kielégítő eredményre.

- A dokumentumban található metaadatok elemzése

  Ilyen megközelítést használ például a Google keresője, mely az egyes oldalakon található hiperlinkek számát, a hiperlinkek struktúráját figyelembe véve rangsorolja az egyes oldalakat. Ez a módszer általában véve kétségtelenül sikeres. Ugyanakkor ez a módszer csak meglehetősen nagy oldalcsoportokra, portálok egészére működik a belső hivatkozások jellemzően aránytalan mértékű előfordulása miatt.

  A mi célunk egy ennél finomabb szinteken is jól működő módszer kidolgozása.

- A dokumentumokon történő böngészések adatainak elemzése

> Ha olyan minősítést szeretnénk megalkotni, mely intuitív elképzeléseinkkel összhangban van, akkor célszerű megvizsgálni, hogy az adott oldal vagy rovat mennyire nyerte meg a közvélemény, azaz a „többség” tetszését. Ezt a felhasználók böngészéseiről gyűjtött adatok elemzésével tehetjük meg.

Fontos tehát leszögezni, hogy az általunk használt minőség fogalom nem objektív abban az értelemben, hogy ne függne az emberek véleményétől – ez azonban nem is feltétlen elvárás. Ez a minősítés csupán abban az értelemben lehet objektív, hogy megalkotásakor igen nagy, heterogén embercsoport - valójában akár az egész olvasói tábor - viselkedéséből vonunk le következtetéseket. Célunk tehát egy ilyen alapokra építő minősítési módszer létrehozása volt.

Mivel a minőség definiálásának lehetősége a felhasználók viselkedésében rejlik, ezért szükségünk van egy modellre, amely képes ennek leírására. Az általunk megalkotott modellnek tükröznie kell azt, hogy a böngészés során hozott emberi döntéseknél jelentős szerep jut az egyes rovatok minőségének is.

A létrehozott modellel lehetőségünk nyílik arra, hogy segítségével böngészéseket szimuláljunk. Megfelelő modellillesztés esetén egy konkrét hírportál rovatainak minősége a kinyert paraméterekből származtatható.

2.3. A dolgozat felépítése

A dolgozatban a témaválasztás indoklása, a szakirodalom bemutatása és az alapvető célok ismertetése után (2. és 3. fejezetek) tárgyaljuk, hogy hogyan készítettük el a felhasználói viselkedés egy olyan modelljét, melyben fontos szerephez jut az egyes hírrovatok minősége is. A modellépítés legfontosabb szempontjait, a problémákat és a rájuk adható válaszokat, végül a kész modellt mutatjuk be a 4. és 5. fejezetekben. Ezek után a 6. fejezetben a szimuláció, a modellillesztés, a paraméterek beállításának kérdéseivel foglalkozunk. Az implementálást és a módszer alkalmazását egy konkrét hírportál rovatainak vizsgálatára a 7. fejezetben foglaljuk össze. Végül az eredmények elemzése és összefoglalása történik meg a 8. fejezetben.

2.4. Köszönetnyilvánítás

Köszönöm Lukács Andrásnak és Rácz Balázsnak széleskörű matematikai tudásuknak és átfogó látásmódjuknak köszönhetően mindig nagyon hasznos ötleteiket, és főként kritikáikat. Szintén köszönettel tartozom Szepesváry Csabának az optimalizálás területén nyújtott segítségéért. Végül köszönöm Réczey Bálintnak a technikai problémák leküzdésében nyújtott segítségét.

Legfőképpen pedig hálás vagyok Gáspár Csabának az állandó támogatásáért, biztatásáért, és nem utolsó sorban rengeteg munkájáért, amivel ennek a dolgozatnak az elkészültéhez hozzájárult.

3. Korábbi eredmények ismertetése

Ebben a fejezetben összefoglaljuk a szakirodalomban megtalálható eddigi eredményeket, bemutatjuk az egymástól eltérő megközelítéseket. Ezek segítségével elhelyezzük a dolgozatunkat abból a szempontból, hogy mennyire illeszkedik egyik vagy másik uralkodó irányzat kereteibe, és ismertetjük saját célkitűzéseinket.

3.1. Eltérő megközelítési módok a szakirodalomban

Az internet robbanásszerű elterjedése maga után vonta egy új tudományág, az internetes adatbányászat kialakulását. Ennek célja, hogy minél több adatot elemezzen, értelmezzen és hasznosítson a világháló használatakor termelődő, vagy annak szerves részét képező nagymennyiségű adatból. A cél tehát bizonyos szempontból mindig azonos: a rendelkezésre álló adatokból kiinduló tudáskinyerés. Ennek a rejtett tudásnak a felderítésekor azonban sokszor lényegesen különböző szempontok, célok és technikák érvényesülnek.

A ma fellelhető publikációk, dolgozatok nagy része alapvetően négy csoportba sorolható, ezek mindegyike teljesen eltérő szempontokat vesz figyelembe:

1. tartalom analízis
2. struktúra analízis
3. fehasználói viselkedés elemzése
4. komplex webes adatbányászatot támogató rendszerek ismertetése

Mint látni fogjuk, a négy eltérő megközelítés más célokat állít maga elé, sokszor más-más adatok feldolgozásán alapul, és eltérő algoritmusokat és módszereket alkalmaz.

A négy típusból az utolsó inkább technológiai, mintsem tudományos szemléletű munkákat foglal össze, ezért ezzel nem foglalkoztam részletesebben.

A struktúra analízis során a cél valamilyen struktúra megtalálása a világháló dokumentumai között. Ennek a struktúrának a felfedésére leginkább a dokumentumokon megtalálható linkek, elérési útvonalak elemzése ad lehetőséget. Ilyen módon keresett összefüggéseket az egyes oldalak között Spertus [4] és Gibson [5].

Ugyanakkor a struktúra analízis nem csak a webes dokumentumok körében fellehető szerkezetek felkutatását jelenti. Az internet segítségével elküldött levelek vagy egyéb kommunikációs eszközök az internetfelhasználók közti kapcsola-

tokra utalnak. Ezek felderítése mind tudományos, mind üzleti szempontból jelentős feladat. Sok kutató ezt a problémát próbálta meg körbejárni kapcsolati hálózatok elemzésével, és erre adnak hatékonyan alkalmazható módszert Allst es Song [7], valamint Tuulos [6] is.

A tartalom analízis esetében a cél valamilyen módon osztályozni a webes dokumentumokat. Ez a fajta megközelítés tehát már sokkal közelebb áll az általunk alkalmazotthoz. Azonban fontos megjegyezni, hogy a legtöbb esetben nincs szó a dokumentumok minősítéséről, csupán azok oszályozásáról [11, 12], vagy feldolgozásáról [9, 10]. Sokszor ezek az elemzések valójában nem adatbányászati módszereket alkalmaznak, hanem a mesterséges intelligencia egyes eredményeit hasznosítják. Erre példát adnak azok a cikkek, melyekben olyan intelligens ágensek létrehozására adnak javaslatot a szerzők, melyek segítik a dokumentumok osztályozását [8, 13].

A legnagyobb, és rohamos iramban bővülő irodalma azonban a felhasználók viselkedéseit leíró, modellező és elemző módszereknek van. Az egyik legfontosabb probléma a felhasználók általános böngészési szokásainak elemzése. Sokan csupán statisztikai alapokon vizsgálják a felhasználói viselkedést, példa erre Catledge és Pitkow tanulmánya [1], melynek célja ajánlásokat tenni jól használható weboldalak készítésére. Sok kutató ad módszereket gyakori útvonalak kiszűrésére, és egyéb tipikus viselkedési mintákra [14, 15]. Ezeket az eredményeket aztán a felhasználói magatartás előrejelzésében [17], és az erre épülő adaptív, személyes profilt nyújtó weboldalak fejlesztésében hasznosítják [16]. Ezek mellett a szinte kizárólag csak statisztikai és adatbányászati alapokat használó módszerek körében megjelent néhány modell alapú megközelítés is, ezek közül a legjelentősebbek a rejtett Markov-modelleken alapuló kutatások, melyet Anderson és társai alkalmaztak [18].

3.2. Célkitűzéseink

Az előző fejezetben láthattuk, hogy sokan, sokféle szempontból vizsgálták már az internetes adatokon alapuló információkinyerés problémáját. A többféle megközelítés közül azt általunk választott témához egyrészt a webes dokumentumok osztályozásával foglalkozó irányvonal, másik oldalról pedig a felhasználói viselkedés modellezése áll közel.

Észre kellett vennünk, hogy mindeddig nem kombinálták ezt a két megközelítést, azaz a szakirodalomban nem ismert olyan eredmény, mely a felhasználó böngészésének modelljét arra használná, hogy végül egy összetartozó oldalcsoport minősítését megalkossa. Már önmagában a dokumentumok osztályzásán túlmutató minőség fogalom sem jelent meg eddig a tanulmányokban. Spiliopoulou és társai ugyan foglalkoztak egy hasonló fogalommal, a „sikerességgel”, azonban

ők szigorúan üzleti szempontokat vettek csak figyelembe, és kizárólag az elektronikus kereskedelemhez kötődő oldalak vizsgálatakor helyeztek hangsúlyt erre a jellemzőre [2].

Az általunk kitűzött cél tehát egy eddig felderítetlen terület problémáinak feltárása, melyben összekapcsoljuk egy webes dokumentum, vagy összetartozó dokumentumcsoport minőségének meghatározását a felhasználói magatartás elemzésével. Mindebben egy valós hírportál adatainak elemzése és az ez alapján végh ez vihető modellalkotás utáni szimuláció lehet segítségünkre.

4. A modellezés elméleti alapjai

Ebben a fejezetben áttekintjük a modell alapvető szereplőit, definiáljuk a használt fogalmakat és egységeket. Megvizsgáljuk a modell egyes szereplőinek alapvető jellemzőit, és a köztük lévő kapcsolatok legfőbb vonásait. Megadjuk a modell alkalmazhatóságához szükséges feltételeket, és megvizsgáljuk, hogy jogosake ezek a feltételezések. Végül rögzítjük a modellezni kívánt jelenségek körét és azok legalapvetőbb tulajdonságait.

4.1. A modellezés egységei

A modellel alapvetően az internetes böngészés folyamatát szeretnénk leírni. A böngészés lényegéből adódóan a modell két legfontosabb elemét egyrészt a böngészést végző felhasználók, másrészt az általuk meglátogatott oldalak, illetve azok csoportjai, a rovatok adják. A két fogalmat kapcsolja össze a böngészés folyamata, melynek kapcsán a modell időbeliségére is kitérünk, és definiáljuk a böngészési egységét. Lássuk, mit értünk pontosabban a fenti fogalmakon.

4.1.1. A felhasználó

A felhasználó fogalma a modellünkben lényegében nem takar mást, mint egy olyan személyt, aki a világhálón keresztül böngészője segítségével meglátogatja az általunk vizsgált hírportál oldalainak valamelyikét. Mielőtt azonban megelégednénk ezzel az egyszerű definícióval, meg kell említeni egy igen fontos tényt: a hírportált látogató emberek közel fele böngészésük során csupán egyetlen oldalt tölt le a portálról. Ez az oldal rendszerint főoldal, hiszen sokan csak a legfontosabb híreket szeretnék megnézni.

Ezt végiggondolva érdemes a felhasználók körét egy egyszerű szűréssel leszűkíteni azokra a látogatókra, akik böngészésük során több oldalt is letöltöttek a

hírportál oldalaiból, hiszen az egyetlen oldalkérést tartalmazó böngészéseket nyilván nem érdemes vizsgálni.

4.1.2. A rovat fogalma

A webes böngészés tárgyai az egyes internetes dokumentumok, oldalak. Azonban mivel ezekből túl sok van, és – főként hírportálok esetén – időben túl gyakran változnak, ezért vizsgálatunk tárgyának inkább az adott hírportál rovatait választottuk. Ez természetesen azt is jelenti, hogy amennyiben a felhasználó böngészés közben nem a hírportál rovatai közül tölt le valamilyen oldalt, akkor azzal egyszerűen nem foglalkozunk.

Rovat alatt az oldalak egy szervesen összetartozó csoportját értjük. Az összetartozást leginkább a téma azonos mivolta jelenti. Amennyiben minősíteni kívánjuk ezeket a rovatokat, mindenképpen fontos, hogy az egy rovatba sorolt dokumentumok minősége valóban, ha nem is azonos, de mindenképpen hasonló legyen. Mivel egy hírportál esetén a rovat nem pusztán tematikai, hanem szervezési egység is, ezért feltehetjük, hogy egy rovat oldalainak minőségét, stílusát és egyéb fontos jellemzőit kellően meghatározza az adott rovat elkészítéséért felelős szerkesztő illetve csoport.

Fontos leszögezni, hogy azzal, hogy a modell egységeként a rovatot definiáltuk, lemondtunk arról a lehetőségről, hogy az egyes dokumentumokkal önmagukban foglalkozzunk, és bármilyen módon jellemezzük őket. Tehát bár az általunk definiált rovat különálló oldalakból épül fel, ezen oldalakat a továbbiakban nem különböztetjük meg.

4.1.3. Az időbeli egység

A böngészés során a felhasználók és a rovatok kapcsolatát a böngészési sorozatok írják le. Egy böngészési sorozat („session”) tulajdonképpen egy adott felhasználótól egy adott időintervallumban beérkező letöltési kérések sorozata. Egy letöltésre vonatkozó kérés számunkra fontos paraméterei a letöltés ideje, a letöltendő dokumentum azonosítója, valamint annak a rovatnak az azonosítója, melyhez a lekért dokumentum tartozik. Vizsgálatunkban a dokumentum azonosítójára csupán azért van szükség, hogy két dokumentumról eldönthessük, vajon azonosak-e.

Lényeges, hogy mekkora időegységet választunk, azaz egy session milyen hosszú. A használandó egység kiválasztásánál két szempontot is figyelembe vehetünk:

- A letöltések sűrűsége

Minél gyorsabban követik egymást a felhasználó letöltései, annál biztosabb, hogy azok összefüggnek. Amennyiben ezeket az összefüggő letöltéseket nevezzük egy böngészési sorozatnak, akkor mindenképpen időben változó hosszú session-öket kapnánk eredményül.

Ez önmagában nem okozna gondot, az igazi problémát a határok megszabása okozza. Hány perc telhet el egy session két letöltése között? Hamar rájöhetünk, hogy a felhasználók sokszor több órára is megszakítják böngészésüket valamilyen más tevékenység miatt. Ennek befejezése után aztán folytatják a böngészést az őket érdeklő témákról. Ez tehát nagyban megnehezíti azt, hogy a letöltések közt eltelt idő alapján definiáljuk a session fogalmát.

- Periodicitás

  Ha valamilyen periodicitást tudnánk felfedezni a felhasználók viselkedésében, akkor az nyilván arra utalna, hogy egy periódus önmagában is teljesnek tekinthető. Egy ilyen zárt egységből már kinyerhetők lennének a böngészés jellemzői.

A legkisebb, várhatóan valóban periodikus egység a hét lenne, de a feldolgozásra kerülő adatok mennyisége (négy hétnyi adat) miatt inkább a napot választottuk alapegységnek. Ez nagyjából megfelel annak az elképzelésnek is, hogy néhány órás megszakítás után még folytathatjuk a böngészést, viszont nem valószínű, hogy különböző napok böngészései szoros egységet alkotnának.

