

# **Yhteistoiminnalliset ja sisältöpohjaiset suosittelujärjestelmät**

Johanna Wahtera

Kandidaatintutkielma  
Helsingin yliopisto  
Tietojenkäsittelytieteen laitos

Helsinki, 30. marraskuuta 2015

Tiedekunta — Fakultet — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
Johanna Wahtera			
Työn nimi — Arbetets titel — Title			
Yhteistoiminnalliset ja sisältöpohjaiset suosittelujärjestelmät			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Kandidaatintutkielma	30. marraskuuta 2015	14	
Tiivistelmä — Referat — Abstract			
<p>Aineessa tutustutaan sekä sisältöpohjaiseen että yhteistoiminnalliseen suodattamiseen perustuvien suosittelujärjestelmien toimintaan yleisellä tasolla. Aineesta selviää, mikä on näiden kahden suosittelujärjestelmätyypin oleellisin ero. Kummastakin järjestelmätyypistä esitellään esimerkki, joka on sisältöpohjaisen suosittelun tapauksessa Flickr-kuvapalvelun tunnisteiden suosittelu käyttäjille. Yhteistoiminnallista suosittelua avataan esittelemällä Netflix Prize -kilpailuun osallistuneen joukkueen suosittelujärjestelmää.</p>			
Avainsanat — Nyckelord — Keywords			
suosittelujärjestelmät, yhteistoiminnallinen suodatus, sisältöpohjainen järjestelmä, Flickr, Netflix			
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>6</b>
3.1	Yleisesti . . . . .	6
3.2	Netflix Prize -kilpailu . . . . .	7
3.3	Uutisartikkelien suosittelu käyttäjille - skaalautuva yhteisoi- minnallinen suodattaminen . . . . .	9
<b>4</b>	<b>Pohdinta</b>	<b>13</b>
<b>5</b>	<b>Yhteenveto</b>	<b>13</b>
	<b>Lähteet</b>	<b>13</b>

# 1 Johdanto

Internet koostuu valtavasta määrästä monimuotoista sisältöä. Verkkoa selaava käyttäjä löytää niin tuotteita, artikkeleita kuin itse sivustojakin lähes loputtomasti. Tällaisen rajattomien mahdollisuuksien edessä voi kuitenkin olla vaikeaa löytää jotakin juuri itseään kiinnostavaa. Esimerkiksi suuren verkkokaupan laajaa valikoimaa ei voida asettaa kerralla käyttäjän näkyville samalla tavoin kuin perinteisessä kivijalkakaupassa. Sama 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. On syntynyt tarve henkilökohtaisille kulutussuosituksille.

Suosittelujärjestelmä valikoi käyttäjän puolesta tuotteita, joita tarjotaan hänelle kulutettavaksi. Järjestelmä kerää dataa käyttäjästä 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. Tässä aineessa keskitymme näiden järjestelmien eroihin ja tutustumme muutamaa esimerkkijärjestelmään.

Sisältöpohjaisissa järjestelmissä kerätään tietoa suositeltavista tuotteista ja vertaillaan näitä toisiinsa. 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 pelkkien tuotteiden ominaisuuksien sijasta tuotteiden ja käyttäjien välisiin suhteisiin. Keskenään samankaltaiset käyttäjät lajitellaan heidän antamiensa arvosteluiden perusteella. Tuotteiden samankaltaisuus määritellään vertailemalla useamman samankaltaisen käyttäjän arvosteluja. Koska tämänkaltaiset järjestelmät perustuvat käyttäjältä kerättyihin arvosteluihin, ne soveltuvat vain sellaisiin

järjestelmiin, joissa tuotteet ovat ylipäänsä arvosteltavissa. *tämä on vähän outo lause, voidaanhan esim. tarkastella vain käyttäjän klikkailukäyttäytymistä ja käsitellä sitä "arvosteluina".*

## 2 Sisältöpohjaiset järjestelmät

### 2.1 Yleisesti

Sisältöpohjaisissa suosittelujärjestelmissä vertaillaan käyttäjälle tarjottavan sisällön ominaisuuksia toisiinsa ja pyritään löytämään niiden väliset samankaltaisuudet. Jokaiselle tuotteelle muodostetaan tuoteprofiili, johon kerätään tuotteen tärkeimmät ominaisuudet. Nämä tiedot ovat yleensä saatavilla suoraan tekstinä tuotteen tiedoista [6].

