

LAPPEENRANNAN TEKNILLINEN YLIOPISTO  
TEKNISTALOUEDELLINEN TIEDEKUNTA  
TIETOTEKNIKAN KOULUTUSOHJELMA

DIPLOMITYÖ

## **Sumea suositin elokuville**

Diplomityön aihe on hyväksytty 21.10.2009

Tarkastajat: Professori Ville Kyrki ja TkT Jarmo Ilonen.

Ohjaaja: Professori Ville Kyrki

Lappeenrannassa 12. kesäkuuta 2012

Simo-Pekka Tiittanen

# TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto  
Teknistaloudellinen tiedekunta  
Tietotekniikan koulutusohjelma  
Tiittanen, Simo-Pekka

## Sumea suositin elokuville

Diplomityö

2012

76 sivua, 16 kuvaa, 10 taulukkoa.

Tarkastajat: Professori Ville Kyrki  
TkT Jarmo Ilonen

Hakusanat: suositusmenetelmä, sumea logiikka, yhteisösuodatus  
Keywords: Recommender, Fuzzy Logic, Collaboratory Filtering

Suositusmenetelmien tarkoituksena on auttaa käyttäjää löytämään häntä kiinnostavia asioita ja välttämään asioita, joista hän ei pitäisi. Suositusmenetelmät antavat suositukset yleensä terävinä lukuina. Tässä työssä kehitetään suositusmenetelmä, joka antaa suositukset arvosanojen sumeina jäsenyysasteina. Menetelmän antamat suositukset voidaan myös perustella käyttäjälle. Menetelmä kuuluu pääosin yhteisösuodatusmenetelmiin, joissa suositukset tehdään käyttäjien antamien arvosanojen perusteella, mutta myös tietoa elokuvien tyylilajeista hyödynnetään suositustarkkuuden parantamiseksi. Sumeiden suositusten suositeltavuusjärjestyksen laskemiseen esitetään myös menetelmä.

Käyttäjien elokuville antamat arvosanat voidaan käsittää sumeana datana. Käyttäjä voi kuvata arvosanaa esimerkiksi ilmaisulla ”noin 4”. Tästä syystä on loogista esittää suosituksetkin sumeina lukuina. Tällöin käyttäjälle voidaan antaa tietoa suosituksen tarkkuudesta ja mahdollisista ristiriidoista. Epävarmojen suositusten tapauksessa käyttäjä voi painottaa enemmän muita tietolähteitä. Kokeiden perusteella kehitetty menetelmä antaa joissa tapauksissa selvästi vertailtavia menetelmiä parempia suosituksia, kun taas toisissa tapauksissa suositukset ovat selvästi heikompia.

# **ABSTRACT**

Lappeenranta University of Technology  
Faculty of Technology Management  
Degree Program of Information Technology  
Tiittanen, Simo-Pekka

## **Fuzzy Recommender for Movies**

Master's Thesis

2012

76 Pages, 16 figures, 10 tables.

Examiners: Professor Ville Kyrki  
D.Sc. Jarmo Ilonen

Keywords: Recommender, Fuzzy Logic, Collaborative Filtering

Recommender systems help users to find items they find interesting and avoid items they do not find interesting. Recommender systems usually give recommendations as crisp values. In this thesis a recommendation system is developed that gives recommendations as fuzzy membership values of the given grades. The system can also explain the recommendations to the user. The developed method is mostly a collaborative filtering method, meaning it uses user given grades to make the recommendations, but it also exploits movie genre information to improve recommendation accuracy. A method to sort the fuzzy recommendations is also presented.

User given movie grades can be considered to be fuzzy data. An user can for example describe some grade with phrase "about 4". For this reason it is logical to give the recommendations also as fuzzy values. This way it is possible to give the user more information for example about recommendation accuracy. Because users will usually consider also other information sources, they can rely more on these other sources when the recommendation indicates poor accuracy. Experiments show that the developed method gives significantly better recommendations in some cases and significantly worse in some other cases compared to other methods.

# ALKUSANAT

Tämä työ on tehty Lappeenrannan teknillisen yliopiston Tietotekniikan koulutusohjelmassa.

Kiitos Professori Ville Kyrjelle ohjaamisesta katkonaisesti edenneen työn aikana ja työn tarkastamisesta. Kiitos TkT Jarmo Iloselle työn tarkastamisesta ja rakentavista kommentteista. Kiitos TkT Risto Tiiaselle paikoitellen humoristisestakin oikolukemisesta ja parannusehdotuksista. Kiitos isälleni Martti Tiittaselle ja siskolleni Anni Tiittaselle oikolukemisesta ja parannusehdotuksista. Kiitokset myös perheelle, ystäville ja työtovereille tukemisesta hieman pitkäksi venähtäneiden opintojen aikana.

Simo-Pekka Tiittanen

12.6.2012

# SISÄLLYSLUETTELO

<b>1</b>	<b>JOHDANTO</b>	<b>9</b>
1.1	Taustaa . . . . .	9
1.2	Tavoitteet ja rajaukset . . . . .	11
1.3	Työn rakenne . . . . .	12
<b>2</b>	<b>SUMEASTA LOGIIKASTA</b>	<b>13</b>
2.1	Sumeudesta yleisesti . . . . .	13
2.2	Sumeista joukoista . . . . .	14
2.3	T-normit ja t-konormit . . . . .	15
2.4	Sumeista relaatioista . . . . .	17
<b>3</b>	<b>SUMEEA LOGIIKAA HYÖDYNTÄVIÄ SUOSITUSJÄRJESTELMIÄ</b>	<b>19</b>
3.1	Film-Conseil . . . . .	19
3.2	Tapahtumien suosittelu sumeilla relaatioilla . . . . .	20
3.3	Hybridi elokuva-suositin . . . . .	21
3.4	FARAMS . . . . .	22
3.5	Looginen kehys sumeaa yhteisösuodatukseen . . . . .	22
3.6	Sumean logiikan metodeja suositukseen . . . . .	24
3.7	Karheat joukot ja sumea klusterointi . . . . .	25
3.8	Komponenttikohtainen vakaa lineaarinen sumea klusterointi yhteisösuodatukseen . . . . .	26
3.9	Agenttipohjainen sumea yhteisösuodatin . . . . .	27
3.10	Kontekstittietoinen musiikkisuositusjärjestelmä . . . . .	28
3.11	Sumea suositusjärjestelmä kuluttajaelektronikalle . . . . .	28
3.12	Sumeaan likimääräiseen päättelyyn perustuva älykäs verkkosivujen suositusjärjestelmä . . . . .	30
3.13	Opetusohjelmien sisällön suosittelu . . . . .	31
<b>4</b>	<b>SUOSITUSMENETELMÄN KUVAUS</b>	<b>32</b>
4.1	Suosituksen esittäminen arvosanojen sumeina jäsenyysasteina . . . . .	32
4.2	Suositusäännöt . . . . .	33
4.3	Arvosanojen vastaavuus . . . . .	37
4.4	Säännön merkittävyys . . . . .	39
4.5	Säännön ulostulo . . . . .	40
4.6	Sääntöjen ulostulojen yhdistäminen . . . . .	41
4.7	Vastaavuuksien optimointi . . . . .	42
4.8	Tyylilajibonus . . . . .	44

4.9	Suosituksen järjestäminen . . . . .	45
<b>5</b>	<b>SUORITETTAVAT KOKEET</b>	<b>48</b>
5.1	Datasta . . . . .	48
5.2	Datan jakaminen kokeita varten . . . . .	48
5.3	Verrattavat menetelmät . . . . .	49
5.3.1	Arvosanojen keskiarvo . . . . .	49
5.3.2	Pearsonin korrelaatio . . . . .	50
5.4	Kokeet . . . . .	50
5.4.1	Suositusarkkuus ja suositusten kattavuus . . . . .	50
5.4.2	Suositusjärjestys . . . . .	51
<b>6</b>	<b>KOKEIDEN TULOKSET</b>	<b>52</b>
6.1	Parhaat parametrien arvot . . . . .	52
6.2	T-konormien vertailu . . . . .	55
6.3	Suosituksen vastaavuuden kerroin . . . . .	56
6.4	Suosituksen vastaavuuden eksponentti . . . . .	58
6.5	Merkittävyyden rajoittaminen yhteisten elokuvien määrän perusteella . . . . .	59
6.6	Tyylilajibonuksen vaikutus . . . . .	61
6.7	Käyttäjän antamien arvioiden määrän vaikutus suosituksiin . . . . .	63
6.8	Suositusjärjestys . . . . .	65
6.9	Suoritusajoista . . . . .	69
6.10	Tulosten yhteenveto . . . . .	70
<b>7</b>	<b>YHTEENVETO JA JOHTOPÄÄTÖKSET</b>	<b>72</b>
	<b>LÄHTEET</b>	<b>74</b>

## SYMBOLIT JA LYHENTEET

FARAMS	Fuzzy Association Rules and Multiple Level Similarity
IMDB	Internet Movie Database
IP	Internet Protocol
IRT	Item Response Theory
OWA	Ordered Weight Average
RMSE	Root Mean Square Error
URL	Uniform Resource Locator
<b>T</b>	T-normi
<b>S</b>	T-konormi
$M_{i,j}(x)$	Käyttäjän $i$ merkittävyys käyttäjän $j$ antamalle arvosanalle $x$
$\Sigma$	Summa
$S_{i,j}(x,y)$	Käyttäjän $i$ vastaavuus käyttäjän $j$ antamien arvosanojen $x$ ja $y$ välillä.
$W_{i,j,x}$	Käyttäjän $i$ paino käyttäjän $j$ antamalle arvosanalle $x$
$\max(x,y)$	maksimi-operaatio
$\min(x,y)$	minimi-operaatio
$\wedge$	Ja-operaatio. Vapaasti valittava t-normi
$\vee$	Tai-operaatio. Vapaasti valittava t-normi
$\bigwedge_i$	$i$ :n tekijän ja-operaatio
$\bigvee_i$	$i$ :n tekijän tai-operaatio
$M_{i,j}^f(x)$	Käyttäjän $i$ rajoitettu merkittävyys käyttäjän $j$ antamalle arvosanalle $x$
$C_{i,j}$	Käyttäjien $i$ ja $j$ yhteisten elokuvien joukko
$f$	Raja-arvo käyttäjien yhteisten elokuvien määrälle
$H_{i,r}(I)$	Käyttäjän $i$ merkittävyys säännölle $r$ sisääntuloon $I$ nähden
$H_i^{\max}$	Korkein mahdollinen säännön merkittävyys käyttäjälle $i$

$O_{i,r}(I)$	Ulostulo käyttäjän $i$ säännölle $r$ sisääntuloon $I$ nähden
$P_{i,x}(I)$	Käyttäjän $i$ ulostulo arvosanalle $x$ sisääntuloon $I$ nähden
$F(P,y)$	Sumean suosituksen $P$ virhe tiedettyyn arvosanaan $y$ nähden
$m$	Kerroin sumean arvosanan virheelle
$d$	EkspONENTTI sumean arvosanan virheelle
$B(g_1, g_2)$	Tyylilajibonus tyylilajijoukoille
$b$	Tyylilajibonuksen kertoimen maksimi
$O_{i,r}^b(I)$	Ulostulo käyttäjän $i$ säännölle $r$ sisääntuloon $I$ nähden, käyttäen bonusta $b$
$V(x^s)$	Suositusjärjestyksen virhe



# 1 JOHDANTO

## 1.1 Taustaa

Tietoyhteiskunnan kehittymisen myötä ihmisistä ja asioista tallennetun tiedon määrä on kasvanut huomattavasti. Tämä tiedon määrän kasvu luo sekä haasteita että mahdollisuuksia. Tiedon hyödyntäminen on sen valtavan määrän vuoksi ongelmallista, sillä oleellisen tiedon löytäminen epäoleellisen tiedon seasta on vaikeaa. Kerätyn tiedon varaan voidaan kuitenkin kehittää monenlaisia sovelluksia.

Yksi kerättyä tietoa hyödyntävä sovellus on suositusjärjestelmät. Suositusjärjestelmä pyrkii kerätyn tiedon perusteella suositteluun käyttäjälle asioita, joista hän mahdollisesti pitäisi. Yksi esimerkki on internetkauppojen suositusjärjestelmät, jotka kertovat, mistä muista tuotteista jotain tuotetta tarkasteleva asiakas voisi olla kiinnostunut. Erilaisia palveluita ja tuotteita myyvät internetsivustot ovatkin suositusjärjestelmien yleisimpiä käyttäjiä.

Suositusmenetelmät voidaan jakaa karkeasti kahteen tyyppiin: yhteisösuodatusmenetelmiin (collaborative filtering) ja asioiden piirteisiin pohjautuviin menetelmiin. Yhteisösuodatusmenetelmissä suositukset perustuvat käyttäjien asioille antamiin arvosanoihin. Asioiden piirteisiin perustuvissa menetelmissä suositin pyrkii etsimään piirteiden perusteella käyttäjän pitämien asioiden kanssa samanlaisia asioita. Molemmissa lähestymistavoissa on omat ongelmansa. Yhteisösuodattimien ongelmana on yleensä tiedettyjen arvosanojen vähäinen määrä tuntemattomiin arvosanoihin nähden. Suositeltavien asioiden piirteisiin perustuvien suositusjärjestelmien ongelmana on piirteiden kerääminen. Suositeltavista asioista täytyy analysoida piirteitä suosittelua varten. Usein analysointia ei voida kokonaan automatisoida, joten se vaatii jonkin verran ihmisten työpanostusta. Suurten datamäärien analysoiminen on tästä syystä varsin työläs tehtävä. Näiden ongelmien takia onkin kehitetty suositusjärjestelmiä, jotka hyödyntävät molempia menetelmiä [1].

Esimerkkejä yleisessä käytössä olevista suositusjärjestelmistä ovat Pandora Radio ja Netflix. Pandora Radio on internetradio, joka pyrkii soittamaan käyttäjän mieltymysten mukaista musiikkia. Soittolista perustuu käyttäjän antamiin mieltymyksiin ja kappaleista kerättyihin piirteisiin. Suosituksessa käytetyt piirteet saadaan muusikoiden ja asiantuntijoiden analyysien perusteella. Kuuntelijalle soitetaan kappaleita, joiden piirteet ovat lähellä hänen pitämiensä kappaleiden piirteitä.

Toinen yleisesti käytetty suositusjärjestelmä on Yhdysvalloissa toimiva videovuokraus-yritys Netflix. Yritys toimittaa asiakkaalle elokuvia ja televisiosarjoja postin ja Internetin välityksellä. Käyttäjä pitää yllä listaa, jonka mukaan hänelle toimitetaan tuotteita. Seuraava tuote lähetetään asiakkaalle, kun hän on ensin palauttanut tai ostanut edellisen tuotteen. Käyttäjät voivat antaa katsomilleen tuotteille arvosanoja yhdestä viiteen. Kerättyjen arvosanojen perusteella suositusjärjestelmä pyrkii arvioimaan, minkä arvosanan käyttäjä antaisi elokuvalla, jota hän ei ole vielä arvioinut. Käyttäjä voi hyödyntää suosituksia tehdessään listaa haluamistaan tuotteista.

Netflix järjesti kilpailun suositusjärjestelmänsä parantamiseksi. Kilpailu alkoi 2.10.2006. Tavoitteena oli kehittää suositusjärjestelmä, joka antaisi vanhaa järjestelmää kymmenen prosenttia parempia suosituksia. Paremmuutta mitattiin suositusten ja todellisten arvosanojen välisenä virheenä. Kilpailuun osallistujille annettiin osa yrityksen keräämistä arvosanoista. Arvosanat olivat kokonaislukuja väliltä yhdestä viiteen. Suositukset piti antaa desimaalilukuina samalta väliltä. [2] Kilpailu päättyi 21.9.2009, kun yksi joukkue saavutti kilpailun voittamiseksi asetetut ehdot [3].

Kilpailu sai varsin paljon huomiota osakseen, minkä seurauksena siinä määriteltäviä käytäntöjä ja arviointimenetelmiä käytetään varsin laajalti. Suositusten antaminen desimaalilukuina hävittää kuitenkin osan käyttäjän kannalta mahdollisesti hyödyllisestä informaatiosta. Ovatko esimerkiksi kaikki saman arvosanan saaneet elokuvat yhdenvertaisia? Annetun datan perusteella on hyvin mahdollista, että joidenkin elokuvien saama suositus perustuu suurempaan määrään tietoa kuin toisen elokuvan suositus. Käyttäjä ei voi kuitenkaan tätä eroa annetun suosituksen perusteella havaita. Koska käyttäjä tekee joka tapauksessa lopullisen päätöksen siitä, katsooko hän elokuvan vai ei, olisi parempi antaa hänelle enemmän informaatiota päätöksen tekemiseksi. Suositusjärjestelmän hyödyllisyyden kannalta on havaittu, että järjestelmän tulisi pystyä kertomaan käyttäjälle, mihin suositus perustuu [1, 4].

Sumea logiikka on klassisen logiikan laajennus, jossa totuusarvot voivat olla jotain toden ja epätoden välillä. Sillä voidaan käsitellä asioita, joita ei voida luontevasti kuvata vain kahdella totuusarvolla. Jäsenyysaste on sumean logiikan käsite, joka kuvaa asian kuuluvuuden astetta johonkin joukkoon. Elokuvien suositelussa joukot voidaan käsittää arvosanoina, jolloin elokuva voi jollain jäsenyysasteella kuulua arvosanan joukkoon. Tällä tavalla ilmaistujen suositusten avulla käyttäjälle voidaan antaa enemmän tietoa. Jäsenyysasteen avulla voidaan ilmaista ristiriitaisia suosituksia, jotka tiedetyn datan perusteella viittaisivat useampaan arvosanaan. Myös täyden tietämättömyyden ilmaiseminen on mahdollista, jolloin elokuvalla ei ole jäsenyysastetta minkään arvosanan joukossa. Tämä

on yhteisösuodatukseen perustuvissa suositusmenetelmissä varsin yleinen tilanne, sillä tiedettyjä arvosanoja on tuntemattomiin arvosanoihin nähden yleensä varsin vähän.

Myös käyttäjien antamat arvostamat voidaan käsittää sumean logiikan avulla. Sanallisesti käyttäjä voisi kuvata antamaansa arvosanaa esimerkiksi ilmaisuilla ”noin viisi”, ”joku kolme” tai ”sellainen kaksi”. Nämä ovat selkeästi sumeutta osoittavia ilmaisuja, jotka johtuvat siitä, että on lähes mahdoton kiteyttää elokuva yhdeksi kokonaisluvuksi. Huomioitavaa on myös se, että arvosanojen antaminen desimaalilukuina ei parantaisi tätä tilannetta laisinkaan, sillä ongelma ei johdu arvostama-asteikon riittämättömästä tarkkuudesta, vaan koko lähtökohdan epämääräisyydestä. Useita sumeaa logiikkaa hyödyntäviä suositusjärjestelmiä on kehitetty. Niitä käsitellään tarkemmin luvussa 3.

## 1.2 Tavoitteet ja rajaukset

Tämän työn tavoitteena on kehittää sumeaa logiikkaan perustuva suositusjärjestelmä, joka antaa suositukset arvosanojen jäsenyysasteiden avulla. Järjestelmän toiminnan tulisi olla läpinäkyvää ja annetut suositukset täytyisi pystyä myös perustelevaan käyttäjälle. Suositusjärjestelmää ei kehitetä todelliseen käyttöön, mutta sen tulisi olla laskentavaatimuksiltaan realistisesti käytettävissä myös oikeassa sovelluksessa.

Tarkoituksena ei ole kehittää maailman parasta suosintijärjestelmää, sillä se vaatisi todennäköisesti huomattavasti diplomityötä suuremman projektin. Kehitetyn järjestelmän suositustarkkuutta pyritään kuitenkin vertaamaan muihin suositusjärjestelmiin, vaikka desimaalilukuina suositukset antaviin järjestelmiin vertaaminen ei olekaan suoraan mahdollista. Myös arvosanojen jäsenyysasteina annettujen suositusten hyötyjä pyritään arvioimaan.

Arvosanojen jäsenyysasteina annettujen suositusten suositeltavuusjärjestykseen laittamiseen kehitetään myös menetelmä, joka oppii järjestyksen tiedettyjen arvosanojen ja niille laskettujen suositusten perusteella.

Datana työssä käytetään MovieLens-sivuston julkaisemaa dataa, joka sisältää käyttäjien elokuville antamia arvosanoja ja tiedon elokuvien tyylilajeista. Kehitettävä järjestelmä ei ole kuitenkaan mitenkään sidottu juuri elokuvien suositteluun. Elokuvat valittiin suosittelukohteeksi, koska niistä on yleisesti saatavilla järjestelmän kehitykseen soveltuvaa dataa.

### **1.3 Työn rakenne**

Luku kaksi on johdatus sumeaan logiikkaan, siinä käydään läpi työn kannalta olennaiset sumean logiikan käsitteet. Luvussa kolme tarkastellaan aikaisempia sumeaa logiikkaa hyödyntäviä suositusjärjestelmiä. Luvussa neljä kuvataan kehitetyn suositusmenetelmän toimintaperiaate. Luvussa viisi esitellään kokeissa käytetyt menetelmät. Luku kuusi sisältää kokeiden tulokset ja tulosten tulkinnan. Luku seitsemän on yhteenveto työstä.

## 2 SUMEASTA LOGIIKASTA

Tässä luvussa esitellään kehitetyn suositusmenetelmän kannalta olennaiset sumean logiikan osa-alueet. Luvussa kolme esiteltävien muiden sumeiden suositusjärjestelmien käyttämiä sumean logiikan menetelmiä ei tässä luvussa käydä läpi, vaan ne esitellään lyhyesti kyseisen menetelmän kuvauksen yhteydessä. Kappale perustuu pääosin Jorma Mattilan kirjaan ”Sumean logiikan oppikirja” [7].

### 2.1 Sumeudesta yleisesti

Ihmisten kielenkäyttö ja ajattelu perustuu enemmän sumeisiin kuin tarkkoihin käsitteisiin. Käsitteet, kuten ”pieni”, ”iso”, ”kuuma”, ”kylmä”, ”kevyt” ja ”painava”, eivät ilmaise mitään tarkkaa arvoa, mutta siitä huolimatta ovat käyttökelpoisia käsitteitä ihmisten välisessä kommunikoinnissa. Tämänlaisten sumeiden käsitteiden merkitys riippuu myös yleensä siitä, missä tilanteessa niitä käytetään. ”Kuumaa” tarkoittaa hyvin erisuuruisia lämpötiloja puhuttaessa huoneistosta, saunasta tai uunista. Sumeasti ilmaistut arvot helpottavat monimutkaistenkin järjestelmien toiminnan kuvaamista, sillä pääpiirteittäinen järjestelmän rakenne ei huku yksityiskohtiin. Esimerkiksi lause ”jos on kuuma, laita tuuletin kovemmalle” kuvaa varsin riittävästi ilmaistun toimenpiteen. Edellinen esimerkki on hyvin yksinkertainen, mutta samanlaista lähestymistapaa voidaan käyttää myös monimutkaisemmissa tapauksissa.

Sumean logiikan avulla voidaan käsitellä sumeutta tarkasti määriteltyjen sääntöjen avulla. Sumea logiikka ei siis itsessään ole mitenkään huonosti tai epäselvästi määriteltyä. Sumea logiikka ei myöskään tarkoita, että logiikan toiminta olisi jotenkin vaikeasti ymmärrettävää tai hämärää. Päinvastoin sumean logiikan tarkoituksena on tuoda ihmismäisempää ajattelua esimerkiksi monimutkaisten järjestelmien kuvaamiseen. Liialliseen tarkkuuteen pyrkiminen monimutkaisten ja vaikeasti määriteltävien järjestelmien kuvaamisessa voi johtaa virheisiin ja heikkoon luotettavuuteen [5]. Järjestelmien toimintaa ei yksinkertaisesti pystytä tuntemaan riittävällä tarkkuudella, jotta niistä voitaisiin luoda täydellinen malli.

Vaikka sumeuden ja todennäköisyyden ilmaisemiseen käytetään yleensä yksikköväliä  $[0,1]$ , ne eivät kuitenkaan tarkoita samaa asiaa. Otetaan esimerkiksi jokin esine, joka kuuluu kevyiden esineiden joukkoon jäsenyysasteella 0,9. Tämä ei ole todennäköisyys, sillä esineen todellinen paino on tiedossa, eikä siihen siis liity minkäänlaista tilastollista toden-

näköisyyttä.