Ezek az egynapos session-ök lesznek tehát a modellünk alapvető logikai egységei. Ez azt is jelenti, hogy egy adott felhasználó böngészéseit intervallumokra felosztva tároljuk. Ezeket a session-öket egy felhasználó böngészéseinek, vagy böngészési sorozatainak nevezzük. „Felhasználói sorozat” alatt egy felhasználó minden session-jének összefűzésével kapott letöltések sorát értjük.

Kérdés még, hogy szükséges-e a napnál rövidebb időegységet is definiálni, célunk-e az egynapi böngészés időbeli szerkezetének vizsgálata. Egy ilyen elemzésnek nyilvánvalóan lenne értelme, hiszen sok kutatás foglalkozik azzal, hogy időben hogyan oszlik meg az egy nap alatt lebonyolított internetforgalom.

Mi azonban nem ezt szeretnénk vizsgálni, hiszen a rovatok minőségének felderítésekor feltehetően nem játszik túl nagy szerepet az egyes letöltések pontos időpontja.

4.2. A modell szereplőinek vizsgálata

Vizsgáljuk, meg részletesebben a modell egyes szereplőit.

4.2.1. Felhasználók

Fontos egyszerűsítés, hogy a modellben a felhasználók feltételezéseink szerint homogének. Mivel a valóságban a böngészést végző emberek természetesen közel sem jellemezhetők homogén tulajdonságokkal, ezt a feltételezést indokolnunk kell.

Valójában több érv is amellett szól, hogy a felhasználók homogenitásának feltételezése jogos. Tekintsük át ezeket.

a) Valójában nem feltételezzük, hogy a felhasználók homogének, a modellben viszont egy minden szempontból „átlagos" felhasználóval számolunk. Ez a megközelítés azért jogos, mert a böngészést végző emberek nagy száma miatt nagy biztonsággal alkalmazhatunk statisztikai módszereket, így a modellben szereplő homogén, de statisztikailag átlagos tulajdonságokat mutató felhasználók sokasága a portál szempontjából egyenértékű lesz a valóságban inhomogén felhasználói halmazzal. Erre az átlagos felhasználóra tehát gondolhatunk úgy is, mint a sokféle valós felhasználó szuperpozíciója.

b) Előfeldolgozás segítségével elérjük, hogy a mérésekben csak a felhasználók egy többé-kevésbé homogén csoportja szerepeljen. Ekkor viszont a modell érvényességi köre is leszűkül ezekre a felhasználókra, vagyis a végcélként meghatározandó minősítése a rovatoknak is csak egy szűkebb kör véleményét fogja tükrözni.

   A homogenitást biztosító előfeldolgozást elvégezhetjük valamilyen klaszterező eljárással, vagy csoportosíthatjuk a felhasználókat az általuk letöltött oldalak száma alapján, az eloszlás ferdesége miatt például logaritmikus skálát használva. A csoportosítás után a releváns felhasználók immár jóval homogénebb körével dolgozhatunk tovább.

c) Az előző pontbeli megközelítést kombinálhatjuk a statisztikai sokaság gondolatára alapuló szuperpozíció elvével, azaz alkalmazhatunk egy kevert modellt is. Ekkor az előfeldolgozás során elvégzett csoportosítás után minden – egyenként homogénnek tekintett – csoportra illesztjük a modellt, majd az utófeldolgozás során egyesítjük a kapott eredményeket.

Alapvetően a legelső, tehát a szuperpozíció elvén alapuló ötletet alkalmazzuk a modellben. Ugyanakkor a harmadik lehetőség egyfajta ellenőrzésként szolgálhat, hiszen ha különböző felhasználói csoportokat vizsgálva hasonló minősítési sorrendet kapunk a rovatokra, akkor ez azt mutatja, hogy a modellünk kifejező ereje nagy.

Amennyiben nem ezt tapasztaljuk majd, azaz az eltérő tulajdonságú felhasználói csoportok viselkedéséből kinyert minősítések jelentősen különböznek, úgy annak oka lehet az is, hogy a különböző felhasználók ténylegesen más-más preferenciákkal rendelkeznek. Ha ez így van, akkor kérdéses egy olyan minősítés megalkotása, mely mindegyik csoport vélemyényét tükrözi. Az viszont még ekkor is igaz marad, hogy ha az összes felhasználó véleményének szuperpozícióját vizsgáljuk, annak mindenképpen az egyes csoportok által megtestesített vélemények között kell elhelyezkedni. Ez az eset is lehetőséget ad a modellezés helyességének ellenőrzésére.

4.2.2. Rovatok

A rovatokat összetartozó oldalcsoportokként definiáltuk az előző szakaszban. Magától értetődő módon a hírportál esetében a köznapi módon értelmezett rovat fogalma, amely valójában egy szervezeti egységet is takar, megfelel ennek a definíciónak.

Problémát okozhat azonban az, hogy ezek a rovatok egy hierarchikus rendszer részei, így felmerült a kérdés, hogy vajon csak rovatokat, vagy alrovatokat is vizsgáljunk, valamint hogy minden rovattal foglalkozzunk-e. Ennek a kérdésnek a magától értetődő megoldását az a feltevés adja, hogy a vizsgálandó rovatokat egyszerűen a feladat bemenetének tekintjük. A rovatok kiválasztása tehát minden esetben a probléma keretein kívül eső, egyébiránt nem túl bonyolult feladat marad.

A rovatokon belül az oldalakat egymástól nem különböztetjük meg. Mégis szükség van néhány, az oldalak szintjét érintő előszűrésre. Ezek célja, hogy csak a számunkra releváns, a vizsgálódásra érdemes letöltéseket vizsgáljuk. A használt oldalszintű előszűrések:

- Nem létező, vagy értelmetlen (például hibaüzenetet tartalmazó) oldalak kiszűrése.
- A főoldal túlzott látogatottsága miatt az arra érkező kéréseket kiszűrjük az adatok közül. Az egyes rovatok főoldalaira vonatkozó kéréseknek viszont már van jelentős információtartalma, így úgy döntöttünk, hogy azokat bevonjuk a vizsgált oldalak körébe.
- A böngésző programok automatikus frissítéséből adódó – az adott oldaltól függő időközönként megtörténő – ismételt oldalkéréseket szintén töröljük.

4.2.3. Böngészési sorozatok

Az adatbázisunk logikai egysége a session, amely napokra és felhasználókra lebontva tartalmazza a böngészés során lekért dokumentumok listáját. Lássuk,

mit tartalmaz tehát az adatbázis egy rekordja, mely megfelel egy dokumentum letöltésének:

1. UserID: a felhasználó egyedi azonosítója
2. SessionID: a session azonosítója
3. ColoumnID: a rovat azonosítója
4. DocID: a letöltött dokumentum egyedi azonosítója
5. TimeStamp: a böngészés időbélyege

Az eddig elmondottak alapján a rekord öt mezője közül néhány külön figyelmet érdemel. Az egyik a dokumentum egyedi azonosítója, melyre – mint azt a rovatok definiálásánál kikötöttük – valójában nem lenne szükség, hiszen a dokumentumokat nem különböztetjük meg egymástól. A másik az időbélyeg, amiből elméletileg csak az aktuális nap sorszámára lenne szükség. Mindkét információt indirekt módon használjuk fel, például mind a pontos letöltési időpontra, mind a letöltött dokumentum azonosítójára szükség van, hogyha ki szeretnénk szűrni a böngészők által automatikusan elvégzett frissítésekből adódó kéréseket.

A felhasználó azonosítója szintén csak arra szolgál, hogy meg tudjuk különböztetni egymástól a különböző felhasználók böngészési sorozatait. Ezen kívül semmit sem tárolunk az egyes felhasználókról, ami összhangban van a különféle adatvédelmi elvárásokkal.

A session és a rovat azonosítója nem igényel külön magyarázatot.

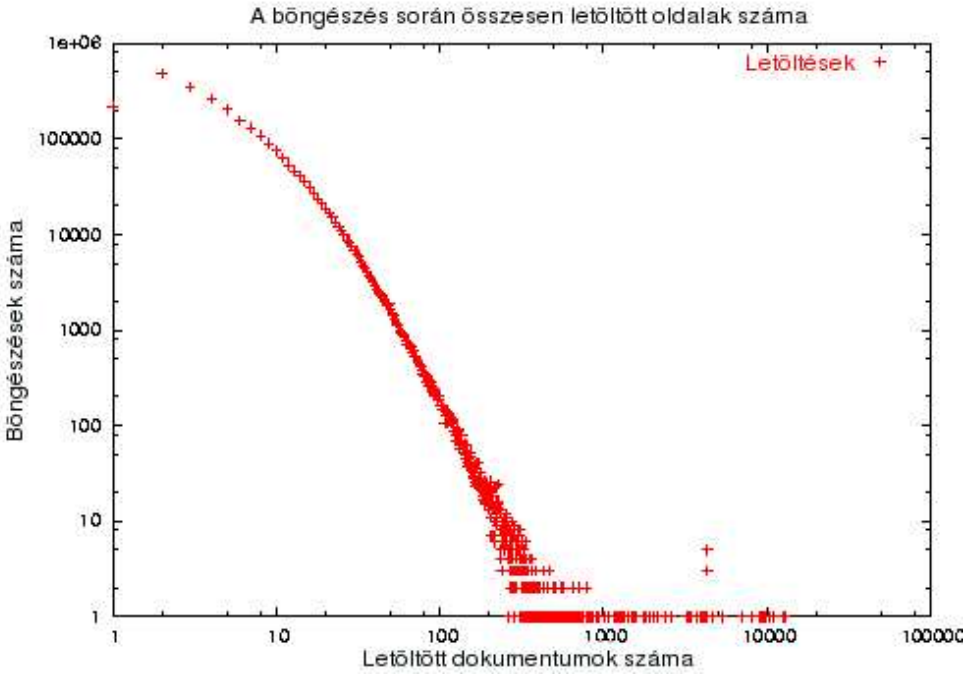
4.3. A modellezni kívánt jelenségek megválasztása

Ennek a szakasznak a célja, hogy sorra vegyük azokat a tényezőket, melyek a felhasználónak a böngészés során meghozott döntéseiben szerepet játszanak. Vegyük sorra, melyek azok a jelenségek, melyek segítenek abban, hogy felfedjük a felhasználói viselkedés mozgatórugóit. Azt szeretnénk, hogy a modell tükrözze a következő jelenségeket:

- Felhasználó fáradása session szinten:

  Ha megvizsgáljuk azt a hisztogramot, ami a felhasználók számát mutatja az általuk egy nap alatt összesen letöltött oldalak számának függvényében (1. ábra), láthatjuk, hogy ez a függvény meredeken csökkenő, hatványfüggvény lefutású. Ez összhangban van azzal az elvárásunkkal, hogy a böngészés során az ember folyamatosan fárad.

1. ábra. Dokumentum – felhasználó hisztogram



Ha szeretnénk modellezni ezt a jelenséget, akkor olyan modellt kell adni, amely garantálja, hogy a felhasználónak az általa eddig letöltött oldalak számának növekedésével egyre inkább csökken az esélye arra, hogy új oldalt töltsön le.

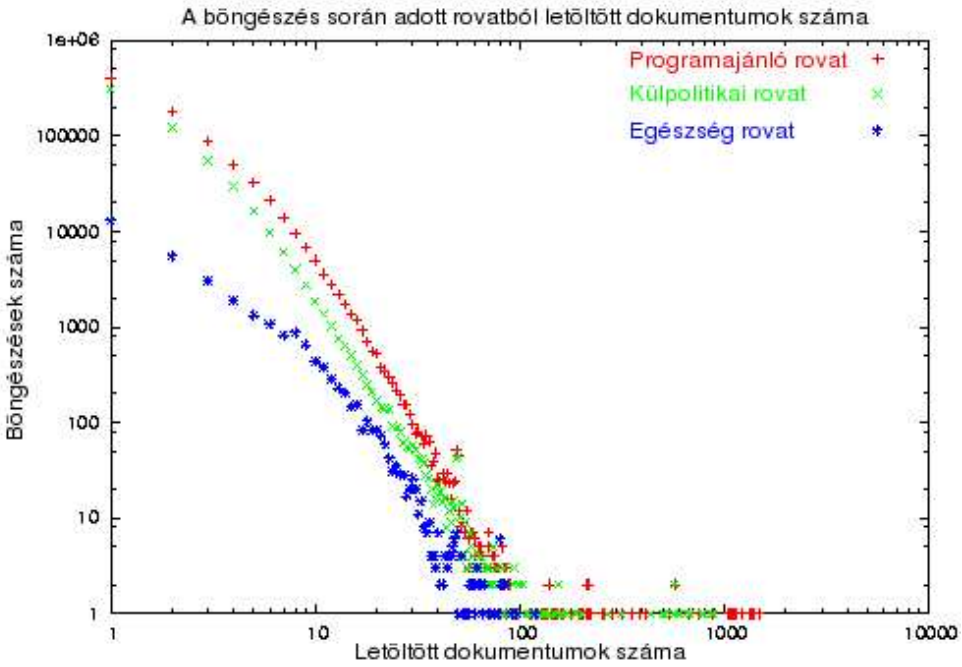
Természetesen ezt a növekvő esélyt a böngészés befejezésére nem kizárólag a letöltött oldalak száma befolyásolja: egy érdekes hír vagy egy jól megírt cikk nyilvánvalóan megnöveli a további böngészés esélyét. Csupán annyit állítunk, hogy a letöltött oldalak száma – azonos körülmények mellett – növeli a böngészés befejezésének esélyét.

- Felhasználó fáradása rovat szinten:

  Az elfáradás jelensége nem csak napi szinten figyelhető meg, hanem rovat szinten is, azaz minél több dokumentumot tölt le valaki egy rovatból, annál nagyobb az esélye, hogy csökken az érdeklődése a rovat iránt. Ezt mutatja a 2. ábra is.

  Ezt a józan ész alapján tett feltételezést könnyen alátámaszthatjuk, ha megfigyeljük, hogy az egy rovatból való letöltések száma tipikusan hogyan változik. Azt látjuk, hogy ez a hisztogram is hatványfüggvény lefutású, azaz egy adott rovat esetén sokkal valószínűbbek az adott rovatból csak kevés

2. ábra. Dokumentum – felhasználó hisztogram adott rovatokra



letöltést tartalmazó böngészések.

Ez azt jelenti, hogy minél többet böngészett már a felhasználó, általánosságban annál esélyesebb, hogy abbahagyja. Persze ezt, akárcsak a rovatbeli fáradást, befolyásolhatja az aktuális, sőt session szinten esetleg néhány, a múltban meglátogatott rovat milyensége is.

- A rovat frissülésének szerepe:

  A rovatok böngészését befolyásolja a rovatban található új oldalak száma is, és ezt szintén tükröznie kell a modellnek. Természetesen az, hogy egy felhasználó egy adott pillanatban hány új oldalt talál egy rovatban, sok tényezőtől függ. Függ attól, hogy a felhasználó hány oldalt látogatott már meg a rovatban, mikor nézte meg a rovatot utoljára, és függ attól is, hogy milyen időközönként frissítik a rovatot. Egy abszolút valósághű modellben mindennek szerepelnie kéne.