Profiilien 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.

Tuoteprofiilin lisäksi myös käyttäjästä rakennetaan käyttäjäprofiili. Käyttäjäprofiili rakentuu samoista osista kuin tuoteprofiili, mutta tuotetiedon paikalle merkitään käyttäjän mieltymys kyseisen tiedon suhteen. Esimerkki tällaisesta tiedosta voisi olla vaikkapa elokuvan lajityyppi, jolloin toiminnasta pitävän käyttäjän profiiliin merkitään lajityypin kohdalle "toiminta".

Joidenkin tuotteiden ominaisuuksia on vaikea vertailla keskenään. Esimerkiksi kuvista on mahdollista saada vain rajallinen määrä suositusjärjestelmän kannalta oleellista tietoa. Tällaisissa tapauksissa järjestelmässä onkin usein käytössä tuotteiden tunnistetoiminto (tags), jossa tuotteiden ominaisuuksia merkitään sanallisin tunnuksin.

Kun vastuu merkitsemisestä on käyttäjällä, voivat tunnisteet jäädä epäselviksi ja niiden määrä vaihdella suurestikin. 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 seuraavassa kappaleessa, joka toimii myös esimerkkinä sisältöpohjaisesta suosittelujärjestelmästä.

## 2.2 Tunnisteiden suosittelu käyttäjille yhteisötiedon perusteella

Sigurbjörnsson ja van Zwol [7] esittelevät eri tapoja suositella käyttäjälle tunnisteita, jotka sopivat hänen palveluun lataamaansa sisältöön. Suositellulla tunnisteista saadaan järjestelmällisempiä ja yhdenmukaisempia.

Käyttäjien motivaatio käyttää tunnisteita 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 tunnisteksi 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 [7]. 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 erikoislaatuisia tai kirjoitusvirheellisiä, ettei niitä kannata suositella. Jos taas tunniste on yksi yleisimmin käytetyistä, esimerkiksi vuosiluku, on se yleensä liian geneerinen suositeltavaksi [7].

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

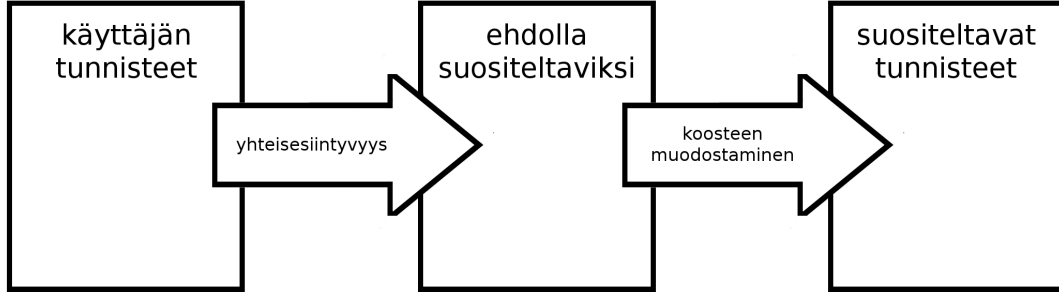
Jaccardin kerrointa käytetään yleisesti kahden objektin tai joukon samantyyppisyyden mittaamiseen. Sen kaltaiset symmetriset mittaukset soveltuvat hyvin kahden tunnisteiden merkitysten vertailuun.

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  $P$  kertoo, millä todennäköisyydellä kuvassa, joka on merkitty tunnisteella  $t_i$ , on myös tunniste  $t_j$ .

Symmetrinen normalisointi tuottaa tunnisteelle hyvin samankaltaisia tunnistesuosituksia [7]. Esimerkiksi tunnisteiden *Eiffel Tower* tapauksessa suurimman yhteisesiintymisluvun saavat sanat *Tour Eiffel*, *Eiffel*, *Seine*, *La Tour Eiffel* ja *Paris*. Epäsymmetrisellä mittauksella saman sanan tulokset ovat



Kuva 1: Tunnistesuosituksen muodostaminen.

