

hyväksymispäivä

arvosana

arvostelija

Suosittelijajärjestelmät

Sami Ojanpää

Helsinki 7. maaliskuuta 2002

Älykkäiden järjestelmien tutkimusseminaari

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Sisältö

1 Johdanto	1
2 Menetelmät	2
2.1 Sisältösuodatusjärjestelmät	3
2.2 Yhteisösuodatusjärjestelmät	4
2.2.1 Muistiperustaiset järjestelmät	5
2.2.2 Malliperustaiset järjestelmät	7
3 Sovellusesimerkkejä	9
4 Lopuksi	9
Lähteet	10

1 Johdanto

Internetissä on valtavasti informaatiota, minkä takia tietyn tiedon löytäminen on hankalaa. Käyttäjää vastaan tulee paljon turhaa informaatiota, mutta oleellinen tieto odottaa löytäjäänsä. Entä jos joku toinen onkin jo löytänyt kaipaamallasi asialle hyvän tietolähteen? Jos tunnet kyseisen henkilön, hän voi suositella sinulle kyseistä lähdettä. Näin onnellisesti käy kuitenkin vain harvoin. Voisiko hyväksi koettuja paikkoja suositella jotenkin automaattisesti?

Verkkokaupat haluavat tarjota asiakkailleen lähikaupan tuntua: asiakkaat tuntevat kauppiaan ja kauppias asiakkaat ja heidän tarpeensa. Tähän kaupat pyrkivät tarjoamalla asiakkailleen personoituja kullekin asiakkaalle yksilöllisiä palveluja. Jos asiakas kokee palvelun tarpeitaan tyydyttäväksi, hän käyttää palvelua useammin. Tämä lisää käyttäjäuskollisuutta ja kauppiaan tuloja. Yksilöllisten palvelujen antaminen edellyttää kuitenkin paljon tietoa asiakkaista ja heidän mieltymyksistään. Sähköistynyt kaupankäynti (verkkokaupat, kanta-asiakaskortit) mahdollistaa suuren informaatiomäärän keräämisen suhteellisen nopeasti. Suuren informaatiomäärän käsitteleminen ei kuitenkaan suju ihmisiltä helposti, vaan käsitteilyyn tarvitaan automaattisia menetelmiä.

Edellisille tilanteille yhteisenä tekijänä on tavoite tarjota käyttäjälle se mitä hän tarvitsee (tai luulee tarvitsevänsä). Tämän tavoitteen toteuttamiseksi on kehitetty *suosittelijajärjestelmiä* (recommender systems), jotka suosittelevat käyttäjälle palveluja ¹, joista hän saattaisi olla kiinnostunut. Suosittelijajärjestelmien toiminta voi perustua esimerkiksi käsin määriteltyihin produktiosääntöihin (ks. esim. www.broadvision.com). Tämä ei kuitenkaan ole tehokasta. Aktiivisen tutkimuksen kohteena ovatkin järjestelmät, jotka osaavat automaattisesti tehdä suosituseh-

¹Yksinkertaisuuden vuoksi puhutaan palveluista, vaikka tarkoitettaisiin myös dokumentteja ja tuotteita.

dotuksia asiakas- ja tuotetietojen sekä asiakkaiden (oletettujen) mieltymysten perusteella.

2 Menetelmät

Suosittelijajärjestelmät hyödyntävät käyttäjäprofiileja, palvelutietoja ja käyttäjien palveluille antamia arvosanoja määrittäessään mitä palveluita kullekin käyttäjälle kannattaa suositella.

Tietoja voidaan kerätä eksplisiittisesti tai implisiittisesti. *Eksplisiittisessä tietojen keräämisessä* käyttäjäprofiilit muodostetaan esimerkiksi rekisteröintilomakkeen pohjalta. Myös palveluiden arvosanojen antaminen on aktiivista toimintaa. Käyttäjä antaa arvosanan jollakin skaalalla esim. yhdestä (huono) viiteen (hyvä).

