SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1569

ANALIZA I IMPLEMENTACIJA ALGORITMA ZA SMANJENJE DIMENZIONALNOSTI DEKOMPOZICIJOM NA SINGULARNE VRIJEDNOSTI

Irena Kezić

Zagreb, studeni 2005.

Zahvaljujem mentorici prof.dr.sc. Dalbelo Bašić na podršci i stručnom vodstvu pri izradi diplomskog rada

Sadržaj

1. Uvod 5
2. Prikaz dokumenata u dubinskoj analizi teksta 6
2.1. Eliminacija 'stop riječi' 6
2.2. Morfološka normalizacija 6
2.3. Matrica 'riječ-dokument' 7
2.4. Pridruživanje težina izrazima 8
3. Pristupi redukciji dimenzionalnosti 10
3.1. Transformacija obilježja 11
3.1.1. Latentno semantičko indeksiranje 11
3.1.2. Kanonska korelacijska analiza 12
3.1.3. Nasumična projekcija 12
3.1.4. Analiza nezavisnih komponenti 13
3.2. Selekcija obilježja 14
3.2.1. Frekvencija dokumenta 14
3.2.2. Metoda približnih skupova 14
4. Dekompozicija na singularne vrijednosti 16
4.1. Uvodni pojmovi 16
4.2. Dekompozicija na singularne vrijednosti 17
4.3. Izračunavanje SVD-a 20
5. Optimizacijske metode za izračunavanje SVD-a 21
5.1. Kratki pregled algoritama 21
5.2. Simetrična Lanczos metoda 22
5.2.1. Svojstva Krylovog podprostora i Krylove matrice 22
5.2.2. Tridijagonalizacija 24
5.2.3. Zaustavljanje i ograničenje pogreške 25
6. Latentno semantičko indeksiranje 28
6.1. Ilustracija rada 28
6.2. Preslikavanje dokumenata iz skupa za testiranje 29
6.3. Dodavanje dokumenata 29
6.3.1. Ponovno izračunavanje 32
6.3.2. Umatenje dokumenata 33
6.3.3. SVD-osvježavanje 35

7. Klasifikacija metodom *k najbližih susjeda* **36**
7.1. Karakteristike metode *k najbližih susjeda* 36
7.2. Opis algoritma 36
8. Implementacija **38**
8.1. Ulazni podaci 38
8.2. Grafičko korisničko sučelje 39
8.2.1. Predprocesiranje 39
8.2.2. Redukcija dimenzionalnosti 40
8.2.3. Klasifikacija 41
8.3. Opis sustava 41
8.3.1. Stvaranje 'riječ-dokument matrice' 42
8.3.2. Redukcija dimenzionalnosti skupa za učenje 43
8.3.3. Preslikavanje dokumenata iz skupa za testiranje u novi *k* - dimenzionalni prostor 45
8.3.4. Klasifikacija i evaluacija 45
9. Eksperimenti i rezultati **47**
9.1. Paralelni hrvatsko-engleski korpus *Croatia Weekly* 47
9.1.1. Ovisnost uspješnosti klasifikacije o jeziku i broju dimenzija 48
9.1.2. Usporedba uspješnosti klasifikacije s i bez redukcije dimenzionalnosti 50
9.1.2.1. Uspješnost klasifikacije po kategorijama 51
9.1.2.2. Usporedba uspješnosti klasifikacije i utrošenog vremena 52
9.1.3. Vrijeme učenja klasifikatora 54
9.2. Baza novinskih članaka *Vjesnik* 56
9.2.1. Automatska morfološka normalizacija 57
9.2.2. Usporedba uspješnosti klasifikacije s i bez redukcije dimenzionalnosti 58
9.2.3. Ovisnost uspješnosti klasifikacije o morfološkom prikazu teksta 59
9.2.4. Uspješnosti klasifikacije bez kategorije *Tema dana* 62
10. Zaključak **64**
11. Literatura **65**
12. Dodatak **68**
12.1. Primjer izračunavnja svojstvenih vrijednosti i vektora matrice 68
12.2. Primjer izračunavanja dekompozicije matrice na singularne vrijednosti 68

1. Uvod

Povećanjem broja dokumenata dostupnih na internetu raste potreba za alatima koji pomažu pri pronalaženju, filtriranju i rukovanju resursima. U tu svrhu koriste se mnogi statistički algoritmi i tehnike strojnog učenja. Ovi algoritmi upotrebljavaju se za automatsko sortiranje novinskih članaka, web stranica i elektronske pošte kao i za učenje interesa čitatelja. Kako količina tekstualnih dokumenata raste sve brže metode dubinske analize teksta (eng. text mining) nailaze na sve veće izazove.

Tekstualni dokumenti po prirodi su nestrukturirani. Velika dimenzionalnost predstavlja problem prilikom primjene standardnih postupaka za grupiranje i klasifikaciju pomoću računala. Upravo utjecaj redukcije dimenzionalnosti na klasifikaciju teksta bit će temom ovog rada.

Rad se sastoji od tri cjeline. U prvoj cjelini biti će objašnjeno što je to redukcija dimenzionalnosti teksta te njena uloga u klasifikaciji teksta. Dati ćemo kratak pregled najčešće korištenih metoda za redukciju dimenzionalnosti teksta. U drugom dijelu detaljno će biti objašnjena metoda l*atentno semantičkog indeksiranja* za redukciju dimenzionalnosti teksta te metoda *k najbližih susjeda* za klasifikaciju teksta. Prve dvije cjeline služe kao osnova za treću koja opisuje implementirani klasifikator te prikazuje i vrednuje dobivene rezultate.

2. Prikaz dokumenata u dubinskoj analizi teksta

Svrha dubinske analize teksta je klasifikacija ili grupiranje tekstualnih dokumenata kada njihova količina nadilazi ljudske sposobnosti obrade podataka u realnom vremenu. Čovjeku jednostavno razumljive tekstualne dokumente potrebno je pretvoriti u oblik koji omogućuje dubinsku analizu računalom. Za razliku od čovjeka koji razumije sadržaj pojedinog dokumenta pisanog prirodnim jezikom računalo zahtijeva strukturirane podatke. Stoga je bitno prije opisa postupaka dubinske analize teksta objasniti način pohrane te predprocesiranja dokumenata.

Kako prikaz teksta ima velik utjecaj na učinkovitost klasifikacije i sposobnost generalizacije potrebno je odabrati prikaz teksta koji odgovara korištenom algoritmu klasifikacije odnosno redukcije dimenzionalnosti. Postoji nekoliko načina predprocesiranja i pohrane dokumenata, ali u ovom poglavlju će biti objašnjene samo one metode bitne za ovaj rad.

2.1. Eliminacija 'stop riječi'

Sve riječi u dokumentu nisu jednako bitne. Riječi koje ne nose informaciju o pripadnosti dokumenta jednoj od kategorija zovu se 'stop riječi'. 'Stop riječi' su riječi s visokom frekvencijom pojavljivosti. Najčešće su to prilozi, prijedlozi i veznici, a razlikuju se u svakom jeziku (npr. neke stop riječi u hrvatskom jeziku su: i, pa, u, ali...). Potrebno je formirati listu 'stop riječi' te se potom iz dokumenata eliminiraju riječi iz liste.

2.2. Morfološka normalizacija

Morfološka normalizacija je postupak svođenja riječi na osnovni oblik[1] i bitan je korak u predprocesiranju tekstualnih dokumenata. Naime, u govornom jeziku jedna riječ pojavljuje se u različitim oblicima (npr. u raznim vremenima, padežima...). Bez morfološke normalizacije svaki od oblika riječi bio bi zabilježen kao poseban vektor riječi. Primjenom morfološke

[1] Osnovni oblik riječi ili lema. Stoga se proces morfološke normalizacije zove i lematizacija.

normalizacije svi oblici riječi svode se na osnovni oblik. Pokažimo to primjerom:

zakon, zakonom, zakoni, zakona → zakon.

Ovom metodom se znatno smanjuje broj pojavnica u skupu dokumenata. Osim smanjenja dimenzionalnosti prostora primjenom morfološke normalizacije dolazi do poboljšanja rezultata klasifikacije koje ovisi o morfološkom bogatstvu[2] jezika. Normalizacija morfološki siromašnih jezika, poput engleskog jezika, nije nužna, ali daje značajna poboljšanja za morfološki bogate jezike kao što je hrvatski jezik.

2.3. Matrica 'riječ-dokument'

Najčešći način prikaza tekstualnih dokumenta za potrebe računalne obrade je vektorska reprezentacija (eng. Vector Space Model). Osnovni model vektorske reprezentacije pretpostavlja da se konceptualno značenje dokumenta može izvesti iz riječi od kojih je dokument sačinjen. Iz skupa dokumenata se formira *m x n* (gdje je *m* broj riječi, a *n* broj dokumenata) matrica čiji redci predstavljaju 'riječ vektore', a stupci 'dokument vektore'. Takva matrica predstavlja skup 'riječ-dokument' asocijacija.

	D_1	D_2	…	D_n
R_1	a_{11}	a_{12}	…	a_{1n}
R_2	a_{21}	a_{22}		a_{2n}
⋮	⋮		⋱	⋮
R_m	a_{m1}	a_{m2}	…	a_{mn}

Slika 1. Riječ-dokument matrica

Elementi matrice su frekvencije pojavljivanja pojedine riječi u dotičnom dokumentu

$$A = [a_{ij}]$$

[2] Jezik je morfološki bogat ako obiluje različitim oblicima riječi.

gdje je a_{ij} frekvencija pojavljivanja riječi *i* u dokumentu *j*. Za računanje elemenata matrice umjesto frekvencije moguće je koristiti i neku od težinskih funkcija.

Model vektorske reprezentacije sve riječi i dokumente preslikava u višedimenzionalni prostor. Budući da dimenzija tog prostora raste s povećanjem broja riječi i raznolikošću sadržaja, kod velikih skupova dokumenata dimenzija prostora može biti prevelika za računalnu obradu.

2.4. Pridruživanje težina izrazima

Pridruživanje težina izrazima bitno utječe na performanse sustava. Zasniva se na sljedećim pretpostavkama:

- izrazi koji se pojavljuju u jako puno ili jako malo dokumenata su manje važni od onih koji se pojavljuju umjereno u skupu dokumenata
- dugački dokumenti nisu važnijih od kratkih
- višestruka pojavljivanja izraza u dokumentu nisu manje važna od jednostrukih.

Težine izraza se obično sastoje od dvije komponente: komponente dokumenta i komponente kolekcije.

(2.1.) $$a_{ij} = L_{ij}G_{ij}.$$

Komponenta dokumenta može se računati na slijedeće načine:

- *binarna metoda*: $L_{ij} = \begin{cases} 1, & ako\ se\ riječ\ pojavila\ u\ dokumentu \\ 0, & inace \end{cases}$
- *frekvencija izraza:* $L_{ij} = TF(w_i, d_j)$, broj pojavljivanja izraza w_i u dokumentu d_j
- normalizirana frekvencija: $L_{ij} = 0{,}5 + 0{,}5\frac{TF(w_i, d_j)}{\max_k TF(w_k, d_j)}$

Komponenta kolekcije može se računati na slijedeće načine:

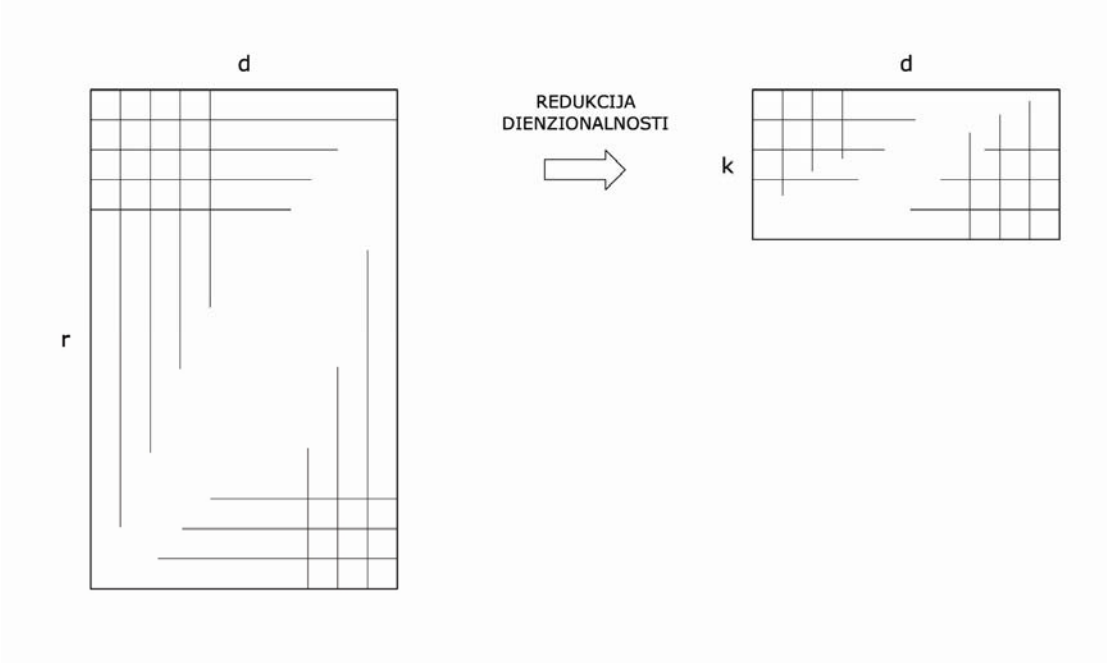
- ignoriramo frekvenciju dokumenta: $G_{ij} = 1{,}0$

- inverzna frekvencija dokumenta: $G_{ij} = \log \frac{N}{DF(w_i)}$, gdje je N ukupan broj dokumenata, $DF(w_i)$ broj dokumenata u kojima se izraz w_i pojavljuje barem jedanput
- probabilistički inverz frekvencije dokumenata: $G_{ij} = \log \frac{N - DF(w_i)}{DF(w_i)}$, gdje je N ukupan broj dokumenata, $DF(w_i)$ broj dokumenata u kojima se izraz w_i pojavljuje barem jedanput

Kako bi izbjegli pridjeljivanje veće važnosti dužim dokumentima koristi se normalizacijska komponenta. Normalizacijska komponenta normalizira vektor na duljinu jedan u 2-normi: $\frac{1}{\sqrt{x_j^2}}$.

3. Pristupi redukciji dimenzionalnosti

Kako novi problemi postavljaju sve veće zahtjeve računalnim resursima sve češće se javlja potreba za analizom podataka velike dimenzionalnosti. Postaje teško razumjeti skrivenu strukturu takvih podataka: «od šume često ne vidimo stablo». Nadalje, pohrana, prijenos i obrada podataka velike dimenzionalnosti postavljaju velike zahtjeve sustavu. Stoga je poželjno smanjiti dimenzionalnost podataka te istovremeno zadržati što više informacija i originalnu strukturu.



Slika 2. Redukcija dimenzionalnosti

U svrhu rješavanja problema velike dimenzionalnosti podataka predložene su brojne metode. Postoje dva osnovna pristupa redukciji dimenzionalnosti:

1. transformacija obilježja i
2. selekcija obilježja.

Kako bismo dobili bolji uvid u samu problematiku redukcije dimenzionalnosti, u ovom poglavlju će ukratko biti objašnjene najčešće korištene metode oba pristupa.

3.1. Transformacija obilježja

Kod transformacije obilježja originalni podaci velike dimenzinalnosti preslikaju se u novi prostor manje dimenzionalnosti. U novom prostoru svaka značajka je linearna ili nelinearna kombinacija značajki originalnog prostora. Cilj je u novom prostoru zadržati udaljenosti među podatkovnim vektorima. Smatra se da prostor manje dimenzionalnosti predstavlja skrivenu latentno semantičku strukturu originalnog skupa podataka. Najčešće upotrebljavane metode su *latentno semantičko indeksiranje*, *analiza nezavisnih komponenti*, *nasumična projekcija*, *kanonska korelacijska analiza*... Utvrđeno je da su ove metode manje učinkovite kada u skupu podataka postoji više nebitnih nego bitnih dimenzija. (Tang, et al., 2004)

3.1.1. Latentno semantičko indeksiranje

Metoda *latentno semantičkog indeksiranja* (eng. Latent Semantic Indexing, LSI) prvotno je osmišljena kao učinkovit alat za automatsko indeksiranje i pretraživanje podataka. Pronalaženjem semantičke strukture visokog reda (veza 'riječ-dokument') rješava problem višeznačnosti govornog jezika. Za preslikavanje originalnog prostora velike dimenzionalnosti u prostor manje dimenzionalnosti s minimalnom pogreškom udaljenosti upotrebljava se dekompozicija na singularne vrijednosti (SVD). Dimenzije novog prostora su ortogonalne što znači da su nezavisne. Pokazalo se da se korištenjem SVD-a zadržavaju najvažnije veze između riječi i dokumenata te učinkovito odstranjuje šum, redundancija i višeznačnost unutar skupa podataka.

Mana ove metode je velik stupanj složenosti. Za 'riječ-dokument matricu' A_{mxn} vremenska složenost je reda $O(m^2n)+ O(m^3)$ (Tang, et al., 2004). Za rijetke matrice dimenzija *m x n*, koristeći računalno učinkovitije metode, složenost je reda O(*cmn*) gdje je *c* prosječan broj ne-nul elemenata za sve podatkovne vektore.

Više o ovoj metodi bit će rečeno nešto kasnije.

3.1.2. Kanonska korelacijska analiza

Kanonska korelacijska analiza (eng. Canonical Correlation Analysis, CCA) je metoda redukcije dimenzionalnosti temeljena na generaliziranom SVD-u. Ova metoda se upotrebljava za procjenu povezanosti dviju višedimenzionalnih varijabli. Metoda omogućava uporabu dviju različitih reprezentacija istog semantičkog objekta (npr. tekstualni dokument napisan na dva jezika) u svrhu pronalaženja semantičke strukture. (Fortuna, 2004)

Pretpostavimo da postoje dvije inačice nekog skupa podataka napisane na dva različita jezika. Cilj *kanonske korelacijske analize* je pronalaženje novog semantičkog prostora zajedničkog objema inačicama te preslikavanje svake od njih u novi prostor. Preslikavanjem svih dokumenata obaju skupova u novi prostor dobiva se reprezentacija neovisna o jeziku. Koristeći ovu metodu klasični algoritmi strojnog učenja mogu biti primijenjeni na višejezične skupove podataka.

3.1.3. Nasumična projekcija

Metoda *nasumične projekcije* (eng. Random Projection, RP) je zamišljena kao računarski manje zahtjevna alternativa metodi *latentno semantičkog indeksiranja*. Za razliku od prostora generiranog *LSI* metodom, u kojem su sve dimenzije ortogonalne, dimenzije novog *k*-dimenzionalnog prostora generiranog *metodom nasumične projekcije* su približno ortogonalne. Može se dokazati da je distorzija pogreške udaljenosti, uvedena ovim pristupom, unutar novog prostora dobro ograničena.

Slično metodi LSI, za preslikavanje originalne matrice A_{mxn} u novi *k*-dimenzionalni prostor koristi se nasumično generirana projekcijska matrica R, $\tilde{A}_{[kxn]} = R_{[kxn]} \times A_{[mxn]}$. Stupanj složenosti generiranja matrice R je O(*mk*). (Tang, et al., 2004.)

Nije sigurno je li uporaba *metode nesumične projekcije* prikladna u dubinskoj analizi teksta. Za razliku od drugih metoda transformacije obilježja, koje generiraju nova obilježja na osnovu statističkih svojstava, u ovom slučaju nova obilježja se generiraju nasumično (nasumične linearne kombinacije originalnih izraza). Ova slučajnost može pridonijeti povećanju

višeznačnosti govornog jezika i smanjiti učinkovitost *metode nasumične projekcije* pri redukciji dimenzionalnosti u klasifikaciji teksta.

3.1.4. Analiza nezavisnih komponenti

Analiza nezavisnih komponenti (eng. Independent Component Analysis, ICA) je statistička metoda opće namjene koja linearno transformira originalne podatke na komponente koje su u statističkom smislu maksimalno nezavisne jedna od druge. Za razliku od metode LSI statistički nezavisne komponente nisu nužno međusobno ortogonalne.

