DEBRECENI EGYETEM

INFORMATIKAI KAR

Webbányászati módszerek alkalmazása a web perszonalizációban

Témavezető:

Dr. Ispány Márton

egyetemi docens

Készítette:

Gonda László

Programtervező informatikus Bsc

Debrecen

2009

Tartalomjegyzék

1. Bevezetés – a perszonalizáció igénye

Ma a világháló a legnagyobb és legszélesebb körben ismert információforrás a világon, mely könnyen hozzáférhető és kereshető. Több milliárd egymásra kölcsönösen hivatkozó weblapból áll, melyet emberek milliói írnak. Megjelenése óta drámaian megváltoztatta keresési szokásainkat, szinte minden pár kattintásnyi távolságba került, az információ könnyen fellelhető és megosztható. Emellett a web kényelmes lehetőségeket nyújt vásárlásra, kommunikálásra, véleménynyilvánításra és vitázásra. Azonban ez a folyamat a visszájára fordul, mivel a web dinamikus ütemben növekszik, naponta milliós nagyságrendű új weboldalt töltenek fel, és ezzel a megfelelő információt egyre nehezebb fellelni. Ez az információtúltengés frusztráló élménye.

A weben található információ mennyiségével együtt növekszik a weben keresztül kínált szolgáltatások mennyisége is, az elektronikus kereskedelem, tanulás és banki ügyintézés megjelenésével, ezért ezek szolgáltatói között is egyre jobban élesedik a verseny.

Az információtúltengés megjelenése és a webes piacért folytatott harc arra a felismerésre juttatta az üzleti weboldalak tulajdonosait, hogy érdekükben áll oldalaik látogatóival, mint reklámfelületeik közönségével és termékeik potenciális vásárlóival, lojális viszont kialakítani, amit azáltal érhetnek el, ha valamilyen plusz értéket adnak hozzá az oldaluk meglátogatása adta élményhez, ami nem feltétlenül a rajta lévő információtömeg növelésével, hanem inkább a felhasználók által keresett információ könnyebb és gyorsabb, megfelelő formában és időben történő elérésével érhetnek el.

Ehhez az oldalon szereplő információkat a várt felhasználókhoz igazítva kell strukturálni, azonban a felhasználók különböző igényei és beállítottsága miatt ugyanazon elrendezéssel nem lehet széleskörű megelégedettséget elérni. Ezért vetődött fel az oldalak testreszabhatóságának, perszonalizálásának lehetősége, hogy mindenki azt kaphassa, ami számára a legmegfelelőbb. Ráadásul a látogató személyes igényeihez való igazodás egyértelműen növeli a komfortérzetét, ami nagyban elősegíti a szolgáltatók által áhított lojális viszony kialakulását.

Kezdetben a perszonalizáció alapja a hosszas űrlapkitöltés eredményeképpen létrejövő felhasználói profil volt, ami ugyan lehetőséget ad a felhasználó igényeinek explicit kifejezésére, de megvalósítása a felhasználótól, az oldal tervezőjétől, fejlesztőjétől és a

tartalomszolgáltatótól is komoly ráfordítást igényel, egyes látogatók számára pedig kényelmetlen, vagy egyenesen riasztó lehet.

De később az adatbányászat módszereit alkalmazó társterület, a webbányászat technikáit alkalmazva megkezdték a perszonalizációs rendszerek automatizálását. Mobasher et al. (1999) a web perszonalizációt már egyszerűen úgy definiálja, mint a webalapú információs rendszerek képessé tételét az egyéni felhasználók igényeihez és érdeklődési köreihez való alkalmazkodásra. Tipikusan, egy perszonalizált oldal felismeri a felhasználóit, információkat gyűjt preferenciáikról, és igazítja szolgáltatásait a felhasználó igényeihez.

A perszonalizációs rendszerek jelenlegi fő használati területe a termékajánló rendszerek, de számos más funkció megvalósítására is alkalmas, az egészen egyszerű felhasználó-felismeréstől egészen a felhasználó helyetti feladatvégzésig (pl. e-mailküldés, aukciós licitálás). A webhasználat-bányászat pedig ötletek és eljárások értékes forrása a perszonalizációs funkcionalitás implementációjához.

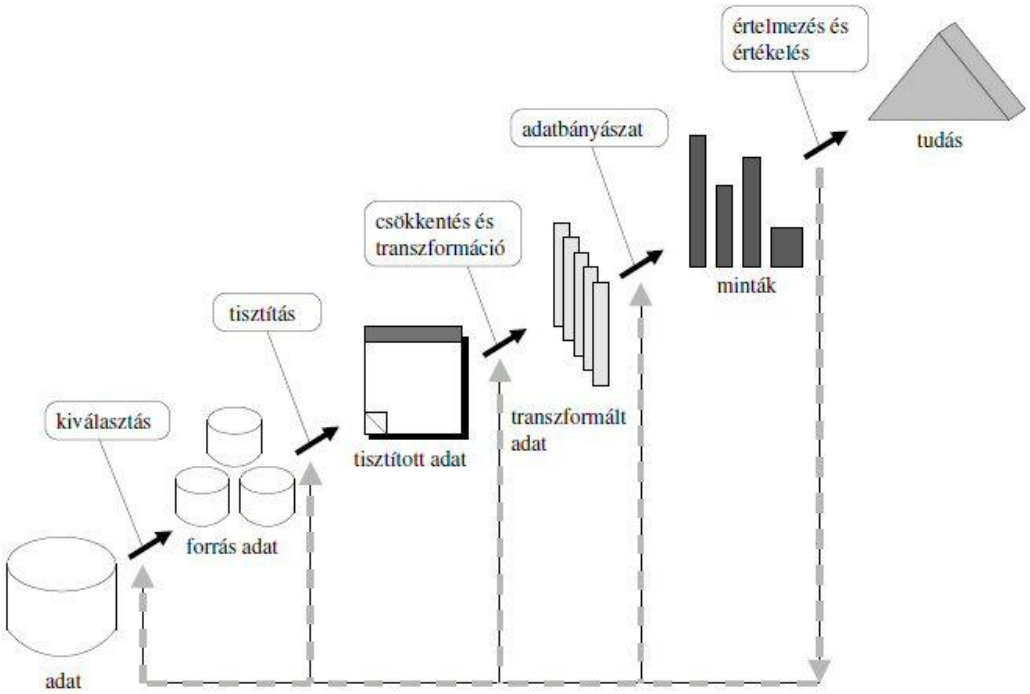
Dolgozatomban ezen automatizált perszonalizációs rendszerek működését, gyakorlati felhasználását, osztályozását mutatom be, közben kitérve a fejlesztésük során felmerült problémák megoldási kísérleteire és a területen kialakult szabványokra.

2. Módszertani háttér – az adatbányászattól a web perszonalizációig

2.1. Adatbányászat

Az adatbányászatot általában úgy definiálják, mint az adatforrásokból, azaz adatbázisokból, szövegekből, képekből, az internetről, vagy egyéb forrásból történő hasznos tudásminták kinyerésének folyamatát. Ezen mintáknak validáltnak, potenciálisan hasznosnak, és érthetőnek kell lennie. Ennek eléréséhez az adatbányászat a gépi tanulás, statisztika, adatbázisok, mesterséges intelligencia, az információkinyerés és vizualizáció módszereit alkalmazza, ezáltal egy igen összetett és számos témához kötődő kutatási területet alkotva.

Egy adatbányászati alkalmazás fejlesztése általában az alkalmazási terület megértésével kezdődik, ezután azonosítják a megfelelő adatforrásokat és a céladatokat. Ezen adatokon végezhető el az adatbányászati folyamat.



Az adatbányászati folyamat menete. Forrás: Bodon, 2008

Mint az ábrán látható, az adatbányászat folyamata egymásra épülő lépésekre bontható, amiket alapvetően három fázisba lehet besorolni:

- Előfeldolgozás: mivel az adathalmazok általában túl nagyok egészükben való elemzésre, és eleve az elemzés szempontjából irreleváns adatokat is tartalmazhatnak, ide tartozik a megfelelő adatok kiválasztása. Ezek sok esetben nem alkalmasak azonnali vizsgálatra, szükség lehet a megtisztításukra a zaj és az abnormális adatok eltávolítása által. A kiválasztott adathalmaz még így is lehet túl nagy, szükség lehet dimenziócsökkentésre, és további transzformációkra az alkalmazott adatbányászati algoritmustól függően.
- Adatbányászat: A feldolgozott adatokat betáplálják egy adatbányászati algoritmusba, ami kinyeri a tudásmintákat.
- Utófeldolgozás: Sok alkalmazás esetében nem minden felfedezett minta hasznos, ez a lépés ismeri fel a felismert minták közül a hasznosakat az alkalmazás szempontjából. Számos kiértékelési és vizualizációs eljárás használatos ezen döntések meghozatalára.

A klasszikus adatbányászat strukturált, relációs táblákban, táblázatokban, vagy egyszerű fájlokban tárolt adatokkal dolgozik. A web kiterjedésének és a strukturálatlan szöveges dokumentumok mennyiségének növekedésével azonban webbányászat és a szövegbányászat egyre fontosabbá és népszerűbbé válnak. A webbányászat megjelenésekor gyakorlatilag az

adatbányászat módszereinek alkalmazása weben összegyűjtött adathalmazokból való információkinyerésre (Etzioni, 1996).

2.2 Webbányászat

A webbányászat célja hasznos információ kinyerése a web hiperlink-struktúrájából, oldalainak tartalmából és a használata során keletkezett adatokból. Bár a webbányászat sok adatbányászati technikát alkalmaz, ma már nem tisztán a klasszikus adatbányászati technikák alkalmazását jelenti, a webes adatok heterogén, félig strukturált vagy strukturálatlan természete miatt. Az adatbányászati folyamat során elsősorban használatos adattípusokra alapozva a webbányászati feladatokat jellemzően három típusba sorolják:

- webstruktúra-bányászat: a web felépítését reprezentáló hiperlinkekből nyer ki értékes információt. Pl. ilyen módszereket alkalmaznak a keresőmotorok az egyes oldalak relevanciájának meghatározására. A klasszikus adatbányászat ilyen feladatot nem végez, mivel ott általában nincsenek linkek a relációs táblák struktúrájában.
- webes tartalombányászat: hasznos tudást nyer ki a weboldalak tartalmából, pl. automatikusan osztályozza és klaszterezi a weboldalakat tematika szerint. Ezek a feladatok hasonlítanak a tradicionális adatbányászathoz. A webes tartalom bányászata során fórumok hozzászólásai vagy felhasználói kritikák alapján feltárhatóak fontos információk a vásárlói hozzáállással kapcsolatban.
- webhasználat-bányászat: felhasználói oldalhozzáférési mintázatok felismerése szervernaplókban, amik minden egyes felhasználó minden egyes klikkelését rögzítik. Sok adatbányászati algoritmust alkalmaz. Ezen a területen különösen fontos az adatok megfelelő előfeldolgozása.

A webbányászati folyamat hasonlít az adatbányászati folyamatra, a különbség általában az adathalmazban van. Adatbányászat esetében az adatok gyakran előre be vannak gyűjtve és le vannak tárolva egy adattárházban. A webbányászat számára jelentős feladat lehet az adatgyűjtés, főleg az első két esetben. Az adatgyűjtés után az adatbányászati folyamathoz hasonlóan az előfeldolgozás, webbányászat és utófeldolgozás lépései következnek. Azonban minden lépéshez nagyban különböző technikákat lehet alkalmazni a klasszikus adatbányászat esetében alkalmazottaktól.

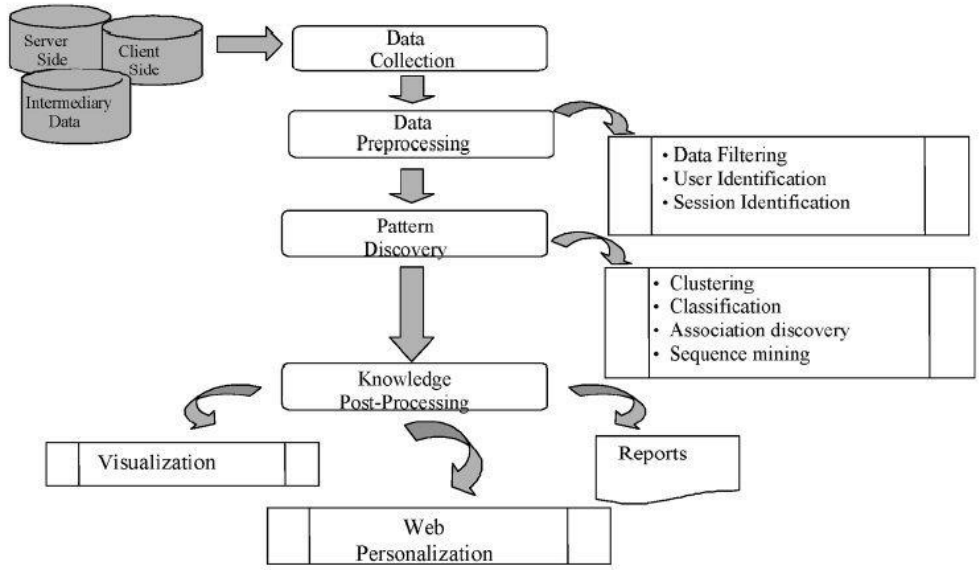
2.3 Webhasználat-bányászat

A webbányászat három alterülete közül a perszonalizációs megoldásokhoz a webhasználat-bányászat módszerei alkalmazhatóak, mivel azok használják forrásként a webes szolgáltatások igénybevétele, a weboldalak látogatása során keletkező adatokat, főleg a szervernaplókat. Bár a webhasználat-bányászat területén megjelent munkák többsége nem perszonalizációval foglalkozik, a terület automatizált perszonalizációval való kapcsolata egyértelműen következik abból, hogy nagy mennyiségű felhasználói adatot kezel.

