Mladen Nikolić Anđelka Zečević

NAUČNO IZRAČUNAVANJE

Beograd
2019.

Sadržaj

Na korisnim sugestijama zahvaljujemo se studentima Nikoli Dimitrijeviću, Aleksandri Ilić, Urošu Stegiću, Rastku Đorđeviću...

Glava 1

Uvod

Naučno izračunavanje je multidisciplinarna oblast koja se fokusira na rešavanje praktičnih problema u naučnim i inženjerskim disciplinama. Mnoštvo inženjerskih, ali i mnogih drugih, problema se rešava upravo metodama koje predstavljaju deo oblasti naučnog izračunavanja. Kao što često važi za discipline čiji je razvoj vođen zahtevima rešavanja praktičnih problema, ovo nije homogena oblast sa jasnim granicma. Međutim, iako ne postoji precizna definicija šta tačno ova oblast obuhvata, moguće je uočiti da većina univerzitetskih udžbenika i kurseva naučnog izračunavanja uključuje jedno šire jezgro znanja koje pre svega obuhvata numeričke metode raznovrsnih namena, a neretko i metode stohastičke simulacije. Pored navedenih, i druge metode koje služe za rešavanje problema u naučnim i inženjerskim disciplinama se mogu podvesti pod ovu oblast.

Pomenuta multidisciplinarnost naučnog izračunavanja kao oblasti se ogleda u primeni opštih znanja iz različitih oblasti matematike poput linearne algebre (matrični račun, sistemi jednačina, sopstveni vektori,...), algebre (polinomi, algebarske strukture,...), matematičke analize (konveksnost, izvodi, integrali,...), verovatnoće (slučajne promenljive, raspodele, matematičko očekivanje,...), statistike (statistički modeli, ocena parametara raspodela,...), potom mnogih specifičnijih znanja matematičkih znanja, poput optimizacije, Furijeove analize i drugih, ali i primeni znanja iz različith oblasti računarstva, uključujući programiranje, algoritmiku, strukture podataka, dizajn softvera i druge. Pored posedovanja pomenutih znanja, primena ovih metoda za rešavanje konkretnih problema često podrazumeva i dublje upućivanje u oblast kojoj problem pripada, što može biti fizika, hemija, biologija, građevina, geodezija, mašinstvo, rudarstvo, arheologija, sociologija, lingvistika, itd. Sve navedeno čini ovu oblast izazovnom, često i teškom za savladavanje, ali samim tim i izuzetno zanimljivom, raznovrsnom i nadasve primenljivom.

Primene metoda naučnog izračunavanja su mnogobrojne. Samo neke od njih, sa neformalno naznačenim znanjima potrebnim za njihovo rešavanje, su:

- prepoznavanje lica na slikama (sopstveni vektori matrica, neuronske mreže),

- prepoznavanje tumora na rendgenskim snimicima (verovatnosno modelovanje, optimizacija, aproksimacija funkcija),
- procena rizika bolesti na osnovu podataka o životnim navikama, uslovima života, rezultatima kliničkih ispitivanja (verovatnosno modelovanje, optimizacija, linearna algebra, aproksimacija funkcija),
- rekonstrukcija oblika i optimizacija mreže na osnovu podataka sakupljenih 3D skenerom (optimizacija),
- homogenizacija kvaliteta uglja na kopovima, upravljanjem radom bagera (optimizacija),
- pretraga zvučnog signala i slika (konvolucija),
- obrada i analiza signala, npr. uklanjanje šuma, kompresija,... (Furijeova analiza, aproksimacija funkcija),
- pretraga stranica na Internetu (Markovljevi lanci, sopstveni vektori matrica),
- preporučivanje filmova, proizvoda, muzike (dekompozicije matrica),
- modelovanje mehaničkog napona na delovima aviona, građevinskim konstrukcijama, mašinskim elementima (diferencijalne jednačine, metoda konačnih elemenata),
- semantička analiza teksta (numerička linearna algebra, neuronske mreže)
- igranje igara, npr. go (Monte Karlo simulacija, neuronske mreže, Markovljevi procesi odlučivanja)
- prepoznavanje tema u tekstu (verovatnosno modelovanje, stohastička simulacija)
- modelovanje dinamike populacije na osnovu arheoloških nalaza (verovatnosno modelovanje, stohastička simulacija)
- pozicioniranje robotske ruke (polinomi, algebarske strukture, Grebnerove baze)
- pronalaženje preseka 3D oblika (polinomi, algebarske strukture, Grebnerove baze)

Glava 2

Rešavanje problema matematičkim metodama

U mnogim domenima, poput prirodnih nauka i inženjerskih disciplina, matematika je, kao jezik za izražavanje teorija i formulisanje problema, ali i kao arsenal metoda za njihovo rešavanje, postala de facto standard već vekovima, ako ne i milenijumima, unazad. Poslednjih decenija, učestalost upotrebe matematike rapidno raste i u društvenim i humanističkim naukama. Osnovne prednosti upotrebe matematike u rešavanju problema kojima se ove discipline bave su preciznost formalnog izražavanja koju matematika nudi, kao i bogat skup metoda za rešavanje različitih matematičkih problema koji su razvijeni u toku više milenijuma razvoja matematike. Izbor i način primene ovih metoda drastično varira od oblasti do oblasti i od problema do problema. Međutim moguće je formulisati grube smernice za rešavanje problema iz različitih domena matematičkim metodama u nekoliko koraka:

Modelovanje U razmatranom domenskom problemu se uočavaju relevantne veličine i načini njihovog kvantitativnog izražavanja i odnosi između tih veličina, na osnovu čega se formuliše matematički model M razmatranog problema. Pored toga, uočava se pitanje na koje je, na osnovu formulisanog modela, potrebno dati odgovor, kako bi se polazni problem rešio i na osnovu toga se formuliše matematički problem P.

Rešavanje Bira se metoda kojom je moguće rešiti problem P i njenom primenom se dobija rešenje S.

Interpretacija Rešenje S, koje je izraženo u terminima modela M se interpretira u terminima polaznog problema i time se dobija željeno rešenje.

Prethodne smernice će prvo biti ilustrovane na trivijalnom primeru.

Primer 1 *Neka se dve vrste čaja dobijaju mešanjem crnog čaja i bergamota u različitim odnosima. Prva smesa se dobija mešanjem a_1 grama crnog čaja i b_1 grama bergamota i košta c_1 dinara, dok se druga dobija mešanjem a_2 grama*

crnog čaja i b_2 grama bergamota, a cena joj je c_2 dinara. Potrebno je odrediti koliko koštaju crni čaj i bergamot. Faze rešavanja problema su sledeće:

Modelovanje *Relevantne veličine su cena crnog čaja i bergamota u dinarima po gramu. Ove veličine se mogu označiti kao x_1 i x_2. U problemu figurišu i količine ovih sastojaka u gramima i ukupne cene. I one predstavljaju promenljive, ali ne nepoznate, već ulazne parametre problema. Model M koji odražava relevantne odnose veličina od interesa se postavlja na osnovu jednostavne analize postavke i glasi:*

$$a_1x_1 + b_1x_2 = c_1$$
$$a_2x_1 + b_2x_2 = c_2$$

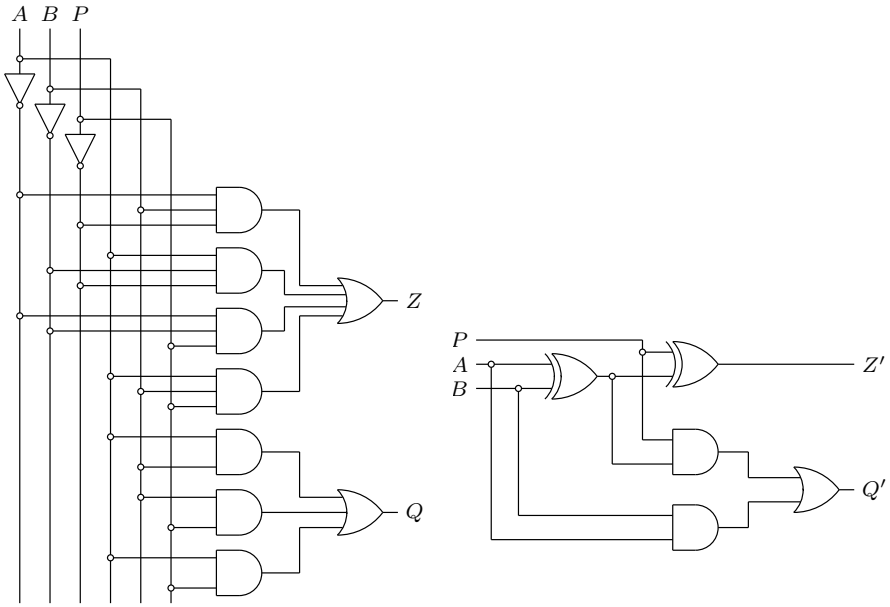
Na osnovu pitanja u polaznoj postavci, uočava se problem P, kao problem nalaženja vrednosti promenljivih x_1 i x_2 koje zadovoljavaju dati sistem jednačina.

Rešavanje *Metoda kojom je moguće rešiti ovaj sistem je metoda Gausove eliminacije i njenom primenom se dobijaju konkretne vrednosti promenljivih od interesa, ukoliko rešenje postoji.*

Interpretacija *Korak interpretacije je u ovom slučaju trivijalan.*

Nešto zanimljiviji primer, dat je u nastavku.

Primer 2 *Kombinatorna kola predstavljaju mreže povezanih logičkih elemenata kod kojih vrednosti izlaza zavise isključivo od vrednosti ulaza. Logički elementi predstavljaju elektronske implementacije logičkih veznika. Jedan problem verifikacije hardvera je provera ekvivalentnosti kombinatornih kola. Dva logička kola su ekvivalentna ukoliko za sve kombinacije vrednosti na svojim ulazima, daju iste izlaze. Ova vrsta provere je korisna u sledećem kontekstu. Kombinatorna kola mogu biti vrlo složena i dizajniraju se u alatima koji podržavaju neki od jezika za opis hardvera kao što su Verilog ili VHDL. Pre nego što se na osnovu kreiranog dizajna pristupi fizičkoj implementaciji logičkog kola, taj dizajn prolazi kroz niz transformacija kojima se vrše optimizacije kola kako bi se uštedelo na njegovoj površini, brzini i slično. Svaki od koraka ovog postupka može biti vrlo složen i, iako alogoritmi na kojima pomenute transformacije počivaju garantuju održanje korektnosti, usled složenosti softvera u kojem su ti algoritmi implementirani, uvek postoji mogućnost da je u nekom koraku napravljena greška i da finalni, optimizovani, dizajn kola više nije ekvivalentan polaznom. Zbog toga je pre fizičke izrade logičkog kola potrebno proveriti ekvivalentnost polaznog i finalnog dizajna kola. Treba primetiti da ustanovljena ekvivalentnost ne garantuje funkcionalnu korektnost kola — to da ono zaista radi ono što bi trebalo. Međutim, i to je moguće ustanoviti proverom ekvivalentosti sa kolom za koje je poznato da je funkcionalno korektno, ukoliko takvo kolo postoji. Koraci provere ekvivalentnosti kombinatornih kola mogu biti sledeći:*



Slika 2.1: Osnovni i optimizovani dizajn sabirača

Modelovanje *Kao što vrednosti na izlazima kombinatornog kola zavise samo od vrednosti na njegovim ulazima, tako i interpretacija iskaznih formula zavisi samo od valuacije pridružene toj iskaznoj formuli. Shodno tome, provera ekvivalentnosti kombinatornih kola se vrši tako što se svakom kolu pridruži iskazna formula koja odgovara njegovom dizajnu. Neka su to formule A i B. Ukoliko su kola ekvivalentna, za sve kombinacije vrednosti ulaza, vrednosti izlaza su iste. U terminima iskaznih formula, za svaku valuaciju v, mora da važi važi $I_v(A) = I_v(B)$. Odnosno, formula $A \Leftrightarrow B$ mora biti tautologija, a formula $\neg(A \Leftrightarrow B)$ nezadovoljiva. Relevantni problem koji je potrebno rešiti je provera zadovoljivosti formule $\neg(A \Leftrightarrow B)$.*

Rešavanje *Zadovoljivost iskazne formule se može proveriti pomoću SAT-rešavača (npr. zasnovanog na DPLL proceduri), koji daje ili zadovoljavajuću valuaciju ili informaciju da formula nije zadovoljiva.*

Intepretacija *Ukoliko je rešavač dao zadovoljavajuću valuaciju, ona predstavlja dokaz da kola nisu ekvivalentna i daje konkretne ulaze za koje se izlazi kola razlikuju. Ukoliko je rešavač pružio informaciju da je formula nezadovoljiva, onda su polazna kola ekvivalentna.*

Postupak provere ekvivalentnosti se može ilustrovati na primeru sabirača. Recimo da je optimizovanjem prvog kola na slici 2.1, dobijeno drugo. Na

osnovu dizajna, za svaki od izlaza može se formirati iskazna formula koja mu odgovara:

$$Z = (\neg A \wedge B \wedge \neg P) \vee (A \wedge \neg B \wedge \neg P) \vee (\neg A \wedge \neg B \wedge P) \vee (\neg A \wedge \neg B \wedge P) \vee (A \wedge B \wedge P)$$

$$Q = (A \wedge B) \vee (B \wedge P) \vee (A \wedge P)$$

$$Z' = (A \underline{\vee} B) \underline{\vee} P$$

$$Q' = (A \wedge B) \vee (A \underline{\vee} B) \wedge P$$

Kola su ekvivalentna ukoliko je formula $\neg((Z \Leftrightarrow Z') \wedge (Q \Leftrightarrow Q'))$ *nezadovoljiva. Treba imati u vidu da formule koje se dobijaju iz ovakvih primena mogu imati i desetine hiljada, pa i stotine hiljada promenljivih, ali da* SAT*-rešavači ipak uspevaju da provere njihovu zadovoljivost zahvaljujući pravilnostima koje su prisutne u tim formulama, a koje* SAT*-rešavači u toku rada mogu da nauče i iskoriste.*

Uprkos svojoj jednostavnosti, prethodni primeri već ukazuju na određena opšta zapažanja. Prvo, prilikom modelovanja, moguće je postaviti problem na različite načine. U slučaju provere ekvivalentnosti kombinatornih kola, bilo je moguće svesti dati problem na bilo koji NP-kompletan problem. Izbori napravljeni u koraku modelovanja, mogu dovesti do toga da su različite metode primenljive u koraku rešavanja ili da ista metoda ima različitu efikasnost. Korak interpretacije je najčešće jednostavan i u velikoj meri je određen konvencijama dogovorenim u vreme modelovanja i svojstvima metoda primenjenog u koraku rešavanja. U nastavku će biti detaljnije diskutovan svaki od navedenih koraka.

2.1 Modelovanje problema

Korak modelovanja je često intelektualno najzahtevniji korak rešavanja problema. Jedan razlog za to je što je precizno uočavanje relevantnih veličina i odnosa među njima, kao i njihovo precizno formalno izražavanje, zahtevaju visok nivo analitičnosti, širinu matematičkog obrazovanja i duboko razumevanje relevantnih matematičkih koncepata. Drugi razlog je jaka interakcija između odluka donesenih u koraku modelovanja i izbora koji su na raspolaganju u koraku rešavanja. Naime, loše odluke u koraku modelovanja mogu drastično uticati na skup raspoloživih metoda i načina njihove primene u koraku rešavanja.

Matematički model nekog fenomena predstavlja skup matematičkih formula kojima se izražavaju odnosi između relevantnih veličina polaznog problema, koje su u tim formulama predstavljene promenljivim. Modeli gotovo uvek predstavljaju apstrakcije, odnosno pojednostavljenja fenomena koji se posmatra. Naime, osnovna svrha modela je da se odgovori na neko specifično pitanje o fenomenu od interesa, a ne da se razmatraju sva njegova svojstva. Otud se prilikom formulisanja modela, veliki broj detalja apstrahuje, odnosno zanemaruje, a modeluju se samo oni aspekti fenomena koji su od interesa za pitanje na koje je potrebno odgovoriti. Recimo, u slučaju primera vezanog za dve vrste

čaja, u modelu nisu potrebni detalji vezani za zemlju porekla sastojaka, njihovu boju, zadovoljstvo mušterija ambalažom, itd. U slučaju primera vezanog za ekvivalenciju kombinatornih kola, potpuno su zanemarena pitanja poput tehnologije izrade logičkih elemenata, tačne veličine logičkih kola, imena kompanija koje su logička kola proizvela, itd. Ništa od ovih detalja nije relevantno za odgovaranje na postavljena pitanja. Stoga, apstrakcija predstavlja jedan od ključnih postupaka u formulisanju matematičkog modela.

Nakon što su identifikovane relevantne veličine i njihovi odnosi, potrebno je tim veličinama pridružiti promenljive, a odnosima formule koje ih predstavljaju. Pritom je poželjno izabrati formulaciju koja se uklapa u neku od poznatih matematičkih teorija i koja ima povoljnija matematička svojstva (npr. neprekidnost, diferencijabilnost, konveksnost), kako bi skup metoda koje se mogu primeniti za rešavanje problema bio širi i kako bi njihova primena bila računski efikasnija.

Jedan od pristupa izgradnji modela, u cilju jednostavnijeg rešavanja problema, je njegovo iterativno pojednostavljivanje dok god je greška koja se time unosi u rešenje prihvatljiva. Pristupi pojednostavljivanju su vrlo raznovrsni. Neki od primera su:

- zamena beskonačnih procesa konačnim procesima, poput zamene integrala i redova konačnima sumama ili izvoda konačnim razlikama,
- zamena opštih matrica matricama jednostavnije forme, poput blok dijagonalnih, trakastih i dijagonalnih ili matricama niskog ranga
- zamena proizvoljnih funkcija jednostavnijim funkcijama, poput polinoma
- zamena proizvoljnih funkcija, funkcijama sa poželjnijim matematičkim svojstvima, poput aproksimacije nekonveksnih funkcija konveksnim funkcijama
- zamena nelinearnih problema linearnim problemima
- zamena diferencijalnih jednačina algebarskim jednačinama
- zamena beskonačno dimenzionalnih prostora konačno dimenzionalnim prostorima

Recimo, kako bi se rešio sistem diferencijalnih jednačina, moguće je prvo zameniti ga sistemom algebarskih jednačina, potom njega sistemom linearnih jednačina, a potom matricu linearnog sistema, matricom jednostavnijeg oblika, koja omogućava efikasniju inverziju i time efikasnije rešavanje sistema. Jasno, svaki od ovih koraka u opštem slučaju unosi dodatnu grešku, pa je potrebno ustanoviti da se prilikom svakog od njih rešenje ili ne menja u odnosu na rešenje polaznog sistema ili da je greška dovoljno mala, tako da je dobijeno rešenje i dalje upotrebljivo u praktičnom kontekstu. Očigledno, da bi ovakva strategija formulisanja modela bila upotrebljiva, potrebno je imati na raspolaganju:

- alternativni problem koji je moguće lakše rešiti, a čije rešenje nije značajno drugačije od polaznog i
- transformaciju tekućeg problema u taj lakši problem koja dozvoljava izražavanje rešenja lakšeg problema u terminima tekućeg problema.

Poslednji zahtev je važan zbog koraka interpretacije.

Pored postavljanja modela koji ustanovljava odnose među relevantnim veličinama, potrebno je obratiti pažnju i na pitanje na koje je potrebno odgovoriti u odnosu na taj model, kako bi se polazni problem rešio. U nekim slučajevima, kao u primeru sa čajem, potrebno je naći jedinstveno rešenje problema. Vrlo često, potrebno je naći najbolje iz većeg skupa mogućih rešenja. U takvim situacijama, potrebno je primeniti metode matematičke optimizacije.

Primer 3 *Signal x se beleži sa šumom (kao na slici 2.2) i usled toga nije dostupan, već je umesto njega dostupan uzorak $y = x + \varepsilon$. Zadatak je doći do pravog signala x ili njegove aproksimacije koja je kvalitetnija od uzorka y. Drugim rečima, potrebno je u što većoj meri eliminisati uticaj šuma ε. Za veliki broj signala, očekuje se određena doza neprekidnosti, pa stoga susedni elementi rekonstruisanog uzorka x ne treba da budu previše daleko. S jedne strane, potrebno je dobiti nov uzorak x koji je u nekoj meri blizak uzorku y, ali da s druge strane ima neprekidno ponašanje, umesto skokova koji su karakteristični za šum. Ovo odgovara minimizacionom problemu*

$$\min_x \|x - y\|^2 + \lambda \sum_{i=1}^{n-1} (x_i - x_{i+1})^2$$

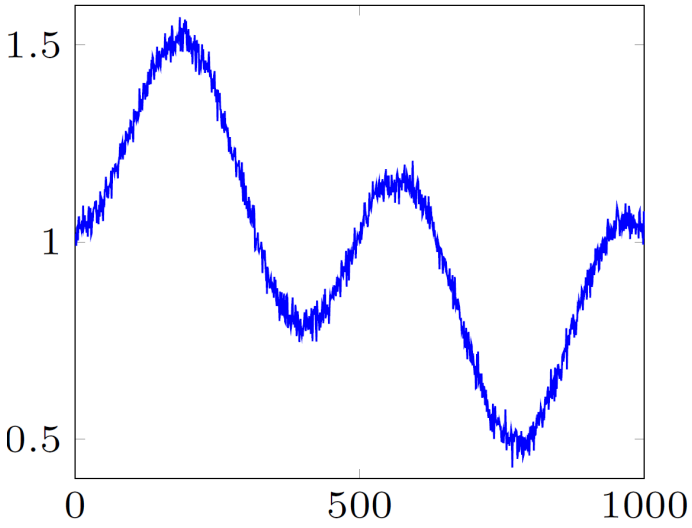
gde parametar λ kontroliše odnos između vernosti zabeleženom uzorku i glatkosti rekonstruisanog signala.

Primer 4 *Potrebno je napraviti sistem koji automatski prepoznaje teme o kojima se govori u delovima nekog teksta, kao na slici 2.3. Pretpostavlja se da je dat skup dokumenata, ali ne i da je poznato o kojim temama se govori u kom delu teksta! Samo je poznato da postoji ukupno k unapred poznatih tema. Jedan pristup je verovatnosno modelovanje pomoću generativnog modela – modela koji opisuje na koji način tekst nastaje, ali zavisi od određenog broja parmetara, koje treba oceniti iz podataka, tako da tekstovi koje bi model mogao da generiše najviše liče na tekstove iz već poznatog skupa dokumenata.*

Neka je korpus dokumenata $\mathcal{D} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$ sastavljen iz dokumenata oblika $\mathbf{w} = (w_1, w_2, \ldots, w_N)$, pri čemu je svaka reč w_i dokumenta, jedan od brojeva iz skupa $\{1, 2, \ldots, V\}$. Taj skup brojeva se naziva vokabular. Slično, svakom dokumentu odgovara neki vektor tema $\mathbf{z} = (z_1, z_2, \ldots, z_N)$, pri čemu je svaka tema z_i, jedan od brojeva iz skupa $\{1, 2, \ldots, k\}$.

Jedan generativni model dokumenta $\mathbf{w}$ (poznat kao latentna Dirihleova alokacija*) je sledeći:*

1. *Izabrati N u skladu sa Puasonovom raspodelom $Poisson(\lambda)$.*



Slika 2.2: Signal zapisan sa šumom.

2. *Izabrati* θ *u skladu sa Dirihleovom raspodelom* $Dir(\alpha)$.

3. *Svaka reč* w_i *(od ukupno* N*), generiše se na sledeći način:*

   - *Izabrati temu* z_i *iz multinomijalne raspodele* $Multinomial(\theta)$.
   - *Izabrati reč* w_i *iz multinomijalne raspodele* $p(w_i|z_i, \beta) = Multinomial(\beta)$.

Pretpostavka o korišćenju Puasonove raspodele nije ključna, već je samo potrebna neka diskretna raspodela nad celim brojevima. Dirihleova raspodela je raspodela nad k *dimenzionalnim vektorima nenegativnih brojeva, čije se koordinate sumiraju na 1. Samim tim, elementi vektora* θ *se mogu interpretirati kao verovatnoće svake od* k *tema. Multinomijalna raspodela je raspodela nad konačnim skupom tema i parametrizovana je vektorom verovatnoća svake od tih tema. Matrica parametara* β *određuje verovatnoće da neka reč odgovara nekoj temi, odnosno* $\beta_{ij} = p(w_i|z_j)$.

Parametri ovog generativnog modela su α *i* β*. Za realizaciju sistema koji prepoznaje teme u tekstovima, potrebno je prvo na osnovu podataka oceniti parametre* α *i* β*, tako da verovatnoća da ovakav generativni model generiše raspoloživi korpus bude maksimalna. Potom je potrebno za svaki dokument od značaja* w *odrediti vektor tema* z.

Ukoliko su poznate vrednosti parametara α *i* β*, zajednička raspodela ima sledeći oblik*

$$P(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = P(N)P(\theta|\alpha, \beta)P(\mathbf{z}, \mathbf{w}|\theta, \alpha, \beta) = P(N)P(\theta|\alpha)P(\mathbf{z}, \mathbf{w}|\theta, \beta) =$$

$$P(N)P(\theta|\alpha)\prod_{i=1}^{N} P(z_i, w_i|\theta, \beta) = P(N)P(\theta|\alpha)\prod_{i=1}^{N} P(z_i|\theta)P(w_i|z_i, \beta)$$

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Slika 2.3: Primer određivanja tema u članku. Svaka tema je označena različitom bojom.

Ova raspodela uključuje i promenljive čije vrednosti u korpusu nisu poznate. Stoga je za ocenu parametara α i β potrebna raspodela $P(\mathbf{w}|\alpha,\beta)$, koja se iz prethodne raspodele dobija integracijom po θ i sumiranjem po z:

$$P(\mathbf{w}|\alpha,\beta) = P(N)\int p(\theta,\alpha)\left(\prod_{i=1}^{N}\sum_{\mathbf{z}}^{P}(z_i|\theta)P(w_i|z_i,\beta)\right)d\theta$$

Pretpostavljajući da su dokumenti nezavisno generisani, verovatnoća korpusa je proizvod verovatnoća pojedinačnih dokumenata:

$$P(\mathcal{D}|\alpha,\beta) = \prod_{d=1}^{M}P(N_d)\int p(\theta_d,\alpha)\left(\prod_{i=1}^{N_d}\sum_{\mathbf{z}_d}^{P}(z_{di}|\theta_d)P(w_{di}|z_{di},\beta)\right)d\theta_d$$

Optimizacioni problem

$$\max_{\alpha,\beta} P(\mathcal{D}|\alpha,\beta)$$

Reč	Dete	trči	niz	kosu	padinu.
Vrsta reči	imenica	glagol	predlog	pridev	imenica

Tabela 2.1: Primer rečenice i vrsta svake od reči.

je težak za rešavanje i obično se umesto njega rešava neki problem koji ga aproksimira.

Kada su parametri α i β poznati, zaključivanje se vrši tako što se za neki dokument $\mathbf{w}$ *u kojem je potrebno identifikovati teme, reši optimizacioni problem*

$$\max_{\theta,\mathbf{z}} P(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)$$

pri čemu važi

$$P(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta) = \frac{P(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta)}{P(\mathbf{w}|\alpha, \beta)}$$

Ovaj problem je takođe težak za rešavanje.

Primer 5 *Neka je u proizvoljnoj rečenici srpskog jezika potrebno rečima pridružiti vrste reči. Na primer, kao u tabeli 2.1. Za neke reči je vrsta uvek jednoznačno određena. S druge strane, za neke nije. Recimo, u datoj rečenici, reč „kosu“ bi mogla biti i pridev i imenica. Razrešavanje je moguće uraditi na osnovu konteksta reči. Ukoliko je poznat neki tekstualni korpus u kojem je za sve reči obeležena i njihova vrsta, moglo bi se izbrojati koliko često svakoj od reči odgovara koja vrsta. Ove frekvencije zapravo aproksimiraju verovatnoće i odlučivanje o vrsti reči bi se uvek moglo sprovesti u skladu sa najverovatnojom vrstom reči. Međutim, ovaj pristup, iako bolji od nasumičnog razrešavanja višeznačnosti, ne uzima u obzir kontekst reči. Ukoliko se rečenica sastoji od reči $w_1, w_2, \ldots, w_N$ i ukoliko $t_1, t_2, \ldots, t_N$ predstavljaju vrste odgovarajućih reči iz nekog konačnog skupa vrsta, bilo bi poželjno rešiti sledeći problem*

$$\max_{t_1,t_2,\ldots,t_N} P(t_1, t_2, \ldots, t_N|w_1, w_2, \ldots, w_N)$$

Osnovni problem sa ovim modelom je nemogućnost ocenjivanja date verovatnoće jer u korpusu verovatno nema dovoljno pojavljivanja date rečenice da bi se verovatnoća pouzdano ocenila. Alternativa bi bila posmatrati manje delove rečenice pojedinačno i ocenjivati verovatnoće pojavljivanja neke reči ili vrste reči nakon prethodne dve. Jedna formulacija takvog problema bi bila

$$\max_{t_1,t_2,\ldots,t_N} \prod_{i=1}^{N} P(t_i|w_i, w_{i-1}, w_{i-2})$$

Iako ovo rešenje može delovati adekvatno, i dalje nije zadovoljavajuće iz istog razloga – trojke reči se u korpusu retko ponavljaju i određivanje navedene verovatnoće nije pozudano. Alternativno rešenje je da se umesto prethodnih reči

posmatraju vrste prethodnih reči, kojih je manje nego reči i stoga se očekuje češće ponavljanje njihovih kombinacija.

$$\max_{t_{-1},t_0,\ldots,t_{N+1}} \left(\prod_{i=1}^{N} P(t_i|t_{i-1},t_{i-2})P(w_i|t_i) \right) P(t_{N+1}|t_N)$$

Vrste reči t_{-1}, t_0, t_{N+1} su oznake van predviđenog skupa reči i služe da označe početak i kraj rečenice. Ovo nije neophodno, ali poboljšava performanse modela. Kada su verovatnoće ocenjene iz korpusa, za određivanje vrsta reči na osnovu ovog modela, koristi se Viterbijev algoritam.

Imajući u vidu da su modeli pojednostavljene predstave stvarnog fenomena, treba imati u vidu nekoliko upozorenja, prilikom njihovog korišćenja:

- Model ne oslikava precizno stvarnost.
- Model može biti dobar u određenim aspektima (intervalu vremena, regionu prostora, određenom režimu rada,...), a da u drugima nije.
- Podešavanje podataka, kako bi se prilagodili modelu vodi pozitivnim ishodima evaluacije, ali i modelima koji ne rade u praksi.
- Ne treba se držati modela koji ne radi.

2.2 Rešavanje problema

Ukoliko je prilikom modelovanja problema vođeno računa da se formulacija problema uklopi u neku od poznatih matematičkih teorija i da problem ima poželjna matematička svojstva, verovatno je da će metoda za njegovo rešavanje već biti na raspolaganju. Ukoliko to nije slučaj, nekada je potrebno razviti i samu metodu. Izbor metode i njene primene je vrlo tesno vezan za konkretnu formulaciju problema, pa u nastavku neće biti reči o rešavanju problema u opštem slučaju, već će biti ilustrovano na koji način odluke donesene prilikom modelovanja utiču na korak rešavanja problema.

Primer 6 *Pretpostavimo da je potrebno predvideti neki relevantan zdravstveni parametar, poput rizika od određene bolesti, na osnovu skupa testova koji se nad pacijentom sprovode i drugih podataka o njemu. Neka je promenljiva koja se predviđa realan broj za koji je poznato da velike vrednosti označavaju visok rizik, a male vrednosti označavaju nizak rizik, kao i da se ostale promenljive, koje predstavljaju rezultate merenja u izvršenim testovima i druge podatke o pacijentu, mogu predstaviti u vidu realnih brojeva. Na primer, to mogu biti nivo šećera i holesterola u krvi, prosečan broj cigareta koje pacijent puši na dan, prosek primanja, da li pacijent živi u gradu ili na selu, itd. Pretpostavimo da je na raspolaganju skup podataka $\mathcal{D} = \{(x_i, y_i)|i = 1, \ldots, N\}$, koji uključuje vrednosti svih pomenutih promenljivih za veliki broj pacijenata. Kako je potrebno*

vršiti predviđanje, potrebno je izraziti vezu između rizika od bolesti i ostalih promenljivih, nekom matematičkom funkcijom. Ta funkcija mora zavisiti od nekih parametara, kako se modelovanje ne bi svelo na pogađanje konkretne funkcije. Jedan jednostavan i uobičajen izbor je izbor linearnog modela zavisnosti:

$$f(x, \alpha) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i$$

gde je $f(x, \alpha)$ funkcija kojom se aproksimira rizik y, a x_i su promenljive koje opisuju stanje pacijenta. Postavlja se pitanje, kako je moguće izabrati koeficijente α, tako da predviđanje bude dobro. Ukoliko se može očekivati da ista zakonitost koja važi kod već poznatih pacijenata važi i kod budućih pacijenata, racionalna strategija bi bila, pronaći one koeficijente α za koje vrednosti funkcije $f(x, \alpha)$ ne odstupaju mnogo od vrednosti rizika za već poznate pacijente. Ovo je očigledno problem u kojem je potrebno naći najbolje iz skupa potencijalnih rešenja. Jedna formulacija ovog problema može biti kroz minimizaciju kvadratne greške:

$$\min_{\alpha} \sum_{i=1}^{N} (f(x_i, \alpha) - y_i)^2$$

Očigledno, ukoliko je data greška mala, odstupanje je uglavnom malo (osim eventualno u slučaju malog broja pacijenata) i dobijeno rešenje dobro izražava zaivsnost rizika od ostalih promenljivih, a ukoliko je velika, dobijeno rešenje očito nije dobro. Konkretan oblik greške koja se minimizuje je mogao biti i drugačiji. Na primer, mogao je uključivati apsolutne vredosti umesto kvadrata razlika:

$$\min_{\alpha} \sum_{i=1}^{N} |f(x_i, \alpha) - y_i|$$

Ovakva formulacija je bolja jer velika odstupanja u pojedinačnim slučajevima, zbog nedostatka kvadrata ne utiču drastično na ukupnu vrednost greške, ali funkcija nije diferencijabilna po parametrima α, što otežava proces minimizacije.

Ukoliko je zarad predviđanja potrebno vršiti testiranja pacijenata, koja mogu biti skupa i za njih neprijatna (poput uzimanja krvi) ili čak rizična ili je potrebno od njih zahtevati privatne informacije (poput visine primanja), poželjno je da promenljivih koje će biti uključene u model bude što manje, čak i ako je kvalitet predviđanja nešto niži. Stoga se polazni problem može zameniti novim, koji preferira rešenja sa što manjim brojem koeficijenata koji nisu nula:

$$\min_{\alpha} \sum_{i=1}^{N} (f(x_i, \alpha) - y_i)^2 + \lambda \sum_{i=1}^{n} I(\alpha_i \neq 0)$$

gde je I indikatorska funkcija koja ima vrednost 1 ukoliko je uslov u zagradama tačan, a 0 u suprotnom. U navedenom optimizacionom problemu, parametar λ kontroliše odnos značaja koji se pridaje kvalitetu aproksimacije i broju testova koje je potrebno izvršiti. Ovaj problem dobro izražava nameru da broj

testova nad pacijentom bude što manji, ali funkcija koju treba minimizovati je nekonveksna, nediferencijabilna u nuli, a konstantnog gradijenta van nule (što onemogućava primenu gradijentnih metoda optimizacija) i čak prekidna, usled čega optimizacioni problem zahteva primenu metoda kombinatorne optimizacije i dokazano je da je NP-težak. Jedan način da se izbegne kombinatorna optimizacija je da se umesto funkcije I koristi neka druga funkcija koja je aproksimira, a koja će imati netrivijalan gradijent. Jedna alternativa datom problemu bi bila:

$$\min_{\alpha} \sum_{i=1}^{N} (f(x_i, \alpha) - y_i)^2 + \lambda \sum_{i=1}^{n} \sqrt[m]{\alpha_i^2}$$

Ova funkcija nije diferencijabilna u nuli (za šta postoje rešenja), ali ima nenula gradijent u ostalim tačkama i neprekidna je. Za visoku vrednost m, aproksimacija je vrlo dobra. Međutim, kvadratna greška je konveksna funkcija parametara α, a koren je konkavna funkcija. Usled toga, minimizacioni problem je nekonveksan, što može dovesti do većeg broja lokalnih minimuma, do sporije konvergencije i manjeg broja primenljivih optimizacionih metoda, nego u slučaju da je optimizacioni problem konveksan. Jedna konveksna aproksimacija funkcije I se može dobiti upotrebom apsolutne vrednosti:

$$\min_{\alpha} \sum_{i=1}^{N} (f(x_i, \alpha) - y_i)^2 + \lambda \sum_{i=1}^{n} |\alpha_i|$$

Sada je problem konveksan, ali ponovo postoji problem nediferencijabilnosti u nuli kad god je neki parametar α_i jedna nuli. Diferencijabilna aproksimacija

$$\min_{\alpha} \sum_{i=1}^{N} (f(x_i, \alpha) - y_i)^2 + \lambda \sum_{i=1}^{n} \alpha_i^2$$

ne mora eliminisati nijedan parametar.

U nizu predloženih aproksimacija, postavlja se pitanje najboljeg odnosa između povoljnosti matematičkih svojstava i greške koja se unosi aproksimacijama. U slučaju navedenog problema, najčešće se koristi formulacija:

$$\min_{\alpha} \sum_{i=1}^{N} (f(x_i, \alpha) - y_i)^2 + \lambda \sum_{i=1}^{n} |\alpha_i|$$

sa specifičnim tehnikama optimizacije razvijenim za ovaj problem.

2.3 Interpretacija rešenja

Korak interpretacije je najjednostavniji od pomenutih koraka. Ukoliko postoji jasna korespodencija između veličina u razmatranom fenomenu i promenljivih u postavljenom modelu i ukoliko svaka od transformacija pojednostavljivanja modela omogućava izražavanje rešenja pojednostavljenog problema u

terminima prethodnog problema, interpretacija se svodi na niz koraka kojima se od rešenja problema koji je rešen u koraku rešavanja dobijaju rešenja problema koji mu prethode. U ovome, kao i u koraku modelovanja, potrebno je obratiti pažnju na merne jedinice koje odgovaraju relevantnim veličinama.

2.4 Aproksimacije i greške u izračunavanju

Metode naučnog izračunavanja su mahom metode numeričkog izračunavanja. Pored takvih metoda, postoje i metode simboličkog izračunavanja. Osnovna razlika među njima je u tome što su metode numeričkog izračunavanja aproksimativne i oslanjaju se na konačne zapise realnih brojeva u računaru, poput zapisa brojeva u pokretnom zarezu prema standardu IEEE754, dok se metode simboličkog izračunavanja oslanjaju na manipluaciju simbolima i brojevima neograničene preciznosti kako bi sva izračunavanja bila sprovedena egzaktno. Numeričko izračunavanje je najčešće efikasnije od simboličkog i stoga je dominantno u praksi. Otud su aproksimacija i greške izračunavanja važna tema u naučnom izračunavanju.

Postoje različiti izvori neegzaktnosti u naučnom izračunavanju. Neki od njih su prisutni već pre samog izračunavanja:

- **Modelovanje:** Modelovanje često uključuje apstrakciju raznih detalja i pojednostavljivanje, kako bi rešavanje problema bilo efikasnije.

- **Empirijska merenja:** Empirijska merenja uvek uključuju dozu neprecizznosti zbog nesavršenosti mernih instrumenata, nepovoljnih uslova radne sredine, itd.

- **Prethodna izračunavanja:** Ulazni podaci mogu biti proizvod prethodnih izračunavanja, tako da se njihova greška akumulira još pre nego što tekuće izračunavanje počne.

Navedeni problemi nekada nisu otklonjivi, ali i dalje utiču na rezultate izračunavanja i stoga moraju biti uzeti u obzir, prilikom evaluacije finalnih rezultata izračunavanja. Izvori greške na koje se može lakše uticati su:

- **Diskretizacija i odsecanje:** Nekada se prilikom modelovanja neprekidne veličine zamenjuju diskretnim. Granularnost diskretizovane skale, utiče na kvalitet dobijenog rešenja. Takođe, greška koja se pravi prilikom zamene beskonačnih procesa konačnim, često se može kontrolisati. Na primer, u slučaju reda, moguće je uticati na broj elemenata konačne sume kojom se red aproksimira.

- **Zaokruživanje:** Računarske reprezentacije realnih brojeva su nužno neegzaktne. Međutim, broj decimala koje se koriste prilikom aproksimacije skupa realnih brojeva značajno utiče na kvalitet rešenja.

Primer 7 *Neka se površina Zemlje računa pomoću formule*

$$A = 4\pi r^2$$

gde je r poluprečnik sfere. Korišćenje ovakve formule nužno dovodi do niza aproksimacija:

- *Modelovanje Zemlje sferom predstavlja idealizaciju njenog stvarnog oblika - geoida.*
- *Vrednost poluprečnika Zemlje, koja je ulaz u navedenu formulu, od približno $6370km$ je proizvod niza merenja i računice nad tim merenjima.*
- *Vrednost broja π se može definisati beskonačnim procesom, ali se za svrhe realnog računanja mora izvršiti odsecanje.*
- *Prilikom čuvanja numeričkih vrednosti i vršenja operacija nad njima, u računaru se nužno vrši zaokruživanje.*

Iz prethodne diskusije se može uočiti da postoje dve vrste grešaka – *greške podataka*, koje nastaju usled raspolaganja nesavršenim reprezentacijama podataka nad kojima je potrebno izvršiti računanje i *greške izračunavanja*, koje nastaju usled nesavršenosti procesa izračunavanja u računaru.

U kontekstu izračunavanja u prisustvu greške, u praksi je često potrebno raspolagati ne samo rešenjem već i procenom greške tog rešenja. Stoga je potrebno definisati načine na koje se greška može kvantifikovati. Ukoliko su prava i približna vrednost neke veličine x i $\hat{x}$, onda je najjednostavnija mera greške razlika tih vrednosti. Ta vrsta greške se naziva *apsolutna greška*:

$$E(x, \hat{x}) = \|x - \hat{x}\|$$

U nekim situacijama apsolutna greška predstavlja adekvatnu meru greške. Recimo, ako je potrebno rasporediti nameštaj u stanu i za komad nameštaja se ostavi 10cm dodatnog prostora u odnosu na predviđenu dužinu, apsolutna greška je upravo relevantna mera greške. Bez obzira na to da li je komad nameštaja dugačak 1m ili 3m, ukoliko bude duži za više od 10cm od predviđenog, neće se uklopiti u planirani raspored. S druge strane, u zavisnosti od toga da li neka kompanija ugovara posao vredan 10.000 evra ili posao vredan 10.000.000 evra, greška u proceni od 1000 evra može biti izrazito važna ili zanemarljiva. U takvim situacijama, apsolutna vrednost nije adekvatna mera. Tada je potrebno izraziti grešku u odnosu na stvarnu vrednost razmatrane veličine i za to služi *relativna greška*:

$$R(x, \hat{x}) = \frac{\|x - \hat{x}\|}{\|x\|}$$

Treba imati u vidu da stvarna vrednost neke veličine najčešće nije poznata. U suprotnom ne bi bilo potrebe da se diskutuje greška. Zbog toga, ni vrednosti greške najčešće nisu poznate, već se često barata njihovim ocenama ili gornjim granicama.

2.5 Stabilnost, uslovljenost i regularizacija

Izračunavanje neke vrednosti se često matematički može formulisati na različite načine, odnosno može se sprovesti različitim algoritmima. Međutim, neki od tih algoritama se ponašaju *nestabilno*, u smislu da se računska greška akumulira, što dovodi do toga da izračunata vrednost ne aproksimira dobro željenu vrednost, dok drugi algoritmi mogu biti *stabilni*, u smislu da prilikom izračunavanja nema nagomilavanja računske greške, što vodi uspešnoj aproksimaciji željene vrednosti.

Primer 8 *Ukoliko je potrebno izračunati vrednosti integrala*

$$I_n = \int_0^1 \frac{x^n}{x+10} dx, \qquad n = 0, 1, 2, \ldots$$

jedan način je da se to uradi pomoću rekurentne formule

$$I_0 = \ln 1.1, \qquad I_n = \frac{1}{n} - 10 I_{n-1}, \qquad n = 1, 2, \ldots$$

Kako se pri dobijanju nove vrednosti, stara vrednost množi sa 10, računska greška se u svakom koraku uvećava, što dovodi do toga da predloženi metod ne može biti pouzdan u prisustvu računskih grešaka, iako je matematički korektan. Alternativni algoritam je, odsecanjem repa, približno izračunati naredni red:

$$I_n = \sum_{i=1}^{\infty} (-1)^{i+1} \frac{10^{-i}}{n+i}$$

Ovaj algoritam je stabilan i dovodi do preciznih rešenja u malom broju koraka.

Jedna od situacija u kojoj se tipično mogu očekivati visoke relativne greške je situacija u kojoj se oduzimaju bliske vrednosti, koje su obe podložne grešci. Ta vrsta problema se naziva *poništavanje* (eng. *cancelation*). Problem je u tome što računari aproksimiraju realne brojeve pomoću konačnog broja cifara i prilikom oduzimanja bliskih brojeva koji su zapisani sa određenom greškom zaokruživanja, dolazi do poništavanja značajnih cifara, a rezultat ostaje na pozicijama koje su podložne pomenutoj grešci, tako da je vrednost razlike reda veličine greške, što znači da dobijene cifre ne nose informaciju i da je relativna greška velika.

Primer 9 *Neka je data kvadratna jednačina* $ax^2 + bx + c = 0$ *pri uslovu* $a \neq 0$ *i neka je* $b^2 >> |4ac|$. *Jedan algoritam za rešavanje ove jednačine je upotrebom formula*

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

U slučaju da važi $b \geq 0$, *važi da je* $b \approx \sqrt{b^2 - 4ac}$ *i dolazi do velike relativne greške usled poništavanja prilikom računanja vrednosti* x_1. *U slučaju da važi*

$b \leq 0$, *isti problem se javlja prilikom izračunavanja vrednosti* x_2. *Treba imati u vidu da iako je b možda i egzaktno poznato, izračunavanje kvadratnog korena lako dovodi do računske greške. Alternativni algoritam se zasniva na izbegavanju poništavanja. Biće razmotren slučaj kada je* $b \geq 0$. *Alternativni slučaj je analogan. Izračunavanje korena* x_2 *se može sprovesti prema navedenoj formuli bez opasnosti od poništavanja. Za izračunavanje* x_1, *potrebno je eliminisati oduzimanje:*

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \cdot \frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}} = \frac{b^2 - b^2 + 4ac}{2a(-b - \sqrt{b^2 - 4ac})} =$$

$$= \frac{2ac}{a(-b - \sqrt{b^2 - 4ac})} = \frac{c}{ax_2}$$

Problemi stabilnosti nekada nastaju zbog lošeg izbora modela ili lošeg izbora algoritma. Takvi problemi se nekad daju otkloniti relativno jednostavnim transformacijama modela ili algoritma. Međutim, postoje problemi koji su po svojoj prirodi, nezavisno od grešaka izračunavanja, vrlo osetljivi na izmene ulaznih podataka ili parametara. Naime, za male promene ulaznih podataka, odnosno parametara, dolazi do velikih izmena u rešenju problema. Takvi problemi se nazivaju *loše uslovljenim* problemima.

Primer 10 *Neka je potrebno izračunati vrednost kosinusa u okolini vrednosti* $\pi/2$. *Neka je* $x \approx \pi/2$ *i neka je h mala greška u odnosu na vrednost x. Tada važi*

$$R(\cos(x), \cos(x+h)) = \frac{|\cos(x) - \cos(x+h)|}{|\cos(x)|} \approx \frac{|h \sin(x)|}{|\cos(x)|} = |h \tan(x)| \approx \infty$$

Primer 11 *Veliki broj algoritama koji se koriste u svrhe različitih predviđanja (npr. linearna regresija) se zasniva na inverziji matrice podataka ili neke matrice koja je iz nje izvedena. U takvim matricama se lako javljaju visoko korelirane kolone. Na primer, ukoliko podaci uključuju temperaturu i vazdušni pritisak, može se očekivati visoka korelacija. Ekstremni slučaj, kada je koeficijent korelacije između dve kolone 1, odgovara slučaju linearne zavisnosti kolona i u tom slučaju nije moguće invertovati matricu. Čak i u slučaju kada taj koeficijent nije 1, ali je visok, inverzija ne daje smislene rezultate, zato što je za takve matrice, problem loše uslovljen.*

Primer 12 *Problem vrlo srodan prethodnom se javlja i prilikom rešavanja sistema jednačina* $Ax = b$. *Ukoliko se b zameni vrednošću* $b + \Delta b$, *dobija se rešenje* $x + \Delta x$. *Kako važi*

$$A(x + \Delta x) = b + \Delta b$$

važi i

$$Ax + A\Delta x = b + \Delta b$$

Imajući u vidu da važi $Ax = b$, dobija se da je greška rešenja:

$$\Delta x = A^{-1}\Delta b$$

Neka je data matrica

$$A = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1+\varepsilon & 1-\varepsilon \end{bmatrix}$$

Tada važi

$$A^{-1} = \begin{bmatrix} 1-\frac{1}{\varepsilon} & \frac{1}{\varepsilon} \\ 1+\frac{1}{\varepsilon} & -\frac{1}{\varepsilon} \end{bmatrix}$$

Ako je $b = (1,1)^T$, tada važi $x = (1,1)$. Greška u rešenju usled promene vrednosti b je

$$\Delta x = A^{-1}\Delta b = \begin{bmatrix} \Delta b_1 - \frac{1}{\varepsilon}(\Delta b_1 - \Delta b_2) \\ \Delta b_1 + \frac{1}{\varepsilon}(\Delta b_1 - \Delta b_2) \end{bmatrix}$$

Očito, za male vrednosti Δb, greška Δx može biti vrlo velika, ako je ε malo. Na primer, ako važi $\Delta b = (0, 10^{-5})^T$ i $\varepsilon = 10^{-10}$.

Uslovljenost problema se kvantifikuje merom koja se tako i naziva – *uslovljenost*. Neka α predstavlja sve ulazne podatke problema P, a $x(\alpha)$ rešenje tog problema. Tada se uslovljenost $Cond(P)$ problema P, definiše kao odnos relativne greške rešenja i relativne greške ulaznih podataka:

$$Cond(P) = \frac{R(x(\alpha), x(\hat{\alpha}))}{R(\alpha, \hat{\alpha})} = \frac{\|x(\alpha) - x(\hat{\alpha})\|/\|x(\alpha)\|}{\|\alpha - \hat{\alpha}\|/\|\alpha\|}$$

Primer 13 *U slučaju problema izračunavanja funkcije f u tački x (ne treba mešati x kao promenljivu funkcije f sa x kao rešenjem problema u prethodnoj formuli), uslovljenost se može oceniti na sledeći način*

$$Cond(f) = \frac{|f(x) - f(x+\Delta x)|/|f(x)|}{|\Delta x|/|x|} \approx \left| x\frac{f'(x)}{f(x)} \right|$$

U slučaju funkcije e^x, važi $Cond(e^x) \approx |x|$. Ovaj rezultat je sasvim intuitivan. Naime, za velike vrednosti argumenta x, funkcija e^x menja vrednost vrlo brzo za male promene argumenta x.

Primer 14 *U slučaju problema P, rešavanja sistema jednačina $Ax = b$ važi*

$$Cond(P) = \frac{\|A^{-1}b - A^{-1}(b+\Delta b)\|/\|A^{-1}b\|}{\|\Delta b\|/\|b\|} = \frac{\|A^{-1}\Delta b\|/\|A^{-1}b\|}{\|\Delta b\|/\|b\|} = \frac{\|A^{-1}\Delta b\|}{\|\Delta b\|}.\frac{\|Ax\|}{\|x\|}$$

Vrlo često se govori o *uslovljenosti matrice A*. Ona se definiše kao maksimalna vrednost uslovljenosti problema iz prethodnog primera, kada se maksimum uzima po svim nenula vrednostima Δb i x. Prema definiciji matričnih normi, koja će biti data kasnije, važi

$$Cond(A) = \|A^{-1}\| \cdot \|A\|$$

Osnovna strategija za rešavanje loše uslovljenih problema je zamena loše uslovljenog problema drugim, pomoćnim problemom, koji je dobro uslovljen, a koji ima blisko rešenje. Dodatno, pretpostavlja se da je razliku između tih problema moguće kontrolisati menjanjem nekog parametra, tako da rešenje pomoćnog problema teži rešenju polaznog problema kada taj parametar teži nuli. Ovaj pristup rešavanju problema se naziva *regularizacijom* loše uslovljenog problema.

Primer 15 *U slučajevima problema inverzije matrice ili rešavanja sistema linearnih jednačina, postupak regularizacije se najčešće izvodi tako što se loše uslovljena matrica A zameni matricom $A + \lambda I$, za neko malo λ, tako da se rešenja polaznog i regularizovanog problema ne razlikuju mnogo. Može se dokazati da ova vrsta regularizacije poboljšava uslovljenost matrice.*

Glava 3

Aproksimacija funkcija

Pod aproksimacijom funkcije f, podrazumeva se određivanje neke funkcije g, koja je funkciji f bliska u nekom unapred definisanom smislu. Razlozi za aproksimaciju mogu biti različiti. Nekada je to zato što je funkcija f previše komplikovana za direktnu upotrebu, pa se aproksimacijom pojednostavljuje (npr. zamena polinomom), nekada nema poželjna matematička svojstva, pa se zamenjuje funkcijom koja ih ima (npr. konveksnom funkcijom), a najčešći razlog je što su vrednosti funkcije f poznate samo na diskretnom, najčešće konačnom, skupu tačaka, a potrebno je koristiti njene vrednosti na širem domenu.

Za funkciju f može postojati više različitih funkcija koje je aproksimiraju. Kriterijumi kvaliteta aproksimacije mogu biti različiti i voditi različitim najboljim aproksimacijama. Kriterijumi se mogu razlikovati u zavisnosti od toga šta se želi postići. Ukoliko je bitno da funkcija g uglavnom dobro aproksimira funkciju f, ali je prihvatljivo i da se negde više razlikuju, kao kriterijum aproksimacije je moguće koristiti sredinu kvadrata razlike između te dve funkcije. U suprotnom, ako je potrebno da aproksimacija svuda bude dobra, moguće je kao kriterijum koristiti maksimum njihove razlike.