Primjena ove metode u tekstualnim aplikacijama započela je nedavno. Proširenje standardne ICA metode uspješno se koristi za identifikaciju naslova u dinamičkom tekstualnom okruženju, npr. chat room konverzacija.

Cilj *analize nezavisnih komponenti* je iz vektora originalnih varijabli ('riječ vektora') y pronaći vektor nepoznatih latentnih varijabli x. Pretpostavka je da su varijable vektora y linearna kombinacija od x:

$$y = Ax,$$

gdje je A *m x k* matrica (*m* ≥ *k* , *m* je broj riječi, *k* je dimenzija novog prostora). Matrica A zove se 'matrica miješanja', a određuje se iz 'separacijske matrice' B reverznog modela

$$x \approx x' = By.$$

Funkcija cilja ICA metode mjeri negaussnost komponenata koja bi trebala biti minimizirana.

3.2. Selekcija obilježja

Selekcija obilježja, kako samo ime govori, odabire podskup bitnih ili korisnih dimenzija (specifičnih za aplikaciju) iz originalnog skupa dimenzija. Cilj je pronaći bitne podskupove prikladne za svaku kategoriju zasebno. Sada će biti opisani predstavnici ove grupe *metoda približnih skupova i frekvencija dokumenta.*

3.2.1. Frekvencija dokumenta

Frekvencija dokumenta (eng. Document Frequency, DF) može biti upotrebljena kao osnova za selekciju obilježja. Temeljna ideja ove metode je da se u vektoru obilježja pojavljuju samo one dimenzije s visokim DF vrijednostima. Iako jednostavna pokazalo se da je ova metoda jednako učinkovita kao mnogo naprednije tehnike kategorizacije teksta.

Metodu *frekvencije dokumenta* formalno možemo definirati na sljedeći način. Promotrimo kolekciju dokumenata predstavljenu matricom A dimenzija *m x n* (gdje je *m* broj različitih riječi koje se pojavljuju u svim dokumentima u kolekciji, a *n* broj dokumenata te je $m >> n$). DF vrijednost izraza *t*, DF_t, je definirana kao broj dokumenata u kojima se *t* pojavljuje barem jednom. Da bi smanjili dimenzionalnost matrice A s *m* na *k* ($k<m$) odabiremo *k* dimenzija s najvećim DF vrijednostima.

Očito je da je stupanj složenosti izračunavanja DF vrijednosti O(*mn*).

3.2.2. Metoda približnih skupova

Teorija približnih skupova[3] (engl. Rough Set Theory) je formalna matematička metoda koja može biti upotrebljena pri reduciranju dimenzionalnosti skupova podataka. Teorija je orijentirana na otkrivanje uzoraka, pravilnosti i znanja u velikim količinama podataka. Teorija je od tada primijenjena u brojnim aplikacijama specijaliziranim za dubinsku analizu podataka (engl. data mining).

[3] Koncept približnih skupova iznio je dr. Zdslaw Pawlak 1982.g.

Pawlak-ova teorija približnih skupova, iako jako dobar model za relativno mali broj atributa, u slučaju klasifikacije teksta potpuno je neupotrebljiva. Zbog toga se u dubinskoj analizi teksta koristi pojednostavljen algoritam koji koristi heuristiku, a zasniva se na Pawlak-ovom modelu. Ideja algoritma je inkrementalno dodavati jednu po jednu riječ, koja se pojavila u skupu za učenje, novom skupu riječi. Novi skup predstavlja reducirani skup dimenzija originalnog skupa. Dimenzija koja povećava udaljenost dokumenata koji pripadaju različitim kategorijama bit će dodana skupu. Redoslijed odabira riječi koju ćemo slijedeću ispitati određuje se heuristički.

Problem ove metode je velik stupanj složenosti koji za najgori slučaj iznosi O(n!). Prikladnost uporabe metode za redukciju dimenzionalnosti teksta nije do kraja ispitana.

4. Dekompozicija na singularne vrijednosti

Implementirani sustav zasniva se na dekompoziciji 'riječ-dokument matrice' na singularne vrijednosti. U ovom poglavlju biti će objašnjena matematička pozadina modela korištenog za redukciju dimenzionalnosti.

4.1. Uvodni pojmovi

Sada će biti definirani osnovni pojmovi potrebni za razumijevanje narednih poglavlja.

Definicija 4.1. (Kurepa, 1985) Za kvadratnu matricu A kažemo da je *regularna* ako postoji kvadratna matrica B takva da je

$$AB = BA = I, \tag{4.1.}$$

gdje je I jedinična matrica. Kažemo da je kvadratna matrica C *singularna* ako nije regularna.

Dokažimo da je matrica B jednoznačno određena s (4.1.). Uzmimo da je

$$AC = CA = I$$

za neku drugu kvadratnu matricu C. Tada je

$$C = C \cdot I = C \cdot (AB) = (CA) \cdot B = I \cdot B = B, \text{ tj. } C = B.$$

Budući da je B jednoznačno određena uvjetom (4.1.), pišemo $B = A^{-1}$ i matricu A^{-1} zovemo *inverzna* matrica matrice A.

Definicija 4.2. (Kurepa, 1985) Broj linearno nezavisnih redaka matrice A jednak je broju linearno nezavisnih stupaca te matrice. Taj broj zove se *rang matrice* A i označava se sa r(A). Matrice A i A^T imaju isti rang.

Definicija 4.3. (Elezović, 2003) Neka je A kvadratna realna matrica *n*-tog reda. Realan broj λ zove se *svojstvena (karakterisitčna) vrijednost matrice* A ako postoji vektor $v \in R_n$ $v \neq 0$ takav da je

$$Av = \lambda v. \tag{4.2.}$$

Za vektor v kažemo da je svojstven (karakterističan) vektor matrice A i da pripada svojstvenoj vrijednosti λ.

Nalaženje svojstvenih vektora i svojstvenih vrijednosti matrice. Jednadžba Av = λv ekvivalentna je s

(4.3.) $$(\lambda I - A)v = 0.$$

Dakle, v je svojstveni vektor ako i samo ako pripada jezgri matrice (λI – A). Ovaj uvijet govori o načinu na koji se mora birati skalar *λ*. Da bi jednadžba (4.3.) imala netrivijalno rješenje, matrica *λI – A* ne smije biti regularna odnosno njena determinanta mora biti jednaka nuli:

$$|\lambda I - A| = 0.$$

Ova je determinanta polinom po nepoznanici *λ*, stupnja *n*. Nazivamo ga *karakteristični polinom* matrice A i označavamo s

$$\circ(\lambda) = \det(\lambda I - A) .$$

Jednadžba

$$\circ(\lambda) = \det(\lambda I - A) = 0$$

naziva se *karakteristična jednadžba* matrice A. Njena rješenja su svojstvene vrijednosti matrice A.

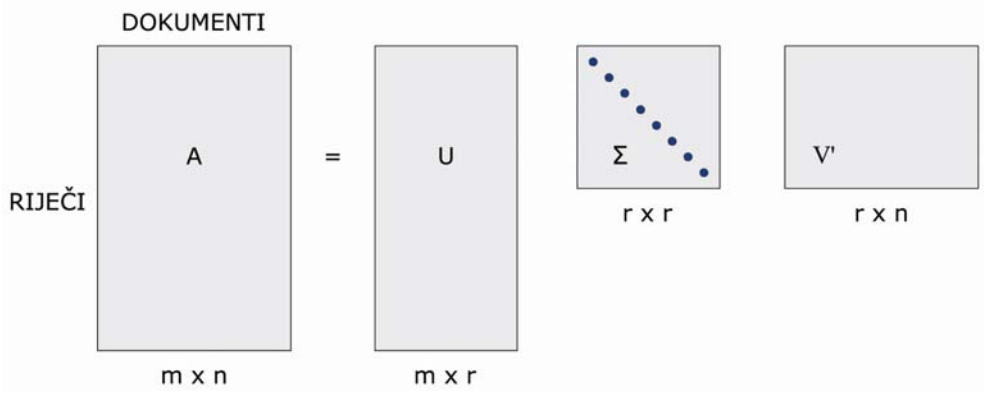
4.2. Dekompozicija na singularne vrijednosti

Za matricu A dimenzija *m x n*, gdje je *m* >> *n* i rang(A) = *r*, dekompozicija na singularne vrijednosti, SVD(A), je definirana kao

(4.4.) $$A = U\Sigma V^T$$

gdje je $U^TU = V^TV = I_n$ i $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$, $\sigma_i > 0$ za $1 \le i \le r$, $\sigma_j = 0$ za $j \ge r + 1$. Prvih *r* stupaca ortogonalnih matrica U i V određuju ortogonalne singularne vektore povezane s *r* ne-nul svojstvenih vrijednosti od AA^T i A^TA . Stupci od U i V se smatraju lijevim i desnim singularnim vektorima. Singularne vrijednosti od A definirane su kao dijagonalni elementi od Σ koji su ne-negativni kvadratni korijeni *n* svojstvenih vrijednosti od A^TA.

Dekompozicija na singularne vrijednosti matrice je jedinstvena do određenog retka, stupca i permutacija predznaka. Na slici 3. prikazana je dekompozicija matrice dimenzija *m x n* (u našem slučaju to je matrica *m* riječi i *n* dokumenata).

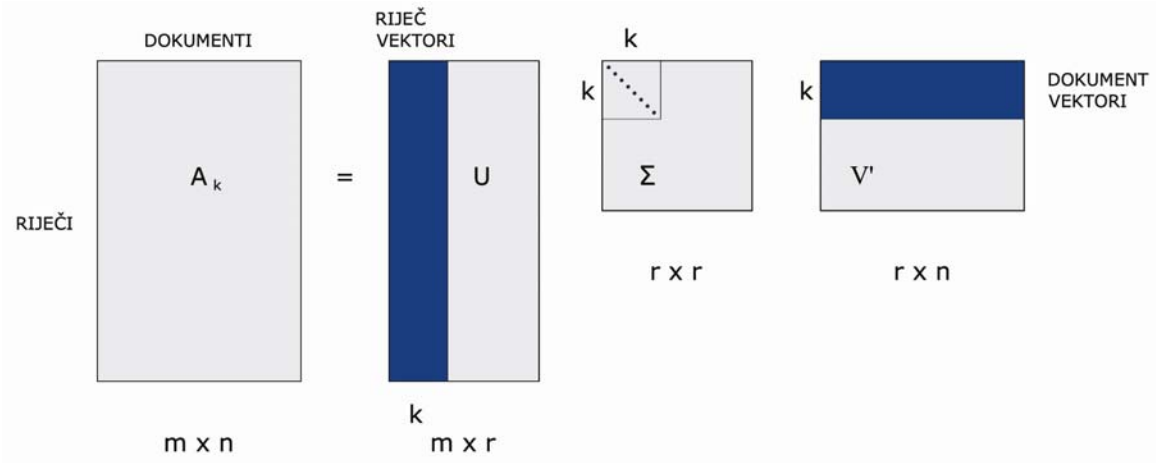


Slika 3. SVD dekompozicija matrice A

Općenito za $A = U\Sigma V^T$ matrice U, Σ i V moraju biti istog ranga. SVD dopušta jednostavnu strategiju optimalne aproksimacije manjim matricama. Kako su singularne vrijednosti matrice Σ sortirane u padajućem poretku možemo zadržati prvih *k* najvećih, a ostale postaviti na nula. Produkt dobivenih matrica je matrica A_k ranga *k* koja je dobra aproksimacija matrice A. Može se pokazati da je nova matrica A_k matrica ranga *k* koja je u smislu najmanjih kvadrata najbliža matrici A. Matricu Σ možemo pojednostavniti brisanjem redaka i stupaca koje smo postavili na nula. Na taj način dobijemo novu matricu Σ_k. Obrišemo li odgovarajuće stupce matrica U i V dobit ćemo nove matrice U_k i V_k. Kao rezultat dobivamo slijedeći model:

$$A \approx A_k = U_k \Sigma_k V_k^T \quad (4.5.)$$

koji je najbolja moguća aproksimacija matrice A ranga *k*.



Slika 4. Aproksimacija matrice A matricom A_k

Slijedeća dva teorema ilustriraju kako SVD može otkriti važne podatke o strukturi matrice.

Teorem 4.1. (Dumais, et al.,1995) Neka je SVD matrice A definiran jednadžbom (4.4.) i neka vrijedi

$$\sigma_1 \geq \sigma_2 \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_n = 0.$$

Neka R(A) i N(A) određuju rang i nul prostor od A respektivno. Onda vrijedi:

- Svojstvo ranga: rang(A) = r, N(A) ≡ span{v_{r+1}, ... , v_n} te R(A) ≡ span{u_1, ... , u_r}, gdje je U = [u_1u_2 ... u_m] i V = [v_1v_2 ... v_n].
- Didaktička dekompozicija: A = $\sum_{i=1}^{r} u_i \cdot \sigma_i \cdot v_i^T$.
- Norme: Frobenius norma $\| A \|_F^2 = \sigma_1^2 + \ldots + \sigma_r^2$ i 2-norma $\| A \|_2^2 = \sigma_1$.

Svojstvo ranga, možda jedno od najvažnijih aspekata SVD-a, omogućava uporabu singularnih vrijednosti od A kao kvantitivnih mjera kvalitativne notacije ranga. Didaktička dekompozicija, koja je mjerilo za redukciju dimenzionalnosti ili kompresiju u mnogim aplikacijama, omogućava kanonsku dekompoziciju matrice kao sume *r* matrica ranga jedan padajuće važnosti.

Teorem 4.2. [Eckart i Young] Neka je SVD od A definiran jednadžbom (4.4.) i neka je r = rang(A) ≤ min(m,n) te definiramo

(4.6.) $$A_k = \sum_{i=1}^{k} u_i \cdot \sigma_i \cdot v_i^T ,$$

tada vrijedi

$$\min_{\text{rang}(B)=k} \| A - B \|_F^2 = \| A - A_k \|_F^2 = \sigma_{k+1}^2 + \ldots + \sigma_p^2 .$$

Drugim riječima, matrica A_k, koja je konstruirana iz *k* najvećih singularnih trojki od A, je matrica ranga *k* najbliža matrici A. A_k je najbolja aproksimacija matrice A za bilo koju unitarno invarijantnu normu. Primijetimo,

(4.7.) $$\min_{\text{rang}(B)=k} \| A - B \|_2 = \| A - A_k \|_2 = \sigma_{k+1}$$

4.3. Izračunavanje SVD-a

Postoji važna veza između dekompozicije singularnih vrijednosti matrice A dekompozicije simetričnih matrica A^TA, AA^T i $\begin{pmatrix} 0 & A^T \\ A & 0 \end{pmatrix}$. Ako je

$$U^TAV = \text{diag}(\sigma_1, \ldots, \sigma_n)$$

SVD matrice $A \in R^{mxn}$ $(m \geq n)$ tada vrijedi

(4.8.) $$V^T(A^TA)V = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2) \in R^{mxn} \quad \text{i}$$

$$U^T(AA^T)U = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2, 0, \ldots 0) \in R^{mxn}.$$

Štoviše, ako je

$$\underset{n}{U_1} \quad \underset{m-n}{U_2}$$

i ako definiramo ortogonalnu matricu $Q \in R^{(m+n)x(m+n)}$ s $Q = \frac{1}{\sqrt{2}} \begin{bmatrix} V & V & 0 \\ U_1 & -U_1 & \sqrt{2}U_2 \end{bmatrix}$

tada vrijedi

(4.9.) $$Q^T \begin{pmatrix} 0 & A^T \\ A & 0 \end{pmatrix} Q = \text{diag}(\sigma_1, \ldots, \sigma_n, -\sigma_1, \ldots, -\sigma_n, \ldots, 0, \ldots 0).$$

Ova povezanost sa simetričnim problemom pronalaženja singularnih vrijednosti nam dopušta da prilagodimo matematičke algoritme pronalaženja singularnih vrijednosti nesimetričnih matrica. (Golub, et al., 1996)

5. Optimizacijske metode za izračunavanje SVD-a

Klasične metode za izračunavanje SVD-a, kao što su Golub-Kahan-Reinscd i Jacobi metode, nisu dobar izbor za rijetke matrice velikih dimenzija. Zbog uporabe ortogonalnih transformacija direktno na rijetku matricu A klasične metode imaju velike memorijske zahtjeve. Nadalje, klasične metode računaju sve singularne vrijednosti matrice A što ih čini računalno rasipnima s obzirom na to da je potrebno izračunati samo podskup skupa singularnih vrijednosti. U ovom poglavlju bit će objašnjene iterativne matematičke metode izračunavanja SVD-a rijetkih matrica velikih dimenzija. Detaljno će biti objašnjen Lanczos algoritam za određivanje najvećih singularnih trojki (singularne vrijednosti i pripadni lijevi i desni singularni vektori) rijetke matrice.

5.1. Kratki pregled algoritama

Kako se povećava dostupnost računala visokih performansi sve više raste interes za razvojem učinkovitih implementacija rješavanja problema pronalaženja singularnih vrijednosti matrica. U području dubinske analize podataka često je potrebno izračunati SVD rijetkih matrica velikih dimenzija. Postoje brojne biblioteke razvijene u tu svrhu kao što su Arpack, Lapack, LSI++, Svdpack, SparseLib++...

Često korištene metode možemo razdvojiti u dvije grupe:

- metode zasnovane na podprostornim iteracijama
- metode zasnovane na Lanczos algoritmu. (Berry, et al., 1993)

Neke od metoda prve grupe su podprostorna iteracija i metoda minimalizacije traga. Razne verzije Lanczos algoritma su simetrična Lanczos metoda, blok Lanczos metoda, nesimetrična Lanczos metoda... Sve navedene metode iterativno pronalaze singularne vrijednosti matrice te ne mijenjaju u originalnu matricu A. To ih čini memorijski i računalno prikladnima za rješavanje ovog problema. Najčešće upotrebljavana metoda je Lanczos metoda koja je

implementirana i u ovom radu. Metoda je detaljno objašnjena u sljedećem poglavlju.

5.2. Simetrična Lanczos metoda

Lanczos algoritam[4] je vrlo popularna metoda izračunavanja singularnih vrijednosti rijetkih matrica velikih dimenzija. Za razliku od klasičnih pristupa metoda ne unosi promjene u originalnu matricu te ne generira potpune međumatrice. Ova metoda generira sekvencu tridijagonalnih matrica T_j sa svojstvom da su najveće singularne vrijednosti *j x j* matrice T_j sve bolje aproksimacije singularnih vrijednosti originalne matrice. Vrijednost najvećih singularnih vrijednosti matrice T_j konvergira pravoj vrijednosti mnogo prije završetka tridijagonalizacije. To svojstvo čini Lanczos algoritam korisnim u slučajevima kada se traži samo nekoliko najvećih vrijednosti.

5.2.1. Svojstva Krylovog podprostora i Krylove matrice

Budući da je Lanczos metoda usko povezana s izračunavanjem ortogonalne baze pripadnog Krylovog podprostora, u ovom ćemo poglavlju razmotriti svojstva Krylovog podprostora i Krylove matrice.

Krylov podprostor i Krylova matrica matrice A povezani s vektorom x definirani su kako slijedi.

Definicija 5.1. (Jiang, 1997) Neka je $A \in R^{nxn}$ i $x \in R^n$. Krylova matrica definirana je jednadžbom

$$K(x, A, m) = [x, Ax, ..., A^{m-1}x] \in R^{nxm} \quad (5.1.)$$

Podprostor razapet stupcima Krylove matrice se zove Krylov podprostor i označava se s K(x, A, m).

Sljedeći teorem utvrđuje postojanje i jedinstvenost tridijagonalizacije Krylove matrice.

