

*Model adaptivnega sistema za priporočanje učnih  
objektov v konstruktivističnem učnem okolju*

Uroš Ocepek

DOKTORSKA DISERTACIJA

PREDANA

FAKULTETI ZA RAČUNALNIŠTVO IN INFORMATIKO

KOT DEL IZPOLNJEVANJA POGOJEV ZA PRIDOBITEV NAZIVA

DOKTOR ZNANOSTI

S PODROČJA

RAČUNALNIŠTVA IN INFORMATIKE



Ljubljana, 2015



*Model adaptivnega sistema za priporočanje učnih  
objektov v konstruktivističnem učnem okolju*

Uroš Ocepek

DOKTORSKA DISERTACIJA

PREDANA

FAKULTETI ZA RAČUNALNIŠTVO IN INFORMATIKO

KOT DEL IZPOLNJEVANJA POGOJEV ZA PRIDOBITEV NAZIVA

DOKTOR ZNANOSTI

S PODROČJA

RAČUNALNIŠTVA IN INFORMATIKE



Ljubljana, 2015



## IZJAVA

*Izjavljam, da sem avtor dela in da slednje ne vsebuje materiala, ki bi ga kdorkoli predhodno že objavil ali oddal v obravnavo za pridobitev naziva na univerzi ali na drugem visokošolskem zavodu, razen v primerih, kjer so navedeni viri.*

— Uroš Ocepek —

oktober 2015

ODDAJO SO ODOBRILI

dr. Igor Kononenko

*redni profesor za računalništvo in informatiko*

*Univerza v Ljubljani, Fakulteta za računalništvo in informatiko*

PRESEDNIK OCENJEVALNE KOMISIJE

dr. Zoran Bosnić

*izredni profesor za računalništvo in informatiko*

*Univerza v Ljubljani, Fakulteta za računalništvo in informatiko*

MENTOR IN ČLAN OCENJEVALNE KOMISIJE

dr. Jože Rugelj

*izredni profesor za področje računalništvo v izobraževanju*

*Univerza v Ljubljani, Pedagoška fakulteta*

SOMENTOR IN ČLAN OCENJEVALNE KOMISIJE

dr. Irena Nančovska Šerbec

*docentka za področje računalništvo v izobraževanju*

*Univerza v Ljubljani, Pedagoška fakulteta*

ZUNANJA ČLANICA OCENJEVALNE KOMISIJE



## PREDHODNA OBJAVA

Izjavljam, da so bili rezultati obravnavane raziskave predhodno objavljeni/sprejeti v objavo v recenzirani reviji ali javno predstavljeni v naslednjih primerih:

- [1] U. Ocepek, Z. Bosnić, I. Nančovska Šerbec in J. Rugelj. Exploring the relation between learning style models and preferred multimedia types. *Computers & Education*, 69:343–355, 2013. doi: [10.1016/j.compedu.2013.07.029](https://doi.org/10.1016/j.compedu.2013.07.029)
- [2] U. Ocepek, J. Rugelj in Z. Bosnić. Improving matrix factorization recommendations for examples in cold start. *Expert Systems With Applications*, 42(19):6784–6794, 2015. doi: [10.1016/j.eswa.2015.04.071](https://doi.org/10.1016/j.eswa.2015.04.071)

Rezultati so bili predstavljeni tudi na treh mednarodnih konferencah.

Potrjujem, da sem pridobil pisna dovoljenja vseh lastnikov avtorskih pravic, ki mi dovoljujejo vključitev zgoraj navedenega materiala v pričujočo disertacijo. Potrjujem, da zgoraj navedeni material opisuje rezultate raziskav, izvedenih v času mojega podiplomskega študija na Univerzi v Ljubljani.





“Slovenci! /.../  
Za uk si prebrisane glave  
pa čedne in trdne postave.  
Išče te sreča,  
um ti je dan,  
našel jo boš, ak  
nisi zaspan.”

— Valentin Vodnik, *Dramilo*, 1795



— Ekipa Mozaik, *Mozaik STPŠ Trbovlje*, 2014



## POVZETEK

Smernice sodobnega učenja kažejo na razvoj prilagodljivih spletnih učnih okolij, ki prilagajajo učno vsebino učenčevim lastnostim in potrebam. Priporočilni sistemi lahko predstavljajo takšne sisteme, saj temeljijo na priporočanju ustreznih učnih objektov glede na uporabnikove lastnosti, kot so: učne karakteristike, predhodna ocenjevanja objektov in demografske lastnosti. Če za uporabnika nimamo začetnih ocen, pravimo, da je uporabnik v stanju hladnega zagona.

Tema disertacije je oblikovanje koncepta priporočilnega sistema, ki vsebuje nov pristop k reševanju problema hladnega zagona. Na splošno se pristopi za reševanje tega problema delijo na dve skupini: pristope, ki temeljijo na začetnem poizvedovanju o uporabnikovih lastnostih, in pristope, ki uporabljajo specifične algoritme za uporabnike, ki so v stanju hladnega zagona. Najprej smo oblikovali pristop, ki sodi v prvo skupino pristopov – začetno poizvedovanje. Z namenom, da bi učencem priporočili čim bolj ustrezna multimedijska gradiva, smo povezali različne učne stile s posameznimi multimedijskimi tipi. Povezali smo Kolbov model učnih stilov, Rancourtov model, hemisferični model in model učnih stilov VAK (vizualni, slušni in kinestetični stil). Zgradili in analizirali smo odločitveni model za priporočanje multimedijskega gradiva. Zgrajeni model je pokazal, da je hemisferični model učnih stilov najpomembnejši odločitveni atribut, Kolbov model in model VAK pa sta druga najpomembnejša odločitvena atributa. Raziskava je pokazala, da študentje za učenje najpogosteje uporabljajo besedila, ki so dobro strukturirana in imajo barvno diskriminacijo. Poleg besedil študentje pogosto uporabljajo animacije in videoposnetke. Z zgrajenim odločitvenim modelom smo tudi potrdili, da je za ustreznjšo izbiro in priporočanje različnih tipov multimedijskih gradiv smiselno kombinirati različne modele učnih stilov. Po drugi strani pa je model statičen, kar pomeni, da v vsaki situaciji priporoča enak multimedijski tip. To ni ustrezno, ker ni nujno, da je neki multimedijski tip ustrezen za vse učne teme.

Drugi del raziskovanja je bil namenjen oblikovanju in razvoju inovativnega pristopa, ki izboljša priporočanje uporabnikom v hladnem zagonu, in sicer tako, da uporabnikom vstavi samo nekaj ocen za izbrane objekte in nato nad celotno matriko izvede matrično faktorizacijo. Naš pristop je sestavljen iz treh korakov: 1) iskanje podobnih uporabnikov; 2) izbor ciljnih atributov; 3) agregacija pridobljene vrednosti v izbrane ciljne attribute. Reševanje hladnega zagona smo razdelili na reševanje problema absolutnega in delnega hladnega zagona. Glede na rezultate eksperimenta (reševanje problema absolutnega hladnega zagona) so rezultati pokazali, da predlagane metode v povprečju izboljšajo nenegativno matrično faktorizacijo s stohastičnim gradientnim spustom (NG). Pri polnenegativni matrični faktorizaciji z manjkajočimi vrednostmi (SN) pa kombinaciji FR-ME (v atribut, ki smo ga izbrali glede na frekvenco najpogostejše vrednosti, vstavimo povprečno vrednost vseh ocen v atributu) in SD-MF (v atribut, ki smo ga izbrali glede na vrednost standardnega odklona, vstavimo najpogostejšo vrednost atributa) bistveno izboljšata priporočanje. Pri preostalih dveh matričnih faktorizacijah (matrična faktorizacija z zlivanjem podatkovnih virov (DF) in nenegativna matrična faktorizacija z izmenjujočimi se najmanjšimi kvadrati (NS)) predlagane kombinacije ne izboljšajo priporočanja. V nadaljevanju smo naš pristop testirali za reševanje delnega hladnega zagona.

Statistična analiza eksperimentalnega dela (testiranja našega pristopa) na umetni množici je pokazala, da vsak posamezen parameter statistično značilno izboljša priporočanje metod matričnih faktorizacij. Najboljše delovanje so pokazale metode, pri katerih upoštevamo 25 % najbližjih sosedov (25-\*-\*-), uporabljamo frekvenco (\*-FR-\*-\*) ali RRelief (\*-RR-\*-\*) za izbiro atributov in vstavljamo vrednosti s povprečenjem (\*-\*-ME-\*) ali z uporabo regresijskega drevesa (\*-\*-RT-\*). V nadaljevanju smo izbrali varianti metod, za kateri smo pričakovali, da bosta dajali dobre rezultate: 25-FR-ME-\* in 25-RR-RT-\*. Izbranimi variantama metod smo dodali še kombinacijo, pri kateri upoštevamo 50 % najbližjih sosedov, za izbor atributov uporabimo RRelief in vstavljamo vrednosti z uporabo linearne regresije (50-RR-LR-\*) – kombinacija je bila glede na povprečni rang najboljša. Vse tri izbrane variante metod smo testirali na dveh realnih množicah: Jester in PEFbase. Evalvacija je pokazala, da vstavljanje manjkajočih vrednosti s predlagano metodo 25-FR-ME-\* in uporabo matrične faktorizacije NG značilno izboljša priporočanje objektov v primerjavi z golimi metodami matrične faktorizacije (DF, NG, NS in SN), in sicer za uporabnike v stanju delnega hladnega zagona. Preverili smo tudi vpliv vstavljanja manjkajočih ocen na druge uporabnike.

Ugotovili smo, da naše spremembe ne povzročajo značilnih sprememb pri priporočanju drugim uporabnikom, ki so zunaj stanja delnega hladnega zagona.

*Ključne besede:* vstavljanje manjkajočih vrednosti, problem novega uporabnika, matrična faktorizacija, priporočilni sistemi



University of Ljubljana  
Faculty of Computer and Information Science

Uroš Očepek

*A model of an adaptive system for recommending learning objects in a constructivist learning environment*

## ABSTRACT

Computer-based multimedia learning environments support the idea that people learn better and more deeply when appropriate pictures (i.e., animations, video, static graphics) are added to text or narration. There are many adaptive learning systems that adapt learning materials to student properties, preferences, and activities. Adaptive learning environments mostly support only traditional concepts of learning. There is a need to design and develop an e-learning system that embodies principles of constructivist learning approach. The solution is in recommenders systems, which suggest items of interest to users based on their preferences (i.e. previous ratings). If there are no ratings for a certain user or item/object, there is a situation called a cold start problem, which leads to unreliable recommendations. Researchers mostly avoid tackling the absolute cold start in recommender systems.

The topic of presented dissertation is designing a recommender system with a novel approach to avoid cold start problem. Approaches for solving the new user cold start problem can be divided into two main groups: the first group performs additional inquiries to gather more information about the users; and the second group uses dedicated algorithms for users in the cold start state.

The first group of approaches aims at performing additional inquiries about the user. According to this approach, we relate combinations of different learning styles (taking into account four different learning styles models) to preferred multimedia types. We explore a decision model aimed at proposing learning material of an appropriate multimedia type. The study includes 272 student participants. The resulting decision model shows that students prefer well-structured learning texts with colour discrimination, and that the hemispheric learning style model is the most important criterion in deciding student preferences for different multimedia learning materials. To provide a more accurate and reliable model for recommending different multimedia types more

learning style models must be combined. Kolb's classification and the VAK classification allow us to learn if students prefer an active role in the learning process, and what multimedia type they prefer. The results also shows that there is an obvious need to combine learning styles model in order to get a wider view of the student's characteristics: an approach to problem solving problems, cognitive modes, way of thinking, and a dominant mode of perceiving information. On another hand, model recommends same multimedia material regardless of the learning topic.

In the second part of our research, we have designed and developed a novel approach for alleviating the cold start problem by imputing missing values into the input matrix, thereby improving recommendation performance. Our approach has three steps: 1) finding similar users to given user in cold start state; 2) selecting relevant attributes for the imputation process; 3) aggregate ratings to input matrix for a user in the cold start state. We separate our approach for solving cold start problem into solving absolute cold start problem and solving partial cold start problem. According to the results of our experiments (solving absolute cold start problem), the results indicate that all our proposed methods improve recommending for non-negative matrix factorization with stochastic gradient descent (NG). For semi-non-negative matrix factorization with missing data (SN), combinations FR-ME (imputing attribute's mean value into the attributes that have the highest frequency of the most frequent values) and SD-MF (imputing attribute's most frequent value into attributes that have the lowest standard deviation) improve recommendations for users in the absolute cold start state. For non-negative matrix factorization with alternating least squares (NS) and matrix factorization by data fusion (DF), none of variations of proposed parameters (methods) improves recommending in absolute cold start state. In the next stage of our research, we evaluated our approach for solving partial cold start problem.

Statistical analysis of experimental evaluation of our approach on the artificial domain showed that each parameter significantly improved recommending of matrix factorization methods. The methods that yield improvements in recommendation accuracy compared with the raw matrix factorization are methods that consider 25 % of similar users (25-\*-\*-\*), select an attribute according to the frequency (\*-FR-\*-\*), or RReliefF (\*-RR-\*-\*), and impute a value aggregated by mean value (ME) or predicted by using regression trees (RT). For further investigation we chose two method combinations (25-FR-ME-\* and 25-RR-RT-\*), which were expected to work well, and compared them with other strategies on real domains. Among all approaches evalu-



ated on the artificial domain, we chose the best performing method with the highest average rank – a method that considers 50 % of similar users, selects an attribute for imputation according to the RRelieff, and imputes a value predicted by linear regression (50-RR-LR-\*). All three combinations of the selected methods were evaluated on two real domains: Jester in PEFbase. An evaluation showed that method 25-FR-ME-\* combined with matrix factorization NG performed statistically better than the raw matrix factorization algorithms (DF, NG, NS in SN) on real domains for users in the partial cold-start state. The results demonstrated the advantage of using imputation approaches in terms of better recommendation accuracy. At the same time, the results have shown that imputing of missing values has no negative impact for recommending to the users, which are not in the cold start state.

*Key words:* imputing missing values, new user problem, matrix factorization, recommender systems



## ZAHVALA

*Iskreno se zahvaljujem obema mentorjema za pomoč pri izdelavi doktorske disertacije. Hvala mentorju izr. prof. dr. Zoranu Bosniću za vso pomoč, konstruktivno kritiko in za vse odgovore ter številne komentarje na elektronska sporočila, ki sem jih bil deležen tudi ob koncih tedna, praznikih in ob dopustniških dnevih. Zahvaljujem se tudi mentorju izr. prof. dr. Jožetu Ruglju za vse strokovne in življenjske nasvete, podporo in za motivacijo pri nastajanju te naloge.*

*Zahvaljujem sem doc. dr. Ireni Nančovski Šerbec za vse kakovostne ideje in za vlivanje pozitivne energije, še posebno pri prvih raziskovalnih korakih. Hvaležen sem tudi dr. Juliji Bele Lapuh, doc. dr. Alenki Polak in prof. dr. Igorju Kononenku za vse kakovostne ideje ter strokovne nasvete pri pisanju znanstvenih prispevkov in nastajanju te naloge. Zahvaljujem se tudi Marinki Žitnik za pomoč pri raziskovanju področja matričnih faktorizacij.*

*Ravnateljici Marjetki Bizjak in vsem sodelavcem Srednje tehniške in poklicne šole Trbovlje se zahvaljujem, ker so me sprejeli v svoje delovno okolje. Hvala za razumevanje in podporo pri študiju. Zahvaljujem pa se tudi vsem svojim dijakom programa tehnik računalništva, ker so spodbujali svojega učitelja pri vztrajanju na poti do cilja.*

*Danili Jan, Simoni Izgoršek, Petri Žvelc in doc. dr. Tomažu Petku se najlepše zahvaljujem za angleško in slovensko lektoriranje znanstvenih prispevkov in disertacije.*

*Največjo podporo pri nastanku disertacije sem imel pri svoji ožji družini. Zahvaljujem se mami Mojci in očetu Jožetu za vso skrb in motivacijo skozi vso življenjsko pot — hvala, ker sta vztrajala pri svojih odločitvah in se nista ozirala na druga mnenja ter mi nudila vso podporo pri odraščanju in izobraževanju. Zahvaljujem se bratu Primožu, ki mi je brezpogojno pomagal, mi vlival moč in mi svetoval pri programerskih težavah. Hvala starim staršem in prastaršem za vse pozitivne misli in podporo. Zahvaljujem pa se tudi prijateljem in vsem drugim, ki so v tej zahvali krivično, a pomotoma izpuščeni.*

— Uroš Ocepek, Ljubljana, oktober 2015



## KAZALO

1	<i>Uvod</i>	1
1.1	Motivacija in cilji . . . . .	3
1.2	Prispevki k znanosti . . . . .	4
1.3	Temeljni pojmi . . . . .	5
1.3.1	Učenec in učitelj . . . . .	5
1.3.2	Učni objekt . . . . .	5
1.3.3	Multimedija in multimedijski tip . . . . .	6
1.3.4	Učni stil . . . . .	7
1.4	Pregled vsebine . . . . .	8
2	<i>Pedagoška zasnova priporočilnega sistema</i>	9
2.1	Konstruktivistična teorija učenja . . . . .	10
2.2	Konstruktivistična učna okolja . . . . .	11
2.3	Prilagodljivi hipermedijski učni sistemi . . . . .	13
2.4	Učni stili . . . . .	14
2.4.1	Kolbov model učnih stilov . . . . .	14
2.4.2	Rancourtov model učnih stilov . . . . .	15
2.4.3	Učni stili in hemisferičnost . . . . .	16
2.4.4	Učni stili VAK . . . . .	16
2.4.5	Združevanje učnih stilov za izbiranje ustreznih multimedij- skih tipov . . . . .	16
2.5	Zasnova raziskave . . . . .	17
2.5.1	Zbiranje podatkov o učnih stilih in pogostosti uporabe mul- timedijskih gradiv . . . . .	18

2.6	Opis vzorca . . . . .	18
2.6.1	Povezave med multimedijskimi tipi in modusi učnih stilov . . . . .	19
2.6.2	Modeliranje rabe multimedijskih gradiv glede na učne stile . . . . .	21
2.7	Interpretacija modela napovedi rabe večpredstavnostnih gradiv glede na učne stile . . . . .	22
2.7.1	Sinergija modelov učnih stilov . . . . .	23
3	<i>Oblikovanje modela priporočilnega sistema</i> . . . . .	27
3.1	Priporočilni sistemi v izobraževanju . . . . .	28
3.2	Zasnova izvirnega sistema za priporočanje . . . . .	31
3.2.1	Registracija in uvodno poizvedovanje o učenčevih učnih lastnostih . . . . .	31
3.2.2	Priporočilni sistem . . . . .	33
3.2.3	Priporočanje učnih objektov . . . . .	33
3.2.4	Priporočanje/Prilagajanje predstavitve učnih objektov . . . . .	35
3.2.5	Formativno preverjanje znanja . . . . .	35
3.3	Uporaba spletnega učnega sistema v izobraževanju . . . . .	36
4	<i>Priporočilni sistemi</i> . . . . .	37
4.1	Priporočilni sistemi in matrična faktorizacija . . . . .	39
4.1.1	Nenegativna matrična faktorizacija s stohastičnim gradientnim spustom (NG) . . . . .	40
4.1.2	Nenegativna matrična faktorizacija z izmenjujočimi se najmanjšimi kvadrati (NS) . . . . .	41
4.1.3	Polnegativna matrična faktorizacija z manjkajočimi vrednostmi (SN) . . . . .	42
4.1.4	Matrična faktorizacija z zlivanjem podatkovnih virov (DF) . . . . .	43
4.2	Priporočilni sistemi in hladni zagon . . . . .	44
4.2.1	Sistemi za reševanje problema hladnega zagona . . . . .	45
4.2.2	Reševanje problema hladnega zagona z uvodnim poizvedovanjem . . . . .	48
4.3	Novi pristop k reševanju hladnega zagona . . . . .	48

5	<i>Reševanje problema hladnega zagona</i>	51
5.1	Vstavljanje manjkajočih vrednosti v vhodno matriko . . . . .	52
5.2	Oblikovanje lastnega pristopa za vstavljanje manjkajočih vrednosti za primere v hladnem zagonu . . . . .	54
5.2.1	Vstavljanje ene začetne manjkajoče vrednosti za uporabnike v absolutnem hladnem zagonu . . . . .	56
5.2.2	Iskanje podobnih uporabnikov . . . . .	56
5.2.3	Izbor ustreznih atributov . . . . .	58
5.2.4	Vstavljanje manjkajoče vrednosti . . . . .	60
5.2.5	Priporočanje z uporabo matričnih faktorizacij . . . . .	60
5.3	Evalvacija pristopa reševanja problema hladnega zagona . . . . .	61
5.3.1	Ocena uspešnosti . . . . .	62
5.3.2	Podatkovne množice . . . . .	64
5.4	Evalvacija reševanja absolutnega hladnega zagona . . . . .	65
5.4.1	Analiza na umetni domeni . . . . .	65
5.4.2	Analiza na realnih domenah . . . . .	67
5.5	Evalvacija reševanja delnega hladnega zagona . . . . .	67
5.5.1	Analiza variant metod v posameznih korakih metode . . . . .	67
5.5.2	Rangiranje uspešnosti posameznih eksperimentov . . . . .	71
5.5.3	Evalvacija najboljših izbranih metod na realnih domenah . . . . .	73
5.6	Vpliv vstavljanja manjkajočih ocen na priporočanje uporabnikom zunanja stanja hladnega zagona . . . . .	78
6	<i>Zaključek</i>	81
6.1	Razprava in nadaljnje delo . . . . .	83
A	<i>Vprašalnik o pogostosti uporabe multimedijskih tipov</i>	85
	<i>Literatura</i>	87





# *Uvod*

*Učenje je kot veslanje proti toku:  
če ne napreduješ, greš nazaj.  
(Kitajski pregovor)*

Prilagodljivost učnih okolij je ena izmed pomembnih raziskovalnih tem na področju računalniško podprtega učenja. Sodobna izobraževalna paradigma poudarja pomen pedagoških pristopov, ki so usmerjeni k učencu. Take pristope podpirajo konstruktivistične teorije učenja, ki zahtevajo individualizacijo poučevanja in uporabo aktivnih učnih oblik, da lahko učencem v kar največji mogoči meri pomagamo pri konstrukciji novih znanj. Vloga učitelja v takem pristopu je vloga mentorja, ki svetuje učencu v procesu učenja, a mu ob tem ne določa učne poti.

Računalniški sistemi, ki se prilagajajo učenju učenca, sodijo med ekspertne sisteme. Omenjeni koncept podpira predvsem tradicionalne teorije učenja, saj določa učno pot, kar pa je v nasprotju s konstruktivistično teorijo učenja. Zato pri konstruktivističnem pristopu k učenju uporabimo sisteme za priporočanje, ki učencu dopuščajo avtonomnost in mu hkrati svetujejo v procesu učenja ter s tem zmanjšajo kognitivne obremenitve.

Sistemi za priporočanje običajno temeljijo na eni izmed treh skupin algoritmov: na algoritmih za vsebinsko izbiranje, algoritmih za izbiranje s sodelovanjem in na hibridnih algoritmih, ki na različne načine združujejo elemente prvih dveh kategorij.

Večina priporočilnih sistemov, ki so namenjeni uporabi v izobraževanju, temelji na algoritmih izbiranja s sodelovanjem. Te algoritme razdelimo na algoritme matrične faktorizacije in algoritme iskanja  $k$ -najbližjih sosedov. V disertaciji smo se odločili za uporabo metod matrične faktorizacije, ker te metode hkrati povzemajo lastnosti vseh uporabnikov in vseh objektov v priporočilnem sistemu; po drugi strani pa so metode sorazmerno uspešne pri priporočanju objektov. Njihova slabost je, da v večini primerov neustrezno priporočajo objekte povsem novim uporabnikom. Zato je ključno raziskovalno vprašanje, kako lahko izboljšamo priporočanje novim uporabnikom: katere objekte kot elemente priporočanja v priporočilnih sistemih naj priporočimo učencu/uporabniku, ko začne prvič uporabljati priporočilni sistem. Ker o uporabniku ničesar ne vemo — nahaja se v stanju hladnega zagona — lahko uporabimo uvodno proizvedovanje kot način pridobivanja začetnih informacij. Druga možnost pa je, da

uporabnike ločimo od preostalih uporabnikov in uporabimo metode, ki so namenjene reševanju hladnega zagona.

Ker je prilagajanje metod matričnih faktorizacij za reševanje hladnega zagona zelo neraziskano področje, smo še posebej motivirani, da oblikujemo lasten pristop za reševanje tega problema.

Disertacija predstavlja nov in inovativen pristop k reševanju hladnega zagona z namenom izboljšanja priporočanja metod matričnih faktorizacij.

## *1.1 Motivacija in cilji*

Združevanje različnih konceptov računalniških sistemov za podporo konstruktivističnemu učenju je ideja, ki lahko pripomore k oblikovanju modelov računalniških sistemov, ki bi se bolje prilagajali posameznikovim potrebam. S tem namenom bi lahko oblikovali prilagodljivo učno okolje, s pomočjo katerega bi učinkoviteje dosegali operativne učne cilje.

Računalniška okolja za podporo konstruktivističnemu pristopu k učenju vključujejo tudi koncept sodelovalnega učenja. Manj raziskano je področje računalniške podpore konstruktivističnemu pristopu k individualnemu učenju z uporabo računalniškega okolja, ki se prilagaja učenčevim potrebam. To lahko dosežemo z uporabo priporočilnih sistemov.

Razviti želimo učno okolje za podporo učencu, ki bi mu lahko ustrezno priporočalo učna gradiva — s tem bi učencem olajšali iskanje ustrezne literature, hkrati pa bi jim lahko pomagali pri iskanju alternativnih učnih gradiv. Priporočilni sistemi na osnovi matrične faktorizacije učinkovito priporočajo objekte uporabnikom, ki dlje časa uporabljajo sistem. Manj raziskano področje pa je, kako naj priporočamo novim uporabnikom, o katerih še ne vemo ničesar. Omenjeno situacijo lahko prenesemo tudi na področje izobraževanja, kjer je pomembno, da učencu priporočamo ustrezna učna gradiva.

Naš pristop temelji na reševanju problema hladnega zagona vsakega novega uporabnika posamezno — novemu uporabniku vstavimo manjkajoče ocene (glede na naš pristop) in nato izvedemo priporočanje z matrično faktorizacijo nad celotno množico. Naša ideja temelji na pristopu delno nadzorovanega učenja (ang. semi-supervised learning), pri katerem pri priporočanju upoštevamo vse podatke, tudi neoznačene [1]. Na podoben način torej želimo pripraviti neoznačene podatke (primere oz. uporabnike v hladnem zagonu), ki jih uporabimo skupaj z označenimi podatki, da bi izboljšali

proces učenja priporočilnega sistema [2].

V disertaciji se ukvarjamo z oblikovanjem, razvojem in s testiranjem novega pristopa, ki bi lahko povečal točnost priporočanja pri uporabi matrične faktorizacije. Razvite pristope primerjamo z golimi matričnimi faktorizacijami in preizkušamo združevanje različnih metod v kombinacije. Metodologijo, predstavljeno v okviru disertacije, razvijamo na umetni množici in testiramo tudi na dveh realnih domenah, in sicer na splošni podatkovni množici, namenjeni testiranju priporočilnih sistemov, in na podatkovni množici, ki smo jo sami zgradili med celotnim raziskovanjem.

## 1.2 Priskevki k znanosti

Deli disertacije so bili objavljeni v dveh pomembnejših člankih. Disertacija vsebuje naslednje prispevke k znanosti:

### 1. *Izboljšava pedagoške prakse in modeliranje učenca* [3]

S pomočjo podatkov, pridobljenih z reševanjem standardiziranih vprašalnikov štirih klasifikacij spoznavnih stilov, smo določili povezave med kombinacijami učnih stilov in učnimi strategijami. Ugotavljali smo, ali učenci s posameznimi prevladujočimi spoznavnimi stili uporabljajo učne strategije, ki jih klasifikacije predvidevajo za posamezen učni stil, kateremu učenci pripadajo. Zgradili smo model, ki vsebuje avtomatsko generirano znanje iz podatkov. Ta model omogoča: 1) razumevanje podatkov o učnih stilih učenca in povezovanje učnega stila učenca z drugimi njegovimi karakteristikami, kot so: način razmišljanja, način reševanja problema in način prejemanja informacij iz okolja in 2) možnost implementacije novega ekspertnega sistema za priporočanje, ki je drugačen od dozdajšnjih. Naš ekspertni sistem bo imel možnost priporočanja in prilagajanja predstavitev učnih objektov. Namenjen bo splošni uporabi v izobraževanju za poljubno učno vsebino in poljuben učni predmet v šoli.

### 2. *Razvoj lastne metode za priporočanje z uporabo metod matričnih faktorizaciji v stanju hladnega zagona* [4]

Razvili smo izviren pristop reševanja stanja hladnega zagona pri novih uporabnikih. Naš inovativen pristop izboljša priporočanje uporabnikom v hladnem zagonu, tako da uporabnikom vstavi samo nekaj ocen za izbrane objekte, ki naj bi pomagali izboljšati napovedno točnost za uporabnike v hladnem zagonu. Naš

pristop je sestavljen iz treh korakov: 1) iskanje podobnih uporabnikov; 2) izbor ciljnih atributov; 3) agregacija pridobljene vrednosti v izbrane ciljne attribute.

Reševanje hladnega zagona smo razdelili na reševanje problema absolutnega hladnega zagona in delnega hladnega zagona. Naš pristop smo evalvirali na eni umetni in dveh realnih domenah. Rezultati eksperimenta so pokazali, da smo z uporabo kombinacije metod 25-FR-ME-NG izboljšali priporočanje za uporabnike, ki se nahajajo v stanju delnega hladnega zagona.

### 1.3 *Temeljni pojmi*

Na področju računalniško podprtega učenja uporabljamo različne izraze, ki z razvojem dobivajo nove pomene. Za lažje razumevanje disertacije v nadaljevanju definiramo ključne pojme, povezane z učenjem.

#### 1.3.1 *Učenec in učitelj*

Besedi učenec in učitelj v splošnem povezujemo z osnovnošolskim izobraževanjem, v katerem imamo učitelja in učence. Z napredovanjem na višjo raven izobraževanja besedo učitelj nadomestijo nekatere sopomenke, kot so na primer profesor, asistent, predavatelj, mentor ali tutor. Enako velja za pojem učenec, ki ga lahko poimenujemo dijak, študent ali vajenec. V naši disertaciji s pojmom učenec opredeljujemo vsakega posameznika, ki se nahaja v procesu učenja – torej se v nekem trenutku uči, s pojmom učitelj pa opredeljujemo osebo, ki v nekem trenutku opravlja vlogo mentorja, ki učence usmerja v procesu učenja, in sicer z namenom, da učenec sam zgradi znanje za reševanje nekega problema.