  A megvalósíthatóság érdekében persze mindenképpen kompromisszumot kell kötnünk az egyszerűség és a modell valóságot leíró ereje között. Vizsgáljuk meg a legfontosabb altenatívákat.

  a) Ha nem akarjuk, hogy a modellben egy felhasználóról számon kelljen tartani annak múltbeli böngészéseit is, akkor valahogy az adott fel-

használó múltját nem ismerve kell becsülnünk az adott rovatban számára megtalálható friss oldalak számát.

Erre alkalmas módszer lehet, hogyha egy, a weblogból számított statisztika alapján megvizsgáljuk a friss lapok számának eloszlását a különböző rovatba való belépések idején az egyes felhasználókra nézve. Ez alapján minden esetben, amikor egy felhasználó belép egy rovatba, azaz akár aznap először, vagy esetleg valamilyen más rovat böngészése után letölt róla egy oldalt, mindig kisorsoljuk a modellben, hogy számára éppen hány új oldal található a rovatban. Ekkor persze a sorsolás során nem játszik szerepet az adott felhasználó múltja.

b) Tárolhatjuk egy felhasználóról azt, hogy mikor böngészett utoljára, és a rovatokról pedig tárolhatjuk azt az eloszlást, hogy egy nap alatt hány új oldal jelenik meg bennük. Így a két szám szorzatával becsülhetjük egy adott napon felhasználó által frissnek látott oldalak számát.

   Természetesen ekkor a modell működésekor valamilyen kezdeti értékről kell indítanunk a szimulációt, valamint szimulálnunk kell azt is, hogy mely napokon böngészik a felhasználó, és mely napokon nem.

c) Kiegészíthetjük a modellt úgy is, hogy a friss oldalak számát nem mindig a nulláról számoljuk újra, hanem inkrementálisan az időközben a rovatba felkerült friss oldalak számát hozzáadjuk a felhasználó által eddig nem látott friss oldalak számához.

d) A legbonyolultabb modellben a fentieket még kiegészíthetjük egy elavulási rátával, mely a friss oldalak számát folyamatosan csökkenti abban az ütemben, melyben az egyes dokumentumok lekerülnek az elérhető oldalak listájáról.

Az a) pontban kifejtett, a felhasználó múltját számításba egyáltalán nem vevő megoldás túlságosan elnagyolt. A második már sokkal közelebb áll a valósághoz. A harmadik pontban leírt inkrementális modell azért nem megfelelő, mert ekkor minden olyan dokumentum, amely valaha új volt, de a felhasználó nem nézte meg, a továbbiakban is újként lesz számon tartva.

Ez adja a negyedik, inkrementális, de a dokumentumok elavulásával is operáló modellt. Ez elméleti szempontból jobb, mint a b) pontban leírt – eleddig leginkább megfelelő – modell, azonban nagy hátránya, hogy az elavulási rátát nagyon nehéz megmérni, vagy akár csak megbecsülni is. Ennek oka, hogy az elavulás foka – az adott témától függően – egészen szélsőséges határok között mozoghat. Jó példa erre az aktuális politikai témájú cikkek, valamint a különféle ismeretterjesztő oldalak ellentéte.

A fenti indokokat figyelembe véve végül úgy döntöttünk, hogy a második megoldás szerint, azaz a rovatokra jellemző naponkénti frissülési rátából valamint a felhasználó utolsó böngészésének időpontjából számoljuk a rovat aktuális frissességét.

Ehhez a szimuláció során az új oldalak számának kezdeti értékeket becsülni kell, erről a 7.3.1 részben írunk bővebben. Gondoskodni kell arról is, hogy az egyes felhasználók böngészései (session-jei) „kellő időközökben" kövessék egymást. (Ez egy mérhető eloszlás lesz.) Szükség van még az egyes oldalak új voltának megállapítására is, ezt az oldal letöltési rátájának hirtelen felszökése fogja megmutatni.

Mindhárom most felsorolt jelenségben tükröződik az egyes rovatok minősége is. Nyilvánvalóan azt, hogy egy felhasználó milyen gyorsan fárad el vagy unja meg a böngészést (akár session, akár rovat szinten vizsgálódunk) nagyban befolyásolja, hogy milyen az eddig általa meglátogatott rovatok minősége.

A rovat frissülése szintén azon tényezők egyike, melyek befolyásolják, hogy a felhasználók hány oldalt töltenek le az adott rovatból. Amennyiben tehát figyelembe vesszük ezt a tulajdonságot, akkor ettől függetlenül tudjuk majd megállapítani a rovat minőségét. Ha viszont nem használjuk a frissülési ráta fogalmát, akkor a minőség implicit módon magába foglalja majd ezt a tulajdonságot is, azaz a sűrűn megújuló rovatok minősége jobb lesz, mint a hasonló színvonalon megírt, de ritkábban frissülő rovatoké.

5. A modell megalkotása

Elsődleges célkitűzésünk a böngészésben szerepet kapó oldalak, jelen esetben egy internetes hírportál oldalainak illetve rovatainak vizsgálata. Azt várjuk, hogy egy ilyen vizsgálat eredményeként megkapjuk a vizsgált rovatoknak egy minősítését.

Ehhez elsősorban egy olyan modellre van szükségünk, amely jellemezni tudja az adott rovatokat, és ezen jellemzőkre alapozva jól leírja a rovatokon történő böngészés folyamatát. Mivel ezt a folyamatot célszerű sztochasztikus folyamatnak tekintenünk, így nyilván a modellnek is tartalmaznia kell sztochasztikus elemeket.

Amennyiben rendelkezésünkre áll egy ilyen elvárásoknak megfelelő modell, akkor képesek vagyunk a modell jóságától függő mértékben szimulálni a valós böngészést. Reményeink szerint így az egyes rovatok minősítésére is lehetőséget kapunk. Ebben a fejezetben a megalkotandó modellel foglalkozunk, míg a következőben azt mutatjuk meg, hogy hogyan lehet szimuláció, és az arra épülő – op-

timalizáló eljárást is alkalmazó – modellillesztés segítségével kinyerni a rovatok minőségét, amennyiben ismerjük a böngészési adatokat.

5.1. Elvárások a modellel kapcsolatban

A modellel szembeni elvárásainknak alapvetően két csoportját definiálhatjuk. Egyrészről léteznek olyan elvek, melyeket minden használható modell megalkotásakor érdemes követni, másrészt a modellezendő területről alkotott előzetes elképzeléseinkből és információinkból szintén adódnak elvárások, melyeket a modellnek teljesítenie kell. Vizsgáljuk meg a következőkben ezeket az elvárásokat részletesebben.

5.1.1. Általános elvárások

Minden jól használható modellnek eleget kell tennie a következőknek:

- Ellenőrizhetőség

  Biztosan elvárjuk egy modelltől azt, hogy ellenőrizhető legyen. Ez azt jelenti, hogy található olyan módszer, melynek segítségével meg tudunk adni egy mértéket, ami alkalmas annak a jellemzésére, hogy a modell mennyire közelíti jól a valóságot.

  Látni fogjuk, hogy ezt az általunk választott modellnél többféleképpen is meg lehet tenni. A szükséges mérték megtalálásához a statisztikaelmélet és a valószínűségi alapú modellezés adja majd az alapot.

- Kiszámíthatóság

  A modellezés során a modell helyességének mérésére használt érték igen gyakran valamilyen összehasonlításon alapul. Amennyiben ez a helyzet, akkor nagyon fontos előnyt jelent egy modell esetén az, hogyha az ehhez az összehasonlításhoz szükséges jellemzők közvetlen számíthatók a modellből. Ekkor minden típusú modellillesztés vagy optimalizálás a modell keretein belül igen hatékonyan elvégezhető.

  Előfordul azonban, hogy a modellből nem számolható ki egyértelműen az összehasonlítás tárgyát képező érték. Erre kézenfekvő példát adnak azok a modellek, melyek a jóságmérték megállapításához valamilyen statisztikai, a véletlenen is múló jellemzőt használnak. Persze attól, hogy a modellből nem számolható közvetlenül annak jósága, még mérhetőek lehetnek ezek a jellemzők is, például szimulációs eszközök alkalmazásával.

Ebben a feladatban ilyen mérhető jellemzők lesznek a különféle alapvető, a böngészést jellemző statisztikák, például az egy felhasználó által átlagosan letöltött oldalak száma, vagy részletesebb szinten egy adott rovat nézettségének lecsengésének meredeksége a rovaton belül letöltött oldalak számának függvényében. Mint látni fogjuk, éppen ilyen statisztikai jellemzők miatt lesz feltétlen szükség szimulációra.

- Értelmezhetőség

  Általános elvárás még, hogy a modellben használt feltételezések indokolhatóak legyenek, és a modellben megjelenő paraméterek intuitív módon értelmezhetőek legyenek. Az egyes modelljelöltek vizsgálata során az átláthatóság és a kisebb hibázási lehetőség érdekében érdemes az egyszerűbb modelltől a komplexebb felé haladni.

5.1.2. Területspecifikus elvárások

Ha végiggondoljuk, hogy milyen előzetes feltételezéseink vannak a böngészésről, valamint általában véve a minőség fogalmáról, akkor a következő elvárásokat támaszthatjuk a modell elé:

- Időbeli stabilitás:

  A modellillesztés során bizonyos paraméterek ne mutassanak erős változásokat rövid távon. Egy paraméter jelentős ingadozása valójában azt jelzi, hogy az adott paraméter nem ír le lényeges tulajdonságot. Ennek a kijelentésnek az az apriori feltételezés ad alapot, hogy sem a modellünk, sem a benne résztvevő szereplők nem változnak gyorsan. Ennek a feltételezésnek a létjogosultsága könnyen belátható, hiszen sem a felhasználók szokásai, sem maguk a rovatok nem rendelkeznek gyorsan változó jellemzőkkel.

- Térbeli stabilitás:

  Térbeli stabilitás alatt azt értjük, hogy a modellnek érzéketlennek kell lennie az aktuálisan vizsgált felhasználók körére. Azaz ha a felhasználóknak csak egy véletlenszerűen kiválasztott hányadát tekintjük, akkor azok viselkedését is jellemezze jól a modell, mindaddig, míg számuk elegendő a sztochasztikus megközelítéshez.

  Itt természetesen nagyon fontos a véletlenszerű kiválasztás, hiszen biztosan lehet találni olyan felhasználókat, akik akár viselkedésükben, akik ízlésükben jelentősen eltérnek valamely irányba az átlagostól. Ekkor rájuk alkalmazva a modellt bizonyára eltérő eredményeket kapnánk.

A térbeli stabilitás fogalmát nem csak a felhasználók oldaláról lehet megközelíteni, hanem a rovatokéról is. Ekkor azt az előzővel analóg elvárást kapjuk, hogy amennyiben csak a rovatok egy véletlenszerűen választott részhalmazát vizsgáljuk a böngészés elemzése során, attól egyrészt ne változzanak jelentősen a kapott minősítések az egyes rovatokra, valamint ne változzon meg jelentősen a felhasználók viselkedésének jellege sem.

- Rovatmérettől való függetlenség.

  Ez egy magától értetődő elvárás: a rovatok mérete, azaz a hozzájuk tartozó webes dokumentumok száma ne befolyásolja nagy mértékben a rovat minőségét.

- Rovat minőségének függetlensége a téma népszerűségétől.

  Ez alatt azt értjük, hogy a rovathoz kötődő téma popularitásától lehetőleg független legyen a modellből adódó minőség értéke. Ez az elvárás egyáltalán nem triviális, ráadásul megvalósulása sajnos nagyon nehezen ellenőrizhető. Ennek oka, hogy a téma népszerűségét nem lehet egzakt módon megmérni.

  Első megközelítésben úgy tűnik, könnyen adható lenne pontos definíció a popularitásra, például megadhatjuk a téma népszerűségként azt, hogy hányan látogatják az adott rovatot összesen. Azonban észre kell vennünk, hogy valójában a látogatók számát a téma népszerűségén kívül – legalábbis hosszú távon – mindenképpen befolyásolja az adott rovat minősége is. Így tehát ez a definíció nem alkalmas a rovat témájának népszerűségének mérésére.

5.2. A modell eseményei és azok paraméterei

A felhasználó viselkedését tekinthetjük úgy, mint egy sztochasztikus folyamat, melynek valószínűségi változói tulajdonképpen azt adják meg, hogy mikor – ez alatt valójában nem a valós időt értjük – és milyen rovatba tartozó oldalt tölt le a felhasználó. E folyamat során a böngészést végző felhasználó a sztochasztikus modellből adódó valószínűséggel hoz meg bizonyos döntéseket, és tesz meg adott cselekvéseket.

Ebben a szakaszban áttekintjük, hogy milyen helyzetekben mik a felhasználó által választható cselekvések, és hogy milyen tényezők befolyásolhatják a felhasználó választását a lehetséges alternatívák közül.

5.2.1. A sztochasztikus böngészés eseményei

Vegyük sorra, milyen lehetőségei vannak a felhasználónak, azaz mik a modell lehetséges eseményei.

Böngészés kezdete Amennyiben a modellezés során figyelembe szeretnénk venni a rovatok tartalmának felfrissülését is, akkor tudnunk kell azt megbecsülni, hogy egy adott napon a felhasználó – saját múltjától függően – hány, számára újnak ható oldalt találhat a rovatban. Ekkor a felhasználó viselkedésének leírásához hozzátartozik az is, hogy mely napokon kezd meg egy böngészési sorozatot, és mely napokon nem. Ezt felfoghatjuk úgy is, hogy a felhasználó minden nap döntést hoz arról, hogy elkezdje-e böngészést.

Ha a döntés igen, akkor ezt az eseményt nevezhetjük a böngészés kezdetének.

Kezdeti rovatba ugrás Ha a felhasználó megkezdte a böngészést, akkor nyilvánvalóan azt is el kell döntenie, hogy melyik rovatot látogatja meg először. Ezt a lépést nevezhetjük kezdeti rovatba ugrásnak.

Ezután a böngészés során minden dokumentum letöltését követően választás elé kerül a felhasználó. Ennek a választásnak a kimenetelétől függően a következő három esemény egyike következik be:

Rovatban maradás Ekkor a felhasználó marad az aktuális rovatban, azaz a következő letöltött oldal ugyanabból a rovatból fog kikerülni, mint amelyikből az előző letöltés történt.