Implisiittisessä tavassa tulkitaan käyttäjän käyttäytymistä ja hänen tekemiään valintoja. Esimerkiksi usein vierailtu www-sivu tai paljon ostettu tuote saa korkeat arvosanat [BHK98]. Tässä yhteydessä arvosanojen sijata puhutaan usein äänistä. Usein vierailtu sivu saa paljon kävijöitä l. ääniä, ja siten se arvostetaan korkealle.

Implisiittisessä tietojen keräämisessä on aina virhetulkinnan mahdollisuus. Käyttäjän ostaessa tuotteen oletetaan, hän pitää tuotteesta tai tarvitsee sitä. Mutta tämä oletus voi olla virheellinen, koska tuote saattaakin esimerkiksi olla lahja jollekin toiselle henkilölle tai muutoin toisen henkilön pyynnöstä ostettu. Eksplisiittinen tietojen kerääminen vaatii käyttäjältä aikaa ja normaaliin toimintaan nähden ylimääräistä ponnistelua. Tämä saattaa olla käyttäjästä vastenmielistä, eikä hän enää palaa takaisin. Toisaalta käyttäjä voi sitoutua palveluun paremmin, kun hän huomaa, että palvelua yritetään parantaa hänelle sopivaksi. Eksplisiittisesti saadut tiedot ovat ehkä tarkempia kuin implisiittisesti saadut, mutta tämä edellyttää, että käyttäjä vastaa totuudenmukaisesti, ja että hän toimii samoin kuin

vastaa. Annettujen tietojen varmentaminen voi siis olla tarpeen.

Tarkimmat käyttäjien profiilit ja suosituimmat palvelut saadaan parhaiten selvälle yhdistämällä eksplisiittinen ja implisiittinen tiedonkeräys. Yhdistämisen avulla voidaan tarkistaa eri menetelmillä saatuja tietoja. Näin käyttäjän profiilista saadaan luotettavampi ja totuudenmukaisempi.

Onnistunut käyttäjäprofiilitietojen määrittäminen ja tietojenkeräys on oleellinen edellytys suosittelijajärjestelmän virheettömälle toiminnalle [Buo01]. Järjestelmiä suunniteltaessa on myös otettava huomioon arvosanojen epätasainen jakautuminen palveluiden kesken [BHK98]. Osa palveluista saa paljon ääniä, osa ei yhtään. On myös huomattava, että absoluuttiset arvostukset eivät välttämättä ole tärkeitä, vaan niiden suhde antajaan tulisi ottaa huomioon [BHC98]. Eri käyttäjien henkilökohtaiset asteikot voivat olla erilaiset.

Suosittelijajärjestelmät voidaan jakaa *yhteisösuodatusjärjestelmiin* (collaborative filtering systems) ja *sisältösuodatusjärjestelmiin* (content-based filtering systems). Toisin on myös olemassa järjestelmiä, jotka ovat näiden hybridejä eikä raja muutenkaan aina ole ihan selvä.

2.1 Sisältösuodatusjärjestelmät

Sisältösuodatussuosittelija analysoi palveluiden sisältöä/ominaisuuksia tehdessään päätöstä suosittelemisesta. Menetelmä sopii tilanteisiin, jossa palveluita voidaan helposti analysoida tietokoneella (selkeät attribuutit) ja käyttäjän päätös tuotteen sopivuudesta ei ole kovinkaan subjektiivinen [Buo01]. Koska suositeltavan palvelun valinta perustuu palvelun ominaisuuksiin, käyttäjälle voidaan tarjota aivan uuttakin vaihtoehtoa eli sellaista, jota kukaan ei ole vielä arvioinut [Pop01]. Tiivistetysti voi sanoa, että sisältösuodatusjärjestelmä on parhaimmillaan suositellessaan vähän tunnettuja palveluja käyttäjille, joilla on yksilölli-

siä mieltymyksiä tilanteessa, jossa käyttäjätiedot ovat heikot, mutta palveluista on hyvät tiedot [MoRK00].