[4] Algoritam se temelji na metodi koju je 1950. godine uveo Cornelius Lanczos

Teorem 5.1. (Jiang, 1997) Neka je $A \in R^{nxn}$ simetrična, $x \in R^n$ s $\|x\|_2 = 1$ i dimenzija Krylovog podprostora K(x, A, m) je m tada vrijedi

- Ako je K[x, A, m] = $Q_m R_m$ QR dekompozicija, tada je $Q_m^T A Q_m = T_m \in R^{mxm}$ tridijagonalna matrica takva da vrijedi
$$AQ_m = Q_m T_m + r_m e_m^T,$$
gdje je $r_m \in R^n$, $Q_m^T r_m = 0$ te e_m m-ti stupac matrice I_n.
- Neka je x n-vektor i neka vrijedi $\|x\|_2 = 1$. Ako je $Q_m \in R^{nxm}$ takav da vrijedi $Q_m e_1 = x$, $Q_m^T Q_m = I_m$ i
$$AQ_m = Q_m T_m + r_m e_m^T,$$
gdje je T_m tridijagonalna, tada je
$$K(x, A, m) = Q_m[e_1, T_m e_1, ..., T_m^{m-1} e_1] := Q_m R_m$$
QR dekompozicija od K(x, A, m), gdje je $R_m = [e_1, T_m e_1, ..., T_m^{m-1} e_1]$ gornja trokutasta matrica.

Postojanje ortogonalne transformacije sličnosti pokazano je sljedećim teoremom.

Teorem 5.2. (Jiang, 1997) Neka je $x = q_1$ takav da vrijedi $\|q_1\|_2 = 1$ te rang(K(x, A, m)) = n tada postoji ortogonalna matrica Q s prvim stupcem q_1 takva da je $Q^T A Q = T$ tridijagonalna matrica.

Gore navedeni teoremi povezuju QR dekompoziciju Krylove matrice K(x, A, m) s tridijagonalizacijom matrice A. Ako možemo pronaći QR dekompoziciju Krylove matrice K(x, A, m) onda možemo pronaći transformaciju Q koja transformira matricu A u T. Stoga je izračunavanje singularnih vrijednosti od A ekvivalentno izračunavanju singularnih vrijednosti od T.

5.2.2. Tridijagonalizacija

Neka je A *n x n* matrica

$$A = B^T B$$

gdje je B *m x n* matrica čije singularne vrijednosti tražimo. Lanczos metoda pronalazi *n x n* ortogonalnu matricu Q koja transformira matricu A u tridijagonalnu matricu T (Golub, et al., 1996)

$$Q^T A Q = T.$$

Neka je r_1 slučajno generiran početni vektor dimenzija *n x 1* takav da vrijedi $\|r_1\|_2 = 1$. Za j = 1, 2 … *k* definiramo pripadne Lanczos matrice T_j uporabom iduće rekurzije. Definiramo $\beta_1 \equiv 0$ te $r_0 \equiv 0$. Zatim za *i* = 1, 2, … *l* definiramo Lanczos vektore q_i te skalare α_i i β_{i+1} gdje je

(5.2.)
$$\begin{aligned} \beta_{i+1} q_{i+1} &= A q_i - \alpha_i q_i - \beta_i q_{i-1}, \\ \alpha_i &= q_i^T (A q_i - \beta_i q_{i-1}), \text{ i} \\ |\beta_{i+1}| &= \|A q_i - \alpha_i q_i - \beta_i q_{i-1}\|_2 . \end{aligned}$$

Za svaki *j* pripadna Lanczos matrica T_j se definira kao realna, simetrična, tridijagonalna matrica s dijagonalnim elementima α_i ($1 \le i \le j$) te subdijagonalnim elementima β_{i+1} ($1 \le i \le (j-1)$),

$$T_j \equiv \begin{pmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \beta_j \\ & & & & \beta_j & \alpha_j \end{pmatrix}.$$

Prema definiciji vektori $\alpha_i q_i$ i $\beta_i q_{i-1}$ su ortogonalne projekcije od Aq_i na posljednje vektore q_i i q_{i-1}. Rezultirajući α_i i β_{i+1} definiraju pripadnu Lanczos matricu. Napišemo li jednadžbu (5.2.) u matričnom obliku za svaki *j* dobijemo

(5.3.)
$$AQ_j = Q_j T_j + \beta_{j+1} q_{j+1} e_j^T,$$

gdje je $Q_j \equiv [q_1, q_2, \ldots, q_j]$ *n x j* matrica čiji je *k*-ti stupac *k*-ti Lanczos vektor, a e_j^T je *j*-ti stupac *n x n* jedinične matrice. Lanczos rekurzija (5.2.) generira obitelj realnih, simetričnih, tridijagonalnih matrica povezanih s A i q_1.
Iz rekurzije dane jednadžbom (5.2.) dobijemo sljedeći pseudokod (Jiang, 1997)

$r_0 = q_1;\ \beta_0 = 1;\ q_0 = 1;\ j = 0;$
while $(\beta_j \neq 0)$
$q_{j+1} = r_j / \beta_j;$
$j = j + 1;$
$\alpha_j = q_j^T A q_j;$
$r_j = (A - \alpha_j I) q_j - \beta_{j-1} q_{j-1};$
$\beta_j = \|r_j\|_2;$
end while

Osnovna Lanczos procedura opisana je kroz sljedeća četiri koraka (Berry, et al., 1993):

- Lanczos rekurzijom (5.2.) generirati obitelj realnih, simetričnih, tridijagonalnih matrica T_j (j = 1, 2, …,k).
- Za k ≤ l izračunati bitne singularne vrijednosti od T_k.
- Odabrati neke ili sve singularne vrijednosti kao aproksimacije singularnih vrijednosti od A.
- Za svaku singularnu vrijednost λ izračunati odgovarajući jedinični singularni vektor z takav da vrijedi $T_k z = \lambda z$. Poredati dobivene vektore u odgovarajuće Ritz vektore $y \equiv Q_l z$ koji se upotrebljavaju kao aproksimacije željenih singularnih vektora matrice A.

5.2.3. Zaustavljanje i ograničenje pogreške

Iteracija opisana u prethodnom poglavlju će se zaustaviti prije završetka tridijagonalizacije ako je početni vektor q_1 u prikladnom invarijantnom podprostoru. Sljedeći teorem pokazuje kako stupci matrice $Q = [q_1, q_2, ..., q_j]$ razapinju Krylov podprostor $K(q_1, A, j)$ nakon *j*-tog koraka.

Teorem 5.3. (Jiang, 1997) Neka je $A \in R^{nxn}$ simetrična matrica te pretpostavimo da $q \in R^n$ ima jediničnu 2-normu. Tada Lanczos algoritam iterira do koraka *j* = *m*, gdje je *m* = rang(K[q_1, A, *n*]). Za j = 1, 2, ...,*m* vrijedi

$$AQ_j = Q_jT_j + r_je_j^T,$$

gdje je

$$T_j \equiv \begin{pmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & . & . & . & \\ & & . & . & . & \\ & & & . & . & \beta_j \\ & & & & \beta_j & \alpha_j \end{pmatrix}$$

i Q = [q_1, q_2, ..., q_j] ima ortonormalne stupce koji razapinju K(q_1, A, *j*).

Primijetimo da za r_j = 0 vrijedi $AQ_j = Q_jT_j$ što znači da je podprostor razapet s Q_j invarijantan podprostor od A. Iz teorema (5.3) je očito da je rang(Q_j) = K(q_1, A, n). Primijetimo da vrijedi *j* = *m* = dim(K(q_1, A, j)). Kako β_j = 0 implicira r_j = 0, nailaženje na β_j = 0 u Lanczos iteraciji je poželjan događaj jer signalizira da je pronađen invarijantan podprostor.

Teorem 5.4. (Jiang, 1997) Pretpostavimo da je izvršeno *j* iteracija Lanczos algoritma i da je $S_j^TTS_j$ = diag(θ_1, θ_2, ..., θ_j) Schur dekompozicija tridijagonalne matrice T_j. Ako je Y_j = [y_1, y_2, ..., y_j] = $Q_jS_j \in R^{jxj}$ tada za *i* = 1, 2, ..., *j* vrijedi

$$\left\| Ay_i - \theta_i y_i \right\|_2 = \left| \beta_j \right\| s_{ji} \left|$$

gdje je $S_j = (s_{pq})$.

Koristeći teorem moguće je izračunati ograničenje pogreške singularnih vrijednosti od T_j:

(5.4.) $$\min_{\mu \in \lambda(A)} | \theta_i - \mu | \le | \beta_j | | s_{ji} | . \qquad i = 1, \ldots, j$$

θ_i su Ritz vrijednosti, a y_i Ritz vektori. Uređeni par (θ_i, y_i) je Ritz par podprostora $K(q_1, A, n)$. Vrijednost s_{ij} je *j*-ta komponenta singularnog vektora s_i koja odgovara *i*-toj singularnoj vrijednosti θ_i od T_j.

Nakon *j* koraka Lanczos algoritmom se dobiju matrice Q_j (matrica Lanczos vektora) i T_j (tridijagonalna matrica) te vektor r_j. Zbog jednostavnije analize pretpostavljamo da vrijedi (Jiang, 1997.):

- $T_j = S_j Q_j S_j^T,\ S_j^T = S_j^{-1},\ Q_j = \mathrm{diag}(q_1, \ldots, q_j)$.
- Lokalna ortogonalnost je zadržana odnosno

$$q_{i+1}^T q_i = 0,\ \ i = 1, \ldots, j-1 \ \text{ i } \ r_j^T q_j = 0.$$

6. Latentno semantičko indeksiranje

Većina metoda pretraživanja tekstualnih dokumenata zasniva se na leksičkoj usporedbi riječi postavljenog upita i riječi pridruženih dokumentima. Raznolikost u izboru riječi (sinonimi) koje ljudi upotrebljavaju za opis pojedinog dokumenta čini leksičke metode neadekvatnim i nepreciznim. Metoda *latentno semantičkog indeksiranja* prevladava nedostatke usporedbe izraza promatrajući nepouzdanost promatranih 'veza riječ-dokument' kao statistički problem. Kako bi se otklonio šum u podacima, koriste se statističke metode. Uporabom SVD-a iz velike 'riječ-dokument matrice' konstruira se semantički prostor u kojem su usko povezane riječi i dokumenti postavljeni jedni blizu drugih. Riječi koje se nisu pojavile u dokumentu mogu se u novom prostoru pojaviti blizu dokumenta ako je to konzistentno s najjačim asocijativnim uzorcima u podacima.

6.1. Ilustracija rada

U svrhu implementacije l*atentno semantičkog indeksiranja* potrebno je konstruirati 'riječ-dokument matricu'. Elementi matrice su frekvencije pojavljivanja pojedine riječi u dotičnom dokumentu

$$A = [a_{ij}]$$

gdje je a_{ij} frekvencija pojavljivanja riječi *i* u dokumentu *j*. Kako se svaka riječ ne pojavljuje u svakom dokumentu matrica A je vrlo rijetka. Matrica A se faktorizira na produkt tri matrice (4.4.) uporabom dekompozicije na singularne vrijednosti (SVD). SVD izračunava latentno semantički strukturni model iz ortogonalnih matrica U i V, koje sadrže lijeve i desne singularne vektore matrice A, te matrice Σ, koja sadrži singularne vrijednosti matrice A. Ove matrice predstavljaju rastav originalnih 'riječ-dokument veza' na linearno nezavisne vektore ili faktor vrijednosti. Uporaba *k* faktora ili *k* najvećih singularnih trojki je ekvivalentna aproksimaciji 'riječ-dokument matrice' A matricom A_k prema jednadžbi (4.5.).

Važno svojstvo SVD-a je da izračunata matrica A_k točno ne rekonstruira originalnu 'riječ-dokument matricu' A. SVD zadržava većinu najvažnije skrivene strukture u povezanosti riječi i dokumenata te

istovremeno odbacuje šum u podacima. Izrazi koji se pojavljuju u sličnim dokumentima biti će blizu u *k*-dimenzionalnom prostoru čak i ako se nikad zajedno ne pojavljuju u istom dokumentu.

6.2. Preslikavanje dokumenata iz skupa za testiranje

Nakon izračunavanja SVD-a matrice skupa za učenje potrebno je odrediti prikladno preslikavanje za dokumente iz skupa za testiranje. Kod klasifikacije dokumenata dokumente iz skupa za testiranje možemo promatrati kao korisničke upite. Svaki takav upit predstavljamo kao vektor u *k*-dimenzionalnom prostoru, a zatim ga uspoređujemo s dokumentima iz skupa za učenje. Dokument iz skupa za testiranje predstavljen je vektor *d* čiji su elementi frekvencije riječi u dotičnom dokumentu. Primijetimo da prema (4.4.) za *d* vrijedi

$$d = U_k \Sigma_k d'^T,$$

odnosno

$$d' = d^T U_k \Sigma_k^{-1}, \tag{6.1.}$$

gdje je d' prikaz dokumenta u *k*-dimenzionalnom prostoru (Deerwester, et al., 1990). Sada dokument može biti uspoređen s dokumentima iz skupa za učenje te pravilno klasificiran.

6.3. Dodavanje dokumenata

Skupovi dokumenata stalno se mijenjaju, neki dokumenti se brišu drugi se dodaju. Kod *latentno semantičkog indeksiranja* logičan pristup za prilagodbu promjenama je ponovno izračunavanje SVD-a nove matrice. Za velike matrice proces ponovnog izračunavanja SVD-a je veoma računalno i vremenski zahtjevan postupak zbog čega se češće koriste manje zahtjevni postupci umetanja dokumenata (eng. folding-in) i SVD-osvježavanja (eng. SVD-update). Umetanje dokumenata je računalno nezahtijevna metoda, ali daje nepreciznu reprezentaciju skupa. Metoda SVD-osvježavanja je računalno zahtjevnija od metode umetanja dokumenata, ali čuva reprezentaciju skupa podataka.

Prikazat ćemo dodavanje dokumenata na primjeru. Neka naslovi[5] u tablici 1. predstavljaju skup dokumenata.

Riječi plave boje su riječi koje su se pojavile u nekoliko dokumenata. Ostale riječi ćemo u ovom primjeru eliminirati, kao što je to inače slučaj sa 'stop riječima'.

Tablica 1 . Skup dokumenata

Oznaka	Naslov
D1	*Opća svojstva algebarskih jednadžbi*
D2	*Jednadžba* s realnim koeficijentima
D3	*Derivacija* složene *funkcije* nekoliko *varijabli*
D4	Rješavanje *sustava linearnih jednadžbi*
D5	Razvoj analitičkih *funkcija* u *redove* potencija
D6	*Opći* postupci pri računanju *integrala*
D7	*Funkcije kompleksnih varijabli*
D8	Pretvaranje *algebarskih jednadžbi* u normalni oblik
D9	Osnovna *svojstva* Fourierovih *redova*
D10	*Derivacija vektorske funkcije*
D11	Taylorov *red* za *vektorske funkcije*
D12	*Vektorske jednadžbe*
D13	*Integrali* u *kompleksnom* području
D14	Neodređeni *integrali*
D15	*Opći* slučaj *sustava linearnih jednadžbi*

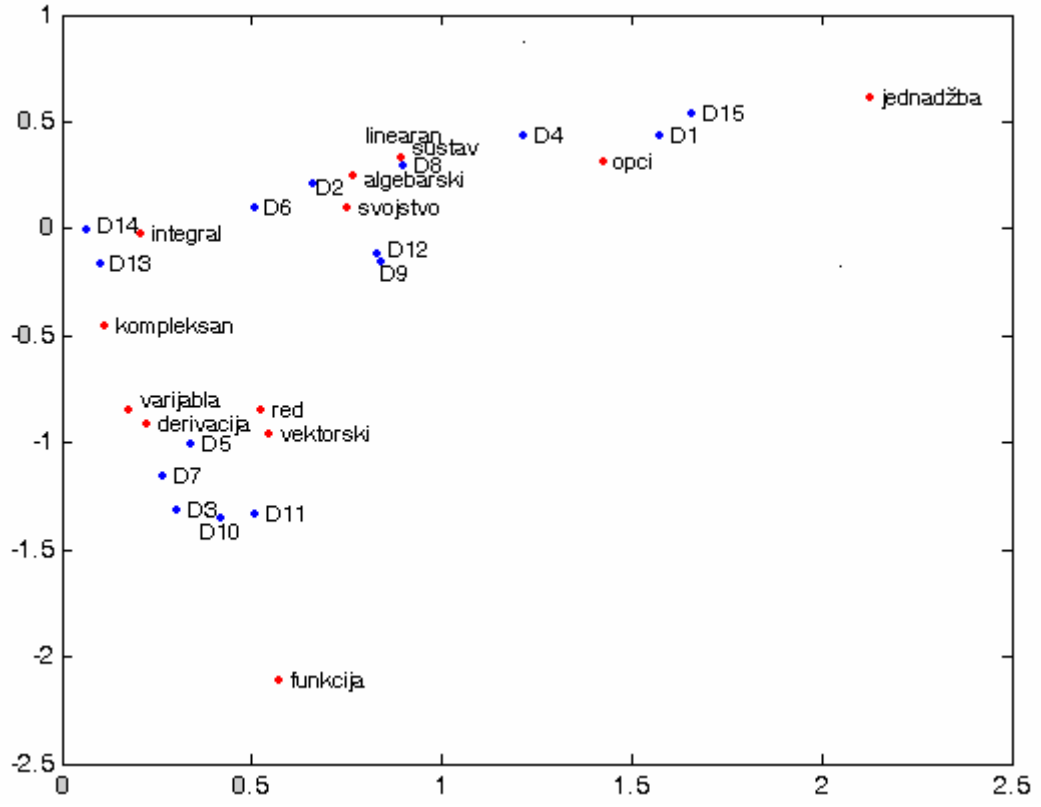
Tablica 2. prikazuje 'riječ-dokument matricu' teksta iz tablice 1. Elementi matrice su frekvencije pojave pojedine riječi u dokumentu.

[5] Naslovi u tablici su podnaslovi uzeti iz I. N. Bronštejn, K. A. Semendjajev, Matematički priručnik za inženjere i studente, Tehnička knjiga, Zagreb, 1964.

Tablica 2. Riječ – dokument matrica

	Dokumenti														
Riječi	*D1*	*D2*	*D3*	*D4*	*D5*	*D6*	*D7*	*D8*	*D9*	*D10*	*D11*	*D12*	*D13*	*D14*	*D15*
algebarski	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
derivacija	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
funkcija	0	0	1	0	1	0	1	0	0	1	1	0	0	0	0
integral	0	0	0	0	0	1	0	0	0	0	0	0	1	1	0
jednadžba	1	1	0	1	0	0	0	1	0	0	0	1	0	0	1
kompleksan	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
linearan	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
opći	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1
red	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0
sustav	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
svojstvo	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
varijabla	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
vektorski	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0

Sada će biti izračunat SVD ove matrice (k = 2). Riječi i dokumenti su na slici 5. prikazani u koordinatnoj ravnini.



Slika 5. Dvodimenzionalni graf riječi i dokumenata

Množenjem prvog stupca matrice U s prvom singularnom vrijednosti σ_1 matrice Σ dobivamo x koordinatu, a množenjem drugog stupca matrice U s drugom svojstvenom vrijednošću σ_2 matrice Σ dobivamo y koordinatu.

Promotrimo li sliku 5. primijetit ćemo da su dokumenti podijeljeni u dvije grupe. Dokumenti sa sadržajem o vektorskim funkcijama i redovima grupirani su bliže y-osi, a oni sa sadržajem o jednadžbama bliže x-osi. Na primjer, možemo zaključiti da dokumenti D_5, D_7, D_3, D_{10} i D_{11} imaju sličan sadržaj.

