Rijetka reprezentacija u dubokom strojnom učenju

Šebalj, Dolores

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Dolores Šebalj

RIJETKA REPREZENTACIJA U DUBOKOM STROJNOM UčENJU

Diplomski rad

Voditelj rada:
prof. dr. sc. Zlatko Drmač

Zagreb, studeni, 2019.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Ovaj rad posvećujem Donatu.
Najveću zahvalu dugujem svojoj obitelji, na povjerenju i bezuvjetnoj potpori tijekom studija.
Zahvaljujem svome dečku na strpljenju i podršci.
Zahvaljujem svojim prijateljima, posebice onima koje sam upoznala tokom studija, s kojima je učenje bilo zabava.
Zahvaljujem mentoru, prof. dr. sc. Zlatku Drmaču na susretljivosti i ukazanoj pomoći tijekom izrade ovog diplomskog rada.

Sadržaj

Uvod

Suvremen čovjek, koristeći različite tehnološke naprave, svakodnevno stvara golemu količinu informacija. Posao podatkovne znanosti je u hrpi naizgled nebitnih podataka pronaći strukturu te analizom iz nje izvući značajne zaključke. Problem kod rukovanja takvom golemom količinom informacija predstavljaju memorija i vrijeme - tu u pomoć uskače rijetka reprezentacija. Pojednostavljivanjem prikaza podataka u računalu, to jest kompresijom signala, štedi se memorija, a računalni se procesi ubrzavaju. Brojne ustaljene metode podatkovne znanosti i strojnog učenja, kao što su klasifikacija, regresija te neuronske mreže, profitiraju koristeći rijetku reprezentaciju upravo zbog efikasnosti koju ona donosi. Nadalje, rijetka reprezentacija može se koristiti za uklanjanje šuma iz signala te rekonstrukciju originalnog signala iz njegove oštećene verzije.

Ideja je rijetke reprezentacije prikazati podatak na što jednostavniji način. Pretpostavljamo da je podatak prikaziv kao linearna kombinacija nekoliko već poznatih podataka, s naglaskom upravo na riječi *nekoliko* — želimo da je što više koeficijenata u tom prikazu jednako nuli. Podaci čije kombinacije uzimamo zovemo atomima, i to ne bez razloga. Ovaj se model prikaza podataka može usporediti s periodnim sustavom elemenata. Naši su podaci molekule koje su sastavljene od različitih atoma, a potrebno nam je poznavati i količinu odgovarajućih atoma u molekuli, što je u linearnoj kombinaciji skalar pridružen pojedinom elementu. U prvom poglavlju uvest ćemo pogodnu funkciju koja mjeri rijetkost vektora, a zatim formirati problem pronalaska rijetke reprezentacije.

Problem koji se prvenstveno nameće je egzistencija i jedinstvnost dovoljno dobrog rijetkog prikaza za dani podatak, a potvrdan odgovor daje elegantna teorijska podloga. U drugom poglavlju dajemo teorijske garancije jedinstvenosti rješenja u određenim uvjetima. Osim toga, teorijska razmatranja dovela su do razvoja brojnih algoritama koji pronalaze rijetku reprezentaciju danog podatka tzv. algoritama potrage (engl. *pursuit algorithms*) kojima se bavimo u trećem poglavlju.

Rijetka reprezentacija pronašla je svoju primjenu u dubokom strojnom učenju uz konvolucijske rječnike. U četvrtom poglavlju dan je teorijski pregled veze između konvolu-

cijskih neuronskih mreža i rijetke reprezentacije.

Nadalje, problem je i pronalazak elemenata koje bismo mogli proglasiti atomima. Skup tako odabranih elemanata nazivamo rječnikom. Od njega zahtjevamo da dovoljno *prorijedi* naše podatke, odnosno da je svaki novi podatak moguće dovoljno dobro prikazati kao linearnu kombinaciju svega nekoliko atoma iz rječnika. Uz poznati rječnik, pronalazak rijetke reprezentacije proizvoljnog podatka postaje već dobro poznat zadatak - pronalazak rješenja pododređenog linearnog sustava jednadžbi. S obzirom na to da je potraga za rječnikom vrlo složen problem, u ovom radu spominjemo ga samo u kontekstu konvolucijskih mreža.

Poglavlje 1

Optimizacijski problem

Za početak dajemo neformalnu definiciju rijetke reprezentacije. Neka je dan signal $\mathbf{b} \in \mathbb{R}^n$ i neka je poznat rječnik $\mathbf{D} \in \mathbb{R}^{n \times m}$. Rječnik zapisujemo kao $n \times m$ matricu gdje je svaki stupac jedan atom iz rječnika. Ovdje je $m > n$ - matrica ima više stupaca nego redaka i stoga nije punog stupčanog ranga. Pretpostavljamo također da $\mathbf{D}$ nema nul-stupaca. Rijetka reprezentacija signala $\mathbf{b}$ uz rječnik $\mathbf{D}$ je vektor $\mathbf{x} \in \mathbb{R}^m$ takav da vrijedi $\mathbf{b} = \mathbf{D}\mathbf{x}$ te da je vektor $\mathbf{x}$ "što rjeđi" u smislu da ima što više ne-nul komponenti. Pretpostavimo da rijetkost vektora mjerimo nekom funkcijom $J : \mathbb{R}^m \to \mathbb{R}$, tada zahtjev rijetkosti vektora $\mathbf{x}$ možemo zapisati kao težnju $J(\mathbf{x}) \to min$.

1.1 Optimizacija konveksnih funkcija

Uočimo da prethodno neformalno opisani problem pronalaska odgovarajuće rijetke reprezentacije za dani signal uz poznati rječnik možemo zapisati kao optimizacijski problem minimizacije funkcije J uz linearne uvjete

$$(P_J): \quad \min_{\mathbf{x}} J(\mathbf{x}) \quad \text{t.d.} \quad \mathbf{b} = \mathbf{D}\mathbf{x}. \tag{1.1}$$

Skup vektora koji zadovoljavaju uvjet problema $\{\mathbf{x} : \mathbf{b} = \mathbf{D}\mathbf{x}\}$ nazivamo dopustivim skupom. Ukoliko je funkcija J strogo konveksna i dopustivi skup neprazan, ovaj problem ima jedinstveno rješenje. Definirajmo formalno konveksnost funkcije te iskažimo rezultat koji garantira jedinstvenost rješenja problema (P_J).

Definicija 1.1.1. *Za skup $K \subseteq \mathbb{R}^m$ kažemo da je konveksan ako $\forall \mathbf{x}_1, \mathbf{x}_2 \in K$ i $\forall t \in [0, 1]$ vrijedi $t\mathbf{x}_1 + (1 - t)\mathbf{x}_2 \in K$.*

Izraz $t\mathbf{x}_1 + (1-t)\mathbf{x}_2$ za $t \in [0,1]$ nazivamo konveksnom kombinacijom. Možemo reći da je skup K konveksan ukoliko sadrži sve konveksne kombinacije svojih elemenata. Definirajmo i konveksnu funkciju:

Definicija 1.1.2. *Za funkciju $f : K \to \mathbb{R}$ gdje je $K \subseteq \mathbb{R}^m$ kažemo da je konveksna ako $\forall \mathbf{x}_1, \mathbf{x}_2 \in K$ i $\forall t \in [0,1]$ vrijedi*

$$f\Big(t\mathbf{x}_1 + (1-t)\mathbf{x}_2\Big) \leq tf(\mathbf{x}_1) + (1-t)f(\mathbf{x}_2). \tag{1.2}$$

Iz definicije zaključujemo da je funkcija konveksna ako je područje iznad grafa funkcije, definirano s $\{(\mathbf{x}, y) : y \geq f(\mathbf{x})\}$, konveksan skup. Nadalje, kažemo da je funkcija f strogo konveksna ukoliko se u relaciji (1.2) ne postiže jednakost. Ako je funkcija $f \in C^2(K)$, konveksnost možemo karakterizirati i na sljedeći način:

Lema 1.1.3. *Neka je $f : K \to \mathbb{R}$, $f \in C^2(K)$ gdje je $K \subseteq \mathbb{R}^m$ otvoren skup. Ekvivalentno je*

1. *f je konveksna funkcija*
2. *$f(\mathbf{x}_2) \geq f(\mathbf{x}_1) + \nabla f(\mathbf{x}_1)^T(\mathbf{x}_2 - \mathbf{x}_1)$, $\forall \mathbf{x}_1, \mathbf{x}_2 \in K$*
3. *$\nabla^2 f(\mathbf{x}_1)$ je pozitivno definitna $\forall \mathbf{x}_1 \in K$.*

Ovdje ∇ označava gradijent funkcije $\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}(\mathbf{x}), ..., \frac{\partial f}{\partial x_m}(\mathbf{x})\right]^T$, a ∇^2 Hesseovu matricu $\left[\nabla^2 f(\mathbf{x})\right]_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}(\mathbf{x})$. Iz ove karakterizacije slijedi egzistencija i jedinstvenost rješenja problema (P_J) kada je J konveksna funkcija i ukoliko dopustivi skup nije prazan. Za egzistenciju je dovoljna konveksnost funkcije, dok je za jedinstvenost rješenja potrebna stroga konveksnost.

Korolar 1.1.4. *Promotrimo optimizacijski problem bez ograničenja*

$$\min f(\mathbf{x}) \quad t.d. \quad \mathbf{x} \in K \tag{1.3}$$

gdje je f strogo konveksna funkcija na $K \subseteq \mathbb{R}^m$ i K konveksan skup. Ako postoji rješenje ovog problema na K, ono je jedinstveno.

Dokaz. Pretpostavimo da su $\mathbf{x}_1, \mathbf{x}_2 \in K$ dva optimalna rješenja. Tada vrijedi

$$f(\mathbf{x}_1) = f(\mathbf{x}_2) \leq f(\mathbf{y}), \forall \mathbf{y} \in K. \tag{1.4}$$

Uzmimo sada $\mathbf{z} = \frac{\mathbf{x}_1+\mathbf{x}_2}{2}$. $\mathbf{z}$ je konveksna kombinacija elemenata iz K, stoga vrijedi $\mathbf{z} \in K$, jer je K konveksan. Sada koristeći svojstvo stroge konveksnosti funkcije f zaključujemo

$$\begin{aligned} f(\mathbf{z}) &= f\left(\frac{\mathbf{x}_1 + \mathbf{x}_2}{2}\right) \\ &< \frac{1}{2}f(\mathbf{x}_1) + \frac{1}{2}f(\mathbf{x}_2) \\ &= \frac{1}{2}f(\mathbf{x}_1) + \frac{1}{2}f(\mathbf{x}_1) = f(\mathbf{x}_1), \end{aligned} \tag{1.5}$$

što je kontradikcija. □

U nastavku podrobnije opisujemo linearni uvjet problema (P_J), a zatim se bavimo odabirom kriterijske funkcije J.

1.2 Pododređeni linearni sustav

Uvjeti minimizacijskog problema opisani su linearnim sustavom $\mathbf{Dx} = \mathbf{b}$ čija matrica ima više stupaca nego redaka, odnosno sustav ima više nepoznanica no jednadžbi. Takav sustav zovemo pododređenim i njegovo rješenje ili ne postoji, ili ih ima beskonačno mnogo. Neka je $\tilde{\mathbf{x}}$ bilo koje rješenje linearnog sustava $\mathbf{Dx} = \mathbf{b}$. Prostor rješenja pripadnog homogenog sustava $\mathbf{Dx} = \mathbf{0}$ je upravo jezgra matrice $\mathbf{D}$. Sada je skup svih rješenja sustava $\mathbf{Dx} = \mathbf{b}$ dan kao linearna mnogostrukost

$$\tilde{\mathbf{x}} + \operatorname{Ker}\mathbf{D} = \left\{\tilde{\mathbf{x}} + \mathbf{h} \ : \ \mathbf{Dh} = \mathbf{0}\right\}. \tag{1.6}$$

Primijetimo da će svaki element ove linearne mnogostrukosti uistinu biti rješenje promatranog sustava jer $\mathbf{D}(\tilde{\mathbf{x}} + \mathbf{h}) = \mathbf{D}\tilde{\mathbf{x}} + \mathbf{Dh} = \mathbf{b} + \mathbf{0} = \mathbf{b}$.

U ovom razmatranju pretpostavit ćemo da je matrica sustava $\mathbf{D}$ punog ranga kako bismo bili sigurni da rješenje postoji. Kako je $m > n$, zahtjev za punim rangom znači da je $r(\mathbf{D}) = n$, što se može protumačiti na način da stupci matrice $\mathbf{D}$ čine sustav izvodnica za prostor $\mathbb{R}^n$. Sada smo sigurni u egzistenciju rješenja danog linearnog sustava. Sljedeći korak je izabrati najbolje rješenje od njih beskonačno mnogo.

Napomenimo još da je dopustivi skup $\{\mathbf{x} : \mathbf{b} = \mathbf{Dx}\}$ konveksan: Neka su $\mathbf{x}_1$ i $\mathbf{x}_2$ takvi da vrijedi $\mathbf{b} = \mathbf{Dx}_1$ i $\mathbf{b} = \mathbf{Dx}_2$. Neka je $t \in [0, 1]$ proizvoljan. Definirajmo $\tilde{\mathbf{x}} = t\mathbf{x}_1 + (1-t)\mathbf{x}_2$ i računamo $\mathbf{D}\tilde{\mathbf{x}} = \mathbf{D}\Big(t\mathbf{x}_1 + (1-t)\mathbf{x}_2\Big) = t\mathbf{Dx}_1 + (1-t)\mathbf{Dx}_2 = t\mathbf{b} + (1-t)\mathbf{b} = \mathbf{b}$.

1.3 Odabir kriterijske funkcije

Kriterijska funkcija je funkcija J koju minimiziramo u problemu (P_J). Ona nam pomaže odabrati najpogodnije rješenje s obzirom na tražena svojstva iz prostora svih rješenja. Funkcije koje ćemo promatrati su norme.

Definicija 1.3.1. *Norma na vektorskom prostoru X nad poljem $\mathbb{F}$ je preslikavanje $\|\cdot\| : X \to \mathbb{R}$ sa sljedećim svojstvima:*

1. $\|\mathbf{x}\| \geq 0$, $\forall \mathbf{x} \in X$;
2. $\|\mathbf{x}\| = 0 \iff \mathbf{x} = 0$;
3. $\|\alpha\mathbf{x}\| = |\alpha| \, \|\mathbf{x}\|$, $\forall \alpha \in \mathbb{F}$, $\forall \mathbf{x} \in X$;
4. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$, $\forall \mathbf{x}, \mathbf{y} \in X$.

U potrazi za pogodnom kriterijskom funkcijom orijentirat ćemo se na ℓ_p-norme koje označavamo s $\|\cdot\|_p$, a za $\mathbf{x} \in \mathbb{R}^m$ definirane su s

$$\|\mathbf{x}\|_p = \sqrt[p]{\sum_{i=1}^{m} |x_i|^p}. \tag{1.7}$$

Važno je napomenuti da ℓ_p-norme zadovoljavaju sva svojstva iz definicije 1.3.1, te su stoga i norme u formalnom smislu, samo za $p \geq 1$, dok za $0 < p < 1$ ovako definirane funkcije ne zadovoljavaju svojstvo 4. iz prethodne definicije. Unatoč tome, radi jednostavnosti u nastavku sve funkcije definirane relacijom (1.7) nazivamo normama vodeći računa o ovoj opasci. Na slici 1.1 dan je grafički prikaz jediničnih ℓ_p-sfera za različite vrijednosti p.

1.3.1 ℓ_2-norma

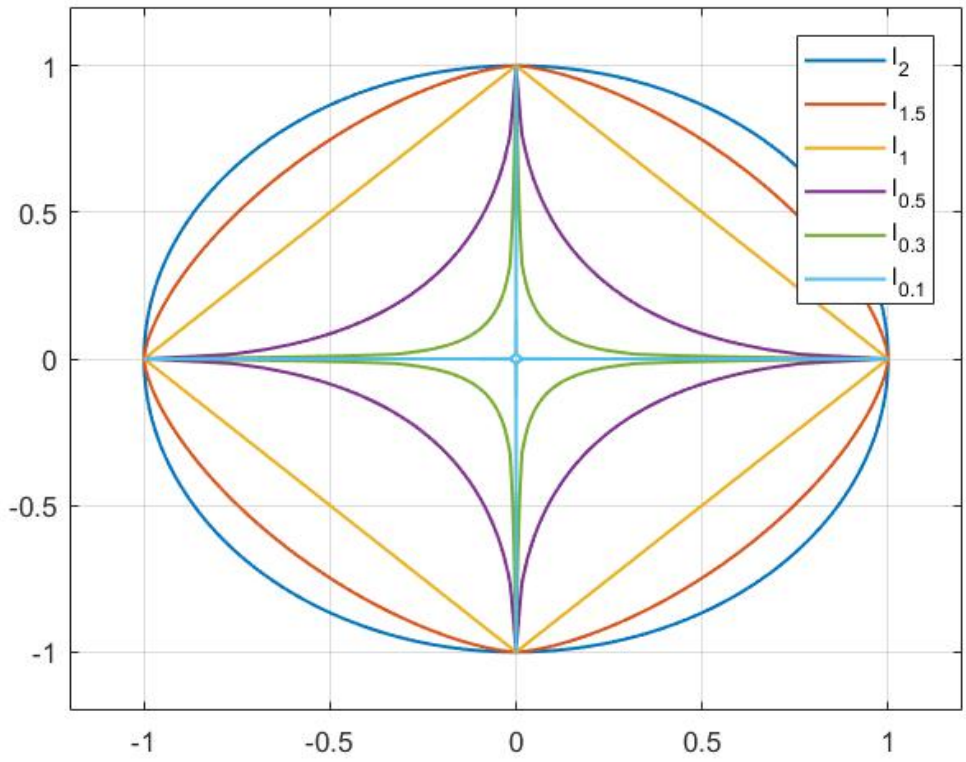
Na putu prema rijetkom rješenju, prva kriterijska funkcija koja se prirodno nameće je ℓ_2-norma, odnosno njen kvadrat, radi jednostavnijeg računa. Za popularnost ℓ_2-norme zaslužni su njena jednostavnost i interpretabilnost, ali najvećim dijelom njena konveksnost. ℓ_2-norma je strogo konveksna funkcija, stoga rješenje problema

$$(P_2): \quad \min_{\mathbf{x}} \|\mathbf{x}\|_2^2 \quad \text{t.d.} \quad \mathbf{b} = \mathbf{D}\mathbf{x}. \tag{1.8}$$

postoji i jedinstveno je, kao što smo pokazali u korolaru 1.1.4. Ovo je vrlo dobro poznat optimizacijski problem čije rješenje postoji u zatvorenoj formi i dano je s

$$\hat{\mathbf{x}}_{opt} = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{b}. \tag{1.9}$$

Izraz s desne strane prepoznajemo - to je Moore-Penroseov generalizirani inverz čiju formalnu definiciju dajemo u nastavku:



Slika 1.1: Jedinične ℓ_p-sfere za različite p u $\mathbb{R}^2$.

Definicija 1.3.2. *Za matricu* $\mathbf{D} \in \mathbb{R}^{n\times m}$ *postoji jedinstvena matrica* $\mathbf{D}^+ \in \mathbb{R}^{m\times n}$ *koja zadovoljava sljedeća svojstva*

1. $\mathbf{D}\mathbf{D}^+\mathbf{D} = \mathbf{D}$
2. $\mathbf{D}^+\mathbf{D}\mathbf{D}^+ = \mathbf{D}^+$
3. $(\mathbf{D}\mathbf{D}^+)^T = \mathbf{D}\mathbf{D}^+$ *(*$\mathbf{D}\mathbf{D}^+$ *je simetrična matrica)*
4. $(\mathbf{D}^+\mathbf{D})^T = \mathbf{D}^+\mathbf{D}$ *(*$\mathbf{D}^+\mathbf{D}$ *je simetrična matrica).*

Matricu $\mathbf{D}^+$ *tada nazivamo Moore-Penroseovim generaliziranim inverzom matrice* $\mathbf{D}$*.*

Generalizirani invez postoji za bilo koju $m \times n$ matricu. Ukoliko matrica ima puni rang po recima, tj. ako je $r(\mathbf{D}) = \min\{m, n\} = m$, kao u našem slučaju, tada se $\mathbf{D}^+$ može izraziti kao

$$\mathbf{D}^+ = \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1} \tag{1.10}$$

što nazivamo desnim generaliziranim invezrom jer vrijedi $\mathbf{D}\mathbf{D}^+ = \mathbf{I}$. Spomenimo da postoji i lijevi generalizirani inverz $\mathbf{D}^+ = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T$ za kojeg vrijedi $\mathbf{D}^+\mathbf{D} = \mathbf{I}$ u slučaju da matrica ima puni stupčani rang. Sada rješenje problema (P_2) možemo zapisati pomoću desnog Moore-Penroseovog generaliziranog inverza kao

$$\hat{\mathbf{x}}_{opt} = \mathbf{D}^+\mathbf{b}. \tag{1.11}$$

Osim ℓ_2-norme, postoje mnoge druge strogo konveksne funkcije J uz koje optimizacijski problem (P_J) ima jedinstveno rješenje. Takve su primjerice sve ℓ_p-norme, gdje je $p \geq 1$ definirane s (1.7).

1.3.2 ℓ_1-norma

Od posebnog nam je interesa slučaj $p = 1$ zbog toga što ℓ_1-norma vodi na rjeđa rješenja. Problem kojeg ovdje promatramo je

$$(P_1): \quad \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{t.d.} \quad \mathbf{b} = \mathbf{D}\mathbf{x}. \tag{1.12}$$

Nadalje, ℓ_1-norma je konveksna (ali ne strogo konveksna) funkcija, stoga znamo da rješenje optimizacijskog problema postoji, no nije nužno jedinstveno. Međutim ipak možemo nešto zaključiti o rasprostranjenosti rješenja problema (P_1) u prostoru.