Sisältöpohjaistasuosittelijaa voidaan käyttää esimerkiksi artikkelitietokannassa. Käyttäjän luettua artikkelin hänelle voidaan tarjota listaa muista samankaltaisista artikkeleista. Tällaisia dokumentteja voidaan löytää esim. siten, että kutakin dokumenttia käsitellään vektorina sanojen frekvensseistä. Laskemalla kahden näin saadun vektorin välisen kulman kosini saadaan mitta dokumenttien samankaltaisuudelle [BHK98].

Toisaalta tavallinen hakukonekin voidaan ajatella sisältöpohjaiseksi suositelijaksi. Hakeehan se joitain avainsanoja vastaavia dokumentteja. Tämä ei tosin usein vielä anna hyviä tuloksia, mutta liittämällä kuhunkin dokumenttiin arvosana päästään parempiin tuloksiin. Esimerkiksi Google:ssa käytetään implisiittistä äänestysmentelmää: linkki sivulta A sivulle B tulkitaan A:n ääneksi B:lle. Mitä enemmän sivulle on linkkejä, sitä paremman arvosanan se saa.

Verkkokaupassa, missä käyttäjät usein pääsevät antamaan arvosanoja tuotteille, sisältösuositelija voisi määrittää, asiakkaan tarkastellessa jotain tuotetta, kaikille saman tuotekategorian sisällä oleville muille tuotteille ennustearvosanat ja suositella käyttäjälle parhaan arvosanan saanutta tuotetta. Huomattavaa on, että suositelija suosittelee vain sellaisia tuotteita, jotka ovat samankaltaisia käyttäjän muuten tarkastelemiin tuotteisiin.

2.2 Yhteisösuodatusjärjestelmät

Yhteisösuodatuksessa käyttäjäprofiilit ja käyttäjien antamat arvosanat nousevat keskeisiksi. Nyt ei siis etsitäkään pidetyn tuotteen kanssa samankaltaisia muita tuotteita, vaan etsitään tuotteita, joista muut samankaltaiset käyttäjät pitävät. Käytännössä yritetään ennustaa tietyn aktiivisena olevan käyttäjän antamaa ar-

vosanaa jollekin palvelulle, kun tiedetään minkälainen hän on suhteessa muihin käyttäjiin [BHK98].

Yhteisösuodatusmenetelmien hyvä puoli on se, että niiden toiminta ei riipu palveluista tai niiden ominaisuuksista [HKR00, MoRK00]. Kuitenkin menetelmä edellyttää, että käyttäjistä tiedetään kohtuullisen paljon [MoRK00, Buo01, BHC98]. Lisäksi käyttäjäprofiilien on oltava osittain päällekkäisiä [BHC98]. Käyttäjien mieltymysten tulisi myös olla samansuuntaisia [MoRK00].

Palveluita, joita ei ole arvosteltu, ei pystytä suosittelemaan [MoRK00]. Tämän takia uuden palvelun ilmaantuessa menee jonkin aikaa ennenkuin järjestelmä rupeaa suosittelemaan sitä [BHC98]. Käyttäjäprofiileissa olevat aukot ja arvosanojen epätasainen jakautuminen palveluille aiheuttaa sen, että vaikka järjestelmien antamat suositukset ovat usein oikein, ne menevät toisinaan myös täysin pieleen [HKR00].

Käyttäjiä voidaan käsitellä yksittäisinä, mutta erityisesti, jos käyttäjiä on paljon, voi olla tarpeen muodostaa käyttäjäryhmiä. Tällöin kukin käyttäjä määrittellään johonkin ryhmään kuuluvaksi ja hänelle tarjotaan hänen ryhmäprofiilinsa mukaisia tuotteita.