6.3.1. Ponovno izračunavanje

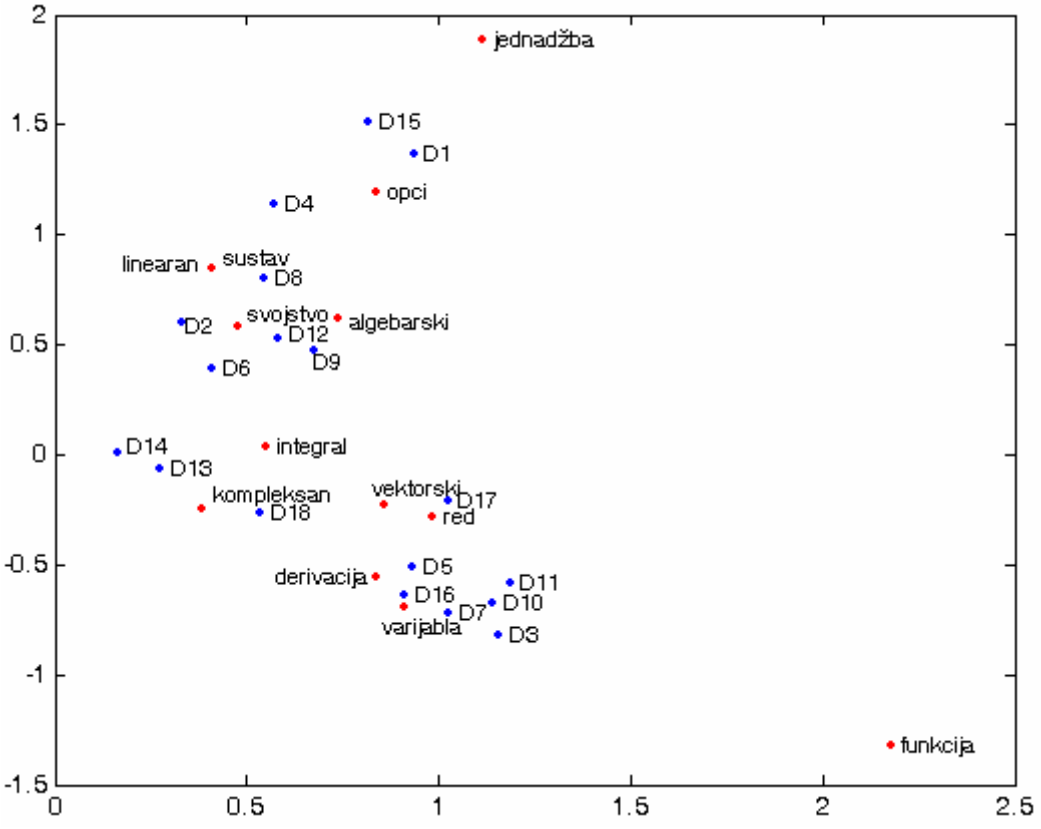
Dodamo li matrici A nove dokumente dobit ćemo matricu A'. Najrobusniji način izračunavanja aproksimacijske matrice A'_k za promijenjenu 'riječ-dokument matricu' je ponovno izračunavanje SVD-a promijenjene matrice. Nažalost to je računalno vrlo zahtjevan pristup, te su za manje promijene mnogo zanimljivije metode umetanja dokumenata i SVD-osvježavanje. Za razliku od metode umetanja dokumenata, koja ne mijenja veze među starim dokumentima, ponovnim izračunavanjem redefinira se latentno semantička struktura cijele matrice.

Prisjetimo se primjera iz prethodnog poglavlja. Zamislimo sada da postojećem skupu od 15 dokumenata želimo dodati tri nova dokumenta iz tablice 3.

Tablica 3. Novi dokumenti

Oznaka	Naslov
D16	*Funkcije* jedne *varijable*
D17	*Integrali algebarskih funkcija*
D18	*Derivacije* i diferencijali višeg *reda*

Sada ćemo matricu prikazanu tablicom 2. proširiti s tri nova stupca koja predstavljaju dokumente D_{16}, D_{17} i D_{18}. Na slici 6. prikazani su dokumenti i riječi u koordinatnoj ravnini nakon izračunavanja SVD-a novonastale matrice.



Slika 6. Dvodimenzionalni graf riječi i dokumenata nove matrice dodavanjem dokumenata metodom ponovnog izračunavanja

Promotrimo li sliku 6. primijetit ćemo da je dodavanjem novih dokumenata došlo do promijene položaja svih dokumenata u koordinatnoj ravnini. Time smo pokazali da se metodom ponovnog izračunavanja redefinira latentno semantička struktura cijele matrice.

6.3.2. Umatenje dokumenata

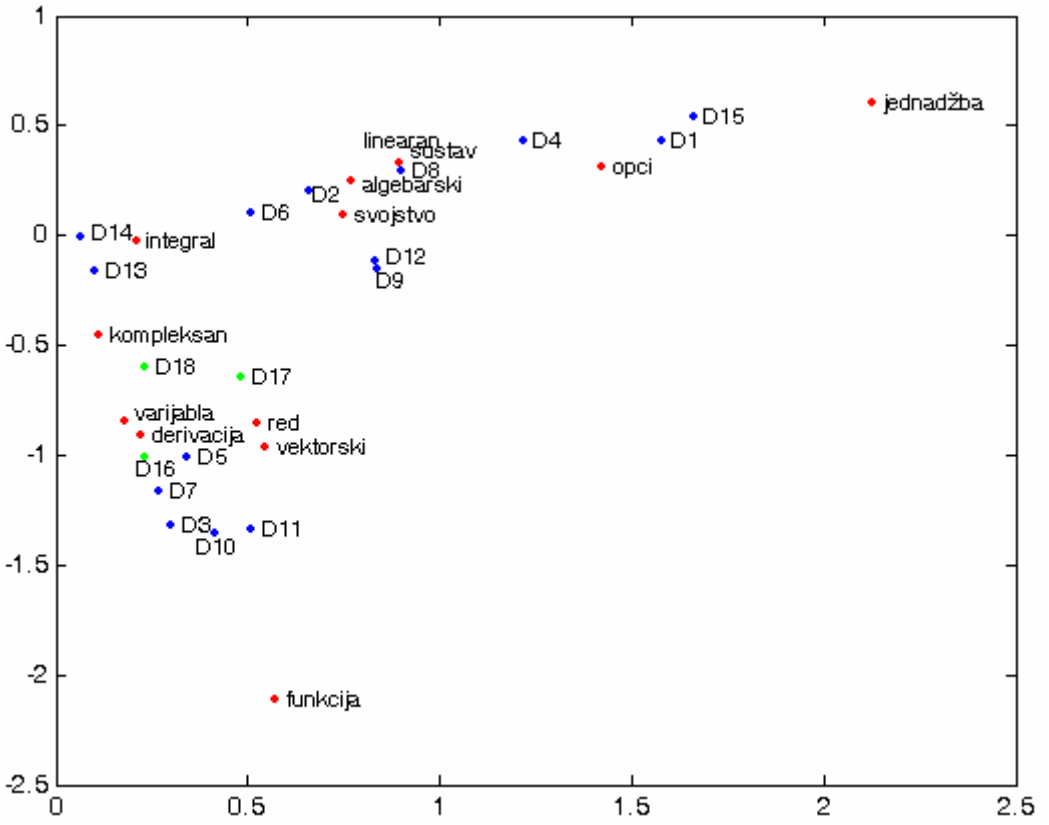
Umetanje novog dokumenta d u postojeću 'riječ-dokument matricu' svodi se na preslikavanje tog dokumenta u prostor U_k pomoću slijedeće jednadžbe

$$d' = U_k U_k^T d. \quad (6.2.)$$

Prema gornjoj jednadžbi koordinate dokumenta d' u bazi U_k dane su elementima vektora $U_k^T d$. *K*-dimenzionalni vektor $U_k^T d$ dodaje se kao novi stupac u *k x d* matricu $\Sigma_k V_k^T$. Kako ne bi računali produkt $\Sigma_k V_k^T$ umetanje

dokumenta je ostvareno dodavanjem vektora $U_k^T d\Sigma_k^{-1}$ kao novi redak matrice V_k. (Deerwester, et al., 1990)

Na slici 7. prikazan je položaj dokumenata u koordinatnoj ravnini nakon dodavanja novih dokumenata metodom umetanja.



Slika 7. Dvodimenzionalni graf riječi i dokumenata nove matrice nastale dodavanjem dokumenata metodom umetanja

Usporedimo li slike 5. i 7. vidimo kako su originalni dokumenti zadržali svoje mjesto u koordinatnoj ravnini, a novi dokumenti su dodani u nepromijenjen prostor. Zaključujemo da novi podaci nemaju nikakav utjecaj na grupiranje postojećih riječi ili dokumenata. To je mana metode umetanja dokumenta u odnosu na metodu ponovnog izračunavanja koja uzima u obzir i promjene unesene u originalni prostor dodavanjem novih dokumenata. Ipak zbog znatno manje računalne i vremenske složenosti metoda umetanja se pokazala kao optimalno rješenje pri preslikavanju dokumenata za testiranje u prostor manje dimenzionalnosti.

6.3.3. SVD-osvježavanje

Metoda SVD-osvježavanja ugrađuje nove podatke u postojeći semantički model (matricu A_k iz jednadžbe 4.3.). Za razliku od metode umetanja dokumenta osigurava veze 'riječ dokument' za nove riječi i dokumente te istovremeno zadržava ortogonalnost. Metoda SVD-osvježavanja umjesto ponovnog izračunavanja SVD-a nove matrice A' iskorištava singularne vrijednosti i singularne vektore originalne 'riječ-dokument matrice' A. Metoda ima tri koraka: osvježavanje riječi, osvježavanje dokumenata te osvježavanje težina riječi.

7. Klasifikacija metodom *k najbližih susjeda*

Metoda *k najbližih susjeda* je često upotrebljavana metoda za klasifikaciju tekstualnih dokumenata. Zbog svoje jednostavnosti i široke primjene ova metoda je odabrana za ispitivanje učinkovitosti redukcije dimenzionalnosti. U ovom poglavlju bit će objašnjen način klasifikacije dokumenata metodom *k najbližih susjeda*.

7.1. Karakteristike metode *k najbližih susjeda*

Metoda *k najbližih susjeda* spada u grupu metoda s odgodom. Metode s odgodom ili «lijene» metode ne određuju općenitu funkciju cilja na temelju koje se vrši klasifikacija. Za svaki novi dokument koji se klasificira određuje se lokalna funkcija cilja za dio prostora u kojem se nalazi taj dokument.

Pretpostavka metode *k najbližih susjeda* je ta da dokumenti koji se nalaze blizu u Euklidskom prostoru pripadaju istoj kategoriji. Stoga će dokument biti svrstan u kategoriju kojoj pripada većina njegovih *k* najbližih susjeda.

Nedostatak ovog algoritma je velika količina računalnog vremena potrebnog za klasifikaciju. Nadalje, udaljenost među dokumentima se određuje na temelju svih atributa (riječi) iako su samo neki bitni za klasifikaciju. Na taj način se prilikom klasifikacije uzima u obzir mnogo nevažnih atributa što znači da slični dokumenti mogu biti udaljeni u Euklidskom prostoru.

7.2. Opis algoritma

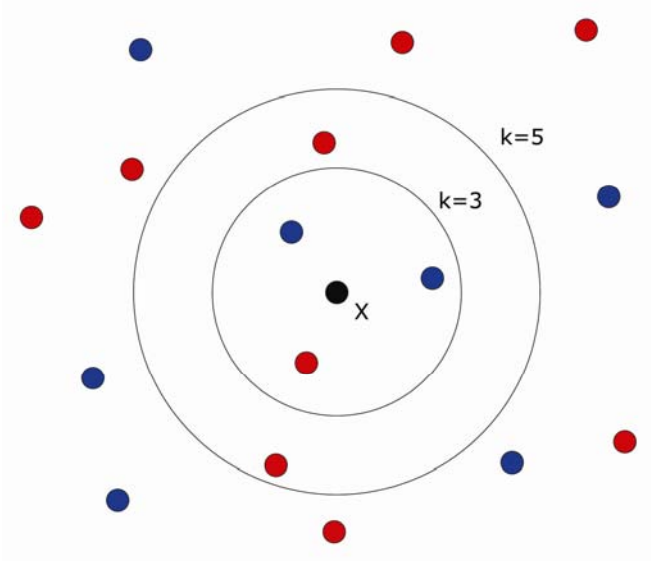
Pretpostavimo da se svaki dokument iz skupova za učenje i testiranje može prikazati u *n*-dimenzionalnom Euklidskom prostoru. Svaki dokument *i* definiran je vektorom

$$\left(r_{1i}, r_{2i}, \ldots, r_{ni}\right),$$

gdje je r_{ji} frekvencija pojavljivanja riječi *j* u dokumentu *i*. Euklidska udaljenost između dva dokumenta definirana je kao

$$d(x,y) \equiv \sqrt{\sum_{i=1}^{n} (r_{ix} - r_{iy})^2} ,$$

gdje je *n* ukupan broj različitih riječi u skupu svih uzoraka. Potrebno je odrediti broj *k* koji označava koliko najbližih susjeda razmatramo za klasifikaciju dokumenta. Dokument će biti dodijeljen onoj kategoriji kojoj pripada većina od k *najbližih susjeda.*



Slika 8. Prikaz rada metode *5 i 3 najbližih susjeda* u dvodimenzionalnom prostoru

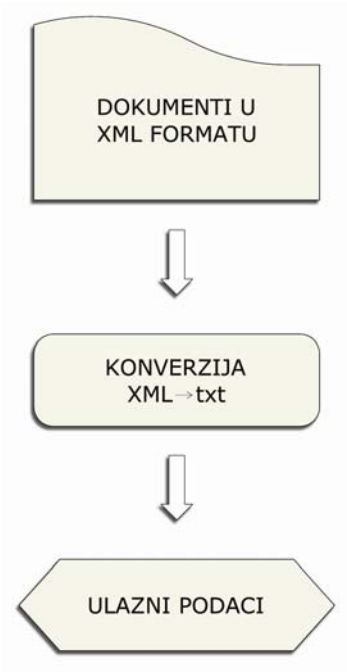
Na slici 8. prikazana je klasifikacija dokumenta *x* metodom *pet i tri najbližih susjeda* u dvodimenzionalnom prostoru. Dokument može pripasti kategoriji “crveni” ili “plavi”. Kako među *pet najbližih susjeda* većina pripada grupi “crvenih” dokument *x* bit će klasificiran kao “crveni”. Odabir vrijednosti *k* bitan je za klasifikaciju dokumenta. Primijetimo da će dokument *x* biti klasificiran kao “plavi” ako ga klasificiramo metodom *tri najbliža susjeda*.

8. Implementacija

U ovom poglavlju bit će iznesena korisnička i programska dokumentacija sustava. Bit će opisan format ulaznih podataka, korisničko sučelje te imeplementacija sustava.

8.1. Ulazni podaci

Početni podaci se nalaze u različitim tipovima XML formata te ih je potrebno konvertirati u jedinstveni format koji predstavlja ulaz u sustav. Skup dokumenata podijeljen je na skupove za učenje i testiranje koji predstavljaju ulaz u sustav opisan u ovom poglavlju. Na slici 9. je prikazan proces pripreme ulaznih podataka.



Slika 9. Priprema ulaznih podataka

Slika 10. predstavlja jedan tip XML formata korištenog kod baze novinskih članaka *Vjesnik*. Takav ili sličan format konvertira se u jednoznačno definiran format objašnjen u poglavlju 8.3.1.

```
<DOC TYPE = "article" FILE = "vj20000112ck03">
       <HEAD TYPE = "na">
              suborac
              globus
              tražiti
       </HEAD>
       <B>
              zagreb
              siječanj
       </B>
       <P>
              djelatnik
              voditi
              grupa
              dogoditi
              pogiblja
       </P>
       <BYLINE>
              vanja
              majetić
               </BYLINE>
</DOC>
```

Slika 10. Zapis dokumenta u XML formatu za bazu *Vjesnik*

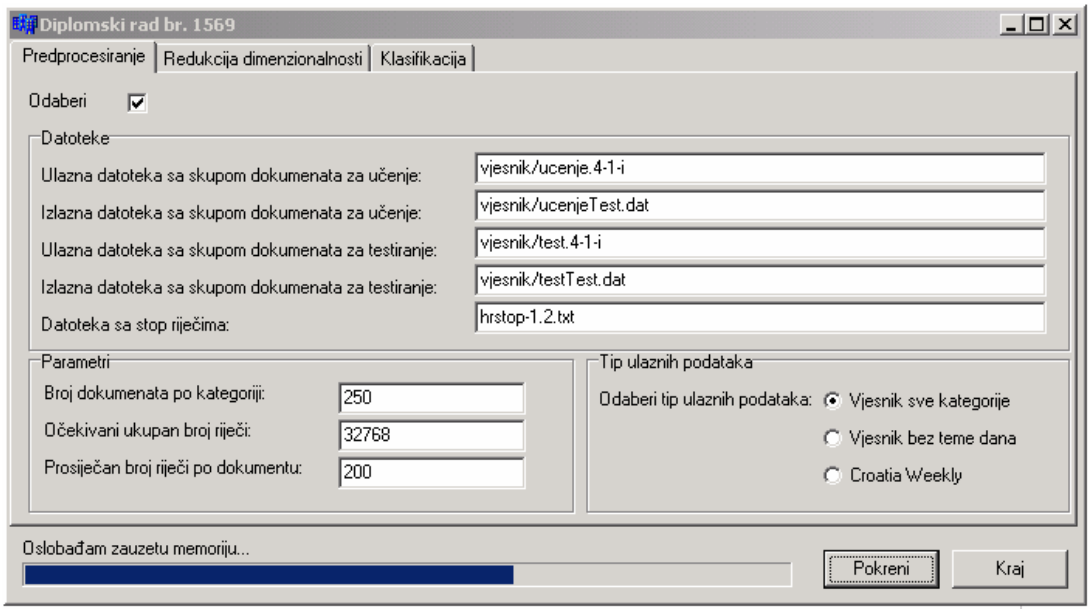
8.2. Grafičko korisničko sučelje

U ovom poglavlju objasnit ćemo način pokretanja programa. Na slikama 11., 12. i 13. prikazano je grafičko korisničko sučelje za svaki od tri dijela implementiranog sustava. Vidimo da je podijeljeno u tri nezavisna dijela: predprocesiranje, redukcija dimenzionalnosti i klasifikacija. Prije pokretanja programa potrebno je označiti koje od navedna tri međusobno nezavisna dijela želimo pokrenuti. Sada ćemo opisati opcije za svaki od tri dijela.

8.2.1. Predprocesiranje

Slika 11. prikazuje grafičko korisničko sučelje dijela za predprocesiranje podataka. U odgovarajuća polja potrebno je upisati naziv datoteka koje sadrže dokumente iz skupa za učenje i testiranje, datoteke koja sadrži 'stop riječi' te datoteka u koje želimo zapisati izlazne podatke. U za to predviđeno polje potrebno je upisati broj dokumenata po kategoriji koje želimo učitati. Ostala dva polja odnose se na optimalno zauzimanje memorije

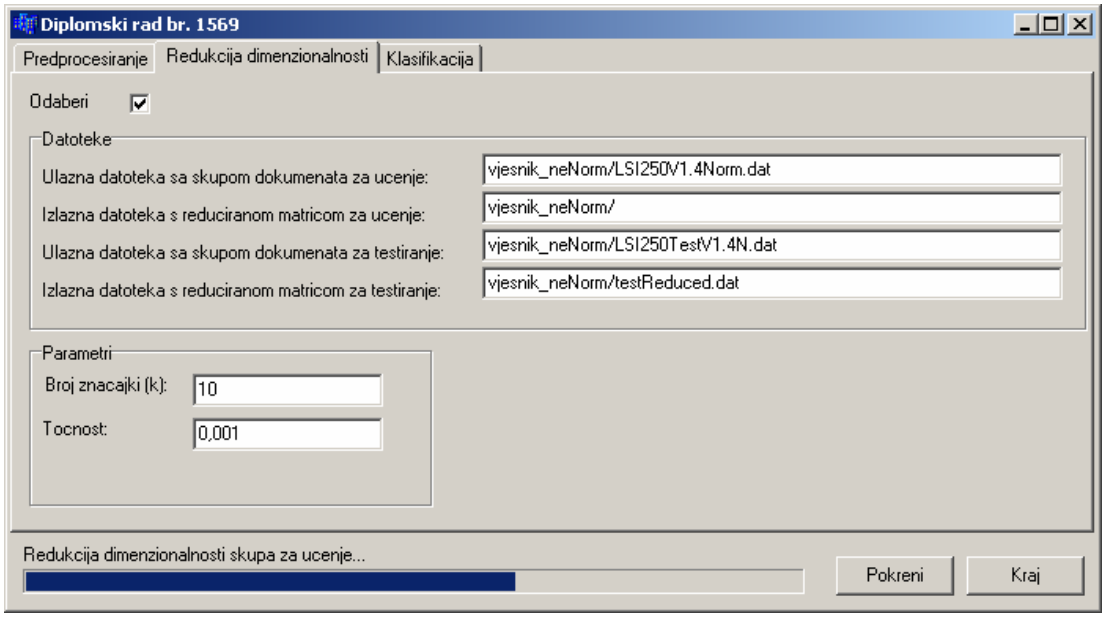
i u njih je potrebno upisati očekivane vrijednosti ukupnog broja riječi u cijelom setu te broja riječi po dokumentu.



