

# **Katsaus suosittelujärjestelmiin - sisältöpohjaiset ja yhteistoiminnallisen suodattamisen järjestelmät**

Johanna Wahtera

Kandidaatintutkielma  
Helsingin yliopisto  
Tietojenkäsittelytieteen laitos

Helsinki, 13. joulukuuta 2015

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Johanna Wahtera			
Työn nimi — Arbetets titel — Title			
Katsaus suosittelujärjestelmiin - sisältöpohjaiset ja yhteistoiminnallisen suodattamisen järjestelmät			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Kandidaatintutkielma		13. joulukuuta 2015	21
Tiivistelmä — Referat — Abstract			
<p>Tutkielmassa tutustutaan sekä sisältöpohjaisten että yhteistoiminnalliseen suodattamiseen perustuvien suosittelujärjestelmien toimintaan yleisellä tasolla. Tutkielmassa selviää, mikä on näiden kahden suosittelujärjestelmätyypin oleellisin ero. Kumpaankin järjestelmätyyppiin syvennyttään esimerkkien kautta. Sisältöpohjaisen suosittelun kohdalla esitellään Flickr-kuvapalvelun tunnisteiden suosittelua käyttäjille. Yhteistoiminnallista suosittelua avataan esittelemällä Netflix Prize -kilpailuun osallistuneen joukkueen elokuvien suosittelujärjestelmää. Lisäksi tutustumme Google Newsin uutisartikkeleiden suosittelujärjestelmään, joka yhdistää kaksi yhteistoiminnallisen suodattamisen tekniikkaa yhdeksi malliksi. Esimerkkien kautta selviää myös, minkätyyppisissä palveluissa suosittelujärjestelmiä yleisesti käytetään.</p>			
<p>ACM Computing Classification System (CCS):</p> <p><b>Information systems—Recommender systems</b></p>			
Avainsanat — Nyckelord — Keywords			
suosittelujärjestelmät, yhteistoiminnallinen suodatus, sisältöpohjaisuus, Flickr, Netflix, Google News			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — Övriga uppgifter — Additional information			

# Sisältö

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Sisältöpohjaiset järjestelmät</b>	<b>2</b>
2.1	Yleisesti . . . . .	2
2.2	Tunnisteiden suosittelu käyttäjille yhteisötiedon perusteella . .	3
<b>3</b>	<b>Yhteistoiminnallisen suodattamisen järjestelmät</b>	<b>9</b>
3.1	Yleisesti . . . . .	9
3.2	Netflix Prize -kilpailu . . . . .	10
3.3	Uutisartikkelien suosittelu käyttäjille - skaalautuva yhteisoi- minnallinen suodattaminen . . . . .	14
<b>4</b>	<b>Pohdinta</b>	<b>18</b>
<b>5</b>	<b>Yhteenveto</b>	<b>19</b>
	<b>Lähteet</b>	<b>20</b>

# 1 Johdanto

Internetissä on valtava määrä monimuotoista sisältöä. Verkon käyttäjällä ei ole puutetta selattavista tuotteista, artikkeleista, keskustelupalstoista tai muusta internetin tarjoamasta sisällöstä. Rajattomien mahdollisuuksien edessä käyttäjän voi kuitenkin olla vaikeaa löytää jotakin juuri häntä kiinnostavaa. Suuren verkkokaupan laajaa valikoimaa ei voida asettaa kerralla käyttäjän näkyville samalla tavoin kuin perinteisessä kivijalkakaupassa.

Tämä tarjonnan esittelemisen haaste tulee vastaan kaikenlaisia tuotteita, kuten musiikkia, elokuvia tai artikkeleita selatessa. Hakukoneet ratkaisevat ongelman osittain löytämällä suuresta määrästä tietoa juuri sen, mitä käyttäjä etsii. On kuitenkin tapauksia, joissa käyttäjä ei *tiedä* mitä hän haluaa löytää. Tässä kuvaan astuu henkilökohtaisia kulutussuosituksia tarjoavat järjestelmät eli suosittelujärjestelmät.

Suosittelujärjestelmä valikoi käyttäjän puolesta tuotteita, joita tarjotaan hänelle kulutettavaksi. Järjestelmä kerää dataa käyttäjistä ja/tai tarjolla olevista tuotteista ja muodostaa datan perusteella suosituksia tuotteista, joita käyttäjä todennäköisesti pitäisi kiinnostavana.

Suosittelujärjestelmän toteutustavat voidaan jakaa kahteen laajempaan ryhmään: sisältöpohjaisiin (content-based) ja yhteistoiminnallisen suodattamisen (collaborative filtering) järjestelmiin [2][8]. Tässä aineessa keskitymme näiden järjestelmien eroihin ja tutustumme muutamaa esimerkkijärjestelmään.

Sisältöpohjaisissa järjestelmissä kerätään tietoa palvelun tuotteista ja vertaillaan näitä toisiinsa. Käyttäjälle suositellaan uusia tuotteita sen perusteella, minkälaisia tuotteita hän on menneisyydessä selannut tai hankkinut. Tuotteen merkittävät piirteet kartoitetaan tuoteprofiiliin ja profiilia verrataan toisen tuotteen profiiliin. Tavoitteena on löytää mahdollisimman samankaltaisia profileja. Tuote voi olla tässä yhteydessä mitä vain käyttäjälle suositeltavaa sisältöä, kuten elokuva tai uutisartikkeli. Elokuvan tapauksessa tuoteprofiiliin merkittäisiin esimerkiksi elokuvan näyttelijät, ohjaaja, valmistusvuosi ja lajityyppi.

Yhteistoiminnallisen suodattamisen järjestelmät keskittyvät yksittäisen

käyttäjän ja tuotteiden ominaisuuksien lisäksi käyttäjäyhteisön välisiin re-laatioihin. Käyttäjät lajitellaan samankaltaisiksi heidän selaushistoriansa tai antamiensa arvosteluiden perusteella. Samanlaisista asioista pitävät käyttäjät muodostavat oman aliryhmänsä koko käyttäjäyhteisöstä. Tuotteiden suosittelavuus määritellään vertailemalla useamman samankaltaisen käyttäjän tuote-arvosteluja tai selaustietoja. Jos yksi käyttäjä pitää tietystä tuotteesta, voidaan sitä suositella muillekin samankaltaisille käyttäjille.

Sekä yhteistoiminnalliseen suodattamiseen perustuvat että sisältöpohjaiset suosittelujärjestelmät vaativat toimiakseen tietoa käyttäjän mieltymyksistä. Käyttäjien mielipiteitä kerätään usein suoraan arvosteluiden kautta, mutta toisinaan preferenssit päätellään käyttäjän käyttäytymisestä muin keinoin, kuten esimerkiksi klikkaushistorian perusteella [5].

## 2 Sisältöpohjaiset järjestelmät

### 2.1 Yleisesti

Sisältöpohjaisissa suosittelujärjestelmissä vertaillaan käyttäjälle tarjottavan sisällön ominaisuuksia toisiinsa ja pyritään löytämään niiden väliset samankaltaisuudet. Jokaiselle tuotteelle muodostetaan tuoteprofili, johon kerätään tuotteen tärkeimmät ominaisuudet. Nämä tiedot ovat yleensä saatavilla suoraan tekstinä tuotteen tiedoista [8].

Profilien samankaltaisuus määräytyy pitkälti sen perusteella, kuinka paljon samoja luokituksia ja oleellisia sanoja niissä esiintyy. Eräs tähän soveltuvista menetelmistä on Jaccardin kerroin (Jaccard coefficient tai Jaccard index) johon palataan seuraavassa kappaleessa.