Ezen adatok elemzése segítheti a szervezeteket meghatározni klienseik élettartamát, kidolgozni marketingstratégiáikat, megvizsgálni hirdetési kampányaik hatékonyságát, optimalizálni webalapú alkalmazásaik funkcionalitását, jobban testre szabható tartalmat nyújtani a látogatóiknak, és megtalálni a leghatékonyabb logikai struktúrát a weboldalukhoz.

A webhasználat-bányászat a clickstream („clickstream" alatt értjük a webet böngésző felhasználók szolgáltatóknál hátrahagyott nyomait, képszerűen a kattintgatásainak sorozatát) és a hozzákapcsolódó összegyűjtött és a felhasználói interakciók eredményeképpen generált adatok halmazában való automatikus mintafelismerést és elemzést jelenti. A cél a viselkedési minták és az oldallal interakcióba kerülő felhasználói profilok megragadása, modellezése és elemzése.

A standard adatbányászati folyamatot követve a webhasználat-bányászati folyamat is három egymástól függő fázisra bontható: adatgyűjtés és előfeldolgozás, mintafelismerés, és mintaelemzés.



A webhasználatbányászati folyamat menete. Forrás: Pierrakos et al., 2003

Az előfeldolgozási fázisban a clickstream adatokat megtisztítják és felhasználói session-ökre osztják, amik az egyes felhasználók egyes az oldalra történő látogatásait reprezentálják. Más tudásforrások, mint pl. az oldal tartalma és felépítése is használhatóak az előfeldolgozásban a felhasználói tranzakciók adatainak erősítésére. A mintafelismerési fázisban statisztikai, adatbáziskezelési műveletek mellett tanuló algoritmusokat alkalmaznak a felhasználók tipikus viselkedésére utaló rejtett minták feltárására, és a webes erőforrások, session-ök és felhasználók statisztikáinak összefoglalására. A folyamat utolsó fázisában a felismert mintákat és statisztikákat további feldolgozásnak vetik alá, és szűrik, ami csoportosított felhasználómodelleket is eredményezhet, amik többek között alapjául szolgálhatnak az adott felhasználó számára legmegfelelőbb ajánlatok kikövetkeztetésének, vagy más perszonalizációs funkcióknak. Ezen funkciók csoportosítása célszerű, mivel rendeltetésük és összetettségük alapján általában meghatározható, hogy milyen webhasználat-bányászati módszer implementálásával valósíthatóak meg.

3. Perszonalizációs funkciók

3.1. Perszonalizációs funkcióosztályok

A web perszonalizáció, mint kifejezés által összefogott, a felhasználó webes interakcióinak könnyebbé tételét célzó technikák, funkcionalitások osztályozásának szükségét Kobsa et al. (2001) vetette fel. Felosztása létrehozásakor figyelembe vette az akkor piacon lévő üzleti rendszereket és kutatási prototípusokat, továbbá a rendszerek által pillanatnyilag megvalósítható képességeket. A perszonalizációs funkciók négy alaposztályát határozta meg: memorizáció, útmutatás, testreszabás, és feladatvégzési támogatás.

3.1.1. Memorizáció

A perszonalizációs funkcionalitás legegyszerűbb formája, a rendszer a „memóriájában” tárol információkat a felhasználóról, mint pl. név vagy böngészési előzmények. Általában önmagában ez a típusú funkcionalitás nem jelenik meg, többnyire egy teljesebb perszonalizációs rendszer részeként van jelen. Az ebbe az osztályba tartozó funkciók pl. a felhasználó üdvözlése, ahol a rendszer a visszatérő felhasználó nevét egy üdvözlő üzenettel együtt kiírja. Egyszerű mivolta ellenére ez az első lépés a felhasználói hűség kialakítása felé,

mivel azt az érzést kelti a felhasználóban, hogy a rendszer mint egyént, és nem mint átlagos látogatót kezeli. Ide tartoznak még a könyvjelzők, és a személyre szabott hozzáférési jogok.

3.1.2. Útmutatás

Az ide tartozó funkciók a felhasználónak segítséget nyújtanak az általa keresett információhoz való gyors eljutásban, ezzel átsegítve az információtúltengés problémáján, ami nagyobb weboldalak esetében gyakran előfordul. Ide tartoznak a hiperlink ajánlások, de a sokkal kevésbé ismert oktatási funkciók is, amelyek a felhasználó tudásszintjét felmérve neki megfelelő módon segítik előbbre jutását lépésről lépésre.

3.1.3. Testreszabás

A weboldal tartalmának, felépítésének és elrendezésének módosítása a felhasználó tudásszintjének, preferenciáinak és érdeklődésének megfelelően. Fő célja a felhasználói interakciók megkönnyítése. Személyre szabható a megjelenített tartalmak részletezése, de az oldalon megjelenő hiperlinkek köre is szűkülhet vagy bővülhet a felhasználó igényeihez igazodva, vagy akár az adott oldalon található termékek árazása is lehet személyre szabott.

3.1.4. Feladatvégzési támogatás

A legbonyolultabb perszonalizációs funkcionalitások csoportja, amik egyfajta személyes asszisztensként működnek. Küldhetnek e-maileket vagy elvégezhetik nagyobb elemhalmazok letöltését a felhasználó helyett, kiegészíthetik lekérdezéseit, ezzel növelve egy információkinyerő rendszer teljesítményét, vagy akár tárgyalófélként is részt vehet a felhasználó helyett, pl. online aukciókon.

3.2. Perszonalizációs politikák

A perszonalizációs funkciók egy kombinációjának kiválasztása az adott perszonalizációs rendszer felépítéséhez attól függ, hogy az adott oldal tulajdonosa milyen perszonalizációs politikát kíván követni. A perszonalizációs politikát olyan faktorok határozzák meg, mint a weboldal felhasználási területe, a rendelkezésre álló emberi és pénzügyi erőforrások, a felhasználóknak kínált tartalom típusa és komplexitása, és az elvárt válaszidőre vonatkozó megszorítások. A perszonalizációs politika kidolgozása során általában figyelembe vett paraméterek:

- Egy rendszer lehet egy vagy több felhasználós, egy felhasználós rendszerről beszélünk, ha funkcionalitása személyi felhasználómodelleken alapszik, azaz minden egyes felhasználó külön-külön felmért érdeklődésén, preferenciáin és tudásán. Ezzel szemben a több felhasználós perszonalizációs politika összesített modelleken alapszik, mint pl. felhasználói közösségek vagy sztereotípiák. Ilyen lépés pl. egy megegyező akciós ár felajánlása minden, egy meghatározott mennyiségű terméket már megvásárolt felhasználónak.
- A perszonalizációs funkciók lehetnek statikusak vagy dinamikusak. Statikus funkciókról beszélünk, ha azokat egy felhasználói session során egyszer alkalmazzák, pl. egy visszatérő felhasználó belépésekor. Ezzel szemben a dinamikus perszonalizáció azt feltételezi, hogy a perszonalizációs funkciók szerepet kapnak a felhasználó és a weboldal interakciójának minden egyes lépésénél. Pl. minden új lekérdezésnél új ajánlatokat kaphat a felhasználó, közelmúltbeli böngészési történelmének függvényében.
- A perszonalizációs politika környezetérzékenysége is fontos tényező, azaz annak kérdése, hogy az egy session-ön belül más oldalakhoz vagy más témákhoz tartozó oldalnézeteket is figyelembe veszi-e a rendszer az adott témában történő ajánláskor.
- A végrehajtott perszonalizációs tevékenységekhez elérhető-e magyarázat.
- A rendszer önállóságának kérdése. Ha bizonyos perszonalizációs funkciók a felhasználó bevonása nélkül, teljesen automatikusan hajtódnak végre, a rendszert proaktívnak nevezzük, ez főleg feladatvégzési támogatás alkalmazásánál fordul elő. Ezzel szemben egy konzervatív rendszer minden tevékenység irányítását a felhasználó kezében hagyja.
- A perszonalizációs funkciók lehetnek konvergensek vagy divergensek, aszerint, hogy egy meghatározott témára fókuszálnak, vagy általánosabb információkat szolgáltatnak, az aktuálistól eltérő típusú termékeket is ajánlanak, amik érdekelhetik a felhasználót.

3.3. Perszonalizációs rendszerek osztályozása

A fentebbi osztályokból kiválasztott funkcionalitások kombinálásával felépíthető az adott oldalnak megfelelő perszonalizációs politika, és az alapján felépíthető az oldalhoz tartozó perszonalizációs rendszer. Ezeket a rendszereket is lehet csoportosítani, a tartalmazott funkciók vagy követett politika mellett a feltárt szakterület alapján is, de ami sokkal

lényegesebb, a rendszer felépítése során alkalmazott implementációs megoldás alapján is. Mobasher et al. (2000) felosztása szerint három megközelítés különíthető el ebből a szempontból:

- Vannak manuális döntési szabályos rendszerek, melyeket a webes szolgáltatást készítőjének manuális beavatkozásával perszonalizálják, gyakran a felhasználó együttműködésével. Jellemzően regisztrációs folyamat során épülnek fel a statikus felhasználómodellek, és bizonyos számú manuálisan megalkotott szabály határozza meg, hogy a különböző modellekkel rendelkező felhasználóknak milyen webes tartalmakat szolgáltatnak.
- A tartalomalapú szűrőrendszerek tanuló algoritmusokat alkalmaznak a webes tartalomra, és azokból próbálják felismerni a felhasználó személyes preferenciáit. Adaptív módon képesek a felhasználó viselkedéséből felépíteni annak felhasználómodelljét.
- A szociális vagy kollaboratív szűrőrendszerek célja egy szolgáltatás perszonalizálása maga a tartalom elemzése nélkül. Ennek eléréséhez közös tulajdonságokat keresnek különböző felhasználóknál, amiket általában explicit fejeznek ki, főleg elemek (árucikkek, cikkek, termékek, stb.) értékelése formájában, és a rendszer ezt rögzíti.

A manuálisan készített döntési szabályokon alapuló rendszerek ugyanazon problémáktól terheltek, mint bármely más manuálisan strukturált komplex rendszer, azaz jelentős erőfeszítésbe kerül a felépítésük és karbantartásuk. Továbbá általában szükségük van a felhasználó bevonására is, ami nem elhanyagolható mértékben csökkenti a rendszer használata iránti kedvet. Ezzel szemben a másik két megközelítés nem igényel ilyen erőfeszítést a rendszer felépítése után, mivel ezek már webhasználat-bányászatra épülnek. Azonban a webhasználat-bányászat alapú perszonalizáció sem problémák nélkül való. A bányászati folyamat minden fázisa során szembekerülhet problémákkal a rendszer kivitelezője, amik egy részére még nem, vagy csak részben találtak megoldást.

4. Webbányászat-alapú perszonalizáció

Amennyiben a perszonalizációs rendszer alapjául webbányászati technikát alkalmazunk, a perszonalizációs célok a webbányászati folyamat minden fázisára ki fognak hatni.

4.1. Adatgyűjtés

Fontos feladat minden adatbányászati alkalmazásban a megfelelő céladatbázis kialakítása, így a webhasználat-bányászat első lépése is a releváns webes információk begyűjtése, melyek elemzése fog hasznos tudást szolgáltatni a felhasználói viselkedésről. A webhasználat-bányászat két fő adatforrása, összefüggésben a web session során együttműködő két szoftverrendszerrel, a szerver oldali és a kliens oldali adatok. Továbbá amikor a kliens-szerver kommunikációba közbenső elemek ékelődnek be, mint pl. proxy szerverek, akkor azok is értékes adatforrásokká válhatnak.

4.1.1. Szerveroldali adatok

A szerver oldali információkat a weboldal szerver oldaláról gyűjtik be, és ezen adatok elsősorban a webszerver által generált különféle naplófájlokból kerülnek ki. A szerver naplófájljai a használat sémáinak felfedezésének fő forrásai, ezek bejegyzései reprezentálják a látogatók finom szemcsézettségű navigációs viselkedését. Azonban nem minden esetben tekinthetőek megbízható információforrásnak, alapvetően két okból: a web gyorsítótárazás (caching) és az IP címek félreértelmezésének problémája miatt.