Slika 11. Grafičko korisničko sučelje za predprocesiranje

8.2.2. Redukcija dimenzionalnosti

Slika 12. prikazuje grafičko korisničko sučelje dijela za redukciju dimenzionalnosti.

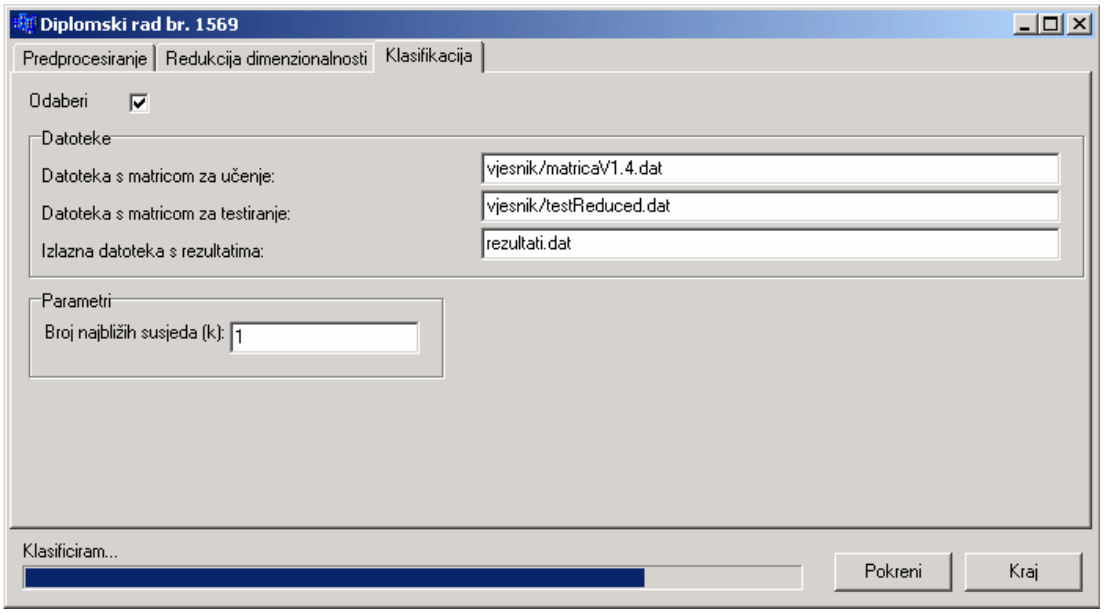


Slika 12. Grafičko korisničko sučelje za redukciju dimenzionalnosti

U odgovarajuća polja potrebno je upisati naziv datoteka koje sadrže matrice s dokumentima iz skupa za učenje i testiranje te datoteka u koje želimo zapisati izlazne podatke. Potrebno je upisati koliko svojstvenih vrijednosti želimo izračunati odnosno broj značajki koje želimo dobiti redukcijom dimenzionalnosti. U za to predviđeno polje potrebno je upisati željenu točnost rezultata. Kako se radi o aproksimaciji singularnih vrijednosti matrice izlaz uvelike ovisi o ovom parametru koji podešavamo eksperimentalno. U ovom radu korištena je vrijednost 0.001.

8.2.3. Klasifikacija

Slika 13. prikazuje grafičko korisničko sučelje dijela za klasifikaciju. U odgovarajuća polja potrebno je upisati naziv datoteka koje sadrže dokumente iz skupa za učenje i testiranje te datoteke u koju želimo zapisati rezultat klasifikacije. Potrebno je učitati koliko najbližih susjeda želimo razmatrati prilikom klasifikacije dokumenta.

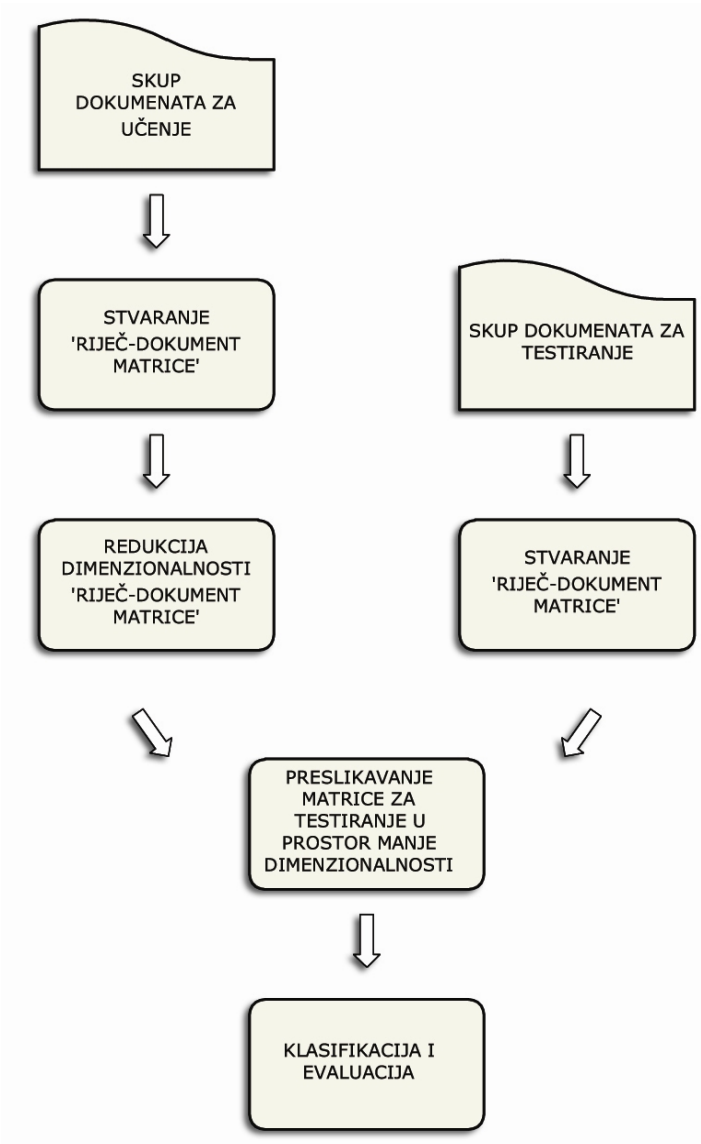


Slika 13. Grafičko korisničko sučelje za klasifikaciju i evaluaciju

8.3. Opis sustava

Implementirani sustav za redukciju dimenzionalnosti i klasifikaciju sastoji se od četiri dijela koji redom obavljaju funkcije učitavanja i formatiranja

ulaznih podataka, redukcije dimenzionalnosti skupa za učenje, preslikavanja dokumenata iz skupa za testiranje u novi prostor te klasifikacije i evaluacije. Sustav je prikazan slikom 14.

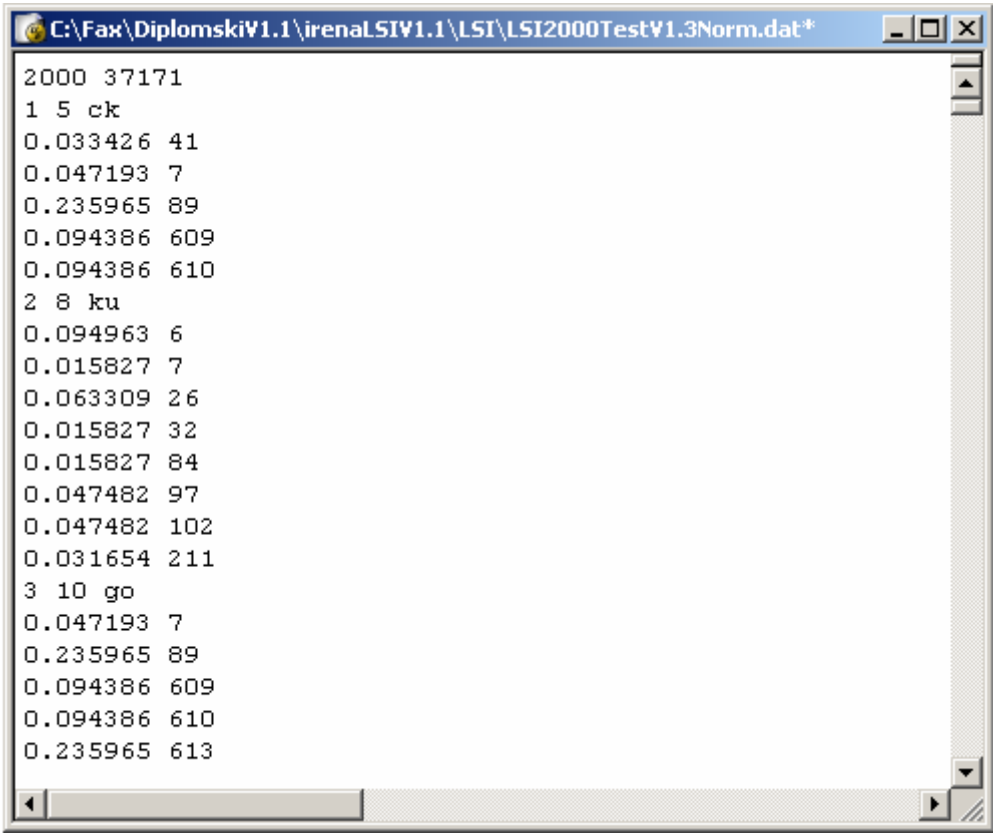


Slika 14. Sustav

8.3.1. Stvaranje 'riječ-dokument matrice'

U prvom koraku se iz ulaznih podataka stvaraju strukture s potrebnim podacima. Nakon eliminacije 'stop riječi', za ostale se riječi računa njihova frekvencija pojavljivanja u svakom dokumentu iz skupa. Izlaz ovog dijela je datoteka u kojoj je zapisana 'matrica riječ-dokument' skupa za učenje odnosno testiranje. Matrica je u datoteci zapisana kao lista dokumenata u

kojoj su svakom dokumentu pridružene sve riječi, odnosno njihovi indeksi, koje su u njemu sadržane te za svaku riječ pripadna frekvencija. Primjer datoteke prikazan je na slici 15.



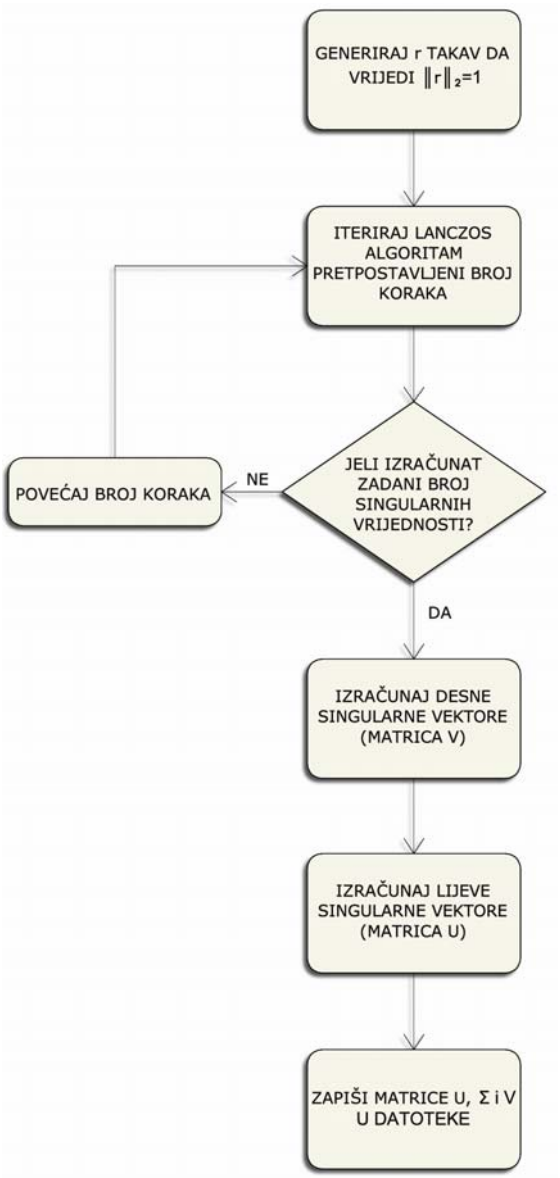
Slika 15. Zapis 'riječ-dokument matrice'

Na početku datoteke zapisan je broj dokumenata i ukupan broj riječi u skupu. Dokumenti su zapisani slijedno jedan iza drugog. Zapis svakog dokumenta sastoji se od njegovog rednog broja, broja riječi koje se pojavljuju u dotičnom dokumentu te kategorije kojoj pripada. Za svaki dokument zapisani su indeksi riječi koje su se u dokumentu pojavile te frekvencija pojavljivanja dotične riječi u tom dokumentu.

8.3.2. Redukcija dimenzionalnosti skupa za učenje

U ovoj fazi se vrši redukcija dimenzionalnosti dokumenata iz skupa za učenje metodom l*atentno Semantičkog Indeksiranja* koja je detaljno opisana u poglavlju 6. Dekompozicija na svojstvene vrijednosti ostvarena je *Lanczos*

algoritmom opisanim u poglavlju 5. Izlaz iz ovog dijela su tri datoteke u kojima je zapisana po jedna od tri matrice koje se dobiju dekompozicijom originalne matrice metodom l*atentno semantičkog indeksiranja*.

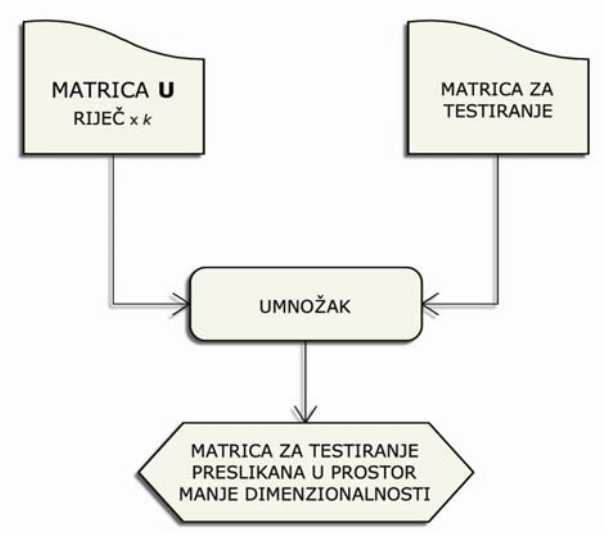


Slika 16. Redukcija dimenzionalnosti skupa za učenje

Na slici 16. je prikazan postupak redukcije dimenzionalnosti dokumenata iz skupa za učenje.

8.3.3. Preslikavanje dokumenata iz skupa za testiranje u novi *k* - dimenzionalni prostor

Koristeći matrice dobivene u prethodnom koraku metodom umetanja dokumenta, opisanom u poglavlju 6., vrši se preslikavanje dokumenata iz skupa za testiranje u novi *k*-dimenzionalni prostor. Preslikavanje se vrši *metodom umetanja* koja se pokazala kao dovoljno dobro rješenje s obzirom na vremensku uštedu u odnosu na *metodu ponovnog izračunavanja*. Izlaz je datoteka u kojoj je zapisana reducirana 'riječ-dokument matrica' skupa za testiranje.



Slika 17. Preslikavanje dokumenata iz skupa za testiranje u prostor manje dimenzionalnosti

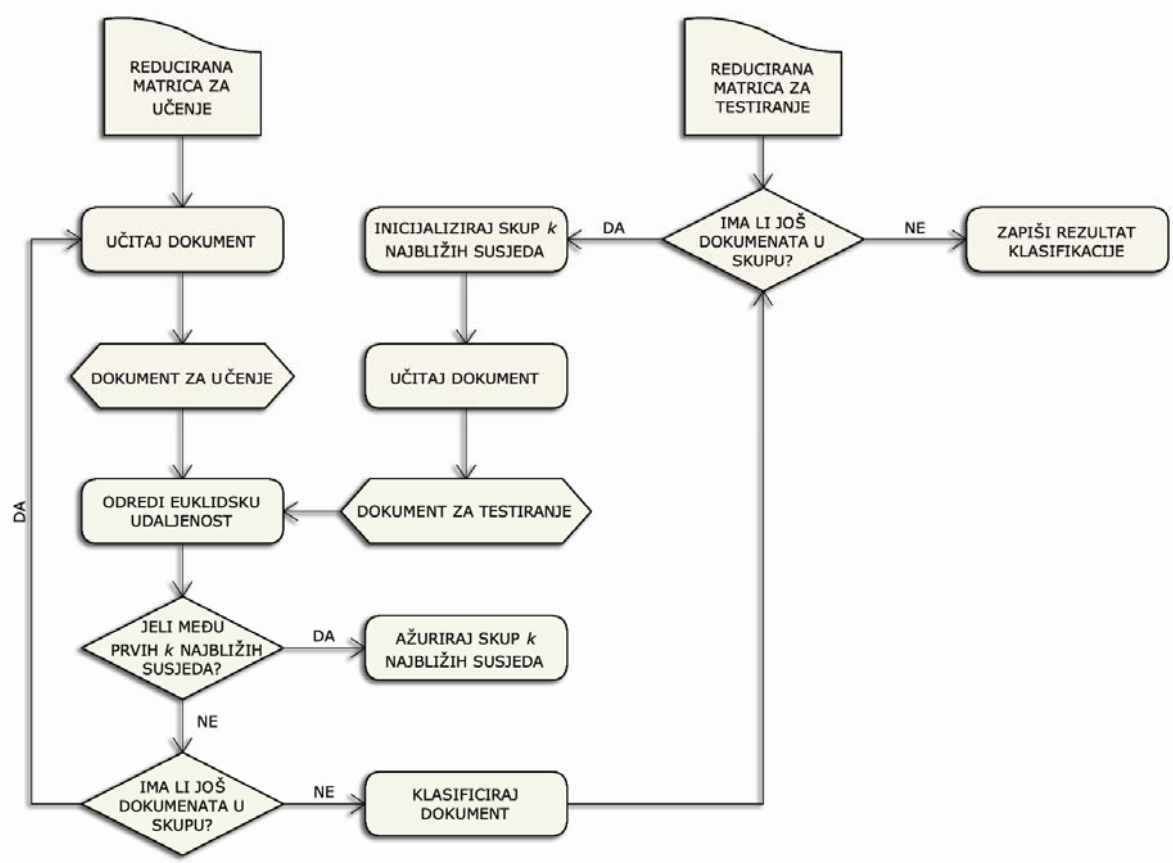
Na slici 17. je prikazan postupak preslikavanja dokumenata iz skupa za testiranje u prostor manje dimenzionalnosti.

8.3.4. Klasifikacija i evaluacija

Posljednja faza je klasifikacija dokumenata iz skupa za testiranje projiciranih u prostor manje dimenzionalnosti. Dokumenti se klasificiraju metodom *k najbližih susjeda*, a pritom se koristi matrica dobivena u trećem koraku.

Na temelju dobivenih rezultata određuje se uspješnost postupka u obliku preciznosti, odziva i F1 mjere uz mikro ili makro usrednjavanje.

Slika 18. prikazuje postupak klasifikacije dokumenata metodom *k-najbližih susjeda*.



Slika 18. Postupak klasifikacije metodom *k-najbližih susjeda*

9. Eksperimenti i rezultati

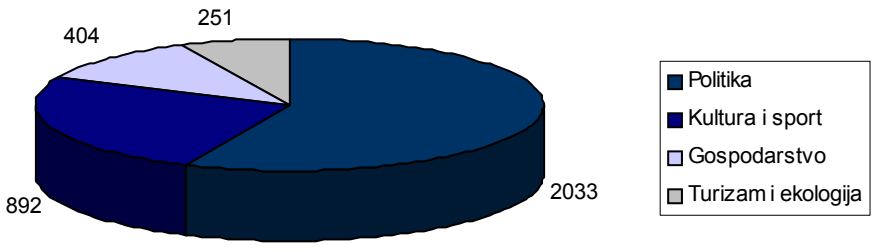
Rad sustava opisanog u prethodnom poglavlju ovisi o odabiru parametara. U ovom poglavlju dat ćemo pregled rezultata ispitivanja sustava na bazama novinskih članaka *Croatia Weekly* i *Vjesnik*.