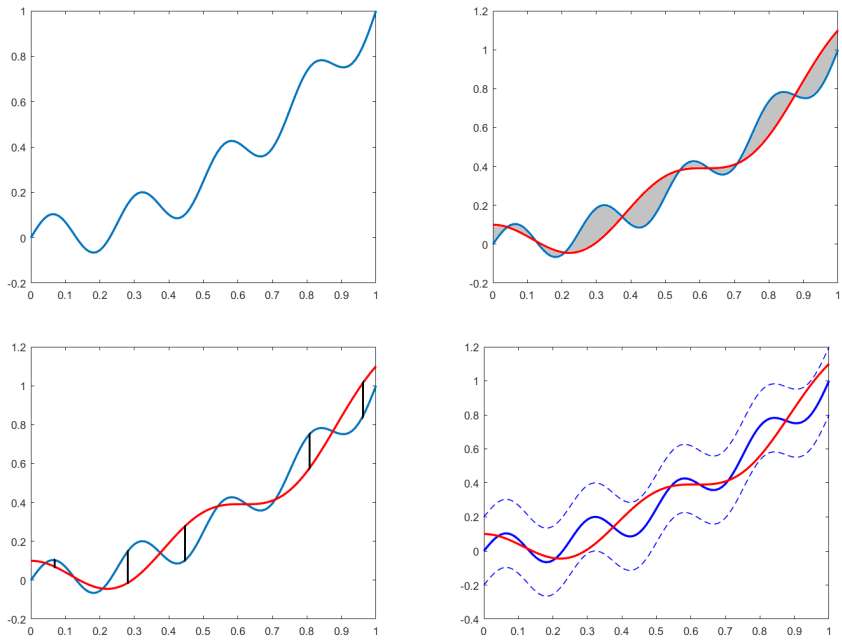
Primer 16 *Ako se funkcija f na intervalu $[a, b]$ aproksimira funkcijom g i ako su obe integrabilne sa kvadratom, jedan izbor kvaliteta aproksimacije je*

$$\|f - g\|_2^2 = \int_a^b (f(x) - g(x))^2 dx$$

U odnosu na ovaj kriterijum, kvalitet aproksimacije se smatra utoliko boljim što je manja površina između funkcija. U slučaju da je funkcija f poznata samo na konačnom skupu tačaka, integral se zamenjuje sumom:

$$\|f - g\|_2^2 = \sum_{i=1}^{n} (f(x_i) - g(x_i))^2$$

Ako je norma razlike mala, to ne znači da funkcije malo odstupaju jedna od druge u svakoj tački, već da je uglavnom tako. Međutim, mogu postojati tačke



Slika 3.1: Prikaz aproksimirane funkcije i njenih aproksimiacija sa istaknutim relevantnim elementima kriterijuma aproksimacije – površinom između funkcija, rastojanjima u diskretnim tačkama i pojasom greške.

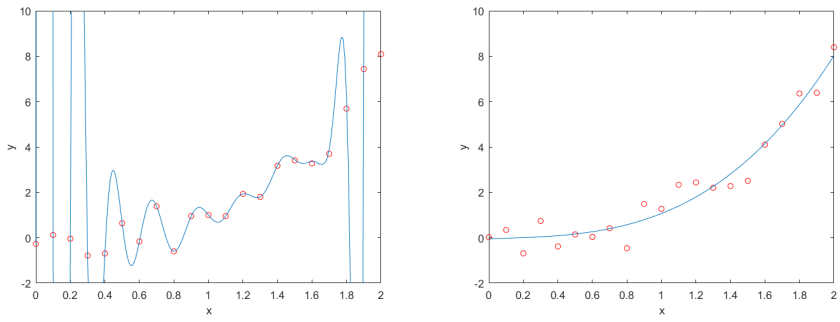
u kojima je odstupanje veliko. Ukoliko je to nepoželjno može se koristiti sledeći kriterijum

$$\|f - g\|_\infty = \sup_{x \in [a,b]} |f(x) - g(x)|$$

i analogno u diskretnom slučaju. Slika 3.1 ilustruje ove kriterijume kvaliteta aproksimacije funkcija.

Funkcija g kojom se vrši aproksimacija najčešće ima neku relativno jednostavnu parametarsku reprezentaciju, poput algebarskog ili trigonometrijskog polinoma, pa se aproksimacija vrši određivanjem vrednosti parametara te reprezentacije. Ovakav pristup je potreban zbog pogodnih matematičkih svojstava ovakvih reprezentacija. S druge strane, nekada se uopšte ne traži eksplicitna reprezentacija aproksimacije, već samo njene vrednosti u specifičnim tačkama.

Primer 17 *Neka je funkcija* $f(x) = x^3$ *izračunata u tačkama intervala* $[0, 2]$ *sa korakom* 0.1*, a onda je na te podatke dodat slučajan šum* ε *koji prati normalnu raspodelu* $\mathcal{N}(0, 0.5)$*. Na slici 3.2 su dati primeri interpolacije Lagranžovim*



Slika 3.2: Ilustracija tačne interpolacije polinomom (levo) i aproksimacije polinomom trećeg stepena.

polinomom i aproksimacije polinomom trećeg stepena koji predstavlja rešenje problema

$$\min_{a_0,a_1,a_2,a_3} \sum_{i=0}^{20} (a_0 + a_1(0.1i) + a_2(0.1i)^2 + a_3(0.1i)^3 - ((0.1i)^3 + \varepsilon_i))$$

Iako interpolacioni polinom potpuno tačno aproksimira funkciju u datim tačkama, očigledno je da on intuitivno ne predstavlja dobru aproksimaciju, dok kubna aproksimacija intuitivno deluje dobro, iako pravi grešku u većini tačaka.

Pristupi aproksimaciji funkcija su raznovrsni i samo određivanje aproksimacije se u različitim pristupima može vršiti na različite načine. Neki pristupi su zasnovani na teoriji Hilbertovih prostora, neki na interpolaciji, neki na matematičkoj optimizaciji, itd.

3.1 Primeri problema aproksimacije funkcija

U nastavku su dati primeri praktičnih problema koji se mogu rešiti tehnikama aproksimacije funkcija, koje su prikazane kasnije u ovoj glavi.

Primer 18 *Primer 6, predstavlja upravo primer aproksimacije funkcije. Funkcija rizika od bolesti, koja je poznata samo na konačnom skupu tačaka $\mathcal{D}$ se aproksimira linearnom funkcijom nekih promenljivih:*

$$g(x, \alpha) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i$$

Pritom, kao osnovni kriterijum aproksimacije, koristi se sredina kvadrata razlike između dve funkcije u tačkama u kojima je vrednost funkcije f poznata

Slika 3.3: Polazna slika, zamućena slika i slika rekonstruisana inverzijom matrice zamućivanja.

(zapis je u duhu nove notacije):

$$\min_{\alpha} \sum_{i=1}^{N} (g(x_i, \alpha) - f(x_i))^2$$

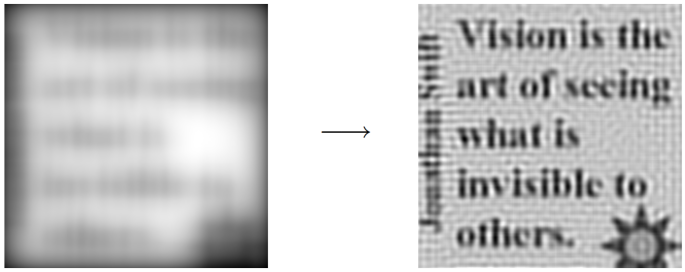
Pritom, naglašeno je da izmene kriterijuma aproksimacije vode aproksimacijama različitih svojstava.

Primer uklanjanja šuma iz signala se takođe može smatrati primerom aproksimacije funkcije, pri čemu se aproksimirana funkcija smatra diskretnom i predstavljena je neposredno svojim vrednostima. Sličan primer naveden je u nastavku.

Primer 19 *Neka je data slika X dimenzija $M \times N$ koja je predstavljena vektorom x dimenzije $MN \times 1$ i neka je od nje dobijena nova slika $y = Ax$ operacijom zamućivanja primenom matrice A. Postavlja se pitanje, ukoliko je poznata matrica A, kako rekonstruisati polaznu sliku x. Imajući u vidu da su zamućena i polazna slika istih dimenzija, matrica A mora biti kvadratna. Prva ideja je izračunati $x = A^{-1}y$. Uprkos tome što rešenje matematički ima smisla, rezultat može biti potpuno neprepoznatljiv, kao na slici 3.3. Razlog za loš ishod prethodnog postupka je vrlo loša uslovljenost matrice A, što dovodi do toga da razlike nastale u zaokruživanju prilikom izračunavanja matrice y, potpuno poremete rekonstruisanu sliku. Odavde se zaključuje da je rekonstrukcija zamućene slike loše uslovljen problem, pa je stoga potrebno izvršiti regularizaciju. Problem koji se rešava umesto polaznog je*

$$\min_{x} \|Ax - y\|^2 + \lambda \left(\sum_{i=1}^{M} \sum_{j=1}^{N-1} (x_{ij} - x_{i,j+1})^2 + \sum_{i=1}^{M-1} \sum_{j=1}^{N} (x_{ij} - x_{i+1,j})^2 \right)$$

gde uslovi regularizacije obezbeđuju da se vrednosti susednih piksela ne razlikuju mnogo. Za neku vrednost parametra λ moguće je dobiti rešenje kao na slici 3.4.



Slika 3.4: Zamućena slika i slika rekonstruisana uz korišćenje regularizacije.

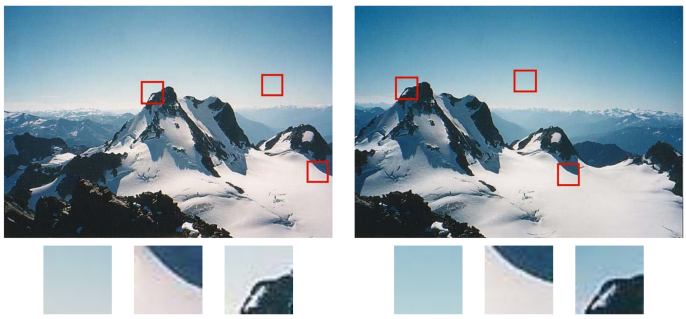
Primer 20 *Neka je od N slika koje predstavljaju različite delove iste scene i koje se međusobno preklapaju potrebno konstruisati kolaž koji predstavlja ukupnu sliku scene. Pritom je očekivano da se slike ne nadovezuju savršeno, usled malih izmena u tački sa koje je slikano, uglu fotoaparata, promena osvetljenosti i slično.*

Kako bi se problem poravnavanja slika modelovao, potrebno je prvo ustanoviti određene pretpostavke o nesavršenosti korespodencije među njihovim delovima. Neka se te razlike mogu objasniti kompozicijom rotacije za ugao θ, translacije za vektor $[u, v]$ i skaliranja za faktor s. Prve dve transformacije su u vezi sa pomerajem kamere između dva fotografisanja, a drugi u vezi sa faktorom uvećanja objektiva. Matrica transformacije nad homogenim koordinatama je:

$$G = \begin{bmatrix} a & -b & u \\ b & a & v \\ 0 & 0 & 1 \end{bmatrix}$$

gde važi $a = s\cos\theta$, $b = s\sin\theta$ i $s = \sqrt{a^2 + b^2}$. Potrebno je naći po jednu transformaciju za svaku od slika, tako da su nakon primenjenih transformacija, slike uklopljene u kolaž.

Algoritmi koji se bave obradom slika, često se oslanjaju na korišćenje određenih detalja (eng. feature*) na slikama, poput pojedinačnih prepoznatljivih delića slike, poput karakterističnih ćoškova, ivica, itd. Na slikama koje predstavljaju preklapajuće delove iste scene, nekada je moguće i uparivati detalje tih slika, ako su poznate njihove pozicije na tim slikama. Pritom, neki detalji su laki za uparivanje, a neki ne, kao što se vidi u slučaju detalja izabranih na slici 3.5. Pritom, postoje gotovi algoritmi za uparivanje detalja dve slike. Imajući u vidu postojanje takvih algoritama, može se pretpostaviti da je ulazni podatak za sliku i skup lokacija pomenutih detalja $\{x_{ij}|j = 1, \ldots M\}$ (pri čemu ako neki detalj nije detektovan na slici, lokacija je nedefinisana), kao i da je za svake dve slike i i j dat skup indeksa $F(i, j)$ detalja koji su uspešno upareni. Onda se kolaž*



Slika 3.5: Nekoliko odgovarajućih detalja na dve različite slike.

konstruiše rešavanjem optimizacionog problema:

$$\min_{\mathbf{a},\mathbf{b},\mathbf{u},\mathbf{v}} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k \in F(i,j)} \|G_i x_{ik} - G_j x_{jk}\|^2$$

gde je G_i matrica transformacije sa parametrima (a_i, b_i, u_i, v_i). Kako bi rešenje bilo jedinstveno, jedna od transformacija se postavlja na jediničnu. Faze procesa su prikazane na slici 3.6.

Primer 21 *Neka je dato nekoliko referentnih tačaka poznate lokacije u odnosu na koje je moguće meriti rastojanja, a na osnovu kojih je potrebno ustanoviti lokaciju tačke sa koje je merenje izvršeno. Jedan takav kontekst se odnosi na pozicioniranje u odnosu na GPS satelite. Pojednostavljen, dvodimenzionalni primer dat je na slici 3.7. Neka su (u, v, w) trenutne koordinate GPS uređaja koje je potrebno izračunati, a (p_i, q_i, r_i), za $i = 1, \ldots, N$ precizno poznate koordinate GPS satelita. Ipak, rastojanja ρ_i do satelita nisu nužno sasvim precizna zbog greške u merenju.*

Za svaki od satelita, koordinate (u, v, w) treba da zadovoljavaju jednačine:

$$\sqrt{(u-p_i)^2 + (v-q_i)^2 + (w-r_i)^2} = \rho_i \qquad i = 1, \ldots, N$$

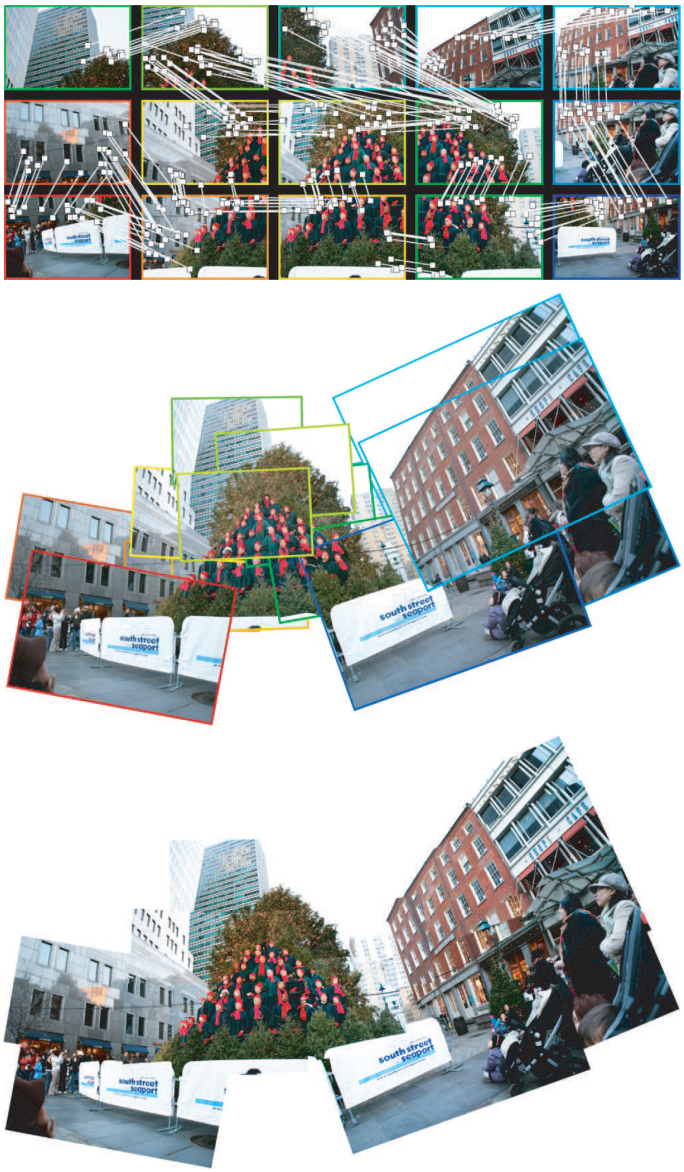
Jedan način određivanja koordinata je rešavanje sledećeg optimizacionog problema:

$$\min_{u,v,w} \sum_{i=1}^{n} (\sqrt{(u-p_i)^2 + (v-q_i)^2 + (w-r_i)^2} - \rho_i)^2$$

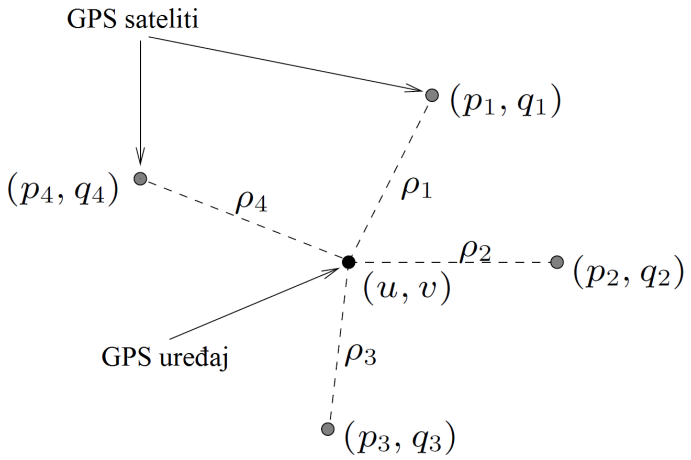
Za razliku od prethodnih primera, izraz pod kvadratom u ovom slučaju nije linearan, što će učiniti njegovo rešavanje izazovnijim.

3.2 Aproksimacija u Hilbertovim prostorima

Vektorski prostor koji je kompletan (svaki Košijev niz elemenata vektorskog prostora konvergira elementu tog prostora) u odnosu na metriku indukovanu



Slika 3.6: Faze kreiranja kolaža: uparivanje svojstava, pronalaženje optimalnog poravnanja i kreiranje finalne slike.



Slika 3.7: Određivanje lokacije na osnovu rastojanja do GPS satelita.

skalarnim proizvodom:

$$d(x, y) = \|x - y\| = \sqrt{(x - y) \cdot (x - y)}$$

se naziva *Hilbertovim prostorom.*

Primer 22 *Prostor $\mathbb{R}^n$ je Hilbertov prostor. Takođe, prostor $\mathcal{L}_2[a, b]$ funkcija koje su na intervalu $[a, b]$ integrabilne sa kvadratom, je Hilbertov prostor.*

Za sistem vektora $\{e_i | i \in \mathbb{N}\}$, kažemo da je ortonormiran ukoliko važi

$$e_i \cdot e_j = \begin{cases} 1, & i = j \\ 0 & i \neq j \end{cases} \qquad i, j \in \mathbb{N}$$

Neka je $\{e_i | i \in \mathbb{N}\}$ ortonormiran sistem vektora Hilbertovog prostora $\mathcal{H}$. Koeficijenti $x \cdot e_i$ nazivaju se *Furijeovim koeficijentima* vektora $x \in \mathcal{H}$ u odnosu na prethodni niz, a red

$$\sum_{i=1}^{\infty} (x \cdot e_i) e_i$$

se naziva *Furijeov red* vektora x u odnosu na dati niz.

Teorema 1 *Za ortonormirani sistem $\{e_i | i \in \mathbb{N}\}$ u Hilbertovom prostory $\mathcal{H}$, sledeća tvrđenja su ekvivalentna:*

- *Za svako $x \in \mathcal{H}$ i svako $\varepsilon > 0$, postoje skalari $\lambda_1, \lambda_2, \ldots, \lambda_n$, takvi da važi $\|x - \sum_{i=1}^{n} \lambda_i e_i\| < \varepsilon$.*
- *Za svako $x \in \mathcal{H}$ važi $\sum_{i=1}^{\infty} (x \cdot e_i) e_i = x$ (pri čemu se podrazumeva konvergencija u smislu metrike prostora $\mathcal{H}$)*

- *Za svako* $x \in \mathcal{H}$ *važi* $\sum_{i=1}^{\infty}(x \cdot e_i)^2 = \|x\|^2$ *(Parsevalova jednakost)*
- *Ako je vektor* $x \in \mathcal{H}$ *takav da je* $x \cdot e_i = 0$ *za svako* $i \in \mathbb{N}$*, onda važi* $x = 0$*.*

Teorema 2 *Neka je* f *element Hilbertovog prostora* $\mathcal{H}$ *i neka je* $\mathcal{H}'$ *njegov potprostor čiju bazu čine elementi* $\{g_1, g_2, \ldots, g_n\}$*. Postoji element najbolje aproksimacije* $g^* = \sum_{i=1}^{n} c_i^* g_i \in \mathcal{H}'$*, takav da važi*

$$\left\| f - \sum_{i=1}^{n} c_i^* g_i \right\| = \inf_{c_1,\ldots,c_n} \left\| f - \sum_{i=1}^{n} c_i g_i \right\|$$

Dodatno, važi da je $(f - g^*) \cdot x = 0$ *za sve* $x \in \mathcal{H}'$ *ako i samo ako je* g^* *element najbolje aproksimacije za* f *iz* $\mathcal{H}'$*.*

Drugim rečima, element najbolje aproksimacije za f je njegova ortogonalna projekcija na prostor $\mathcal{H}'$. Na osnovu ove teoreme sledi da se koeficijenti najbolje aproksimacije mogu odrediti iz sistema:

$$\sum_{i=1}^{n} c_i(g_i \cdot g_j) = f \cdot g_j \qquad j = 1, \ldots, n \tag{3.1}$$

Ukoliko je baza $g_1, \ldots, g_n$ ortogonalna, svi skalarni proizvodi $g_i \cdot g_j$ su jednaki nuli ako $i \neq j$, tako da u tom slučaju nije potrebno rešavati sistem jednačina, već je dovoljno izračunati skalarne proizvode i izraziti koeficijente c_i iz dobijenih jednakosti u kojima učestvuje po jedan koeficijent c_i. Zato se pri aproksimaciji često koriste ortogonalni sistemi, poput sistema Ležandrovih polinoma, sistema Čebišovljevih polinoma, trigonometrijskog sistema sinusa i kosinusa različitih frekvencija, itd.

3.3 Srednjekvadratna aproksimacija

Ako se za Hilbertov prostor $\mathcal{H}$ uzme prostor funkcija $\mathcal{L}_2[a, b]$, odnosno prostor funkcija integrabilnih sa kvadratom na intervalu $[a, b]$, u kome je norma definisana integralom

$$\|f\|^2 = \int_a^b f^2(x)dx \qquad f \in \mathcal{L}_2[a, b]$$

onda se element najbolje aproksimacije naziva elementom *najbolje srednjekvadratne aproksimacije*. Ako je funkcija f definisana samo na konačnom skupu tačaka $x_0, \ldots, x_m$ iz intervala $[a, b]$, integral se zamenjuje sumom, pa su skalarni proizvod i odgovarajuća norma dati jednakostima:

$$f \cdot g = \sum_{i=1}^{m} f(x_i)g(x_i)$$

$$\|f\|^2 = f \cdot f = \sum_{i=1}^{m} f^2(x_i)$$

Metoda koja odgovara srednjekvadratnoj aproksimaciji u ovom slučaju se naziva *metodom najmanjih kvadrata* (eng. *least squares method*). Jednačine 3.1 za ovaj konkretan izbor skalarnog proizvoda dobijaju sledeći oblik:

$$\sum_{i=1}^{n} c_i \sum_{k=1}^{m} g_i(x_k) g_j(x_k) = \sum_{k=1}^{m} f(x_k) g_j(x_k) \qquad j = 1, \ldots, n$$

Razmenom redosleda sumiranja, dobija se:

$$\sum_{k=1}^{m} g_j(x_k) \left(\sum_{i=1}^{n} c_i g_i(x_k) \right) = \sum_{k=1}^{m} f(x_k) g_j(x_k) \qquad j = 1, \ldots, n$$

Prelaskom na matričnu notaciju

$$\underbrace{\begin{bmatrix} g_1(x_1) & \cdots & g_1(x_m) \\ \vdots & \ddots & \vdots \\ g_n(x_1) & \cdots & g_n(x_m) \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} g_1(x_1) & \cdots & g_n(x_1) \\ \vdots & & \vdots \\ g_1(x_m) & \cdots & g_n(x_m) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} g_1(x_1) & \cdots & g_1(x_m) \\ \vdots & \ddots & \vdots \\ g_n(x_1) & \cdots & g_n(x_m) \end{bmatrix}}_{A^T} \underbrace{\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_m) \end{bmatrix}}_{b}$$

i uvođenjem naznačenih oznaka, dobijaju se takozvane *jednačine normale* (eng. *normal equations*):

$$A^T A x = A^T b \tag{3.2}$$

odnosno

$$x = (A^T A)^{-1} A^T b$$

koje predstavljaju rešenje optimizacionog problema

$$\min_x \|Ax - b\|^2 \tag{3.3}$$

U konkretnom slučaju problema 3.3, rešenje se moglo izvesti i na drugačije načine. Veličina koja se minimizuje se može zapisati na sledeći način:

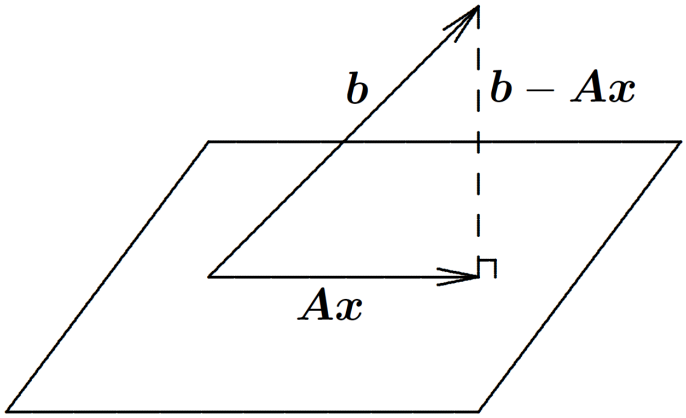
$$\|Ax - b\|^2 = (Ax - b)^T (Ax - b) = b^T b - 2x^T A^T b + x^T A^T A x$$

gde se podrazumeva euklidska norma $\|\cdot\|_2$. Izjednačavanjem gradijenta po x sa nulom dobija se:

$$2A^T A x - 2A^T b = 0$$

što daje jednačine 3.2.

Treći način izvođenja istog rešenja je geometrijski. Matrica A je dimenzija $m \times n$, pri čemu u primenama obično važi $m > n$. Prostor kolona marice A,



Slika 3.8: Projekcija vektora b na prostor kolona matrice A.

odnosno prostor linearnih kombinacija kolona matrice A se može predstaviti kao skup vektora Ax za različite vektore x. Ukoliko vektor b leži u prostoru matrice A, onda postoji takvo x, da važi $Ax = b$. Međutim, ako je $m > n$ to nije moguće. U tom slučaju poželjno je naći ortogonalnu projekciju vektora b na prostor kolona matrice A, kao na slici 3.8. Da bi projekcija bila ortogonalna, vektor $b - Ax$ mora biti ortogonalan na prostor kolona matrice A, pa i na same njene kolone, odnosno mora važiti

$$A^T(b - Ax) = 0$$

što daje jednačine 3.2.

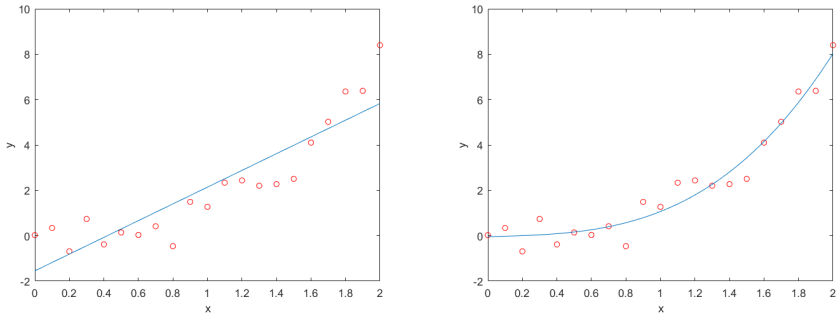
Matrica $(A^TA)^{-1}A^T$ je *Mur-Penrouzov pseudoinverz* matrice A. Naziv pseudoinverz odražava činjenicu da važi $(A^TA)^{-1}A^TA = I$.

Metod najmanjih kvadrata nalazi primenu u mnogim domenima. Jedna od najvažnijih je u statistici, konkretno vezano za linearnu regresiju. Model linearne regresije se izražava narednom matričnom jednačinom

$$y = Xw + \varepsilon$$

pri čemu je y vektor dimenzije m, X, matrica dimenzija $m \times n$, w vektor dimenzije n i ε slučajni vektor dimenzije m. Takođe, matrica X se u praksi često proširuje prvom kolonom jedinica, kako bi regresioni model imao slobodan član.

Primer 23 *U primeru 17 je diskutovana aproksimacija funkcije x^3 čije su vrednosti poremećene normalno raspodeljenim šumom. Linearnom regresijom je moguće odrediti aproksimaciju tih podataka pravom ili polinomom. Pri aprok-*



Slika 3.9: Slika levo prikazuje aproksimaciju tačaka pravom, a slika desno polinomom trećeg stepena.

simaciji pravom, koriste se naredne matrice:

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 0.01 \\ 1 & 0.02 \\ \vdots & \vdots \\ 1 & 2 \end{bmatrix} \qquad b = \begin{bmatrix} 0.027 \\ 0.345 \\ -0.689 \\ \vdots \\ 8.397 \end{bmatrix}$$

a pri aproksimaciji kubnim polinomom, koriste se naredne matrice

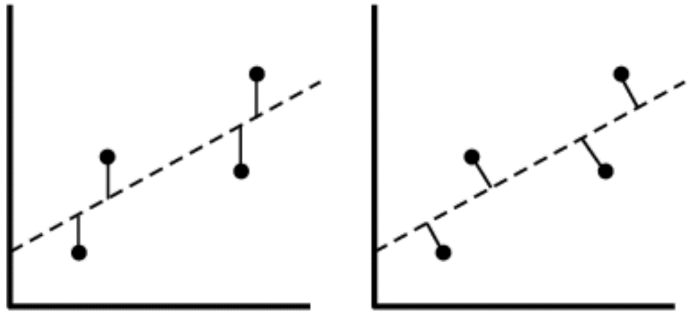
$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 10^{-2} & 10^{-4} & 10^{-6} \\ 1 & 2 \cdot 10^{-2} & 4 \cdot 10^{-4} & 8 \cdot 10^{-6} \\ \vdots & \vdots & & \\ 1 & 2 & 4 & 8 \end{bmatrix} \qquad b = \begin{bmatrix} 0.027 \\ 0.345 \\ -0.689 \\ \vdots \\ 8.397 \end{bmatrix}$$

Jedna od čestih grešaka je razumevanje da linearna regresija, odnosno metod najmanjih kvadrata pronalaze hiperravan koja minimizuje najkraća rastojanja, odnosno rastojanja duž normalnog pravca do tačaka (X_i, y_i), gde je X_i i-ta vrsta matrice X. Zapravo, minimizuju se rastojanja duž y ose, kao što je ilustrovano slikom 3.10.

Za ocenu parametara w se kaže da je linearna, ukoliko je oblika $Ay + a$ za neku matricu A i vektor a odgovarajućih dimenzija.

Teorema 3 (Gaus-Markov) *Ukoliko važi $E(\varepsilon) = 0$ i $Cov(\varepsilon) = \sigma^2 I$, za konstantno $\sigma^2 > 0$, onda za ocenu $\hat{w} = (X^T X)^{-1} X^T y$ važi*

$$E(\hat{w}) = w \text{ i } Cov(\hat{w}) = \sigma^2 (X^T X)^{-1}$$



Slika 3.10: Metod najmanjih kvadrata minimizuje sumu kvadrata rastojanja na levoj, a ne na desnoj slici.

Takođe, za svaku nepristrasnu linearnu ocenu $\tilde{w}$ parametara w, važi

$$\sum_{i=1}^{n}(w_i - \hat{w}_i)^2 \leq \sum_{i=1}^{n}(w_i - \tilde{w}_i)^2$$

Drugim rečima, ocena parametara w, koja se dobija metodom najmanjih kvadrata je najbolja linearna nepristrasna ocena tih parametara, pri čemu se kvalitet meri u odnosu na srednjekvadratno odstupanje ocene od pravih vrednosti koeficijenata.

Primer 24 *Primer 6, koji se tiče ocene rizika od bolesti, predstavlja upravo problem linearne regresije i rešava se metodom najmanjih kvadrata.*

Ukoliko je matrica A^TA loše uslovljena, što je moguće ukoliko su kolone ili vrste matrice A visoko korelisane, problem najmanjih kvadrata je i sam loše uslovljen. U tom slučaju se pribegava regularizaciji i umesto polaznog problema

$$\min_x \|Ax - b\|^2$$

rešava se regularizovani problem

$$\min_x \|Ax - b\|^2 + \lambda\|x\|^2$$

Ova vrsta regularizacije se naziva *Tihonovljevom regularizacijom* ili *grebenom regularizacijom* (eng. *ridge regularization*). Analogno prethodnom slučaju važi:

$$\|Ax - b\|^2 + \lambda\|x\|^2 = (Ax - b)^T(Ax - b) + \lambda x^T x$$

a postavljanjem gradijenta po x na nulu dobija se:

$$A^TAx - A^Tb + \lambda x = 0$$

odnosno

$$x = (A^T A + \lambda I)^{-1} A^T b$$

Ovim se rešava problem loše uslovljenosti. U statističkim terminima, data ocena parametara je ocena *grebene regresije* (eng. ridge regression). Ona nema svojstvo nepristrasnosti, ali greška ocene može biti manja nego u slučaju standardnog problema. Ovo zapažanje ne protivreči Gaus-Markovljevoj teoremi, pošto ona govori o optimalnosti u klasi nepristrasnih linearnih ocena, dok prethodna ocena nije nepristrasna.

Primer 25 *Imajući u vidu da metod najmanjih kvadrata pronalazi minimum srednjekvadratne greške, koji se dostiže kada se vrednosti aproksimacije poklapaju sa vrednostima aproksimirane funkcije, ukoliko se za funkciju kojom se vrši aproksimacija u primeru 17 upotrebi polinom stepena 20, rezultat bi trebalo da bude baš interpolacioni polinom, pošto je funkcija zadata pomoću 21 tačke. Neka je x vektor ekvidistantnih podeoka na intervalu $[0, 2]$ sa rastojanjem 0.5 i neka x^i označava vektor čija je j-ta koordinata broj x^i_j. U tom slučaju, matrica A ima za kolone vektore $x^0, x^1, x^2, \ldots, x^{20}$. Vektori x^i su, očekivano, među sobom visoko korelisani, što znači da je matrica A loše uslovljena. Zbog toga dolazi do velike greške u izračunavanju i polinom dobijen metodom najmanjih kvadrata predstavlja vrlo lošu aproksimaciju, kao što je prikazano na slici 3.11. Međutim, ako se primeni regularizacija, već za male vrednosti parametra λ, dobijeni polinom predstavlja mnogo bolju aproksimaciju. Sa povećavanjem vrednosti regularizacionog parametra, dobijeni polinom počinje sve više da odstupa od tačaka koje treba da aproksimira.*

Na sličan način, kao u slučaju Tihonovljeve regularizacije, mogu se izvesti i rešenja nekih prethodno razmatranih problema.

Primer 26 *U primeru 3, koji se tiče uklanjanja šuma iz signala, javlja se problem*

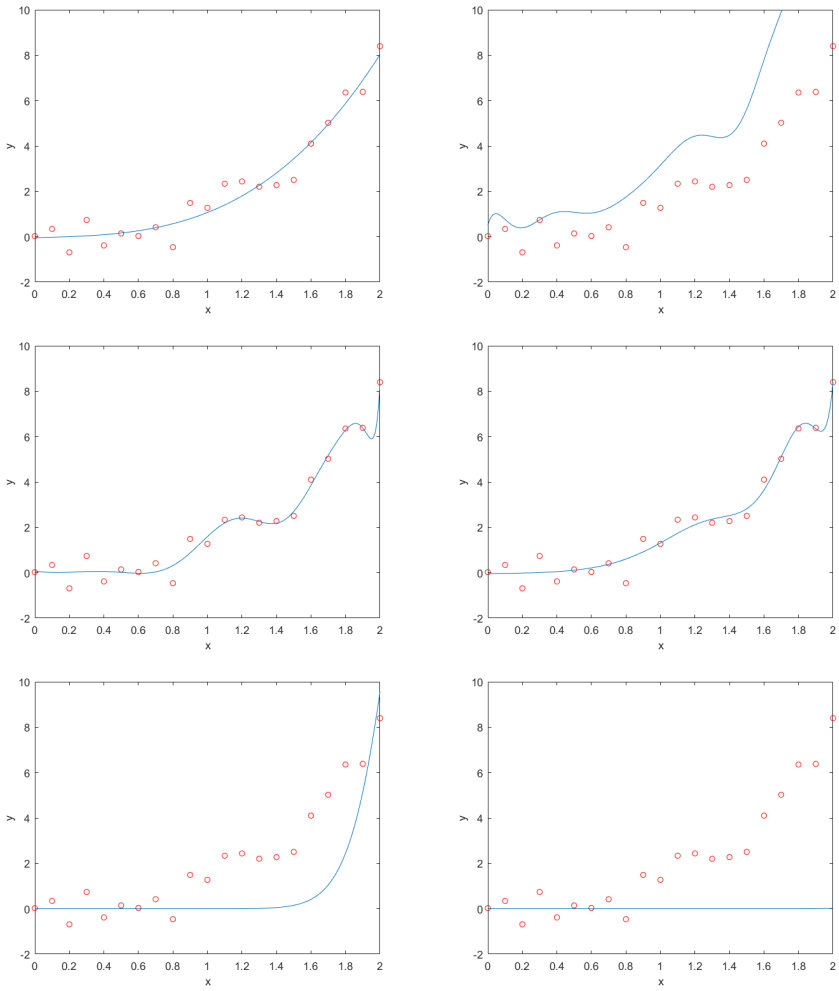
$$\min_x \|x - y\|^2 + \lambda \sum_{i=1}^{n-1} (x_i - x_{i+1})^2$$

Ukoliko se sa D označi matrica

$$\begin{bmatrix} 1 & -1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & -1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & -1 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & -1 \end{bmatrix}$$

dati problem se može zapisati kao

$$\min_x \|Ix - y\|^2 + \|\sqrt{\lambda} Dx - 0\|^2$$



Slika 3.11: Na slikama su sleva nadesno odozgo nadole prikazane aproksimacije metodom najmanjih kvadrata korišćenjem kubnog polinoma i polinoma stepena 20 bez regularizacije, kao i polinomom stepena 20 sa korišćenjem regularizacije ($\lambda = 10^{-4}, 1, 10^{9}, 10^{15}$).

odnosno

$$\min_x \left\| \begin{bmatrix} I \\ \sqrt{\lambda} D \end{bmatrix} x - \begin{bmatrix} y \\ 0 \end{bmatrix} \right\|^2$$

Rešenje se može izvesti na način analogan ranijim izvođenjima i glasi:

$$x = (I + \lambda D^T D)^{-1} y$$

Primer 27 *U primeru 19, koji se tiče rekonstrukcije zamućene slike, javlja se problem:*

$$\min_x \|Ax - y\|^2 + \lambda \left(\sum_{i=1}^{M} \sum_{j=1}^{N-1} (x_{ij} - x_{i,j+1})^2 + \sum_{i=1}^{M-1} \sum_{j=1}^{N} (x_{ij} - x_{i+1,j})^2 \right)$$

Nalik prethodnom problemu, ako se uvede oznaka D_h za matricu

$$\begin{bmatrix} I & -I & 0 & \dots & 0 & 0 & 0 \\ 0 & I & -I & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & I & -I & 0 \\ 0 & 0 & 0 & \dots & 0 & I & -I \end{bmatrix}$$

odgovarajućih dimenzija i D_v za matricu

$$\begin{bmatrix} D & 0 & \dots & 0 \\ 0 & D & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & D \end{bmatrix}$$

odgovarajućih dimenzija, gde je D matrica iz prethodnog primera, problem se može zapisati u obliku

$$\min_x \|Ax - y\|^2 + \|\sqrt{\lambda} D_v x - 0\|^2 + \|\sqrt{\lambda} D_h x - 0\|^2$$

odnosno

$$\min_x \left\| \begin{bmatrix} A \\ \sqrt{\lambda} D_v \\ \sqrt{\lambda} D_h \end{bmatrix} x - \begin{bmatrix} y \\ 0 \\ 0 \end{bmatrix} \right\|^2$$

a rešenje je:

$$x = (A^T A + \lambda D_v^T D_v + \lambda D_h^T D_h)^{-1} A^T y$$

Primer 28 *U primeru 20, koji se tiče pravljenja kolaža od fotografija, javlja se problem:*

$$\min_{\mathbf{a},\mathbf{b},\mathbf{u},\mathbf{v}} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{M} \|G_i x_{ik} - G_j x_{jk}\|^2$$

Kao i u prethodnim slučajevima, ključni problem je naći pogodnu reprezentaciju problema u vidu problema najmanjih kvadrata.

Primer 29 *U primeru 21, koji se tiče određivanja pozicije na osnovu GPS satelita, javlja se problem:*

$$\min_{u,v,w} \sum_{i=1}^{n} (\sqrt{(u-p_i)^2 + (v-q_i)^2 + (w-r_i)^2} - \rho_i)^2$$

Ovaj problem ne predstavlja problem najmanjih kvadrata zbog svoje nelinearnosti. Imajući u vidu da su sateliti izrazito daleko, uz pretpostavku poznavanja pozicije (u_0, v_0, w_0) u prethodnom trenutku, euklidsko rastojanje se može linearizovati u okolini te tačke

$$\sqrt{(u-p_i)^2 + (v-q_i)^2 + (w-r_i)^2} =$$

$$\sqrt{(u_0 + \Delta u - p_i)^2 + (v_0 + \Delta v - q_i)^2 + (w_0 + \Delta w - r_i)^2} =$$

$$\sqrt{(u_0-p_i)^2 + (v_0-q_i)^2 + (w_0-r_i)^2} + \frac{(u_0-p_i)\Delta u + (v_0-q_i)\Delta v + (w_0-r_i)\Delta w}{\sqrt{(u_0-p_i)^2 + (v_0-q_i)^2 + (w_0-r_i)^2}}$$

Ukoliko važi $x = (\Delta u, \Delta v, \Delta w)^T$, kao i

$$b_i = \rho_i - \sqrt{(u_0-p_i)^2 + (v_0-q_i)^2 + (w_0-r_i)^2}$$

$$a_{i1} = \frac{u_0 - p_i}{\sqrt{(u_0-p_i)^2 + (v_0-q_i)^2 + (w_0-r_i)^2}}$$

$$a_{i2} = \frac{v_0 - q_i}{\sqrt{(u_0-p_i)^2 + (v_0-q_i)^2 + (w_0-r_i)^2}}$$

$$a_{i3} = \frac{w_0 - r_i}{\sqrt{(u_0-p_i)^2 + (v_0-q_i)^2 + (w_0-r_i)^2}}$$

rešenje polaznog problema se može aproksimirati rešenjem problema

$$\min_x \|Ax - b\|^2$$

Primer 30 *Pretpostavimo da gledaoci ocenjuju filmove na nekom od sajtova na kojima ih je moguće gledati. Gledaoci ocenjuju one filmove koje su gledali, a trebalo bi da postoji sistem koji procenjuje koliko bi im se svideo neki drugi film i na osnovu toga je u stanju da daje preporuke. Neka je jedan takav skup podataka dat tabelom 3.1. Pored navedenih podataka, nekada mogu biti dostupne dodatne informacije o filmovima, poput procena nivoa neke vrste sadržaja, koje se mogu smatrati atributima. Na primer, kao što je prikazano u tabeli 3.2. U slučaju da su dati atributi za svaki od filmova, moguće je konstruisati regresione modele kojima se za svaku od N osoba, na osnovu atributa, određuje ocena koju bi osoba dala nekom filmu. Neka je Y matrica ocena svih osoba za sve filmove, sa posebnim simbolom za nedostajuću vrednost. Neka je X matrica vrednosti atributa filmova. Neka za matricu A, A_i označava i-tu kolonu, a A^i i-tu vrstu.*

Film	Ana	Jovan	Marko	Dragana
Uspavana dolina	5	4	2	5
Panov lavirint	5	?	1	2
Odbegla mlada	1	5	?	5
Terminator 2	?	3	5	1
Hobit 3	1	1	5	?

Tabela 3.1: Ocene koje gledaoci daju filmovima.

Film	Akcija	Romantika
Uspavana dolina	4	3
Panov lavirint	4	2
Odbegla mlada	2	5
Terminator 2	5	1
Hobit 3	5	2

Tabela 3.2: Atributi filmova.

Neka je $R(i)$ skup indeksa filmova koje je ocenio i-ti gledalac. Tada je moguće odrediti modele za svaku od osoba rešavanjem narednog problema:

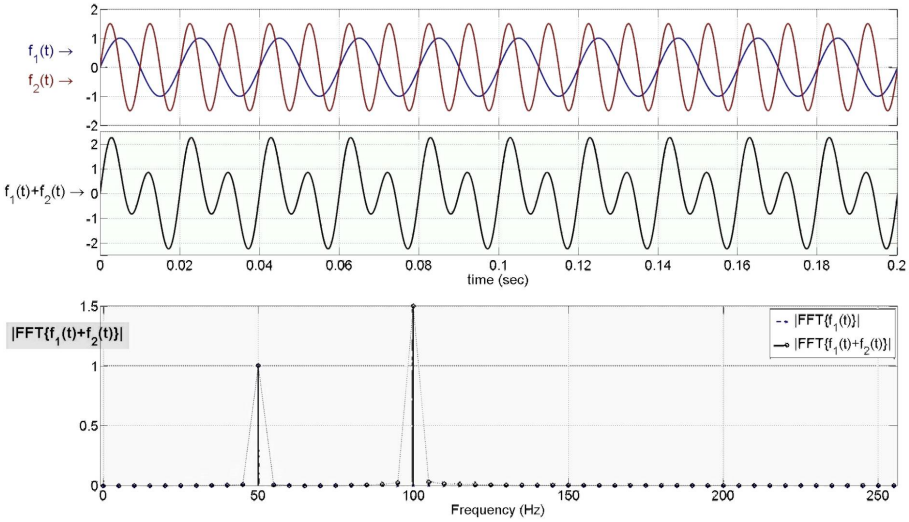
$$\min_{w} \sum_{i=1}^{N} \sum_{j \in R(i)} (X^j w_i - Y_{ij})^2$$

koji se može rešiti metodom najmanjih kvadrata. Predviđanje ocene i-tog gledaoca za j-ti film je dato izrazom $X^j w_i$.

Problem je nešto izazovniji ukoliko nisu date vrednosti atributa filmova. Ovo je takođe realistična pretpostavaka, pošto do takvih atributa ne mora biti trivijalno doći. S druge strane, neka je za svakog od gledalaca poznato koliko voli koju vrstu filmova (poput akcionih ili romantičnih), recimo ponovo tako što će dati ocene od 1 do 5. Date ocene predstavljaju inicijalne vrednosti odgovarajućih vektora w. Tu vrstu informacije nije teško dobiti – recimo tako što će gledaoci popuniti kratak upitnik prilikom registrovanja. Tada se polazne procene atributa mogu dobiti rešavanjem narednog probelma:

$$\min_{X} \sum_{i=1}^{N} \sum_{j \in R(i)} (X^j w_i - Y_{ij})^2$$

Ukupan problem se rešava iteriranjem između ocena parametara w i atributa X. Takođe, ovaj iterativni algoritam je mogao da bude inicijalizovan nasumice, a ne nužno izraženim preferencama gledalaca, ali bi tada smisao tih atributa bio nejasan, čak i ako algoritam pruža dobro predviđanje.



Slika 3.12: Ilustracija Furijeove transformacije. Na dnu se nalazi prikaz koeficijenata Furijeove transformacije.

3.4 Furijeova transformacija

Jedan od tipičnih primera Furijeovog reda je trigonometrijski Furijeov red, koji počiva na sistemu sinusa i kosinusa različitih frekvencija ($\cos(kx)$ i $\sin(kx)$ za $k = 0, 1, \ldots$). Štaviše, periodične funkcije pod određenim uslovima mogu biti proizvoljno dobro aproksimirane linearnim kombinacijama funkcija iz ovog sistema. Furijeovi koeficijenti funkcije omogućavaju analizu signala u odnosu na frekvencije koje su u njemu zastupljene, odnosno *spektar signala*. *Furijeova transformacija* upravo omogućava prevođenje reprezentacije funkcije iz vremenskog domena u frekvencijski domen. Obrnuto se postiže *inverznom Furijeovom transformacijom*. Ilustracija je data na slici 3.12.

U zavisnosti od svojstava funkcije, nad funkcijom se mogu definisati različite vrste Furijeovih transformacija – *razvoj u Furijeov red*, *neprekidna Furijeova transformacija* i *diskretna Furijeova transformacija*. U literaturi se pod rečju Furijeova transformacija najčešće podrazumeva neprekidna Furijeova transformacija. Dodatno, taj izraz se najčešće ne koristi za razvoj u Furijeov red, ali zbog vrlo srodne prirode ovih pojmova, u nastavku se za sve njih koristi opšti izraz Furijeova transformacija.

U narednim razmatranjima se pretpostavlja da je funkcija integrabilna sa kvadratom na relevantnom intervalu.

Neka je funkcija f periodična na intervalu $[a, b]$. Tada se može razviti u

Furijeov red:

$$f(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left(a_k \cos\left(\frac{2\pi kt}{b-a}\right) + b_k \sin\left(\frac{2\pi kt}{b-a}\right)\right)$$

gde važi

$$a_k = \frac{2}{b-a} \int_a^b f(t) \cos\left(\frac{2\pi kt}{b-a}\right) dt \qquad k = 0, 1, 2, \ldots$$

$$b_k = \frac{2}{b-a} \int_a^b f(t) \sin\left(\frac{2\pi kt}{b-a}\right) dt \qquad k = 1, 2, \ldots$$

Umesto celog reda, u praksi se može razmatrati samo suma nekog broja početnih elemenata, koji pruža zadovoljavajuću aproksimaciju. Treba primetiti da pri određivanju aproksimacije nije potrebno rešavati sistem 3.1, već je, zahvaljujući ortogonalnosti trigonometrijskog sistema funkcija, dovoljno izračunati skalarne proizvode koji definišu koeficijente a_k i b_k.

Primer 31 *Funkcija* $f(t) = 5\cos(2t) + 3\sin(8t)$ *je periodična na intervalu* $[0, \pi]$. *Lako se može izračunati da su svi Furijeovi koeficijenti ove funkcije 0, osim* $a_1 = 5$ *i* $b_4 = 3$, *što znači da ona sama predstavlja svoj konačan Furijeov red.*

Primer 32 *Neka je data funkcija*

$$f(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \end{cases}$$

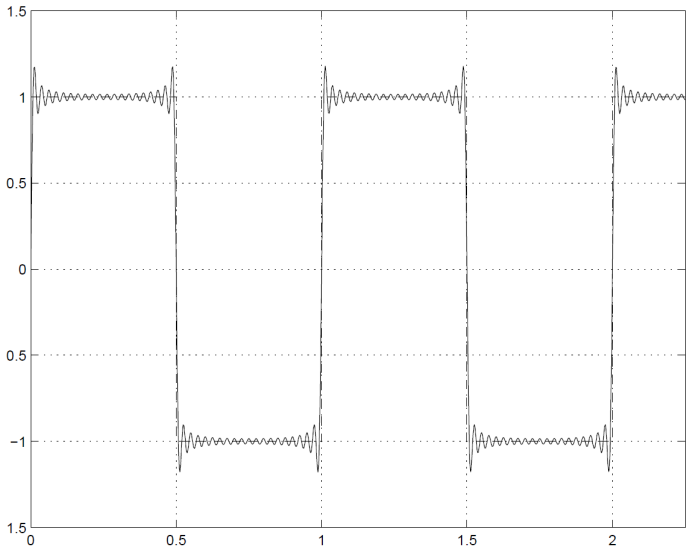
i periodična funkcija dobijena njenim nadovezivanjem, prikazana na slici 3.13, sa svojom aproksimacijom trigonometrijskim Furijeovim redom, izračunatim na osnovu 39 prvih članova. Primećuje se da su tačke prekida posebno problematične za aproksimaciju, što je i očekivano, imajući u vidu da su sve funkcije sistema neprekidne. Dodatno, primećuje se da se upravo u okolini takvih tačaka – oštrih ćoškova, uočava visoka frekvencija. Zbog toga se prilikom filtriranja zvučnog signala pazi da ne dođe do prekida, jer će se u suprotnom visoke frekvencije čuti kao pucketanje i kvariti doživljaj zvuka.

Prikazana reprezentacija Furijeovog reda je nešto teža za algebarsku manipulaciju i često se umesto nje koristi kompleksna reprezentacija, koja se oslanja na naredne identitete:

$$e^{i\theta} = \cos\theta + i\sin\theta$$

$$\cos(\theta) = \frac{e^{i\theta} + e^{-i\theta}}{2}$$

$$\sin(\theta) = i\frac{e^{-i\theta} - e^{i\theta}}{2}$$



Slika 3.13: Aproksimacija prekidne funkcije trigonometrijskim Furijeovim redom.

Kompleksna reprezentacija Furijeovog reda, koja će biti korišćena u nastavku je data narednim jednakostima:

$$f(t) = \sum_{k=-\infty}^{\infty} \hat{f}_k e^{-2\pi i k t/(b-a)} \tag{3.4}$$

$$\hat{f}_k = \frac{1}{b-a} \int_a^b f(t) e^{2\pi i k t/(b-a)} dt$$

Realna i kompleksna reprezentacija su u tesnoj vezi i važi:

$$a_0 = 2\hat{f}_0$$

$$a_k = \hat{f}_k + \hat{f}_{-k}$$

$$b_k = i(\hat{f}_{-k} - \hat{f}_k)$$

Treba imati u vidu da se u literaturi (a i u implementacijama) često mogu sresti i različite formulacije od one date jednakostima 3.4. Recimo, deljenje dužinom intervala može biti ispred sume, umesto ispred integrala. Takođe, znak $-$ u eksponentu može biti prisutan u drugoj, umesto u prvoj, itd. U zavisnosti od toga, i formule za konverziju između kompleksne i realne reprezentacije trigonometrijskog Furijeovog reda mogu biti različite.

Koeficijenti $\hat{f}_k$ su kompleksni brojevi, čak i ako predstavljaju koeficijente realne funkcije. Naravno, prilikom sumiranja, kompleksni delovi se poništavaju.

Dodatno, za koeficijente važi $\overline{\hat{f}_k} = \hat{f}_{-k}$:

$$\overline{\hat{f}_k} = \overline{\frac{1}{b-a}\int_a^b f(t)e^{2\pi ikt/(b-a)}dt} = \frac{1}{b-a}\int_a^b \overline{f(t)}\overline{e^{2\pi ikt/(b-a)}}dt =$$

$$= \frac{1}{b-a}\int_a^b f(t)e^{-2\pi ikt/(b-a)}dt = \hat{f}_{-k}$$

U nastavku će prvo biti prikazivana druga jednakost, koja predstavlja Furijeovu transformaciju, a potom prva, koja predstavlja *inverznu Furijeovu transformaciju*. Obično se zamišlja da promenljiva t predstavlja vreme, dok Furijeovi koeficijenti $\hat{f}_k$ predstavljaju intenzitete odgovarajućih frekvencija u signalu. U razvoju u Furijeov red, vreme je neprekidno, ali je frekvencijski domen diskretan, iako beskonačan. Odnosno, periodična funkcija se može predstaviti kao linearna kombinacija beskonačnog broja sinusa i kosinusa, ali sa diskretnim frekvencijama. Takođe, važna je pretpostavka da je funkcija periodična.