#### 1.3.2 *Učni objekt*

Učni objekt je elektronski oziroma digitalni vir, ki vsebuje vse elemente, ki so namenjeni učenju. Namenjen je večkratni uporabi. V prilagodljivih učnih okoljih in priporočilnih sistemih lahko učni objekt predstavlja eno izmed naslednjih možnosti [3]:

- *gradivo za zaključeno učno enoto*, kar pomeni, da objekt predstavlja vsa potrebna multimedijska gradiva za realizacijo ene učne ure v razredu;

- *gradivo za del učne enote* (npr. krajša razlaga nekega elementa znotraj učne ure)
  - takšne objekte uporabljamo predvsem, ko želimo učencem ponuditi dodatno razlago ali pa želimo nekatere dele prikazati le ob določenih situacijah;
- *predstavitev učne vsebine* (npr. animacija) — nekateri spletni učni sistemi temeljijo zgolj na prilagajanju predstavitve učne vsebine, kar pomeni, da gradivo za en učni objekt predstavlja eno izmed možnih predstavitev iste učne vsebine.

V tej disertaciji učni objekt predstavlja učno gradivo za manjšo učno enoto. To pomeni, da imamo za eno učno temo več različnih predstavitev učne vsebine, kot so npr. različne multimedijske predstavitve učnih objektov. S tem lahko učencu prilagodimo sam prikaz učne vsebine.

Pri procesu učenja je treba vsak učni objekt razdeliti in predvideti, katere operativne učne cilje želimo doseči. Operativni učni cilji so opredeljeni kot konkretne vzgojno-izobraževalne aktivnosti, ki vodijo do rezultatov. Temeljni gradniki operativnih učnih ciljev so [5]:

- opredelitev glagolskega naziva za ravnanje, v katerem bo enopomensko opisana učenčeva zmožnost po končani učni uri;
- navedba pogojev in okoliščin, v katerih bo učenec pokazal, da je učni cilj dosegel;
- navedba meril, kdaj je določeno pokazano znanje učencev še zadovoljivo.

Glede na vsebino operativnih učnih ciljev jih razdelimo na podrejene in nadrejene učne cilje. Nadrejeni učni cilj dosežemo z doseganjem manjših, njemu podrejenih ciljev. Podobno lahko sklepamo v nasprotni smeri. Če s preverjanjem znanja ocenimo, da je učenec dosegel višji nadrejeni učni cilj, lahko sklepamo, da je dosegel tudi manjše podrejene cilje, ki so vsebovani v tem nadrejenem cilju. Zato lahko učne cilje koreliramo s posameznimi nalogami, ki jih postavljamo učencu. Nadrejenost in podrejenost učnih ciljev lahko zapišemo kot množico pravil IF–THEN, ki lahko nato služijo za priporočanje posameznih učnih objektov.

### 1.3.3 *Multimedija in multimedijski tip*

Multimedija (ali večpredstavnost) je predstavitev učne snovi z uporabo več medijev hkrati z namenom, da učenec prejme podatke prek različnih čutil [6].

Multimedijski tip je izbor enega izmed od osnovnih elementov multimedije, kot so: besedilo, slika, zvok in video. Z uporabo različnih multimedijskih tipov učencu omogočimo prilagojeno predstavitev učne vsebine z upoštevanjem njegovih učnih lastnosti in potreb [3].

Multimedijsko podprto učenje torej predstavlja proces učenja, v katerem učenec gradi znanje z uporabo multimedijskih tipov, s katerimi je predstavljena učna vsebina [6].

Učno okolje mora učenca s pravo izbiro multimedijskih tipov spodbuditi, da je pri učenju aktiven, kar pomeni, da mora pri učenju uporabljati različne aktivne učne oblike [6]. Na področju prilagodljivih učnih okolij je prilagajanje predstavitev učnih objektov podrejeno izbrani učni vsebini. Raziskovalci so najprej izbrali učno vsebino, ki jo bo zajelo učno okolje, nato pa glede na učno vsebino zgradili model prilagajanja predstavitev učnih objektov. Če želimo prilagajati učne objekte neodvisno od učne vsebine, mora sistem v največji mogoči meri poznati učenčeve učne preference. To so lastnosti, ki tvorijo učenčev profil.

Zasnova našega sistema temelji na priporočanju učnih objektov in izbiranju ustreznih multimedijskih tipov. Če učenecem povsem prepustimo iskanje učnih gradiv, lahko pride do kognitivne preobremenitve, kar pomeni, da se učenec, namesto da bi se osredinil na ključne kognitivne procese, osredini na iskanje informacij. S tem pa se bistveno zmanjša učinkovitost učenja [7].

Kakor smo že omenili, smo zasnovali prilagodljivo učno okolje in definirali multimedijske tipe, ki jih bomo v našem okolju podpirali in jih uporabljali za ustrezne predstavitve učne vsebine.

#### *1.3.4 Učni stil*

Chang in sodelujoči [8] so povzeli, da je učni stil indikator, ki pove, kako se učenec uči in kako bi moral učitelj učiti, da bi uspešno zadovoljil individualne učne potrebe posameznega učenca. Graf in Liu [9] sta pokazala, da učenci z različnimi učnimi stili uporabljajo različne učne strategije in se različno obnašajo v učnem sistemu. Na izbiro je veliko različnih modelov učnih stilov [10]. Vsak izmed modelov temelji na specifičnem zornem kotu učenčevih karakteristik, ki odsevajo, kako učenec dojema, zbira in miselno obdeluje podatke v procesu učenja [11, 12]. V poglavju 2.4 smo natančno in pregledno predstavili učne stile, ki smo jih uporabili za gradnjo modela, ki predstavlja eno izmed možnosti za prilagajanje predstavitve učne vsebine.

#### *1.4 Pregled vsebine*

Disertacija obsega 6 poglavij. Drugo poglavje je namenjeno pregledu področja z didaktičnega vidika. V njem osmislimo oblikovanje priporočilnega sistema in razvoj novega pristopa za reševanje hladnega zagona ter predstavimo pedagoške koncepte, ki so ključni za razvoj priporočilnega sistema za podporo učenju. Tretje poglavje predstavlja zasnovano priporočilnega sistema za podporo učenju. V poglavju 4 predstavljamo obstoječe priporočilne sisteme in pregled metod matričnih faktorizacij, ki smo jih uporabili znotraj disertacije. Poglavje sklenemo s pregledom obstoječih rešitev problema hladnega zagona, in sicer s poudarkom na problemu novega uporabnika.

Poglavje 5 predstavlja metodologijo razvoja in testiranje inovativnega pristopa za izboljšanje priporočanja metod matričnih faktorizacij, in sicer za uporabnike v stanju absolutnega in delnega hladnega zagona. Poglavje prav tako analizira rezultate eksperimentalnega dela, pri katerem smo oblikovali nov pristop in ga testirali na oblikovani umetni podatkovni množici. V nadaljevanju z izbrano kombinacijo metod analiziramo uspešnost našega pristopa na dveh realnih množicah.

Povzetek in ovrednotenje skupnih rezultatov podaja sklepno poglavje 6, ki navaja tudi pregled idej za nadaljnje delo.



*Pedagoška zasnova  
priporočilnega sistema*

*Samoučenje je, trdno verjamem, edino resnično učenje.*  
(Isaac Asimov)

## 2.1 *Konstruktivistična teorija učenja*

Konstruktivistična teorija učenja temelji na predpostavki, da je učenje aktiven proces konstruiranja znanja [13].

Konstruktivistična teorija učenja ne predvideva klasičnega vodenja učenca po korakih do cilja, ampak ga le usmerja, kar pomeni, da mora učenec sam z lastnim trudom, s spretnostmi in predhodnim znanjem priti do cilja. Učna pot ni nujno optimalna; ključno je, da jo prehodi učenec sam, po potrebi z upoštevanjem učiteljevih smernic [14].

Tabela 2.1 prikazuje razlike med tradicionalnim in konstruktivističnim učenjem. Iz nje je razvidno, kako bi moral učitelj v procesu učenja vso pozornost preusmeriti na vsakega posameznika, saj bi moral poznati njegove učne karakteristike, kot so: stili učenja, razmišljanja in pristopanja k reševanju problemov. Prav tako se spremeni njegova vloga, saj ni več središče učnega procesa v razredu, ampak opravlja vlogo vodiča, svetovalca in motivatorja, ki učencu pripravi probleme in ga spremlja med procesom učenja.

Učni proces, zasnovan na konstruktivizmu, torej poteka na različne načine, ker je od vsakega posameznega učenca odvisno, katero pot za dosego učnih ciljev bo izbral. Tam poudarja, da učencem lahko podajamo znanje, vendar morajo podano sami interpretirati in povezati s preteklim znanjem. Ker se učenci med seboj razlikujejo, se lahko njihove interpretacije razlikujejo [15]. Nekemu učencu lahko neki dani podatek pove, kako priti do zelenega cilja, medtem ko drugim isti podatek predstavlja le en manjši korak na poti do cilja ali pa ga sploh ne znajo uporabiti. Poseben poudarek je na okolju, v katerem se dogaja učna situacija (v razredu ali pa učenje doma).

Jonassen [16] je opredelil konstruktivistično učno okolje kot okolje, v katerem je treba zagotoviti: 1) učni problem, ki bo nudil učencu motivacijo; 2) povezavo na podobne probleme, učne primere, iz katerih se bo lahko učenec naučil; 3) učna gradiva oziroma povezavo na učno literaturo; 4) orodja oziroma naloge, ki spodbujajo učenčeve razmišljanje; 5) pogovor s tutorjem; 6) sodelovalna orodja.

Tabela 2.1

Ključni elementi v tradicionalnem in konstruktivističnem učnem okolju v visokošolskem izobraževanju [17]

<i>Tradicionalno okolje</i>	<i>Konstruktivistično okolje</i>
<i>Ključni pedagoški poudarki</i>	
Poučevanje, reprodukcija znanja, samostojno učenje, tekmovanje.	Učenje, gradnja znanja, sodelovanje, refleksija.
<i>Učne aktivnosti (v razredu)</i>	
Osredotočeno na učitelja, natančna navodila, didaktične aktivnosti, individualno delo.	Osredotočeno na učenca, sokratski razgovor, avtentične aktivnosti, individualno in skupinsko delo.
<i>Učiteljeve vloge</i>	
Ekspert, vir razumevanja (vir znanja), predavatelj.	Sodelavec, mentor, moderator, motivator, oblikovalec učnih skupin.
<i>Učenčeve vloge</i>	
Pasiven, poslušalec, uporabnik znanja, izdelovalec zapiskov.	Aktiven, sodelavec, graditelj znanja, samoopazovanje.
<i>Preverjanja znanja</i>	
Preverjanje poznavanja dejstev.	Avtentična uporaba znanja, portfelji, projekti, demonstracije in učni nastopi.

Jonassen [16] poudarja, da mora konstruktivistično učno okolje učencem zagotoviti še: a) povratno informacijo o tem, kaj že vedo; b) refleksijo o tem, kaj so se naučili; c) podporo pri gradnji novega znanja. Rovai [17] je predlagal, da mora konstruktivistično učno okolje poleg že naštetega omogočati še predstavitev učne vsebine, tako da je učna vsebina razporejena po poglavjih, besedilo vsebuje učiteljeve zaznamke, različno multimedijsko predstavitev in možnost učenčevega dostopa do vsebinskega kazala.

## 2.2 Konstruktivistična učna okolja

Konstruktivistična teorija učenja predvideva učno situacijo, v kateri učenec sam raziskuje in išče informacije ter s pomočjo reševanja nalog in problemov gradi neko novo znanje. Načrtovanje in implementacija spletnega konstruktivističnega učnega okolja sta zelo kompleksni, ker je treba predvideti, da ima lahko vsak učenec svoj učni scena-

rij, ki se lahko povsem razlikuje od scenarijev preostalih učencev. Omenjeno dejstvo je tudi razlog, da ni spletnih učnih okolij, ki bi bila eksplicitno namenjena konstruktivističnemu učenju. Po Jonassenu [16] bi torej morali predvideti preveč stvari, če bi želeli povsem pokriti in zagotoviti učno situacijo, kot jo predvideva konstruktivizem.

Alonso in sodelujoči [18] so ta problem rešili tako, da so povezali in uporabili več različnih učnih teorij in ob tem samo delno realizirali konstruktivistična načela. Preostala konstruktivistična spletna okolja pa vsebujejo le posamezne značilnosti konstruktivistične teorije učenja, kar pomeni, da so s tem zmanjšali kompleksnost sistema.

Wang [19] je razvil spletno konstruktivistično učno okolje s pomočjo sistema Moodle. Samo spletno okolje ni nič novega, saj temelji na funkcijah, ki jih predvideva že sistem Moodle. Učitelji so samo oblikovali učno snov, naloge in probleme, tako da so upoštevali načela konstruktivistične teorije učenja. Omenjena realizacija konstruktivističnega učnega okolja ne predvideva razgovora in sodelovanja med učenci.

Curtis in Lawson [20] sta uporabila orodje, ki je omogočalo komunikacijo med člani skupin – prek elektronske pošte, forumov, izmenjave datotek itn. Orodje je torej omogočilo komunikacijo na daljavo za podporo skupinskemu učenju. Hernández in sodelujoči [21] so razvili orodje COLLAGE, ki pomaga učitelju pri organizaciji in izvedbi sodelovalnega dela v skupinah. Orodje omogoča prikazovanje učnih gradiv in izvajanje učnih aktivnosti, ki jih pripravijo učitelji s pomočjo orodja. Učitelj tako izbere samo eno izmed oblik sodelovalnega dela, s pomočjo orodja pa pripravi ustrezna gradiva in aktivnosti. Cho in sodelujoči [22] so šli še korak naprej, saj so pogledali, kako lahko povežejo podobne uporabnike v družbenih omrežjih. Podobne uporabnike povežejo med seboj glede na komunikacijske stile in učne dosežke pri računalniško podprtem sodelovalnem učenju. Pokazali so, da obstaja korelacija med tvorjenjem družbenih mrež in komunikacijskim stilom. Študenje, ki nimajo težav pri izražanju svojega mnenja in so komunikativni, bistveno lažje spletajo nova virtualna prijateljstva in sodelujejo z neznanci. Brindley in sodelujoči [23] so se prav tako posvetili oblikovanju okolij za podporo sodelovalnemu učenju. Raziskali so, kako oblikovati učne skupine znotraj spletnega učnega okolja. Shaw [24] je oblikoval spletno učno okolje, znotraj katerega je raziskal odvisnosti med učnimi stili, sodelovalnimi tipi (načini sodelovanja znotraj spletnega foruma) in učnim dosežkom. Z raziskavo je pokazal, da sta sodelovalni tip in učni stil povezana z učnim dosežkom. Vsi naštetih primeri temeljijo na teoretični zasnovi spletnih učnih okolij za podporo sodelovalnega učenja. Zelo malo izmed njih je realiziranih, saj gre za kompleksnejše sisteme.

Na področju konstruktivističnih spletnih učnih okolij je neraziskano področje prilagajanja predstavitev učnih vsebin v konstruktivizmu. V literaturi se za to uporabljajo poimenovanje prilagodljivi hipermedijski sistemi.

### 2.3 Prilagodljivi hipermedijski učni sistemi

Prilagodljivi hipermedijski sistemi (AHS) prilagajajo svoje lastnosti, strukturo gradiv in elemente gradiv glede na neki model, na primer model učenca, model znanja in model učnih ciljev. S pomočjo podatkov iz modela torej sistem prilagodi ponujene vsebine učenčevim učnim potrebam [25].

Glede na specifičnost in vrsto prilagoditve posameznih prilagodljivih hipermedijskih učnih sistemov lahko v grobem razdelimo sisteme AHS v naslednji skupini [26]:

- *Sistemi, ki temeljijo na prilagajanju grafične predstavitve vsebine:* sistemi prilagajajo in izbirajo ustrezne predstavitve vsebine (na primer slikovno gradivo, videoposnetke, animacije, zvočne posnetke, simulacije itn.) ali pa prilagajajo samo besedilo uporabnikovim potrebam (glede na uporabnikove preference pokažejo del besedila, besedilo barvno diskriminirajo, ga prilagajajo jeziku uporabnika itn).
- *Sistemi, ki temeljijo na prilagodljivi navigaciji:* sem sodijo sistemi, ki se osredinjajo na usmerjanje uporabnika po vsebinah v sistemu. Uporabnika lahko neposredno usmerimo na neko vsebino oziroma ga preusmerimo na drugo vsebino v neki točno določeni situaciji, v kateri se nahaja. Sistem lahko razvrsti povezave (hiperlinke) po prioriteti glede na uporabnika. Nekatere povezave lahko skrrije ali pa priporoči le tiste, za katere je sistem prepričan, da so za njegovega uporabnika pomembne.

Veliko raziskovalcev je razvilo prilagodljive računalniške sisteme, ki temeljijo na spoznavnih učnih stilih. Odprto ostaja vprašanje, kateri vidiki posameznega učnega stila imajo ključno vlogo pri modeliranju in katere učne metode najboljše ustrezajo posameznemu spoznavnemu učnemu stilu. Večina prilagodljivih okolij temelji na prilagajanju besedilnih in slikovnih elementov učne vsebine. Primer učnega sistema, ki temelji na prilagodljivosti in prožnosti okolja, je iWeaver [27], ki je namenjen učenju programskega jezika Java. Okolje oblikuje učenčev profil, ki temelji na učnih spoznavnih stilih,

pridobljenih s pomočjo začetnega vprašalnika. iWeaver v procesu učenja prilagaja predstavitev učnih enot in navigacijo učenčevemu učnemu stilu. Sistem, ki smo ga idejno zasnovali v okviru disertacije, vsebuje lastnosti obeh skupin — prilagaja predstavitev učnih gradiv, hkrati pa tudi priporoča izbiro učnih objektov. Za prilagajanja predstavitve učne vsebine smo zgradili napovedni model (razdelek 2.7), ki temelji na upoštevanju različnih kombinacij učnih stilov.

## 2.4 Učni stili

V literaturi zasledimo veliko nedoslednosti glede uporabe izrazov spoznavnega (kognitivnega) in učnega stila [28]. Cassidy [28] pravi, da je bistvena razlika med izrazoma ta, da je spoznavni stil “posameznikov značilen način reševanja problemov, razmišljanja, sprejemanja in pomnjenja informacij”, medtem ko učni stil označuje specifično uporabo značilnosti spoznavnih stilov v učnih situacijah. Da bi se izognili nedoslednosti, bomo v disertaciji ves čas uporabljali izraz “učni stil”, saj se bomo osredinili na uporabo različnih modelov stilov pri učenju. Osredinili smo se na naslednje modele stilov: kognitivni stili, epistemološki stili, stili razmišljanja in zbiranja informacij.

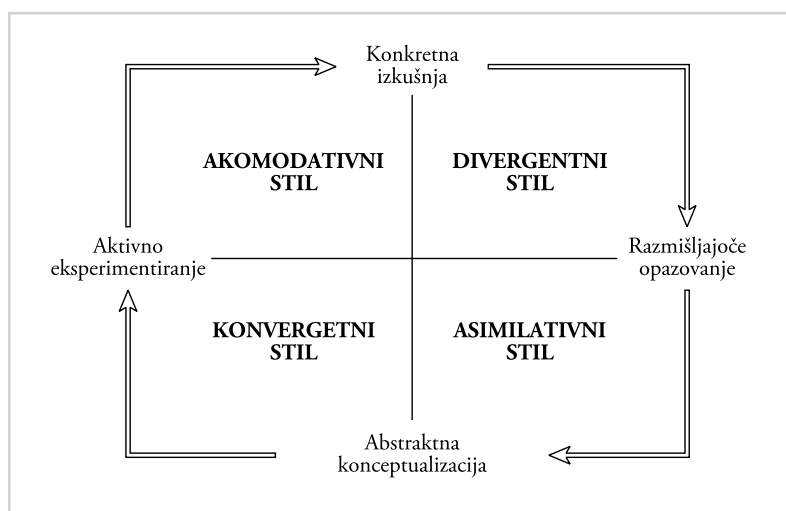
Za določanje primarnega učnega stila nekega modela se v splošnem uporabljata dva načina. Pri prvem uporabimo standardiziran vprašalnik za posamezen model [24, 29, 30]. Prednosti uporabe vprašalnikov so predvsem v tem, da nam dajo numerične vrednosti za posamezen modus učnih stilov učenca, kar nam lahko pomaga pri modeliranju, vendar so običajno časovno potratni in tudi moteči. Obenem je učencu včasih težko enoznačno določiti prevladujoči učni stil, saj obstaja možnost, da dva modusa učnih stilov hkrati dosegata maksimalni vrednosti. Druga možnost za pridobivanje informacij o učnih stilih je njihovo samodejno zaznavanje glede na učenčevo interakcijo in njegovo obnašanje v sistemu [8, 31, 32]. Zadnja možnost je časovno manj zahtevna, vendar se po drugi strani postavi vprašanje, ali je mogoče enolično povezati ustrezen učni stil in neki učenčev vedenjski vzorec pri učenju s sistemom.

Pri naši študiji smo osredinjeni na pridobivanje prevladujočih spoznavnih stilov s pomočjo standardiziranih vprašalnikov. Vsak model spoznavnih stilov temelji na modusih, katerih vrednosti dobimo z reševanjem vprašalnika.

### 2.4.1 Kolbov model učnih stilov

Kolbov model učnih stilov je eden izmed najpogosteje uporabljenih modelov pri oblikovanju prilagodljivih učnih okolij [10]. Model temelji na ravnini aktivnega uče-

nja (slika 2.1) in je razdeljen na štiri moduse: abstraktna konceptualizacija (AC), konkretna izkušnja (CE), aktivno eksperimentiranje (AE) in reflektivno opazovanje (RO) [33]. Glede na kombinacije prevladujočih modusov vsebuje Kolbov model naslednje učne stile: *asimilativni* (prevladujoča modusa AC in RO), *konvergentni* (prevladujoča modusa AC in AE), *akomodativni* (prevladujoča modusa CE in AE) in *divergentni stil* (prevladujoča modusa CE in RO) [34]. Učenci z asimilativnim učnim stilom dajejo poudarek teoretičnim vsebinam z logičnim in s praktičnim vidikom. Konvergentni učenci prav tako dajejo ključno vlogo praktičnemu vidiku v učnih vsebinah, saj se lažje učijo, če stvari poskusijo v praksi. Učenci z akomodativnim stilom se osredinjajo na uporabo različnih pristopov in metod za doseg učnih ciljev. Divergentni učenci so pozorni na zaznavanje različnih vidikov konkretne situacije [33].



Slika 2.1

Krog aktivnega učenja s pripadajočimi stili po Kolbu. Učni stil predstavlja kombinacijo dveh prevladujočih modusov.

#### 2.4.2 Rancourtov model učnih stilov

Rancourtov model učnih stilov temelji na psihoepistemoloških modusih, ki so predstavljeni kot “značilni način udejstvovanja spoznavnega (kognitivnega) in/ali čustvenega pojava” [35, 36]. Psihoepistemološki načini so: racionalni, empirični in intuitivni modus [35]. Rancourt je omenjeni model nadgradil v “Knowledge Accessing Modes Inventory (K. A. M. I.)”; v njem je definiral naslednje učne stile: racionalni stil (R),

empirični stil (E) in noetični stil (N) [37]. Racionalni stil temelji predvsem na logičnem sklepanju in argumentiranju, empirični stil temelji na opazovanju in zbiranju informacij iz okolja, noetični stil pa temelji na subjektivnih spoznanjih [35, 38].

#### 2.4.3 *Učni stili in hemisferičnost*

Hemisferna dominantnost je pomemben element v učenju, saj se možganski hemisferi razlikujeta glede na način razmišljanja in pristopanja k reševanju problemov [39, 40]. Glede na hemisferno dominantnost lahko govorimo o treh učnih stilih: levohemisferični, desnohemisferični in integrativni stil [41]. Učenci z levohemisferičnim stilom imajo raje analiziranje stvari, uporabo dejstev, logično sklepanje in operiranje s števili. Njihovo "nasprotje" so učenci z desnohemisferičnim stilom, ki imajo raje eksperimentiranje, intuitivno sklepanje in kreativno razmišljanje. Učenci, ki se lahko identificirajo z levohemisferičnimi in desnohemisferičnimi učenci, imajo integrativni učni stil [39–41]. Vprašalnik "Your Style of Learning and Thinking (YSLT)" smo v naši raziskavi uporabili za pridobivanje prevladujočega učnega stila glede na hemisferično dominantnost [41].

#### 2.4.4 *Učni stili VAK*

Model učnih stilov VAK temelji na uporabi senzoričnih poti, po katerih sprejemamo informacije iz okolja. Oznaka VAK označuje začetnice zaznavnih načinov: vizualni (V), slušni (A) in kinestetični (K) modus oz. način [42]. Omenjeni model je okrnjena različica modela VARK, ki poleg omenjenih načinov vsebuje še bralno-pisalni modus. Učenci z vizualnim stilom se lažje učijo, če imajo učno vsebino predstavljeno slikovno namesto besedilno. Kinestetični učenci imajo rajši praktično učenje z gibanjem in simulacijami. Slušni učenci pa se raje učijo s poslušanjem predavanj in drugih audioposnetkov [42–44]. Za pridobivanje informacij, kateremu učnemu stilu najbolj ustrezamo, smo v naši raziskavi uporabili vprašalnik "VAK Learning Style Inventory" [45].

#### 2.4.5 *Združevanje učnih stilov za izbiranje ustreznih multimedijskih tipov*

Večina učnih okolij kot podatek o učenčevih značilnostih uporablja samo enega izmed modelov učnih stilov. Z našo raziskavo želimo pokazati, da je smiselno kombinirati različne modele učnih stilov, saj lahko tako dobimo popolnejše informacije o



učencu in posledično lahko z večjo zanesljivostjo priporočimo ustrezne multimedijske predstavitve učnih vsebin.

Povezovanje modelov učnih stilov s preostalimi učenčevimi karakteristikami ni nova ideja. Hauptman in Cohen [29] sta pokazala, da so učenci, ki so uporabljali sistem s prilagajanjem glede na povezovanje modela VAK in modela učnih stilov glede na osebnostne značilnosti (model Myers - Briggs [46]), dosegali boljši uspeh v primerjavi s kontrolno skupino pri sumativnem (končnem) preverjanju znanja. Raziskovalci so prav tako iskali povezave med učnimi stili in kognitivnimi lastnostmi učencev [47, 48] ter učnimi stili in njihovim obnašanjem v učnem sistemu [49].

Namen prvega dela naše raziskave je poiskati povezave med različnimi modeli učnih stilov in multimedijskimi tipi učnih gradiv, in sicer z namenom, da bi dobili popolnejšo sliko o učenčevih lastnostih, ki lahko vplivajo na učinkovitejše prilagajanje predstavitve učnih gradiv. To smo dosegli, tako da smo pri oblikovanju modela upoštevali različne modele učnih stilov, ki se med seboj razlikujejo glede na svoj fokus. Tako smo lahko s podatki o načinu reševanja problemov, načinu spoznavanja, načinu razmišljanja in načinu sprejemanja informacij izbrali ustrezne multimedijske tipe. Hkrati pa smo oblikovali model, s katerim lahko priporočamo učna gradiva. Rezultate raziskave smo objavili v prispevku [3].

## 2.5 Zasnova raziskave

Raziskava je v grobem razdeljena na dva ključna dela, in sicer na pridobivanje podatkov o študentovih učnih stilih in pogostosti uporabe multimedijskih tipov in na uporabo napovednega regresijskega modela za priporočanje najprimernejših multimedijskih tipov.

Za gradnjo odločitvenega modela, ki bi ga lahko uporabili za izbiranje primernih predstavitev učnih vsebin, smo izbrali štiri modele učnih stilov. S pomočjo v slovenščino prevedenih vprašalnikov smo dobili informacijo o: učenčevemu pristopu k reševanju problemov (Kolbov model [50], "Kolbov vprašalnik o učnih stilih" [51]), kognitivnem načinu (Rancourtov model [37], "Vprašalnik načinov in stilov spoznavanja po Rancourtu" [52]), načinu razmišljanja (model hemisferične dominantnosti [41], "Vprašalnik: Tvoj stil učenja in razmišljanja" [53]) in o prevladujočem načinu sprejemanja informacij (model VAK [45], "Učenje in pomnjenje: Predelava podatkov za dolgotrajno pomnjenje" [54]).

### 2.5.1 Zbiranje podatkov o učnih stilih in pogostosti uporabe multimedijskih gradiv

Za zbiranje podatkov smo oblikovali spletišče. Na spletišču učnih stilov so študentje izpolnili vprašalnike za izbrane štiri modele učnih stilov, prevedene v slovenščino, in izpolnili anketo o pogostosti uporabe različnih multimedijskih tipov učnih gradiv. Študentje so izpolnili vprašalnike v okviru spoznavanja lastnega učenja, saj so dobili povratno informacijo o svojem učnem stilu in nasvete za izboljšanje različnih strategij učenja.

V anketi o pogostosti uporabe različnih multimedijskih tipov učnih gradiv smo spraševali, kako pogosto uporabljajo naslednje multimedijske tipe učnih gradiv:

- animacije in videoposnetke (M1),
- simulacije in izobraževalne igre (M2),
- besedila z barvno diskriminacijo, pri katerih so ključni deli besedila označeni z drugačno barvo kot preostalo besedilo (M3),
- strukturirane učna gradiva, v katerih so elementi v logičnem vrstnem redu (M4),
- avdioposnetke (M5).