9.1. Paralelni hrvatsko-engleski korpus *Croatia Weekly*

Baza članaka *Croatia Weekly* je paralelni hrvatsko-engleski korpus članaka. Članci su preuzeti iz časopisa *Croatia Weekly*, brojevi 5.-118. izdani u razdoblju od 1998. do 2000. Članci su raspoređeni u četiri kategorije:

1. po - politika
2. go - gospodarstvo
3. te - turizam i ekologija
4. ks - kultura i sport

Korpus se sastoji od 3580 članaka na hrvatskom jeziku i isto toliko članaka na engleskom jeziku. Skup za učenje i skup za testiranje svaki sadrže 50% dokumenta svake kategorije. Iz članaka na hrvatskom jeziku eliminirane su 'stop riječi', ali članci nisu normalizirani. Problem ove baze je nejednako raspoređen broj dokumenata po kategorijama.



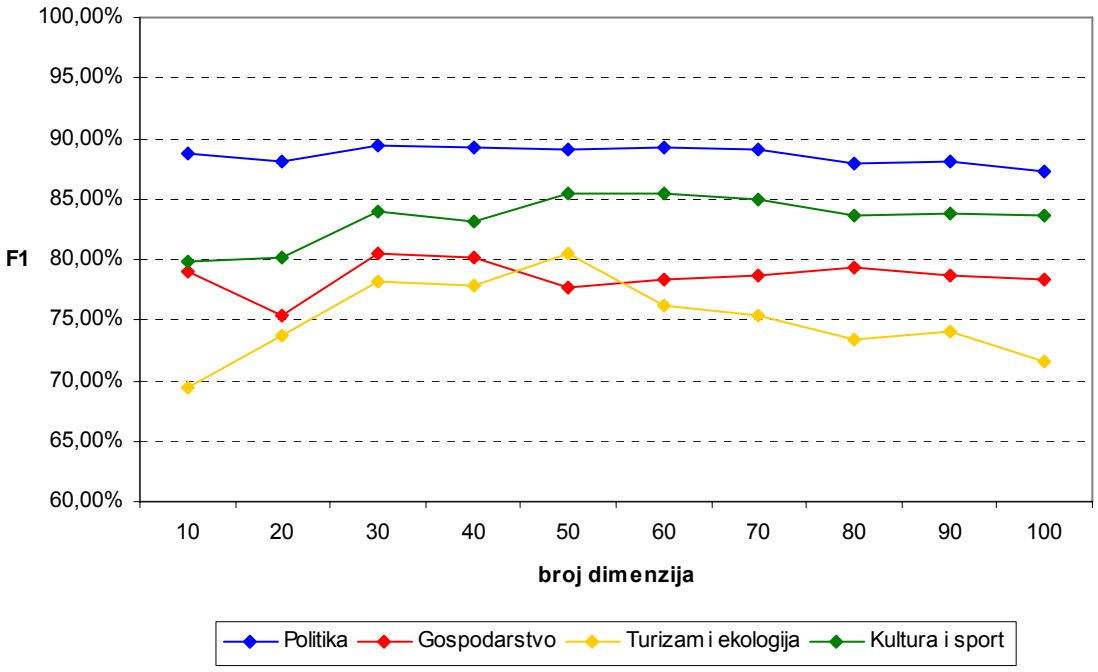
Slika 19. Broj dokumenata po kategorijama za hrvatsko-engleski korpus.

Obrađene članke ustupio je prof. dr. sc. Marko Tadić sa Zavoda za lingvistiku Filozofskog fakulteta Sveučilišta u Zagrebu.

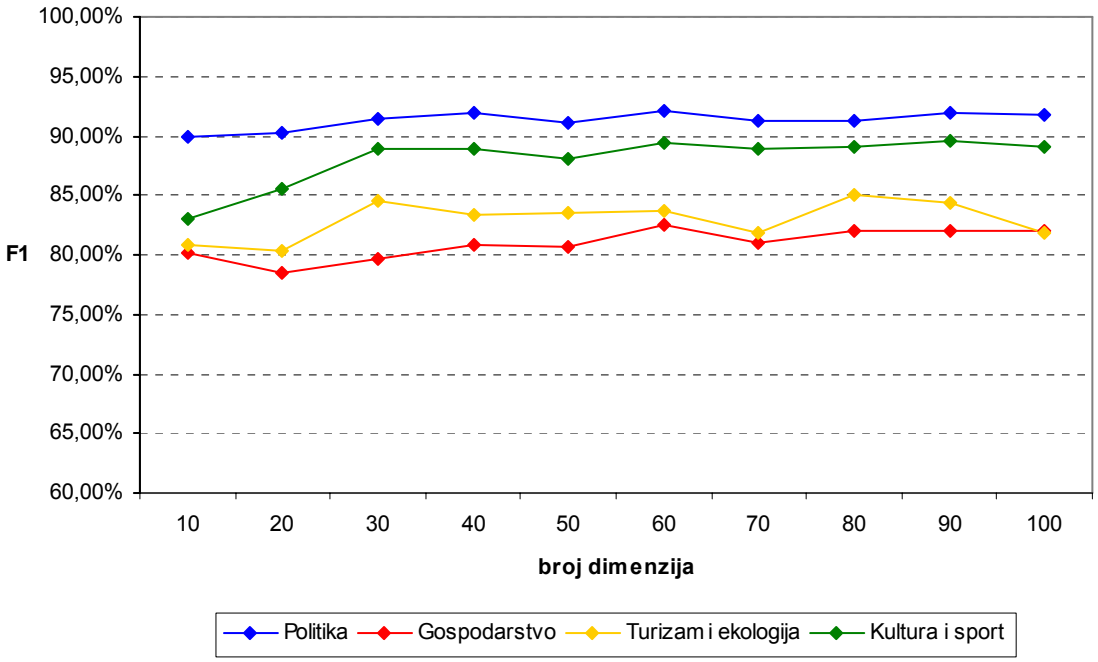
9.1.1. Ovisnost uspješnosti klasifikacije o jeziku i broju dimenzija

U ovom eksperimentu ispitana je razlika uspješnosti klasifikacije za bazu članaka na hrvatskom i engleskom jeziku u ovisnosti o broju dimenzija. Pri redukciji dimenzionalnosti korištena je *LSI* metoda, a klasifikacije je provedena metodom *k* najbližih susjeda uz *k* =1.

Engleski jezik je, za razliku od hrvatskog, morfološki vrlo siromašan. Pri radu s tekstovima na engleskom jeziku ne postoji potreba za morfološkom obradom. Morfološko bogatstvo jezika može imati velik utjecaj na rezultat klasifikacije. Korištenjem *latentno semantičkog indeksiranja* znatno je smanjen broj pojavnica te je za očekivati da morfološko bogatstvo jezika neće imati velik utjecaj na uspješnost klasifikacije. Promotrimo li slike 20., 21. i 22. vidimo da su neznatno bolji rezultati za sve kategorije dobiveni za engleski jezik. Utjecaj morfološke normalizacije na rezultat klasifikacije bit će detaljnije ispitan na bazi novinskih članaka *Vjesnik*.



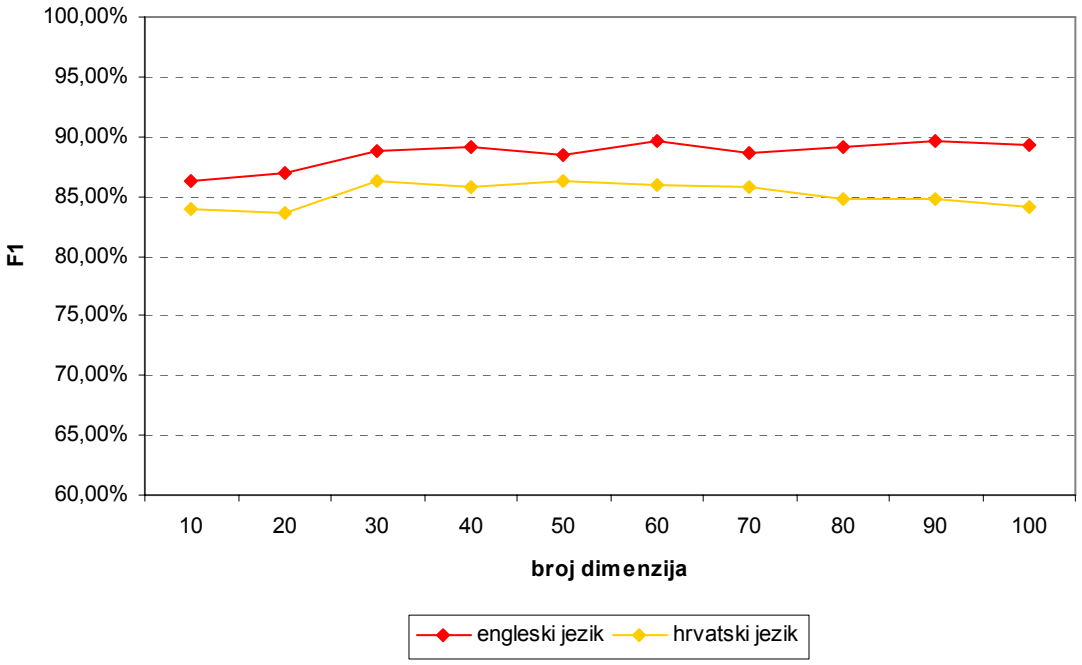
Slika 20. Mjera F1 po kategorijama za hrvatski korpus



Slika 21. Mjera F1 po kategorijama za engleski korpus

Na slici 20. prikazana je vrijednost mjere F1 po kategorijama za hrvatski jezik, a na slici 21. za engleski jezik. Ispitivanje je provedeno u ovisnosti o broju dimenzija. Promotrimo li slike možemo primijetiti da povećanjem broja dimenzija ne dolazi do bitnih promjena u rezultatu klasifikacije. Vrijednost mjere F1 najveća je za otprilike 40 do 60 dimenzija. Povećanjem broja dimenzija iznad tih vrijednosti ne dobivamo bolje rezultate iz čega možemo zaključiti da je većina bitnih informacija sadržana u prvih pedesetak dimenzija, a sve ostalo predstavlja šum. Za očekivati je da bi se rezultat klasifikacije pogoršao za prevelik broj dimenzija koje predstavljaju šum.

Promotrimo li slike 20. i 21. primijetit ćemo da oba jezika daju najbolje rezultate za kategoriju *politika*, nešto lošije za kategoriju *kultura i sport*, a najlošije za kategorije *gospodarstvo* i *turizam i ekologija*. Pogledamo li ponovo skup za učenje primjećujemo da smo takav rezultat mogli predvidjeti s obzirom na to da kategorije *politika* i *kultura* imaju znatno veći broj primjera za učenje.



Slika 22. Mikrousrednjena mjera F1 za hrvatski i engleski

Na slici 22. prikazana je vrijednost mikrousrednjene mjere F1 za bazu na hrvatskom i engleskom jeziku. Primijetimo da su rezultati za engleski jezik otprilike 5% bolji od onih za hrvatski.

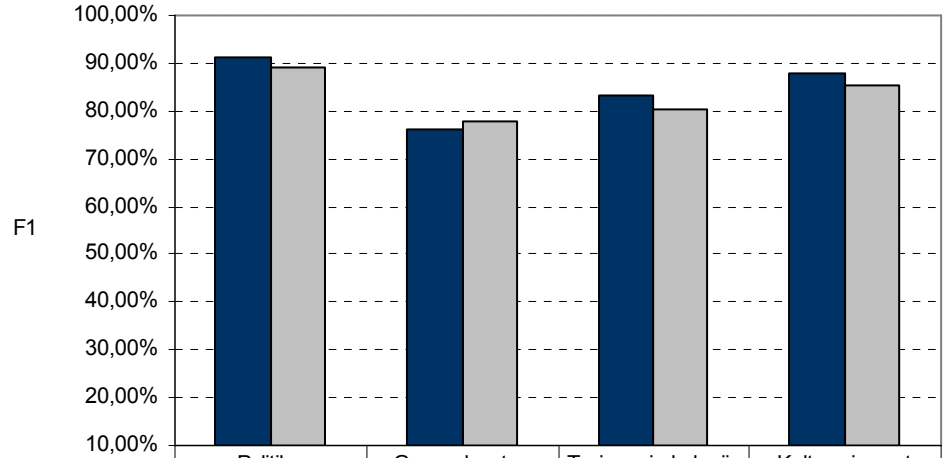
9.1.2. Usporedba uspješnosti klasifikacije s i bez redukcije dimenzionalnosti

Redukcija dimenzionalnosti je bitan korak u klasifikaciji teksta zbog velikih vremenskih i memorijskih zahtijeva pri rukovanju nereduciranom 'matricom riječ-dokument'. Cilj je dovoljno dobro poopćiti originalnu matricu manjim brojem dimenzija (atributa) kako bi zadržali dobre rezultate klasifikacije. Posljedica manje dimenzionalnosti podataka je znatna ušteda vremena potrebnog za klasifikaciju dokumenata iz skupa za testiranje. Memorijski zahtjevi eksponencijalno se povećavaju s porastom broja dokumenata te je redukcija dimenzionalnosti nužno potrebna za velike 'riječ-dokument matrice' čak i kada su rezultati lošiji od onih dobivenih bez redukcije dimenzionalnosti.

U ovom eksperimentu pokazat ćemo razliku u rezultatima dobivenim klasifikacijom s i bez redukcije dimenzionalnosti. Ispitivanje je provedeno za matrice na hrvatskom i engleskom jeziku uz k = 1 najbližih susjeda. Pri redukciji dimenzionalnosti matrica korištena je LSI metoda, a broj atributa reduciran je na 50. Detaljnije o programu korištenom za klasifikaciju dokumenata bez redukcije dimenzionalnosti pogledati u (Tominac, 2004).

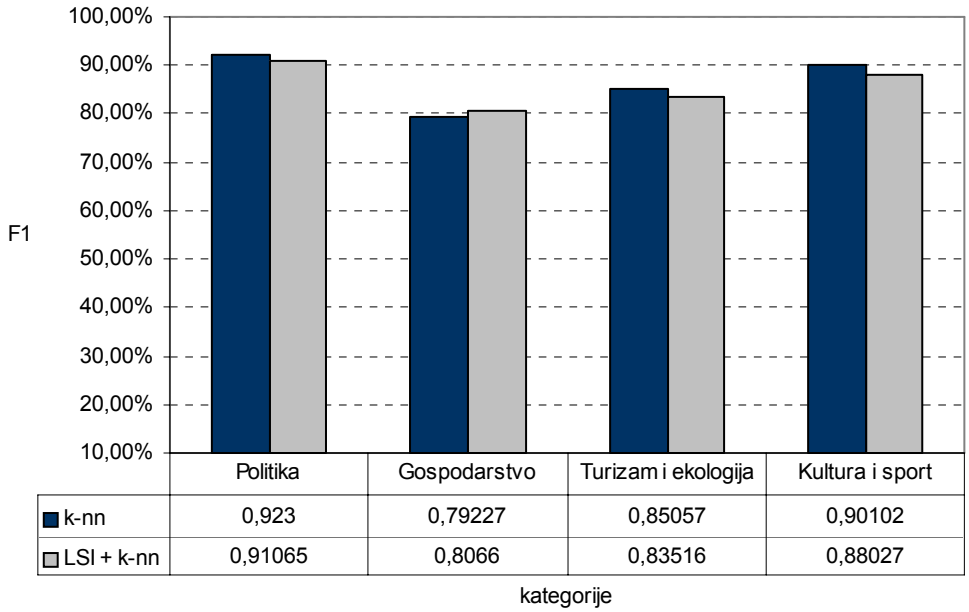
9.1.2.1. Uspješnost klasifikacije po kategorijama

Na slikama 23. i 24. prikazana je uspješnost klasifikacije po kategorijama za hrvatski i engleski jezik s i bez redukcije dimenzionalnosti. Primijetimo da klasifikacija s redukcijom dimenzionalnosti daje približno jednake rezultate kao i bez redukcije dimenzionalnosti za oba jezika. Zaključujemo da smo s drastično manje značajki uspjeli dovoljno dobro poopćiti originalne podatke i zadržali dobre rezultate klasifikacije.



	Politika	Gospodarstvo	Turizam i ekologija	Kultura i sport
k - nn	0,91081	0,76275	0,83137	0,87659
LSI + k-nn	0,8909	0,77698	0,80435	0,85523

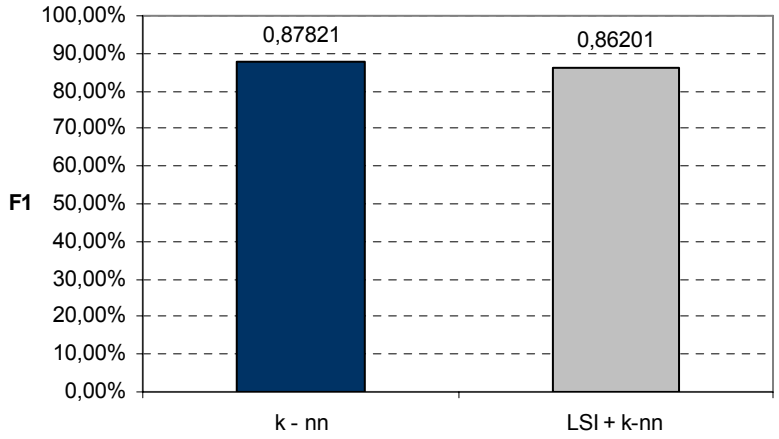
Slika 23. Mjera F1 po kategorijama za klasifikaciju s i bez redukcije dimenzionalnosti za hrvatski jezik



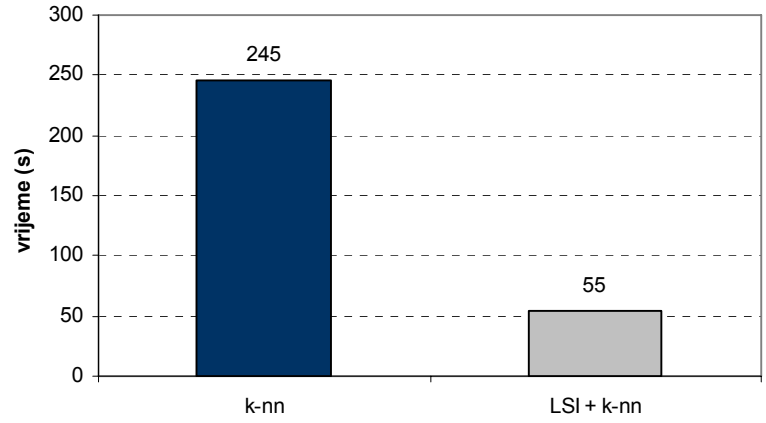
Slika 24. Mjera F1 po kategorijama za klasifikaciju s i bez redukcije dimenzionalnosti za engleski jezik

9.1.2.2. Usporedba uspješnosti klasifikacije i utrošenog vremena

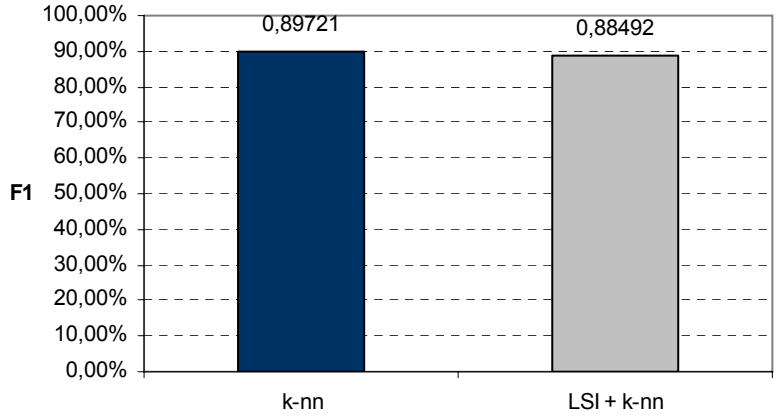
Svrha redukcije dimenzionalnosti teksta pri klasifikaciji je smanjiti vremenske i memorijske zahtjeve. Sada ćemo ispitati koliko smo smanjili vrijeme potrebno za klasifikaciju u odnosu na uspješnost klasifikacije.