Yhteisösuodatusjärjestelmät voidaan jakaa *muistiperustaisiin* (memory-based) ja *malliperustaisiin* (model-based) järjestelmiin.

2.2.1 Muistiperustaiset järjestelmät

Muistiperustaiset järjestelmät käyttävät koko käyttäjätietokantaa tehdessään suosituksia. [BHK98]

Merkitään käyttäjän i palvelulle j antamaa arvosanaa v_{ij} :llä. Jos merkitään I_i :llä palveluja, jotka käyttäjä i on arvostellut, niin käyttäjän i arvosanojen keskiarvo

saadaan lausekkeesta

$$\bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{ij}.$$

Aktiivisen käyttäjän a arvosanaa ennustettaessa käytetään tietoja käyttäjästä ja tietokannasta laskettuja painoarvoja. Ennuste aktiivisen käyttäjän palvelulle j antamalle arvosanalle on

$$p_{aj} = \bar{v}_a + \kappa \sum_{i=1}^n w(a, i)(v_{ij} - \bar{v}_i),$$

missä n on sellaisten käyttäjien lukumäärä, joiden painoarvot ovat (aidosti) positiivisia. Termi κ on normalisoiija, jonka avulla huolehditaan siitä, että painoarvot $w(a, i)$ summautuvat 1:ksi.

Painoarvot kuvaavat käyttäjien samankaltaisuutta. Eräs tapa määrittää painoarvot on tarkastella käyttäjien antamia arvosanoja [BHK98]. Tällöin käyttäjien a ja i välinen korrelaatio on

$$w(a, i) = \frac{\sum_j (v_{aj} - \bar{v}_a)(v_{ij} - \bar{v}_i)}{\sum_j (v_{aj} - \bar{v}_a)^2 \sum_j (v_{ij} - \bar{v}_i)^2},$$

missä summat lasketaan niiden palveluiden yli, jotka sekä käyttäjä a että käyttäjä i ovat arvostelleet.

Painoarvot voidaan laskea myös menetelmällä, joka perustuu dokumenttien samankaltaisuuden laskemiseen [BHK98]. Nyt kukin dokumentti ajatellaan käyttäjäksi, palvelut sanoiksi ja arvosanat frekvensseiksi. Painoarvot saadaan lausekkeesta

$$w(a, i) = \sum_j \frac{v_{aj}}{\sqrt{\sum_{k \in I_a} v_{ak}^2}} \frac{v_{ij}}{\sqrt{\sum_{k \in I_i} v_{ik}^2}}.$$

Muistiperustaisten järjestelmien ongelmana on laskennan hidastuminen käyttäjien lisääntyessä. Tällöin myös todennäköisyys yhteisille palveluille käyttäjien A ja B välillä laskee.

2.2.2 Malliperustaiset järjestelmät

Malliperustaisten järjestelmien ideana on, että käyttäjätietokannan perusteella rakennetaan malli, jota käytetään tekemään ennustuksia [BHK98]. Malli opitaan jonkin opetusaineiston pohjalta ja sitä voidaan korjata tarvittaessa. Malli voi olla mikä tahansa esimerkiksi produktiosääntökokoelma [BHC98] tai neuraaliverkko. Useimmiten kuitenkin on käytetty todennäköisyysperustaisia malleja.

Todennäköisyysnäkökulmasta tarkasteltuna me yritämme laskea odotusarvon käyttäjän antamalle arvosanalle, kun me tiedämme jotain käyttäjästä. Tämä voidaan formuloida seuraavasti

$$p_{aj} = E(v_{aj}) = \sum_{i=0}^m P(v_{aj} = i | v_{ak}, k \in I_a) i,$$

missä arvosanat saavat kokonaislukuarvoja 0:sta m:ään [BHK98].