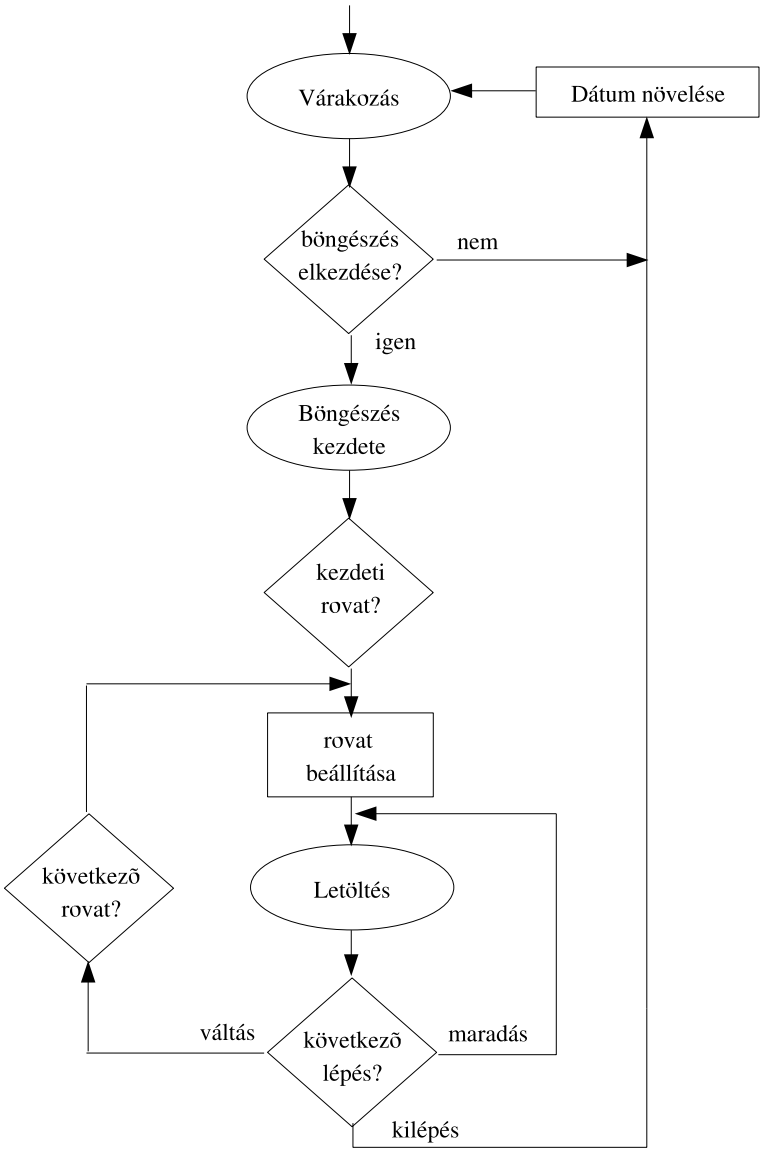
Rovatváltás A felhasználó dönthet úgy is, hogy egy másik rovatból tölti le a következő dokumentumot. Ezt nevezzük rovatváltásnak. Ekkor azon kívül, hogy a felhasználó elhatározza, hogy kilép az aktuális rovatból, nyilván azt is el kell döntenie, hogy milyen rovatból választ dokumentumot a következő letöltéshez.

Böngészés vége Végül minden session végén bekövetkezik az az esemény, mikor a felhasználó úgy dönt, hogy nem tölt le több oldalt, ekkor a böngészési sorozat véget ér.

A fenti események mindegyike valamilyen módon feltételez egy bizonyos szituációt. Tulajdonképpen itt arról van szó, hogy a böngészésnek vannak állapotai, és ezek az események állapothoz kötöttek.

Tekintsük át ezt 3. ábrán látható folyamatábrán.

3. ábra. A felhasználói modell folyamatábrája



5.2.2. A viselkedést meghatározó tényezők

Az előző szakaszban definiált döntéshelyzetekben nagyon sok tényező szerepet játszik, ebből mi természetesen csak a legfontosabbakkal foglalkozunk. Tekintsük át ezeket.

- Böngészés megkezdése: Az, hogy egy adott napon egy felhasználó böngészik vagy sem, csak attól függ, hogy *hány napja böngészett utoljára*.

- Kezdeti rovatba lépéskor történő rovatválasztás:

  A kezdeti rovatba lépéskor egyszerűen a lehetséges rovatok közül kényszerül választani a felhasználó. Azt, hogy egy adott rovat lesz a választás eredménye, jellemezhetjük úgy, mint egy, a rovatra jellemző konstans valószínűséggel bekövetkező eseményt. Azaz a böngészés elején kizárólag a választható rovatoktól függő valószínűséggel ugorhatunk egyik vagy másik rovatba.

  E mögött a felfogás mögött az a gondolat húzódik meg, hogy a rovatba ugrás esélye függ a rovat témájának érdekességétől, és függ egyfajta akkumulált minőségtől is, hiszen egy már többször is tetszést aratott rovatba szívesebben lép be az ember. Amennyiben nem kifejezetten hosszú távú trendeket szeretnénk vizsgálni, akkor tekinthetjük úgy, hogy ez a sok böngészés során kialakult szubjektív minőségi rangsor valamint a rovat domináns témájának – az információ közlésének módjától független – érdekessége nem változik, így valóban tekinthető konstansnak.

  A szimuláció során ennek a rovatonkénti konstansnak az értéket kell becsülnünk. Ehhez nyújt segítséget a rovat látogatottsági mutatójának fogalma.

  A *látogatottsági mutató*t többféleképpen is értelmezhetjük:

  1. *felhasználói látogatottság:*
     hány felhasználó látogatta meg az oldalt összesen a vizsgált időtartam alatt?
  2. *session látogatottság:*
     az egy nap alatt képződött session-ök közül átlagosan hányban szerepel az adott rovat ?
  3. *rovatkezdési látogatottság:*
     átlagosan hányszor kezdték a felhasználók az adott rovattal a böngészést?

4. *belépési látogatottság:*
   átlagosan hányszor léptek az adott rovatba (egy másik rovatból vagy először) a felhasználók egy nap alatt?
5. *letöltési látogatottság:*
   átlagosan hány oldalt töltöttek le a rovatból egy nap alatt?

Azt, hogy melyik meghatározás lesz számunkra a legjobb, mindig az aktuális alkalmazási mód fogja eldönteni. Látható, hogy ha éppen a böngészés kezdeti rovatválasztásához szeretnénk felhasználni a látogatottságot, akkor nyilvánvalóan akkor kapjuk a legpontosabb modellt, hogyha a harmadik definíciót, a rovatkezdési látogatottságot vesszük figyelembe. Ebben az a trükk, hogy ilyenkor tulajdonképpen nem becsüljük ezt a – rovatonként különböző – valószínűséget, hanem valójában megmérjük azt.

Hogy ez a megközelítés mikor alkalmazható, és mikor nem, arról később lesz szó.

- A böngészés közbeni legfőbb döntések:

  A böngészés során a következő események közül kell választanunk: rovatban maradás, rovatváltás vagy kilépés a böngészésből. Amennyiben a rovatváltást eseményét választja a felhasználó, úgy ezt a döntést egy újabb követi: annak a rovatnak a kiválasztása, melyből a következő letöltés során dokumentumot kér majd le. Ezt a második döntést a következő pontban fejtjük ki.

  A három alapvető esemény – session vége, rovatváltás vagy rovatban maradás – közti választásban alapvetően négy tényezőnek van szerepe. Ezek a következők:

  – *Frissesség:*
    A felhasználó által meglátogatható friss oldalak száma a rovatban. Mint ahogy már említettem, ezt a paramétert sztochasztikus módszerekkel fogjuk megbecsülni a felhasználó utolsó böngészési időpontjának ismeretében.
  – *Rovat minősége:*
    A rovatra jellemző paraméter, ami a rovat „olvasó-megtartási” képességét jellemzi. A későbbiekben valójában ez lesz a rovat szubjektív minőségét tükröző paraméter.
  – *Rovatban töltött „idő”:*
    A felhasználó rovatbeli fáradtságát befolyásolja a rovatban eltöltött böngészési idő. Mivel az aktív böngészést inkább a letöltött oldalak

száma jellemzi, ezért érdemesebb ezt figyelembe venni, mint valamilyen valós időmértéket használni. Az elolvasott és nem elolvasott, hanem esetleg csak megnézett oldalak között az egyszerűség érdekében nem próbálunk meg különbséget tenni.

  - *Böngészéssel töltött összes „idő”:*
    A rovat szintű fáradáson kívül a felhasználó session szintjén is fárad, ezt a fáradást pedig az eleddig összesen böngészéssel eltöltött idő jellemzi. Persze akár csak az előző pontban, itt is érdemes inkább az eddig összesen letöltött oldalak számát vizsgálni.

- Rovatváltáskor történő rovatválasztás

  A modellünkben értelmezhetjük a rovatváltáskor fellépő rovatválasztást úgy, mint a kezdeti rovat kiválasztását, azaz tekintet nélkül az eddigi eseményekre, csupán a látogatottság alapján, rovatonként konstans valószínűséggel választjuk egyik vagy másik rovatot a következő letöltéshez.

  Ennek a modellnek egy finomítása, hogy ezt az esélyt nemcsak a látogatottságtól, hanem az *eddig meglátogatott rovatok*tól is függőnek tekintjük. Ha ez csak az utolsó rovattól való függést jelenti, akkor ez lényegében egy egyszerű Markov-folyamatnak is tekinthető. Ez a megkötés már egészen jól leírhatja a valóságot, de persze értelme lehet több memóriával rendelkező Markov-szerű folyamatok használatának is.

  Bizonyos kutatások [20] azt mutatják, hogy a böngészés leginkább egyszerű Markov-folyamatként írható le, azaz valójában nem érvényesülnek olyan hatások a felhasználók választásaiban, melyek arra utalnának, hogy egy több lépéssel korábban megnézett oldal döntő szerepet játszana az aktuális választásban.

Azt, hogy ezek a paraméterek konkrétan hogyan befolyásolják az adott események bekövetkezésének valószínűségét, biztosan csak mérések segítségével lehet megállapítani. Intuitív módon mégis érezhető, hogy például a friss oldalak száma csak egy küszöb alatt csökkenti a rovatban maradás esélyét, viszont ha eléri a nullát, azaz a felhasználó már a rovat összes oldalát ismeri, akkor a kilépés igen nagy valószínűséggel bekövetkezik. A két érték között, azaz a releváns tartományban a rovatban maradás esélye függhet például valamilyen lineáris módon a friss oldalak számától.

5.3. Analitikus és empirikus megközelítések

Már szóltunk arról a problémáról, hogy a rendelkezésünkre álló adatok segítségével bizonyos valószínűségeket becsülnünk kell. Ezt alapvetően kétféleképpen

tehetjük meg.

Szinte minden esetben fennáll annak a lehetősége, hogy az adott valószínűség megbecslése helyett a valós adatokon pontosan megmérjük azt – nevezzük ezt a módszert empirikusnak. Ugyanakkor megtehetjük azt is, hogy bizonyos parametrikus eloszlásokat használunk (pl. geometriai, polinomiális vagy normális eloszlásokat), és ezek paramétereit próbáljuk meg becsülni az általunk mérhető adatokból. Használjuk ez utóbbi a módszer leírására az analitikus jelzőt.

A két megközelítés közül egyik sem abszolút értelemben jobb a másiknál, mindkettőnek megvannak a maga hátrányai és előnyei. Mindkét módszerhez található olyan szituáció, melyben az adott megközelítés oldalára billen a mérleg nyelve.

Amennyiben az empirikus megközelítés szerint megmérjük a használni kívánt valószínűséget, akkor nyilván megtaláltuk azt a módszert, melynek segítségével a legnagyobb valószínűséggel tudjuk szimulálni a valóságot. Így viszont nem tudunk meg semmit arról, hogy ez a valószínűség valójában milyen tényezőktől függ, ezzel tulajdonképpen a modellezés terét szűkítjük le.

Ha kifejezetten az adott esemény bekövetkeztekor szerepet játszó tényezőkre vagyunk kíváncsiak, akkor ebben nyilván nem segít, hogyha megmérjük ezeket a valószínűségeket. Ilyenkor mindenképpen az analitikus megközelítést kell használnunk, ami persze – annak becslő jellege miatt – biztosan kevésbé pontos szimulációt tesz lehetővé. Viszont tény, hogy ilyen módon közelebb jutunk a jelenségek megértéséhez, hiszen az analitikus módszer alkalmazásakor tulajdonképpen azt választjuk, hogy az adott jelenséget bevesszük a modellezendő jelenségek körébe.

A munkánk során mindkét megközelítést alkalmazni fogjuk.

5.4. Definíciók és formális jelölések

Legyen a rovatok száma r. Ekkor értelmezhető egy általánosított állapotátmeneti mátrix a következő módon:

1. Definíció. *Legyen az állapotátmeneti mátrix a következő* $(r+1) \times (r+1)$*-es mátrix:*

$$\mathbf{P} = \begin{pmatrix} p_{00} & p_{01} & \cdots & p_{0r} \\ p_{10} & p_{11} & \cdots & p_{1r} \\ \vdots & \vdots & \vdots & \ddots \\ p_{r0} & p_{r1} & \cdots & p_{rr} \end{pmatrix}$$

Ebben az átmenetmártixban a nulladik sor illetve oszlop az ún. kilépés rovatra vonatkozik, mely az éppen „nem böngésző" állapotot jelöli. Így az egyes elemek jelentése a következő (i *és* j *egészek):*

$$p_{ij} = \begin{cases} \textit{az } i\textit{. rovatból a } j\textit{. rovatba átugrás valószínűsége,} & \textit{ha } 1 \leq i,j \leq r \textit{ és } i \neq j \\ \textit{az } i\textit{. rovatban maradás valószínűsége,} & \textit{ha } 1 \leq i,j \leq r \textit{ és } i = j \\ \textit{az } i\textit{. rovattal kezdődő böngészés valószínűsége,} & \textit{ha } i = 0 \textit{ és } 1 \leq j \leq r \\ \textit{az } i\textit{. rovatból a böngészés végének valószínűsége,} & \textit{ha } 1 \leq i \leq r \textit{ és } j = 0 \\ \textit{a böngészés megkezdésének valószínűsége,} & \textit{ha } i = 0 \textit{ és } j = 0 \end{cases}$$

Mivel a mátrix elemei olyan valószínűségeket tartalmaznak, melyek a böngészés során lépésről lépésre változnak, így maga a $\mathbf{P}$ sem lesz konstans. Észre kell vennünk, hogy $\mathbf{P}$ i. sorának az ismeretében eldönthető, hogy az i. rovatból milyen eséllyel fejezzük be a sessiont, ugrunk más rovatba vagy maradunk az i. rovatban, azaz meg tudjuk hozni a böngészés során előálló döntéseket.

Az egy sorban (az első sor kivételével) megtalálható elemek egy teljes eseményrendszert alkotó, de bizonyos értelemben feltételes események. Például a p_{ij} valószínűség azzal a feltétellel jelenti az i. rovatból a j. rovatba ugrás esélyét, hogyha az már adott, hogy az i.rovatban voltunk. Hasonló igaz a rovatban maradás és a kilépés eseményekre is. Ezek az azonos feltétellel bíró események teljes eseményrendszert alkotnak, ezért

$$\sum_{j=0}^{r} p_{ij} = 1 \tag{1}$$

ahol $i \in \{1, \ldots, r\}$.

Ugyanakkor ha a mátrix első sorát vizsgáljuk, annak elemei (a legelső elem kivételével) azon eseményeknek felelnek meg, hogy a böngészés egy adott rovatban kezdődik el. Ezek az események tehát szintén feltételes események abban az értelemben, hogy feltételezik, hogy a böngészés egyáltalán elkezdődik. Ekkor ezek is teljes eseményrendszert adnak, azaz

$$\sum_{j=1}^{r} p_{0j} = 1 \tag{2}$$

Vezessünk be néhány jelölést, amelyekkel leírjuk majd a fenti események paramétereit.