A web cache egy várakozás- és forgalomcsökkentő eljárás. Egy web cache számon tartja a lekért weboldalakat, és egy másolatot ment el belőlük egy meghatározott időtartamra. Ha ugyanarra az oldalra érkezik kérés ezen az időtartamon belül, annak újbóli lekérése helyett a cache-ben lévő másolat kerül használatra. Ha a lekért weboldal a cache-ben van, a kliens kérése nem jut el az oldalt kezelő webszerverig, tehát a szerver nem is tud a lekérésről, és az oldalhozzáférés így nem kerül be a naplóba. Ez ellen megoldás lehet a cache-busting, azaz speciális http-fejrészek (Cache-Control response headers) használata, amik megszabják az adott oldalak cache-elését, direktívákat tartalmaznak az egyes elemek cache-elésére nézve, azonban ez a „megoldás” ellentmond a cache-használat elsődleges céljának, a forgalomcsökkentésnek.

Az IP címek félreértelmezése két fő okból fordulhat elő. Az egyik a proxy szerverek használata, ugyanis ezek felhasználóiknak könnyen megegyező IP-címeket oszthatnak ki. Hasonló probléma merül fel mikor ugyanazt a host-ot több személy használja (pl. egy család tagjai). Ennek ellenkezője fordul elő, ha ugyanazon felhasználóhoz sok különböző IP cím rendelődik hozzá, pl. az internetszolgáltatók által alkalmazott dinamikus címkiosztás miatt, vagy mert sok különböző helyről használja a rendszert.

A szervernaplók mellett szerver oldali adatforrások a felhasználó által explicit megadott és letárolt adatok és a szerver lekérdezéseinek adatai is, azonban a legfőbb információforrás a szerver hozzáférési naplója.

4.1.2. Kliensoldali adatok

A weboldalhoz hozzáférő host-ról összegyűjtött adatok. Általában agent-ek alkalmazásával gyűjtik össze őket, amiket leggyakrabban Java vagy JavaScript nyelven íródott, a weboldalakba ágyazott programokat jelentenek, és közvetlenül a klienstől gyűjtenek információkat, pl. a weboldalhoz való hozzáférés és annak elhagyásának időpontját, az aktuális oldal előtt és után meglátogatott oldalak listáját, stb.

A kliensoldali adatok megbízhatóbbak a szerveroldali adatoknál, mivel áthidalják a cache-elési és IP félreértelmezési problémákat. Egy probléma viszont, hogy számos agent információgyűjtése befolyásolja a kliens rendszerének teljesítményét, emellett ezek a metódusok igénylik a felhasználók együttműködését, akik nem minden esetben engedélyezik bizonyos agent-ek futását, gyakran olyan biztonsági megoldásokat aktiválva, amik tiltják a Java és JavaScript programok futtatását a böngészők számára, a káros szoftverek elkerülésre érdekében. Ilyen körülmények között az agent-ek használatával történő adatgyűjtés abszolút nem hatékony.

Gyakori kliens oldali adatgyűjtési mód a sütik (cookie-k) alkalmazása is, azonban ezek esetében is felmerül az egy felhasználó több számítógépről való hozzáférésének problémája, továbbá a sütik 4 kilobájtban maximált mérete limitálja az adatgyűjtés mértékét, ráadásul az agent-ekhez hasonlóan letilthatóak, szintén biztonsági megfontolásokból.

4.1.3. Közbülső adatok

A felhasználói host és webszerver közé ékelődhetnek proxy szerverek, amik olyan szoftverek, amit általában cégek használnak az internethez való csatlakozásra, hogy a cég

biztosítani tudja a biztonságot, az adminisztratív irányítást és cache-elési szolgáltatásokat belső host-jai számára. A fentebb említett általuk okozott problémák ellenére fontos információforrásokká válhatnak, mivel a webszerverekhez hasonlóan vezetnek hozzáférési naplót, amiből kinyerhetőek weboldal lekérések és az azokra adott válaszok, azonban a cache és IP félreértelmezési problémák, amik felmerültek a webszerveknél, a proxy szerverekről kinyert adatokra is jellemzőek.

4.1.4. Adatok kiválasztása

A felsorolt adatforrások közül a kivitelezni kívánt perszonalizációs funkcionalitásnak megfelelően kell kiválasztani a vizsgálni kívánt adatokat. Ezeket általában a megfelelő előfeldolgozás és mintafelismerés eléréséhez ki kell egészíteni a weboldalakon található meta-adatokkal, és a megfelelő konklúziók levonásához az adott szakterület ismerete is elengedhetetlen.

A memorizációs funkciók általában explicit felhasználói inputot igényelnek, már az egyszerű üdvözléshez is szükséges regisztráció, hogy a rendszer hozzájuthasson a felhasználó nevéhez. A felhasználónév tárolható helyi adatbázisban vagy cookie fájlban. További regisztrációs adatok szükségesek személyre szabott hozzáférési politikák kialakításához. Könyvjelzők viszont egyszerűen megvalósíthatóak a felhasználó által meglátogatott oldalak lekövetésével, naplófájlok vagy kliensoldali agent-ek által gyűjtött adatok alapján.

Útmutatási funkciók megvalósításához tipikusan arra vonatkozó információkra van szükség, hogy a felhasználó mit keres egy weblapon, a felhasználó tudásszintjére vonatkozó információk mellett. Ezek szerveroldali és kliensoldali információk együtteséből nyerhetőek ki, vagy akár közbenső információforrásokból is.

A testreszabás lehetőségének biztosításához főleg a felhasználó érdeklődésére és preferenciáira vonatkozó információ szükséges, amit a felhasználó böngészési előzményeiből (browsing history) lehet kinyerni, általában a szerver naplófájljai alapján. Az egyszerű naplófájlokat kiegészíthetik vásárlási információkkal az adott cég megfelelő adatbázisaiból, a testreszabott árazási séma megvalósítását lehetővé téve.

A feladatvégzési támogatás olyan adatok gyűjtését igényli, amik felfedhetik a felhasználó szándékát egy bizonyos feladat elvégzésére. Ez adatgyűjtő módszerek egy kombinációjával érhető el. A szerver naplóiból származó és kliensoldali agent-ekkel gyűjtött információk le tudják írni egy felhasználó böngészési viselkedését, és alkalmazhatóak a

felhasználó szándékainak kikövetkeztetésére. Azonban sokszor ezen szándékok egy sokkal pontosabb átlátása szükséges, amit csak regisztráció vagy lekérdezések során begyűjtött információk felhasználásával érhetünk el.

4.2. Előfeldolgozás

Ha kiválasztottuk és összegyűjtöttük a szükséges adatokat, azokat az adatbányászati alkalmazások többségéhez hasonlóan meg kell tisztítani. Ki kell szűrni a redundáns és irreleváns adatokat, a hiányzó értékeket becslésekkel kell pótolni, zajszűrést kell végezni, az esetleges inkonzisztenciákat fel kell oldani. Ezek elvégzésével elkerülhető, hogy a felhasználó viselkedésével semmilyen összefüggésben nem álló információk félrevezessék a mintafelismerési eljárást. Nagyobb szabású weboldalak esetén gyakori, hogy a felhasználóknak nyújtott tartalmakat több web- vagy alkalmazásszerver szolgáltatja. Bizonyos esetekben több, redundáns tartalmú szervert alkalmaznak a szerverek terhelésének csökkentésére. Ez esetben adatfúziót alkalmaznak a különböző szerverek naplófájljainak egyesítésére. Ha a szerverek nem használnak megosztott session-azonosítókat (session id), az egyesítést heurisztikus alapon végzik.

Az adattisztítás általában oldalspecifikus. Eltávolítják a külső hivatkozásokat az elemzés szempontjából érdektelen beágyazott tartalmakra, ideértve a stílusfájlokra, grafikára és hangokra való hivatkozásokat. A folyamat része lehet néhány adatmező eltávolítása (pl. átvitt bájtok száma vagy a http protokoll típusa). A crawlerek navigációja által létrehozott bejegyzések eltávolítása is szükséges, mivel a crawlerek nem valós felhasználók, hanem általában keresőmotorokhoz használt webbejáró alkalmazások, de nem ritkán egy napló bejegyzéseinek akár fele is származhat tőlük.

4.2.1. Absztrakció

A tisztítás végeztével az elemzés céljaitól függően az adatokat át kell alakítani és össze kell vonni különböző absztrakciós szinteken. A webhasználat-bányászatban a legalapvetőbb absztrakciós szint az oldalnézet. Az oldalnézet webes objektumok egy kollekciójának együttes megjelenítése a felhasználó böngészőjén keresztül, amit egyetlen felhasználói tett vált ki. Koncepció szintjén minden oldalnézet megjeleníthető webes objektumok egy kollekciójaként, amik egy specifikus „felhasználói eseményt” reprezentálnak, pl. egy cikk olvasását, egy termék megtekintését, vagy egy termék hozzáadását a kosárhoz.

A felhasználó szintjén a legalapvetőbb viselkedési absztrakciós szint a session. A session egy oldalnézet-sorozat, amit egy felhasználó tekintett meg egy látogatása során. Ez továbbabsztrahálható az oldalnézetek egy részhalmazának kiválasztásával, amik szignifikánsak, vagy relevánsak a vizsgálat szempontjából, ezt nevezik epizódnak vagy tranzakciónak is.

Az előfeldolgozás lényegi része az egyes absztrakciós szintek felismerése és behatárolása. Először az oldalnézeteket kell felismerni. Az adott oldalnézethez a rendszer rugalmasságának növelése érdekében alapvető alkotóelemein kívül további adatok is hozzárendelhetőek, mint pl. az oldalnézet azonosítója (rendes esetben az URL egyedi módon reprezentálja az oldalnézetet), az oldalnézet statikus típusa (információs lap, kategórianézet, terméknézet, indexoldal), és más metaadatok, pl. a tartalmának attribútumai (kulcsszavak vagy az adott termék attribútumai).

A perszonalizációs rendszerek szempontjából viszont a legkritikusabb pont a különböző felhasználók megkülönböztetése, mivel azok megfelelő felismerése nélkül nem rendelhető hozzájuk egyéni viselkedés. Mivel egy felhasználó többször is meglátogathat egy weboldalt, a szervernaplók minden felhasználóhoz több session-t rögzítenek. A felhasználói aktivitási rekord kifejezést használják az egyazon felhasználóhoz tartozó naplózott tevékenységek sorozatára. Autentikációs mechanizmusok hiányában a legelterjedtebb módszer a felhasználók megkülönböztetésére a kliensoldali cookie-k használata, bár ezt biztonsági okokból sokszor letiltják. Az IP címek önmagukban viszont nem elegendőek az egyedi felhasználók azonosítására, mivel az internetszolgáltatók proxy-jai dinamikusan osztják ki az IP-ket. Egyesek kísérleteztek az egyes felhasználók az IP mellett más adatok, pl. a használt operációs rendszer vagy böngésző típusa és verziószáma alapján azonosítani felhasználókat, és pl. Schwarzkopf (2001) alternatív félautomatikus könyvjelzőalapú megoldással próbálkozott.

Ha sikerült azonosítani egy felhasználót, akkor a hozzá tartozó felhasználói tevékenységi rekordot szegmentálni kell session-ökre, amelyek mindegyike egy az oldalra történő látogatást reprezentál. Amennyiben nincs autentikáció, ennek elérésére is szükség lesz heurisztikus módszerekre. A session-felbontáshoz alkalmazott heurisztikák alapvetően két csoportba sorolhatóak, lehetnek időalapúak vagy struktúraalapúak. Az időalapú heurisztika globális vagy lokális időbecsléseket készít, pl. az egy oldalon töltött idő vagy a teljes session időtartamának maximalizálásával. A struktúraalapú heurisztika az oldal statikus szerkezetét

vagy a szervernaplók ajánló (referer) mezőjéből kinyert implicit linkstruktúrát alapul véve szerkezetileg összetartozó oldalakból épít fel egy session-t.

Sikeres session-ökre bontás után általában a kliens- és proxy-oldali cache-elés miatt szükség lehet a cache-elt oldalakra vonatkozó kérések utólagos pótlására. A hiányzó hivatkozásokat heurisztikus úton ki lehet egészíteni, az oldal struktúrájáról rendelkezésre álló tudás és a szervernaplókból származó információk alapján.