*Paris, France, Tour Eiffel, Eiffel* ja *Europe*. Nähdään, että epäsymmetrisellä yhteisesiintymisellä löydetään monipuolisempia suositeltavia tunnisteita.

Jos käyttäjän määrittelemiä tunnisteita on enemmän kuin yksi, mahdollisia suositeltavia tunnisteita on paljon. Tällöin muodostetaan tunniste-kooste ja lyhennetään suositeltavien tunnisteiden listaa. Kuvasta 1 nähdään tunniste-kooste-askleen sijainti suositteluprosessissa. Esitellään esimerkiksi summaukseen perustuva tunniste-koosteen muodostaminen. Määritellään kolme tunnistryhmää seuraavasti:

1. Käyttäjän määrittelemät tunnisteet  $U$ .
2. Ehdolla olevat tunnisteet  $C_u$ , jossa  $C_u$  on järjestetty lista  $m$ :stä useimmin tunnisteiden  $u$  kanssa yhteisesiintyvistä tunnisteista, kun  $u \in U$ .
3. Suositeltavat tunnisteet  $R$ , jossa  $R$  on järjestetty lista  $n$ :stä suositteluun parhaiten soveltuvasta tunnisteesta.

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



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

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 esitellyssä epäsymmetrisen normalisoinnin funktiossa.

*tähän tulossa vielä tekstiä ja kuva, jossa näytetään miten toimii konkreettisesti*

## 3 Yhteistoiminnallisen suodattamisen järjestelmät

### 3.1 Yleisesti

Yhteistoiminnalliseen suodattamiseen perustuvissa suosittelevissa järjestelmissä 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 ryhmiä. Jos joku tällaisen ryhmän jäsenistä on pitänyt jostakin tietystä tuotteesta, suositellaan tuotetta muillekin ryhmän jäsenille.

Automaattisen yhteistoiminnallisen suodattamisen (ACF) järjestelmiä väitetään perinteisiä sisältöpohjaisia tehokkaammaksi, sillä tuotteiden suodattaminen pohjautuu koneanalyysin sijasta käyttäjäyhteisön relaatioihin [4]. 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.

Yhteistoiminnallisen suodattamisen järjestelmät voidaan jakaa vielä kahteen alatyypin: muistipohjaisiin ja mallipohjaisiin. Muistipohjaiset algoritmit tekevät ennusteita suoraviivaisesti käyttäjien arvosteluhistorioiden perusteella laskemalla eri käyttäjien tai tuotteiden samankaltaisuuksia. Yleisesti käytettäviä samankaltaisuuden mittaustapoja ovat Pearsonin korrelaatioker-

roin 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 päätöksentekoprosessi (Markov Decision process) [3].

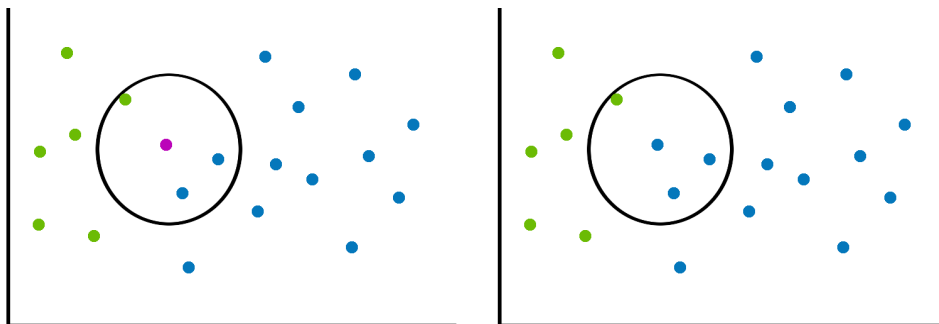
Yleinen tapa toteuttaa yhteistoiminnallisen suodattamisen suosittelujärjestelmä on käyttää muistipohjaisen ja mallipohjaisen tyyppin yhdistelmää. Seuraavissa kappaleissa on esitelty kaksi eri projektia, jotka käyttivät kumpikin näiden tyyppien yhdistelmää.