2. Jelölés. *Legyen $i \in \{1, \ldots, r\}$, t pedig természetes szám.*

Csak a rovatot jellemző paraméterek:

$\underline{pop} = (pop_i)_{i=1}^r$ *ahol* pop_i *az* i*. rovat relatív látogatottsága*

$\underline{qual} = (qual_i)_{i=1}^r$ *ahol* $qual_i$ *az* i*. rovat minősége*

$\underline{newp} = (newp_i)_{i=1}^r$ *ahol* $newp_i$ *az* i*. rovatba naponta bekerülő friss oldalak száma*

Csak a felhasználó állapotát jellemző paraméterek:

$\underline{dp} = (dp_i)_{i=1}^r$ *ahol* dp_i *az aktuális felhasználó által az* i*. rovatból eddig letöltött oldalak száma*

$dpsum = \sum_{i=1}^r dp_i$ *azaz* $dpsum$ *az aktuális felhasználó által eddig összesen letöltött oldalak száma*

$tprev$ *az aktuális felhasználó utolsó böngészésének napja*

A felhasználótól és a rovattól is függő, származtatott paraméterek:

$\underline{fresh} = (fresh_i)_{i=1}^r$ *ahol* $fresh_i$ *az* i*. rovat frissességi értéke*

$\underline{fr(t)} = (fr_i(t))_{i=1}^r$ *ahol* fr_i *az* i*. rovatban a friss oldalak száma a t. napon*

Látható, hogy időbeli függést csak a friss oldalak számánál jelöltük, ennek oka, hogy a többi paraméternél mindig csak az aktuális napra számított értéket tartjuk nyilván. Így, ahol nem jelöljük, ott az aktuális napra vonatkozik a paraméter. (Ennek természetesen mindig egyértelműnek kell lennie.)

Tegyük fel, hogy létezik egy korlát ($frmax$), amely felett a friss oldalak számának csökkenését még nem észleli a felhasználó. Ekkor a frissességi értéket a következőképpen számíthatjuk:

3. Definíció. *Az* i*. rovat frissessége legyen a következő:*

$$fresh_i = \begin{cases} \frac{fr_i}{frmax} & \text{ha } fr_i < frmax \\ 1 & \text{különben} \end{cases}$$

Ehhez az i*. rovatban a t. napon aktuálisan található friss oldalak számát,* $fr_i(t)$ *-t számítsuk a következő módon:*

$$fr_i(t) = (t - tprev_i) \cdot newp_i - dp_i$$

Nézzük meg most az állapotátmenet mátrix elemeinek függését a fenti paraméterektől. Az egyes események tárgyalásakor nagyjából vázoltuk, hogy melyik esemény milyen paraméterektől függ. Itt csak ezt kell felhasználni, hiszen a mátrix elemei megfelelnek az egyes eseményeknek.

A böngészés kezdetének esélye csak az utolsó böngészés óta eltelt időtől függ, a kezdeti rovatba ugrás pedig csak a látogatottságtól. A rovatban maradás, a rovatváltás és a böngészés vége az előző szakaszban felírt négy paramétertől függ. Ezen kívül a rovatváltást követő rovatválasztásnál – Markov-modellt feltételezve – csak az aktuális rovattól, és az egyes rovatok látogatottságától függ, hogy melyik rovatba ugrik át a felhasználó.

Ennek megfelelően a modellünk a következő függvényeket használná:

$p_{00} = f_{start}(t - tprev)$

$p_{0i} = f_{in}(pop_i)$

$p_{ii} = f_{stay}(dpsum, dp_i, qual_i, fresh_i)$

$p_{i0} = f_{exit}(dpsum, dp_i, qual_i, fresh_i)$

$p_{ij} = f_{change}(dp_i, qual_i, fresh_i, i, pop_j)$

Itt $i, j \in \{1, \ldots, r\}, i \neq j$, t természetes szám és feltételezzük, hogy mindegyik függvény értékkészlete a $[0, 1]$ intervallum.

A probléma az, hogy az öt függvény által reprezentált események közül három – a rovatban maradás, a rovatváltás illetve a kilépés – összefügg, hiszen valószínűségeik összege 1. Így sajnos mindhárom esemény függ mindegyik paramétertől, ami túlságosan bonyolult paraméterteret eredményez. Ezen kívül érezhetően fontosabb szerepe van például a böngészésből való kilépés során az eddig összesen letöltött oldalak számának, mint mondjuk az aktuális rovatból meglátogatott oldalak számának.

A probléma áthidalása többféleképpen is megolható, de mindegyik megoldás során bizonyos egyszerűsítő feltételezésekkel kell élnünk. Ezek az megoldások arra alapulnak, hogy feltételezik bizonyos események függetlenségét valamely tényezőktől.

Néhány ilyen lehetséges egyszerűsítés:

- Tegyük fel, hogy a felhasználót leginkább a letöltött oldalak száma befolyásolja, azaz hogyha már túl hosszú ideig tart a böngészés, akkor biztosan befejezi azt, függetlenül az éppen nézegetett oldaltól. Feltesszük még, hogy ha a felhasználó nem hagyja abba a böngészést, akkor az adott rovattól függően olvas tovább, vagy ugrik egy másik rovatba.

  Ekkor a függvényeket így módosíthatjuk:

$$p_{i0} = f_{exit}(dpsum)$$
$$p_{ii} = f_{stay}(dp_i, qual_i, fresh_i)(1 - p_{i0})$$
$$p_{ij} = f_{change}(i, pop_j))(1 - p_{ii} - p_{i0})$$

- Most azt feltételezzük, hogy a felhasználó addig olvas egy rovatot, míg azt meg nem unja, azaz fontosabb az adott rovat hatása, mint a globális böngészési fáradtságé. Ekkor feltehetjük, hogy a rovatban maradás esélye nem függ az eddig összesen letöltött oldalak számától, sem a többi rovat minőségétől. Ezen kívül feltesszük még, hogy ha a felhasználó kilép egy rovatból, akkor az eddig meglátogatott oldalak számának függvényében lép ki, vagy ugrik egy másik rovatba.

  Ekkor a függvényeket így módosíthatjuk:

  $$p_{ii} = f_{stay}(dp_i, qual_i, fresh_i)$$
  $$p_{i0} = f_{exit}(dpsum)(1 - p_{ii})$$
  $$p_{ij} = f_{change}(i, pop_j))(1 - p_{ii} - p_{i0})$$

- A fenti modellt kiegészíthetjük úgy, hogy a rovatban maradás esélyének paraméterei közé még felvesszük az eddig böngészéssel eltelt időt is. Ekkor a függvények:

  $$p_{ii} = f_{stay}(dp_i, qual_i, fresh_i, dpsum)$$
  $$p_{i0} = f_{exit}(dpsum)(1 - p_{ii})$$
  $$p_{ij} = f_{change}(i, pop_j))(1 - p_{ii} - p_{i0})$$

Látható, hogy míg az első esetben a böngészés befejezésének „időbeli” eloszlását lehet könnyebben egy adott eloszláshoz igazítani, addig a második esetben az egy rovaton belül egyhuzamban letöltött oldalak számát könnyebb manipulálni.

A későbbiekben az első megközelítést alkalmaztuk, ennek oka, hogy még egy durva, igen kevés rovatot tartalmazó modelltől is elvártuk, hogy tükrözze a böngészés során összesen letöltött oldalak számának alakulását. Éppen az előbbiek miatt ez az alapvető elvárás nagy eséllyel a legelső megközelítés alkalmazásával valósítható meg könnyebben.

5.5. A kialakított modell

Bármelyik típusú modellt válasszuk is a fentiek közül, a következő kihívás, hogy a benne szereplő függvénykapcsolatokat konkrét függvényekkel helyettesítsük. A legegyszerűbb választások a konstans, a lineáris, a hatványfüggvény, és az exponenciális függvények.

Természetesen többféle modell is elképzelhető, ezek közül leginkább a következőket tartottuk alaposabb vizsgálatra érdemesnek:

- I. modell

  $f_{start}(t - tprev) = x$

  $f_{in}(pop_i) = p_{0i,emp}$

  $f_{change}(i, pop_j) = p_{ij,emp}$

  $f_{exit}(dpsum) = y$

  $f_{stay}(dp_i, qual_i, fresh_i) = fresh_i \cdot z^{(1-qual_i)\cdot dp_i}$

  ahol $0 < x, y, z < 1$ valós számok. $p_{ij,emp}$ jelentése az, hogy az átmenetmátrix p_{ij} elemét nem analitikusan becsüljük meg, hanem az eredeti weblogon mért empirikus értéket használjuk.

- II. modell

  Ez a modell csak az f_{stay} függvényben különbözik az I. modelltől:

  $f_{stay}(dp_i, qual_i, fresh_i) = fresh_i \frac{qual_i}{dp_i}$

- III. modell

  Ez a modell szintén csak az f_{stay} függvényben tér el az előzőektől:

  $f_{stay}(dp_i, qual_i, fresh_i) = fresh_i \cdot \frac{qual_i}{dp_i^z}$

  Itt z pozitív valós szám.

A fenti modelleket tulajdonképpen nevezhetjük rendre I., II. illetve III. analitikus, frissességi modellnek, hiszen analitikusan adott bennük az f_{start} és az f_{exit} függvény. Ezeket meg lehet adni empirikusan is, így további három vizsgálandó modellt kapunk, amennyiben ezen függvényeket a következő módon számítjuk:

$f_{start}(t - tprev) = p_{00,emp}$

$f_{exit}(dpsum) = p_{i0,emp}$

Ezen kívül vizsgálhatjuk ezen modelleknek a frissességet figyelembe nem vevő változatait is, ehhez csupán az f_{stay} függvényben megjelenő $fresh_i$ tényezőt kell eltörölni.

6. Modellillesztés és szimuláció

Ebben a fejezetben azt vizsgáljuk meg, hogy a már felépített modell segítségével hogyan lehet szimulációt végezni, valamint hogyan lehet a modellben szereplő

ismeretlen paramétereket úgy hangolni, hogy segítségükkel a lehető legjobban közelítsük a valóságot.

A modellillesztés alapja egy optimalizáló eljárás lesz. Maga a modell már önmagában elegendő arra, hogy segítségével böngészést szimuláljunk. Az így előálló adatok, egy szimulált weblog segítségével aztán megállapítjuk, hogy a szimulációhoz használt modell mennyire sikeresen közelíti a valóságot. Ezután ezt a jóságmértéket felhasználva valamilyen optimumkereső algoritmus segítségével módosítjuk a modell paramétereit. Az optimalizáló eljárás megfelelő számú iteráció után végül megadja azokat a paramétereket, melyekkel a lehető legjobb szimuláció érhető el.

Vegyük észre, hogy itt valójában egy adott modell paramétereinek optimalizálásáról van szó. Ezek közé tartoznak a rovatminőség paraméterek is, adott modell esetén tehát kiadódnak az optimalizáló kimenetéből ezek a kérdéses tényezők is.

A probléma az, hogy ahhoz, hogy megállapítsuk a rovatok minőségét, szükségünk van a modell ismeretére. Mivel – legalábbis az általunk bemutatott módszerrel – csak véges számú modellt lehet megvizsgálni, ezért minden valószínűség szerint lehetséges lesz az általunk adott modellnél jóval komplexebb, a valósághoz közelebb álló, bonyolult modellt adni. Ideális esetben maga a modellválasztás is optimalizáció útján jönne létre.

Azonban e dolgozatnak nem egy minden szempontból tökéletes felhasználói modell megalkotása az elsődleges célja, hanem egy általános modellalapú minősítő eljárás kidolgozása. Ehhez természetesen kezdetben mindenképpen szükségeltetik egy alkalmas modell megadása is, ám itt célunk nem a tökéletesség, hanem egy értelmes kompromisszum megkötése volt a bonyolultság és az átláthatóság, az értelmezhetőség között.

Ezért nem képezi a modellcsalád kiválasztása optimalizációval végrehajtott modellillesztés tárgyát.

6.1. A modellek összehasonlíthatósága

A modellillesztés problémáját tekintve alapvetően kétféle megközelítés közül választhatunk. Az egyik a bayesi döntéselmélet eredményein alapul, a másik pedig a különféle – gradiens alapú vagy annak ismeretét nem igénylő – szélsőértékkereső eljárásokat használja fel. Célunk mindkét esetben valamilyen jóságfüggvény megadása az egyes modellekre, mely alapján az optimalizáló végrehajthatja a modell paramétereinek hangolását.

6.1.1. Bayesi döntéselmélet

A böngészést tekinthetjük olyan sztochasztikus folyamatnak, amely valamely belső paraméterektől függ. A függés módját az általunk felállított modell adja. Ekkor a feladat tulajdonképpen a modell rejtett paramétereinek becslése. A bayesi döntéselmélet és a maximum likelihood becslés pontosan erre ad megoldást.

A formalizált feladat:

Legyen X egy valószínűségi változó, melynek értékkészlete az egyes rovatok azonosítói. A modell rejtett paramétereinek vektorát jelölje $\underline{\Theta}$. Ezek segítségével a következő valószínűségeket definiálhatjuk:

$P(X = i|\underline{\Theta})$

legyen az i. rovatba tartozó oldal letöltésének valószínűsége, ha a modell paraméterei $\underline{\Theta}$.

$P(X_1, X_2, \ldots, X_n|\underline{\Theta}) = P(\underline{X}^n)$

jelölje egy n letöltésből álló session rovatsorozatának eloszlását.

$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n|\underline{\Theta})$

legyen ennek megfelelően annak az esélye, hogy $\underline{\Theta}$ paramétereket feltételezve egy böngészési sorozat n hosszú, és a letöltött oldalak rendre az $x_1, x_2, \ldots, x_n$ rovatokból kerülnek ki.

$P(\underline{X}^{n_1}, \underline{X}^{n_2}, \ldots, \underline{X}^{n_m}|\underline{\Theta})$

pedig jelölje az összes session-t tartalmazó egész weblog rovatainak eloszlását.

$P(\underline{X}^{n_1} = \underline{x}_1, \underline{X}^{n_2} = \underline{x}_2, \ldots, \underline{X}^{n_m} = \underline{x}_m|\underline{\Theta})$

ekkor egy konkrét weblog valószínűségét adja meg, még mindig $\underline{\Theta}$ paramétereket feltételezve.

Vegyük észre, hogy elméletileg ezeket a valószínűségeket $\underline{\Theta}$ ismeretében ki lehet számolni a rendelkezésre álló modellből.

A feladat valójában a fordított feltételes valószínűségek meghatározása, azaz egy adott weblog esetén a legvalószínűbb paraméterek megtalálása. Erre nyújt megoldást a bayesi becslés alkalmazása:

$$P(\underline{\Theta}|\underline{X}^{n_1}, \underline{X}^{n_2}, \ldots, \underline{X}^{n_m}) = cP(\underline{X}^{n_1}, \underline{X}^{n_2}, \ldots, \underline{X}^{n_m}|\underline{\Theta})P(\underline{\Theta}) \qquad (3)$$

Itt c konstans, $P(\underline{\Theta})$ pedig az adott paramétervektor a priori valószínűsége, amelyet – mivel nem rendelkezünk kitüntetett paraméterekkel vagy valamely egyéb

előzetes feltételezéssel – szintén választhatunk konstansra, így eredményül a maximum likelihood becslést kapjuk.