Tuoteprofilin lisäksi myös käyttäjästä rakennetaan käyttäjäprofili. Käyttäjäprofili rakentuu samoista osista kuin tuoteprofili, mutta tuotetiedon paikalle merkitään käyttäjän mieltymys kyseisen tiedon suhteen. Mieltymyksellä tarkoitetaan tässä käyttäjän historiasta opittua tietoa siitä, minkätyyppistä sisältöä käyttäjä vaikuttaa kuluttavan paljon. Esimerkki käyttäjäprofilin tiedosta voisi olla vaikkapa elokuvan lajityyppi, jolloin paljon toimintasisältöisiä elokuvia katsovan käyttäjän profiliin merkittäisiin lajityypin kohdalle

*toiminta*. Kaikissa järjestelmissä käyttäjästä ei luoda konkreettista profiilia, vaan raaka selaushistoria toimii riittävänä kuvauksena käyttäjästä.

Suosittelu on helppo perustaa tuotteen ominaisuuksille, jos tuotteen sisältö on helposti selvitettävissä, kuten esimerkiksi tekstimuodossa. On kuitenkin tapauksia, joissa tuotteen sisältöä on hankala määritellä ja vertailla toisiin tuotteisiin. Esimerkiksi kuvista on mahdollista saada vain rajallinen määrä suositusjärjestelmän kannalta oleellista tietoa. Yksi mahdollinen ratkaisu on lisätä tuotteisiin tietoa niiden sisällöstä sanallisesti. Esimerkki tällaisesta toiminnosta on monissa kuvapalveluissa käytössä oleva tunnisteiden (tags), eli sanallisten kuvauksien, merkitseminen tuotteisiin.

Tunnisteet ovat onnistuessaan hyvä tapa merkitä tuotteiden ominaisuuksia. Hyödyllinen tunnistejärjestelmä tarvitsee kuitenkin jonkinlaisen säännön siitä, millaisilla tunnisteilla minkäkinlainen tuote merkitään. Jos vastuu merkitsestä on yksittäisillä käyttäjillä, voivat tunnisteet jäädä epäselviksi ja niiden määrä vaihdella suurestikin [9]. Käyttäjien välillä ei myöskään välttämättä vallitse yhtenevä mielipide oikeanlaisesta tunnistetyylistä. Kaksi eri käyttäjää saattaakin merkitä samanlaisen kuvan täysin toisistaan poikkeavilla tunnisteilla. Ratkaisua tähän ongelmaan käydään läpi seuraavassa kappaleessa, joka toimii myös esimerkkinä sisältöpohjaisesta suositelujärjestelmästä.

## **2.2 Tunnisteiden suositelu käyttäjille yhteisötiedon perusteella**

Sigurbjörnsson ja van Zwol [9] esittelevät eri tapoja suositella käyttäjälle tunnisteita, jotka sopivat hänen palveluun lataamaansa sisältöön. Suositelulla tunnisteista saadaan järjestelmällisempiä ja yhdenmukaisempia.

Käyttäjien motivaatio merkitä kuvansa tunnisteilla tuntuu olevan niiden kyvyssä tuoda heidän lataamansa sisältö paremmin muiden käyttäjien näkyville [1]. Johdonmukaisilla tunnisteilla merkittyihin tuotteisiin törmää palvelua selatessa helpommin kuin sellaisiin, joita ei ole merkitty lainkaan tai jotka on merkitty epäselvästi. Esimerkiksi kuvien kohdalla käyttäjä voi lisätä tunnisteiksi kuvauspaikan ja mitä kuva hänen mielestään esittää. Näiden käyttäjän kirjoittamien tunnisteiden ja kaikkien kuvapalvelusta kerättyjen

tunnistetietojen perusteella voidaan kuvaan ehdottaa yleisiä lisätunnisteita. Suositeltujen tunnisteiden käyttö lisää tunnistetyylin johdonmukaisuutta ja helpottaa tietynlaisten kuvien hakua ja suosittelua.

Flickr-kuvapalvelu koostui vuonna 2008 8,5 miljoonasta rekisteröityneestä käyttäjästä ja valtavasta kuvamäärästä. Sigurbjörnsson ja van Zwol tarkastelivat tästä datasta satunnaisesti koostettua 52 miljoonan kuvan osajoukkoa [9]. Jokaisessa kuvassa oli vähintään yksi tunniste. Yhteensä tunnisteita oli noin 188 miljoonaa, joista 3,7 miljoonaa olivat uniikkeja.

Vain kerran esiintyvät tunnisteet ovat yleensä niin erikoislaatuista tai kirjoitusvirheellisiä, ettei niitä kannata suositella. Jos taas tunniste on yksi yleisimmistä käytetyistä, esimerkiksi vuosiluku, on se yleensä liian geneerinen suositeltavaksi [9]. Tämän ongelman ratkaisuun palaamme tuonnempana.

Käyttäjien antamien tunnisteiden määrä vaihtelee yksilökohtaisesti ja vaikuttaa suosittelun kannattavuuteen. Enimmillään tarkasteltavan joukon käyttäjät olivat merkinneet kuvaan yli 50 tunnistetta. Tällaisissa tilanteissa on vaikeaa tarjota hyödyllisiä tunnistesuosituksia. 64 % kuvista oli merkitty 1-3 tunnisteella, jolloin suosituksia on helppo muodostaa ja järkevää tarjota.

Tunnisteet voidaan jakaa eri kategorioihin käsittelyn helpottamiseksi. Aineiston suosituimpia tunnistetyyppejä olivat paikat (28 %), esineet tai artefaktit (16 %), ihmiset tai ryhmät (13 %), toiminnot tai tapahtumat (9 %) ja ajankohdat (7 %). Loput 27 % eivät menneet minkään näiden kategorian alle. Ne voitiin kuitenkin luokitella omiin alikategorioihinsa WordNet-kategorisoinnin (WordNet broad categories) avulla.

Tunnisteiden samassa yhteydessä esiintymisten (tag co-occurrence) laskeminen on suosittelun ydin. Menetelmä toimii luotettavasti vain suuren datamäärän kanssa, mutta käsiteltävästä aineistosta koostettu alijoukko oli tässä tapauksessa riittävä. Kuten edellä jo käsiteltiin, ovat eri tunnisteet suosittelun kannalta eriarvoisia keskenään. Pelkkä tunnisteiden yhteisesiintymisten laskeminen ei ota huomioon tunnisteiden esiintymisyleisyyttä, joten on suositeltavaa normalisoida tulos tunnisteiden kokonaisesiintymisellä. Normalisointiin esitellään kaksi tapaa: symmetrinen ja epäsymmetrinen.

Symmetrisessä normalisoinnissa voidaan normalisoida kahden tunnisteiden

$t_i$  ja  $t_j$  yhteiset esiintymiset Jaccardin kertoimen (Jaccard coefficient)

$$J(t_i, t_j) := \frac{|K(t_i) \cap K(t_j)|}{|K(t_i) \cup K(t_j)|}$$

mukaan, jossa  $K(t) = \{\text{kuva} \mid \text{kuvalla tunniste } t\}$ .

Jaccardin kerrointa käytetään yleisesti kahden objektin tai joukon samantyyppisyyden mittaamiseen. Sen kaltaiset symmetriset mittaukset soveltuvat hyvin kahden tunnisteiden merkitysten vertailuun.