Kako bi se prevazišla dva uočena ograničenja razvoja u Furijeov red i kako bi se nešto slično moglo uraditi za funkcije definisane na intervalu $(-\infty, \infty)$, red se zamenjuje integralom, a Furijeovi koeficijenti funkcijom koja izražava intenzitet frekvencija:

$$\hat{f}(u) = \int_{-\infty}^{\infty} f(t)e^{2\pi iut}dt$$

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(u)e^{-2\pi iut}du$$

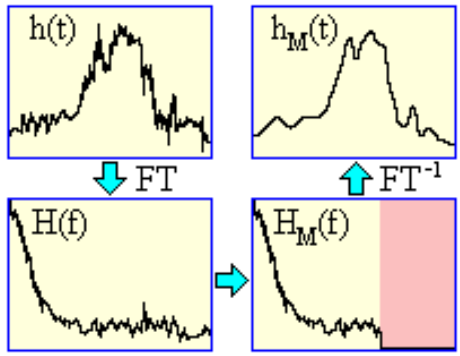
Promenljiva t predstavlja vreme, a promenljiva u frekvenciju. Očito, za sve moguće frekvencije su definisani intenziteti. Ponovo, funkcija f je nešto nalik linearnoj kombinaciji sinusa i kosinusa, ali svih mogućih frekvencija.

Mana oba prethodna pristupa je što je nekada funkcija f poznata samo na konačnom skupu tačaka. U takvim situacijama, nije moguće osloniti se na integrale, pa se oni zamenjuju odgovarajućim sumama. Neka su vrednosti funkcije $f_j = f(t_j)$ poznate samo u tačkama $t_j = t_0 + jh$, za $j = 0, 1, \ldots, n-1$ i $h > 0$ i neka je funkcija periodična, sa periodom nh. Tada je diskretna Furijeova transformacija definisana narednim jednakostima.

$$\hat{f}_k = \frac{1}{n}\sum_{j=0}^{n-1} f_j e^{2\pi ikj/n} \qquad k = 0, 1, \ldots, n-1$$

$$f_j = \sum_{k=0}^{n-1} \hat{f}_k e^{-2\pi ikj/n} \qquad j = 0, 1, \ldots, n-1$$

U slučaju diskretne Furijeove transformacije, funkcija se aproksimira konačnom linearnom kombinacijom sinusa i kosinusa, pri čemu su i vreme i frekvencije diskretni.



Slika 3.14: Uklanjanje šuma iz signala pomoću Furijeove transformacije.

Primer 33 *Klasičan primer upotrebe Furijeove transformacije je uklanjanje šuma iz signala i sastoji se u uklanjanju visokih frekvencija ili drugih delova frekvencijskog spektra, kojima šum dominira. Ovo je ilustrovano na slici 3.14.*

Kako Furijeove transformacije daju kompleksne vrednosti, mogu se odvojeno posmatrati njihove realne i imaginarne komponente. Neka je $f_-(x) = f(-x)$. Za realne i imaginarne komponente Furijeove transformacije važi:

$$Re(\hat{f}) = \frac{\widehat{f + f_-}}{2} \qquad Im(\hat{f}) = -i\frac{\widehat{f - f_-}}{2}$$

Furijeova transformacija se intenzivno koristi u obradi slika. U tom slučaju je potrebno definisati je u dve dimenzije. U neprekidnom slučaju, definiše se kao:

$$\hat{f}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{2\pi i(xu+yv)}dxdy$$

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \hat{f}(u,v)e^{-2\pi i(xu+yv)}dudv$$

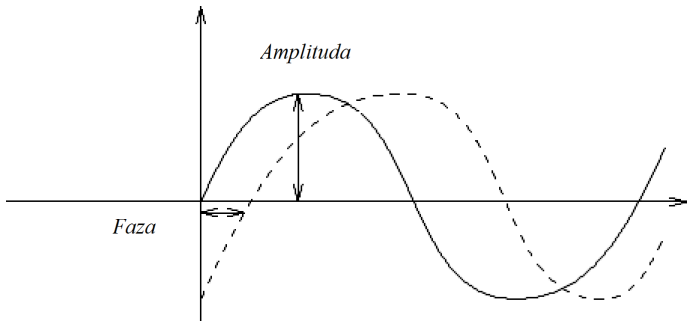
U diskretnom slučaju, definiše se kao:

$$\hat{f}_{lm} = \frac{1}{PQ}\sum_{j=0}^{P-1}\sum_{k=0}^{Q-1} f_{jk}e^{2\pi i(jl/P+km/Q)}$$

$$f_{jk} = \sum_{l=0}^{P-1}\sum_{m=0}^{Q-1} \hat{f}_{lm}e^{-2\pi i(jl/P+km/Q)}$$

Elementi dvodimenzionalnog sistema nad kojim se vrši razvoj su prikazani na slici 3.15.

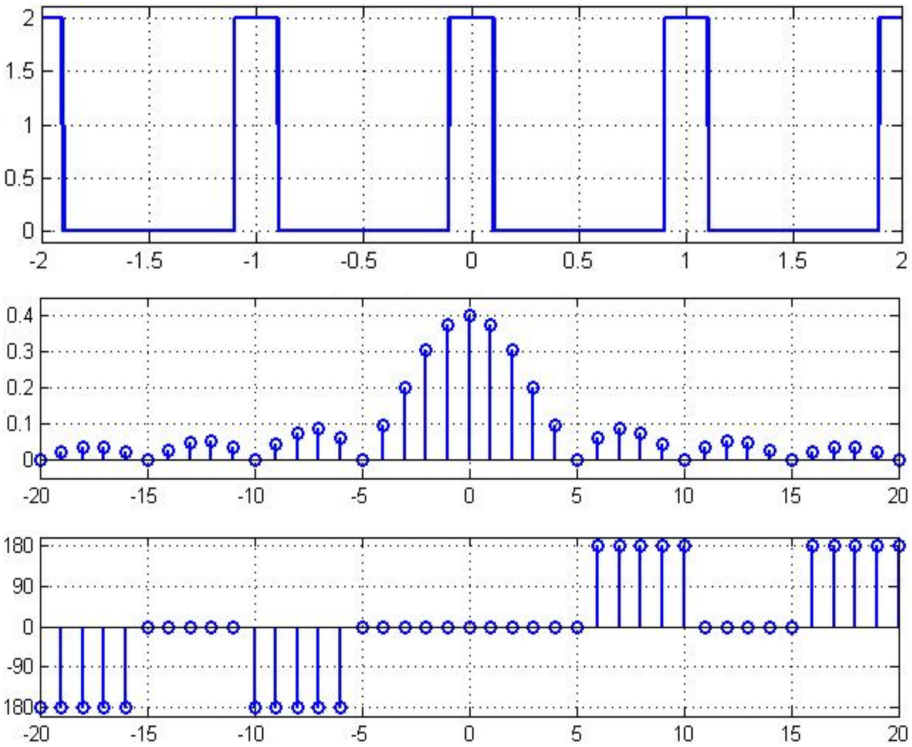
Slika 3.15: Prikaz nekih elemenata trigonometrijskog sistema za funkcije dve promenljive.



Slika 3.16: Ilustracija amplitude i faze sinusoide.

Kompleksni brojevi, kakve su i vrednosti Furijeove transformacije, su određeni svojim modulom i argumentom iliti fazom – intenzitetom radijus vektora i uglom u kompleksnoj ravni. Modul predstavlja intenzitet iliti amplitudu neke frekvencije u nekom signalu, dok faza predstavlja pomeraj sinusne ili kosinusne funkcije te frekvencije duž vremenske ose, kao što je prikazano na slici 3.16. Prilikom prikaza Furijeove transformacije funkcije, najčešće se prikazuje *spektar modula* – grafik modula u zavisnosti od frekvencije, ali je moguće prikazati i *fazni spektar* – grafik faza u zavisnosti od frekvencije. Za jedan signal, oba spektra su prikazana na slici 3.17. Obe informacije su vrlo bitne u primenama. Prirodno se postavlja pitanje koja od ovih informacija je važnija. Odnosno, ukoliko bi trebalo sačuvati samo jednu od njih, koju bi bilo bolje izabrati. Ovakva pitanja su relevantna za dizajn filtera kojima se signali obrađuju, a odgovor će biti ilustrovan sledećim primerom. Prilikom prikaza spektra modula slike, on se obično prikazuje tako da centralni koeficijent $\hat{f}(0,0)$ bude prikazan u sredini, a koeficijenti dalje od centra predstavljaju više frekvencije.

Primer 34 *Na slici 3.18 data je originalna slika i odgovarajući spektar modula i fazni spektar dobijeni diskretnom Furijeovom transformacijom. Na slici 3.19 prikazane su rekonstrukcije slika inverznom Furijeovom transformacijom*



Slika 3.17: Signal, njemu odgovarajući spektar modula i odgovarajući fazni spektar.

nad i) Furijeovim transformacijama polaznih slika kojima su anulirane faze, ii) Furijeovim transformacijama polaznih slika kojima su normirani moduli i iii) Furijeovim transformacijama polaznih slika kojima su zamenjene vrednosti modula. Iz primera se vidi da faza nosi značajniju informaciju nego modul.

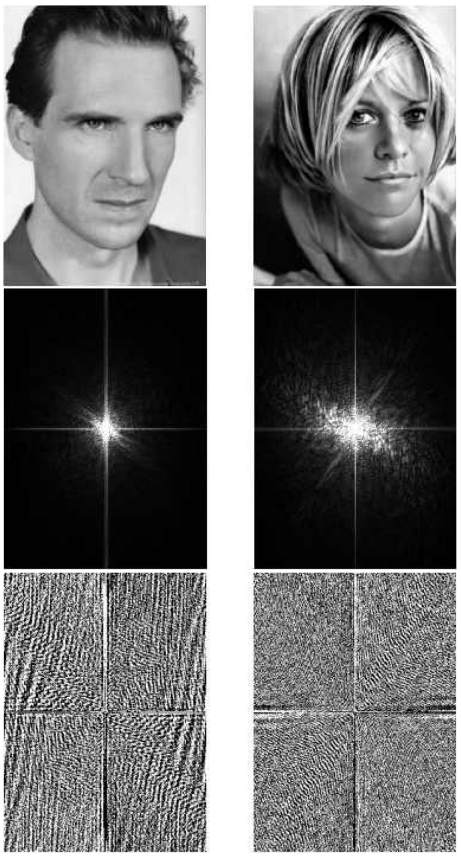
U nastavku će biti razmotreno nekoliko primera vezanih za obradu slika.

Primer 35 *Dirakova delta funkcija se* neformalno *opisuje kao funkcija*

$$\delta(x) = \begin{cases} +\infty & x = 0 \\ 0 & x \neq 0 \end{cases}$$

pri čemu važi

$$\int_{-\infty}^{\infty} \delta(x)dx = 1$$



Slika 3.18: Slika i odgovarajući spektar modula i fazni spektar.

Neka je data funkcija $f(x,y) = \delta(x,y) = \delta(x)\delta(y)$. *Prikaz ove funkcije dat je na slici 3.20 gore levo. Neprekidna Furijeova transformacija daje:*

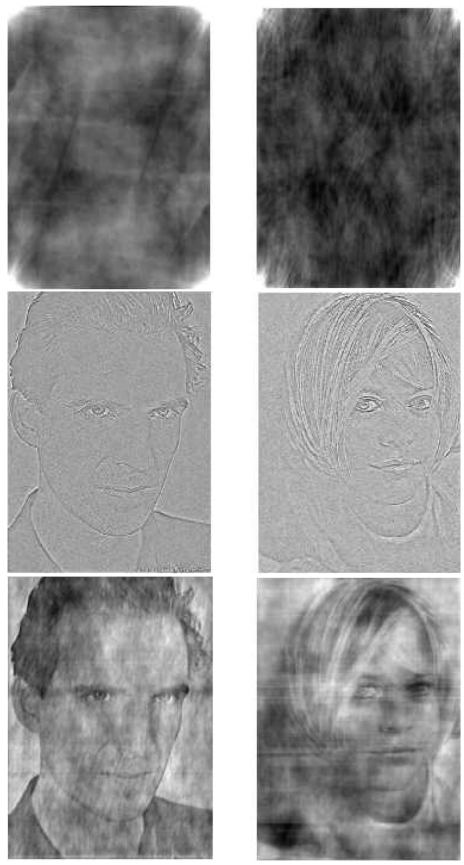
$$\hat{f}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(x,y)e^{2\pi i(ux+vy)}dxdy = e^{2\pi i 0} = 1$$

što objašnjava sliku gore desno. Neka je funkcija $f(x,y) = \frac{1}{2}(\delta(x,y-a) + \delta(x,y+a))$. *Prikaz ove funkcije dat je na slici 3.20 dole levo. Neprekidna Furijeova transformacija daje:*

$$\hat{f}(u,v) = \frac{1}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(\delta(x,y-a) + \delta(x,y+a))e^{2\pi i(ux+vy)dxdy} =$$

$$\frac{1}{2}(e^{2\pi iav} + e^{-2\pi iav}) = cos(2\pi av)$$

što se vidi na slici dole desno. Generalno, slika se razlaže na sinusoidalne



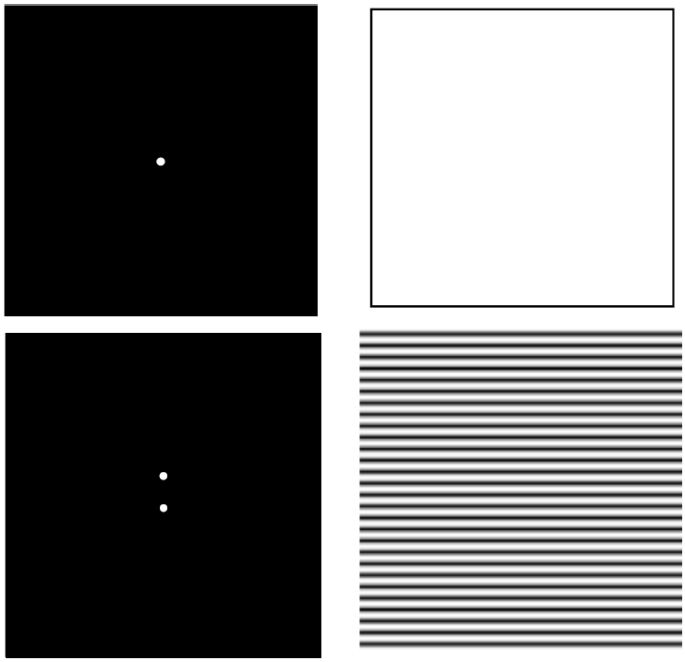
Slika 3.19: Ishodi koji se dobijaju rekonstrukcijom slike nakon uklanjanja faze, normiranja modula i zamene vrednosti modula.

komponente nalik prikazu na slici 3.21.

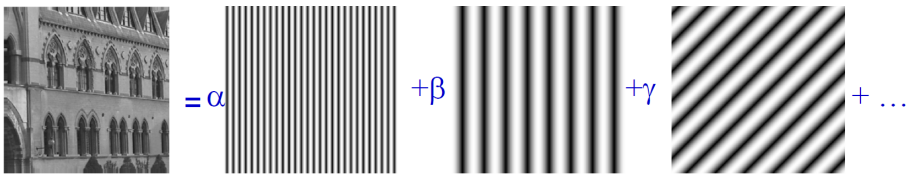
Primer na slici 3.22 prikazuje kako se odsecanjem dela spektra mogu zadržati najniže ili najviše frekvencije i kakvim efektima na slici to rezultuje. Odsecanje viših frekvencija omogućava grub prikaz slike, dok odsecanje nižih frekvencija omogućava prepoznavanje ivica, upravo zato što ivice predstavljaju tačke prekida u dvodimenzionalnom signalu.

Periodične strukture na slikama se često dobro uočavaju na prikazu Furijeove transformacije. Uklanjanjem prepoznatljivih maksimuma iz Furijeove transformacije i vraćanjem nazad inverznom Furijeovom transformacijom, u polaznom signalu se odstranjuju periodične strukture. Ovo se vidi na slikama 3.23, 3.24 i 3.25.

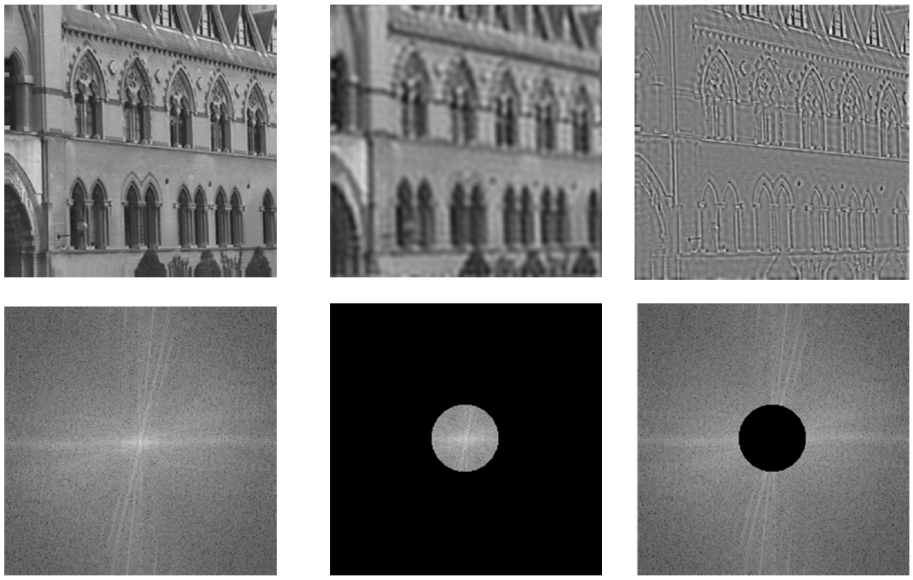
Primer 36 *Jedna od najpoznatijih primena Furijeove transformacije, odnosno transformacije koja joj je vrlo bliska, je JPEG kompresija. Kompresija ima*



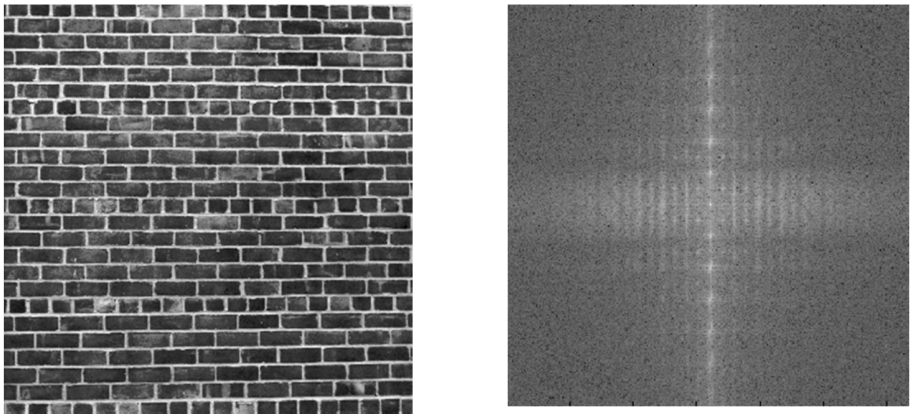
Slika 3.20: Furijeove transformacije dve jednostavne slike.



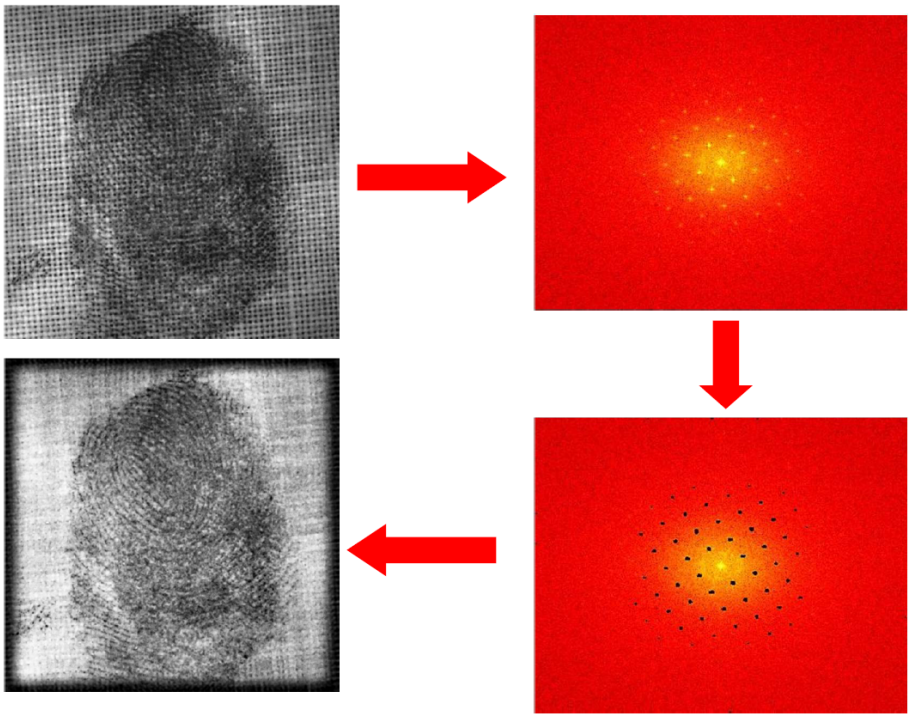
Slika 3.21: Razlaganje slike nad sistemom funkcija pomoću Furijeove transformacije.



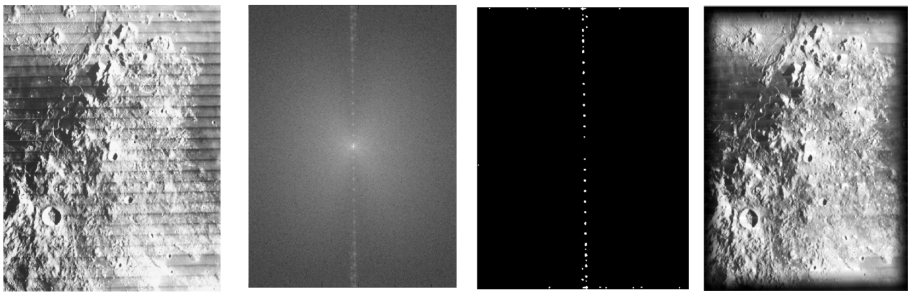
Slika 3.22: Efekti filtera kojima se u spektru slike odsecaju više i niže frekvencije.



Slika 3.23: Periodičnost slike se odražava prepoznatljivim maksimumima.



Slika 3.24: Uklanjanjem prepoznatljivih maksimuma, uklanjaju se i periodične strukture sa slike. Na donjoj desnoj slici, crne tačke predstavljaju uklonjene regione spektra.



Slika 3.25: Uklanjanjem prepoznatljivih maksimuma, uklanjaju se i periodične strukture sa slike. Treća slika ističe maksimume koji se uklanjaju.

nekoliko koraka kojima se postiže veći faktor kompresije, ali jezgro metode se sastoji u određivanju Furijeovih koeficijenata i odbacivanju onih koji imaju niske vrednosti, što su koeficijenti koji odgovaraju višim frekvencijama.

Prvi korak algoritma je prevođenje iz RGB zapisa u YCbCr sistem u kojem komponenta Y označava osvetljenost, a Cb i Cr zajednički označavaju ton i zasićenost boje. Transformacija glasi:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.334 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

Kako je ljudsko oko osetljivije na osvetljenost, nego na komponente koje predstavljaju boju, ovaj sistem omogućava smanjivanje dinamičkog raspona komponenti Cb i Cr. Nakon tih koraka, slika se deli na blokove dimenzija 8×8 *i na sve tri komponente slike se odvojeno primenjuje* diskretna kosinusna transformacija *– transformacija slična Furijeovoj koja se oslanja na razvoj nad sistemom kosinusa:*

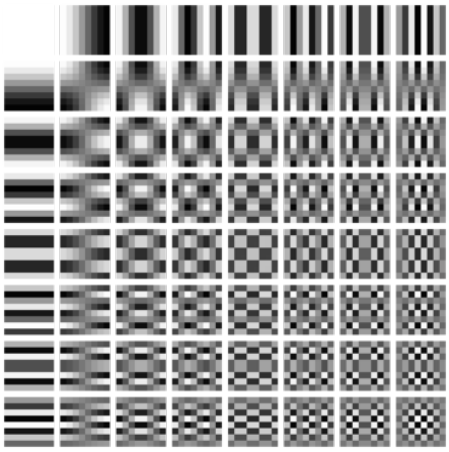
$$\hat{f}_{lm} = \frac{1}{4} c_l c_m \sum_{j=0}^{7} \sum_{k=0}^{7} f_{jk} \cos \frac{(2j+1)l\pi}{16} \cos \frac{(2k+1)m\pi}{16}$$

$$f_{jk} = \frac{1}{4} \sum_{l=0}^{7} \sum_{m=0}^{7} c_l c_m \hat{f}_{lm} \cos \frac{(2j+1)l\pi}{16} \cos \frac{(2k+1)m\pi}{16}$$

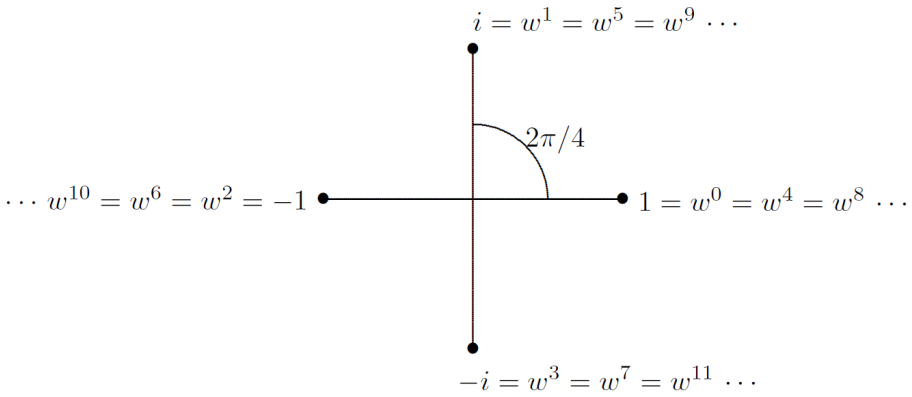
gde važi

$$c_k = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 1 & k \neq 0 \end{cases}$$

Sistem funkcija nad kojima se vrši razvoj ovom transformacijom dat je na slici 3.26. Osnovni razlog što je diskretna kosinusna transformacija preferirana u odnosu na diskretnu Furijeovu transformaciju je što dodeljuje veće koeficijente nižim frekvencijama i samim tim niže višim frekvencijama, nego diskretna Furijeova transformacija. To olakšava odbacivanje koeficijenata viših frekvencija što vodi višem stepenu kompresije. Koeficijent najveće vrednosti je u gornjem levom uglu matrice, a ostali su manji što su dalji od gornjeg levog ugla. Eliminacija koeficijenata se vrši tako što se svi elementi transformisane matrice dele odgovarajućim elementima unapred definisane matrice Q, koja definiše gubitak kvaliteta od 50%. Drugi procenti gubitka se dobijaju tako što se ova matrica množi odgovarajućim skalarom. Sama matrica je dobijena kroz eksperimente vezane za osetljivost ljudske vizualne percepcije. Nakon deljenja, mnogi koeficijenti postaju jednaki nuli. Nakon toga, elementi matrice se ređaju cik-cak dijagonalnim nabrajanjem pošavši od gornjeg levog ugla. Na taj način većina nula se grupiše i daje sekvence nula koje se lako kompresiju. Finalna kompresija se vrši nekom od metoda kompresije bez gubitka, poput Hafmanovog kodiranja.



Slika 3.26: Sistem funkcija nad kojim se vrši razvoj diskretnom kosinusnom transformacijom.



Slika 3.27: Stepeni w^k četvrtog korena jedinice u kompleksnoj ravni.

3.4.1 Brza Furijeova transformacija

Od prethodnih varijanti Furijeove transformacije, u praksi se najčešće koristi diskretna Furijeova transformacija ili skraćeno DFT. Njena složenost, kada se računa prema formulama koje je definišu, je $\Theta(n^2)$, gde je n dužina uzorka signala. Imajući u vidu da zapisi koji se danas čuvaju i obrađuju mogu biti prilično obimni, ovo vreme izvršavanja predstavlja ozbiljnu prepreku za primenu Furijeove transformacije u praksi. Umesto njega, u praksi se koristi algoritam brze Furijeove transformacije (eng. fast Fourier transform) ili skraćeno FFT. Za dati broj n, n-ti koren jedinice je $w = e^{2\pi i/n}$. Ponašanje ovog korena pri stepenovanju u kompleksnoj ravni je ilustrovano slikom 3.27. Koristeći tu

oznaku, diskretna Furijeova transformacija glasi:

$$\hat{f}_k = \frac{1}{n}\sum_{j=0}^{n-1} f_j w^{kj} \qquad k = 0, 1, \ldots, n-1$$

Važi

$$\hat{f}_{k+n} = \frac{1}{n}\sum_{j=0}^{n-1} f_j w^{(k+n)j} = \frac{1}{n}\sum_{j=0}^{n-1} f_j w^{kj} w^{nj} = \frac{1}{n}\sum_{j=0}^{n-1} f_j w^{kj} = \hat{f}_k$$

zahvaljujući tome što je $w = e^{2\pi i/n}$, pa otud važi $w^n = 1$. Dodatno važi

$$w^{k+n/2} = e^{\frac{2\pi i(k+n/2)}{n}} = e^{\frac{2\pi ik}{n}+\pi i} = e^{\pi i} e^{\frac{2\pi ik}{n}} = -w^k$$

Još treba primetiti da ako DFT niza dužine n, gde je n parno, odgovara koren jedinice w, onda DFT niza dužine $n/2$ odgovara koren $w' = w^2$. Iz formula DFT, može se primetiti da se DFT niza parne dužine n, može rekurzivno izraziti preko DFT dva njegova podniza, parnih i neparnih elemenata:

$$\hat{f}_k = \frac{1}{n}\sum_{j=0}^{n-1} f_j w^{kj} = \frac{1}{n}\sum_{j=0}^{n/2-1} f_{2j} w^{2jk} + \frac{1}{n}\sum_{j=0}^{n/2-1} f_{2j+1} w^{(2j+1)k} =$$

$$\underbrace{\frac{1}{2}\frac{1}{n/2}\sum_{j=0}^{n/2-1} f_{2j} w'^{jk}}_{\text{DFT parnih}} + \frac{1}{2}w^k \underbrace{\frac{1}{n/2}\sum_{j=0}^{n/2-1} f_{2j+1} w'^{jk}}_{\text{DFT neparnih}} = \frac{1}{2}(E_k + w^k O_k)$$

Zahvaljujući pokazanoj periodičnosti Furijeove transformacije i imajući u vidu da su E_k i O_k Furijeove transformacije nad $n/2$ tačaka, važi

$$E_{k+n/2} = E_k$$

$$O_{k+n/2} = O_k$$

Stoga se Furijeovi koeficijenti mogu zapisati i u obliku

$$\hat{f}_k = \begin{cases} \frac{1}{2}(E_k + w^k O_k) & 0 \le k < n/2 \\ \frac{1}{2}(E_{k-n/2} + w^k O_{k-n/2}) & n/2 \le k < n \end{cases}$$

Zahvaljujući pokazanoj činjenici

$$w^{k+n/2} = -w^k$$

važi i

$$\hat{f}_k = \frac{1}{2}(E_k + w^k O_k)$$

$$\hat{f}_{k+n/2} = \frac{1}{2}(E_k - w^k O_k)$$

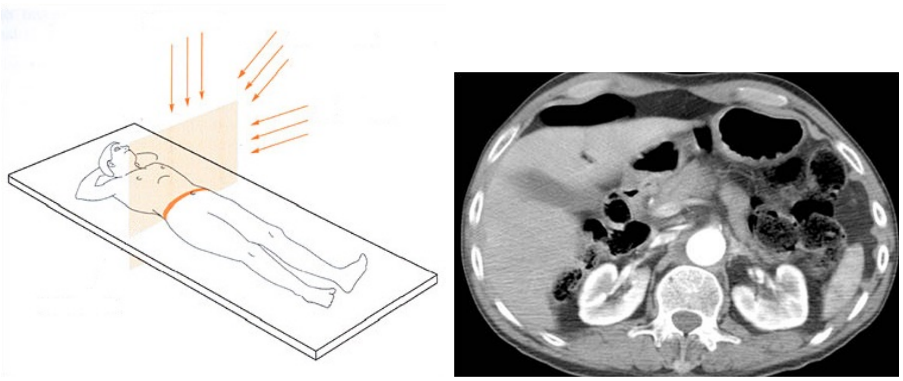
```
1  fft(f,n,s):
2      if n=1 then
3          f̂_0 ← f_0
4      else
5          f̂_0, ..., f̂_{n/2-1} ← fft(f,n/2,2s)
6          f̂_{n/2}, ..., f̂_{n-1} ← fft(f+s,n/2,2s)
7          k ← 0
8          while k < n/2 − 1 do
9              f̂_{k+n/2} ← f̂_k − w^k f̂_{k+n/2}
10             f̂_k ← f̂_k + w^k f̂_{k+n/2}
11             k ← k + s
12         end
13     end
14     return f̂_0, ..., f̂_{n-1}
```

Slika 3.28: Algoritam brze Furijeove transformacije. `f` je pokazivač na početak niza brojeva koji treba transformisati.

za $0 \leq k < n/2$. Očigledno, novi algoritam za izračunavanje DFT, može se formulisati kao na slici 3.28. Zbog saglasnosti sa uobičajenim implementacijama i drugim formulacijama, u algoritmu je izostavljeno množenje polovinom, koje je prisutno u formulama. Kako bi se to kompenzovalo, potrebno je deljenje rezultata brojem n, nakon izvršavanja algoritma. Razlog za takvu formulaciju je razlika u definiciji diskretne Furijeove transformacije, koja može da uključuje deljenje brojem n u rekonstrukciji funkcije, a ne u računanju koeficijenata.

Analiza složenosti algoritma FFT liči na analizu složenosti algoritma sortiranja spajanjem (eng. merge sort) i ishod je isti – vremenska složenost je $\Theta(n \log n)$. Algoritam FFT se može formulisati i iterativno – bez rekurzije. Očigledna mana algoritma FFT je u tome što dužina ulaznog niza n mora biti stepen dvojke, kako bi se niz uvek mogao deliti na dva jednaka dela. Postoje i modifikacije ovog algoritma koje ublažavaju ovaj problem dok god n može da se faktoriše na male proste brojeve. Na primer, niz dužine 45 bi se obrađivao deljenjem na 3 podniza dužine 15, od kojih bi svi bili podeljeni na 3 podniza dužine 5. Kada je dužina podniza prost broj, na tom podnizu se koristi standardni DFT algoritam. U slučaju da je dužina niza prost broj, ubrzanje nije moguće. Pored primene drugačijeg algoritma, često se pribegava lakšem rešenju – popunjavanja niza nulama dok ne dostigne dužinu koja je stepen dvojke. Međutim, ovakva strategija može nepovoljno uticati na rezultate izračunavanja.

Imajući u vidu očiglednu sličnost u formulacijama Furijeove transformacije i inverzne Furijeove transformacije, očekivano je da se FFT algoritam može upotrebiti i za izračunavanje inverzne Furijeove transformacije. Najjednostavniji način je da se pre upotrebe algoritma FFT ulaz konjuguje, a da se nakon njegove primene konjuguje i izlaz. Pri inverznoj transformaciji ne treba deliti



Slika 3.29: Ilustracija postupka snimanja vezanog za racunsku tomografiju i primer snimka.

rezultat brojem n.

3.4.2 Računska tomografija

Jedna od najimpresivnijih praktičnih primena Furijeove transformacije je računska tomografija (eng. computed tomography), koja predstavlja tehniku kombinovanja rendgenskih snimaka nekog objekta iz različitih uglova zarad rekonstrukcije slike poprečnog preseka tog objekta. Ova tehnika se intenzivno koristi u medicinskoj dijagnostici za dobijanje prikaza koje nije moguće dobiti standardnim rendgenskim snimanjem, koje ne zadržava dubinsku informaciju. Tehnika počiva na korišćenju niza jednodimenzionalnih rendgenskih snimaka koji se prave tako što izvor X zraka rotira oko pacijenta u ravni željenog preseka i zrači snopom X zraka koji leži u toj ravni i ima dovoljnu širinu da obuhvati pacijenta. Postupak snimanja i snimak su ilustrovani slikom 3.29.

Pre prikaza računske tomografije, biće reči o apsorpciji zračenja pri prolasku kroz neki medijum. To može biti prolazak svetlosti kroz obojeno staklo, ali i prolazak X zraka kroz telo. U slučaju uniformne gustine, intenzitet zračenja se smanjuje za isti procenat po pređenoj jedinici dužine. Odnosno, intenzitet eksponencijalno opada. Stoga se intenzitet zračenja za pređeno rastojanje t, može opisati formulom

$$I(t) = I_0 e^{-\mu t}$$

gde je I_0 polazni intenzitet, a μ koeficijent apsorpcije. Neka je materijal kroz koji zračenje prolazi nehomogen i neka $\mu(x, y)$ predstavlja koeficijent apsorpcije u zavisnosti od lokacije. Ovaj koeficijent će u nastavku biti poistovećen sa gustinom. X zrak se kreće pravom sa koordinatama $(x(s), y(s))$, gde je s dužina puta i kreće se od s_0 do s_1. U tom slučaju, intenzitet zračenja se računa kao

$$I(t) = I_0 exp\left(-\int_{s_0}^{t} \mu(x(s), y(s))ds\right) \qquad t \in [s_0, s_1]$$

Ako se pretpostavi da funkcija $\mu(x, y)$ ima vrednost 0 van razmatranog tela, za granice se mogu uzeti vrednosti $-\infty$ i ∞. Jednačina prave po kojoj se zrak kreće se može zapisati kao

$$x \cos\phi + y \sin\phi - \rho = 0 \qquad x, y \in \mathbb{R}$$

Korišćenjem Dirakove delta funkcije se prilikom integracije može ograničiti domen integracije na tu pravu i definisati *Radonova transformacija* funkcije $\mu(x, y)$:

$$\tilde{\mu}(\rho, \phi) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(x, y)\delta(x \cos\phi + y \cos\phi - \rho)dxdy$$

Za svaki ugao rotacije ϕ, pri variranju parametra ρ, razmatra se integral duž jedne od paralelnih linija koje su sve normalne na pravu koja zaklapa ugao ϕ sa osom y. Kolekcija vrednosti $\tilde{\mu}(\rho, \phi)$ za fiksirano ϕ se naziva *projekcijom funkcije* μ *pod uglom* ϕ, a problem je odrediti funkciju (sliku) μ iz njenih projekcija. Treba imati u vidu da izraz projekcija u ovom kontekstu nema geometrijsku interpretaciju. Ključno mesto u rešavanju ovog problema ima *centralna teorema o presecima*. U njenoj formulaciji podrazumeva se sledeća oznaka za projekciju $\tilde{\mu}_\phi(\rho) = \tilde{\mu}(\phi, \rho)$.

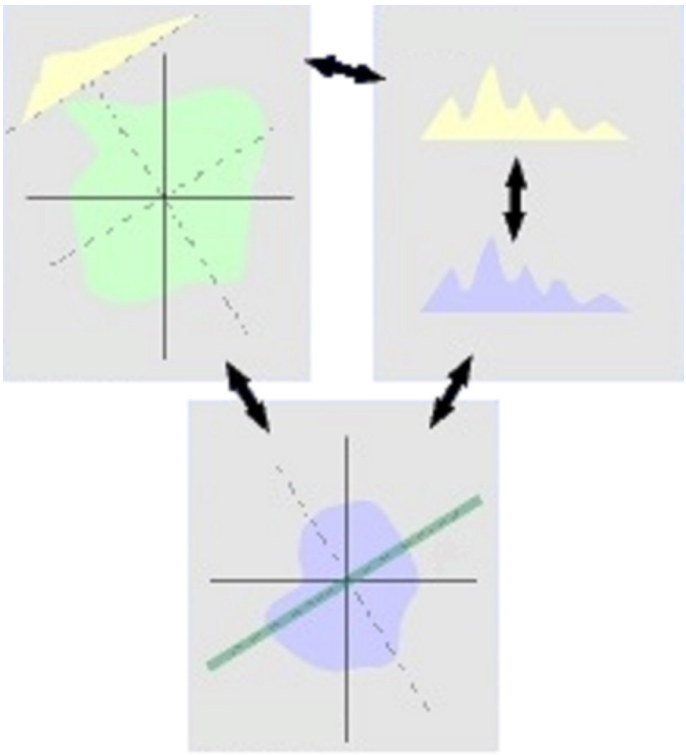
Teorema 4 (Centralna teorema o presecima) *Važi*

$$\widehat{\tilde{\mu}}_\phi(r) = \hat{\mu}(r \cos\phi, r \sin\phi).$$

Dokaz.

$$\begin{aligned}
\widehat{\tilde{\mu}}_\phi(r) &= \int_{-\infty}^{\infty} \tilde{\mu}(\rho, \phi) e^{2\pi i r\rho} d\rho \\
&= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(x, y)\delta(x \cos\phi + y \cos\phi - \rho)dxdy \right) e^{2\pi i r\rho} d\rho \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mu(x, y) \left(\int_{-\infty}^{\infty} \delta(x \cos\phi + y \cos\phi - \rho) e^{2\pi i r\rho} d\rho \right) dxdy \\
&= \int_{-\infty}^{\infty} \mu(x, y) e^{2\pi i r(x \cos\phi + y \sin\phi)} dxdy \\
&= \int_{-\infty}^{\infty} \mu(x, y) e^{2\pi i (xr \cos\phi + yr \sin\phi)} dxdy \\
&= \hat{\mu}(r \cos\phi, r \sin\phi)
\end{aligned}$$

Smisao prethodne teoreme je da je Furijeova transformacija projekcije funkcije pod nekim uglom jednaka vrednosti Furijeove transformacije te funkcije duž prave pod tim uglom, što je ilustrovano slikom 3.30. To omogućava da se Furijeova transformacija funkcije μ, odnosno slike, odredi na uzorcima pravih duž različitih uglova iz projekcija pod tim uglovima, a da se potom inverzonom Furijeovom transformacijom dobije slika. Problem je očito što uzorci ne leže na



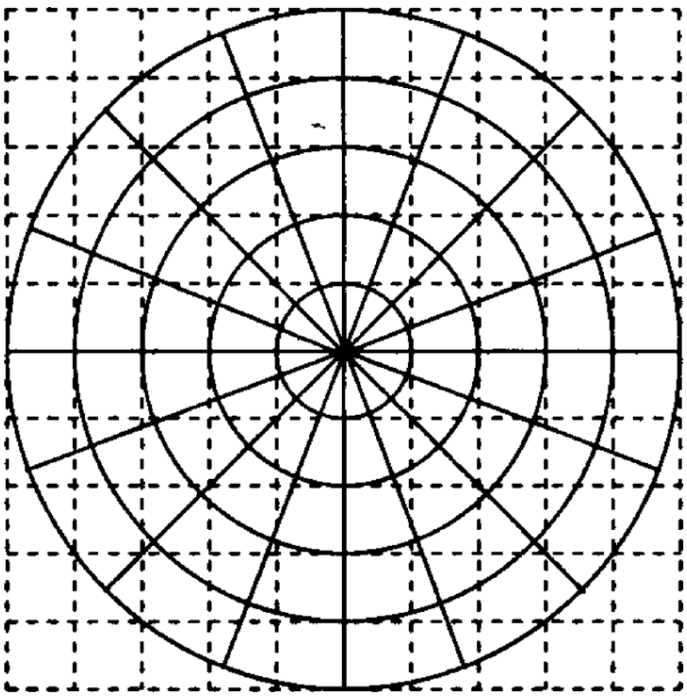
Slika 3.30: Furijeova transformacija projekcije slike pod nekim uglom je jednaka preseku Furijeove transformacije slike pod istim uglom.

pravougaonoj mreži kao što se očekuje za primenu diskretne Furijeove transformacije. Ovaj problem se rešava interpolacijom u tačkama neke pravougaone mreže. Međutim, kao što ilustruje slika 3.31, uzorak je gušći bliže centru, odnosno nižim frekvencijama. Kako su više frekvencije te od čijeg prisustva zavisi prikaz detalja, ovako rekonstruisane slike će imati loš prikaz detalja i mogu delovati mutno. Postoje i naprednije metode rekonstrukcije slike iz projekcija, koje se takođe oslanjaju na Furijeovu transformaciju.

Postupak rekonstrukcije slike u procesu računske tomografije, može se sumirati na sledeći način:

- Za snimanje se koristite izvor zračenja jačine I_0, koji odašilje snop zraka u ravni željenog preseka i naspramno postavljen senzor, koji zajedno mogu da rotiraju pod uglom $0 \leq \phi < \pi$.

- Za svaki ugao rotacije ϕ_i, $i = 1, 2, \ldots, m$, senzor beleži jačinu zračenja $I_{\phi_i}(\rho)$ duž prave na koju padaju zraci, gde je ρ označeno rastojanje od centralne linije snopa.



Slika 3.31: Prirodna mreža uzorka i pravougaona mreža na kojoj je potrebno uraditi interpolaciju.

- Za svako ϕ_i, $i = 1, 2, \ldots, n$, računa se $g_{\phi_i}(\rho_j) = -log\left(\frac{I_{\phi_i}(\rho_j)}{I_0}\right)$, $j = 1, 2, \ldots, n$.
- Za svako ϕ_i, $i = 1, 2, \ldots, n$, računa se $\hat{g}_{\phi_i}(\rho_j)$, $j = 1, 2, \ldots, n$, i na osnovu teoreme o centralnom preseku te vrednosti se uzimaju za vrednosti $\hat{\mu}(\rho_i \cos\phi_i, \rho_i \sin\phi_i)$.
- Na osnovu tog uzorka vrednosti funkcije $\hat{\mu}$, interpolacijom se dobijaju vrednosti funkcije $\hat{\mu}(x_i, y_j)$ za $i = 1, 2, \ldots, p$ i $j = 1, 2, \ldots, q$, na nekoj pravougaonoj mreži.
- Vrši se inverzna Furijeova transformacija nad dobijenim uzorkom i formira se slika.

3.4.3 Konvolucija

Jedna od glavnih primena Furijeove transformacije je obrada signala. Postavlja se pitanje postoji li veza između rezultata jednostavnih aritmetičkih operacija nad signalima i nekih operacija nad njihovim Furijeovim transforma-

cijama. Trivijalno je uveriti se da važi:

$$\widehat{f+g} = \hat{f} + \hat{g}$$

Postavlja se pitanje, postoji li kombinacija signala f i g, takva da je njena Furijeova transformacija $\hat{f}\hat{g}$. Odgovor se može dobiti polazeći od proizvoda Furijeovih transformacija:

$$\begin{aligned}
\hat{f}(u)\hat{g}(u) &= \int_{-\infty}^{\infty} f(x)e^{2\pi ixu}dx \int_{-\infty}^{\infty} g(y)e^{2\pi iyu}dy \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x)g(y)e^{2\pi i(x+y)u}dxdy \\
&\quad (y = v - x) \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x)g(v-x)e^{2\pi ivu}dxdv \\
&= \int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} f(x)g(v-x)dx\right)e^{2\pi ivu}dv \quad = \int_{-\infty}^{\infty}(f*g)(v)e^{2\pi ivu}dv
\end{aligned}$$

gde važi

$$(f*g)(v) = \int_{-\infty}^{\infty} f(x)g(v-x)dx$$

Operacija $*$ se naziva konvolucijom. Ovime je dokazana sledeća teorema.

Teorema 5 (Teorema o konvoluciji) *Važi*

$$\widehat{f*g} = \hat{f}\hat{g}.$$

Pored navedenog, važe još neka svojstva konvolucije:

$$\widehat{fg} = \hat{f} * \hat{g}$$

$$f*g = g*f$$

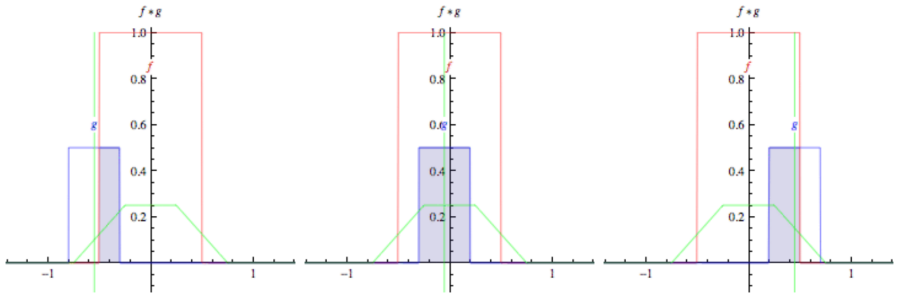
$$(f*g)*h = f*(g*h)$$

$$f*(g+h) = f*g + f*h$$

$$f*\delta = f$$

gde je δ već pomenuta Dirakova delta funkcija.

Primer 37 *Ako se funkcija g reflektuje oko vertikalne ose i prevuče duž x ose, vrednost konvolucije u nekoj tački je površina ispod grafika funkcije njihovog proizvoda. Na slici 3.32 je data ilustracija konvolucije. Kako funkcija f ima vrednost 0 ili 1, proizvod odgovara vrednosti funkcije g. Otud zeleni grafik predstavlja površinu preklopa između funkcija f i g. Stoga zeleni grafik prvo raste pod uglom od $45°$, pa je konstantan, pa opada.*



Slika 3.32: Ilustracija konvolucije.

Konvoluciju je moguće definisati i u diskretnom slučaju, upotrebom suma umesto integrala:

$$(f * g)_i = \sum_{j=0}^{n-1} f_j g_{i-j} \qquad i = 0, 1, \ldots, n-1$$

pri čemu se podrazumeva da je $g_k = g_{n+k}$, ukoliko je $k < 0$. Za diskretnu konvoluciju važe već navedena svojstva neprekidne konvolucije. Takođe, konvolucija se može definisati u više dimenzija, na primer dve:

$$(f * g)(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) g(u - x, v - y) dx dy$$

$$(f * g)_{ij} = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} f_{kl} g_{i-k, j-l}$$

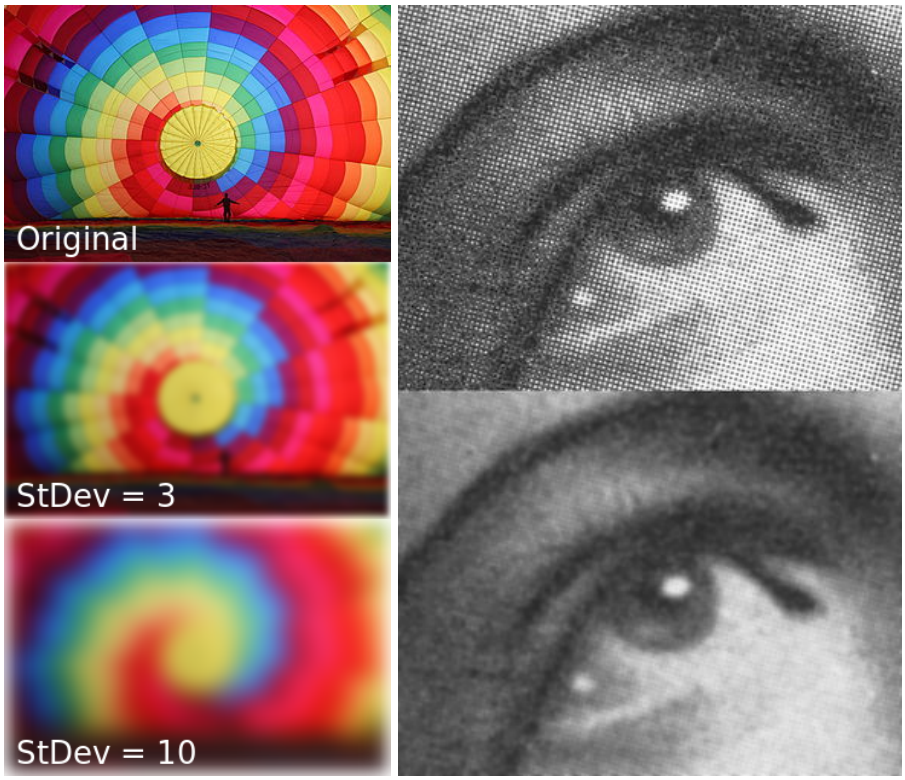
sa istom konvencijom indeksiranja kao u jednodimenzionalnom slučaju.

Očigledno, izračunavanje diskretne konvolucije dva signala prema formuli datoj u definiciji ima vremensku složenost $\Theta(n^2)$. Međutim, teorema o konvoluciji daje način za izračunavanje konvolucije u vremenu $\Theta(n \log n)$ – algoritmom FFT se izračunaju Furijeove transformacije signala f i g, koje se pomnože, a potom se istim algoritmom izračuna inverzna Furijeova transformacija proizvoda.

Primer 38 *Jedna primena konvolucije je u množenju polinoma. Neka su dati polinomi*

$$f(x) = \sum_{i=1}^{m} f_i x^i \qquad g(x) = \sum_{i=1}^{n} g_i x^i$$

onda su koeficijenti polinoma proizvoda rezultat diskretne konvolucije nad $m + n + 1$ *dimenzionalnim vektorima* $(f_1, f_2, \ldots, f_m, 0, \ldots, 0)^T$ *i* $(g_1, g_2, \ldots, g_n, 0, \ldots, 0)^T$. *Dodatno, celi brojevi se mogu predstaviti kao vrednosti polinoma sa koeficijentima koji odgovaraju ciframa broja, izračunatog za osnovu brojčanog sistema.*



Slika 3.33: Ilustracija upotrebe Gausovog zamućenja na slikama.

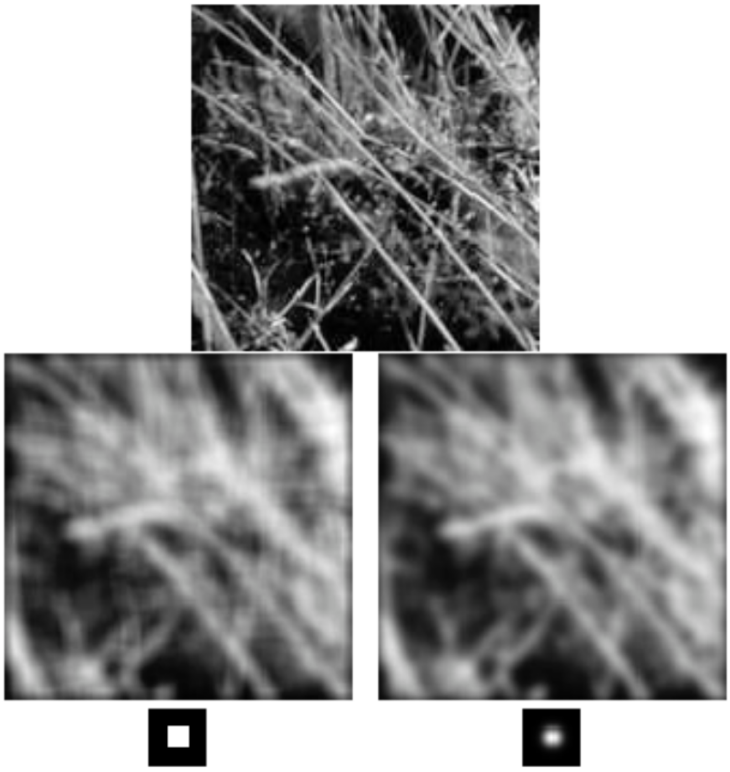
Stoga, diskretna konvolucija se može koristiti za brzo računanje proizvoda brojeva sa velikim brojem cifara.