## 2.2 Sumeista joukoista

Sumean logiikan perusta on sumeiden joukkojen käsite, jonka Lotfi A. Zadeh esitteli artikkelissaan ”Sumeat joukot” (Fuzzy Sets) [6] vuonna 1965. Sumeat joukot ovat klassisen joukko-opin laajennus, jossa alkio voi kuulua vain osittain joukkoon. Klassisen joukko-opin joukoissa, joita kutsutaan myös teräviksi joukoiksi, alkio joko kuulu tai ei kuulu joukkoon. Olkoon  $A$  perusjoukon  $X$  terävä osajoukko. Joukon  $A$  karakteristinen funktio voidaan esittää seuraavasti:  $f_A : X \rightarrow \{0, 1\}$ , missä  $f_A$  määritellään yhtälöllä 1.

$$f_A(x) = \begin{cases} 1, & \text{kun } x \in A \\ 0, & \text{muulloin} \end{cases} \quad (1)$$

Sumeiden joukkojen tapauksessa karakteristinen funktio ei ole rajoitettu joukon  $\{1,0\}$  alkioihin, vaan se voi saada minkä tahansa arvon suljetulta yksikköväliä  $[0,1]$ . Sumeiden joukkojen yhteydessä tällaista funktiota kutsutaan jäsenyysfunktioiksi. Jäsenyysfunktio määrittelee sumean joukon. Sumeista joukoista käytetään myös joskus nimitystä sumea osajoukko, joka viittaa siihen, että sumea joukko on kaikki mahdolliset alkiot sisältävän joukon osajoukko. Sumea joukko on tyhjä, jos yhdelläkään alkiolla ei ole siinä nollasta poikkeavaa jäsenyysastetta. Sumea joukko on normaali, jos ainakin yhden alkion jäsenyysaste on siinä 1. Sumean joukon  $A \subset X$  jäsenyysfunktio  $\mu_A$  voidaan esittää yhtälöllä

$$\mu_A : X \rightarrow [0, 1] \quad (2)$$

jossa jokaisella joukon  $X$  alkiolla  $x$  on jäsenyysaste  $\mu_A(x) \in [0, 1]$ . Sumea joukko  $A$  voidaan esittää alkoiden ja niiden jäsenyysasteiden parien joukkona yhtälöllä

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (3)$$

Sumeille joukoille voidaan määritellä teräviä joukkoja vastaavia operaatioita, kuten unioni, leikkaus ja komplementti. Sumeiden joukkojen  $A, B \subset X$  unioni määritellään yhtälöllä

4, leikkaus yhtälöllä 5 ja komplementti yhtälöllä 6.

$$\mu_{A \cup B} = \max(\mu_A(x), \mu(x)_B), x \in X \quad (4)$$

$$\mu_{A \cap B} = \min(\mu_A(x), \mu(x)_B), x \in X \quad (5)$$

$$\mu_{\bar{A}} = 1 - \mu_A(x), x \in X \quad (6)$$

### 2.3 T-normit ja t-konormit

Edellä esitetyt minimi- ja maksimi-operaatio ovat vain yksi mahdollinen tapa määritellä sumeiden joukkojen leikkaus ja unioni. Ne ovat erikoistapauksia yleisemmistä operaatioista joita kutsutaan t-normeiksi ja t-konormeiksi. T-konormeja kutsutaan myös joskus s-normeiksi. T-normi tai t-konormi voi olla mikä tahansa funktio, jotka täyttää yhtälöissä 7, 8 ja 9 esitetyt ehdot. Tämän lisäksi funktio on t-normi, jos se täyttää yhtälön 10 ehdon. Funktio on t-konormi, jos se täyttää yhtälön ehdon 11. Lisäksi T-normilla on niin sanottu Arkhimedeen ominaisuus, jos se täyttää yhtälön 12 ehdon. [7]

$$L = [0, 1], \mathbf{T} : L \times L \rightarrow L, x, y, x', y' \in L$$

kommutatiivisuus:

$$\mathbf{T}(x, y) = \mathbf{T}(y, x) \quad (7)$$

monotonisuus:

$$x \leq x', y \leq y' \Rightarrow \mathbf{T}(x, y) \leq \mathbf{T}(x', y') \quad (8)$$

assosiatiivisuus:

$$\mathbf{T}(\mathbf{T}(x, y), z) = \mathbf{T}(x, \mathbf{T}(y, z)) \quad (9)$$

T-normi:

$$\mathbf{T}(x, 1) = x \quad (10)$$

T-konormi:

$$\mathbf{T}(x, 0) = x \quad (11)$$

Arkhimedeen ominaisuus:

$$\forall x \in ]0, 1[: \mathbf{T}(x, x) < x \quad (12)$$

Yleisesti käytettyjä t-normeja ja niitä vastaavia t-konormeja[7]:

Minimi ja maksimi:

$$\mathbf{T}_m(x, y) = \min(x, y), \quad \mathbf{S}_m(x, y) = \max(x, y)$$

Algebraalinen tulo ja summa:

$$\mathbf{T}_a(x, y) = x \cdot y, \quad \mathbf{S}_a(x, y) = x + y - x \cdot y$$

Einsteinin tulo ja summa:

$$\mathbf{T}_e(x, y) = \frac{x \cdot y}{1 + (1-x) \cdot (1-y)}, \quad \mathbf{S}_e(x, y) = \frac{x+y}{1+x \cdot y}$$

Rajoitettu erotus ja summa:

$$\mathbf{T}_r(x, y) = \max(0, x + y - 1), \quad \mathbf{S}_e(x, y) = \min(1, x + y)$$

Hamacherin tulo ja summa:

$$\mathbf{T}_h(x, y) = \frac{x \cdot y}{x + y - x \cdot y}, \quad \mathbf{S}_h(x, y) = \frac{x + y - 2x \cdot y}{1 - x \cdot y}$$

Hamacherin yleistetty tulo ja summa:

$$\mathbf{T}_\gamma(x, y) = \frac{x \cdot y}{\gamma + (1-\gamma)(x+y-x \cdot \gamma)}, \quad \mathbf{S}_\gamma(x, y) = \frac{x+y-x \cdot y - (1-\gamma)x \cdot y}{\gamma + (1-\gamma)(1-x \cdot y)}, \quad \gamma \in [0, \infty)$$



## 2.4 Sumeista relaatioista

Sumeat relaatiot ovat funktioita vastaavassa asemassa sumeassa logiikassa. Sumea relaatio on terävän relaation erikoistapaus, joka voi saada arvokseen minkä tahansa arvon suljetulta yksikköväliltä  $[0, 1]$ . Sumea binäärinen relaatio on yleinen  $n$ -paikkaisen sumean relaation erikoistapaus. Esimerkki binäärisestä sumeasta relaatiosta on esimerkiksi  $xRy =$  ” $x$  likimain  $y$ ”. Olkoot  $X = \{1, 2, 3\}$  ja  $Y = \{2, 3, 4\}$ . Relaatio  $xRy$  voidaan esittää relatiomatriisina:

$$\begin{pmatrix} 0,5 & 0,25 & 0,0 \\ 1,0 & 0,5 & 0,25 \\ 0,5 & 1,0 & 0,5 \end{pmatrix}$$

” $x$  likimain  $y$ ” voidaan esittää myös monella muulla tavalla tilanteesta riippuen, sillä sumean relaation arvot määritellään yleensä sen käyttötarkoituksen perusteella. Formaalisimmin ilmaistuna sumea  $n$ -paikkainen relaatio  $R$  on karteesisen tulon  $X_1 \times X_2 \times \dots \times X_n$  sumea osajoukko  $R$ , jonka jäsenyysfunktio on

$$\mu_R(x_1, x_2, \dots, x_n), x_i \in X, i = 1, 2, \dots, n,$$

missä  $\mu_R$  on funktio  $\mu_R : X_1 \times X_2 \times \dots \times X_n \rightarrow [0, 1]$  [7].

Sumeat relaatiot voidaan jakaa niiden ominaisuuksien perusteella erilaisiin ryhmiin. Seuraavassa muutamia yleisiä ominaisuuksia [7].

$R$  on sumea relaatio joukossa  $X$ ,  $x, y, z \in X$ .  $R$  on refleksiivinen jos se täyttää yhtälön 13 ehdon.  $R$  on antirefleksiivinen, jos se ei täytä ehtoa.

$$\mu_R(x, x) = 1 \tag{13}$$

$R$  on symmetrinen, jos se täyttää yhtälön 14 ehdon. Jos  $R$  täyttää ehdon vain joillain pareilla  $(x, y)$ , se on antisymmetrinen. Jos  $R$  ei täytä ehtoa millään parilla  $(x, y)$ , se on asymmetrinen.

$$\mu_R(x,y) = \mu_R(y,x) \quad (14)$$

R on transitiivinen, jos se täyttää yhtälön 15 ehdon.

$$\mu_R(x,z) \geq \max(\mu_R(x,y), \mu_R(y,z)) \quad (15)$$

Relaatiota, joka on refleksiivinen, symmetrinen ja transitiivinen, kutsutaan similaarisuus relaatioksi. Similaarisuusrelaation komplementti on dissimilaarisuusrelaatio. Jos relaatio on refleksiivinen ja symmetrinen, sitä kutsutaan läheisyysrelaatioksi.

## 3 SUMEAA LOGIIKKA HYÖDYNTÄVIÄ SUOSITUSJÄRJESTELMIÄ

Tässä luvussa esitellään sumeaa logiikkaa hyödyntäviä suositusjärjestelmiä. Esittelyssä painotetaan järjestelmien sumean logiikan osuutta jättäen muu järjestelmä vähemmälle huomiolle.

### 3.1 Film-Conseil

Film-Conseil [8] on sumeisiin relaatioihin perustuva suositusjärjestelmä elokuville. Käytetyt sumeat relaatiot ovat: käyttäjien mieltymys elokuvaan, käyttäjien samankaltaisuus, elokuvien samanlaisuus, käyttäjien oletettu mieltymys elokuvaan ja käyttäjän pätevyys arvioida elokuvaa.

Järjestelmä yhdistää yhteisösuodatukseen perustuvan menetelmän asioiden ominaisuuksiin perustuvaan suositusmenetelmään. Eri menetelmien keskinäistä painoa lopullisessa suosituksessa voidaan muuttaa, riippuen esimerkiksi siitä, kuinka monelle elokuvalla käyttäjä on antanut arvosanan. Jos käyttäjä on antanut vain muutaman arvosanan, voidaan suosituksissa painottaa asioiden ominaisuuksiin perustuvaa suositusta. Toisaalta, jos käyttäjä on arvostellut monta elokuvaa, voidaan asioiden ominaisuuksiin perustuvan suosituksen antamien suositusten vaikutusta lopulliseen suositukseen vähentää.

Molemmissa järjestelmän käyttämissä menetelmissä naapurien määrää voidaan rajoittaa johonkin kokonaislukuun. Naapurit ovat käyttäjää tai elokuvaa muistuttavien käyttäjien tai elokuvien joukko. Rajoituksen seurauksena suositukset perustuvat vain siis aktiivista käyttäjää lähimpiin käyttäjiin tai elokuvaa lähimpiin elokuviin. Naapuruston rajoittaminen sopivasti sekä parantaa suositusten tarkkuutta että vähentää laskentatehon tarvetta. Tässä työssä kehitetyssä menetelmässä käytetään hieman vastaavaa toimintatapaa, joka eroaa kuitenkin varsin paljon toteutuksen osalta.

Järjestelmässä on kiinnitetty erityistä huomiota suositusten selitettävyyteen. Se pystyy siis antamaan syitä, miksi se antoi elokuvalla juuri tämän suosituksen. Tämän ominaisuuden on havaittu lisäävän käyttäjien luottamusta järjestelmään [1]. Samalla se myös kannustaa käyttäjää arvostelemaan useampia elokuvia, jolloin käyttäjille annettujen suositusten tasoa voidaan parantaa. Film-Conseil oppii myös käyttäjän antaman palautteen perusteella. Jos käyttäjä ei ole samaa mieltä järjestelmän suositukselle antaman perustelun kanssa,

hän voi antaa esimerkin elokuvasta, jonka tapauksessa annettu syy suositukselle ei pidä paikkaansa.

Lopullisen suosituksen järjestelmä antaa elokuvan suositeltavuutta kuvaavana lukuna, jonka avulla elokuvat voidaan järjestää suositeltavuusjärjestykseen. Järjestyksessä otetaan huomioon myös annetun suosituksen luotettavuus.

Suosituksessa otetaan huomioon vain aktiivisen käyttäjän kanssa samanlaisista elokuvista pitävät käyttäjät, joten siinä ei hyödynnetä käyttäjiä, joilla on päinvastaiset mieltymykset. Tämä tarkoittaa sitä, että järjestelmä vaatii useampia käyttäjiä antaakseen hyviä suosituksia kuin järjestelmä, joka osaa hyödyntää muitakin korrelaatioita. Ilmeisesti muunlaisten korrelaatioiden lisääminen järjestelmään ei olisi kuitenkaan kovin vaativaa.

### **3.2 Tapahtumien suosittelu sumeilla relaatioilla**

Artikkelissa [9] esitellään edellisessä luvussa esitettyyn menetelmään perustuva suositusjärjestelmä, joka pystyy suositteluun aikaan sidottuja tapahtumia. Menetelmä yrittää myös korjata aikaisemmassa menetelmässä havaitut kaksi ongelmaa: yhteisösuodatuksen ja asioiden ominaisuuksiin perustuvan suosituksen painotuksen arvon määrittäminen on hankalaa, ja yhteisösuodatus ei pysty suositteluun asioita, joita ei ole vielä arvosteltu. Jälkimmäisen ongelman takia aikaisempi menetelmä ei sovellu hyvin aikaan sidottujen tapahtumien suositteluun. Parannetun menetelmän pääperiaate voidaan kiteyttää yhteen virkkeeseen: ”suositellaan tulevaa tapahtumaa, jos se on samanlainen niiden menneiden tapahtumien kanssa, joista aktiivisen käyttäjän kanssa samanlaiset käyttäjät ovat pitäneet.”

Menetelmä käyttää kolmea sumeaa relaatiota: käyttäjien samanlaisuutta, asioiden samanlaisuutta ja käyttäjien mieltymystä asiaan. Asioiden samanlaisuutta ilmaiseva relaatio on sekä symmetrinen että refleksiivinen. Käyttäjien samanlaisuutta ilmaisevan relaation ei sen sijaan tarvitse välttämättä olla symmetrinen. Menetelmä ei käytä aikaisemman menetelmän relaatiota, joka ilmaisee käyttäjän pätevyyden arvostella kyseinen asia. Tämä ominaisuus on kuitenkin helposti lisättävissä järjestelmään. Pätevyys määrittää rajat käyttäjän positiivisen ja negatiivisen mieltymyksien vaikutuksille.

Sisäisesti menetelmä esittää käyttäjän positiivisen ja negatiivisen mieltymyksen asiaan erillisinä jäsenyysasteina. Lopullisen suosituksen se kuitenkin antaa yhtenä terävänä lukuna. Eri positiivisen ja negatiivisen arvojen pareille annetaan selitykset: ristiriidassa se-

kä positiivinen että negatiivinen mieltymys saa arvokseen yksi, tietämättömydessä molempien mieltymysten arvo on nolla ja neutraaliutta ilmaistaan molempien mieltymysten arvolla 0,5. Tämä määritelmä ei tunnu pitävän yhtä aikaisemman määritelmän kanssa, jossa ristiriita esitettiin molempien arvojen saadessa arvon yksi. Missä vaiheessa neutraalius muuttuu ristiriidaksi? Positiivisen ja negatiivisen mieltymyksen yhtäsuuren arvon tulisi aina ilmaista ristiriitaa. Tämä lähestymistapa sopisi hyvin esitettyyn tietämyksen relaatioon, joka määrittelee tietämyksen positiivisen ja negatiivisen mieltymyksen rajatuksi summaksi.

Käyttäjien samanlaisuudessa otetaan huomioon tapaukset, joissa käyttäjillä on samantyyppiset mieltymykset, mutta eivät ole arvostelleet yhtään samaa asiaa. Käyttäjät ovat siis arvostelleet korkealle asioita, jotka ovat keskenään samantyyppisiä olematta silti samoja. Tämä tehdään hyödyntämällä asioiden samankaltaisuutta ilmaisevaa relaatiota. Pelkkään yhteisösuodatukseen perustuvat menetelmät eivät tähän pysty, sillä suositeltavien asioiden ominaisuuksien samanlaisuutta ei oteta huomioon.

Menetelmän toimivuutta ei testattu kokeellisesti, joten sitä, miten paljon se parantaa aikaisemman järjestelmän toimivuutta, ei pystytä arvioimaan.

### 3.3 Hybridi elokuvasuosittin

Artikkelissa [10] esitetty järjestelmä on yhteisösuodatusmenetelmä, joka perustuu käyttäjistä luotuihin malleihin. Mallien käyttö nopeuttaa käyttäjien naapurustojen etsimistä. Naapuruston antamien arvosanojen perusteella käyttäjälle lasketaan ennustetut arvosanat. Käyttäjien samankaltaisuuden arvioinnissa käytetään sumeaa konkordanssi/diskonkordanssi-periaatetta.

Sumean konkordanssi/diskonkordanssi-periaatteen mukaan asiat ovat samantyyppisiä, jos suurin osa syistä puoltaa samantyyppisyyttä, eikä ole olemassa vahvaa pientä vastustusta. Konkordanssikoalition, eli puoltavien syiden joukon, täytyy siis olla vahva ja diskonkordanssikoalitiolla ei saa olla vetoa. Menetelmä vähentää tilanteita, joissa puoltavat ja vastustavat syyt kompensoivat toisiaan. Pienet, mutta ratkaisevat erot voivat siis estää asioiden määrittelyä samantyyppisiksi. Sumeaa logiikkaa käytetään puoltamisen ja vastustuksen vahvuuden määrittelyssä. Tämä mahdollistaa käyttäjien samankaltaisuuden hienojakoisemman määrittelyn.

Testidatana käytetään MovieLens-dattaa, jonka käyttäjät ovat arvostelleet vähintään 20

elokuvaa. Testauksessa käytetään vain käyttäjiä, jotka ovat arvostelleet vähintään 60 elokuvaa. Tämä tuntuu varsin korkealta määrältä, joten testi ei anna hyvää kuvaa siitä, kuinka hyvin järjestelmä suoriutuu vähän elokuvia arvostelleiden käyttäjien kohdalla. Järjestelmä näyttäisi parantavan suosittelutarkkuutta ja samalla parantaa suositusten kattavuutta, vertailussa käytettyihin menetelmiin nähden. Arviot annetaan terävinä arvoina samalta arvosana-asteikolta kuin käyttäjien antamat arvosanat.

### 3.4 FARAMS

FARAMS (Fuzzy Association Rules And Multiple Level Similarity) on suositusjärjestelmä, joka perustuu sumeiden assosiaatiosääntöjen louhimiseen datasta [11]. Säännöt ovat muotoa: ”jos käyttäjä pitää asiasta A, hän pitää asiasta B”. Sumeaa logiikkaa hyödynnetään suositusjoukkojen reunojen pehmentämiseen. Käyttäjien antamat arvosanat jaetaan kolmeen sumeaan joukkoon: ”dislike”, ”neutral”, ”like”. Arvosanalla voi olla nollasta poikkeava jäsenyysaste useassa joukossa yhtä aikaa, mikä mahdollistaa ristiriitaisuuden ja tietämättömyyden ilmaisemisen.

Tapauksissa, joissa käyttäjä on antanut arvosanan vain hyvin pienelle osalle suositeltavista asioista, FARAMS hyödyntää asioiden kategorioita suositusten antamisessa. Kategoriaan, johon kuuluu paljon käyttäjän pitämiä elokuvia, kuuluvaa asiaa suositellaan vaikka suoraa yhteyttä suositeltavan asian ja käyttäjän arvioimien asioiden välillä ei olisikaan. Tämä kasvattaa suositusjärjestelmän suosittelmien asioiden määrää. Suositukset FARAMS antaa suosituksen hyvyttä kuvaavana terävänä lukuna.

### 3.5 Looginen kehys sumeaan yhteisösuodatukseen

Artikkelin [12] menetelmä perustuu Łukasiewiczin moniarvologiikkaan, jossa totuusarvo voi olla mikä tahansa rationaaliluku väliltä  $[0,1]$ . Tämä on Łukasiewiczin kolmiarvoisen moniarvologiikan laajennus, joka puolestaan on klassisen kaksiarvoisen logiikan laajennus. Łukasiewiczin moniarvologiikan käsitys toden ja epätoden väliin jäävistä totuusarvoista poikkeaa sumean logiikan käsityksestä. Sumeassa logiikassa totuusarvot kuvaavat järjestelmän epämääräisyyttä, kun puolestaan Łukasiewiczin moniarvologiikka käyttää todesta tai epätodesta poikkeavia totuusarvoja epävarmuuden ilmaisemiseen. Epävarmuus tarkoittaa asian tilaa, jossa totuusarvoa ei vielä tiedetä, mutta todellinen totuusarvo voi selvitä myöhemmin. Asian tilalle on siis olemassa yksiselitteinen totuusarvo.

Järjestelmä laskee käyttäjän ennustetut mieltymykset asioille. Se antaa myös ennustetuille mieltymyksille luottamusta kuvaavan arvon. Järjestelmä mahdollistaa siis epävarmuuden ilmaisemisen, mutta sillä ei voida ilmaista ristiriitaisia tilanteita, joissa annetun datan perusteella mieltymyksen ennuste osoittaa kahteen tai useampaan toisistaan poikkeavaan arvoon. Sekä ennustettua mieltymystä että ennusteen luottamusta ilmaistaan arvona väliltä  $[0,1]$ . Mieltymyksen ennusteet väliltä  $[0, 1/2)$  tulkitaan negatiiviseksi mieltymykseksi, eli käyttäjä ei pitäisi kyseisestä asiasta. Puolestaan ennusteet väliltä  $(1/2,1]$  käsitetään positiiviseksi mieltymykseksi, eli käyttäjä pitäisi asiasta. Arvo  $1/2$  käsitetään neutraalina. Ennusteen luotettavuudelle järjestelmässä on olemassa parametrina annettava raja-arvo, jonka ylitettyään ennusteen oletetaan olevan luotettava. Kun käyttäjä antaa mieltymyksen uudelle asialle, sen ennustettu mieltymys asetetaan annetuksi arvoksi ja ennusteen luotettavuus asetetaan arvoon yksi. Käyttäjän antamiin mieltymyksiin luotetaan siis täysin.

Käyttäjien välille lasketaan käyttäjien mieltymysten samankaltaisuutta kuvaava arvo, jota käytetään painona mieltymysten ennusteiden laskemisessa. Käyttäjien katsotaan olevan samanlaisia, jos jokainen molempien käyttäjien korkean luottamuksen saanut mieltymys on mieltymyksen arvosta lähellä toistaan. Järjestelmän antamat suositukset muuttuvat ajan mukana. Uusien arvosanojen antaminen aiheuttaa uusien ennustettujen mieltymysten ja ennusteiden luotettavuuden laskemisen. Esitettyssä järjestelmässä jo annettujen arvosanojen muuttaminen ei ole sallittua, mutta sitä voidaan muuttaa hyväksymään myös tällaiset tilanteet. Ajan hetkellä nolla vain käyttäjän suoraan antamiin mieltymyksiin luotetaan, eli niiden luottamus on 1. Asioille, joille käyttäjä ei ole antanut mieltymystä, annetaan ennustetuksi mieltymyksen arvoksi  $1/2$  ja ennusteen luotettavuudeksi 0.

Esitetty yhteisösuodatukseen perustuva järjestelmä voidaan yhdistää asioiden ominaisuuksiin perustuvaan suositteluun. Tämä voidaan tehdä kahdella tavalla: laskemalla erikseen yhteisösuodatuksella ja ominaisuuksiin perustuvalla suosituksella saadut ennusteet, ja lopuksi yhdistämällä ne, tai yhdistämällä asioiden ominaisuuksiin perustuva suositus kiinteämmin osaksi yhteisösuodatusta. Erikseen eri menetelmillä lasketut suositukset voidaan yhdistää painottamalla saatuja ennusteita eri painoarvoilla. Menetelmät tiukemmin yhdistämällä ominaisuuksiin perustuvaa suositusta käytetään vahvistamaan yhteisösuodatuksen antamia suosituksia. Tässä työssä esitettyssä menetelmässä käytetään samanlaista tekniikkaa, sillä elokuvien tyylilajeja käytetään vahvistamaan yhteisösuodatuksella laskettuja suosituksia.

Järjestelmälle ei anneta minkäänlaisia kokeellisia tuloksia, joten sen mahdollinen suorituskyky jää muihin menetelmiin nähden arvoitukseksi.

### 3.6 Sumean logiikan metodeja suositukseseen

Viitteessä [13] esitellään asioiden ominaisuuksiin ja yksittäisen käyttäjän mieltymyksiin perustuvia suositusmenetelmiä. Käyttäjien mieltymyksiä ei siis verrata toisiinsa, joten menetelmät eivät ole yhteisösuodatusmenetelmiä. Suosituksia varten asioiden ominaisuuksista tarvitaan kuvaus. Elokuvien tapauksessa tällaisia ominaisuuksia tai piirteitä voi olla esimerkiksi elokuvan nimi, valmistumisvuosi, ohjaaja tai elokuvassa esiintyvät näyttelijät.