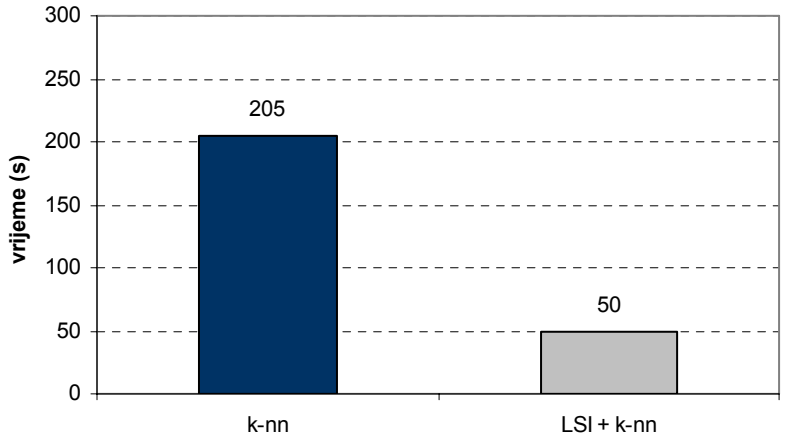
Slika 25. Mikrousrednjena mjera F1 za klasifikaciju s i bez redukcije dimenzionalnosti za hrvatski jezik



Slika 26. Vrijeme potrebno za klasifikaciju s i bez redukcije dimenzionalnosti za hrvatski jezik



Slika 27. Mikrousrednjena mjera F1 za klasifikaciju s i bez redukcije dimenzionalnosti za engleski jezik

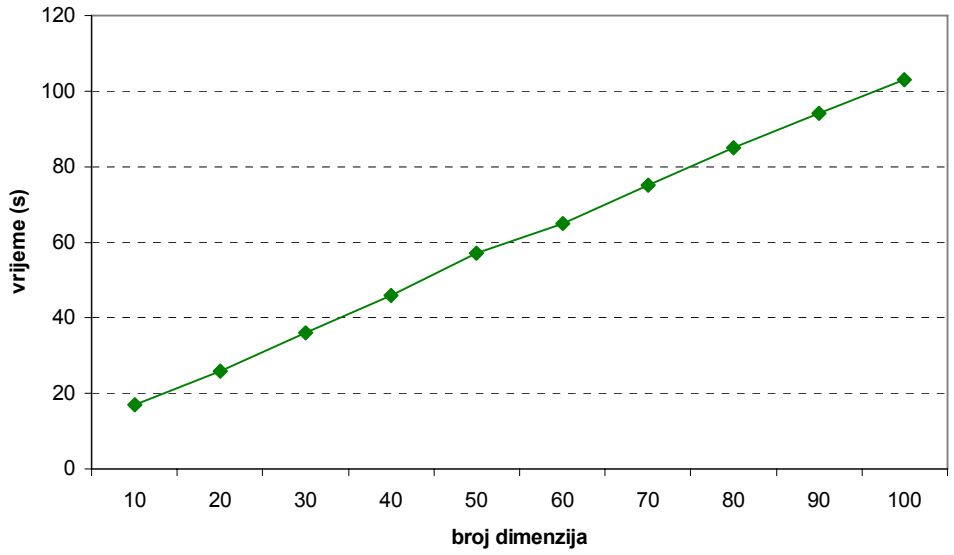


Slika 28. Vrijeme potrebno za klasifikaciju s i bez redukcije dimenzionalnosti za engleski jezik

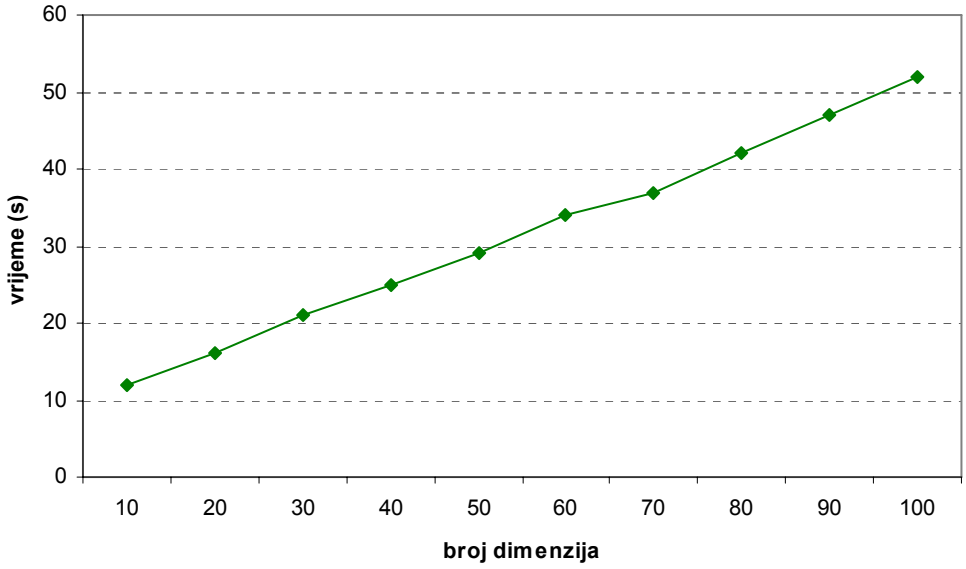
Na slikama 25., 26., 27. i 28. prikazani su rezultati i utrošeno vrijeme klasifikacije s i bez redukcije dimenzionalnosti za bazu na hrvatskom i engleskom jeziku. Vidimo da je uspješnost klasifikacije s i bez redukcije dimenzionalnosti približno jednaka. Vrijeme potrebno za klasifikaciju s redukcijom dimenzionalnosti drastično je manje od vremena potrebnog za klasifikaciju bez redukcije dimenzionalnosti. Takav rezultat je upravo ono čemu smo se nadali, naime smanjili smo vrijeme potrebno za klasifikaciju, a uspješnost je i dalje ostala ista.

9.1.3. Vrijeme učenja klasifikatora

Bitan zahtjev pri implementaciji sustava bio je smanjiti vremensku i računalnu složenost. Bilo je potrebno implementirati sustav tako da se zauzima što manje memorijskog prostora. Kao što je objašnjeno u poglavlju 6. pri redukciji dimenzionalnosti korištena je Lanczos metoda koja singularne vrijednosti matrice računa iterativno te u usporedbi s drugim metodama ima jako male memorijske zahtjeve.



Slika 29. Vrijeme potrebno za klasifikaciju baze na hrvatskom jeziku u ovisnosti o broju dimenzija



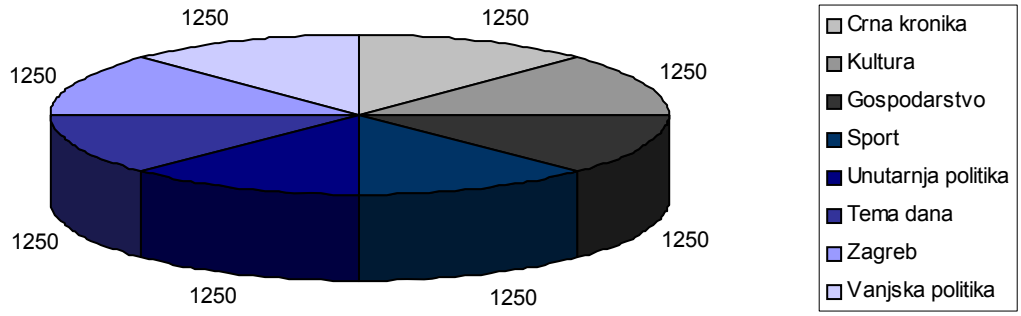
Slika 30. Vrijeme potrebno za redukciju dimenzionalnosti skupa za učenje za bazu na hrvatskom jeziku u ovisnosti o broju dimenzija

Vrijeme potrebno za redukciju dimenzionalnosti i klasifikaciju prikazano je na korpusu na hrvatskom jeziku. Vrijeme u ovisnosti o broju dimenzija prikazano je na slikama 29. i 30.. Kako za ovaj eksperiment nije bitna struktura podataka on neće biti ponovljen na bazi *Vjesnik*.

9.2. Baza novinskih članaka *Vjesnik*

Baza se sastoji od 10 000 novinskih članaka (dnevni list *Vjesnik*, 2000. - 2003.), preuzetih iz hrvatskog nacionalnog korpusa [http://www.hnk.ffzg.hr/]. Članci su podijeljeni u osam kategorija (svaka kategorija sastoji se od 1250 dokumenata) :

1. ck - crna kronika
2. go - gospodarstvo
3. ku - kultura
4. sp - sport
5. td - tema dana
6. un - unutarnja politika
7. vp - vanjska politika
8. zg – Zagreb.



Slika 31. Broj dokumenata po kategoriji za Vjesnikovu bazu članaka

U ovom radu nisu korišteni svi članci već samo 250 članaka po kategoriji jer je pokazano da se korištenjem svih 10000 dokumenata ne dobivaju bolji rezultati klasifikacije.

Novinske članke je ustupio prof. dr. sc. Marko Tadić sa Zavoda za lingvistiku Filozofskog fakulteta Sveučilišta u Zagrebu.

9.2.1. Automatska morfološka normalizacija

Eksperimenti nad bazom Vjesnik provodili su se s normaliziranom inačicom baze verzija 4.1. Normalizirana baza stvorena je iz originalne baze postupkom automatske morfološke normalizacije (AMN) kojeg je razvio dipl. ing. Jan Šnajder sa Zavoda za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva u Zagrebu.

Automatska morfološka normalizacija je postupak kojim se pojavnice u tekstu svode na svoje morfološke norme. Osim što na taj način smanjujemo dimenzionalnost ulaznih podataka, povećavamo kvalitetu teksta koji želimo kategorizirati jer uklanjamo rasipanje značenja istog pojma na više leksički različitih oblika.

Korištene su tri vrste morfološke normalizacije:

- *Flektivna*

  Korištena je oznaka "-i" pri označavanju pripadnih baza (eng. Inflective). Eliminira efekt flektivne norme na način da sve oblike neke riječi svodi na jedan oblik, lemu ako je nju moguće automatski odrediti.

- *Derivacijska*

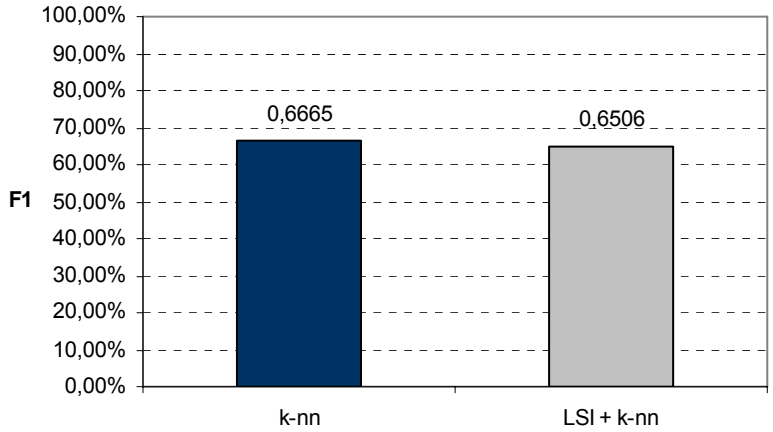
  Korištena je oznaka "-id" pri označavanju pripadnih baza. Nakon svođenja različitih oblika riječi na njihovu lemu, nalazi zajedničkog predstavnika više lema prema derivacijskim pravilima morfologije. Derivacijska pravila morfologije opisuju tvorbu riječi (iz jedne leme u drugu). Smisao derivacijskih normi jest ostvarivanje još veće redukcije dimenzionalnosti ulaznih podataka od one koja se postiže primjenom inflekcijskih pravila.

- *Terminirajuć*a

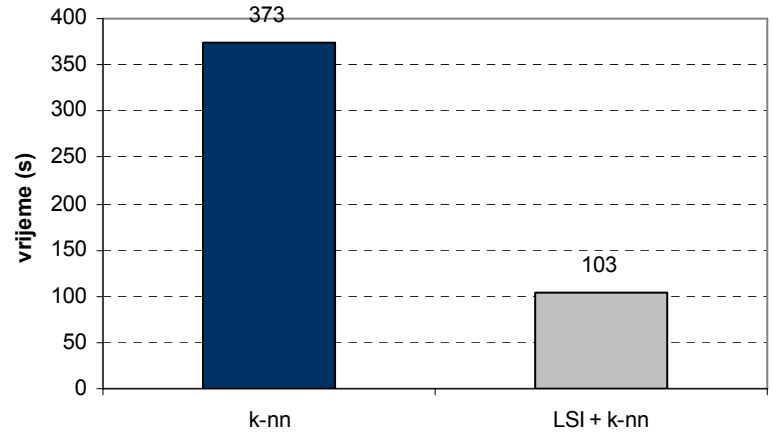
  Korištena je oznaka "-idt" pri označavanju pripadnih baza. U slučajevima kad ne djeluju derivacijska pravila onda se reže riječ nakon sedmog slova.

9.2.2. Usporedba uspješnosti klasifikacije s i bez redukcije dimenzionalnosti

Ovim eksperimentom je ispitana uspješnost i vrijeme potrebno za klasifikaciju s i bez redukcije dimenzionalnosti. Eksperiment je proveden na morfološki nenormaliziranoj bazi članaka *Vjesnik* metodom k = 1 najbližih susjeda. Dimenzionalnost 'matrice riječ-dokument' je reducirana na 50 dimenzija metodom *LSI*. Slike 32. i 33. prikazuju rezultate eksperimenta.



Slika 32. Mikrousrednjena mjera F1 za klasifikaciju s i bez redukcije dimenzionalnosti



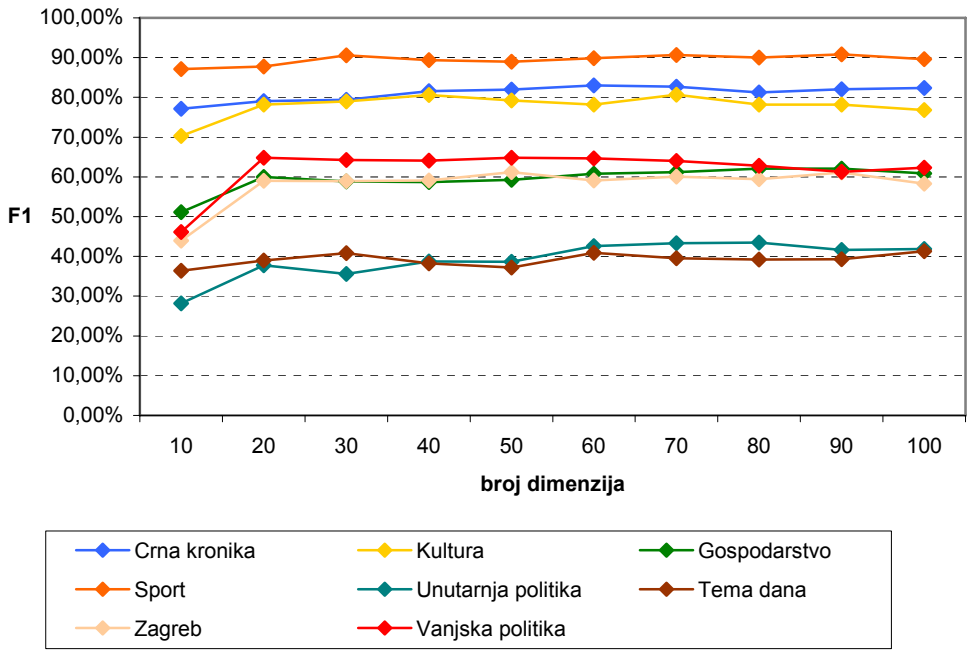
Slika 33. Vrijeme potrebno za klasifikaciju s i bez redukcije dimenzionalnosti

Primijetimo da je uspješnost klasifikacije s i bez redukcije dimenzionalnosti približno jednaka dok je vrijeme potrebno za klasifikaciju s redukcijom dimenzionalnosti znatno manje od vremena potrebnog za klasifikaciju bez redukcije dimenzionalnosti.

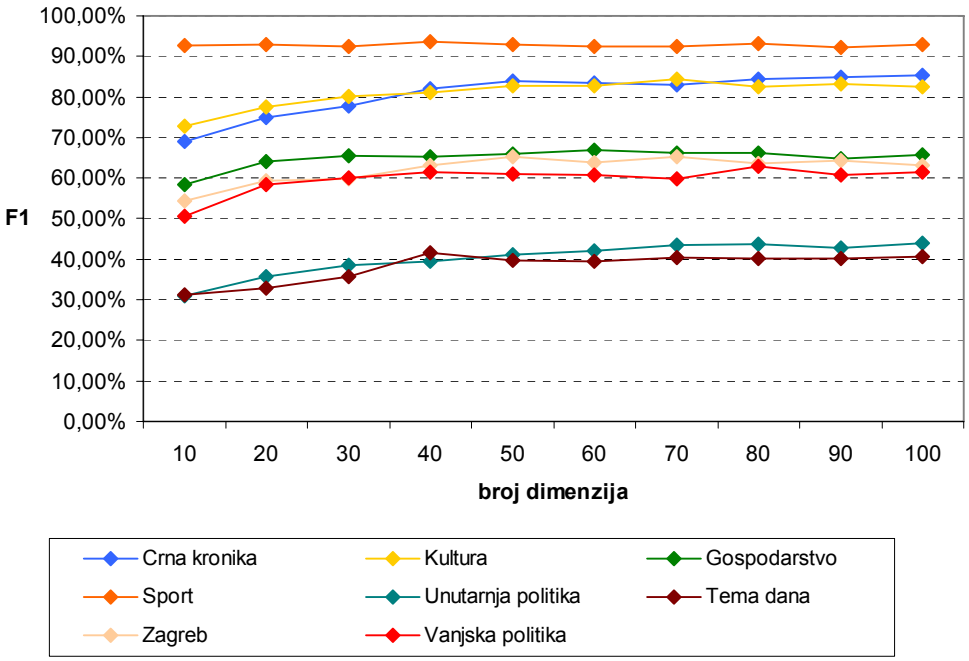
9.2.3. Ovisnost uspješnosti klasifikacije o morfološkom prikazu teksta

Prisjetimo se rezultata klasifikacije za hrvatski i engleski jezik za bazu članaka *Croatia Weekly* iz prethodnog poglavlja. Zaključili smo da je uspješnost klasifikacije nešto bolja za engleski nego za hrvatski jezik zbog toga što je engleski morfološki siromašniji jezik od hrvatskog. Zaključili smo i to da bi razlika u uspješnosti klasifikacije bila mnogo veća za ova dva jezika kada ne bi bila provedena redukcija dimenzionalnosti metodom *latentno semantičkog indeksiranja*. Sada ćemo ispitati utjecaj različitih načina morfološke normalizacije na uspješnost klasifikacije.

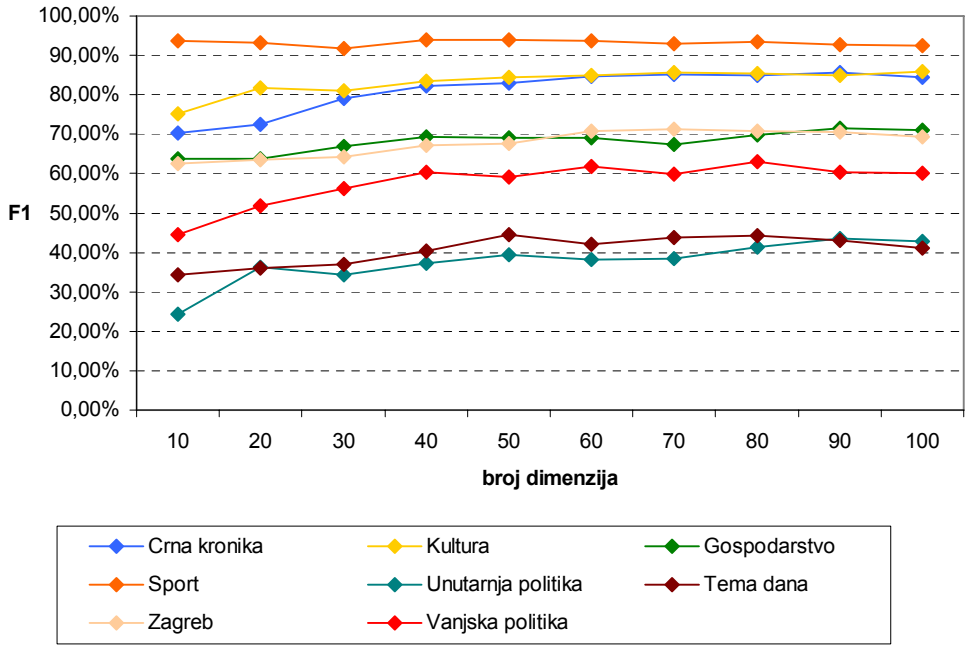
Eksperiment je proveden na bazi članaka Vjesnik uz k = 1 najbližih susjeda. Dimenzionalnost je reducirana metodom *LSI*. Ispitivanje je provedeno u ovisnosti o broju dimenzija. Na slikama 34., 35., 36., 37. i 38. prikazana je redom uspješnost klasifikacije po kategorijama za nenormaliziranu te inflekcijski, derivacijski i terminirajuće normaliziranu bazu *Vjesnik*.