Konvolucija se intenzivno koristi u obradi signala. Jedna funkcija u konvoluciji predstavlja signal, a druga je obično jednostavnija i predstavlja specifičnu funkciju kojom se postiže neki efekat transformacije signala i naziva se *filterom*.

Primer 39 *Česta primena konvolucije je u obradi slika. Pretpostavka je da su slike u nijansama sive. Klasičan primer filtera je Gausovo zamućenje, čiji su efekti prikazani na slici 3.33, predstavlja konvoluciju sa Gausovim zvonom*

$$\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Kako je Gausovo zvono normirana funkcija, jasno je da konvolucija sa njime predstavlja otežano uprosečavanje. Pritom, prilikom izračunavanja vrednosti piksela u zamućenoj slici, najveća težina se daje vrednosti tog istog piskela u polaznoj slici, dok doprinosi ostalih piksela eksponencijalno opadaju sa rastojanjem od tog piskesla.

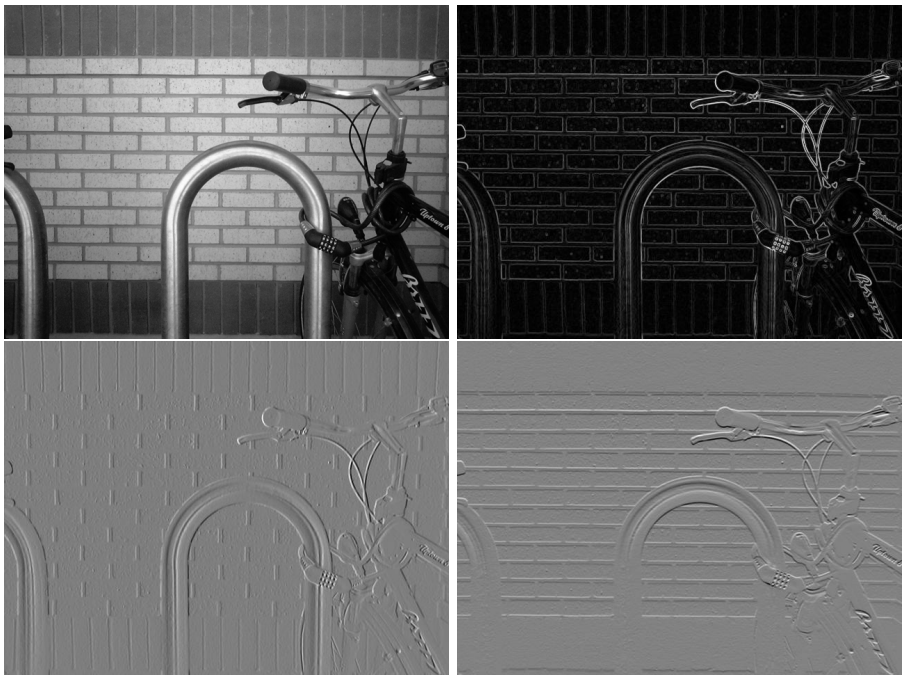


Slika 3.34: Efekti uprosečavanja (levo) i Gausovog filtera (desno) na slici trave i na slikama pojedinačnog piksela.

Kako su slike diskretni objekti, vrednosti Gausovog zvona se izračunavaju u diskretnim tačkama. U slučaju obrade slika, filteri se često predstavljaju matricama. Još jedan filter koji se koristi za zamućenje slika je uprosečavanje, kojem odgovara sledeća matrica:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Gausov filter bolje uklanja iz slika visoke frekvencije, a samim tim i detalje. Poređenje ova dva filtera je dato na slici 3.34. Još jedan zanimljiv zadatak u obradi slika je detekcija ivica. Ivice predstavljaju tačke naglih skokova u vrednosti osvetljenja slike. Nagli skokovi odgovaraju visokim vrednostima izvoda ili razlika susednih vrednosti piksela, kojima se izvodi često aproksimiraju. Za početak biće razmotreni samo izvodi u horizontalnom i vertikalnom pravcu, koji odgovaraju dvema komponentama gradijenta. Izvodi slike A u horizontalnom i vertikalnom pravcu se često izračunavaju konvolucijama sa Sobel-Feldmanovim



Slika 3.35: Polazna slika, intenzitet gradijenta i njegove komponente u horizontalnom i vertikalnom pravcu.

filterima:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A \qquad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

Aproksimacija intenziteta gradijenta je onda

$$G = \sqrt{G_x^2 + G_y^2}$$

Ilustracija detekcije ivica na ovaj način data je na slici 3.35. Umesto ovih filtera, mogli su biti korišćeni i jednostavniji. Na primer, horiznotalni filter bi mogao biti $[1 \ -1]$*. Međutim, vrednosti konvolucije sa ovim filterom ne bi bile pridružene pikselima, već sredinama između njih, što je nepoželjno. Modifikacija* $[1 \ 0 \ -1]$ *rešava ovaj problem, ali je, kao i prethodna verzija, osetljiva na šum. Kako bi se to rešilo, mogli bi se uključiti uticaji susednih piksela i koristiti filter*

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

međutim, ovaj filter daje podjednak značaj i tekućem pikselu i njegovim susetimda. Sobel-Feldmanov filter se i u tom smislu ponaša poželjnije.

Primer 40 *Konvoluciju je moguće primeniti za brzo nalaženje uzorka slike g u drugoj slici f. Slike f i g se često prvo predprocesiraju oduzimanjem njihovih proseka i deljenjem njihovim standardnim devijacijama. Ako je f' uzorak slike f istih dimenzija kao uzorak g, prirodan kriterijum sličnosti je srednjekvadratno odstupanje*

$$\sum_{i=1}^{m}\sum_{j=1}^{n}(f'_{ij}-g_{ij})^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}(f'^2_{ij}-2f'_{ij}g_{ij}+g^2_{ij}) = \sum_{i=1}^{m}\sum_{j=1}^{n}f'^2_{ij}-2\sum_{i=1}^{m}\sum_{j=1}^{n}f'_{ij}g_{ij}+\sum_{i=1}^{m}\sum_{j=1}^{n}g^2_{ij}$$

Samo srednja suma izražava odnos između f' i g i predstavlja jednu vrednost diskretne konvolucije slike f' i uzorka g, reflektovanog u odnosu i na x i na y osu. Lokacija uzorka g u slici f, odgovara maksimumu konvolucije slike f sa reflektovanim uzorkom g, kao što je ilustrovano na slici 3.36. Takođe, imajući u vidu oduzimanje proseka i deljenje standardnom devijacijom, treba primetiti da ova konvolucija izračunava koeficijent korelacije u svim tačkama slike f.

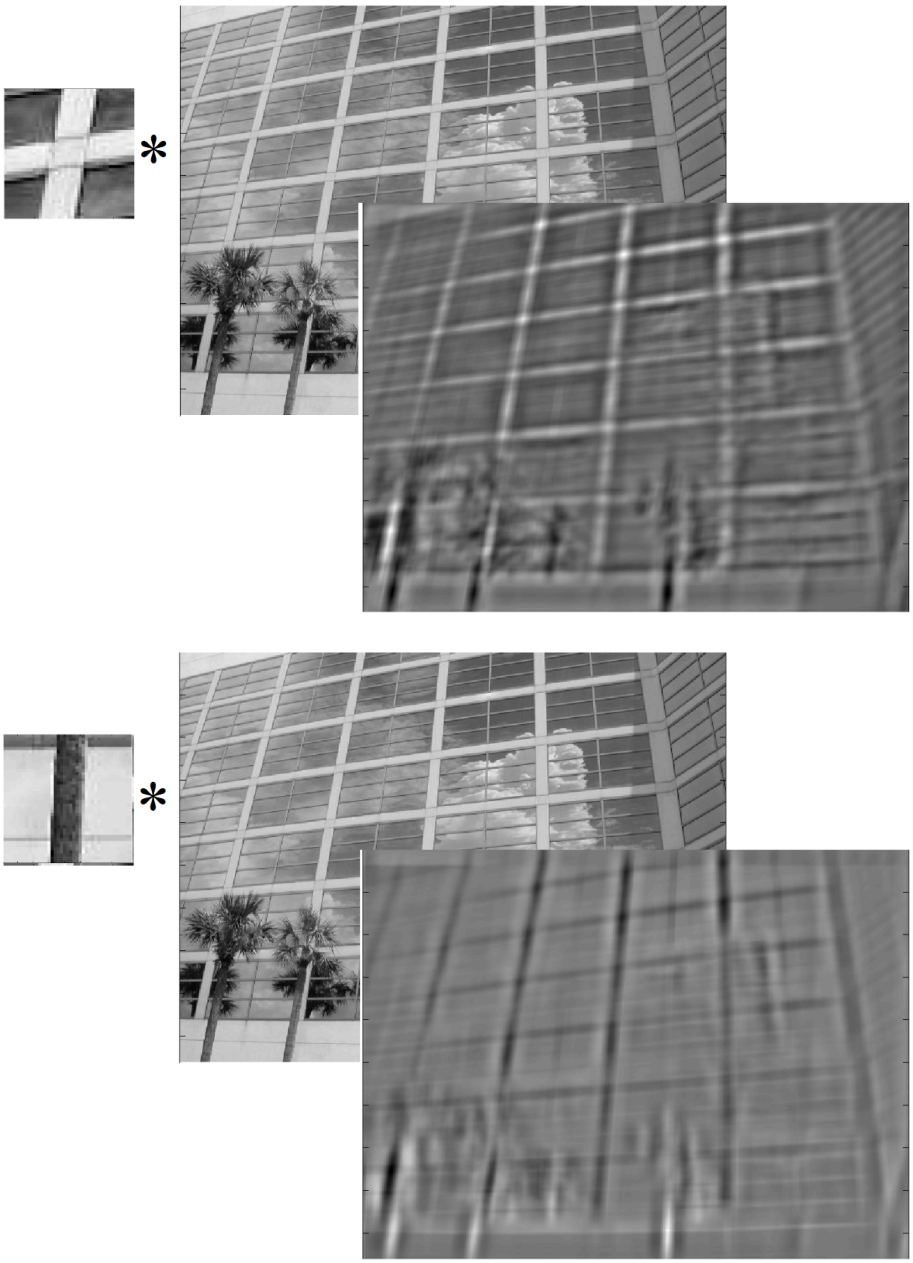
3.5 Osnovni koncepti obrade signala

U nastavku će biti dat osvrt na najjednostavnije teme obrade signala – uzorkovanje, curenje spektra i filtriranje.

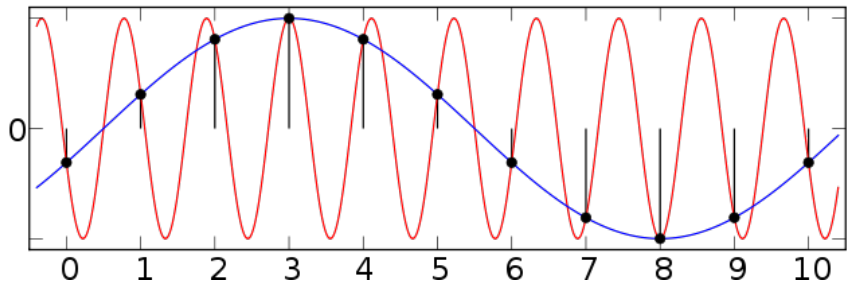
3.5.1 Uzorkovanje

U praksi se zbog diskretne prirode većine računskih mašina najčešće operiše sa diskretnim reprezentacijama signala. Da bi se dobile ovakve reprezentacije, neophodno je proći kroz proceduru uzorkovanja signala i njegove kvantizacije. Ukoliko je, recimo, reč o zvučnim signalima, uzorkovanje predstavlja odabir vremenskih trenutaka u kojima će se meriti jačina zvuka, a kvantizacija odabir numeričke skale za predstavljanje izmerenih vrednosti (na primer, vrednosti se mogu zapisivati pomoću jednog bajta, pomoću dva, itd).

Ako se na svakih T sekundi vrši odabir uzorka signala tj. merenje relevantnih parametara signala, govori se o uzorkovanju signala sa frekvencijom uzorkovanja f_s. Veza koja važi između frekvencije uzorkovanja i vremena koje protekne između dobijanja uzastopnih uzoraka je $f_s = \frac{1}{T}$. Ova veličina se izražava u Hercima. Frekvencija uzorkovanja se može izražavati i u radijanima, računa se vezom $\omega_s = \frac{2\pi}{T} = 2\pi f_s$ i naziva kružnom frekvencijom. U nastavku će biti reči o frekvenciji f_s, ali bi bilo moguće govoriti i o kružnoj frekvenciji. Ukoliko postoji neka frekvencija f_b za koju važi $f_b > f$ gde f predstavlja frekvenciju signala izraženu u Hercima, kaže se da je signal ograničen, a frekvencija f_b se zove granična frekvencija. Informacija o ograničenosti signala je osobito važna pri izboru frekvencije uzorkovanja jer se prema Najkvistovoj



Slika 3.36: Ilustracija pretrage slike. Zarad lakšeg razumevanja, uzorak nije reflektovan, a trebalo bi da bude prilikom konvolucije. Vidi se da su lokacije na slici, koje se najbolje poklapaju sa uzorkom, najsvetlije.



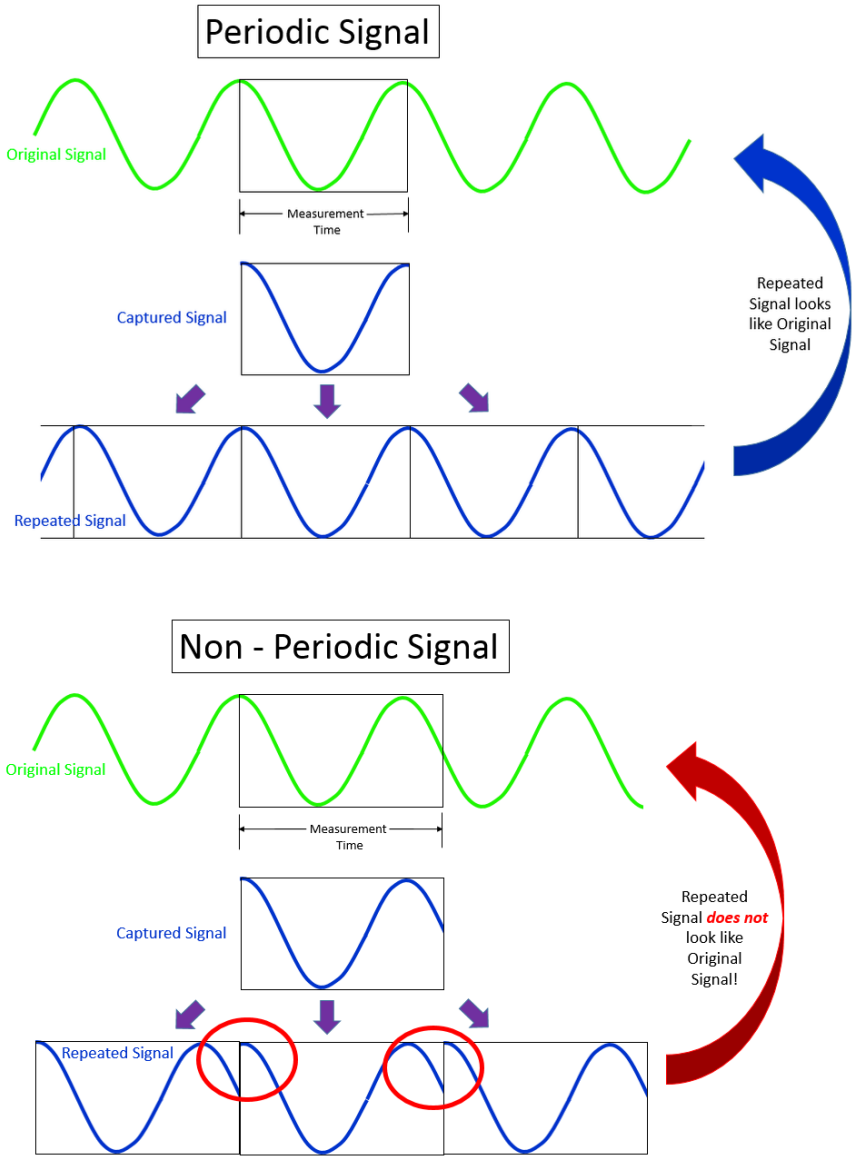
Slika 3.37: Dati uzorak ne razlikuje dve prikazane sinusoide.

teoremi signal može verodostojno reprodukovati samo ako je frekvencija uzorkovanja *više* od dva puta veća od granične frekvencije. Zato se, recimo, zbog ograničenosti čulnog aparata čoveka i maksimalne frekvencije od oko $20KHz$ koje uvo može da detektuje, najčešće vrši uzorkovanje zvučnog signala frekvencijom $44.1KHz$.

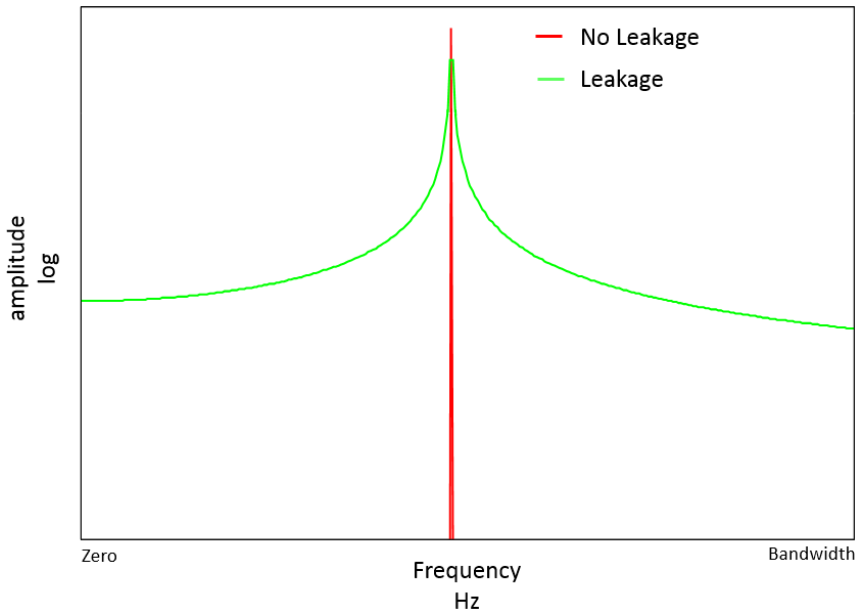
Ukoliko frekvencija uzorkovanja nije više od dva puta veća od svih frekvencija u signalu, dolazi do njegove pogrešne rekonstrukcije zbog problema *alijasovanja* (eng. *aliasing*). Neka se na obodu diska koji rotira u smeru kazaljke na satu brzinom od jednog obrataja u sekundi, odnosno frekvencijom od $1Hz$, nalazi tačka. Na osnovu fotografija tačke napravljenih deset puta u sekundi, može se steći jasna predstava o njenom kretanju. Na osnovu dve fotografije po sekundi, što odgovara frekvenciji od $2Hz$, lako je primetiti da kretanje postoji, ali ne i u kom smeru. Otud, frekvencija mora biti više nego duplo viša od najviše frekvencije u signalu. U slučaju 3 fotografije na 2 sekunde, što je $1.5Hz$, delovalo bi čak da se tačka kreće u suprotnom smeru. Još jedan primer ovog fenomena može se videti na slici 3.37. Ovakvi problemi su fundamentalni u rekonstrukciji signala.

3.5.2 Curenje spektra

Razvoj u Furijeov red i diskretna Furijeova transformacija pretpostavljaju periodičnost signala i diksretan frekvencijski domen. Realističnost ovih pretpostavki u praksi je upitna. Signal najčešće nije periodičan. Čak i ako jeste, ako nije savršeno uzorkovan u dužini perioda, dobijeni uzorak neće imati neprekidno periodično ponašanje i očekuju se prekidi ukoliko se uzorak periodično nadoveže. Ovo je prikazano slikom 3.38. Ti prekidi dovode do pojave visokih frekvencija u frekvencijskom spektru, koje nisu zaista prisutne u spektru originalnog signala. Neka je softver za analizu spektra kalibrisan tako da izražava frekvencije u celobrojnim Hercima (naspram oscilacija u periodima trajanja 2π, što se postiže skaliranjem vremenske ose u jednačinama koeficijenata Furijeovog reda). Signal frekvencije $3Hz$ prilikom Furijeove transformacije daje vrlo jasan pik za frekvenciju $3Hz$. Postavlja se pitanje šta bi trebalo da bude



Slika 3.38: Primer savršenog (gore) i nesavršenog (dole) uzorkovanja signala.

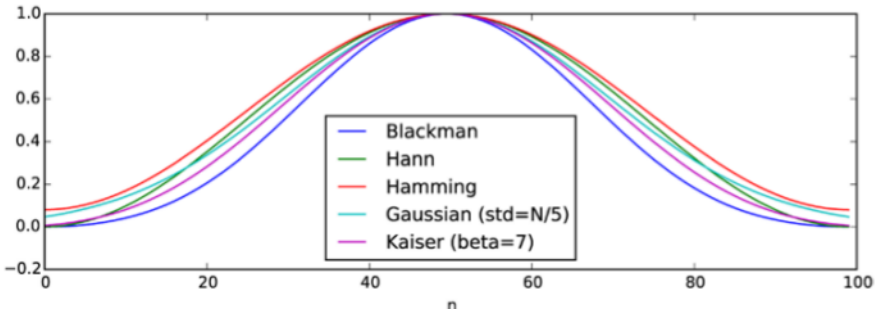


Slika 3.39: Frekvencijski spektar signala kod kojeg nema curenja spektra (crveno) i signala kod kojeg ga ima (zeleno).

ishod u slučaju signala frekvencije $2.8Hz$. Iako bi se naivnom intuicijom moglo očekivati da ishod neće biti savršen, ali da će se javiti pikovi u okolini frekvencije $2.8Hz$, to nije tačno. Zapravo, dolazi do fenomena poznatog pod nazivom *curenje spektra*, odnosno do toga da se taj signal predstavlja u celom frekvencijskom spektru, što je predstavljeno slikom 3.39. Kako se jačina signala rasporedila duž više frekvencija i jačina signala u središnjoj frekvenciji je manja.

Kako bi se ovaj problem ublažio, pribegava se modifikovanju signala. Jedan način za to predstavljaju *prozorske funkcije*, kojima se u vremenskom domenu množi polazni signal tako da se u krajevima potiskuje ka nuli. Prozorske funkcije obično zadovoljavaju određena jednostavna svojstva kao da su svuda izvan nekog intervala jednake nuli i da maksimum dostižu na polovini tog intervala na kojem su različite od nule. Neke od prozorskih funkcija koje se koriste u praksi su:

- Blekmanova $w(n) = a_0 - a_1 \cdot \cos(\frac{2\pi n}{N-1}) + a_2 \cdot \cos(\frac{4\pi n}{N-1})$, $a_0 = \frac{1-\alpha}{2}$, $a_1 = \frac{1}{2}$, $a_2 = \frac{\alpha}{2}$
- Hanova $w(n) = 0.5 \cdot [1 - \cos(\frac{2\pi n}{N-1})]$
- Hamingova funkcija $w(n) = 0.54 - 0.46 \cdot \cos(\frac{2\pi n}{N-1})$



Slika 3.40: Grafici nekih prozorskih funkcija.

- Kasijerova funkcija $w(n) = \frac{I_0(\pi\alpha\sqrt{1-(\frac{2n}{N-1}-1)^2}}{I_0(\pi\alpha)}$, $\beta = \pi\alpha$ u kojoj je I_0 Beselova funkcija nultog reda.

Vrednost N u definicijama ovih funkcija odgovara širini prozora. Njihovi grafici prikazani su na slici 3.40. Prozorske funkcije koriste se i u drugim kontekstima osim popravljanja curenja spektra.

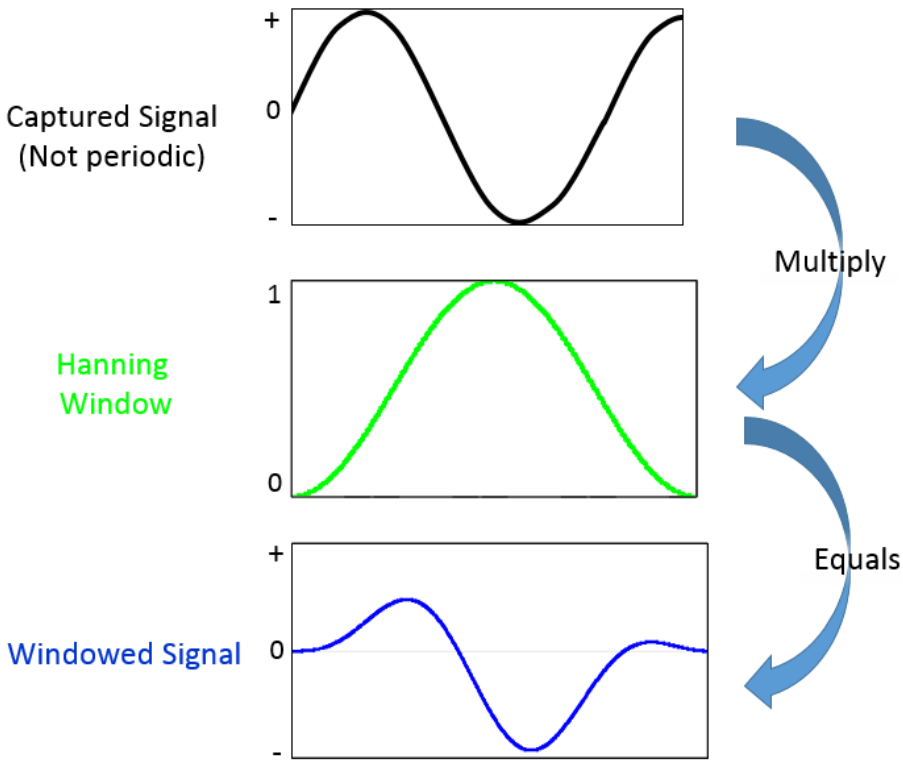
Na slici 3.41, prikazana je primena prozorske funkcije na neperiodičan signal. Dobijeni signal je osetno deformisan. S druge strane, kao što pokazuje slika 3.42, moguće je izvršiti njegovo periodično i neprekidno nadovezivanje. Postavlja se pitanje – da li je veća korist ili šteta? Uprkos deformisanju signala, zahvaljujući primeni prozorske funkcije, spektar više ne sadrži mnoštvo visokih frekvencija, što ipak dovodi do kvalitetnije reprezentacije u frekvencijskom domenu, kao što pokazuje slika 3.43.

Dosadašnji primeri su se fokusirali na vrlo jednostavne signale, predstavljene jednom sinusoidom. Realni signali predstavljaju dosta kompleksniju kombinaciju osnovnih harmonika (sinusoida različite frekvencije i faznog pomeraja). U takvim slučajevima mešanjem curenja različitih harmonika, dolazi do još izraženijih problema. Curenje spektra u slučaju signala koji se sastoji od dva harmonika prikazan je na slici 3.44. Očito upotreba prozorske funkcije olakšava razaznavanje komponenti signala.

3.5.3 Filtriranje signala

Filtriranje signala obuhvataju širok skup operacija koje se mogu izvoditi nad signalima. Filtriranje se obično koristi kao prvi korak selekcije informacija koje signal nosi. Može se vršiti u polaznom (vremenskom ili prostornom) domenu ili u frekvencijskom domenu.

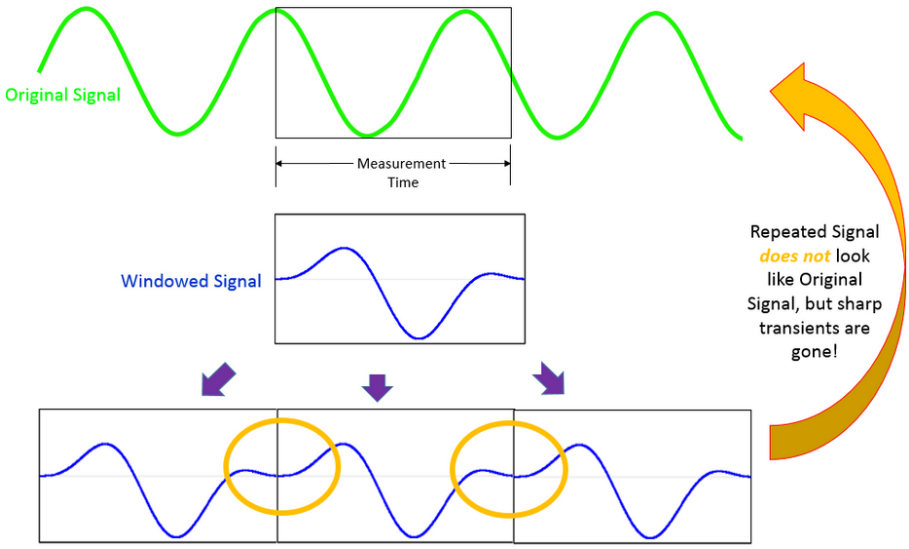
Jedan od načina da se realizuje filtriranje signala bi bio da se primeni Furijeova transformacija nad signalom, zatim da se na nivou dobijene reprezentacije u frekvencijskom domenu izvrši modifikacija signala, i konačno, da se ovako modifikovana reprezentacija inverznom Furijeovom transformacijom vrati u polazni



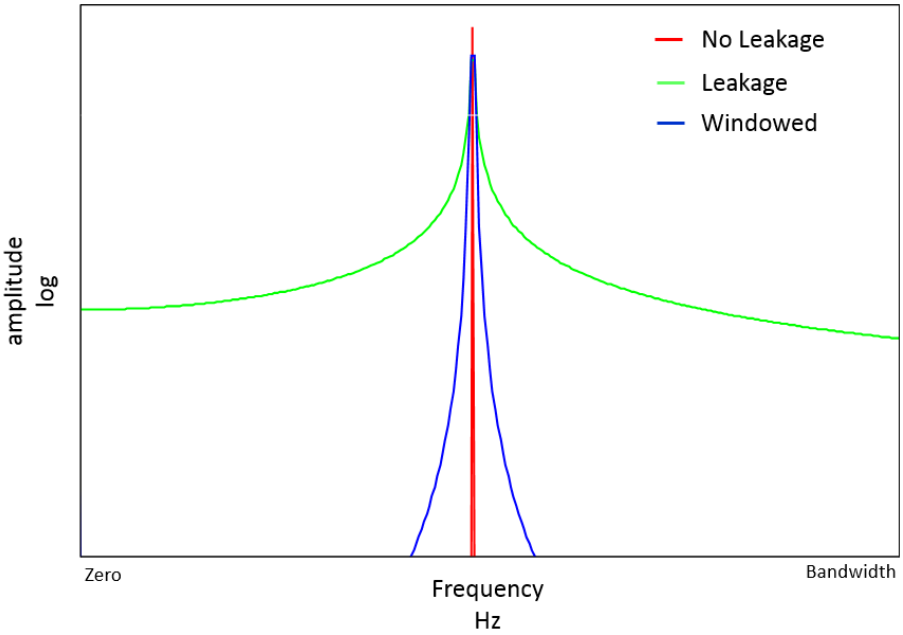
Slika 3.41: Primena prozorske funkcije na signal.

domen. Iako sasvim validan, ovaj pristup može dugo trajati i zahteva poznavanje celog uzorka, pa se u praksi njime ne mogu realizovati obrade u realnom vremenu (kako zbog brzine, tako i zbog nepoznavanja budućih vrednosti). Zbog toga se pribegava rešenjima koja se zasnivaju na konvoluciji signala i koja se realizuju u polaznom domenu. Jedan od signala koji učestvuje u konvoluciji je uvek ulazni signal, dok je drugi signal iz grupe predefinisanih signala kojima se može ostvariti željeni efekat. Signali iz druge grupe se generišu softverski ili hardverskim modulima koji se, takođe, nazivaju filterima.

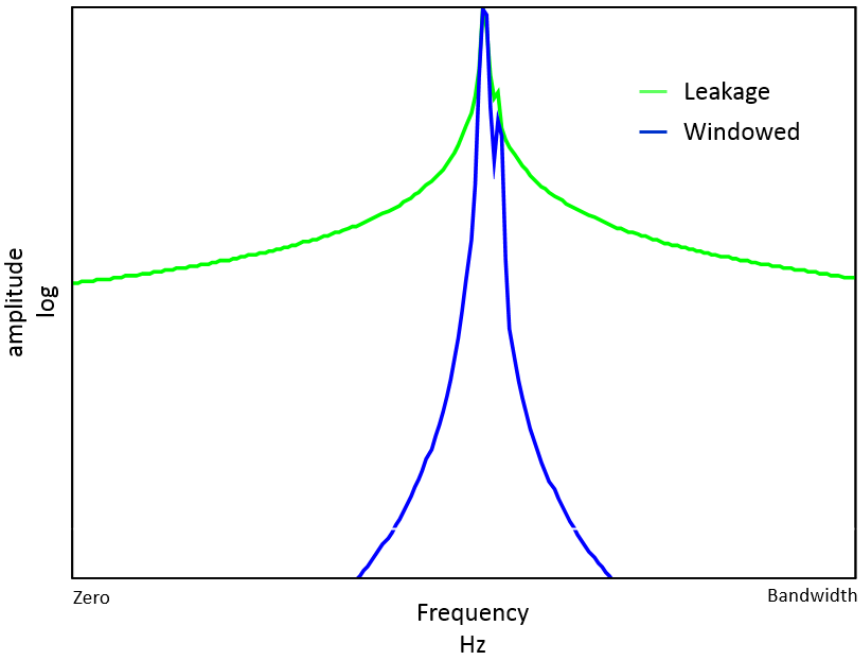
Pre daljih razmatranja vezanih za filtere, važno je zapitati se zašto bi konvolucija bila efikasnija od Furijeove transformacije. Naime, već je konstatovano da naivna implementacija konvolucije zahteva kvadratno vreme. Ukoliko se upotrebni brza Furijeova transformacija, može se izvršiti, brže. Ipak, to nije ništa brže od Furijeove transformacije. Suština je u tome da u vremenskom domenu filteri vrlo često uzimaju vrednost nula u skoro svim tačkama osim u ograničenom broju uzastopnih tačaka. Na taj način, vreme obrade je linearno, a budući uzorci signala nisu potrebni, ako su odgovarajuće vrednosti filtera jednake nuli. Na primeru Gausovog zamućenja sa malom standardnom devi-



Slika 3.42: Periodično i neprekidno proširenje signala zahvaljujući primeni prozorske funkcije.



Slika 3.43: Frekvencijski spektar u slučaju signala bez curenja spektra (crveno), sa curenjem spektra (zeleno) i nakon primene prozorske funkcije (plavo).



Slika 3.44: Curenje spektra signala sastavljenog od dva harmonika (zeleno) i rezultat primene prozorske funkcije (plavo).

jacijom, filter se opisuje malim brojem nenula koeficijenata zahvaljujući tome što Gausovo zvono eksponencijalno brzo opada.

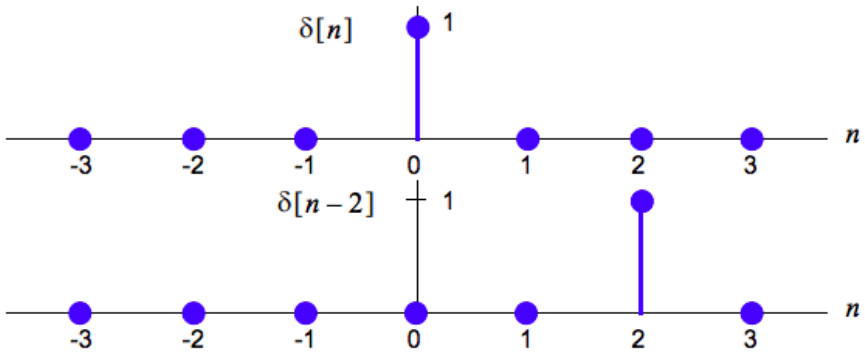
Filteri deluju nad ulaznim signalima, koje transformišu i daju izlazne signale. Kao važan slučaj razmatraju se *linearni vremenski invarijantni sistemi* (eng. *linear time-invariant systems*) koji za linearne kombinacije ulaznih signala generišu linearne kombinacije izlaznih signala i čije ponašanje se ne menja u zavisnosti od vremena. Matematički se ova svojstva za linearni invarijantni sistem H koji preslikava signal $x(t)$ u signal $y(t)$ opisuju jednakostima:

- $H(ax(t)) = aH(x(t))$ za sve konstante a i ulazne signale $x(t)$
- $H(x_1(t)+x_2(t)) = H(x_1(t))+H(x_2(t))$ za sve ulazne signale $x_1(t)$ i $x_2(t)$

Očito, ovo je samo svojstvo linearnosti. Svojstvo vremenske invarijantnosti se ogleda u tome da H nije funkcija promenljive t, odnosno da u svakom trenutku na signal deluje na isti način. U suprotnom bi bila korišćena oznaka H_t.

Primer 41 *Preslikavanja*

- $y(t) = tx(t)$ *i*
- $y(t) = 2x(t)$



Slika 3.45: Diskretni impulsi $\delta[n]$ i $\delta[n-2]$.

su oba linearna, ali prvo nije vremenski invarijantno, dok drugo jeste.

Jedna od važnih karakterizacija linearnih vremenski invarijantnih sistema je *impulsni odgovor sistema*. Pod impulsima se podrazumevaju kratkotrajni signali u vremenskom domenu, najčešće Dirakova δ funkcija. Treba imati u vidu razliku u definicijama ove funkcije u kontinualnom i u diskretnom slučaju. U kontinualnom, ona u nuli uzima vrednost $+\infty$, a u diskretnom vrednost 1. Na slici 3.45 dat je primer dva diskretna impulsa $\delta[n]$ i $\delta[n-2]$, gde uglaste zagrade naglašavaju diskretnu prirodu funkcije, što je konvencija u literaturi vezanoj za obradu signala. Impulsni odgovor sistema predstavlja odgovor sistema za ovakve ulaze. Formalno, ako je ulaz sistema signal $x[n] = \delta[n]$, impulsni odgovor sistema je signal $h[n] = H(\delta[n])$. Zbog prirode linearnih invarijantnih sistema ova informacija je dovoljna za određivanje izlaznog signala svih ulaza. Naime, diskretni signali se mogu zapisati u vidu sume

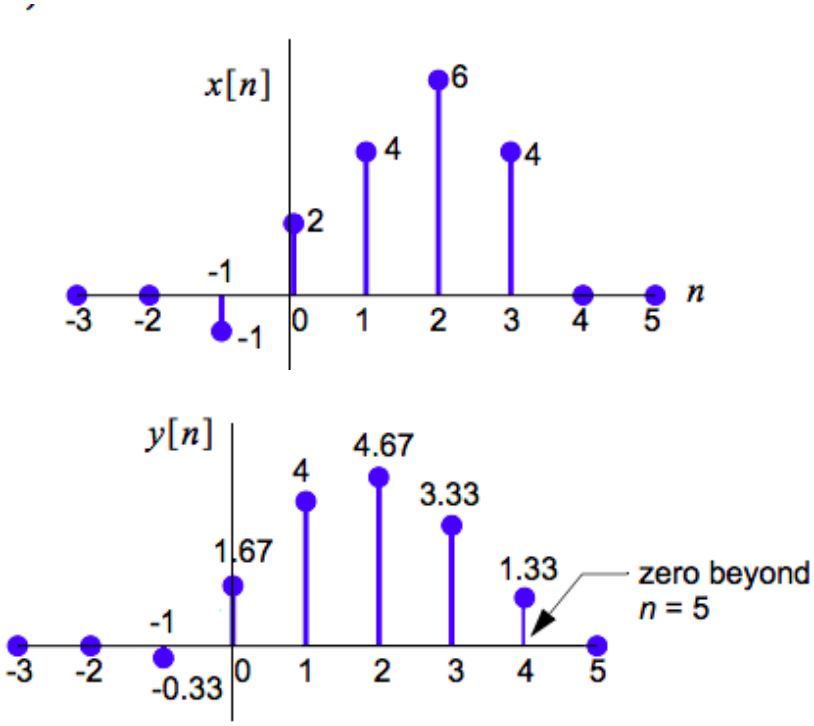
$$x[n] = \sum_k x[k]\delta[n-k]$$

pa se njima odgovarajući izlazi mogu zapisati kao

$$y[n] = H(x[n]) = \sum_k x[k]h[n-k]$$

Pored impulsnih odgovora, može se govoriti i o frekvencijskim odgovorima, ali o njima će biti reči samo u specifičnom kontekstu FIR filtera.

O filterima se može razmišljati kao o posebnoj vrsti sistema. Njihova dalja podela se može vršiti prema prirodi signala nad kojim se primenjuju, po načinu na koji se hardverski implementiraju, po efektima koji se njihovom primenom postižu, itd. Po prirodi signala, filteri se dele na analogne i digitalne. Analogni filteri se primenjuju nad analognim signalima, dok se digitalni filteri primenjuju na digitalnim signalima. Nadalje će biti reči samo o digitalnim filterima.



Slika 3.46: Prikaz ulaznog i uprosečenog signala.

Prema dužini impulsnog odgovora filteri se dele na *filtere sa konačnim trajanjem impulsnog odgovra* iliti FIR filtere (eng. *finite impulse response*) i *filtere sa beskonačnom dužinom impulsnog odgovora* iliti IIR filtere (eng. *infinite impulse response*) filtere.

FIR filteri imaju konačne impulsne odgovore i mogu se predstaviti kao težinska suma prethodnih, trenutnih i, moguće, budućih ulaza filtera

$$y[n] = \sum_{i=-M_1}^{M_2} b_i x[n-i]$$

Primer jednog jednostavnog filtera bi bio filter uprosečavanja triju uzastopnih vrednosti

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$$

Na primer, za signal prikazan na slici 3.46 izlaz $y[2]$ se može izračunati po formuli $y[2] = \frac{1}{3}(6+4+2) = 4$. Primenom ovog filtera dobija se izlaz koji je glatkiji u odnosu na ulaz i prikazan je na istoj slici.

Pomenuti frekvencijski odgovor sistema oslikava kako sistem reaguje na ulaze oblika

$$x[n] = e^{i\omega n}$$

odnosno harmonike. Po definiciji FIR filtera, važi

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] = \sum_{k=0}^{M} b_k e^{i\omega(n-k)} = e^{i\omega n} \sum_{k=0}^{M} b_k e^{-i\omega k}$$

Pod frekvencijskim odgovorom filtera, podrazumeva se funkcija

$$H(\omega) = \sum_{k=0}^{M} b_k e^{-i\omega k}$$

Ova funkcija ima svoj moduo iliti amplitudu i argument iliti fazu. Dejstvo filtera na neki harmonik je to da se njegova amplituda množi amplitudom frekvencijskog odgovora, a faza sabira sa fazom frekvencijskog odgovora. Kao i u slučaju impulsnog odgovora, kako se signali mogu razložiti na harmonike, dovoljno je poznavati frekvencijski odgovor filtera da bi filter bio definisan. Koji je značaj ovoga? Recimo, ukoliko je amplituda frekvencijskog odgovora filtera za harmonik neke frekvencije jednaka nuli, to znači da taj filter eliminiše tu frekvenciju iz signala.

Jedan od načina predstavljanja filtera je preko para grafika modula i faze frekvencijskog odgovora. Na primer, za FIR filter

$$y[n] = x[n] + x[n-1] + 3x[n-2] + x[n-3] + x[n-4]$$

dobija se frekvencijski odgovor u obliku

$$H(\omega) = e^{-i2\omega}[2cos(2\omega) + cos(\omega) + 3]$$

za koji su, dalje, moduo

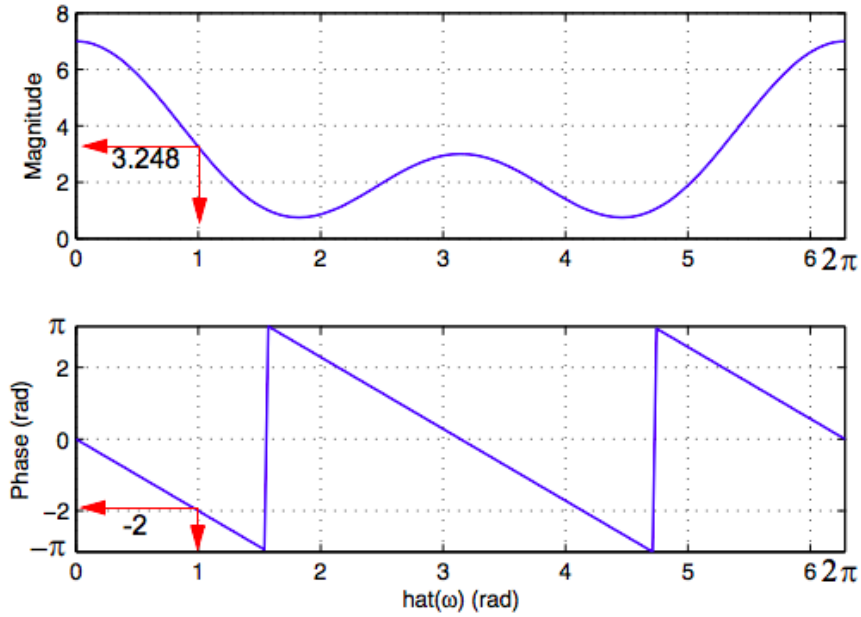
$$|H(\omega)| = 3 + 2cos(\omega) + 2cos(2\omega)$$

i argument

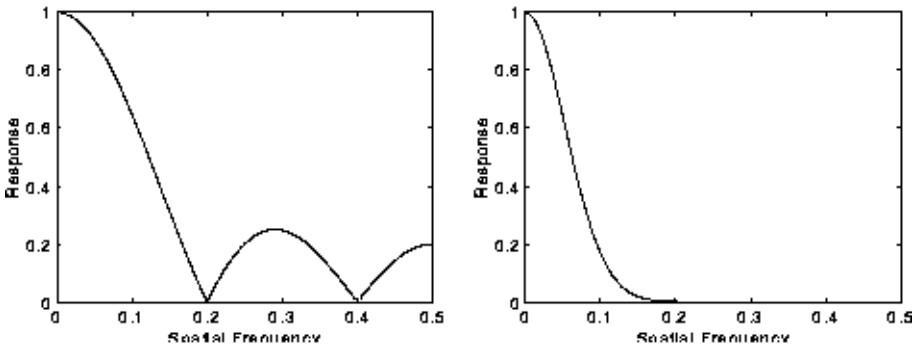
$$arg(H(\omega)) = -2\omega$$

Njima odgovarajuci grafici su dati na slici 3.47. Sa slike se vidi da dejstvom ovog filtera, frekvencije oko 1.8 i 4.5 bivaju praktično eliminisane iz signala dok frekvencije 0 i 2π bivaju pojačane oko 7 puta.

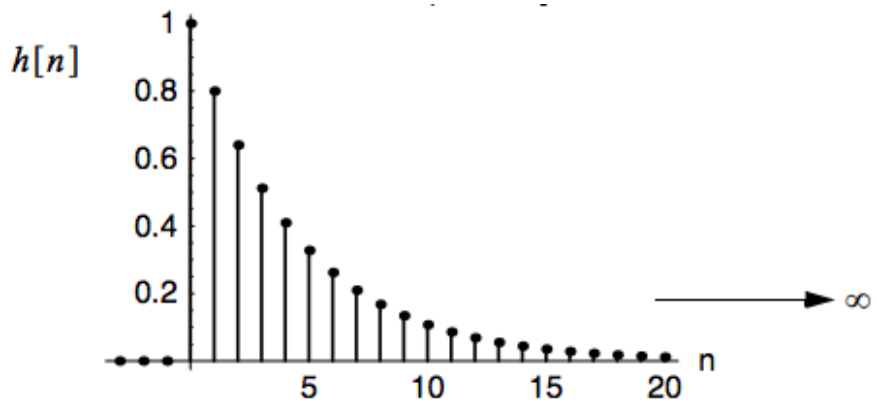
Primer 42 *Zamućenje prostim uprosečavanjem, Gausovo zamućenje i Sobel-Feldmanov filter predstavljaju primere FIR filtera. Na slici 3.48 prikazana je amplituda frekvencijskog odgovora zamućenja uprosečavanjem i Gausovog zamućenja. Očito Gausovo zamućenje ima bolja svojstva – ukoliko uklanja neku frekvenciju iz signala, uklanja i sve frekvencije više od nje. Za prosto uprosečavanje to ne važi, što je neočekivano ponašanje.*



Slika 3.47: Amplituda i faza frekvencijskog odgovora filtera.



Slika 3.48: Amplituda frekvencijskog odgovora prostog uprosečavanja (levo) i Gausovog zamućenja (desno).



Slika 3.49: Impulsni odgovor jednog jednostavnog IIR filtera.

IIR filteri su filteri čiji impulsni odgovori zavise od tekućih i prethodnih ulaza, ali i od prethodnih izlaza filtera. Izražavaju se formulama oblika

$$y[n] = \sum_{l=1}^{N} a_l y[n-l] + \sum_{k=0}^{M} b_k x[n-k]$$

sa ukupno $N+M+1$ koeficijenata. Jedan primer IIR filtera bi bio filter prvog reda za koji je N=1 i M=0 i čija je jednačina

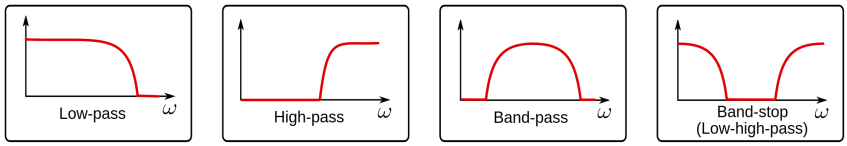
$$y[n] = a_1 y[n-1] + b_0 x[n]$$

Za ovaj filter se impulsni odgovor može odrediti zamenom $x[n] = \delta[n]$ uz uslov da je sistem inicijalno u stanju mirovanja tj. da važi $x[0] = 0$ i $y[0] = 0$. Rekurentno se stiže do rešenja

$$y[n] = a_1^n b_0 x[0]$$

za $n > 0$. Na slici 3.49 dat je impulsni odgovor filtera prvog reda za koji je $b_0 = 1$ i $a_1 = 0.8$. Odavde se vidi i zašto se ovi filteri nazivaju filterima sa beskonačnim impulsnim odgovorom. Naime, odgovor filtera na impuls koji se jednom desio propagira se beskonačno, samo sa eksponencijalno opadajućim doprinosom.

Na osnovu frekvencijskog odgovora filtera može se izvršiti klasifikacija na niskopropusne filtere, visokopropusne filtere ili filtere propusnike određenih opsega. Niskopropusni filteri su filteri koji propuštaju signale sa frekvencijama koje su niže od zadate. Visokopropusni filteri su filteri koji propuštaju signale sa frekvencijama višim od zadate. Filteri propusnici opsega su filteri koji propustaju signale sa frekvencijama iz zadatog opsega. Na slici 3.50 prikazani su grafici modula frekvencijskih odgovora u zavisnosti od tipa filtera. Granične



Slika 3.50: Impulsni odgovor jednog jednostavnog IIR filtera.

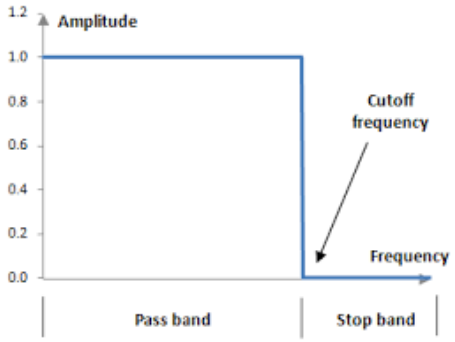
frekvencije koje se pojavljuju u definicijama ovih filtera se najčešće zovu frekvencijama odsecanja.

Čest zadatak u teoriji sistema predstavlja dizajn filtera tj. određivanje koeficijenata bilo FIR bili IIR filtera kojim bi se dobili filteri sa željenim ponašanjem.

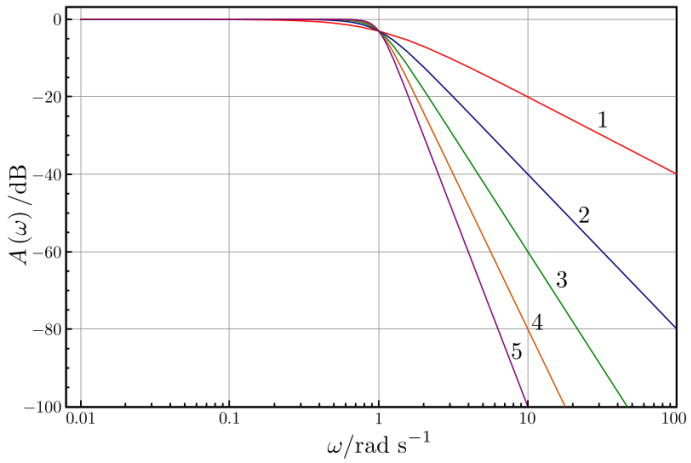
Primer 43 *Magnetna rezonanca je neinvazivna tehnika praćenja moždanih aktivnosti koja se koristi u dijagnostici i istraživanjima koja bi trebala da približe način na koji mozak radi. Kada se u nekom delu mozga poveća aktivnost neurona, MR signali koji odgovaraju ovoj regiji se, takođe, blago uvećaju. Reč je o uvećanju od svega oko* 1% *pa instrumenti koji prate ovakve promene signala moraju biti neosetljivi na različite vrste šumova poput otkucaja srca, disanja, pomeranja ispitanika ili tehničkih svojstava aparature. Tako se, recimo, otkucaji srca mogu videti kao periodični signali sa frekvencijom od* $0.66Hz$ *do* $4Hz$*, disanje kao periodični signal frekvencije od* $0.2Hz$ *do* $0.33Hz$*. Zbog toga se u praksi koriste filteri koji odsecaju niske ili visoke frekvencije koje mogu dovesti do pogrešnih rezultata. Kako su frekvencije neuronskih aktivnosti ispod* $0.15Hz$*, mogu se koristiti niskopropusni filteri sa ovom gornjom granicom.*

Idealan niskopropusni filter bi imao grafik kao na slici 3.51. Ovakve realizacije je tehnički teško ostvariti ako treba da rade u realnom vremenu sa visokom učestalošću uzorkovanja pa se prihvataju aproksimacije željenih ponašanja. Obično postoji opseg frekvencija $[\omega_1, \omega_2]$ takve da frekvencije manje od ω_1 bivaju praktično netaknute, frekvencije više od ω_2 bivaju uklonjene (amplituda im postaje 0), dok amplituda frekvencija u opsegu pada sa rastom frekvencije. Jedan specifičan filter, koji se zbog svojih dobrih svojstava često koristi za filtriranje frekvencija, poznat kao niskopropusni Batervortov filter, ilustrovan je slikom 3.52.