Käyttäjä antaa suoraan tietoa, millaisista asioita hän pitää. Käyttäjä tekee eräänlaisia hakuja asioiden ominaisuuksia sisältävään tietokantaan. Suositeltavat asiat esitetään väittämien avulla. Jokaisen väittämän tulos on luku väliltä  $[0,1]$  riippuen miten hyvin asia vastaa esitettyä väittämää. Elokuvien tapauksessa väittäjä voi olla esimerkiksi ”tämä elokuva on komedia”, ”tämän elokuva tähti on Robert DeNiro” tai ”elokuva on valmistunut vuonna 1998”. Joillain väittämällä tulos voi olla vain joko 0 tai 1, kun taas toisilla tulos voi olla mikä tahansa arvo kyseiseltä väliltä. Asialla voi olla useita samaan piirteeseen kuuluvia väittämiä. Esimerkiksi elokuvalla voi olla useita tähtiä.

Käyttäjä voi esittää mieltymyksensä erityisesti sitä varten kehitetyn Hi-Ret-nimisen kielen avulla. Kielen tulkinnessa käytetään OWA-operaattoria (Ordered Weight Average). OWA-operaattorin määrittää  $n$ -ulotteinen painovektori, jonka painot ovat yksikköväliltä ja painojen summa on yksi. Operaattorille annetaan argumenttina suuruusjärjestykseen järjestetty vektori. Operaattorin tulos lasketaan kertomalla argumenttivektorin ja painovektorin vastaavat elementit keskenään ja summaamalla näin saadut arvot. Painovektorista riippuen OWA-operaattorilla voidaan esittää esimerkiksi sellaisia operaatiota kuin minimi, maksi ja keskiarvo.

Painot OWA-operaattorille saadaan kielellisten kvanttorien avulla. Kielelliset kvanttorit ilmaisevat mieltymyksen vahvuutta. Esimerkkejä käytetyistä kielellisistä kvanttoreista ovat ”ainakin yksi”, ”muutama”, ”kaikki”. Väittämien ja kielellisten kvanttorien avulla käyttäjä voi esittää mieltymyksiä asioista.

Käyttäjien profiilit koostuvat edellä kuvatun kielen avulla esitetystä positiivisista ja negatiivisista mieltymyksistä. Asian suositeltavuuden aste lasketaan ottamalla maksimi käyttäjän profiilin sisältämien mieltymysten vastaavuuksista laskettavan asian kanssa. Käyttäjä voi antaa myös jokaiselle mieltymykselle painon sille, kuinka paljon se voi vaikuttaa laskennan lopputulokseen. Paino annetaan yksikköväliltä ja se määrää ylärajan mieltymyksen vastaavuuden asteelle. Positiivisten ja negatiivisten mieltymysten aste voidaan yhdis-



tää lopulliseksi suositusasteeksi esimerkiksi vähentämällä positiivisesta asteesta negatiivinen aste. Negatiiviset tulokset asetetaan nolaksi. Suositeltavuuden aste voidaan laskea myös miniminä negatiivisen asteen negatiivisesta ja positiivisesta asteesta. Tällöin negatiivinen mieltymyksen aste asettaa ylärajan, kuinka paljon positiivinen mieltymyksen aste voi vaikuttaa lopputulokseen.

Suosituksia voidaan tehdä myös ilman käyttäjän suoraan antamia mieltymyksiä, jos saatavilla on käyttäjän asioille antamia arvosanoja. Tällöin asiaa suositellaan kahdessa tapauksessa: asia on samanlainen toisen asian kanssa, josta käyttäjä on pitänyt, tai asia on vastaava ainakin usean asian kanssa, joista käyttäjä on jossain määrin pitänyt. Asioiden samankaltaisuuden määrittelemiseen ei anneta mitään menetelmää, sillä se on hyvin tilannekohtaista. Toisessa tapauksessa käytetty ilmaus ”ainakin usea” on esimerkki kielellisestä määrästä, jonka tarkoituksena on pehmentää säännön rajoja. Mitä laajempi mieltymyksen vaihtelu sallitaan, sitä suurempi määrä säännön täyttäviä asioita on oltava, jotta asiaa suositellaan. Asiaa voidaan suositella myös, jos käyttäjä on antanut arvosanan usealle suositeltavaa asiaa lähellä olevalle asialle. Elokuviin tapauksessa tämä tarkoittaa sitä, että käyttäjä katsoo tietynlaisia elokuvia riippumatta miten paljon hän niistä varsinaisesti pitää.

Suosituksia voidaan antaa asiantuntijoiden päättämien prototyyppien perusteella. Elokuvia suositeltaessa tällaisia prototyyppisiä voisi olla esimerkiksi film noir elokuvat ja eepilliset elokuvat. Suositus perustuu siihen, kuinka paljon käyttäjä pitää prototyypistä, ja mikä on suositeltavan asian kuuluvuuden aste prototyyppiin. Kuten edellisessäkin lähestymistavassa, asiaa voidaan silti suositella, jos se kuuluu prototyyppiin, johon kuuluvia asioita käyttäjä on arvostellut useita. Tämä käsitetään käyttäjän mieltymyksenä kyseiseen prototyyppiin.

Esitettyjä menetelmiä ei testata kokeellisesti, joten niiden suorituskyky muihin menetelmiin nähden jää tutkimatta. Menetelmät kehoitetaan yhdistämään yhteisösuodatusmenetelmään parhaan lopputuloksen aikaansaamiseksi.

### **3.7 Karheat joukot ja sumea klusterointi**

Haasteena yhteisösuodatuksen perustuvissa suositusmenetelmissä on, että suurin osa mahdollisista käyttäjien asioille antamista arvosanoista on tyhjiä. Viitteessä [14] esitetty menetelmä käyttää karheita joukkoja käyttäjä-asiamatriisin täydentämiseen. Täydennästä matriisista etsitään käyttäjien naapurustot sumealla klusteroinnilla. Pearsonin kor-

relaatiota käytetään käyttäjien samanlaisuuden laskemiseen. Lopullinen arvosanan arvio lasketaan  $k:n$  lähimmän naapurin painotetulla keskiarvolla. Lopullinen suositus on siis aktiivisen käyttäjän oletettu arvosana asialle.

### **3.8 Komponenttikohtainen vakaa lineaarinen sumea klusterointi yhteisösuodatukseen**

Artikkelin [15] menetelmä on yhteisösuodatusmenetelmä, jossa etsitään sumean klusteroinnin ja pääkomponenttianalyysin avulla datasta paikallisia lineaarisia malleja. Mallien avulla voidaan laskea suositukset ennustamalla puuttuvat arvosanat. Järjestelmässä pyritään hyödyntämään myös virheellisiä arvosanoja antaneita käyttäjiä.

Pääkomponenttianalyysissä datasta etsitään komponentit, joiden avulla koko datan piirteet voidaan kuvata riittävällä tarkkuudella. Datasta muodostetaan siis yksinkertaistettu yleistyys, joka kuvaa annettua dataa pääpiirteissään, mutta joka samalla hävittää jonkin verran datan yksityiskohdista. Mallien käyttö pienentää suositusten laskemiseen tarvittavan muistin määrää, sillä laskennassa ei tarvita kaikkia käyttäjien asioille antamia arvosanoja.

Klusteroinnissa datasta pyritään löytämään ryppäitä, joiden datapisteet ovat keskenään samanlaisia. Sumeassa klusteroinnissa ryppäiden väliset rajat eivät ole teräviä, vaan datapisteellä voi olla nollasta poikkeava jäsenyysaste useassa ryppäessä yhtä aikaa. Datassa olevat häiriöt vaikeuttavat hyvien lineaaristen mallien löytämistä datasta. Suosittelemuksen tapauksessa näin voi käydä, jos käyttäjä on antanut, vahingossa tai tahallaan, virheellisiä arvosanoja. Yksinkertaisimmillaan ongelma voidaan korjata jättämällä käyttäjä pois, kun suosituksia lasketaan muille käyttäjille. Tällöin kuitenkin menetetään hyödyllistä dataa mahdollisesti vain muutaman virheellisen arvosanan takia. Esitetyn klusterointimenetelmän avulla tällaisesta näytteiden sisäisistä virheistä kärsivästä datasta voidaan muodostaa hyviä lineaarisia malleja.

Järjestelmän testaukseen käytettiin MovieLens-dataa, josta karsittiin pois kaikki elokuvat, jotka oli arvosteltu alle neljä kertaa. Menetelmä antaa suositukset terävinä lukuina arvosana-asteikolla. Testien perusteella menetelmä antaa hieman parempia suosituksia kuin vertailun muut menetelmät.

### 3.9 Agenttipohjainen sumea yhteisösuodatin

Järjestelmä [16] käyttää hajautettua laskentaa suositusten laskemisen nopeuttamiseksi välttämällä keskitettyjen menetelmien mahdollisia pullonkauloja. Käyttäjien suositusten laskenta on jaettu itsenäisiin osiin, joita kutsutaan agenteiksi. Agentit pystyvät kommunikoimaan suoraan vain välittömien naapuriensa kanssa. Agenteilla on kuitenkin epäsuorasti yhteys myös muihin agentteihin naapuriensa välityksellä. Agentti laskee käyttäjälle annettavat suositukset naapurien antamista suosituksista. Keskitetyistä suositusmenetelmistä poiketen agenttipohjaisella järjestelmällä on vain osittainen näkymä kaikkien käyttäjien antamiin arvosanoihin, ja se joutuu siis tekemään suositukset vähemmällä määrällä dataa. Järjestelmä perustuu puhtaasti yhteisösuodatukseen, eli siinä ei hyödynnetä käyttäjistä tai suositeltavista asioista kerättyä dataa.

Menetelmässä sumeaa logiikkaa käytetään numeeristen arvosanojen epämääräisyyden käsittelyyn. Arvosana-asteikko jaetaan sumeisiin joukkoihin, jolloin lopullinen suositus saadaan aggregoimalla sumeita joukkoja. Aggregoinnissa otetaan huomioon, miten kauan sitten suositus on annettu. Agentti voi siis unohtaa vanhat suositukset ja suosia uusia, tai päin vastoin. Tällä tavoin pyritään jäljittelemään ihmisen päätöksentekoa.

Agentti muuttaa myös käyttäytymistään riippuen siitä, miten tarkkoja sen käyttäjälle tekemät suositukset ovat olleet. Suositusten tarkkuutta voidaan mitata, kun käyttäjä antaa arvosanan uudelle asialle, jolle on olemassa agentin laskema suositus. Jos suositus vastasi todellista arvosanaa, agentti muuttaa käyttäytymistään suosimalla vanhoja suosituksia enemmän kuin uusia. Jos suositus ei vastannut todellista arvosanaa, agentti muuttaa käyttäytymistään suosimalla uusia naapurin antamia suosituksia vanhoja enemmän.

Agenttipohjaisissa suositusjärjestelmissä käyttäjällä on suurempi vaara joutua manipuloinnin kohteeksi. Useat agentit voivat yhteistyössä antaa käyttäjälle virheellisen kuvan suositeltavan asian hyvydestä. Testien perusteella esitetty sumeaa logiikkaa hyödyntävä menetelmä on ainakin yhtä vastustuskykyinen manipulointia vastaan kuin paras ei sumeaa logiikkaa hyödyntävä menetelmä. [17]

Kokeissa käytetään MovieLens-dataa. Kokeet keskittyy käyttäjiin, jotka ovat arvostelleet vähintään 70 elokuvaa, ja elokuviin, jotka on arvostelu vähintään 35 kertaa. Testi ei anna hyvää kuvaa menetelmän toiminnasta vähän data kanssa. Menetelmä näyttäisi kokeiden mukaan antavan parempia suosituksia kuin aiemmat hajautettuun yhteisösuodatukseen perustuvat menetelmät.

### 3.10 Kontekstitietoinen musiikkisuositusjärjestelmä

Artikkelissa [18] kuvattu järjestelmä suosittelee käyttäjälle kontekstiin sopivaa musiikkia. Se perustuu musiikista analysoituihin ominaisuuksiin, käyttäjän profiliin ja ympäristöstä kerättyihin tietoihin. Ympäristöstä kerätyt tiedot ovat lämpötila, kosteus, melu, valoisuus, sää, sääennuste, vuodenaika ja aika. Käyttäjistä tiedossa on ikä ja sukupuoli.

Järjestelmässä käytetään Bayesilaisia verkkoja kontekstin päättelyyn järjestelmään syötetystä datasta. Bayesilaisen verkon avulla voidaan tehdä tilastollisia päätelmiä epävarmoissa tilanteissa. Verkko tarvitsee diskreettejä sisääntuloja, joten sillä ei voida tehdä päätelmiä monista lähteistä tulevasta datasta, joka voi olla diskreettiä tai jatkuvaa. Data joudutaan muuttamaan Bayesilaisen verkolle soveltuvaan muotoon, mistä voi seurata päättelyn kannalta oleellisen informaation katoamista.

Sumeaa logiikkaa voidaan käyttää useanlaisista datalähteistä tulevan data muuttamiseen bayesilaiselle verkolle soveltuvaan muotoon siten, että päättelyn kannalta oleellista informaatiota ei pääse juuri katoamaan. Diskreetti ja jatkuva data voidaan muuttaa vektoreiksi, jotka sisältävät sisääntulon mahdollisia tiloja vastaavat jäsenyysasteet. Esimerkiksi lämpötila jakautuu tiloihin kylmä, sopiva ja kuuma. Tiedetyn lämpötilan perusteella voidaan ennalta määrätyn jäsenyysfunktion perusteella datalle laskea jäsenyysaste jokaisessa tilassa. Näin lasketut vektorit annetaan syötteenä bayesilaiselle verkolle, jota on laajennettu toimimaan jäsenyysasteita kuvaavien vektorien kanssa.

Bayesilaisen verkon päättelemän kontekstin ja käyttäjän kontekstia vastaavien mieltymysten perusteella lasketaan musiikkikappaleille suositusta kuvaava arvo. Järjestelmä suosittelee joukon korkeimman arvon saaneita kappaleita käyttäjälle. Testien mukaan sumeaan bayesilaisiin verkkoihin perustuva järjestelmä antaa parempia suosituksia kuin tavalliseen bayesilaiseen verkkoon perustuva suositusjärjestelmä.

### 3.11 Sumea suositusjärjestelmä kuluttajaelektronikalle

Artikkelissa [19] esitetyn järjestelmän tarkoituksena on helpottaa kuluttajaelektronikkatuotteiden valintaa. Suositusjärjestelmät, jotka hyödyntävät käyttäjän aikaisempaa ostokäyttäytymistä tuotteiden suosittelussa eivät sovellu esimerkiksi kännyköiden ja kannettavien tietokoneiden suositteluun kovinkaan hyvin, sillä näitä tuotteita ostetaan varsin harvoin. Järjestelmällä ei siis ole tarvittavaa dataa suositusten tekemiseen. Käyttäjän vaatimukset tuotteiden suhteen muuttuvat myös varsin nopeasti. Käyttäjän on puolestaan vai-

keaa valita monia ominaisuuksia sisältävien tuotteiden suuresta joukosta omiin tarkoituksiinsa paras vaihtoehto. Useimmilla käyttäjillä ei ole myöskään tarvittavaa teknistä tietämystä valita eri tuotteiden väliltä. Käyttäjät kuitenkin pystyy antamaan ei-teknisiä vaatimuksia laitteen käyttötarkoituksesta.

Asiantuntijat arvioivat miten hyvin mitkäkin tuotteet soveltuvat mihinkin käyttötarkoitukseen. Kun arviot yhdistetään käyttäjän antamiin vaatimuksiin tuotteesta, järjestelmä pystyy antamaan suosituksia käyttäjän tarkoituksiin sopivista tuotteista. Sekä käyttäjän vaatimukset että asiantuntijan arviot annetaan järjestelmälle sumeina lukuina. Mahdollisia lukuja on seitsemän, jotka ovat väliltä hyvin matalasta hyvin korkeaan. Kielellisinä kuvauksina annettujen lukujen tarkoituksena on helpottaa käyttäjää vaatimusten antamisessa.

Käyttäjän antamia vaatimuksia verrataan tuotteisiin mittaamalla sumeaa euklidista etäisyyttä sumeiden lukujen välillä. Laskettua etäisyyttä käyttäjän vaatimuksen ja tuotteen ominaisuuden välillä painotetaan käyttäjän vaatimukselle antaman tärkeyden mukaan. Ero vaatimuksen ja ominaisuuden välillä vaikuttaa etäisyyteen siis sitä enemmän mitä korkeampi käyttäjän vaatimuksen tärkeys on.

Tuotteen komponenttien ominaisuuksien vaikutusta komponenttiin painotetaan erilaisilla painoilla ominaisuuden mukaan. Esimerkiksi kannettavan tietokoneen suorittimen kellotaajuus vaikuttaa suorittimen toimintaan enemmän kuin välimuistin määrä. Painojen määrittämiseen käytetään asiantuntijoita.

Järjestelmä tarjoaa käyttäjälle interaktiivisen käyttöliittymän, jonka avulla hän voi kertoa haluamansa tuotteen vaatimukset. Käyttäjän kerrottua tuotteen vaatimukset, järjestelmä näyttää listan kymmenestä parhaiten vaatimukset täyttävästä tuotteesta. Jos käyttäjä ei löydä listasta mieluista tuotetta, hän voi muuttaa vaatimuksiaan ja pyytää uudet kymmenen suositusta.

Järjestelmän suorituskykyä arvioitiin seitsemän käyttäjän avulla, jotka käyttivät järjestelmää omiin vaatimuksiinsa soveltuvan kannettavan tietokoneen etsimiseen 138 kannettavan tietokoneen joukosta. Jokaisen käyttäjän saaman suosituslistan tarkkuutta arvioitiin asiantuntijoiden avulla. Asiantuntijat arvioivat, mitkä suositetuista tuotteista todellisuudessa soveltuivat käyttäjän vaatimuksiin ja mitkä ei suositelluista tuotteista soveltuivat käyttäjän vaatimuksiin. Tulosten mukaan järjestelmä antaa varsin hyviä suosituksia. Vääriä negatiivisia suosituksia järjestelmä antoi hieman vääriä positiivisia enemmän.

### 3.12 Sumeaan likimääräiseen päättelyyn perustuva älykäs verkkosivujen suositusjärjestelmä

Järjestelmä [20] suosittelee internetsivuston käyttäjälle häntä mahdollisesti kiinnostavia sivuston muita sivuja. Suosittelemus perustuu käyttäjän session aikana vieraillemiin sivuston URL-osoitteisiin. Järjestelmä ei siis varsinaisesti pidä kirjaa käyttäjistä, vaan jokaista sessiota pidetään itsenäisenä kokonaisuutena, johon ei liity varsinaista tietoa käyttäjästä. Sessio on yksittäisestä IP-osoitteesta tulleet käynnit URL-osoitteissa tietyn ajanjakson sisällä. Sessio on siis binäärinen vektori kaikista mahdollisista URL-osoitteista, mikä kertoo onko URL-osoitteessa käyty vai ei.

Profiilit ovat samanlaisista asioista kiinnostuneiden käyttäjien sessioista luotuja malleja. Aktiiviselle käyttäjän sessiolle merkitykselliset URL-osoitteet voidaan päätellä vertaamalla session ja profiilien samankaltaisuutta ja yhdistämällä session kanssa samanlaisten profiilien sisältämät URL-osoitteiden merkittävyydet. Tuloksena on vektori, joka sisältää jokaista URL-osoitetta vastaavat merkittävyydet aktiiviselle sessiolle.

Menetelmän toimintaan voidaan vaikuttaa usealla tavalla. Sessioiden ja profiilien samanlaisuuden laskemiseen voidaan käyttää useita ehdotettuja menetelmiä. Aktiivisen session ja profiilien samanlaisuutta kuvaavat jäsenyysasteet voidaan normalisoida jollain tavalla tai jättää ne kokonaan normalisoimatta. Profileista voidaan karsia pois vähämerkityksellisiä URL-osoitteita. Vähämerkityksellisten URL-osoitteiden karsinta voidaan suorittaa myös vasta lopullisista tuloksista. Myös t-normin ja t-konormin valinnalla voidaan vaikuttaa menetelmän suorituskykyyn ja toimintaan.

Järjestelmän suorituskykyä testattiin Missourin yliopiston tietotekniikan laitoksen verkkosivun käyttödatan avulla. Testeissä laskettiin suositusten tarkkuus ja kattavuus. Yleisesti voidaan sanoa, että menetelmän variaatiot, jotka paransivat suositustarkkuutta, huononsivat tulosten kattavuutta ja päinvastoin. Tuloksista voidaan kuitenkin esimerkiksi päätellä, että maksimin käyttäminen t-konormina toimii paremmin kuin rajatun summan käyttäminen. Myös terävän  $\alpha$ -leikkauksen käyttäminen profiiliin kuuluvien URL-osoitteiden karsimiseen tuotti hyviä tuloksia. Eri profiileihin kuuluvien sessioiden suositusten tarkkuudessa oli havaittavissa selkeitä eroja.

### 3.13 Opetusohjelmien sisällön suosittelu

Viitteessä [21] esitetty järjestelmä suosittelee opetusohjelman materiaaleja ohjelman käyttäjille heidän arvioidun oppimistasonsa perusteella. Suosittelu tehdään käyttäjän aikaisemmasta materiaalista antaman palautteen pohjalta.

Ei-sumea IRT (Item Response Theory) -menetelmä arvioi oppijan tasoa kaksiarvoisen palautteen pohjalta: käyttäjä osaa täysin materiaalin tai käyttäjä ei osaa täysin materiaalia. Sumeassa järjestelmässä käyttäjä antaa palautteen prosenttiluvun, joka kuvaa, kuinka suuren osan materiaalista oppija mielestään osaa. Annettu arvo sumennetaan kolmen sumean joukon jäsenyysteiksi: matala taso, keskitaso ja korkea taso. Joukkojen jäsenyysfunktio on määritelty etukäteen ja ovat muodoltaan kellokäyräisiä. Järjestelmä käyttää korkeimman jäsenyysasteen joukkoa oppimisen asteen arviointiin. Käyttäjän arvioitua osaamisen tasoa muutetaan riippuen korkeimman jäsenyysasteen saaneesta joukosta. Matalan tasoon kuuluvien oppimisen tason arviota lasketaan ja korkeaan tasoon kuuluvien oppimisen tason arviota nostetaan. Keskitasoon kuuluvien oppimisen tason arvio pysyy samana. Käyttäjä antaa palautteen myös mielipiteen opetusmateriaalin vaikeudesta, jota käytetään materiaalin yleisen vaikeuden määrittelyyn asiantuntijan antaman arvion ohella.

Tulosten perusteella esitetty järjestelmä pystyy varsin hyvin suositteluun oppijan tasoon nähden sopivan vaikeusasteen omaavaa oppimismateriaalia. Järjestelmän toimintaa ei kuitenkaan verrata vastaavaan ei sumeaan järjestelmään, joten sumean logiikan mahdollisesti tuomaa hyötyä ei pystytä eristämään.

## 4 SUOSITUSMENETELMÄN KUVAUS

Tässä luvussa esitellään kehitetyn suositusmenetelmän toimintaperiaate yksityiskohtaisesti. Toteutuksessa käytettyjen ratkaisujen ohella esitetään vaihtoehtoisia toteutustapoja, jotka saattaisivat toimia paremmin, mutta jotka jätetään mahdolliseen jatkokehitykseen.

### 4.1 Suositusten esittäminen arvosanojen sumeina jäsenyysasteina

Suositusjärjestelmät antavat suositukset yleensä arvosanoina samalta asteikolta, jolta käyttäjän on antamat arvosanatkin ovat. Tämä on käyttäjän kannalta varsin selkeä menettelytapa. Suosituksissa pyritään myös välttämään väärin positiivisten suositusten antamista, eli pyritään välttämään antamasta korkeaa oletettua arvosanaa asialle, josta käyttäjä ei todellisudessa pidä. Väärän negatiivisen suosituksen antamista ei pidetä yhtä haitallisena. Edellä mainituista asioista seuraa, että epävarmoissa tilanteissa suosituksen arvosanalle kannattaa antaa todellista matalampi arvo, jolloin vältetään väärin positiivisten suositusten antamista. Ongelmana on, että tällöin suosituksena annettu arvosana ei enää olekaan varsinainen arvosana. Siitä tulee jonkinlainen suositeltavuutta kuvaava luku. Tämä voi olla käyttäjän kannalta hämäävää.

Jotkin suositusjärjestelmät antavat suosituksen suositeltavuutta esittävänä lukuna, eikä arvosanoina. Tällöin vältetään edellisessä kappaleessa esitetyt ongelmat, mutta samalla menetetään suositusjärjestelmän sisääntulojen ja ulostulojen samankaltaisuus. Suositeltavuus ja arvosana voidaan yhdistää pariaksi arvoja, joista toinen kuvaa suositeltavuutta ja toinen oletettua arvosanaa. Jos suositeltavuus annetaan arvona suljetulta yksikköväliltä  $[0,1]$  ja arvosana on kokonaisluku väliltä  $[1,5]$ , niin suositusjärjestelmän antaman suositus voi olla esimerkiksi pari  $(0,6; 4)$ . Tällainen suositus on varsin helposti käyttäjän ymmärrettävissä.