4.2.2. Adatmodellezés, adatábrázolás

Az előfeldolgozási feladat végül felhasználói session-ök vagy epizódok egy halmazát eredményezi, amik mind egy lehatárolt oldalnézet-szekvenciát jelképeznek. Tehát az eredmény egy n db oldalnézetből álló halmaz, $P = \{p_1, p_2, \ldots, p_n\}$, és egy m elemű tranzakcióhalmaz, $T = \{t_1, t_2, \ldots, t_m\}$, ahol minden T-beli t_i P egy részhalmaza, az oldalnézetek pedig szemantikailag értelmes entitásokat jelképeznek, amikre alkalmazzák a bányászati módszereket, mint pl. oldalak vagy termékek. Koncepció szintjén minden t tranzakciót egy l hosszúságú rendezett elempár-sorozatnak tekintünk:

$$t = \langle (p_1^t, w(p_1^t)), (p_2^t, w(p_2^t)), \ldots, (p_l^t, w(p_l^t)) \rangle,$$

ahol minden $p_i^t = p_j$ valamely j ∈ {1, 2,…,n}-re, és $w(p_i^t)$ az adott p_i^t t tranzakcióbeli oldalnézethez hozzárendelt súly, ami reprezentálja a szignifikanciáját. A súlyokat számos módon meg lehet határozni, részben az elvégzendő vizsgálat vagy a tervezett perszonalizációs keretrendszer milyensége alapján. Pl. a kollaboratív szűrési alkalmazások esetében, amik hasonló felhasználók profilja alapján tesznek ajánlatot az adott felhasználónak, a súlyok alapulhatnak az elemek felhasználói értékelésein. Legtöbb esetben a súlyok vagy binárisak, ezzel reprezentálva az adott oldalnézet bele- vagy bele nem tartozását az adott tranzakcióba, vagy lehetnek az adott felhasználói munkamenet során az adott oldalnézet időtartamának kifejezői. A gyakorlatban általában normalizált értéket használnak a nyers időadatok helyett a felhasználók varianciájának figyelembe vétele céljából. Néhol az oldalnézetek időtartamának logaritmusát alkalmazzák, mert ezzel csökkenthető a zaj.

A tranzakciók súlyainak vektoraként előállítható a tranzakcióvektor, $t = (w_{p1}^t, w_{p2}^t, \ldots, w_{pn}^t,)$ alakban, ahol minden $w_{pi}^t = w(p_j)$ valamely j ∈ {1, 2,…,n}-re, ha p_j megjelenik a t tranzakcióban, és $w_{pi}^t = 0$ egyébként. Ebből adódóan az összes felhasználói tranzakció tekinthető egy m×n-es mátrixként, amit felhasználói oldalnézetmátrixnak vagy tranzakciómátrixnak neveznek. Erre a mátrixra számos tanuló algoritmusos technika

alkalmazható mintakinyerés céljából, pl. a tranzakciók vagy munkamenetek klaszterezése, az elemek klaszterezése vagy asszociációfeltárása fontos kapcsolatokat fedhet fel az elemek között.

Ha az oldalak mögöttes szemantikai tartalmát is integrálni akarjuk a modellbe, mivel az oldalnézetek a szemantikát reprezentáló szöveges elemeket is tartalmaznak, minden p oldalnézet reprezentálható egy r-dimenziós elemvektorral, ahol r az összes az oldalakból kinyert elemek (szavak és koncepciók) száma. Ezt a vektort

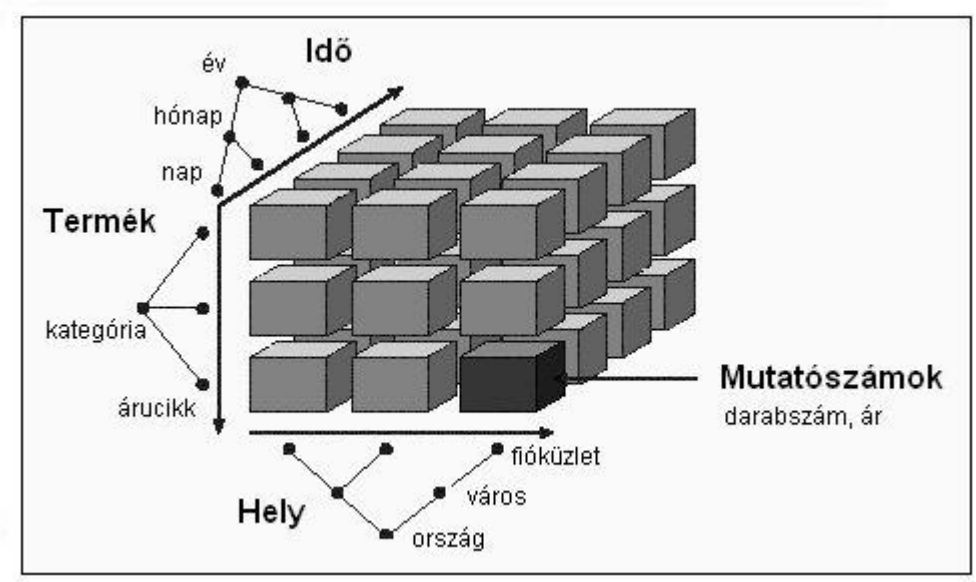
$$p = \left(fw^p(f_1), fw^p(f_2), \ldots, fw^p(f_r)\right)$$

formában adhatjuk meg, ahol $fw^p(f_j)$ a j-edik elem súlya a p oldalnézetben, $1 \leq j \leq r$-re. Az oldal összes oldalnézetére ez megad egy n×r-es oldalnézet-elemmátrixot. Az adatok effajta integrálásának célja minden felhasználói munkamenet, vagy még inkább minden felhasználói profil szemantikus elemek vektoraként történő ábrázolása egy oldalnézetek fölötti vektor helyett. Így nem csak a meglátogatott oldalakat, de a felhasználói interakció szempontjából fontos különböző koncepciók és konkrét elemek szignifikanciáját is megmutatja egy felhasználói munkamenet. A szemantikus tartalmak integrációja a webhasználat-bányászatba potenciálisan jobb megértéséhez vezethet az objektumok közötti mélyebb kapcsolatoknak.

A naplókon kívüli más adatforrásokból származó adatok integrációja az előfeldolgozott clickstream adatokba fontos üzleti intelligenciai metrikák feltárását segíthetik, mint pl. vásárlói átváltási arányok és élettartam-értékek. Ezen adattípusok sikeres integrációja egy oldalspecifikus „esemény-modell" elkészítéséhez vezetnek, amire alapozva egy felhasználóhoz tartozó clickstream részhalmazait csoportosítják és leképezik speciális eseményekre, mint pl. egy termék hozzáadása a kosárhoz.

Általánosságban az integrált üzleti adatok egy végső tranzakció-adatbázisban kerülnek tárolásra, amely lehetséges az előbbiekben vázolt vagy ahhoz hasonló egyszerű módon, de hogy teljes értékű webes elemzési módszereket alkalmazhassanak rájuk, ezeket az adatokat általában egy adatpiacon tárolják. Ennek megvalósítása általában egy többdimenziós adatbázissal történik, ami különböző forrásokból származó, és különböző aggregációs szinteken lévő adatokat egyesít, és előre kiszámolt metrikákat adhat több dimenzióban. Az ilyen multidimenzionális adatabsztrakció sok előnnyel jár, ezen alapszik pl. az OLAP (On Line Analitical Processing) által realizált online elemzési funkcionalitás. Multidimenzionális adatabsztrakció alatt értjük az adatok egy speciális, könnyen elemezhető és képszerű kezelési módszerét, mely szerint az adatainkat úgy kezeljük, mintha egy n-dimenziós kocka pontjai

lennének, ahol az egyes dimenziók megfelelnek egy-egy mérhető attribútumnak. Az ábra egy ilyen adatkockára vonatkozó példát mutat egy termékajánló rendszer vonatkozásában.



Adatkocka felépítése. Forrás: Sidló, 2003

4.3. Mintafelismerés

Az előfeldolgozáson átesett és a felhasználó- vagy eseménymodellbe beépített adatokon végzett elemzés típusai és szintjei, amit az integrált adatokon végeznek, függenek az elemzés végső céljaitól, és a kívánt eredményektől. A meglévő perszonalizációs rendszerek túlnyomó többsége a link vagy termékajánlások funkcionalitására koncentrál, amit általában tartalomalapú vagy kollaboratív szűrőrendszereken keresztül valósítanak meg. A tartalomalapú szűrők az adott oldalnézetben vagy termékhez kapcsolódó szöveges, tartalmi információk alapján keresnek az adott felhasználó számára kívánatos termékeket, míg a kollaboratív szűrők a felhasználók hasonlóságára koncentrálnak, és más hasonló felhasználók preferenciái alapján teszik meg ajánlataikat. Emellett a meglévő rendszerek adataikat kizárólag vagy legnagyobb részben a szerverek hozzáférési naplóiból nyerik ki, ezért az alkalmazott webhasználat-bányászaton alapuló megoldások nagy része is ezen körülményekhez idomulva alakult ki. Ezek legjelentősebb irányzatai a következők:

4.3.1. Session és felhasználó-elemzés

Sessionök vagy felhasználók elemzéséhez az adatokat előre definiált egységekbe rendezik, pl. napok, sessionök, felhasználók vagy érdeklődési területek szerint. Ezt a megközelítést választja a legtöbb jelenleg forgalomban lévő üzleti webes naplóelemző eszköz.

Ide tartozhat a leggyakrabban látogatott oldalak, egy oldal átlagos látogatási idejének, egy oldalon keresztülvezető út átlagos hosszának, a leggyakoribb be- és kilépési pontok, és más csoportosított mértékek kinyerése. Az eredményül kapott tudás hasznos lehet a rendszer teljesítményének növelése és marketing döntések támogatása szempontjából. Ezek a módszerek legnagyobb részben az előfeldolgozási lépéseken alapulnak.

4.3.2. Klaszteranalízis és a felhasználók szegmentációja

A felhasználók klaszterezése a hasonló böngészési mintákkal rendelkező felhasználókat rendezi egy csoportba. Az ilyen tudás nagyon hasznos a felhasználók demográfiai adatainak kikövetkeztetésére piaci szegmentáció érdekében, vagy személyre szabott tartalom nyújtására a hasonló érdeklődésű felhasználóknak. A csoportok további vizsgálata demográfiai adataik alapján (kor, nem, jövedelemszint, stb.) értékes üzleti tudás kinyeréséhez vezethet. A használatalapú klaszterezést használták webalapú felhasználói közösségek létrehozására közös érdeklődés alapján.

4.3.3. Asszociációk és korrelációk vizsgálata

Asszociációs szabályok feltárása és egy statisztikai korrelációvizsgálat olyan elemcsoportokat tárhat fel, amiket együtt érnek el és vesznek meg, tehát úgynevezett gyakori elemhalmazok tárhatóak fel a segítségükkel. Ez segíthet a weboldalak hatékonyabb elrendezésében és termékajánlatoknál.

A legtöbb elterjedt asszociáció-feltárási megközelítés még mindig az egyik legrégebbi adatbányászati módszeren, az Apriori algoritmuson alapszik.

Algorithm 1 Apriori

Require: $\mathcal{T}$: tranzakciók sorozata,
min_supp: támogatottsági küszöb,

$\ell \leftarrow 0$
$J_\ell \leftarrow \{\emptyset\}$
while $|J_\ell| \neq 0$ **do**
  támogatottság_meghatározás($\mathcal{T}, J_\ell$);
  for all $j \in J_\ell$ **do**
    if $supp(j) \geq min\_supp$ **then**
      $GY_\ell \leftarrow j$;
    end if
  end for
  $J_{\ell+1} \leftarrow$ jelölt_előállítás(GY_ℓ);
  $\ell \leftarrow \ell + 1$;
end while
return GY

Az Apriori adatbányászati algoritmus. Forrás: Bodon, 2008

Az Apriori algoritmus szélességi bejárást valósít meg, az üres halmaztól kezdve iterációnként az előzőnél eggyel nagyobb elemszámú gyakori mintákat keres, és itt a kinyert l elemszámú gyakori mintákat a GY_l halmazban tárolja.

Webhasználat-bányászati esetben ez azt jelenti, hogy az algoritmus az előfeldolgozott naplóban megjelenő oldalnézetek olyan csoportjait fedezi fel, amelyek gyakran fordulnak elő együtt a tranzakciókban. Ezután egy minimális szignifikanciaküszöböt elérő asszociációs szabályok generálása következik. Az ilyen szabályok használhatóak az oldal struktúrájának optimalizálására. Pl. ha nincs közvetlen összeköttetés két gyakran együtt előforduló elem között, akkor a két oldal összekötése segíthet a felhasználónak megtalálni a keresett információt. Ezeket sikerrel alkalmazzák a testreszabás és az ajánlórendszerek területén.

Gyakori alkalmazási forma a top-N típusú ajánlórendszer is, amely döntési szabályokon alapszik. A célfelhasználó preferenciáit összehasonlítja a meghatározott szabályok előtagjában szereplő formulával, és ezután a szabályok jobb oldalán lévő elemeket az elért konfidenciaértékek szerint rendezi, és ezek közül az N legnagyobb értéket kapott elemet listázza ki ajánlásként a felhasználónak. Többek között ennek a megoldásnak is komoly problémája, hogy nem tud ajánlatokat tenni ritka adatbázisból, ami gyakran előfordul, pl. új felhasználó esetében, aki még nem tett semmilyen értékelést a rendszerben.