Primer 44 *Već je pomenuto da ukoliko frekvenija uzorkovanja nije veća od dvostruke najviše frekvencije signala, može doći do alijasinga. Rešenje može biti dosta češće uzorkovanje, međutim, češće uzorkovanje vodi većoj količini podataka koje treba čuvati što može biti nepoželjno. Umesto toga, moguće je privremeno uzorkovati na višoj frekvenciji, uz upotrebu niskopropusnog filtera kojim se čiste sve frekvencije koje su više od polovine željene frekvencije uzorkovanja. Na taj način se izbegava alijasovanje, jer te frekvencije više ne ostavljaju tragove u uzorku koji se mogu pogrešno rekonstruisati.*

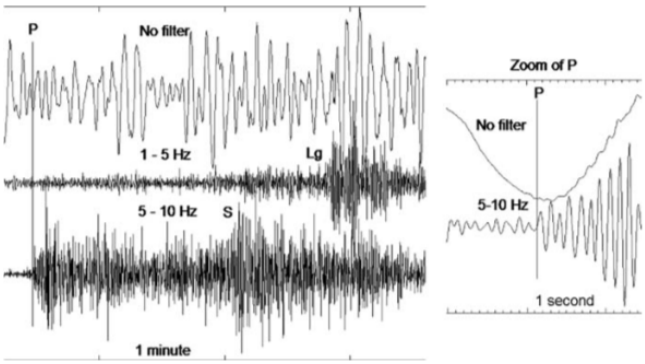


Slika 3.51: Idealan niskopropusni filter.



Slika 3.52: Moduo frekvencijskog odgovora Batervortovih filtera različitog reda.

Primer 45 *Seizmičke promene se prate kroz ponašanja različitih vrsta talasa: primarnih (P) talasa koji iniciraju pojave i čija je brzina oko* $6km/s$*, sekundarnih (S) talasa koji dolaze posle primarnih talasa i čija je brzina oko* $3.5km/s$ *i površinskih talasa koji dolaze posle sekundarnih talasa i čija je brzina između* $3.5km/s$ *i* $4.5km/s$*. Jedan od osnovnih zadataka jeste detektovanje početka P faze zbog daljeg utvrđivanja epicentra zemljotresa i procene njegove jačine. Na slici 3.53 je dat prikaz ulaznog signala i rezultata primene filtera fiksnih opsega. Analizom originalnog signala događaj se ne može detektovati jer je maskiran mikroseizmičkim šumom. Primenom filtera sa opsegom od* $1Hz$ *do* $5Hz$ *i dalje nije moguće detektovati početak P faze. Konačno, primenom filtera sa opsegom od* $5Hz$ *do* $10Hz$*, mogu se jasno detektovati počeci P i S faze.*



Slika 3.53: Ilustracija detektovanja primarnih i sekundarnih seizmičkih talasa. Pre filtriranja ili nakon filtriranja u pogrešnom opsegu P i S faza se ne mogu uočiti. Međutim, nakon filtriranja u pravom opsegu, mogu.

3.6 Talasići

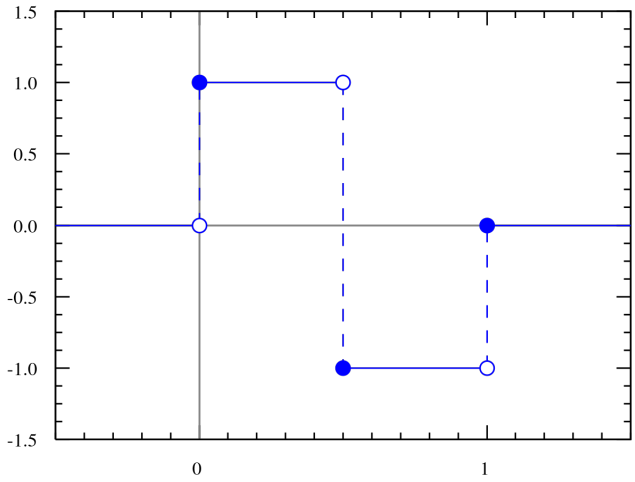
Sistem trigonometrijskih funkcija koji je korišćen u prethodnim odeljcima je samo jedan od mogućih sistema i nije najbolji izbor za primenu u svakom problemu. Pored pretpostavke o periodičnosti, korišćenje trigonometrijskog sistema ne dozvoljava lokalizaciju frekvencija. Naime, ukoliko je neka frekvencija prisutna u signalu, prisutna je u toku celog trajanja signala, ali se u intervalima u kojima nije izražena (npr. ne čuje se, ako se radi o zvuku) poništava kombinovanjem sa drugim frekvencijama. Ovo može otežati analizu signala. Takođe, trigonometrijski sistem nije pogodan za aproksimaciju funkcija koje nisu glatke, na primer u slučaju analize seizmičkih podataka, Braunovog kretanja čestica, slika otisaka prstiju itd. U takvim slučajevima, aproksimacija trigonometrijskim sistemom se oslanja na sinusoide visokih frekvencija. Alternativa je korišćenje sistema koji po konstrukciji odgovara ovakvim situacijama. Ključna ideja je definisati na konačnom intervalu osnovnu funkciju, koja ne mora biti glatka, čak ni neprekidna. Ovakva funkcija se naziva osnovnim *talasićem*, a od nje se generiše sistem *talasića* translacijama i skaliranjem. Jedan primer osnovnog talasića je Harova funkcija

$$\psi(x) = \begin{cases} 1 & x \in [0, \frac{1}{2}) \\ -1 & x \in [\frac{1}{2}, 1) \\ 0 & x \notin [0, 1) \end{cases}$$

prikazana na slici 3.54, a ostali talasići, koji čine ortonormirani sistem kojim se mogu proizvoljno dobro aproksimirati funkcije prostora $L^2(\mathbb{R})$ su definisani na sledeći način

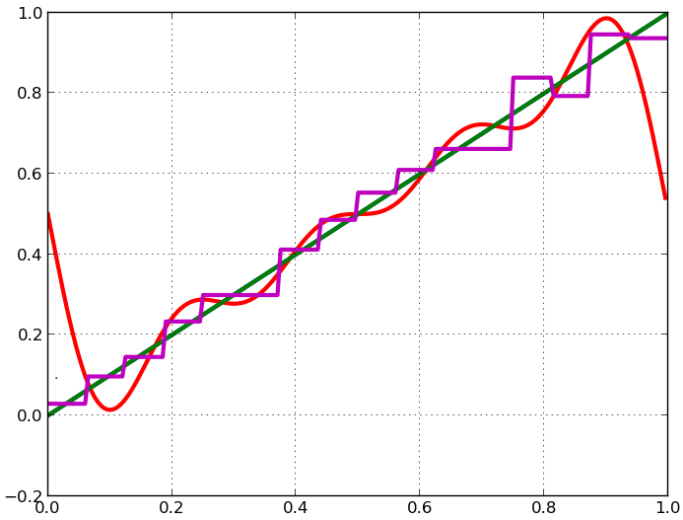
$$\psi_{ij} = 2^{i/2}\psi(2^i x - j) \qquad i, j \in \mathbb{Z}$$

Na slici 3.55 su prkazani i neki drugi primeri osnovnih talasića.



Slika 3.54: Harov osnovni talasić.

Slika 3.55: Neki osnovni talasići.



Slika 3.56: Aproksimacija funkcije $y = x$ (zeleno) pomoću talasića (ljubičasto) i pomoću trigonometrijskog polinoma (crveno).

Očigledno menjanjem parametra j, vrši se translacija elementa baze, dok se menjanjem parametra i, vrši njegovo skaliranje – kako se povećava i, povećava se i amplituda talasića, a smanjuje oblast na kojoj je različit od nule. Pored Harove funkcije, moguće je koristiti i druge osnovne talasiće, ali je bitno od njih konstruisati ortonormirani sistem poput Harovog. Kao i u slučaju trigonometrijskog sistema, moguće je definisati različite varijante *talasaste transformacije*, od kojih je u praksi najvažnija *diskretna talasasta transformacija*. Analogno algoritmu za brzu Furijeovu transformaciju, postoji i algoritam za *brzu talasastu transformaciju*, ali koji radi u vremenu $\Theta(n)$, a ne $\Theta(n \log n)$.

Primer 46 *Slika 3.56 prikazuje aproksimaciju funkcije $y = x$ pomoću talasića i pomoću trigonometrijskog polinoma. Vidi se da je aproksimacija zasnovana na talasićima daje nediferencijabilnu aproksimaciju, ali dosta bližu stvarnoj funkciji, posebno pri krajevima intervala u kojima trigonometrijski polinom značajno odstupa zbog svoje neprekidne i periodične prirode.*

Primer 47 *Jedna od primena talasića je u kompresiji slike. Zapravo, standard JPEG 2000 predviđa upotrebu talasića umesto diskretne kosinusne transformacije. Proces kompresije je drugačiji od onoga zasnovanog na diskretnoj kosinusnoj transformaciji, ali u nastavku neće biti više reči o detaljima. Pristup zasnovan na talasićima pruža viši stepen kompresije, oštrije ivice i izbegava pojavu blokova. Blokovi na slikama kompresovanim pomoću uobičajenog JPEG algoritma, predstavljaju posledcu deljenja slike na blokove dimenzija 8×8 u okviru kojih se zadržavaju niske frekvencije. U ekstremnom slučaju visokog*

Slika 3.57: Prikaz iste slike u standardnom JPEG formatu i formatu JPEG 2000.

stepena kompresije, zadržava se samo prosečna vrednost bloka, što jasno dovodi do pojave jednobojnih blokova i naglih prelaza između njih. Očigledno, problem je utoliko više izražen, što je stepen kompresije viši. JPEG 2000 se ne zasniva na ovom principu i stoga nema izražen ovaj problem. Na slici 3.57, prikazana je ista slika u standardnom JPEG formatu i formatu JPEG 2000.

Glava 4

Numerička linearna algebra

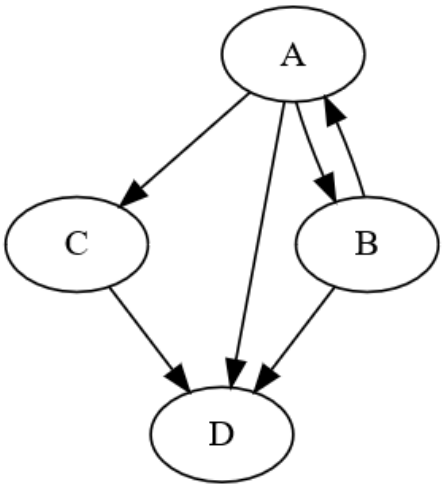
Linearna algebra je praktično nezaobilazan alat u najrazličitijim naučnim i inženjerskim disciplinama. Bilo da se radi o saobraćaju, električnim kolima, genetici, kriptografiji ili nekoj drugoj oblasti, prisustvo linearnih veza između veličina od interesa, predstavlja osnovu za primenu jezika linearne algebre i metoda zasnovanih na njoj. Pored direktnih praktičnih primena u konkretnim domenima, rešavanje problema linearne algebre, poput recimo inverzije matrice, predstavlja nezaobilazan deo raznorodnih matematičkih metoda.

Glavni problemi numeričke linearne algebre su rešavanje sistema jednačina, inverzija matrica, dekompozicija matrica i izračunavanje sopstvenih vektora i sopstvenih vrednosti matrica. Pomenuti problemi su među sobom vrlo povezani.

4.1 Primeri problema numeričke linearne algebre

Jedan od najilustrativnijih primera primene metoda numeričke linearne algebre je problem rangiranja stranica na internetu i poznati algoritam PageRank, jedan od glavnih elemenata Guglovog pristupa pretrazi. Sam algoritam ne uzima u obzir upit pretrage, već se samo tiče rangiranja stranica po značaju.

Primer 48 *Ideja vodilja iza algoritma PageRank je da na značajne stranice pokazuje puno drugih značajnih stranica. Preciznija formulacija pretpostavlja da osoba krstari internetom prateći veze na stranama, pri čemu su izbori veza koje se prate nasumični. Ukoliko strnaica nema veze ni ka jednoj drugoj stranici, izbor sledeće stranice se vrši u skladu sa personalizovanim vektorom preferenci korisnika v, čije koordinate su sve strogo pozitivne, i sumiraju se na 1. Očito v_s se može interpretirati kao verovatnoća da korisnik odluči da poseti stranicu s. Stranica je utoliko značajnija ukoliko je veća verovatnoća da će pri ovakvom krstarenju, korisnik naići na nju. Za modelovanje ovakvog ponašanja korisnika, potrebno je matematički modelovati internet, zanemarujući sve njegove aspekte osim strukture veza među stranicama. Prirodni matematički formalizam kojim se ovakva struktura može modelovati je usmereni graf. Neka je*



Slika 4.1: Primer usmerenog grafa.

skup strana, njegovih čvorova, označen sa S i neka je $|S| = N$. Neka je K skup strana bez veza ka drugim stranama. Ponašanje korisnika se može izraziti u terminima verovatnoće. Kako je do svake stranice moguće doći preko stranica koje ukazuju na nju i kako se sa tih stranica sa jednakom verovatnoćom može otići na bilo koju stranu na koju se sa njih ukazuje, verovatnoća pristupa nekoj stranici s, može se izraziti na sledeći način:

$$P(s) = \sum_{r \in N(s)} \frac{P(r)}{n_r} + \sum_{r \in K} v_s P(r) \qquad s \in S$$

gde je $N(s)$ skup strana koje ukazuju na stranu s, a n_r broj strana na koje se ukazuje sa strane r. Neka je p vektor verovatnoća posete svih stranica, a A matrica takva da važi $A_{ij} = 1/n_j$ ukoliko postoji veza sa strane j ka strani i, a 0 u suprotnom. Takođe, neka je k vektor takav da važi $k_i = 1$ ako čvor i nema naslednika, a $k_i = 0$ u suprotnom. Onda se prethodna jednakost može izraziti matrično:

$$p = (A + vk^T)p$$

U slučaju usmerenog grafa prikazanog na slici 4.1, vektor k je $(0, 0, 0, 1)^T$. Neka je vektor v jednak $(1/4, 1/4, 1/4, 1/4)^T$. Onda je matrica $A + vk^T$ jednaka

$$\begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix}$$

Očito, p predstavlja sopstveni vektor matrice $A + vk^T$. Stoga se prethodni problem svodi na problem nalaženja sopstvenih vektora te matrice. Imajući u vidu da je matrica $A + vk^T$ stohastička matrica*, odnosno da je suma svake njene kolone jednaka 1, lako je dokazati da je njena najveća sopstvena vrednost jednaka* 1*. Stoga, p predstavlja sopstveni vektor matrice $A + vk^T$, koji odgovara najvećoj sopstvenoj vrednosti. Međutim, ta sopstvena vrednost može biti višestruka, pa p nije jedinstveno određeno. Ipak, treba imati u vidu da korisnici ne biraju uvek samo strane dostupne sa tekuće strane koju su posetili, već nekada biraju narednu stranu nezavisno od tekuće. Neka je $0 \leq \alpha < 1$ verovatnoća izbora jedne od veza dostupnih na tekućoj strani. Tada važi*

$$p = \alpha(A + vk^T)p + (1 - \alpha)v = (\alpha(A + vk^T) + (1 - \alpha)ve^T)p$$

gde je e vektor čiji su svi elementi jednaki 1, pri čemu poslednja jednakost važi zahvaljujući tome što važi $e^T p = \sum_{i=1}^{n} p_i = 1$. U ovom slučaju, p je sopstveni vektor matrice

$$G = \alpha(A + vk^T) + (1 - \alpha)ve^T$$

Kako su za $\alpha < 1$ svi elementi matrice G strogo pozitivni, na osnovu Peron-Frobenijusove teoreme sledi da je po modulu najveća sopstvena vrednost ove matrice pozitivna, da nema druge sopstvene vrednosti sa istom apsolutnom vrednošću i da njoj odgovara sopstveni vektor čije su sve koordinate strogo pozitivne. Kako je matrica stohastička, ta sopstvena vrednost mora biti 1*. Samim tim, vektor p je sopstveni vektor koji odgovara najvećoj sopstvenoj vrednosti matrice G, onoj koja je jednaka 1.*

Još jedan primer primene agoritama linearne algebre je modelovanje semantike reči na osnovu njihovih pojavljivanja u dokumentima, a u cilju pronalaženja relevantnih informacija u velikim korpusima dokumenata.

Primer 49 *Oblast pretraživanja informacija (eng. information retrieval) se bavi pronalaženjem relevantnih informacija u velikim korpusima dokumenata. Ulaz u pretragu je* upit*, što je obično niz reči koje su relevantne za datu pretragu (u najjednostavnijem slučaju, tako što se nalaze u dokumentu od interesa), a izlaz iz pretrage je niz dokumenata ili, eventualno, njihovih delova koji se smatraju relevantnim za dati upit. Mehanizmi pretrage su raznorodni, a izazovi na koje treba odgovoriti kako bi se ovaj problem adekvatno rešio su mnogobrojni i teški. Neki od izazova su sledeći:*

- *Sinonimija – u opštem slučaju označava postojanje više različitih načina da se opiše isti objekat. Želja za istim informacijama, u slučaju različitih korisnika, može voditi različitim upitima. Zapravo, utvrđeno je da dva korisnika biraju istu glavnu ključnu reč za isti dobro poznati objekat u manje od 20% slučajeva. Korisnici u upitima često koriste reči koje se same ne nalaze u dokumentima od značaja za dati upit, ali su po smislu bliske rečima u tim dokumentima. Ovaj problem se u nekoj meri ublažava proširivanjem upita sinonimima iz rečnika, ali takav pristup može voditi uvođenju reči koje imaju i drugačija značenja.*

- *Polisemija – označava postojanje većeg broja značenja istog upita. Homonimija reči je uobičajen primer. Razrešavanje polisemije, odnosno ustanovljavanje jednog od više mogućih značenja reči u upitu, vrlo je teško. Jedan od pristupa razrešavanju polisemije je analiza prisustva ostalih reči u upitu i sužavanje mogućnosti značenja svake reči na osnovu toga. Ipak, ovaj pristup nije dovoljan jer značajno zavisi od korisnikovog upita, a korisnici ne razmišljaju o potrebi za zadavanjem reči koje će ublažiti problem polisemije.*

Jedan pristup ublažavanju ovih problema je u traženju vektorskih reprezentacija reči, koje bi bile utoliko bliže što se reči češće nalaze zajedno u dokumentima. Iza ovoga principa stoji ideja da reči koje se često javljaju zajedno nose srodna značenja i obrnuto. Takav vektorski pristor u kojem bi reči bile predstavljene, predstavlja semantički prostor reči. Za ovako postavljen problem, od očiglednog su značaja frekvencije pojavljivanja reči u dokumentima koje se čuvaju u vidu matrice u kojoj kolone predstavljaju reči, a vrste dokumente (eng. term-document frequency matrix). Dodatno, bilo bi korisno imati i reprezetnacije dokumenata u istom prostoru. Time bi se postigle dve stvari. Prvo, lako poređenje sličnosti reči iz upita i dokumenata, ali i dodavanje novih reči i dokumenata. Novom dokumentu se može pridružiti reprezentacija koja je centroida*, odnosno srednja vrednost, reprezentacija reči koje se u njemu pojavljuju. Novoj reči se može pridružiti reprezentacija koja je centroida reprezentacija dokumenata u kojima se pojavljuje. Postavlja se pitanje kako polazeći od takve matrice konstruisati željeni prostor i reprezentacije reči u njemu, ali nije iznenađujuće da odgovor, koji će biti dat kasnije, nudi linearna algebra.*

U nastavku se diskutuju dekompozicije matrica, izračunavanje sopstvenih vektora, rešavanje sistema linearnih jednačina specifičnih formi i inkrementalni pristup rešavanju problema linearne algebre, ali je prethodno potrebno formalno uvesti pojam norme i razmotriti nekoliko specifičnih normi.

Neka je X konačno dimenzionalni vektorski prostor. *Norma* je funkcija $\|\cdot\| : X \to \mathbb{R}$ takva da za svako $\alpha \in \mathbb{R}$ i $x, y \in X$ važi:

- $\|\alpha x\| = |\alpha|\|x\|$
- $\|x + y\| \leq \|x\| + \|y\|$
- Ako važi $\|x\| = 0$, onda važi $x = 0$.

Takozvane p norme nad vektorima iz $\mathbb{R}^n$ se definišu na sledeći način:

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^{n} x_i^p}$$

Za svake dve p norme $\|\cdot\|_a$ i $\|\cdot\|_b$ postoje konstante $0 < c_1 \leq c_2$, takve da za svako $x \in X$ važi

$$c_1\|x\|_b \leq \|x\|_a \leq c_2\|x\|_b$$

Ovo svojstvo se naziva *ekvivalentnošću* p normi. Jedna od implikacija je da konvergencija u jednoj p normi znači konvergenciju u bilo kojoj drugoj p normi.

Norme se mogu lako definisati i nad matricama. U slučaju p normi, definicija se oslanja na p norme nad vektorima:

$$\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$

Još jedna često korišćena matrična norma, ekvivalentna matričnim p normama je Frobenijusova norma:

$$\|A\|_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2$$

4.2 Dekompozicije matrica

Dekompozicije matrica igraju veliku ulogu u numeričkom rešavanju osnovnih problema linearne algebre. Često pružaju veću brzinu, ali i veću numeričku stabilnost metoda u odnosu na druge pristupe. U nastavku se razmatraju četiri vrste dekompozicija – LU, Čoleski, QR i SVD.

4.2.1 LU dekompozicija

LU dekompozicija kvadratne matrice A se sastoji u njenom predstavljanju u vidu proizvoda dve matrice

$$A = LU$$

gde je matrica L donjetrougaona sa jedinicama na dijagonali, a matrica U gornjetrougaona. Ukoliko je takva dekompozicija poznata, može se upotrebiti za rešavanje sistema jednačina $Ax = b$. Naime, važi

$$Ax = (LU)x = L(Ux) = b$$

Odavde se vidi da je rešavanjem sistema

$$Ly = b$$

a potom sistema

$$Ux = y$$

moguće dobiti rešenje polaznog sistema. S druge strane, ne mora biti očigledno zašto je postojanje ovakve dekompozicije poželjno. Prvo, Krautov algoritam, kojim se izračunava LU dekompozicija matrice, zahteva $\frac{1}{3}n^3$ množenja, dok na primer Gaus-Žordanova metoda zahteva n^3. Otud postoji dobitak u brzini za konstantan faktor. Dodatno, pri rešavanju većeg broja sistema oblika $Ax = b_1, \ldots, Ax = b_n$, što je nekad potrebno u praksi, prvi put je potrebno uraditi dekompoziciju matrice A u vremenu $O(n^3)$, a za rešavanje svakog sledećeg je

potrebno vreme reda $O(n^2)$, pošto se rešavaju dva gornjetrougaona sistema. Takvo ubrzanje nije moguće ako se koristi Gaus-Žordanova metoda.

Oslanjajući se na LU dekompoziciju, moguće je izračunati i inverz matrice. Neka su $I_1, \ldots, I_n$, kolone jedinične matrice dimenzija $n \times n$. Onda se i-ta kolona inverza A^{-1} dobija tako što se reši sistem

$$Ly = I_i$$

a potom sistem

$$Ux = y$$

Za ovaj način izračunavanja je potrebno jednako operacija kao za računanje inverza Gaus-Žordanovom metodom.

Ipak, retko je potrebno računati inverz matrice zarad njega samog. Inverz je obično potrebno računati da bi se njime množio neki vektor ili druga matrica. Ako je potrebno izračunati $A^{-1}B$ i ako su $B_1, \ldots, B_n$ kolone matrice B, onda se i-ta kolona matrice $A^{-1}B$ dobija tako što se reši sistem

$$Ly = B_i$$

a potom sistem

$$Ux = y$$

Broj operacija je u ovom slučaju jednak kao u slučaju računanja samog inverza, pošto se umesto kolona jedinične matrice koriste kolone matrice B, što znači da se štedi na jednom matričnom množenju, ali se time takođe dobija i na preciznosti. Kao što ovo razmatranje sugeriše, retko je uopšte potrebno čuvati inverz matrice, pošto je moguće osloniti se na čuvanje njene dekompozicije. Čak i u slučaju da je matrica A^{-1} dostupna besplatno, preferira se korišćenje dekompozicije zbog preciznosti. Štaviše, u praktičnim primenama je čuvanje inverza problematično i zbog memorijske zahtevnosti. Naime, u praksi, sistemi jednačina često imaju vrlo specifičnu strukturu – obično su retki, što znači da je najveći broj elemenata matrice A jednak nuli. Retka struktura sugeriše da ne interaguju svi elementi sistema sa svim drugim elementima sistema. Na primer, ako je potrebno modelovati ljudsko ponašanje u društvenim mrežama, ili povezanost stranica na internetu, ovo je u potpunosti očekivano. U slučaju retke strukture, često je moguće baratati sistemima koju uključuju i milione promenljivih. U tim slučajevima se sistem $Ax = b$ može predstaviti u memoriji i često se može efikasno rešiti, ali u opštem slučaju inverz A^{-1}, nema retku strukturu, što rezultuje hiljadama milijardi brojeva koje je potrebno čuvati u memoriji, što čini čuvanje inverza u memoriji nemogućim. Postoje posebni algoritmi za dekompoziciju retkih matrica, koji bolje čuvaju retku strukturu od standardnih metoda dekompozicije.

Vezano za LU dekompoziciju, postavlja se pitanje, kada ju je moguće izvršiti i da li je jedinstvena. Postoje uslovi koji garantuju mogućnost izvođenja LU dekompozicije i njenu jedinstvenost, međutim oni su oštriji nego što je u praksi

potrebno. Naime, jedinstvenost nije od praktičnog značaja, dok, iako LU dekompoziciju nije uvek moguće dobiti od polazne matrice, uvek je moguće razmeniti njene redove, tako da je moguće sprovesti dekompoziciju, za šta postoji i konkretna procedura. Kako razmena redova matrice odgovara razmeni redosleda jednačina u sistemu, to se ne odražava na smisao problema i stoga u praksi predstavlja zadovoljavajuće rešenje.

Ako je poznata LU dokompozicija matrice, lako je izračunati determinantu matrice – ona se dobija kao proizvod dijagonalnih elemenata matrice U (dijagonalni elementi matrice L su jedinice). Pritom, treba imati u vidu da ukoliko je vršeno permutovanje vrsta polazne matrice, to utiče na znak determinante, ali broj permutacija, pa otud i znak determinante, je moguće pratiti prilikom rada algoritma koji vrši dekompoziciju.

Primer 50 *Prilikom rešavanja problema najmanjih kvadrata, potrebno je rešiti sistem linearnih jednačina*

$$A^T Ax = A^T b$$

ili njegove regularizovane varijante. Rešavanje ovog sistema LU dekompozicijom predstavlja bolji pristup od izračunavanja Mur-Penrouzovog pseudoinverza koji uključuje inverziju matrice $A^T A$.

4.2.2 Čoleski dekompozicija

Ukoliko je kvadratna matrica A simetrična i pozitivno definitna, postoji mogućnost njene efikasnije trougaone dekompozicije. Matrica je pozitivno definitna ukoliko za sve vektore x odgovarajuće dimenzije, različite od nule, važi

$$x^T Ax > 0$$

Iako je uslov simetričnosti i pozitivne definitnosti restriktivan, ovakve matrice se javljaju u različitim primenama. Na primer, u verovatnosnom modelovanju, pretpostavka zajedničke normalne raspodele promenljivih je prilično česta. Matrica kovarijacije normalne raspodele mora biti simetrična i pozitivno definitna,[1] a u samoj formulaciji normalne raspodele figuriše njen inverz, što znači da poznavanje dekompozicije te matrice može biti od koristi.

Čoleski dekompozicija koju je moguće izvesti u pomenutom slučaju predstavlja dekompoziciju matrice u obliku

$$A = LL^T$$

gde je L donjetrougaona matrica sa strogo pozitivnim dijagonalnim elementima. Čoleski dekompozicija matrice dimenzija $n \times n$ se izračunava narednim

[1] Može biti i pozitivno semidefinitina, ali ako nije pozitivno definitna, to vodi određenim tehničkim problemima.

formulama

$$l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik} \right)^{1/2} \qquad i = 1, \ldots, n$$

$$l_{ji} = \frac{1}{l_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} \right) \qquad i = 1, \ldots, n-1$$

$$j = i+1, \ldots, n$$

Izračunavanje Čoleski dekompozicije zahteva $\frac{1}{6}n^3$ množenja, što je duplo bolje od LU dekompozicije. Pored veće efikasnosti izračunavanja, kvalitet Čoleski dekompozicije je i veća numerička stabilnost njenog izračunavanja. Treba primetiti da pozitivna definitnost implicira invertibilnost, što znači da ovu vrstu dekompozicije nije moguće primeniti na matrice koje nemaju pun rang, ali to takođe znači i da je stabilnost izračunavanja lošija za loše uslovljene matrice.

Ako je poznata Čoleski dekompozicija, rešavanje sistema jednačina se vrši tako što se prvo reši sistem

$$Ly = b$$

a potom sistem

$$L^T x = y$$

Determinantu je moguće izračuanti na sledeći način

$$\det(A) = \det(L) \cdot \det(L^T) = \det{}^2(L)$$

pri čemu je determinanta matrice L proizvod njenih dijagonalnih elemenata.

Čoleski dekompoziciju je moguće izvršiti kad god je matrica simetrična i pozitivno definitna. Zanimljivo je da važi i obrnuto – da postojanje Čoleski dekompozicije matrice implicira simetričnost i pozitivnu definitnost te matrice. Neka se matrica A može predstaviti u obliku LL^T, gde je L donjetrougaona matrica sa strogo pozitivnim dijagonalnim elementima. Simetrija je očigledna, a takođe važi i

$$x^T A x = x^T L L^T x = (L^T x)^T (L^T x) = \|L^T x\|^2 \geq 0$$

što pokazuje da je matrica A pozitivno semidefinitna. Kako je matrica L^T gornjetrougaona sa strogo pozitivnim dijagonalnim elementima, onda je ona i invertibilna, pa Lx može imati vrednost 0, samo u slučaju da važi $x = 0$. Odnosno, važi $x^T A x = 0$, ako i samo ako važi $x = 0$. Zajedno sa pozitivnom semidefinitnošću, ovo implicira pozitivnu definitnost. Zapravo, sprovođenje Čoleski dekompozicije predstavlja efikasan način da se ustanovi pozitivna definitnost matrice. Ukoliko ju je moguće izvesti, to znači pozitivnu definitnost. U slučaju da matrica nije pozitivno defintna, algoritam će se zaustaviti usled nemogućnosti izračunavanja korena negativnih brojeva ili greške deljenja nulom.

Primer 51 *Regularizovani problem najmanjih kvadrata se rešava rešavanjem sistema*

$$(A^T A + \lambda I)x = A^T b$$

za neko $\lambda > 0$. *Matrica* $A^T A + \lambda I$ *je očigledno kvadratna i simetrična i važi*

$$x^T(A^T A + \lambda I)x = x^T A^T Ax + \lambda x^T x = \|Ax\|^2 + \lambda\|x\|^2 \geq 0$$

Kako je $\|x\|^2 = 0$ *samo ako je* $x = 0$, *matrica* $A^T A + \lambda I$ *je pozitivno definitna, pa je umesto LU dekompozicije, regularizovani problem najmanjih kvadrata bolje rešavati Čoleski dekompozicijom. Ukoliko nije primenjena regularizacija, moguć je problem loše uslovljenosti matrice* $A^T A$ *ili, još gore, njene neinvertibilnosti u kom slučaju nije moguće primeniti Čoleski dekompoziciju.*

4.2.3 QR dekompozicija

Za kvadratnu matricu Q se kaže da je ortogonalna, ukoliko za nju važi $Q^T Q = QQ^T = I$. Pored toga što nije potrebno nalaziti inverz ovakvih matrica, one imaju i druga poželjna svojstva. Na primer, čuvaju euklidsku normu vektora pri množenju:

$$\|Qx\|_2^2 = (Qx)^T Qx = x^T Q^T Qx = x^T x = \|x\|_2^2$$

Dalje, uslovljenost matrice se definiše kao maksimum uslovljenosti sistema jednačina $Ax = b$ za koji je ranije pokazano da je jednak

$$\max_{x,\Delta b} \frac{\|Q^{-1}\Delta b\|}{\|\Delta b\|} \cdot \frac{\|Qx\|}{\|x\|} = \max_{x,\Delta b} \frac{\|Q^T \Delta b\|}{\|\Delta b\|} \cdot \frac{\|Qx\|}{\|x\|}$$

što je zahvaljujući upravo pokazanom očuvanju euklidske norme jednako 1. Odnosno, ortogonalne matrice imaju minimalnu moguću uslovljenost $Cond(Q) = 1$, što znači da množenje ovim matricama ne uvećava do tada akumuliranu grešku, osim ukoliko dovede do novih grešaka zaokruživanja. Zato su ovakve matrice posebno važne u naučnom izračunavanju.

Proizvoljna matrica A, dimenzija $m \times n$, gde je $m \geq n$, može se predstaviti u obliku

$$A = QR$$

gde je Q *ortogonalna matrica*, dimenzija $m \times m$, dok je R matrica dimenzija $m \times n$ oblika

$$R = \begin{bmatrix} R' \\ 0 \end{bmatrix}$$

gde je matrica R' gornjetrougaona matrica dimenzija $n \times n$. Odnosno, matrica R ima formu

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

Kako se prilikom množenja matrica Q i R, poslednjih $m - n$ kolona matrice Q množe nulama, QR dekompoziciju je moguće predstaviti u redukovanom obliku. Ako se podmatrica matrice Q, koja se sastoji od njenih prvih n kolona, označi sa Q', onda se QR dekompozicija može predstaviti i kao

$$A = Q'R'$$

Ovo je ilustrovano na slici 4.2.

U slučaju da je matrica A kvadratna, kao i u slučaju prethodnih dekompozicija, poznavanje QR dekompozicije se može upotrebiti za rešavanje sistema jednačina $Ax = b$ tako što će se rešavati gornjetrougaoni sistem

$$Rx = Q^T b$$

pri čemu su ovi sistemi ekvivalentni zahvaljujući činjenici da je matrica Q^T inverz matrice Q. Kako izračunavanje QR dekompozicije zahteva više operacija nego izračunavanje LU dekompozicije, s tačke gledišta računske efikasnosti ovo nije preferirani način rešavanja sistema. S druge strane, kako je matrica R gornjetrougaona, a matrica Q ortogonalna, upotreba QR dekompozicije može voditi boljoj numeričkoj stabilnosti.

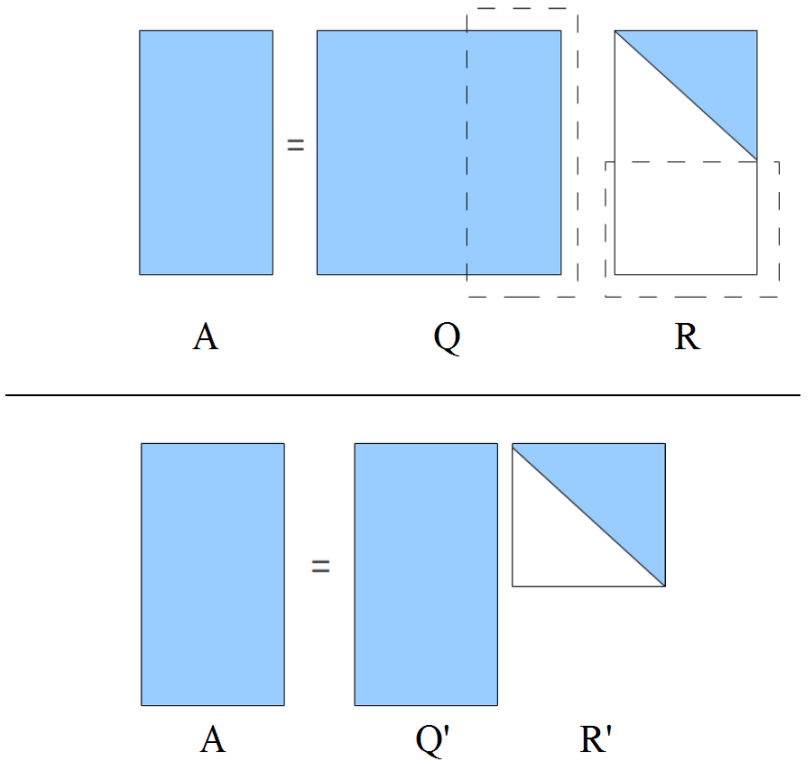
Upotreba QR dekompozicije je posebno zanimljiva u kontekstu kada matrica A nije kvadratna, pošto prethodne metode u tom slučaju nisu primenljive. Konkretno, kada postoji više jednačina nego promenljivih. Ovo je očigledno kontekst metode najmanjih kvadrata i upravo u rešavanju ovog problema QR dekompozicija nalazi jednu od svojih važnih primena.

Primer 52 *Neka je poznata QR dekompozicija matrice A. Tada važi*

$$\|b - Ax\|_2 = \|b - QRx\|_2 = \|Q^T(b - QRx)\|_2 = \|Q^T b - Rx\|_2$$

pri čemu je u drugom prelazu upotrebljeno svojstvo da množenje ortogonalnom matricom čuva normu vektora. Ako se vektor $Q^T b$ predstavi u obliku

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$



Slika 4.2: Puna (gore) i redukovana (dole) QR dekompozicija

gde važi $c_1 \in \mathbb{R}^n$ *i* $c_2 \in \mathbb{R}^{m-n}$*, onda se prethodne jednakosti svode na*

$$\|b - Ax\|_2^2 = \|c_1 - R'x\|_2^2 + \|c_2\|_2^2$$

Kako se u metodu najmanjih kvadrata ovaj izraz minimizuje po x *i kako je matrica* R' *gornjetrougaona i kvadratna, rešavanje problema najmanjih kvadrata se vrši rešavanjem gornjetrougaonog sistema*

$$R'x = c_1$$

Ovo je preferirani način rešavanja problema najmanjih kvadrata. Isto rešenje, samo u terminima matrica Q *i* R*, dobija se polazeći od Mur-Penrouzovog inverza koji se koristi u rešavanju problema najmanjih kvadrata:*

$$(A^TA)^{-1}A^T = ((QR)^TQR)^{-1}(QR)^T = (R^TQ^TQR)^{-1}(QR)^T$$

$$= (R^TR)^{-1}R^TQ^T = R^{-1}Q^T$$

U slučaju kvadratne matrice A*, isti izraz predstavlja inverz te matrice.*

Poznavanje QR dekompozicije matrice očigledno predstavlja prednost, ali se postavlja pitanje na koji ju je način moguće izračunati. Jedan način se zasniva na zanimljivom zapažanju vezanom za odnos matrica A i Q. Neka važi $A = QR$. Prostor kolona matrice A, odnosno skup svih linearnih kombinacija kolona matrice A, je

$$C(A) = \{Ax | x \in \mathbb{R}^n\}$$

Onda važi

$$\begin{aligned} u \in C(A) &\iff u = Ax \text{ za neko } x \\ &\iff u = QRx \\ &\iff u = Qy \text{ za } y = Rx \\ &\iff u \in C(Q) \end{aligned}$$

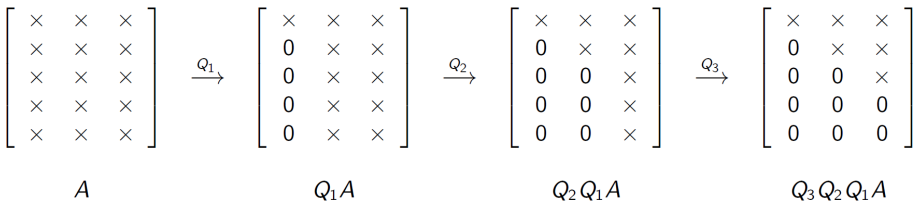
odnosno $C(A) = C(Q)$, pri čemu je u trećem redu upotrebljena pretpostavka da je matrica R, a samim tim i matrica A invertibilna, što ne mora biti slučaj. Ako nije tako, onda ne važi ekvivalencija, već implikacija i, stoga, $C(A) \subseteq C(Q)$. Kako je matrica Q ortogonalna, što znači da su joj kolone normirane i međusobno ortogonalne, to znači da kolone matrice Q predstavlja ortonormiranu bazu prostora kolona matrice A, ukoliko je matrica A invertibilna, a u suprotnom to važi za neke od kolona matrice Q. S jedne strane, ovo zapažanje ukazuje na to da se bilo koji algoritam za izračunavanje QR dekompozicije može iskoristiti za ortonormiranje sistema vektora – QR dekompozicijom matrice koja za kolone ima vektore tog sistema vektora. S druge strane, isto zapažanje daje prvu ideju o tome kako se može izvršiti QR dekompozicija matrice – Gram-Šmitovim procesom ortonormiranja njenih kolona. Ovaj postupak neće biti opisivan detaljnije, tim pre što je ovaj algoritam numerički vrlo nestabilan i zahteva oko $2mn^2$ operacija za izračunavanje faktorizacije. Postoji modifikovani Gram-Šmitov postupak koji je numerički stabilniji, ali zahteva isti broj operacija.

Algoritam koji se najčešće koristi za izračunavanje QR dekompozicije je Haushelderov algoritam, koji je numerički stabilan, zahvaljujući tome što počiva na upotrebi ortogonalnih matrica, a zahteva oko $2mn^2 - \frac{2}{3}n^3$ operacija. Haushelderov algoritam konstruiše niz matrica $Q_1, \ldots, Q_n \in \mathbb{R}^{m \times m}$ i njima množi matricu A kako bi se od nje dobila gornjetrougaona matrica, odnosno:

$$Q_n \cdots Q_2 Q_1 A = R$$

Pritom, efikasnost ovakvog algoritma proizilazi iz mogućnosti takvog izbora matrica Q_i, da se množenjem i-tom od njih anuliraju svi elementi i-te kolone, ispod glavne dijagonale, tako da proces izlgeda kao na slici 4.3. Ako se izračunavanje matrice R sprovodi na ovaj način, onda je QR dekompozicija matrice A jednaka

$$A = Q_1^T Q_2^T \cdots Q_n^T R$$



Slika 4.3: Proces dobijanja matrice R Haushelderovim algoritmom.

Naravno, postavlja se pitanje, kako treba da izgledaju matrice Q_i, kako bi ovakav proces bio moguć. Svaka od matrica ima sledeću formu

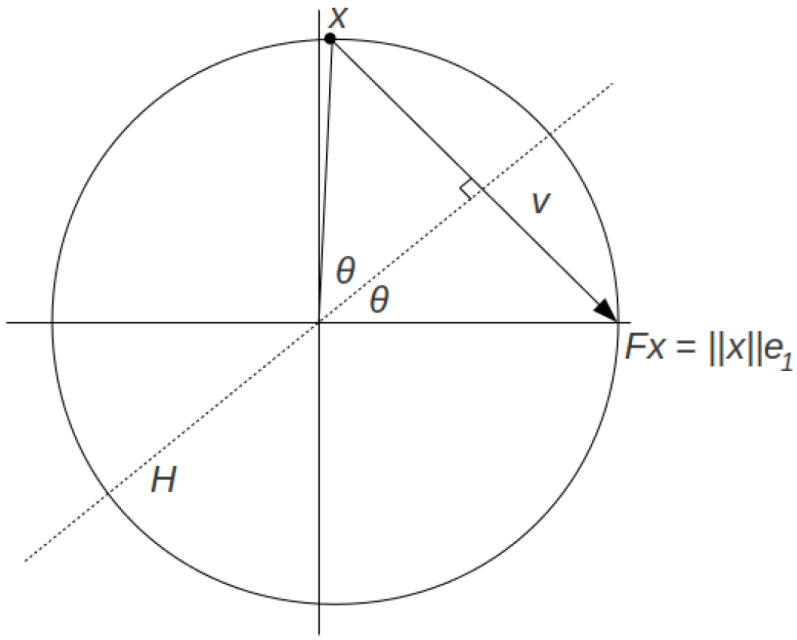
$$Q_i = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}$$

pri čemu je matrica I kvadratna jedinična matrica dimenzije $i-1$, a F ortogonalna kvadratna *Haushelderova matrica refleksije* dimenzije $m-i+1$. Blok I čini da svaka od transformacija sačuva ono što su prethodne postigle, dok matrica F postiže željeni rezultat anuliranja elemenata. Kako je matrica F ortogonalna, to je i matrica Q_i. Osnovna ideja za konstrukciju matrice F je da treba da realizuje refleksiju vektora tako da mu svi elementi osim prvog budu nula. Ovo je moguće uraditi zato što se vektor uvek može reflektovati u odnosu na neku hiperravan tako da završi na koordinatnoj osi prvog baznog vektora i zato što se refleksija može predstaviti ortogonalnom matricom. Kako množenje ortogonalnom matricom čuva euklidsku normu vektora, to znači da se vektor x dejstvom ove matrice preslikava u vektor $Fx = \|x\|_2 e_1$, gde je e_1 prvi bazni vektor, odnosno mora važiti:

$$Fx = \begin{bmatrix} \|x\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Kako je namena matrice Q_i da se anuliraju svi elementi ispod glavne dijagonale, i kako matrica F deluje upravo na podmatricu čiji je gornji levi element i-ti element dijagonale, jasno je da ova matrica ima željeno dejstvo. Transformacija koju matrica F treba da realizuje, prikazana je na slici 4.4, gde je H podprostor u odnosu na koji se vrši refleksija, a v je vektor koji treba dodati na x, tako da se dobije $\|x\|_2 e_1$, odnosno $v = \|x\|_2 e_1 - x$. Pritom, kako se radi o refleksiji, prostor u odnosu na koji se vrši refleksija deli duž koja spaja x i Fx na pola. Prvi korak u određivanju transformacije F je određivanje projekcije vektora x na podprostor H. Lako se proverava da je ta projekcija izražena matricom

$$P_H = I - \frac{vv^T}{v^T v}$$



Slika 4.4: Refleksija koju realizuje Haushelderova matrica.

Kako transformacija F treba da „pomeri“ x dvaput dalje nego projekcija, željena matrica je

$$F = I - 2\frac{vv^T}{v^T v}$$

Lako se pokazuje da je F ortogonalna matrica. Jedna stvar je pojednostavljena u prethodnoj diskusiji. Naime, pored vektora $\|x\|_2 e_1$, kandidat za rezultat transformacije Fx je podjednako i vektor $-\|x\|_2 e_1$, koji ima istu normu i nule na istim mestima. Pritom, nije sasvim svejedno koji se bira, pošto to može uticati na grešku izračunavanja. Naime, ukoliko su vektori x i Fx blizu, moguće je da dođe do greške poništavanja, pa je poželjno da se izabere onaj vektor koji je dalje od x. Stoga se za vektor v uzima

$$v = -sgn(x_1)\|x\|_2 e_1 - x$$

odnosno, pazi se da ako vektor x ima pozitivnu komponentu duž pravca e_1, vektor Fx ima negativnu i obrnuto. Kako matrica F ne zavisi od znaka vektora v, uzima se lepša forma

$$v = sgn(x_1)\|x\|_2 e_1 + x$$

Haushelderov algoritam je dat na slici 4.5. Ovim algoritmom se očigledno dobija matrica R, ali ne i matrica Q. Matricu Q često nije ni neophodno

```
1 for i=1 to n do
2 |  x = a_{i:m,i}
3 |  v_i = sgn(x_1)‖x‖_2 e_1 + x
4 |  v_i = v_i/‖v_i‖_2
5 |  a_{i:m,i:n} = a_{i:m,i:n} − 2v_k(v^T a_{i:m,i:n})
6 end
```

Slika 4.5: Haushelderov algoritam. Oznaka $a_{i:j,k:l}$ označava podmatricu matrice A u rasponu datih indeksa.

```
1 for i=n to 1 do
2 |  x_{i:m} = x_{i:m} − 2v_i(v_i^T x_{i:m})
3 end
```

Slika 4.6: Algoritam za množenje vektora x matricom Q kada je poznat samo niz vektora $v_1, \ldots, v_n$.

eksplicitno izračunati. Na primer, ukoliko je potrebno izračunati proizvod Qx, dovoljno je koristiti algoritam na slici 4.6 koji se oslanja samo na vektore v_i. Slično, ako je kao u slučaju rešavanja problema najmanjih kvadrata potrebno izračunati $Q^T b$, dovoljno je zameniti granice indeksa u `for` petlji, tako da i ide od 1 do n. Ipak, ako je potrebno konstruisati matricu Q, to je moguće uraditi tako što će se datim algoritmom izračunati

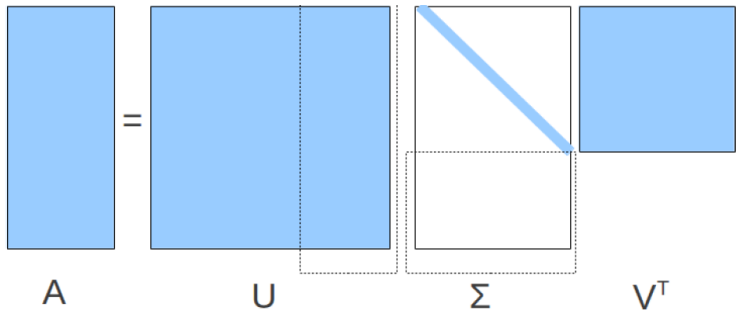
$$\begin{bmatrix} Qe_1 & Qe_2 & \cdots & Qe_n \end{bmatrix}$$

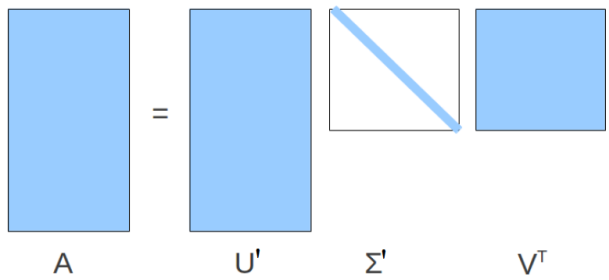
4.2.4 Singularna dekompozicija

U slučaju matrica koje su bliske singularnim, prethodne metode ipak imaju svoja ograničenja u kontekstu rešavanja sistema jednačina – problem je sam po sebi loše uslovljen. U slučaju singularnih matrica, pomoću njih nije moguće doći ni do kakvih rešenja. Singularna dekompozicija (eng. singular value decomposition), ili skraćeno SVD, pruža mogućnost za smisleno baratanje ovakvim sistemima jednačina, bilo u cilju njihovog rešavanja, bilo u cilju razumevanja problema koji ih čine problematičnim za rešavanje.

Neka je A matrica dimenzija $m \times n$. Obično se radi jednostavnosti diskusije pretpostavlja da važi $m \geq n$, što je pretpostavka koja će važiti i u nastavku, ali ta pretpostavka nije od suštinskog značaja i sve što bude rečeno, važi i u slučaju $m < n$, uz adekvatne izmene koje razlika u dimenzijama matrica nalaže. Za bilo koju takvu matricu A, postoje ortogonalna matrica U dimenzija $m \times m$, dijagonalna matrica Σ dimenzija $m \times n$ i ortogonalna matrica V dimenzija $n \times n$, takve da važi

$$A = U\Sigma V^T$$





Slika 4.7: Varijante singularne dekompozicije u slučaju kada je $m \geq n$.

Slično slučaju QR dekompozicije, matrica Σ ima $m - n$ vrsta koje se sastoje isključivo od nula i koje se mogu zanemariti. Stoga, matricu A je moguće razložiti i na matricu ortonormiranih kolona U' dimenzija $m \times n$, dijagonalnu matricu Σ' dimenzija $n \times n$ i ortogonalnu matricu V dimenzija $n \times n$, tako da važi

$$A = U'\Sigma'V^T$$

Obe varijante su ilustrovane na slici 4.7.

Dijagonalni elementi matrice Σ, koji će biti označavani σ_i za $i = 1, 2, \ldots, n$, nazivaju se *singularnim vrednostima* matrice A. Može se pretpostaviti da su singularne vrednosti σ_i poređane u nerastućem poretku. Važe sledeća svojstva singularne dekompozicije:

- Singularne vrednosti matrice A su sve nenegativne i predstavljaju kvadratne korene sopstvenih vrednosti matrica AA^T i A^TA.
- Rang matrice A je jednak broju singularnih vrednosti koje nisu jednake nuli.
- Kolone matrice U su sopstveni vektori matrice AA^T i nazivaju se *levim singularnim vektorima* matrice A.

- Kolone matrice V predstavljaju sopstvene vektore matrice $A^T A$ i nazivaju se *desnim singularnim vektorima* matrice A.

- Kolone matrice V koje odgovaraju nultim singularnim vrednostima predstavljaju ortonormiranu bazu jezgra matrice A, odnosno prostora svih vektora x, takvih da važi $Ax = 0$.

- Kolone matrice V koje odgovaraju nenultim singularnim vrednostima predstavljaju ortonormiranu bazu prostora vrsta matrice A.

- Kolone matrice U koje odgovaraju nenultim singularnim vrednostima predstavljaju ortonormiranu bazu prostora kolona matrice A.

- Kolone matrice U koje odgovaraju nultim singularnim vrednostima predstavljaju ortonormiranu bazu levog jezgra matrice A, odnosno prostora svih vektora x, takvih da važi $x^T A = 0$.

- Važi $\|A\|_2 = \sigma_1$

- Važi $\|A\|_F = \sqrt{\sum_{i=1}^{n} \sigma_i^2}$

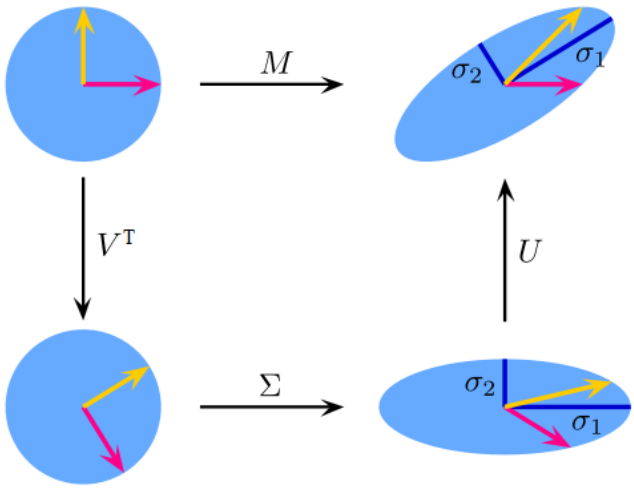
- Važi
$$Cond(A) = \frac{\sigma_1}{\sigma_n}$$
pri čemu se u definiciji uslovljenosti matrice, podrazumeva norma $\|\cdot\|_2$.

Algoritmi za izračunavanje singularne dekompozicije su komplikovani i o njima neće biti reči. Izračunavanje singularne dekompozicije je računski zahtevnije od prethodno razmatranih dekompozicija – potrebno je $\Theta(mn^2)$ operacija, sa nepovoljnim konstantnim faktorom. Međutim, izračunavanje singularne dekompozicije je numerički vrlo stabilno.

Iz navedenih svojstava, vidi se da se pomoću singularne dekompozicije može izvesti ortogonalizacija skupa vektora – tako što se uradi singularna dekompozicija matrice čije kolone oni čine. Ortonormiranu bazu čine kolone matrice U koje odgovaraju nenultim singularnim vrednostima. Ovaj pristup je mnogo bolji od korišćenja nestabilnog Gram-Šmitovog algoritma.