Epäsymmetrisessä normalisoinnissa normalisointi tehdään yhden tunnisteiden esiintymismäärän perusteella. Voimme laskea kahden tunnisteiden  $t_i$  ja  $t_j$  yhteisesiintymisten todennäköisyyden ja normalisoida tuloksen tunnisteiden  $t_i$  esiintymisyleisyydellä

$$P(t_j|t_i) := \frac{|K(t_i) \cap K(t_j)|}{|K(t_i)|}$$

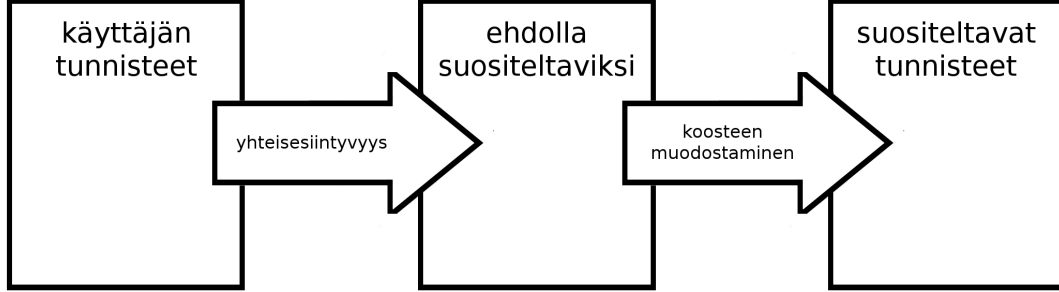
mukaisesti. Tässä  $P(t_j|t_i)$  kertoo, millä todennäköisyydellä kuvassa, joka on merkitty tunnisteella  $t_i$ , on myös tunniste  $t_j$ .

Symmetrinen normalisointi tuottaa tunnisteelle suosituksena useita samaa merkitseviä sanoja tai kuvan kohteen lähimaastossa sijaitsevia kohteita [9]. Esimerkiksi tunnisteiden *Eiffel Tower* tapauksessa suurimman yhteisesiintymisluvun saavat sanat *Tour Eiffel*, *Eiffel*, *Seine*, *La Tour Eiffel* ja *Paris*. Epäsymmetrisellä mittauksella saman sanan tulokset ovat *Paris*, *France*, *Tour Eiffel*, *Eiffel* ja *Europe*. Voidaankin olettaa, että epäsymmetrisellä yhteisesiintymisellä löydetään monipuolisempia suositeltavia tunnisteita.

Jos käyttäjän määrittelemiä tunnisteita on enemmän kuin yksi, mahdollisia suositeltavia tunnisteita on paljon. Tällöin muodostetaan tunnistekooste ja lyhennetään suositeltavien tunnisteiden listaa. Kuvasta 1 nähdään tunnistekooste-askeleen sijainti suositteluprosessissa. Esitellään esimerkiksi summaukseen perustuva tunnistekoosteen muodostaminen. Määritellään kolme tunnisteryhmää seuraavasti:

1. Käyttäjän määrittelemät tunnisteet  $U$ .
2. Ehdolla olevat tunnisteet  $C_u$ , jossa  $C_u$  on järjestetty lista  $m$ :stä useim-





Kuva 1: Tunnistesuosituksen muodostaminen.

min tunnisteeseen  $u$  kanssa yhteisesiintyvistä tunnisteesta, kun  $u \in U$ .

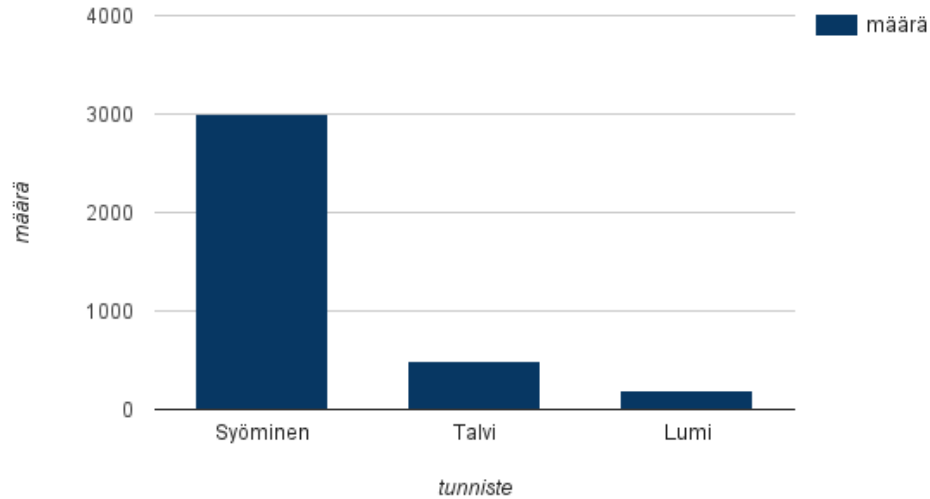
3. Suositeltavat tunnisteet  $R$ , jossa  $R$  on järjestetty lista  $n$ :stä suositeltuun parhaiten soveltuvasta tunnisteesta.

Tunnistekooste otetaan kaikkien ehdolla olevien tunnisteiden joukosta  $C = \cup_{u \in U} C_u$  ja tulokseksi saadaan lopullinen lista suositeltavista tunnisteista  $R$ . Summaukseen perustuvassa koosteenmuodostuksessa käytämme  $m$ :ää useimmin yhteisesiintyvää tunnistetta. Otetaan kaikkien ehdolla olevien tunnisteiden joukosta  $C$  yhdiste ja lasketaan yhteen tunnisteiden yhteisesiintymisarvot. Lasketaan ehdolla olevan tunnisteeseen  $c \in C$  arvosana (score)

$$score(c) := \sum_{u \in U} 1_{c \in C_u} (P(c|u)),$$

mukaisesti, jossa  $1_{c \in C_u}$  saa arvon 1 jos  $c \in C_u$  ja arvon 0 muuten. Funktio  $P(c|u)$  laskee epäsymmetrisen yhteisesiintymisen kuten aiemmin esiteltyssä epäsymmetrisen normalisoinnin funktiossa.

Havainnollistetaan seuraavaksi tunnisteiden suosittelua esimerkillä. Tarkastellaan kolmea eri tunnistetta: *Talvi* (500), *Lumi* (200) ja *Syöminen* (3000). Kuvasta 2 nähdään näiden tunnisteiden esiintymismäärät suhteessa toisiinsa. Tarkastellaan tilannetta jossa käyttäjä merkitsee kuvansa tunnisteella *Talvi*. Oletetaan, että tunnisteet *Talvi* ja *Lumi* esiintyvät muissa kuvissa yhdessä 150 kertaa ja tunnisteet *Talvi* ja *Syöminen* 45. Tätä tunnisteiden yhteisesiin-



Kuva 2: Esimerkin tunnisteiden jakautuminen kuvitteellisessa datassa.