Suositusjärjestelmän käyttämän datan harvuuden takia joissain tapauksessa ei pystytä antamaan minkäänlaista suositusta. Suositeltavuuden ja arvosanan parina annettu suositus on näihin tilanteisiin riittämätön. Jos datan perusteella ei pystytä tekemään minkäänlaista suositusta, on suositeltavuuden arvo parissa selkeästi 0, mutta mikä on oletetun arvosanan arvo? Arvosanalle voitaisiin antaa jokin arvo arvosana-asteikolta, mutta tämä antaisi kuvan, että tämä arvosana olisi jotenkin muista arvosanoista poikkeava. Ristiriitaisissa tilanteissa, joissa data viittaisi kahteen tai useampaan arvosanaan, suositeltavuuden ja arvosanan pari on riittämätön tapa esittää suositus. Ongelma voidaan korjata antamalla



suosituksena jokaiselle arvosanalle suositeltavuus, jolloin ristiriita voidaan ilmaista antamalla suositeltavuuksille yhtä suuret arvot. Koska arvosana-asteikko on tiedossa, suositus voidaan ilmaista vektorina, joka sisältää jokaisen arvosanan jäsenyysasteen. Esimerkiksi vektori  $[0; 0; 0; 0,5; 1,0]$  esittää suositusta, jossa arvosanan 4 jäsenyysaste on 0.5 ja arvosanan 5 jäsenyysaste on 1.0. Tällaisen suosituksen voisi käsittää ristiriitaisena, mutta se osoittaa kuitenkin selkeästi, että käyttäjä pitäisi suositeltavasta asiasta. Tietämättömyys voidaan ilmaista antamalla kaikille suositeltavuuksille arvoksi 0. Samalla vältetään ongelma, joka muodostui tietämättömyyden ilmaisemisesta suositeltavuuden ja arvosanan parina.

Tällaiset suositukset voidaan käsittää arvosanojen jäsenyysasteina. Ne eivät ole todennäköisyyksiä, sillä niiden toiminnan kannalta on oleellista ettei kaikkien arvosanojen jäsenyysasteiden summa ole yksi. Esimerkiksi sumeissa säätöjärjestelmissä ulostulo täytyy yleensä terävöittää [22], sillä esimerkiksi venttiiliä ei voida asettaa useaan asentoon yhtä aikaa. Suositusjärjestelmän tapauksessa lopullisen päätöksen tekee kuitenkin ihminen, joka pystyy varsin hyvin tulkitsemaan annettua suositusta. Suositusjärjestelmän käyttäjä ei todennäköisesti perusta päätöstään pelkästään suositusjärjestelmän antamaan suositukseen, vaan hän saa päätöksensä tueksi informaatiota myös muualta. Käyttäjä yhdistää eri lähteistä saadun informaation lopulliseksi päätökseksi, joten suosituksen ilmaisema epävarmuus hyödyttää käyttäjää, sillä epävarmoissa tilanteissa hän voi painottaa enemmän muita tietolähteitä päätöksen tekemisessä.

Suositusjärjestelmän tarvitsee joissain tapauksissa järjestää suositeltavat asiat suositeltavuusjärjestykseen. Tämä ei ole arvosanojen jäsenyysasteina ilmaistujen suositusten tapauksessa yksiselitteistä. Mihin väliin esimerkiksi tietämättömyyttä ilmaiseva suositus kuuluu, entä ristiriitaiset suositukset? Luvussa 4.9 esitetään yksi mahdollinen menetelmä tällaisten suositusten järjestämiseksi.

## 4.2 Suositussäännöt

Käyttäjien antamien arvosanojen perusteella jokaiselle käyttäjälle luodaan joukko sääntöjä, jotka kuvaavat hänen elokuvamieltymyksiään suhteessa muihin käyttäjiin.

Jokaista käyttäjän elokuvalla antamaa arvosanaa vastaan luodaan sääntö, jonka ulostulo on jäsenyysaste annetulle arvosanalle. Itse sääntö muodostuu muiden käyttäjien antamista arvosanoista. Sääntöjä käytetään suositusten laskemiseen. Helpointa järjestelmän toiminta on kuvata esimerkkidatan avulla. Taulukko 1 sisältää esimerkkidatan, jossa sarakkeet

ovat käyttäjiä ja rivit elokuvia. Arvosanat on rajattu taulukossa yksinkertaisuuden vuoksi yhdestä kolmeen.

Taulukko 1: Esimerkkidata käyttäjien  $K_1 - K_2$  antamista arvosanoista elokuville  $E_1 - E_{15}$ .

	$K_1$	$K_2$	$K_3$	$K_4$
$E_1$	1	3	1	1
$E_2$	1	3	2	1
$E_3$	1	3	3	1
$E_4$	2	2	1	2
$E_5$	2	2	2	2
$E_6$	2	2	3	2
$E_7$	3	1	1	3
$E_8$	3	1	2	3
$E_9$	3	1	3	3
$E_{10}$	1			1
$E_{11}$	1			
$E_{12}$	1	3		
$E_{13}$	1	1		
$E_{14}$			2	
$E_{15}$		3	2	

Käyttäjän  $K_4$  antamille arvosanoille luodaan kuvatulla tavalla datasta seuraavat säännöt:

$R_1$ : **jos**  $K_1$  on antanut 1 vastaavan arvosanan **ja**  $K_2$  on antanut 3 vastaavan arvosanan **ja**  $K_3$  on antanut 1 vastaavan arvosanan **ja** säännöllä on merkitystä sisääntulon kannalta **niin**  $K_4$  arvosana on 1

**myös**

$R_2$ : **jos**  $K_1$  on antanut 1 vastaavan arvosanan **ja**  $K_2$  on antanut 3 vastaavan arvosanan **ja**  $K_3$  on antanut 2 vastaavan arvosanan **ja** säännöllä on merkitystä sisääntulon kannalta **niin**  $K_4$  arvosana on 1

**myös**

$R_3$ : **jos**  $K_1$  on antanut 1 vastaavan arvosanan **ja**  $K_2$  on antanut 3 vastaavan arvosanan **ja**  $K_3$  on antanut 3 vastaavan arvosanan **ja** säännöllä on merkitystä sisääntulon kannalta **niin**  $K_4$  arvosana on 1

**myös**

$R_4$ : **jos**  $K_1$  on antanut 2 vastaavan arvosanan **ja**  $K_2$  on antanut 2 vastaavan arvosanan **ja**  $K_3$  on antanut 1 vastaavan arvosanan **ja** säännöllä on merkitystä sisääntulon kannalta **niin**  $K_4$  arvosana on 2

**myös**

$R_5$ : **jos**  $K_1$  on antanut 2 vastaavan arvosanan **ja**  $K_2$  on antanut 2 vastaavan arvosanan **ja**  $K_3$  on antanut 3 vastaavan arvosanan **ja** säännöllä on merkitystä sisääntulon kannalta **niin**  $K_4$  arvosana on 2

**myös**

$R_6$ : **jos**  $K_1$  on antanut 2 vastaavan arvosanan **ja**  $K_2$  on antanut 2 vastaavan arvosanan **ja**  $K_3$  on antanut 4 vastaavan arvosanan **ja** säännöllä on merkitystä sisääntulon kannalta **niin**  $K_4$  arvosana on 2

**myös**

$R_7$ : **jos**  $K_1$  on antanut 3 vastaavan arvosanan **ja**  $K_2$  on antanut 1 vastaavan arvosanan **ja**  $K_3$  on antanut 1 vastaavan arvosanan **ja** säännöllä on merkitystä sisääntulon kannalta **niin**  $K_4$  arvosana on 3

**myös**

$R_8$ : **jos**  $K_1$  on antanut 3 vastaavan arvosanan **ja**  $K_2$  on antanut 1 vastaavan arvosanan **ja**  $K_3$  on antanut 2 vastaavan arvosanan **ja** säännöllä on merkitystä sisääntulon kannalta **niin**  $K_4$  arvosana on 3

**myös**

$R_9$ : **jos**  $K_1$  on antanut 3 vastaavan arvosanan **ja**  $K_2$  on antanut 1 vastaavan arvosanan **ja**  $K_3$  on antanut 3 vastaavan arvosanan **ja** säännöllä on merkitystä sisääntulon kannalta **niin**  $K_4$  arvosana on 3

**myös**

$R_{10}$ : **jos**  $K_1$  on antanut 1 vastaavan arvosanan **ja** säännön tulos on 1 **ja** säännöllä on merkitystä sisääntulon kannalta **niin**  $K_4$  arvosana on 1

Sääntöjoukosta lasketaan sisääntulona annetulle elokuvalla suosituksen muodostavat arvosanojen jäsenyysasteet. Useat sumeaa logiikkaa käyttävät järjestelmät perustuvat tällaisten sääntöjoukkojen aggregointiin [7]. Muille käyttäjille voidaan vastaavalla tavalla muodostaa sääntöjoukko.

Säännöt voitaisiin muodostaa myös sarakkeiden, eli käyttäjien, perusteella. Koska tarkoituksena on käyttää elokuvien tyyllilajeja vahvistamaan saman tyyllilajiin kuuluvien elokuvien sääntöjä, säännöt muodostetaan rivien, eli elokuvien, perusteella. Mahdollista olisi myös muodostaa sekä riveistä että sarakkeista sääntöjä ja yhdistää näiden tulos lopulliseksi tulokseksi. Tämän vaihtoehdon tutkiminen jätetään kuitenkin mahdolliseen jatko-tutkimukseen.

Kaikista elokuvista, joille aktiivinen käyttäjä on antanut arvosanan, voidaan siis muodostaa sääntö, jonka tulos on jäsenyysaste annetun arvosanan joukossa. Sääntöjen ”ja”-sanat ovat t-normeja ja ”myös”-sanat t-konormeja. Vastaavuus on sumea relaatio, joka kuvaa käyttäjän antaman arvosanan suhdetta sääntöjen tulokseen. Jos käyttäjä on antanut aina saman arvosanan kun aktiivinen käyttäjä on antanut jonkin tietyn arvosanan, on sisääntulon ja säännön arvosanojen vastattava tarkemmin toisiaan. Esimerkkidatasta voidaan nähdä, että  $K_1$  on antanut aina saman arvosanan kuin  $K_4$ . Sisääntulon  $K_4$  arvosanan on siis vastattava säännön  $K_4$  arvosanaa hyvin tarkasti.  $K_3$  on puolestaan antanut  $K_1$  arvoso-noihin nähden useita eri arvosanoja. Tällöin säännön lopputuloksen kannalta käyttäjän  $K_3$  antamilla arvosanoilla ei ole juuri merkitystä. Käyttäjä  $K_2$  on antanut aina päinvastaisen arvosanan kuin aktiivinen käyttäjä  $K_4$ . Tämäkin on hyödyllinen tieto, jota voidaan käyttää suositusten laskemisessa. Menetelmän kannalta on samantekevää onko käyttäjä antanut aina saman arvosanan kuin aktiivinen käyttäjä, vain onko käyttäjä antanut aina jonkin tietyn muun arvosanan. Vastaavuus perustuu siis siihen, kuinka vahvasti voidaan sanoa käyttäjän antaessa jonkin arvosanan, että aktiivinen käyttäjä, eli käyttäjä, jolle suosituksia ollaan laskemassa, antaa jonkin tietyn, mahdollisesti toisen, arvosanan. Tarkemmin vastaavuuksien laskemista tarkastellaan seuraavassa luvussa.

Säännön merkittävyyden tarkoituksena on estää sellaisten sääntöjen vaikutus, joilla ei ole sisääntuloon nähden merkitystä, vaikuttamasta liikaa lopputulokseen. Esimerkiksi, jos käytetään elokuvan  $E_{14}$  saamia arvosanoja järjestelmän sisääntulona, millään säännön tulolla ei ole liiemmin merkitystä, sillä kyseiselle elokuvalla on antanut arvosanan vain käyttäjä  $K_3$ , jonka arvosanoilla ei ole juuri merkitystä aktiivisen käyttäjän antamiin arvoso-noihin. Säännön merkitys on myös pienempi, jos säännön ja sisääntulon yhteisten käyttäjien määrä on pieni. Esimerkiksi elokuvaa  $E_{11}$  käytettäessä sisääntulona sääntöjen merkittävyys on pienempi kuin käytettäessä elokuvaa  $E_{12}$  sisääntulona, sillä elokuvalla

$E_{11}$  on vähemmän arvosanoja kuin elokuvalla  $E_{12}$ . Näin estetään myös lyhyitä sääntöjä, kuten sääntöä  $R_{10}$ , vaikuttamasta liikaa suosituksen lopputulokseen. Lyhyillä säännöillä olisi muuten etulyöntiasema pitkiin sääntöihin nähden, sillä lyhyt sääntö vastaa helpommin sisääntuloa kuin pitkä sääntö. Suositukset halutaan kuitenkin mieluiten perustuvan pitkiin kuin lyhyisiin sääntöihin, koska pitkien sääntöjen voisi olettaa kuvaavan tarkemmin käyttäjän mieltymyksiä. Merkittävyyden laskemisesta tarkemmin luvussa 4.4.

Jokainen sääntö antaa jäsenyysasteen vain yhdelle arvosanalle. Tästä on seurauksena, että järjestelmä voi antaa suosituksena nolasta suuremman jäsenyysasteen vain arvosanoille, joita aktiivinen käyttäjä on antanut. Tätä voidaan kuitenkin jossain mielessä pitää järkevänä, sillä jos aktiivinen käyttäjä ei ole antanut muita arvosanoja kuin 1 ja 5, niin ei voida olettaa, että hän antaisi muita arvosanoja niillekään elokuville, joita hän ei ole vielä nähnyt. Järjestelmää olisi myös mahdollista muokata siten, että sääntö voisi antaa arvosanan sumeassa muodossa, jolloin sääntö vaikuttaisi myös viereisten arvosanojen jäsenyysasteisiin. Tämä voisi olla hyödyllinen lähestymistapa, sillä käyttäjän antamia arvosanoja ei välttämättä kannata pitää terävinä lukuina myöskään säännön tuloksessa. Tätä lähestymistapaa ei kuitenkaan tässä työssä tutkita tarkemmin.

Suurimpia ulostuloja antaneita sääntöjä voidaan käyttää perustelemaan annettu suositus käyttäjälle. Käyttäjälle voidaan esimerkiksi kertoa, että kyseinen suositus johtuu siitä, että hän on antanut tietyn arvosanan jollekin toiselle elokuvalla tai elokuville. Myös yksittäisten sääntöjen eniten vaikuttaneita arvosanoja voitaisiin käyttää suosituksen perustelussa, mutta silloin jouduttaisiin paljastamaan muiden käyttäjien elokuvamieltymyksiä, mitä ei välttämättä voida pitää hyväksyttävänä yksityisyydensuojan kannalta.

Seuraavaksi tarkastellaan tarkemmin, mitä sääntöjen eri osat tarkoittavat.

### 4.3 Arvosanojen vastaavuus

Arvosanojen vastaavuudet määritellään sumean relaation avulla, joka puolestaan määritellään painojen avulla. Säännön arvosanan ja sisääntulon arvosanan vastaavuus lasketaan tarkoitukseen kehitetyn yhtälön 16 avulla. Yhtälössä  $i$  on aktiivinen käyttäjä,  $j$  toinen käyttäjä,  $x$  säännön arvosana,  $y$  sisääntulon arvosana ja  $W_{i,j,x}$  aktiivisen käyttäjän  $i$  paino käyttäjän  $j$  antamalle arvosanalle  $x$ .

$$S_{i,j}(x,y) = \max(0, 1 - W_{i,j,x}|x - y|) \quad (16)$$

Taulukossa 2 on esimerkki painoista, joista voidaan laskea yhtälöä 16 käyttäen arvosanojen vastaavuudet. Saatu sumea relaatio esitetään taulukossa 3. Esimerkin arvosanalla viisi painoksi on annettu nolla, jolloin kaikki arvosanat täysin vastaavat arvosanaa viisi. Tällöin arvosana ei missään tapauksessa pysty rajoittamaan säännön ulostuloa. Näin esitetty relaatio on painoista riippumatta aina refleksiivinen, ja ilmeisesti myös transitiivinen. Tämä tekisi siitä läheisyysrelaation. Relaatio on myös symmetrinen, kun kaikki painot ovat samoja. Käytännössä tämä on kuitenkin varsin harvinainen tilanne. Relaatiolla ei kuitenkaan tarvitse olla mitään erityistä ominaisuutta soveltuakseen järjestelmän käyttötarkoitukseen, tosin arvosanojen merkittävyyden laskennassa tällä hetkellä oletetaan, että relaatio on refleksiivinen. Arvosanojen vastaavuudet määräävä sumea relaatio on järjestelmän keskeinen osa, jota muuttamalla vaikutetaan koko järjestelmän toimintaan. Tässä esiteltävä sumea relaatio on valittu, koska se antaa hyvän kompromissin yksinkertaisuuden ja ilmaisuvoiman välillä. Sillä voidaan esimerkiksi ilmaista tilannetta, jossa joku toinen käyttäjä on aina antanut arvosanan 1, kun aktiivinen käyttäjä on antanut arvosanan 5, mutta muutoin käyttäjien antamat arvosanat eivät korreloi millään tavalla. Optimoitavien painojen määrä on kohtuullisen pieni, joten hyvien painojen löytäminen ei muodostu liian raskaaksi ongelmaksi laskea. Painojen optimoinnista tarkemmin kappaleessa 4.7.

Yksi mahdollinen muutos esitettyyn relaation olisi määritellä erikseen painot positiivisille ja negatiivisille poikkeamille säännön arvosanasta. Tällöin käytettäisiin eri painoja riippuen siitä, onko sisääntulon arvosana pienempi vai suurempi kuin säännön arvosana. Painojen määrä siis kaksinkertaistuisi, mikä aiheuttaisi lisää laskentaa. Negatiivisten ja positiivisten painojen ollessa samoja relaatio vastaisi aiemmin esiteltyä relaatiota. Tämän lisäksi sillä voitaisiin esittää esimerkiksi tilanteita, joissa sisääntulon arvosana vastaisi säännön arvosanaa vain, jos se olisi sama tai suurempi. Tämän mahdollisuuden tarkempi tutkiminen jätetään kuitenkin tämän työn ulkopuolelle.

Taulukko 2: Esimerkki arvosanojen vastaavuuksien painoista.

1	2	3	4	5
1	0,75	0,25	0,5	0

Taulukko 3: Esimerkki arvosanojen vastaavuuksien sumeasta relaatiosta.

	1	2	3	4	5
1	1	0	0	0	0
2	0,25	1	0,25	0	0
3	0,25	0,75	1	0,75	0,25
4	0	0	0,5	1	0,5
5	1	1	1	1	1

#### 4.4 Säännön merkittävyys

Säännön merkittävyydellä pyritään estämään tilanteita, joissa vähämerkitykselliset säännöt pääsevät vaikuttamaan liikaa suosituksen tulokseen. Jokaiselle säännön arvosanalle lasketaan merkittävyys yhtälöllä 17. Mitä suurempi paino arvosanalla on, sitä merkittävämpi se on. Painolla 1 arvosana saa merkittävyydeksi 1 ja painolla 0 merkittävyydeksi 0. Yhtälössä  $n$  tarkoittaa korkeinta arvosanaa, joka esimerkiksi taulukossa 3 on 5. Matalimman arvosanan oletetaan olevan 1.

$$M_{i,j}(x) = 1 - \frac{\left[ \sum_{y=1}^n S_{i,j}(x,y) \right] - 1}{n - 1} \quad (17)$$

Esimerkiksi taulukosta 3 voidaan laskea merkittävyys arvosanalle 2 seuraavalla tavalla:  $1 - \frac{0,25+1+0,25+0+0-1}{5-1} = 1 - \frac{0,5}{4} = 0,875$ . Käyttäjän antamalla arvosanalla 2 on siis varsin korkea merkittävyys aktiivisen käyttäjän antamiin arvosanoihin nähden.

Arvosanan merkittävyyttä rajoitetaan aktiivisen käyttäjän ja arvosanan antaneen käyttäjän yhteisten elokuvien määrän mukaan. Tällä on tarkoitus rajoittaa arvosanojen merkittävyyttä, jos merkittävyys perustuu vain muutamaankin yhteiseen arvosanaan. Yhtälössä 18 rajoitetaan yhtälöllä 17 laskettua merkittävyyttä käyttäjien yhteisten elokuvien määrän mukaan. Yhtälössä  $C_{i,j}$  tarkoittaa aktiivisen käyttäjän  $i$  ja arvosanan antaneen käyttäjän  $j$  yhteisten elokuvien joukkoa. Parametri  $f$  määrää, miten paljon yhteisten arvosanojen lukumäärä voi rajoittaa arvosanan merkittävyyttä. Kaavassa  $\wedge$  tarkoittaa vapaasti valittavaa  $t$ -normia. Arvosanan merkittävyyttä rajoitetaan siis käyttäjien kaikkien yhteisten elokuvien määrän mukaan, eikä yhteisten elokuvien mukaan, joille käyttäjä on antanut kyseisen arvosanan. Arvosanan merkittävyyttä voitaisiin rajoittaa myös jälkimmäisellä tavalla, mutta se toisi hieman lisää laskentaa. Tosin samalla se saattaisi parantaa hieman suositusten tasoa. Tätä tapaa ei kuitenkaan tarkastella tässä työssä tämän enempää, vaan se jätetään mahdolliseen jatkotutkimukseen.

$$M_{i,j}^f(x) = M_{i,j}(x) \wedge \min \left( 1, \frac{|C_{i,j}|}{f} \right) \quad (18)$$

Säännön merkittävyys lasketaan summaamalla säännön muodostavien arvosanojen merkittävyydet yhteen ja jakamalla se korkeimmalla mahdollisella merkittävyydellä, jonka sääntö voi saada. Säännön merkittävyys on riippuvainen sisääntulosta, sillä vain sisääntulon ja säännön yhteiset arvosanat vaikuttavat säännön merkittävyyteen. Jollekin sisään-

tulolle säännöllä voi olla suuri merkitys, kun taas toiselle sillä ei ole mitään merkitystä. Säännön merkittävyys lasketaan yhtälöllä 19, jossa käytetty säännön korkein mahdollinen merkittävyys  $H_i^{max}$  lasketaan yhtälöllä 20.  $C_{i,r}^I$  on aktiivisen käyttäjän  $i$  säännön  $r$  ja sisääntulon  $I$  yhteisten käyttäjien antamien arvosanojen joukko.  $X_{i,j,r}$  on aktiivisen käyttäjän  $i$  säännön  $r$  arvosana käyttäjälle  $j$ .

$$H_{i,r}(I) = \frac{\sum_{j \in C_{i,r}^I} M_{i,j}^f(X_{i,j,r})}{H_i^{max}} \quad (19)$$

$$H_i^{max} = \sum_j \max_{x=1}^n (M_{i,j}^f(x)) \quad (20)$$

$H_i^{max}$  on korkein mahdollinen merkittävyys, jonka käyttäjän  $i$  sääntö voi saada. Se on summa jokaisen käyttäjän korkeimmista arvosanan merkittävyyksistä, eli se on merkittävyys, jonka sääntö saisi, jos jokainen käyttäjä olisi antanut säännön muodostavalle elokuvalle eniten merkitsevän arvosanansa. Käytännössä näin ei pääse koskaan käymään, sillä ei ole todennäköisesti olemassa elokuvaa, jonka jokainen käyttäjä on nähnyt, saati elokuvaa, jolle jokainen käyttäjä olisi antanut eniten merkitsevän arvosanansa. Korkeinta mahdollista säännön merkittävyyttä käytetään normalisoimaan sääntöjen merkittävyys suljetulle yksikköväliä  $[0,1]$ . Toteutetussa menetelmässä tätä normalisoitua arvoa käytetään suoraan, mutta sitä voitaisiin myös pehmentää esimerkiksi ottamalla arvosta neliöjuuri. Tällöin säännön merkittävyyden ei tarvitsisi olla yhtä lähellä korkeinta mahdollista säännön merkittävyyttä saadakseen korkean normalisoidun merkittävyyden.

## 4.5 Säännön ulostulo

Säännön ulostulo sisääntulolle lasketaan säännön ja sisääntulon yhteisten käyttäjien perusteella. Jos sisääntulolla ja säännöllä ei ole yhtään yhteisen käyttäjän antamaa arvosanaa, on säännön ulostulo nolla. Aktiivisen käyttäjän  $i$  säännön  $r$  ulostulo  $O_{i,r}$  sisääntulolle  $I$  lasketaan yhtälöllä 21, jossa  $X$  on säännön arvosanat ja  $C_{i,r}^I$  on aktiivisen käyttäjän  $i$  säännön  $r$  ja sisääntulon  $I$  yhteiset käyttäjät.  $H_{i,r}(I)$  on aktiivisen käyttäjän  $i$  säännön  $r$  merkittävyys sisääntulolle  $I$ .  $\wedge$  ja  $\wedge$  ovat vapaasti valittavia t-normeja.



$$O_{i,r}(I) = \left( \bigwedge_{j \in C_{i,r}^I} S_{i,j}(X_{j,r}, I_j) \right) \wedge H_{i,r}(I) \quad (21)$$

Jotta säännöllä olisi nollasta poikkeava ulostulo, on sen ja sisääntulon kaikkien arvosanojen vastattava toisiaan ja säännöllä on oltava nollasta poikkeava merkittävyys sisääntuloon nähden. Säännön ulostulo on tietysti myös nolla silloin kun säännöllä ja sisääntulolla ei ole yhtään saman käyttäjän antamaa arvosanaa.