Lema 1.3.3. *Skup rješenja problema (P_1) ograničen je i konveksan skup.*

Dokaz. Neka su $\mathbf{x}_1$ i $\mathbf{x}_2$ dva rješenja problema (P_1). Označimo minimum funkcije s ν_{min},

$$\nu_{min} = \min_{\substack{\mathbf{x}, \\ \mathbf{b}=\mathbf{D}\mathbf{x}}} \|\mathbf{x}\|_1. \tag{1.13}$$

Kako su oba rješenja optimalna, to znači da vrijedi

$$\|\mathbf{x}_1\|_1 = \|\mathbf{x}_2\|_1 = \nu_{min} \tag{1.14}$$

iz čega zaključujemo da je skup rješenja ograničen. Uzmimo sada njihovu konveksnu kombinaciju i pogledajmo njenu ℓ_1-normu. Neka je $t \in [0, 1]$. Koristeći svojstvo konveksnosti ℓ_1-norme i prethodnu jednakost dobivamo

$$\begin{aligned} \|t\mathbf{x}_1 + (1-t)\mathbf{x}_2\|_1 &\leq t\|\mathbf{x}_1\|_1 + (1-t)\|\mathbf{x}_2\|_1 \\ &= t\|\mathbf{x}_1\|_1 + (1-t)\|\mathbf{x}_1\|_1 \\ &= \|\mathbf{x}_1\|_1 \\ &= \nu_{min}, \end{aligned}$$

a s obzirom na to da $\mathbf{x}_1$ minimizira ℓ_1-normu, dobivamo i jednakost

$$\|t\mathbf{x}_1 + (1-t)\mathbf{x}_2\|_1 = \nu_{min}. \tag{1.15}$$

Zaključno, proizvoljna konveksna kombinacija dvaju rješenja problema (P_1) je i sama rješenje, stoga je skup rješenja konveksan.

□

Označimo sada skup rješenja problema (P_1) sa $S = \{\mathbf{x} \in \mathbb{R}^m : \mathbf{b} = \mathbf{D}\mathbf{x}, \ \ \|\mathbf{x}\|_1 = \nu_{min}\}$, gdje je ν_{min} definiran s (1.13). Prije sljedećeg rezultata navodimo definiciju nosača vektora.

Definicija 1.3.4. *Neka je* $\mathbf{x} \in \mathbb{R}^n$. *Nosač vektora* $\mathbf{x}$ *definiran je sa*

$$supp(\mathbf{x}) = \{i \in \mathbb{N} : 1 \leq i \leq n \ \ \& \ \ x_i \neq 0\}.$$

Lema 1.3.5. *Postoji* $\mathbf{x}_s \in S$ *koji ima najviše n elemenata različitih od nule.*

Dokaz. Neka je $\mathbf{x}_{opt} \in S$ jedno pronađeno rješenje koje ima $k > n$ ne-nul elemenata. Kako je $k > n$, zaključujemo da je skup odabranih k atoma iz rječnika $\mathbf{D}$ linearno zavisan. Označimo odabrane atome s $d_{i_1}, d_{i_2}, ..., d_{i_k}$, tj. $\text{supp}(\mathbf{x}_{opt}) = \{i_1, i_2, ..., i_k\}$. Tada postoji vektor $\mathbf{h} \in \mathbb{R}^n$ takav da vrijedi $\text{supp}(\mathbf{h}) \subseteq \text{supp}(\mathbf{x}_{opt})$ i

$$h_{i_1}d_{i_1} + h_{i_2}d_{i_2} + ... + h_{i_k}d_{i_k} = 0, \tag{1.16}$$

odnosno barem jedan atom od odabranih možemo prikazati kao linearnu kombinaciju preostalih odabranih atoma. Bez smanjenja općenitosti možemo pretpostaviti da je to posljednji atom d_{i_k}. Sada postoji netrivijalan vektor $\alpha \in \mathbb{R}^{k-1}$ takav da vrijedi

$$d_{i_k} = \alpha_1 d_{i_1} + \alpha_2 d_{i_2} + ... + \alpha_{k-1} d_{i_{k-1}}. \tag{1.17}$$

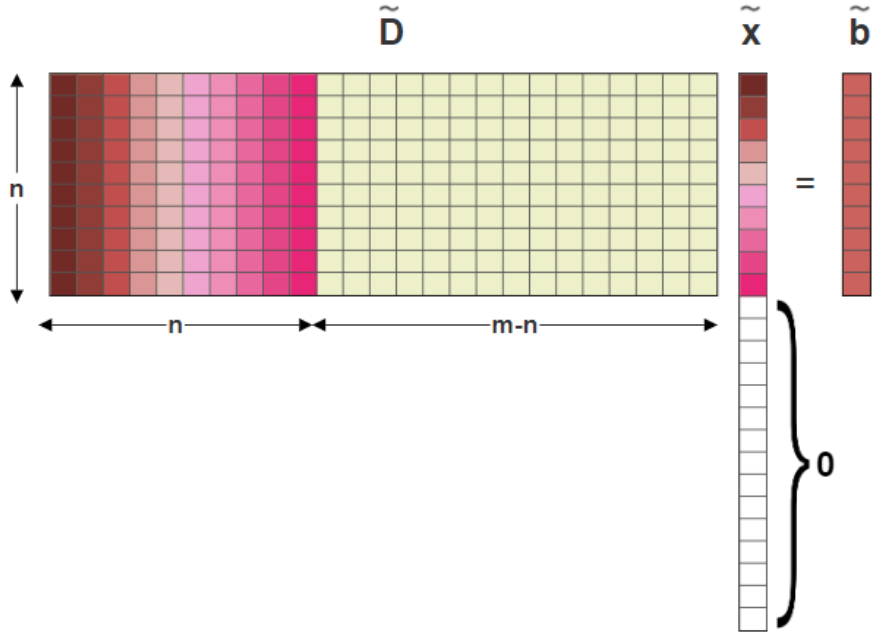
Vektor $\mathbf{h}$ sada možemo modificirati

$$\mathbf{h}_\alpha = [h_1 + \alpha_1, h_2 + \alpha_2, ..., h_{k-1} + \alpha_{k-1}, 0], \tag{1.18}$$

a jednakost (1.16) ostaje zadovoljena. Vidimo da je vektor $\mathbf{h}_\alpha$ rjeđi od vektora $\mathbf{h}$, tj. $\text{supp}(\mathbf{h}_\alpha) \subset \text{supp}(\mathbf{h}) \subseteq \text{supp}(\mathbf{x}_{opt})$. Ovaj postupak ponavljamo sve dok je $k > n$ i nakon $k - n$ koraka dolazimo do optimalnog rješenja koje ima točno n ne-nul elemenata. □

Alternativno, dokaz smo mogli provesti i na sljedeći način: pronađemo n linearno nezavisnih stupaca u rječniku $\mathbf{D}$. Permutirajući stupce dovodimo linearno nezavisne stupce na prvih n pozicija u rječniku i tako permutiran rječnik označimo s $\tilde{\mathbf{D}}$. Istu permutaciju primijenimo i na vektor $\mathbf{x}$ i dobivamo $\tilde{\mathbf{x}}$ Označimo sada s $\tilde{\mathbf{D}}_n$ $n \times n$ podmatricu matrice $\tilde{\mathbf{D}}$ dobivenu uzimanjem njenih prvih n stupaca. Tako dobivena matrica $\tilde{\mathbf{D}}$ je regularna jer je

D punog retčanog ranga, a odabranih n stupaca su linearno nezavisni. Zbog toga postoji jedinstven netrivijalan vektor $\tilde{\mathbf{x}}_n \in \mathbb{R}^n$ takav da vrijedi $\tilde{\mathbf{D}}_n\tilde{\mathbf{x}}_n = \mathbf{b}$. S obzirom na netrivijalnost, vektor $\tilde{\mathbf{x}}_n$ ima najmanje jednu, a najviše n ne-nul komponenti. Postavimo $\tilde{\mathbf{x}}_n$ na prvih n komponenti vektora $\tilde{\mathbf{x}}$, a preostalih $m-n$ komponenti postavimo na 0. Sada vrijedi $\tilde{\mathbf{D}}\tilde{\mathbf{x}} = \mathbf{b}$. Primijenimo inverznu permutaciju stupaca i dobivamo $\mathbf{Dx} = \mathbf{b}$. Ilustracija ovog postupka prikazana je na slici 1.2.



Slika 1.2: Ilustracija tvrdnje leme 1.3.5 u smislu permutiranih stupaca.

Vidimo da koristeći ℓ_1-normu dobivamo rjeđa rješenja nego koristeći ℓ_2-normu za kriterijsku funkciju. Nastavit ćemo slijediti ovaj trag promatrajući općenite ℓ_p-norme za $p < 1$.

1.3.3 ℓ_p-norme, $0 < p < 1$

Kao što smo već napomenuli, u ovom slučaju funkcije koje promatramo, ℓ_p-norme za $0 < p < 1$, nisu formalne norme jer ne zadovoljavaju posljednje svojstvo iz definicije 1.3.1 (nejednakost trokuta). Unatoč tome, radi jednostavnosti ćemo ih zvati normama i želimo vidjeti vode li na još rjeđa rješenja.
Neka je $q < p$. Potražimo vektor koji je normiran u ℓ_p-normi i takav da je njegova ℓ_q-norma najmanja moguća, tj. tražimo

$$\min_{\mathbf{x}} \|\mathbf{x}\|_q^q \quad \text{t.d.} \quad \|\mathbf{x}\|_p^p = 1. \tag{1.19}$$

Lema 1.3.6. *Za svaki par normi ℓ_p i ℓ_q gdje je $q < p$ rješenje problema (1.19) postiže se za najrjeđi vektor* **x**.

Dokaz. Pretpostavimo da vektor **x** ima a elemenata različitih od nule i bez smanjenja općenitosti pretpostavimo da su sve ne-nul vrijednosti pozitivne. Pretpostavimo da se a ne-nul vrijednosti nalaze na indeksima $1, ..., a$ (vodećih a vrijednosti). Definiramo Lagrangeovu funkciju s Lagrangeovim multiplikatorom λ

$$\mathcal{L}(\mathbf{x}) = \|\mathbf{x}\|_q^q + \lambda\Big(\|\mathbf{x}\|_p^p - 1\Big) = -\lambda + \sum_{k=1}^{a}\Big(x_k^q + \lambda x_k^p\Big). \tag{1.20}$$

Tražimo stacionarne točke Lagrangeove funkcije. Neka je $i \in \{1, ..., a\}$ proizvoljan.

$$\nabla_{x_i}\mathcal{L}(\mathbf{x}) = q x_i^{q-1} + \lambda p x_i^{p-1} = 0 \quad \Rightarrow \quad x_i^{p-q} = -\frac{1}{\lambda}\frac{q}{p} \quad \Rightarrow \quad x_i = c \in \mathbb{R}, \tag{1.21}$$

a kako je i bio proizvoljan, to mora vrijediti $\forall i \in \{1, ..., a\}$, odnosno sve ne-nul vrijednosti moraju biti jednake. Sada koristimo uvjet normiranosti u ℓ_p-normi i dobivamo

$$\|\mathbf{x}\|_p^p = \sum_{k=1}^{a} x_k^p = a c^p = 1 \quad \Rightarrow \quad c = a^{-\frac{1}{p}}, \tag{1.22}$$

tj. $x_i = a^{-1/p}$, $\forall i \in \{1, ..., a\}$. Izračunajmo ℓ_q-normu ovako dobivenog vektora **x**

$$\|\mathbf{x}\|_q^q = \sum_{k=1}^{a} a^{-\frac{q}{p}} = a^{1-\frac{q}{p}}. \tag{1.23}$$

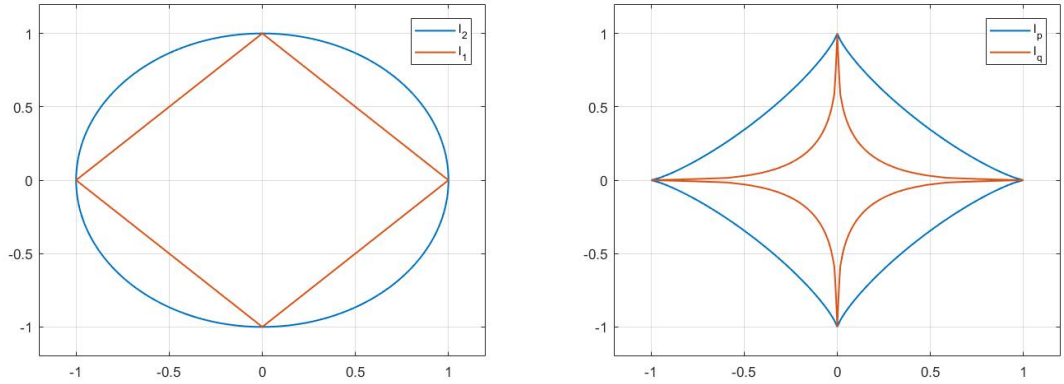
Kako je $q < p$, $1 - \frac{q}{p} > 0$, minimum ℓ_q-norme postiže se za $a = 1$, tj. onda kada vektor **x** ima samo jednu komponentu različitu od nule. □

Geometrijski ovu tvrdnju možemo opisati na sljedeći način: Jedinična ℓ_p-sfera oko ishodišta u prostoru $\mathbb{R}^m$ predstavlja dopustivi skup rješenja za problem (1.19). "Napušimo" sada ℓ_q-sferu oko ishodišta sve dok ne dotakne ℓ_p-sferu. Točke gdje se ℓ_p- i ℓ_q-sfere sijeku su rješenja promatranog problema. Na slici 1.3 vidimo kako se to događa upravo na osima, gdje je samo jedna komponenta vektora različita od nule .

Promotrimo sada generalizaciju problema (P_1) uz općenitu ℓ_p-normu, $p < 1$, kao kriterijsku funkciju.

$$(P_p): \quad \min_{\mathbf{x}} \|\mathbf{x}\|_p^p \quad \text{t.d.} \quad \mathbf{b} = \mathbf{D}\mathbf{x} \tag{1.24}$$

i pokušajmo provesti istu geometrijsku analizu. Sada je dopustivi skup definiran linearnim uvjetom i određuje hiperravninu u prostoru. Rješenja gornjeg problema tražimo u točkama gdje hiperravnina siječe ℓ_p-sferu. Slutimo da će, kao u gornjem primjeru, rijetka rješenja



(a) $p = 2, q = 1$ (b) $p = 0.8, q = 0.4$

Slika 1.3: Geometrijska interpretacija problema 1.19 u $\mathbb{R}^2$.

ležati na koordinatnim osima.
Na slici 1.4 dana je geometrijska interpretacija ove slutnje, tj. ilustracija rješenja problema (P_p) za različite vrijednosti $p \in \{0.7, 1, 1.5, 2\}$ gdje je $\mathbf{x} \in \mathbb{R}^3$. Primijetimo da se smanjujući p točka dirališta hiperravnine, koja predstavlja linearan uvjet, i ℓ_p-sfere približava koordinatnim osima. Rješenje u ℓ_2- i $\ell_{1.5}$- normi na svima trima komponentama ima vrijednosti različite od nule, dok rješenje već u ℓ_1- pa onda i u $\ell_{0.7}$-normi leži na koordinatnoj osi stoga ima samo jednu ne-nul komponentu. Time smo ilustrirali težnju k rijetkim rješenjima kada $p \to 0$.

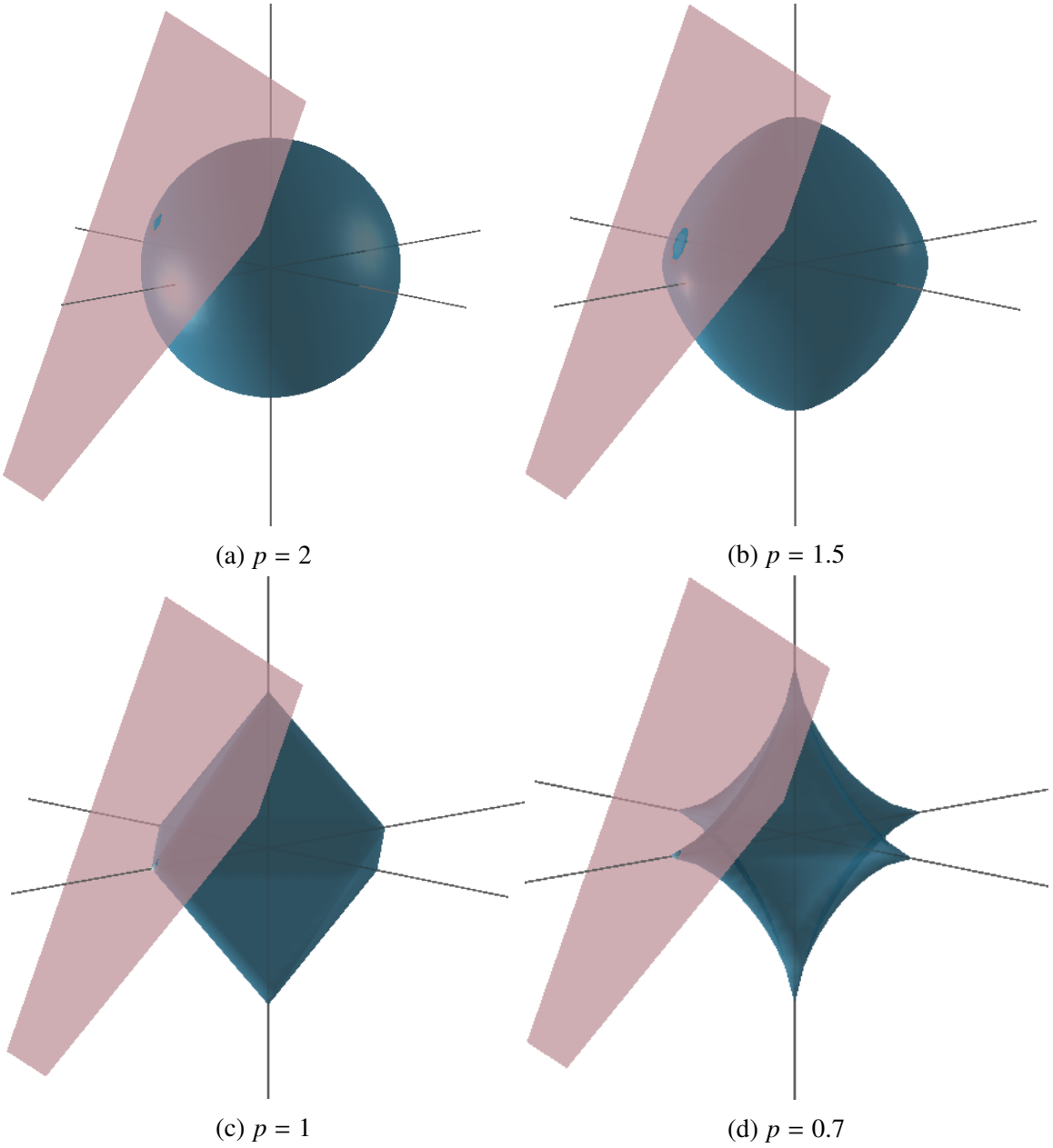
Postoje mnoge druge funkcije koje, poput ℓ_p-normi, vode na rijetkost. Primjerice slaba ℓ_p-norma za $0 < p \leq 1$

$$\|\mathbf{x}\|_{p_w}^p = \sup_{\epsilon>0} N(\epsilon, \mathbf{x}) \cdot \epsilon^p, \tag{1.25}$$

gdje je $N(\epsilon, \mathbf{x})$ broj komponenti vektora $\mathbf{x}$ koji su veći od ϵ. Nadalje, funkcije oblika

$$J(\mathbf{x}) = \sum_i \rho(x_i), \tag{1.26}$$

gdje je ρ simetrična i rastuća te takva da joj je derivacija rastuća za $x \geq 0$ također vode na rijetka rješenja. Često korišteni primjeri ovakvih funkcija $\rho(x) = 1 - \exp(|x|)$, $\rho(x) = \ln(1 + |x|)$ te $\rho(x) = \frac{|x|}{1+|x|}$.
Unatoč brojnim funkcijama koje vode na rijetkost, radi jednostavnosti analize i računa, fokusiramo se na ℓ_p-norme i pokušavamo doći do pogodne mjere rijetkosti.



(a) $p = 2$ (b) $p = 1.5$

(c) $p = 1$ (d) $p = 0.7$

Slika 1.4: Geometrijska interpretacija problema (P_p) u $\mathbb{R}^3$.

1.3.4 ℓ_0-norma

Nastavljajući u istom trendu, tj. smanjujući p za ℓ_p-norme, dolazimo do pojma ℓ_0-norme.

$$\|\mathbf{x}\|_0 = \lim_{p\to 0} \|\mathbf{x}\|_p^p = \lim_{p\to 0} \sum_{k=1}^{m} |x_k|^p = \#\{i : x_i \neq 0\} \tag{1.27}$$

odnosno ℓ_0-norma vektora je upravo broj njegovih ne-nul elemenata iz čega zaključujemo da je ℓ_0-norma vrlo pogodna za traženje rijetkih rješenja. Njenom minimizacijom pod određenim uvjetima dolazimo do najrjeđeg rješenja. Za razliku od ℓ_p-normi za $0 < p < 1$ opisanih u prethodnom potpoglavlju, ℓ_0-norma zadovoljava nejednakost trokuta, no ne zadovoljava svojstvo 3. iz definicije 1.3.1 tj. homogenost ($\|\alpha\mathbf{x}\|_0 = \|\mathbf{x}\|_0 \neq |\alpha|\|\mathbf{x}\|_0$).
ℓ_0-norma ima i svoju slabu verziju koja je u primjeni ponekad pogodnija. Omogućuje nam da zahtjev rijetkosti relaksiramo na način da sve vrijednosti koje su manje od zadane tolerancije smatramo nulama. Za zadanu toleranciju $\delta > 0$ možemo je definirati s

$$\|\mathbf{x}\|_{0,\delta} = \#\{x_i : |x_i| > \delta\}. \tag{1.28}$$

Zapišimo optimizacijski problem (P_J) s ℓ_0-normom kao kriterijskom funkcijom:

$$(P_0): \quad \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{t.d.} \quad \mathbf{b} = \mathbf{D}\mathbf{x}. \tag{1.29}$$

Ono što tražimo je najrjeđe rješenje linearnog sustava $\mathbf{b} = \mathbf{D}\mathbf{x}$, to jest za dani signal $\mathbf{b}$ tražimo rijetki vektor $\mathbf{x}$ koji ga najbolje opisuje uz zadani rječnik $\mathbf{D}$. Problem (P_0) glavni je problem kojim ćemo se baviti u ovom radu.

Varijanta problema (P_0) izvedena je uz pretpostavku da u signalu kojeg dobivamo postoji šum ε. Ako s $\mathbf{b}_\varepsilon$ označimo signal $\mathbf{b}$ sa šumom ε, problem (P_0) možemo zapisati kao

$$(P_0^{\varepsilon}): \quad \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{t.d.} \quad \|\mathbf{b}_\varepsilon - \mathbf{D}\mathbf{x}\|_2 \leq \varepsilon. \tag{1.30}$$

Problem s kojim se susrećemo je jedinstvenost rješenja problema (P_0) i (P_0^{ε}). Egzistencija rješenja nije problem jer je ℓ_0-norma pozitivna i konačna funkcija, ali problem je možemo li tvrditi da je pronađeni minimum i globalni minimum. Za razliku od neprekidnog problema (P_2) koji je strogo konveksan i čije je rješenje jedinstveno te neprekidnog problema (P_1) koji je konveksan, probleme (P_0) i (P_0^{ε}) ne možemo analizirati na analogan način jer ℓ_0-norma nije neprekidna i poprima diskretne vrijednosti. Također, oba problema su NP-teški, kao što je pokazano u [5], stoga ćemo se za traženje rješenja u praksi oslanjati na razne algoritme, ali prvo analizirajmo jedinstvenost rješenja u teorijskom pogledu.