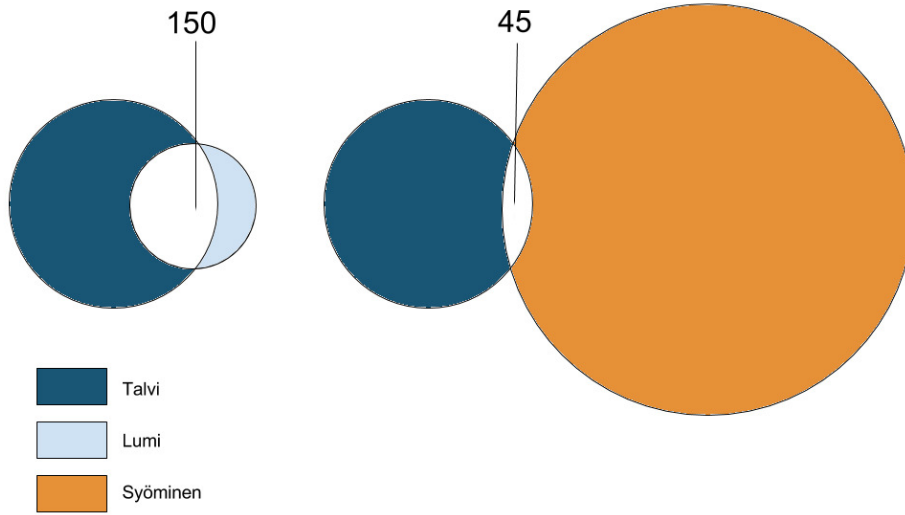
tyvyyttä havainnollistetaan kuvassa 3. Vertaillaan ensin tunnisteiden *Talvi* ja *Lumi* yhteisesiintyvyyttä kaavalla

$$P(Lumi|Talvi) := \frac{|K(Talvi) \cap K(Lumi)|}{|K(Talvi)|}$$

josta saadaan luku 0,3. Kun tehdään sama tunnisteille *Talvi* ja *Syöminen* niin saadaan luku 0,09.

Tunniste *Lumi* sai korkeamman arvosanan, joten suosittelemme sitä käyttäjälle mieluummin kuin tunnistetta *Syöminen*. Lumi vaikuttaa luontevamalta parilta talvelle kuin syöminen, joten tunnistesuosituksen voidaan sanoa olevan hyödyllinen. Oikeassa tapauksessa yhteisesiintyviä tunnisteita käyttäjän antamalle tunnisteelle olisi paljon enemmän kuin kaksi ja suositeltavien tunnisteiden listakin pitenisi.

Realistisemmassa esimerkissä tulisi olla kriittinen sen suhteen, minkälaiset tunnisteet saavat korkeimman arvosanan. Ensimmäisenä listassa on nimittäin usein niin yleisluontoisia tunnisteita, etteivät ne tarjoa tarpeeksi kuvaavaa informaatiota käsiteltävästä kuvasta [9]. Kuten jo kappaleen alussa mainittiin, tällaisia kaikkein yleisimmin käytettyjä tunnisteita ei kannata suositella



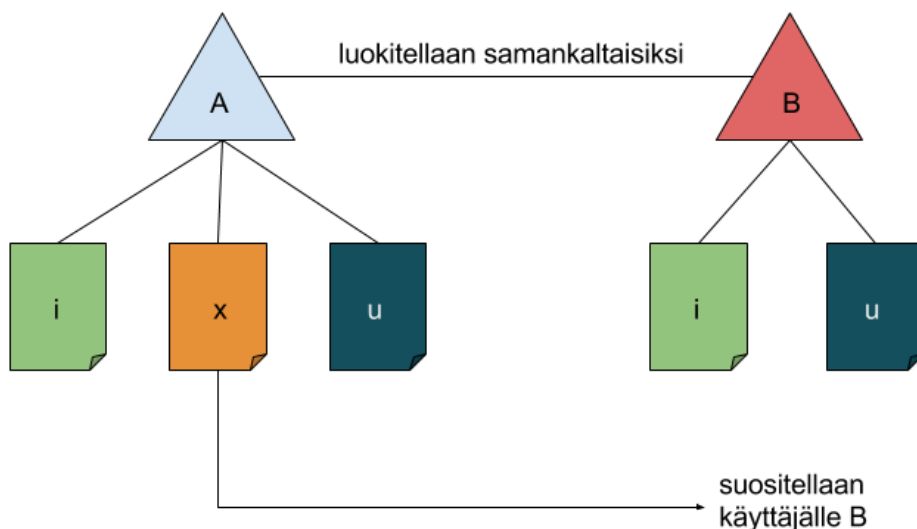
Kuva 3: Tarkasteltavien tunnisteiden yhteisesiintyminen.

käyttäjille. Ongelman ratkaisuksi tehdään askel nimeltä kuvailevuuden edistäminen (descriptiveness-promotion) ja alennetaan todella usein esiintyvien tunnisteiden arvosanaa funktiolla

$$kuvailevuus(c) := \frac{k_d}{k_d + (|k_d - \log(f_c)|)}$$

jossa  $f_c$  on tunnisteiden  $c$  esiintymistiheisyys aineistossa ja  $k_d$  on parametri, joka opitaan datasta. Artikkelissa parametrin  $k_d$  oppimiseen käytettiin 131 kuvan joukkoa.

Tarkasteltavassa artikkelissa evaluoitiin kaikkia edellä esiteltyjä funktioita suurilla kuva- ja tunnistejoukoilla. Lisäksi käytettiin toistakin koosteenmuodostusalgoritmia ja esiteltiin kolme suosittelua edelleen parantavaa askelta, joita emme tässä käsitelleet. Evaluoinnin lopuksi todettiin edellä esitellyn suosittelustrategian tuottavan hyödyllisiä tunnistesuosituksia [9].



Kuva 4: Havainnollistus yhteistoiminnalliseen suodattamiseen perustuvan suosittelujärjestelmän peruseriaatteesta.

### 3 Yhteistoiminnallisen suodattamisen järjestelmät

#### 3.1 Yleisesti

Yhteistoiminnalliseen suodattamiseen perustuvissa suosittelujärjestelmissä otetaan yksittäisen käyttäjän ja tuotteiden sijasta huomioon kaikkien käyttäjien väliset riippuvuudet. Suositusjärjestelmä vertaa käyttäjästä kerättyjä tietoja muiden käyttäjien tietoihin ja luokittelee käyttäjät samanlaisten mieltymysten perusteella pienemmiksi ryhmiksi. Jos joku tällaisen ryhmän jäsenistä on pitänyt jostakin tietystä tuotteesta, suositellaan tuotetta muillekin ryhmän jäsenille.

Kuvassa 4 on esitelty tällaisen järjestelmän periaate hyvin yleisellä tasolla. Käyttäjät  $A$  ja  $B$  ovat arvostelleet tuotteet  $i$  ja  $u$  samalla tavalla siten, että kumpikin pitää näistä tuotteista. Lisäksi käyttäjä  $A$  on arvostellut tuotteen  $x$  ja osoittanut pitävänsä siitä. Tämän perusteella suositellaan tuotetta  $x$  käyttäjälle  $B$ , jolla ei vielä ole arvostelua kyseisestä tuotteesta.

Automaattisen yhteistoiminnallisen suodattamisen (ACF) järjestelmiä väitetään perinteisiä sisältöpohjaisia tehokkaammaksi, sillä tuotteiden suodattaminen pohjautuu koneanalyysin sijasta käyttäjäyhteisön relaatioihin [6]. ACF-järjestelmät toimivat hyvin kaikenlaisen tiedon arvioinnissa, sellaisenkin, jota koneen on vaikea arvioida, kuten ihmisten makutottumukset tai laatuvaatimukset. ACF-järjestelmät eivät kuitenkaan ole syrjäyttäneet sisältöpohjaisia järjestelmiä, vaan niitä käytetään usein rinnakkain [4].