Ennek megoldását célozta a kollaboratív szűrés technikája, ami közeli szomszédokat talál, akik ízlése nagyban hasonlít a célfelhasználóéhoz, és az ő értékeléseik alapján ad ajánlatokat. A Lin et al. (2002) által kidolgozott asszociációs szabály alapú kollaboratív ajánlórendszer a felhasználók és a termékek között is hoz létre asszociációkat, és utóbbiakat a felhasználók közötti szabályok konfidenciaszint alatt maradása esetén alkalmazza.

Ezek a rendszerek a teljes egyezés nehéz létrejötte miatt egy elcsúszó ablakot alkalmaznak a célfelhasználó aktív profilja vagy sessionje felett, és csak az ablakon belül eső elemek kerülnek összehasonlításra, miközben az ablak mérete iteratív módon csökken az egzakt egyezésű előtaggal rendelkező szabály megtalálásáig. Probléma, hogy ez a szabályok halmazának iteratív keresésével jár, ami időigényes, de a rendszerek hatékonysága gráfalapú reprezentációval nagyban növelhető.

4.3.4. Osztályozási feladatok, előrejelzési módszerek

Az osztályozás feladata általánosságban az adatelemek valamelyik előredefiniált osztályba való besorolása. Webes körökben ez általában egy felhasználói profil valamelyik

kategóriába vagy osztályba való besorolásához szükséges. Ehhez az adott osztályt legjobban leíró tulajdonságok vagy elemek kinyerése és kiválasztása szükséges. Az osztályozás elvégezhető olyan tanuló algoritmusokkal, mint pl. döntési fák, k legközelebbi szomszéd alapú osztályozók, stb., de korábban feltárt klaszterek is használhatóak új felhasználók osztályozására.

Az osztályozási és előrejelzési feladatok fontos használati módja a két korábban említett automatikus szűrő-alapú megközelítés. A tartalomalapú szűrés könnyen problémákba ütközhet, mivel a weboldalak tartalmának elemzése és a szemantikai hasonlóságok felfedezése meglehetősen nehéz lehet. Még ha a multimédiás tartalmakat figyelmen kívül is hagyjuk, a természetes nyelv akkor is nagyon gazdag és strukturálatlan információforrást nyújt, melynek automatizált értelmezése nagyon komoly kihívás, és ráadásul a szöveges tartalmak mennyisége sok esetben erősen korlátozott.

A tartalom szerepének csökkentésével ezt a problémát kívánja orvosolni a kollaboratív szűrés technikája. Továbbá a kollaboratív szűrés elősegíti olyan használati sémák feltárását, amelyek nem behatárolhatóak szigorú szemantikai szabályokkal. Ugyanakkor ez a módszer sem problémamentes. Felismertek már a kollaboratív szűrők számára problémás eseteket, amikor nem áll rendelkezésre elég értékelés, a túl kevés felhasználó vagy az egy felhasználóra jutó túl kevés értékelés miatt, ami a következő esetekben fordulhat elő:

- új felhasználó esete: kevés vagy semennyi értékelése van, nem megbízhatóan összevethető más felhasználókkal
- új objektum esete: ha egy új objektum kerül a gyűjteménybe, és még senki nem értékelte, elhanyagolttá válhat
- rendszertöltés esete: ez a fenti 2 kombinációja, és tipikus az új ajánlórendszerekre, amik összességében rendelkeznek kevés értékeléssel. Nehéz elcsábítani a felhasználókat, mivel kezdetben nagyon gyenge ajánlásokat kapnak, és így a rendszer rendszertöltési dilemmában maradhat. (Kohrs és Merialdo, 1999)

Ezzel szemben a tartalomalapú sémák kevésbé érzékenyek az értékelések ritkaságára. Ráadásul a pusztán memóriaalapú tanuló algoritmusokat használó kollaboratív szűrő rendszerek két további probléma által is terheltek: nem skálázhatóak megfelelően nagy adatmennyiségekre, és semmilyen betekintést nem nyújtanak az adatokban felismert használati sémákba (Pennock et al., 2000). Nemrégiben kezdtek el megoldást keresni ezen problémákra, modellalapú illetve hibrid memória- és modellalapú kollaboratív szűrő

rendszerek fejlesztésével. A teljesítmény javítására többen javasolták már a kollaboratív és tartalomalapú rendszerek kombinációját is.

Technikai hátterüket tekintve a legtöbb kollaboratív szűrő alkalmazás és meglévő ajánló rendszer a k legközelebbi szomszéd módszerén alapuló osztályozókat használ. Ez alapvetően a célfelhasználó aktivitási rekordjának összehasonlítását jelenti más felhasználók tevékenységi rekordjainak halmazával, T-vel, hogy megtalálják a k leghasonlóbb ízlésű és érdeklődésű felhasználót. Ennek modellezése alapulhat az elemek értékelésének, az oldalhozzáférések vagy a megvett termékek hasonlóságán. A legtöbb kollaboratív szűrőrendszerben a felhasználói profilok értékeléshalmazokat jelentenek elemek halmazai felett. A kinyert szomszédsági viszonyok alapján korábban nem hozzáfért elemeket vagy meg nem vett termékeket ajánlanak a rendszerek.

A folyamat során a célfelhasználó *u* aktív sessionje összehasonlításra kerül az összes korábbi *v* tranzakciós vektorral, ahol $v \in T$. A k darab *u*-hoz leghasonlóbb tranzakciót tekintjük az *u* session környezetének. Az *u* és *v* közötti hasonlóság a Pearson korrelációs együtthatóval számolható ki, melynek képlete:

$$sim(\mathbf{u},\mathbf{v}) = \frac{\sum_{i\in C}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})}{\sqrt{\sum_{i\in C}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{i\in C}(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})^2}},$$

ahol *C* az *u* és *v* által is értékelt elemek halmaza, $r_{u,i}$ és $r_{v,i}$ az *i* elemre *u* illetve *v* által adott értékelések, és $\bar{r}_u$ és $\bar{r}_v$ az *u*-hoz és *v*-hez tartozó értékelési átlagok. A hasonlóságok kiszámolása után kiválasztásra kerülnek a leghasonlóbb felhasználók.

Általánosnak mondható, hogy az egy bizonyos küszöbértéknél kisebb hasonlóságú szomszédokat kiszűrik, hogy megakadályozzák a nagyon távoli vagy negatív korrelációkon alapuló ajánlásokat.

A hasonlóságok feltárása után egy *i* elemre az *u* felhasználó által adott értékelés előrejelzését a következő formulával számítják ki:

$$p(\mathbf{u},i) = \bar{r}_{\mathbf{u}} + \frac{\sum_{\mathbf{v}\in V} sim(\mathbf{u},\mathbf{v}) \times (r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})}{\sum_{\mathbf{v}\in V} |sim(\mathbf{u},\mathbf{v})|},$$

ahol *V* a k darab hasonló felhasználó halmaza, $r_{v,i}$ a *V*-beli felhasználók értékelései *i*-re, és *sim(u,v)* a Pearson korrelációs együttható. A formula hasonlóságuk fokával súlyozva megadja minden szomszéd preferenciaszintjét az adott elemre, és aztán ezt hozzáadja a

célfelhasználó átlagos értékelési értékéhez. Ennek oka az az alapötlet, hogy a felhasználóknak különböző „alapértékük” van, ami körül az értékeléseik eloszlanak.

Ennek hátránya a már említett skálázhatóság hiánya, valós idejű összehasonlítást követel meg minden felhasználói rekorddal, hogy előrejelzéseket generálhasson. Ennek a problémának a megoldására született a módszer egy variációja, a termékalapú kollaboratív szűrés. Működése során termékeket hasonlít össze a felhasználók által rájuk adott értékelések mintázatai alapján. Ez is k legközelebbi szomszéd módszert alkalmaz, ami ez esetben k darab hasonló elemet keres hasonló felhasználók helyett, melyek különböző felhasználók együttes értékelései alapján hasonlóak. Az általában használt hasonlósági mérték a javított koszinusz hasonlóság:

$$sim(i,j) = \frac{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})(r_{\mathbf{u},j} - \bar{r}_{\mathbf{u}})}{\sqrt{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},j} - \bar{r}_{\mathbf{u}})^2}},$$

ahol U az összes felhasználó halmaza, i és j termékek, $r_{u,i}$ az $u \in U$ felhasználó i-re adott értékelése, és $\bar{r_u}$ az u felhasználó átlagos értékelése. Ez a termékek hasonlóságát méri az egyes felhasználók által rájuk adott értékelések alapján. A hasonlóságok kiszámítása után kiválasztják az adott termékhez leghasonlóbb k elemet, és a következő formulával generálják az u felhasználó várható értékelését az adott elemre:

$$p(\mathbf{u}, i) = \frac{\sum_{j\in J} r_{\mathbf{u},j} \times sim(i,j)}{\sum_{j\in J} sim(i,j)},$$

ahol J a k hasonló termék halmaza, $r_{u,j}$ az u felhasználó értékelése a j elemre, és $sim(i,j)$ az i és j termékek fentebb definiált hasonlósága. Általában figyelmen kívül hagyják a negatív hasonlóságú termékeket a téves ajánlások elkerülése érdekében. Ezen módszer alapötlete a felhasználó saját, hasonló elemekre adott értékeléseinek felhasználása a céltermék értékelésének kikövetkeztetésére.

4.3.5. Módszerek alkalmazása

A klaszterezés, osztályozás, és asszociációfeltárás gyakran használt metódusok a perszonalizációs rendszerekben, a használati minták felismerésére ezek közül a klaszterezés bizonyult a leghatékonyabbnak, főleg hozzáférési naplókból származó adatok esetében, ahol

nem állt rendelkezésre semmilyen előzetes tudás előredefiniált osztályok formájában, mint ahogy az általában lenni szokott.

Azonban majdnem minden klaszterezést alkalmazó rendszer figyelmen kívül hagy a navigációs viselkedés vizsgálatának egy fontos szempontját, a munkamenet lekéréseinek sorrendjét. Ez általában az adatreprezentációra vezethető vissza, mivel az oldalnézeteket gyakran egyszerű halmazokban tárolják. Ennek kiküszöbölése megoldható lenne sorrendiséget is nyilvántartó adatreprezentációval.

Az asszociációs szabályok bányászata is jó módszer az egyszerű logokból való információkinyerésre, főleg webes elemek, pl. oldalak vagy termékek közötti kapcsolatok feltárására. Meglepő, hogy a szekvenciális minták felismerését nagyon ritkán alkalmazzák, annak ellenére, hogy a felhasználók navigációs viselkedésének modellezésére különösen alkalmas. Ennek egyik oka lehet az eredményül kapott modellek értelmezésének nehézsége.

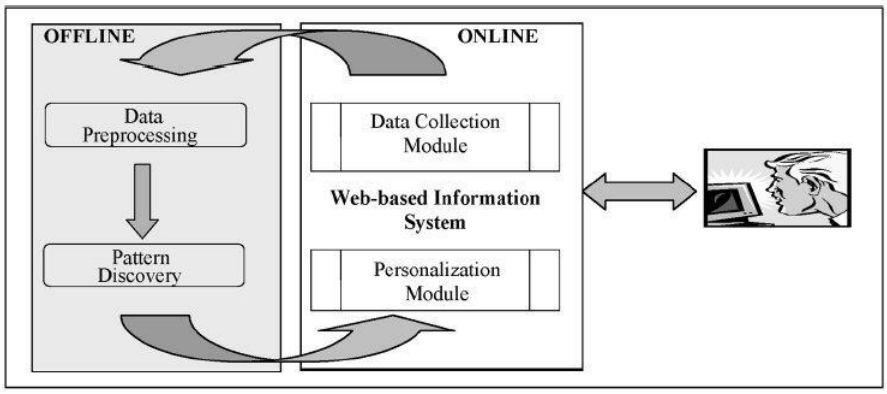
4.4. Utófeldolgozás, a folyamat integrálása a perszonalizációs rendszerbe

A legtöbb webhasználat-bányászati módszer esetén alkalmaznak egy utófeldolgozási fázist, melyben a feltárt sémákat szűrik és elemzik, hogy megkönnyítsék a döntéshozatalt az emberi szakértőknek, azaz a weboldalak adminisztrátorainak. Bizonyos esetekben azon emberek döntései, akik megkapták a kinyert tudást, elvezethet a webes szolgáltatások perszonalizációjához, de a webhasználat-bányászat még ez esetben sem szerves része a perszonalizációs folyamatnak. Továbbá a kinyert tudás manuális feldolgozásának igénye késleltetést és információveszteséget von maga után.

Ezzel szemben egy másik megközelítés a webhasználat-bányászat beépítése a perszonalizációs folyamatba azáltal, hogy a kinyert tudást közvetlenül betáplálják a perszonalizációs modulba, ami annak megfelelően fogja alakítani a webalapú rendszer viselkedését.