Poglavlje 2

Rješenje problema P_O

Izabravši ℓ_0-normu kao kriterijsku funkciju, glavni problem kojim se bavimo je problem (P_0). Kao što smo napomenuli u zaključku prethodnog poglavlja, pitanja na koja želimo odgovoriti su jedinstvenost i pouzdanost rješenja. Prije nego što uronimo u formalnu teorijsku analizu problema (P_0), analizirat ćemo specifičan slučaj problema (P_0) kada je matrica $\mathbf{D}$ nastala konkatenacijom dviju ortogonalnih matrica. Motivacija iza ovakvog pristupa je sljedeća: neka je $\mathbf{Q}$ ortogonalna matrica. Tada vrijedi $\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ pa je $\mathbf{Q}$ regularna s inverzom $\mathbf{Q}^T$ i linearni sustav $\mathbf{Q}\mathbf{x} = \mathbf{b}$ ima jedinstveno rješenje. Ovakvi problemi nisu nam od interesa, upravo zbog jedinstvenosti rješenja, pa pokušajmo stvoriti rječnik spajanjem dviju ortogonalnih matrica. Na taj način dobivamo pododređen sustav čija nam je forma poznata i koji ima beskonačno mnogo rješenja među kojima možemo tražiti najrjeđe.

2.1 Sparene ortogonalne matrice

Neka su $\mathbf{\Psi}, \mathbf{\Phi} \in \mathbb{R}^{n\times n}$ dvije različite ortogonalne matrice i neka je $\mathbf{D} = [\mathbf{\Psi}, \mathbf{\Phi}] \in \mathbb{R}^{n\times 2n}$. Linearni uvjet problema (P_0) sada glasi

$$[\mathbf{\Psi}, \mathbf{\Phi}]\mathbf{x} = \mathbf{b} \tag{2.1}$$

gdje je $\mathbf{x} \in \mathbb{R}^{2n}$ te $\mathbf{b} \in \mathbb{R}^n$ netrivijalan vektor. Tada se $\mathbf{b}$ može reprezentirati kao linearna kombinacija stupaca iz $\mathbf{\Psi}$ ili stupaca iz $\mathbf{\Phi}$, to jest postoje jedinstveni vektori α i β takvi da vrijedi

$$\mathbf{b} = \mathbf{\Psi}\alpha = \mathbf{\Phi}\beta. \tag{2.2}$$

Nadalje, za $\mathbf{\Psi}$ i $\mathbf{\Phi}$ možemo tvrditi, s obzirom na njihovu udaljenost: ili je α rijedak vektor, ili je β rijedak vektor. Za mjerenje udaljenosti dviju matrica uvodimo pojam međusobne koherencije kao maksimalan skalarni produkt stupaca matrice $\mathbf{\Psi}$ sa stupcima matrice $\mathbf{\Phi}$.

Definicija 2.1.1. *Za proizvoljan par ortonormiranih baza* $\mathbf{\Psi}$, $\mathbf{\Phi}$ *takve da je* $\mathbf{D} = [\mathbf{\Psi}, \mathbf{\Phi}]$ *definiramo međusobnu koherenciju kao*

$$\mu(\mathbf{D}) = \max_{1 \leq i,j, \leq n} |\psi_i^T \phi_j|. \tag{2.3}$$

U nastavku navodimo rezultat o granicama međusobne koherencije.

Lema 2.1.2. *Međusobna koherencija dviju ortonormiranih matrica* $\mathbf{\Psi}$, $\mathbf{\Phi}$ *koje čine matricu* $\mathbf{D}$ *zadovoljava nejednakosti*

$$\frac{1}{\sqrt{n}} \leq \mu(\mathbf{D}) \leq 1. \tag{2.4}$$

Dokaz. Vrijedi

$$(\mathbf{\Psi}^T\mathbf{\Phi})^T(\mathbf{\Psi}^T\mathbf{\Phi}) = \mathbf{\Phi}^T(\mathbf{\Psi}\mathbf{\Psi}^T)\mathbf{\Phi} = \mathbf{\Phi}^T\mathbf{I}\mathbf{\Phi} = \mathbf{\Phi}^T\mathbf{\Phi} = \mathbf{I} \tag{2.5}$$

pa je matrica $\mathbf{\Psi}^T\mathbf{\Phi}$ ortogonalna. Označimo s $U = \mathbf{\Psi}^T\mathbf{\Phi}$. Po definiciji matrične 2-norme imamo

$$\|\mathbf{U}\|_2^2 = \sup_{\|\mathbf{x}\|_2=1} \|\mathbf{U}\mathbf{x}\|_2^2 = \sup_{\|\mathbf{x}\|_2=1} (\mathbf{U}\mathbf{x})^T\mathbf{U}\mathbf{x} = \sup_{\|\mathbf{x}\|_2=1} \mathbf{x}^T\mathbf{U}^T\mathbf{U}\mathbf{x} = \sup_{\|\mathbf{x}\|_2=1} \mathbf{x}^T\mathbf{x} = \sup_{\|\mathbf{x}\|_2=1} \|\mathbf{x}\|_2^2 = 1. \tag{2.6}$$

Sada računamo

$$\mu(\mathbf{D}) = \max_{1 \leq i,j \leq n} |u_{i,j}| \leq \|\mathbf{U}\|_2 = 1 \tag{2.7}$$

što je upravo tražena gornja ograda.
Stupci i retci ove matrice su ortonormirani vektori, stoga imamo

$$\sum_{i=1}^{n} u_{i,j}^2 = 1, \quad \forall j = 1, ..., n \tag{2.8}$$

Kada bi svi sumandi bili manji od $\frac{1}{n}$, suma kvadrata bi bila manja od 1 što je kontradikcija s prethodnom jednakošću. Iz toga zaključujemo da vrijedi

$$\frac{1}{\sqrt{n}} \leq \max_{1 \leq i \leq n} |u_{i,j}| \leq \max_{1 \leq i,j, \leq n} |\psi_i^T \phi_j| = \mu(\mathbf{D}). \tag{2.9}$$

□

Teorem 2.1.3. *Za proizvoljan par ortonormiranih baza* $\mathbf{\Psi}$, $\mathbf{\Phi}$ *takve da je* $\mathbf{D} = [\mathbf{\Psi}, \mathbf{\Phi}]$ *s međusobnom koherencijom* $\mu(\mathbf{D})$ *i za proizvoljan netrivijalan vektor* $\mathbf{b} \in \mathbb{R}^n$ *s reprezentacijama* α *i* β *respektivno, vrijedi sljedeća nejednakost*

$$\|\alpha\|_0 + \|\beta\|_0 \geq \frac{2}{\mu(\mathbf{D})}. \tag{2.10}$$

Dokaz. Bez smanjenja općenitosti možemo pretpostaviti da je $\|\mathbf{b}\|_2 = 1$. Kako je $\mathbf{b} = \mathbf{\Psi}\alpha = \mathbf{\Phi}\beta$, koristeći definiciju međusobne koherencije dviju ortonormiranih baza imamo

$$\begin{aligned} 1 &= \mathbf{b}^T\mathbf{b} \\ &= \alpha^T\mathbf{\Psi}^T\mathbf{\Phi}\beta \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\beta_j\psi_i^T\phi_j \\ &\leq \mu(\mathbf{D})\cdot\sum_{i=1}^{n}\sum_{j=1}^{n}|\alpha_i||\beta_j| = \mu(\mathbf{D})\cdot\|\alpha\|_1\|\beta\|_1. \end{aligned} \tag{2.11}$$

Koristeći aritmetičko-geometrijsku nejednakost $\sqrt{ab} \leq \frac{a+b}{2}, \forall a, b \geq 0$ imamo

$$\|\alpha\|_1\|\beta\|_1 \geq \frac{1}{\mu(\mathbf{D})} \quad \Rightarrow \quad \|\alpha\|_1 + \|\beta\|_1 \geq \frac{2}{\sqrt{\mu(\mathbf{D})}}, \tag{2.12}$$

što možemo interpretirati kao princip neodređenosti u smislu da ne mogu oba vektora α i β imati proizvoljno malu ℓ_1-normu.

Potražimo sada reprezentaciju α za koju vrijedi $\|\alpha\|_2 = 1$ i koja ima A ne-nul elemenata, a koja ima najveću ℓ_1-normu - tražimo rješenje sljedećeg optimizacijskog problema

$$\max_{\alpha}\|\alpha\|_1 \quad \text{t.d.} \quad \|\alpha\|_2 = 1 \quad \& \quad \|\alpha\|_0 = A. \tag{2.13}$$

Pretpostavimo da ovaj problem ima rješenje koje je dano s $g(A) = g(\|\alpha\|_0)$. Analogno tome, postavimo isti optimizacijski problem za β B ne-nul elemenata i pretpostavimo da je rješenje dano s $g(B) = g(\|\beta\|_0)$. Koristeći relaciju (2.11) imamo

$$\frac{1}{\mu(\mathbf{D})} \leq \|\alpha\|_1\|\beta\|_1 \leq g(\|\alpha\|_0)\cdot g(\|\beta\|_0) \tag{2.14}$$

gdje smo ℓ_1-normu vektora α zamijenili s gornjom ogradom tj. rješenjem optimizacijskog problema (2.13) (analogno za β).
Bez smanjenja općenitosti sada pretpostavljamo da je A ne-nul komponenti vektora α upravo prvih A komponenti te da su sve ne-nul komponente pozitivne, dok je preostalih $n - A$ komponenti jednako nuli. Koristeći metodu Lagrangeovih multiplikatora dobivamo Langrangeovu funkciju

$$\mathcal{L}(\alpha) = \sum_{i=n}^{A}\alpha_i + \lambda\left(1 - \sum_{i=1}^{A}\alpha_i^2\right) \tag{2.15}$$

čija je derivacija po komponentama dana s

$$\frac{\partial \mathcal{L}(\alpha)}{\partial \alpha_i} = 1 - 2\lambda\alpha_i = 0 \tag{2.16}$$

iz čega zaključujemo da su optimalne vrijednosti dane s $\alpha_i = \frac{1}{2\lambda}$ te da su sve jednake. S obzirom na uvjet ℓ_2-norme dobivamo da mora vrijediti $\alpha_i = \frac{1}{\sqrt{A}}$, pa je stoga $g(A) = \frac{A}{\sqrt{A}} = \sqrt{A}$ maksimalna ℓ_0-vrijednost vektora α tj. rješenje problema (2.13). Analogno dobivamo da je $g(B) = \sqrt{B}$ te uvrstivši dobiveno u (2.14) slijedi

$$\frac{1}{\mu(\mathbf{D})} \leq \|\alpha\|_1\|\beta\|_1 \leq g(\|\alpha\|_0) \cdot g(\|\beta\|_0) = \sqrt{\|\alpha\|_0 \cdot \|\beta\|_0} \tag{2.17}$$

iz čega ponovno koristeći aritmetičko-geometrijsku nejednakost dobivamo

$$\frac{1}{\mu(\mathbf{D})} \leq \sqrt{\|\alpha\|_0 \cdot \|\beta\|_0} \leq \frac{1}{2}(\|\alpha\|_0 + \|\beta\|_0) \tag{2.18}$$

što smo i trebali dokazati. □

Teorem 2.1.3 nam govori da u slučaju da je međusobna koherenecija dviju baza mala, reprezentacije α i β ne mogu obje biti rijetke. Princip neodređenosti koristit ćemo u nastavku kako bismo dokazali jedinstvenost rijetkog rješenja problema (P_0).

Teorem 2.1.4 (Neodređenost redundantnih rješenja)**.** *Neka su* $\mathbf{x}_1$ *i* $\mathbf{x}_2$ *dva različita rješenja linearnog sustava* $\mathbf{Dx} = [\mathbf{\Psi}, \mathbf{\Phi}]\mathbf{x} = \mathbf{b}$*. Vrijedi*

$$\|\mathbf{x}_1\|_0 + \|\mathbf{x}_2\|_0 \geq \frac{2}{\mu(\mathbf{D})}, \tag{2.19}$$

to jest, $\mathbf{x}_1$ *i* $\mathbf{x}_2$ *ne mogu oba biti proizvoljno rijetka.*

Dokaz. Označimo s $\mathbf{e} = \mathbf{x}_1 - \mathbf{x}_2 \neq 0$. Vrijedi

$$\mathbf{De} = \mathbf{D}(\mathbf{x}_1 - \mathbf{x}_2) = \mathbf{Dx}_1 - \mathbf{Dx}_2 = \mathbf{b} - \mathbf{b} = 0 \tag{2.20}$$

pa je vektor $\mathbf{e} \in \operatorname{Ker}\mathbf{D}$. Označimo sada prvih n komponenti vektora $\mathbf{e}$ s $\mathbf{e}_\Psi$ te posljednjih n komponenti s $\mathbf{e}_\Phi$. Sada imamo, s obzirom na (2.20)

$$\mathbf{\Psi e}_\Psi + \mathbf{\Phi e}_\Phi = 0 \quad \Rightarrow \quad \mathbf{\Psi e}_\Psi = -\mathbf{\Phi e}_\Phi = \mathbf{y}. \tag{2.21}$$

Kako je $\mathbf{e} \neq 0$, a matrice baza $\mathbf{\Psi}$ i $\mathbf{\Phi}$ su regularne, zaključujemo da je $\mathbf{y}$ netrivijalan vektor. Sada koristimo teorem 2.1.3

$$\|\mathbf{e}\|_0 = \|\mathbf{e}_\Psi\|_0 + \|\mathbf{e}_\Phi\|_0 \geq \frac{2}{\mu(\mathbf{D})}, \tag{2.22}$$

a s obzirom na to da smo $\mathbf{e}$ definirali s $\mathbf{e} = \mathbf{x}_1 - \mathbf{x}_2$ i koristeći nejednakost trokuta za ℓ_0-normu dobivamo

$$\frac{2}{\mu(\mathbf{D})} \leq \|\mathbf{e}\|_0 = \|\mathbf{x}_1 - \mathbf{x}_2\|_0 \leq \|\mathbf{x}_1\|_0 + \|\mathbf{x}_2\|_0. \tag{2.23}$$

□

Izravna posljedica teorema 2.1.4 je sljedeći, naizgled jednostavan, rezultat koji nam garantira jedinstvenost rijetkog rješenja

Korolar 2.1.5. *Ako rješenje sustava* $\mathbf{Dx} = [\mathbf{\Psi}, \mathbf{\Phi}]\mathbf{x} = \mathbf{b}$ *ima manje od* $\frac{1}{\mu(\mathbf{D})}$ *ne-nul komponenti, tada je ono nužno najrjeđe rješenje.*

Dokaz. Dokaz je jednostavan i direktan koristeći nejednakost iz prehodnog teorema. Neka je $\mathbf{x}_1$ rješenje sustava koje ima manje od $\frac{1}{\mu(\mathbf{D})}$ ne-nul komponenti, odnosno neka vrijedi $\|\mathbf{x}_1\|_0 < \frac{1}{\mu(\mathbf{D})}$. Neka je $\mathbf{x}_2 \neq \mathbf{x}_1$ neko drugo proizvoljno rješenje sustava. Sada koristeći nejednakost (2.19) računamo ℓ_0-normu rješenja $\mathbf{x}_2$:

$$\|\mathbf{x}_2\|_0 > \frac{2}{\mu(\mathbf{D})} - \|\mathbf{x}_1\|_0 > \frac{1}{\mu(\mathbf{D})} > \|\mathbf{x}_1\|_0. \tag{2.24}$$

Rješenje $\mathbf{x}_2$ ima više ne-nul komponenti od rješenja $\mathbf{x}_1$, stoga slijedi da je $\mathbf{x}_1$ najrjeđe rješenje promatranog sustava. □

Ovaj nam rezultat daje jedinstvenost i globalnu optimalnost dobivenog rješenja za slučaj sparenih ortogonalnih matrica. Općenito, kod nekonveksnih problema ne možemo tvrditi globalnu optimalnost dobivenog rješenja, stoga ovaj rezultat ima posebnu težinu. No, pojava rječnika kao sparenja ortogonalnih matrica u praksi je rijetka, stoga ćemo u nastavku pokušati dobiti garanciju jedinstvenosti i globalne optimalnosti rješenja u općenitom slučaju za proizvoljnu matricu $\mathbf{D}$.

2.2 Općeniti slučaj

Za općenite matrice $\mathbf{D} \in \mathbb{R}^{n\times m}$ jedinstvenost rijetkog rješenja pokazat ćemo koristeći *spark* matrice, već spomenutu međusobnu koherenciju te Babel funkciju koju još zovemo i kumulativnom koherencijom, navest ćemo odnose ovih triju veličina i dati teorijsku garanciju globalne optimalnosti rješenja problema (P_0).

2.2.1 Jedinstvenost koristeći *spark* matrice

Spark matrice pojam je kojeg je prvi skovao Joseph Kruskal, stoga se po njemu još naziva i Kruskalov rang.

Definicija 2.2.1. *Spark matrice* $\mathbf{D}$ *definiramo kao najmanji broj stupaca matrice* $\mathbf{D}$ *koji su linearno zavisni.*

Uočimo razliku definicije *sparka* i ranga matrice. Rang definiramo kao najveći broj stupaca (ili redaka) matrice koji su linearno nezavisni. Nadalje, razlika je u tome što je računanje *sparka* matrice puno zahtjevniji posao od računanja ranga. Računanje *sparka* kombinatorne je prirode i eksponencijalne složenosti te je NP-težak problem što je pokazano u [10] dok je računanje ranga matrice polinomijalne složenosti (Gaussova metoda eliminacije).

Primjer 2.2.2. *Pogledajmo matricu*

$$\mathbf{A} = \begin{bmatrix} -3 & -2 & -1 & 3 & 6 \\ 0 & 3 & 5 & 9 & 0 \\ -1 & 0 & 2 & -1 & 2 \\ -2 & 2 & 1 & -8 & 4 \end{bmatrix}. \tag{2.25}$$

Nije teško izračunati njen rang - vrijedi $r(\mathbf{A}) = 3$. *Označimo s* $\mathbf{a}_i$ *i-ti stupac matrice* $\mathbf{A}$. *Uočimo da vrijedi* $\mathbf{a}_5 = -2\mathbf{a_0}$, *stoga zaključujemo* $spark(\mathbf{A}) = 2$.

Primjer 2.2.3. *Pogledajmo matricu*

$$\mathbf{A} = \begin{bmatrix} -1 & -3 & 0 & 8 & -1 & -1 & -2 \\ -3 & -3 & -2 & 8 & -7 & 11 & 10 \\ 0 & -4 & -2 & 14 & -14 & 4 & -4 \\ -2 & -2 & 0 & 4 & 2 & 2 & 4 \end{bmatrix}. \tag{2.26}$$

Uočimo da za stupce matrice $\mathbf{A}$ *vrijedi*

$$\begin{aligned} \mathbf{a}_4 &= \mathbf{a}_1 - 3\mathbf{a}_2 - \mathbf{a}_3 \\ \mathbf{a}_5 &= -2\mathbf{a}_1 + \mathbf{a}_2 + 5\mathbf{a}_3 \\ \mathbf{a}_6 &= -2\mathbf{a}_1 + \mathbf{a}_2 - 4\mathbf{a}_3 \\ \mathbf{a}_7 &= -4\mathbf{a}_1 + 2\mathbf{a}_2 - 2\mathbf{a}_3 \end{aligned} \tag{2.27}$$

dok su prva tri stupca linearno nezavisni. Zbog toga zaključujemo $r(\mathbf{A}) = 3$. *Pogledamo sada matricu* $\tilde{\mathbf{A}} = [\mathbf{a}_4, \mathbf{a}_5, \mathbf{a}_6, \mathbf{a}_7]$. *Njen rang iznosi 3, što znači da je jedan vektor matrice*

moguće prikazati kao linearnu kombinaciju preostala 3 ili manje stupaca. Računamo i dobivamo

$$\mathbf{a}_7 = -\frac{8}{5}\mathbf{a}_4 - \frac{74}{45}\mathbf{a}_5 - \frac{52}{45}\mathbf{a}_6 \tag{2.28}$$

Kako su svi dobiveni koeficijenti različiti od nule, zaključujemo da je najmanji broj linearno zavisnih stupaca u matrici $\tilde{\mathbf{A}}$ *jednak 4. Nadalje, kako su stupci iz* $\tilde{\mathbf{A}}$ *linearna kombinacija prvih triju, zaključujemo da je* $spark(\mathbf{A}) = 4$.

Spark matrice karakterizira njenu jezgru koristeći ℓ_0-normu. U tom kontekstu definiciju *sparka* možemo zapisati i kao

Definicija 2.2.4. *Spark matrice* $\mathbf{D}$ *je rješenje problema*

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad t.d. \quad \mathbf{D}\mathbf{x} = \mathbf{0}. \tag{2.29}$$

Kako je *spark* upravo minimum ℓ_0-norme po vektorima iz jezgre, mora vrijediti $\|\mathbf{x}\|_0 \geq spark(\mathbf{D})$, $\forall \mathbf{x} \in \operatorname{Ker}\mathbf{D}$. Napomenimo još koje vrijednosti *spark* može poprimiti.

Lema 2.2.5. *Za spark matrice* $\mathbf{D} \in \mathbb{R}^{n\times m}$*, gdje je* $n \neq m$*, vrijedi*

$$2 \leq spark(\mathbf{D}) \leq n + 1. \tag{2.30}$$

Dokaz. Dokaz je vrlo jednostavan i temelji se na osnovama linearne algebre. Prvo, vrijednost *sparka* mora biti pozitivna. S obzirom na definiciju *sparka* kao kardinaliteta skupa linearno zavisnih stupaca matrice $\mathbf{D}$, *spark* ne može iznositi 1 jer je skup od jednog (netrivijalnog) vektora nužno linearno nezavisan skup. Gornja granica očita je iz činjenice da je skup od $n + 1$ elemenata u prostoru $\mathbb{R}^n$ nužno linearno zavisan skup. □

Napomenimo da *spark* kvadratne matrice punog ranga nema smisla promatrati, s obzirom na to da nema linearno zavisnih stupaca, stoga u tom slučaju *spark* ili nije definiran, ili se definira kao 0. U svakom slučaju, gornja ocjena *sparka* vrijedi za matrice strogo različitih dimenzija. Sljedeći rezultat daje nam garanciju jedinstvenosti i globalne optimalnosti rješenja problema (P_0).