Todennäköisyyksistä on helppo siirtyä Bayes-verkkoihin. Palveluja voidaan ajatella muuttujina, jotka kuvataan Bayes-verkon solmuina. Muuttujien arvot ovat kyseisen palvelun saamat arvosanat. Opitussa Bayes-verkossa solmun vanhempien voidaan ajatella olevan parhaita ennustajia solmun arvolle. Solmujen ehdolliset todennäköisyydet voidaan esittää päätöspuuna, jonka avulla suosittelijajärjestelmä tekee päätöksensä [BHK98].

Suositteluongelma voidaan myös nähdä klusterointiongelmana, jolloin käytettävissä on lukuisia ratkaisuvaihtoehtoja. Yksi vaihtoehto on olettaa kunkin arvosan todennäköisyys ehdollisesti riippumattoksi jonkin latentin luokkamuuttujan C suhteen:

$$P(C = c, v_1, \dots, v_n) = P(C = c) \prod_{i=1}^n P(v_i | C = c).$$

Luokkamuuttujan arvojen määrittäminen ja mallin oppiminen palautuu äärellisten sekoitemallien (finite mixture models) [TSM85] oppimisongelmaksi, joihin on useita ratkaisuja. Toisaalta, jos käyttäjistä on tehty ryhmäprofiileja, voidaan

ongelmaa tarkastella naiivi-Bayes -luokitteluongelmana, jossa luokkamuuttujan C arvot vastaavat näitä ryhmäprofiileja.

Sekoitemallien perheeseen kuuluva *aspektimalli* (aspect model) [HoP99, Pop01] on vielä yksi tapa ratkaista yhteisösuositusongelma. Siinä jokaiseen pariin (i, j) liitetään latenti luokkamuuttuja z . Oletuksena on, että käyttäjä i ja palvelu j ovat riippumattomia annettuna z . Formaalisti siis

$$P(i, j) = \sum_{z \in Z} P(z) P(i|z) P(j|z).$$

Mallia voidaan opettaa *odotusarvonmaksimointialgoritmillä* (expectation maximization, EM). EM-algoritmissa on kaksi askelta: ensimmäisessä askeleessa (E) lasketaan puuttuvan arvon odotusarvo ja toisessa askeleessa (M) päivitetään mallin parametreja, siten että parametrien posterioritodennäköisyys maksimoituisi. Aspektimallin tapauksessa E-askel on

$$P(z|i, j) = \frac{P(z)P(i|z)P(j|z)}{\sum_{z' \in Z} P(z')P(i|z')P(j|z')}.$$

M-askeleessa tehdään kaksi päivitystä:

$$P(i|z) = \frac{\sum_j n(i, j) P(z|i, j)}{\sum_{j, i'} P(z|j, i')}$$

ja

$$P(j|z) = \frac{\sum_i n(i, j) P(z|i, j)}{\sum_{j', i} P(z|j', i)},$$

missä $n(i, j)$ on parien (i, j) lukumäärä. Kun malli on opittu, käyttäjälle a suositeltavat palvelut j voidaan laittaa järjestykseen todennäköisyyden $P(a, j)$ perusteella. Korkeimman todennäköisyyden saanut pari on paras suositus.

Edellä kuvattu aspektimalli ei ota mitään kantaa siihen millaisia palvelut ovat. Palveluiden sisältöön kantaa ottavaa aspektimallia on tarkasteltu ResearchIndex-artikkelitietokannan kontekstissa Popescu et al. artikkelissa [Pop01]. Edellä kuvatussa mallissa ei myöskään oteta huomioon käyttäjien palveluille antamia ar-

vosanoja. Tämä voidaan tehdä ottamalla malliin mukaan muuttuja, jonka arvot vastaavat palveluiden mahdollisia arvosanoja [HoP99].

3 Sovellusesimerkkejä

Enemmän tai vähemmän kaupallisia sovelluksia on tehty pilvin pimein ja lisää tulee koko ajan. Tässä viitteitä joihinkin sovellusesimerkkeihin.