Így a tudás a felhasználókhoz közvetlenül, emberi beavatkozás nélkül jut el, egy vagy több perszonalizációs funkció formájában. A felkínált perszonalizációs funkció nagyban függ az adott rendszer által követett perszonalizációs politikától, azaz a módtól, ahogy eljuttatja az információt a végfelhasználóhoz.

Ezen megközelítést követve a rendszer jellemzően két részre osztásra kerül, az online és az offline részre, Mobasher et al. (2000) javaslata alapján.



A webbányászat-alapú perszonalizációs rendszer. Forrás: Pierrakos et al., 2003

Az online modulok azok, amelyek begyűjtik a használati adatokat, és a konkrét perszonalizációt végzik a kinyert tudás alapján. Az adatbányászati folyamat összes többi fázisa jellemzően offline történik, annak érdekében, hogy ne rontsák a rendszer teljesítményét.

Érdekes, hogy a módszer hatékonysága és csekély utólagos munkaigénye ellenére a perszonalizációs rendszerek nagy része nem alkalmaz webhasználat-bányászati eszközöket.

5. A webhasználat-bányászat alapú perszonalizációs rendszerek problémái és felmerülő kérdései

5.1. Alkalmazott funkcionalitások

A meglévő perszonalizációs rendszerek legnagyobb része a hiperlinkek ajánlása terén nyújt funkcionalitást, ami valószínűleg annak tudható be, hogy a web perszonalizáció terén végzett kutatások elsősorban az információtúltengés kiküszöbölésére kezdődtek el. Emiatt lehetséges az is, hogy az erre a területre fejlesztett webhasználat-bányászati módszerek nagyrésze is linkek ajánlatait számolja ki. Ezután a második leggyakrabban alkalmazott funkcionalitás az oldalak testreszabásának lehetősége, bár ez sem túl gyakori. Más elméletileg létező, és a felhasználó segítésére nagyon alkalmas funkcionalitásokat pedig teljesen elhanyagolnak, pl. oktatási vagy magyarázó funkcionalitással nem lehet találkozni, és feladatvégzési támogatást nyújtó rendszerekkel is alig.

Az alkalmazott perszonalizációs politikát tekintve a több felhasználós szemléletmód az elterjedtebb, mivel ez nem igényel autentikációt, és nem kell minden egyes felhasználót külön modellezni, ami nagyobb rendszereknél gondot jelenthetne. Az oldalak tartalmi céljait illetően a konvergens szemlélet a jellemzőbb, mivel a weboldalak általában erősen tematikusak, és

tulajdonosaiknak nem fűződik érdeke az oldaluk elhagyására való ösztönzéshez. Érdekes, hogy abszolút nem jellemző a magyarázatok használata, pedig ezzel elérhető lenne a perszonalizációs funkciók transzparenssé tétele, amivel meg lehetne nyerni az adataik biztonsága miatt aggódó felhasználók bizalmát. A bizalmatlanság megszüntetése pedig fontos lenne, hogy a komplexebb perszonalizációs funkcionalitások elterjedhessenek.

5.2. Biztonsági kérdések

A webbányászat, különösen a webhasználat-bányászat alkalmazásánál nagyon fontos a felhasználói adatok biztonságának kérdése, mivel a felhasználó minden egyes tevékenységét rögzítő adatokkal dolgozik, így valamilyen szinten jogos a felhasználók félelme az ilyen módszerek alkalmazásától. Legtöbbjük számára a biztonságuk és anonimitásuk biztosítása még kényelmüknél is előbbre való, ezért feltétlenül biztosítani kell az adatbiztonságot. Kobsa (2001) szerint ahhoz, hogy egy web perszonalizációs rendszer megfeleljen a biztonsági elvárásoknak, a következő feltételeknek kell teljesülnie:

- az adatokat és a metódusokat intelligensen közzé kell tenni, hogy ez elősegítse a funkciók működésének és a róla gyűjtött adatok természetének megértését a felhasználó által
- gondoskodás a felhasználó technikai lehetőségeiről felhasználói modelljének módosítására, azaz a felhasználó beengedése a perszonalizációs folyamatba
- felhasználómodell-szerverek, melyek több anonimizáló metódust támogatnak, hogy a felhasználók megvédhessék anonimitásukat
- a felhasználómodellező metódusok a biztonsági beállításokhoz és a jogi környezethez való igazítása, ami magasabb fokú rugalmasságot biztosít a web perszonalizációs rendszereknek.

Tehát nagyon fontos, hogy az új webhasználat-bányászati módszerek legyenek átláthatóak a felhasználó számára, azáltal, hogy elérhetővé teszik az összegyűjtött információkat, és tisztázzák azok felhasználását, azok potenciális hasznával együtt. Ennek megvalósítására egy magyarázó perszonalizációs politika követése lehet a megoldás.

5.3. Változáskezelés

A web perszonalizációban használt tanuló algoritmusokkal szemben felmerül a követelmény, ha az általuk létrehozott modelleket képesek legyenek inkrementálisan

frissíteni, ugyanis a weboldalak dinamikus természetéből fakadóan nem feltételezhető, hogy adataik később is elérhetőek lesznek az adatbegyűjtés után.

Másik kérdés az időtényező a modellekbe való beépítésének igénye. A felhasználók viselkedése idővel változik, és ennek hatással kellene lennie a modellalkotásra. Egy web perszonalizációs rendszernek képesnek kellene lennie alkalmazkodni a felhasználó viselkedéséhez, amikor az megváltozik, nem feltételezhető egy felhasználóról, hogy nagyon hosszú távon is fennmarad az először róla kialakult kép időszerűsége.

Ezen tényező modellezése azonban igen nehéz feladat, mivel nagyon erősen függ a felhasználó egyéni tulajdonságaitól és az oldal szakterületi hovatartozásától.

5.4. Teljesítmény

Komoly probléma a mintafelismerő módszerek nagy részével, hogy nehezen kezelik a webre jellemzően nagy adathalmazokat. Annak ellenére, hogy a webhasználat-bányászati folyamat nagy része megoldható offline, a webes adatok mérete, főleg a hozzáférési naplóké, nagyobbak a tanuló algoritmusok átlagos alkalmazásainál tapasztaltaknál. Egy olyan perszonalizációs rendszernél, melynek valós időben kell működnie, a mintafelismerési módszer skálázhatósága kritikus ponttá válhat. Ennél még fontosabb lehet a felismert mintákat beépítő perszonalizációs modul számítási teljesítménye. Ezért problematikusak a memóriaalapú algoritmusok, melyek elodázzák az általánosítást a futásig, míg a modellalapúak jobb megoldást biztosítanak, mivel az eredi adatokat általánosított modellekbe sűrítik be. Azonban ezen modellek skálázhatósága is különös odafigyelést igényelne, és feltehetően a használt adatszerkezetek megújítására is szükség lenne.

Ezt a problémát tovább mélyíti, hogy érdemben még nem foglalkoztak a perszonalizációs rendszerek teljesítményének mérésével, bár ez valószínűleg annak tudható be, hogy nagyon nehéz objektív mérési szempontokat találni egy ilyen komplex rendszer használatára.

5.5. Hordozhatóság

A perszonalizációs rendszerekkel és más webbányászati alkalmazásokkal kapcsolatban is felmerült az a praktikus igény, hogy egységesítsék a kinyert tudás ábrázolását, azaz perszonalizáció esetében a legenerált felhasználómodelleket. A korábban kifejlesztett rendszerek mind saját formátumban tárolták le a kinyert tudást, ami csak saját rendszerükkel

kompatibilis, és nem osztható meg másokkal. Egy ezeknél alkalmazkodóképesebb reprezentációs szabvány elősegítheti a rendszerek együttműködését. Pl. Cingil et al. (2000) az XML és RDF W3C szabványokat alkalmazza kollaboratív szűrőrendszerének modellezéséhez. Ezen egységesítési igény mára számos modellezési szabványhoz vezetett, nem csak az adatmodellezés, hanem gyakorlatilag az egész adat- és webbányászati folyamat terén.

6. Szabványok

Az adatbányászat terén mára számos szabvány jelent meg. Ezek nem egyforma célokat szolgálnak, némelyik csak az adatmodellezés terén ad útmutatást, mások a teljes adat- vagy webbányászati folyamatot képesek lefedni, egyesek elméleti szemszögből közelítik meg az adatbányászat kérdését, mások inkább fejlesztői szemmel tekintik a kérdést. Ez utóbbi szemlélet eredményeképpen jöttek létre komplett fejlesztői környezetek (API-k) az adatbányászat megvalósítására. A megjelent szabványok többsége XML alapokon nyugszik, és nagy részük nem képez teljesen önálló szabványt, hanem egy meglévő keretrendszer részeként jelent meg. A szabványosítás elvben megoldást jelenthet az adat- és webbányászat számos problémájára, a hatékony modellek és módszerek terjeszthetővé tétele által.

6.1. PMML

A PMML egy XML leíró nyelv statisztikai és adatbányászati modellek leírására, amit a Data Mining Group fejleszt. Egy PMML dokumentum alapvető felépítése a következő:

```
<?xml version="1.0"?>
<PMML version="3.2"
  xmlns="http://www.dmg.org/PMML-3_2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <Header copyright="Example.com"/>
  <DataDictionary> ... </DataDictionary>
  ... modell ...
</PMML>
```

kód forrása: dmg.org

ahol a `<DataDictionary>` elemben kerülnek definiálásra az adatmodell elemei, maga az adatbányászati modell pedig egy

```
<xs:element name="ExampleModel">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
```

```
        <xs:element ref="MiningSchema"/>
        <xs:element ref="Output" minOccurs="0"/>
        <xs:element ref="ModelStats" minOccurs="0"/>
        <xs:element ref="Targets" minOccurs="0"/>
        <xs:element ref="LocalTransformations" minOccurs="0" />
        ...
        <xs:element ref="ModelVerification" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="modelName" type="xs:string" use="optional"/>
      <xs:attribute name="functionName" type="MINING-FUNCTION"
use="required"/>
      <xs:attribute name="algorithmName" type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>
```

kód forrása: dmg.org

alakú modellelem által kerül definiálásra, amelyben hivatkozott elemek fejtik ki a modell részegységeit (MiningSchema, Output, stb.).

Elsődleges célja az adatbányászati modellek rendszerek közötti és akár implementációk forgalmazói közötti cseréjének lehetővé tétele. Támogatja az adatbányászati folyamat input-jának, a szükséges adat-átalakításoknak, és az adatbányászati modellt leíró paraméterek leírását (dmg.org).

A PMML a modellek reprezentációjára koncentrál, ezért algoritmus-központú megközelítést alkalmaz, mivel minden algoritmus jellemzően különböző adatszerkezeteket használ a modell-szint megtartásához. Az algoritmus implementációja irányítja a modell-reprezentációt, az átalakításokat és bizonyos algoritmus-specifikus beállításokat, ezért egy új algoritmus hozzáadása a PMML keretrendszerhez gyakran XML reprezentációs sémák egy csoportjának létrehozását vonja maga után. A PMML ma már adatbányászati modellek nagyon széles spektrumát támogatja.

A PMML céljai közé tartozik az is, hogy csökkentse a forgalmazók a modellekhez illesztett kiegészítéseinek szükségét, azáltal, hogy a nagy forgalmazókkal együttműködve próbál standardizált kiegészítéseket kidolgozni.

6.2. CWM/DM

A CWM/DM is XML-alapú szabvány, és a Common Warehouse Metadata (CWM) specifikációján alapul, így azok közé a szabványok közé tartozik, amelyek egy már létező, fejlett keretrendszerbe illeszkednek bele. Olyan metamodellt biztosít, mely képes az adatbányászathoz szükséges meta-adatok XML-ben történő ábrázolására. (omg.org). Elég komplex és átfogó adatmodellel rendelkezik.

6.3. XMLA

XML for Analysis, röviden XMLA nyílt szabvány a világhálóról származó adatforrásokhoz való hozzáférést támogatja. A szabvány legfontosabb implementációs közege a Microsoft SQL Server. A Microsoft SQL Server 2005 Analysis Services (SSAS) az XMLA-t használja kizárólagos protokollként a kliensalkalmazásokkal való kommunikációra.

Az XMLA szabvány alapvetően két XML-ben definiálható metódussal dolgozik, a Discover és az Execute metódusokkal.

A Discover metódus megszerzi az adott webes szolgáltatásból kinyerhető információkat és meta-adatokat. Ezen információ a rendelkezésre álló adatforrások mellett tartalmazhatja az adatforrások szolgáltatóinak adatait is. Tulajdonságmezőkkel megadható, hogy mely adatok és milyen formában kerüljenek kinyerésre az adatforrásból. A Discover általánosan arra használatos, hogy definiálja az adattípusokat, amikre egy kliensoldali alkalmazásnak szüksége lehet. A beállítható tulajdonságok és a generikus interfész extensibility-t nyújt annak szüksége nélkül, hogy a felhasználónak újra kelljen írnia a funkciókat egy kliensoldali alkalmazásban.