Teorem 2.2.6 (Jedinstvenost koristeći *spark*)**.** *Ako linearan sustav* $\mathbf{b} = \mathbf{D}\mathbf{x}$ *ima rješenje* $\mathbf{x}$ *koje zadovoljava* $\|\mathbf{x}\|_0 < spark(\mathbf{D})/2$*, tada je to rješenje nužno najrjeđe moguće.*

Dokaz. Neka je $\hat{\mathbf{x}} \neq \mathbf{x}$ drugo rješenje sustava, tj. vrijedi $\mathbf{b} = \mathbf{D}\hat{\mathbf{x}}$. Označimo kao i prije s $\mathbf{e}$ vektor razlike dvaju rješenja $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$. Kao i prije, vrijedi $\mathbf{e} \in \operatorname{Ker}\mathbf{D}$. Po definiciji *sparka* i koristeći nejednakost trokuta slijedi

$$spark(\mathbf{D}) \leq \|\mathbf{e}\|_0 = \|\mathbf{x} - \hat{\mathbf{x}}\|_0 \leq \|\mathbf{x}\|_0 + \|\hat{\mathbf{x}}\|_0, \tag{2.31}$$

a s obzirom na to da po pretpostavci teorema vrijedi $\|\mathbf{x}\|_0 < spark(\mathbf{D})/2$, zaključujemo da mora vrijediti

$$\|\hat{\mathbf{x}}\|_0 > spark(\mathbf{D}) - \|\mathbf{x}\|_0 > spark(\mathbf{D})/2, \tag{2.32}$$

dakle $\hat{\mathbf{x}}$ mora imati više od $spark(\mathbf{D})/2$ ne-nul komponenti. □

2.2.2 Jedinstvenost koristeći međusobnu koherenciju

Bez obzira na sve vrijedne teorijske garancije koje nam nudi *spark* matrice, problem nam ipak predstavlja složenost algoritama za njegovo računanje. Prisjetimo se međusobne koherencije matrice koju smo u slučaju sparenih ortogonalnih matrica definirali koristeći ortogonalne matrice baze. U tom slučaju računanje međusobne koherencije matrice možemo zapisati koristeći Gramovu matricu $\mathbf{D}^T\mathbf{D}$:

$$\mathbf{D}^T\mathbf{D} = \begin{bmatrix} \mathbf{I} & \mathbf{\Psi}^T\mathbf{\Phi} \\ \mathbf{\Phi}^T\mathbf{\Psi} & \mathbf{I} \end{bmatrix}. \tag{2.33}$$

Sada međusobnu koherenciju možemo jednostavno pronaći - ona je apsolutno najveći vandijagonalni element ove Gramove matrice. U tom kontekstu generaliziramo definiciju međusobne koherencije za općenite matrice:

Definicija 2.2.7. *Međusobnu koherenciju matrice $\mathbf{D} \in \mathbb{R}^{n\times m}$ definiramo s*

$$\mu(\mathbf{D}) = \max_{\substack{1\le i,j\le m,\\ i\neq j}} \frac{|\mathbf{a}_i^T\mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \cdot \|\mathbf{a}_j\|_2}, \tag{2.34}$$

gdje $\mathbf{a}_i$ označava i-ti stupac matrice $\mathbf{D}$.

Međusobna koherencija matrice nam u ovom slučaju govori o zavisnosti stupaca matrice $\mathbf{D}$, to jest "mjeri kut" među stupcima matrice. Kod ortogonalnih matrica stupci čine ortonormiranu bazu, stoga je, zbog ortogonalnosti stupaca, međusobna koherencija ortogonalne matrice jednaka 0. Za matrice koje imaju više stupaca nego redaka vrijedi $\mu(\mathbf{D}) > 0$, no težimo što manjim vrijednostima kako bismo dobili matrice što bliže ortogonalnima. Također valja napomenuti kako u općem slučaju ne vrijedi nejednakost (2.4). Za $n \times m$ matrice gdje je $m > n$ donja ograda međusobne koherencije dana je s

$$\mu(\mathbf{D}) \geq \sqrt{\frac{m-n}{n(m-1)}}. \tag{2.35}$$

Ovu relaciju često nazivamo Welchovom nejednakošću. Važno je primijetiti kako ova ocjena ne ovisi o matrici $\mathbf{D}$, već samo o njenim dimenzijama.

Međusobnu koherenciju matrice lako je izračunati, stoga je korisno pomoću nje ograničiti *spark* odozdo. Prije formalnog iskaza leme koja govori o odnosu međusobne koherencije i *sparka* matrice, iskazujemo pomoćni rezultat, teorem o Geršgorinovim krugovima, kojeg ćemo koristiti u dokazu:

Teorem 2.2.8 (Geršgorinovi krugovi)**.** *Neka je $\mathbf{A} \in \mathbb{C}^{n\times n}$ i neka su $\lambda_1, ..., \lambda_n$ njene svojstvene vrijednosti. Za $i = 1, ..., n$ definiramo Geršgorinove krugove*

$$\mathcal{G}_i = \left\{ z \in \mathbb{C} : |z - a_{i,i}| \leq \rho_i \right\},\ \rho_i = \sum_{\substack{j=1,\\ j\neq i}}^{n} |a_{i,j}|. \tag{2.36}$$

Tada vrijedi

1. *Sve svojstvene vrijednosti matrice $\mathbf{A}$ su sadržane u uniji Geršgorinovih krugova,*

$$\sigma(\mathbf{A}) \subseteq \bigcup_{i=1}^{n} \mathcal{G}_i$$

2. *Ako je unija $G_{i_1 \dots i_k} = \bigcup_{j=1}^{k} \mathcal{G}_{i_j}$ nekih k Geršgorinovih krugova disjunktna s unijom preostalih $n - k$ krugova, onda se u $G_{i_1 \dots i_k}$ nalazi točno k svojstvenih vrijednosti matrice $\mathbf{A}$*

Lema 2.2.9. *Za svaku matricu $\mathbf{D} \in \mathbb{R}^{n\times m}$ vrijedi sljedeća nejednakost*

$$spark(\mathbf{D}) \geq 1 + \frac{1}{\mu(\mathbf{D})}. \tag{2.37}$$

Dokaz. Matrici $\mathbf{D}$ normiramo stupce s obzirom na ℓ_2-normu. Ova operacija čuva *spark* i međusobnu koherenciju. Dobivenu matricu označimo s $\tilde{\mathbf{D}}$ te izračunajmo Gramovu matricu $\mathbf{D} = \tilde{\mathbf{D}}^T\tilde{\mathbf{D}} \in \mathbb{R}^{m\times m}$. Za tako dobivenu matricu $\mathbf{G}$ vrijedi

$$\{G_{k,k} = 1 : 1 \leq k \leq m\} \quad i \quad \{|G_{k,j}| \leq \mu(\mathbf{D}) : 1 \leq k, j \leq m, k \neq j\}. \tag{2.38}$$

Pogledajmo proizvoljnu $p \times p$ podmatricu matrice $\mathbf{G}$ dobivenu tako da odaberemo p stupaca iz $\tilde{\mathbf{D}}$ i pomoću njih računamo podmatricu Gramove matrice $\mathbf{G}^p$. Kada bi $\mathbf{G}^p$ bila dijagonalno-dominantna, tj. kada bi vrijedilo

$$\sum_{j\neq i} |G^p_{i,j}| \leq |G^p_{i,i}|, \quad \forall i = 1, ..., m \tag{2.39}$$

mogli bismo, koristeći Geršgorinove krugove, zaključiti da su sve svojstvene vrijednosti matrice $\mathbf{G}^p$ pozitivne, odnosno da je $\mathbf{G}^p$ pozitivno-definitna matrica te su stoga odabranih p stupaca matrice $\tilde{\mathbf{D}}$ linearno nezavisni.

Koristeći (2.38), uvjet dijagonalne dominantnosti glasi $1 > (p-1)\mu(\mathbf{D}) \Rightarrow p < 1 + \frac{1}{\mu(\mathbf{D})}$ što povlači pozitivnu definitnost matrice $\mathbf{G}^p$ za svaku vrijednost p. Zbog toga zaključujemo da je $p = 1 + \frac{1}{\mu(\mathbf{D})}$ najmanji mogući broj stupaca koji mogu biti linearno zavisni te stoga $spark(\mathbf{D}) \geq 1 + \frac{1}{\mu(\mathbf{D})}$. □

Sada možemo iskazati varijantu teorema 2.2.6 koristeći ocjenu *sparka* pomoću međusobne koherencije:

Teorem 2.2.10 (Jedinstvenost koristeći međusobnu koherenciju)**.** *Ako linearan sustav* $\mathbf{b} = \mathbf{Dx}$ *ima rješenje* $\mathbf{x}$ *koje zadovoljava* $\|\mathbf{x}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$*, tada je to rješenje nužno najrjeđe moguće.*

Dokaz. Neka je $\tilde{\mathbf{x}}$ neko drugo rješenje promatranog sustava. Kao u dokazu teorema 2.2.6 označimo s $\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}}$. Ponovno vrijedi $\mathbf{e} \in \operatorname{Ker} \mathbf{D}$. Iz relacije (2.31) koristeći donju ocjenu *sparka* zaključujemo da vrijedi

$$1 + \frac{1}{\mu(\mathbf{D})} \leq spark(\mathbf{D}) \leq \|\mathbf{x}\|_0 + \|\tilde{\mathbf{x}}\|_0, \tag{2.40}$$

a po pretpostavci teorema vrijedi $\|\mathbf{x}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$ zaključujemo da mora vrijediti

$$\|\tilde{\mathbf{x}}\|_0 \geq 1 + \frac{1}{\mu(\mathbf{D})} - \|\mathbf{x}\|_0 > \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right). \tag{2.41}$$

□

Primjer 2.2.11. *Promotrimo nasumično generiranu matricu dimenzija* 80×200 *(*$n = 80, m = 200$*) koja je prikazana na slici 2.1. Po Welchovoj nejednakosti donja ograda za međusobnu koherenciju dana je s*

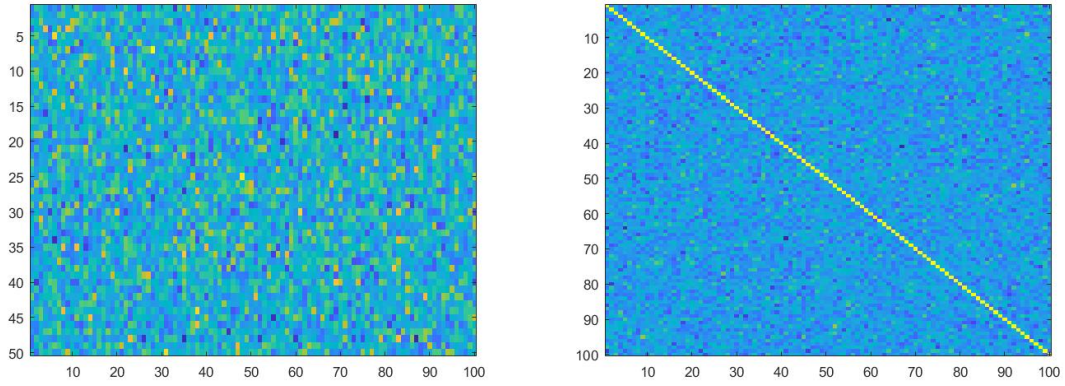
$$\mu(\mathbf{D}) \geq \sqrt{\frac{200 - 80}{80(200 - 1)}} = 0.0868. \tag{2.42}$$

Koristeći rutinu u Matlabu računamo koherenciju i dobivamo

$$\mu(\mathbf{D}) = 0.4251. \tag{2.43}$$

Pomoću leme 2.2.9 možemo ocjeniti spark

$$spark(\mathbf{D}) \geq 1 + \frac{1}{0.4251} = 3.3521, \tag{2.44}$$



(a) Matrica $\mathbf{D}$. (b) Matrica $\mathbf{D}^T\mathbf{D}$.

Slika 2.1: Matrica iz primjera 2.2.11.

a s obzirom na to da je $spark(\mathbf{D}) \in \mathbb{N}$*, mora vrijediti*

$$spark(\mathbf{D}) \geq 4, \tag{2.45}$$

tj. postoji skup od najmanje 4 stupca matrice $\mathbf{D}$ *koji su linearno zavisni. Po teoremu 2.2.6, jedinstvenost rješenja možemo tvrditi tek za rješenje koje ima manje od* $4/2 = 2$ *ne-nul elemenata, tj. koje ima samo jedan ne-nul element, što je onda uistinu najrjeđe moguće rješenje.*

```
function mu = mutual_coherence(A)
[n m] = size(A);

if(m<2)
    disp('error - matrix has only one column');
   return;
end

colNorms = sqrt(diag(A'*A));
if(~all(colNorms))
    disp('error - matrix has a null-column');
    return;
end

A=A*diag(1./colNorms);
```

```
16 mu = max(max(abs(triu((A')*A,1))));
```

Izvorni kôd 2.1: MATLAB kôd za računanje međusobne koherencije.

2.2.3 Jedinstvenost koristeći kumulativnu koherenciju

Definicija 2.2.12. *Za matricu $\tilde{\mathbf{D}}$ s normiranim stupcima promatramo podskup od p stupaca čiji skup indeksa nazivamo Λ. Kumulativnu koherenciju dobivamo kao sumu apsolutnih vrijednosti skalarnih produkata stupaca iz Λ sa stupcem koji nije u Λ, maksimiziramo po Λ i po stupcima koji nisu u Λ te dobivamo*

$$\mu_1(p) = \max_{\Lambda,|\Lambda|=p} \max_{j\notin\Lambda} \sum_{i\in\Lambda} |\tilde{\mathbf{a}}_i^T \tilde{\mathbf{a}}_j|. \tag{2.46}$$

Do ove definicije došli smo analizom dokaza leme 2.2.9 - da bismo u punoj općenitosti osigurali dijagonalnu dominantnost matrice, moramo provjeriti da je suma apsolutnih vrijednosti vandijagonalnih elemenata u svakom retku manja od 1 kako bismo mogli iskoristiti teorem 2.2.8, bez ograničavanja vandijagonalnih elemenata kao u dokazu leme 2.2.9, točnije u relaciji (2.38). Kada je $p = 1$, kumulativna koherencija postaje jednostavno međusobna koherencija.

S obzirom na to da za svaki p, pri računanju kumulativne koherencije moramo proći kroz sve $(p+1)$-torke (p stupaca iz Λ i jedan vanjski stupac) što znači da s porastom p raste i veličina skupa u kojem tražimo maksimum, a kako je maksimum po većem skupu veći od maksimuma po manjem, zaključujemo da je kumulativna koherencija rastuća funkcija (ne strogo). Računanje kumulativne koherencije može se pojednostaviti na sljedeći način:

$$\mu_1(p) = \max_{1\le j\le m} \sum_{i=2}^{p+1} |\mathbf{G}^S_{j,i}| \tag{2.47}$$

gdje je $\mathbf{G}^S$ matrica dobivena iz Gramove matrice $\mathbf{G} = \tilde{\mathbf{D}}^T\tilde{\mathbf{D}}$ sortirajući svaki redak u padajućem redoslijedu. Prvi element svakog retka je 1 (dijagonalni element), stoga sumacija kreće od drugog elementa. Zbrajajući sljedećih p komponenti (od druge nadalje), dobivamo za svaki vanjski j sumu p najvećih vrijednosti skalarnih produkata j-tog stupca sa stupcima iz Λ, to jest dobivamo Λ takav da je $|\Lambda| = p$ koji je najrazličitiji od promatranog j-tog stupca. Zatim izaberemo najveću vrijednost od m pronađenih.
Napomenimo još da za svaki p vrijedi

$$\mu_1(p) \le p \cdot \mu(\mathbf{D}). \tag{2.48}$$

Kako možemo koristiti kumulativnu koherenciju da bismo ograničili *spark* odozdo? Ako je $\mu_1(p) < 1$, zaključujemo da su svi skupovi Λ s $p+1$ elemenata linearno nezavisni zbog dijagonalne dominantnosti i Geršgorinova teorema. Ovime smo dali intuitivni dokaz sljedeće leme:

Lema 2.2.13. *Za svaku matricu* $\mathbf{D} \in \mathbb{R}^{n\times m}$ *vrijedi sljedeća nejednakost*

$$spark(\mathbf{D}) \geq \min_{1\leq p\leq n}\{p : \mu_1(p-1) \geq 1\}. \tag{2.49}$$

Sada, koristeći ovu ocjenu, možemo navesti teorem o jedinstvenosti

Teorem 2.2.14 (Jedinstvenost koristeći kumulativnu koherenciju)**.** *Ako linearan sustav* $\mathbf{b} = \mathbf{Dx}$ *ima rješenje* $\mathbf{x}$ *koje zadovoljava* $\|\mathbf{x}\|_0 < \frac{1}{2}\min_{1\leq p\leq n}\{p : \mu_1(p-1) \geq 1\}$*, tada je to rješenje nužno najrjeđe moguće.*

Dokaz. Dokaz se provodi analogno kao dokaz teorema 2.2.10 uz ocjenu (2.49). □

2.2.4 Ocjena *sparka* odozgo

Kao što smo već napomenuli na početku ovog potpoglavlja, računanje *sparka* matrice je vrlo zahtjevan problem, štoviše, NP-težak. Dosada smo tražili ocjenu *sparka* odozdo i pomoću nje iskazali i dokazali teoreme o jedinstvenosti rijetkog rješenja linearnog sustava. U ovom poglavlju fokusiramo se na traženje gornje ograde *sparka*. Više ne možemo tvrditi jedinstvenost kao prije, ali možemo dobiti vrlo dobru aproksimaciju intervala jedinstvenosti.

Kako bismo dobili gornju ogradu, koristimo definiciju 2.2.4. Problem iz te definicije zapisat ćemo kao niz od m optimizacijskih problema (P_0^i). U svakom problemu (P_0^i) pretpostavljamo da je i-ta komponenta različita od nule, $i = 1, ..., m$.

$$(P_0^i): \quad \mathbf{x}_{opt}^i = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{t.d.} \quad \mathbf{Dx} = \mathbf{0}\ i\ x_i = 1. \tag{2.50}$$

Spark je potom dan kao najrjeđe od pronađenih rješenja

$$spark(\mathbf{D}) = \min_{1\leq i\leq m} \|\mathbf{x}_{opt}^i\|_0. \tag{2.51}$$

S obzirom na to da su problemi (P_0^i) kompleksni za rješavanje jer koriste ℓ_0-normu kao kriterijsku funkciju, relaksiramo ih koristeći ℓ_1-normu. Tako dobivene probleme označavamo s (P_1^i) i glase:

$$(P_1^i): \quad \mathbf{z}_{opt}^i = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{t.d.} \quad \mathbf{Dx} = \mathbf{0}\ i\ x_i = 1. \tag{2.52}$$

Problemi (P_1^i) su konveksni, njihova rješenja postoje i nije ih teško pronaći jer za njihovo rješavanje možemo koristiti metode linearnog programiranja. S obzirom na to da linearni sustav $\mathbf{Dx} = \mathbf{0}$ ima jedinstveno rijetko rješenje kao što smo pokazali u prethodnom dijelu poglavlja, zaključujemo da mora vrijediti

$$\|\mathbf{x}_{opt}^i\|_0 \leq \|\mathbf{z}_{opt}^i\|_0. \tag{2.53}$$

Na ovaj smo način dobili gornju ocjenu *sparka*

$$spark(\mathbf{D}) \leq \min_{1\leq i\leq m} \|\mathbf{z}_{opt}^i\|_0. \tag{2.54}$$

2.2.5 Grassmannove matrice

Numerički prikaz međusobne koherencije matrice ilustrirat ćemo pomoću familije matrica koje nazivamo Grassmannove matrice.

Definicija 2.2.15. *Matricu* $\mathbf{D} \in \mathbb{R}^{n\times m}$*, gdje je* $m \geq n$*, nazivamo Grassmannovom matricom ako njena Gramova matrica* $\mathbf{G} = \mathbf{D}^T\mathbf{D}$ *zadovoljava*

$$\forall k \neq j,\ |G_{k,j}| = \sqrt{\frac{m-n}{n(m-1)}}. \tag{2.55}$$

Iz definicije slijedi da se za Grassmannove matrice postiže jednakost u Welchovoj nejednakosti (2.35), odnosno međusobna koherencija je najmanja moguća. Iz toga nadalje zaključujemo da su svi stupci u parovima pod istim kutom, koji je ujedno i najmanji mogući kut među stupcima matrice $\mathbf{D}$. Za Grassmannove matrice znamo i odnos međusobne i kumulativne koherencije - postiže se jednakost u relaciji (2.48). Za *spark* Grassmannove matrice znamo da vrijedi

$$spark(\mathbf{D}) = n + 1, \tag{2.56}$$

to jest *spark* je najveći mogući.

Pokazuje se da Grassmannove matrice postoje samo u slučaju kada vrijedi

$$m \leq \min\left\{\frac{n(n-1)}{2}, \frac{(m-n)(m-n+1)}{2}\right\}.$$

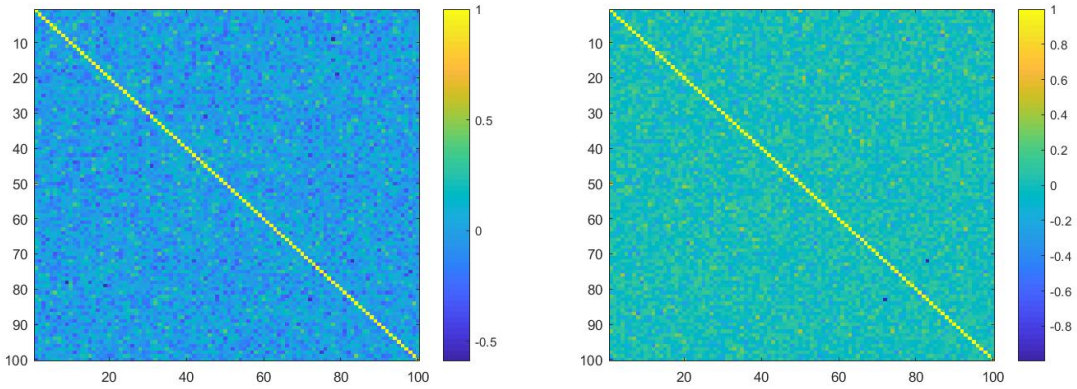
U slučaju $m = n$ Grassmannove matrice su jednostavno ortogonalne matrice. Konstrukcija Grassmannovih matrica može se provesti iterativno kao što je opisano u [2]. Algoritam za konstrukciju Grassmannove matrice mora pratiti tri glavne smjernice a to su

- Stupci moraju ostati normirani s obzirom na ℓ_2-normu;
- U Gramovoj matrici smanjujemo jako velike vrijednosti i povećavamo jako male vrijednosti (s obzirom na zadani prag);
- Rang matrice ne smije postati veći od n - o tome vodimo računa koristeći SVD dekompoziciju te anulirajući singularne vrijednosti nakon prvih n.