A cél ennek a valószínűségnek a maximalizálása, melyet az egyenlet jobboldalának ismeretében megtehetünk. Fontos, hogy nem csak az adott valószínűség értékét tudjuk meghatározni, hanem a modell ismeretében kiszámíthatjuk a szélsőérték-keresést segítő gradienst is.

Ennek a módszernek egyértelmű előnye, hogy nemcsak matematikaiag megalapozott, hanem a gyakorlatban is sikeresen alkalmazott. Ugyanakkor a szükséges valószínűségek valamint a gradiens kiszámolása, bár megtehető, igen hosszadalmas. Ez önmagában nem jelentene gondot, a probléma az, hogy amennyiben több modell esetén is kíváncsiak vagyunk az optimalizálás eredményeire, akkor minden esetben újra kell számolni ezeket a képleteket.

6.1.2. Homogenitásvizsgálat χ^2-próbával

A modellillesztést végző optimalizáló eljárás bemenet mindenképpen valamilyen hibaérték vagy jóságfüggvény. (A kettő között nincs lényeges különbség, mindenképpen szélsőértéket kell keresni.)

Célunk tehát valamilyen módon a szimulált weblog jóságának mérése. Egy szimulációt akkor tekintünk jónak, hogyha eléggé hasonlít az általa generált weblog a valós weblogra.

Ezt a hasonlóságot többféleképpen mérhetjük. Mivel maga a weblog meglehetősen nagy méretű, ezért érdemes belőle különféle statisztikákat kinyerni, és azok hasonlóságát becsülni. A legtöbb statisztika általában valamilyen hisztogram formájában áll rendelkezésünkre. Egy hisztogram felfogható olyan adattárolási formának, melynél azt az adatot tartjuk számon, hogy egy adott valószínűségi változó hány esetben vett fel egy bizonyos értéket. (Például hány böngészési sorozat volt adott hoszzúságú.) Ekkor a hisztogram valójában egy adott eloszlású valószínűségi változó konkrét megnyilvánulásainak mintáit, azaz a mért gyakoriságokat tartalmazza.

Ha így szemlélve a statisztikáinkat páronként össze szeretnénk hasonlítani azokat, akkor eljutunk az ún. homogenitásvizsgálat problémájához, ami a hipotézisvizsgálatok egyik fajtája. A homogenitásvizsgálat célja az, hogy megállapítsuk, hogy két valószínűségi változó tekinthető-e azonos eloszlásúnak, vagyis, hogy az $X_1, \ldots, X_n$ és $Y_1, \ldots, Y_m$ azonos és független minták származhatnak-e azonos sokaságból. Itt a két mintahalmaz az eredeti illetve a szimulált weblogból készített statisztikákhoz tartozó valószínűségi változók mintái.

A probléma megoldását a χ^2-próba szolgáltatja [21]. Ehhez a következő mennyiséget kell kiszámolni, és összevetni az ún. kritikus értékkel, melyet a bizonyosság elvárt valószínűsége, valamint a probléma dimenziószáma határoz meg:

$$\chi^2 = nm \sum_{i=1}^{r} \frac{(\frac{\nu_i}{n} - \frac{\mu_i}{m})^2}{\nu_i + \mu_i} \tag{4}$$

Itt $\nu_i = |\{k \,|\, X_k = x_i\}|$ és $\mu_i = |\{k \,|\, Y_k = x_i\}|$ $(i = 1, \ldots, r)$, azaz ν_i és μ_i valójában a hisztogramok i. helyen felvett értékei.

A két statisztika hasonlóságának megfelelő mértékét nyújta az, ha megvizsgáljuk, hogy milyen valószínűséggel bízhatunk meg abban, hogy a két hisztogram azonos eloszlású valószínűségi változókból származik. Ez a valószínűség (4) értékéből a χ^2 -eloszlás ismeretében könnyen kiszámolható.

6.1.3. Eloszlások távolsága

Az előző szakaszban az egyes modellek jóságát úgy határoztuk meg, hogyaz eredeti valamint a modell segítségével generált weblogokból kinyert statisztikákat hasonlítottunk össze, mégpedig a statisztikai homogenitásvizsgálat segítségével. Ez a módszer sajnos nem minden esetben alkalmazható. Ennek oka, hogy a χ^2-próba olyan eloszlásokat illetve méréseket feltételez, melynek eredményeképpen az adott valószínűségi változó minden lehetséges értékét felveszi néhányszor (legalább ötször). Előfordulhat ezért, hogy a homogenitásvizsgálat eredménye nem tükrözi teljesen a valóságot.

Célszerű tehát egyéb módszereket is megadni két eloszlás összehasonlítására. Megfelelő módszer lehet egyszerűen két hisztogram (vagy eloszlás) távolságának definiálása.

4. Definíció. *Legyen X és Y két független valószínűségi változó, értékkészletük rendre* $\{x_1, \ldots, x_r\}$ *és* $\{y_1, \ldots, y_r\}$. *Ekkor a mintáikból származó hisztogramok értékei:*

$\nu_i = |\{k \,|\, X_k = x_i\}|$ *és* $\mu_i = |\{k \,|\, Y_k = x_i\}|$ *ahol*$(i = 1, \ldots, r)$

Legyen ekkor a két eloszlás távolsága:

$$d(X, Y) = \sum_{i=1}^{r} (\nu_i - \mu_i)^2$$

Ez alapján már megadható az eredeti és a szimulált weblogból adódó statisztikák távolsága. Az optimalizáló feladata az egyes statisztikapárok távolságainak súlyozott összegének minimalizálása.

6.2. Mérendő statisztikák

A weblogokon végzett méréseknek két fajtája van. Előfordulhat, hogy a mérés célja valamilyen adat gyűjtése, melyet a szimuláció során felhasználunk majd. Ebben az esetben nevezzük a mérést bemeneti mérésnek. Az ilyen méréseket nyilvánvalóan elegendő az eredeti weblogon elvégezni.

A másik eset az, amikor a mérés arra szolgál, hogy segítségével öszehasonlítsuk az eredeti és a szimuláció során generált weblogot. Az efféle méréseket nevezzük összehasonlító mérésnek. Ezeket nyilván mindkét weblogon el kell végezni. Fontos megjegyezni, hogy elméletileg a bemeneti mérések közül bármelyik játszhatja az összehasonlító mérés szerepét is. Gyakorlati okokból a mátrixos formában adott statisztikákat nem használjuk összehasonlítás céljára, ennek oka, hogy a χ^2-próbán alapuló hasonlóságvizsgálat leginkább egydimenziós statisztikákon – hisztogramokon – működik megfelelően.

Vegyük sorra a felhasználandó statisztikákat.

6.2.1. Bemeneti statisztikák

Ebben a részben a modell bemeneteként is szerepet kapó statisztikák rövid leírását adjuk meg. Ezeket a statisztikákat lehet összehasonlítás céljára is használni, ám ez – éppen az adatok bemenő jellege miatt – leginkább csak ellenőrzésre szolgál.

- *Szomszédos session-ök között eltelt napok számának hisztogramja*

  Megadja, hogy az azonos felhasználó által gyártott, időben egymást követő session-ök hányad részében telt el közöttük adott számú nap.

- *Dokumentum - session hisztogram*

  Megadja, hogy a session-ök mekkora hányadában történt összesen adott számú letöltés.

- *Rovatba lépésen alapuló rovat-látogatottság hisztogram*

  Megadja, hogy a rovatba lépések mekkora hányada volt az adott rovatba történő belépés esete.

- *Általánosított rovatátmenet mátrix*

  Az általánosított rovatátmeneti mátrix definiciójával egyező eredményt ad, egyetlen kivétel a p_{00} elem, ami ebben a mérésben nem a böngészés megkezdésének esélyét jelöli, hanem konstans módon nullát ad. Ennek oka, hogy a session-ök gyakoriságának vizsgálatára más mérés szolgál (ld. szomszédos session-ök között eltelt napok számának hisztogramja).

- *Új dokumentumok bekerülésének rovatonkénti mátrixa*

  A mérés során valahogy meg kell állapítani egy oldalról, hogy az egy adott időpontban új-e vagy sem. Ezt legegyszerűbben úgy oldhatjuk meg, hogy azon a napon tekintünk újnak egy dokumentumot, mikor arra legelőször érkezik egy adott mennyiségnél több lekérés. Ennek lehet oka, hogy az oldal valóban akkor került fel a rovat dokumentumai közé, vagy előfordulhat az is, hogy valamilyen okból ismét aktuálissá vált. Ez azonban egyáltalán nem zavarja meg az intuitív képünket egy új rovatról, hiszen egy régi információ új kontextusban egészen más jelentőséggel bírhat.

  Ennek a definíciónak az alapján már megadható, hogy egy adott (i azonosítójú) rovatra a j. napon hány új dokumentum került fel. Ez a szám még soronként, azaz naponta normálva van, és a kapott értéket tárolja a mátrix i. sorának j. eleme.

- *A naponta bekerülő új dokumentumok számának rovatonkénti hisztogramja*

  Megadja, hogy egy napon hány új dokumentum keletkezik egy adott rovatban. Ez a statisztika igen egyszerűen számolható az előzőből.

6.2.2. Összehasonlító statisztikák

A bemenetként nem, csak összehasonlítás céljaira elvégzett mérésekből adódó statisztikákat soroljuk fel az alábbiakban.

- *Session - felhasználó hisztogram*

  Megadja, hogy a felhasználók mekkora hányada böngészett adott számú alkalommal (session) során.

- *Rovatba lépés - felhasználó hisztogram*

  Megadja, hogy a felhasználók mekkora hányada lépett bele a böngészései során adott számú alkalommal valamely rovatba. Ez a belépés történhetett a böngészés kezdetén, vagy valamely rovatból történő rovatváltás során is. Az egyes rovatba lépések rovatonként nem akkumulálódnak.

- *Rovat - felhasználó hisztogram*

  Megadja, hogy a felhasználók mekkora hányada lépett bele adott számú rovatba valaha böngészései során. Itt természetesen akkumulálódnak a rovatb lépések, hiszen nem az a kérdés, hogy hányszor lépett rovatba egy felhasználó, hanem hogy hányfajta rovatot látogatott meg.

- *Dokumentum - felhasználó hisztogram*

  Megadja, hogy a felhasználók mekkora hányada töltött le adott számú dokumentumot összesen a böngészései során.

- *Rovatba lépés - session hisztogram*

  Megadja, hogy a session-ök mekkora hányadában történt adott számú rovatba lépés. Itt az egy rovathoz tartozó letöltések szintén nem gyűlnek össze.

- *Rovat - session hisztogram*

  Megadja, hogy a session-ök mekkora hányadában történt adott számú rovatokból letöltés. Azt használja fel, hogy egy session során hány rovatból töltött le a felhasználó oldalakat.

- *Dokumentum - session hisztogram*

  Megadja, hogy a session-ök mekkora hányadában történt adott számú letöltés összesen.

- *Rovatbeli folyamatos letöltések hosszának hisztogramja*

  Megadja, hogy az esetek hányad részében fordult elő, hogy egy session során a folyamatosan egy rovatból letöltött oldalak száma éppen a megadott érték volt. Itt a normálás tehát azzal a számmal történt, amely azt adja meg, hogy hány rovatba lépés történt az összes session alatt.

- *Rovatokbeli letöltések hosszának hisztogramja*

  Megadja, hogy az esetek hányad részében fordult elő, hogy az egy session során egy adott rovatból összesen letöltött oldalak száma adott számú volt.

- *Látogatók számán alapuló rovat-látogatottság hisztogram*

  Megadja, hogy egy adott rovatot hány felhasználó látogatott meg valaha. Normálva van az egyes rovatok között.

- *Session-ök számán alapuló rovat-látogatottság hisztogram*

  Megadja, hogy egy adott rovatot hány session tartalmaz összesen. Normálva van az egyes rovatok között.

- *Dokumentum letöltésen alapuló rovat-látogatottság hisztogram*

  Megadja, hogy a letöltések mekkora hányada történt az adott rovatból.

- *Dokumentum - session hisztogram, adott rovat esetén*

  Megadja, hogy a session-ök mekkora hányadában történt adott számú letöltés egy meghatározott rovatból.

6.3. Optimalizálási módszerek

Széleskörű alkalmazhatóságuk miatt az optimalizáló eljárások rengeteg változata vált ismertté, a legegyszerűbb eljárásoktól a mesterséges intelligencia eredményein alapuló komplex algoritmusokig. Ezen algoritmusokat sokféle szempontból lehet vizsgálni, a legelemibb csoportosítás azonban abból adódik, hogy igen nagy a különbség a hibafelület gradiensét felhasználó, valamint annak kiszámolását nem igénylő eljárások között.

6.3.1. Gradiens alapú módszerek

A gradiens alapú módszerek legfontosabb előnye a gyorsaság. Hátulütőjük azonban, hogy mindenképpen szükséges ezen eljárások alkalmazásához legalábbis a gradiens, de sokszor még a hibafelület második deriváltjának ismerete is. Ez esetünkben nagy probléma, hiszen a statisztikákon alapuló összehasonlítás miatt nem számolható ki a gradiens. Így tehát mindaddig, amíg annak körülményessége miatt nem a maximum likelihood megközelítést alkalmazzuk, ezeket a módszereket el kell vetnünk.

Marad tehát a gradiens ismeretét nem igénylő eljárások széles skálája. Ezeket a következő szakaszban tekintjük át.

6.3.2. A gradiens ismeretét nem igénylő módszerek

A gradienssel kiszámítására nem alapozó eljárások egy része egyáltalán nem is használja a gradiens fogalmát. Ilyen például a genetikus algoritmusok vagy a lokális keresés módszere. Ezek hátránya, hogy igen lassúak.

A kétfajta módszer között próbál megoldást keresni a gradienst becsülő optimalizáló algoritmusok családja. Ezek közül a ma ismert egyik leghatékonyabb az SPSA (Simultaneous Perturbation Stochastic Approximation) algoritmus [19]. Meglehetős gyorsasága valamint robosztussága miatt mi is ezt az eljárást választottuk.

6.3.3. Az SPSA algoritmus

Az algoritmus alapja, hogy véletlenszerű alapon kiválaszt néhány irányt a paramétertérben (minden dimenzió mentén megengedve az elmozdulást), majd ezen lépések megtétele után a kapott hibaértékekből becsli a gradienst az adott pontban. A mi esetünkben egy ilyen lépés megtétele gyakorlatilag egy adott paraméterekkel bíró felhasználói modell alapján elvégzett szimulációt, valamint a szimuláció

során generált weblogon mérhető összehasonlítás célú statisztikák elkészítését jelenti. Ezután a lépés, azaz a paramétertér adott pontjának jósága az eredeti és a szimulált weblogok statisztikáin végzett összehasonlítás eredménye lesz. Ez az összehasonlítás vagy a 6.1.2 pontban leírt homogenitásvizsgálattal, vagy a 6.1.3 részben definiált távolságmérték alkalmazásával történik.