## 4.6 Sääntöjen ulostulojen yhdistäminen

Sääntöjen ulostulojen yhdistämiseen käytetään t-konormia jokaisen arvosanan jäsenyyssasteen laskemiseksi. Aktiivisen käyttäjän  $i$  sääntöjoukon sisääntuloa  $I$  vastaava ulostulo  $P_{i,x}(I)$  arvosanalle  $x$  lasketaan yhtälöllä 22, jossa  $R_{i,x}$  on aktiivisen käyttäjän  $i$  säännöt, joiden ulostulo on arvosanan  $x$  jäsenyysaste.  $\bigvee$  on vapaasti valittava t-konormi, joka antaa mieluusti maksimi-operaatiota suuremman tuloksen, eli jos  $K$  on t-konormi, niin  $K(x,y) > \max(x,y)$ ,  $x,y \in ]0,1[$ . Tämä sen takia, että sääntöjen ulostuloista yhdistetty suositus voisi saada korkeita jäsenyysasteita arvosanojen joukossa, vaikka millään yksittäisellä ulostulolla ei olisi korkeaa jäsenyysastetta. Jos monta sääntöä antaa sisääntulolle nollasta poikkeavan jäsenyysasteen tietyille arvosanalla, voidaan olettaa, että tällä sisääntulolla on kohtuullisen korkea jäsenyysaste arvosanan joukossa. Käytettäessä maksimi-operaatiota vain korkeimman jäsenyysasteen antaneella säännöllä on merkitystä arvosanan jäsenyysasteelle.

$$P_{i,x}(I) = \bigvee_{r \in R_{i,x}} O_{i,r}(I) \quad (22)$$

Sääntöjen ulostulojen yhdistämiseen voisi myös käyttää funktiota, jotka eivät täytä t-konormin määritelmää. Esimerkiksi OWA-operaattorit voisivat olla yksi tutkimisen arvoinen vaihtoehto.

## 4.7 Vastaavuuksien optimointi

Kehitetyn suositusmenetelmän toiminta perustuu käyttäjien antamien arvosanojen vastaavuuksien optimointiin siten, että säännöt antavat mahdollisimman korkean jäsenyysasteen oikealle arvosalalle ja pienen väärille arvosalalle. Aktiivisen käyttäjän antamia arvosalanoja käytetään vastaavuuksien määrittävien arvosanojen painojen optimoinnissa. Sääntöjoukolle annetaan syötteenä säännöt muodostaneet elokuvat siten, että laskettavasta elokuvasta muodostettua sääntöä ei käytetä. Tämä siksi, että tämä yksittäinen sääntö ei pääsisi liikaa vaikuttamaan optimoinnin tulokseen, sillä sääntö antaa korkeimman mahdollisen arvon silloin, kun sille annetaan syötteenä arvosalat, joista sääntö on muodostettu.

Jotta tiedettäisiin, mitkä painot tuottavat hyviä tuloksia, on laskettuja suosituksia jotenkin pystyttävä vertaamaan tiedettyihin arvosaloihin. Tähän käytetään tarkoitukseen kehitettyä yhtälöä 23, jolla lasketaan suosituksen virhe  $F$  tiedettyyn arvosalanaan nähden. Yhtälössä  $P$  on vektori, joka sisältää elokuvan jäsenyysasteet eri arvosaloina,  $y$  on elokuvan tiedetty arvosalana ja  $n$  on korkein arvosalana.  $m$  ja  $d$  ovat parametreja, joilla voidaan säätää sitä, miten paljon oikeasta arvosalasta poikkeavista jäsenyysasteista rangaistaan. Parametri  $m$  määrää lineaarisesti kasvavan rangaistuksen määrä ja parametri  $d$  eksponentiaalisesti kasvavan rangaistuksen määrän. Yhtälön lähtökohtana on oikean arvosalan jäsenyysaste, josta vähennetään kaikkien väärin arvosalojen jäsenyysasteet, jotka ovat oikeaa jäsenyysastetta korkeampia. Mitä kauempana väärä arvosalana on oikeasta arvosalasta, sitä enemmän virheestä rangaistaan.

$$F(P, y) = P_y - \sum_{i=1}^n \max(0, P_i - P_y) m |i - y|^d \quad (23)$$

Koska käyttäjien antamia arvosalanoja ei voida pitää erityisen terävinä lukuina, yhtälö 23 ei rankaise vääristä jäsenyysasteista, jotka eivät ylitä oikeaa jäsenyysastetta. Suosituksella voi siis olla positiivinen jäsenyysaste muissakin arvosaloina ilman, että siitä rangaistaan. Esimerkiksi on hyvin todennäköistä, että oikean arvosalan viereiset arvosalat saavat myös positiivisen jäsenyysasteen. Liian ankara rankaiseminen tällaisissa tilanteissa voisi johtaa siihen, että suositusjärjestelmä antaisi paljon suosituksia, joiden jäsenyysaste kaikissa arvosaloina olisi nolla.

Esimerkiksi jos järjestelmän antama suositus on  $[0,0;0,5;0,4;0,4]$  ja tiedetty arvosalana on 4, voidaan suosituksen tarkkuus laskea seuraavasti:  $F([0,0;0,5;0,4;0,4], 4) = 0,4 - [\max(0;0,0-0,4)m|1-4|^d + \max(0;0,5-0,4)m|2-4|^d + \max(0;0,0-0,4)m|3-$

$$4|d + \max(0; 0, 0 - 0, 4)m|4 - 4|d + \max(0; 0, 0 - 0, 4)m|5 - 4|] = 0, 4 - [0m3^d + 0, 1m2^d + 0m1^d + 0m0^d + 0m1^d = 0, 4 - 0, 1m2^d$$

Suosituksen tarkkuuden arvioinnissa voitaisiin käyttää muunlaisiakin menetelmiä, mutta esitetty yhtälö on valittu sen kohtuullisen yksinkertaisuuden vuoksi; siinä on vain kaksi parametria, joiden arvo täytyy selvittää kokeellisesti.

Vastaavuuksien painojen optimointi toteutetaan yksinkertaisesti kokeilemalla joitain erisuuruisia painoja, ja valitsemalla niistä paras jatko-optimointia varten, josta puolestaan paras valitaan vielä uutta optimointia varten. Ensiksi listasta  $[\frac{1}{10}, \frac{1}{9}, \frac{1}{8}, \frac{1}{7}, \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{1}{1}]$  etsitään paras paino. Seuraavaksi kokeillaan löytyykö valitun painon läheltä parempaa painoa. Kokeiltavat painot saadaan lisäämällä ja vähentämällä arvon jakajasta 0,5. Näistä kahdesta ja alkuperäisestä painosta valitaan paras, jonka jakajaa muutetaan vielä kerran, kuten edelläkin, mutta tällä kertaa arvolla 0,25. Tämän optimoinnin tulos on lopullinen painon arvo. Painojen optimointiin on olemassa todennäköisesti parempiakin menetelmiä, mutta tällä menetelmällä saadut arvot ovat todennäköisesti riittävän lähellä optimia ilman, että optimointiin kuluu liian paljon aikaa. Optimointi saattaa todennäköisesti juuttua joissain tapauksissa lokaaliin optimiin, mutta tätä pyritään välttämään laajalla optimoinnin alkuarvojen valinnalla. Alkuperäinen lista painoista on valittu sillä perusteella, että ne edustavat arvosanojen rajoja. Esimerkiksi painolla  $\frac{1}{2}$  arvosanan viereiset arvosanat saavat vielä positiivisen vastaavuuden, mutta kahden arvosanan päässä jäsenyysaste on jo nolla.

Painoja optimoitaessa ei tarvita kuin ne säännöt, joissa käyttäjä, jonka painoja ollaan optimoimassa, on antanut arvosanan. Säännöistäkään ei tarvitse laskea kuin kyseisen käyttäjän osa. Optimoinnissa tarvitsee siis laskea vain hyvin pieni osa säännöistä painon hyvyyden selvittämiseksi. Laskettujen suositusten hyvyys arvioidaan yhtälön 23 avulla. Kaikkien suositusten hyvyydet lasketaan yhteen painon hyvyyden saamiseksi.

Painoja optimoitaessa kaikille säännöille oletetaan merkittävyyden maksimiarvoksi yksi. Optimoinnissa ei oteta huomioon käyttäjien yhteisten elokuvien määrää, joten yhtälössä 19 käytetään yhtälöä 17 yhtälön 18 sijaan. Tämä siksi että, yhtälössä 18 esiintyvän muuttujan  $f$  eri arvojen kokeilu olisi nopeampaa. Jos arvo määrättäisiin jo optimoinnin yhteydessä, olisi eri arvojen kokeilemiseksi optimoitava kaikki painot uudestaan. Painojen optimointi on kaikesta huolimatta varsin hidas operaatio, joten ajan säästämiseksi painot optimoidaan vain kerran. Tämä päätös vaikuttaa tietysti jonkin verran järjestelmän antamiin suosituksiin, mutta koska arvosanojen merkittävyyden rajoittaminen käyttäjien yhteisten arvosanojen määrän perusteella ei ole järjestelmän toiminnan kannalta oleellinen ominaisuus, voidaan tämä menettely hyväksyä.

Kun aktiivinen käyttäjä antaa uuden arvosanan elokuvalle, joudutaan kaikkien elokuvale arvosanan antaneiden käyttäjien painot laskemaan uudestaan. Laskentaa ei kuitenkaan välttämättä tarvitse suorittaa, jos vanha paino toimii riittävän hyvin myös uuden säännön tapauksessa. Optimoinnissa on myös mahdollista hyödyntää vanhaa painoa, mutta silloin on vaarana jäädä lokaaliin optimiin. Aikaisemman optimoinnin välituloksien avulla painojen uudelleenoptimointia voitaisiin myös nopeuttaa, mutta silloin vaadittavan tallennuskapasiteetin tarve suurenisi. Eri käyttäjien painojen optimointi voidaan suorittaa osissa, sillä yhden painon optimoinnissa ei tarvita tietoa muista, saman tai muun käyttäjän, painojen arvoista.

## 4.8 Tyylilajibonus

Voidaan olettaa, että suosituksia voitaisiin parantaa painottamalla sääntöjä, jotka kuuluvat samaan elokuvatyyllilajiin sisääntulon kanssa.

Elokvien välinen tyylilajibonus lasketaan yhtälöllä 24. Bonus määrää voidaan muuttaa muuttujan  $b$  avulla. Bonus toimii kertoimena elokuvan säännön arvosanan jäsenyysasteelle. Bonus voi siis vain korottaa säännön tulosta. Korotettu tulos rajoitetaan maksimissaan yhteen, jotta saatu arvo pysyy jäsenyysasteen rajoissa. Yhtälössä  $g_1$  ja  $g_2$  ovat elokuvien tyylilajien joukot. Elokuva voi kuulua yhtä aikaa moneen tyylilajiin, se voi esimerkiksi olla sekä komedia- että toimintaelokuva.

$$B(g_1, g_2) = 1 + b \frac{|g_1 \cap g_2|}{|g_1 \cup g_2|} \quad (24)$$

Esimerkki: jos säännön tyylilajit ovat joukko  $\{toiminta, komedia\}$ , sisääntulon tyylilajit on joukko  $\{toiminta\}$  ja bonus  $b$  on 0,5, on bonuskerroin  $B(\{toiminta, komedia\}, \{toiminta\}) = 1 + 0,5 \frac{|\{toiminta, komedia\} \cap \{toiminta\}|}{|\{toiminta, komedia\} \cup \{toiminta\}|} = 1 + 0,5 \frac{|\{toiminta\}|}{|\{toiminta, komedia\}|} = 1 + 0,5 \frac{1}{2} = 1,25$ . Huomioitavaa on, että argumentteina annettujen tyylilajijoukkojen järjestyksellä ei ole väliä, sillä myös  $B(\{toiminta\}, \{toiminta, komedia\})$  antaa saman vastauksen kuin esimerkki.

Bonuskerrointa käytetään säännön ulostulon kertoimena rajoittaen tulo korkeintaan arvoon yksi. Säännön ulostulo tyylilajibonus kanssa lasketaan yhtälöllä 25. Kaavaa voidaan käyttää tavallisen ulostulon asemasta yhtälössä 22.

$$O_{i,r}^b(I) = \min[1, O_{i,r}(I)B(G_{i,r}, g_I)] \quad (25)$$

Elokuvan tyyllilajin lisäksi samankaltaista bonusmenetelmää voitaisiin käyttää esimerkiksi elokuvan näyttelijöiden kanssa, jolloin säännön ulostuloa kasvatettaisiin, jos sillä olisi samoja näyttelijöitä sisääntulon kanssa.

## 4.9 Suositusten järjestäminen

Suositusmenetelmän antamia suosituksia ei voida yksiselitteisesti järjestää paremmuusjärjestykseen. Oletetaan arvosanojen olevan kokonaislukuja väliltä 1 ja 5. Arvosanojen 1 ja 2 voidaan sanoa olevan huonoja arvosanoja ja arvosanojen 4 ja 5 hyviä arvosanoja. Arvosana 3 on neutraali. Mihin väliin kuulu suositus, joka ei ole saanut yhtään nolasta poikkeavaa arvosanan jäsenyysastetta? Entä miten järjestetään ristiriitaisuutta ilmaisevat suositukset?

Lasketaan suositukset tiedossa oleville arvosanoille, siten että ei käytetä sääntöä, joka on luotu laskettavasta elokuvasta. Näin on tiedossa suositukset ja järjestys, johon suositukset tulisi järjestää. Oletetaan, että on olemassa funktio, joka antaa suositeltavuusarvon suosituksen perusteella, siten että suositeltavuusarvoja käytettäessä päästään samaan järjestykseen kuin tiedettyjä arvosanoja käytettäessä.

Määritellään yksinkertainen malli, jota voidaan optimoida tiedettyjen arvosanojen ja suositusten perusteella siten, että se antaa hyviä suositeltavuusarvoja. Näin vältetään tehtävältä määrittää manuaalisesti suositusten paremmuusjärjestys. Paremmuusjärjestys määrittyy automaattisesti tiedettyjen arvosanojen ja niille laskettujen suositusten perusteella.

Suosittelavuusarvon laskemiseen voitaisiin yksinkertaisemmillaan käyttää arvosanojen jäsenyysasteille määriteltyä painovektoria. Esimerkiksi viiden arvosanan asteikolle voitaisiin määrittää painovektori  $[-3; -2; 1; 2; 3]$ , jossa  $-3$  on arvosanan 1 jäsenyysasteen paino ja  $3$  arvosanan 5 jäsenyysasteen paino. Suositeltavuusarvo saataisiin kertomalla jäsenyysasteet painolla ja laskemalla saadut arvot yhteen. Painoja muuttamalla voitaisiin vaikuttaa suositusjärjestykseen. Tämä malli olisi hyvin yksinkertainen, mutta sillä ei voida esittää esimerkiksi taulukossa 4 esitettyä suositusten järjestystä. Jotta suositukset saataisiin taulukossa esitettyyn järjestykseen, olisi arvosanojen neljä ja viisi painojen oltava positiivisia, sillä muutoin ne tulisivat järjestyksessä vasta kaikille arvosanojen jäsenyysasteille nollan saaneen suosituksen jälkeen. Sekä arvosanan neljä että viisi painojen ollessa

positiivisia käy kuitenkin niin, että suositus [0,0; 0,0; 0,0; 1,0; 1,0] päättyy suositusjärjestyksessä ensimmäiseksi.

Taulukko 4: Suositusten esimerkkijärjestys, jossa ylimpänä ensimmäisenä suositeltava suositus.

1	2	3	4	5
0,0	0,0	0,0	0,0	1,0
0,0	0,0	0,0	1,0	1,0
0,0	0,0	0,0	1,0	0,0
0,0	0,0	1,0	0,0	0,0
0,0	0,0	0,0	0,0	0,0
0,0	1,0	0,0	0,0	0,0
1,0	0,0	0,0	0,0	0,0

Edellä esitettyä suositusjärjestyksen laskemistapaa voidaan laajentaa ottamalla huomioon korkeimman jäsenyysasteen saanut arvosana. Korkeimman jäsenyysasteen saanut arvosana voidaan olettaa olevan suosituksen kannalta merkittävin arvosana, joka siis vaikuttaa eniten suositusjärjestykseen. Jokaiselle korkeimmalle arvosanalle on määritelty oma järjestyksen painovektori, jossa kyseisellä arvosanalla on joko negatiivinen tai positiivinen paino ja muilla arvosanoilla on painot, jotka joko korottavat tai laskevat suositeltavuusarvoa. Esimerkiksi arvosanalla viisi saadessa korkeimman jäsenyysasteen järjestyksen painovektori voisi olla [-4;-3;-2;-1;5]. Arvosanaa viisi matalampien arvosanojen jäsenyysasteet siis laskisivat suositeltavuusarvoa, näin suositus, jolla olisi vain nolasta poikkeava jäsenyysaste arvosanan viisi joukossa, saisi korkeamman suositeltavuusarvon kuin suositus, jolla olisi jäsenyysasteita myös pienemmillä arvosanoilla. Jos kaikki korkeimman arvosanan painovektorit olisivat samoja, olisi tämä menetelmä vastaava edellisessä kappaleessa esitetyn menetelmän kanssa. Se on siis ainakin yhtä ilmaisuvoimainen. Taulukossa 5 on esimerkki matriisista, jota voitaisiin käyttää suositusten järjestämisessä. Rivit ovat korkeimman jäsenyysasteen saaneet arvosanat ja sarakkeet arvosanojen painot korkeimman arvosanan suhteen. Esimerkkimatriisi ei ole erityisen hyvä järjestämään suosituksia paremmuusjärjestykseen, mutta se antaa kuvan järjestysmatriisin toiminnasta. Suositeltavuusarvo lasketaan yhtälöllä 26, jossa  $X_i$  on suositus arvosanalle  $i$  ja  $J_{i,k}$  suosituskerroin arvosanalle  $i$  arvosanan  $k$  ollessa korkeimman jäsenyysasteen saanut arvosana.

$$Q(X) = \sum_i X_i J_{i,k} \quad (26)$$

Taulukko 5: Esimerkki järjestysmatriisista viiden arvosanan asteikolle.

	1	2	3	4	5
1	-6	1	2	3	4
2	-1	-4	1	2	3
3	-2	-1	2	1	2
4	-3	-2	-1	4	1
5	-4	-3	-2	-1	6

Järjestyksessä voidaan olettaa esiintyvän jonkin verran tilanteita, joissa pieni arvosanan jäsenyysasteen muutos aiheuttaisi varsin suuren muutoksen suosituksen suositeltavuusarvossa. Tämä on arvosanojen jäsenyysasteina annettujen suositusten tapauksessa väistämätöntä, sillä moniulotteista suositusta ei voi sulavasti muuttaa yksiulotteiseksi arvoksi. Järjestysmatriisiin optimointiin käytetään differentiaalievoluutioalgoritmia. Differentiaalievoluutioalgoritmi on optimointimenetelmä, joka perustuu mahdollisista vastauksista koostuvan populaation evoluutioon, risteyttämällä ja mutatoimalla populaation jäseniä. Populaation elinkelpoisimmat jäsenet pääsevät jatkamaan uuteen populaatioon. Algoritmin toiminta on kuvattu tarkemmin esimerkiksi viitteessä [23]. Algoritmin parametreille  $c_r$  ja  $f$  annettiin arvoksi 0,9.

Jotta järjestyspainot voitaisiin optimoida differentiaalievoluutioalgoritmillä, täytyy painoilla laskettua suositusten hyvyyttä pystyä mittaamaan. Järjestyksen virhettä voidaan mitata tarkoitukseen kehitetyllä yhtälöllä 27, jossa  $x^s$  on suositeltavuusjärjestykseen järjestetyt suositukset. Suositeltavin elokuva on ensimmäisenä ja vähiten suositeltava viimeisenä. Järjestyksen katsotaan olevan virheellinen, jos matalampi arvosana on järjestyksessä ylempänä kuin korkea arvosana. Esimerkiksi arvosanojen [3,1,2] järjestyksen virhe on  $|\min(0, 3 - 1) + \min(0, 3 - 2) + \min(0, 1 - 2)| = |0 + 0 + -1| = 1$

$$V(x^s) = \left| \sum_{i=1}^{n-1} \sum_{j=i+1}^n \min(0, (x_i^s - x_j^s)) \right| \quad (27)$$

Jokaiselle käyttäjälle lasketaan samaa painomatriisia käyttäen suositusten järjestykset, joista lasketut virheet lasketaan yhteen järjestysmatriisiin kokonaisvirheeksi.

Jokaiselle käyttäjälle voitaisiin myös käyttää omaa järjestysmatriisia, mutta vain muutamman elokuvan arvostelleille käyttäjille tämä olisi hankalaa datan vähyyden vuoksi. Siksi päädyttiin käyttämään samaa järjestysmatriisia kaikille käyttäjille.

## 5 SUORITETTAVAT KOKEET

Kehitetyn menetelmän suorituskykyä testataan usealla tavalla, jotta saadaan selville, miten hyvin se toimii muihin menetelmiin nähden, ja mitkä ovat sen vahvuudet ja heikkoudet. Kokeiden tulosten perusteella voidaan myös ohjata mahdollisen jatkokehityksen suuntaa.

### 5.1 Datasta

Datana käytetään MovieLens-sivuston julkaisemaa dataa käyttäjien elokuville antamista arvosanoista. Data sisältää 100 000 arviota 943 käyttäjältä 1682 elokuvalla. Kaikista mahdollisista käyttäjien elokuville antamista arvosanoista data sisältää vain noin 6,3%, loput noin 93,7% täytyy arvioida varsin vähän datamäärän perusteella. Arvosanat ovat kokonaislukuja väliltä yhdestä viiteen. Yksi ja kaksi tarkoittavat, että käyttäjä ei pitänyt elokuvasta, neljä ja viisi tarkoittavat, että käyttäjä piti elokuvasta, kolme tarkoittaa, että käyttäjä ei erityisemmin pitänyt elokuvasta tai inhonnut elokuvaa. Kaikki käyttäjät ovat arvioineet vähintään 20 elokuvaa. Keskimäärin käyttäjät ovat arvioineet 71 elokuvaa.

Data on kerätty 19.9.1997 ja 22.4.1998 välisenä aikana MovieLens-sivulta (movie-lens.umn.edu) osana Minnesotan yliopiston GroupLens-tutkimusprojektia.

Data sisältää myös tiedon elokuvan tyylilajista. Kaikki elokuvat kuuluvat yhteen tai useampaan seuraavista tyylilajeista: Unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western.

Data sisältää myös seuraavat tiedot käyttäjistä: ikä, sukupuoli, ammatti. Näitä tietoja ei kuitenkaan hyödynnetä tässä työssä kehitetyssä menetelmässä.

### 5.2 Datan jakaminen kokeita varten

Data täytyy jollain tavalla jakaa opetusdataan, jolla järjestelmää opetetaan tunnistamaan käyttäjän mieltymykset, ja testidataan, jolla järjestelmän suorituskykyä voidaan mitata. Koska suositukset pyritään järjestämään paremmuusjärjestykseen, testidata jaetaan ottamalla jokaiselta käyttäjältä kymmenen arviota testidataksi. Loput datasta on opetusdataa.



Käyttäjien saamien suositusten järjestyksen vertaileminen on helpompaa, kun kaikilla käyttäjillä on yhtä monta suositusta. Tätä datan jakamista testidataan ja opetusdataan käytettiin myös muussa suositusjärjestelmiä tutkivassa artikkelissa [24].

### 5.3 Verrattavat menetelmät

Kehitettyä menetelmää verrataan kahteen muuhun menetelmään sen suositustarkkuuden selvittämiseksi. Menetelmät ovat yksinkertainen käyttäjien antamien arvosanojen keskiarvo ja Pearsonin korrelaation perustuva suositusmenetelmä. Kehitettyä järjestelmää ei suoraan verrata muihin sumeaa logiikkaa hyödyntäviin järjestelmiin, sillä tällainen vertailu ei antaisi hyvää kuvaa järjestelmän suorituskyvystä yleisesti muihin suositusmenetelmiin nähden, vaan kertoisi vain sen suorituskyvystä kyseiseen sumeaa logiikka hyödyntävään menetelmään nähden. Pearsonin korrelaatioon perustuvaa menetelmää käytetään varsin paljon vertailukohtana muissa suositusjärjestelmiä koskevissa tutkimuksissa, joten menetelmiä voidaan epäsuorasti verrata riippuen, miten hyvin ne pärjäävät Pearsonin korrelaatioon perustuvan menetelmän kanssa.

Kaikkien käyttäjien antama arvosanojen keskiarvo on myös hyvin yleinen tapa ilmaista elokuvan hyvyttä, joten sitä käytetään myös yhtenä vertailtavana menetelmänä.