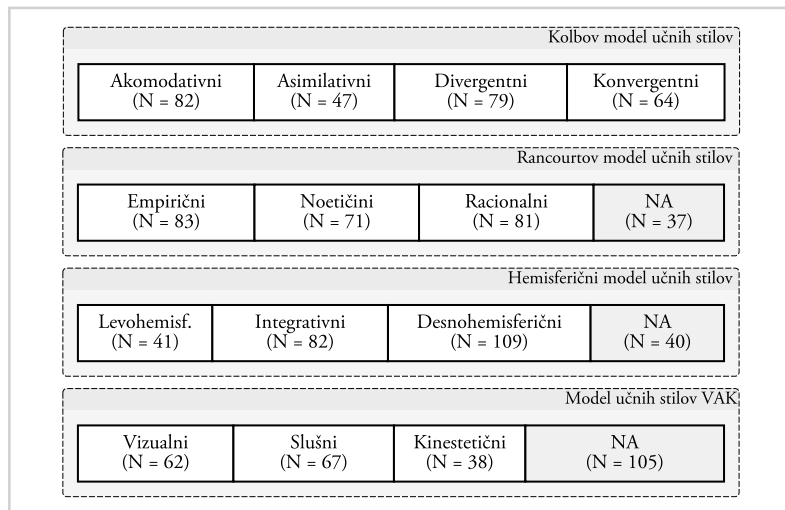
Za zbiranje podatkov o pogostosti uporabe multimedijskih tipov pri učenju smo oblikovali štiristopenjsko lestvico: 1 – nikoli, 2 – včasih, 3 – pogosto, 4 – vedno. Vprašalnik je v prilogi A.

### 2.6 Opis vzorca

V raziskavo je bilo vključenih 272 dodiplomskih študentov Pedagoške fakultete Univerze v Ljubljani. V naš vzorec so bili vključeni študentje naslednjih smeri: matematika (57), računalništvo (37), razredni pouk (75), naravoslovje (51) in družboslovje (52).

Slika 2.2 prikazuje porazdelitev študentov glede na različne stile znotraj modelov učnih stilov. V situaciji, ko študent doseže enako vrednost pri dveh najbolj izraženih dimenzijah učnega stila, študentu ni mogoče določiti primarnega učnega stila. Takšne študente smo razvrstili v kategorijo NA – informacija o učnem stilu ni na voljo. Pri posameznih modelih učnih stilov je število študentov, ki jih ne moremo razvrstiti, različno. Teh študentov nismo vključili v analizo korelacij med posameznimi učnimi stili

in multimedijскими tipi, saj lahko takšnim študentom ponudimo poljuben multimedijški tip, ker jim ustrezajo različne predstavitve učnih vsebin. V naslednjem koraku modeliranja pa smo upoštevali vse primere, tudi študente, ki jim ne moremo določiti primarnega stila, saj nas je zanimalo, kako različne kombinacije učnih stilov (različnih modelov učnih stilov) vplivajo na izbor ustreznih multimedijških tipov.



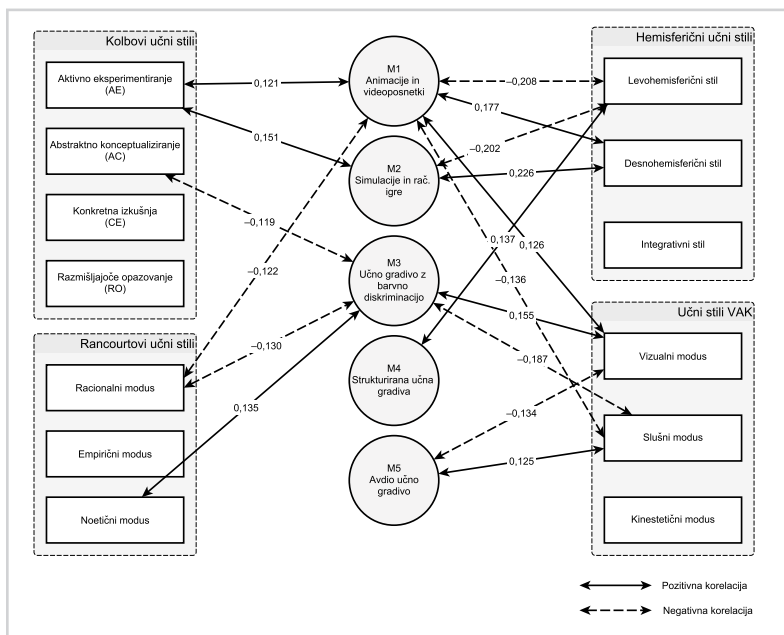
Slika 2.2

Porazdelitev študentov glede na posamezne učne stile. NA označuje primere, ki jim nismo mogli enolično določiti ustreznega stila učenja za posamezen model učnih stilov.

### 2.6.1 Povezave med multimedijскими tipi in modusi učnih stilov

Za nadaljnji razvoj modela, ki nam bo služil za priporočanje, smo pogledali, ali obstajajo statistično pomembne povezave med posameznimi multimedijскими tipi učnih gradiv in modusi modelov učnih stilov. Za analizo povezanosti spremenljivk smo uporabili Spearmanov korelacijski test. Slika 2.3 prikazuje grafično predstavitev rezultatov testa. Na grafu so multimedijški tipi označeni s krogom, modusi spoznavnih stilov pa so v pravokotnikih. Povezava med multimedijским tipom in modusom učnega stila označuje, da je med njima statistično značilna korelacija. Tabela 2.2 vsebuje izračunane vrednosti korelacijskih koeficientov in p-vrednosti.

Analiza korelacij nakazuje, da študentje, ki imajo izrazit modus aktivnega eksperimentiranja, modus desnohemisferičnosti ali vizualni modus, pri učenju z IKT pogosteje uporabljajo animacije in videoposnetke. Po drugi strani študentje z izrazitejšim



Slika 2.3

Statistično pomembne ( $p \leq 0,05$ ) korelacije med multimedijskimi tipi in posameznimi modusi modelov učnih stilov.

levohemisferičnim modusom, z izrazitejšim slušnim modusom ali z izrazitejšim racionalnim modusom redkeje uporabljajo multimedijski tip M1 (animacije in videoposnetki).

Za multimedijski tip M2 so rezultati testa pokazali, da študentje s prevladujočim modusom aktivnega eksperimentiranja ali prevladujočim desnohemisferičnim modusom pri učenju pogosteje uporabljajo simulacije in izobraževalne računalniške igre. Ravno nasprotno pa je test pokazal rezultat, da študentje s prevladujočim levohemisferičnim modusom redkeje uporabljajo omenjeni multimedijski tip.

Analiza je prav tako pokazala, da učenci z izstopajočim noetičnim modusom ali s prevladujočim vizualnim modusom pogosto uporabljajo besedila, ki vsebujejo barvno diskriminacijo (M3). Po drugi strani pa so rezultati pokazali, da študentje z bolj izrazitim modusom abstraktno konceptualizacije, racionalnim modusom ali slušnim modusom redkeje uporabljajo omenjeni multimedijski tip.

Glede pogostosti uporabe strukturiranih učnih gradiv (M4) je test pokazal, da jih po-

Tabela 2.2

Izračunani Spearmanovi korelacijski koeficienti in njihove p-vrednosti (v oklepajih) za izračun korelacij med multimedijskimi tipi in modusi modelov učnih stilov. V krepki pisavi so tiste korelacije, ki so statistično pomembne ( $p \leq 0,05$ ).

Učni modus	M1	M2	M3	M4	M5
v1_KI	0,019 (0,753)	-0,036 (0,550)	0,064 (0,291)	-0,070 (0,250)	-0,067 (0,269)
v1_RO	-0,033 (0,586)	-0,105 (0,085)	-0,054 (0,379)	-0,038 (0,531)	0,054 (0,376)
v1_AK	-0,04 (0,506)	0,007 (0,905)	<b>-0,119 (0,50)</b>	-0,004 (0,942)	-0,053 (0,380)
v1_AE	<b>0,121 (0,046)</b>	<b>0,151 (0,013)</b>	0,057 (0,348)	-0,003 (0,959)	-0,006 (0,921)
v2_R	<b>-0,122 (0,044)</b>	-0,096 (0,114)	<b>-0,130 (0,032)</b>	0,082 (0,177)	-0,111 (0,068)
v2_N	0,039 (0,520)	0,002 (0,973)	<b>0,135 (0,026)</b>	-0,067 (0,274)	0,056 (0,361)
v2_E	0,108 (0,076)	0,091 (0,133)	-0,027 (0,653)	0 (0,994)	0,051 (0,400)
v3_L	<b>-0,208 (0,003)</b>	<b>-0,202 (0,001)</b>	-0,071 (0,244)	<b>0,137 (0,024)</b>	-0,046 (0,453)
v3_D	<b>0,177 (0,003)</b>	<b>0,226 (0,000)</b>	0,075 (0,219)	-0,039 (0,518)	0,031 (0,612)
v3_I	0,015 (0,800)	-0,076 (0,211)	-0,048 (0,432)	-0,094 (0,121)	0,016 (0,793)
v4_V	<b>0,126 (0,0038)</b>	-0,043 (0,478)	<b>0,155 (0,011)</b>	0,064 (0,293)	<b>-0,134 (0,027)</b>
v4_A	<b>-0,136 (0,025)</b>	-0,038 (0,533)	<b>-0,187 (0,002)</b>	-0,095 (0,116)	<b>0,125 (0,040)</b>
v4_K	0,009 (0,879)	0,085 (0,164)	0,027 (0,654)	0,001 (0,981)	0,016 (0,791)

gostejše uporabljajo študentje s prevladujočim levohemisferičnim stilom. Pričakovano je test pokazal tudi, da študentje z izstopajočim slušnim modusom pogosto uporabljajo avdioposnetke (M5) pri učenju, medtem ko jih študentje s prevladujočim vizualnim modusom uporabljajo redkeje.

### 2.6.2 Modeliranje rabe multimedijskih gradiv glede na učne stile

Poiskati želimo povezave med kombinacijami učnih stilov in napovedjo pogostosti uporabe večpredstavnostnih materialov. Ker je naš problem večciljen, saj želimo hkrati napovedati pripadnost uporabe več razredom; za tovrstno modeliranje potrebujemo tudi takšen model. S tem namenom smo izbrali večciljna regresijska drevesa [55, 56], pri katerih smo upoštevali vse štiri modele učnih stilov za napovedovanje pogostosti uporabe petih multimedijskih tipov.

Večciljno regresijsko drevo je napovedni model, podoben običajnemu regresijskemu drevesu, le da ima več ciljnih (regresijskih) spremenljivk. Listi večciljnega drevesa vsebujejo vektor, ki je sestavljen iz povprečnih vrednosti posameznih ciljnih spremenljivk vseh učnih primerov, ki so bili razvrščeni v dani list [55–57]. Metoda deluje po principu gradnje odločitvenih dreves od zgoraj navzdol (algoritem TDITD), ki rekurzivno uvršča vzorec v poddrevesa, in sicer ob upoštevanju rezultatov testov, ki so v notranjih vozliščih drevesa [57].

Za gradnjo drevesa smo uporabili sistem CLUS [58]. Uporabili smo standardno hevrstiko, ki minimizira varianco ciljnih vrednosti homogenih listov, ki so znotraj poddrevesa. Za lažji pregled, interpretacijo in za iskanje kombinacij modusov učnih stilov smo uporabili metodo rezanja dreves (angl. "pruning") garofalakis [59], ki v zgrajenem drevesu izbere tisto najboljšo poddrevo, ki ima vnaprej izbrano največje dovoljeno število vozlišč in najmanjšo vrednost napake na učni množici [60]. Za lažjo interpretacijo in nadaljnjo uporabo smo model obrezali tako, da je drevo največje globine 3 in ima hkrati čim manjše število vseh vozlišč. Parametre orodja CLUS smo nastavili na  $\text{MaxDepth} = 3$  (največja dovoljena globina drevesa) in omejili skupno število vozlišč drevesa na največ 13.

### *2.7 Interpretacija modela napovedi rabe večpredstavnostnih gradiv glede na učne stile*

Naš cilj je povezati različne kombinacije modelov učnih stilov z ustreznimi multimedijскими tipi, da bi študentom na čim ustrežnejši način izbrali različne predstavitve učnih vsebin. S tem namenom smo zgradili večciljno regresijsko drevo (slika 2.4). Pri gradnji modela smo uporabili podatke, opisane v razdelku 2.6. Vhodni atributi predstavljajo naslednji atributi:

- KLS: nominalna spremenljivka, ki vsebuje eno izmed štirih vrednosti za ime Kolbovega učnega stila (divergentni, konvergentni, asimilativni in akomodativni stil).
- RLS: nominalna spremenljivka, ki vsebuje eno izmed treh vrednosti za ime Rancourtovega učnega stila (racionalni, noetični in empirični).
- HLS: nominalna spremenljivka, ki vsebuje eno izmed treh vrednosti za ime hemisfernega učnega stila (levohemisferični, desnohemisferični in integrativni stil).
- VLS: nominalna spremenljivka, ki vsebuje eno izmed treh vrednosti za ime učnega stila VAK (vizualni, slušni in kinestetični stil).

Ciljne spremenljivke predstavlja pet numeričnih spremenljivk: 1) animacije in video-posnetke (M1); 2) simulacije in izobraževalne igre (M2); 3) besedila z barvno diskriminacijo, kjer so ključni deli besedila označeni z drugačno barvo kot preostalo besedilo (M3); 4) strukturirana učna gradiva, v katerih so elementi v logičnem vrstnem redu (M4);

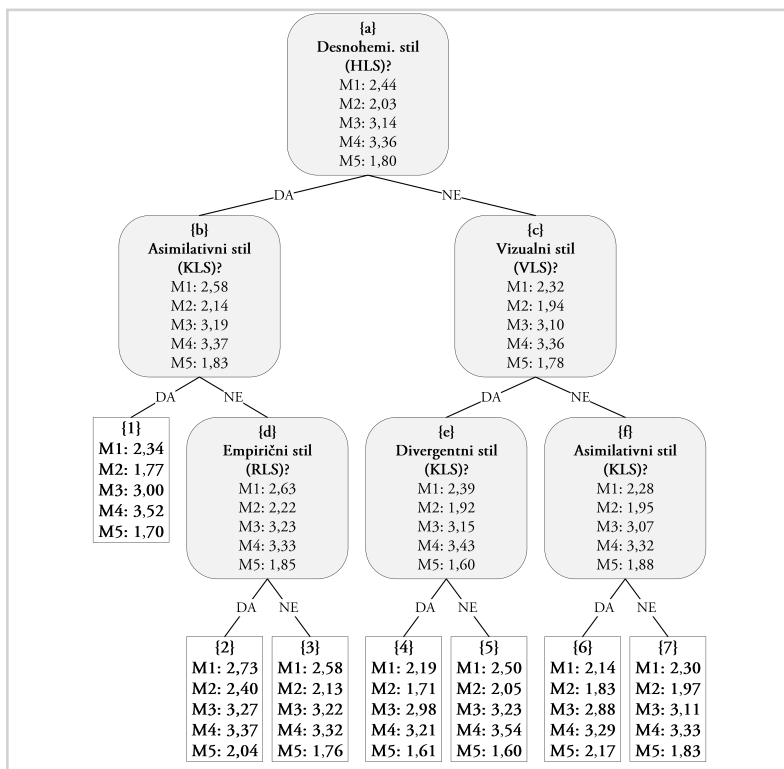
5) avdioposnetke (M<sub>5</sub>). Ciljne spremenljivke vsebujejo celoštevilске vrednosti (od 1 do 4), kako pogosto študent uporablja posamezen multimedijски tip (1 – nikoli, 2 – redko, 3 – pogosto in 4 – vedno).

Slika 2.4 prikazuje zgrajeno večciljno regresijsko drevo, ki ima v listih vektor napovedi regresijskih vrednosti posameznih tipov. V vozliščih drevesa so vektorji, ki vsebujejo povprečne vrednosti vseh tistih učnih primerov, ki so razvrščeni v poddrevo. Za lažje sklicevanje na vozlišča pri interpretaciji smo na sliki 2.4 označili s številko (npr. {5}) liste drevesa in s črko (npr. {b}) vozlišča drevesa.

Omenili smo že situacijo, v kateri ni mogoče vsakemu študentu določiti dominantnega učnega stila (glede na prevladujoči modus), kar pa ne pomeni, da mu ne moremo določiti primarnih učnih stilov pri preostalih treh modelih učnih stilov. Ker želimo pogledati vse kombinacije učnih stilov in vpliv kombinacij na napoved regresijskih vrednosti, smo pri gradnji drevesa upoštevali vse učne primere.

### 2.7.1 Sinergija modelov učnih stilov

Zgrajeno večciljno regresijsko drevo na sliki 2.4 v korenu vsebuje hemisferični model učnih stilov. Vidimo, da se vse vrednosti vektorja povprečnih vrednosti multimedijских tipov v korenu levega poddrevesa povečajo (vozlišče {b}) in zmanjšajo v korenu desnega poddrevesa (vozlišče {c}). To pomeni, da desnohemisferični študentje pogosteje uporabljajo različne multimedijске tipe kot študentje z levohemisferičnim in integrativnim učnim stilom. Ta ugotovitev potrjuje, da študentje z levohemisferičnim in integrativnim učnim stilom dajejo večji poudarek teoriji in vsebini učnega gradiva ne glede na način predstavitve. Po drugi strani pa se takšni študentje raje učijo s pomočjo besedil, ki ne vsebujejo multimedijских elementov. Glede na vozlišče {b} študentje, ki so desnohemisferični in asimilativni, redkeje uporabljajo animacije (M<sub>1</sub>), simulacije (M<sub>2</sub>), besedila z barvno diskriminacijo (M<sub>3</sub>) in avdiogradio (M<sub>5</sub>). Ta ugotovitev podpira izhodišče, da se asimilatorji osredinjajo na študijo besedilnih učnih virov in analitično raziskovanje različnih modelov. Zato dajejo večji poudarek teoretičnim izhodiščem. Študentje, ki so desnohemisferični, niso asimilatorji in so empirični (list {2}), v povprečju pogosteje uporabljajo vse multimedijске tipe kakor študentje, ki so desnohemisferični, a niso asimilatorji in niso empirični. Tudi ta ugotovitev podpira tezo, da empirični ljudje dajejo bistveno večji poudarek zaznavanju okolja pri učenju, zato je smiselno, da jim ponudimo raznovrstne multimedijске predstavitve. Poglejmo še, kaj velja za študente, ki niso desnohemisferični. Glede na drevo študentje, ki niso



Slika 2.4

Večciljno regresijsko drevo za napovedovanje pogostosti uporabe multimedijskih tipov glede na študentove učne stile.

desnohemisferični, so pa vizualni in divergentni (list {4}), pogosteje uporabljajo multimedijske tipe M<sub>1</sub>, M<sub>3</sub> in M<sub>4</sub>, kar ponovno potrди tezo, da se študentje z vizualnim in divergentnim učnim stilom pri učenju opirajo predvsem na informacije, ki so jih zaznali z vidom. Zato so jim blizu animacije, videoposnetki in besedila, ki so barvno obogatena. Podobno lahko sklepamo tudi za študente, ki niso desnohemisferični, so vizualni in niso divergentni (list {5}). Če primerjamo vrednosti v listih {4} in {5}, opazimo, da študentje, ki so razvrščeni v list {5}, pogosteje uporabljajo vsa multimedijska gradiva (razen avdioposnetkov M<sub>5</sub>) kakor študentje, razvrščeni v list {4}. Ta ugotovitev potrjuje trditev, da se ljudje z divergentnim stilom lažje prilagodijo različnim učnim situacijam, medtem ko drugi učni stili po Kolbu dajejo večji poudarek na izboru ustreznih



ne predstavitve učne vsebine. Študentje, ki niso desnohemisferični, niso vizualni in so asimilativni (list {6}), pri učenju pogosteje uporabljajo avdioposnetke kakor študentje, ki niso desnohemisferični, niso vizualni in niso asimilativni (list {7}). Po drugi strani pa študentje, razvrščeni v list {7}, pogosteje uporabljajo preostale multimedijske tipe kakor študentje, razvrščeni v list {6}. Ti ugotovitvi potrjujeta teoretično izhodišče, saj za študente, razvrščene v list {7}, velja, da dajejo večji poudarek raznolikim multimedijskim predstavitev učne vsebine. V dani situaciji želijo preučiti dejstva, pridobljena s pomočjo raznolikih multimedijskih gradiv oz. virov.

Modeli učnih stilov razdelijo študente na podobne skupine, saj se nekateri vidiki modelov lahko prekrivajo (npr. desnohemisferični stil je podoben noetičnemu in vizualnemu stilu). Zato obstaja možnost, da nam drevo nakaže kombinacije, ki bi nam dale redundantno informacijo. Glede na drevo lahko torej sklepamo, da študentje v povprečju najpogosteje uporabljajo besedila, ki so dobro strukturirana in ki vsebujejo barvno diskriminacijo. Eden izmed ključnih razlogov za takšen rezultat je, da so študentje navajeni na klasično učenje ob knjigah; po drugi strani pa smo navajeni vsakodnevno spremljati filme s podnapisi, kar se odraža tudi na dejstvu, da se raje učimo z besedilnim gradivom [61].

Če upoštevamo zgornjo ugotovitev, bi morali pri učnem procesu učencem ponuditi samo besedilna gradiva, kar pa ni smiselno. Pri učenju namreč ne moremo vedno ponuditi besedilnih učnih gradiv. Lahko nastane situacija, ko študent želi poiskati alternativo besedilnim predstavitev, zato je pomembno, da naš sistem upošteva tudi napovedane vrednosti za preostale multimedijske tipe. Zgrajeno drevo lahko uporabimo za to, da različnim študentom ponudimo čim bolj raznolika učna gradiva z upoštevanjem njihovih učnih stilov.

Poglejmo na primeru, kako lahko zgrajeni model uporabimo za priporočanje ustreznih predstavitev učnih vsebin. Za primer vzemimo študenta, ki je akomodativen, racionalen, desnohemisferičen in vizualen. Študent želi poiskati alternativo besedilni predstavitvi delu učne snovi. Model bi študenta razvrstil v list {3}, kar pomeni, da bi študentu priporočil animacije in videogradiva za del učne vsebine. Odločitev je povsem smiselna, saj ima študent vizualen učni stil. Hkrati pa je študent tudi desnohemisferičen in racionalen, kar pomeni, da daje poudarek zaznavanju informacij iz okolja, te pa smiselno analitično in teoretično ovrednoti.

V poglavju 2.6 smo omenili, da včasih ni mogoče določiti prevladujočega učnega stila za vsak model. Študentu želimo ponuditi alternativo besedilnim gradivom.

Predpostavimo, da imamo študenta, ki ima asimilativni, empirični in slušni stil; za hemisferični model ni bilo mogoče določiti primarnega učnega stila. Glede na drevo bi ga model razvrstil v list {6}, kar pomeni, da bi mu kot alternativo ponudil avdiogradiva, kar je popolnoma ustrezno, ker ima študent slušni stil. Ta ugotovitev tudi potrjuje dejstvo, da lahko, če ne moremo določiti prevladujočega učnega stila po enem modelu, s pomočjo preostalih učnih stilov vseeno priporočimo ustrezen tip multimedijskih gradiv.

*Oblikovanje modela  
priporočilnega sistema*

*Nobena pot ni ravna, nobena pot ni revna,  
a vsaka je zahtevna in tvoja ena sama – glavna.  
(Tone Pavček)*

### 3.1 Priporočilni sistemi v izobraževanju

Priporočilni sistemi so posebne aplikacije, ki temeljijo na filtriranju informacij. Cilj sistemov je določiti bistveno manjšo podmnožico ustreznih objektov za točno določenega uporabnika [62]. V splošnem so priporočilni sistemi prilagojeni komercialnim namenom, vendar lahko njihove lastnosti analogno predrugачimo in prilagodimo potrebam izobraževanja. Njihovo lastnost pa lahko uporabimo za podporo konstruktivističnemu pristopu k učenju, ki temelji na usmerjanju učenca v procesu učenja. Seveda pa je treba definirati, kaj želimo s pomočjo priporočilnega sistema priporočati.

Eden izmed prvih priporočilnih sistemov, ki so ga uporabili za izobraževanje, je Altered Vista [63], ki temelji na skupni rabi ocen in mnenj o različnih spletnih povezavah do učnih virov. Uporabnik poda oceno nekega spletnega mesta, ta pa služi kot informacija sistemu, ki zbira nabor spletnih povezav in njihovih ocen. Preostali uporabniki si s pomočjo pregledovanja zbirnih ocen drugih uporabnikov ustvarijo neko mnenje in se na njegovi podlagi odločijo, ali bodo neki spletni vir obiskali in prebrali. Posledično s tem sistem omogoča tudi sodelovanje med učenci, ki je pomemben element socialnega konstruktivizma.

Podobno učno okolje, ki temelji na priporočilnem sistemu, je QSIA [64]. Okolje omogoča priporočanje in ocenjevanje spletnih učnih virov znotraj učne skupine. Učenci oblikujejo vprašanja in manjše naloge, s pomočjo katerih se sprašujejo med seboj in preverjajo razumevanje učne snovi. Podobno deluje tudi sistem CYCLADES [65], ki uporablja priporočanje z uporabo skupinskega izbiranja (angl. “collaborative filtering”) ter upošteva učenčevo ocenjevanje znotraj učne skupine in glede na druge učne skupine. Oba sistema temeljita na načelih konstruktivističnega učenja, vendar sta namenjena uporabi šele v zadnji fazi učenja, tj. pri utrjevanju in preverjanju znanja. Do tega stanja pa je treba učence še pripeljati, za kar bi lahko uporabili prej omenjena sistema, v katerih bi si lahko učenci postavljali različne učne probleme, ki bi jih reševali tako, da bi sami poiskali učno vsebino za rešitev problema. Zastavlja pa se vprašanje, ali

so učenci dovolj usposobljeni, da bi si zastavljali problemske situacije, ki bi bile dovolj kompleksne in hkrati izvedljive.

Huang in sodelujoči [66] so razvili priporočilni sistem LSRS, ki temelji na priporočanju z upoštevanjem sekvenc, torej v kakšnem zaporedju so učenci pregledovali učne objekte. S pomočjo Markovskih modelov so sestavili profile učencev v obliki seznamov, ki so vsebovali predviden vrstni red obiskovanja posameznih učnih objektov. Zelo podobno deluje priporočilni sistem, ki sta ga razvila Shen in Shen [67], vendar način priporočanja temelji na zgrajenih sekvenčnih pravilih, ki pomagajo učencem pri navigaciji oziroma brskanju med različnimi koncepti, ki so predstavljeni kot ontologija. Na prvi pogled bi predpostavili, da oba sistema temeljita na konstruktivizmu, vendar to ne drži. Konstruktivizem namreč ne predpostavlja, da bi učitelj omejil učenčev gibanje po sistemu, ampak ravno nasprotno. Učenec mora raziskovati tako, da pregleduje različne učne objekte. To dosežemo tako, da mu ne prikažemo točno določenega učnega objekta z neko učno vsebino, ampak mu le označimo tiste objekte, za katere sistem predvideva, da bi bili učencu zanimivi in bi jih tudi pregledal.

Priporočilni sistem ReMashed [68] omogoča, da lahko učenci ocenjujejo različne učne objekte in si s tem prilagodijo način priporočanja učnih objektov. Sistem izkorišča lastnost priporočilnih sistemov, ki jih sicer uporabljajo v komercialne namene. Danemu učencu poišče podobne učence glede na njihovo ocenjevanje učnih objektov in tako zgradi okolico podobnih učencev. Priporočanje nato poteka z uporabo skupinskega izbiranja. Hkrati pa pri priporočanju tudi upošteva značke (angl. "tags"), ki so namenjene označevanju vsebine posameznih učnih objektov. Sistem sicer podpira konstruktivistični pristop, vendar učencu samo priporoča učne objekte, pozablja pa na prilagajanje predstavitve učne vsebine, saj ni nujno, da je vsaka predstavitev neke učne teme za nekega učenca primerna. S tem ko učenec išče ustrezne predstavitve učne vsebine, pa lahko situacija vodi do kognitivne preobremenitve, ki nastane, kadar učenci več pozornosti posvečajo iskanju primernih informacij kakor pa samemu učenju [6].

Klašnja - Miličević in sodelujoči [69, 70] so oblikovali priporočilni sistem Protus za podporo učenju programskega jezika Java. Sistem temelji na izbiranju s sodelovanjem, in sicer na metodi na osnovi pomnjenja. Raziskovalci so razdelili učence v skupine glede na Felder – Solomonov model učnih stilov [71]. Posamezni skupini so torej vnaprej dodelili predstavitev učne vsebine. Za izračun neznane cene učnega objekta (kot primer učne vsebine) so upoštevali ocene študentov, ki so si med seboj podobni. Za izračun podobnosti dveh učencev so izračunali vrednost Pearsonovega korelacijskega

koeficienta dveh vektorjev vseh ocen, ki sta jih podala uporabnika za neko množico učnih objektov. Priporočanje učnih objektov tako temelji na učenčevem ocenjevanju učnih objektov. Prednost njihovega sistema je predvsem v tem, da upoštevajo dejstvo, da vse ocene učnih objektov različnih učencev niso enakovredne. Slabost sistema je, da metoda priporočanja temelji le na podobnostih učencev glede na ocenjevanje učnih objektov. To bi bilo mogoče izrabit, saj bi lahko učenci namenoma ocenjevali neustrezno. To bi lahko privedlo do napačnega priporočanja, saj ocene ne bi bile verodostojne. Rešitev take situacije je, da uporabimo pristope za zaznavanje anomalij v podatkih.

Raziskovalki Ghauth in Abdullah [72] sta razvili metodo za priporočanje učnih objektov na osnovi pomnjenja. Metoda najprej prek modula na osnovi zapisov na strežniku ustvari množico boljših učencev. Nato izračuna napoved ocene še neoocenjenega učnega objekta prek funkcije podobnosti oz. oddaljenosti učencev glede na preteklo ocenjevanje. Sistem upošteva težo ocenjevanja učnih objektov glede na učenčevo uspešnost. Pomanjkljivost sistema je, da priporočanje temelji le na ocenjevanju učnih objektov, kar pomeni, da ne upošteva učenčevih karakteristik, od katerih je odvisna ustreznost predstavitve posamezne učne vsebine. Za to bi bilo treba upoštevati različne učenčeve lastnosti, kot so na primer spoznavni stili.

Eden izmed nazadnje razvitih priporočilnih sistemov, namenjenih izobraževanju, je sistem IWT [73, 74], ki upošteva dosežene učne cilje in glede na že naučeno sistem ustrezneje priporoči učne objekte. Sistem temelji na skupinskem izbiranju na osnovi pomnjenja. Učencu poišče podobne učence, za oceno podobnosti pa sistem izračuna Pearsonov korelacijski koeficient dveh vektorjev ocen dveh uporabnikov. Hkrati pa pri priporočanju upoštevajo tudi samo zaporedje – učno pot, ki jo je učenec “prehodel” znotraj učnih gradiv. Ideja sistema je samo delno skladna s konstruktivistično učno teorijo, saj je zadnja lastnost v nasprotju z omenjeno teorijo, ker priporoča zaporedje učnih objektov.