Az ilyen módon megtett véletlenszerű lépések segítségével becsült gradiens alapján az SPSA algoritmus a gradiens módszer működésének megfelelően megtesz egy lépést a hibafelületen, várhatóan az optimum irányába elmozdulva. Ezen iterációs lépés többszöri megtétele végül elvezet egy lokális optimum közelébe.

Az SPSA algoritmus alkalmazásának kérdései:

- Kiindulási pont

  Mivel a rendszer paraméterterében nincs valódi kitüntetett pont, ezért legjobb az algoritmust véletlen kezdőponttal elindítani. A rovatminőség paramétereket tekintve hasznos lehet még az azonos minőségeket feltételező kiindulás is.

- Lokális optimum problémája

  Mivel minden, a gradiensmódszeren alapuló algoritmus csak lokális optimum megtalálására képes, ezért hasznos, ha az algoritmust többször is lefuttatjuk, mindig más véletlen pontból indulva. Ez tulajdonképpen minden gradiens alapú optimumkereső algoritmus esetén hasznos, hiszen ezek egyetemes hártánya, hogy semmiféle garanciával nem tudnak szolgálni arra vonatkozóan, hogy a kapott eredmény globális szélsőérték-e.

- Lépésköz mérete

  A lépésköz nagyságának megválasztására nincsenek egzakt módszerek, ugyanakkor léteznek jó heurisztikák. Ilyen heurisztika például, hogy amennyiben a becsült gradiens alapján megtett utolsó két lépés valóban az optimumhoz közelebb eső értéket adott, akkor a lépésközt növeljük, egy jó lépés utáni hibás lépés esetén viszont csökkentjük. A növelés általában lineáris, míg a csökkentés valamilyen multiplikatív módon történik, a konzervatív óvatosság jegyében.

- A zajosság problémája

  A statisztikák használata miatt természetes módon belép a rendszerbe egy adott nagyságú zaj. Ennek a csökkentését oly módon érhetjük el, hogy egy lépés megtételekor a jóságfüggvény számításához a szimuláció során feltételezett felhasználószámot megfelelően nagyra vesszük.

6.4. Paraméterek beállítása

A paraméterek hangolása nyilván az optimalizáló segítségével történik meg. A modellillesztés sikerének valószínűségét a megfelelő optimalizáló módszer kiválasztásán kívül azonban tovább növelhetjük azzal, hogyha az optimalizálás során a paraméterek terét lecsökkentjük.

A modell magas dimenziószámát leginkább a benne paraméterként szereplő, és így optimalizálandó rovatminőség értékek adják. Ezen kívül szerepelnek még ismeretlen, kitüntetett jelentéssel nem bíró belső paraméterek is a modellben. Erre példának hozhatók az egyes becsült eloszlások paraméterei.

Az alapötlet az, hogy kezdetben a rovatok vizsgálatának háttérbe szorításával csak igen kevés (tipikusan egy, esetleg kettő) rovatot definiálva próbáljuk meg a modell belső paramétereit hangolni, majd ezután egy újabb optimalizálások keretében fokozatosan továbbfinomítani a modellt. Ezzel a módszerrel várhatóan sikerül biztosítani, hogy a modell szerves részét képező paraméterek kevésbé függjenek az aktuálisan vizsgálandó rovatok tulajdonságaitól.

7. A modell implementálása és alkalmazása

Ebben a fejezetben bemutatjuk a modell implementálásának legfőbb kérdéseit, és megmutatjuk, hogy hogyan alkalmaztuk a kész rendszert egy gyakorlati prbléma megoldásában.

7.1. A rendszer felépítése

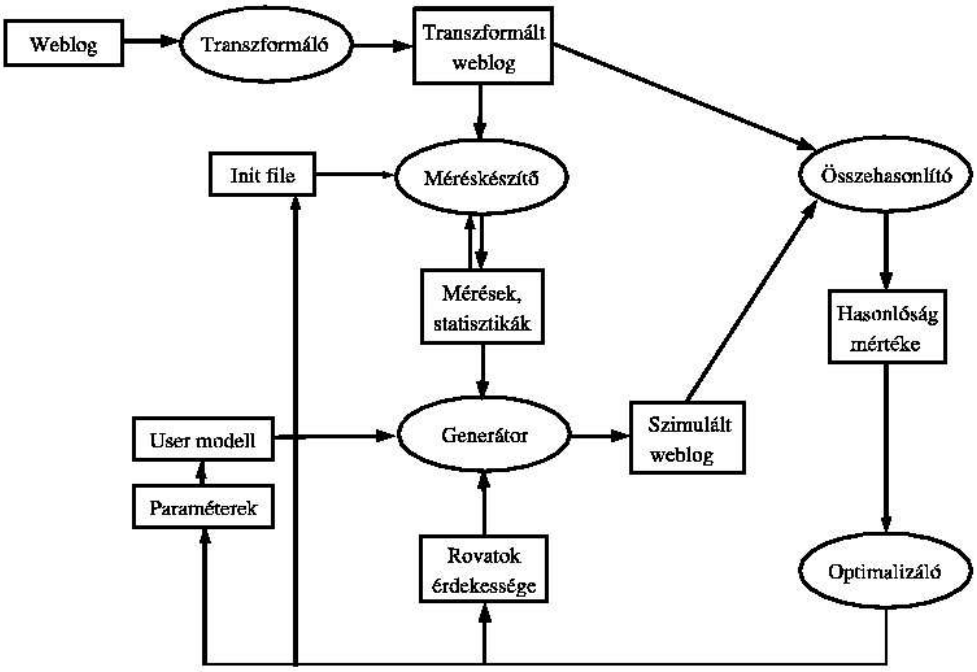
Lássuk, hogy az általunk implemetált rendszerben milyen elemek jutnak szerephez. A rendszer felépítését a 4. ábra mutatja.

Az ábrán szereplő téglalapok mindig valamiféle adatokat, míg az ellipszisek folyamatokat, valamilyen feladatot elvégző objebtumokat jelentenek. Bár ezen az objektumokok közül a legtöbbről már esett szó, álljon itt egy összefoglaló az egyes rendszerelemekről.

A rendszerben szereplő adatok:

- Weblog: az eredetileg rendelkezésre álló naplóállomány.
- Transzformált weblog: a szükséges előfeldolgozó lépések után kapott adatbázis, mely már a session-öket tartalmazza, felhasználónként egybegyűjtve.
- User modell: a felhasználóról alkotott modell. Tartalmaz egy függvénytárat, egy a modellhez kapcsolódó paramétertárat, valamint az adott felhasználó állapotát leíró paramétereket.

4. ábra. A rendszer vázlatos terve



- Paramétertár: a felhasználó viselkedését leíró modell paramétereit tartalmazza.
- Rovatok minősége: a rovatok objektív minősítése, melynek meghatározása a cél.
- Mérések, statisztikák: tartalmazza a rovatok különféle mért adatait, azaz mind a bemeneti, mind az összehasonlító statisztikákat.
- Szimulált weblog: A felhasználó modellje és a rovatok tulajdonságai alapján szimulált weblog.
- Hasonlóság mértéke: A szimulált és az eredeti weblog hasonlósága, mely vagy az egyes statisztikák homogenitásvizsgálatából adódó egyezési valószínűségek, vagy a statisztikák korábban definiált páronkénti távolságainak súlyozott összege.

  (Utóbbi esetben valójában nem hasonlóságmértékről van szó, melyet maximalizálni kell, hanem távolságmértékről, melyet minimalizálni szeretnénk. Ez persze nem lényegi eltérés.)
- Init file: Inicializáló fájl, tartalmazza például a rovatok számát, valamint egyéb, a szimulációhoz szükséges paramétereket.

A rendszerben szereplő folyamatok:

- Transzformáló: az eredeti weblogot átalakítja a megfelelő alakra, és elvégzi az előfeldolgozást. Ezzel részletesebben a 7.2.2 szakaszban foglalkozunk.
- Méréskészítő: a transzformált weblog és az inicializáló adatok alapján – esetleg már elkészített statisztikákat is felhasználva – kiszámítja a szükséges statisztikákat. Ezekről bővebben a 6.2 részben olvashatunk.
- Generátor: a felhasználóról alkotott modell, a benne szereplő paraméterek és a rovatok tulajdonságai alapján sztochasztikusan szimulálja a böngészést, és elkészíti a hozzá tartozó weblogot.
- Összehasonlító: a szimulált és az eredeti modellből készített statisztikák alapján megadja a modell jóságát. Működésének alapjait a 6.1.2 illetve a 6.1.3 részekben fejtettük ki.
- Optimalizáló: a modell jóságának, azaz a hasonlóságmértéknek függvényében megadja, hogy a paramétertérben merre kell elmozdulni, így változtatja a generátor bemeneteit, a minőséget és a felhasználói modell paramétereit. Hatással lehet az init file-ra is.

  Az használt optimalizálót a 6.3.3 szakaszban ismertettük.

7.2. A megoldandó probléma

Az alkalmazási feladat egy magyarországi vezető internetes hírportál rovatainak minősítése volt. Ehhez rendelkezésünkre állt a hírportál üzemeltetői által a rendelkezésünkre bocsájtott naplózófájl, melyben 2004. szeptemberének négy hetében történt böngészések adatai szerepeltek. A naplóállomány mérete megközelítőleg öt Gigabyte volt.

7.2.1. Az implementálás alapkérdései

Az implementálás Linux környezetben történt, a használt programnyelv pedig a C++ volt. A használt fordító a gcc 3.3-as verziója volt. A különféle hatékony adatszerkezetek használatakor a C++ Standard Template Library nevű csomagja hagyatkoztam, míg a szimulációk matematikai hátterének megteremtésében a GNU Scientific Library nyújtott segítséget. A szimulációkat hat számítógépen végeztük. A gépek Pentium 4-es processzorral és 512 Mbyte memóriával rendelkeztek, operációs rendszerük Debian Linux volt.

Az aktuális feladatot megismerve el kellett döntenünk hogy pontosan milyen rovatokat definiálunk. A csupán az elérési útvonalra alapozott rovatdefiníció igen sokszor félrevezető lett volna, mivel így például bizonyos hibaüzeneteket, hirdetéseket vagy egyéb más irreleváns információt tartalmazó „rovatokat” is definiáltunk volna.

Így a lehetséges elérési útvonalakból adódó rovatjelöltek valós tartalmát megvizsgálva kiválasztottuk a valóban relevánsakat. Ezek között definiáltunk többféle felosztást is annak érdekében, hogy lehetőségünk legyen a modell fokozatos finomítására. Ez azt jelentette, hogy definiáltunk olyan hierarchikus kapcsolatokat is, melyek segítségével a szimulációk során össze tudtunk vonni bizonyos hasonló tartalmú rovatokat.

Végül huszonhat vizsgálandó rovat keletkezett.

7.2.2. Az adatok előfeldolgozása

1. Hibás vagy hiányos adatok kiszűrése.

2. A felhasználói sorozatok felépítése.

   A felhasználói sorozatok, valamint az egy felhasználóhoz tartozó sessionök felépítése – a feldolgozandó adathalmaz óriási mennyisége okán – közel sem volt triviális feladat. A megoldás lépései következők voltak:

   - Az azonos felhasználók adatainak összefésülése

- A felhasználói sorozatok szétdarabolása session-ökre

  Ennél a feladatnál nem csak a session hosszát (24 óra), hanem annak kezdetét illetve végét is meg kellett állapítani. Az internetforgalom aktivitásának egy napon belüli változását figyelembe véve a fordulópontot éjjel 3 órára választottuk.

  Ebben a szakaszban történik a session-azonosítók kiosztása is.

3. A rovatcímkék hozzárendelése.

   A letöltendő dokumentum elérési útvonala alapján az egyes lekérésekhez hozzárendeltük annak a rovatnak az azonosítóját, melybe a meglátogatott oldal tartozott.

4. Felesleges adatok kiszűrése.

   Ide tartozik a főoldalra vonatkozó kérések kiszűrése, valamint a böngészők automatikus frissítéséből adódó ismételt letöltések elhagyása is.

7.2.3. Adatvédelmi megfontolások

Az általunk implementált rendszer eleget tesz az adatvédelmi előírásoknak. Ezt a következő módszerekkel értük el:

– Már az eredeti naplóállományban is csupán "cookie"-k segítségével azonosítjuk a felhasználókat. Az adatvédelmi törvény előírásai szerint ennek a használatát minden böngészőprogramban le lehet tiltani.

– Különválasztjuk az eredeti weblog feldolgozásának és a modellillesztés (tehát a szimulációk) szakaszát. Ezáltal lehetőség nyílik arra, hogy az eredeti weblog feldolgozását a portál üzemeltetője végezze. Ennek a részfeladatnak a kimenete – a különféle statisztikákat tartalmazó állomány – egyáltalán nem tartalmaz a felhasználók azonosítását szolgáló elemeket. Így a rovatok vizsgálatakor semmilyen lehetőség nincs az adatvédelmi elvek megsértésére.

  Egyébiránt ez a különválasztás implementációs szempontból is hatékony, hiszen többszöri szimuláció esetén is elegendő az eredeti – igen nagy méretű – adatbázis egyszeri végigolvasása.

7.3. Szimulációk

Ebben a szakaszban előbb a szimulációnál használt paramétereket, azok kiindulási értékeit ismertetjük, majd ismertetjük az elvégzendő szimulációkat.

7.3.1. Kiindulási értékek

A szimulációhoz szükséges kezdeti értékeket két csoportba oszthatjuk:

- Fix paraméterek

  Ezeket a paramétereket nem változtatjuk a szimuláció során. Ide tartozik a friss oldalak számának kezdeti értéke, valamint annak a küszöbnek a megadása, hogy hány új oldal elegendő ahhoz, hogy a felhasználó ne érezze a rovat elavulását, azaz a frissességi tényező 1 legyen.

  Ezen a paraméterek értékét rendszerint valamilyen intuitív módon állítjuk egy konstans értékre. Ezt azért tehetjük meg, mert ezek a tényezők nem befolyásolják meghatározóan a szimuláció eredményét.

- Optimalizálandó paraméterek

  Ezeket a paramétereket a szimulációk egymást követő iterációjában folyamatosan változtatjuk, ezért kiindulási értékük kevésbé lényeges, általában véletlenszerű.

  Ide tartoznak például a rovatok minőségének értékei, valamint a felhasználó modelljében szereplő belső paraméterek.

7.3.2. Az elvégzett szimulációk

A legmegfelelőbb modell kiválasztása érdekében az minden vizsgálandó modell esetén végeztünk az adott modellt használó szimulációkat. Az ezekre alapuló optimalizálás eredményeképpen megkaptuk az legalkalmasabb modellt. Ezután ennek segítségével újabb futtatásokat végeztünk, véletlenszerű kezdőpontokból indítva az optimalizálást, ezzel is elősegítve a lehető legjobb eredmények megtalálását.

Az egyes szimulációk eredményét a 8.1 szakaszban ismertetjük.