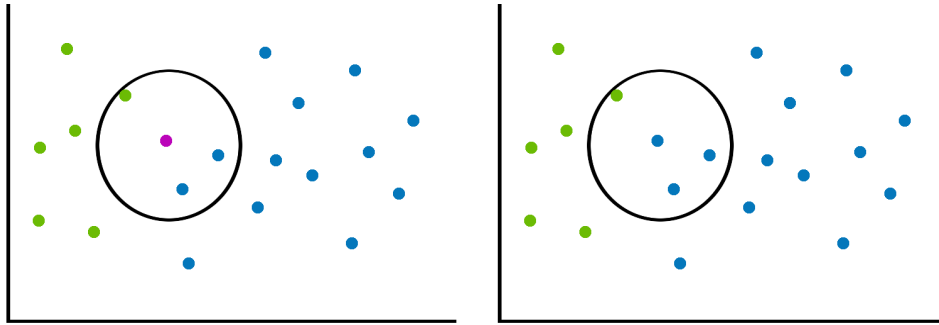
Yhteistoiminnallisen suodattamisen järjestelmät voidaan jakaa vielä kahteen alatyyppiin: muistipohjaisiin ja mallipohjaisiin [5]. Muistipohjaiset algoritmit tekevät ennusteita suoraviivaisesti käyttäjien arvosteluhistorioiden perusteella laskemalla eri käyttäjien tai tuotteiden samankaltaisuuksia. Yleisesti käytettäviä samankaltaisuuden mittaustapoja ovat Pearsonin korrelaatiokerroin ja kosinisamankaltaisuus (cosine similarity) arvosteluista muodostettujen vektoreiden välillä.

Mallipohjaiset algoritmit käyttävät historioita käyttäjien mallintamiseen ja ennustavat näillä malleilla tulevia arvosteluja kohteille, joita käyttäjät eivät ole vielä nähneet. Mallipohjaisia menetelmiä ovat esimerkiksi Bayesiläinen klusterointi, piilosemanttinen indeksointi (latent semantic indexing, LSI), todennäköisyyspohjainen piilosemanttinen indeksointi (PLSI) ja Markovin päätösprosessi (Markov Decision process) [5].

Yleinen tapa toteuttaa yhteistoiminnallisen suodattamisen suosittelujärjestelmä on käyttää muistipohjaisen ja mallipohjaisen tyyppien yhdistelmää [4]. Seuraavissa kappaleissa on esitelty kaksi eri projektia, jotka käyttivät kumpikin näiden tyyppien yhdistelmää.

## 3.2 Netflix Prize -kilpailu

Suosittelujärjestelmät nousivat suuremman yleisön puheenaiheeksi digitaalisen elokuvavuokraamo Netflixin vuonna 2006 järjestämän Netflix Prize -kilpailun ansiosta. Kilpailun tarkoituksena oli parantaa Netflixin käyttämää suosittelujärjestelmää ja pienentää annetun testidatan keskineliöpoikkeamaa 10 prosentilla. Testidata koostui yli 100 miljoonasta elokuva-arvostelusta noin 480 000 käyttäjältä 17 770 eri elokuvasta. Bell ja Koren käsittelevät artikke-



Kuva 5: Naapurustomallin (k-NN) peruseriaate. Valitaan violetin yksilön  $k$  lähintä naapuria (tässä  $k = 3$ ). Koska näissä naapureissa on enemmän sinisiä naapureita, luokitellaan violetti sinisten joukkoon.

lissaan [3] tässä kilpailussa vuoden sisällä parhaiten pärjänneen joukkueen suosittelujärjestelmämallia, joka saavutti 8,43 %:n parannuksen.

Huomattavaa joukkueen käyttämässä mallissa oli se, että se käytti kahden tärkeimmän yhteistoiminnallisen suodattamisen mallityypin yhdistelmää. Kummassakin mallissa on omat puutteensa, mutta yhdessä ne tuottivat hyviä tuloksia. Kappaleessa 3 esitellyn jaon mukaan toinen käytetyistä menetelmistä voidaan luokitella muistipohjaisiin ja toinen mallipohjaisiin algoritmeihin.

Muistipohjaisena toimii naapurustomalli (Neighborhood model, k-NN), joka on hyvä havaitsemaan paikallisia riippuvuuksia. Mallilla tarkastellaan kunkin alkion  $k$ :ta lähintä naapuria ja luokitellaan käsiteltävä alkio siihen ryhmään, jonka jäseniä on tarkasteltavissa naapureissa eniten. Kuvassa 5 on havainnollistettu naapurustomallin toimintaa. Mallilla koko tarkasteltava joukko saadaan jaettua pienempiin ryhmiin.

Joukkueen käyttämässä naapurustomallissa elokuvat jaotellaan pareihin, jotka on arvosteltu pääsääntöisesti samalla tavalla. Näiden parien avulla pyritään ennustamaan käyttäjän mielipide elokuvasta, jota hän ei ole arvostellut. Naapurustomalli ottaa huomioon vain osan käyttäjän arvosteluista eikä siis havaitse niissä piileviä heikkoja signaaleja.

Piilomuuttujamallit tulevat apuun siinä, missä naapurustomallissa on puutteita. Kun naapurustomalli havaitsee vain samankaltaisten elokuvien suhteet, hahmottaa piilomuuttujamalli elokuvien väliset riippuvuudet kattavammin.

Se pystyy esimerkiksi havaitsemaan saman lajityypin elokuvien olevan samankaltaisia keskenään. Toisin kuin naapurustomallit, se ei kuitenkaan huomioi pieniä, keskenään samankaltaisista elokuvista muodostuvia ryhmiä. Malli ei esimerkiksi pysty suositteluun trilogian ensimmäisen osan katsoneelle käyttäjälle sarjan toista osaa. Joukkueen käyttämän piilomuuttujamallin toiminta perustuu käyttäjästä ja elokuvasta muodostettaviin vektoreihin, joiden avulla saadaan ennuste käyttäjän arviolle elokuvasta.

Koren esittelee käytettyjen mallien teknistä puolta tarkemmin omassa artikkelissaan [7]. Tutustumme kahdesta esittelystä mallista lähemmin naapurustomalliin. Perinteistä naapurustomallia ei käytetty sellaisenaan, vaan sitä paranneltiin käyttötarkoitukseen sopivaksi. Pohjalla ollut, yleisesti käytössä oleva korrelaatioihin perustuva naapurustomalli,

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in S^k(i;u)} s_{ij}(r_{uj} - b_{uj})}{\sum_{j \in S^k(i;u)} s_{ij}}$$

antaa käyttäjän  $u$  ennustetun arvion  $r_{u_i}$  tuotteelle  $i$ . Mallissa

1. Ennuste  $r_{u_i}$  otetaan naapurituotteiden arvosteluiden painotettuna keskiarvona.
2. Funktio  $b_{ui} = \mu + b_u + b_i$  antaa pohja-arvion (baseline estimate) jossa  $b_u$  on käyttäjän  $u$  havaittu poikkeama keskiarvosta ja  $b_i$  vastaavasti tuotteen  $i$  havaittu poikkeama keskiarvosta. Parametri  $\mu$  on keskimääräinen arvosana tuotteen arviolle.
3. Joukko  $S^k(i; u)$  sisältää  $k$  käyttäjän  $u$  arvostelemaa kaikkein samankaltaisinta tuotetta tuotteen  $i$  kanssa.
4. Funktio  $s_{ij}$  mittaa tuotteiden samankaltaisuutta  $s_{ij} \stackrel{def}{=} \frac{n_{ij}}{n_{ij} + \lambda_2} p_{ij}$ , jossa  $n_{ij}$  on niiden käyttäjien lukumäärä, jotka arvostelivat tuotteet  $i$  ja  $j$ . Tyypillinen arvo parametrille  $\lambda_2$  on 100. Samankaltaisuuden mittaamisen pohjana on Pearsonin korrelaatiokerroin  $p_{ij}$ .