Na osnovi pregleda obstoječih priporočilnih učnih sistemov lahko povzamemo naslednje: v večini primerov, z izjemo sistema Protus, raziskovalci predpostavljajo, da je vseeno, kako je učna vsebina predstavljena. Večji poudarek dajejo priporočanju učnih tem. Idejna zasnova našega sistema gre korak naprej od sistema Protus. Pri ocenjevanju učnih objektov ne upoštevamo samo enega modela učenčevih učnih stilov, ampak več modelov, saj tako dobimo popolnejšo sliko o učenčevih učnih karakteristikah. S tem poleg priporočanja posameznih učnih objektov priporočamo tip predstavitve učne vsebine.

Tehnološko gledano pa predstavljeni sistemi temeljijo na preprostih metodah, ki so ob veliki količini podatkov časovno zelo potratne. Naš sistem uporablja sodobnejšo in primernejšo metodo za priporočanje, matrično faktorizacijo, ki jo predstavljamo v poglavju 4.

### *3.2 Zasnova izvirnega sistema za priporočanje*

Model adaptivnega sistema za priporočanje učnih objektov smo zasnovali glede na faze, v katerih je učenec med učnim procesom. Slika 3.1 predstavlja koncept našega učnega okolja ALECA (Adaptive Learning Environment with Constructivist Approach), ki je sestavljen iz naslednjih ključnih delov:

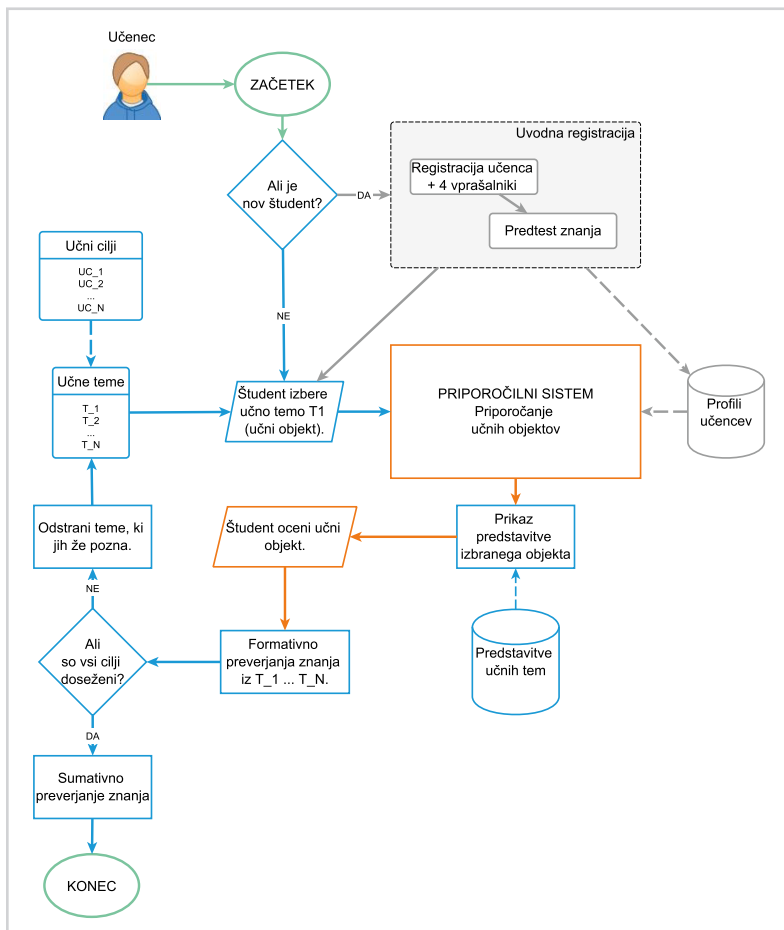
1. registracija in uvodno poizvedovanje o učenčevih učnih lastnostih,
2. priporočilni sistem,
  - a) priporočanje učnih objektov,
  - b) priporočanje/prilagajanje predstavitve učnih objektov,
3. formativno/sumativno preverjanje znanja.

#### *3.2.1 Registracija in uvodno poizvedovanje o učenčevih učnih lastnostih*

Registracija učenca je prvi stik učenca z našim učnim sistemom. Za samo spletno učno okolje ime in priimek učenca nista pomembna, saj želimo le spremljati razliko med doseganjem učnih ciljev pred začetkom procesa učenja s sistemom in po njem.

Učenec bi se na začetku registriral in si izbral poljubno uporabniško ime. Nato bi izpolnil vse štiri vprašalnike posameznih modelov učnih stilov:

- Kolbov model učnih stilov – Learning Style Inventory [51],
- Rancourtovi učni stili – Knowledge Accessing Modes Inventory [52],
- Hemisferičnost in učni stili – Your Style of Learning and Thinking (YSLT) [53],
- Učni stili VAK – VAK Learning Style Inventory [54].



Slika 3.1

Skica pedagoškega sistema  
ALECA.



S pomočjo prevedenih in prirejenih vprašalnikov bo sistem dobil informacijo o učenčevemu pristopu k reševanju problemov (Kolbov model), kognitivnem načinu spoznavanja (Rancourtov model), načinu razmišljanja (hemisferičnost in učni stili) in prevladujočem načinu sprejemanja informacij (model VAK). Potem ko učenci uspešno izpolnijo vse štiri vprašalnike, rešijo še predtest, ki preveri njihovo predznanje.

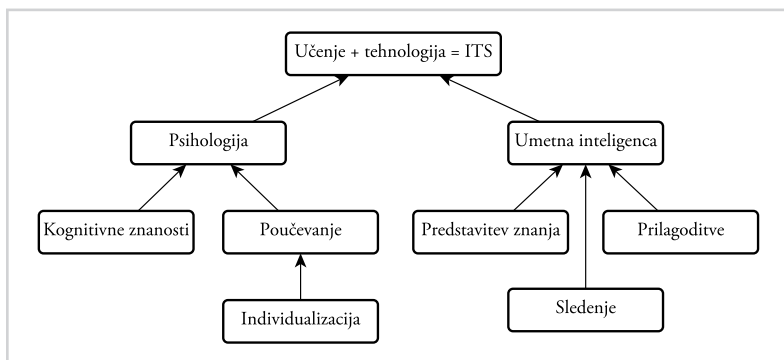
### 3.2.2 *Priporočilni sistem*

Po začetni registraciji in reševanju predtesta, bi sistem prilagodil seznam doseganja učnih ciljev. Učencem bi ponudil le tiste učne teme/objekte glede na ustrezne učne cilje, ki si jih mora še pogledati, se jih naučiti in preučiti, da bo dosegel vse zastavljene učne cilje. Prav tako bi sistem prilagodil začetne predstavitve učnih objektov, in sicer do trenutka, ko učenec ne poda prve ocene nekega učnega objekta. Od prve ocene naprej bi za prilagajanje skrbelo jedro priporočilnega sistema, ki bi reševalo tudi probleme stanja hladnega zagona. Stanje hladnega zagona je stanje, ko sistem nima dovolj ocen, da bi lahko natančneje priporočal učne objekte.

Priporočilni sistem je ključni gradnik učnega sistema ALECA. Glede na zasnovano sistema (3.1) je sistem priporočanja razdeljen na dva povsem različna koncepta priporočanja. Ideja priporočanja učnih objektov je, da bi priporočanje objektov delovalo neodvisno od priporočanja predstavitev učnih objektov. Prvo priporočanje bi temeljilo na domenskem modelu, ki bi povezoval operativne učne cilje z ustreznimi učnimi objekti. Ti bi omogočali učencu, da s pomočjo njihove obravnave doseže pričakovane učne cilje. Na drugi strani pa bi imeli priporočanje tipa predstavitev učnih objektov, ki bi temeljilo na ocenah, ki jih predhodni uporabniki namenijo za neki tip predstavite učnega objekta. V naslednjih dveh razdelkih natančneje opišemo oba idejna koncepta priporočanja.

### 3.2.3 *Priporočanje učnih objektov*

Naša izdejna zasnova računalniškega sistema za podporo konstruktivistični teoriji sodi med ekspertne sisteme, za katere je značilno, da se v določenih situacijah obnašajo kot človeški eksperti na posameznem področju [75]. Ekspertni sistemi temeljijo na različnih predstavitvah znanja v samem sistemu, kot so na primer: sklepanje na osnovi primerov, upravljanje odločitev, predstavitev znanja v obliki odločitvenih pravil itn. Znotraj področja ekspertnih sistemov smo se odločili za uporabo koncepta inteligentnih tutorskih sistemov (ITS) v kombinaciji s priporočilnimi sistemi.



Slika 3.2

Inteligentni tutorski sistemi (ITS).

Področje ITS predstavlja interdisciplinarno področje, ki povezuje različne znanstvene veščine (slika 3.2 [76]). ITS imajo naslednje lastnosti [76, 77]:

- simulirajo človeškega tutorja z oponašanjem (udejstvom) njegovega znanja in sklepanja,
- inteligentno komunicirajo z uporabniki,
- sklepajo v zvezi s študentovim znanjem na podlagi znanja, ki ga je pokazal med učenjem, vadbo in preverjanjem znanja,
- razlagajo študentovo znanje na podlagi njegovih interakcij s sistemom in analizirajo doseganje učnih ciljev, ki smo si jih zadali med učenjem,

Idejna zasnova našega računalniškega sistema temelji na lastnostih ITS za spremljanje učenčevih vedenjskih vzorcev pri učenju s pomočjo sistema. Po vsakem formativnem preverjanju znanja bi sistem preveril, katere operativne učne cilje je učenec že dosegel. Po določenem zaporedju učenčevih učnih iteracij bi sistem prikrl tiste učne objekte, ki jih učenec ne potrebuje več, saj je v danem trenutku že dosegel posamezne učne cilje. Seveda pa bo lahko še vedno do njih dostopal, če bi si želel osvežiti znanje. Po nekem zaporednem ciklu formativnega (sprotnega) preverjanja znanja bi se končala faza učenja in bi se učenčevo znanje preverilo s sumativnim (končnim) preverjanjem znanja.

### 3.2.4 *Priporočanje/Prilagajanje predstavitve učnih objektov*

Zasnova našega sistema, ki smo ga zasnovali v okviru te doktorske disertacije, temelji na dejstvu, da priporočanje učnih objektov in izbiranje ustreznih multimedijskih tipov učnih gradiv pripomoreta k učinkovitejšemu učenju. S priporočanjem tipa predstavitve učne vsebine oz. učnih objektov želimo učencu olajšati iskanje ustreznih učnih gradiv.

Naše izhodišče je, da sicer veljajo korelacije med učnimi stili in z njimi povezanimi multimedijskimi tipi, vendar se hkrati zavedamo, da vsake učne snovi ne moremo predstaviti vedno z istim multimedijskim tipom. Pokazali smo [3], da se učenci pogosteje učijo z besedilnimi učnimi objekti, kar je po svoje razumljivo, saj gre za naučene vedenjske vzorce učenja. Po drugi strani pa se tudi zavedamo, da vseeno obstajajo pristopi, s pomočjo katerih se lahko učenci bistveno lažje in učinkovitejše učijo. Zato naša idejna zasnova sistema temelji na spremljanju ocenjevanja učnih objektov — kateri učni objekti se zdijo učencem najuporabnejši za učenje neke učne snovi.

Priporočanje predstavitev učnih objektov temelji na iskanju podobnosti med učenci, za kar lahko uporabljamo metodo priporočanja z matrično faktorizacijo, ki jo predstavimo v poglavju 4. Zakaj iščemo podobnosti med učenci? Trdimo namreč, da se učenci s podobnimi učnimi preferencami podobno učijo. Če sta dva učenca podobno ocenila učne objekte v preteklosti, je zelo velika verjetnost, da bosta podobno ocenila tudi tiste učne objekte, ki jih do sedaj še nista ocenila.

V učnem procesu je ključno, da učenec dobi nasvete oziroma priporočila, katere učne objekte in katere njegove predstavitve naj izbere, da bo z njihovo pomočjo dosegel operativne učne cilje. Doseganje operativnih učnih ciljev pa preverjamo v fazi sprotnega in končnega preverjanja znanja.

### 3.2.5 *Formativno preverjanje znanja*

Formativno preverjanje znanja je oblika sprotnega preverjanja znanja, ki podpira proces učenja in je namenjena učencu ter učitelju. S pomočjo formativnega preverjanja znanja lahko sistem ugotovi delne cilje — preveriti mora, kaj učenec že zna in česa še ne zna (“Terminološki slovar vzgoje in izobraževanja” [78]).

V sistemu bi bilo formativno preverjanje znanja realizirano kot praktične naloge, ki bi bile vključene v posamezen učni objekt. Z reševanjem nalog bi lahko učenec sproti preveril, ali razume in zna uporabiti pridobljeno znanje v praksi. Glede na doseganje

učnih ciljev bi sistem priporočal učne objekte, kot smo pojasnili v razdelku 3.2.3. Sumativno preverjanje znanja pa je končno ugotavljanje doseganja ciljev — gre za učne dosežke, ki se običajno ocenjujejo. Sumativno ocenjevanje se izvede ob koncu večjih učnih enot. Rezultati se vrednotijo glede na postavljene standarde znanja skladno z vnaprej predstavljenimi kriteriji (“Terminološki slovar vzgoje in izobraževanja” [78]).

### 3.3 *Uporaba spletnega učnega sistema v izobraževanju*

V prejšnjem razdelku smo predstavili zasnovano računalniškega sistema za podporo konstruktivističnemu učenju, in sicer sistema, ki bi bil namenjen samostojnemu učenju. Učenec bi se prijavil v sistem, v katerem bi moral na začetku rešiti vprašalnike za detekcijo učnih stilov posameznih modelov učnih stilov. Po uvodni registraciji bi imel učenec na voljo množico multimedijskih gradiv in drugih učnih aktivnosti za posamezno učno enoto.

Za primer vzemimo učenca, ki želi rešiti programerski problem znotraj učne vsebine “Osnove programiranja”. Če želi učenec rešiti problem, mora imeti temeljna znanja iz programiranja in si pridobiti temeljne veščine programiranja v izbranem programskem jeziku. Predpostavimo, da se v nekem trenutku učenec želi naučiti pomen ključnih pojmov iz programiranja (npr. spremenljivka, odločitveni stavek, zanka itn.). Učenec bi se prosto gibal po spletišču in pregledoval učna gradiva, ki vsebujejo podatke, ki jih učenec potrebuje za reševanje njegovega problema. Predpostavimo, da je učenec izbral učno temo “Kaj je spremenljivka?”. Spletno učno okolje bi učencu priporočilo gradiva iz množice vseh multimedijskih gradiv za izbrano učno enoto. Po koncu ogleda učnega gradiva bi učenec ocenil učinkovitost in uporabnost gradiva, ki ga je preučeval. Tako bi priporočilni sistem prejel uporabnikovo (učenčevo) povratno informacijo. Te ocene objektov bi poleg drugih podatkov o učencu (učni stili, učenčevo predznanje, doseženo znanje ipd.) predstavljale vhodne podatke za priporočilni sistem, ki bi priporočal učne objekte, ki bi bili za učenca najprimernejši.

Poleg učenja s pomočjo multimedijskih učnih gradiv bi učenec skozi ves proces učenja reševal manjše naloge, ki bi bile namenjene formativnemu (sprotnemu) preverjanju znanja. Sprotno preverjanje znanja bi učencu služilo kot povratna informacija, ali naučeno snov ustrezno razume. Hkrati pa bi spletno učno okolje prejelo povratno informacijo, kako primerna so bila za učenca priporočena multimedijska gradiva. Proces učenja bi trajal toliko časa, dokler sistem ne bi presodil, da je učenec dosegel vse vnaprej določene učne cilje. Po koncu učnega procesa bi sledilo sumativno preverjanje znanja.

# *Priporočilni sistemi*

*Če ne premoreš reči nekaj lepega,  
potem ne spregovori.  
(iz risanega filma "Bambi", 1942)*

Sistemi za priporočanje v splošnem temeljijo na reševanju problema, ki je sestavljen iz trojice  $(U, I, R)$ , kjer množica  $U$  predstavlja množico vseh uporabnikov v sistemu, množica  $I$  predstavlja množico vseh objektov, katerim uporabniki pripisujejo neko oceno  $r$ , množica  $R$  pa hrani vse ocene  $r$ , ki so jih uporabniki namenili posameznim objektom [79].

Če problem matematično formuliramo, je priporočanje z matrično faktorizacijo funkcija  $f$ , za katero velja:

$$f : U \times I \rightarrow R.$$

Trojica  $(u, i, r)$  predstavlja zapis, da je uporabnik  $u$  objektu  $i$  priredil oceno  $r$ . Cilj reševanja problema je torej, da zna sistem čim natančneje predvideti oceno  $\hat{r}$  za nekega uporabnika  $u$  in objekt  $i$ .

Področje sistemov za priporočanje se običajno deli v tri kategorije: 1) algoritmi za vsebinsko izbiranje (angl. "content-based filtering"); 2) algoritmi za skupinsko izbiranje (angl. "collaborative filtering"); 3) hibridni algoritmi, ki na različne načine združujejo elemente prvih dveh kategorij [79].

Priporočilni sistemi, ki temeljijo na vsebinskem izbiranju, priporočajo objekte glede na to, katere objekte je neki uporabnik v preteklosti že ocenil kot zanj primerne. Temeljijo torej na uporabnikovi pretekli uporabi objektov znotraj sistema. Druga skupina so algoritmi za izbiranje s sodelovanjem, ki predpostavljajo, da so si uporabniki med seboj podobni, kar pomeni, da bosta dva uporabnika, ki sta v preteklosti podobno ocenjevala objekte, z veliko verjetnostjo podobno ocenjevala objekte, ki jih še nista ocenila. Za hibridne priporočilne sisteme je značilno, da kombinirajo lastnosti obojih, že predstavljenih konceptov. Lahko gre za združevanje obeh skupin algoritmov ali pa se en koncept uporabi samo za uporabnike v stanju hladnega zagona, drugi pa za uporabnike izven omenjenega stanja [79].

Večina računalniških sistemov za priporočanje, ki so bili razviti za uporabo v izobraževanju, temelji na priporočanju učnega objekta, ki ga bodo obiskali učenci v naslednjem koraku [80], ali pa kot sistem za priporočanje učnih objektov s ciljem usmerjanja

učencev za doseganje posameznega učnega cilja [81].

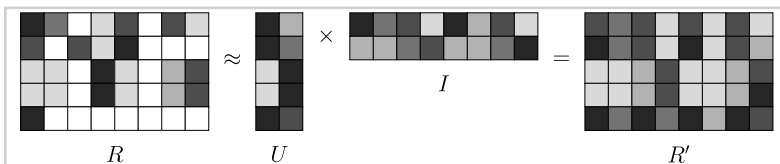
Večina razvitih računalniških sistemov, ki temeljijo na prilagajanju, uporablja algoritme za izbiranje s sodelovanjem za izvedbo priporočanja učencem v učnem omrežju (angl. “learning network”) [82, 83]. Kritika omenjenih sistemov je, da so namenjeni le lokalni uporabi, kar otežuje njihovo primerjanje, hkrati pa sistemi temeljijo na osnovnih tehnikah strojnega učenja in zato raziskovalci predlagajo, da bi uporabili še kakšne kompleksnejše pristope [69].

#### 4.1 Priporočilni sistemi in matrična faktorizacija

Algoritmi za izbiranje s sodelovanjem se v grobem delijo na: 1) metode na osnovi matričnih faktorizacij (angl. “matrix factorization”) in 2) metode na osnovi  $k$ -najbližjih sosedov [84].

V tem delu smo se odločili za uporabo matrične faktorizacije v kontekstu priporočilnih sistemov, ker metoda hkrati povzema karakteristike uporabnikov in tudi objektov. Metode matričnih faktorizacij so torej družina metod. Temeljijo na matrični predstavitvi podatkov, priporočanje na njihovi osnovi pa temelji na napovedovanju vrednosti celotne vhodne matrike. Cilj matrične faktorizacije je aproksimacija vhodne matrike  $R$  (velikosti  $m \times n$ ) kot produkta dveh manjših matrik:  $R \approx U \cdot I$ , kjer je matrika  $U$  velikosti  $m \times k$  in matrika  $I$  velikosti  $k \times n$ , kakor prikazuje slika 4.1. Matriki  $U$  pravimo tudi faktorska matrika uporabnikov, v kateri vsaka vrstica predstavlja enega od uporabnikov v priporočilnem sistemu. Matriki  $I$  pa pravimo faktorska matrika objektov, v kateri vsaka vrstica predstavlja enega od objektov v priporočilnem sistemu. [85].

Rezultat matrične faktorizacije sta torej matriki  $U$  in  $I$ , kjer je  $R'$  rezultat njenega skalarnega produkta. Hkrati pa  $R'$  predstavlja aproksimacijo originalni matriki  $R$ . Slika 4.1 prikazuje grafično predstavitev uporabe matrike  $R'$  za priporočanje atributov, ki še nimajo vrednosti, tj. ocene.



Slika 4.1

Primer matrične faktorizacije – različno osenčene celice predstavljajo različne vrednosti v matriki. Slika predstavlja primer uporabe matrike  $R'$  za priporočanje neznanih ocen (belih celic) v izvorni matriki  $R$ .

Najpogosteje uporabljene različice matričnih faktorizacij so: nenegativna matrična faktorizacija s stohastičnim gradientnim spustom (angl. “non-negative matrix factorization with stochastic gradient descent” [84, 86]), nenegativna matrična faktorizacija z izmenjujočimi se najmanjšimi kvadrati (angl. “non-negative matrix factorization with alternating least squares” [84, 87, 88]) in polnenegativna matrična faktorizacija z manjkajočimi vrednostmi (angl. “semi-non-negative matrix factorization with missing data” [89, 90]). Omenjene metode so namenjene predvsem uporabi v primerih, ko nimamo uporabnikov, ki so v stanju hladnega zagona. V nadaljnjih razdelkih od 4.1.1 do 4.1.4 so predstavitev omenjenih metod.

#### 4.1.1 Nenegativna matrična faktorizacija s stohastičnim gradientnim spustom (NG)

Nenegativna matrična faktorizacija s stohastičnim gradientnim spustom (NG) (angl. “non-negative matrix factorization with stochastic gradient descent” [84, 86]) temelji na iteracijah skozi vse obstoječe ocene vhodne matrike  $R$  – za vsako obstoječo oceno (vrednost) sistem izračuna njeno napoved in ustrezno napako napovedi. Cilj algoritma je minimizirati naslednjo funkcijo – kvadratno napako napovedi na dani učni množici znanih ocen:

$$\frac{1}{2} \sum_{(u,i) \in R} (r_{ui} - p_u q_i)^2 + \lambda (\|p_u\|^2 + \|q_i\|^2), \quad (4.1)$$

kjer  $r_{ui}$  predstavlja znano oceno za uporabnika  $u$  in objekt  $i$ ,  $p_u q_i$  predstavlja napovedano oceno – skalarni produkt vektorja  $p_u$  za uporabnika  $u$  in vektorja  $q_i$  za objekt  $i$ ,  $\lambda$  pa predstavlja regularizacijski faktor.

Simon Funk [86] je opisal metodo za optimizacijo napake 4.1. Pri vsaki iteraciji sistem za vsako oceno v učni množici izračuna napako za napoved vrednosti znane ocene  $r_{ui}$ :

$$e_{ui} = r_{ui} - \hat{r}_{ui} = r_{ui} - p_u q_i \quad (4.2)$$

Vsakokrat ko izračuna napako, popravi vrednosti faktorja, ki se ga v danem trenutku uči. Metoda popravi faktor tako, da upošteva hitrost učenja (angl. “learning rate”)  $\gamma$  in regulacijski faktor  $\lambda$  (angl. “regularization factor”) [84, 85] – algoritem v enem koraku popravi vektor  $p_u$  uporabnika  $u$  in vektor  $q_i$  za objekt  $i$  na naslednji način:

$$\begin{aligned} p'_u &= p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u) \\ q'_i &= q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i) \end{aligned} \quad (4.3)$$



Algoritem NG popravlja ocene v vsaki iteraciji, dokler ne doseže približka globalnemu minimumu tj. dokler se v vsaki iteraciji kvadratna napaka napovedi na dani učni množici znanih ocen zmanjšuje (enačba 4.1).

Prednost algoritma je, da deluje tudi na redkih vhodnih matrikah — to so matrike, ki vsebujejo zelo malo ocen oz. imajo prazne celice. Po drugi strani pa je algoritem časovno zelo potraten, saj temelji na iskanju globalnega minimuma celotne matrike, kar pomeni, da v vsaki iteraciji algoritem za vsako podano obstoječo oceno izračuna njeno napoved in napako. Ta lastnost lahko vodi do dolgotrajnega čakanja konca algoritma, še posebno če imamo velike matrike — algoritem se lahko izvaja tudi več ur ali celo več dni [84–86].

#### 4.1.2 *Nenegativna matrična faktorizacija z izmenjujočimi se najmanjšimi kvadrati (NS)*

Nenegativna matrična faktorizacija z izmenjujočimi se najmanjšimi kvadrati (NS) (angl. “non-negative matrix factorization with alternating least squares” [87, 88]) temelji na iteracijah, v katerih izmenično računamo eno izmed faktorskih matrik  $U$  in  $I$  z namenom, da dosežemo lokalni minimum funkcije:

$$\frac{1}{n} \sum_{(u,i) \in R} (r_{ui} - p_u q_i)^2, \quad (4.4)$$

kjer je  $r_{ui}$  znana ocena (iz množice vseh znanih ocen  $R$ ) za uporabnika  $u$  in objekt  $i$ ,  $p_u q_i$  predstavlja napovedano oceno — skalarni produkt vektorja  $p_u$  za uporabnika  $u$  in vektorja  $q_i$  za objekt  $i$ .

Potek učenja na dani učni množici (z znanimi ocenami) deluje tako, da izmenjujoče fiksiramo eno izmed matrik  $U$  in  $I$  za popravljanje druge matrike. Ko računamo oz. popravljamo matriko  $U$ , fiksiramo matriko  $I$ , tako da pri izračunu matrike  $U$  minimiziramo funkcijo 4.4. Ko računamo matriko  $I$ , fiksiramo matriko  $U$  in izračunamo  $I$ , tako da ponovno minimiziramo funkcijo 4.4. Postopek ponavljamo, dokler funkcija 4.4 ne doseže konvergenca.

Pristop v osnovi ni namenjen redkim matrikam, saj predpostavlja, da imamo za vsakega uporabnika in vsak objekt oceno. Ta problem so raziskovalci rešili, tako da so manjkajočim ocenam vstavili povprečno vrednost po stolpcu in nato izvedli metodo NS [84, 87, 88].

### 4.1.3 Polnenegativna matrična faktorizacija z manjkajočimi vrednostmi (SN)

Metoda polnenegativne matrične faktorizacije z manjkajočimi vrednostmi (SN) (angl. “semi-non-negative matrix factorization with missing data” [89, 90]) loči pozitivni in negativni del matrike (enačba 4.7) vključenih matrik. Cilj algoritma je minimizacija spodnje funkcije:

$$J(U, I) = \min_{U, I} \|(R - UI)\|_F; I \geq 0, \quad (4.5)$$

kjer je  $R$  vhodna matrika vseh znanih ocen,  $U$  factorska matrika uporabnikov in  $I$  factorska matrika objektov.

Algoritem SN deluje podobno kot algoritem NS (razdelek 4.1.2) – metoda iterativno izmenjujoče popravlja matriki  $U$  in  $I$ , dokler se zmanjšuje vrednost napake (enačba 4.5). Metoda SN pri vsaki iteraciji najprej popravi matriko  $U$  in nato matriko  $I$ , in sicer s praviloma:

$$U = RI(I^T I)^{-1}$$

$$I_{ik} \leftarrow I_{ik} \sqrt{\frac{(R^T U)_{ik}^+ + [I(U^T U)^-]_{ik}}{(R^T U)_{ik}^- + [I(U^T U)^+]_{ik}}}, \quad (4.6)$$

kjer je  $R$  vhodna matrika vseh znanih ocen,  $U$  factorska matrika uporabnikov,  $I$  factorska matrika objektov in  $ik$  pomeni, da gledamo  $k$ -to vrednost v vektorju objekta  $i$ .

Pri popravljanju – vnovičnem izračunu matrike  $I$  – ločimo pozitivni in negativni del dane matrike. Za lažje razumevanje predpostavimo, da imamo matriko  $A$ . Glede na naslednji pravili ločimo pozitivni in negativni del matrike:

$$A_{ik}^+ = \frac{|A_{ik}| + A_{ik}}{2},$$

$$A_{ik}^- = \frac{|A_{ik}| - A_{ik}}{2}, \quad (4.7)$$

pri čemer vsako matriko  $A$  preoblikujemo, tako da pri  $A_{ik}^+$  ohranimo prvotno vrednost  $A_{ik}$  v matriki, če je vrednost nenegativno število (v nasprotnem primeru ima vrednost števila nič (0)), in pri  $A_{ik}^-$  ohranimo prvotno vrednost  $A_{ik}$  v matriki, če je vrednost negativna (v nasprotnem primeru ima vrednost števila nič (0)).

Sam pristop ne temelji na redkih matrikah, saj predvideva, da so matrike polne, tj. nimamo praznih celic oz. vrednosti. Zato sta Mo in Draper [90] predstavila razširitev

metode, tako da metoda deluje tudi na redkih matrikah. Cilj metode je minimizirati naslednjo funkcijo:

$$J(U, I) = \min_{U, I} \|W \otimes (R - UI)\|_U; I \geq 0, \quad (4.8)$$

kjer  $W$  označuje matriko uteži – matrika je sestavljena iz ničel in enic, vrednost  $w_{ui} = 0$  na mestu  $u, i$  pomeni, da vhodna matrika  $R$  za uporabnika  $u$  in objekt  $i$  nima ocene. Če ima matrika  $R$  na mestu  $u, i$  oceno, pa ima matrika uteži  $W$  vrednost  $w_{ui} = 1$ . Operator  $\otimes$  v enačbi 4.8 označuje operacijo množenja istoležnih vrednosti v matrikah in operator  $\oslash$  označuje operacijo deljenja istoležnih vrednosti v matrikah. Metoda SN pri vsaki iteraciji najprej popravi matriko  $U$ , in sicer s praviloma:

$$\begin{aligned} R &= ((W \otimes R)I) \oslash ((W \otimes UI^T)I) \\ U &= U \otimes R \end{aligned} \quad (4.9)$$

V naslednjem koraku (znotraj iste iteracije) metoda SN popravi matriko  $I$ , in sicer s pravili:

$$\begin{aligned} R_1 &= ((W^T \otimes R^T)U)^+ + ((W^T \otimes IU^T)U)^- \\ R_2 &= ((W^T \otimes R^T)U)^- + ((W^T \otimes IU^T)U)^+ \\ I &= I \otimes R_1 \oslash R_2 \end{aligned} \quad (4.10)$$

Metoda je dobra alternativa metodi NG (razdelek 4.1.1), ker je izjemno hitra in daje sorazmerno dobre rezultate. Prednost metode je tudi, da brez večjih težav izračuna matrično faktorizacijo za večje podatkovne baze, pri katerih je matrika večja in ima tudi večje število začetnih vrednosti [89, 90].