8. Eredmények

8.1. Eredmények bemutatása és elemzése

8.1.1. A legmegfelelőbb modell kiválasztása

Mivel a legfontosabb célunk a modellezés során az egyes rovatok minőségének megállapítása volt, ezért a vizsgálandó modellek leglényegesebb eleme az ettől közvetlen módon függő f_{stay} függvény volt.

1. táblázat. A különböző empirikus modellek összehasonlítása

	Frissességet használva			Frissességet nem használva		
	I. modell	II. modell	III. modell	I. modell	II. modell	III. modell
p	98.1303%	98.1274%	98.1207%	98.1092%	98.1088%	98.1077%
d	2.9996	3.1334	3.7714	5.2468	5.1931	5.4060

2. táblázat. Az empirikus modellek összehasonlításából adódó sorrend

	Frissességet használva			Frissességet nem használva		
	I. modell	II. modell	III. modell	I. modell	II. modell	III. modell
p szerinti sorrend	1.	2.	3.	4.	5.	6.
d szerinti sorrend	1.	2.	3.	5.	4.	6.

Lássuk, melyik típusú modell adta a legjobb eredményeket.

Az empirikus modellek esetén kapott eredményeket az 1. táblázat foglalja össze. A kiértékelést mind a χ^2-próbán alapuló homogenitásvizsgálattal, mind a hisztogramok távolságának kiszámításával elvégeztük. Az előbbi eredménye p-vel jelölve látható az első sorban, míg az utóbbi megközelítésnél a hisztogramok távolságösszegét tüntettük fel (d). A táblázat vizsgálatakor természetesen szem előtt kell tartani, hogy az előbbi esetben egy jóságfüggvény maximalizálása, míg az utóbbiakban egy hibaérték minimalizálása a cél – így az első sorban a legmagasabb, a második sorban pedig a legalacsonyabb értéket adó oszlop adja a legmegfelelőbb modellt.

Látható, hogy a homogenitásvizsgálat eredménye csak igen kisléptékű eltéréseket ad az egyes modellekre, a hisztogramok távolságán alapuló összehasonlítás jóval határozottabb különbségeket mutat. Az optimalizálások során mindig ez utóbbi módszert használtuk, hiszen így jóval meredekebb hibafelülethez jutottunk.

Érdekes és fontos, a módszer helyességét alátámasztó tény ugyanakkor, hogy mindkét összehasonlítást alkalmazva az I. modell frissesség nélküli változata adta a legjobb eredményt. Ennél lényegesen több is igaz: hogyha megvizsgáljuk a kétféle minősítésből adódó sorrendezését a modelleknek (2. táblázat), akkor szinte ugyanazt kapjuk. Eltérés csak egyetlen helyen jelentkezik, a sorrendben 4. és 5. helyen álló modellek rangsorolásánál. Ez igazán lényeges eredmény, hiszen azt mutatja, hogy a modellek minősítése valóban helyesen működik.

3. táblázat. Az I. modell analitikus és empirikus változatainak hibái

Analitikus modell	Empirikus modell
3.2782	2.9996

Az, hogy a frissesség nélküli modell jobban írta le a valóságot, azt mutathatja, hogy a böngészés során a tipikus felhasználó számára nem szab gátat a még nem ismert oldalak számának csökkenése. Ennek legvalószínűbb oka, hogy a felhasználó gyakran visszatér már meglátogatott oldalakra, például egy részletesebben megismerni kívánt téma miatt. Ez intuitív alapon igen hitelesnek tűnő magyarázat a frissességgel nem operáló modellek sikerességére.

A megfelelő alapmodell kiválasztása után megvizsgáltuk, hogy vajon az empirikus vagy az analitikus modell ad-e jobb eredményt. Ehhez a hisztogramok távolságán alapuló hibamértéket használtuk. A vizsgálat eredménye a 3. táblázatban található.

A kapott eredmény az volt, hogy az empirikus megközelítés jobban közelíti a valóságot. Ez tulajdonképpen várható volt a 5.3 szakaszban leírtak alapján. Jelen esetben nem okoz problémát az, hogy az empirikus megközelítés során mind f_{start}, mind f_{exit} értékét becslés helyett megmérjük, hiszen a számunkra fontos minőség paramétertől – az általunk alkamazott modellben – egyik sem függ.

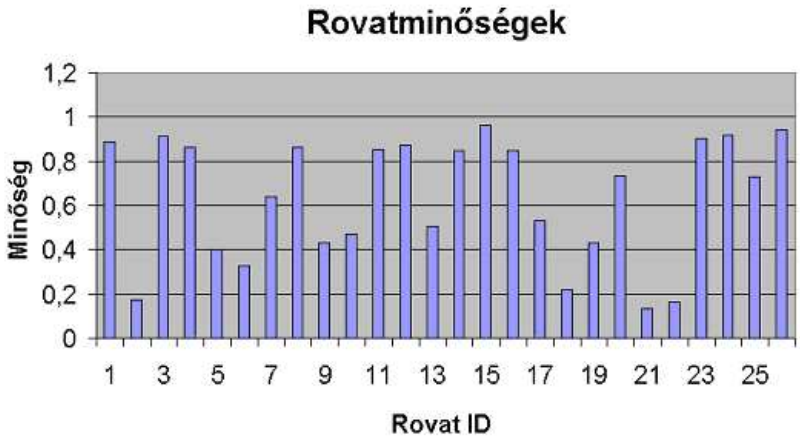
Emlékeztetőül álljanak itt a kiválasztott, azaz az empirikus, frissességet nem használó I. modell azon függvényei, melyek megadják a böngészési döntések során felhasznált valószínűségeket:

$f_{start}(t - tprev) = p_{00,emp}$
$f_{in}(pop_i) = p_{0i,emp}$
$f_{change}(i, pop_j) = p_{ij,emp}$
$f_{exit}(dpsum) = p_{i0,emp}$
$f_{stay}(dp_i, qual_i, fresh_i) = z^{(1-qual_i)\cdot dp_i}$

8.1.2. A rovatok minősége

A legalkalmasabb modell kiválasztása után megvizsgáltuk, milyen eredményeket kaptunk annak alkalmazásával az egyes rovatok minőségére. A modellillesztés eredményeképpen megkaptuk az optimális paramétereket, ezek között szerepeltek a minőségértékek is. Az eredményt numerikusan a 4. táblázat tartalmazza, a rovatazonosítókat, a rovatok témáját és azok felhasználói látogatottságát feltüntetve.

5. ábra. Az egyes rovatok minősége



4. táblázat. A rovatok minősége

Azonosító	Rovat témája	Minőség	Látogatottság
1	Portálismertető	0.887452	90
2	Promóciós rovat	0.171310	39751
3	Szezonális programok	0.911102	8235
4	Internet	0.863116	20881
5	Erotika	0.402990	169257
6	Kulturális ajánló	0.327606	328474
7	Női rovat	0.638273	223155
8	Multimédia	0.863695	102842
9	Külpolitika	0.430787	232074
10	Üzleti hírek	0.470267	224284
11	Számítástechnika	0.852283	281963
12	Autó – motor	0.871305	193179
13	Tudomány	0.505149	208673
14	Belpolitika	0.850240	286008
15	Sport	0.960240	288556
16	Egészség	0.849711	27250
17	Időjárás	0.531337	35229
18	Aktuális hírek	0.218532	74960
19	Képtár	0.429690	18381
20	Archívum	0.733150	18014
21	Utónévkönyv	0.133704	14863
22	Állatok	0.163821	28573
23	Ipresszum	0.901490	1
24	Utazás	0.915788	18499
25	Mobiltelefon	0.726635	5165
26	Bűnözés	0.941772	1383

A 5. ábra diagramon ábrázolja az egyes rovatok minőségét.

Ezen eredmények helytállóságának eldöntése nem a mi feladatunk, hiszen a rovat minőségét csak a hozzáértő szakemberek, jelen esetben a hírportál szerkesztői tudnák megállapítani.

8.1.3. Futási idők

Munkánk során megvizsgáltuk az implementált szimulációs program végrehajtásából származó futási időket. Tapasztalataink alapján egyetlen letöltés generálása mintegy 0.7-0.8 ms nagyságrendű időt vesz igénybe. Ismerve a modell által generált felhasználói sorozatokban naponta letöltött dokumentumok átlagos számát (2-3 letöltés), kiszámolható, hogy a 28 napos szimulációk során alkalmazott 500-as felhasználói létszám mellett egyetlen szimuláció körülbelül 30 másodpercet vezs igénybe.

Egyetlen optimalizálás során ugyanakkor minden lépésben pontosan kilenc szimulációt végzünk, mivel az SPSA algoritmus négy-négy irányban számol hibaértéket ahhoz, hogy aztán ebből a gradienst becsülje. Az ehhez szükséges szimulációk száma nyolc, amit a végül valóban megtett optimalizációs lépéshez szükséges hiba- vagy jóságérték számolása kilencre egészít ki. Feltéve, hogy az optimalizáció során megtett lépések száma a 30-as nagyságrendben van – ez tapasztalataink szerint legtöbbször teljesül –, az ehhez szükséges futási idő már átlagosan 130-140 perc.

Ha még meggondoljuk, hogy több véletlenszerűen kiválasztott pontból is célszerű optimalizációkat indítani, akkor ez 10-12 próbálkozás esetén már kitesz egy teljes napot.

Ez a viszonylagos lassúság nem küszöbölhető ki, hiszen már egyetlen szimuláció során is nagyszámú, esetünkben 500 felhasználó böngészéseit kell generálnunk, ami meglehetősen nagymennyiségű adatot jelent. Az ezen szimulációk sokszori megismétlését igénylő optimalizálás tehát mindenképpen időigényes feladat.

8.2. Értékelés

Összefoglalva az eredményeket megállapíthatjuk, hogy sikerült egy alapvetően jól működő modellt adnunk, melynek segítségével a valóságot jól közelítő szimulációkat tudtunk végezni. Ezen szimulációk eredményeként kiadódott az elsődleges célként kitűzött minőségi mérce, mely az elvárásainkkal összhangban lévő módon értékelte az egyes rovatok minőségét.

9. Összefoglaló

E dolgozat célja internetes hírportálok rovatközpontú böngészési modelleken alapuló minősítése volt. Ehhez a szükséges alapfogalmak definiálása után megvizsgáltuk, hogy milyen elvárásaink vannak egy megfelelő böngészési modellel szemben, majd ezekből kiindulva számba vettük a lehetséges megoldásokat.

A modellillesztéshez szükséges optimalizációs eljárást, valamint az annak bemenetéül szolgáló kiértékelési módszereket körültekintően megválasztottuk. Ezek, valamint egy konkrét hírportál naplóállományának segítségével sikerült megadnunk az általunk vizsgált modellek közül a valóságot legjobban közelítőt. Végül ezt a modellt az optimalizálás során illesztve a valós adatokra, a kapott eredményeket elemezve megkaptuk az egyes rovatok minőségét is.

Az általunk a dolgozatban bemutatott eljárás során eleddig egyedülálló, új szempontokat figyelembe vevő megközelítést alkalmaztunk. Az általunk elért eredmények minden bizonnyal hozzájárulnak tehát egy egyre fontosabbá váló terület, a webes tartalomminősítés fejlődéséhez.

Eredményeinket publikálni szeretnénk a Salford System Data Mining 2005 (Barcelona, Spain) és esetleg a 14th International World Wide Web 2005 (Chiba, Japan) konferencián is.

Hivatkozások

[1] Lara Catledge and James Pitkow: *Characterizing browsing strategies in the world-wide web.* Computer Networks and ISDN Systems, 26(6):1065-1073, 1995.

[2] M. Spiliopoulou, Carsten Pohle, and Max Teltzrow. *Modelling and mining web site usage strategies.* In Proceedings of the Multi-Konferenz Wirtschaftsinformatik, Nurnberg, Germany, Sept. 9-11, 2002.

[3] J. Srivastava, R. Cooley, M. Deshpande, P-T. Tan. *Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data.* In SIGKDD Explorations, (1) 2, 2000.

[4] E. Spertus. *Parasite : Mining structural information on the web.* Computer Networks and ISDN Systems: The International Journal of Computer and Telecommunication Networking, 29:1205-1215, 1997.

[5] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. *Inferring web communities from link topology.* In Conference on Hypertext and Hypermedia. ACM, 1998.

[6] Ville H. Tuulos, Henry Tirri. *Combining Topic Models and Social Networks for Chat Data Mining.* Web Intelligence, IEEE/WIC/ACM International Conference on (WI'04), China, Beijing. pp 206-213, 2004.

[7] Wil M.P. van der Aalst, Minseok Song. *Mining Social Networks: Uncovering Interaction Patterns in Business Processes.* Business Process Management 2004: 244-260, 2004.

[8] M. Pazzani, L. Nguyen, and S. Mantik. *Learning from hotlists and coldlists: Towards a www information filtering and seeking agent.* In IEEE 1995 International Conference on Tools with Artificial Intelligence, 1995.

[9] V. R. Borkar, K. Deshmukh, and S. Sarawagi. *Automatic Segmentation of Text into Structured Records.* Proc. ACM-SIGMOD Int. Conf. Management of Data (SIGMOD 2001), ACM Press, New York, 2001, pp. 175-186.

[10] S. Chakrabarti. *Mining the Web: Statistical Analysis of Hypertex and Semi-Structured Data.* Morgan Kaufmann, San Francisco, 2002.

[11] M. Steinbach, G. Karypis and V. Kumar. *A comparison of document clustering techniques.* In KDD Workshop on Text Mining, 2000.

[12] Douglass R. Cutting, Jan O. Pedersen, David Karger, John W. Tukey. *Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections.* Proc. 15th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. pp. 318-329, 1992.

[13] Eui-Hong (Sam) Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis. *A Web Agent for Document Categorization and Exploration.* Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98).

[14] M.S. Chen, J.S. Park, P.S. Yu. *Data miningfor path traversal patterns in a web environment.* In 16th International Conference on Distributed Computing Systems, pp. 385-392, 1996.

[15] B. Huberman, P. Pirolli, J. Pitkow, R. Kukose. *Strong regularities in world wide web surfing.* Technical report, Xerox PARC, 1998.

[16] Mike Perkowitz, Oren Etzioni. *Adaptive Sites: Automatically Learning from User Access Patterns.* In Proc. WWW6, April 1997.

[17] S. Schechter, M. Krishnan, and M. D. Smith. *Using path profiles to predict http requests.* In 7th International World Wide Web Conference, Brisbane, Australia, 1998.

[18] C. Anderson, P. Domingos, and D. Weld. *Relational Markov Models and their Application to Adaptive Web Navigation.* In Proc. 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, pages 143-152, Edmonton, Canada, 2002.

[19] John L. Maryak, Daniel C. Chin *Global random optimization by simultaneous perturbation stochastic approximation* Proc. 33nd conf. on Winter simulation, pages 307-312, Virginia, 2001.

[20] A. A. Benczúr, K. Csalogány, B. Rácz, Cs. Sidló, M. Uher and L. Végh *An Architecture for Mining Massive Web Logs with Experiments* http://www.sztaki.hu/ alukacs/Papers/origomining.pdf, 2003.

[21] Borovkov A. A. *Matematikai statisztika.* Typotex, Budapest, 1999.