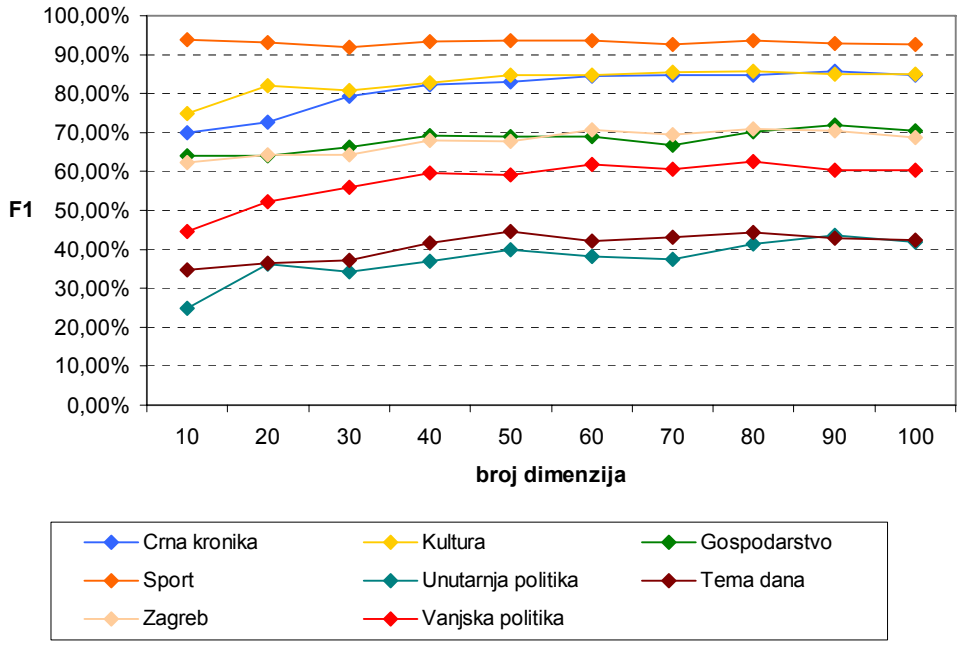
Slika 34. Mjera F1 po kategorijama za nenormaliziranu bazu *Vjesnik*



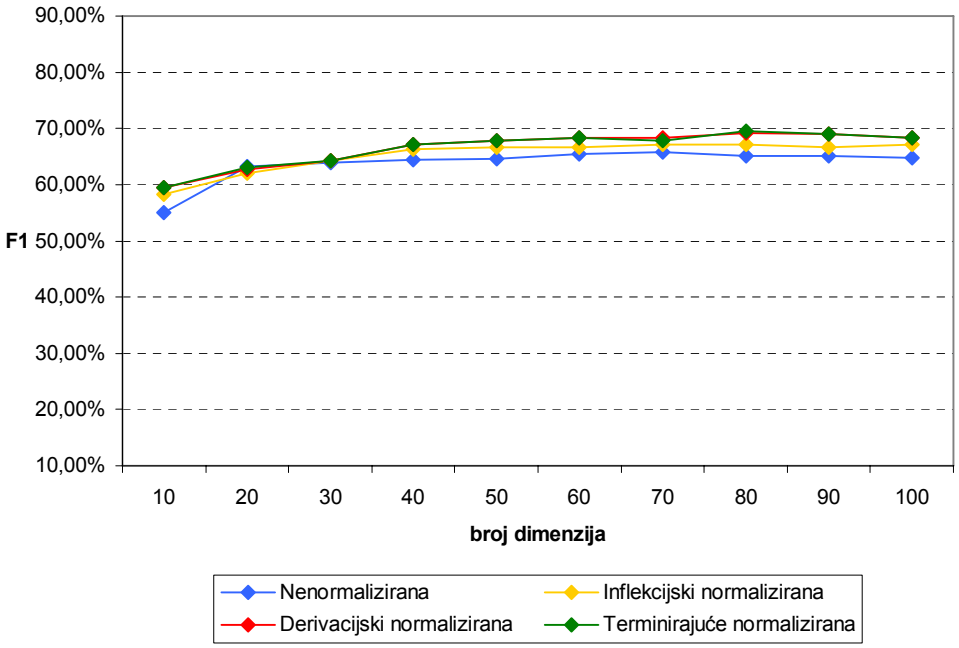
Slika 35. Mjera F1 po kategorijama za inflekcijski normaliziranu bazu *Vjesnik*



Slika 36. Mjera F1 po kategorijama za derivacijski normaliziranu bazu *Vjesnik*



Slika 37. Mjera F1 po kategorijama za terminirajući normaliziranu bazu *Vjesnik*



Slika 38. Mikrousrednjena mjera F1 za različite vrste morfološke normalizacije

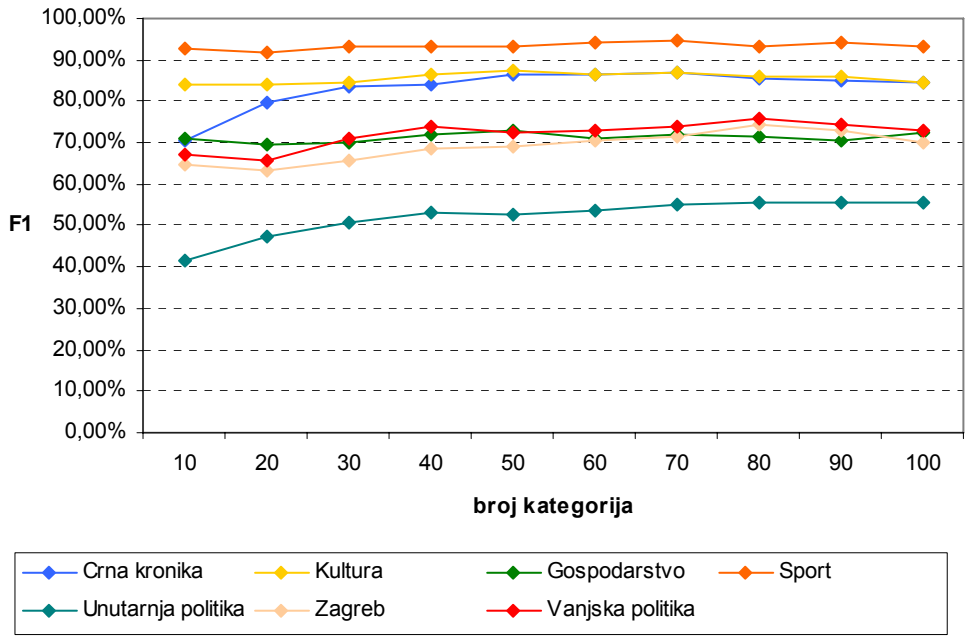
Pogledamo li grafove prikazane na slikama 34., 35., 36., 37. i 38. vidimo da najlošije rezultate dobivamo za nenormaliziranu bazu, nešto bolje za inflekcijski normaliziranu, a najbolje za derivacijski i terminirajuće

normaliziranu bazu. Rezultati su za otprilike 5% lošiji za nenormaliziranu bazu. Zaključujemo da morfološka normalizacija hrvatskog jezika, kao što smo i predvidjeli, utječe na uspješnost klasifikacije, ali s obzirom da je proveden postupak redukcije dimenzionalnosti razlika uspješnosti nije velika.

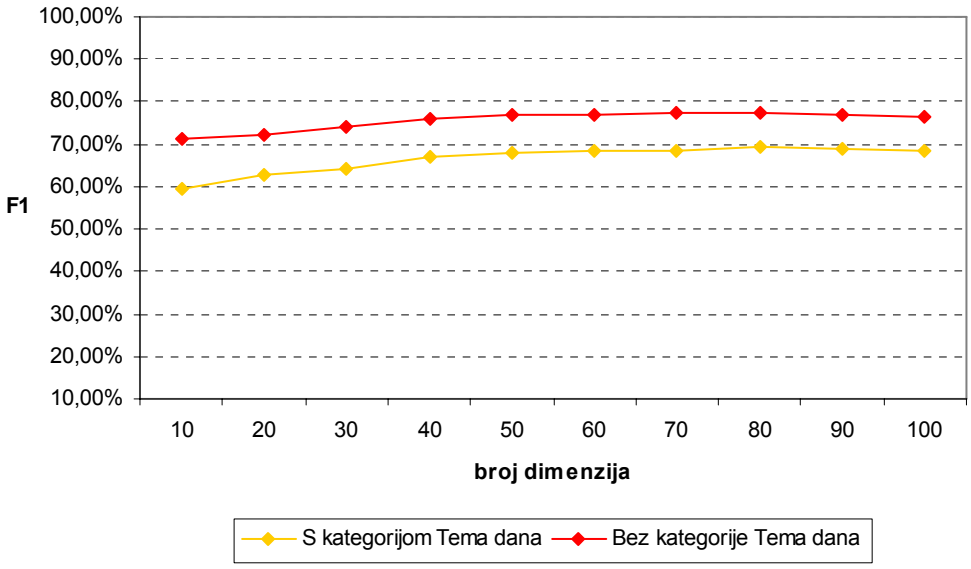
9.2.4. Uspješnosti klasifikacije bez kategorije *Tema dana*

Problem baze novinskih članaka Vjesnik je postojanje kategorije *Tema dana*. Članci koji pripadaju toj kategoriji po sadržaju pripadaju jednoj od ostalih sedam kategorija, ali na dan tiskanja novina bile su proglašene temom dana. Za očekivati je da će ta kategorija negativno utjecati na učinkovitost klasifikacije za sve kategorije. U ovom ispitivanju pokazat ćemo u kojoj mjeri ova kategorija utječe na rezultate klasifikacije.

Ispitivanje je provedeno na derivacijski normaliziranoj bazi Vjesnik s tim da su izbačeni dokumenti koji pripadaju kategoriji *tema dana.* Klasifikacija je provedena *metodom k = 1 najbližih susjeda*. Dimenzionalnost je reducirana metodom LSI. Ispitivanje je provedeno u ovisnosti o broju dimenzija. Uspješnost klasifikacije prikazana je na slikama 39. i 40.



Slika 39. Vrijednost mjere F1po kategorijama za derivacijski normaliziranu bazu Vjesnik bez kategorije *Tema dana*



Slika 40. Vrijednost mikrousrednjene mjere F1 za derivacijski normaliziranu bazu Vjesnik s i bez kategorije *Tema dana*

Promotrimo li slike 38. i 40. primijetit ćemo da je uspješnost klasifikacije za bazu Vjesnik bez kategorije *Tema dana* za otprilike 10% bolja od one koja uključuje sve kategorije. Usporedimo li slike 37. i 39. vidimo da je eliminacija kategorije *Tema dana* dovela do poboljšanja uspješnosti klasifikacije svih kategorija, ali najviše kategorije *Unutarnja politika* i *Vanjska politika*. Iz toga zaključujemo da je kao vijest dana najčešće proglašavan neki politički događaj.

10. Zaključak

Zadatak ovog diplomskog rada bio je ispitati utjecaj redukcije dimenzionalnosti metodom *latentno semantičkog indeksiranja* na uspješnost klasifikacije skupa dokumenata na hrvatskom i engleskom jeziku. Za tu svrhu je izgrađen sustav za redukciju dimenzionalnosti dekompozicijom na singularne vrijednosti. Implementirani sustav se po pitanju vremenskih i memorijskih zahtjeva u ovom istraživanju pokazao kao dobro rješenje pri rukovanju rijetkim matricama velikih dimenzija.

Ispitivanje je provedeno na bazama novinskih članaka *Croatia Weekly* i *Vjesnik*. Na temelju rezultata eksperimenata nameće se zaključak kako je redukcija dimenzionalnosti metodom *latentno semantičkog indeksiranja* bitan korak pri klasifikaciju dokumenata. Pokazano je kako je većina bitnih informacija sadržana u prvih pedesetak dimenzija. Budući da je početni broj dimenzija nekoliko desetaka tisuća, ovako drastična redukcija dimenzionalnosti znači znatno bržu klasifikaciju dokumenata. Osim toga, metoda *latentno semantičkog indeksiranja* pokazala se iznimno uspješnom u smanjenju broja pojavnica te rješavanja problema sinonima. To je posebno bitno za morfološki bogate jezike, kao što je hrvatski jezik, jer uspješnost klasifikacije čini manje ovisnom o postupku morfološke normalizacije.

11. Literatura

[1] R. Ahel, Postupak klasifikacije teksta temeljen na k-nn metodi i naivnom Bayesovom klasifikatoru, Diplomski rad, Fakultet elektrotehnike i računarstva, Zagreb, 2003.

[2] B. D. Bašić, B. Bereček, A. Cvitaš, Mining Textual Data In Croatian, MIPRO, (2005), Opatija.

[3] M. W. Berry, Large Scale Singular Value Computations, Interantional Journal of Super-Computer Applications, 6(1992), pp 13-49.

[4] M. W. Berry, T. Do, G. O'Brien, V. Krishna, S. Varadhan, SVDPACKC (Version 1.0) User's Guide, University of Tenessee, 1993.

[5] M. W. Berry, Z. Drmač, E. R. Jessup, Matrices, Vector Spaces and Information Retrieval, SIAM Review, 41(1999), pp 335-362.

[6] E. Bingham, H. Mannila, Random projection in Dimensionality Reduction: Application to Image and Text Data, Knowledge Discovery and Data Mining, 1998.

[7] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis, J. American Society for Information Science, 41(1990), pp 391-407.

[8] S. T. Dumais, M. W. Berry, G. W. O'Brien, Using Linear Algebra for Intelligent Information Retrieval, SIAM Review, 37(1995), pp 573 – 595.

[9] N. Elezović, Linearna algebra, Element, Zagreb, 2003.

[10] R. D. Fierro, E. P. Jiang, Lanczos and the Riemannian SVD in information retrieval applications, Numerical Linear Algebra with Applications, 2003.

[11] B. Fortuna, Kernel Canonical Correlation Analysis With Applications, SIKDD 2004 at multiconference IS 2004, (2004), Ljubljana.

[12] G. H. Golub, C. F. Van Loan, Matrix computations, 3. izdanje, USA: The Johns Hopkins University Press, Baltimore, MD, 1996.

[13] J. Jiang, Using Latent Semantic Indexing for Data Mining, The University of Tennessee, 1997.

[14] J. Jiang, M. W. Berry, J. M. Donato, G. Ostrouchov, N. W. Grady, Mining consumer product data via latent semantic indexing, Intelligent Data Analysis 3, (1999), pp 377-398.

[15] J. Jiang, On The Lanczos-Type Algorithms And Look-Ahead Remedies, Institute of Applied Mathematics, National Chao Tung University, 1997.

[16] A. Kontostathis, W. Pottenger, Detecting Patterns in the LSI Term – Term Matrix, Workshop on the Foundation of Data Mining and Discovery, The 2002 IEEE International Conference on Data Mining, (2002), pp. 243-248.

[17] S. Kurepa, Uvod u lienarnu algebru, Školska knjiga, Zagreb, 1985.

[18] M. Malenica, Primjena jezgrenih metoda u kategorizaciji teksta, Diplomski rad, Fakultet elektrotehnike i računarstva, Zagreb, 2004.

[19] J. Šnajder, Automatska normalizacija hrvatskog jezika, Tehnički izvještaj, Fakultet elektrotehnike i računarstva, Zagreb, 2004.

(http://www.zemris.fer.hr/~jan/textm/)

[20] M. Tadić, Hrvatski nacionalni korpus na Internetu, 2005. (http://www.hnk.ffzg.hr/)

[21] B. Tang, X.Luo, M. I . Heywood, M. Shepherd, A Comparative Study of Dimension Reduction Techniques for Document Clustering, Technical report, Halifax, 2004.

[22] D. Tominac, Klasifikacija teksta pomoću K-NN algoritma i naivnog Bayesovog klasifikatora, S eminarski rad, Fakultet elektrotehnike i računarstva, 2004.

12. Dodatak

12.1. Primjer izračunavnja svojstvenih vrijednosti i vektora matrice

Neka je zadana matrica $\mathrm{A} = \begin{bmatrix} 1 & 2 \\ -1 & 4 \end{bmatrix} \in R^{2x2}$.

Nađimo svojstvene vrijednosti i vekotre matrice A. Karakterisitčna jednadžba je:

$$\det(\lambda \cdot I - A) = \det\begin{bmatrix} \lambda - 1 & -2 \\ 1 & \lambda - 4 \end{bmatrix} = 0$$

iz čega slijedi:

$$\lambda^2 - 5\lambda + 6 = 0 .$$

Rješenja jednadžbe su: $\lambda_1 = 2$ i $\lambda_2 = 3$, a to su ujedno svojstvene vrijednosti matrice A.

Vlastiti vektori su rješenja jednadžbi:

$$2 \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ -1 & 4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 \text{ i}$$

$$3 \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 1 & 2 \\ -1 & 4 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0 .$$

Za $\lambda_1 = 2$ svojstveni vektor matrice A je $\begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$, a za $\lambda_2 = 3$ svojstveni vektor matrice A je $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

12.2. Primjer izračunavanja dekompozicije matrice na singularne vrijednosti

Neka je zadana matrica $\mathrm{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \in \mathrm{R}^{3x2}$.

Izračunajmo matrice U, Σ i V iz jednadžbe $A = U\Sigma V^T$. Prvo računamo singularne vrijednosti matrice A odnosno kvadratne korijene svojstvenih vrijednosti matrice A^TA.

Svojstvene vrijednosti od $A^TA = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$ dobiju se rješavanjem

karakteristične jednadžbe:

$$\det(\lambda \cdot I - A^TA) = \det\begin{bmatrix} \lambda - 2 & 0 \\ 0 & \lambda - 1 \end{bmatrix} = 0.$$

Rješenja jednadžbe su $\lambda_1 = 1$ i $\lambda_3 = 2$, a pripadni svojstveni vektori matrice A^TA su $v_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ i $v_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Svojstveni vektori matrice A^TA su desni singularni vektori matrice A iz čega slijedi:

$$V = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Kvadratni korijeni svojstvenih vrijednosti matrice A^TA su singularne vrijednosti matrice A koje čine dijagonalu matrice Σ, a ostatak matrice čine nule

$$\Sigma = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{1} \\ 0 & 0 \end{bmatrix}.$$

Stupci matrice U, lijevi singularni vektori matrice A, su svojstveni vektori matrice

$$AA^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Karakteristična jednadžba je

$$\det(\lambda \cdot I - AA^T) = \det\begin{bmatrix} \lambda - 1 & 0 & -1 \\ 0 & \lambda - 1 & 0 \\ -1 & 0 & \lambda - 1 \end{bmatrix} = 0$$

iz čega se jednostavno izračunaju svojstveni vektori matrice AA^T koji čine stupce matrice U:

$$U=\begin{bmatrix} \sqrt{2}/2 & 0 & \sqrt{2}/2 \\ 0 & -1 & 0 \\ -\sqrt{2}/2 & 0 & \sqrt{2}/2 \end{bmatrix}.$$

Primijetimo da vrijedi

$$A=\begin{bmatrix} \sqrt{2}/2 & 0 & \sqrt{2}/2 \\ 0 & -1 & 0 \\ -\sqrt{2}/2 & 0 & \sqrt{2}/2 \end{bmatrix}\begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{1} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}^{T}.$$