Tällaisenaan käytettynä malli ei kuitenkaan ole täysin ongelmaton. Ryhmä näki suurimpana ongelmana sen, että metodin toiminnalle ei ole olemassa

formaalia perustelua. Lisäksi samankaltaisuusmittauksen huomautettiin keskittyvän vain kahden tuotteen relaatioihin jättämällä näin huomiotta suhteet muuhun naapuristoon. Kolmanneksi todettiin, että interpolaatiopainojen summautessa yhteen on mallin pakko nojautua naapureihin sellaisissakin tapauksissa, kun naapurustotietoa ei löydy, eli käyttäjä  $u$  ei ole arvioinut tuotteen  $i$  kanssa samanlaisia tuotteita. Tällöin olisi järkevämpää katsoa pohja-arviota naapuruuksien sijaan [7]. Ongelmat ratkaistiin päivitetyllä mallilla

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in S^k(i;u)} \theta_{ij}^u (r_{uj} - b_{uj})$$

jossa annetulle joukolle naapureita  $S^k(i;u)$  lasketaan interpolaatiopainot  $\theta_{ij}^u | j \in S^k(i;u)$ . Uudistettu malli tuotti testeissä huomattavasti parempia tuloksia kuin alkuperäinen naapurustomalli, kun  $k$ :n arvo oli vähintään 500 [7].

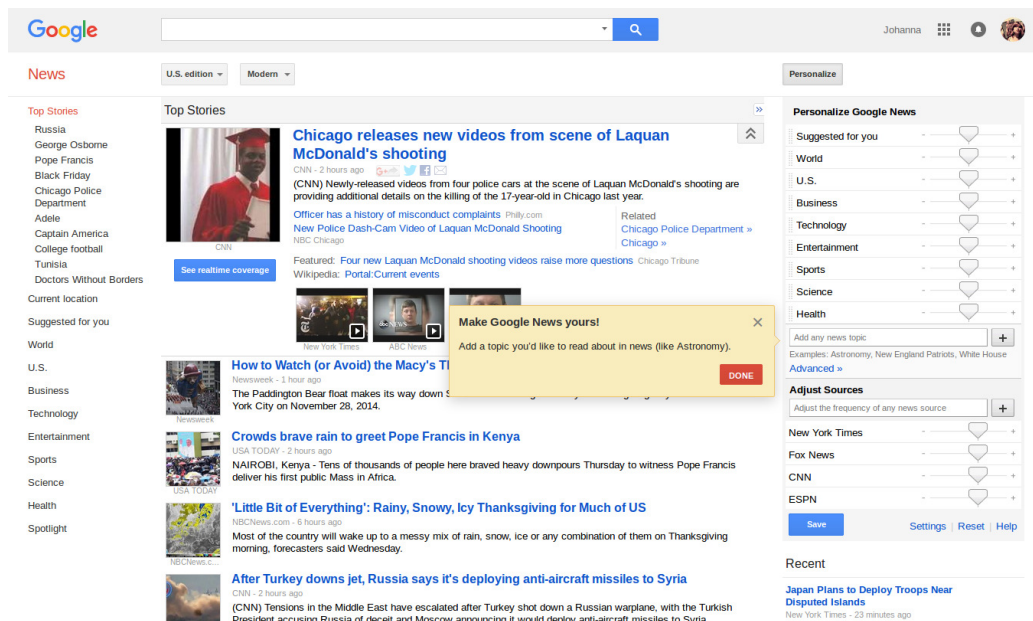
*tänne vähän naapurusto- ja piilomuuttujamallin yhdistelmäkäytöstä ja havaituista hyödyistä*

Joukkue huomasi olevan oleellista katsoa dataa muutenkin kuin arvostelujen sisältöjen osalta. Laajemman kuvan saamiseksi joukkue keskittyi myös siihen, minkä tyyppisiä elokuvia käyttäjä ylipäätään vaivautuu arvostelemaan. Se, mikä elokuva on niin vaikuttava, että se kannattaa arvostella, opettaa paljon käyttäjästä. Jotkut mallit ottivat huomioon myös muun muassa arvostelujen määrän, keskiarvon ja päivämäärät.

Haasteetta projektille toi ihmisten elokuvamaun mallintamisen vaikeus. Mallinnuksessa tulisi ottaa huomioon muun muassa sellaisia piirteitä kun tietynlainen tunnelma, äänimaailma tai dialogin laadukkuus ja päätellä niistä käyttäjän elokuvamakua. Tällaisten ominaisuuksien huomioon ottaminen on kuitenkin algoritmisesti hyvin vaikeaa.

Projektia vaikeuttivat myös käyttäjien vähäiset tai hajanaiset arviot testidatassa. Joukkue kehitti sekä naapurustomallia että piilomuuttujamallia tehokkaammaksi ja tarkoitukseen sopivammiksi. Eniten ongelmia tuotti datassa esiintyvä puutteellisuus arvojen osalta. Käytettävät mallit ovat standardimuodoltaan sellaisia, etteivät ne huomioi arvosteluiden vähäisyyttä. Jossakin tapauksessa olisi järkevämpää jättää jokin arvo kokonaan huomiotta kuin





Kuva 6: Kuvankaappaus Google Newsistä.

vääristää laskentaa puutteellisilla tiedoilla. Tästä aiheutuva ylisovittaminen (over fitting) oli yksi ongelma, jota saatiin vähennettyä parannetuilla malleilla.

Artikkelin julkaisemisen jälkeen Netflixin toimintaperiaate on muuttunut elokuvavuokraamosta suoratoistopalveluksi. Tämä vaikuttaa datan muotoon ja käyttäjien käyttäytymiseen. Myös suosittelujärjestelmien toimintaa on siksi muutettu edellä esitellyistä malleista. Artikkelissa ennakoitiin muutosta pohdinnassa, jossa esitettiin parempia suosittelutuloksia saatavan aikaiseksi seuraamalla itse arvostelujen ohella muita tietoja. Tällaisia ovat esimerkiksi käyttäjän selaus- tai hakuhistoria. [3]

### 3.3 Uutisartikkelien suosittelu käyttäjille - skaalautuva yhteistoiminnallinen suodattaminen

Google News on palvelu, joka kokoaa käyttäjilleen uutisartikkeleita monelta eri uutissivustolta ja luokittelee keskenään samanlaiset artikkelit ryhmiin. Käyttäjille tarjotaan suosituksia luettavista artikkeleista heidän lukuhistoriansa perusteella. Das ym. [5] esittelevät artikkelissaan yhteistoiminnalliseen suodat-

tamiseen perustuvan ratkaisuehdotuksensa suosittelujen tarjoamiseen. Ryhmän päätavoitteenaan oli rakentaa skaalautuva online-suosittelevä järjestelmä, jota voitaisiin käyttää suurissa palveluissa kuten Google Newsissä.

Google News -palvelun ominaisuudet loivat joitakin haasteita järjestelmän rakentamiselle. Valtava kävijämäärä ja miljoonat uutisartikkelit asettavat vaatimuksen skaalautuvuudelle. Palvelun osiot ovat myös jatkuvassa muutoksen tilassa, kun artikkeleita poistuu ja lisätään muutaman minuutin välein. Toisin kuin monissa muissa staattisissa palveluissa, käytettävä suosittelumalli vanhennee nopeasti eikä korjaannu pienillä muutoksilla. Osioden jatkuva muutos on merkittävin tekijä, joka erottaa rakennettavan järjestelmän muiden suurien palveluiden suosittelujärjestelmistä [5].