Minnesotan yliopiston tutkimusprojektit GroupLens [MRK97, Res94] ja MovieLens (www.movielens.com) ovat tunnettuja. Näistä projekteista on kehittynyt kaupallinen sovellus NetPerceptions (www.netperceptions.com).

Alexa (www.alexa.com) on ilmainen Microsoftin Internet Exploreriin asennettava selainplugin, joka suosittelee linkkejä muiden käyttäjien tapojen pohjalta.

Suosittelijajärjestelmiä käytetään myös useissa verkkokaupoissa. Tällaisia ovat esimerkiksi Amazon.com, CDNow.com ja Levis.com.

Monissa ECommerce-palvelimissa on mukana personointipalveluja mahdollistavia työkaluja. Tällaisia ovat esimerkiksi Microsoftin SiteServer (www.microsoft.com/siteserver/) ja Bean WebLogic Portal (www.bea.com/products/weblogic/portal/).

4 Lopuksi

Suositteluongelma voidaan palauttaa moniin muihin ongelmiin, joten suosittelijajärjestelmiä voidaan toteuttaa monin eri tavoin. Kaikilla toteutustavoilla on omat hyvät ja huonot puolensa. Valittava menetelmä riippuu aina sovelluskohteesta.

Suosittelijajärjestelmät ovat tärkeä tutkimuskohde, koska informaation määrä li-

sääntyy koko ajan. Tarpeellisen tiedon löytämistä on helpotettava - ihmisillä ei ole aikaa tarpoa turhan tiedon suossa.

Suosittelijajärjestelmiin liittyy myös uhkakuvia. Voiko käyttäjä olla varma, että hän saa suosituksia puolueettomasti? Suositteleeko verkkokauppa vain tuotteita, joissa on suuri kate? On myös esitetty, että suosittelijajärjestelmän tulisi osata tarvittaessa perustella päätöksiään käyttäjälle [HKR00].

Sähköinen kaupankäynti pitää personoituja palveluja ja tehokkaita suosittelijajärjestelmiä edellytyksenä tuottoisaan kaupankäyntiin. Ensin olisi kuitenkin saatava perusta kuntoon. Heikko käytettävyys on yksi suurimpia syitä verkkokauppojen huonolle menestykselle [Nie01].

Lähteet

- BHC98 Basu C., Hirsh H., Cohen W., Recommendation as classification: Using social and content-based information in recommendation. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, July 1998, 714–720.
- BHK98 Breese J., Heckerman D., Kadie C., Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, July 1998, 43–52.
- Buo01 Buono P. et al., Integrating user data and collaborative filtering in a web recommendation system. *UM2001 Proc. Third Workshop on Adaptive Hypertext and Hypermedia*, Sonthofen, Germany, July 2001.
- HKR00 Herlocker J., Konstan J., Riedl J., Explaining collaborative filtering recommendations. *Proc. of CSCW*, 2000.

- HoP99 Hofmann T., Puzicha J., Latent class models for collaborative filtering. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1999, 688–693.
- MRK97 Miller B., Riedl J., Konstan J., Experiences with GroupLens: Making UseNet useful again. *Proceedings of the 1997 Usenix Winter Technical Conference*, Jan 1997.
- MoRK00 Mooney R., Roy L., Content-based book recommendations using learning for text categorization. *Proceedings of the Fifth ACM Conference on Digital Libraries*, 2000, 195–204.
- Nie01 Nielsen J., Did poor usability kill e-commerce?. *Jakob Nielsen's Alertbox*, Aug. 2001 *E-commerce and usability*, August 19, 2001, www.useit.com.
- Pop01 Popescui A. et al., Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-2001)*, 2001, 437–444.
- Res94 Resnick P. et al., GroupLens: An open architecture for collaborative filtering of Netnews. *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, 1994, 175–186.
- TSM85 Titterington D., Smith A., Makov U., *Statistical analysis of finite mixture distributions*. John Wiley & Sons, 1985.