#### 4.1.4 Matrična faktorizacija z zlivanjem podatkovnih virov (DF)

Žitnik in Zupan [91, 92] sta šla korak naprej pri razvoju matričnih faktorizacij, saj sta predlagala matrično faktorizacijo z zlivanjem podatkovnih virov (DF) (angl. “matrix factorization by data fusion”). Algoritem lahko združi heterogene podatkovne vire, ki jih predstavimo z matrikami.

Algoritem v prvem koraku organizira vhodne podatke (v nadaljevanju vhodne matrike) v bločno predstavitev. Matrična faktorizacija poteka simultano na vseh vhodnih matrikah, in sicer tako, da si nižjenivojski matrični faktorji med seboj delijo trifaktorsko dekompozicijo vhodnih matrik.

Trifaktorska matrična faktorizacija temelji na dekompoziciji vhodne matrike  $R$  na produkt treh nižjedimenzionalnih faktorskih matrik, in sicer tako, da velja:

$$R_{xy} \approx G_x S_{xy} G_y^T, \quad (4.11)$$

kjer je  $R_{xy}$  ena izmed podanih matrik za dva različna tipa objektov  $x$  in  $y$  – v našem primeru so to uporabniki in učni objekti,  $G_x$  je faktorska matrika, ki jo sestavljajo vektorji za posamezen objekt  $x$  (vrednosti v vektorju so numerične vrednosti),  $G_y$  je faktorska matrika, ki jo sestavljajo vektorji za posamezen objekt  $y$  (vrednosti v vektorju so numerične vrednosti),  $S_{xy}$  je faktorska matrika. Rezultat matrične faktorizacije z zlivanjem vsebuje matriko  $S_{x,y}$ , ki je specifična za vsako vhodno matriko (podatkovni vir) in matriko  $G_x$ , ki je specifična za vsak podatkovni tip (glede na vrsto entitet oz. atributov v vhodni matriki).

V raziskovalnem delu smo nadgradili omenjeno metodo [91, 92], in sicer za reševanje hladnega zagona na način, kot ga opisujemo v razdelku 5.2.5.

## 4.2 Priporočilni sistemi in hladni zagon

Matrične faktorizacije pogosto uporabljamo v priporočilnih sistemih, saj hkrati izračunajo (manjkajoče) vrednosti za vse uporabnike in vse objekte. Ključna pomanjkljivost matričnih faktorizacij je, da ne dosežajo dobrih rezultatov napovedi, če imamo o uporabniku ali objektu zelo malo ali skoraj nobene ocene. Takrat govorimo o stanju hladnega zagona [93].

V grobem poznamo tri različne scenarije hladnega zagona [93–96]:

- Problem novega uporabnika: nov uporabnik začne uporabljati priporočilni sistem. Ker ta nima vhodnih uporabnikovih ocen, novemu uporabniku težje priporoči objekte.
- Problem novega objekta: v priporočilni sistem damo nov objekt, ker nima ocen, pa ga sistem ne more primerjati (glede na ocene) s preostalimi objekti.
- Problem novega sistema: implementiramo nov priporočilni sistem, v katerem so vsi novi uporabniki in novi objekti. V takem primeru je treba bazo napolniti z množico ocen za ustrezno priporočanje objektov.

V tem raziskovalnem delu smo se osredinili na problem *novega uporabnika*, o katerem ne vemo ničesar ali pa ima sistem premalo ocen, da bi lahko podajal personalizirane in relevantne napovedi oz. priporočila.

V našem delu smo problem novega uporabnika razdelili na tri stanja, med katerimi prehaja nov uporabnik [96]:

1. Absolutni hladni zagon: uporabnik ni podal ocene — vektor novega uporabnika  $u$  v matriki  $R$  je povsem prazen.
2. Delni hladni zagon: uporabnik ima premalo ocen za ustrezno priporočanje objektov — vektor novega uporabnika  $u$  ima premalo ocen — vektor je reddek.
3. Stabilno stanje: uporabnik ima dovolj ocen — vektor  $u$  ni več reddek in ga lahko uporabljamo za napovedovanje oz. priporočanje objektov uporabniku  $u$ .

V grobem lahko obstoječe načine reševanja hladnega zagona razdelimo v dve skupini, in sicer na: 1) skupino algoritmov, ki rešujejo zgolj stanje hladnega zagona kot neodvisen problem oz. neozirajoč se na uporabnike, ki niso v hladnem zagonu, in 2) skupino algoritmov, ki dodatno poizvedujejo o uporabnikih.

#### 4.2.1 Sistemi za reševanje problema hladnega zagona

Eden izmed običajnih pristopov reševanja hladnega zagona je oblikovanje ločenega napovedanega sistema, ki bo namenjen samo uporabnikom v hladnem zagonu. Tovrstne rešitve v grobem delijo uporabnike v dve skupini: uporabniki v stanju hladnega zagona in uporabniki zunaj omenjenega stanja. Za vsako skupino raziskovalci oblikujejo lasten algoritem in uporabijo ločeno priporočanje objektov. Čeprav je ta pristop uporaben, je težje nadzorovati posameznega uporabnika in mu slediti, kdaj bomo zanj uporabili prvi algoritem in kdaj drugi.

Leung in sodelujoči [97] so razvili večnivojsko iskanje povezovalnih pravil (angl. “cross-level association rule mining”), ki upošteva hierarhijo med koncepti/objekti v priporočilnem sistemu. Objekti, ki so hierarhično višje, običajno predstavljajo splošnejše koncepte — posledično so močnejše distribuirani kakor objekti, ki se hierarhično nahajajo nižje. Razvili so sistem CLARE, ki je namenjen iskanju večnivojskih povezovalnih pravil za objekte, ki so v stanju hladnega zagona. Priporočanje uporabnikom je sestavljeno iz več korakov. Najprej sistem upošteva uporabnikove preference, in sicer tako, da pogleda, kateri objekti so bili za uporabnika ustrezni, in glede na zgrajena

povezovalna pravila sestavi množico objektov, ki so primerni za priporočanje. Za vsak objekt poznamo njegovo ustreznost, ki predstavlja številčno vrednost (npr. podpora (angl. “support”). Glede na numerične vrednosti povezovalnih pravil sistem izbere  $N$  najustreznejših objektov. Sistem je namenjen reševanju hladnega zagona, vendar moramo imeti za gradnjo povezovalnih pravil zadostno število začetnih podatkov, kar pomeni, da mora sistem že poznati uporabnika oz. objekte. Torej gre za reševanje delnega hladnega zagona (ko že imamo nekaj začetnih ocen).

Raziskovalci so poskusili rešiti stanje hladnega zagona tudi z razvojem novih metrik za iskanje podobnih uporabnikov. Ahn [98] je razvil metriko PIP za iskanje podobnosti med uporabniki. Razvita je bila predvsem za reševanje stanja hladnega zagona, pri katerem ima uporabnik premalo ocen, da bi mu sistem lahko ustrezno priporočal. Metrika PIP je produkt numeričnih vrednosti naslednjih količin: neposredna bližina ocen (angl. “proximity”), vpliv ocenjevanja (angl. “impact”) in priljubljenost objektov (angl. “popularity”). Metrika torej upošteva, kako podobno dva uporabnika ocenjujeta objekte. Podobnost dveh uporabnikov se izračuna tako, da se posamezna mera znotraj metrike PIP izračuna za vsak par ocen dveh objektov — metrika upošteva ocene tistih objektov, ki sta jih oba uporabnika ocenila. To pomeni, da moramo imeti za uporabnika, ki je v stanju hladnega zagona, vsaj dve začetni oceni. Metrika je torej neuporabna za stanje absolutnega hladnega zagona, ko za uporabnika nimamo ocen — njegov vektor je povsem prazen. Liu in sodelujoči [99] so razvili podobno metriko PSS, ki je produkt numeričnih vrednosti naslednjih količin: neposredna bližina ocen (angl. “proximity”), pomembnost (angl. “significance”) in singularnost (angl. “singularity”). Enako kakor za metriko PIP tudi za metriko PSS velja, da moramo poznati nekaj začetnih ocen, če želimo uporabiti omenjeno metriko.

Bobadilla in sodelujoči [100] so prav tako oblikovali novo metriko MJD, in sicer tako, da so linearno povezali preproste (obstoječe) metrike (metrika Jaccard, število podobno ocenjenih objektov, število povsem različno ocenjenih objektov, povprečna razlika ocen objektov, standardni odklon itn.), ki se uporabljajo za računanje oz. ugotavljanja podobnosti dveh (numeričnih) vektorjev. Slabost metrike MJD je, da predvideva, da imamo za uporabnika začetne ocene, kar je neuporabno za reševanje absolutnega hladnega zagona.

Formoso in sod. [96] so reševali problem hladnega zagona tako, da uporabniku, ki je v stanju hladnega zagona, razširijo profil (angl. “profile expansion”), in sicer tako, da dodajo ocene objektom, ki jih sicer še ni ocenil. To pomeni, da sistem na

premišljen način izbere neke objekte za novega uporabnika — izbranim objektom priredi manjkajoče ocene. Govorimo lahko o vstavljanju manjkajočih vrednosti v redek vektor. Formoso in sodelujoči so oblikovali razširitev prvotnega profila glede na dve merili — globalna ali lokalna razširitev profila ter glede na izbran pristop — uporabimo lahko pristop, ki temelji na objektih (angl. “item-based techniques”), ali pristop, ki temelji na uporabnikih (angl. “user-based techniques”). Raziskovalci so testirali naslednje načine razširjanja uporabnikovega začetnega profila: dodajanje manjkajočih ocen (povprečna utežena vrednost ocene objekta), dodajanje povprečne ocene najbolj ocenjenih objektov, dodajanje povprečne ocene največkrat ocenjenih objektov. Poskus je pokazal, da je najprimernejša metoda za reševanje hladnega zagona pristop dodajanja povprečne ocene največkrat ocenjenih objektov. Takšen pristop reševanja hladnega zagona je zelo perspektiven, saj začetnemu profilu dodaja nekaj ocen in tako začasno reši problem hladnega zagona, saj lahko sistem povsem običajno priporoča objekte. Težava lahko nastane, če uporabnikom v hladnem zagonu dodajamo takšne ocene, ki lahko sistem zavedejo, in posledično sistem priporoča neustrezne objekte. Pogost problem priporočilnih sistemov so t. i. “popularni” produkti, saj imajo veliko ocen in običajno uporabniki take produkte zelo visoke ocenjujejo. S tem lahko sistem priporočanja zavedemo, saj bo uporabniku priredil neki “okus”, ki je lahko bistveno drugačen od resničnega. Če uporabniku dodamo ocene objektov, ki imajo najboljše ocene, bomo našega uporabnika navidezno naredili preveč podobnega neki množici uporabnikov. Nekateri produkti so vsečni večji množici uporabnikov, zato bi bilo primerneje, če uporabniku dodamo reprezentativnejše ocene, saj ga bomo tako lažje ločili od večje množice “navidezno” podobnih uporabnikov.

Če povzamemo: raziskovalci so ločili uporabnike na dve skupini: na tiste, ki so v stanju hladnega zagona, in na tiste, ki so zunaj stanja hladnega zagona. Oblikovali so nove metrike, ki sicer zelo uspešno rešujejo problem delnega hladnega zagona, vendar pa ne rešujejo problema absolutnega hladnega zagona. Pri večini metrik moramo za uporabnika, ki je v stanju hladnega zagona, poznati vsaj dve oceni. Postavijo pa se vprašanja, kako pridobiti prvi dve oceni oz. kako naj uporabniku priporočamo do trenutka, ko bo že ocenil vsaj dva objekta. Raziskovalci so ta problem reševali tako, da so uvodoma poizvedovali o uporabniku. V naslednjem razdelku sledi pregled področja reševanja hladnega zagona z uvodnim poizvedovanjem.

#### 4.2.2 Reševanje problema hladnega zagona z uvodnim poizvedovanjem

Druga skupina pristopov reševanja hladnega zagona temelji na dodatnem poizvedovanju o uporabniku. Glede na namen uvodnega poizvedovanja se pristopi delijo v dve skupini: 1) poizvedovanje o demografskih podatkih uporabnika; 2) uvodno pridobivanje (začetnih) ocen.

Rashid in sod. [101, 102] so uporabili pristop spraševanja uporabnikov, pri katerem so uporabnika vprašali za pet produktov, kako bi jih ocenil. Razvili so sistem, ki uporabniku premišljeno ponuja prvih pet produktov, ki jih uporabnik oceni takoj ob prvi prijavi v sistem. Ko produkte oceni, sistem pridobi zadostno število ocen. Tako uporabnik že takoj ob vstopu pridobi zadostno število ocen in zato ne pride do stanja hladnega zagona. Slabost pristopa je, da je težko izbrati, katere produkte mora novi uporabnik oceniti, saj lahko oceni produkte, ki so priljubljeni — s tem sistem ne bo pridobil novih in ključnih podatkov o uporabniku, ker bo dobil podatke, ki bi jih lahko tudi sam predvidel. Hkrati lahko potegnemo vzporednico z dejstvom, da sistem ne bo mogel razbrati uporabnikovega “okusa”.

Alternativa začetnemu ocenjevanju izbranih atributov (objektov) je pridobivanje sekundarnih podatkov o uporabniku — to so podatki, ki niso uporabnikove ocene objektov, ampak karakteristike uporabnika, ki lahko sistemu služijo za iskanje podobnih uporabnikov glede na osebnostne značilnosti. Z omenjenim pristopom so raziskovalci oblikovali začetni uporabnikov profil [95]. Na podoben način so Lika in sod. [103] uporabili splošne klasifikacijske metode za gručenje uporabnikov glede na demografske podatke.

Rosli in sodelujoči [104] so za reševanje hladnega zagona povezali njihov priporočilni sistem s portalom Facebook — za uporabnika so pogledali, katere filme je označil, da so mu všeč. S pomočjo izračuna kosinusne razdalje so izračunali podobnosti med uporabniki in tako lažje priporočali filme novim uporabnikom. Pridobivanje začetnih podatkov je lahko časovno potratno in ni nujno, da sistemu pomaga do dobrega priporočanja novim uporabnikom.

#### 4.3 Novi pristop k reševanju hladnega zagona

V našem raziskovalnem delu predlagamo alternativo obema skupinama reševanja hladnega zagona. Naš pristop ne predvideva začetnega poizvedovanja in prav tako ne ločenega obravnavanja uporabnikov, ki so v hladnem zagonu. Naša rešitev teme-



lji na ideji “popravljanja” vhodne matrike, tako da uporabnikom v hladnem zagonu smiselno vstavimo nekaj začetnih manjkajočih vrednosti in nato izvedemo matrično faktorizacijo.

Naša ideja temelji na pristopu delno nadzorovanega učenja, pri katerem za priporočanje objektov upoštevamo vse podatke, tudi neoznačene [1]. Na podoben način torej želimo pripraviti neoznačene podatke (primere oz. uporabnike v hladnem zagonu), ki jih uporabimo skupaj z označenimi podatki z namenom izboljšanja procesa učenja priporočilnega sistema [2].

Naša rešitev je podprta tudi s konceptom priporočilnih sistemov. Ti sistemi temeljijo na skupinskem izbiranju, saj rešitev predpostavlja, da podobni uporabniki podobno ocenjujejo objekte. Ko sistem dobi množico podobnih uporabnikov, lahko lažje izbere, katerim atributom lahko vstavimo manjkajočo vrednost, tako da izboljšamo delovanje matrične faktorizacije.

V okviru predlaganega pristopa sklepamo o manjkajočih vrednostih na podlagi podobnih uporabnikov in ocen, ki so jih podobni uporabniki dali posameznim objektom — glede na to se odločimo, katerim atributom (objektom) bomo vstavili manjkajočo oceno, in izračunamo manjkajoče vrednosti. Pristop je podoben delu Chena in sodelujočih [105], ki so pokazali, da z gradnjo omrežij podobnih uporabnikov lahko objekte uspešno priporočimo uporabnikom v hladnem zagonu. Razširanje uporabnikovega profila z objekti, ki so podobni že ocenjenim objektom, pa je tudi ključno za izboljšanje priporočanja uporabnikom v hladnem zagonu [96].



*Reševanje problema hladnega  
zagona*

5

*Obupavanje je za začetnike.  
(iz risanega filma "Hercules", 1997)*

V tem poglavju opisujemo metodologijo za reševanje stanja hladnega zagona pri novih uporabnikih. V našem priporočilnem sistemu smo se želeli izogniti začetnemu spraševanju, zato smo se pri matrični faktorizaciji osredinili na nove uporabnike – problem redkega vektorja. Znotraj raziskovanja smo se osredinili predvsem na razvoj lastnega pristopa k reševanju hladnega zagona z namenom izboljšanja priporočanja uporabnikom v delnem in v absolutnem hladnem zagonu.

Za lažje razumevanje celotnega metodološkega postopka reševanja hladnega zagona smo definirali pojma *uporabnik* in *objekt*:

- *uporabnik*: predstavlja primer v podatkovnem viru, oz. če si predstavljamo podatkovni vir kot matriko, je uporabnik  $u$  predstavljen z  $u$ -to vrstico, pri čemer so v vrstici  $u$  podane ocene za posamezne objekte, ki jih je že ocenil;
- *objekt*: predstavlja atribut oz.  $i$ -ti stolpec matrike, v katerem so ocene, ki jih uporabniki podajajo za objekt  $i$ .

### 5.1 Vstavljanje manjkajočih vrednosti v vhodno matriko

Naša ideja je oblikovati pristop, ki bo smiselno vstavil le nekatere manjkajoče ocene v vhodno matriko, in sicer samo novim uporabnikom, tj. redkim vektorjem. Z omejenim pristopom dodamo nekaj ocen, kar nam omogoči, da lahko nato uporabimo algoritme matrične faktorizacije.

Kot motivacijski primer, ki prikazuje, kako lahko vstavljanje vrednosti izboljša napovedno točnost, si naprej pogledjmo nekaj manjših kontroliranih poskusov. Slika 5.1 prikazuje razširjanje sprememb skozi matriko ob več iteracijah matrične faktorizacije pri petih različnih scenarijih. Prvi stolpec predstavlja vhodno matriko, preostali trije stolpci pa rezultat matrične faktorizacije po 10, 25 in vseh iteracijah (algoritem doseže lokalni/globalni minimum). Za poskus smo uporabili matrično faktorizacijo z gradientnim spustom (NG).

Oblikovali in preverili smo delovanje matrične faktorizacije NG za naslednje eksperimente:

- a) Polna vhodna matrika z enakimi vrednostmi: namen eksperimenta je opazovanje delovanja matrične faktorizacije skozi iteracije in pridobiti referenčen primer za primerjavo s primerom b). Opazimo, da je matrična faktorizacija zelo uspešno napovedala ustrezne ocene za vse uporabnike oz. objekte v matriki.
- b) V vhodno matriko iz primera a) na izbrano mesto vstavimo bistveno večjo vrednost in opazujemo vpliv vrednosti na druge vrednosti v matriki. Že po prvih 25 iteracijah je razvidno, da vstavljena ocena bistveno vpliva na vrednosti v stolpcu in vrstici, v kateri se ta ocena nahaja. Po koncu algoritma pa je razvidno, da se je bistveno poslabšalo priporočanje vseh komponent matrike, saj tudi komponente zunaj izbrane vrstice in stolpca skonvergirajo v vrednost 0.57, namesto v vrednost blizu 1. Iz eksperimenta lahko sklepamo, da z neustreznim vstavljanjem lahko bistveno pokvarimo priporočanje tudi za tiste uporabnike, ki jim ni treba vstaviti manjkajoče vrednosti.
- c) Izmenični vrednosti 1 in 5 ter prisotnost redkega vektorja (brez vrednosti). Oblikovali smo eksperiment, v katerem smo želeli pogledati, kakšen je rezultat napovedovanja za uporabnika v absolutnem hladnem zagonu. Iz eksperimenta je razvidno, da je priporočanje povsem neustrezno – vektor uporabnika se namreč med iteracijami ne spreminja, saj zanj nimamo referenčnih ocen, ki bi bile temelj popravljanja uporabnikovega vektorja v matriki; posledično je to razlog, da je metoda za uporabnike, ki so v stanju hladnega zagona, neuporabna.
- d) Za uporabnika v absolutnem hladnem zagonu iz primera c) vstavimo dve skladni vrednosti. Če pogledamo vhodno matriko, je razvidno, da je naš novi uporabnik glede na poznane ocene podoben uporabniku v drugi vrstici – sklepamo, da bo podobno sklepala tudi metoda matrične faktorizacije, kar tudi dejansko vidimo iz nadaljnjih iteracij. To nas motivira, da z ustreznim načinom vstavljanja manjkajoče ocene lahko vplivamo na ustrežnejše priporočanje priporočilnega sistema.
- e) Primer c) smo predrugačili v redko matriko – nekaterim uporabnikom smo izbrisali nekaj ocen. Po pričakovanjih je metoda matrične faktorizacije brez težav ustrezno napovedala manjkajoče vrednosti uporabnikom, ki so zunaj stanja hladnega zagona. Ker imamo v matriki redek vektor oz. vektor brez vrednosti, je priporočanje za uporabnika v hladnem zagonu povsem neuporabno.

f) Enako kot pri primeru d) v matriko e) za novega uporabnika (njegov vektor je prazen) vstavimo dve vrednosti. Če primerjamo rezultate priporočanja po vseh iteracijah, je razvidno, da je metoda bistveno ustrežnejše napovedala vrednosti. Opazimo, da se priporočanje ni bistveno poslabšalo tudi za preostale uporabnike.

Glede na predstavljene manjše eksperimente in prikaz, kako lahko vstavljanje vrednosti v primerih d) in f) pomaga pri napovedovanju, smo oblikovali lasten inovativni pristop za vstavljanje vrednosti v matriko. Ta na premišljen način izbere, kako vstaviti manjkajoče ocene, da dosežemo večjo napovedno točnost.

## 5.2 Oblikovanje lastnega pristopa za vstavljanje manjkajočih vrednosti za primere v hladnem zagonu

Problem absolutnega hladnega zagona je situacija, v kateri uporabnik ni podal ocene – vektor novega uporabnika  $u$  v matriki  $R$  je povsem prazen. Takšno stanje smo označili z oznako CSo, kar pomeni, da uporabnik nima ocen v svojem vektorju. Kakor smo že napisali v pregledu raziskovalnega področja, so raziskovalci za absolutni hladni zagon uporabili uvodno poizvedovanje, da so pridobili nekaj začetnih ocen ali pa ločene napovedne metode za uporabnike v hladnem zagonu in zunaj njega. Drugi pristopi, ki temeljijo na posebnih metrikah za iskanje podobnih uporabnikov, pa pri absolutnem stanju ne morejo učinkoviteje reševati hladnega zagona, saj uporabnika ne moremo primerjati glede na ocene, ker jih nima.

Z našim pristopom se želimo izogniti uvodnemu poizvedovanju (pridobivanju ocen ali pridobivanju psiholoških/demografskih podatkov). Naš cilj je dopolniti začetno matriko z vstavljanjem nekaj manjkajočih ocen uporabnikom v hladnem zagonu z namenom izboljšati priporočanje objektov z uporabo matrične faktorizacije.

Predlagani postopek reševanja hladnega zagona tvorijo štirje koraki (slika 5.2). Če je uporabnik v stanju absolutnega hladnega zagona, začnemo postopek izvajati s predkorakom 0, sicer pa s korakom 1:

- o. predkorak: uporabniku v stanju absolutnega hladnega zagona vstavimo samo eno oceno, da ga s tem prevedemo v stanje delnega hladnega zagona in lahko uporabimo 1. korak predlaganega postopka. Način vstavljanja vrednosti je predstavljena v razdelku 5.2.1.
1. uporabniku v stanju delnega hladnega zagona poiščemo  $N$  najbolj podobnih uporabnikov;

		št. iter. = 10	št. iter. = 25	po koncu izvajanja algoritma
a)	1 1 1 1 1	0,02 0,02 0,02 0,02 0,02	0,13 0,13 0,13 0,13 0,13	0,96 0,96 0,96 0,95 0,96
	1 1 1 1 1	0,02 0,02 0,02 0,02 0,02	0,13 0,13 0,13 0,13 0,13	0,96 0,96 0,96 0,95 0,96
	1 1 1 1 1	0,02 0,02 0,02 0,02 0,02	0,13 0,13 0,13 0,13 0,13	0,96 0,96 0,96 0,95 0,96
	1 1 1 1 1	0,02 0,02 0,02 0,02 0,02	0,13 0,13 0,13 0,13 0,13	0,96 0,96 0,96 0,96 0,96
b)	1 1 1 1 1	0,02 0,02 0,04 0,02 0,02	0,18 0,18 0,42 0,18 0,18	0,58 0,58 1,60 0,57 0,57
	1 1 5 1 1	0,03 0,03 0,05 0,03 0,03	0,38 0,38 0,89 0,38 0,39	1,48 1,49 4,12 1,47 1,48
	1 1 1 1 1	0,02 0,02 0,04 0,02 0,02	0,18 0,18 0,43 0,18 0,19	0,57 0,57 1,59 0,57 0,57
	1 1 1 1 1	0,02 0,02 0,04 0,02 0,02	0,19 0,19 0,44 0,19 0,19	0,57 0,57 1,59 0,57 0,57
c)	5 1 5 1 5	0,25 0,15 0,26 0,15 0,27	4,49 1,96 4,50 1,96 4,51	4,98 1,03 4,98 1,03 4,99
	1 5 1 5 1	0,17 0,10 0,18 0,10 0,18	2,13 0,93 2,13 0,93 2,13	1,04 4,91 1,02 4,90 1,01
	5 1 5 1 5	0,27 0,17 0,29 0,16 0,30	4,49 1,96 4,50 1,96 4,51	4,99 1,01 4,99 1,01 4,99
	? ? ? ? ?	0,03 0,02 0,04 0,02 0,04	0,15 0,06 0,15 0,06 0,15	0,16 0,05 0,16 0,05 0,16
d)	5 1 5 1 5	0,26 0,22 0,27 0,16 0,30	4,36 2,79 4,37 2,01 4,28	4,99 1,00 4,99 1,00 4,99
	1 5 1 5 1	0,18 0,16 0,19 0,11 0,21	2,31 1,48 2,32 1,07 2,27	1,07 5,14 1,08 4,90 0,80
	5 1 5 1 5	0,28 0,24 0,30 0,17 0,32	4,36 2,80 4,38 2,01 4,28	4,99 1,00 4,99 1,00 4,98
	? 5 ? ? 1	0,12 0,10 0,13 0,07 0,14	2,02 1,29 2,02 0,93 1,98	1,40 4,88 1,41 4,66 1,15
e)	5 ? 5 1 5	0,33 0,17 0,34 0,18 0,36	4,56 2,18 4,73 2,34 4,55	4,98 1,07 4,97 1,11 4,99
	1 5 ? 5 1	0,20 0,11 0,21 0,11 0,21	2,35 1,13 2,44 1,21 2,34	1,03 4,94 3,21 4,94 1,01
	5 1 5 ? 5	0,36 0,18 0,36 0,20 0,39	4,59 2,20 4,76 2,35 4,58	4,98 1,10 4,99 1,14 4,99
	? ? ? ? ?	0,05 0,02 0,05 0,03 0,05	0,17 0,08 0,18 0,09 0,17	0,17 0,09 0,20 0,09 0,17
f)	5 ? 5 1 5	0,34 0,25 0,34 0,19 0,38	4,53 3,44 4,78 2,46 4,41	4,93 1,70 5,11 1,01 4,92
	1 5 ? 5 1	0,22 0,17 0,23 0,13 0,25	2,61 2,00 2,76 1,43 2,54	1,08 5,16 3,45 4,88 0,86
	5 1 5 ? 5	0,37 0,28 0,37 0,21 0,42	4,36 3,31 4,60 2,37 4,24	5,03 1,05 4,88 0,37 5,05
	? 5 ? ? 1	0,15 0,12 0,16 0,09 0,17	2,07 1,58 2,18 1,13 2,01	1,33 4,87 3,52 4,56 1,12

Slika 5.1

Vpliv vstavljanja vrednosti na delovanje/priporočanje matrične faktorizacije: za vsak primer smo pogledali rezultat priporočanja, in sicer po 10 zaporednih iteracijah in ko se algoritem povsem konča. Znak '?' označuje, da za danega uporabnika (vrstica) in za dani objekt (stolpec) ni podane vrednosti.

2. na osnovi podobnih uporabnikov iz prejšnjega koraka določimo attribute/objekte, ki jim bomo vstavili manjkajoče vrednosti;
3. izbranim atributom glede na izbrano metodo izračunamo/vstavimo manjkajoče ocene;
4. uporabimo izbrano matrično faktorizacijo nad vsemi uporabniki.

### 5.2.1 Vstavljanje ene začetne manjkajoče vrednosti za uporabnike v absolutnem hladnem zagonu

V predkoraku (korak 0) želimo prevesti stanje absolutnega hladnega zagona (označimo ga s  $CS_0 - 0$  podanih ocen) v stanje delnega hladnega zagona (označimo ga s  $CS_1' - 1$  umetno vstavljena ocena), kar pomeni, da bomo za uporabnika (nahaja se v absolutnem hladnem zagonu) poiskali tisti objekt, ki mu lahko z največjo zanesljivostjo vstavimo eno oceno in ga tako privedemo v stanje delnega hladnega zagona.