Google News -palvelulla on käyttäjänä sekä rekisteröitymättömiä että rekisteröityneitä käyttäjiä, joista jälkimmäisille tarjotaan enemmän käytettäviä ominaisuuksia. Kuvassa 6 näkyy rekisteröityneen käyttäjän näkymä palvelussa. Käyttäjän niin salliessa Google tallentaa rekisteröityneen käyttäjän uutisartikkeleihin liittyviä aktiviteetteja muiltakin Googlen palveluilta ja tallentaa nämä artikkelit käyttäjän lukuhistoriaan Google News -palvelussa. Esimerkki tällaisesta aktiviteetista voisi olla vaikka uutisartikkelin hakeminen Googlen hakukoneella. Kootun historian pohjalta muodostetaan artikkelisuosituksia, joista kolmea tarjotaan käyttäjän luettavaksi. Artikkelin projektissa keskityttiin suositusten antamiseen juurikin rekisteröityneille käyttäjille.

Toisin kuin joissakin suositusjärjestelmiä käyttävissä palveluissa, Google Newsissä tarkasteltavat tuotteet eivät saa suoraa arvosanaa käyttäjältään. Projektissa päätettiin käsitellä käyttäjän käyttäytymistä arvioinnin pohjana siten, että klikkaus uutisartikkeliin tulkitaan myönteisenä äänenä artikkelille. Päätöstä perusteltiin sillä, että käyttäjille tarjotaan lyhyt, selkeä kuvaus jokaisesta artikkelista listausnäkyvässä. Voidaan siis olettaa, että käyttäjä on todennäköisimmin kiinnostunut artikkelista, jos hän vielä kuvauskappaleen lukemisenkin jälkeen päättää klikata artikkelia. Klikkaukset kuvastavat kuitenkin vain käyttäjien myönteisiä mielipiteitä, eivätkä kerro mitään siitä, mistä käyttäjät eivät pidä [5].

Google News -palvelu on yksi maailman suosituimmista uutissivustoista. Tarkasteltavassa projektissa havainnoitiin uutisartikkeleita yhden kuukauden

ajalta ja artikkeleita kertyi useita miljoonia. Käyttäjien klikkausaktiivisuus artikkeleihin on hyvin vaihtelevaa ja klikkaushistorioiden koko vaihtelee nolasta satoihin tai jopa tuhansiin.

Google asettaa tarkkoja vaatimuksia palveluidensa vasteajoille myös Google Newsin kohdalla. Esimerkiksi kotisivun näkymä generoidaan tyypillisesti sekunnissa. Tästä sekunnista jää muiden toimintojen jälkeen jäljelle muutama sata millisekuntia suositteluiden muodostamiseen. Tiukat aikavaatimukset olivatkin yksi projektin haasteista.

Kappaleessa 3.1 esiteltiin jako malli- ja muistipohjaisiin yhteistoiminnallisen suodattamisen järjestelmiin. Artikkelissa käsiteltävässä järjestelmässä käytettiin niin sanottua hybridimallia, eli sekoitusta kummankin tyyppisestä järjestelmästä.

Muistipohjaisena algoritmina toimii Covisitation, jota esitellään tarkemmin jäljempänä. Mallipohjaisessa lähestymisessä käytetään kahta klusterointitekniikkaa, algoritmeja PLSI (todennäköisyyspohjainen piilosemanttinen indeksointi) ja MinHash. Kaikki nämä kolme algoritmia asettavat tarkasteltaville uutisartikkeleille arvosanat siten, että paremmat suositukset saavat korkeamman numeerisen arvon. Lopussa kaikki kolme arvosanaa yhdistetään painottaen kaavalla

$$\sum_a w_a r_s^{(a)}$$

missä  $w_a$  on algoritmin  $a$  paino ja  $r_s^{(a)}$  on algoritmin  $a$  antama arvosana artikkelille  $s$  ja saadaan järjestetty lista artikkeleita. Tästä listasta valitaan  $K$  parhaimman arvosanan saanutta artikkelia ja suositellaan niitä käyttäjälle.

Ensimmäisenä esittelemme todennäköisyyspohjaisen klusterointialgoritmin MinHash. MinHash jakaa parin käyttäjiä samaan klusteriin sen todennäköisyyden perusteella, jolla käyttäjät ovat äänestäneet eli klikanneet samoja artikkeleita. Jokainen käyttäjä  $u \in U$  esitetään tämän klikkaushistoriana  $C_u$ , eli joukkona artikkeleita, joita kyseinen käyttäjä on klikannut.

Kahden käyttäjän  $u_i$  ja  $u_j$  samankaltaisuus on mahdollista mitata käyttämällä jo kappaleessa 2.2 esiteltyä Jaccardin kerrointa  $S(u_i, u_j) = \frac{|C_{u_i} \cap C_{u_j}|}{|C_{u_i} \cup C_{u_j}|}$ . Jos haluaisimme tarjota käyttäjälle  $u_i$  suosittelua, laskisimme ensin tämän samankaltaisuuden kaikkien muiden käyttäjien  $u_j$  kanssa ja suosittelisimme

sitten (käyttäjälle  $u_i$ ) muiden käyttäjien  $u_j$  äänestämiä artikkeleita, joiden paino on yhtäläinen suureen  $S(u_i, u_j)$  kanssa. Tämän tekeminen reaaliajassa ei kuitenkaan ole skaalautuvaa, joten Jaccardin kerroin ei semmoisenaan kelpaa artikkelin projektin käyttöön.

Ongelman ratkaisuna käytetään tiivistesten muodostamista LSH-tekniikalla (Locality Sensitive Hashing). LSH:n keskeinen ajatus on muodostaa datapisteistä tiiviste useita tiivistefunktioita käyttäen ja päätellä läheiset naapurit kyselypisteen tiivisteiden avulla. Jaccardin kertoimen kanssa käytettäväksi soveltuu LSH-tekniikka MinHash. MinHashin perusidea on permutoida satunnainen joukko artikkeleita ( $S$ ) ja laskea jokaiselle käyttäjälle  $u_i$  tiivistearvo indeksinään ensimmäinen artikkeli permutaatiossa, joka kuuluu käyttäjän artikkelijoukkoon  $C_{u_i}$ . Todennäköisyys, että kahdella kaikkien  $S$ :n permutaatioiden joukosta valitulla permutaatiolla on sama tiiviste, on täysin yhtenevä niiden Jaccardin kerroin -luvun eli samankaltaisuuden kanssa. MinHashin jokaisen tiivisteluokan voidaan ajatella vastaavan klusteria. Samaan klusteriin laitetaan kaksi käyttäjää todennäköisyydellä, joka on yhtenevä näiden käyttäjien artikkelijoukkojen samankaltaisuudella  $S(u_i, u_j)$ .