Parhaimmat suositusjärjestelmät perustuvat useiden eri menetelmien yhdistämiseen, kuten esimerkiksi Netflix-kilpailun voittanut suositusjärjestelmä [25]. Nämä menetelmät ovat siis varsin monimutkaisia toteuttaa, joten niihin vertaaminen on varsin työläs tehtävä, joka ei tehtyyn työhön nähden toisi mitään erityistä hyötyä. Kehitetty menetelmä on suorituskyvyltään varmasti parhaita menetelmiä huonompi, eikä huonouden tarkka määrä ole kokeiden kannalta erityisen mielenkiintoinen tieto.

#### 5.3.1 Arvosanojen keskiarvo

Elokuvalle annettujen arvosanojen keskiarvo on yleinen ja yksinkertainen tapa ilmaista elokuvan yleistä suositeltavuutta, se ei kuitenkaan varsinaisesti ole käyttäjäkohtainen suositusmenetelmä. Keskiarvo antaa kuitenkin hyvän vertailukohdan kehitetylle menetelmälle.

### 5.3.2 Pearsonin korrelaatio

Pearsonin korrelaatioon perustuvat suositusjärjestelmät olivat ensimmäisiä käytettyjä suositusjärjestelmiä. Niissä käyttäjien välisiä Pearsonin korrelaatioita käytetään painoina käyttäjien antamille arvosanoille. Menetelmä on toteutettu viitteessä [24] esitettyjen suositusten perusteella.

## 5.4 Kokeet

Kokeilla pyritään selvittämään parhaat parametrien arvot kehitetylle menetelmälle, ja arvioimaan kehitetyn menetelmän suorituskykyä vertailtaviin menetelmiin nähden. Suositusjärjestyksen hyvyttä verrataan myös vertailtavien menetelmien suositusjärjestyksiin.

### 5.4.1 Suositustarkkuus ja suositusten kattavuus

Menetelmän suositustarkkuutta arvioidaan RMSE (Root Mean Square Error) -arvon avulla. Tätä tapaa käytetään yleisesti suositusjärjestelmien suositustarkkuuden laskemiseen. RMSE-arvo lasketaan yhtälöllä 28, jossa  $x_n$  on käyttäjän antama arvosana asialle  $n$ ,  $r_n$  menetelmän antama suositus asialle  $n$ . Yhtälöstä huomataan, että suosituksen täytyy olla terävä luku suositustarkkuuden laskemiseksi. Koska kehitetty menetelmä antaa suositukset arvosanojen jäsenyysasteina, on ne terävöitettävä jollain menetelmällä. Terävöittäminen tehdään painotetulla keskiarvolla, jossa arvosanoja painotetaan niiden jäsenyysasteilla. Terävöittämisestä johtuen menetetään todennäköisesti osa kehitetyn menetelmän suositustarkkuudesta, mutta sen vaikutuksen ei kuitenkaan voida olettaa olevan liian suuri kehitetyn menetelmän yleisen suositustarkkuuden arvioimiseksi.

$$x_{rmse} = \sqrt{\frac{\sum (x_n - r_n)^2}{n}} \quad (28)$$

Suosituksien kattavuus on se prosenttiosuus suosituksista, joilla on ainakin yksi nollasta poikkeava jäsenyysaste jonkin arvosanan joukossa.

Kehitetylle menetelmälle yritetään löytää paras parametrien yhdistelmä. Muutettavat parametrit ovat yhtälön 23 kerroin  $m$  ja eksponentti  $d$ . Kertoimelle kokeillaan arvoja 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9 ja 1,0. Eksponentille puolestaan kokeillaan arvoja

0,8; 0,9; 1,0; 1,1 ja 1,2. Yhtälössä 24 muuttujalle  $b$  kokeillaan arvoja 0,5; 1,0; 1,5; 2,0; 2,5; 3,0; 3,5; 4,0; 4,5; 5,0; 5,5; 6,0 ja 6,5. Käyttäjien yhteisten elokuvien määrän rajan parametrin  $f$  arvoksi yhtälössä 18 kokeillaan arvoja 1, 5, 10, 15, 20 ja 25. Menetelmän  $t$ -konormeiksi kokeillaan seuraavia funktioita: algebrallinen summa, Einsteinin summa, Hamacherin summa ja rajoitettu summa. Alustavassa testauksessa huomattiin, että muut  $t$ -normit kuin minimi antavat hyvin matalia suositusten jäsenyysasteita, joten testeissä ei käytetä muita  $t$ -normeja.  $T$ -normien ja  $t$ -konormien vaikutus menetelmään kannattaa selvittää kokeellisesti myös siksi, koska niiden toiminnasta on varsin vähän kokeellisia tuloksia [26].

Kaikkiaan parametrien yhdistelmistä muodostuu 16825 parametrien joukkoa. Parhaan joukon valitseminen näiden joukosta on varsin vaativaa. Parametrijoukoille lasketaan suositusten hyvyttä kuvaava vertailuluku seuraavalla menetelmällä: suositukset jaetaan ryhmiin niiden korkeimman jäsenyysasteen mukaan. Ryhmät muodostetaan 0,05 jäsenyysasteen välein siten, että suositus kuuluu ryhmään, jos sen korkein jäsenyysaste on vähintään tämän arvon verran. Poikkeuksena ryhmä 0,0, johon kuuluvat kaikki jonkin positiivisen jäsenyysasteen saaneet suositukset. Nollan saaneita suosituksia ei siis oteta suositustarkkuuden laskennassa huomioon. Jokaiselle ryhmälle lasketaan RMSE-arvo ja kattavuus, jotka normalisoidaan ryhmän korkeinta ja matalinta arvoa käyttäen välille nolasta yhteen. Normalisoitu RMSE-arvo vähennetään arvosta 1 ja kerrotaan normalisoidulla suositusten kattavuudella. Lopuksi kaikkien ryhmien näin lasketut arvot summataan yhteen parametrijoukon lopulliseksi hyvyttä ilmaisevaksi arvoksi. Tämän menettelytavan tarkoituksena on löytää ne parametrien joukot, jotka antavat hyviä suosituksia korkealla kattavuudella laajalla alalla jäsenyysasteita.

#### 5.4.2 Suositusjärjestys

Kehitetyllä menetelmällä ja verrattavilla menetelmillä lasketut suositukset järjestetään paremmuusjärjestyksen, ja järjestyksille lasketaan virhe yhtälöllä 27 . Käyttäjien suositusjärjestyksien virheet lasketaan yhteen kyseisen menetelmän suositusjärjestyksen kokonaisvirheeksi.

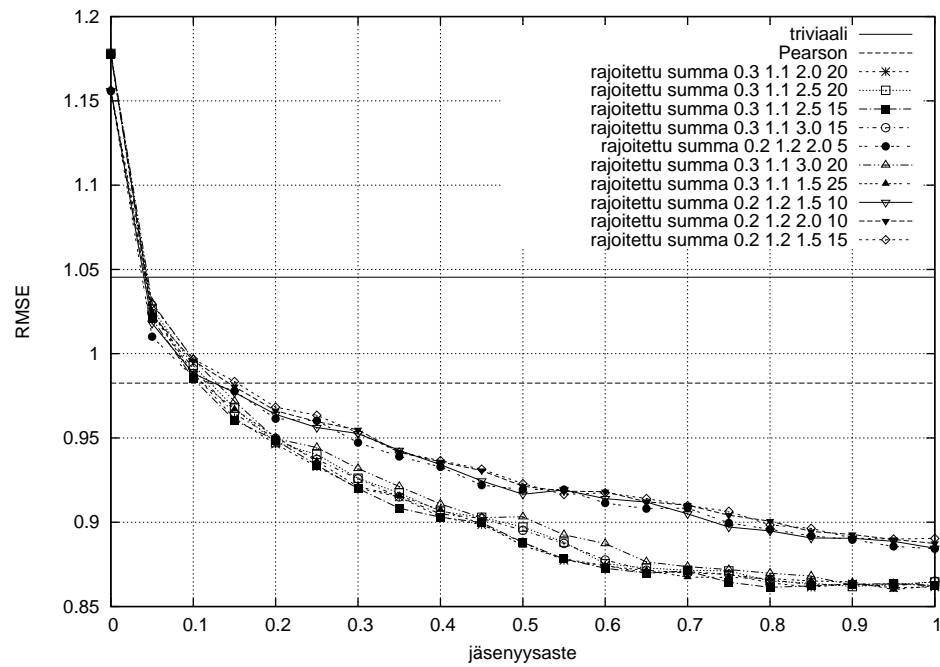
## 6 KOKEIDEN TULOKSET

Tässä kappaleessa esitellään edellisessä kappaleessa kuvattujen kokeiden tulokset. Ensiksi etsitään kaikista parametrivaihtoehdoista paras yhdistelmä. Tämän jälkeen tutkitaan, miten eri parametrien arvot vaikuttavat kehitetyn suositusmenetelmän toimintaan.

### 6.1 Parhaat parametrien arvot

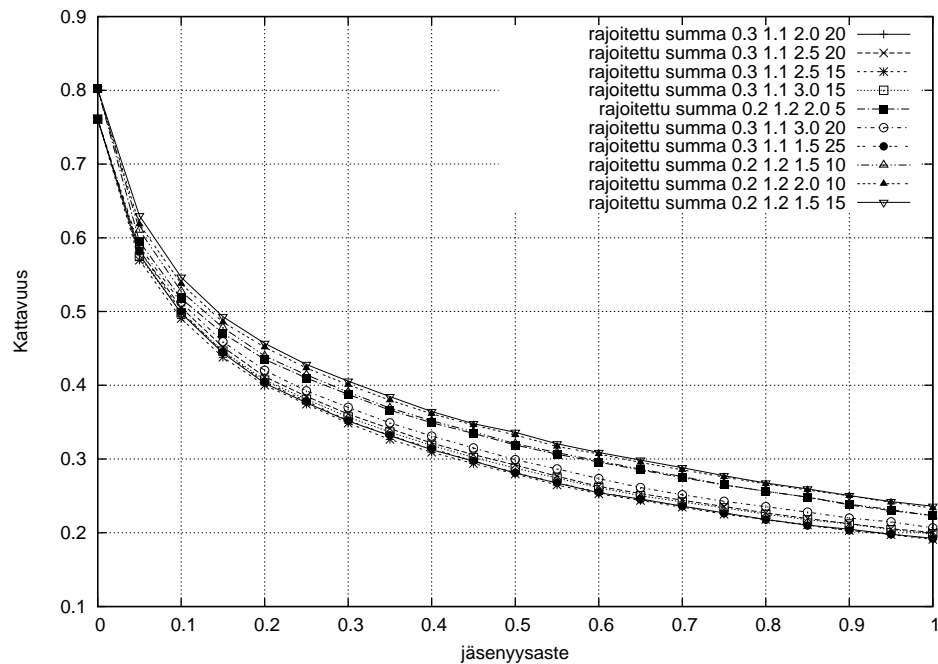
Parhaan parametriyhdistelmän valitseminen mahdollisten yhdistelmien joukosta ei ole täysin yksiselitteistä. Kuten useissa muissakin suositusmenetelmissä, joudutaan tekemään valinta suositusten kattavuuden ja tarkkuuden välillä. Suositustarkkuutta voidaan siis parantaa pienentämällä kattavuutta ja päin vastoin. Sumeat suositukset tuovat vielä lisää haastavuutta parhaan parametriyhdistelmän valintaan. Järjestelmälle asetetut vaatimukset vaikuttavat myös parametriyhdistelmän valintaan. Kuvassa 1 esitetään kymmenen parhaan parametrijoukon suositustarkkuus suhteessa jäsenyysasteeseen. Kuvaan on myös piirretty Pearsonin korrelaatioon perustuvan suositusmenetelmän ja triviaalin suositusmenetelmän suositustarkkuudet. Ne molemmat antavat suositukset terävinä lukuina, joten niiden antamia suosituksia ei voida jakaa jäsenyysasteen mukaan. Jäsenyysasteakselilla on suositukset, joiden korkein arvosana jäsenyysaste on vähintään annetun arvon verran. Suosituksia, joiden jokaisen arvosanan jäsenyysaste on nolla, ei ole otettu mukaan. Täyden arvosanan jäsenyysasteen saaneet suositukset ovat kuvan oikeassa laidassa.

Kuvasta voidaan havaita, että kehitetyn suositusjärjestelmän tarkkuus ylittää triviaalin toteutuksen suositustarkkuuden, kun korkein jäsenyysaste on hieman alle 0,05. Pearsonin korrelaation perustuvan menetelmän suositustarkkuuden järjestelmä ylittää, kun jäsenyysaste on, parametreista riippuen, jossain 0,1 ja 0,15 välillä. Kehitetty järjestelmä antaa tämän kokeen mukaan hyvin huonoja suosituksia, kun korkeimman jäsenyysasteen arvo on lähellä nollaa. Korkeimman jäsenyysasteen ollessa lähellä yhtä, kehitetyn suositusmenetelmän antamat suositukset ovat selvästi tarkempia kuin vertailtavien menetelmien antamat suositukset.



Kuva 1: Kymmenen parasta kehitetyn menetelmän parametrijoukkoa.

Kuvasta 1 voidaan havaita, että eri parametreilla suositustarkkuudet jakautuvat kahteen ryhmään. Myös kattavuutta esittävässä kuvassa 2 eri tavalla parametroidut suositusmenetelmät jakautuvat kahteen ryhmään, tosin ei yhtä selvästi kun suositustarkkuutta esittävässä kuvassa. Parametriryhmät, jotka kuuluvat heikomman suositustarkkuuden ryhmään, kompensoivat sitä jossain määrin korkeammalla suositusten kattavuudella.



Kuva 2: Kymmenen parhaan kehitetyn menetelmän parametrijoukon suositusten kattavuus suosituksen korkeimman jäsenyysasteen funktiona.

Taulukossa 6 on lueteltuina kymmenen parasta parametrijoukkoa ja taulukossa 7 kymmenen huonointa parametrijoukkoa. Tuloksista voidaan päätellä, että vastaavuuden kerroin kannattaa pitää alhaisena. Eksponentti kannattaa asettaa korkeaksi, tosin myös huonoimpien parametrijoukkojen eksponentti on korkea. Matalan eksponentin arvon voidaan olettaa siis antavan keskitasoa olevia suosituksia.

Tyylilajibonus kannattaa asettaa arvoon 1,5 ja 3,0 väliltä. Huonoimmilla parametrijoukoilla bonuksen arvo on selvästi parhaita matalampi. Bonusta ei kuitenkaan ilmeisesti kannata asettaa liian korkeaksi.

Sisääntulon ja säännön yhteisten elokuvien määrän rajaksi kannattaa asettaa kohtuullisen korkea arvo, tosin myös joillain huonoimmilla parametrijoukoilla raja on korkea.

T-konormiksi kannattaa selvästi valita rajoitettu summa. Hamacherin summa näyttäisi puolestaan antavan huonoimpia tuloksia. Kertoimen ja t-konormin valinnalla näyttäisi olevan eniten merkitystä suositusten hyvyyden kannalta. Seuraavissa luvuissa tarkastellaan tarkemmin kunkin parametrin vaikutusta menetelmän antamiin suosituksiin.

Taulukko 6: Parhaat parametrien arvot.

t-konormi	Kerroin	Eksponentti	Bonus	Raja	Vertailuarvo
rajoitettu summa	0,3	1,1	2,0	20	4,11092163317132
rajoitettu summa	0,3	1,1	2,5	20	4,10484699906578
rajoitettu summa	0,3	1,1	2,5	15	4,10236868656265
rajoitettu summa	0,3	1,1	3,0	15	4,09578922232539
rajoitettu summa	0,2	1,2	2,0	5	4,09476912322286
rajoitettu summa	0,3	1,1	3,0	20	4,09316452896311
rajoitettu summa	0,3	1,1	1,5	25	4,08935980649806
rajoitettu summa	0,2	1,2	1,5	10	4,08905766839132
rajoitettu summa	0,2	1,2	2,0	10	4,08830047328174
rajoitettu summa	0,2	1,2	1,5	15	4,08754938400824

Taulukko 7: Huonoimmat parametrien arvot.

t-konormi	Kerroin	Eksponentti	Bonus	Raja	Vertailuarvo
Hamacherin summa	1,0	1,2	0,0	1	0,00105500892082821
Hamacherin summa	1,0	1,2	0,0	5	0,00895512577927658
Hamacherin summa	1,0	1,2	0,0	10	0,0398941461632811
Hamacherin summa	1,0	1,2	0,5	1	0,0436449962088046
Hamacherin summa	1,0	1,2	0,5	5	0,0526321191550553
Hamacherin summa	1,0	1,2	0,0	15	0,07335069093587
Hamacherin summa	1,0	1,2	1,0	1	0,0805751962613309
Hamacherin summa	1,0	1,1	0,0	1	0,0810720809255789
Hamacherin summa	1,0	1,2	0,5	10	0,0838244333151157
Hamacherin summa	1,0	1,1	0,0	5	0,0875234192836349

## 6.2 T-konormien vertailu

Edellisen kappaleen taulukosta 6 voidaan havaita, että paras t-konormi menetelmän kannalta on rajoitettu summa. Taulukossa 8 parametrijoukkojen vertailuluvut on jaoteltu t-konormin mukaan. Taulukosta voidaan havaita, että rajoitettu summa erottuu varsin selkeästi parhaana t-konormina tähän käyttötarkoitukseen. Seuraavaksi paras t-konormi on Einsteinin summa, joka on varsin tasaväkinen algebrallisen summan kanssa. Hamacherin summa puolestaan antaa selkeästi muita t-konormeja heikompia tuloksia. Menetelmä näyttää toimivan parhaiten t-konormien kanssa, jotka saavat korkeita arvoja aggregoituina usean pienen jäsenyysasteen yli. Kehitetyn menetelmän kannalta tämä tarkoittaa, että useat pienen saman arvosanan jäsenyysasteen tuottavat säännöt vaikuttavat paljon suosituksen tulokseen. Parhaat suositukset eivät siis perustu vain muutaman säännön tuottamiin jäsenyysasteisiin.

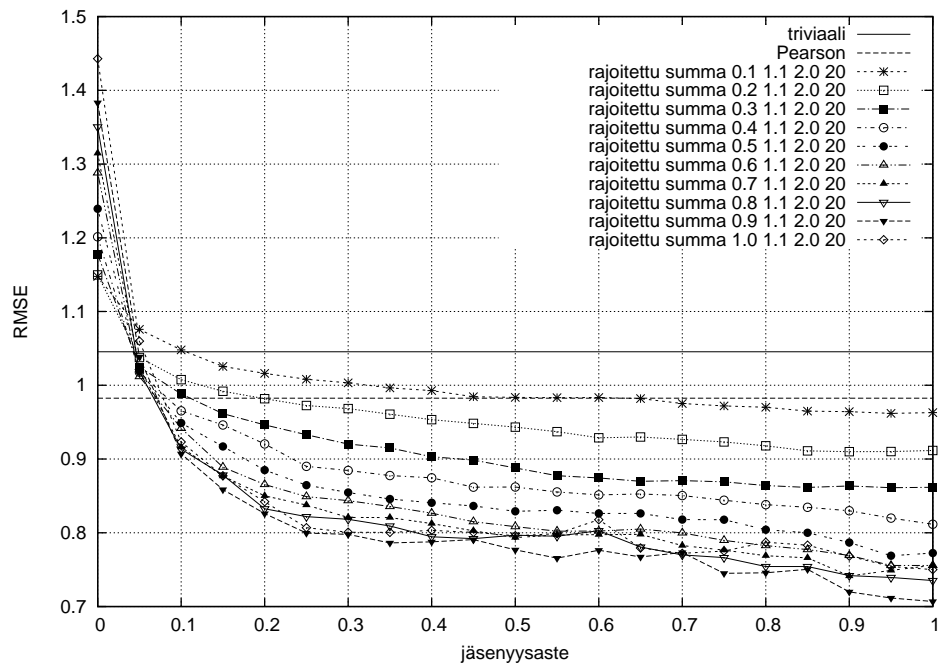
Taulukko 8: T-konormien vertailuluvut eri parametrijoukoille.

	algebrallinen summa	Einsteinin summa	Hamacherin summa	rajoitettu summa
maksimi	3,83628043239	3,8718504445	3,48478393944	4,11092163317
75. persentiili	1,712014662983	1,90118317256	1,355566757071	2,12393068934
mediaani	1,22931522992	1,383831650675	0,936716087615	1,531247478697
25. persentiili	0,838924774382	0,970645150859	0,62172977584	1,11680765287
minimi	0,129307351352	0,233407600846	0,00105500892082	0,305727499593

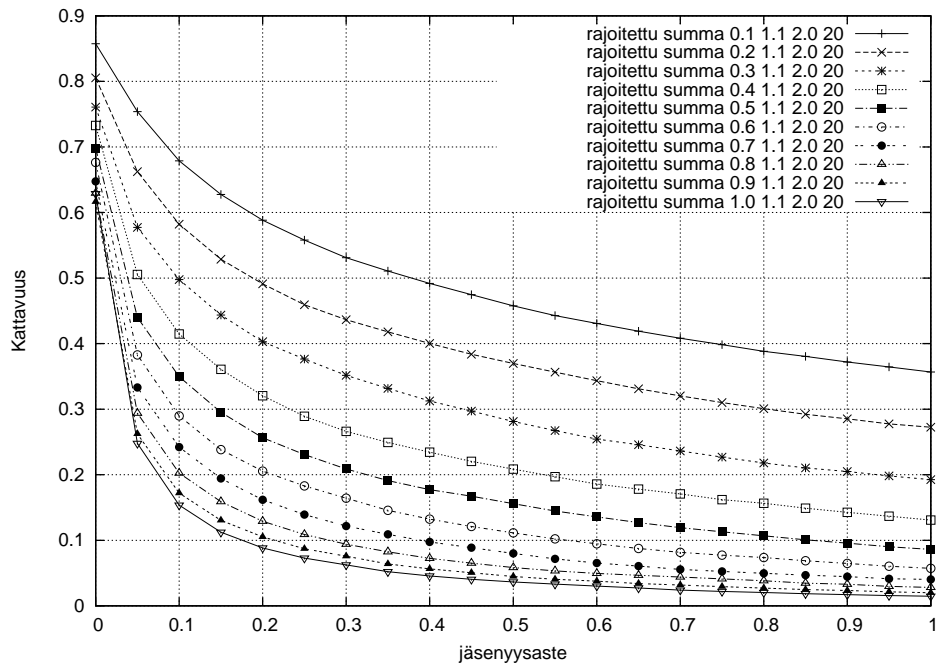
### 6.3 Suosituksen vastaavuuden kerroin

Suosituksen vastaavuus tiedettyyn arvosanaan nähden lasketaan yhtälön 23 avulla. Yhtälön parametrin  $m$  vaikutus suositustarkkuuteen on nähtävissä kuvassa 3. Kuva 4 puolestaan esittää parametrin vaikutusta suositusten kattavuuteen. Matala parametrin arvo vähentää suositustarkkuutta, mutta samalla kasvattaa suositusten kattavuutta. Toisaalta korkeat parametrin arvot parantavat suositustarkkuutta, mutta samalla vähentävät suositusten kattavuutta. Kuvissa lähtökohtana on aiemmin parhaaksi arvioitu parametrijoukko, jonka suosituksen vastaavuuden kerrointa muutetaan muiden parametrien arvojen säilyessä samana. Näin voidaan havaita parametrin vaikutus lopputulokseen. Parhaassa parametrijoukossa muutettavan parametrin arvo on 0,3, joka kuuluu suositustarkkuudessa kuvan 3 keskijoukkoon. Parametrin arvo 0,3 kuuluu myös suositusten kattavuudessa keskijoukkoon kuvan 4 mukaan. Tämä voidaan selittää sillä, että parametrin arvon ollessa korkea, suosituksen vastaavuuden laskennassa rangaistaan paljon siitä, että suosituksella on korkea jäsenyysaste myös oikean arvosanan viereisellä arvosanalla. Tämä on varsin yleinen tilanne, sillä samanlaisen elokuvamaun omaavat käyttäjät voivat hyvinkin antaa yhden arvosanan toisistaan eroavia arvioita. Tällaisista tilanteista ei selvästi kannata rangaista liikaa, sillä muuten suositusten kattavuus kärsii.





Kuva 3: Kertoimen vaikutus suositusten tarkkuuteen.