## 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 artikkelissaan [2] 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.

Toinen näistä malleista on 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 2 on havainnollistettu naapurustomallin toimintaa. Mallilla koko tarkasteltava joukko saadaan jaettua pienempiin ryhmiin.



Kuva 2: Naapurustomallin (k-NN) perusperiaate. Valitaan violetin yksilön  $k$  lähintä naapuria (tässä  $k = 3$ ). Koska näissä naapureissa on enemmän sinisiä naapureita, luokitellaan violetti sinisten joukkoon.

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.

*joukkue käytti PARANNELTUA naapurustomallia, tänne MITEN he sitä paransivat ja miksi! tästä tarkemmin artikkelissa [5]*

*naapurustomallia ja piilomuuttujamallia käytettiin yhdistelmänä, tänne MITEN*

Joukkue huomasi olevan oleellista katsoa dataa muutenkin kuin arvostelujen sisältöjen osalta. Laajemman kuvan saamiseksi joukkue keskittyi myös

siihen, minkä tyyppisiä elokuvia käyttäjä ylipäättään 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.

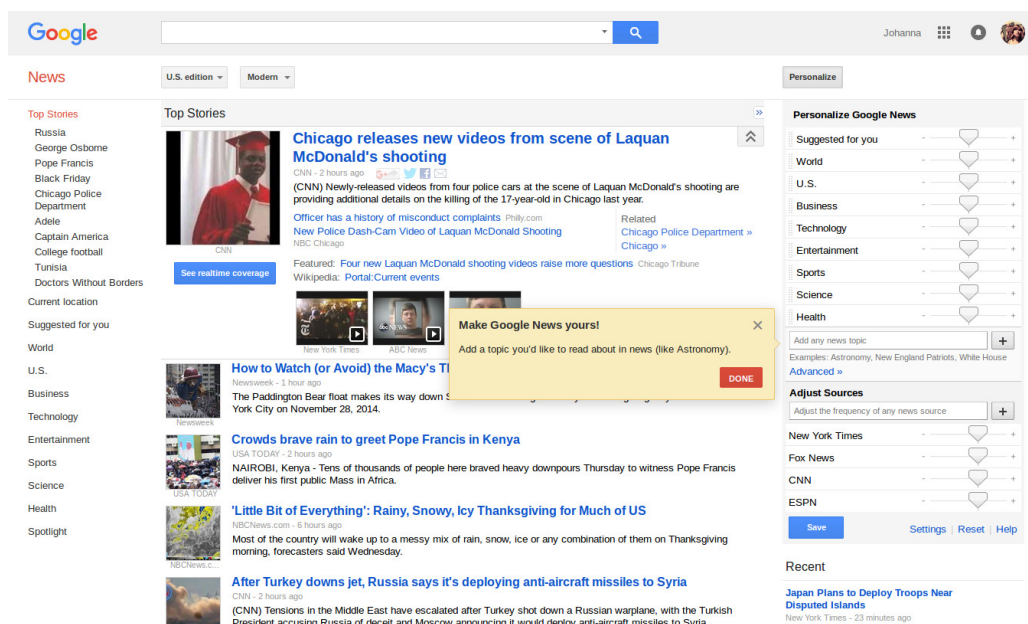
Haasteetta projektille toi ihmisten elokuvamaun mallintamisen vaikeus. Mallinnuksessa tulisi ottaa huomioon muun muassa sellaisia piirteitä kun tietynlainen tunnelma, äänimaailma tai dialogin laadukkuus ja päätellä niistä käyttäjän elokuvamakua. Tällaisten ominaisuuksien huomioon ottaminen on kuitenkin 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 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. [2]

### **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. [3] esittelevät artikkelissaan yhteistoiminnalliseen suodattamiseen perustuvan ratkaisuehdotuksensa suosittelujen tarjoamiseen. Ryh-



Kuva 3: Kuvankaappaus Google Newsistä.

män päätavoitteenaan oli rakentaa skaalautuva online-suositteleva 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 suosittelemalli vanhennee nopeasti eikä korjaannu pienillä muutoksilla. Osioiden jatkuva muutos on merkittävin tekijä, joka erottaa rakennettavan järjestelmän muiden suurien palveluiden suositteleva järjestelmästä [3].