Iako ne postoji garancija da ovaj algoritam konvergira niti da pronalazi odgovarajuću Grassmannovu matricu ukoliko ona postoji, provest ćemo 10000 iteracija algoritma u nadi da ćemo se približiti Grassmannovoj matrici te ćemo promatrati promjene u međusobnoj koherenciji.

Primjer 2.2.16. *Promotrimo nasumično generiranu matricu* $\mathbf{D}$ *dimenzija* 50×100 *čiji su stupci normirani. Provest ćemo* 10000 *iteracija algoritma za konstrukciju Grassmannove matrice preuzetog iz [3]. Najmanja moguća vrijednost međusobne koherencije je*

$$\mu(\mathbf{D}) \geq \sqrt{\frac{100-50}{50(100-1)}} = 0.1005. \tag{2.57}$$



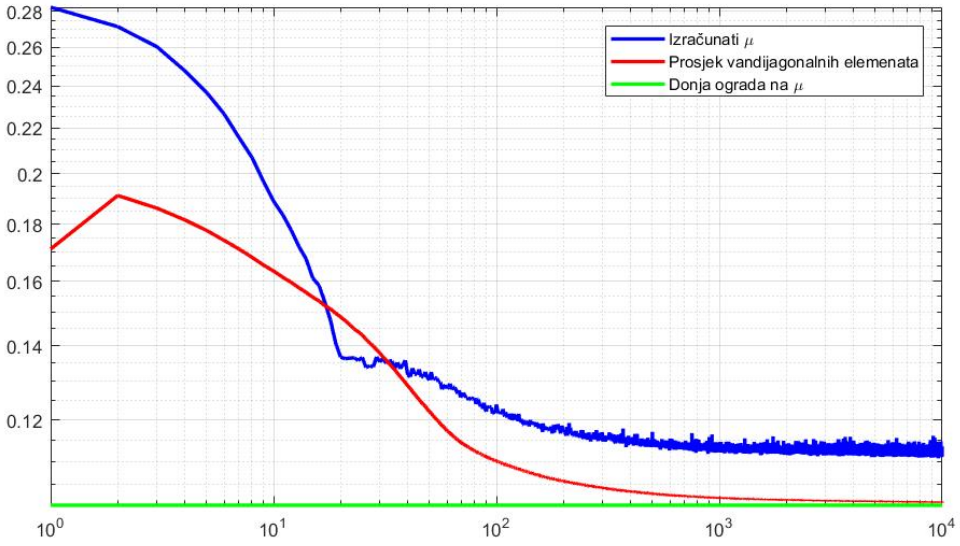
(a) Inicijalna Gramova matrica.

(b) Gramova matrica nakon 10000 iteracija.

Slika 2.2: Prikaz Gramovih matrica u algoritmu konstrukcije Grassmannove matrice.

Na slici 2.2 vidimo da su primjenom algoritma dijagonalni elementi Gramove matrice $\mathbf{D}^T\mathbf{D}$ *ostali nepromijenjeni i iznose* 1 *zbog toga što su stupci matrice normirani. Promjenu vidimo na vandijagonalnim elementima. Nakon 10000 iteracija algoritma oscilacije se smanjuju i vandijagonalni elementi teže istim vrijednostima, tj. matrica* $\mathbf{D}$ *teži Grassmannovoj.*

Na slici 2.3 prikazujemo (u logaritamskoj skali) kako se međusobna koherencija mijenja s iteracijama. Uočimo kako se s iteracijama međusobna koherencija smanjuje i približava svojoj donjoj ogradi danoj Welchovom nejednakošću. Međusobna koherencija finalne matrice iznosi 0.1125. *Međutim, to nije minimalna vrijednost međusobne koherencije postignuta tijekom izvođenja ovog algoritma. Minimalna postignuta vrijednost međusobne koherencije iznosila je* 0.1108. *Ovo je posljedica spomenute neoptimalnosti promatranog algoritma.*



Slika 2.3: Međusobna koherencija kroz iteracije algoritma za konstrukciju Grassmannove matrice.

U ovom trenutku imamo teorijsku garanciju egzistencije, jedinstvenosti i globalne optimalnosti rijetkog rješenja problema (P_0). U nastavku se bavimo praktičnim metodama traženja rješenja problema (P_0), takozvanim algoritmima potrage (engl. *pursuit algorithms*).

```
n=50; m=100; iter=10000;
dd1=0.9; dd2=0.9;

D=randn(n,m);
D=D*diag(1./sqrt(diag(D'*D)));
G=D'*D;

mu=sqrt((m-n)/n/(m-1));
muVec = []; muCo = []; muMean = [];

for k=1:1:iter
    gg=sort(abs(G(:)));
    pos=find(abs(G(:))>gg(round(dd1*(m*m-m))) & abs(G(:)) <1);
    G(pos)=G(pos)*dd2;
    [U,S,V]=svd(G);
    S(n+1:end,1+n:end)=0;
    G=U*S*V';
    G = G*diag(1./sqrt(diag(G'*G)));

    muCo = cat(1,muCo,max(max(abs(triu(G,1)))) );
    muMean = cat(1,muMean, mean(abs(G(pos))));
    muVec = cat(1,muVec,mu);
end

[U,S,V]=svd(G);
D=sqrt(S(1:n,1:n))*U(:,1:n)';
D=D*diag(1./sqrt(diag(D'*D)));
```

Izvorni kôd 1: MATLAB kôd za konstrukciju Grassmannove matrice uz konstrukciju vektora međusobne koherencije te prosjeka apsolutnih vrijednosti vandijagonalnih elemenata po iteracijama, adaptiran iz [3].

Poglavlje 3

Algoritmi potrage

U ovom poglavlju analiziramo neke od metoda za rješavanje problema (P_0) kao što su OMP (engl. *Orthogonal-Matching Pursuit*), algoritma s pragom tolerancije (engl. *Thresholding Algorithm*) te metode koje se zasnivaju na relaksaciji problema kao što je BP algoritam (engl. *Basis Pursuit*). Na problem možemo gledati sa dva stajališta: diskretno i neprekidno. Kad kažemo diskretno stajalište, mislimo konkretno na problem pronalaženja skupa indeksa na kojima se nalaze ne-nul vrijednosti (nosač vektora). Tada same vrijednosti lako pronalazimo rješavajući problem najmanjih kvadrata. Ovakav pristup vodi na pohlepne algoritme kojima pripadaju OMP i algoritam s pragom tolerancije. Neprekidno stajalište odnosi se na relaksaciju diskretne prirode problema što vodi na neprekidan problem u kojem možemo koristiti razne metode (konveksne) optimizacije kao što je BP algoritam.

3.1 Pohlepni algoritmi

Pohlepni algoritmi konsturiraju i/ili prilagođavaju rješenje radeći lokalno optimalne promjene. U ovoj kategoriji bavimo se dvama algoritmima: OMP-om te algoritmom s pragom tolerancije koji je vrlo jednostavan, a pod određenim uvjetima iznenađujuće precizan.

3.1.1 OMP

Ideja je vrlo jednostavna: krenuvši od nul-vektora iterativno gradimo rješenje – dodajemo vrijednosti na jednu po jednu komponenetu, a zatim provjeravamo ℓ_2-normu pogreške. Kada pogreška padne ispod zadane točnosti, ovako konsturirano rješenje proglašavamo optimalnim za zadanu točnost. U nastavku navodimo formalne korake algoritma:

Problem: (P_0) : $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ t.d. $\mathbf{b} = \mathbf{D}\mathbf{x}$
Podatci: matrica $\mathbf{D}$, vektor desne strane $\mathbf{b}$, tražena točnost ϵ
Inicijaliziraj: k = 0,
inicijalno rješenje $\mathbf{x}^0 = \mathbf{0}$,
inicijalni rezidual $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0 = \mathbf{b}$,
inicijalni nosač rješenja $\mathcal{S}^0 = \text{supp}(\mathbf{x}^0) = \emptyset$.
dok $\|\mathbf{r}^k\|_2 < \epsilon$ **čini**
k = k+1;
izračunaj greške $e(j) = \min_{z_j} \|\mathbf{a}_j z_j - \mathbf{r}^{k-1}\|_2^2$, $\forall$ j, koristeći optimalni $z_j^* = \frac{\mathbf{a}_j^T \mathbf{r}^{k-1}}{\|\mathbf{a}_j\|_2^2}$;
pronađi $j_0 = \arg\min_j \{ e(j) : j \notin \mathcal{S}^{k-1}\}$ i ažuriraj nosač $\mathcal{S}^k = \mathcal{S}^{k-1} \cup \{j_0\}$;
izračunaj $\mathbf{x}^k = \arg\min_{\mathbf{x}} \{ \|\mathbf{D}\mathbf{x} - \mathbf{b}\|_2^2 : \text{supp}(\mathbf{x}) = \mathcal{S}^k\}$;
izračunaj $\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k$;
kraj
Rezultat: aproksimativno rješenje $\mathbf{x}^k$ problema (P_0)

Algoritam 1: OMP algoritam

U koraku računanja trenutno optimalnog rješenja $\mathbf{x}^k$ minimiziramo izraz $\|\mathbf{D}\mathbf{x} - \mathbf{b}\|_2^2$ po $\mathbf{x}$ čiji je nosač sadržan u $\mathcal{S}^k$. Ovo možemo zapisati i kao

$$\min_{\mathbf{x}} \|\mathbf{D}_{\mathcal{S}^k}\mathbf{x}_{\mathcal{S}^k} - \mathbf{b}\|_2^2 \tag{3.1}$$

gdje smo s $\mathbf{D}_{\mathcal{S}^k} \in \mathbb{R}^{n \times |\mathcal{S}^k|}$ označili matricu dobivenu iz $\mathbf{D}$ uzimanjem stupaca koji se nalaze u $\mathcal{S}^k$, a s $\mathbf{x}_{\mathcal{S}^k}$ smo označili vektor dobiven iz $\mathbf{x}$ dobiven uzimanjem samo ne-nul komponenti, tj. komponenti iz $\mathcal{S}^k$. Deriviranjem ove kvadratne forme i izjednačavanjem s $\mathbf{0}$ dobivamo

$$\mathbf{D}_{\mathcal{S}^k}^T\mathbf{D}_{\mathcal{S}^k}\mathbf{x}_{\mathcal{S}^k} - \mathbf{D}_{\mathcal{S}^k}^T\mathbf{b} = \mathbf{D}_{\mathcal{S}^k}^T(\mathbf{D}_{\mathcal{S}^k}\mathbf{x}_{\mathcal{S}^k} - \mathbf{b}) = -\mathbf{D}_{\mathcal{S}^k}^T\mathbf{r}^k = 0 \tag{3.2}$$

gdje smo iskoristili izraz za rezidual u k-tom koraku algoritma $\mathbf{r}^k = \mathbf{b} - \mathbf{D}\mathbf{x}^k = \mathbf{b} - \mathbf{D}_{\mathcal{S}^k}\mathbf{x}_{\mathcal{S}^k}$. Ovime smo pokazali da su stupci iz $\mathcal{S}^k$ nužno ortogonalni na rezidual $\mathbf{r}^k$, što nam garantira da se u idućim iteracijama algoritma već odabrani stupci nosača neće ponovno odabirati. Iz ove ortogonalnosti algoritam dobiva svoje ime. Kako bismo opravdali korištenje OMP algoritma, navodimo teorijsku garanciju pronalaska rješenja najprije u specijalnom slučaju sparenih ortogonalnih matrica, a zatim u punoj općenitosti za proizvoljne matrice $\mathbf{D}$.

Teorem 3.1.1 (OMP za sparene ortogonalne matrice)**.** *Za linearni sustav* $\mathbf{D}\mathbf{x} = [\mathbf{\Psi}, \mathbf{\Phi}] = \mathbf{b}$ *gdje su* $\mathbf{\Psi}, \mathbf{\Phi} \in \mathbb{R}^{n \times n}$ *ortogonalne matrice vrijedi: ako postoji rješenje* $\mathbf{x}$ *koje ima* k_p *ne-nul vrijednosti na prvih n komponenti te* k_q *ne-nul vrijednosti na posljednjih n komponenti te ako* k_q *i* k_p *zadovoljavaju*

$$\max(k_p, k_q) < \frac{1}{2\mu(\mathbf{D})}, \tag{3.3}$$

tada ga OMP algoritam s $\epsilon_0 = 0$ *pronalazi u točno* $k_0 = k_p + k_q$ *iteracija.*

Teorem 3.1.2 (OPM za opći slučaj)**.** *Za linearni sustav* $\mathbf{Dx} = \mathbf{b}$*, gdje je* $\mathbf{D} \in \mathbb{R}^{n\times m}$*,* $n < m$*, matrica punog ranga vrijedi: ako postoji rješenje* $\mathbf{x}$ *za koje vrijedi*

$$\|\mathbf{x}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right), \tag{3.4}$$

tada ga OMP algoritam s $\epsilon_0 = 0$ *pronalazi u konačno mnogo iteracija.*

Dokazi ovih dvaju teorema mogu se pronaći u [3].

Razne varijante OMP algoritma prisutne su u primijeni, s poboljšanjima u vidu točnosti ili složenosti. Jedna od takvih metoda je primjerice LS-OMP (engl. *Least Squares Orthogonal-Matching Pursuit*) koja je složenija od klasičnog OMP-a jer ne-nul komponenete ne dodajemo jednu po jednu, već tražimo kumulativno, rješavajući problem najmanjih kvadrata u prvom koraku algoritma.

Nadalje, koriste se i takozvane MP (engl. *Matching Pursuit*) metode koje su jednostavnije od OMP-a. Razlika je u računanju trenutno optimalnog rješenja $\mathbf{x}^k$ u k-tom koraku algoritma - umjesto rješavanja problema najmanjih kvadrata kako bismo pronašli optimalne vrijednosti ne-nul komponenti u vektoru rješenja i promijenili postojeće, u ovoj varijanti prethodno izračunate vrijednosti (one koje odgovaraju komponentama iz $\mathcal{S}^{k-1}$) se ne mijenjaju, već samo računamo vrijednost komponente na trenutno odabranom indeksu $j_0 \in \mathcal{S}^k$. Postoji jednostavnija varijanta MP algoritma, tzv. slabi-MP (engl. *Weak*-MP) gdje u svakom koraku odabiremo suboptimalan indeks kojeg dodajemo u nosač $\mathcal{S}^k$.

3.1.2 Algoritam s pragom tolerancije

U ovom odjeljku bavimo se najjednostavnijim pohlepnim algoritmom za rješavanje problema (P_0), algoritmom s pragom tolerancije. Za razliku od OMP-algoritma koji iterativno pronalazi nove indekse koje dodaje u nosač, algoritam s pragom tolerancije koristi se za unaprijed zadani broj ne-nul komponenti vektora rješenja k. U slučaju da je željeni broj ne-nul komponenti k nepoznat ili ga je nemoguće odrediti prije provođenja algoritma, moguće je napraviti modifikaciju algoritma na način da k iterativno povećavamo sve dok greška aproksimacije ne dosegne zadanu točnost. Funkcija koju koristimo za računanje pogreške jednaka je onoj korištenoj u OMP-algoritmu. S obzirom na njegovu jednostavnost, nije potrebno posebno promatrati slučaj sparenih ortogonalnih matrica. U nastavku navodimo korake algoritma s pragom tolerancije:

Problem: (P_0) : $\min_{\mathbf{x}} ||\mathbf{x}||_0$ t.d. $\mathbf{b} = \mathbf{D}\mathbf{x}$
Podatci: matrica $\mathbf{D}$, vektor desne strane $\mathbf{b}$, broj ne-nul komponenti k
dok $||\mathbf{r}^k||_2 < \epsilon$ **čini**
izračunaj greške $e(j) = \min_{z_j} ||\mathbf{a}_j z_j - \mathbf{r}^{k-1}||_2^2$, $\forall$ j, koristeći optimalni $z_j^* = \frac{\mathbf{a}_j^T \mathbf{r}^{k-1}}{||\mathbf{a}_j||_2^2}$;
pronađi nosač $\mathcal{S}$ takav da $|\mathcal{S}| = k$ kao skup k indeksa za koje je greška najmanja, tj. $\forall j \in \mathcal{S}$, $\epsilon(j) \leq \min_{i \notin \mathcal{S}} \epsilon(i)$;
izračunaj $\mathbf{x}^k = \arg\min_{\mathbf{x}} \{ ||\mathbf{D}\mathbf{x} - \mathbf{b}||_2^2 : \text{supp}(\mathbf{x}) = \mathcal{S}\}$;
kraj
Rezultat: aproksimativno rješenje $\mathbf{x}^k$ problema (P_0)

Algoritam 2: Algoritam s pragom tolerancije

Algoritam s pragom tolerancije vrlo je jednostavan i brz algoritam, a pokazuje se da je u određenim situacijama dovoljno točan. Potrebno je još navesti pod kojim uvjetima algoritam s pragom tolerancije pronalazi rješenje:

Teorem 3.1.3 (Algoritam s pragom tolerancije)**.** *Za linearni sustav* $\mathbf{D}\mathbf{x} = \mathbf{b}$*, gdje je* $\mathbf{D} \in \mathbb{R}^{n \times m}$*,* $n < m$*, matrica punog ranga vrijedi: ako postoji rješenje* $\mathbf{x}$*, koje ima minimalnu apsolutnu ne-nul vrijednost* $x_{min} = \min\{ |x_i| : x_i \neq 0, i = 1, ..., n\}$ *te maksimalnu apsolutnu ne-nul vrijednost* $x_{max} = \max\{ |x_i| : x_i \neq 0, i = 1, ..., n\}$ *,za koje vrijedi*

$$||\mathbf{x}||_0 < \frac{1}{2}\left(1 + \frac{1}{\mu}\frac{x_{min}}{x_{max}}\right), \tag{3.5}$$

tada ga algoritam s pragom tolerancije $\epsilon_0 = 0$ *pronalazi u konačno mnogo koraka.*

Dokaz ovog teorema može se pronaći u [3].

3.2 Metode relaksacije

Kao što smo napomenuli u uvodnom dijelu poglavlja, metode relaksacije diskretnu ℓ_0-normu zamjenjuju neprekidnim funkcijama kao što su primjerice ℓ_p-norme za $p \in \langle 0, 1]$ svodeći problem (P_0) na problem (P_p) koji je konveksne prirode. Zatim se koriste neki od poznatih algoritma za rješavanje problema konveksne optimizacije kao što je npr. IRLS algoritam (engl. *Iterative-Reweighed Least Squares*) gdje ℓ_p-normu aproksimiramo koristeći ℓ_2-normu i to na sljedeći način: neka je $\mathbf{x}^{k-1}$ trenutno rješenje u iterativnom postupku. Pomoću tog rješenja kreiramo dijagonalnu matricu $\mathbf{X}_{k-1} = diag(|\mathbf{x}^k|^q)$. Pod pretpostavkom

da je ova matrica regularna, vrijedi

$$\begin{aligned}\|\mathbf{X}_{k-1}^{-1}\mathbf{x}\|_2^2 &= (\mathbf{X}_{k-1}^{-1}\mathbf{x})^T(\mathbf{X}_{k-1}^{-1}\mathbf{x}) \\ &= \mathbf{x}^T\mathbf{X}_{k-1}^{-T}\mathbf{X}_{k-1}^{-1}\mathbf{x} \\ &= \mathbf{x}^T\begin{bmatrix}(\mathbf{x}_1^{k-1})^{-2q} & 0 & \dots & 0 \\ 0 & (\mathbf{x}_2^{k-1})^{-2q} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (\mathbf{x}_m^{k-1})^{-2q}\end{bmatrix}\mathbf{x} \\ &= \|\mathbf{x}\|_{2-2q}^{2-2q}\end{aligned} \tag{3.6}$$

Ako izaberemo $q = 1 - p/2$, dobivamo

$$\|\mathbf{X}_{k-1}^{-1}\mathbf{x}\|_2^2 = \|\mathbf{x}\|_p^p, \tag{3.7}$$

čime smo dobili način kako ℓ_p-normu zapisati pomoću ℓ_2-norme. Uočimo da ako izaberemo $q = 1$, dobivamo interpretaciju ℓ_0-norme pomoću ℓ_2-norme.