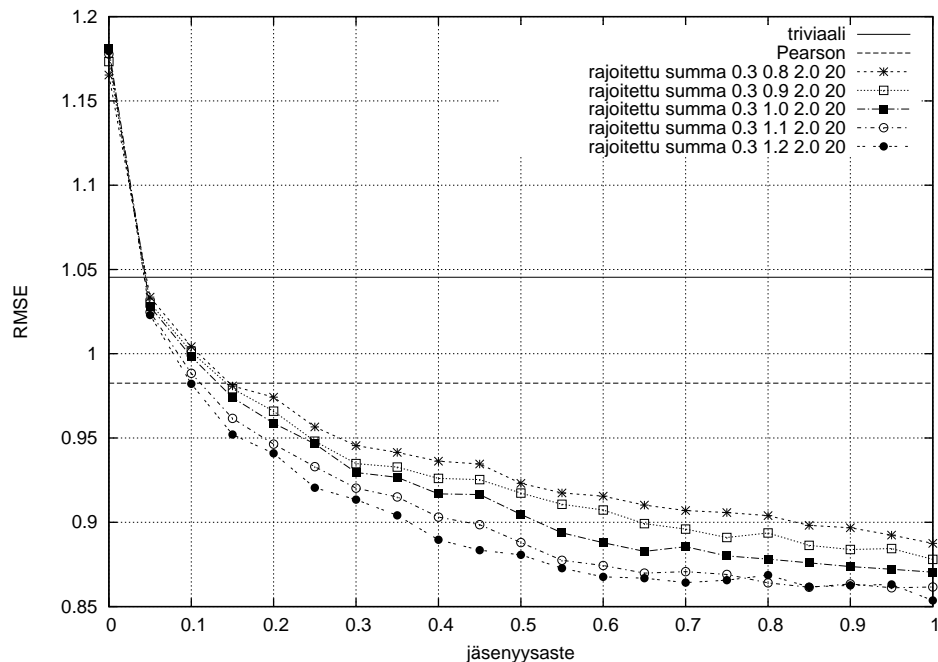


Kuva 4: Kertoimen vaikutus suositusten kattavuuteen.

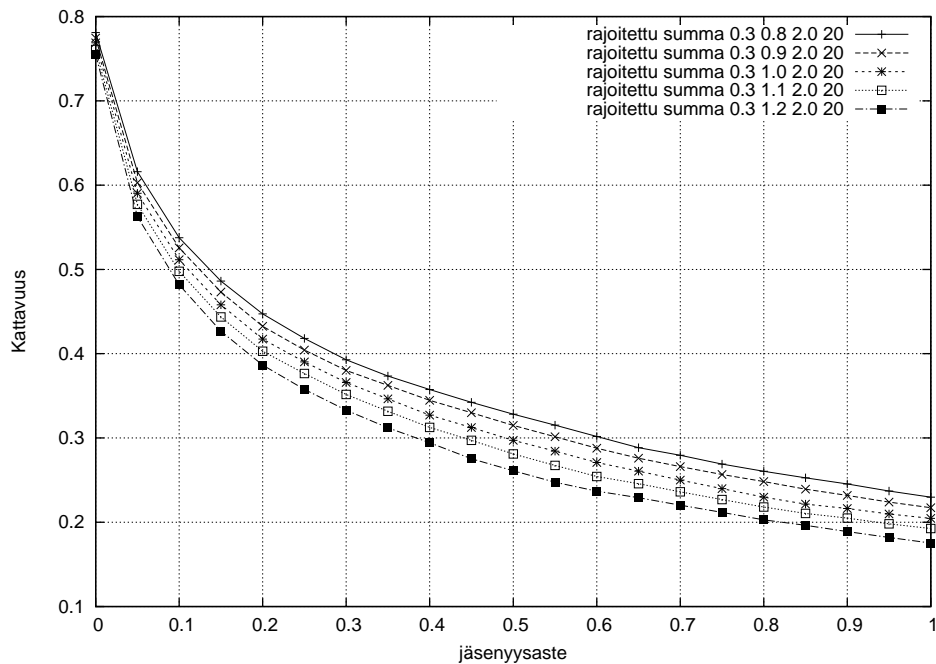
## 6.4 Suosituksen vastaavuuden eksponentti

Suosituksen vastaavuuden eksponentilla on hyvin samanlainen vaikutus suosituksiin kuin edellisessä luvussa esitetyllä kertoimellakin. Kuvista 5 ja 6 nähdään, että korkea eksponentin arvo antaa tarkempia suosituksia, mutta samalla laskee suositusten kattavuutta. Kertoimesta poiketen eksponentin kasvattaminen ei kuitenkaan vaikuta merkittävästi suositusten kokonaiskattavuuteen. Eksponentin korottamisen seurauksena tarkkojen suositusten suhde huonoihin suosituksiin heikkenee. Kokonaisuudessaan suositusten kattavuus pysyy silti lähes samana. Pieni kokonaiskattavuuden pieneneminen johtuu todennäköisesti siitä, että tietokone esittää laskennassa käytetyt luvut rajallisella määrällä bittejä, jolloin suosituksen jäsenyysaste voi päätyä nolllaksi, vaikka se ei sitä matemaattisessa mielessä olisikaan.

Parhaan eksponentin arvon valitseminen riippuu siitä, halutaanko painottaa suositustarkkuutta vai suositusten kattavuutta. Yleisesti voidaan kuitenkin sanoa, että korkeammat eksponentin arvot toimivat matalia arvoja paremmin.



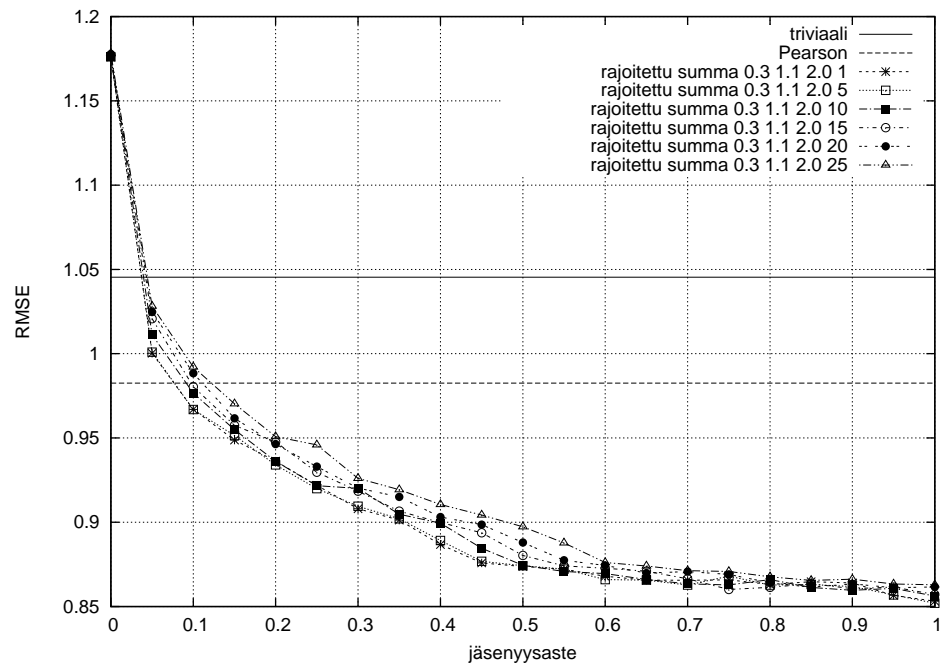
Kuva 5: Eksponentin vaikutus suositustarkkuuteen.



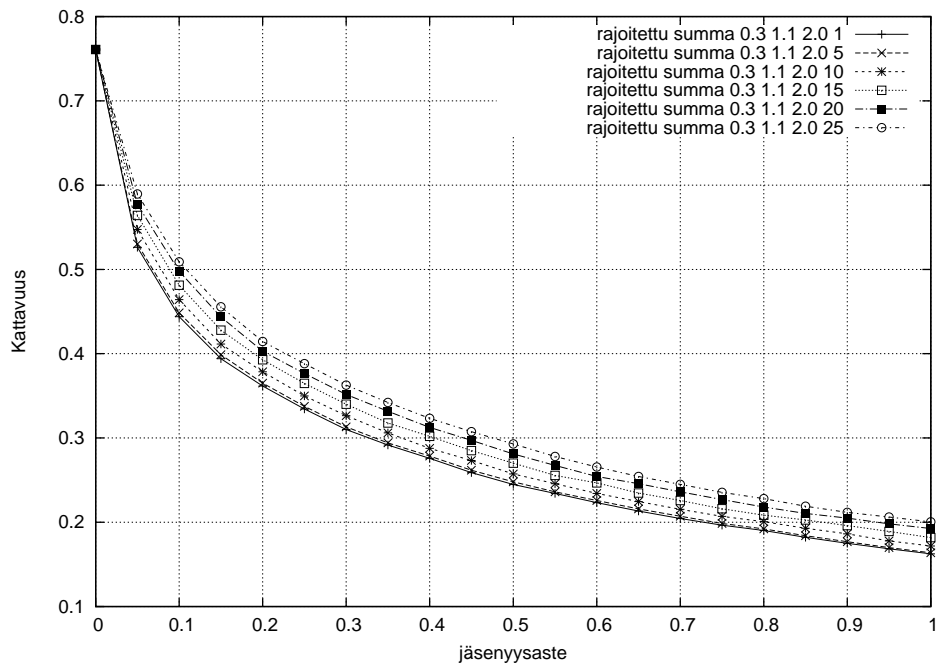
Kuva 6: Eksponentin vaikutus suositusten kattavuuteen

## 6.5 Merkittävyyden rajoittaminen yhteisten elokuvien määrän perusteella

Säännön merkittävyyden rajoittaminen molempien käyttäjien arvostelemien elokuvien määrän perusteella ei vaikuta kuvien 7 ja 8 perusteella kaikkien suositusten kokonaistarkkuuteen tai suositusten kattavuuteen. Se vaikuttaa suositustarkkuuteen heikentämällä korkean jäsenyysasteen saaneiden suositusten tarkkuutta, mutta samalla se nostaa korkean jäsenyysasteen saaneiden suositusten kattavuutta. Liian korkeaksi asetettu yhteisten elokuvien määrän raja vaikuttaa suosituksiin negatiivisesti, mutta matalaksi asetetulla rajalla saadaan parempia tuloksia kuin ilman rajaa. Kuten aikaisempienkin parametrien tapauksessa, on tämänkin parametrin paras arvo riippuvainen suositusjärjestelmältä vaadituista ominaisuuksista. Testidatan tapauksessa voidaan kuitenkin sanoa, että rajaksi kannattaa valita lähellä kymmentä oleva arvo.



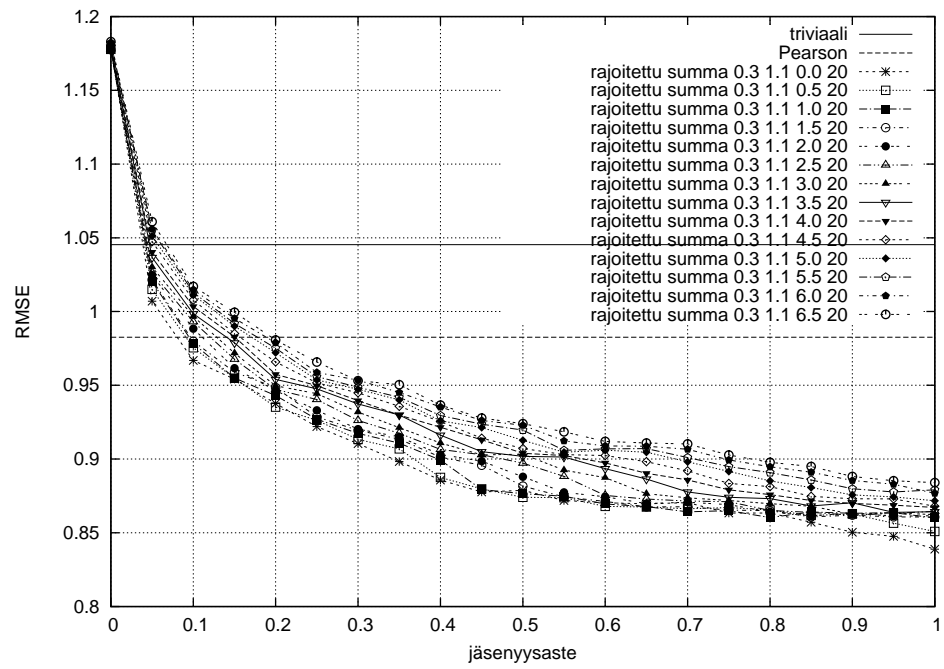
Kuva 7: Säännön merkittävyyden rajoittamisen vaikutus suositustarkkuuteen.



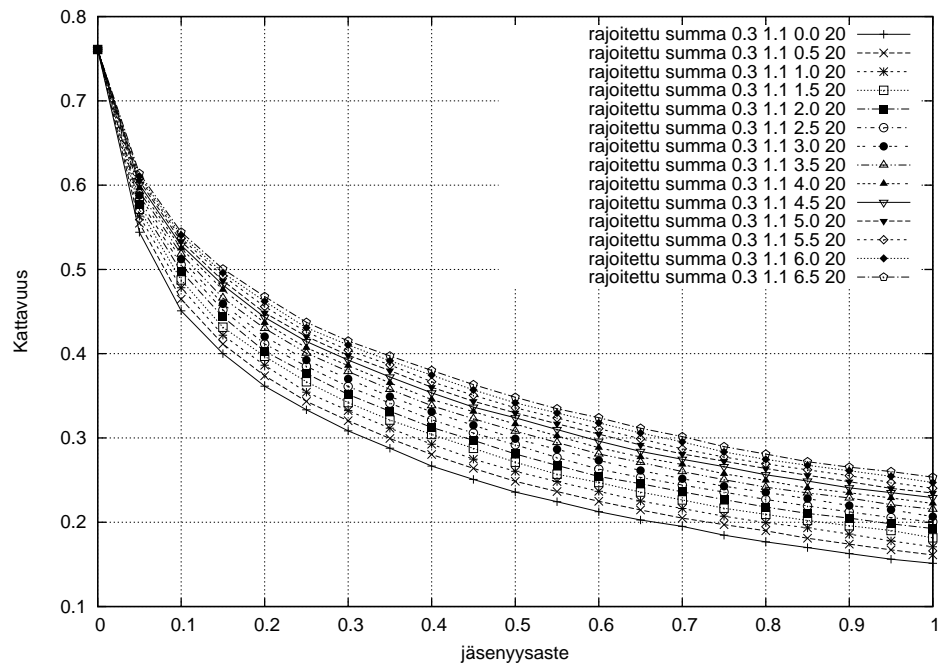
Kuva 8: Säännön merkittävyyden rajoittamisen vaikutus suositusten kattavuuteen.

## 6.6 Tyylilajibonuksen vaikutus

Kuvista 9 ja 10 nähdään, että tyylilajibonuksen arvo vaikuttaa suosituksiin nostamalla tarkkojen suositusten kattavuutta, mutta samalla huonontaa tarkkojen suositusten tarkkuutta. Pienillä bonuksen arvoilla kattavuuden kasvusta saatu hyöty on kuitenkin suurempi kuin tarkkuuden laskusta aiheutuva haitta, joten tyylilajibonuksen avulla voidaan parantaa suositusten laatua. Testidatalla tyylilajibonuksen arvot lähellä kahta antavat parhaita tuloksia. Muulla datalla paras bonuksen arvo on todennäköisesti hieman toinen, mutta kuitenkin lähes varmasti nollaa suurempi. Yhtälön 24 perusteella tyylilajibonuksen saadessa arvon kaksi, säännön ulostulo kerrotaan enintään kolmella, riippuen säännön ja syötteen tyylilajeista. Tyylilajitiedolla on siis varsin suuri merkitys säännön ulostuloon, sillä kahdesta samanlaisesta säännöstä, joista toinen kuuluu syötteen kanssa samaan tyylilajiin ja toinen ei, samaan tyylilajiin kuuluva antaa kolminkertaisen ulostulon jäsenyysasteen tyylilajiin kuulumattomaan nähden.



Kuva 9: Tyylilajibonuksen vaikutus suositustarkkuuteen.



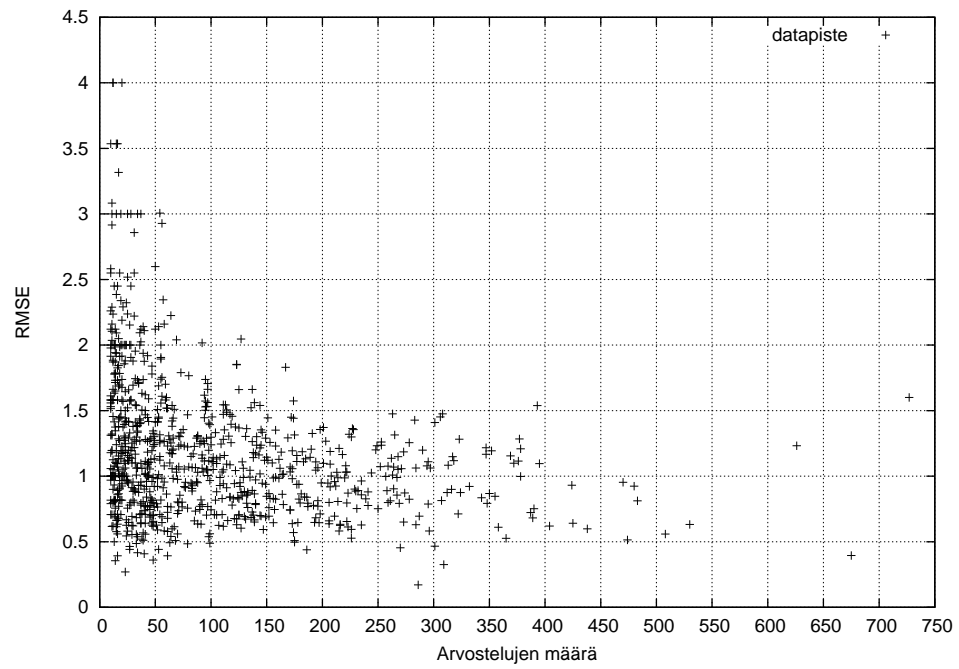
Kuva 10: Tyyllilajibonuksen vaikutus suositusten kattavuuteen.

## 6.7 Käyttäjän antamien arvioiden määrän vaikutus suosituksiin

Useita elokuvia arvostelleet käyttäjät saavat kuvan 11 perusteella selkeästi keskimääräistä parempia suosituksia kuin vähän elokuvia arvostelleet käyttäjät. Useita satoja elokuvia arvostelleet käyttäjät eivät kuitenkaan saa selkeästi parempia suosituksia kuin sadasta kahdeksataan elokuvaa arvostelleet käyttäjät. Käyttäjä voi saada tarkkoja suosituksia, vaikka hän olisikin arvostellut vain muutamia elokuvia. Arvioiden määrällä ei näytä olevan tietyn rajan jälkeen suurempaa vaikutusta suositusten tarkkuuteen. Suositusten tarkkuuden kannalta on todennäköisesti tärkeämpää vastaavan elokuvamaun omaavien käyttäjien löytyminen käyttäjien joukosta. Käyttäjän elokuvamaun selvittäminen näyttäisi vaativan toista sataa arviota, jonka jälkeen testidatalla ei ole havaittavissa merkittävää muutosta suositustarkkuudessa.

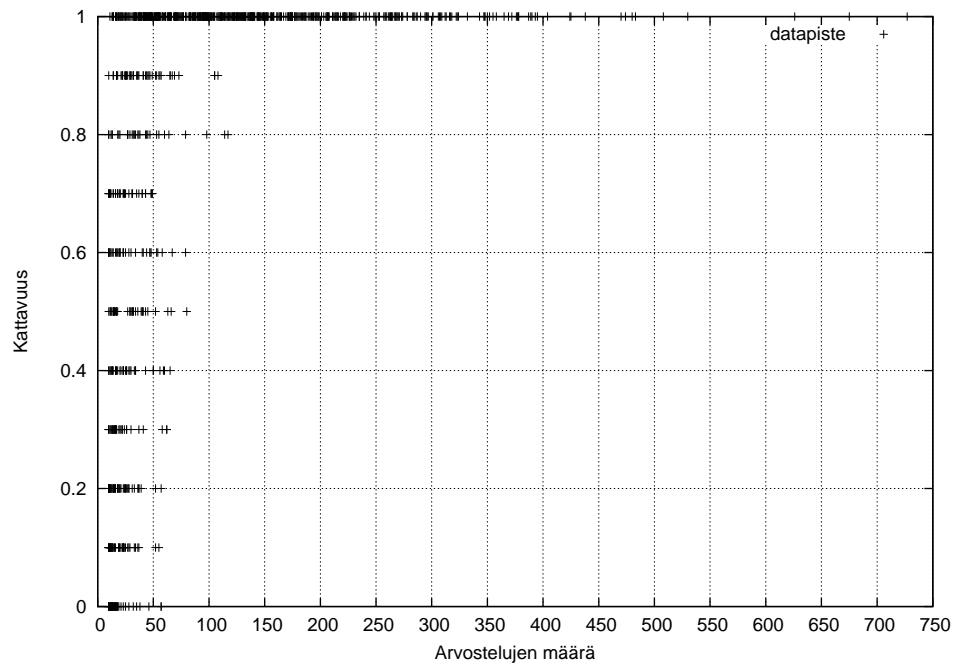
Kuvasta 12 nähdään, että suositusten kattavuus on joitain kymmeniä elokuvia arvostelneiden käyttäjien kohdalla vaihteleva. Joillekin voidaan antaa nollasta poikkeava suositus kaikille elokuville, kun taas toisille ei voida antaa yhtään nollasta poikkeavaa suositusta. Tämä johtunee siitä, että testidatan käyttäjien joukosta ei löydy käyttäjiä, joiden avul-

la voitaisiin aktiiviselle käyttäjälle antaa luotettavia suosituksia. Lähes kaikille enemmän kuin sata arvostelua antaneille käyttäjille voidaan antaa nollasta poikkeava suositus kaikille testidatan elokuville.



Kuva 11: Suositusten virhe suhteessa arvostelujen elokuvien määrään.





Kuva 12: Suositusten kattavuus suhteessa arvosteltujen elokuvien määrään.

## 6.8 Suositusjärjestys

Optimoimalla suositusjärjestyksen luvussa 4.9 esitetyllä tavalla tuotti taulukossa 9 esitetyn järjestysmatriisin. Suositusmenetelmän parametrien arvoina käytettiin aiemmin parhaaksi todettuja arvoja taulukosta 6.

Taulukko 9: Optimoitu järjestysmatriisi.

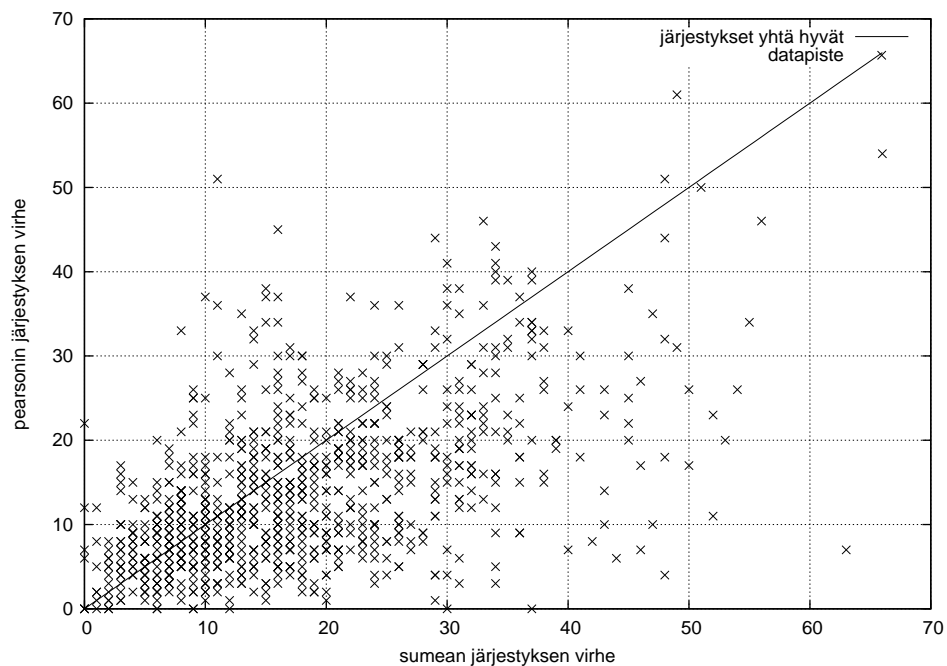
	1	2	3	4	5
1	-3,81	1,79	-1,42	1,2	-4,3
2	-9,18	-7,54	8,48	-2,73	2,51
3	-1,08	-2,06	-7,80	3,31	6,59
4	-3,40	1,95	-4,76	1,32	8,24
5	-7,41	-1,62	-3,36	2,34	3,02

Taulukossa 10 esitetään keskimääräiset järjestyksen virheet käyttäjille eri menetelmillä. Huomioitavaa on, että sumea suositus antaa muita menetelmiä huonompia suositusjärjestyksiä käyttäjille. Kuva 13 esittää Pearsonin korrelaatioon perustuvan menetelmän ja

kehitetyn menetelmän suositusjärjestyksien virheitä käyttäjiä kohden. Vaikka kehitetty menetelmä antaakin joissain tapauksissa parempia suositusjärjestyksiä, kokonaisuudessa se kuitenkin tekee enemmän virheitä. Se myös antaa varsin paljon suositusjärjestyksiä, joiden virhe on suuri, vaikka Pearsonin menetelmällä laskettujen suositusten suositusjärjestyksen virhe on kyseiselle käyttäjälle pieni.