Google News -palvelulla on käyttäjänä sekä rekisteröitymättömiä että rekisteröityneitä käyttäjiä, joista jälkimmäisille tarjotaan enemmän käytettäviä ominaisuuksia. Kuvassa 3 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 [3].

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 algoritmia toimii Covisitation, jota esitellään tarkemmin kappaleen loppupuolella. Mallipohjaisessa lähestymisessä käytetään kahden klusterointitekniikkaa, algoritmeja PLSI (probabilistic latent semantic indexing) ja MinHash. Kaikki nämä kolme algoritmia asettavat tarkasteltaville uutisartikkeleille arvosanat siten, että paremmat suositukset saavat korkeamman numeerisen arvon. Lopussa kaikki kolme arvosanaa yhdistetään

painottaen

$$\sum_a w_a r_s^a$$

, (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 todennäköisyyden, jolla käyttäjät ovat äänestäneet eli klikanneet samoja artikkeleita, perusteella. Jokainen käyttäjä  $u \in U$  esitetään tämän klikkaushistoriana  $C_u$ , eli joukkona artikkeleita ( $s \in S$ ), joita kyseinen käyttäjä on klikannut.

Kahden käyttäjän  $u_i$  ja  $u_j$  samankaltaisuus mitataan käyttämällä jo kappa-leessa 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  $S(u_i, u_j)$  kanssa. Tämän tekeminen reaaliajassa ei kuitenkaan ole skaalautuvaa, joten Jaccardin kerroin ei semmoisenaan kelpaa artikkelin projektin käyttöön.

Ongelman ratkaisuna käytetään tiivisteiden muodostamista LSH-algoritmillla.  
*lyhyesti siitä tähän*

Klusterointialgoritmi PLSI mallintaa käyttäjät ( $u \in U$ ) ja artikkelit ( $s \in S$ ) satunnaismuuttujina. Käyttäjien ja artikkeleiden väliset suhteet opitaan mallintamalla yhteisjakauma käyttäjistä ja artikkeleista yhdistelmäjakautana (mixture distribution). Suhdetta merkitään piilomuuttujalla  $Z$ , jonka voidaan ajatella kuvastavan käyttäjä- ja artikkeliyhteisöjä siten, että samankaltaiset käyttäjät ovat oma yhteisönsä ja samankaltaiset artikkelit omansa. *tämä luonnollisesti vielä jatkuu...*

*tähän Covisitation*

## 4 Pohdinta

Voisiko täällä olla vaikka kappalekohtaista pohdintaa. Esim jotain tägeistä, jotain ihmisten elokuvamausta jne.

## 5 Yhteenveto

Suosittelujärjestelmiä tarvitaan monessa eri kontekstissa. Esimerkkini sisältöpohjaisen suodattamisen suosittelujärjestelmästä on hyvälaatuisten tunnisteteiden suosittelu käyttäjille. Tässä tulee esille ikään kuin kahden kerroksen suosittelua. Ensin käyttäjille suositellaan tunnisteita, jotka sopivat heidän kuviinsa. Näitä hyviä tunnisteita käyttämällä mahdollistuu toinen suositteluominaisuus, eli kuvien suosittelu niitä selaaville käyttäjille. Palvelu voi suositella käyttäjille heitä kiinnostavia kuvia käyttämällä joko sisältöpohjaista tai yhteistoiminnallista suosittelujärjestelmää.

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 [2].

## 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] Bell, Robert M. ja Koren, Yehuda: *Lessons from the Netflix Prize Challenge*. SIGKDD Explor. Newsl., 9(2):75–79, joulukuu 2007.
- [3] 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.



- [4] 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.
- [5] 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.
- [6] Leskovec, Jure, Rajaraman, Anand ja Ullman, Jeffrey D.: *Mining of Massive Datasets*. Cambridge University Press, Cambridge, United Kingdom, 2. painos, 2014.
- [7] 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.