A Discover metódus szintaxisa a következő:

```
<Discover>
   <RequestType>...</RequestType>
   <Restrictions>...</Restrictions>
   <Properties>...</Properties>
</Discover>
```

kód forrása: msdn.microsoft.com

Ebben az alakban kéri le a futó Analysis Services példány meta-adatait, melyeket egy XMLA Rowset adattípusban ad vissza.

Az Execute metódus szolgáltató-specifikus parancsok lefuttatását teszi lehetővé az alkalmazásoknak az XML adatforrásokra. Az Execute metódus szintaxisa a következő:

```
<Execute>
   <Command>...</Command>
   <Properties>...</Properties>
   <Parameters>...</Parameters>
</Execute>
```

kód forrása: msdn.microsoft.com

Az XMLA protokoll a webes alkalmazásokra lett optimalizálva. Ezt az XML-alapú API-t a kliens és szerver között flexibilis technológia alkalmazását igénylő kliens-szerver alkalmazások, valamint több operációs rendszert célzó kliens-szerver alkalmazások fejlesztéséhez ajánlják. (msdn.microsoft.com)

A Microsoft SQL Server 2005 Analysis Services az XMLA-t használja az adatok definiálásához, manipulálásához és kezelésének támogatására is, az XMLA specifikációt kiegészítő plusz parancsokat is támogat.

6.4. SQL/MM DM

Az SQL specifikációhoz tartozó SQL Multimedia an Applicaton Packages részeként jelent meg az SQL/MM Part 6 Data Mining, röviden SQL/MM DM, ami SQL-en keresztül nyújt hozzáférést az adatbázisokban való adatbányászathoz. A szabvány fejlesztői felismerték az igényt egy standard SQL-en alapuló modellépítési, tesztelési és alkalmazási interfészre, míg a modellreprezentációs formátumot a már meglévő PMML-ből vették át. Az SQL/MM DM nem specifikálja a modellek reprezentációját, de felteszi, hogy a PMML és hasonló szabványok alkalmas jelöltek az adatbázisban tárolt adatok modellezésére.

A SQL/MM DM alapkoncepciója tehát, hogy maga nem készít konkrét modelleket, hanem a fejlesztőnek a már meglévő, pl. PMML-ben definiált modelljének betöltésére és használatára ad egy interfészt, amelyet a standard SQL-ben a szabvány által definiált felhasználói típusokkal határol be. Ezek a felhasználói típusok alapvetően két csoportra oszthatóak, a bányászati technikáktól független és a bányászati technikákhoz kapcsolódó csoportokra.

A független felhasználói típusok az adatbányászati adatok, az adatbányászati adatmodellezés és az adatbányászati alkalmazás felhasználói típusai. Ezek adják a konkrét technikákhoz kötődő típusok alapszintű infrastruktúráját.

- A `DM_MiningData` felhasználói típus a valós táblákban vagy nézetekben tárolt adatok absztrakciója, ami csak meta-adatokat tárol, és rajta keresztül elérhetőek a valós adatok.
- A `DM_MiningMapping` felhasználói típus az adatbányászati alkalmazások bemeneti mezőit definiálja, közben lehetőséget adva azokhoz tartozó járulékos információk definiálására. Fontos része az adatbányászati mezőtípus, ami meghatározza, hogyan kezelje az adatbányászati modul az adott mezőt.
- A `DM_ApplicationData` felhasználói típus egy konténer az adatbányászati modell alkalmazásához szükséges adatoknak. Alapvetően a bementi oszlopokhoz kapcsolt nevek halmaza.

A technikafüggő típusok használatosak az adatbányászati feldolgozásra. A főbb bányászati módszerek alapján a rendszer biztosít lehetőséget asszociációs szabályokon alapuló, klaszterezési, regressziós és osztályozási modellek alkalmazására. Ezek mindegyikéhez tartozik egy-egy típus a következő kategóriákban:

- bányászati feladat-típus, ami minden információt tartalmaz az adatbányászati modell felépítéséhez, pl. osztályozási modellhez `DM_ClasTask`.
- bányászati modell-típus, ami egy konkrét adatbányászati modell absztrakciójaként van definiálva. Metódusokat tartalmaz a modell tulajdonságainak eléréséhez, a modell alkalmazásához és teszteléséhez. Ennek értékei csak a hozzá tartozó feladatpéldány build metódusával vagy importálással generálhatóak. Ez az osztály pl. osztályozási feladatok esetében a `DM_ClasModel`, mely

```
CREATE TYPE DM_ClasModel
  AS (
    DM_content CHARACTER LARGE OBJECT(DM_MaxContentLength)
  )
  INSTANTIABLE
  NOT FINAL

STATIC METHOD DM_impClasModel
  (input CHARACTER LARGE OBJECT(DM_MaxContentLength))
  RETURNS DM_ClasModel
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL
  RETURNS NULL ON NULL INPUT,

METHOD DM_expClasModel()
  RETURNS CHARACTER LARGE OBJECT(DM_MaxContentLength)
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL,

METHOD DM_clasGetTask()
  RETURNS DM_ClasTask
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL,

METHOD DM_clasCostRate()
  RETURNS DOUBLE
  LANGUAGE SQL
  DETERMINISTIC
  CONTAINS SQL,
```

```
METHOD DM_clasModelMapping()
   RETURNS DM_MiningMapping
   LANGUAGE SQL
   DETERMINISTIC
   CONTAINS SQL,

METHOD DM_applyClasModel
   (input DM_ApplicationData)
   RETURNS DM_ClasResult
   LANGUAGE SQL
   DETERMINISTIC
   CONTAINS SQL
   RETURNS NULL ON NULL INPUT,

METHOD DM_testClasModel
   (input DM_MiningData)
   RETURNS DM_ClasTestResult
   LANGUAGE SQL
   DETERMINISTIC
   CONTAINS SQL
   RETURNS NULL ON NULL INPUT
```

kód forrása: SQL/MM DM ISO/IEC Committee Draft

alakú, ahol a definiált metódusok rendre az osztályozási modellt, az osztályozási modell `DM_CONTENT` értékét szimbolizáló `CHARACTER LARGE OBJECT` típusú objektumot, a modell adott adatokra való alkalmazásának eredményét, az adott adatokkal végzett teszt eredményét, a modell költségértékét, a modell létrehozásához felhasznált `DM_ClasTask` példányt, illetve a modell adott alkalmazásához felhasznált `DM_MiningMapping` példányt adják vissza.

- teszteredmény-típus, ami az adott tesztelés során keletkezett értékeket tárolja (pl. `DM_ClasTestResult`)
- alkalmazási eredmény-típus, ami a modell adott alkalmazásának eredményeit tárolja (pl. `DM_ClasResult`)

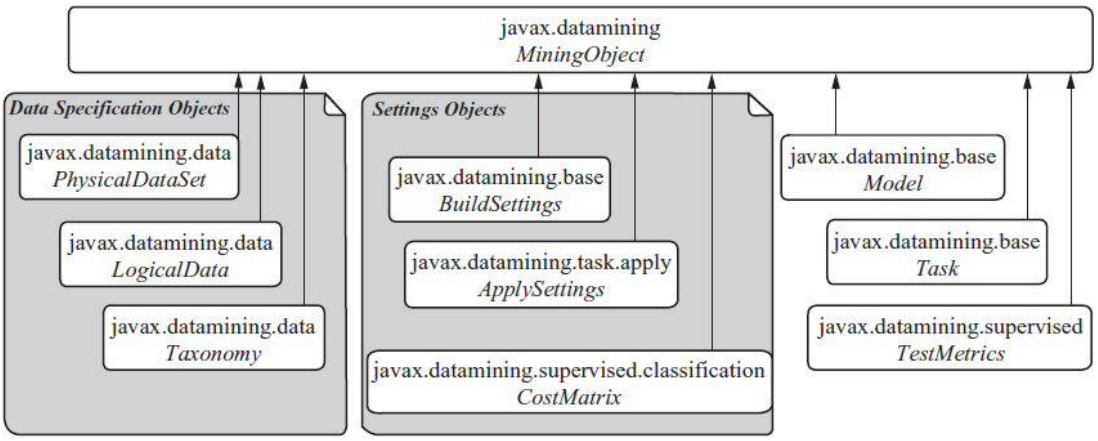
Ezen típusok segítségével az adatbányászati módszerek modelljeik importálása után elérhetővé válnak akár egyetlen SQL-lekérdezésen keresztül, pl. az alábbi utasítás létrehoz egy `DM_MiningData` példányt a `DM_defMiningData` metódussal és egy `DM_MiningMapping` példányt a `DM_MiningData` példány `DM_genMiningMap` metódusával, létrehoz egy `DM_ClasSettings` példányt annak alapértelmezett konstruktorával és hozzárendeli a létrehozott `DM_MiningMapping` példányt, majd az „r” oszlopot választja ki vizsgálatra a `DM_clasSetTarget` metódussal, ezután létrehozza egy `DM_ClasTask` értéket a `DM_defClasTask` metódussal, és végül a létrehozott `DM_ClasTask` példányt letárolja az MT táblában.

```
WITH MyData AS (
   DM_MiningData::DM_defMiningData('CT')
)
INSERT INTO MT (ID, TASK)
VALUES (
   1,
   DM_ClasTask::DM_defClasTask(
     MyData, NULL,
     (
     (new DM_ClasSettings())
       .DM_clasUseMapping(MyData.DM_genMiningMap())
     ).DM_clasSetTarget('r')
   )
)
```

kód forrása: SQL/MM DM ISO/IEC Committee Draft

6.5. JDM

A Java Data Mining (JDM) API a konkrét adatbányászati funkcionalitások hordozhatóságát elősegítő szabvány, mellyel olyan adatbányászati megoldások fejelszthetőek, melyek hordozhatóak több DME-re azaz adatbányászati motorra (Data Mining Engine). A JDM interfészei tisztán Java nyelven íródtak, amelyekhez a metódusok specifikálhatóak Java nyelven, de lehetőséget biztosítanak a forgalmazóknak bármilyen más implementációs technológia és programozási nyelv használatára a Java interfészek mögötti implementáció megvalósításához. Ez lehetőséget nyújt a forgalmazóknak termékeik becsomagolására a JDM interfésszel. Így a JDM nyitott, tisztán Javában íródott és több forgalmazós szabvány marad.



A JDM osztálydiagramja. Forrás: Hornick et al., 2007

Az ábrán látható a JDM osztálydiagramja, melynek csúcsán a *MiningObject* áll, amely a *javax.datamining* csomagban található. Ez a JDM legfontosabb osztálya, ennek példányai elmenthetőek egy ezek számára fenntartott tárolóba, a MOR-ba (Mining Object Repository), ahonnan az alkalmazások később név alapján kikereshetik azokat futásuk során. A JDM

felosztja a bányászati objektumokat adatspecifikáció, beállítás, modell, tesztmetrika, és feladat csoportokba.

Az adatspecifikációs objektumok használatosak az input adatok leírására, mivel fontos a DME hatékonysága érdekében, hogy ismerje az adatok tulajdonságait. A JDM külön definiál fizikai és logikai adatosztályokat, a *PhysicalDataSet* osztály a fizikai tulajdonságokat ragadja meg, mint az adat elérési útja, neve, míg a *LogicalDataSet* specifikálja az adatok értelmezési módját, pl. típusának megadásával, vagy validált értékeinek megadásával. Ez a szétválasztás lehetővé teszi egy *PhysicalDataSet*-hez több *LogicalDataSet* hozzárendelését és fordítva, ami nagyban elősegíti az adatok feladatközi újrahasznosítását.

A beállítási objektumok a modellek felépítése és alkalmazása során alkalmazott beállításokhoz használatosak, a *BuildSettings* a modellépítéshez meghatározható bányászati funkció vagy algoritmusszintű beállításokat, az *ApplySettings* az alkalmazás eredményeként előálló output kívánt tartalmát határozza meg.

A modellobjektumok tartalmazzák az adott modell felépítéséhez szükséges adatokban található tudást, funkció és algoritmus szintjén is részletezve, és a modell létrehozása során használt modellépítési beállításokat is. A *Model* ezen típusok szülőinterfésze, és ehhez tartozhatnak a *ModelDetail* szülőinterfészen keresztül algoritmus-specifikus implementációs részletek is, és ezen szülőinterfészek alinterfészei tartoznak minden különböző adatbányászati feladathoz (pl. *ClassificationModel*).

A *TestMetrics* osztály és alosztályai implementálják a bányászati funkciók tesztelése során használt metrikákat.

A *Task* osztály arra szolgál, hogy a JDM feladatait, a modellépítést, tesztelést, alkalmazást, exportálást és importálást modellezze. A *Task* objektum enkapszulálja az ahhoz szükséges input és output objektumok specifikációját, hogy a DME elvégezze az adott feladatot. Pl. a *BuildTask* objektum egy modell felépítéséhez használható, és ehhez az adatspecifikációkat, a modellépítési beállításokat, és a kimeneti modell nevét veszi fel argumentumként.