Prevedbo stanja  $CS_0$  v stanje  $CS_1'$  smo razdelili na dva koraka:

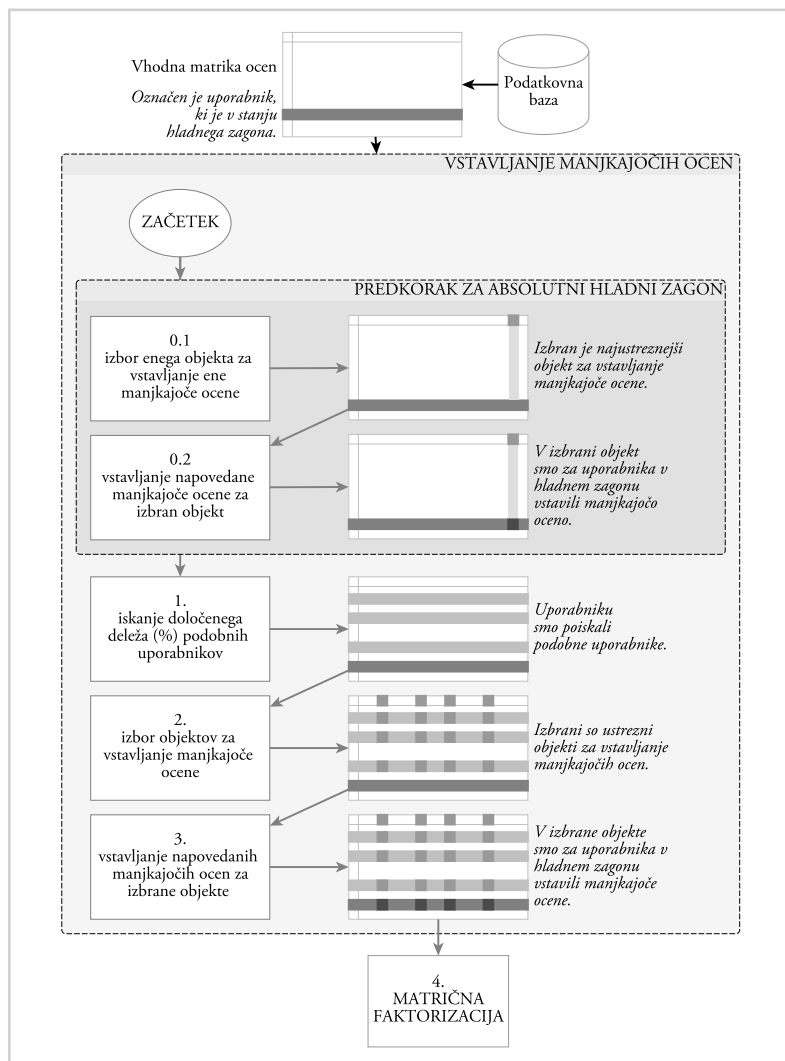
1. Znotraj množice uporabnikov zunaj hladnega zagona določimo atribut/objekt, ki mu bomo vstavili manjkajočo vrednost. Kriterija, ki sta primerna za izbor ustreznega atributa za uporabnika v absolutnem hladnem zagonu, sta standardni odklon (SD) in frekvenca najpogostejše vrednosti v atributu (FR). Oba kriterija sta predstavljena v razdelku 5.2.3.
2. Izbranemu atributu glede na izbrano metodo izračunamo/vstavimo manjkajočo oceno. Primerni metodi za vstavljanje manjkajoče vrednosti za uporabnika v absolutnem hladnem zagonu sta izračunana povprečna ocena (ME) in najpogostejša vrednost (MF). Obe metodi sta predstavljena v razdelku 5.2.4.

V nadaljevanju so predstavljeni koraki 1–4 (glede na sliko 5.2) predlaganega algoritma.

### 5.2.2 Iskanje podobnih uporabnikov

Za uporabnika (v delnem hladnem zagonu) v tem koraku poiščemo njemu podobne uporabnike. Če je uporabnik v stanju absolutnega hladnega zagona, ga najprej privedemo v stanje delnega hladnega zagona s predkorakom, predstavljenim v prejšnjem





Slika 5.2

Koraki predlaganega postopka za vstavljanje vrednosti uporabniku v hladnem zagonu.

razdelku (razdelek 5.2.1). V nadaljevanju lahko nato uporabnikom, ki že imajo nekaj začetnih ocen (so v stanju delnega hladnega zagona), poiščemo njim podobne uporab-

nike.

Za iskanje uporabniku podobnih uporabnikov smo uporabili evklidsko razdaljo, ki izračuna podobnost med dvema uporabnikoma in primerja njune ocene za iste objekte. Danega uporabnika v hladnem zagonu smo primerjali z vsemi uporabniki zunaj omenjenega stanja. Ko smo dobili množico podobnih uporabnikov, smo uporabnike v dobljeni množici razvrstili glede na podobnost in nato izbrali različne deleže podobnih uporabnikov, ki jih upoštevamo v naslednjem koraku — izbor ustreznih atributov. Določili smo odstotke najpodobnejših uporabnikov, ki jih uporabljamo v nadaljnjih fazah: 1 %, 5 %, 10 %, 25 % in 50 %.

### 5.2.3 Izbor ustreznih atributov

Za uporabnika (v hladnem zagonu) v naslednjem (drugem) koraku sistem izbere najustreznejše attribute (objekte, ki jih priporočamo), ki jim bomo vstavili manjkajočo vrednost. Za ta korak smo izbrali štiri različne kriterije izbora teh atributov, ki jim bomo v naslednjem koraku vstavili manjkajočo oceno. Attribute razvrstimo glede na kriterije in izberemo 10 % najustreznejših atributov. Prag 10 % smo izbrali z namenom, da ne vstavimo prevelikega števila manjkajočih ocen. Ker je razsežnost prostora različnih parametrov metode velika, smo analizo vpliva tega parametra pustili za nadaljnje delo.

Prva dva kriterija za izbor ustreznih atributov temeljita na preprostih statističnih funkcijah in sta primerna za uporabnike v delnem pa tudi v absolutnem hladnem zagonu:

- *SD* (standardni odklon — angl. “standard deviation”): za vsak atribut (stolpec) izračunamo standardni odklon. Attribute razvrstimo glede na vrednost standardnega odklona in izberemo 10 % atributov z najmanjšim standardnim odklonom. Ideja za tem kriterijem je, da želimo izbrati takšne attribute, ki jim lahko z največjo mogočo verjetnostjo napovemo pravo oceno. Če ima atribut majhno standardno deviacijo, lahko sklepamo, da uporabniki ta objekt ocenjujejo podobno, in je velika verjetnost, da ga bo podobno ocenil tudi uporabnik v hladnem zagonu, saj gledamo uporabnike, ki so njemu podobni glede na ocenjevanje objektov;
- *FR* (frekvenca najpogostejše vrednosti v atributu — ang. “frequent values”): druga možnost izbora atributa je glede na frekvenco najpogostejše vrednosti atributa

ta. Za vsak atribut določimo najpogostejšo vrednost/oceno in nato preštejemo oz. izračunamo frekvenco te ocene. Nato izberemo 10 % atributov z najvišjo vrednostjo frekvence najpogostejše vrednosti. Predpostavljamo, da večja kot je frekvenca najpogostejše vrednosti, z večjo gotovostjo lahko vstavimo najpogostejšo vrednost v izbrani atribut v naslednjem koraku.

Za preostala dva kriterija smo izbrali naprednejša kriterija za izbor atributov, ki se tudi sicer v strojnem učenju uporabljata za ocenjevanje kakovosti atributov, primerna pa sta le za uporabnike v delnem hladnem zagonu:

- *IG* (informacijski prispevek — angl. “*information gain*” [106]): definiran je kot prispevana informacija atributa za določitev njegove vrednosti. *IG* je ena izmed najosnovnejših mer, ki meri pomembnost atributa [107]. Slabost mere nastopi, ko imamo attribute z zelo velikim številom različnih vrednosti, ker *IG* atributa raste s številom vrednosti. Metodo *IG* lahko uporabimo tudi v kontekstu priporočilnih sistemov, pri katerih lahko problem napovedovanja ocene prevedemo na klasifikacijski problem s petimi razredi (5 ocen od 1 do 5);
- *RR* (*RR*elief [108]): sodi v skupino algoritmov Relief, ki veljajo za najuspešnejše algoritme za izbor podmnožic pomembnih atributov. Algoritmi Relief poskušajo izmeriti kakovost vseh podanih atributov glede na to, kako dobro vrednosti, ki jih imajo atributi, ločijo med primeri, ki so si različni. Relief je razširitev Reliefa, ki za vsak učni primer poišče najbližji primer iz istega razreda in hkrati najbližji primer iz nasprotnega razreda. Temelji torej na ocenjevanju lokalnih značilnosti razločevanja razredov. Mera *RR*elief pa je prilagoditev algoritma Relief, ki je namenjen uporabi v klasifikacijskih problemih, medtem ko je *RR*elief namenjen regresijskim problemom.

Za uporabo enega izmed kriterijev *IG* ali *RR* razdelimo množico atributov na dve podmnožici: v prvi podmnožici so tisti atributi, pri katerih imamo podane ocene za uporabnika, ki je v stanju delnega hladnega zagona; drugo podmnožico pa tvorijo preostali atributi, ki nimajo ocen. Za lažje razumevanje predpostavimo, da elementom prve podmnožice rečemo vhodni atributi (neodvisne spremenljivke), elementom druge podmnožice pa ciljni atributi (odvisne spremenljivke).

Postopek ocenjevanja atributov izvedemo iterativno, tako da za vsak ciljni atribut izračunamo ocene vseh vhodnih atributov z ocenama *IG* in *RR*. Tako smo vsako oceno

za vsak vhodni atribut izračunali tolikokrat, kolikor je ciljnih atributov. V naslednjem koraku izračunamo oceno primernosti ciljnega atributa, in sicer tako, da množico ocen vseh vhodnih atributov (za dani ciljni atribut) v iteraciji povprečimo. To vrednost interpretiramo kot povprečno oceno kakovosti vseh vhodnih atributov, ki jih lahko uporabimo za napovedovanje vrednosti ciljnih atributov. Izmed vseh ciljnih atributov nato izberemo 10 % atributov z najvišjo oceno. S takšnim pristopom pričakujemo, da bomo izbrali attribute, ki so "univerzalno" najboljši oz. so najbolj primerne neodvisne spremenljivke za napovedovanje vseh odvisnih spremenljivk (ciljnih atributov).

Po koncu tega koraka prejmemo množico ustreznih objektov, ki jim lahko vstavimo manjkajoče ocene s pomočjo ene izmed metod, predstavljenih v naslednjem razdelku.

#### 5.2.4 Vstavljanje manjkajoče vrednosti

Tretji korak je namenjen vstavljanju manjkajočih vrednosti. Iz prejšnjega koraka (razdelek 5.2.3) smo prejeli množico atributov, ki jim lahko napovemo manjkajoče ocene.

Predlagamo naslednje možnosti za vstavljanje manjkajočih vrednosti atributom, od katerih sta prvi dve (ME in MF) primerni za delni in absolutni hladni zagon, drugi dve (LR in RT) pa samo za delni hladni zagon):

- *ME* (povprečna vrednost — angl. "mean value"): vstavimo povprečno vrednost vseh vrednosti (ocen) atributa,
- *MF* (najpogostejša vrednost — angl. "most frequent value"): vstavimo najpogostejšo vrednost izmed ocen, ki jih vsebuje atribut,
- *LR* (linearna regresija — angl. "linear regression") in *RT* (regresijsko drevo — angl. "regression tree"): dva modela nadzorovanega učenja, ki vsakemu izbranemu atributu napovesta manjkajočo vrednost z upoštevanjem atributov, ki že imajo ocene.

#### 5.2.5 Priporočanje z uporabo matričnih faktorizacij

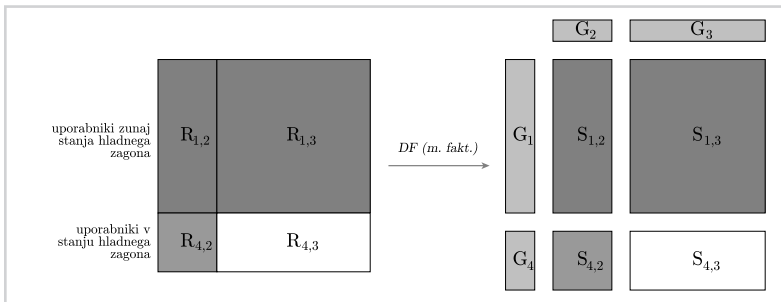
V našem raziskovanju smo uporabili štiri različne metode matričnih faktorizacij, ki se uporabljajo na področju priporočilnih sistemov. Najprej smo primerjali delovanje metod v stanju hladnega zagona — za nove uporabnike, nato pa smo primerjali njihovo delovanje skupaj z našim pristopom reševanja problema novih uporabnikov.

Za naš eksperiment smo uporabili vse štiri matrične faktorizacije, ki so predstavljenе v razdelku 4.1). Za potrebe reševanja hladnega zagona smo prilagodili tudi metodo

matrične faktorizacije z zlivanjem podatkovnih virov (DF) [91, 92], in sicer za reševanje hladnega zagona in uporabo na samo enim podatkovnem viru. Če imamo za dani podatkovni vir vhodno matriko  $R$ , smo vhodno matriko  $R$  razdelili na 4 bloke/podmatrice, in sicer na:  $R_{1,2}$ ,  $R_{1,3}$ ,  $R_{4,2}$  in  $R_{4,3}$ . Pred razdelitvijo smo vrstice zamenjali/razvrstili, tako da spodnje vrstice (matrik  $R_{4,2}$  in  $R_{4,3}$ ) predstavljajo uporabnike v stanju hladnega zagona. Nato smo stolpce prerazvrstili tako, da je matrika  $R_{4,2}$  čim bolj polna in matrika  $R_{4,3}$  čim bolj prazna. Slika 5.3 prikazuje predstavljeno razdelitev vhodne matrike na 4 bloke.

Za matriko  $R_{4,3}$  lahko predpostavimo, da je prazna. Njene vrednosti lahko dobimo kot matrično množenje faktorskih matrik, ki smo jih dobili z učenjem na blokih  $R_{1,2}$ ,  $R_{1,3}$  in  $R_{4,2}$ . S pomočjo množenja faktorskih matrik lahko torej dobimo  $R'_{4,3}$ , in sicer s formulo:

$$R'_{4,3} = G_4 \cdot S_{4,2} \cdot S_{1,2}^T \cdot S_{1,3} \cdot G_3^T \quad (5.1)$$



Slika 5.3

Levo: vhodna matrika, v kateri smo prerazporedili vrstice in stolpce tako, da smo dobili čim bolj prazno matriko  $R_{4,3}$ . Desno: rezultat matrične trifaktorizacije z zlivanjem podatkov.

### 5.3 Evalvacija pristopa reševanja problema hladnega zagona

V tem razdelku predstavljamo testiranje našega pristopa za reševanje hladnega zagona na različnih podatkovnih bazah. Cilj eksperimentalnega dela je oceniti naš pristop reševanja problema novih uporabnikov, ki se pojavi pri priporočilnih sistemih. Preizkusili smo vse mogoče kombinacije metod iz posameznih korakov našega predlaganega pristopa (opisanih v poglavju 5.2 in predstavljeni na sliki 5.2), kar skupaj tvori 1.280 kombinacij glede na naslednje parametre:

- 2 metodi za izbor ustreznega objekta/atributa, ki mu bomo vstavili manjkajočo

vrednost: SD (standardni odklon) in FR (frekvenca najpogostejše vrednosti) (korak 0.1),

- 2 metodi za vstavljanje manjkajoče vrednosti: ME (povprečna vrednost atributa) in MF (najpogostejša vrednost) (korak 0.2),
- 5 različnih razmerij za upoštevanje števila podobnih uporabnikov: 1 %, 5 %, 10 %, 25 % in 50 % (1. korak),
- 4 metode za izbor ustreznih objektov/atributov, ki jim bomo vstavili manjkajoče vrednosti: SD (standardni odklon), FR (frekvenca najpogostejše vrednosti), IG (informacijski prispevek) in RR (RReliefF) (2. korak),
- 4 metode za vstavljanje manjkajočih vrednosti: ME (povprečna vrednost atributa), MF (najpogostejša vrednost), LR (linearna regresija) in RT (regresijsko drevo) (3. korak),
- 4 metode matrične faktorizacije: NG (nenegativna matrična faktorizacija s stohastičnim gradientnim spustom), NS (nenegativna matrična faktorizacija z izmenjujočimi najmanjšimi kvadrati), SN (polnenegativna matrična faktorizacija z manjkajočimi vrednostmi) in DF (matrična faktorizacija z zlivanjem podatkovnih virov) (4. korak).

Vsako kombinacijo smo označili na naslednji način: XX-YY-ZZ-WW, pri čemer XX predstavlja odstotek upoštevanih podobnih uporabnikov (1 %, 5 %, 10 %, 25 % in 50 %), YY predstavlja izbrano metodo za iskanje atributov, ki jim bomo vstavili manjkajoče vrednosti (SD, FR, IG in RR), ZZ predstavlja metodo, s katero izračunamo oz. napovemo manjkajočo oceno (ME, MF, LR in RT) in WW predstavlja uporabljeno matrično kombinacijo (NG, NS, SN in DF).

### 5.3.1 Ocena uspešnosti

Naš pristop smo testirali na eni umetni in dveh realnih množicah, ki jih predstavljamo v razdelku 5.3.2. Uspešnost posameznih kombinacij v primerjavi z golimi metodami matričnih faktorizacij smo primerjali glede na koren relativne kvadratne napake

(RRSE):

$$RRSE(R) = \sqrt{\frac{\sum_{(i,j) \in \tau} (r_{i,j} - \hat{r}_{i,j})^2}{\sum_{(i,j) \in \tau} (r_{i,j} - \bar{r})^2}},$$

pri čemer  $\tau$  predstavlja testno množico ocen,  $r_{i,j}$  (iz množice podanih ocen  $R$ ) je dejanska ocena objekta  $j$  za uporabnika  $i$ ,  $\hat{r}_{i,j}$  pa predstavlja napovedano vrednost, ki jo izračuna matrična faktorizacija.

Ocena RRSE je izračunana tako, da smo razdelili celotno množico na 10 podmnožic (vsaka enkrat nastopa kot testna množica); povprečno napako RRSE smo izračunali z 10-kratnim prečnim preverjanjem.

S testiranjem želimo preveriti, ali smo izboljšali priporočanje uporabnikom v hladnem zagonu, ko smo uporabili naš pristop reševanja problema hladnega zagona v kombinaciji z izbranimi matričnimi faktorizacijami. Primerjali smo torej naš pristop in gole metode matričnih faktorizacij. Izraz “gola matrična faktorizacija” pomeni, da izvedemo metodo matrične faktorizacije brez vstavljanja manjkajočih vrednosti – vhodna matrika je nespremenjena.

Rezultate različnih kombinacij in golih matričnih faktorizacij smo statistično primerjali s Fridmanovim testom in z Nemenyijevem post-hoc testom [60]. Friedmanov test je neparametričen statistični test, ki rangira uspešnost posameznih metod (kombinacij) na vsaki domeni posebej in nato izračuna povprečen rang vsake metode prek vseh domen. Ničelna hipoteza predpostavlja dejstvo, da so povprečni rangi vseh metod enaki [60].

Nemenyijev post-hoc test oceni, ali sta uspešnosti dveh metod značilno različni, tako da preveri, če je njuna razlika povprečnih rangov večja od kritične razdalje (CD). Če želimo pogledati, ali sta si metodi  $i$  in  $j$  značilno različni, primerjamo njuna povprečna ranga  $PR_i$  in  $PR_j$ . Za razliko njunih povprečnih rangov mora torej veljati:

$$|PR_i - PR_j| \geq CD = q_\alpha \sqrt{\frac{k(k+1)}{6D}},$$

pri čemer  $q_\alpha$  predstavlja kritično vrednost Nemenyijevega testa za stopnjo zaupanja  $\alpha$ ,  $k$  predstavlja število metod, ki jih primerjamo,  $D$  pa predstavlja število podatkovnih množic, na katerih smo testirali naše metode [60].

### 5.3.2 Podatkovne množice

Vse možne kombinacije smo ločili glede uporabljenno metodo matrične faktorizacije v 4 skupine po 320 kombinacij. Te smo nato testirali na eni umetni podatkovni množici, ki smo jo poimenovali *ArtificialDS*. Umetno množico podatkov smo oblikovali tako, da vsebuje objekte, katerih ocene imajo med seboj pozitivno korelacijo, in objekte, katerih ocene imajo med seboj negativno korelacijo. Podatkovni množici smo dodali tudi nekaj šuma. Končna umetna podatkovna množica vsebuje 15 atributov s celoštevilskimi vrednostmi od 1 do 5. Prve tri attribute ( $A_1$ – $A_3$ ) smo generirali naključno (priredili smo jim naključne vrednosti), vrednosti preostalih 12 atributov ( $A_4$ – $A_{15}$ ) pa smo generirali z naslednjimi funkcijami:

- $A_4, A_7, A_{10}, A_{13}$  priredimo vrednost  $A_1 + \text{rand}(\{-1, 0, 1\})$ ,
- $A_5, A_8, A_{11}, A_{14}$  priredimo vrednost  $6 - A_2 + \text{rand}(\{-1, 0, 1\})$ ,
- $A_6, A_9, A_{12}, A_{15}$  priredimo vrednost  $\frac{A_{i-1} + A_{i-2}}{2} + \text{rand}(\{-1, 0, 1\})$ ,

pri čemer funkcija *rand* naključno izbere enega izmed elementov dane množice. Kakor smo že omenili, smo podatkovni množici dodali 10 % šumnih podatkov, pri čemer smo za vseh 15 atributov priredili naključne celoštevilске vrednosti med 1 in 5. Tabela 5.1 prikazuje lastnosti umetne podatkovne množice, ki smo jo generirali z zgornjimi pravili.

Tabela 5.1

Osnovne lastnosti umetne podatkovne množice *ArtificialDS*.

pod. množica	št. uporabnikov	št. objektov	št. ocen	redkost	ocene
<i>ArtificialDS</i>	200	15	3000	0 %	1–5

V naslednjem koraku smo predlagane metode testirali na dveh realnih domenah: podatkovna baza Jester [109], ki vsebuje ocene stotih smešnic, in naša podatkovna baza, v katerih so ocene, ki so jih študentje podali pri ocenjevanju različnih mulmedijskih učnih gradiv [3]. Tabela 5.2 predstavlja lastnosti obeh podatkovnih množic.



Tabela 5.2

Lastnosti realnih podatkovnih množic: Jester in PEFbase

pod. množica	št. uporabnikov	št. objektov	št. ocen	redkost	ocene
<i>Jester</i>	300	100	30000	0	1–5
<i>PEFbase</i>	286	35	10010	0	1–4

## 5.4 Evalvacija reševanja absolutnega hladnega zagona

### 5.4.1 Analiza na umetni domeni

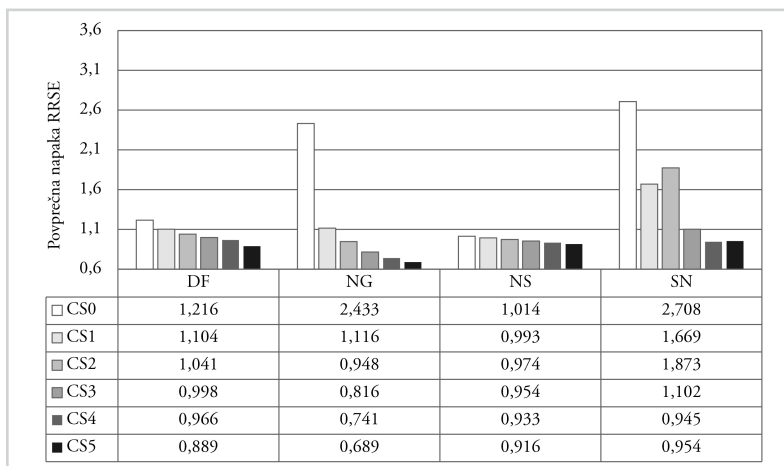
V razdelku 5.2 smo predstavili predkorak za reševanje absolutnega hladnega zagona, če nimamo danih ocen za uporabnike v omenjenem stanju.

Naprej smo želeli preveriti temeljno predpostavko, tj. ali se z večjim številom podanih uporabnikov ocen izboljša tudi kakovost napovedovanja oz. priporočanja. Zato smo zagnali gole matrične faktorizacije (DF, NG, NS in SN) na različno obsežnih situacijah hladnega zagona glede na število začetnih podanih ocen. Te situacije smo označili z oznakami CS0, CS1, CS2, CS3, CS4 in CS5, in sicer glede na število začetnih podanih ocen (CS0 torej predstavlja absolutni hladni zagon). Rezultati (slika 5.4) potrjujejo našo predpostavko — napaka RRSE se v skoraj vseh primerih zmanjša za vsako dodatno uporabnikovo oceno. Rezultati kažejo tudi dejstvo, da je pri večini golih matričnih faktorizacij očitna razlika med stanjema CS0 in CS1, kar pomeni, da nam lahko že ena sama znana ocena bistveno izboljša priporočanje objektov. Po drugi strani pa opazimo, da so v večini primerov prva tri stanja hladnega zagona za priporočanje neuporabna, ker je napaka RRSE (bistveno) večja od vrednosti 1, kar pomeni, da priporočanje deluje slabše, kot če bi priporočali povprečne ocene. Hkrati lahko opazimo tudi, da je matrična faktorizacija NS na naši umetni domeni najprimernejša za napovedovanje uporabnikom v stanju hladnega zagona (CS1–CS5), medtem ko matrična faktorizacija NG po stanju CS2 prevzame vlogo najboljše gole metode.

Kakor smo že omenili v razdelku 5.2, želimo stanje CS0 spremeniti v stanje, ki je podobno stanju CS1 delnega hladnega zagona, saj sicer ne moremo poiskati podobnih uporabnikov, kakor predvideva naš pristop. Ker gre za stanje, pri katerem nimamo ocen za uporabnike v absolutnem hladnem zagonu, lahko uporabimo le določene metode za izbor atributov (razdelek 5.2.3) in vstavljanje manjkajoče ocene (razdelek 5.2.4)

Slika 5.4

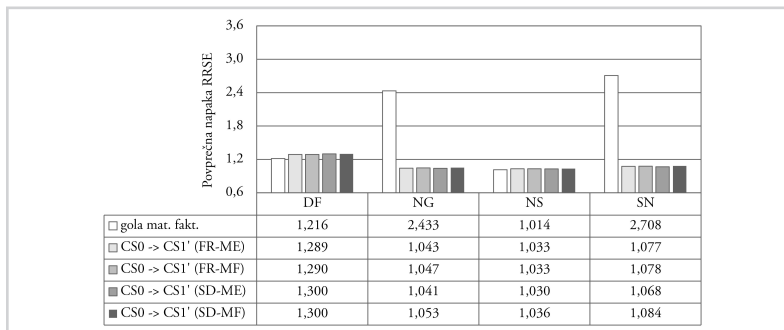
Uspešnost priporočanja golih matričnih faktorizacij različno obsežnih hladnih zagonov (označene s CS0–CS5), glede na napako RRSE. Rezultati kažejo, da je napaka RRSE pri treh matričnih faktorizacijah sicer nižja, a so hkrati priporočanja glede na rezultate še vedno neuporabna.



— preostalih metod ni mogoče uporabiti, saj modeli predpostavljajo, da imamo vsaj eno vrednost v atributu, če želimo napovedati vrednost drugemu atributu. Za izbor najustreznejšega atributa lahko torej uporabimo metodi standardnega odklona (SD) in frekvenco najpogostejše vrednosti (FR). Za vstavljanje manjkajočih vrednosti pa lahko uporabimo vstavljanje povprečne vrednosti atributa (ME) in vstavljanje najpogostejše vrednosti v danem atributu (FR). S pomočjo teh 4 kombinacij torej uporabnikom v stanju CS0 vstavimo samo eno oceno in tako prevedemo stanje iz CS0 v CS1'.

Slika 5.5

Primerjava uspešnosti golih metod matrične faktorizacije na uporabnikih v absolutnem hladnem zagonu in uporabnikih, ki smo jim vstavili eno manjkajočo oceno – spremenili smo stanje iz absolutnega v delni hladni zagon. Prikazana je uspešnost gole matrične faktorizacije in štirje pristopi spreminjanja iz scenarija CS0 v scenarij CS1'



Slika 5.5 prikazuje delovanje štirih metod matričnih faktorizacij na umetni domeni, ko smo uporabnikom v testni množici izbrisali vse ocene in s tem simulirali absolutni hladni zagon. Na sliki je razvidno, da je napaka RRSE pri treh matričnih faktorizacijah sicer nižja, a so hkrati priporočanja glede na rezultate še vedno neuporabna, ker je napaka RRSE večja od vrednosti 1.

#### 5.4.2 Analiza na realnih domenah

Eksperiment smo izvedli na povsem enak način kot v razdelku 5.4, v katerem smo testirali vse kombinacije na umetni podatkovni množici. Na sliki 5.6 opazimo, da so rezultati najboljši, če uporabimo golo matrično faktorizacijo NS za primere, ki se nahajajo v absolutnem hladnem zagonu (CS0). Slika prav tako kaže, da so pri matričnih faktorizacijah DF in SN priporočanja neuporabna za vseh pet scenarijev. Pri matrični faktorizaciji NG pa je na sliki razvidno, da se rezultat pri scenarijih CS3, CS4 in CS5 nekoliko popravi.

V naslednjem koraku smo testirali naš pristop reševanja absolutnega hladnega zagona še na realnih domenah. Glede na rezultate eksperimenta (slika 5.7) je razvidno, da se priporočanje izboljša pri tistih primerih, pri katerih smo uporabili ali matrično faktorizacijo NG ali SN v kombinaciji z našimi predlaganimi metodami. Pri metodi NS smo rezultate nekoliko poslabšali, pri metodi DF pa so rezultati nepomembni, saj povprečna napaka RRSE ostaja večja od 1.

V naslednjem razdelku predstavljamo evalvacijo našega pristopa za reševanje delnega hladnega zagona.