Primer 53 *Geometrijska ilustracija singularne dekompozicije prikazana je na slici 4.8. Matrica M transformiše krug u elipsu. Singularna dekompozicija daje razlaganje te operacije na delove. Prvo se dejstvom matrice V^T, vrši izometrijska transformacija, u ovom slučaju rotacija, nad krugom. Potom se krug transformiše u elipsu homotetijom Σ. Na kraju se elipsa transformiše izometrijskom transformacijom određenom matricom U.*

Vrlo zanimljiva primena singularne dekompozicije je u aproksimaciji matrica. Na osnovu singularne dekompozicije, ako su U_i kolone matrice U, a V_i



Slika 4.8: Ilustracija singularne dekompozicije na primeru dejstva matrice na krug.

kolone matrice V, onda se singularna dekompozicija može zapisati i na sledeći način

$$A = \sum_{i=1}^{n} \sigma_i U_i V_i^T$$

Matrice $U_i V_i^T$ su očito matrice ranga 1 – j-ta kolona te matrice je $v_{ij} U_i$ za $j = 1, \ldots, n$, odnosno sve kolone matrice $U_i V_i^T$ su samo skalirane varijante kolone U_i. Neka važi $0 \leq r \leq n$. Tada je

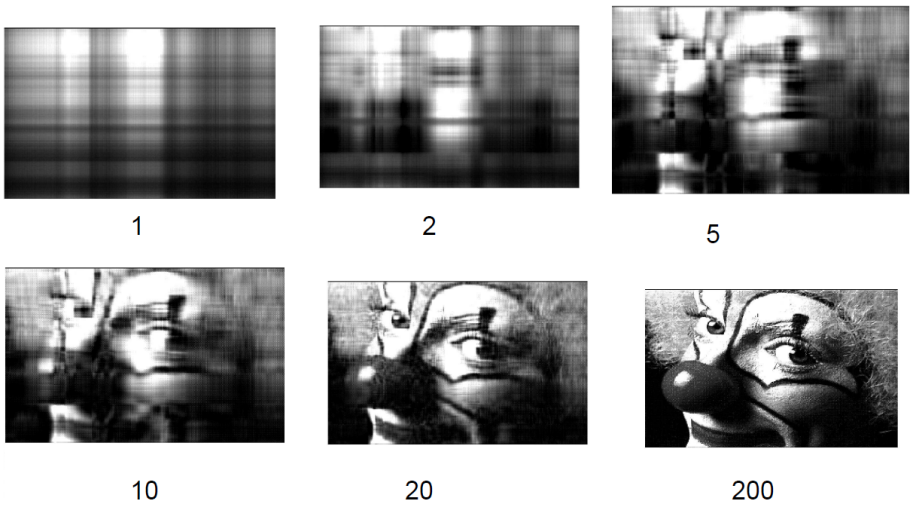
$$A_r = \sum_{i=1}^{r} \sigma_i U_i V_i^T$$

najbolja aproksimacija matrice A ranga r, odnosno važi da je

$$\|A - A_r\| = \min_{B \in \mathbb{R}^{m \times n} \wedge rang(B) = r} \|A - B\|$$

pri čemu je $\|\cdot\|$ Frobenijusova norma. Imajući u vidu definiciju Frobenijusove norme, jasno je da se radi o najboljoj srednjekvadratnoj aproksimaciji ranga r. Da bi se izvršila ova vrsta aproksimacije dovoljno je u matrici Σ anulirati najmanje singularne vrednosti, tako da ostane r vrednosti koje nisu nula.

U nekim slučajevima se aproksimacija matrice matricom nižeg ranga smatra poželjnijom od originalne matrice, pošto se prilikom odsecanja članova sume



Slika 4.9: Ilustracija aproksimacije slike pomoću singularne dekompozicije za različite rangove matrice kojom se vrši aproksimacija.

koji odgovaraju najnižim singularnim vrednostima može eliminisati šum ili neki manje bitni elementi signala.

Primer 54 *Jedna od mogućih primena singularne dekompozicije je u aproksimaciji slika. Slika 4.9 prikazuje aproksimacije različitog ranga slike dimenzije* 200×320. *Slika u svom izbornom obliku je opisana pomoću* $320 \cdot 200 = 64000$ *brojeva. Ukupan broj singularnih vrednosti je 200. Ukoliko se vrši aproksimacija ranga 20, potrebno je* $200 \cdot 20 + 20 + 320 \cdot 20 = 10420$ *brojeva. Prilikom ove računice, prirodno, pretpostavljeno je da se matrica* Σ *ne čuva kao puna matrica, već kao niz singularnih vrednosti.*

Kao i druge metode za dekomponovanje matrica, i singularna dekompozicija ima veoma važnu primenu u inverziji matrica i rešavanju sistema linearnih jednačina. Kako su matrice U i V ortogonalne, inverz kvadratne matrice A se lako računa ukoliko je poznata singularna dekompozicija matrice A:

$$A^{-1} = V\Sigma^{-1}U^T$$

pri čemu Σ^{-1} predstavlja matricu u kojoj je svaka od singularnih vrednosti σ_i, zamenjena vrednošću $1/\sigma_i$. Ono što je zanimljivo je da je na isti način moguće definisati i pseudoinverz proizvoljne matrice, pošto se singularna dekompozicija može izvršiti za bilo koju matricu. Takav pseudoinverz matrice A se označava A^+. Ispostavlja se da je A^+ upravo Mur-Penrouzov pseudoinverz. Ovo omogućava i definisanje uslovljenosti za proizvoljnu matricu:

$$Cond(A) = \|A\|\|A^+\|$$

Ono što je problem u ovakvom pristupu računanju inverza ili pseudoinverza je mogućnost da neka od singularnih vrednosti bude nula ili toliko mala da je računska greška relativno velika u odnosu na njenu vrednost. U takvom slučaju računanje recipročnih vrednosti nema smisla. Stoga, u matrici Σ^{-1}, figurišu vrednosti 0, tamo gde je σ_i jednako 0 ili je dovoljno malo. Otud matrica Σ^{-1} nije uvek inverz matrice Σ. Postavlja se pitanje, da li je ovakva definicija matrice Σ^{-1} intuitivno opravdana. Biće razmotren samo slučaj nultih singularnih vrednosti. Slučaj kada su singularne vrednosti vrlo male se odnosi na slučaj loše uslovljenosti, što je u praktične svrhe isto što i neinvertibilnost, što opravdava tretiranje takvih singularnih vrednosti na isti način kao da su nule. Neka je potrebno rešiti sistem $Ax = b$, neka su neke singularne vrednosti nulte i neka je

$$x' = V\Sigma^{-1}U^T b$$

Tada važi

$$Ax' = AV\Sigma^{-1}U^T b = U\Sigma V^T V\Sigma^{-1}U^T b = U\Sigma\Sigma^{-1}U^T b$$

Proizvod $\Sigma\Sigma^{-1}$ ima nule van dijagonale, a jedinice na dijagonali na pozicijama singularnih vrednosti koje nisu nula i nule na pozicijama onih koje jesu. Stoga će i vektor Ax' biti jednak vektoru b na pozicijama koje odgovaraju nenultim singularnim vrednostima, a nula na pozicijama nultih singularnih vrednosti. Nulte singularne vrednosti odgovaraju jednačinama koje se mogu izraziti pomoću drugih (otud i nepotpun rang matrice). Ako je sistem $Ax = b$ neprotivrečan, onda te vrste matrice i odgovarajući elementi vektora b mogu da se izostave bez menjanja skupa rešenja, pa su relevantne koordinate vektora b i Ax' jednake.

Može se pokazati da je vektor x', rešenje sistema najmanje $\|\cdot\|_2$ norme. Kako kolone matrice V koje odgovaraju nultim singularnim vrednostima predstavljaju bazu jezgra matrice A, sva rešenja ovog sistema se mogu dobiti dodajući njihove linearne kombinacije na vektor x'. Ukoliko je sistem protivrečan, može se pokazati da vektor x' minimizuje veličinu $\|Ax-b\|_2$. Odavde se vidi da singularna dekompozicija nudi još jedan način za rešavanje problema najmanjih kvadrata.

Singularna dekompozicija može poslužiti za prethodno pomenuti problem konstruisanja vektorskih reprezentacija reči.

Primer 55 *Već je ukazano na značaj nalaženja vektorskih reprezentacija reči na osnovu matrice frekvencija reči u dokumentima, ali je ostalo nerazjašnjeno kako se pomenute reprezentacije mogu konstruisati. Prvo, pojavljivanje reči u dokumentima se ugrubo može predstaviti matricom frekvencija A. Neka kolone predstavljaju reči, a vrste dokumente i neka elementi matrice predstavljaju broj pojavljivanja reči u dokumentu. Vrste matrice U koja se dobija singularnom dekompozicijom predstavlja nove reprezentacije dokumenata, a vrste matrice V nove reprezentacije reči. Sinonimi će u ovoj reprezentaciji imati vrlo slične reprezentacije zato što se oba javljaju u dokumentima u sličnim kontekstima,*

definisanim pojavljivanjem drugih reči. Odnosno, ako se jedna reč često javlja u dokumentima sa nekim rečima, verovatno se i njen sinonim često javlja u dokumentima sa tim istim rečima.

Iz matrice U i V mogu se izbaciti kolone koje odgovaraju najmanjim singularnim vrednostima, a iz matrice Σ se mogu izbaciti i kolone i vrste koje im odgovaraju. Neka je broj preostalih kolona k i neka su U_k, V_k i Σ_k odgovarajuće redukovane matrice dekompozicije, a A_k njima definisana aproksimacija matrice A. Smatra se da je reprezentacija A_k pouzdanija i da bolje oslikava stvarnu strukturu u podacima ukoliko je k dobro izabrano.

Postavlja se pitanje kako se ove reprezentacije mogu koristiti u praksi. Sličnost dva dokumenta ili dve reči se može lako meriti kosinusnim rastojanjem nad vrstama matrica koje sadrže njihove nove reprezentacije. Međutim, u praksi je potrebno za dati upit, koji ne odgovara nijednom redu ovih matrica, pronaći odgovarajući dokument. Prvi korak je predstavljanje upita u vidu dokumenta – frekvencijama reči u njemu. Na taj način, upit se predstavlja isto kao i dokumenti, koji su skupa bili predstavljeni matricom A. Nove reprezentacije U tih dokumenata se dobijaju sledećim operacijama

$$U = AV\Sigma^{-1}$$

Kako se fokusiramo na aproksimaciju ranga k, adekvatna formula je

$$U_k = A_k V_k \Sigma_k^{-1}$$

Kao što se primenom ove operacije dolazi do novih reprezentacija dokumenata, tako se može doći i do nove reprezentacije upita. Ako je upit predstavljen vektorom frekvencija reči x, nova reprezentacija je

$$x_k = x^T V_k \Sigma_k^{-1}$$

Tada je reprezentaciju x_k moguće porediti sa redovima matrice U_k i naći adekvatan dokument. Moguće je i sortirati dokumente prema sličnosti i vratiti zahtevani broj dokumenata. Pre upotrebe, matrice se obično standardizuju po kolonama.

Ispostavlja se da se dobri rezultati pretrage dobijaju već za vrednosti k između 50 i 150, što daje ekonomičniju reprezentaciju dokumenata, zahvaljujući eliminiaciji redundantnih informacija. Kako je za očekivati da se sinonimi relativno često javljaju u istim kontekstima, oni imaju i slične reprezentacije, pa nije bitno koji od sinonima se koristi. Stoga, ovaj pristup značajno pomaže u otklanjanju problema sinonimije. Problem homonimije i dalje nije otklonjiv na ovaj način, pošto će različita značenja imati pridruženu istu reprezentaciju.

Još jedan sličan primer primene singularne dekompozicije je u *sistemima za preporučivanje* (eng. *recommender systems*)

Primer 56 *U primeru 30, razmotren je problem preporučivanja filmova korisnicima na osnovu ocena koje su različitim filmovima dali različiti gledaoci, pri*

Film	Ana	Jovan	Marko	Dragana
Uspavana dolina	5	4	2	5
Panov lavirint	5	?	1	2
Odbegla mlada	1	5	?	5
Terminator 2	?	3	5	1
Hobit 3	1	1	5	?

Tabela 4.1: Ocene koje gledaoci daju filmovima.

čemu mnogi gledaoci nisu ocenili mnoge filmove. Primer takvih podataka dat je u tabeli 4.1. Ovaj problem se često rešava primenom singularne dekompozicije na takvu tabelu, odnosno matricu, ali uz dve izmene. Prvo, kako bi singularna dekompozicija bila primenjena, ne sme biti nedostajućih vrednosti. Te vrednosti se popunjavaju prosečnim ocenama svakog filma. Na taj način njihova prosečna poželjnost biva očuvana. Drugo, umesto matrice koja odgovara prikazanoj tabeli, češće se koristi transponovana matrica - u kojoj kolone odgovaraju filmovima, a vrste korisnicima. Ovo nema nikakve praktične posledice već samo predstavlja uobičajenu praksu. Nakon izračunavanja singularne dekompozicije, vrši se aproksimacija ranga k.

Ako je potrebno proceniti koliko bi se nekom korisniku svideo neki film, to se obično radi oslanjajući se na sličnost korisnika ili na sličnost filmova. U prvom slučaju se za datog korisnika nalazi najsličniji korisnik koji je gledao taj film i ocena koju je dao taj korisnik se koristi za procenu koliko bi se film svideo korisniku kojem se film preporučuje. Kao mera sličnosti se obično koristi kosinusno ili euklidsko rastojanje. Kao i u prošlom primeru, kao reprezentacije korisnika, koriste se redovi redukovane matrice U_k, a i reprezentacija novog korisnika se konstruiše na isti način. Suštinska razlika je u popunjavanju matrice prosecima i u fokusiranju na korisnike koji su gledali film. U prethodnom primeru, frekvencije svih reči su bile poznate i pri pretrazi su uzimani u obzir svi dokumenti.

Osnovni problem ovog pristupa je što korisnika često ima više nego filmova. Otud se nekada umesto sličnosti koristinika, koristi sličnost filmova, pa se za dati film nalazi njemu najsličniji koji je dati koristnik ocenio i ocena tog filma se uzima za ocenu datog filma. Prvi pristup se ipak pokazao kao precizniji.

Treba primetiti da je u oba prethodna primera vršena aproksimacija neke matrice matricom nižeg ranga. Ovaj korak se može posmatrati kao korak učenja. Naime, kao što modeli mašinskog učenja, sumiraju celokupne podatke nekakvim modelom koji zavisi od mnogo manjeg broja parametara, nego što ima brojeva u matrici podataka, tako i aproksimacija nižeg ranga smanjuje reprezentaciju podataka. Pritom, očekuje se da najmanjim singularnim vrednostima ne odgovaraju značajne komponente, već pre šum. Zapravo, polazni podaci se smatraju nepozudanima, a aproksimacijom nižeg ranga se izdvaja struktura koja postoji u podacima i razdvaja se od šuma.

4.3 Sopstveni vektori matrica

Kvadratna matrica A ima *sopstveni vektor* x i *sopstvenu vrednost* λ, ukoliko važi

$$Ax = \lambda x \qquad x \neq 0$$

Drugim rečima, matrica A ne menja pravac sopstvenog vektora. Poznato je da različitim sopstvenim vrednostima odgovaraju linearno nezavisni vektori, kao i da su sopstvene vrednosti jednake nulama karakterističnog polinoma

$$\det(A - \lambda I)$$

Svakoj sopstvenoj vrednosti λ_i, $i = 1, \ldots, m$ odgovara sopstveni potprostor V_i. Ukoliko se dimenzije ovih sopstvenih potprostora sabiraju na dimenziju matrice A, matrica se može svesti na dijagonalnu na sledeći način. Ako je D dijagonalna matrica sopstvenih vrednosti, u kojoj je svaka sopstena vrednost λ_i ponovljena $dim(V_i)$ puta i ako je X matrica čije su kolone sopstveni vektori, važi

$$AX = XD$$

Onda važi i

$$A = XDX^{-1} \qquad D = X^{-1}AX$$

Ukoliko pomenuti uslov vezan za dimenzije prostora V_i nije ispunjen, matrica X neće biti invertibilna. U slučaju realnih simetričnih matrica, uvek je moguće konstruisati odgovarajuću dijagonalnu matricu.

Značaj problema pronalaženja sopstvenih vektora je već demonstriran na primeru algoritma PageRank, koji predstavlja samo jednu u mnoštvu primena. Otud ne čudi da postoji mnoštvo metoda kojima se rešava ovaj problem. Pristupi se mogu podeliti na potpune, koji pronalaze sve sopstvene vektore date matrice, i delimične, koji pronalaze neke od njenih sopstvenih vektora. Ove metode najčešće zahtevaju ispunjenost određenih uslova. Međutim, provera ispunjenosti ovih uslova i sama može biti težak problem, tako da se obično vrši primena metode, a onda se naknadno proverava smislenost rezultata.

4.3.1 Potpune metode

Prvo će biti razmotrene potpune metode. Jedna grupa metoda se zasniva na pronalaženju sopstvenih vrednosti kao nula karakterističnog polinoma i rešavanju sistema jednačina $(A - \lambda I)x = 0$, ali ovaj pristup se u praksi ne pokazuje dobro. Umesto toga češće se koriste metode koje direktno operišu nad matricama. U specijalnom slučaju realnih simetričnih matrica, najpouzdanija je Jakobijeva metoda, zasnovana na primeni matrica rotacije kako bi se anulirali vandijagonalnih elementi. Iako svaka sledeća primena remeti ono što je prethodno postignuto, vandijagonalni elementi ipak postaju sve manji i matrica teži dijagonalnoj. Od matrica rotacije se množenjem može konstruisati matrica sopstvenih vektora. Ipak, ova metoda je za veće matrice značajno neefikasnija od alternativne, QR metode.

QR metoda se zasniva na zapažanju da prilikom konstrukcije niza matrica

$$A_1 = A$$

$$A_{i+1} = R_i Q_i$$

gde je $Q_i R_i$ QR dekompozicija matrice A_i, zahvaljujući ortogonalnosti matrica Q_i, važi

$$A_{i+1} = R_i Q_i = Q_i^T Q_i R_i Q_i = Q_i^T A_i Q_i$$

što znači da se prilikom zamene redosleda množenja čuva sličnost sa polaznom matricom. Pritom, pod određenim uslovima, niz matrica R_i konvergira matrici koja na dijagonali ima sopstvene vrednosti, a matrica sopstvenih vektora X se dobija kao prizovod $Q_1 Q_2 \cdots$. Ova metoda je brža od Jakobijeve, ali ima dodatne pretpostavke. Postoje varijante QR metode i za nesimetrične i za simetrične matrice. Obe su značajno efikasnije ukoliko iteriranje počne tek nakon svođenja matrice na posebne forme. U slučaju simetričnih matrica, to je gornjetorugaona forma, a u slučaju nesimetričnih matrica, to je gornja Hesenbergova forma, koja pored gornjetrougaonog dela uključuje još jednu sporednu dijagonalu. O detaljima ovih metoda neće biti reči.

4.3.2 Delimične metode

Delimične metode omogućavaju traženje samo nekih sopstvenih vektora, po pravilu dominantnog sopstvenog vektora – onog koji odgovara po modulu najvećoj sopstvenoj vrednosti. Među njima je najpoznatija *metoda stepenovanja* (eng. *power method*). Neka su $x_1, \ldots, x_n$ sopstveni vektori matrice $A \in \mathbb{R}^{n \times n}$. Neka je v_0 proizvoljan vektor. Kako sopstveni vektori čine bazu vektorskog prostora, važi

$$v_0 = \alpha_1 x_1 + \ldots + \alpha_n x_n$$

Neka je

$$v_k = A^k v_0$$

Onda važi

$$v_k = \lambda_1^k \alpha_1 x_1 + \lambda_2^k \alpha_2 x_2 + \ldots + \lambda_n^k \alpha_n x_n$$

Ako važi $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|$ i $\alpha_1 \neq 0$, onda važi

$$\lambda_2^k \alpha_2 x_2 + \ldots + \lambda_n^k \alpha_n x_n = o(\lambda_1^k \alpha_1 x_1)$$

odnosno vektor v_k konvergira ka sopstvenom vektoru kolinearnom sa x_1. Prilikom računanja, kako ne bi došlo do prekoračenja, potrebno je vektor v_k normirati povremeno ili u svakom koraku. Odgovarajuća sopstvena vrednost je aproksimirana izrazom

$$\frac{v_k^T A v_k}{v_k^T v_k}$$

Čak i ako važi $\alpha_1 = 0$, u praksi se zbog računaskih grešaka pojavljuje komponenta u pravcu sopstvenog vektora x_1, što dovodi do konvergencije, ali sporije. Ukoliko matrica ima nekoliko po modulu jednakih sopstvenih vrednosti, konvergencija nije garantovana.

Primer 57 *Algoritam PageRank se zasniva na metodi stepenovanja, pošto matrica G zadovoljava uslove primene ovog metoda. Očigledan izazov primeni metoda stepenovanja je velika dimenzija matrice G i njena gustina – svi elementi su strogo pozitivni. Međutim, važi:*

$$\begin{aligned} Gp &= (\alpha(A + vk^T) + (1-\alpha)ve^T)p \\ &= \alpha(A + vk^T)p + (1-\alpha)v \\ &= \alpha Ap + \alpha vk^T p + (1-\alpha)v \end{aligned}$$

Očigledno, jedina matrica kojom je potrebno vršiti množenje je retka matrica A. Vektor $\alpha vk^T p$ se može izračunati tako što se prvo izračuna vektor $k^T p$, a potom se pomnoži sleva vektorom αv. Na taj način se ne formira gusta matrica αvk^T. Odnosno, kako bi se primenio metod stepenovanja, dovoljno je vršiti množenje retkom matricom uz dodatna sabiranja vektora, što čini ovaj metod efikasnim.

Postoje i druge delimične metode, koje takođe izračunavaju dominantni sopstveni vektor, ali o njima neće biti reči. Važnije pitanje je kako izračunati i ostale sopstvene vektore, koristeći neku od tih metoda. Neka važi $|\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$ i neka su $y_1, \ldots, y_n$ sopstveni vektori matrice A^T, skalirani tako da važi $x_i \cdot y_j = \delta_{ij}$[2] (što je pod datim uslovom moguće). Neka je

$$A_1 = A$$

$$A_{i+1} = A_i - \lambda_i x_i y_i^T \qquad i = 2, \ldots, n$$

Sopstveni vektori i vrednosti se dobijaju primenom metoda stepenovanja na matrice A_i i A_i^T za $i = 1, \ldots, n$. Ova metoda se naziva *metodom iscrpljivanja.*

4.3.3 Analiza glavnih komponenti

U primenama, podaci se često predstavljaju visokodimenzionalnim vektorima. Primera radi, slike se često sastoje iz velikog broja piksela. Ipak, iako je prostor visokodimenzionalan, većina vektora podataka često leži u niskodimenzionalnoj površi unutar tog prostora ili vrlo blizu neke takve površi. Ukoliko bi se podaci projektovali na nju, izgubio bi se deo informacije, ali bi se dimenzionalnost podataka mogla, često i drastično, smanjiti. Na primer, u skupu svih mogućih slika, skup lica čini prostor manje dimenzije. Ovakva transformacija je poželjna kako zbog skladištenja podataka, tako i zbog toga što statistički

[2] $\delta_i j = 1$ ako važi $i = j$, a $\delta_{ij} = 0$, inače.

algoritmi i algoritmi mašinskog učenja često daju bolje rezultate ukoliko podaci imaju nižu dimenzionalnost. U nekim slučajevima olakšava i razumevanje podataka.

Primer 58 *Veliki broj algoritama mašinskog učenja (poput algoritma k najbližih suseda), počiva na upotrebi neke mere rastojanja, poput euklidskog rastojanja ili sličnosti, poput kosinusa ili skalarnog proizvoda. Sve ove funkcije su u praksi osetljive na prisustvo redundantnosti u opisu podataka. Neka se objekti predstavljaju vektorima vrednosti nekih atributa. Neka je broj atributa 101 i neka su prvih 100 atributa identični. U tom slučaju prilikom računanja euklidskog rastojanja tih 100 atributa dominiraju nad onim jednim koji se od njih razlikuje i njegov uticaj je praktično zanemarljiv.*

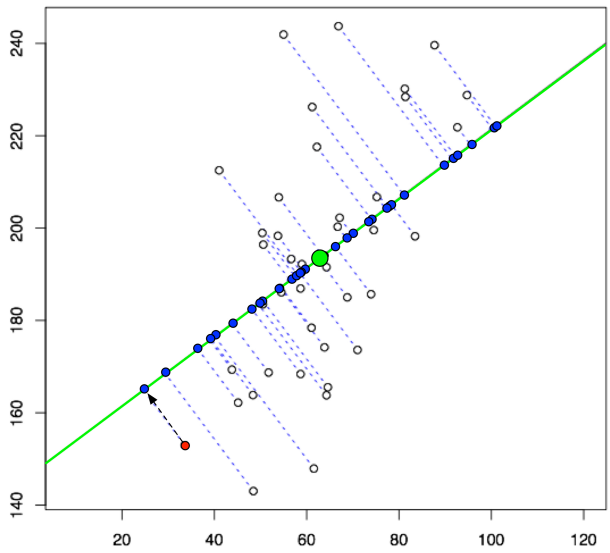
$$d(x,y) = \sqrt{100(x_1 - y_1)^2 + (x_{101} - y_{101})^2} = 10\sqrt{(x_1 - y_1)^2 + \frac{1}{100}(x_{101} - y_{101})^2}$$

Na taj način se umesto informacija koju pružaju dva atributa, koristi samo informacija koju pruža jedan atribut.

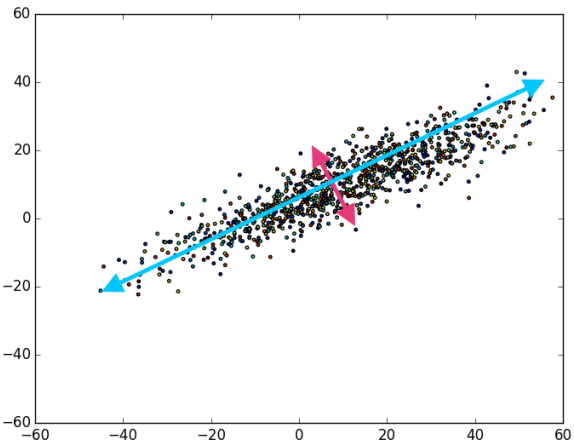
Navededeni primer je očito ekstreman, ali je efekat ovog problema osetan i u realnim slučajevima. Stoga bi bilo poželjno eliminisati redundantnosti u informaciji koju pružaju atributi. Međutim, redundantnost ne mora biti i najčešće nije proizvod pukog ponavljanja atributa, već se češće radi o korelacijama i približnim linearnim zavisnostima među atributima. Eliminacija ovakvih redundantnosti nije trivijalan zadatak.

Jedno od zapažanja na kojem se zasnivaju metode analize podataka je da varijacija promenljivih nosi informaciju. S druge strane, koreliranost promenljivih znači da promenljive nose manje informacije nego što njihova brojnost sugeriše. Stoga, racionalan pristup je naći skup vektora duž kojih je varijacija podataka najveća, takvih da projekcije podataka na njih nisu korelirane, izabrati ih za novu bazu prostora podataka i zanemariti one koordinate koje odgovaraju vektorima duž kojih je varijacija podataka najmanja. Kako bi projekcije bile nekorelirane, konstruisani vektori treba da budu ortogonalni. Ovi vektori, koji se nazivaju *glavnim komponentama* se mogu pronaći tako što se pronađe, prvo vektor duž kojeg podaci najviše variraju, tako što se varijabilnost podataka duž tog vektora ukloni, a potom se proces nastavi. Varijabilnost podataka duž nekog pravca se tiče raštrkanosti projekcija podataka na taj pravac, kao što je ilustrovano slikom 4.10. Dve glavne komponente na primeru dvodimenzionih podataka prikazane su na slici 4.11. Prva pretpostavka metode je da su proseci svih kolona matrice podataka jednaki 0. To u praksi nije zadovoljeno i zbog toga se od svih elemenata svake kolone oduzima prosek te kolone i nadalje se radi sa tako transformisanim podacima. Takvo transformisana matrica će biti označena sa X. Prva glavna komponenta se dobija kao rešenje problema

$$\max_{\|v\|=1} \sum_{i=1}^{N} (x_i \cdot v)^2 = \max_{\|v\|=1} \|Xv\|_2^2 = \max_{v} \frac{\|Xv\|_2^2}{\|v\|_2^2} = \max_{v} \frac{\|Xv\|_2}{\|v\|_2}$$



Slika 4.10: Varijabilnost podataka duž nekog pravca se odnosi na raštrkanost projekcija tih podataka na dati pravac. Prikazani pravac nije optimalan, već bi ga trebalo rotirati suprotno smeru kretanja kazaljke na satu.



Slika 4.11: Dve glavne komponente dovdimenzionalnog skupa podataka.

Očigledno, prema definiciji matričnih p normi, maksimalna vrednost veličine koja se maksimizuje je $\|X\|_2$, što je takođe najveća singularna vrednosti σ_1 matrice X, što je koren po modulu najveće sopstvene vrednosti λ_1 matrice X^TX. Ako se za vektor v uzme odgovarajući sopstveni vektor v_1 matrice X^TX, dobija se

$$\frac{\|Xv\|_2}{\|v\|_2} = \left(\frac{v^TX^TXv}{v^Tv}\right)^{1/2} = \left(\lambda_1\frac{v^Tv}{v^Tv}\right)^{1/2} = \sqrt{\lambda_1}$$

Očito, maksimalna vrednost se dostiže kada je v dominantan sopstveni vektor matrice X^TX. Sledeća glavna komponenta se dobija tako što se ovaj postupak primeni na matricu

$$X_1 = X - Xv_1v_1^T$$

Ispostavlja se da je dominantni sopstveni vektor matrice X_1^TX, drugi sopstveni vektor matrice X^TX (koji odgovara sopstvenoj vrednosti λ_2). Ovaj postupak se može nastaviti i ispostavlja se da je potrebno naći sve sopstvene vektore i sopstvene vrednosti matrice X^TX. Poznato je da su sopstveni vektori ove matrice desni singularni vektori matrice X, odnosno ako je singularna dekompozicija matrice X jednaka $U\Sigma V^T$, onda su kolone matrice V, sopstveni vektori matrice X^TX, pa je očito, još jedan način nalaženja glavnih komponenti, singularnom dekompozicijom matrice X.

Ako su $v_1, v_2, \ldots, v_n$, glavne komponente, svaki podatak x se može predstaviti pomoću koordinata $(x\cdot v_1, \ldots, x\cdot v_n)$, odnosno matrica X se može zameniti matricom XV, čime su polazni podaci predstavljeni u odnosu na bazu glavnih komponenti. Smanjenje dimenzionalnosti se postiže izostavljanjem kolona koje odgovaraju manje važnim glavnim komponentama. Zahvaljujući vezi sa singularnim vektorima i vrednostima matrice X, svakoj od komponenti se može pridružiti udeo varijanse koju ona predstavlja, prema ranije navedenoj formuli

$$\frac{\sigma_i^2}{\sum_{j=1}^n \sigma_j^2}$$

Tako se može izračunati procenat varijabilnosti koji se gubi izostavljanjem određenih komponenti, a udeo varijanse objašnjen pomoću prvih k komponenti je

$$\sum_{i=1}^{k} \frac{\sigma_i^2}{\sum_{j=1}^n \sigma_j^2}$$

Prethodno opisana tehnika se naziva *analizom glavnih komponenti* (eng. *principal component analysis*) i primenljiva je u najraznovrsnijim kontekstima u kojma je potrebno naći ortonormiranu bazu potprostora u kojem je sadržana glavnina varijabilnosti podataka. Ipak, ovim postupkom nije moguće pronaći proizvoljne površi, već linearne potprostore. Uopštenje na neprekidne površi moguće je postići, na primer, korišćenjem specifičnih vrsta neuronskih mreža – *autonekodera*.

Matrica $X^T X$ predstavlja matricu kovarijacije kolona polazne matrice podataka. Pored centriranja, kolone matrice podataka je moguće i podeliti standardnim devijacijama tih kolona. Tako se dobijaju kolone kojima je prosek 0, a standardna devijacija 1, a koje se nazivaju *standardizovanim*. U tom slučaju, matrica $X^T X$, predstavlja matricu korelacije kolona polazne matrice podataka. Postavlja se pitanje da li je prilikom analize glavnih pravaca bolje koristiti matricu kovarijacije ili matricu korelacije, odnosno da li treba standardizovati kolone polazne matrice podataka. U zavisnosti od ove odluke, rezultati mogu biti različiti. Kolone se ne standardizuju kada promenljive prirodno variraju u sličnim rasponima i mere se na istim skalama. Na primer, ako svaka kolona matrice predstavlja visine ljudi iz različitih država izražene u centimetrima. Iako ljudi iz različitih krajeva sveta mogu imati različite visine, ipak je skala ista i ne bi trebalo da bude drastičnih odstupanja (na primer, za red veličine). U ovakvoj situaciji, standardizacija može dovesti do gubitka informacije. Naime, analiza glavnih komponenti se bavi pronalaženjem pravaca u kojima je varijacija maksimalna. Pre standardizacije, kovarijacije promenljivih sa samim sobom mogu biti različite što ukazuje na promenljive koje više variraju, dok je nekon standardizacije, kovarijacija svake promenljive sa samom sobom jednaka 1, što utiče na ishod primene metode. S druge strane, ukoliko kolone nisu merene na istoj skali (na primer, visina u centimetrima i visina u metrima) ili ne variraju u istim rasponima (na primer, ljudska visina u metrima i visina planina u metrima), poželjno je koristiti matricu korelacije, a ne kovarijacije, pošto će u suprotnom kolone koje sadrže vrednosti sa najvećim brojevima, dominirati glavnim komponentama, zbog svoje naizgled velike varijacije. Odnosno, promenom skale, na primer tako što će se visina meriti u milimetrima, umesto u metrima, data promenljiva može postati važnija od ostalih u smislu da bi joj prva glavna komponenta mogla biti skoro kolinearna.

Primeri upotrebe analize glavnih komponenti su mnogobrojni – praktično u bilo kom poslu analize podataka u prisustvu velikog broja koreliranih promenljivih. Ipak, problem detekcije i prepoznavanja lica je karakterističan po tome što u njemu ova metoda igra centralnu ulogu.

Primer 59 *Potrebno je napraviti sistem koji je u stanju da automatski zaključuje da li se na slici nalazi lice ili ne. U tom cilju, neka je dat skup slika lica. Jednostavnosti radi, neka su lica prikazana u nijansama sive. Svaka slika se može predstaviti kao vektor – nizanjem kolona piksela jedne ispod druge. Primer skupa lica je dat na slici 4.12. Iako slike lica mogu imati veliki broj piksela kojima su predstavljene, one predstavljaju samo mali udeo u skupu svih mogućih slika istih dimenzija, odnosno, može se pretpostaviti da leže u potprostoru značajno manje dimenzije u odnosu na prostor svih slika. Međutim, postavlja se pitanje koji su bazni vektori tog prostora, odnosno, da li se sva lica mogu približno dobro predstaviti kao linearne kombinacije nekog skupa slika. Za očekivati je da i te slike predstavljaju nekakva lica. Ideja je da se ona odrede analizom glavnih komponenti, a kako ona počiva na pronalaženju sopstvenih vektora, ta lica se nazivaju* sopstvenim licima *(eng.* eigenfaces*). Neka*

Slika 4.12: Nekoliko slika lica.

su $x_1, \ldots, x_N$ *sve raspoložive slike lica i neka je* $\overline{x}$ *prosečno lice. Kao što je rečeno, ono se oduzima od svih lica i matrica podataka* X*, predstavlja matricu čije su vrste* $x_1 - \overline{x}, \ldots, x_N - \overline{x}$*. Singularnom dekompozicijom matrice* X*, dobija se matrica* V *sopstvenih lica. Prosečno lice i neka sopstvena lica, prikazana su na slici 4.13. Prostor lica je prostor*

$$\left\{ \overline{x} + \sum_{i=1}^{n} \alpha_i V_i | \alpha_i \in \mathbb{R}, i = 1, \ldots, n \right\}$$

koji razapinju sopstvena lica. Umesto svih sopstvenih lica V_i*, zadržavaju se samo najvažnija u skladu sa procenom gubitka informacije na osnovu singularnih vrednosti. Matrica tih sopstvenih lica se označava sa* W*.*

Kako bi se odredilo da li neka slika predstavlja lice, potrebno ju je projektovati na potprostor lica i odrediti rastojanje od slike do njene projekcije, što je najbliža slika u prostoru lica. Ukoliko je to rastojanje manje od nekog unapred određenog praga, onda se smatra da slika predstavlja lice, a u suprotnom se smatra da ne predstavlja lice. Pomenuto rastojanje slike x *do njene projekcije se računa po formuli*

$$\|(I - WW^T)(x - \overline{x})\|$$

Naime $x - \overline{x}$ *predstavlja odstupanje od prosečnog lica, množenje ovog vektora matricom* W^T *predstavlja određivanje koordinata ovog vektora u prostoru*

Slika 4.13: Prosečno lice, dva najvažnija sopstvena lica i tri najmanje važna sopstvena lica.

lica, što uključuje i njegovo projektovanje, a množenje matricom W predstavlja izražavanje projekcije u koordinatama polaznog prostora. Otud se množenjem matricom $I - WW^T$ dobija vektor normale, čija norma predstavlja traženo rastojanje. Slika 4.14 prikazuje sliku x i njenu reprezentaciju pomoću sopstvenih lica

$$WW^T(x - \overline{x}) + \overline{x}$$

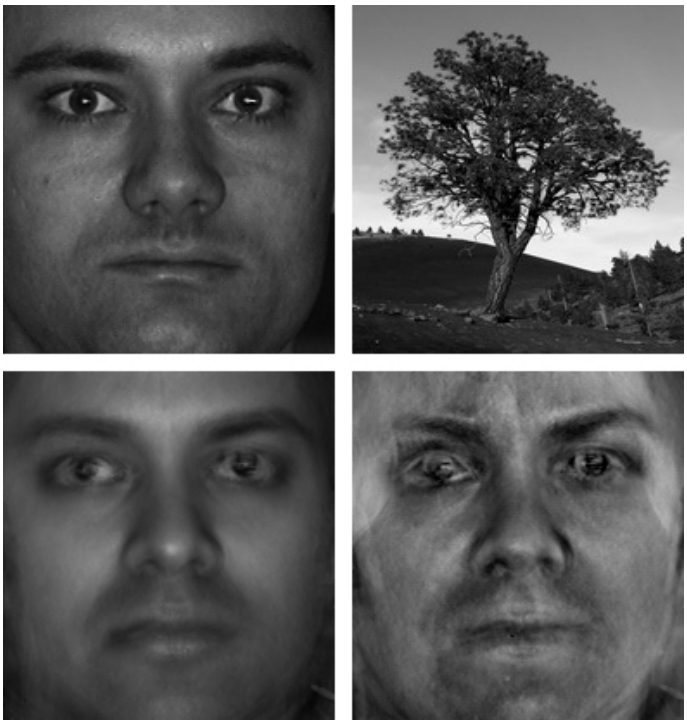
u slučaju slike lica i u slučaju slike drveta. U slučaju slike lica, očigledno je da joj je najbliža slika u prostoru lica vrlo blizu. S druge strane, u slučaju slike stabla, očigledno je da joj je najbliža slika u prostoru lica vrlo daleko.

Ukoliko nije dovoljno samo detektovati lice na slici, već i prepoznatli lice, odnosno odrediti osobu kojoj lice pripada, potrebno je ovo lice uporediti sa već znatim licima $\{y_1, \ldots, y_M\}$ za koja se pretpostavlja da su predstavljena u koordinatama prostora lica, odnosno potrebno je rešiti problem

$$\min_k \|W^T(x - \overline{x}) - y_k\|$$

4.4 Retki sistemi linearnih jednačina

U praksi je čest slučaj da sistemi jednačina koji se rešavaju imaju neko specifično svojstvo ili neku specifičnu strukturu. U takvim situacijama, moguće je



Slika 4.14: Izvorne slike (gore) i njihove reprezentacije pomoću sopstvenih lica.

dizajnirati posebne metode rešavanja takvih sistema (poput metoda dekompozicije matrice) koje uzimaju u obzir pomenute specifičnosti kako bi se postupak rešavanja sproveo efikasnije, kako u odnosu na potrebni broj operacija, tako i u odnosu na potrebni memorijski prostor.

Umesto da se govori o svojstvima i strukturi samih sistema, obično se govori o svojstvima i strukturi odgovarajućih matrica. Od posebnog značaja su takozvane *retke matrice* (eng. *sparse matrix*). Retke matrice se karakterišu malim brojem nenula elemenata, što omogućava efikasno skladištenje matrice, zahvaljujući tome što se nule ne moraju čuvati, kao i efikasnije izračunavanje, zahvaljujući jednostavnosti aritmetičkih operacija koje uključuju nulu. Pojam retke matrice nije formalno definisan i može se koristiti kad god je zahvlajujući broju nula moguće smanjiti memorijsku ili vremensku zahtevnost izračunavanja.

Retke matrice mogu imati specifičnu strukturu, odnosno raspored nula i nenula elemenata u matrici, koji može voditi različitim algoritmima rešavanja odgovarajućih sistema. Trivijalan primer je dijagonalna matrica. Za njeno čuvanje je potrebno $\Theta(n)$ memorijskih lokacija, a za rešavanje odgovarajućeg sistema, očigledno $\Theta(n)$ računskih operacija. Isto važi i u slučaju trodijagonal-

nih matrica oblika

$$\begin{bmatrix} b_1 & c_1 & 0 & \cdots & & & \\ a_2 & b_2 & c_2 & \cdots & & & \\ & & & \cdots & & & \\ & & & \cdots & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & \cdots & 0 & a_n & b_n \end{bmatrix}$$

Na primer, algoritam LU dekompozicije se može prilagoditi tako da zahteva $\Theta(n)$ operacija, a potom je dovoljan isti red broja operacija za rešavanje sistema. Još jedan vrlo važan slučaj specifične strukture matrice je blok-dijagonalna matrica oblika

$$\begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_n \end{bmatrix}$$

gde su $A_1, A_2, \ldots, A_n$ kvadratne matrice koje se često nazivaju blokovima ili dijagonalnim blokovima matrice A. Ako sistemu $Ax = b$ odgovara ovakva matrica, nije potrebno rešavati ga odjednom, već je dovoljno odvojeno rešiti podsisteme koji odgovaraju blokovima

$$A_1 x_1 = b_1$$

$$A_2 x_2 = b_2$$

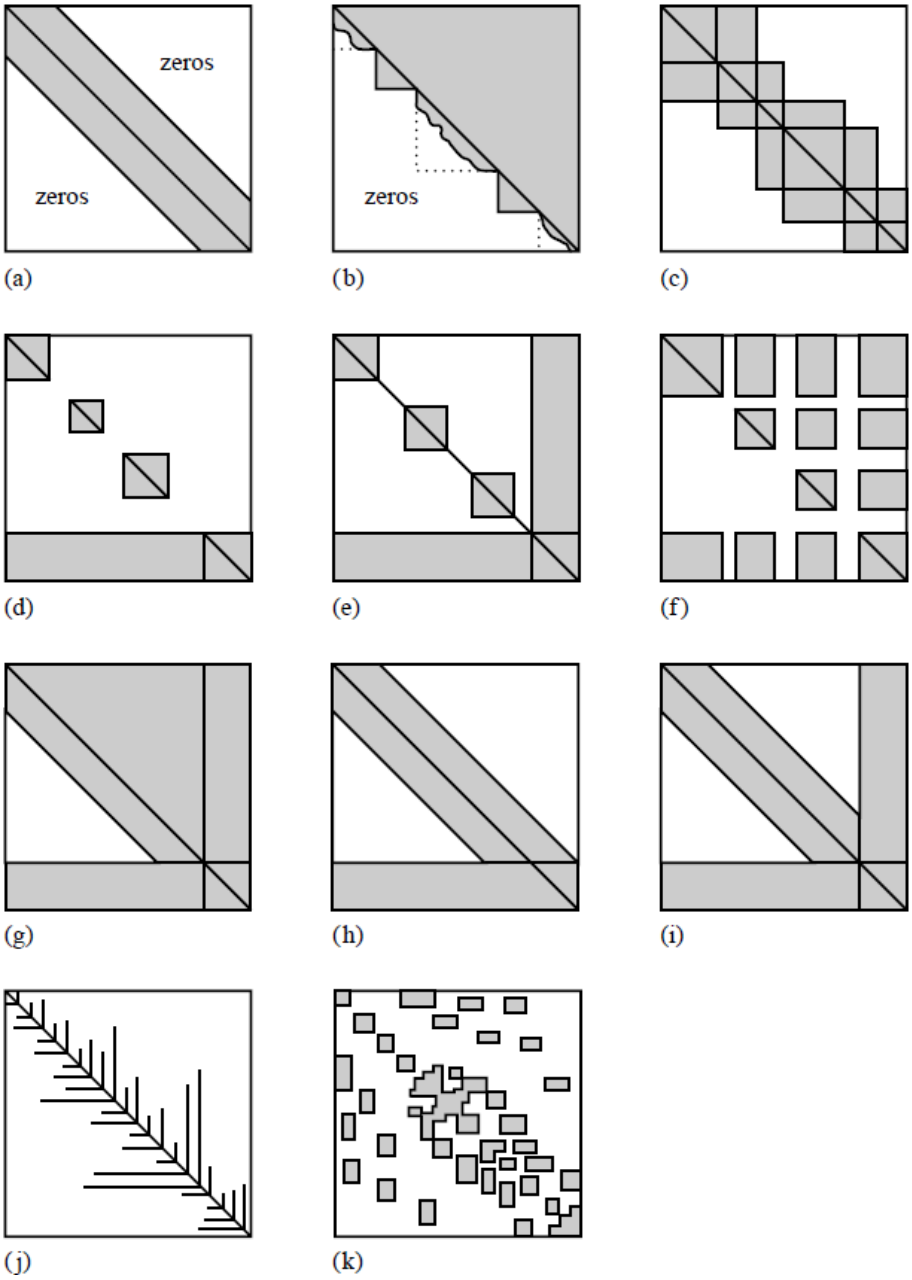
$$\vdots$$

$$A_n x_n = b_n$$

gde su $b_1, b_2, \ldots, b_n$ delovi vektora b koji odgovaraju blokovima $A_1, A_2, \ldots, A_n$, a $x_1, x_2, \ldots, x_n$ odgovarajući delovi vektora x. Ukoliko je za rešavanje punog sistema potrebno $\Theta(n^3)$ operacija, za odvojeno rešavanje m sistema koji odgovaraju blokovima dimenzije n/m, potrebno je $m\Theta((n/m)^3) = \Theta(n^3)/m^2$, što u praksi može biti vrlo značajno ubrzanje. Slično, inverz matrice A je

$$\begin{bmatrix} A_1^{-1} & 0 & \cdots & 0 \\ 0 & A_2^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_n^{-1} \end{bmatrix}$$

Neke od matrica prepoznatljive strukture su date na slici 4.15. To su *trakasta*, *blok-trougaona*, *blok-trodijagonalna*, *blok-dijagonalna sa ivicom*, *blok-dijagonalna sa dve ivice*, *blok trougaona sa ivicom*, *trakasto-trougaona sa ivicom*, *trakasta sa ivicom*, *trakasta sa dve ivice* i ostale.



Slika 4.15: Različite vrste retkih matrica.

4.5 Inkrementalni pristup rešavanju problema linearne algebre

U nekim situacijama su dekompozicija ili inverz matrice već poznati. U slučaju male promene matrice čiji je inverz poznat, naivni pristup podrazumeva izračunavanje iz početka. Postavlja se pitanje, da li je u slučaju malih promena moguće efikasno ažurirati već postojeću dekompoziciju ili inverz. Slično razmatranje je moguće u slučajevima matrica specifične strukture kada je do dekompozicije ili inverza lakše doći nego u opštem slučaju. U slučaju malog odstupanja od specifične strukture za koju je poznat efikasan algoritam, taj algoritam postaje neupotrebljiv i potrebno je primeniti opštiji, manje efikasan algoritam. Tada se postavlja pitanje da li je moguće rešiti sistem zanemarujući to odstupanje, a potom korigovati rešenje, tako da se ta razlika uzme u obzir. Odgovori na ova pitanja su od velikog pratkičnog značaja. Naime, izmene u polaznoj matrici mogu dolaziti od izmena u određenim podacima koje ta matrica predstavlja. Na primer, gledalac može promeniti svoju ocenu filma ili se može pojaviti jedan nov gledalac ili jedan nov film. Pritom, matrice kojima se barata u ovakvim primenama mogu biti ogromne i često rešavanje sistema iz početka nije dopustivo.

Odgovor na prethodna pitanja je pod određenim pretpostavkama pozitivan. Najpoznatiji primer je dat *Šerman-Morison-Vudburijevom formulom* (eng. *Sherman-Morrison-Woodburry formula*) u slučaju modifikacija niskog ranga:

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1} \tag{4.1}$$

gde je A matrica dimenzija $n \times n$, a U i V matrice dimenzija $n \times p$, za $p < n$, a očekivano je da p, koje predstavlja rang modifikacije, bude mnogo manje od n. Ova formula zahteva inverziju matrice $I + V^T A^{-1}U$, dimenzija $p \times p$, što se značajno pojednostavljuje u slučaju da važi $p = 1$. U literaturi se često govori o Šerman-Morisonovoj formuli u slučaju da važi $p = 1$, a o Vudburijevoj formuli kao o opštem slučaju.

Primer 60 *Neka je A blok-dijagonalna matirca i neka je A' matrica čiji su svi elementi jednaki matrici A, osim elementa u gornjem desnom uglu, koji je u matrici A' jednak 1, umesto 0. Rešavanje sistema $Ax = b$ je efikasno zahvlajujući blokdijagonalnoj strukturi, ali ne važi isto za sistem $A'x = b$ zbog toga što matrica A' nije blok dijagonalna. Međutim, ona se može predstaviti kao*

$$A + uv^T$$

gde važi $u = (1, 0, \ldots, 0, 0)^T$ i $v = (0, 0, \ldots, 0, 1)^T$. Primena formule 4.1, omogućava efikasno rešavanje sistema.

Data formula pretpostavlja da je u memoriji moguće čuvati inverz A^{-1}. Tako nešto nije uvek moguće. Recimo, u slučaju kada je polazna matrica retka, a inverz to nije, što nije neuobičajeno. U tom slučaju je potrebno čuvati

retku dekompoziciju matrice A i osloniti se na rešavanje sistema pomoću nje. U tom slučaju, za rešenje sistema $(A + UV^T)x = b$ važi

$$\begin{aligned} x &= (A + UV^T)^{-1}b = \\ &= (A^{-1} - A^{-1}U(I + V^TA^{-1}U)^{-1}V^TA^{-1})b \\ &= \underbrace{A^{-1}b}_{y} - \underbrace{A^{-1}U}_{Z}\underbrace{(I + V^T\underbrace{A^{-1}U}_{Z})^{-1}}_{H}V^T\underbrace{A^{-1}b}_{y} \\ &= y - ZH^{-1}V^Ty \end{aligned}$$

Očigledno, vektor y je rešenje sistema

$$Ay = b$$

i može se izračunati bez eksplicitnog računanja inverza, recimo korišćenjem metoda dekompozicije za retke matrice. Matrica Z se može izračunati, tako što se izračunaju njene kolone $z_1, z_2, \ldots, z_p$, rešavanjem p retkih sistema

$$Az_1 = U_1$$

$$Az_2 = U_2$$

$$\vdots$$

$$Az_p = U_p$$

Slično, vektor $H^{-1}V^Ty$ se može izračunati rešavanjem sistema

$$Hw = V^Ty$$

umesto inverzijom. Iako je matrica H malih dimenzija i njen inverz nije teško čuvati u memoriji, rešavanje sistema je često numerički stabilnije od računanja inverza.

U prethodnom razmatranju pomenuto je čuvanje retke dekompozicije. Isto razmatranje se odnosi i na direktno ažuriranje rešenja sistema $Ax = b$ bez čuvanja bilo čega drugog. U tom slučaju potrebno je računati matricu Z, ali ako je već rešavanje sistema koji uključuje matricu A jednostavno, to je prihvatljivo.

Primene Šerman-Morison-Vudburijeve formule su mnogobrojne. Prvi primer pokazuje kako se može graditi sistem u koji se inkrementalno dodaju podaci, u slučaju primene metode najmanjih kvadrata.

Primer 61 *Neka je rešen problem najmanjih kvadrata, na primer zarad rešavanja problema linearne regresije. Rezultat je vektor parametara w. U jednačinama*

normale figuriše matrica $X^T X$. Ukoliko se među postojeće podatke doda i vektor x, u jednačinama normale treba umesto matrice X da figuriše matrica

$$X' = \begin{bmatrix} X \\ x \end{bmatrix}$$

a umesto matrice $X^T X$, matrica $X^T X + x^T x$. Ovo je očigledno modifikacija ranga 1. Ukoliko je inverz matrice $X^T X$ već poznat, onda se inverz modifikacije ažurira prema formuli 4.1.

Formula 4.1 izražava kako se ažurira inverz matrie. Međutim, kao što je ranije rečeno, računanje i čuvanje inverza nije poželjna praksa, već je uvek bolje rešavati sistem oslanjajući se na dekompoziciju matrice. Stoga se postavlja pitanje, mogu li se dekompozicije matrica lako modifikovati ukoliko se modifikuje polazna matrica. Odogovor je pozitivan – zaista postoje tehnike ažuriranja dekompozicija matrica, zasnovane na formuli 4.1, ali o kojima neće biti reči. Idealan primer za primenu ovakvih tehnika su sistemi za preporučivanje, koji počivaju na singularnoj dekompoziciji, a kod kojih je uobičajeno da se podaci menjaju – dodavanjem ocena koje su gledaoci dodelili filmovima, njihovom izmenom ili dodavanjem novih gledalaca ili filmova.

Primer 62 *Prilikom rangiranja strana, umesto metoda stepenovanja, moguće je i direktno rešavati sistem jednačina*

$$\begin{aligned} p &= Gp \\ &= \alpha Ap + \alpha v k^T p + (1-\alpha) v e^T p \\ &= \alpha Ap + \alpha v k^T p + (1-\alpha) v \end{aligned}$$

odnosno

$$(I - \alpha A - \alpha v k^T) p = (1-\alpha) v$$

Kako je matrica $S = I - \alpha A$ retka, sistem $Sy = (1-\alpha)v$ se rešava lako. Imajući u vidu da je $-\alpha v k^T$ modifikacija ranga 1, primenom formule 4.1 dobija se

$$(S - \alpha v k^T)^{-1} = S^{-1} + \frac{S^{-1} v k^T S^{-1}}{\frac{1}{\alpha} + k^T S^{-1} v}$$

Odatle sledi da važi

$$\begin{aligned} p &= (1-\alpha)(S-\alpha v k^T)^{-1} v \\ &= (1-\alpha)S^{-1}v + \frac{(1-\alpha)S^{-1}vk^T(1-\alpha)S^{-1}v}{\frac{1-\alpha}{\alpha} + k^T(1-\alpha)S^{-1}v} \\ &= y + \frac{yk^Ty}{\frac{1-\alpha}{\alpha} + k^Ty} \\ &= \left[1 + \frac{k^Ty}{\frac{1-\alpha}{\alpha} + k^Ty}\right] y \end{aligned}$$

Kako su vektori p i y kolinearni, a poznato je da se elementi vektora p sabiraju na 1, p se računa tako što se vektor y podeli sumom svojih koordinata.