A JDM használatához első lépésként csatlakozni kell egy DME-hez, ami alapvetően a *javax.datamining.resource* csomag *ConnectionFactory, ConnectionSpec* és *Connection* interfészein keresztül történik. A *ConnectionFactory* osztály a legtöbb esetben forgalmazó-specifikus. A *ConnectionFactory* objektum beállítása után, és a DME kapcsolat

tulajdonságainak a *ConnectionSpec* objektumba való betöltése után a kapcsolat létrehozható valamely *getConnection* metódussal. Ennek menete a következő:

```
if(connFactory.supportsCapability(ConnectionCapability.connectionSpec)) {
  //üres ConnectionSpec a Factory-tól
  ConnectionSpec connSpec = connFactory.getConnectionSpec();
  //kapcsolat tulajdonságainak beállítása
  connSpec.setURI("DMELocation");
  connSpec.setUsername("user");
  connSpec.setPassword("passwd");
  //DME kapcsolat létrehozása
  Connection dmeConn = connFactory.getConnection(connSpec);
}
```

kód forrása: Hornick et al., 2007

Az itt használatba kerülő *Connection* interfész biztosít metódusokat *MiningObject*-ek létrehozására és kezelésére, bányászati feladatok elvégzésére, és a DME képességeinek felderítésére, tehát ezekkel a metódusokkal oldható meg a tulajdonképpeni adatbányászati feladatvégzés. Az itt látható procedúrák erre mutatnak példát, fennálló DME kapcsolat esetén az adott *MiningObject*-re az első felépíti annak modelljét a DME-ben, míg a második kinyeri onnan egy *ClassificationModel* objektumba.

```
//BuildTask létrehozása és elvégzése a modell felépítésére
public boolean run() throws JDMException {
BuildTask btk = btkFactory.create("build_data","build_settings","model");
Long timeOut=null;//Run until completion
ExecutionStatus status = dmeConn.execute( btk, timeOut );
if( ExecutionState.success.equals( status.getState() ) )
  return true;
else
  return false;
}

//Felépített modell kinyerése
public void output() throws JDMException {
ClassificationModel clsModel = (ClassificationModel)dmeConn.retrieveObject(
"model", NamedObject.model );
}
```

kód forrása: Hornick et al., 2007

A JDM különösen alkalmas lehet webbányászati alkalmazásokra, mivel a szabvány definiál egy webes szolgáltatás interfészt, a Java Data Mining Web Services-t (JDMWS), ami XML sémákon alapulva a JDM API-val konzisztens funkcionalitást képes biztosítani, minden abban megjelenő koncepció és objektummodell is alkalmazható benne.

7. Kitekintés – a szemantikus web bányászata

A megjelenő szabványok nagyban befolyásolják az adatbányászat területének előrehaladását, segítik a jó megoldások alkalmazások közötti megosztását, és leveszik a modellek és módszerek implementációjának terhét a nem erre koncentráló alkalmazásfejlesztőktől.

Emellett azonban a webbányászatot a web fejlődése is formálja, annak trendjei is kihatnak rá. A webbányászati kutatási terület és a szemantikus web fejlődése között egyre erősebb kapcsolat fedezhető fel. Egyre többen kutatják annak lehetőségét, hogy a web szemantikai strukturális tartalmának felhasználásával javítsák a webbányászat teljesítményét, a webbányászatot pedig a szemantikus web felépítésére használják egyre többen. (Stumme et al., 2006)

A szemantikus web (más néven web 3.0) a felhasználói folyamatokat automatizáltan feldolgozható szemantikai információkkal segíti, ennek hatására egyre jobban terjed a weboldalak szemantikájának és a navigációs viselkedések formalizációja.

Ezek a változások a webbányászat minden területét érintik. A webhasználat-bányászat szempontjából ez azt jelenti, hogy ha a meglátogatott weboldalak halmazának vagy sorozatának oldalaihoz hozzárendeljük a mögöttes szemantikai tartalmakat, azzal plusz információk fedezhetőek fel a felhasználói viselkedésről, amivel többek között a perszonalizációs funkcionalitás is jól fejleszthető. Amennyiben pedig nincs hozzáadott szemantikus tartalom, webhasználat-bányászati módszerekkel meghatározható egy struktúra a viselkedések alapján. Ez főleg igaz a perszonalizált ajánlórendszerekre.

A webhasználat-bányászat szempontjából különösen fontos, hogy a szemantikus weben már lehetőség nyílik a hiperlinkek explicit leírására, ami mélyebb tudás megszerzését teszi lehetővé a weboldal tartalmával kapcsolatban, és az oldal tartalma is formai szemantikát kap, ami különösen előnyös, ha az oldalnézeteket elemeik együtteseként tekintik a részletesebb használati elemzés céljából.

A hozzáadott szemantika hatalmas előnye, hogy megfelelő felépítése esetén kizárhatja a szerver elemzéséből az oldalnézetek heurisztikus felhasználói eseményekbe rendezését, mivel az egyes eseményekhez tartozó oldalnézetek, lekérések szemantika hozzáadásával is behatárolhatóak.

A szemantikus web fejlődése így növelheti a webhasználat-bányászati módszerek hatékonyságát, pontosságát, ami így használhatóbb, kényelmesebben használható weboldalakat eredményezhet.

Stumme et al. (2006) szerint ez a szemantikán alapuló fejlődés odáig juthat, hogy a webbányászat egyszerre, integrálva kezelheti majd a tartalmat, a struktúrát és a használati adatokat, amit a szemantika kinyerésének és hasznosításának iteratív ciklusaiban alkalmazhat, ezzel eljutva a webes információk sokkal jobb megértéséhez.

8. Összegzés

A felhasználókban felmerülő kényelmi kérdések és a weboldalak üzemeltetőinek üzleti érdekei együttesen alakították ki a web perszonalizáció igényét és megvalósítását. Az egyre inkább a szolgáltatások és az elektronikus kereskedelem felé eltolódó web piacáért küzdő weboldalak jobban meg akarják ismerni ügyfeleiket, és igényeikhez akarják igazítani szolgáltatásaikat, hogy növeljék azok hűségét és vásárlókedvét. A felhasználóknak pedig igénye van az intelligens szolgáltatásra, hogy ne vesszenek el a rajtuk túlnőtt adatrengetegben, és ne kelljen egy oldalra való belépéskor minden alkalommal „mindent előről kezdeniük", a rendszer „emlékezzen rájuk", beállításaikra és érdeklődési körükre.

Ezen célok elérését célozza a perszonalizáció, mely ma már korábbi problémás, hosszú űrlapok kitöltésével járó vagy akár az adatfeldolgozás fázisában is folyamatos emberi beavatkozást igénylő megoldásain túllépve a webhasználat-bányászat módszereit alkalmazva automatizálásra került.

A webhasználat-bányászat alapú perszonalizáció ma már igen sokféle és bonyolult funkcionalitásokat képes nyújtani. Ezekre jó példa az ismertebb weboldalak közül például az Amazon.com internetes boltja vagy a Facebook közösségi portál, melyek a felhasználó korábbi aktivitásai alapján tesznek ajánlásokat könyvekre és egyéb termékekre, vagy a felhasználó online ismerőseinek ismerőseire.

Ezeket a funkcionalitásokat, és a legfontosabbak, a leggyakoribbak mögött lévő módszertant, működési elveket, valamint a megvalósításuk során felmerülő problémákat helyeztem dolgozatom középpontjába.

A teljesítmény és a hordozhatóság növelése érdekében az adatbányászatot is elérte a szabványosítás, ez pedig annak minden alterületére, így a webhasználat-bányászatra és az

azon alapuló perszonalizációra is kihat. Komoly lehetőségek rejlenek a jövőben az általánosan adatbányászatra kifejlesztett szabványok mellett főleg a webre koncentráló megoldásokban, mint pl. a főleg szerver-kliens architektúrájú webes alkalmazásokra optimalizált XMLA, vagy a JDM részeként alkalmazható JDMWS.

A szemantikus web megjelenésével pedig a felhasználók viselkedésének és érdeklődésének megismerése tovább mélyíthető és finomítható, ha pedig az abból kinyert adatokat valóban felhasználják a szemantikus web továbbépítésére, adott lesz a környezet egyebek mellett a perszonalizáció szélesebb körű alkalmazására.

Ezek a tények előrevetítik, hogy a jövőben a jelenlegi domináns ajánlattevő rendszerekkel szemben akár lehetőség nyílhat sokkal bonyolultabb perszonalizációs funkcionalitások elterjedésére is.

Ez viszont felvet számos biztonsági kérdést is, ugyanis sok felhasználó már jelenleg is veszélyben érzi adatainak biztonságát, a testre szabhatóság és kényelem igényének fenntartása mellett sokan nem értik, miért kell róluk adatokat gyűjteni. Ezért fontos, hogy amennyiben a perszonalizációs funkcionalitások továbbfejlődnek és pontosabb, részletesebb adatokra lesz szükségük, biztosítani kell egyrészt az alkalmazott módszerek transzparenciáját a felhasználó számára, másrészt a felhasználók minden körülmények között megőrzött anonimitását.

Összességében véve viszont a biztonsági aggályoktól eltekintve a perszonalizáció a ma tapasztalható adatmennyiség mellett elengedhetetlenné vált a web hatékony használatához, és a szemantikus web alapú fejlesztésekre is szükség van, ugyanis a mai napig vannak bizonyos helyzetek, melyekben a szolgáltatások komoly pontatlanságokat mutatnak.

Irodalomjegyzék

Cingil, I.; Dogac, A.; Azgin, A.: A Broader Approach to Personalization, *Communications of the ACM*, 43(8), 136-141, 2000

Eztioni, O.: The world wide Web: Quagmine gold mine, *Communications of the ACM*, 39(11), 65-68, 1996

Hornick, Mark F.; Marcadé, Erik; Venkayala, Sunil: Java Data Mining: Strategy, Standard and Practice. Elsevir, 2007

Kobsa,A.; Koenemann,J.; Pohl,W.: Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. *The Knowledge Engineering Review*, 16(2), 111-155., 2001

Kohrs, A.; Merialdo, B.: Clustering for collaborative filtering applications. *Computational Intelligence for Modelling, Control and Automatication* (CIMCA'99), IOS Press, Vienna, 199-204, 1999

Kohrs, Arnd; Merialdo, Bernard: Creating user-adapted Websites by the use of collaborative filtering. *Interacting with Computers*, 13, 695-716. Elsevier Science B.V., 2001

Mobasher, B. ,Cooley, R., Srivastava, J.: Automatic personalization based on Web usage mining. Department of Computer Science, DePaul University, 1999

Mobasher, Bamshad: Web Usage Mining. In: Liu, Bing: Web Data Mining. Springer-Verlag Berlin Heidelberg, 2007

Mobasher,B.; Cooley,R.; Srivastava,J.: Automatic personalization based onWeb usage mining. *Communications of the ACM*, 43(8), 142-151, 2000

Pennock,D.; Horvitz,E.; Lawrence, S.; Lee Giles, C.: Collaborative Filtering by Personality Diagnosis: A Hybrid Memory and Model-Based Approach. *UAI-2000: The 16th Conference on Uncertainty in Artificial Intelligence*. Stanford University, Stanford, CA, 73-480, 2000

Pierrakos, Dimitrios; Paliouras, Georgios; Papatheodorou, Christos; Spyropoulos, Consantine D.: Web Usage Mining as aTool for Personalization: A Survey. User Modeling and User Interaction, 13, 311-372. Kluwer Academic Publishers, 2003

Schwarzkopf,E.: An adaptive web site for the UM2001 conference, In: Proceedings of the UM2001Workshop on Machine Learning for User Modeling, 77-86, 2001

Stumme, Gerd; Hotho, Andreas; Berendt, Bettina: Semantic Web Mining State of art and future directions. *Web Semantics: Science, Services and Agents ont he World Wide Web*, 4, 124-143. Elsevir, 2006

W. Lin; S. A. Alvarez; C. Ruiz: Efficient Adaptive-Support Association Rule Mining for Recommender Systems. *Data Mining and Knowledge Discovery*, 6, 83–105, 2002

Webes irodalomjegyzék

ISO/IEC JTC 1/SC 32 N 0606 - Information technology — Database languages — SQL Multimedia and Application Packages — Part 6: Data Mining: ISO/IEC Committee Draft, 2000

CWM – Common Warehouse Metamodel Specification: http://www.omg.org/docs/formal/03-03-02.pdf

PMML – Data Mining Group – PMML 3.2 specification: http://www.dmg.org/v3-2/GeneralStructure.html

XML for Analysis (XMLA) – msdn - http://msdn.microsoft.com/en-us/library/ms187190(SQL.90).aspx

Bodon Ferenc, Dr.: Adatbányászati algoritmusok, http://www.cs.bme.hu/~bodon/magyar/adatbanyaszat, 2008

OLAP – Sidló Csaba István – ClickS fejlesztési dokumentáció - http://scs.web.elte.hu/Work/ClickS/doc/Fejleszti dokumentci.doc, 2003