Sada umjesto inverza matrice $\mathbf{X}_{k-1}$, u čije postojanje nismo sigurni, koristimo pseudoinverz kojeg ćemo označiti s $\mathbf{X}_{k-1}^+$. S obzirom na to da je $\mathbf{X}_{k-1}$ dijagonalna matrica, singularna je ako postoji dijagonalni element koji je jednak nuli. U tom smislu pseudoinverz definiramo kao

$$[\mathbf{X}_{k-1}^+]_{i,j} = \begin{cases}0, & [\mathbf{X}_{k-1}]_{i,j} = 0 \\ \frac{1}{[\mathbf{X}_{k-1}]_{i,j}}, & [\mathbf{X}_{k-1}]_{i,j} \neq 0,\end{cases} \tag{3.8}$$

tj. kao dijagonalnu matricu koja je dobivena iz $\mathbf{X}_{k-1}$ tako da su nule ostale nule, a sve nenul vrijednosti zamijenjene su svojim recipročnim vrijednostima. Uz relaciju (3.7), koja vrijedi i za pseudoinverz, zapišimo problem (P_p) kao

$$(M_k): \quad \min_{\mathbf{x}} \|\mathbf{X}_{k-1}^+\mathbf{x}\|_2^2 \quad \text{t.d.} \quad \mathbf{b} = \mathbf{D}\mathbf{x} \tag{3.9}$$

i riješimo ga pomoću Lagrangeovih multiplikatora

$$\mathcal{L}(\mathbf{x}) = \|\mathbf{X}_{k-1}^+\mathbf{x}\|_2^2 + \lambda^T(\mathbf{b} - \mathbf{D}\mathbf{x}) \Rightarrow \frac{\partial\mathcal{L}(\mathbf{x})}{\partial\mathbf{x}} = 2(\mathbf{X}_{k-1}^+)^2\mathbf{x} - \mathbf{D}^T\lambda = 0 \tag{3.10}$$

iz čega slijedi da je rješenje problema (M_k) dano s

$$\mathbf{x}^k = \frac{1}{2}(\mathbf{X}_{k-1}^+)^{-2}\mathbf{D}^T\lambda. \tag{3.11}$$

Pogledajmo sada inverz pseudoinverza. Ako je $\mathbf{X}_{k-1}^+$ regularna, vrijedi $(\mathbf{X}_{k-1}^+)^{-1} = \mathbf{X}_{k-1}$, stoga prethodna relacija prelazi u

$$\mathbf{x}^k = \frac{1}{2}\mathbf{X}_{k-1}^2\mathbf{D}^T\lambda. \tag{3.12}$$

U slučaju da pseudoinverz nije regularan, postoje dijagonalne vrijednosti koje su jednake nuli, to jest vektor $\mathbf{x}_{k-1}$ imao je neke nul-vrijednosti. Gornja jednakost nam sada garantira da će i $\mathbf{x}_k$ imati nule na odgovarajućim komponentama, čime se održava rijetkost rješenja. Uvrstimo dobiveno rješenje $\mathbf{x}_k$ u uvjet $\mathbf{b} = \mathbf{D}\mathbf{x}$:

$$\frac{1}{2}\mathbf{D}\mathbf{X}_{k-1}^2\mathbf{D}^T\lambda = b \quad \Rightarrow \quad \lambda = 2\left(\mathbf{D}\mathbf{X}_{k-1}^2\mathbf{D}^T\right)^{-1}\mathbf{b}. \tag{3.13}$$

Primijetimo da inverz $(\mathbf{D}\mathbf{X}_{k-1}^2\mathbf{D}^T)^{-1}$ moramo zamijeniti pseudoinverzom $(\mathbf{D}\mathbf{X}_{k-1}^2\mathbf{D}^T)^+$ s obzirom na to da matrica $\mathbf{D}\mathbf{X}_{k-1}^2\mathbf{D}^T \in \mathbb{R}^{n\times n}$ nije nužno regularna. Kako je matrica $\mathbf{X}_{k-1}^2 \in \mathbb{R}^{m\times m}$ dijagonalna, ovo matrično množenje možemo shvatiti kao da pomoću matrice $\mathbf{X}_{k-1}$ izabiremo određeni broj stupaca iz $\mathbf{D}$, a broj odabranih stupaca ovisi o broju ne-nul elemenata u $\mathbf{X}_{k-1}$, tj. o broju ne-nul elemenata u $\mathbf{x}_{k-1}$. Označimo broj odabranih stupaca s $k = \|\mathbf{x}_{k-1}\|_0$. Ako je $k \geq n$ te ako tih k odabranih stupaca razapinju cijeli prostor $\mathbb{R}^n$, tada je $\mathbf{D}\mathbf{X}_{k-1}^2\mathbf{D}^T$ regularna. U slučaju da $k \geq n$ stupaca ne razapinju cijeli prostor ili ako je $k < n$, $\mathbf{D}\mathbf{X}_{k-1}^2\mathbf{D}^T$ nije nužno regularna i u tom slučaju ne možemo govoriti o njenom inverzu. Dakle, u punoj općenitosti λ je dan s

$$\lambda = 2\left(\mathbf{D}\mathbf{X}_{k-1}^2\mathbf{D}^T\right)^{+}\mathbf{b}. \tag{3.14}$$

Uvrštavanjem u izraz (3.12) dobivamo

$$\mathbf{x}^k = \mathbf{X}_{k-1}^2\mathbf{D}^T\left(\mathbf{D}\mathbf{X}_{k-1}^2\mathbf{D}^T\right)^{+}\mathbf{b}. \tag{3.15}$$

Sljedeću iteraciju algoritma provodimo na analogan način, koristeći $\mathbf{x}^k$ kako bismo ažurirali matricu $\mathbf{X}_k$. U nastavku dajemo formalne korake algoritma.

Problem: (P_p) : $\min_{\mathbf{x}} \|\mathbf{x}\|_p^p$ t.d. $\mathbf{b} = \mathbf{D}\mathbf{x}$
Podatci: matrica $\mathbf{D}$, vektor desne strane $\mathbf{b}$, tražena točnost ϵ
Inicijaliziraj: $k = 0$,
inicijalno rješenje $\mathbf{x}^0 = \mathbf{1}$,
inicijalna matrica težina $\mathbf{X}_0 = \mathbf{I}$
inicijalna pogreška $\mathbf{e}^0 = 0$
dok $\|\mathbf{e}^k\|_2 < \epsilon$ **čini**
k = k+1;
izračunaj rješenje linearnog sustava
$\mathbf{x}^k = \mathbf{X}_{k-1}^2\mathbf{D}^T(\mathbf{D}\mathbf{X}_{k-1}^2\mathbf{D}^T)^{+}\mathbf{b}$;
ažuriraj matricu težina $\mathbf{X}$ koristeći $\mathbf{x}^k$
$[\mathbf{X}_k]_{i,j} = |x_j^k|^{1-p/2}$;
izračunaj pogrešku $\mathbf{e}^k = \mathbf{x}^k - \mathbf{x}^{k-1}$;
kraj
Rezultat: aproksimativno rješenje $\mathbf{x}^k$ problema (P_p)

Algoritam 3: IRLS algoritam

Početno rješenje inicijaliziramo na vektor jedinica iz razloga što, kao što smo napomenuli ranije, relacija (3.15) garantira da, prilikom konstrukcije rješenja, komponenete koje su u i-toj iteraciji postavljene na nulu, ne mogu postati ne-nul u daljnjim iteracijama algoritma garantirajući rjeđa (ili barem jednako rijetka) rješenja s porastom broja iteracija.

Sustav u prvom koraku možemo riješiti direktno ili primjenom neke od iterativnih metoda za rješavanje linearnih sustava (primjerice metodom konjugiranih gradijenata).

Strategija rješavanja problema (P_0) relaksacijom na (P_p) te potom koristeći IRLS-algoritam za pronalazak rješenja naziva se FOCUSS algoritmom (engl. *Focal Underdetermined System Solver*). Problem kod FOCUSS algoritma je što ne možemo tvrditi da je pronađena aproksimacija lokalno optimalnog rješenja dovoljno dobra aproksimacija globanog optimuma problema (P_0). Više o samom algoritmu i njegovim nedostacima može se pronaći u [4].

Prelaskom s problema (P_0) na problem (P_p) moramo biti oprezni zbog razlika u promatranim normama. ℓ_0-norma samo "broji" ne-nul elemente, bez obzira na njihovu vrijednost, dok općenite ℓ_p-norme uzimaju u obzir i vrijednost elementa. Zbog toga imaju tendenciju da, prilikom provođenja IRLS algoritma, za ne-nul komponente izabiru one koje pripadaju atomima s velikim normama. Iz tog razloga moramo koristiti matricu težina kako bismo što bliže aproksimirali ℓ_0 normu pomoću neke druge ℓ_p-norme.

U nastavku se bavimo relaksacijom ℓ_0-norme ℓ_1-normom čime prelazimo s problema (P_0) na problem (P_1). Matricu težina koja će voditi računa o prethodno opisanom problemu skalirajući elemente nazovimo $\mathbf{W}$. Sada problem glasi

$$(P_1^W): \quad \min_{\mathbf{x}} \|\mathbf{W}^{-1}\mathbf{x}\|_1 \quad \text{t.d.} \quad \mathbf{b} = \mathbf{D}\mathbf{x}, \tag{3.16}$$

gdje je $\mathbf{W}$ dijagonalna, pozitivno-definitna, s vrijednostima $w_{i,i} = \frac{1}{\|\mathbf{a}_i\|_2}$. Pod pretpostavkom da matrica rječnika $\mathbf{D}$ nema nul-stupaca, vrijedi $\|\mathbf{a}_i\|_2 \neq 0$ te je problem (P_1^W) dobro definiran. Ako su svi stupci matrice $\mathbf{D}$ normirani, tada je $\mathbf{W} = \mathbf{I}$ i u tom slučaju algoritam nazivamo BP algoritmom. Primijetimo da stupce matrice $\mathbf{D}$ možemo normirati prije provođenja algoritma. Normiranjem matrice $\mathbf{D}$ dobivamo matricu $\tilde{\mathbf{D}}$ pa problem (P_1^W) poprima opći oblik problema (P_1). Uz $\tilde{\mathbf{x}} = \mathbf{W}^{-1}\mathbf{x}$ imamo

$$(\tilde{P}_1): \quad \min_{\tilde{\mathbf{x}}} \|\tilde{\mathbf{x}}\|_1 \quad \text{t.d.} \quad \mathbf{b} = \mathbf{D}\mathbf{W}\tilde{\mathbf{x}} = \tilde{\mathbf{D}}\tilde{\mathbf{x}}. \tag{3.17}$$

Važno je zapamtiti transformaciju skaliranja jer rješavanjem gornjeg problema BP algoritmom (ili pohlepnim algorimom) dobivamo normirano rješenje iz kojeg potom moramo rekonstruirati originalno koristeći istu transformaciju. U nastavku pretpostavljamo da je matrica rječnika $\mathbf{D}$ normirana i navodimo dva rezultata koji osiguravaju ekvivalenciju problema (P_0) i (P_1):

Teorem 3.2.1 (BP za sparene ortogonalne matrice)**.** *Za linearni sustav* $\mathbf{Dx} = [\mathbf{\Psi}, \mathbf{\Phi}]\mathbf{x} = \mathbf{b}$ *gdje su* $\mathbf{\Psi}, \mathbf{\Phi} \in \mathbb{R}^{n\times n}$ *ortogonalne matrice vrijedi: ako postoji rješenje* $\mathbf{x}$ *koje ima* k_p *ne-nul vrijednosti na prvih n komponenti te* k_q *ne-nul vrijednosti na posljednjih n komponenti,* $k_q \leq k_p$*, te ako* k_q *i* k_p *zadovoljavaju*

$$2\mu(\mathbf{D})^2 k_p k_q + \mu(\mathbf{D})k_p < 1 \tag{3.18}$$

tada je to rješenje ujedno jedinstveno rješenje problema (P_1) *i jedinstveno rješenje problema* (P_0)*.*

S obzirom na to da je uvjet dan u teoremu kompliciran i neintuitivan, dajemo jednostavniji, slabiji uvjet na rješenje

$$\|\mathbf{x}\|_0 = k_p + k_q < \frac{\sqrt{2} - \frac{1}{2}}{\mu(\mathbf{D})} \tag{3.19}$$

Teorem 3.2.2 (BP za opći slučaj)**.** *Za linearni sustav* $\mathbf{Dx} = \mathbf{b}$*, gdje je* $\mathbf{D} \in \mathbb{R}^{n\times m}$*,* $n < m$*, matrica punog ranga vrijedi: ako postoji rješenje* $\mathbf{x}$ *za koje vrijedi*

$$\|\mathbf{x}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right), \tag{3.20}$$

tada je to rješenje ujedno jedinstveno rješenje problema (P_1) *i jedinstveno rješenje problema* (P_0)*.*

Dokazi ovih teorema, kao i prethodnih teorema o garanciji pronalaska rješenja za ostale spomenute algoritme potrage, mogu se pronaći u [3]. Primjetimo da se u teoremu 3.2.2 javlja jednaka ocjena kao u teoremu 3.1.2, no napomenimo da to ne znači da se algoritmi BP i OMP uvijek ponašaju slično, već da se ponašaju jednako samo u rubnim slučajevima.

3.3 Numerička usporedba algoritama potrage

U ovoj cjelini promotrit ćemo efikasnost algoritama potrage na izabranim primjerima. Algoritmi koje ćemo promatrati su OMP, MP, algoritam s pragom tolerancije te BP algoritam. Implementacija OMP, MP i BP algoritama preuzeta je s [1], dok je implementacija algoritma s pragom tolerancije izvedena samostalno.

```
import numpy as np

def Thresholding(A, y, k):
    m = A.shape[1]

    x = np.zeros(m)
    err = np.zeros(m)

    r = y

    for i in range(m):
        c = np.dot(A[:,i], r)/(np.linalg.norm(A[:,i]))**2
        err[i] = np.linalg.norm(A[:,i].dot(c) - r)

    threshold = np.partition(err,k)[k]
    chosenInd = np.nonzero(err < threshold)
    nullInd = np.nonzero(err >= threshold)

    Aa = A[:,chosenInd[0]];
    [xlsq, _, _, _] = np.linalg.lstsq(Aa, y, rcond=-1);
    x[chosenInd[0]] = xlsq;
    x[nullInd[0]] = 0;

    r = y - np.dot(A, x)

    return x
```

Izvorni kôd 2: Python kôd za algoritam s pragom tolerancije.

Primjer 3.3.1 (Slučaj sparenih ortogonalnih matrica)**.** *Prvo generiramo dvije* 100×100 *ortogonalne matrice* $\mathbf{\Psi}$ *i* $\mathbf{\Phi}$*. Matricu rječnika* $\mathbf{D}$ *dimenzija* 100×200 *konsturiramo konkatenacijom matrica* $\mathbf{\Psi}$ *i* $\mathbf{\Phi}$*. Izračunamo njihovu međusobnu koherenciju kako bismo potvrdili ortogonalnost, a potom računamo međusobnu koherenciju matrice* $\mathbf{D}$*:*

$$\mu(\mathbf{\Psi}) = 1.0277\text{e}{-15}$$
$$\mu(\mathbf{\Phi}) = 1.3608\text{e}{-15}$$
$$\mu(\mathbf{D}) = 0.3517.$$

Po teoremima 3.1.1, 3.1.3, 3.2.1 rješenje problema pronalaska rijetke reprezentacije za dani vektor $\mathbf{b}$ *je jedinstveno i promatrani algoritam ga pronalazi u konačno mnogo itera-*

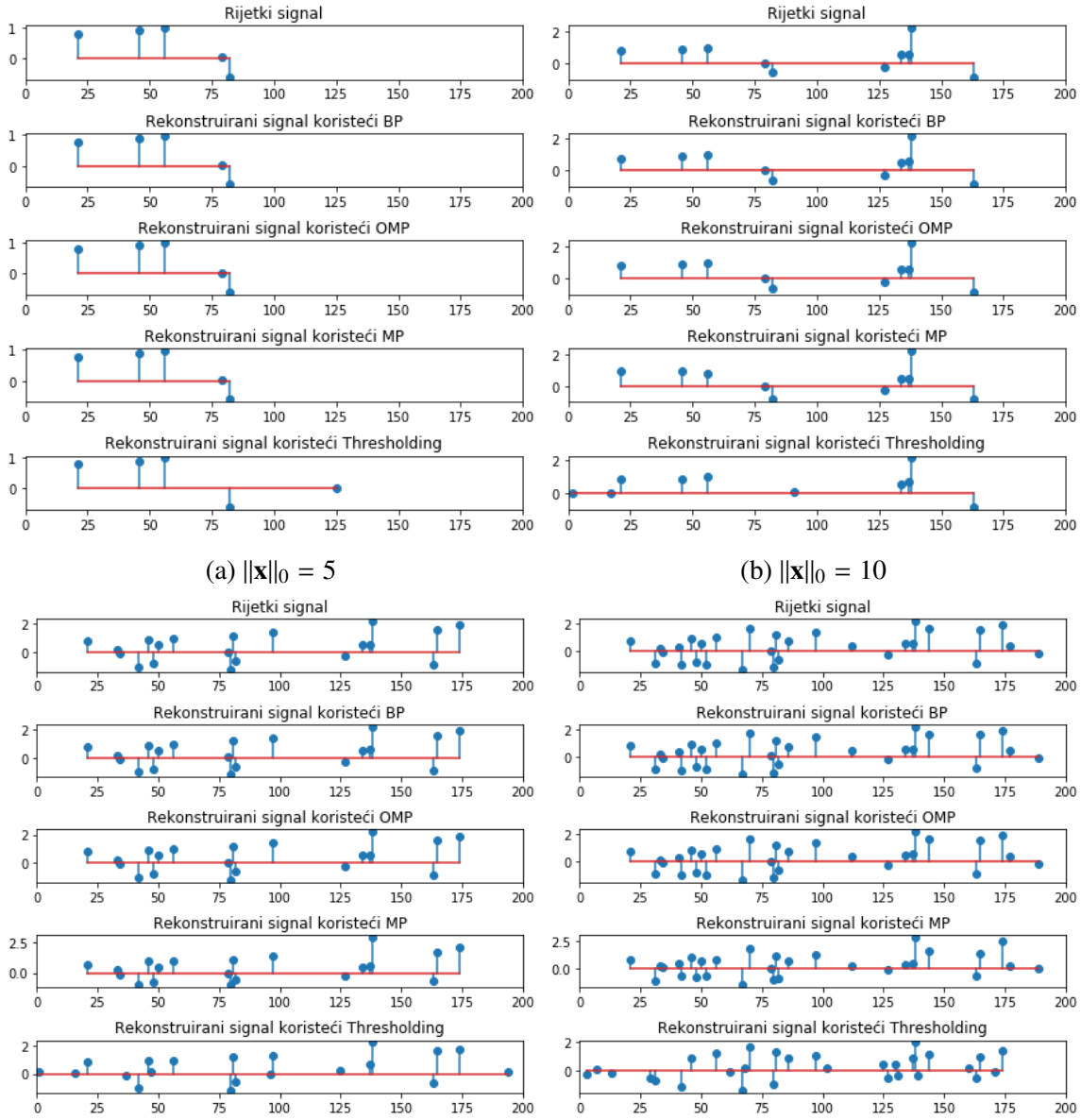
cija ukoliko za njega vrijedi

$$\begin{aligned} \max(k_{p,OMP}, k_{q,OMP}) &< 1.4217 \\ \|\mathbf{x}_{BP}\|_0 &< 2.5994 \\ \|\mathbf{x}_T\|_0 &< 1.5439, \end{aligned}$$

to jest rješenje bi trebalo imati 1 ili 2 ne-nul elemenata kako bi bilo jedinstveno. U praksi se pokazalo kako su gornje ocjene prilično rigorozne, s obzirom na to da algoritmi s pragom dobro aproksimiraju rješenje i za gušće vektore. Na slici 3.2 vidimo ponašanje greške pri povećanju broja ne-nul elemenata rijetke reprezentacije. Primijetimo kako BP i OMP algoritmi najbolje aproksimiraju rijetku reprezentaciju, dok algoritam s pragom tolerancije i MP algoritam daju nešto lošije rezultate s povećanjem broja ne-nul elemenata. S obzirom na to da težimo što rjeđoj reprezentaciji, možemo reći da u tom kontekstu i algoritam s pragom i MP algoritam dosta dobro funkcioniraju za $\|\mathbf{x}\|_0 < 7$*. BP algoritam počinje griješiti za* $\|\mathbf{x}\|_0 > 30$*, dok OMP algoritam velikim pogreškama odolijeva sve do* $\|\mathbf{x}\|_0 \approx 50$*. Pogreške algoritama potrage za neke vrijednosti* $\|\mathbf{x}\|_0$ *dane su u tablici 3.1. Stupac APT označava pogrešku algoritma s pragom tolerancije. Prikaz dobivene aproksimacije rijetkog vektora prikazan je na slici 3.1.*

$\|\mathbf{x}\|_0$	BP	OMP	MP	APT
1	8.87e-08	3.47e-18	3.47e-18	3.46e-18
2	8.33e-08	2.25e-16	1.73e-17	3.30e-02
3	4.15e-07	5.67e-16	1.11e-16	3.02e-02
4	5.66e-07	3.70e-16	5.09e-16	3.03e-02
5	8.90e-07	4.33e-16	6.85e-16	3.03e-02
10	5.07e-07	9.94e-16	0.3207	0.6844
15	2.77e-06	3.21e-15	0.6406	1.4070
20	2.53e-06	3.66e-15	0.8933	1.2130
30	7.70e-06	4.40e-15	1.2316	2.5067
40	1.1387	4.51e-15	3.0499	3.4149

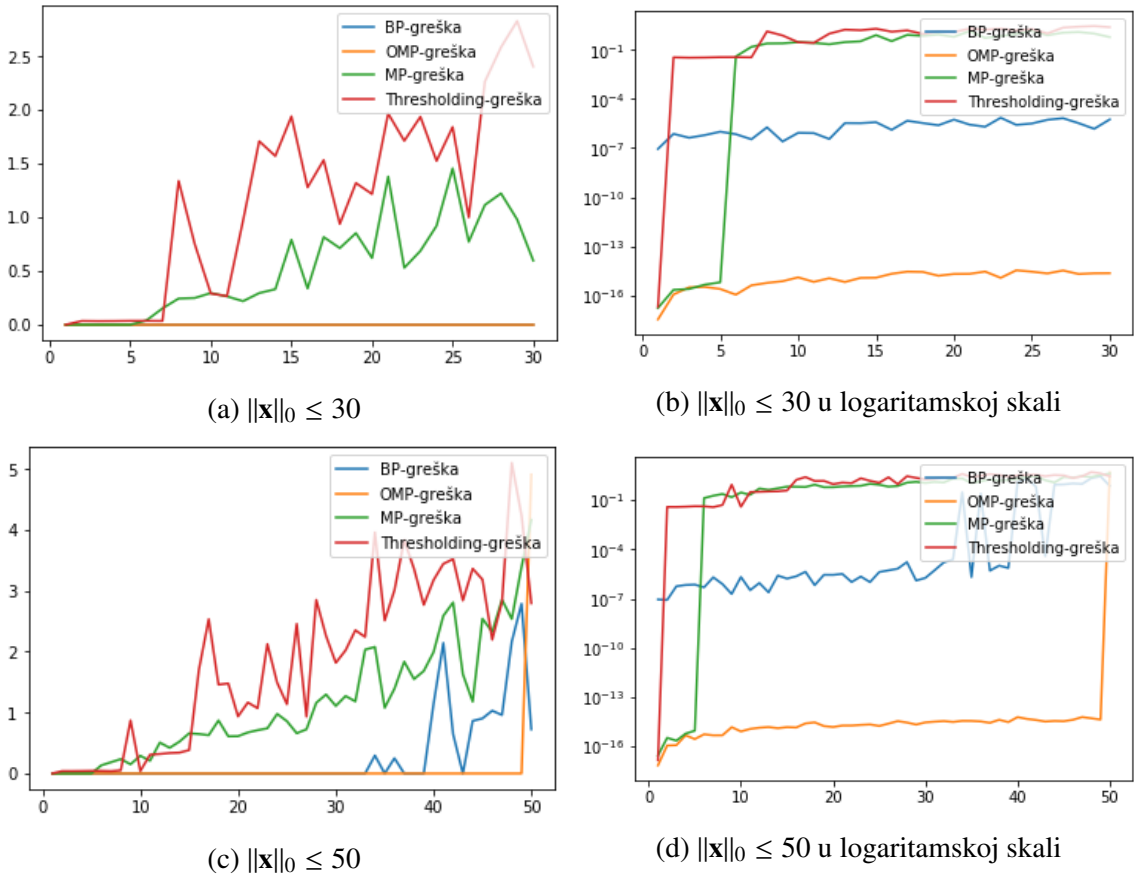
Tablica 3.1: Tablica grešaka za algoritme potrage u slučaju sparenih ortogonalnih matrica.



(a) $\|\mathbf{x}\|_0 = 5$ (b) $\|\mathbf{x}\|_0 = 10$

(c) $\|\mathbf{x}\|_0 = 20$ (d) $\|\mathbf{x}\|_0 = 30$

Slika 3.1: Rijetki signal i aproksimacije rijetkog signala algoritmima potrage u slučaju sparenih ortogonalnih matrica.