Glava 5

Matematička optimizacija

Matematička optimizacija je jedna od najprimenjenijih grana matematike. Njen cilj je definisanje metoda pronalaženja minimuma i maksimuma funkcija. Opšti *problem optimizacije* je obično oblika:

$$\begin{aligned} &\min_{x \in \mathcal{D}} f(x) \\ &\text{pri ogr.} \quad g_i(x) \leq 0 \qquad i = 1, \ldots, M \end{aligned}$$

pri čemu se funkcija f naziva *ciljnom funkcijom*, skup $\mathcal{D}$ domenom, a uslovi vezani za g_i *ograničenjima*. Objekat iz domena koji zadovoljava sva ograničenja, naziva se *dopustivo rešenje*. Potrebno je među svim dopustivim rešenjima naći ono za koje je vrednost ciljne funkcije najmanja. Ova formulacija obuhvata i pronalaženje maksimuma, pošto se pronalaženje maksimuma funkcije f može svesti na pronalaženje minimuma funkcije $-f$. Zato će u nastavku biti reči isključivo o metodama pronalaženja minimuma, odnosno *minimizacije*. Takođe, treba primetiti da se i jednakosna ograničenja lako uklapaju u navedeni okvir. Naime, ograničenje $g(x) = 0$ se može predstaviti pomoću dva ograničenja – $g(x) \leq 0$ i $-g(x) \leq 0$.

Kako je česta potreba da se neki posao uradi na najbolji način, primene metoda matematičke optimizacije su mnogobrojne. Obuhvataju najrazličitije probleme raspoređivanja (npr. raspoređivanje časova, avionskih letova, aerodroma, proizvodnje), transporta (npr. optimizacija automobilske transportne mreže ili mreže transporta gasa) i komunikacija (optimizacija računarskih mreža), skoro sve probleme mašinskog učenja, neke metode automatskog dizajna hardvera, računarskog opažanja, robotike, odlučivanja, ekonomije i finansija (npr. optimizacija bankarskog portfolija), biologije (npr. ustanovljavanje načina savijanja proteina), građevine, geonauka (npr. ocena Zemljinog magnetnog polja, automatsko kartiranje, preraspodela zemljišta), arheologije (npr. rekonstrukcija objekata od pronađenih fragmenata) i tako dalje.

Broj metoda matematičke optimizacije je ogroman. Razlikuju se pre svega po pretpostavkama o svojstvima problema na koje se primenjuju, a potom i

po pristupima rešavanju. Metode koje podrazumevaju više pretpostavki su obično efikasnije na takvim problemima od opštijih metoda. Često se definišu optimizacione metode i za pojedinačne probleme. Ove metode se mogu podeliti u grupe po više kriterijuma vezanih za važna svojstava problema optimizacije:

Lokalnost U definiciji problema optimizaije nije naglašeno da li se traži lokalni ili globalni minimum. Obe vrste problema su od praktičnog značaja. Poznavanje globalnog minimuma je najpoželjnije, ali ne postoje egzaktne metode *globalne optimizacije*. Takve metode zasnovane su na različitim heuristikama i ne daju garancije pronalaženja globalnog minimuma. Metode *lokalne otpimizacije* su često egzaktne, odnosno često daju garancije nalaženja lokalnog minimuma.

Neprekidnost Problemi i metode optimizacije se drastično razlikuju u zavisnosti od toga da li je domen diskretan ili neprekidan skup. Metode *diskretne optimizacije* su često zasnovane na principima koji svoje opravdanje nalaze u kombinatorici. U njima se obično javlja kombinatorna eksplozija, pa se, kako bi se ona kontrolisala, često, ali ne i nužno, zasnivaju na heurističkim pristupima. U tom slučaju, ne garantuju nalaženje optimalnog rešenja. Problemi raspoređivanja su tipični problemi diskretne optimizacije. Metode *neprekidne optimizacije* su zasnovane na matematičkoj analizi i tipično su efikasnije, mada je ovakvo poređenje nezahvalno, pošto se ne primenjuju na isti skup problema. Primer neprekidne optimizacije može biti većina metoda mašinskog učenja. Ostali kriterijumi se odnose samo na neprekidnu optimizaciju.

Diferencijabilnost Ukoliko su i ciljna funkcija i ograničenja diferencijabilni, radi se o diferencijabilnom optimizacionom problemu. Većina metoda neprekidne optimizacije je upravo ovog tipa i najčešće se u svojoj formulaciji oslanjaju na pojam *gradijenta* – vektora parcijalnih izvoda. U slučaju da su funkcije dva puta diferencijabilne, kao dodatni izvor informacije o funkciji koja se minimizuje, može se koristiti i *hesijan* – matrica drugih parcijalnih izvoda.

Konveksnost Optimizacioni problem je konveksan ako su i ciljna funkcija i ograničenja konveksni. Ovo je veoma poželjno svojstvo optimizacionog problema, zato što garantuje postojanje samo jednog optimuma, kao i zato što u tom slučaju metode optimizacije tipično brže pronalaze željeni optimum. Ukoliko problem nije konveksan, moguće je pronalaženje lokalnog optimuma, koji nije i globalni, kao i veća vremenska zahtevnost procesa optimizacije. Moguće je i da neke metode nisu primenljive. Iako je vrlo poželjno, optimizacioni problemi u praksi, neretko nemaju ovo svojstvo.

Prisustvo ograničenja U opštem slučaju broj ograničenja M, može biti i 0. Takvi problemi su tipično lakši za rešavanje i moguće ih je rešavati

jednostavnijim metodama. Prisustvo ograničenja zahteva nešto komplikovanije i neretko sporije metode otpimizacije. Obe varijante se često javljaju u praksi.

U nastavku će prvo biti reči o problemima i metodama neprekidne lokalne optimizacije, a onda i o metodama globalne diskretne optimizacije.

5.1 Primeri praktičnih problema neprekidne matematičke optimizacije

Svi primeri diskutovani na početku glave o aproksimaciji funkcija predstavljaju probleme optimizacije i na njih su primenljive metode koje će biti opisane u nastavku. Ipak, svi razmatrani problemi, osim problema globalnog pozicioniranja, se mogu rešiti metodom najmanjih kvadrata. Problem globalnog pozicioniranja se može rešiti tom metodom uz aproksimaciju, dok bi za puno rešenje upravo bilo potrebno primeniti metode optimizacije.

Čak i u kontekstu problema najmanjih kvadrata, za koju je poznata procedura koja ga egzaktno rešava, često se zbog računske ili memorijske zahtevnosti, pri njegovom rešavanju pribegava primeni tehnika matematičke optimizacije, koje nalaze približno rešenje. Naime u slučaju da je broj kolona matrice X vrlo veliki, što današnjim primenama nije retkost, veličina matrice $X^T X$ može biti ogromna, a njeno smeštanje u memoriju problematično. Takođe, nekada optimizacione metode mogu brže da dođu do približnog rešenja koje je dovoljno dobro, čak i ako nije egzaktno. Otud, i problem najmanjih kvadrata se nekad rešava optimizacionim metodama.

Pored tih, već pomenutih primera, zanimljivo je razmotriti i nekoliko novih.

Primer 63 *Potrebno je odrediti intenzitete osvetljenja za m lampi fiksirane pozicije, maksimalnog intenziteta p_{max}, tako da adekvatno osvetljavaju prostoriju. Neka se osvetljenost meri na zidovima stana (pošto se od njih i odbija svetlost ka oku posmatrača). Neka je prostorija koju treba osvetlili konveksna. Ako je p_i intentzitet osvetljenja koju pruža lampa i, r_{ij} rastojanje od lampe i do tačke na zidu j u kojoj se osvetljenje meri, a θ_{ij} ugao pod kojim lampa i osvetljava zid j, intenzitet osvetljenja I_j na zidu j zavisi od navedenih parametara na sledeći način:*

$$I_j = \sum_{i=1}^{m} \frac{cos\theta_{ij}}{r_{ij}^2} p_i$$

U slučaju da prostorija nije konveksna, zavisnost od ugla osvetljenja bi bila komplikovanija. Neka je željeni nivo osvetljenja I. Kako ljudsko oko osvetljenje doživljava logaritamski, poželjno je da logaritmi proizvedenog i željenog osvetljenja budu što bliži na svim zidovima, odnosno da razlike $|\log I_j - \log I|$ budu što manje za svako j. Željeni intenziteti osvetljenja lampi su rešenja

narednog optimizacionog problema

$$\min_{p_1,\ldots,p_m,I_1,\ldots,I_n} \max_{j=1,\ldots,n} |\log \frac{I_j}{I}|$$
$$\textit{pri ogr. } I_j = \sum_{i=1}^{m} \frac{cos\theta_{ij}}{r_{ij}^2} p_i \qquad j = 1,\ldots,n$$
$$0 \le p_i \le p_{max} \qquad i = 1,\ldots,m$$

Ovaj problem je težak jer je nediferencijabilan i nekonveksan. Jedan način rešavanja bi bio pomoću aproksimacije problemom koji se može rešiti metodom najmanjih kvadrata:

$$\min_{p_1,\ldots,p_m,I_1,\ldots,I_n} \sum_{j=1}^{n} \left(\sum_{i=1}^{m} \frac{cos\theta_{ij}}{r_{ij}^2} p_i - I \right)^2$$

a potom modifikovanjem intenziteta p_i na 0, ukoliko važi $p_i < 0$ ili na p_{max}, ukoliko važi $p_i > p_{max}$. Jasno, ovakva aproksimacija je manjkava iz više razloga. Ne uzima se u obzir logaritamska zavisnost, umesto apsolutne vrednosti, koristi se kvadrat, umesto maksimuma suma, a rešenje ne mora poštovati ograničenja, već se naknadno modifikuje. Nešto bolja aproksimacija je:

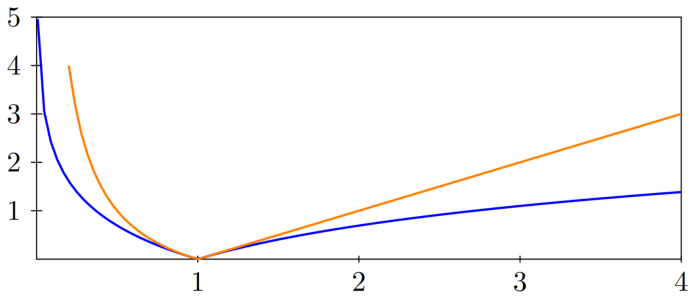
$$\min_{p_1,\ldots,p_m,I_1,\ldots,I_n} \max_{j=1,\ldots,n} |I_j - I|$$
$$\textit{pri ogr. } I_j = \sum_{i=1}^{m} \frac{cos\theta_{ij}}{r_{ij}^2} p_i \qquad j = 1,\ldots,n$$
$$0 \le p_i \le p_{max} \qquad i = 1,\ldots,m$$

Ovaj problem se može rešiti tehnikama linearnog programiranja*, o kojem će biti reči kasnije. Ipak, najbolji prustup je zamena polaznog problema konveksnim problemom koji ima isto rešenje:*

$$\min_{p_1,\ldots,p_m,I_1,\ldots,I_n} \max_{j=1,\ldots,n} h\left(\frac{I_j}{I}\right)$$
$$\textit{pri ogr. } I_j = \sum_{i=1}^{m} \frac{cos\theta_{ij}}{r_{ij}^2} p_i \qquad j = 1,\ldots,n$$
$$0 \le p_i \le p_{max} \qquad i = 1,\ldots,m$$

gde je

$$h(x) = \max\{x, 1/x\}$$



Slika 5.1: Funkcije $|\log \frac{I_j}{I}|$ i $h\left(\frac{I_j}{I}\right)$ dostižu minimum u istoj tački.

Funkcija h je konveksna jer su funkcije x i $1/x$ konveksne, a maksimum konveksnih funkcija je konveksan. Imajući u vidu poslednje svojstvo konveksnih funkcija, i cela ciljna funkcija je konveksna. Grafici funkcija $|\log \frac{I_j}{I}|$ i $h\left(\frac{I_j}{I}\right)$, prikazani na slici 5.1 ilustruju navedenu činjenicu da ove funkcije dostižu minimum u istoj tački, usled čega se rešavanjem poslednjeg problema dobija tačno rešenje polaznog problema. Naravno, ovakvo rešavanje problema, zamenom drugim problemom ili aproksimacijom koja se lakše rešava, uopšte nije trivijalno.

Primer 64 *Jedna od tipičnih primena matematičke optimizacije je optimizacija portfolija, odnosno ulaganja novca. Neka je potrebno uložiti svotu novca Q u akcije nekih od N različitih kompanija. Kako bi se investicija isplatila, potrebno je na kraju meseca ostvariti udeo dobiti bar q (npr. 5%). Kako su poslovi ulaganja rizični, potrebno je da taj rizik bude minimalan.*

Veličine od značaja u datom problemu su iznosi ulaganja x_i za svako preduzeće i, koji se mogu birati i povraćaji po jedinici novca r_i za svako od preduzeća, a koji predstavljaju slučajne promenljive. Kako povraćaj predstavlja slučajnu promenljivu, ovaj problem će biti razmatran aproksimativno – tako što će se umesto povraćaja koristiti njegovo očekivanje. Jasno, takav pristup ne daje garancije za udeo dobiti. Ograničenja koja je potrebno zadovoljiti su lako uočljiva, ali funkcija cilja nije. Zahtevi se odnose na određeni dobitak, koji se predstavlja ograničenjem i minimalnost rizika. Upravo rizik treba da bude ciljna funkcija. U svom radu za koji je nagrađen Nobelovom nagradom, Markovic je pokazao da se minimizacija rizika može dobro aproksimirati minimizacijom varijanse (disperzije) povraćaja ulaganja. U konkretnom slučaju, ta

varijansa je

$$\begin{aligned} Var\left(\sum_{i=1}^{N} r_i x_i\right) &= \mathbb{E}\left[\left(\sum_{i=1}^{N} r_i x_i - \mathbb{E}\left(\sum_{i=1}^{N} r_i x_i\right)\right)^2\right] \\ &= \mathbb{E}\left[\left(\sum_{i=1}^{N} r_i x_i - \left(\sum_{i=1}^{N} \overline{r}_i x_i\right)\right)^2\right] \\ &= \mathbb{E}\left[\left(\sum_{i=1}^{N} (r_i - \overline{r}_i) x_i\right)\left(\sum_{i=1}^{N} (r_i - \overline{r}_i) x_i\right)\right] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \mathbb{E}[(r_i - \overline{r}_i)(r_j - \overline{r}_j)] \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij} \end{aligned}$$

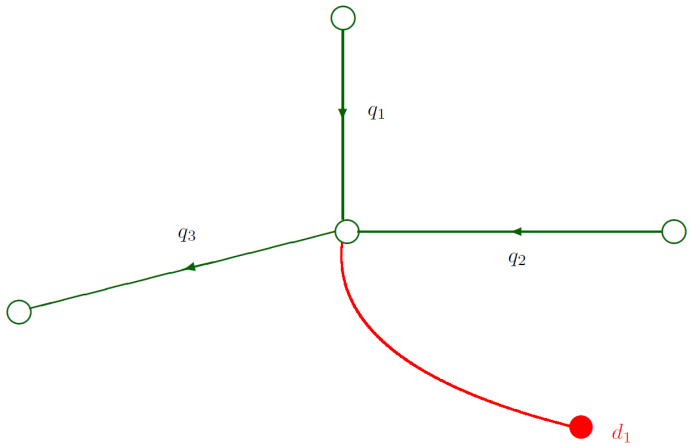
gde $\overline{r}_i$ predstavlja prosek promenljive r_i, a σ_{ij} kovarijaciju promenljivih r_i i r_j. Veličine σ_{ij} se mogu oceniti iz podataka o ranijim povraćajima ulaganja u date kompanije.

Onda je željeni optimizacioni problem:

$$\begin{aligned} \min_x \ & \sum_{i=1}^{N}\sum_{j=1}^{N} x_i x_j \sigma_{ij} \\ \textit{pri ogr.} \ & \sum_{i=1}^{N} x_i \le Q \\ & \sum_{i=1}^{N} \overline{r}_i x_i \ge q \\ & x_i \ge 0 \qquad i = 1, \ldots, N \end{aligned}$$

Ili, u matričnom obliku, ako e predstavlja kolonu jedinica, a Σ matricu kovarijacije:

$$\begin{aligned} \min_x \ & x^T \Sigma x \\ \textit{pri ogr.} \ & e^T x \le Q \\ & \overline{r}^T x \ge q \\ & x \ge 0 \end{aligned}$$



Slika 5.2: Količina gasa koja odlazi iz čvora, mora biti jednaka količini koja u njega dolazi, odnosno, mora važiti $q_1 + q_2 - q_3 = d_1$.

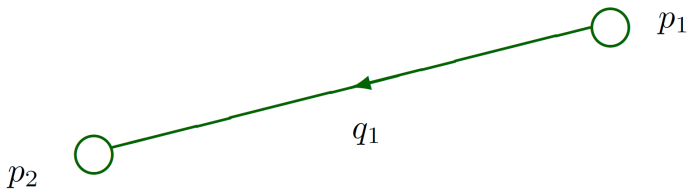
pri čemu poređenje vektora i skalara označava da svi elementi tog vektora moraju biti u datom odnosu sa datim skalarom.

Primer 65 *U Velikoj Britaniji se gas crpi iz Severnog mora i transportuje mrežom cevi do različitih gradova. Gasovod uključuje 6600km cevi, 26 strateški raspoređenih kompresora koji održavaju pritisak u gasovodu i održavaju protok gasa, kao i 140 izlaznih tačaka, koje predstavljaju industrijske komplekse, elektrane i rezidencijalne oblasti. U gasovodu je nekada potrebno maksimizovati pritisak, a nekad ga minimizovati ili minimizovati trošak rada kompresora, u zavisnosti od potreba. Nekada je potrebno ostvariti više ciljeva odjednom. U nastavku se pretpostavlja da je potrebno minimizovati rad kompresora, kako bi se smanjio trošak rada gasovoda, pri određenim ograničenjima normlanog rada gasovoda.*

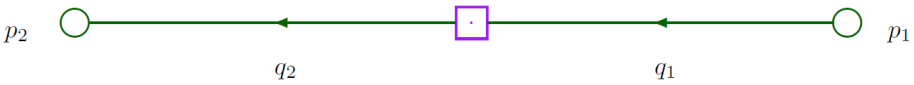
Gasovod se može modelovati u vidu grafa, čiji su čvorovi spojevi cevi, a grane same cevi. Neka je n broj čvorova, a m broj cevi u gasovodu. Matrica A, dimenzija $n \times m$ predstavlja gasovod i sastoji se od redova koji odgovaraju čvorovima, tako što red sadrži 1 na mestu cevi iz koje gas dolazi u čvor, -1 na mestu cevi u koju gas odlazi iz datog čvora, a 0 ako data cev nije povezana na dati čvor. Pritisak u čvoru gasovoda i se označava sa p_i, protok po jedinici vremena duž cevi j, sa q_j, a potražnja po jedinici vremena u čvoru i sa d_i. Ograničenja snabdevanja koja garantuju da će u svakoj od izlaznih tačaka biti onoliko gasa kolika je potražnja su:

$$Aq = d$$

Ova ograničenja su ilustrovana slikom 5.2. Prilikom transporta gasa, gubi se na pritisku. Empirijski je utvrđeno da je pad u kvadratu pritiska između dva



Slika 5.3: Zbog pada pritiska, različiti krajevi cevi moraju zadovoljavati ograničenje $p_1^2 - p_2^2 = k_1 q_1^{2.8359}$.



Slika 5.4: Zbog prisustva kompresora, protoci kroz dve cevi povezane na njega zadovoljavaju ograničenje $q_2 = q_1 + z_1$.

kraja cevi proporcionalan 2.8359*-om stepenu protoka duž cevi, što se matrično izražava kao:*

$$A^T p^2 = K q^{2.8359}$$

gde je K dijagonalna matrica svojstava cevi, ustanovljenih empirijskim merenjima, a koja predstavljaju faktor proporcionalnosti. Ova ograničenja su ilustrovana slikom 5.3. Kompresori se nalaze u nekima od čvorova i u slučaju da su uključeni povećavaju pritisak, a u suprotnom se ponašaju kao standardni čvorovi. Pritom, uvek imaju tačno jednu ulaznu i jednu izlaznu cev. Neka je A' matrica čiji je svaki red jednak redu matrice A ukoliko odgovarajući čvor predstavlja kompresor, a jednak nula vektoru u suprotnom. Takođe, neka je z vektor koji sadrži promenljive na mestima koja odgovaraju čvorovima sa kompresorima, a nule na ostalim. Onda se ograničenja vezana za rad kompresora mogu zapisati kao

$$A'^T q + z = 0$$

Ova ograničenja su ilustrovana slikom 5.4. Za svaki čvor i svaku cev, postoje priodna ograničenja opterećenja koje oni mogu da izdrže, a za kompresore povećanje u protoku koje mogu da proizvedu:

$$p_{min} \leq p \leq p_{max}$$

$$q_{min} \leq q \leq q_{max}$$

$$0 \leq z \leq z_{max}$$

Pritisak u tačkama snabdevanja gasovoda se može zadati tako što se za odgovarajuće koordinate vektora p, koordinate vektora p_{min} i p_{max} postave na tu istu vrednost.

Zanimljivo je primetiti da su sve matrice izrazito retke. Ukupan optimizacioni problem je

$$\begin{aligned} \min_{z} \; & e^T z \\ \textit{pri ogr.} \; & Aq - d = 0 \\ & A^T p^2 - K q^{2.8359} = 0 \\ & A'^T q + z = 0 \\ & p_{min} \leq p \leq p_{max} \\ & q_{min} \leq q \leq q_{max} \\ & 0 \leq z \leq z_{max} \end{aligned}$$

5.2 Neprekidna optimizacija

U ovom delu će biti diskutovane metode neprekidne optimzacije. Prvo će biti diskutovani uslovi optimalnosti, odnosno neophodni i dovoljni uslovi da neka tačka bude optimum funkcije, a potom metode optimizacije prvog i drugog reda, kao i prisustvo ograničenja.

5.2.1 Uslovi optimalnosti

Uslovi optimalnosti se odnose na teorijske uslove pod kojim tačka predstavlja minimum ili maksimum funkcije. U nastavku se pretpostavlja da su funkcije koje se razmatraju diferencijabilne. *Gradijent* funkcije u tački x je vektor parcijalnih izvoda funkcije u toj tački:

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n} \right)$$

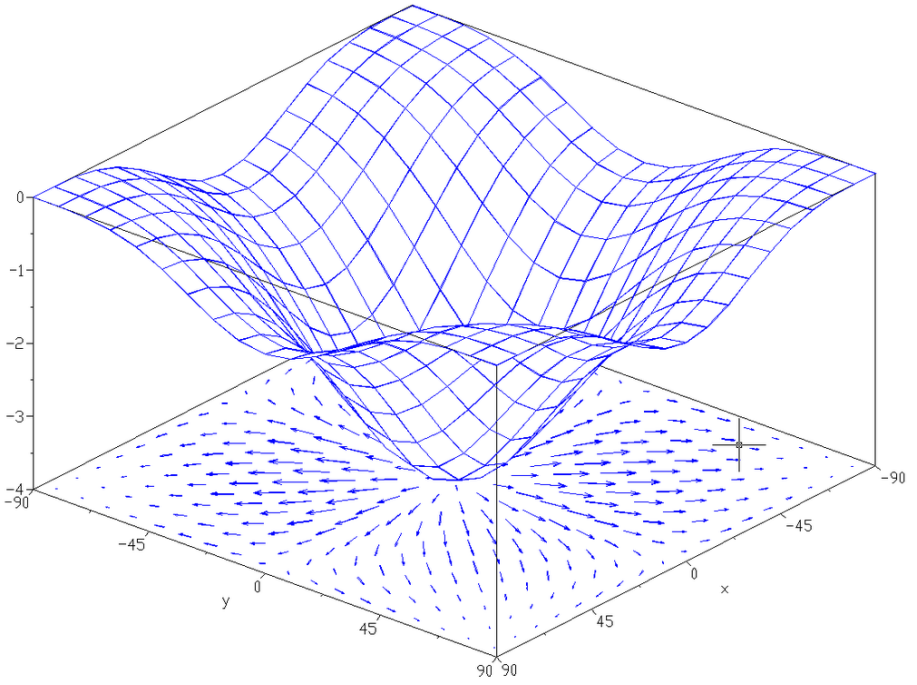
Gradijent predstavlja vektor pravca u kojem funkcija najbrže raste u toj tački. Gradijent funkcije u različitim tačkama, ilustrovan je slikom 5.5. Gradijent funkcije f u nekoj tački u kojoj funkcija ima vrednost y normalan je na površ koja prolazi kroz tu tačku i definisana je uslovom $f(x) = y$. Odnosno, gradijenti su normalni na konture iste vrednosti funkcije (nalik izohipsama na geografskim kartama).

Opšte je poznato da je neophodan uslov optimalnosti neke tačke da je gradijent u toj tački jednak nuli, odnosno ako je tačka x^* optimum, važi:

$$\nabla f(x^*) = 0$$

Intuicija iza ovog rezultata je da je tangentna površ

$$\{(x, f(x^*) + \nabla f(x^*)(x - x^*)) | x \in \mathbb{R}^n\}$$



Slika 5.5: Strelice u ravni argumenata funkcije predstavljaju gradijente funkcije u različitim tačkama.

u optimumu horizontalna. Analitičko nalaženje optimuma počiva na rešavanju ovog sistema jednačina. Naravno, u većini praktičnih situacija te jednačine nije moguće analitički rešiti. Čak i kad je nađeno rešenje, to ne znači da se radi o optimumu pošto je prethodni uslov neophodan uslov optimalnosti, ali ne i dovoljan. Naravno, problem predstavljaju sedlene tačke. Sve tačke u kojima važi $\nabla f(x^*) = 0$ nazivaju se *stacionarnim*. U slučaju dva puta diferencijabilnih funkcija, matrica drugih parcijalnih izvoda

$$\nabla^2 f(x) = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right]_{i,j=1,\ldots,n}$$

se naziva *hesijan*. Da bi stacionarna tačka zaista bila optimum, hesijan u datoj tački mora biti pozitivno ili negativno definitna matrica, odnosno mora važiti

$$h^T \nabla^2 f(x^*) h > 0$$

za svako $h \neq 0$. Ovaj uslov ima jednostavno objašnjenje. U dovoljno maloj okolini tačke x^*, funkcija f se može dobro aproksimirati svojom kvadratnom aproksimacijom oblika

$$f(x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^*)(x - x^*)$$

(lienarnog dela nema, jer je gradijent u optimumu jednak nuli). Da bi tačka x^* bila optimum, potrebno je da se bar u maloj okolini sa odaljavanjem od nje, vrednost funkcije uvećava. Ukoliko je hesijan pozitivno definitan, to je za prethodni izraz garantovano.

Iako navedeni uslovi u praksi često nisu direktno upotrebljivi, korisni su kako u svrhe matematičke analize mnogih problema i metoda, tako i u svrhe konstrukcije metoda optimizacije. Naime, neke od metoda koje će biti opisane, mogu se videti kao metode pronalaženja tačaka u kojima su uslovi optimalnosti zadovoljeni.

U slučaju optimizacionih problema sa ograničenjima (pretpostavljajući da su sve funkcije funkcije realnih promenljivih), postoje nešto komplikovaniji, ali slični uslovi optimalnosti, poznati kao Karuš-Kun-Takerovi (KKT) uslovi. Neka je dat problem

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) \\ \text{pri ogr.}\quad &g_i(x) \leq 0 \qquad i = 1,\ldots,M \\ &h_j(x) = 0 \qquad j = 1,\ldots,L \end{aligned}$$

Iako nije neophodno, jednakosna ograničenja su posebno istaknuta. Neka je x^* optimalno rešenje i neka su sve funkcije diferencijabilne u x^*. Ukoliko problem zadovoljava još neke *uslove regularnosti* koje diskutujemo niže, postoje konstante μ_i^* i λ_j^* takve da važi:

$$g_i(x^*) \leq 0$$

$$h_j(x^*) = 0$$

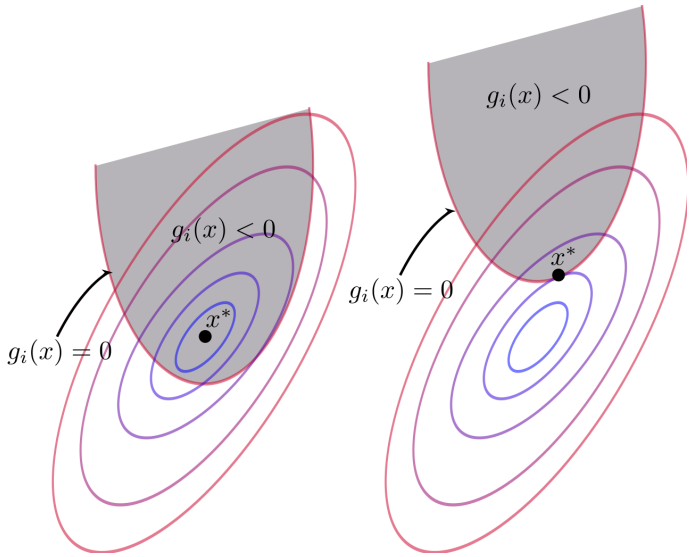
$$-\nabla f(x^*) = \sum_{i=1}^{M} \mu_i^* \nabla g_i(x^*) + \sum_{j=1}^{L} \lambda_j^* \nabla h_j(x^*)$$

$$\mu_i^* \geq 0$$

$$\mu_i^* g_i(x^*) = 0$$

Za ograničenja g_i za koja važi $g_i(x^*) = 0$ kažemo da su *aktivna*.

Probajmo da interpretiramo ove uslove. Ograničenja $g_i(x^*) \leq 0$ i $h_j(x^*) = 0$ govore samo da je optimalno rešenje dopustivo rešenje, što je zahtevano postavkom problema. Uslov $\mu_i^* g_i(x^*) = 0$ je malo sofisticiraniji, ali i dalje jednostavan. Naime, vrednost $g_i(x^*)$ može biti ili 0 ili strogo negativna. Ukoliko je nula, ovaj uslov je ispunjen. Ukoliko je strogo negativna, to znači da ograničenje $g_i(x^*) \leq 0$ ne utiče na rešenje. Naime, razmišljajmo o ograničenjima kao o preprekama koje onemogućavaju prilazak minimumu funkcije f. Zamislimo da površ $g_i(x) = 0$ predstavlja zid. Oblast $g_i(x) \leq 0$ predstavlja stranu zida sa koje smemo da se krećemo. Ukoliko $g_i(x)$ nije jednako 0, to znači da krećući se prema optimumu nismo ni imali potrebu da dođemo do zida, što znači da za



Slika 5.6: Ilustracija uticaja nejednakosnog ograničenja na rešenje optimizacionog poblema. Ukoliko ograničenje nije aktivno (levo), njegovo isključivanje iz problema ne menja rešenje. Očigledno, isključivanje aktivnog ograničenja (desno) bi promenilo optimalno rešenje.

naše potrebe taj zid kao da ni ne postoji, odnosno kao da ograničenja $g_i(x) \leq 0$ ni nema. Otud mu odgovara koeficijent $\mu_i^* = 0$, čime se ovo ograničenje u uslovima optimalnosti efektivno zanemaruje. Zaključujemo da važi $\mu_i^* g_i(x^*) = 0$, odnosno da su bitna samo ograničenja koja su aktivna u optimumu. Ovo je ilustrovano slikom 5.6.

Uslovi

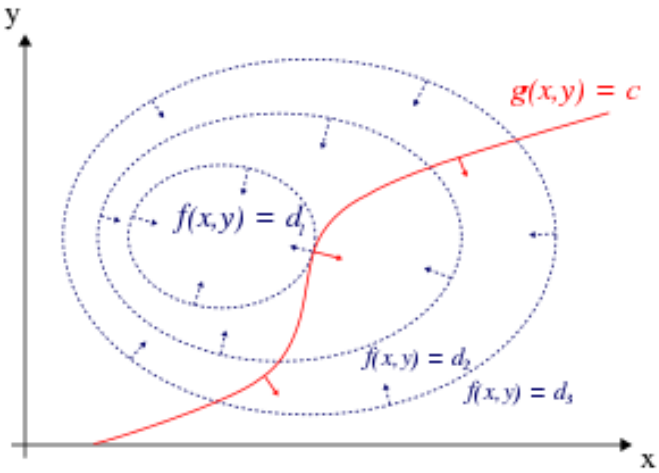
$$-\nabla f(x^*) = \sum_{i=1}^{M} \mu_i^* \nabla g_i(x^*) + \sum_{j=1}^{L} \lambda_j^* \nabla h_j(x^*)$$

$$\mu_i^* \geq 0$$

su suštinski. Razmotrimo prvo slučaj kad važi $M = 0$ i $L = 1$. U tom slučaju mora važiti

$$-\nabla f(x^*) = \lambda \nabla h(x^*)$$

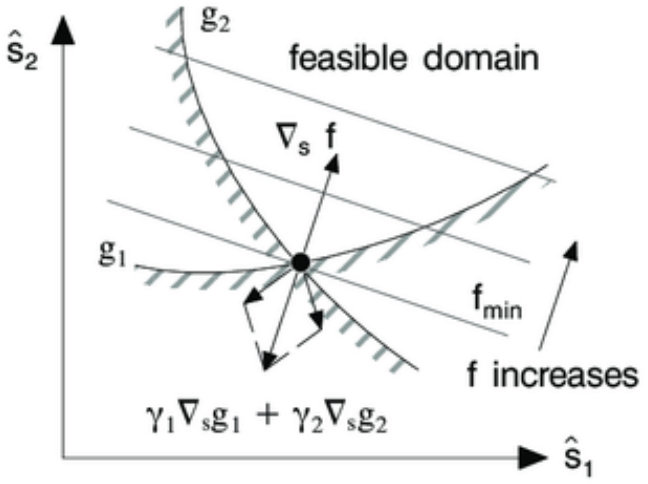
Ovo je ilustrovano slikom 5.7. Zašto je to tako? Zamislimo da je minimum funkcije centar privlačne sile koja deluje na objekat koji se može kretati samo po oblasti $h(x) = 0$. Negativni gradijent funkcije daje pravac najbržeg kretanja ka minimumu, odnosno pravac u kojem deluje sila. Ukoliko se objekat nalazi u tački u kojoj normala na oblast $h(x) = 0$ (odnosno njen gradijent) nije kolinearna sa pravcem negativnog gradijenta, kretanjem u pravcu projekcije negativnog gradijenta na tangentu oblasti $h(x) = 0$ u datoj tački, može se



Slika 5.7: Kolinearnost negativnog gradijenata funkcije i gradijenta ograničenja u optimalnoj tački u slučaju samo jednog jednakosnog uslova.

doći do tačke sa manjom vrednošću funkcije f. Stoga tačka u kojoj se objekat nalazi u tom slučaju nije minimum. Otud je jasna neophodnost ovog uslova. Da sumiramo, u slučaju jednog jednakosnog ograničenja, zaključak je da negativni gradijent mora biti normalan na površ definisanu ograničenjem. Kako je već ukazano da su od nejednakosnih ograničenja bitna samo ona koja su aktivna i stoga se ponašaju isto kao jednakosna, isto važi i za slučaj jednog nejednakosnog ograničenja. Pitanje je šta u slučaju kada postoji više ograničenja.

Razmotrimo sada slučaj više aktivnih ograničenja, ilustrovan na slici 5.8. Usled presecanja više oblasti koje odgovaraju aktivnim ograničenjima, u opštem slučaju, ne može se definisati normala na oblast dopustivih rešenja u optimumu. Ipak, pojedinačna ograničenja imaju svoje normale, odnosno gradijente, u toj tački. Gradijenti nejednakosnih ograničenja definišu pravac rasta tih ograničenja. Kako je unutrašnjost skupa dopustivih rešenja definisana negativnim vrednostima ograničenja, to znači da su oni usmereni van skupa dopustivih rešenja, kao i negativni gradijent funkcije. Štaviše, negativni gradijent mora biti linearna kombinacija gradijenata ograničenja. Iz prethodnog razmatranja vezano za smer gradijenata, zaključuje se da koeficijenti linearne kombinacije koji odgovaraju nejednakosnim uslovima moraju biti nenegativni, odnosno važe dva diskutovana uslova. Ukoliko gradijent ne bi bio takva linearna kombinacija, argument iz slučaja jednog ograničenja bi se mogao uopštiti kako bi se našla tačka sa manjom vrednošću funkcije, što bi bila kontradikcija sa time da razmatramo situaciju u optimalnoj tački. Ipak, pitanje je zašto se ovaj uslov odnosi samo na koeficijente nejednakosnih ograničenja. Zašto ne i na koeficijente jednakosnih ograničenja? Jednakosno ograničenje $h_j(x) = 0$ se slobodno može zameniti ograničenjem $-h_j(x) = 0$. Kako se znak funkcije može



Slika 5.8: Negativni gradijent funkcije cilja je linearna kombinacija gradijenata ograničenja.

slobodno promeniti, znak odgovarajućeg koeficijenta ne može biti unapred poznat.

Primetimo da se prvi uslov može posmatrati kao tvrdnja da je (x^*, μ^*, λ^*) stacionarna tačka funkcije

$$L(x, \mu, \lambda) = f(x^*) + \sum_{i=1}^{M} \mu_i g_i(x) + \sum_{j=1}^{L} \lambda_j h_j(x)$$

koju nazivamo *lagranžijan*.

Što se tiče uslova regularnosti pod kojima prethodna tvrdnja važi, ima ih raznih, uključujući sledeće:

- Sva ograničenja su afine funkcije.
- Gradijenti aktivnih ograničenja i jednakosnih ograničenja u tački x^* su linearno nezavisni.
- Sve funkcije u problemu su konveksne i postoji tačka x takva da je $h_j(x) = 0$ za sve j i $g_i(x) < 0$ za sve i.

Naravno, kao i pre, stacionarnost nije dovoljna. Dovoljan uslov, koji se opet može definisati u terminima hesijana, je

$$h^T \nabla^2_{xx} L(x^*, \lambda^*, \mu^*) h > 0$$

za svako h koje zadovoljava uslov $h^T \nabla g_i(x) = 0$ za svako aktivno nejednakosno ograničenje g_i i gde ∇^2_{xx} označava da se parcijalni izvodi računaju samo po promenljivim x.

Primer 66 *Nekada KKT uslovi mogu dati i praktične i teorijske rezultate. Razmotrimo jedan problem ekonomije. Neka je q proizvedena količina proizvoda, $R(q)$ prihod od prodaje količine q proizvoda, $C(q)$ cena proizvodnje količine q proizvoda. Profit je onda $R(q) - C(q)$. Ukoliko važi $R'(q) > C'(q)$ pri povećanju q prihod raste brže od troška, pa se isplati povećati proizvodnju. Ukoliko važi $R'(q) < C'(q)$, pri smanjenju q trošak pada brže od profita, pa se isplati smanjiti proizvodnju. Očito, ima smisla povećavati proizvodnju dok god je $R'(q) > C'(q)$ i stati kad to više ne važi, odnosno kad važi $R'(q) = C'(q)$. Imajmo u vidu da u praksi takva tačka postoji, pošto beskonačan profit nije realističan. Na primer, za dovoljnu količinu proizvoda, zafaliće kupaca. Ukratko, za firmu koja želi da maksimizuje profit, optimalna strategija je naći vrednost q za koju važi $R'(q) = C'(q)$.*

Razmotrimo rad firme koja maksimizuje prihod od prodaje, pri zadatom uslovu minimalnog profita $m > 0$ koji se mora ostvariti. Primetimo, firma ne pokušava da maksimizuje profit, već promet. Ovako nešto je poželjno ukoliko firma želi da poveća svoju zastupljenost na tržištu ili da se izbori sa konkurencijom, a da ne padne ispod neke granice isplativosti rada. Neka je R nenegativna funkcija za koju važi $R(q) = 0$, neka je C nenegativna funkcija i neka su, očekivano, obe funkcije strogo rastuće. Pitanje je koja je optimalna količina koju treba proizvesti. Primetimo da ovaj problem ima smisla samo ukoliko funkcija C za neku vrednost q i nadalje prestiže funkciju R. Ova pretpostavka ima smisla u realnosti. Recimo, ukoliko se proizvodi više proizvoda nego što potrošačima treba, cena proizvodnje raste brže od prihoda. Problem se može postaviti kao

$$\begin{gathered} \min_q -R(q) \\ \text{pri ogr. } R(q) - C(q) \geq m \\ q \geq 0 \end{gathered}$$

KKT uslovi su:

$$\begin{gathered} C(q) - R(q) + m \leq 0 \\ -q \leq 0 \\ R'(q) = \mu_1(C'(q) - R'(q)) - \mu_2 \\ \mu_1, \mu_2 \geq 0 \\ \mu_1(C(q) - R(q) + m) = 0 \\ -\mu_2 q = 0 \end{gathered}$$

Ukoliko bi važilo $q = 0$, uslov $R(q) - C(q) \geq m$ bi bio narušen. Otud mora važiti $\mu_2 = 0$, pa se može izvesti da važi

$$R'(q) = \frac{\mu_1}{\mu_1 + 1} C'(q)$$

Dalje rešavanje zavisi od konkretnih funkcija R i C. Dodatno, primetimo da kako su R' i C' strogo pozitivni, i μ_1 mora biti strogo pozitivan, pa otud važi $R(q) - C(q) = m$.

Zanimljiv teorijski uvid je i da je izvedeni uslov različit od slučaja firme koja pokušava da maksimizuje profit, jer iako je $R'(q) < C'(q)$ u izabranoj tački, ne teži se smanjenju proizvodnje dok se ne ostvari uslov $R'(q) = C'(q)$. Naravno, neka suštinska razlika između optimalnih režima rada dve diskutovane vrste firmi se mogla i očekivati.

5.2.2 Metode lokalne optimizacije prvog reda bez ograničenja

Pod metodama optimizacije prvog reda podrazumevaju se metode koje kao jedine informacije o funkciji koriste njene vrednosti i vrednosti njenog gradijenta u proizvoljnim tačkama. Pored pojma gradijenta, za razumevanje osnovnih metoda prvog reda, potrebno je uvesti još nekoliko matematičkih pojmova. Neka je $X \subseteq \mathbb{R}^n$. Funkcija $f : X \to \mathbb{R}$ je *Lipšic neprekidna*, ukoliko postoji konstanta L, takva da za sve $x, y \in X$ važi

$$|f(x) - f(y)| \leq L\|x - y\|$$

Ovo svojstvo je jače od svojstva obične, pa i ravnomerne neprekidnosti. Diferencijabilna funkcija $f : X \to \mathbb{R}$ je *konveksna*, ako za svako $x, y \in X$ važi:

$$f(x) \geq f(y) + \nabla f(y)^T (x - y)$$

Odnosno, ukoliko je površ $(x, f(x))$ koja predstavlja grafik funkcije f u svakoj tački iznad tangente u toj tački. Ukoliko važi stroga nejednakost, funkcija je *strogo konveksna*. Naravno, konveksnost se može definisati i za nediferencijabilne funkcije, ali je u kontekstu dalje diskusije diferencijabilnost standardna pretpostavka. Funkcija f je *konkavna* ukoliko je funkcija $-f$ konveksna. Funkcija f je *jako konveksna*, ukoliko postoji konstanta $m > 0$ i za svako $x, y \in X$ važi:

$$f(x) \geq f(y) + \nabla f(y)^T (x - y) + \frac{m}{2}\|x - y\|^2$$

Neformalno se kaže da je jako konveksna funkcija konveksna bar koliko kvadratna funkcija. Razmatranjem funkcije $f(x) = x^\alpha$ u kontekstu gornjeg svojstva, lako se uočava da je najmanje α za koje to svojstvo važi $\alpha = 2$, otud i pomenuta kvalifikacija. Za diferencijabilnu funkciju, svojstvo konveksnosti implicira da je hesijan $\nabla^2 f(x)$ pozitivno semidefinitna matrica, a svojstvo jake konveksnosti, da je $\nabla^2 f(x) - mI$ pozitivno semidefinitna matrica.

Konveksne funkcije imaju naredna svojstva, koja često mogu olakšati prepoznavnaje da je neka funkcija konveksna:

- Ako su $f_1, \ldots, f_m$ konveksne funkcije i važi $w_1 \geq 0, \ldots, w_m \geq 0$, onda je i funkcija
$$w_1 f_1(x) + \ldots, w_m f_m(x)$$

konveksna funkcija.

- Ako je f konveksna funkcija, A matrica i b vektor odgovarajućih dimenzija, onda je i $f(Ax+b)$ konveksna funkcija.
- Ako su $f_1, \ldots, f_m$ konveksne funkcije, onda je i funkcija

$$\max\{f_1(x), \ldots, f_m(x)\}$$

konveksna funkcija. Isto važi i za supremum nad beskonačnim skupom konveksnih funkcija.

- Kompozicija $f \circ g$ je konveksna ako je funkcija f konveksna i neopadajuća po svim argumentima, a funkcija g konveksna ili ako je funkcija f konveksna i nerastuća po svim argumentima, a g konkavna.

Primer 67 *Funkcija greške u problemu linearne regresije* $\|Xw-y\|_2^2$ *je konveksna. Naime,*

$$(Xw-y)^T(Xw-y) = w^TX^TXw - w^TX^Ty - y^TXw + y^Ty$$

Pozitivna semidefinitnost matrica oblika X^TX *je već pokazana u diskusiji Čoleski dekompozicije. Pošto je matrica* X^TX *hesijan prethodne funkcije, onda je ona konveksna funkcija.*

Funkcija greške u problemu regularizovane (grebene) linearne regresije $\|Xw-y\|_2^2 + \lambda\|w\|_2^2$ *je jako konveksna. Naime,*

$$(Xw-y)^T(Xw-y) + \lambda w^Tw = w^TX^TXw - w^TX^Ty - y^TXw + y^Ty + w^Tw$$

Hesijan ove funkcije je

$$X^TX + \lambda I$$

Za $m = \lambda$*, dobija se*

$$X^TX + \lambda I - \lambda I = X^TX$$

što je pozitivno semidefinitna matrica. Otud polazna funkcija mora biti jako konveksna.

Najjednostavnija i najpoznatija metoda optimizacije prvog reda za diferencijabilne funkcije je *gradijentni spust* (eng. gradient descent). Ova metoda, kao i većina metoda optimizacije, zasniva se na postepenom, iterativnom, približavanju rešenju problema. Kako gradijent ukazuje na pravac najbržeg uspona, negativna vrednost gradijenta ukazuje na pravac najbržeg spusta. Osnovna ideja gradijentnog spusta je da se, polazeći od neke nasumice izabrane tečke, nizom koraka u pravcu gradijenta dođe vrlo blizu rešenju. Ako je polazna tačka x_0, svaka naredna se dobija primenom pravila

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

U vezi sa ovakvim pristupom, postavlja se više pitanja. Prvo je kako se bira dužina koraka α_k koji se preduzima u pravcu suprotnom gradijentu. Postoje različiti pristupi. Jedan, jednostavan izbor je korišćenje konstantne vrednosti koraka $\alpha_k = \alpha$, za neko α, za svako i. Drugi pristup je oslanjanje na Robins-Monroove uslove

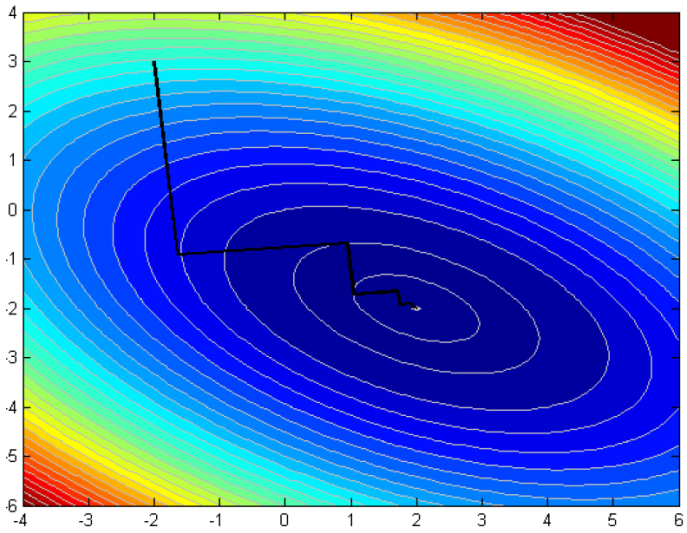
$$\sum_{i=0}^{\infty} \alpha_k = \infty \qquad \sum_{i=1}^{\infty} \alpha_k^2 < \infty$$

Intuitivno, smisao prvog uslova je da su koraci dovoljno veliki da se može dostići rešenje problema. Smisao drugog uslova je da su koraci dovoljno mali da niz tačaka x_k konvergira rešenju, umesto da osciluje. Jedan od izbora koji zadovoljava ove uslove je $\alpha_k = \frac{1}{k}$. Pored ovih pristupa, postoje i drugi. Drugo pitanje je kada se staje sa izračunavanjem. Kriterijuma zaustavljanja koji se u praksi koriste ima više. Najčešći su zaustavljanje nakon unapred zadatog broja iteracija, nakon što razlika između susednih koraka $\|x_{k+1} - x_k\|$ postane manja od unapred zadate vrednosti ε, nakon što razlika između vrednosti funkcije u susednim koracima $|f(x_{k+1}) - f(x_k)|$ postane manja od ε ili nakon što ova razlika u odnosu na polaznu vrednost funkcije $|f(x_{k+1} - f(x_k)|/|f(x_0)|$ postane manja od ε. Moguće je kombinovati i više ovakvih kriterijuma.

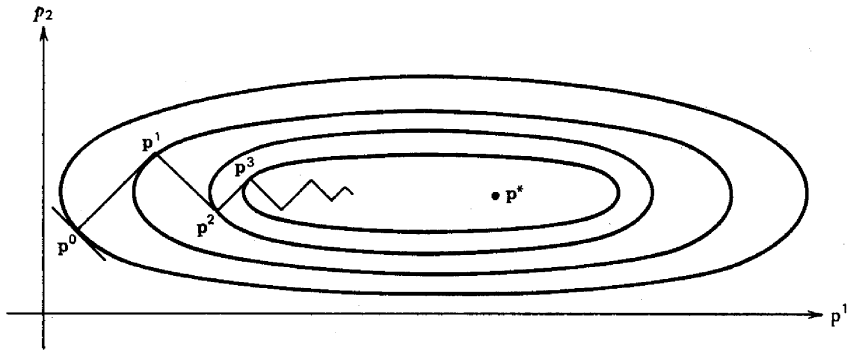
U slučaju konstantnog koraka, moguće je dokazati konvergenciju metoda ka okolini pravog rešenja, ali sa određenom nesavladivom greškom, koja je utoliko veća ukoliko je veličina koraka veća. U slučaju oslanjanja na Robins-Monroove uslove, za konveksne funkcije sa Lipšic neprekidnim gradijentom, greška metode $\|x_k - x^*\|$ u koraku k, gde je x^* tačka minimuma, je reda $O\left(\frac{1}{k}\right)$, što očito implicira konvergenciju. Za jako konveksne funkcije sa Lipšic neprekidnim gradijentom, greška je reda $O\left(c^k\right)$ za neko $0 < c < 1$. Očito, u slučaju rešavanja problema najmanjih kvadrata, brzina konvergencije je eksponencijalna. U slučaju nekonveksnih funkcija, gradijentni spust i njegove varijante prikazane u nastavku konvergiraju, ali navedene brzine konvergencije ne važe. Konvergencija gradijentnog spusta se smatra relativno sporom.

Poznato je da je gradijent u svakoj tački normalan na konturu (poput izohipse na geografskoj karti) funkcije sa istom vrednošću koju funkcija ima u toj tački. Ovo ponašanje je ilustrovano slikom 5.9. Imajući ovo u vidu, ne čudi da se gradijentni spust ne ponaša dobro u slučajevima funkcija čije su konture izdužene, kao na slici 5.10. U takvim situacijama, gradijentni spust bira tačke koje leže duž cik-cak putanje ka minimumu i broj koraka do zadovoljavajućeg rešenja može biti veliki. Očito, pravac najbržeg uspona uopšte ne mora biti pravac najbržeg kretanja ka minimumu.

U sumi, prednosti metode gradijentnog spusta su njena jednostavnost i široki uslovi primenljivosti, a mane su spora konvergencija, to što je izabrani pravac samo lokalno optimalan, što dodatno usporava konvergenciju cik-cak kretanjem i to što se u mnogim slučajevima za izračunavanje tog neoptimalnog pravca troši puno vremena, na primer u metodama mašinskog učenja, poput neuronskih mreža, ali i drugim.



Slika 5.9: Pravac gradijenta u nekoj tački je normalan na odgovarajuću konturu funkcije.



Slika 5.10: Ponašanje gradijentnog spusta u slučaju funkcije sa izduženim konturama.

Jedna, vrlo široko primenjena, modifikacija gradijentnog spusta je *stohastički gradijentni spust* (eng. stochastic gradient descent). Intenzivno se primenjuje u treniranju modela mašinskog učenja sa velikim količinama podataka, ali i u drugim problemima. Modifikacija se sastoji u tome da je umesto gradijenta dovoljno koristiti neki slučajni vektor čije je očekivanje kolinearno sa gradijentom i istog je smera. Ovakva modifikacija ima smisla pre svega kada se funkcija koja se optimizuje može predstaviti kao prosek drugih funkcija koje

se lakše izračunavaju:

$$f(x) = \frac{1}{n} \sum_{i=1}^{N} f_i(x)$$

Ovo je tipičan slučaj u kontekstu mašinskog učenja, gde se minimizuje funkcija greške koja je zbir grešaka na pojedinačnim instancama. Tada je pravilo izračunavanja novog koraka moguće zameniti sledećim pravilom:

$$x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$$

Jasno, kako je funkcija f, prosek funkcija f_i, ako se i bira u skladu sa uniformnom raspodelom, očekivanje slučajnog vektora $\nabla f_i(x)$ je baš $\nabla f(x)$. Obično se i bira tako da bude jednako $(k \ mod \ N) + 1$, odnosno tako da se u svakom koraku koristi naredna funkcija f_i dok se ne dođe do poslednje, a onda se ponovo nastavlja od prve. Ovaj pristup predstavlja jeftinu aproksimaciju gradijenta. Ipak, ona može biti prilično neprecizna, pa se kao kompromis često, umesto samo jedne od funkcija f_i, koristi prosek nekog podskupa ovih funkcija (eng. minibatch). Ovo je praktično uvek pristup koji se koristi u treniranju neuronskih mreža.

Brzina konvergencije stohastičkog gradijentnog spusta merena *u broju iteracija* je dosta manja nego kod običnog gradijentnog spusta. U slučaju konveksnih funkcija sa Lipšic neprekidnim gradijentom, greška je reda $O\left(\frac{1}{\sqrt{k}}\right)$, a u slučaju jako konveksnih funkcija sa Lipšic neprekidnim gradijentom, greška je reda $O\left(\frac{1}{k}\right)$. Uprkos ovome, u mašinskom učenju, u kojem se danas često koriste ogromne količine podataka, vreme jedne iteracije gradijentnog spusta, koji u svakoj iteraciji koristi sve podatke, je drastično veće nego u slučaju stohastičkog gradijentnog spusta, koji u svakoj iteraciji koristi samo po jednu instancu iz skupa podataka.