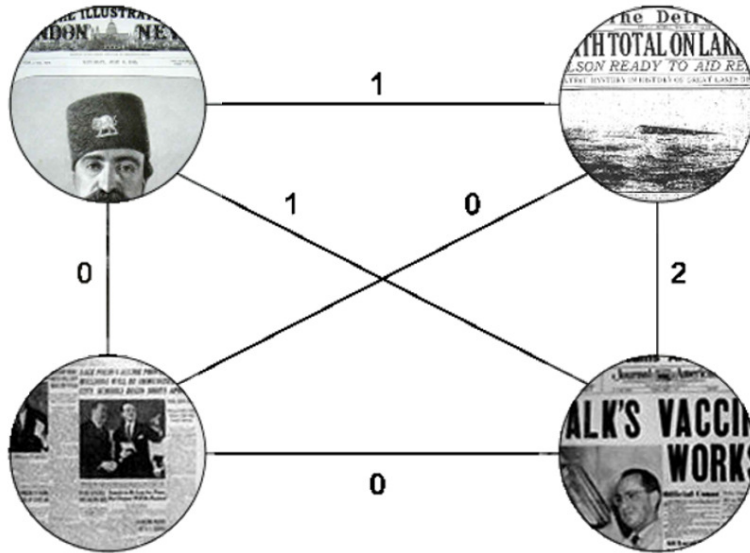
Toinen esitelty klusterointialgoritmi PLSI mallintaa käyttäjät ( $u \in U$ ) ja artikkelit ( $s \in S$ ) satunnaismuuttujina. Käyttäjien ja artikkeleiden väliset suhteet opitaan mallintamalla yhteisjakauma käyttäjistä ja artikkeleista sekoitejakaumana (mixture distribution). Suhdetta merkitään piilomuuttujalla  $Z$ , jonka voidaan ajatella kuvastavan käyttäjä- ja artikkeliyhteisöjä siten, että samankaltaiset käyttäjät ovat oma yhteisönsä ja samankaltaiset artikkelit omansa. Malli voidaan kirjoittaa sekoitemallina

$$p(s|u; \theta) = \sum_{z \in Z} p(z|u)p(s|z)$$

kun  $\theta$  on todennäköisyysjakauman parametrit sisältävä vektori.

Covisitation("kanssavierailu") esiteltiin projektin muistipohjaisena osiona. Kanssavierailu on tapahtuma, jossa sama käyttäjä klikkaa kahta artikkelia tietyn ajan sisällä. Voimme ajatella uutisartikkeleita kuvan 7 tapaisena verkkona, jossa artikkelisolmut on painotettu kanssavierailujen lukumäärillä.

Verkkoa käsitellään vierekkäisyyslistana, jonka avaimina on artikkeli-id:t.



Kuva 7: Uutisartikkelit verkkona, jossa painoina kanssavierailujen lukumäärät.

Kun havaitaan käyttäjän  $u_i$  klikkaavan artikkelia  $s_k$ , käydään läpi kyseisen käyttäjän klikkaushistoria  $C_{u_i}$ . Jokaisen artikkelin  $s_k \in C_{u_i}$  kohdalla muokataan sekä  $s_j$  ja  $s_k$  vierekkäisyyslistoja lisäämällä niihin kyseiseen klikkaukseen viittaava uusi merkintä. Jos kyseiselle parille on jo olemassa merkintä, päivitetään verkon painoja siten, että vanhojen kanssavierailujen merkitys vähenee ajan mittaan.

## 4 Pohdinta

Suosittelujärjestelmiä käytetään runsaasti kaikenlaisia tuotteita tarjoavissa palveluissa. Esimerkiksi kappaleessa 2.2 esitellyn Flickr-kuvapalvelun tunnistneiden suosittelussa tulee esille ikään kuin kahden kerroksen suosittelua. Ensimmäisellä kerroksella käyttäjille suositellaan tunnisteita, jotka sopivat heidän kuviinsa. Näitä hyviä tunnisteita käyttämällä mahdollistuu toinen suosittelukerros, eli kuvien suosittelu niistä selaaville käyttäjille. Palvelu voi suositella käyttäjille heitä kiinnostavia kuvia käyttämällä joko sisältöpoh-

jaista tai yhteistoiminnallista suosittelujärjestelmää, mutta emme perehtyneet tähän "ylempään kerrokseen" tarkemmin.

Kappaleessa 2 mainittiin tunnisteiden käytön olevan hyvä keino kerätä tietoa kuvan kaltaisista tuotteista, joista on muuten saatavissa vain vähän tietoa. Mainittakoot kuitenkin, että on olemassa erilaisia tapoja analysoida kuvien sisältöä ilman tunnisteita [9]. Voitaisiinkin ajatella olevan mahdollista perustaa tunnisteiden suosittelu näille tiedoille yhteisötiedon sijaan.

mitä kuvista voi päätellä tunnistetietoihin, eri tunnistemenetelmät mitä ei käyty

## 5 Yhteenveto

Suosittelujärjestelmiä käytetään muun muassa kuva- elokuva- ja uutisartikkelipalveluissa. Järjestelmät voidaan jakaa sisältöpohjaisiin tai yhteistoiminnallisen suodattamisen järjestelmiin sen perusteella, tarkastellaanko vain käyttäjän omaa osto-, arvostelu- tai selaushistoriaa vai myös muiden käyttäjien vastaavia historioita.

Sisältöpohjainen suosittelu perustuu yhden käyttäjän historiatietoihin ja tuotteiden vertailuun keskenään. Esimerkkinä sisältöpohjaisen suodattamisen suosittelujärjestelmästä esiteltiin hyödyllisten tunnisteiden suosittelu käyttäjille.

Yhteistoiminnalliseen suodattamiseen perustuvassa suosittelussa keskitytään käyttäjäyhteisön välisiin relaatioihin. Samoja tuotteita kuluttaneiden käyttäjien oletetaan pitävän myös sellaisista tuotteista, joita vain toinen heistä on kuluttanut.

Sisältöpohjaisen ja yhteistoiminnallisen suosittelujärjestelmän ero on pohjimmiltaan siinä, käytetäänkö suositteluun vain yhden käyttäjän vai koko käyttäjäyhteisön mieltymyksiä. Tehokkaimmat tulokset näytettäisiin saavan käyttämällä useamman kuin yhden suosittelumallin yhdistelmää [3].

## Lähteet

- [1] Ames, Morgan ja Naaman, Mor: *Why We Tag: Motivations for Annotation in Mobile and Online Media*. Teoksessa *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, sivut 971–980, New York, NY, USA, 2007. ACM, ISBN 978-1-59593-593-9.
- [2] Balabanović, Marko ja Shoham, Yoav: *Fab: Content-based, Collaborative Recommendation*. Commun. ACM, 40(3):66–72, maaliskuu 1997, ISSN 0001-0782.
- [3] Bell, Robert M. ja Koren, Yehuda: *Lessons from the Netflix Prize Challenge*. SIGKDD Explor. Newsl., 9(2):75–79, joulukuu 2007.
- [4] Burke, Robin: *Hybrid Recommender Systems: Survey and Experiments*. User Modeling and User-Adapted Interaction, 12(4):331–370, marraskuu 2002, ISSN 0924-1868.
- [5] Das, Abhinandan S., Datar, Mayur, Garg, Ashutosh ja Rajaram, Shyam: *Google News Personalization: Scalable Online Collaborative Filtering*. Teoksessa *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, sivut 271–280, New York, NY, USA, 2007. ACM.
- [6] Herlocker, Jonathan L., Konstan, Joseph A. ja Riedl, John: *Explaining Collaborative Filtering Recommendations*. Teoksessa *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, CSCW '00, sivut 241–250, New York, NY, USA, 2000. ACM.
- [7] Koren, Yehuda: *Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model*. Teoksessa *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, sivut 426–434, New York, NY, USA, 2008. ACM, ISBN 978-1-60558-193-4.
- [8] Leskovec, Jure, Rajaraman, Anand ja Ullman, Jeffrey D.: *Mining of Massive Datasets*. Cambridge University Press, Cambridge, United Kingdom, 2. painos, 2014.

- [9] Sigurbjörnsson, Börkur ja van Zwol, Roelof: *Flickr Tag Recommendation Based on Collective Knowledge*. Teoksessa *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, sivut 327–336, New York, NY, USA, 2008. ACM.