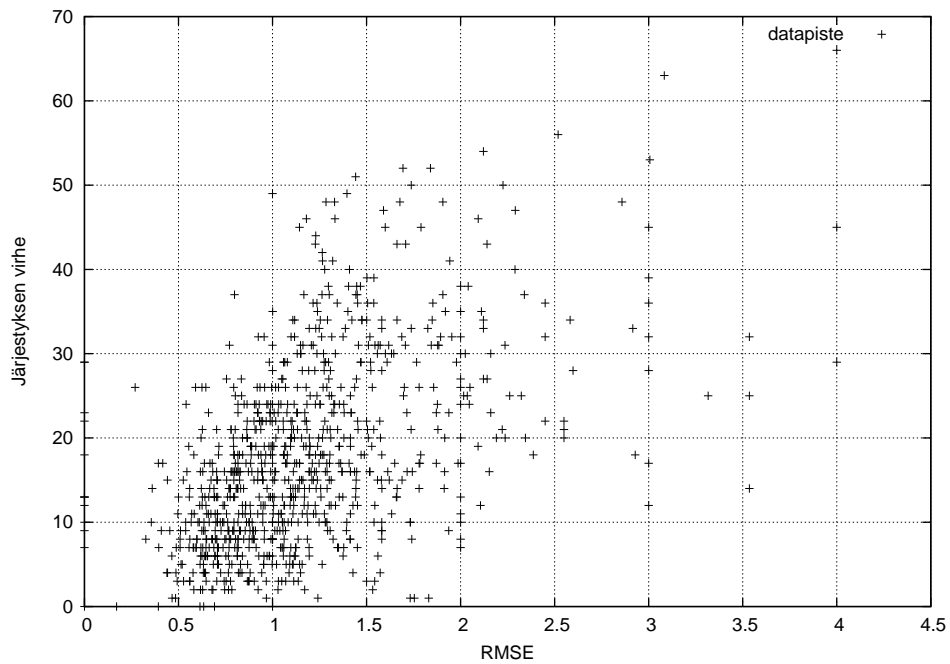
Taulukko 10: Keskimääräinen suositusjärjestyksen virhe käyttäjää kohti eri menetelmille.

menetelmä	käyttäjän keskimääräinen suositusjärjestyksen virhe
Pearson	13,97
triviaali	14,87
sumeaa	17,62



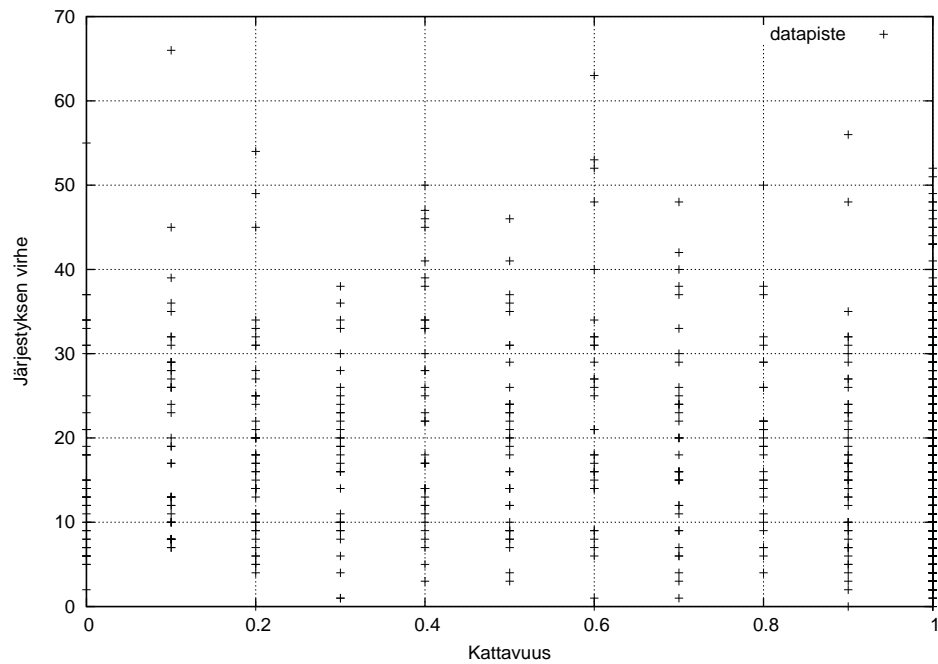
Kuva 13: Sumeaa järjestyksen virheen vertaaminen Pearsonin suositusten järjestyksen virheeseen.

Missä tapauksissa kehitetty menetelmä antaa huonoja suositusjärjestyksiä? Kuva 15 esittää käyttäjän suositusten RMSE-arvon suhdetta käyttäjän suositusjärjestyksen virheeseen. Kuvasta havaitaan, että suositusjärjestyksen virhe kasvaa RMSE-arvon kasvaessa, eli suositusjärjestyksen tarkkuus heikkenee suositustarkkuuden heikentyessä. Tämä on odotettavaa. Suositusjärjestyksien tulisi siis parantua, jos koko järjestelmän suositustarkkuutta voidaan parantaa.



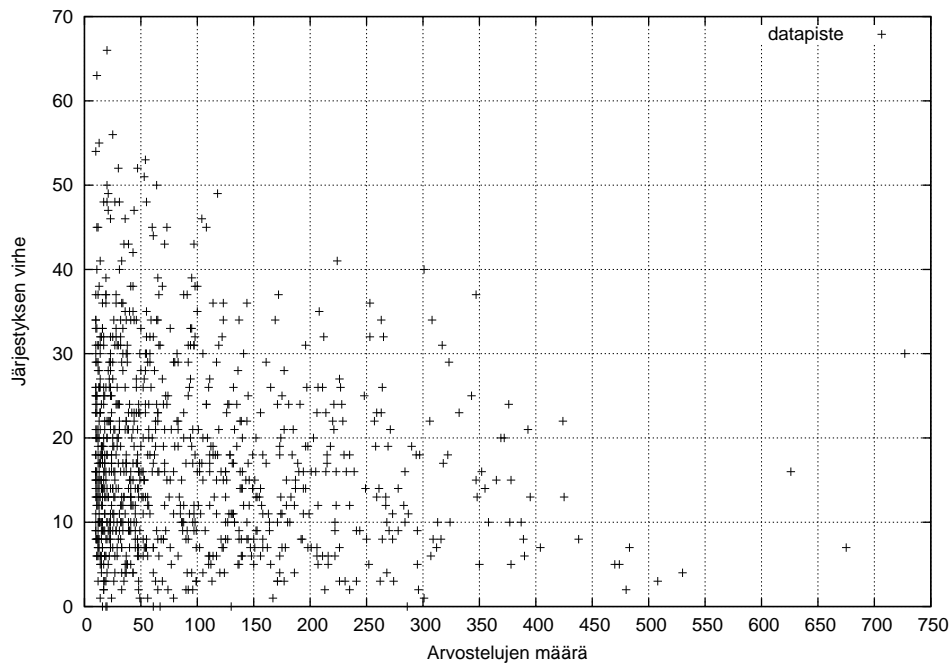
Kuva 14: suositusten virheen suhde järjestyksen virheeseen.

Kuva 15 esittää suositusten kattavuuden suhdetta suositusjärjestyksen virheeseen. Suositusten kattavuudella ei näytä olevan kovinkaan suurta merkitystä suositusjärjestyksen virheeseen. Tämä on jokseenkin yllättävää. Asian voi selittää se, että saman suositusarvon saaneet elokuvat järjestetään kyseisen elokuvan arvosanojen keskiarvon mukaan. Tällöin kattavuuden ollessa nolla suositusjärjestys määräytyy täysin elokuvien arvosanojen keskiarvon mukaan. Taulukosta 10 nähdään, että triviaali suositusmenetelmä, joka perustuu arvosanojen keskiarvoon, antaa varsin hyviä suositusjärjestyksiä. Suositusten kattavuus ei siis vaikuta suositusjärjestyksen virheeseen, koska matalan kattavuuden tapauksessa käytetty varajärjestelmä antaa vastaavan tasoisia suositusjärjestyksiä varsinaiseen menetelmään nähden.



Kuva 15: Kattavuuden suhde järjestyksen virheeseen.

Kuva 16 esittää käyttäjän antamien arvostelun määrän suhdetta suositusjärjestyksen virheeseen. Suositusjärjestyksen virhe näyttää pienenevän arvostelujen määrän kasvaessa. Tämä käy yhteen kuvien 14 ja 11 kanssa, sillä suositustarkkuus paranee arvostelujen määrän kasvaessa, ja suositusjärjestyksen virhe pienenee suositustarkkuuden kasvaessa. On siten oletettavaa, että suositusjärjestyksen virhe pienenee arvostelujen määrän kasvaessa.



Kuva 16: Arvioiden määrän suhde järjestyksen virheeseen.

Kehitetyn menetelmän suositusjärjestyksien virhe on selkeästi vertailtuja menetelmiä korkeampi. Tämä näyttäisi johtuvan pääosin menetelmän suositustarkkuuden heikkoudesta tietyissä tilanteissa. Jos kehitettävän menetelmän suositustarkkuutta voidaan parantaa, myös suositusjärjestyksien tarkkuus paranee. Kehitetty sumeiden suositusten suositeltavuusjärjestyksen laskeminen näyttäisi testien mukaan toimivan kohtuullisesti, mutta siinäkin on parannettavaa. Kehitetty järjestämismenetelmä antaa kuitenkin hyvän lähtökohdan mahdolliselle jatkokehitykselle.

## 6.9 Suoritusajoista

Kokeet suoritettiin tietokoneella, jossa on neljä gigitavua muistia ja suorittimena AMD Athlon II 605e. Menetelmä toteutettiin Clojure-ohjelmointikielellä, ja ohjelmaa ajettiin OpenJDK-virtuaalikoneella. Kaikkien opetusdatan painojen optimointi kesti vähiten arvosanoja antaneiden käyttäjien tapauksessa hieman yli sekunnin. Eniten arvosanoja antaneen käyttäjän tapauksessa kaikkien painojen optimointi kesti noin kymmenen minuuttia. Vähiten arvosanoja antaneille käyttäjille suositusten laskeminen kesti noin 0,2 sekuntia.

Eniten arvosanoja antaneelle käyttäjälle suositusten laskeminen kesti noin 1,5 sekuntia. Suositusten laskeminen on siis painojen optimointiin nähden varsin nopeaa.

Käytetty ohjelmointialusta ei ole erityisen nopea, eikä suoritusaikojen optimointiin käytetty kovinkaan paljon aikaa. Käytetyn tietokoneen suorituskykyään ei ole erityisen suuri. Suoritusajoja voidaan todennäköisesti varsin helposti parantaa tässä esitetyistä luke-  
mista.

## 6.10 Tulosten yhteenveto

Yleisesti voidaan sanoa, että tulokset ovat lupaavia. Parantamisen varaa on erityisesti vähän arvosteluja antaneiden käyttäjien tapauksessa. Käyttäjä on kuitenkin tietoinen heikoista suosituksista arvosanojen jäsenyysasteiden ansioista, sillä korkean arvosanan jäsenyysasteen saaneet suositukset ovat selkeästi tarkempia kuin matalan jäsenyysasteen saaneet suositukset. Käyttäjä on siis tietoinen jokaisen suosituksen luotettavuudesta.

Helpointa menetelmässä havaitut heikkoudet olisi korjata yhdistämällä se esimerkiksi Pearsonin korrelaatioon perustuvaan menetelmään. Pearsonin korrelaatioon perustuvan menetelmän antamia suosituksia voitaisiin käyttää korjaamaan matalan jäsenyysasteen saaneita suosituksia. Tällainen yhdistelmä tuottaisi todennäköisesti parempia suosituksia kuin kumpikaan menetelmä yksinään. Esimerkiksi hahmontunnistusjärjestelmillä, joita suositusjärjestelmät muistuttavat, on yleistä, että usean menetelmän yhdistäminen tuottaa yhdistelmän osatekijöitä parempia tuloksia.

Mielenkiinnon vuoksi tarkastellaan kahta esimerkkiä käyttäjän saamasta suosituksesta elokuvalla ”Pulp Fiction”. Erään käyttäjän saama suositus elokuvalla on [0,0; 0,0; 0,061; 0,115; 0,017]. Hänen oikeasti elokuvalla antama arvosana on 5, joten suosituksessa on virhettä. Samainen käyttäjä on antanut elokuvalla ”Reservoir Dogs” arvosanaksi 4, mikä voi selittää osittain suosituksen arvon, sillä molemmilla elokuvilla on sama ohjaaja, ja ne ovat tyyliltään samanlaisia. Toinen käyttäjä sai samalle elokuvalla suositukseksi [0,31; 0,069; 0,29; 0,43; 0,38]. Käyttäjän todellinen arvosana elokuvalla on 4. Hän on antanut elokuvalla ”Reservoir Dogs” arvosanaksi 1, joka selittänee korkean jäsenyysasteen tälle arvosanalle. Suositus osoittaa selvästi jonkinlaista ristiriitaa datassa. Tästä huolimatta suositus osoittaa kuitenkin kokonaisuudessa lähelle oikeaa arvosanaa.

Sumeiden suositusten järjestäminen antaa kohtuullisia tuloksia. Sen antamat tulokset näyttävät olevan kuitenkin suurelta osin riippuvaisia suositusten tarkkuudesta, joten jär-

jestämismenetelmää kannattaa kehittää enemmän vasta, kun yleistä suositustarkkuutta on saatu parannettua.

## 7 YHTEENVETO JA JOHTOPÄÄTÖKSET

Suositusjärjestelmän tehtävänä on suositella käyttäjille asioita, joista he voisivat pitää. Asiat voivat olla esimerkiksi musiikkia, kirjoja, elokuvia, ravintoloita, oikeastaan mitä tahansa asioita, joiden keskinäistä samanlaisuutta voidaan jotenkin kuvata. Suositusmenetelmät voidaan jakaa karkeasti kahteen ryhmään: yhteisösuodatusmenetelmiin ja asioiden ominaisuuksiin perustuviin menetelmiin. Yhteisösuodatusmenetelmissä suositukset tehdään käyttäjien antamien arvioiden perusteella. Asioiden ominaisuuksiin perustuvissa menetelmissä suositukset tehdään asioiden ominaisuuksien perusteella. Elokuvien tapauksessa ominaisuuksia voivat esimerkiksi elokuvan tyylilaji ja elokuvassa esiintyvät näyttelijät. Ominaisuudet voidaan kerätä joko automaattisesti tai manuaalisesti. Manuaalinen ominaisuuksien kerääminen rajoittaa kuitenkin menetelmän käytettävyyttä. Useat suositusjärjestelmät yhdistävät molempiin ryhmiin kuuluvia menetelmiä parempien suositusten saamiseksi.

Sumealla logiikalla voidaan käsitellä epämääräisiä asioita ja vahaata dataa sisältäviä ongelmia. Käyttäjien elokuville antamat arvosanat voidaan käsittää tällaiseksi epämääräiseksi dataksi. Jos käyttäjä antaa elokuvalla arvosanaksi viisi, lukuväliltä yhdestä viiteen, se ei tarkoita tarkkaa lukua, vaan jotain sumeaa lukua, joka voidaan ymmärtää vaikka kuvauksilla ”noin viisi” tai ”sellainen viisi”. Sumeaa logiikkaa on hyödynnetty useissa erilaisiin käyttötarkoituksiin tehdyissä suositusjärjestelmissä lupaavin tuloksin.

Jos suositusjärjestelmään syötetty data on luonteeltaan sumeaa, on loogista antaa ulostulo myös sumeana vastauksena. Käyttäjä perustaa päätöksensä paljon muuhunkin kuin pelkästään suositusjärjestelmän suositukseen, joten annetun vastauksen kannattaa sisältää yhtä arvosanaa enemmän tietoa, jonka avulla käyttäjä voi tehdä päätöksensä. Suositusten antaminen arvosanojen sumeina jäsenyysasteina mahdollistaa monipuolisempien suositusten antamisen käyttäjälle. Tällaisella suosituksella voidaan ilmaista ristiriitaisuutta ja tietämättömyyttä.

Tässä työssä kehitettiin sumeaa logiikkaa hyödyntävä suositusmenetelmä, joka yhdistää asioiden ominaisuuksiin perustuvaa suositusta yhteisösuodatukseen perustuvaan suositusmenetelmään. Menetelmä antaa suositukset arvosanojen sumeina jäsenyysasteina, jolloin käyttäjälle annetaan enemmän tietoa päätöksen tekemiseksi. Suositusmenetelmän antamat suositukset voidaan myös perustella. Käyttäjälle voidaan esimerkiksi kertoa, että kyseinen elokuva sai tämän suosituksen, koska käyttäjä on antanut tietyn arvosanan jollekin toiselle elokuvalla.



Kokeiden perusteella kehitetty menetelmä antaa joissakin tapauksissa todella hyviä suosituksia, kun taas toisissa tapauksissa suositukset ovat todella huonoja. Keskimäärin menetelmä toimii kuitenkin huonommin kuin vertailussa käytetyt menetelmät, tosin sumeat suositukset jouduttiin terävöittämään vertailua varten, mistä saattaa seurata suositustarkkuuden heikkenemistä. Toisin kuin vertailtavissa menetelmissä, käyttäjä saa tiedon suosituksen varmuudesta arvosanojen jäsenyysasteiden avulla, minkä takia vertailtavia menetelmiä heikompia suositustarkkuus ei anna oikeaa kuvaa suositusten hyödyllisyydestä. Kokeiden mukaan suositustarkkuus paranee suosituksen korkeimman jäsenyysasteen kasvaessa, joten käyttäjä pystyy välttämään huonoja suosituksia valitsemalla korkeita jäsenyysasteita saaneita suosituksia.

Kokeet kertovat myös, missä tapauksissa menetelmä toimii huonosti, jolloin jatkokehitys voidaan kohdentaa oikeaan paikkaan. Heikoiten menetelmä suoriutuu vähän arvosanoja antaneiden käyttäjien tapauksessa. Yksinkertainen tapa parantaa huonoja suosituksia olisi käyttää esimerkiksi Pearsonin korrelaatioon perustuvaa menetelmää huonojen suositusten arvosanojen jäsenyysasteiden korjaamiseen. Itse menetelmässä on myös monia parannusmahdollisuuksia, kuten esimerkiksi arvosanojen vastaavuuksien sumean relaatiosuhteen muuttaminen epäsymmetriseksi tai suosituksen laskeminen toiseen kertaan vertailemalla elokuvia keskenään käyttäjien sijaan. Näin saataisiin toiset suositukset, jotka voitaisiin yhdistää aikaisempiin suosituksiin suositustarkkuuden ja suositusten kattavuuden parantamiseksi. Tämä vaatisi tietysti lisää laskentatehoa ja tallennuskapasiteettia, mutta samalla se todennäköisesti parantaisi suosituksia varsinkin vähän arvosteluja antaneiden käyttäjien tapauksessa. Myös vastaavuuksien optimoinnista löytyy monia parannusmahdollisuuksia.

Sumeiden suositusten järjestämiseksi kehitetyn menetelmän tulokset ovat suositusten tasoon nähden hyviä. Suositusjärjestyksen virhe pienenee suositustarkkuuden mukana, joten suositustarkkuutta parantamalla voidaan samalla parantaa suositusjärjestyksen tarkkuutta. Kokonaisuutena kehitetty menetelmä on toimiva, vaikka siinä onkin selkeitä, osin kohtuullisen helposti korjattavissa olevia, heikkouksia.

## LÄHTEET

- [1] Patrice Perny, Jean-Daniel Zucker, *Collaborative Filtering Methods based on Fuzzy Preference Relations*, Proceedings of the 4th Meeting of the Euro Working Group on Fuzzy Sets (EUROFUSE-SIC '99), 1999.
- [2] Netflix Prize Rules [viitattu 2.10.2010]. Saatavissa: <http://www.netflixprize.com/rules>
- [3] Netflix prize homepage [viitattu 16.10.2010]. Saatavissa: <http://www.netflixprize.com/index>.
- [4] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedel: *Explaining Collaborative Filtering Recommendations*. Proceedings of the 2000 ACM conference on Computer supported cooperative work. 2000. Sivut 241 – 250. ISBN:1-58113-222-0.
- [5] Lotfi A. Zadeh, *Outline of a New Approach to the Analysis of Complex Systems and Decision Processes*. IEEE Transactions on Systems, Man and Cybernetics, sivut 28 - 44, vol. 3, num. 1, 1973. ISSN : 0018-9472.
- [6] Lotfi A. Zadeh, *Fuzzy Sets*, Information and Control, vol. 8, no. 3, sivut 338-353, 1965.
- [7] Jorma K. Mattila: *Sumean logiikan oppikirja*. Kolmas, uusittu painos, Helsinki: Art House Oy, 2002, ISBN 951-884-300-7
- [8] Patrice Perny, Jean-Daniel Zucker: *Preference-based Search and Machine Learning for Collaborative Filtering: the "Film-Conseil" Movie Recommender System*. Information , Interaction, Intelligence, Vol. 1(1):1-40, 2001.
- [9] Cornelis C, Guo X, Lu J, Zhang, *A Fuzzy Relational Approach to Event Recommendation*, Proceedings of the Second Indian International Conference on Artificial Intelligence (IICAI-05). Pune, India, sivut 2231–2242
- [10] Mohammad Yahya H. Al-Shamri, Kamal K. Bhaadwaj: *A Compact User Model for Hybrid Movie Recommender System*. International Conference on Computational Intelligence and Multimedia Applications, sivut 519-524, 2007. ISBN: 0769530508.
- [11] Cane Wing-ki Leung, Stephen Chi-fai Chan ja Fu-lai Chung: *A Collaborative Filtering Framework Based on Fuzzy Association Rules and Multpile-Level Similarity*. Knowledge and Information Systems, Vol. 10, Num. 3, 2006.

- [12] Stefano Aguzzoli, Paolo Avesani, Brunella Gerla: *A Logical Framework for Fuzzy Collaborative Filtering*. In Proceedings of the 10th IEEE International Conference on Fuzzy Systems, sivut 1043-1046, 2002. ISBN: 078037293X
- [13] Ronald R. Yager, *Fuzzy Logic Methods in Recommender Systems*, Fuzzy Sets and Systems, Elsevier, sivut 133-149, 2003.
- [14] DanEr Chen, YoLong Ying, SongJie Gong: *A Collaborative Filtering Algorithm Based on Rough Set and Fuzzy Clustering*. Fifth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 1, sivut 17 - 20, 2008. ISBN: 978-0-7695-3305-6
- [15] Katsuhiko Honda, Hidetomo Ichihashi, *Component-wise robust linear fuzzy clustering for collaborative filtering*. International Journal of Approximate Reasoning 37, num. 2, sivut 127–144, 2004. ISSN: 0888-613X.
- [16] Javier Carbo, Jose M. Molina, *Agent-based Collaborative Filtering Based on Fuzzy Recommendations*. Web Engineering and Technology, Inderscience, vol. 1, num. 4, sivut 414-426, 2004.
- [17] Javier Carbo, Jose M. Molina, *Avoiding Malicious Agents in E-Commerce using Fuzzy Recommendations*. Journal of Organizational Computing and Electronic Commerce. Vol. 17, num. 2, sivut 101-117, 2007.
- [18] Han-Saem Park, Ji-Oh Yoo, Sung-Bae Cho, *A Context-Aware Music Recommendation System Using Fuzzy Bayesian Networks with Utility Theory*, Fuzzy In Proceedings Third International Conference, Systems and Knowledge Discovery, sivut, 970-979, 2006.
- [19] Yukun Cao, Yunfeng Li, *An Intelligent Fuzzy-based Recommendation System for Consumer Electronic Products*, Expert Systems with Applications, vol. 33, sivut 230-240, Elsevier, 2007, ISSN: 09574174.
- [20] Nasraoui O, Petenes C, *An Intelligent Web Recommendation Engine Based on Fuzzy Approximate Reasoning*, Proceedings of the IEEE International Conference on Fuzzy Systems, vol. 2, sivut 1116–1121, 2003.
- [21] Chen CM, Duh LJ, Liu CY, *A Personalized Courseware Recommendation System Based on Fuzzy Item Response Theory*, Proceedings of 2004 IEEE e-Technology, e-Commerce and e-Service, sivut 305–308, 2004, ISBN: 0-7695-2073-1.

- [22] Lefteri H. Tsoukalas, Robert E. Uhrig, *Fuzzy and Neural Approaches in Engineering*, 1997 John Wiley & Sons, Inc. ISBN 0.471-16003-2
- [23] Rainer Storn, Kenneth Price: *Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces*. Technical Report TR-95-012, International Computer Science Institution, 1995.
- [24] Jon Herlocker, Joseph A. Konstan, John Riedl, *An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms*, Information Retrieval, vol. 5, num. 4, sivut 287-310, Springer, 2002, ISSN: 13864564.
- [25] Yehuda Koren, *The BellKor Solution to the Netflix Grand Prize*, 2009. [viitattu 20.5.2012]. Saatavissa: [http://www.netflixprize.com/ruleswww.netflixprize.com/assets/GrandPrize2009\\_BPC\\_BellKor.pdf](http://www.netflixprize.com/ruleswww.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf)
- [26] Kalle Saastamoinen, *Many values algebraic structures as measures of comparison*, dissertation, Lappeenrannan teknillinen yliopisto, Digipaino 2008. ISBN 978-952-214-645-8.