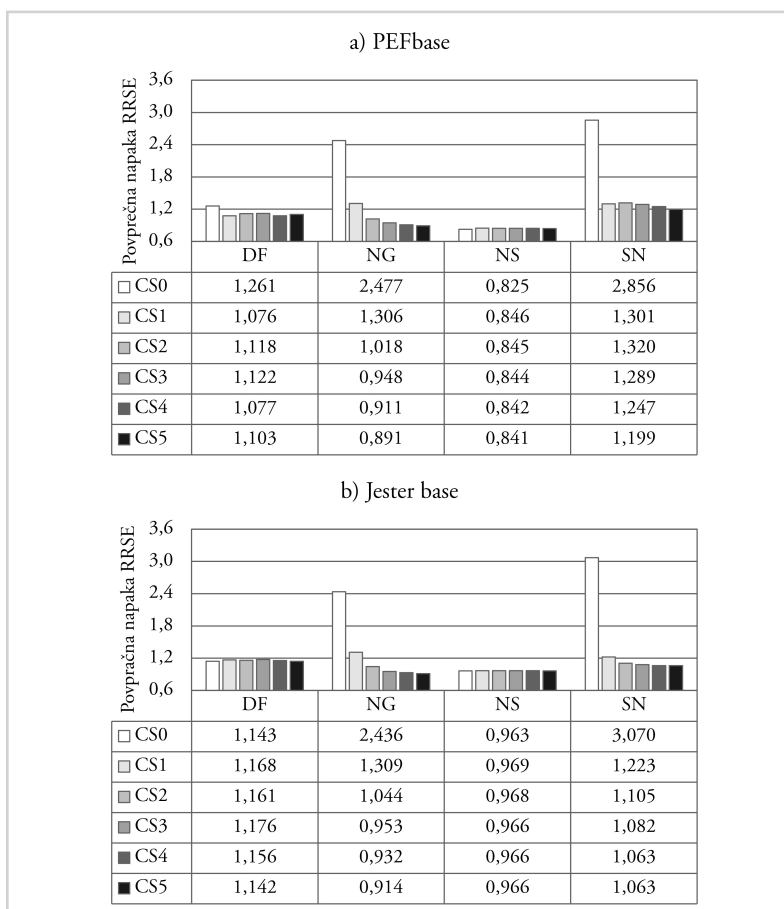
### 5.5 Evalvacija reševanja delnega hladnega zagona

V tem razdelku smo se osredinili na reševanje problema delnega hladnega zagona. Predstavljamo analizo našega pristopa po posameznih korakih metode.

#### 5.5.1 Analiza variant metod v posameznih korakih metode

V nadaljevanju se osredinjamo na vprašanje, kako izbor parametrov za posamezen korak reševanja hladnega zagona (prikazan na sliki 3.1) vpliva na uspešno priporočanje objektov z matrično faktorizacijo.

Vse možne kombinacije smo ločili glede uporabljeno metodo matrične faktorizacije v 4 skupine po 320 kombinacij (5 deležev najbližjih sosedov  $\times$  4 metode za izbiro atributov  $\times$  4 metode za vstavljanje vrednosti). Za vseh 5 stopenj hladnega zagona

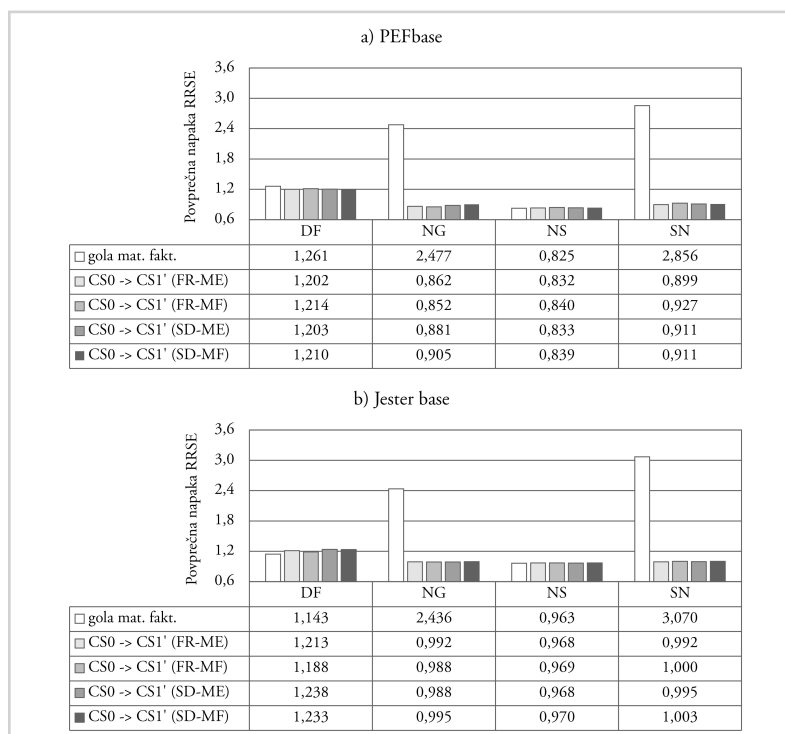


Slika 5.6

Uspešnost priporočanja golih matričnih faktorizacij za različno obsežne hladne zagone (označene s CS0–CS5) glede na napako RRSE na podatkovnih množicah PEFbase in Jester.

(CS1–CS5) smo izračunali povprečno vrednost napake RRSE. Uporabili smo Friedmanov test, da smo preverili, ali so razlike v uspešnosti priporočanja statistično značilne.

Najprej smo na umetni množici ocenili, kako različni odstotki najpodobnejših uporabnikov (prvi del našega pristopa) vpliva na delovanje algoritmov matrične faktorizacije (slika 5.8) v povprečju. S slike vidimo, da uporaba poljubne vrednosti parametra v



Slika 5.7

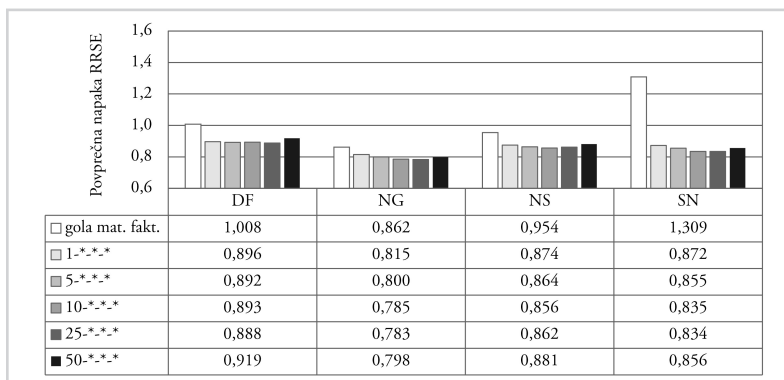
Primerjava uspešnosti golih matričnih faktorizacij in matričnih faktorizacij s pristopom odpravljanja absolutnega hladnega zagona. Metode smo primerjali na dveh realnih podatkovnih množicah.

prvem koraku metode zniža napako RRSE. Slika 5.9 prikazuje statistično analizo rezultatov s slike 5.8, iz katere se tudi vidi, da vrednost parametra 25 % statistično značilno izboljša priporočanje uporabnikom v hladnem zagonu.

V naslednjem koraku smo na podoben način na umetni množici ocenili povprečne uspešnosti različnih atributov za izbor atributov, ki jim bomo vstavili manjkajočo oceno (drugi korak našega pristopa). Primerjava povprečnih napak RRSE prikazuje slika 5.10, na kateri vidimo, da uporaba poljubne vrednosti parametra v drugem koraku metode zniža napako RRSE v primerjavi z golo matrično faktorizacijo. Slika 5.11 prikazuje statistično primerjavo rezultatov s slike 5.10 in to, da parametra \*-FR-\* in \*-RR-\* značilno izboljšata napovedno točnost v primerjavi z golo matrično faktorizacijo.

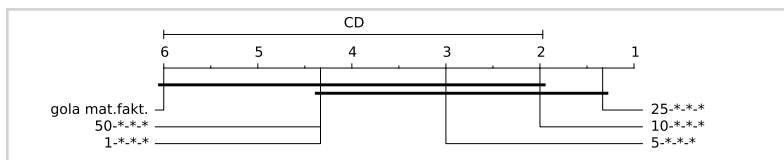
Slika 5.8

Evalvacija delovanja golih matričnih faktorizacij (brez vstavljanja manjkajočih vrednosti) in matrične faktorizacije z različnimi izbirami metod v prvem koraku (upoštevajo odstotek najbližjih sosedov) na umetni množici. Povprečna napaka RRSE je povprečna preko vseh metod, ki upoštevajo enak delež podobnih uporabnikov; ker so ostali parametri poljubni, so označeni z \*.



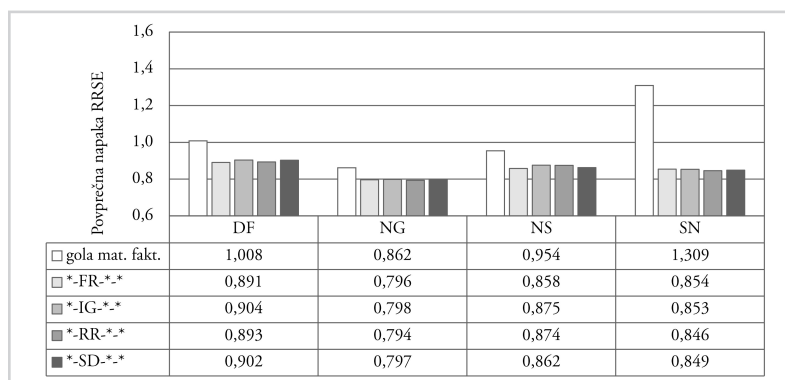
Slika 5.9

Statistična primerjava uspešnosti gole matrične faktorizacije in matrične faktorizacije z različnimi izbirami metod v prvem koraku pristopa. Različne vrednosti prvega parametra (XX) so povprečene po vseh vrednostih parametrov (označenih z \*) – npr. 5-\*.\*\* označuje, da za vstavljanje manjkajočih vrednosti upoštevamo 5 % podobnih uporabnikov. Skupine metod, ki se statistično značilno ne razlikujejo (pri  $p \leq 0,05$ ), so povezane z vodoravno črto.



Poleg analize rezultatov povprečnih vrednosti napak RRSE prvega in drugega parametra smo na umetni množici ocenili tudi, kako izbor metode za vstavljanje manjkajoče vrednosti (tretji korak našega pristopa) vpliva na povprečno delovanje algoritmov matrične faktorizacije (slika 5.12). S slike vidimo, da uporaba poljubne vrednosti parametra v tretjem koraku metode zniža napako RRSE v primerjavi z golo matrično faktorizacijo. Slika 5.13 prikazuje statistično analizo omenjenih rezultatov, iz katere se vidi, da se varianti metod \*-\*.RF-\* in \*-\*.ME-\* značilno razlikujeta od golih metod matričnih faktorizacij za priporočanje uporabnikom v hladnem zagonu.

Glede na pridobljene rezultate smo izbrali kombinaciji metod, za kateri pričakuje-



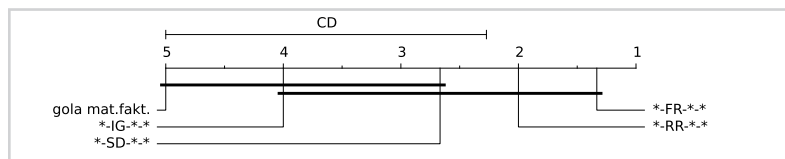
Slika 5.10

Evalvacija delovanja golih matričnih faktorizacij (brez vstavljanja manjkajočih vrednosti) in matrične faktorizacije z različnimi izbirami metode v drugem koraku (način izbire atributov) na umetni množici. Napaka RRSE je povprečna prek vseh metod, ki izbirajo ustrezne atribute z enako metodo; ker so drugi parametri poljubni, so označeni z \*.

Slika 5.11

Statistična primerjava uspešnosti gole matrične faktorizacije in matrične faktorizacije z različnimi izbirami metod v drugem koraku pristopa. Različne vrednosti drugega parametra (YY) so povprečene po vseh vrednostih parametrov (označenih z \*) – npr. \*-SD-\*

označuje, da za vstavljanje manjkajočih vrednosti izberemo atribute z najmanjšim standardnim odklonom. Skupine metod, ki se statistično značilno ne razlikujejo (pri  $p \leq 0,05$ ), so povezane z vodoravno črto.



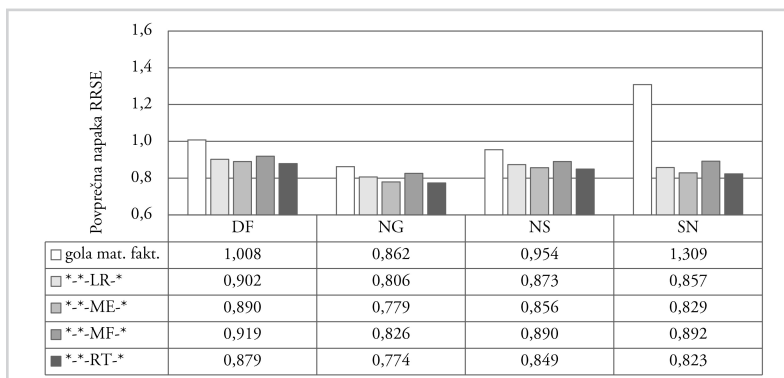
mo, da bosta dobro delovali. Izbrani kombinaciji sta: 25-FR-ME-\* in 25-RR-RT-\*. Uspešnost obeh kombinacij smo testirali na dveh realnih domenah – rezultati so predstavljeni v razdelku 5.5.3.

### 5.5.2 Rangiranje uspešnosti posameznih eksperimentov

V naslednjem koraku smo primerjali povprečne range vseh 320 kombinacij za vstavljanje manjkajočih vrednosti (glede na posamezne parametre s slike 3.1) – vse kombinacije smo testirali na umetni domeni. V analizo smo vključili tudi gole metode matrične faktorizacije (NG, NS, SN in DF). Slika 5.14 prikazuje povprečne range po-

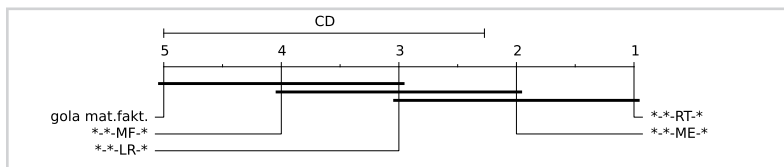
Slika 5.12

Evalvacija delovanja golih matričnih faktorizacij (brez vstavljanja manjkajočih vrednosti) in izbirami metode v tretjem koraku (metoda za vstavljanje manjkajoče vrednosti) na umetni množici. Napaka RRSE je povprečna preko vseh metod, ki v izbrane attribute vstavljajo manjkajoče ocene z enako metodo; ker so ostali parametri poljubni, so označeni z \*.



Slika 5.13

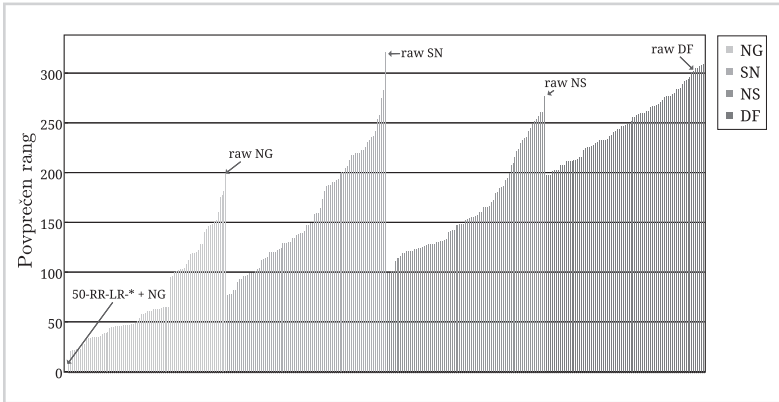
Statistična primerjava uspešnosti gole matrične faktorizacije in matrične faktorizacije z različnimi izbirami metod v tretjem koraku pristopa. Različne vrednosti tretjega parametra (ZZ) so povprečne po vseh vrednostih parametrov (označenih z \*) – npr. \*-ME-\* označuje, da za vstavljanje manjkajočih vrednosti uporabimo izračun povprečne vrednosti ocen v izbranem atributu. Skupine metod, ki se statistično značilno ne razlikujejo (pri  $p \leq 0,05$ ), so povezane z vodoravno črto.



sameznih kombinacij v primerjavi z golimi metodami. S slike je razvidno, da imajo vse metode za reševanje problema hladnega zagona z izjemo nekaterih metod v kombinaciji z metodo DF bistveno nižji rang, kar pomeni, da so se nekatere metode v povprečju odrezale bistveno boljše kot gole matrične faktorizacije.

Med vsemi pristopi, prikazanimi na sliki 5.14, smo izbrali kombinacijo, ki ima v povprečju najboljši povprečni rang – izbrali smo metodo 50-RR-LR-\*. Čeprav sklepamo, da se je metoda preveč prilagodila vhodnim podatkom, smo jo skupaj z že izbranimi metodama iz poglavja 5.5.1 (25-FR-ME-\* in 25-RR-RT-\*) testirali na dveh realnih množicah. Rezultate predstavljamo v razdelku, ki sledi.





Slika 5.14

Primerjava vseh 320 kombinacij za vstavljanje manjkajočih vrednosti in golih metod matrične faktorizacije. Rezultati so na umetni domeni grupirani glede na posamezno faktorizacijsko metodo (NG, SN, NS, DF).

### 5.5.3 Evalvacija najboljših izbranih metod na realnih domenah

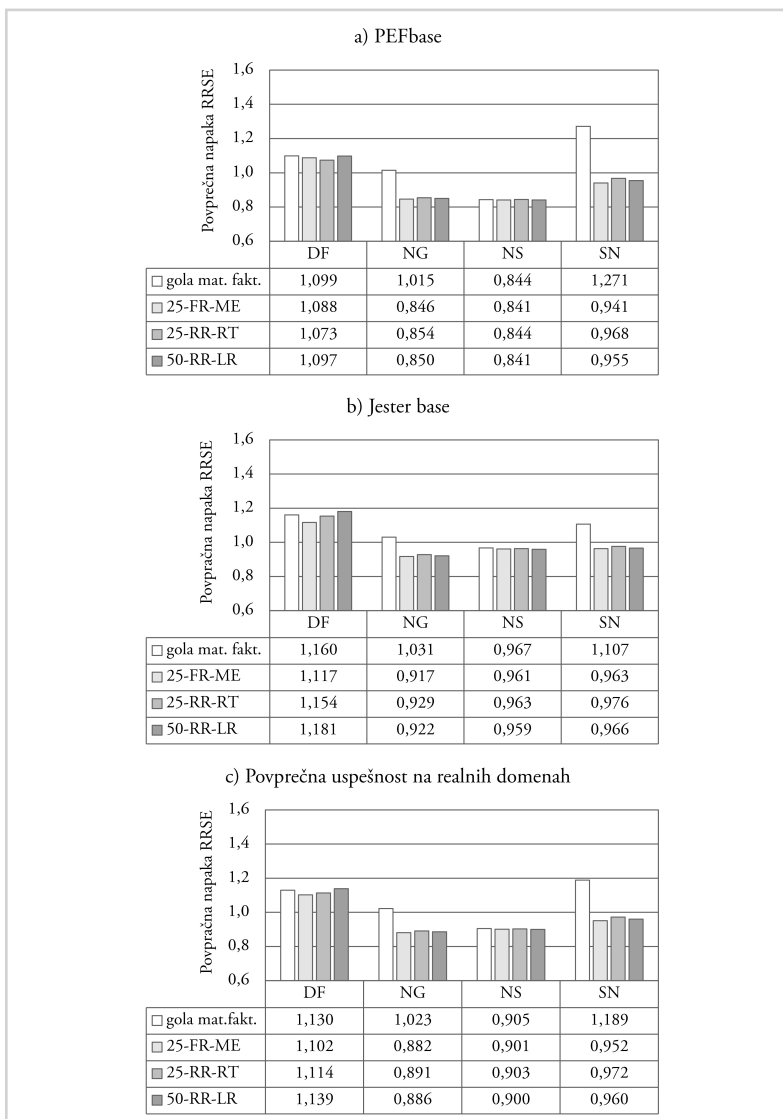
V naslednjem koraku smo izbrane kombinacije metod iz prejšnjega razdelka testirali na dveh realnih domenah: podatkovni bazi Jester [109] in naši podatkovni bazi o ocenjevanju različnih mulmedijskih učnih gradiv [3].

Eksperiment smo izvedli na enak način kot druge dozdašnje. Slika 5.15 prikazuje (za posamezno podatkovno množico) povprečno napako RRSE golih matričnih faktorizacij v primerjavi z metodami matričnih faktorizacij, ki uporabljajo naš pristop reševanja hladnega zagona, in sicer z uporabo posamezne izbrane metode za vstavljanje manjkajočih vrednosti (50-RR-LR-\*, 25-FR-ME-\* in 25-RR-RT-\*).

Rezultati (slika 5.15) kažejo, da smo z našim pristopom izboljšali priporočanje v kombinaciji z uporabo matričnih faktorizacij NG, NS, SN na podatkovni množici *PEFbase*. Pri matrični faktorizaciji NS so razlike zelo majhne. Pri matrični faktorizaciji DF pa so rezultati nepomembni, saj je povprečna napaka RRSE večja od 1.

Glede na rezultate na podatkovni množici *Jester base* (slika 5.15) vidimo, da se je povprečna napaka RRSE zmanjšala pri uporabi našega pristopa v kombinaciji z matričnimi faktorizacijami NG, NS in SN, pri čemer napaka RRSE pri metodi DF ostaja večja od 1.

Graf c) (slika 5.15) prikazuje povprečne napake RRSE na obeh realnih podatkovnih množicah. Na grafu je razvidno, da ima kombinacija 25-FR-ME-NG najmanjšo povprečno napako RRSE, kar bi lahko pomenilo, da je kombinacija najprimernejša za

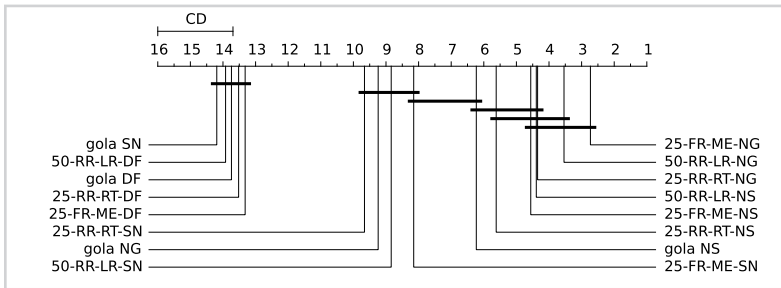


Slika 5.15

Primerjava uspešnosti golih matričnih faktorizacij in matričnih faktorizacij z izbrano metodo za vstavljanje manjkajočih vrednosti. Metode smo primerjali na dveh realnih podatkovnih množicah (a) *PEFbase* in b) *Jester base*. V tabeli c) pa se nahajajo povprečne napake RRSE, povprečne po obeh realnih podatkovnih množicah.

priporočanje uporabnikom, ki se nahajajo v stanju hladnega zagona. Vendar pa smo pri analizi rezultatov (graf a) na sliki 5.15) omenili, da so razlike med uspešnostjo gole matrične faktorizacije NS in uspešnostjo matrične faktorizacije NS v kombinaciji s predlaganimi metodami zelo majhne. Zato smo v naslednjem koraku pogledali, ali obstajajo statistično pomembne razlike med uspešnostjo priporočanj vseh 16 metod.

Friedmanov test ( $F= 1138.03, p= 0.00 < 0.05$ ) je potrdil, da so med primeri statistično pomembne razlike. S post-hoc Nemenyi testom smo pogledali, ali je katera izmed kombinacij značilno boljša od gole matrične faktorizacije NS. Statistična analiza (slika 5.16) je pokazala, da se povprečni rang kombinacije 25-FR-ME-NG statistično značilno razlikuje od povprečnih rangov vseh golih matričnih faktorizacij (razlika povprečnih rangov je večja od kritične razdalje CD). To potrjuje, da vstavljanje manjkajočih vrednosti z metodo 25-FR-ME-\* in uporabo matrične faktorizacije NG statistično značilno izboljša priporočanje objektov v primerjavi z golo metodo NS, in sicer za uporabnike v stanju delnega hladnega zagona.



Slika 5.16

Statistična analiza uspešnosti golih matričnih faktorizacij in metod za vstavljanje manjkajočih vrednosti v kombinaciji z izbrano matrično faktorizacijo. Kombinacije metod smo primerjali na dveh realnih podatkovnih množicah. Rezultati potrjujejo, da vstavljanje manjkajočih vrednosti z metodo 25-FR-ME-\* in uporabo matrične faktorizacije NG statistično značilno izboljša priporočanje objektov v primerjavi z golo metodo NS.

Pri izbiri primerne metode matrične faktorizacije in zelenega pristopa za vstavljanje manjkajočih vrednosti je potrebno upoštevati tudi časovne omejitve, ki so odvisne od priporočilnega sistema, ki ga želimo zgraditi. Pogledali smo čas izvajanja naših predlaganih metod za vstavljanje manjkajočih vrednosti in čas izvajanja posameznih matričnih faktorizacij. Vsak algoritem smo 50-krat ponovili in izmerili čas izvajanja. Matrično faktorizacijo smo izvedli za vse uporabnike hkrati (tudi za vse CS-uporabnike). Časi izvajanja so bili izmerjeni na napravi s specifikacijami: CPU: Intel Core i7-4700HQ, RAM: 16 GB.

Iz izmerjenih časov izvajanja (tabela 5.3) opazimo, da je metoda SN najhitrejša, kar pomeni, da je primerna tudi za večje podatkovne baze. To lastnost smo tudi že izpostavili v razdelku 4.1.3, v katerem smo teoretično opisali metodo SN.

Tabela 5.3

Primerjava časov izvajanja matričnih faktorizacij na prvotnih podatkih (vhodnih matrikah) in na podatkih, v katere smo z izbranimi metodami vstavili manjkajoče vrednosti za uporabnike (28 uporabnikov), ki se nahajajo v stanju hladnega zagona.

<i>pristopi/metode</i>	<i>matrična faktorizacija</i>			<i>alg. vstavljanja + mat. fakt.</i>		
	<i>NG</i>	<i>NS</i>	<i>SN</i>	<i>NG</i>	<i>NS</i>	<i>SN</i>
gola mat. fakt.	11,829	4,057	0,159	11,829	4,057	0,159
25-FR-ME-*	12,018	4,010	0,167	15,102	7,094	3,251
25-RR-RT-*	12,024	4,139	0,167	19,153	11,268	7,296
50-RR-LR-*	12,019	4,065	0,168	19,384	11,430	7,532

Iz časov izvajanja ob uporabi metod za dopolnjevanje vrednosti vidimo, da je metoda 25-FR-ME-\* hitrejša kakor metodi 25-RR-RT-\* in 50-RR-LR-\*. Ključni vzrok za daljši čas izvajanja se nahaja v drugem koraku našega pristopa (opisan v razdelku 5.2). Metoda \*-RR-\* za iskanje ciljnih atributov (primernih za vstavljanje manjkajoče vrednosti) iterativno za vsak ciljni atribut izračuna ocene vseh vhodnih atributov z oceno RR. Tako smo vsako oceno za vsak vhodni atribut izračunali tolikokrat, kolikor je ciljnih atributov. To pomeni, da se s številom ciljnih atributov čas izvajanja algoritma RRelief še poveča.

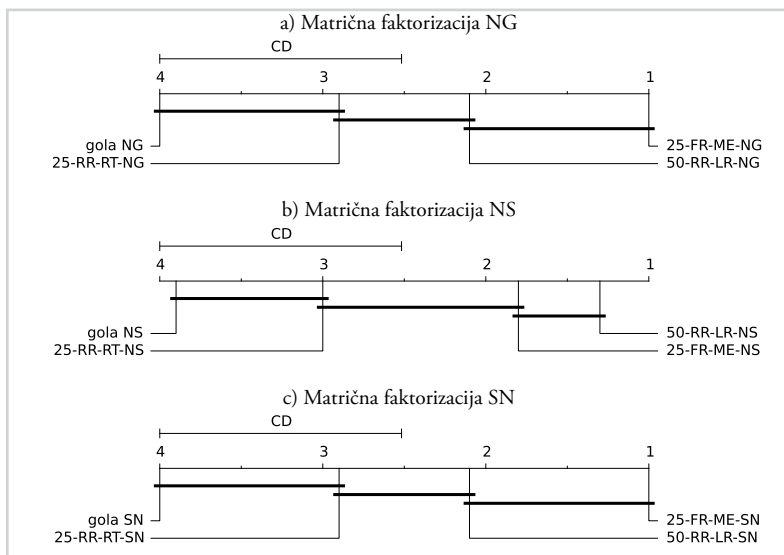
Če pogledamo priporočanje z matrično faktorizacijo NS, vidimo, da se v povprečju algoritem izvaja približno 4 sekunde. Metoda NS je prav tako kot metoda SN primerna za uporabo na velikih matrikah, ki imajo veliko število uporabnikov in objektov. Kakor smo že omenili v razdelku 4.1.2, metoda v prvotni različici ni namenjena redkim matrikam, saj metoda predpostavlja, da imamo za vsakega uporabnika in vsak objekt oceno. Večji kot je interval zaloge vrednosti (število možnih vrednosti, ki jih ima atribut), večja je možnost, da povprečenje ocen po atributih pokvari priporočanje – razen v primeru našega pristopa, kjer selektivno vstavimo manjkajoče vrednosti. Zato metoda NS ni primerna v situacijah, kjer ima priporočilni sistem večje število možnih celoštevilskih vrednosti ocen za posamezni objekt.

Matrična faktorizacija NG ima v povprečju najdaljši čas izvajanja glede na eksperimente. Rezultat ni presenetljiv, saj smo v razdelku 4.1.1 omenili, da je algoritem

časovno zelo potraten, saj temelji na iskanju globalnega minimuma celotne matrike, kar pomeni, da v vsaki iteraciji algoritem za vsako podano obstoječo oceno izračuna njeno napoved in napako. Večja kot je matrika, daljši čas izvajanja ima metoda NG. Raziskovalci poročajo o trajanju izvajanja tudi po več ur in celo več dni. Prednost matrične faktorizacije NG je, da deluje tudi na redkih vhodnih matrikah.

Na osnovi zgornjih ugotovitev lahko zaključimo, da ima lahko uporabnik utemeljene razloge za uporabo prav določene metode za matrično faktorizacijo; ti lahko temeljijo na času izvajanja ali ustreznosti algoritma. Zato smo v nadaljevanju s statistično analizo preverili, ali obstajajo statistično značilne razlike med priporočanji posameznih matričnih faktorizacij v kombinaciji z našim pristopom, kar je prikazano na sliki 5.17. Če primerjamo statistično analizo 16 metod hkrati (slika 5.16) in statistično analizo priporočanja z matrično faktorizacijo NS (graf b), slika 5.17) opazimo, da se rangi metod spremenijo, ker smo iz primerjave odstranili nekatere metode. Friedmanov test je pokazal, da so med primeri značilne razlike, in sicer pri uporabi našega pristopa v kombinaciji z matričnimi faktorizacijami: NG ( $F=28.92, p=0.00 < 0.05$ ), NS ( $F=24.82, p=0.00 < 0.05$ ) in SN ( $F=28.92, p=0.00 < 0.05$ ). Post-hoc Nemenyi test je pokazal, da se priporočanje metod 25-FR-ME-\* in 50-RR-ME-\* pri vseh treh matričnih faktorizacijah NG, NS in SN značilno razlikuje od priporočanja z golimi matričnimi faktorizacijami za uporabnike v delnem hladnem zagonu, medtem metoda 25-RR-RT-\* ne dosega značilnih razlik od gole matrične faktorizacije.