(a) $\|\mathbf{x}\|_0 \leq 30$

(b) $\|\mathbf{x}\|_0 \leq 30$ u logaritamskoj skali

(c) $\|\mathbf{x}\|_0 \leq 50$

(d) $\|\mathbf{x}\|_0 \leq 50$ u logaritamskoj skali

Slika 3.2: Pogreške algoritama potrage u slučaju sparenih ortogonalnih matrica.

Primjer 3.3.2 (Općeniti slučaj)**.** *Generirana je matrica rječnika* $\mathbf{D}$ *dimenzija* 100×500*, tj. rječnik se sastoji od* 500 *atoma iz* $\mathbb{R}^{100}$*.* $\mathbf{D}$ *je punog ranga (rang iznosi* 100*) te su stupci normirani. Generiran je i vektor desne strane* $\mathbf{b} \in \mathbb{R}^{100}$ *te rijedak vektor* $\mathbf{x} \in \mathbb{R}^{500}$ *koji ima 2 ne-nul elementa kojeg ćemo pokušati rekonstruirati. Označimo s* $\mathbf{x}_{OMP}, \mathbf{x}_{MP}, \mathbf{x}_T, \mathbf{x}_{BP}$ *aproksimacije vektora* $\mathbf{x}$ *dobivene algoritmima OMP, MP, algoritam s pragom tolerancije i BP respektivno. Međusobna koherencija ove matrice iznosi* 0.4059*, ocjene za jedinstvenost rijetke reprezentacije iz teorema 3.1.2, 3.1.3, 3.2.2 iznose*

$$\|\mathbf{x}_{OMP}\|_0,\ \|\mathbf{x}_{MP}\|_0,\ \|\mathbf{x}_{BP}\|_0 < 1.73187$$
$$\|\mathbf{x}_T\|_0 < 1.13114.$$

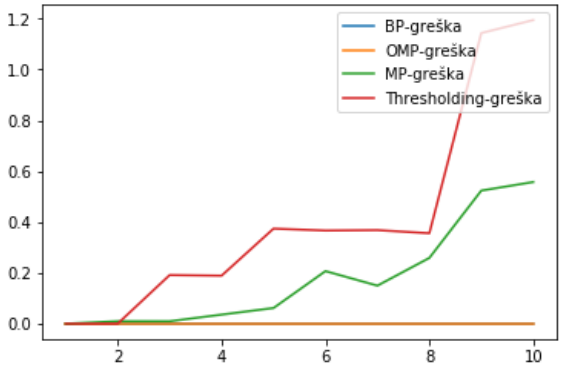
Primijetimo da su ocjene na rješenje za algoritme OMP, MP i BP jednake, dok je ocjena za algoritam s pragom tolerancije nešto manja. U svakom slučaju, ove ocjene nam govore kako rješenje mora imati samo jedan ne-nul element kako bi ono bilo jedinstveno.

Na slici 3.4 vidimo kako svi algoritmi vrlo dobro aproksimiraju rješenje za vektore s jednim ili 2 ne-nul elementa, što je i očekivano ponašanje s obzirom na teorijsku ocjenu. Rigoroznost teorijske ocjene uočavamo u činjenici da većina algoritama potrage vrlo dobro procjenjuje vektore s 3 ili 4 ne-nul elementa kao i mnogo gušće vektore, što možemo vidjeti na slici 3.5. Nešto veću pogrešku uočavamo kod algoritma s pragom i MP algoritma za $\|\mathbf{x}\|_0 \geq 10$*, dok BP i OMP algoritam vrlo dobro procjenjuju vektore sve do* $\|\mathbf{x}\|_0 \leq 25$*.*

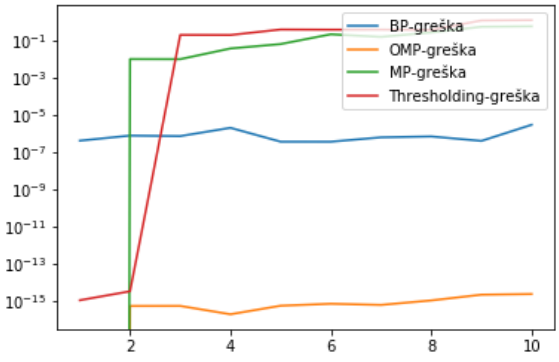
Greška, očekivano, raste s porastom broja ne-nul elemenata. Vrijednosti grešaka za neke vrijednosti dane su u tablici 3.2, a na slici 3.3 vidimo ponašanje pogreške za različite vrijednosti $\|\mathbf{x}\|_0$*. Primjećujemo kako OMP i BP algoritam funkcioniraju izvrsno za* $\|\mathbf{x}\|_0 \leq 25$*, nakon čega BP počinje sve gore i gore aproksimirati rijetku reprezentaciju, dok OMP algoritam i dalje procjenjuje s minimalnom pogreškom, sve do* $\|\mathbf{x}\|_0 = 37$*. MP i algoritam s pragom tolerancije su vidno slabiji, posebice za nešto gušće vektore, iako su, radi jednostavnosti, ponekad pogodniji za aproksimaciju vrlo rijetkih reprezentacija vektora.*

$\|\mathbf{x}\|_0$	BP	OMP	MP	APT
1	3.89e-07	0	0	5.55e-16
2	7.31e-07	3.33e-16	9.57e-03	7.02e-16
3	6.87e-07	4.23e-16	9.47e-03	0.1840
4	1.91e-06	3.34e-16	3.55e-02	0.1784
10	2.79e-06	1.57e-15	0.5572	1.1943
17	2.25e-06	2.01e-15	1.1649	2.1726
25	1.33e-04	3.27e-15	2.2579	3.9586
50	3.2705	8.5085	5.2892	6.3391
100	7.2257	13.4712	8.6753	37.5685

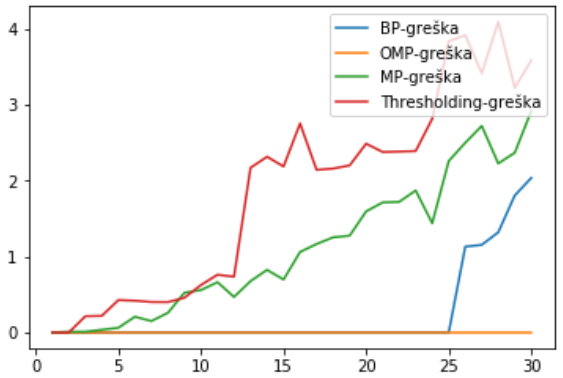
Tablica 3.2: Tablica grešaka za algoritme potrage u općenitom slučaju.



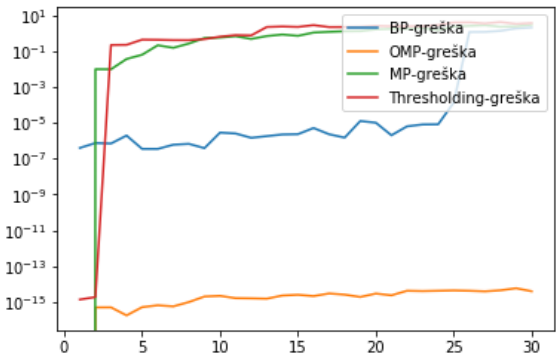
(a) $\|\mathbf{x}\|_0 \leq 10$



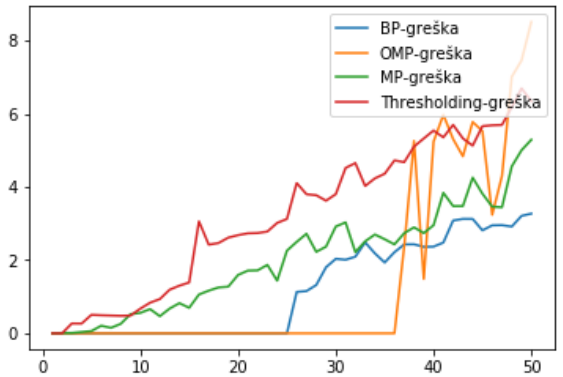
(b) $\|\mathbf{x}\|_0 \leq 10$ u logaritamskoj skali



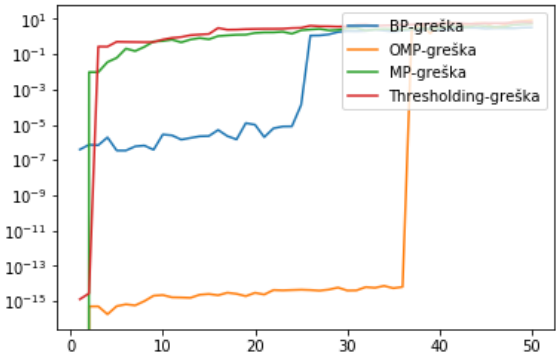
(c) $\|\mathbf{x}\|_0 \leq 30$



(d) $\|\mathbf{x}\|_0 \leq 30$ u logaritamskoj skali

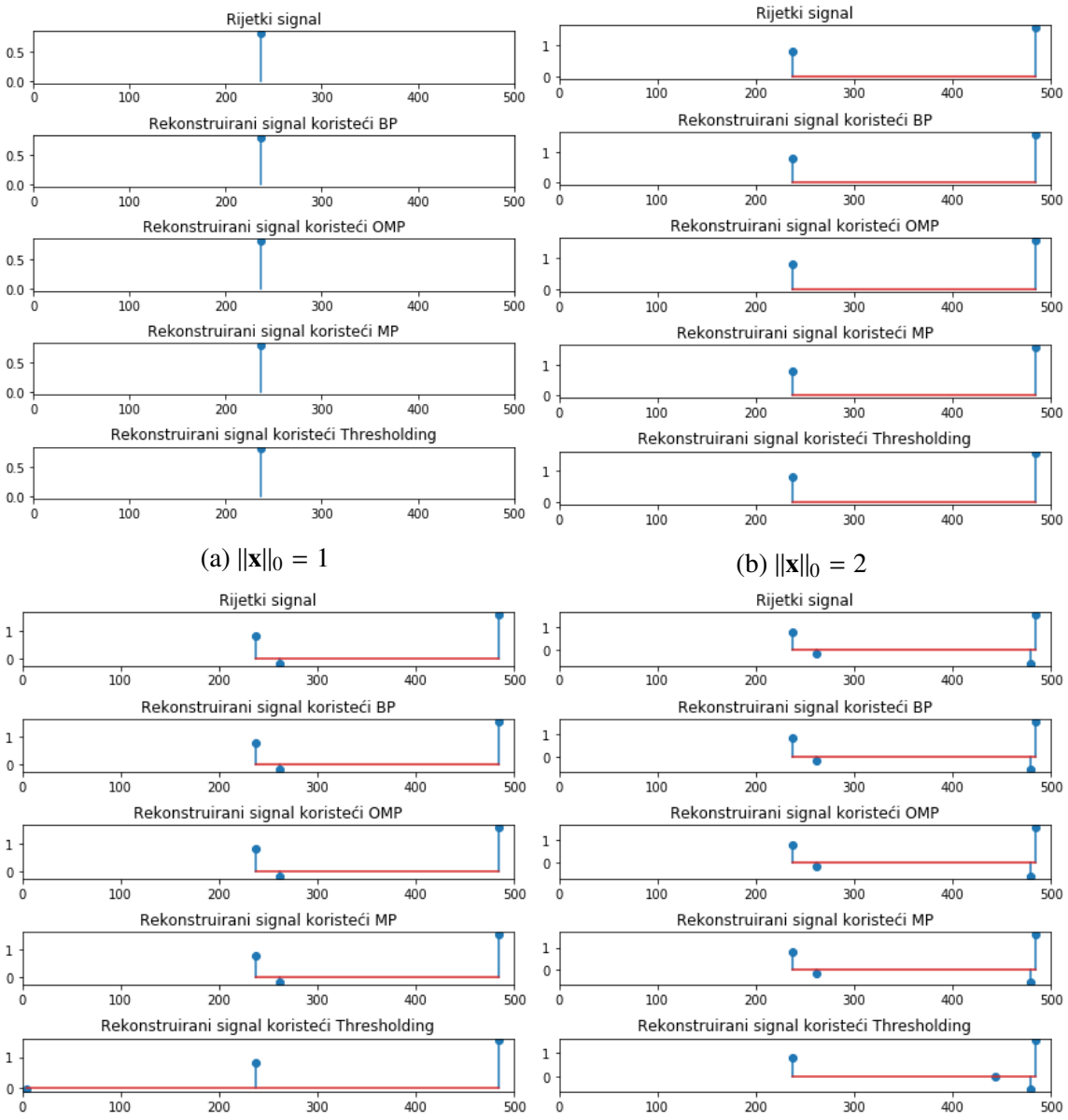


(e) $\|\mathbf{x}\|_0 \leq 50$



(f) $\|\mathbf{x}\|_0 \leq 50$ u logaritamskoj skali

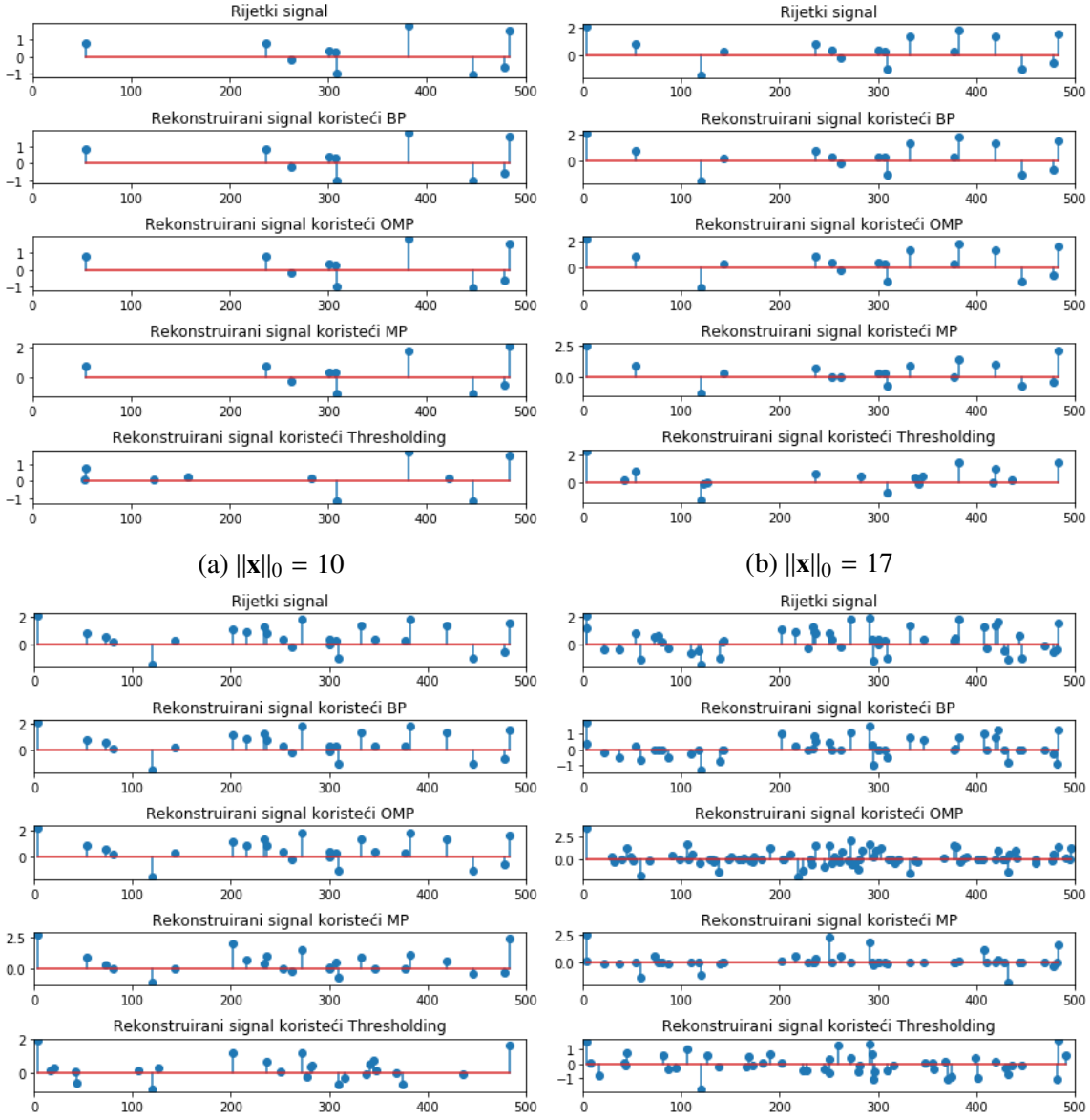
Slika 3.3: Pogreške algoritama potrage u općenitom slučaju.



(a) $||\mathbf{x}||_0 = 1$ (b) $||\mathbf{x}||_0 = 2$

(c) $||\mathbf{x}||_0 = 3$ (d) $||\mathbf{x}||_0 = 4$

Slika 3.4: Vrlo rijetki signal i aproksimacije vrlo rijetkog signala algoritmima potrage u općenitom slučaju.



(a) $\|\mathbf{x}\|_0 = 10$ (b) $\|\mathbf{x}\|_0 = 17$

(c) $\|\mathbf{x}\|_0 = 25$ (d) $\|\mathbf{x}\|_0 = 50$

Slika 3.5: Rijetki signal i aproksimacije rijetkog signala algoritmima potrage u općenitom slučaju.

Poglavlje 4

Rijetki modeli i strojno učenje

Rijetka reprezentacija svoju primjenu pronašla je u brojnim aspektima podatkovne znanosti. Ona omogućuje kompaktan zapis podatka s obzirom na rječnik, što uvelike smanjuje potrebnu memoriju za pohranu velikog skupa podataka. Također se koristi za uklanjanje šuma iz signala ili sa slike, rekonstrukciju signala i slično. U ovom radu opisat ćemo primjenu rijetke reprezentacije u dubokom strojnom učenju kroz konvolucijski model rijetke reprezentacije *Convolutional Sparse Coding Model*) (CSC). Rijetka reprezentacija u strojnom učenju može se koristiti i za nadzirano i za nenadzirano učenje. Nadziranim učenjem smatramo probleme u kojima imamo skup za učenje koji se sastoji od primjera (signala) i njihovih oznaka (npr. u problemu klasifikacije oznaka je indeks klase kojoj primjer pripada) te pomoću pogreške aproksimacije učimo parametre modela, a kod nenadziranog učenja ne poznajemo oznake primjera i želimo naći pravilnost u podacima (npr. problem grupiranja podataka). Za početak ćemo dati kratak uvod u duboko učenje i konvolucijske mreže. Zatim ćemo opisati problem učenja rječnika, a nakon toga bavit ćemo se primjenom rijetke reprezentacije u konvolucijskim mrežama.

4.1 Konvolucijske mreže

Prije svega, definirajmo pojam konvolucije koja nam omogućuje modeliranje lokalne interakcije dviju funkcija.

Definicija 4.1.1. *Konvolucija dviju funkcija* $f, g : K \to \mathbb{R}$, $K \subseteq \mathbb{R}^m$, *je funkcija* $f * g$ *definirana s*

$$(f * g)(\mathbf{x}) = \int_K f(\mathbf{t}) g(\mathbf{x} - \mathbf{t}) d\mathbf{t} \tag{4.1}$$

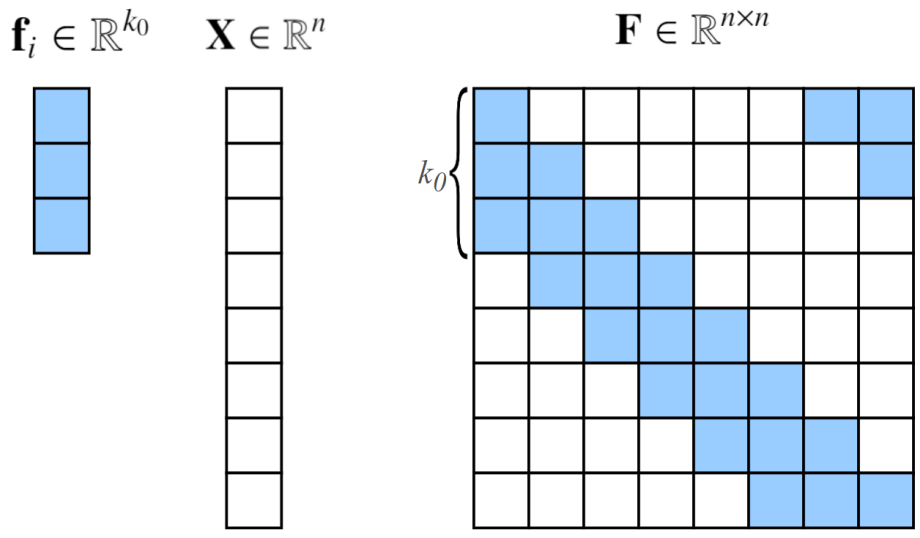
U diskretnom slučaju, kao što je naš, konvolucija funkcija $f, g : \mathbb{N} \to \mathbb{R}$ dana je s

$$(f * g)(n) = \sum_{k=-\infty}^{+\infty} f(k)g(n-k). \tag{4.2}$$

Kratki uvod u konvolucijske mreže dajemo na primjeru jednodimenzionalnog signala, vektora $\mathbf{X} \in \mathbb{R}^n$. Osnovni algoritam koji se koristi kod dubokog učenja je prolazak unaprijed (engl. *forward pass*). U prvom koraku algoritma primljeni signal, vektor $\mathbf{X}$, konvoluiramo s m_1 filtera duljine k_0 koje još nazivamo i jezgrama. Filteri su vektori iz $\mathbb{R}^{k_0}$ koje primijenjujemo na svaki podvektor vektora $\mathbf{X}$ duljine k_0. Dobiveni skalar predstavlja odaziv podvektora na filter. Kažemo da smo na vektor $\mathbf{X}$ primijenili filter sa svim njegovim pomacima (engl. *shift*). Intuitivno možemo reći da filteri traže svojstva kroz cijeli vektor $\mathbf{X}$ pa rezultat primjene filtera na svakom podvektoru možemo shvatiti kao sličnost podvektora i filtera tj. je li promatrano svojstvo pronađeno i u kojoj mjeri. Na slici 4.1 ilustriran je filter $\mathbf{f}$, vektor $\mathbf{X}$ i svi pomaci filtera po vektoru $\mathbf{X}$. Sve pomake svih filtera po vektoru $\mathbf{X}$ možemo zapisati kao

$$\mathbf{W}_1^T \mathbf{X} \in \mathbb{R}^{nm_1} \tag{4.3}$$

gdje je $\mathbf{W}_1 \in \mathbb{R}^{n \times nm_1}$ matrica čiji su stupci svi filteri sa svojim pomacima, tj. prvi $n \times n$ blok je prvi filter sa svim svojim pomacima, sljedeći $n \times n$ blok je drugi filter sa svim svojim pomacima, a posljednji $n \times n$ blok je posljednji filter sa svim svojim pomacima.