U odnosu na gradijentni spust, prednosti stohastičkog gradijentnog spusta su mnogostruke. Gradijent, koji inače može biti skup za izračunavanje, jeftino se aproksimira. U kontekstu metoda mašinskog učenja nad velikim količinama podataka, to često vodi bržoj konvergenciji. Greška aproksimacije gradijenta može poslužiti i kao vid regularizacije kod metoda mašinskog učenja, pošto sprečava preciznu konvergenciju ka minimumu, što u slučaju vrlo fleksibilnih modela ili male količine podataka može voditi ka preteranom prilagođavanju modela podacima za obučavanje. Manje je podložan problemu redundantnosti podataka prilikom treniranja modela mašinskog učenja. Pod redundantnošću se podrazumeva ponavljanje istih ili sličnih instanci u skupu podataka. Poslednja poenta zahteva opširnije obrazloženje, koje je dato narednim primerom. Mana je očito veći broj iteracija do konvergencije, što u slučaju da korak gradijentnog spusta nije vremenski skup, vodi sporijem zaustavljanju nego u slučaju gradijentnog spusta.

Primer 68 *Neka se skup podataka sastoji od instanci* $\{(x_1, y_1), \ldots, (x_N, y_N)\}$. *Neka su* w *parametri modela mašinskog učenja* $f(x, w)$, *koji se određuju optimizacijom. U kontekstu srednjekvadratne greške funkcija koja se minimizuje*

je

$$R(w) = \frac{1}{n}\sum_{i=1}^{N}(f(x_i, w) - y_i)^2$$

Onda je pravilo ažuriranja koeficijenata u skladu sa stohastičkim gradijentnim spustom:

$$w_{k+1} = w_k - 2\alpha_k(f(x_i, w_k) - y_i)\nabla f(x_i, w_k)$$

dok je u slučaju gradijentnog spusta to

$$w_{k+1} = w_k - 2\alpha_k\frac{1}{n}\sum_{i=1}^{N}(f(x_i, w_k) - y_i)\nabla f(x_i, w_k)$$

Ukoliko se ceo skup podataka uveća tako što ponovi za redom M puta u poretku

$$\underbrace{(x_1, y_1), \ldots, (x_N, y_N), \ldots, (x_1, y_1), \ldots, (x_N, y_N)}_{M \times N}$$

pravac koraka u gradijentnom spustu se neće promeniti ni u jednom koraku, pa je stoga i broj koraka u primeni algoritma isti. Kako svaki korak zahteva M puta više vremena, ceo proces M puta duže traje. Stohastički gradijentni spust u ovom slučaju ne zahteva ništa više vremena nego inače, zahvaljujući tome što u svakom koraku koristi samo po jednu instancu, pa korak košta jednako vremena, i što se instance u uvećanom skupu za obučavanje nižu na isti način na koji ih stohastički gradijentni spust i inače smenjuje.

Očigledno, ovo je ekstreman primer redundantnosti podataka, koji se ne očekuje u praksi, ali je ova prednost stohastičkog gradijentnog spusta osetna i u manje ekstremnim slučajevima.

Već je rečeno da pri gradijentnom spustu gradijent u nekim situacijama naglo menja pravac, što dovodi do cik-cak kretanja i sporije konvergencije. Slično, stohastički gradijentni spust takođe može značajno menjati pravac usled toga što se pravac gradijenta za celu funkciju f ocenjuje na osnovu samo jedne njene komponente f_i. *Metod inercije* se zasniva na ideji akumuliranja prethodnih gradijenata, pri čemu je značaj starijih gradijenata manji, a novijih veći, a onda se umesto gradijenta u datoj tački koristi ukupan akumulirani gradijent. Kako prosek nekih vrednosti, manje varira nego same vrednosti, ovakva tehnika dovodi do manjih promena pravca u gradijentu i često do povećanja brzine konvergencije. Metod inercije je definisan na sledeći način:

$$d_0 = 0$$

$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k)$$

$$x_{k+1} = x_k - d_{k+1}$$

pri čemu važi $0 \leq \beta < 1$. U vektoru d_k se akumuliraju gradijenti prvih k koraka. Pritom, kako se d_k u svakoj iteraciji množi brojem manjim od 1, uticaj ranijih

gradijenata eksponencijalno brzo opada, tako da skoriji gradijenti dosta više utiču na pravac koraka. Ovaj metod se često koristi za treniranje neuronskih mreža, ali se češće koristi u kombinaciji sa stohastičkim gradijentnim spustom, kada se umesto funkcije f, u definiciji pravila metode koristi njena komponenta f_i.

Nesterovljev ubrzani gradijentni spust je modifikacija metoda inercije, koja predstavlja asimptotski optimalan algoritam prvog reda za konveksne funkcije. Ukoliko je funkcija konveksna sa Lipšic neprekidnim gradijentom, greška je reda $O\left(\frac{1}{k^2}\right)$, naspram $O\left(\frac{1}{k}\right)$ u slučaju običnog gradijentnog spusta, pod istim uslovima.

$$d_0 = 0$$
$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k - \beta_k d_k)$$
$$x_{k+1} = x_k - d_{k+1}$$

Algoritam definiše specifičan izbor vrednosti α_k i β_k, ali o njemu neće biti reči. Ovaj algoritam je posebno pogodan u slučaju podataka visoke dimenzionalnosti. Naime, u tom slučaju je teško primeniti metode drugog reda, zbog toga što veličina hesijana može biti ogromna. Zbog svoje brzine, Nesterovljev algoritam je tada najbolja alternativa. I on se često koristi u treniranju neuronskih mreža, najčešće u kombinaciji sa stohastičkim gradijentnim spustom.

5.2.3 Metode lokalne optimizacije drugog reda bez ograničenja

Metode optimizacije drugog reda pored vrednosti funkcije i gradijenta, koriste hesijan. Kao što prvi parcijalni izvodi pružaju informaciju o brzini promene funkcije duž različitih koordinatnih pravaca, tako hesijan pruža informaciju o brzini promene gradijenta duž različitih koordinatnih pravaca. Zahvaljujući većoj količini informacije koju koriste, obično konvergiraju u mnogo manje iteracija. Zahvaljujući veličini hesijana i operacijama koje ga uključuju, mogu zahtevati dosta više memorije, a pojedinačne iteracije mogu koštati dosta više vremena. Ipak, što se tiče vremena zaustavljanja, smanjenje u broju iteracija nadjačava povećanje cene iteracije, što rezultuje kraćim trajanjem procesa optimizacije.

Najpoznatiji metod drugog reda je *Njutnov metod*. U slučaju funkcija jedne promenljive, tekuće rešenje se ažurira po pravilu:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

U slučaju funkcija više promenljivih, pravilo je analogno:

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

Za jako konveksne funkcije sa Lipšic neprekidnim hesijanom, greška je reda $O\left(c^{2^k}\right)$, za neko $0 < c < 1$, što je neuporedivo brže od metoda prvog reda.

Očigledno, hesijan mora biti invertibilan. Odnosno, metoda zahteva strogu konveksnost.

Ukoliko je funkcija koja se minimizuje kvadratna, odnosno važi

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c$$

gradijent je $\nabla f(x) = b + Ax$, a Hesijan je $\nabla^2 f(x) = A$. Korak Njutnove metode daje

$$x_1 = x_0 - A^{-1}(b + Ax) = -A^{-1}b$$

Očigledno, tačka x_1 ne zavisi od tačke x_0, pa i sve naredne iteracije daju istu tačku. Ako se ovo rešenje uvrsti u gradijent, dobija se

$$\nabla f(-A^{-1}b) = b + A(-A^{-1}b) = 0$$

Kako je gradijent jednak nuli, radi se o stacionarnoj tački. Ukoliko je funkcija f konveksna, odnosno ako je matrica A pozitivno semidefinitna, sigurno se radi o minimumu, a ukoliko je funkcija f konkavna, odnosno ako je matrica A negativno semi definitna, sigurno se radi o maksimumu. U ostalim slučajevima radi se o sedlenim tačkama.

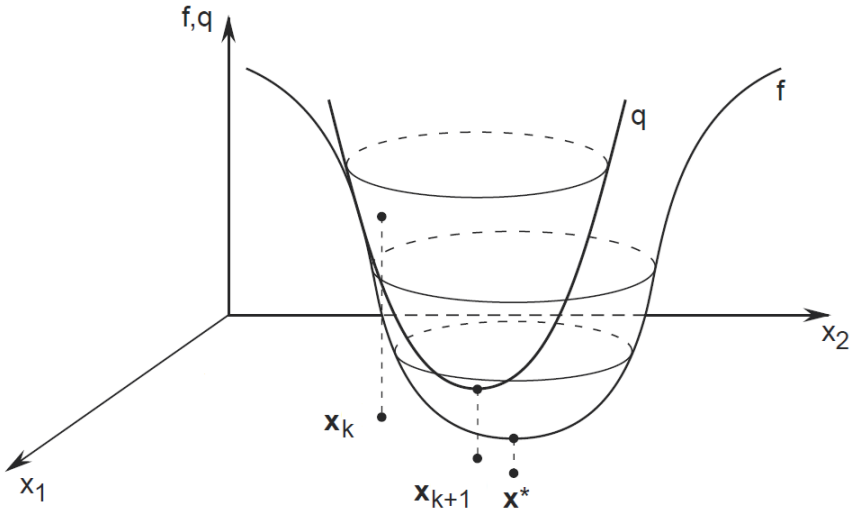
Iz prethodnog razmatranja proizilaze dva zanimljiva zapažanja. Prvo je to da Njutnova metoda zapravo ne traži minimum funkcije, već nulu gradijenta, što može biti minimum, ali može biti i maksimum i sedlena tačka. Ovo je argument više za prethodno pomenuti uslov stroge konveksnosti funkcije koja se minimizuje. Kako Njutnova metoda raspolaže samo vrednostima funkcije, gradijentom i hesijanom, može se doći do drugog zapažanja, a to je da Njutnova metoda zapravo vrši niz uzastopnih minimizacija lokalnih kvadratnih aproksimacija funkcije. Ovo je ilustrovano na slici 5.11.

U sumi, prednost Njutnove metode je brzina, a mane su zahtev za strogom konveksnošću funkcije i memorijska zahtevnost vezana za skladištenje hesijana, što je posebno izraženo u slučaju problema visoke dimenzije.

U nekim slučajevima hesijan nije dostupan, a nekad je preveliki za čuvanje i inverziju. U ovakvim situacijama, odgovor nude *kvazi-Njutnove metode*. Osnovna ideja ovih metoda je da se inverz hesijana aproksimira na osnovu gradijenata, tako da se operacija invertovanja i ne vrši, a aproksimacija se popravlja u svakom koraku. Kako se popravlja aproksimacija hesijana, tako se popravlja i kvadratna aproksimacija funkcije koju Njutnova metoda zapravo rešava. Najpoznatija metoda ovog tipa je *BFGS* (Brojden-Flečer-Goldfarb-Šano). Opšta forma pravila za ažuriranje tekućeg rešenja kod kvazi-Njutnovih metoda je

$$x_{k+1} = x_k - H_k^{-1}\nabla f(x_k)$$

pri čemu je matrica H_k^{-1} simetrčna i predstavlja pomenutu aproksimaciju matrice hesijana. Treba imati u vidu da se ni u jednom koraku ne računa inverz, već da se inverz neposredno aproksimira. Postavlja se pitanje, na koji se način



Slika 5.11: Funkcija i njena kvadratna aproksimacija u okolini tačke x_k, čijom se minimizacijom dobija nova tačka x_{k+1}.

ova aproksimacija može izabrati. Pored uslova simterije, algoritam BFGS pretpostavlja i slaganje gradijenata funkcije f i njene kvadratne aproksimacije $\overline{f}_k$ u tački x_k:

$$\overline{f}_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T H_k(x - x_k)$$

Gradijent ove fukcije je očito

$$\nabla \overline{f}_k(x) = \nabla f(x_k) + H_k(x - x_k)$$

Gradijenti se očito slažu u tački x_k. Dodatno se zahteva da se slažu i u tački x_{k-1}, odnosno

$$\nabla f(x_k) + H_k(x_{k-1} - x_k) = \nabla f(x_{k-1})$$

Ovaj uslov se oslanja na matricu H_k, što je nepoželjno, pošto je aproksimirana H_k^{-1}, pa se uslov transformiše u ekvivalentan:

$$H_k^{-1}(\nabla f(x_k) - \nabla f(x_{k-1})) = x_k - x_{k-1}$$

Dodavanjem ovog uslova, aproksimacija i dalje nije jedinstveno određena i do kraja se definiše zahtevom da pod datim uslovima bude što bliža prethodnoj aproksimaciji, odnosno da H_k^{-1} predstavlja rešenje narednog optimizacionog

problema

$$\begin{aligned} &\min_{H^{-1}} \|H^{-1} - H_{k-1}^{-1}\|_2^2 \\ \text{pri uslovima } &H^{-1}(\nabla f(x_k) - \nabla f(x_{k-1})) = x_k - x_{k-1} \\ &{H^{-1}}^T = H^{-1} \end{aligned}$$

Ispostavlja se da ovaj problem ima rešenje u zatvorenoj formi, koje se brzo izračunava. Interesantno je da u njegovom izvođenju važnu ulogu igra Šerman-Morison-Vudburijeva formula.

BFGS algoritam otklanja problem poznavanja i invertovanja hesijana. Time se dobija na brzini jedne iteracije, a gubi na broju iteracija. Red greške je između $O\left(c^k\right)$ i $O\left(c^{2^k}\right)$. Odnosno, može se očekivati da ova metoda bude sporija od Njutnove, ali brža od metoda prvog reda. Ipak, ona i dalje ne rešava problem skladištenja hesijana. Ovaj problem rešava metoda *LBFGS* (eng. low memory BFGS), koja se zasniva na čuvanju određenog broja poslednjih razlika gradijenata i razlika rešenja koje figurišu u prethodnom minimizacionom problemu, umesto čuvanja matrice H_k^{-1}, pri čemu na osnovu tih razlika postoji efikasan način izračunavanja približne vrednosti proizvoda $H_k^{-1}\nabla f(x_k)$.

5.2.4 Linijska pretraga

Svi diskutovani metodi su iterativni i zasnivaju se na ideji izračunavanja nekog pravca kretanja i preduzimanju koraka određene dužine u tom pravcu u svakoj iteraciji. Način na koji su diskutovani se fokusirao na izbor pravca, dok su za dužinu koraka na početku dati neki dovoljni uslovi. Ipak, i na ovom aspektu se može raditi kako bi bio bolji. Na primer, ni jedan od pristupa ne garantuje da će se vrednost funkcije monotono smanjivati, već je moguće da se povremeno dođe i do većih vrednosti. Linijska pretraga je jedan pristup izboru dužine koraka. Zasniva se na pretrazi duž odabranog pravca, za najboljom, ili makar povoljnom (što je definisano određenim dodatnim uslovima) dužinom koraka i zavisno od toga može biti egzaktna ili približna.

Pre diskusije dužine koraka, treba primetiti jedno svojstvo pravaca kretanja koje obično biraju optimizacione metode. Tipično se bira pravac d takav da važi $\nabla f(x)^T d < 0$. Drugim rečima, projekcija izabranog pravca na pravac gradijenta je negativna, odnosno radi se o pravcu u kojem vrednost funkcije opada. Takav pravac nazivamo *pravac spusta*. U nastavku se pretpostavlja da izabrani pravac zadovoljava ovaj uslov.

Egzaktna linijska pretraga u tački x_k, nakon izbora pravca kretanja d bira dužinu koraka rešavanjem narednog problema:

$$\min_{\alpha \geq 0} f(x_k + \alpha d)$$

U nekim slučajevima, ovaj problem se može tačno rešiti analitički, ali ukoliko se ne može rešiti analitički, pitanje je isplati li se rešavati ga iterativnim metodama. U praksi, egzaktna linijska pretraga se retko koristi.

Ukoliko se odustane od nalaženja najbolje dužine koraka, postavlja se pitanje, kakva dužina koraka se želi. Odgovora može biti više. Svakako, potrebno je da ukupan optimizacioni algoritam konvergira. Poželjno je da korak nije previše veliki, te da se zahvaljujući tome dobije monotono opadanje vrednosti ciljne funkcije. Linijska pretraga sa Armihovim uslovom ispunjava ove zahteve. Ona sa fokusira na izbor najvećeg koraka koji zadovoljava takozvani Armihov uslov, koji garantuje pad vrednosti ciljne funkcije dovoljan za konvergenciju. Neka je $\alpha_k = \alpha_0\beta^k$ za $k > 0$, $\alpha_0 > 0$ i $\beta \in (0, 1)$. Linijska pretraga sa Armihovim uslovom bira najmanje k, odnosno najveće α_k za koje važi Armihov uslov

$$f(x + \alpha_k d) \leq f(x) + \alpha_k \nabla f(x)^T d$$

Tu vrednost α_k označavamo α^*. Kako je d pravac spusta, važi $\nabla f(x)^T d < 0$, odnosno važi $f(x + \alpha^* d) < f(x)$. Trebalo bi formalno dokazati da je smanjenje vrednosti funkcije dovoljno veliko da garantuje konvergenciju, ali u dokaz nećemo ulaziti.

Postavlja se još jedno pitanje, a to je da li se postupak izbora vrednosti α^* nužno završava u konačnom vremenu. Za dovoljno malo α važi

$$f(x + \alpha d) \approx f(x) + \alpha \nabla f(x)^T d$$

Kako α_k eksponencijalno opada, za dovoljno veliko k važi $\alpha_k < \alpha$. Očito, α^* se onda pronalazi u k koraka. Kako je d pravac spusta, važi

$$f(x) + \alpha \nabla f(x)^T d < f(x) + \alpha^* \nabla f(x)^T d$$

pa za dovoljno malo α važi

$$f(x + \alpha d) < f(x) + \alpha^* \nabla f(x)^T d$$

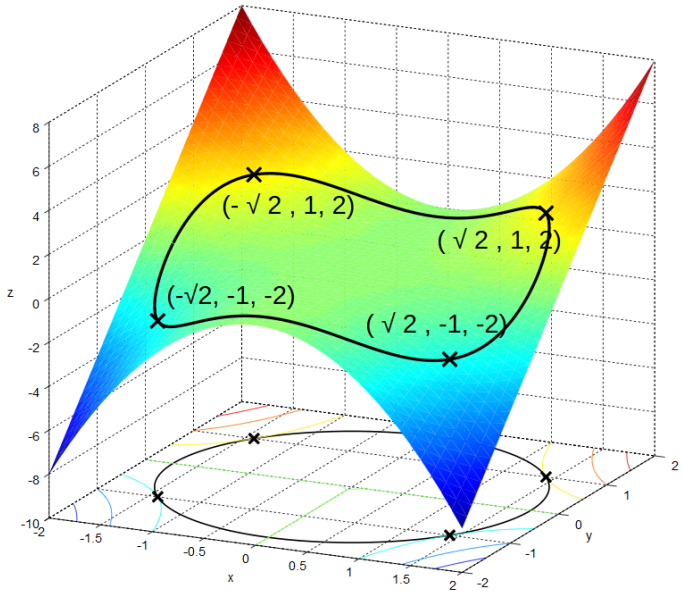
čime je pokazano da će uslov biti ispunjen u konačno mnogo (odnosno k) koraka.

Postoje i drugi uslovi koji obezbeđuju monotono opadanje i, u praksi, brzu konvergenciju. Oni tipično uključuju Armihov uslov, ali neće biti diskutovani.

5.2.5 Metode lokalne optimizacije sa ograničenjima

Ograničenja sužavaju domen na manji skup dopustivih rešenja. Minimum funkcije pri datim ograničenjima često nije jednak pravom minimumu funkcije. Dodatno, funkcije koje nemaju minimum, mogu ga imati u prisustvu ograničenja. Ovo je ilustrovano na slici 5.13. Problemi sa ograničenjima zahtevaju posebne metode rešavanja. Naime, praćenje gradijenta funkcije može lako dovesti do izlaska iz skupa dopustivih rešenja. Metode optimizacije u prisustvu ograničenja su raznovrsne i često se konstruišu za specifične klase problema.

Najjednostavnija klasa problema sa ograničenjima su linearni problemi, odnosno problemi *linearnog programiranja*. Pritom, pod pojmom programa se ne



Slika 5.12: Minimumi i maksimumi funkcije u prisustvu ograničenja. Funkcija nema minimum ni maksimum ukoliko se razmatra bez ograničenja.

podrazumeva kod u nekom programskom jeziku, već postavka optimizacionog problema. Problemi linearnog programiranja se karakterišu linearnom funkcijom cilja i linearnim ograničenjima. Pored njih, još jedna jednostavna klasa problema sa ograničenjima su problemi *kvadratnog programiranja*, kod kojih je funkcija cilja kvadratna, a ograničenja su linearna.

Primer 69 *Za problem raspoređivanja lampi, predložena je jedna aproksimacija za koju je rečeno da se može rešiti metodama linearnog programiranja:*

$$\min_{p_1,\ldots,p_m,I_1,\ldots,I_n} \max_{j=1,\ldots,n} |I_j - I|$$

$$\text{pri uslovima } I_j = \sum_{i=1}^{m} \frac{cos\theta_{ij}}{r_{ij}^2} p_i \qquad j = 1,\ldots,n$$

$$0 \le p_i \le p_{max} \qquad i = 1,\ldots,m$$

Ciljna funkcija ovog problema očigledno nije linearna. Međutim, optimizacioni problemi se često mogu reformulisati, tako da imaju povoljniju formu. Konkretno, minimizacija maksimuma i apsolutne vrednosti se lako predstavlja u vidu linearnog problema, isto važi i za njihovu kompoziciju. Reformulisani pro-

blem je:

$$\begin{aligned} &\min_t t \\ \textit{pri uslovima} \quad & I_j - I \leq t && k = 1, \ldots, n \\ & -(I_j - I) \leq t && k = 1, \ldots, n \\ & I_j = \sum_{i=1}^{m} \frac{cos\theta_{ij}}{r_{ij}^2} p_i && j = 1, \ldots, n \\ & 0 \leq p_i \leq p_{max} && i = 1, \ldots, m \end{aligned}$$

Problem optimizacije portfolija je očigledno problem kvadratnog programiranja. Problem transporta gasa nije problem ni linearnog ni kvadratnog programiranja, pošto među ograničenjima ima nelinarnih.

Najpoznatiji algoritam za rešavanje problema linearnog programiranja je istovremeno i prvi – *simpleks algoritam*. Ovaj algoritam ima eksponencijalnu složenost, ali u praksi vrlo često efikasno pronalazi tačna (do na grešku zaokruživanja) rešenja problema. Ipak, postoje algoritmi linearnog programiranja koji imaju polinomijalnu složenost.

U slučaju konveksnog skupa dopustivih rešenja, moguće je primeniti metod *projektovanog gradijenta*. Euklidska projekcija $P_U(x)$ tačke x na skup U je tačka skupa U najbliža tački x, odnosno rešenje sledećeg optimizacionog problema:

$$\min_{u \in U} \|x - u\|_2$$

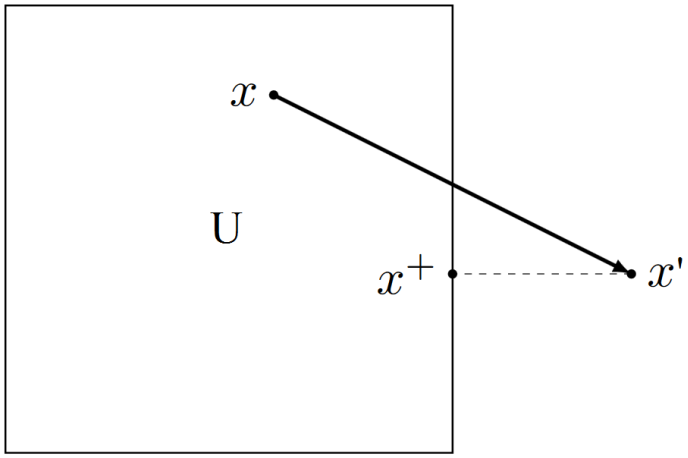
Metod projektovanog gradijenta se zasniva na pravilu:

$$x_{k+1} = P_U(x_k - \alpha_k \nabla f(x_k))$$

Odnosno, dovoljno je u svakoj iteraciji projektovati novo rešenje gradijentnog spusta na skup dopustivih rešenja, kao na slici 3.8. Naravno, postavlja se pitanje kako se može izračunati projekcija na neki konveksan skup. Jedan način je – optimizacijom. Međutim, takav pristup je previše neefiksan i ovaj metod se stoga primenjuje samo u situacijama u kojima je moguće analitički izraziti projekciju. Takvi primeri su projekcija na proizvoljan zatvoren interval orijentisan u skladu sa koordinatnim osama i projekcije na lopte u odnosu na metrike indukovane $\|\cdot\|_1$ i $\|\cdot\|_2$ normama.

Brzina konvergencije projektovanog gradijentnog spusta je ista kao kod običnog gradijentnog spusta.

Nešto opštiji su metodi zasnovani na *kaznenim funkcijama*, kojih ima različitih vrsta. Jedna varijanta počiva na takozvanoj *logaritamskoj barijeri*. Ideja je da



Slika 5.13: Tekuće rešenje x.

se opšti problem minimizacije

$$\begin{aligned} &\min_{x\in\mathcal{D}} f(x) \\ &\text{pri uslovima} \quad g_i(x) \leq 0 \qquad i = 1, \ldots, L \end{aligned}$$

reši iterativno, tako što se u k-toj iteraciji rešava problem

$$\min_x f(x) + \frac{1}{\mu_k} \sum_{i=1}^{L} -\log(-g_i(x))$$

pri čemu se u svakoj iteraciji za polaznu tačku uzima rešenje prethodne iteracije, i pri čemu je niz μ_k strogo rastući. Polazna tačka prve iteracije mora biti iz skupa dopustivih rešenja. Ukoliko nije trivijalno pronaći neko dopustivo rešenje, postoje metode koje nalaze neko dopustivo rešenje.

Očigledno, ukoliko je $g_i(x)$ blisko nuli, za neko i, vrednost kaznene funcije $-\log(-g_i(x))$ je veliki pozitivan broj. Stoga će svaka metoda optimizacije birati rešenja koja nisu previše blizu granici skupa dopustivih rešenja. Naravno, moguće je da minimum leži baš na granici ili u njenoj blizini. Povećavanjem parametra μ se omogućava smanjenje uticaja kaznene funkcije daleko od granice, bolja aproksimacija originalnog problema i približavanje granici skupa dopustivih rešenja.

Metodama zasnovanim na kaznenim funkcijama se mogu rešavati opštiji problemi nego metodom projektovanog gradijentnog spusta, pošto ne zahtevaju poznavanje projekcije, niti konveksnost skupa dopustivih rešenja, ali su, s druge strane, osetno sporije.

5.3 Diskretna optimizacija

Diskretna optimizacija pretpostavlja diskretnost nekog od elemenata optimizacionog problema. Diskretan može biti domen, funkcija cilja ili oba. Diskretnost domena, na primer ako je domen $\mathbb{Z}^n$, čini da razmatranje okolina (u smislu okolina kakve podrazumevamo u $\mathbb{R}^n$) nema smisla i da analitički alat zasnovan na izvodima više nije upotrebljiv. Slično, ako je funkcija cilja diskretna, izvodi su tipično svuda jednaki nuli osim u tačkama prekida funkcije gde su nedefinisani. Oba slučaja traže nove pristupe i vode metodama koji se tipično prirodno posmatraju kao algoritmi pretrage na prostoru potencijalnih rešenja. Nekada je ta pretraga egzaktna, odnosno garantuje pronalaženje optimalnog rešenja, a nekada je heuristička, što znači da ne pruža takve garancije, ali tipično u praksi uz malo truda daje dobre rezultate. Egzaktne metode su tipično računski vrlo zahtevne i najčešće se ne mogu primeniti na velike prostore pretrage, dok heurističke metode obično daju zadovoljavajuće rezultate i na mnogo većim prostorima pretrage. Metode diskretne optimizacije nazivamo i metodama kombinatorne optimizacije.

Za razliku od metoda neprekidne optimizacije koje se prirodno formulišu kao metode lokalne pretrage, metode diskretne optimizacije se tipično prirodno formulišu kao metode globalne optimizacije.

5.3.1 Egzaktne metode

Postoje različite egzaktne metode kombinatorne optimizacije. Ono što je karakteristično za njih je da garantuju optimalnost rešenja. Zbog odsustva alata poput gradijenata i hesijana koji mogu efiksano da navode optimizaciju, ove metode se u najgorem slučaju svode na iscrpnu pretragu prostora rešenja i na problemima na kojima se primenjuju tipično imaju eksponencijlanu ili lošiju složenost najgoreg slučaja. Naravno, sami ti problemi su obično teški (npr. NP-teški). Prethodna konstatacija podrazumeva da je prostor pretrage razmatranog problema konačan. U nastavku će biti opisan jedan pristup konstrukciji egzaktnih metoda, poznat po nazivom *grananje i ograničavanje* (eng. *branch and bound*).

Grananje i ograničavanje nije jedna konkretna metoda već opšti okvir za konstrukciju egzaktnih metoda koji je potrebno precizirati za neki konkretan kombinatorni problem. Zasniva se na ideji da se prostor rešenja može podeliti na dva ili više delova koji u uniji čine ceo prostor koji se dalje mogu rekurzivno deliti dok se ne dođe do pojedinačnih dopustivih rešenja. Na ovaj način prostor pretrage se može struktuirati u vidu stabla pretrage, a proces deljenja nazivamo grananjem. Ova tehnika sama za sebe se naravno svodi na iscrpnu pretragu i stoga nije previše zanimljiva. Međutim, ukoliko bi bilo moguće odseći određena podstabla stabla pretrage, bilo bi moguće i uštedeti na vremenu. Odsecanje počiva na mogućnosti brzog određivanja donje granice vrednosti funkcije cilja na bilo kom od delova prostora pretrage koji mehanizam deljenja može konstruisati. Kad god je donja granica nekog od potprostora veća

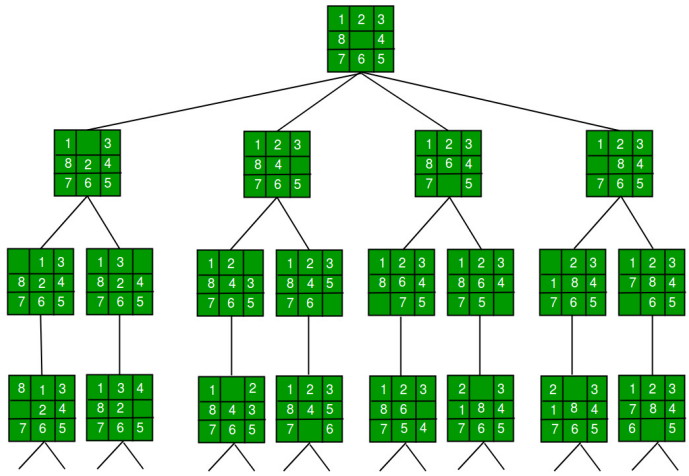
od najniže vrednosti pronađene u toku pretrage, celo podstablo koje odgovara tom potprostoru se može zanemariti. Da bi se na osnovu ovog principa mogao implementirati algoritam, potrebno je da bude definisana neka struktura podataka koja predstavlja konkretnu instancu optimizacionog problema, odnosno prostor pretrage određen tim problemom, potom funkcija *branch* koja za datu instancu izračunava listu instanci potproblema koji se dobijaju grananjem, funkcija *bound* koja izračunava donju granicu funkcije cilja na nekoj instanci i funkcija *single* koja za datu instancu problema određuje da li je njegov skup rešenja jednočlan, što je bitno zbog zaustavljanja grananja.

Ovaj princip je ilustrovan narednim algoritmom. Neka je P ulazna instanca problema i neka je Q pomoćna struktura reda. Tipično se koristi red sa prioritetom kod kojeg se prioritet daje elementima sa manjom vrednošću funkcije cilja.

- Ukoliko je moguće, nekom heuristikom odrediti neko (potencijalno što bolje) dopustivo rešenje x problema, neka je $B = f(x)$ i $s = x$. Ukoliko takva heuristika nije dostupna, neka je $B = \infty$.
- Neka je $Q = [P]$.
- Ponavljati dok $Q \neq \emptyset$
  - Uzeti instancu I iz reda Q
  - Ukoliko važi $single(I)$ i ako je x jedino rešenje instance I i važi $f(x) < B$, onda dodeliti $B = f(x)$ i $s = x$ i preskočiti ostatak iteracije.
  - Neka je $[I_1, \ldots, I_n] = branch(I)$
  - Svaku instancu I_j za koju važi $bound(I_i) < B$ staviti u red.
- Vratiti (s, B)

Očigledno, od kritičnog značaja za efikasnost pretrage je kvalitet funkcije *bound*. U slučaju da je $bound(I) = -\infty$ za svaku instancu I, algoritam se svodi na iscrpnu pretragu, što je i intuitivno jer funkcija *bound* ne pruža baš nikakvu informaciju. U drugom ekstremnom slučaju, ukoliko bi $bound(I)$ bila baš vrendost najboljeg rešenja instance I, algoritam bi mogao da odseče veliki broj potproblema. Naravno, bitan je i način podele koji realizuje funkcije *branch*. U slučaju idealne funkcije *bound*, ako bi funkcija *branch* delila prostor pretrage na dva potprostora jednake veličine, rešenje bi bilo pronađeno u logaritamskom vremenu u odnosu na veličinu polaznog prostora rešenja. Konačno i očigledno, poželjno je da su sve korišćene funkcije što efikasnije.

Primer 70 *Neka je data slagalica pravougaonog oblika sa n polja. Na tih n polja, rasporedjeno je $n-1$ delova slagalice numerisanih brojevima od 1 do $n-1$, a jedno polje je prazno. Delovi se mogu pomerati levo, desno, gore i dole, ali uvek samo na prazno polje. Za potrebe diskusije, može se smatrati da se prazno*

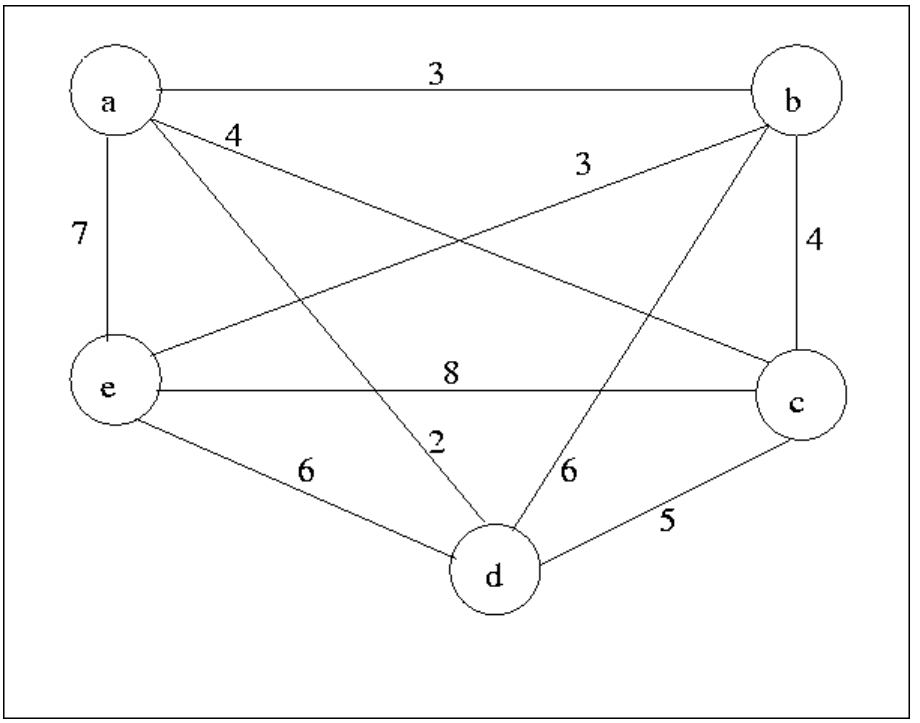


Slika 5.14: Stablo rešavanja slagalice od 8 delova.

polje pomera na mesto nekog od delova (koji prelazi na mesto gde je bilo prazno polje). Cilj je pronaći redosled poteza kojim se najbrže slaže slagalica. Postoji mogućnost da je polazno stanje takvo da je to nemoguće uraditi, ali u nastavku se pretpostavlja da nije dato takvo stanje. Kako bi se rešenje dobilo principom grananja i ograničavanja, potrebno je definisati prethodno pomenute funkcije.

Grananje je moguće realizovati pomeranjem praznog polja u jednom od najviše četiri pravca (ukoliko je prazno polje pri ivici, na raspolaganju je manje od četiri pravca). Novodobijeni problemi se dalje mogu granati na isti način. Kako ne bi došlo do ciklusa, svako posećeno stanje se na neki način označava kao posećeno i u kasnijim grananjima se zanemaruje. Na taj način, dobija se stablo kao na slici 5.14. Svaki od čvorova stabla odgovara skupu rešenja koja počinju nizom poteza koji predstavljaju grane od korena do tog čvora i nastavljaju se mogućim potezima iz tog čvora do lista koji predstavlja složenu slagalicu. Utvrđivanje da li je x jedino rešenje tekućeg potproblema se sprovodi tako što se ustanovi da li su sva stanja u koja se može dospeti iz tekućeg već posećena. Ključni element je ocena donje granice broja poteza za neki potproblem. Ova granica se može dobiti kao zbir tačnog broja poteza do tekućeg stanja (pošto grananje i ograničavanje vrši pretragu u širinu, onda je put kojim se dolazi do nekog stanja sigurno najkraći put po broju poteza) i procenjenog broja poteza od tekućeg stanja do ciljnog stanja. Pritom, bitno je da procena bude optimistična, kako bi zbir predstavljo donju granicu broja koraka. Jedna moguća optimistična procena bi mogla biti jednaka broju polja koja se ne nalaze na željenim mestima. Druga bi mogla biti jednaka sumi Menhetn rastojanja svih polja od željenih pozicija.

Primer 71 *Jedan od najpoznatijih kombinatornih problema i jedan od NP-kompletnih problema je problem trgovačkog putnika. Pretpostavlja se da je dat težinski graf kod kojeg težine interpretiramo kao dužine grana i da je u*



Slika 5.15: Primer težinskog grafa.

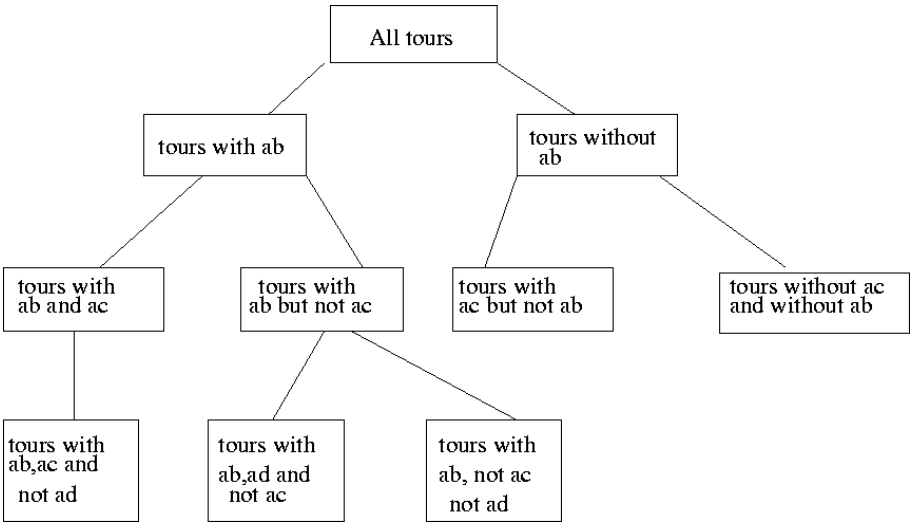
njemu potrebno naći zatvorenu putanju najmanje ukupne dužine koja sadrži sve čvorove tačno jednom ili ustanoviti da takva putanja ne postoji. U nastavku, kad god se govori o putanji, misli se na zatvorenu putanju koja sadrži sve čvorove po jednom. Primer jednog grafa dat je na slici 5.15.

Kao i uvek, pitanje je kako formulisati prethodno diskutovane funkcije. Kao i obično, najteže je definsati funkciju koja procenjuje donju granicu. Ako je data putanja, svaki čvor u putanji ima dve susedne grane. Cena putanje se može predstaviti kao polovina zbira po svim čvorovima grana koje su im susedne u putanji. Na primer, ako se razmatra putanja $a - b - c - d - e$*, navedena suma je* $(7 + 3 + 3 + 4 + 4 + 5 + 5 + 6 + 6 + 7)/2$*. Na primer, prva dva sabirka* 7 *i* 3 *dolaze od dve grane koje su susedne čvoru* a*. Za donju granicu dužine svih putanja može se uzeti suma po dve najkraće susedne grane svih čvorova. U slučaju prikazanog grafa, izabrane grane, po čvorovima, su:*

a: (a, d), (a, b)

b: (b, a), (b, e)

c: (c, b), (c, a)



Slika 5.16: Stablo pretrage za problem trgovačkog putnika na grafu sa slike 5.15.

d: (d,a), (d,c)

e: (e,b), (e,d)

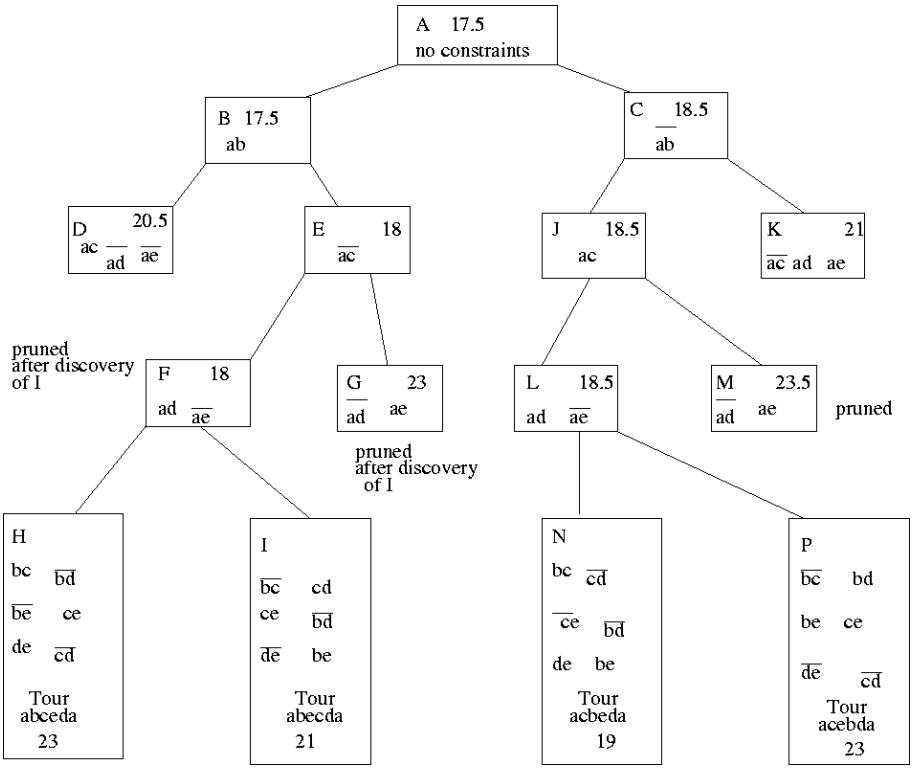
a polovina sume svih navedenih grana je 17.5. *Naravno, ova donja granica važi samo za skup svih putanja, a nije očigledno kako se izvodi grananje i ocena donje granice za podskupove skupa svih putanja. Grananje se može vršiti prema prisustvu neke grane u putanji i jedan podskup putanja mogu činiti sve putanje koje sadrže tu granu, a drugi podskup sve putanje koje ne sadrže tu granu. Ilustracija takvog stabla za graf na slici 5.15, dat je na slici 5.16.*

Prema konstrukciji podskupova, nema smisla koristiti prethodno definisanu ocenu (koja bi, usput, bila konstantna), već je potrebno uzeti u obzir prisustvo ili odustvo date grane. Ako grana (x,y) mora biti uključena, onda u prethodno diskutovanoj listi ta grana menja skuplju od dve grane pridruženu svakom od svoja dva krajnja čvorova x i y. Ukoliko grana ne sme biti uključena, ako se ona javlja u prethodnoj listi za neki čvor, onda je na tom mestu zamenjuje sledeća najjeftinija grana susedna tom čvoru. Na primer, ukoliko mora biti uključena grana (a,e), a isključena grana (b,c), lista grana na osnovu kojih se izračunava donja granica je

a: (a,d), (a,e)

b: (b,a), (b,e)

c: (c,a), (c,d)



Slika 5.17: Stablo pretrage sa gonjim granicama dužine putanje.

d: (d, a), (d, c)

e: (e, b), (e, a)

U ovom slučaju donja granica je 20.5.

Prilikom grananja, potrebno je obratiti pažnju na sledeće dve mogućnosti. Ukoliko isključivanje grane (x, y) *onemogućava čvorove* x *i* y *da imaju po dve susedne grane, ova grana mora biti uključena u putanju. Ukoliko uključivanje grane* (x, y) *čini da čvor* x *ili* y *ima više od dve susedne grane ili zatvara putanju koja ne obuhvata sve gradove, onda ova grana mora biti isključena iz putanje. Celo stablo pretrage sa donjim granicama dužine putanje za sve razmatrane potprobleme prikazano je na slici 5.17.*

5.3.2 Heurističke metode

Heurističke metode optimizacije ne pružaju garancije optimalnosti svojih rešenja, kao što ih pružaju egzaktne metode. Umesto o heuristikama, preciznije je govoriti o takozvanim *metaheuristikama*, koje predstavljaju opšte šablone

čijim preciziranjem se dolazi do heurističkih metoda za konkretne probleme. Metaheurističkih pristupa ima mnogo. Naglasićemo postojanje njihove dve klase. Jednu čine populacione metaheuristike koje se zasnivaju na održavanju populacije dopustivih rešenja koja se paralelno menjaju, popravljaju, kombinuju, interaguju i slično. Tipičan predstavnik ove klase i verovatno najpoznatiji od svih metaheurističkih pristupa su genetski algoritmi, ali ima i drugih poput optimizacije pomoću kolonije mrava, roja čestica itd. Pored njih postoje metaheuristike koje pretpostavljaju održavanje jednog dopustivog rešenja. Takve su na primer simulirano kaljenje, tabu pretraga i metoda promenljivih okolina.

Metaheuristike obično uključuju dva važna mehanizma – mehanizam intenzifikacije i mehanizam diversifikacije. Prvi služi popravljanju tekućeg ili tekućih rešenja i očigledno je neophodan za jednu metodu optimizacije. Drugi služi bekstvu iz lokalnih minimuma i povećavanju šansi da će pronađeni mimum biti globalni. Naravno nikakve garancije u tom smislu ne postoje u praksi, iako postoje teorijski rezultati koji garantuju konvergenciju, pod određenim, često nerealističnim uslovima.

Treba naglasiti da se metaheuristike tipično mogu primeniti i na probleme neprekidne optimizacije, posebno ukoliko se želi povećati šansa nalaženja globalnog optimuma umesto loklanog, ali njihove prednosti posebno dolaze do izražaja u slučaju problema kombinatorne optimizacije.

Metod koji će biti opisan u nastavku je *metod promenljivih okolina* (eng. *variable neighbourhood search*) ili skraćeno VNS. Ova metaheuristika pretpostavlja održavanje jednog rešenja i pretpostavlja postojanje metoda lokalne optimizacije za dati problem, koji realizuje mehanizam intenzifikacije i postojanje skupa okolina za svako od dopustivih rešenja iz kojih se nasumice biraju druga dopustiva rešenja, čime se realizuje mehanizam diversifikacije koji se naziva *razmrdavanje* (eng. *shaking*). Ovu metodu ćemo, po ugledu na autore metode, uvesti u tri varijante rastuće složenosti.

Prva varijanta je takozvana redukovana metoda promenljivih okolina. Pretpostavlja se da je definsan skup okolina $\mathcal{N}_i(x)$, za $i = 1, \ldots, K$ za svako dopustivo rešenje $x \in D$. Na primer, ukoliko je $D = \mathbb{R}^n$, okoline mogu biti definisane sferama različitih prečnika. Tipično, okolina sa manjim indeksom je ugnežđena u okolinu sa većim indeksom. Konkretno, ako su B_1 i B_2 dve lopte prečnika r_1 i r_2, takvih da važi $r_1 < r_2$, okoline mogu biti skupovima B_1 i $B_2 \setminus B_1$. Primetimo, nakon što je lopta B_1 pretražena, nema potrebe da se ponovo pretražuje u okviru lopte B_2. Ipak, ne postoji formalno ograničenje ovog tipa vezano za izbor okolina. Metod se može opisati na sledeći način:

1. Inicijalizovati polazno dopustivo rešenje x
2. Ponavljati naredne korake dok nije ispunjen kriterijum zaustavljanja:
   a) Neka je $k = 1$
   b) Ponavljati dok važi $k < K$
      i. Razmrdavanje: nasumice generisati tačku $x' \in \mathcal{N}_k(x)$

ii. Kretanje: ukoliko je $f(x') < f(x)$, neka je $x = x'$ i $k = 1$, a u suprotnom, neka je $k = k + 1$

U ovoj formulaciji metode, nema upotrebe lokalne pretrage, već intenzifikacija počiva samo na filtriranju rešenja koja ponudi diversifikacija. Očito, metoda se fokusira na nalaženje novih dopustivih rešenja koja su bliska tekućem (pod pretpostavkom da su okoline sa manjim indeksima u nekom smislu uže), ali bolja od njega, a u slučaju da takvih nema, popravka se traži nešto dalje.

Naredna varijanta, takozvana osnovna metoda promenljivih okolina uvodi loklanu pretragu:

1. Inicijalizovati polazno dopustivo rešenje x
2. Ponavljati naredne korake dok nije ispunjen kriterijum zaustavljanja:
   a) Neka je $k = 1$
   b) Ponavljati dok važi $k < K$
      i. Razmrdavanje: nasumice generisati tačku $x' \in \mathcal{N}_k(x)$
      ii. Lokalna pretraga: primeniti neki metod lokalne optimizacije počevši od x' i označiti rezultat sa x''
      iii. Kretanje: ukoliko važi $f(x'') < f(x)$, neka je $x = x''$ i $k = 1$, a u suprotnom, neka je $k = k + 1$

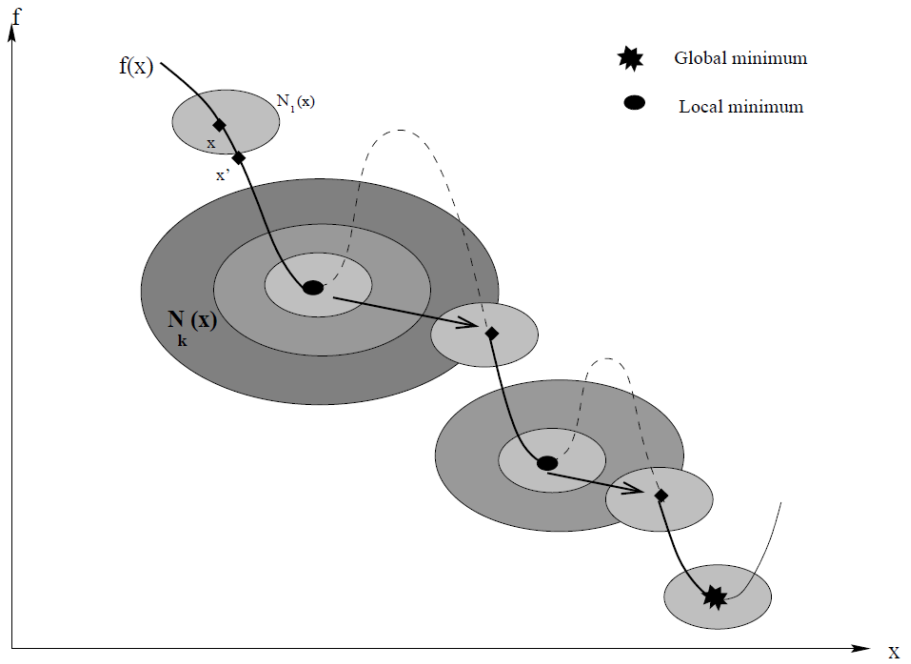
Ponašanje ove metode je ugrubo ilustrovano na slici 5.18.

Opšta metoda promenljivih okolina se dobija kada se za mehanizam lokalne pretrage izabere takozvana metoda *spusta sa promenljivim okolinama* (eng. *variable neighbourhood descent*). U nastavku je opisan taj mehanizam. On pretpostvlja da je dat skup okolina $N_l(x)$ za $l = 1, \ldots, L$ i polazno dopustivo rešenje x i može se opisati na sledeći način:

1. Neka je $l = 1$
2. Ponavljati dok važi $l < L$
   a) Pretraga: naći najbolje $x' \in N_l(x)$
   b) Kretanje: ukoliko važi $f(x') < f(x)$, neka je $x = x'$ i $l = 1$, a u suprotnom $l = l + 1$

Skupovi okolina $\mathcal{N}_k$ i N_l se ne moraju podudarati.

Primer 72 *Razmotrimo kako bi se pomoću ove heuristike rešavao problem trgovačkog putnika. Metoda lokalne optimizacije može biti* 2*-opt, koja polazeći od neke putanje, ispituje sve putanje koje se od nje mogu dobiti izborom dve nesusedne grane* (a, b) *i* (c, d) *i njihovom zamenom granama* (c, b) *i* (a, d)*. Ako se na taj način može dobiti putanja bolja od tekuće, ona se uzima za tekuću. Struktura okolina se mož e dobiti definisanjem rastojanja. Za dve putanje* t_1



Slika 5.18: Ilustracija rada osnovne metode promenljivih okolina.

i t_2*, rastojanje* $\rho(t_1, t_2)$ *se definiše kao broj grana koje nisu sadržane u obe putanje. Onda se okoline definišu, recimo na sledeći način:*

$$\mathcal{N}_k(x) = \{x' \mid \rho(x, x') = k,\ x' \in D\}$$

za $k = 2, \ldots, n$*. Time je definisana metoda promenljivih okolina za problem trgovačkog putnika.*

Primer 73 *Još jedan težak problem je problem optimalnog klasterovanja u odnosu na sumu kvadrata rastojanja od centara klastera. Ako je* $X = \{x_1, \ldots, x_n\}$ *skup vektora koje je potrebno podeliti u* k *disjunktnih skupova, odnosno klastera* $C_1, \ldots, C_m$ *koji obuhvataju sve instance iz skupa* X*, jedan često razmatran kriterijum kvaliteta klasteraovanja je sledeći*

$$\sum_{i=1}^{k} \sum_{x \in C_i} \|x - \overline{x}_i\|^2$$

gde važi

$$\overline{x}_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

Jedan poznat način lokalne optimizacije ove sume je algoritam k sredina koji nasumice bira k elemenata iz skupa X, takozvanih centroida, pridružuje svaku instancu najbližoj centroidi i tako formira k klastera, potom izračunava proseke tih klastera i uzima ih za nove centroide i nastavlja sa ponavljanjem ova dva koraka dok se podela menja. Ovaj algoritam garantovano konvergira lokalnom optimumu. Na osnovu jednog klasterovanja, odnosno particionisanja na skup klastera, susedno se može dobiti prebacivanjem jedne instance iz jednog klastera u drugi. Okoline jednog klasterovanja se onda mogu formirati na osnovu broja ovakvih izmena pošavši od tog klasterovanja.