V tem razdelku smo pogledali, kako uspešne so predlagane metode za priporočanje na dveh realnih podatkovnih množicah. Statistična analiza je pokazala, da je kombinacija 25-FR-ME-NG statistično značilno različna/boljša kakor gole matrične faktorizacije (NG, NS, SN in DF). Metode smo primerjali tudi glede na čas izvajanja. Testiranje je pokazalo, da ima metoda NG bistveno daljši čas izvajanja kakor metodi NS in SN. To pomeni, da je metoda NG primerna za manjše podatkovne množice. V nadaljevanju pa smo izvedli statistično analizo na posameznih matričnih faktorizacijah. Analiza je pokazala, da smo z našim pristopom (uporabo metod 25-FR-ME-\* in 50-RR-ME-\*) v kombinaciji z matričnimi faktorizacijami NG, NS in SN značilno izboljšali priporočanje golih matričnih faktorizacij za uporabnike v delnem hladnem zagonu.



Slika 5.17

Statistična analiza povprečnih uspešnosti golih matričnih faktorizacij in metod za vstavljanje manjkajočih vrednosti v kombinaciji z izbrano matrično faktorizacijo – primeri so ločeni glede na matrično faktorizacijo. Rezultati potrjujejo, da smo z našim pristopom (metodama 25-FR-ME-\* in 50-RR-ME-\*) statistično značilno izboljšali priporočanje objektov za matrične faktorizacije NG, NS in SN.

### 5.6 Vpliv vstavljanja manjkajočih ocen na priporočanje uporabnikom zunaj stanja hladnega zagona

Čeprav smo v prejšnjih razdelkih pokazali, da smo izboljšali priporočanje uporabnikom v stanju delnega hladnega zagona, nas zanima, kako vstavljanje vpliva na priporočanje uporabnikom zunaj stanja delnega hladnega zagona. Pričakujemo, da naš pristop ne poslabša bistveno samega priporočanja drugim uporabnikom – da torej nima globalnega (negativnega) vpliva na samo priporočanje. Uporabili smo algoritme gole matrične faktorizacije in algoritme v kombinaciji z metodo 25-FR-ME-\*, ki se je izkazala kot najuspešnejša na realnih domenah.

Eksperiment smo oblikovali tako, da smo vsaki domeni (tabeli) naključno izbrisali 20 % vseh vrednosti. Nato smo izvedli matrično faktorizacijo z uporabo našega pristopa ali brez nje in pogledali uspešnost priporočanja samo uporabnikom, ki so zunaj stanja hladnega zagona.

Tabela 5.4 vsebuje (za posamezno podatkovno množico) povprečne uspešnosti (napake RRSE) golih metod matričnih faktorizacij v primerjavi z metodami matričnih faktorizacij, ki uporabljajo 25-FR-ME-\*. Če pogledamo povprečne vrednosti napak

Tabela 5.4

Primerjava metod matrične faktorizacije (gole mat. fakt. — brez vstavljanja manjkajočih vrednosti) in metode matrične faktorizacije v kombinaciji s pristopom vstavljanja manjkajočih vrednosti. Odebeljene so tiste vrednosti, ki imajo najmanjšo povprečno napako RRSE za dano matrično faktorizacijo.

pristopi/metode	PEFbase				Jester base			
	DF	NG	NS	SN	DF	NG	NS	SN
<i>gola mat. fakt.</i>	0,978	<b>0,818</b>	0,939	1,113	0,930	<b>0,820</b>	<b>0,821</b>	1,216
<i>25-FR-ME-*</i>	<b>0,973</b>	<b>0,818</b>	<b>0,938</b>	<b>0,897</b>	<b>0,925</b>	<b>0,820</b>	<b>0,821</b>	<b>0,914</b>

RRSE obeh pristopov, je razvidno, da ima naša metoda v vseh primerih manjšo ali enako povprečno napako RRSE kot gole matrične faktorizacije. Po drugi strani pa je Friedmanov test pokazal, da med pristopoma (golimi metodami in naši kombinaciji metod) ni statistično pomembnih razlik ( $F = 0,143$ ,  $p = 0,705 > 0,05$ ), kar pomeni, da naša metoda ni značilno izboljšala samega delovanja matrične faktorizacije za uporabnike zunaj stanja hladnega zagona. To dejstvo namreč potrjuje naše predvidevanje, da naš pristop ne poslabša samega priporočanja drugim uporabnikom, kar je spodbudno za nadaljnje raziskovanje in izboljšanje našega pristopa.





# *Zaključek*

6

*Preteklost je lahko boleča – imaš le dve možnosti:  
ali zbežiš pred njo ali pa se iz nje nekaj naučiš.  
(iz risanega filma "The Lion King", 1994)*

V prvem delu disertacije smo predstavili pristop za izbiranje primernih predstavitev učne vsebine. Z zgrajenim drevesom smo prikazali, da je smiselno kombinirati različne modele učnih stilov za ustrežnejšo izbiro in priporočanje različnih tipov multimedijskih gradiv. Z našim pristopom smo povezali pedagoška in psihološka znanja v skupen model, ki je ustrezen za statično priporočanje multimedijskih gradiv. Zgrajeni model je pokazal, da je hemisferični model učnih stilov najpomembnejši odločitveni atribut; Kolbov model in model VAK sta druga najpomembnejša odločitvena atributa. Raziskava je prav tako pokazala, da študentje za učenje najpogosteje uporabljajo besedila, ki so dobro strukturirana in imajo barvno diskriminacijo. Poleg besedil pogosto uporabljajo tudi animacije in videoposnetke.

V drugem delu disertacije je predstavljen inovativen pristop, ki omogoča uspešnejše priporočanje objektov s pomočjo metod matričnih faktorizacij za uporabnike v stanju hladnega zagona. Razvili smo pristop, ki temelji na reševanju hladnega zagona vsakega novega uporabnika posamezno – novemu uporabniku vstavimo manjkajoče ocene (glede na naš pristop) in nato izvedemo določeno matrično faktorizacijo nad celotno množico. Znotraj raziskave smo ločili dve stanji hladnega zagona, in sicer na absolutni hladni zagon (vektor novega uporabnika  $u$  v matriki  $R$  je povsem prazen) in delni hladni zagon (vektor novega uporabnika  $u$  ima premalo ocen; vektor je reddek). Omenjeni postopek reševanja hladnega zagona sestavljajo naslednji koraki: predkorak 0) če je uporabnik v stanju absolutnega hladnega zagona, mu vstavimo eno oceno in ga privedemo do stanja delnega hladnega zagona; 1) uporabniku v delnem hladnem zagonu poiščemo  $N$  podobnih uporabnikov; 2) znotraj te množice določimo attribute/objekte, ki jim bomo vstavili manjkajočo vrednost; 3) izbranim atributom glede na izbrano metodo izračunamo oz. vstavimo manjkajočo oceno; 4) uporabimo izbrano matrično faktorizacijo nad vsemi uporabniki.

Statistična analiza eksperimentalnega dela (testiranja našega pristopa) na umetni množici je pokazala, da vsak posamezen parameter (posameznega koraka pri predlaganem pristopu) statistično značilno izboljša priporočanje metod matričnih faktoriza-

cij. Najboljše delovanje so pokazale metode, pri katerih upoštevamo 25 % najbližjih sosedov (25-\*-\*-\*), uporabljamo frekvenco (\*-FR-\*\*) ali ReliefF (\*-RR-\*\*) za izbiro atributov in vstavljamo vrednosti s povprečenjem (\*-\*ME-\*) ali z uporabo regresijskega drevesa (\*-\*RT-\*). Glede na pridobljene rezultate smo izbrali varianti metod, za kateri smo pričakovali, da bosta dajali dobre rezultate: 25-FR-ME-\* in 25-RR-RT-\*. Uspešnost obeh variant smo testirali na dveh realnih domenah — rezultati so bili predstavljeni v podpoglavju 5.5.3. Izbranimi variantama metod smo dodali še kombinacijo 50-RR-LR-\*, ki je bila glede na povprečni rang najboljša. Evalvacija je pokazala, da je naš pristop izboljšal priporočanje na realni množici Jester base, in sicer pri matričnih faktorizacijah NG, NS in SN, pri čemer vse tri predlagane metode izboljšajo napovedno točnost. Na realni domeni PEFbase pa je evalvacija pokazala, da se je napovedna točnost priporočanja z upoštevanjem našega pristopa približala napovedni točnosti matrične faktorizacije NS oziroma je naš pristop za manjšo razliko izboljšal priporočanje. V naslednjem koraku smo pogledali, ali so med izbranimi pristopi statistično pomembne razlike. Statistična analiza je pokazala, da je kombinacija 25-FR-ME-NG statistično značilno različna/boljša kakor gole matrične faktorizacije (NG, NS, SN in DF). Evalvacija je torej pokazala, da je kombinacija metod 25-FR-ME-NG najuspešnejša za primere v delnem hladnem zagonu in da ima značilno nižjo napako RRSE od golih matričnih faktorizacij.

Na koncu smo preverili tudi vpliv vstavljanja manjkajočih ocen na ostale uporabnike. Ugotovili smo, da naše spremembe ne povzročajo značilnih sprememb pri priporočanju drugim uporabnikom, ki so zunaj stanja hladnega zagona.

Evalvacija našega pristopa je pokazala, da je naš pristop z uporabe kombinacije metod 50-RR-LR-\* značilno izboljšal (zmanjšal napako RRSE) gole matrične faktorizacije za priporočanje uporabnikom v hladnem zagonu.

## 6.1 Razprava in nadaljnje delo

V poglavju 5 smo opisali metodologijo reševanja situacije hladnega zagona, in sicer na primeru novega uporabnika. Pristop smo razdelili na tri korake. Za vsakega smo predlagali nekaj metod, ki so po našem mnenju ustrezne za posamezno fazo. Pri tem se pojavi vprašanje, katere metode bi še lahko poskusili, saj je lahko mogočih več metod, vendar bi se kompleksnost analize še dodatno povečala. Zato bi bilo v prihodnje primerno raziskati, kako zmanjšati kompleksnost iskanja dobrih in novih kombinacij za vstavljanje manjkajočih vrednosti.

V prvem koraku smo vsem uporabnikom v hladnem zagonu poiskali  $N$  podobnih uporabnikov, in sicer smo za izračun podobnosti uporabili evklidsko razdaljo. V prihodnje bi lahko pogledali še druge mogoče razdalje, ki se uporabljajo za izračun podobnosti dveh vektorjev (npr. PIP-razdalja, kosinusna razdalja itn.)

Pri drugem koraku, v katerem smo znotraj množice CS-uporabniku podobnih uporabnikov določili attribute/objekte, ki jim bomo vstavili manjkajočo vrednost, bi lahko izbrali še druge kriterije za izbor ustreznega atributa. Uporabili smo preproste kriterije za izbor atributov. Po našem mnenju je še veliko neraziskanega na našem raziskovalnem področju — predvsem področje selektivnega vstavljanja manjkajočih vrednosti. V splošnem namreč metode vstavijo vse manjkajoče vrednosti — v našem primeru pa jih želimo vstaviti le nekaterim atributom in za nekatere uporabnike.

V tretjem koraku smo izbranim atributom glede na izbrano metodo izračunali/vstavili manjkajočo oceno. V prihodnje bi bilo zanimivo uporabiti še druge metode strojnega učenja. Pri večini metod smo uporabili intuitivno izbrane pragove (npr. izbira 10 % najboljših vhodnih atributov (neodvisnih spremenljivk), izbira 10 % najboljših ciljnih atributov (odvisnih spremenljivk) ...), pri katerih bi bilo treba vpliv v okviru nadaljnega dela podrobneje raziskati.

Z našim raziskovalnim delom smo odprli povsem novo podpodročje, in sicer področje selektivnega vstavljanja manjkajočih vrednosti redkim vektorjem z namenom izboljšanja priporočanja matričnih faktorizacij. Prepričani smo, da je področje zanimivo in da ima še veliko možnosti za razširitve ter oblikovanje novih pristopov reševanja problema hladnega zagona.

Delo bo v nadaljnje osredotočeno na gradnjo splošnega priporočilnega sistema za podporo učenju, pri katerem bomo izdelali različna učna gradiva (učne objekte) in tako zagotovili zadostno število učnih objektov. Implementirali bomo naš pristop reševanja hladnega zagona v priporočilni sistem in se osredinili na izboljševanje metodologije priporočanja uporabnikom zunaj stanja hladnega zagona. Poseben poudarek bomo namenili zlivanju različnih podatkov (ocen učencev kot rezultat ocenjevanja znanja in ocen učnih objektov kot ocene, ki so jih podali učenci za posamezen učni objekt). Nadaljnje delo bo torej usmerjeno v razvoj in izboljšanje naših pristopov, ki smo jih razvili znotraj disertacije.

*Vprašalnik o pogostosti  
uporabe multimedijskih tipov*

*A*

Ljudje pri učenju s pomočjo IKT (informacijsko-komunikacijsko tehnologijo) uporabljamo različne učne strategije, zato bi vas radi vprašali, kako pogosto in kako radi uporabljate navedene načine učenja (učne strategije) pri svojem učenju.

Pri obeh sklopih ocenite, kako pogosto/radi uporabljati multimedijske tipe pri učenju, pri čemer 1 pomeni nikoli, 2 – včasih, 3 – pogosto, 4 – vedno.

Učim se tako, da:	1 – nikoli	2 – včasih	3 – pogosto	4 – vedno
opazujem/gledam animacije oz. videoposnetke.				
izvajam simulacije oz. igram izobraževalne rač. igre.				
uporabljam gradiva, ki so barvno oblikovno privlačna.				
uporabljam gradiva, ki so dobro strukturirana.				
poslušam avdiogradiva.				

## LITERATURA

- [1] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.
- [2] O. Chapelle, B. Schölkopf, A. Zien et al. *Semi-supervised learning*, zvezek 2. MIT press Cambridge, 2006.
- [3] U. Očepek, Z. Bosnić, I. Nančovska Šerbec in J. Rugelj. Exploring the relation between learning style models and preferred multimedia types. *Computers & Education*, 69, 2013.
- [4] U. Očepek, J. Rugelj in Z. Bosnić. Improving matrix factorization recommendations for examples in cold start. *Expert Systems With Applications*, 42(19), 2015.
- [5] F. Strmčnik. *Didaktika*. Razprave Filozofske fakultete. Ljubljana, 2001.
- [6] R. E. Mayer in R. Moreno. Aids to computer-based multimedia learning. *Learning and Instruction*, 12(1), 2002.
- [7] J. L. Plass, D. M. Chun, R. E. Mayer in D. Leutner. Supporting visual and verbal learning preferences in a second-language multimedia learning environment. *Journal of educational psychology*, 90(1), 1998.
- [8] Y.-C. Chang, W.-Y. Kao, C.-P. Chu in C.-H. Chiu. A learning style classification mechanism for e-learning. *Computers & Education*, 53, 2009.
- [9] S. Graf in T.-C. Liu. Analysis of learners navigational behaviour and their learning styles in an online course. *Journal of Computer Assisted Learning*, 26, 2010.
- [10] F. Dağ in A. Geçer. Relations between online learning and learning styles. *Procedia-Social and Behavioral Sciences*, 1(1), 2009.
- [11] C. Johnston in C. Orwig. Your learning style and language learning. *Lingua Links Library*, 3, 1999.
- [12] V. Yannibelli, D. Godoy in A. Amandi. A genetic algorithm approach to recognise students' learning styles. *Interactive Learning Environments*, 14, 2006.
- [13] L. Vygotsky. Interaction between learning and development. *Readings on the development of children*, 1978.
- [14] J. J. Vogel-Walcutt, J. Gebrim, C. Bowers, T. Carper in D. Nicholson. Cognitive load theory vs. constructivist approaches: which best leads to efficient, deep learning? *Journal of Computer Assisted Learning*, 27, 2011.
- [15] M. Tam. Constructivism, instructional design, and technology: Implications for transforming distance learning. *Educational Technology & Society*, 3, 2000.
- [16] D. Jonassen. Activity theory as a framework for designing constructivist learning environments. *Educational Technology Research and*, 47, 1999.
- [17] A. P. Rovai. A constructivist approach to online college learning. *The Internet and Higher Education*, 7, 2004.
- [18] F. Alonso, D. Manrique in J. Viñes. A moderate constructivist e-learning instructional model evaluated on computer specialists. *Computers & Education*, 53, 2009.
- [19] Q. Wang. Designing a web-based constructivist learning environment. *Interactive Learning Environments*, 17, 2009.
- [20] D. D. Curtis in M. J. Lawson. Exploring collaborative online learning. *Journal of Asynchronous learning networks*, 5(1), 2001.
- [21] D. Hernández-Leo, E. Villasclaras-Fernández, I. Jorrín-Abellán, J. Asensio-Pérez, Y. Dimitriadis, I. Ruiz-Requies in B. Rubia-Avi. Collage, a collaborative learning design editor based on patterns. *Journal of Educational Technology and Society*, 9, 2006.

- [22] H. Cho, G. Gay, B. Davidson in A. Ingrassia. Social networks, communication styles, and learning performance in a cscl community. *Computers & Education*, 49, 2007.
- [23] J. E. Brindley, C. Walti in L. M. Blaschke. Creating effective collaborative learning groups in an online environment. *International Review of Research in Open and Distance Learning*, 10(3), 2009.
- [24] R.-S. Shaw. A study of the relationships among learning styles, participation types, and performance in programming language learning supported by online forums. *Computers & Education*, 58, 2012.
- [25] P. Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6, 1996.
- [26] P. Brusilovsky. Adaptive hypermedia. *User modeling and user-adapted interaction*, 11, 2001.
- [27] C. Wolf. iweaver: towards 'learning style'-based e-learning in computer science education. V *Proceedings of the fifth Australasian conference on Computing education-Volume 20*. Australian Computer Society, Inc., 2003.
- [28] S. Cassidy. Learning styles: An overview of theories, models, and measures. *Educational Psychology*, 24, 2004.
- [29] H. Hauptman in A. Cohen. The synergetic effect of learning styles on the interaction between virtual environments and the enhancement of spatial thinking. *Computers & Education*, 57, 2011.
- [30] E. Y. Huang, S. W. Lin in T. K. Huang. What type of learning style leads to online participation in the mixed-mode e-learning environment? a study of software usage instruction. *Computers & Education*, 58, 2012.
- [31] S. Schiaffino, P. Garcia in A. Amandi. eteacher: Providing personalized assistance to e-learning students. *Computers & Education*, 51(4), 2008.
- [32] E. Özpölat in G. B. Akar. Automatic detection of learning styles for an e-learning system. *Computers & Education*, 53(2), 2009.
- [33] D. A. Kolb. *Experiential learning: Experience as the source of learning and development*, zvezek 1. Prentice-Hall Englewood Cliffs, NJ, 1984.
- [34] A. Y. Kolb in D. A. Kolb. Learning styles and learning spaces: Enhancing experiential learning in higher education. *Academy of Management Learning & Education*, 4, 2005.
- [35] L. Mos, J. Royce in U. of Alberta. Center for Advanced Study in Theoretical Psychology. *Manual, Psycho-epistemological Profile*. Center for Advanced Study in Theoretical Psychology, University of Alberta, 1980.
- [36] A. Abdennur. *Cultural Background and the Epistemic Orientation of University Students: An Exploratory Study*. University of Ottawa, 1998.
- [37] R. Rancourt. *K.A.M.I.: Consultant's manual to style interpretation*. Impact Inc., Ontario, 1988.
- [38] K. Kilpinen. *Reflective teacher using the computer-network in teaching; how the psycho-epistemological learning styles help to better design learning environments*. Tampereen yliopisto, 2004.
- [39] R. Zenhausern. Hemispheric dominance. *Learning Styles Network Newsletter*, 1(2), 1980.
- [40] R. Dunn. Hemispheric preference: the newest element of learning style. *American Biology Teacher*, 44(5), 1982.
- [41] C. R. Reynolds, B. Katsounis in E. P. Torrance. A children's form of your style of learning and thinking: Preliminary norms and technical data. *Gifted Child Quarterly*, 23(4), 1979.
- [42] N. D. Fleming in C. Mills. Not another inventory, rather a catalyst for reflection. *To Improve the Academy*, 11, 1992.
- [43] O. Akdemir in T. A. Koszalka. Investigating the relationships among instructional strategies and learning styles in online environments. *Computers & Education*, 50, 2008.
- [44] N. Othman in M. H. Amiruddin. Different perspectives of learning styles from vark model. *Procedia - Social and Behavioral Sciences*, 7, 2010.
- [45] C. Rose. *Accelerated learning*. Dell Pub. Co., 1987.
- [46] K. C. Briggs. *Myers-Briggs type indicator*. Consulting Psychologists Press Palo Alto, CA, 1976.
- [47] S. Graf in T. Lin. The relationship between learning styles and cognitive traits – getting additional information for improving student modelling. *Computers in Human Behavior*, 24, 2008.
- [48] S. Graf, T.-C. Liu, N.-S. Chen in S. J. Yang. Learning styles and cognitive traits – their relationship and its benefits in web-based educational systems. *Computers in Human Behavior*, 25, 2009.
- [49] J. Tseng, H. Chu, G. Hwang in C. Tsai. Development of an adaptive learning system with two sources of personalization information. *Computers & Education*, 51, 2008.



- [50] D. A. Kolb. *Experiential learning: Experience as the source of learning and development*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1985.
- [51] B. Marentič-Požarnik. *Izziv raznolikosti: stili spoznavanja, učenja, mišljenja*, poglavje Kolbov vprašalnik o učnih stilih. Educa, Nova Gorica, 1995.
- [52] B. Marentič-Požarnik in C. Peklaj. *Izziv raznolikosti: stili spoznavanja, učenja, mišljenja*, poglavje Vprašalnik načinov in stilov spoznavanja po Rancourtu. Educa, Nova Gorica, 1995.
- [53] C. Peklaj. *Izziv raznolikosti: stili spoznavanja, učenja, mišljenja*, poglavje Vprašalnik: Tvoj stil učenja in razmišljanja. Educa, Nova Gorica, 1995.
- [54] M. Stražisar. *Psihologija, spoznanja in dileme*, poglavje Učenje in pomnjenje: Predelava podatkov za dolgotrajno pomnjenje. DZS, Ljubljana, 2004.
- [55] D. Demšar, S. Džeroski, T. Larsen, J. Struyf, J. Axelssen, M. B. Pedersen in P. H. Krogh. Using multi-objective classification to model communities of soil microarthropods. *Ecological Modelling*, 191(1), 2006.
- [56] J. Struyf in S. Džeroski. Constraint based induction of multi-objective regression trees. V *4th Int'l Workshop on Knowledge Discovery in Inductive Databases: Revised Selected and Invited Papers*, Lecture OPTPages in Computer Science. Springer, 2006.
- [57] D. Kocev, S. Džeroski, M. D. White, G. R. Newell in P. Griffioen. Using single- and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecological Modelling*, 220(8), 2009.
- [58] H. Blockeel in J. Struyf. Efficient algorithms for decision tree cross-validation. *The Journal of Machine Learning Research*, 3, 2003.
- [59] M. Garofalakis, D. Hyun, R. Rastogi in K. Shim. Building decision trees with constraints. *Data Mining and Knowledge Discovery*, 7(2), 2003.
- [60] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7, 2006.
- [61] J. Lapuh Bele in J. Rugelj. Comparing efficiency of web based learning contents on different media. *International Journal of Emerging Technologies in Learning (ijET)*, 4(S13), 2009.
- [62] G. Adomavicius in A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6), 2005.
- [63] M. M. Recker in D. A. Wiley. A non-authoritative educational metadata ontology for filtering and recommending learning objects. *Interactive learning environments*, 9(3), 2001.
- [64] S. Rafaeli, M. Barak, Y. Dan-Gur in E. Toch. Qsia—a web-based environment for learning, assessing and knowledge sharing in communities. *Computers & Education*, 43(3), 2004.
- [65] H. Avancini in U. Straccia. User recommendation for collaborative and personalised digital archives. *International Journal of Web Based Communities*, 1(2), 2005.
- [66] Y.-M. Huang, T.-C. Huang, K.-T. Wang in W.-Y. Hwang. A markov-based recommendation model for exploring the transfer of learning on the web. *Educational Technology & Society*, 12(2), 2009.
- [67] L.-P. Shen in R.-M. Shen. *Learning content recommendation service based-on simple sequencing specification*. Springer, 2004.
- [68] H. Drachsler, D. Pecceu, T. Arts, E. Hutten, L. Rutledge, P. Van Rosmalen, H. Hummel in R. Koper. *ReMashed—recommendations for mash-up personal learning environments*. Springer, 2009.
- [69] A. Klačnja-Miličević, B. Vesin, M. Ivanović in Z. Budimac. E-learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education*, 56, 2010.
- [70] A. Klačnja-Miličević, B. Vesin, M. Ivanović in Z. Budimac. Integration of recommendations and adaptive hypermedia into java tutoring system. *Computer Science and Information Systems*, 8, 2011.
- [71] B. A. Solomon in R. M. Felder. Index of learning styles. *Raleigh, NC: North Carolina State University*. Available online, 1999.
- [72] K. I. Ghauth in N. A. Abdullah. Learning materials recommendation using good learners' ratings and content-based filtering. *Educational Technology Research and Development*, 58, 2010.
- [73] N. Capuano, G. Mangione, A. Pierri in S. Salerno. Learning goals recommendation for self regulated learning. *International Journal of Engineering Education*, 28, 2012.
- [74] N. Capuano, R. Iannone in M. Gaeta. A recommender system for learning goals. *Information Systems, E-learning, and Knowledge Management Research*, 2013.
- [75] R. M. Wachter in J. N. Gupta. Distance education and the use of computers as instructional tools for systems development projects: A case study of the construction of expert systems. *Computers & Education*, 29(1), 1997.

- [76] B. P. Woolf. *Building intelligent interactive tutors*. Elsevier Inc., Burlington, USA, 2009.
- [77] I. Arroyo, B. Woolf, W. Burelson, K. Muldner, D. Rai in M. Tai. A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect. *International Journal of Artificial Intelligence in Education*, 24(4), 2014.
- [78] Projektna skupina. Pedagoški terminološki slovar, url <http://pedagoški.slovarji.si/search.aspx>, 2008. Dostopano dne: 6. 5. 2015.
- [79] P. B. Kantor, F. Ricci, L. Rokach in B. Shapira. *Recommender systems handbook*. Springer, 2011.
- [80] A. Kristofić. Recommender system for adaptive hypermedia applications. V *IIT. SRC 2005: Student Research Conference*, 2005.
- [81] R. Farzan in P. Brusilovsky. Social navigation support in a course recommendation system. V *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer, 2006.
- [82] R. Koper. Increasing learner retention in a simulated learning network using indirect social interaction. *Journal of Artificial Societies and Social Simulation*, 8(2), 2005.
- [83] X. Su in T. Khoshgoftaar. Collaborative filtering for multi-class data using belief nets algorithms. 2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06), 2006.
- [84] Y. Koren, R. Bell in C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.
- [85] G. Takács in I. Pálásy. Investigation of various matrix factorization methods for large recommender systems. *Data Mining Workshops, 2008. ICDMW'08. IEEE International*, 1, 2008.
- [86] S. Funk. Netflix update: Try this at home, 2006. URL <http://sifter.org/~simon/journal/20061211.html>, 2011.
- [87] R. M. Bell in Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 2007.
- [88] G. Takács in D. Tikk. Alternating least squares for personalized ranking. *Proceedings of the sixth ACM conference on Recommender systems - RecSys '12*, 2012.
- [89] C. Ding, T. Li in M. I. Jordan. Convex and semi-nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1), 2010.
- [90] Q. Mo in B. A. Draper. Semi-nonnegative matrix factorization for motion segmentation with missing data. V *Computer Vision—ECCV 2012*. Springer, 2012.
- [91] M. Žitnik in B. Zupan. Data fusion by matrix factorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(1), 2015.
- [92] M. Žitnik in B. Zupan. Matrix factorization-based data fusion for gene function prediction in baker's yeast and slime mold. V *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, zvezek 19, 2013.
- [93] J. Bobadilla, F. Ortega, A. Hernando in J. Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Systems*, 26, 2012.
- [94] A. Schein in A. Popescul. Methods and metrics for cold-start recommendations. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002.
- [95] S. Park in W. Chu. Pairwise preference regression for cold-start recommendation. *Proceedings of the third ACM conference on Recommender systems*, 2009.
- [96] V. Formoso, D. Fernández, F. Cacheda in V. Carneiro. Using profile expansion techniques to alleviate the new user problem. *Information Processing & Management*, 49, 2013.
- [97] C. W.-K. Leung, S. C.-F. Chan in E.-L. Chung. An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge-Based Systems*, 21, 2008.
- [98] H. J. Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178, 2008.
- [99] H. Liu, Z. Hu, A. Mian, H. Tian in X. Zhu. A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, 56, 2014.
- [100] J. Bobadilla, F. Ortega in A. Hernando. A collaborative filtering similarity measure based on singularities. *Information Processing & Management*, 48, 2012.
- [101] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan in J. Riedl. Getting to know you: learning new user preferences in recommender systems. V *Proceedings of the 7th international conference on Intelligent user interfaces*. ACM, 2002.
- [102] A. Rashid, G. Karypis in J. Riedl. Learning preferences of new users in recommender systems: an information theoretic approach. *ACM SIGKDD Explorations Newsletter*, 2008.

- [103] B. Lika, K. Kolomvatos in S. Hadjiefthymiades. Facing the cold start problem in recommender systems. *Expert Systems with Applications*, 41(4, Part 2), 2014.
- [104] A. Rosli, T. You, I. Ha, K.-Y. Chung in G.-S. Jo. Alleviating the cold-start problem by incorporating movies facebook optpages. *Cluster Computing*, 2014.
- [105] C. C. Chen, Y. H. Wan, M. C. Chung in Y. C. Sun. An effective recommendation method for cold start new users using trust and distrust networks. *Information Sciences*, 224, 2013.
- [106] J. T. Kent. Information gain and a general measure of correlation. *Biometrika*, 70(1), 1983.
- [107] I. Kononenko. *Strojno učenje*. Fakulteta za računalništvo in informatiko, 2005.
- [108] M. Robnik-Šikonja in I. Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, 53(1-2), 2003.
- [109] K. Goldberg, T. Roeder, D. Gupta in C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2), 2001.