Slika 4.1: Filter $\mathbf{f}$, vektor $\mathbf{X}$, svi pomaci filtera $\mathbf{f}$ po vektoru $\mathbf{X}$. Matrica $\mathbf{W}_1$ sastoji se od m blokova tipa $\mathbf{F}$.

Nakon sloja filtera slijedi sloj aktivacijske funkcije koja djeluje po elementima vektora $\mathbf{W}_1^T \mathbf{X}$ kojem dodajemo vektor $\mathbf{b}_1 \in \mathbb{R}^{nm_1}$ koji predstavlja vrstu pristranosti ili praga.

Najčešće korištena aktivacija funkcija je ReLU-funkcija (engl. *Rectifier Linear Unit*) definirana sa

$$\mathrm{ReLU}(z) = \max(z, 0). \tag{4.4}$$

Izlaz prvog sloja - konvolucije i aktivacijske funkcije je

$$\mathbf{Z}_1 = \mathrm{ReLU}\Big(\mathbf{W}_1^T\mathbf{X} + \mathbf{b}_1\Big). \tag{4.5}$$

Sljedeći sloj kreiramo na isti način vodeći računa o dimenzijama. Konvolucijska matrica je sada $\mathbf{W}_2 \in \mathbb{R}^{nm_1 \times nm_2}$ i sastoji se od m_2 filtera duljine $k_1 m_1$, a vektor pristranosti je $\mathbf{b}_2 \in \mathbb{R}^{nm_2}$. Sada je izlaz nakon drugog sloja

$$\mathbf{Z}_2 = \mathrm{ReLU}\Big(\mathbf{W}_2^T\mathbf{Z}_1 + \mathbf{b}_2\Big) = \mathrm{ReLU}\Bigg(\mathbf{W}_2^T\ \mathrm{ReLU}\Big(\mathbf{W}_1^T\mathbf{X} + \mathbf{b}_1\Big) + \mathbf{b}_2\Bigg). \tag{4.6}$$

Slijedeći ovu shemu, možemo slagati neuronske mreže veće dubine.

Ovime smo prikazali osnovnu strukturu konvolucijske neuronske mreže. Napomenimo još da se ponekad pojavljuju dodatni slojevi koje ovdje nismo spomenuli kao što je npr. sloj sažimanja (engl. *pooling layer*) koji skupu prostorno bliskih značajki na ulazu pridružuje jednu značajku na izlazu kako bi se smanjila dimenzionalnost ulaza. Također, ilustrirali smo strukturu samo za slučaj jednodimenzionalnog signala na ulazu, no ona se može poopćiti i na način da obrađuje matrice tj. slike.

U primjeni se konvolucijske neuronske mreže koriste za primjerice klasifikaciju signala i slika. Tada se izlaz posljednjeg sloja daje kao ulaz klasifikacijskog algoritma. Kod nadziranog učenja potrebno je naučiti sve parametre mreže - matrice $\mathbf{W}_i$, vektore $\mathbf{b}_i$ kao i parametre klasifikatora. Ako parametre klasifikatora označimo s $\mathbf{U}$ broj slojeva mreže s K, učenje parametara mreže se može zapisati kao minimizacijski problem

$$\min_{\mathbf{W}_i, \mathbf{b}_i, \mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j), \mathbf{U}, \mathbf{Z}_K\Big) \tag{4.7}$$

gdje je ℓ funkcija gubitka koja daje vrijednosti kojima kažnjavamo pogrešno klasificirane vektore. Pogrešku možemo izračunati zbog toga što znamo točne oznake primjera $h(\mathbf{X})$.

4.2 Učenje rječnika

Dosadašnja analiza potrage za rijetkom reprezentacijom bazirala se na pretpostavci da poznajemo rječnik koji dovoljno prorjeđuje dane podatke. Međutim, to nije uvijek slučaj. Za prikaz određenih signala pogodni su već ustaljeni Wavelet i Fourirer rječnici, no pokazuje se da su ipak ograničeni za općenitu primjenu. Ne možemo očekivati postojanje jednog globalnog rječnika za sve probleme rijetke reprezentacije. Rječnik ovisi o skupu signala

koje obrađujemo i najveću efikasnost i točnost reprezentacije postižemo upravo kada je rječnik naučen iz podataka. Primjerice, imamo rječnik slika svih namirnica u nekom supermarketu, a promatramo problem klasifikacije voća. Takav je rječnik preopširan s obzirom na to da su nam dovoljne samo slike voća kako bismo opisali svoje podatke. S druge strane, ako imamo rječnik voća, a želimo njime opisati i povrće, zaključujemo da je takav rječnik preuzak.

Zadatak učenja rječnika iz skupa signala $\{\mathbf{X}_j\}_j$ možemo zapisati kao

$$\min_{\mathbf{D},\{\mathbf{\Gamma}^j\}_j} \sum_j \|\mathbf{X_j} - \mathbf{D}\mathbf{\Gamma}^j\|_2^2 + \xi\|\mathbf{\Gamma}^j\|_0 \tag{4.8}$$

što je problem nenadziranog učenja. Sada algoritmima potrage pronađemo $\mathbf{\Gamma}^j$ za svaki ulazni signal $\mathbf{X}_j$ uz rječnik $\mathbf{D}$ tako dobivši

$$\mathbf{\Gamma}^*(\mathbf{X}_j, \mathbf{D}) = \arg\min_{\mathbf{\Gamma}} \Big\{\|\mathbf{\Gamma}\|_0 \ : \ \mathbf{D}\mathbf{\Gamma} = \mathbf{X}_j\Big\}. \tag{4.9}$$

Učenje rječnika sada postaje minimizacija pogreške reprezentacije

$$\min_{\mathbf{D}} \sum_j \|\mathbf{X}_j - \mathbf{D}\mathbf{\Gamma}_j^*\|_2^2. \tag{4.10}$$

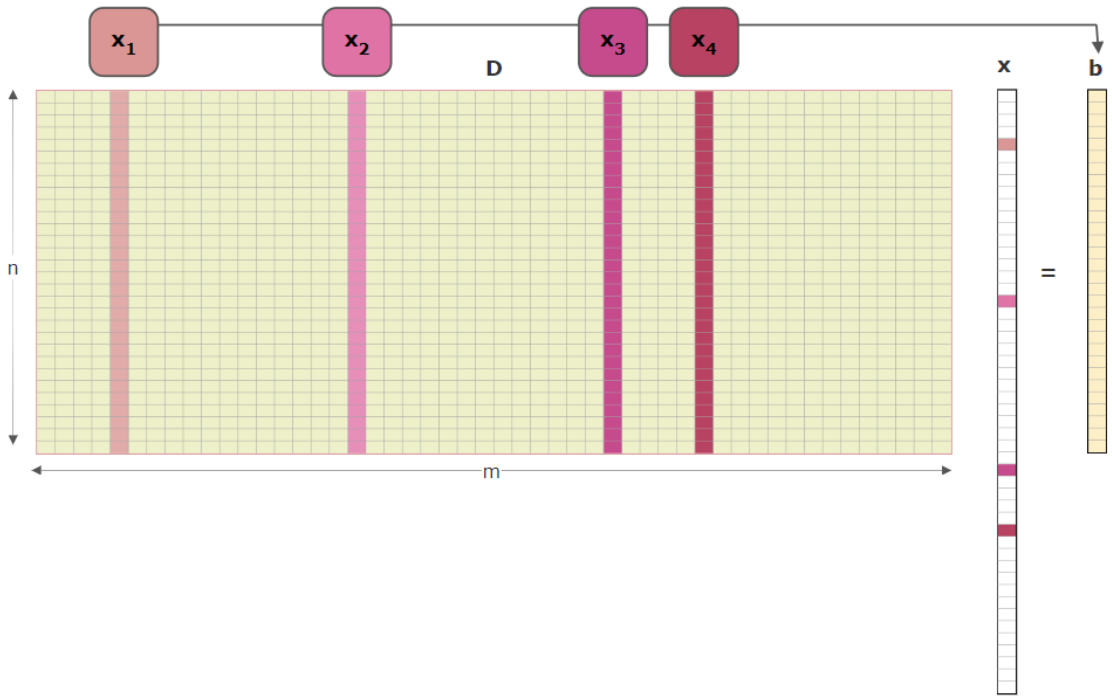
Ukoliko rijetku reprezentaciju koristimo za nadziranu klasifikaciju (uz poznate oznake primjera $h(\mathbf{X}_j)$), učenje rječnika može se provoditi paralelno s učenjem parametara klasifikatora $\mathbf{U}$ na način da mu predajemo dobivene rijetke reprezentacije $\mathbf{\Gamma}^j$ i minimiziramo pogrešku klasifikacije koju zapisujemo pomoću funkcije gubitka ℓ

$$\min_{\mathbf{D},\mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j), \mathbf{U}, \mathbf{\Gamma}^*(\mathbf{X}_j, \mathbf{D})\Big). \tag{4.11}$$

Više o učenju rječnika može se pronaći u [9].

4.3 Konvolucijska rijetka reprezentacija

Model rijetke reprezentacije je generativni model - opisuje na koji način je podatak nastao koristeći k atoma iz rječnika $\mathbf{D}$, kao što je ilustrirano na slici 4.2. Ovim smo se problemom bavili u prethodnim poglavljima rada. S obzirom na to da rijetku reprezentaciju tražimo za cijeli vektor, ovaj pristup nazivamo globalnim. U nastavku ćemo se baviti lokalnom verzijom koja je pogodnija za konvolucijske mreže zbog toga što je na taj način moguće prikazati lokalnu strukturu signala. Takav model nazivamo konvolucijskim modelom rijetke reprezentacije (CSC), a baziran je na pretpostavci da svaki manji dio signala ima svoju rijetku reprezentaciju s obzirom na lokalizirani rječnik.



Slika 4.2: Prikaz rijetke reprezentacije signala $\mathbf{b}$ kao $\mathbf{b} = \mathbf{D}\mathbf{x}$ gdje je $\|\mathbf{x}\|_0 = 4$.

Kao i u prvom potpoglavlju gdje smo ukratko opisali konvolucijske neuronske mreže, pretpostavljamo da su promatrani signali vektori, iako se analogna analiza može provesti i za matrice ili tenzore, koji reprezentiraju slike i videa, itd. S $\mathbf{X} \in \mathbb{R}^n$ označimo globalni signal, s $\{\mathbf{d}_i\}_{i=1}^m \in \mathbb{R}^k$ označimo m lokalnih filtera duljine $k << n$, te s $\mathbf{\Gamma}_i \in \mathbb{R}^n$ označimo rijetke vektore. CSC model temelji se, kao i model konvolucijskih neuronskih mreža, na konvoluciji filtera s vektorom. U tu svrhu konstruirat ćemo m matrica koje predstavljaju konvoluciju svakog filtera $\mathbf{d}_i$ s vektorom $\mathbf{\Gamma}_i$. Takve matrice su cirkularnog oblika,

$$\mathbf{C}_i = \begin{bmatrix} c_0 & c_{n-1} & c_{n-2} & \dots & c_1 \\ c_1 & c_0 & c_{n-1} & \dots & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & \vdots & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \dots & c_1 & c_0 \end{bmatrix}, \tag{4.12}$$

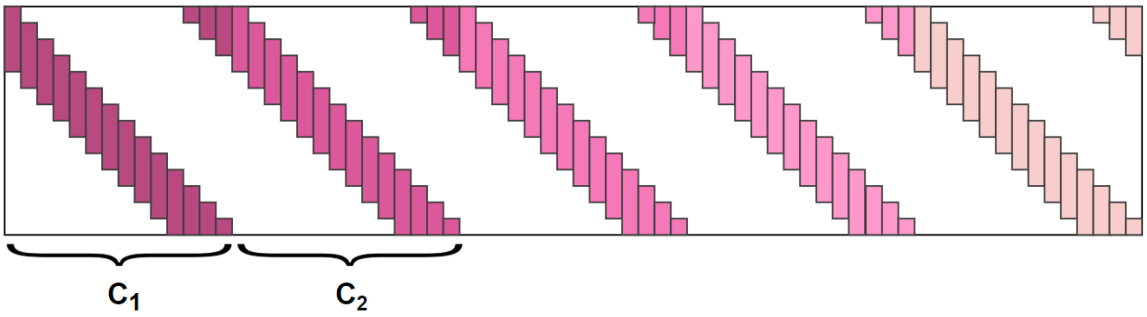
gdje na samo k dijagonala stoje ne-nul elementi. Primijetimo sličnost matrice $\mathbf{C}_i$ s matri-

com **F** na slici 4.1. Sada globalni signal **X** možemo zapisati kao

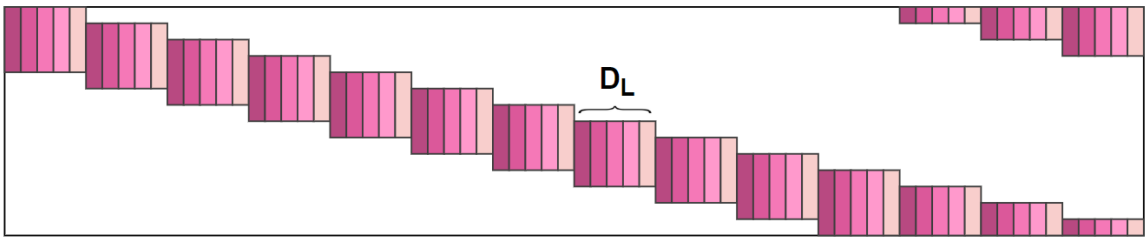
$$\mathbf{X} = \sum_{i=1}^{m} \mathbf{C}_i \boldsymbol{\Gamma}_i = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 & \dots & \mathbf{C}_m \end{bmatrix} \begin{bmatrix} \boldsymbol{\Gamma}_1 \\ \boldsymbol{\Gamma}_2 \\ \vdots \\ \boldsymbol{\Gamma}_m \end{bmatrix} = \mathbf{D}\boldsymbol{\Gamma}. \tag{4.13}$$

Ovdje je $\mathbf{D} \in \mathbb{R}^{n \times mn}$ globalni rječnik dobiven konkatenacijom matrica $\mathbf{C}_i$, a $\boldsymbol{\Gamma} \in \mathbb{R}^{mn}$ je globalni rijetki vektor reprezentacije dobiven spajanjem lokalnih rijetkih vektora. Permutiranjem stupaca dobivamo blok-dijagonalnu matricu $\tilde{\mathbf{D}}$ (uz ostatak na gornjem desnom kutu) u kojoj je svaki blok jednak. Tako dobivene blokove nazivamo lokalnim rječnicima i označavamo s $\mathbf{D}_L \in \mathbb{R}^{n \times m}$. Njihov je i-ti stupac upravo filter $\mathbf{d}_i$. Kažemo da matrica $\tilde{\mathbf{D}}$ ima konvolucijsku strukturu.

Slika 4.3 ilustrira opisan način kreiranja rječnika **D** pomoću cirkularnih matrica. Na slici 4.3a vidimo prikaz matrice rječnika kao konkatenirane cirkularne matrice što ima strukturu konvolucijske matrice. Na slici 4.3b vidimo blok-dijagonalnu strukturu koju dobivamo permutacijom stupaca matrice **D**.



(a) Matrica rječnika **D** kao konkatekacija cirkularnih matrica $\mathbf{C}_i$.



(b) Konvolucijska matrica rječnika $\tilde{\mathbf{D}}$ s lokalnim rječnicima $\mathbf{D}_L$.

Slika 4.3: Različiti prikazi matrice rječnika.

4.4 Lokalno rijetka reprezentacija

Pretpostavimo sada da želimo uzeti dio $\mathbf{p}_i$ iz $\mathbf{X}$ duljine k. Kako bismo to učinili, definirajmo operator ekstrakcije $\mathbf{R}_i \in \mathbb{R}^{k\times n}$. Sada imamo

$$\mathbf{p}_i = \mathbf{R}_i\mathbf{X} = \mathbf{R}_i\mathbf{D}\boldsymbol{\Gamma}. \tag{4.14}$$

Množenje $\mathbf{R}_i\mathbf{D}$ reprezentira ekstrakciju k redaka iz rječnika $\mathbf{D}$, no većina ekstrahiranih elemenata jednaka je 0. Stoga ćemo, kako bismo izbjegli nepotrebne nule, uvodimo operator $\mathbf{S}_i \in \mathbb{R}^{(2k-1)m\times mn}$ koji uzima samo ne-nul stupce. Imamo

$$\mathbf{R}_i\mathbf{D} = \mathbf{R}_i\mathbf{D}\mathbf{S}_i^T\mathbf{S}_i. \tag{4.15}$$

Definirajmo dva vrlo važna pojma:

Definicija 4.4.1. *Za globalni rijetki vektor* $\boldsymbol{\Gamma}$ *definiramo njegovu i-tu traku s* $\gamma_i = \mathbf{S}_i\boldsymbol{\Gamma} \in \mathbb{R}^{(2k-1)m}$.

Definicija 4.4.2. *Za konvolucijski rječnik* $\mathbf{D} \in \mathbb{R}^{n\times nm}$ *koji se sastoji od lokalnih rječnika* $\mathbf{D}_L \in \mathbb{R}^{n\times m}$*, definiramo trakasti rječnik* $\boldsymbol{\Omega} \in \mathbb{R}^{k\times(2k-1)m}$ *dobiven iz* $\mathbf{D}$ *ekstrakcijom n uzastopnih redaka te uklanjanjem nulstupaca, tj.* $\boldsymbol{\Omega} = \mathbf{R}_i\mathbf{D}\mathbf{S}_i^T$.

Sada dio $\mathbf{p}_i$ možemo zapisati kao

$$\mathbf{p}_i = \mathbf{R}_i\mathbf{D}\mathbf{S}_i^T\mathbf{S}_i\boldsymbol{\Gamma} = \boldsymbol{\Omega}\gamma_i. \tag{4.16}$$

Primijetimo da $\boldsymbol{\Omega}$ ne ovisi o i. Zbog toga vrijedi

$$\mathbf{p}_{i+1} = \mathbf{R}_{i+1}\mathbf{X} = \boldsymbol{\Omega}\gamma_{i+1}, \tag{4.17}$$

odnosno svi dijelovi $\mathbf{p}_i$ iz $\mathbf{X}$ imaju svoju rijetku reprezentaciju s obzirom na zajednički rječnik $\boldsymbol{\Omega}$. Za mjerenje rijetkosti koristimo $\ell_{0,\infty}$-normu (nije formalna norma) koja, za razliku od ℓ_0-norme, uzima u obzir lokalnu rijetkost vektora. Definiramo je kao maksimalan broj ne-nul komponenti u trakama iz $\boldsymbol{\Gamma}$:

Definicija 4.4.3. *Za globalni vektor* $\boldsymbol{\Gamma}$ *definiramo* $\ell_{0,\infty}$*-normu (lokalnu kardinalnost) s*

$$\|\boldsymbol{\Gamma}\|_{0,\infty}^s = \max_{1\le i\le n} \|\gamma_i\|_0. \tag{4.18}$$

Ituitivno, ako je $\|\Gamma\|_{0,\infty}^s$ mala vrijednost, to znači da su sve trake rijetke te stoga svaki dio $\mathbf{p}_i$ ima rijetku reprezentaciju s obzirom na rječnik $\boldsymbol{\Omega}$. Pokazuje se da za konvolucijski rijetki model vrijede analogne tvrdnje o jedinstvenosti rješenja koje su vrijedile i za rijetki model promatran na početku ovog rada, te da algoritmi potrage vrlo uspješno rješavaju ovaj problem, o čemu se detaljnije može naći u [8].

Za dani signal $\mathbf{X}$, pronalazak njegove rijetke reprezentacije $\mathbf{\Gamma}$ u kontekstu norme $\ell_{0,\infty}$ opisan je sljedećim optimizacijskim problemom

$$(P_{0,\infty}): \quad \min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\|_{0,\infty}^{s} \quad \text{t.d.} \quad \mathbf{D\Gamma} = \mathbf{X}. \tag{4.19}$$

Zapravo tražimo globalni vektor $\mathbf{\Gamma}$ koji može rijetko reprezentirati svaki dio $\mathbf{p}_i$ signala $\mathbf{X}$ s obzirom na rječnik $\mathbf{\Omega}$. U [8] je pokazano da problem $(P_{0,\infty})$ ima jedinstveno rješenje ukoliko je broj ne-nul elemenata u svakoj traci manji od $\frac{1}{2}(1 + \frac{1}{\mu(\mathbf{D})})$ i može se pronaći koristeći algoritme potrage opisane u prethodnom poglavlju.

Nadalje, i u ovom slučaju možemo pretpostaviti da će podaci u primjeni imati neke nesavršenosti, tj. pretpostavljamo postojanje šuma u podacima. Sada je $\mathbf{Y} = \mathbf{X} + \mathbf{E} = \mathbf{D\Gamma} + \mathbf{E}$ gdje je $\mathbf{E}$ vektor šuma koji je ograničen u ℓ_2-normi. Varijanta problema $(P_{0,\infty})$ uz šum dana je s

$$(P_{0,\infty}^{\epsilon}): \quad \min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\|_{0,\infty}^{s} \quad \text{t.d.} \quad \|\mathbf{Y} - \mathbf{D\Gamma}\|_2^2 \leq \epsilon^2. \tag{4.20}$$

Za ovaj problem također znamo da ima jedinstveno rješenje, uz pretpostavku o dovoljnoj rijetkosti, koje je moguće pronaći algoritmima potrage.

4.5 Višeslojna konvolucijska rijetka reprezentacija

Glavna misao vodilja kod konstrukcije višeslojnih konvolucijskih mreža koristeći rijetku reprezentaciju je nasljednost rijetkosti. Pretpostavimo da signal $\mathbf{X} \in \mathbb{R}^n$ možemo zapisati kao produkt konvolucijskog rječnika $\mathbf{D}_1 \in \mathbb{R}^{n \times nm_1}$ koji se sastoji od m filtera duljine k_0 i rijetkog vektora $\mathbf{\Gamma}_1 \in \mathbb{R}^{nm_1}$

$$\mathbf{X} = \mathbf{D}_1\mathbf{\Gamma}_1. \tag{4.21}$$

Nadalje, pretpostavimo da se vektor $\mathbf{\Gamma}_1$ također može zapisati na isti način uz konvolucijsku matricu rječnika $\mathbf{D}_2 \in \mathbb{R}^{nm_1 \times nm_2}$ te rijetki vektor $\mathbf{\Gamma}_2 \in \mathbb{R}^{nm_2}$. $\mathbf{D}_2$ je konvolucijska matrica od m_2 filtera duljine $k_1 m_1$ sa korakom (engl. *stride*) m_1, što znači da u konvoluciji preskačemo m_1 elemenata. Sada je

$$\mathbf{\Gamma}_1 = \mathbf{D}_2\mathbf{\Gamma}_2. \tag{4.22}$$

Odnosno, promatrani signal $\mathbf{X}$ možemo zapisati kao

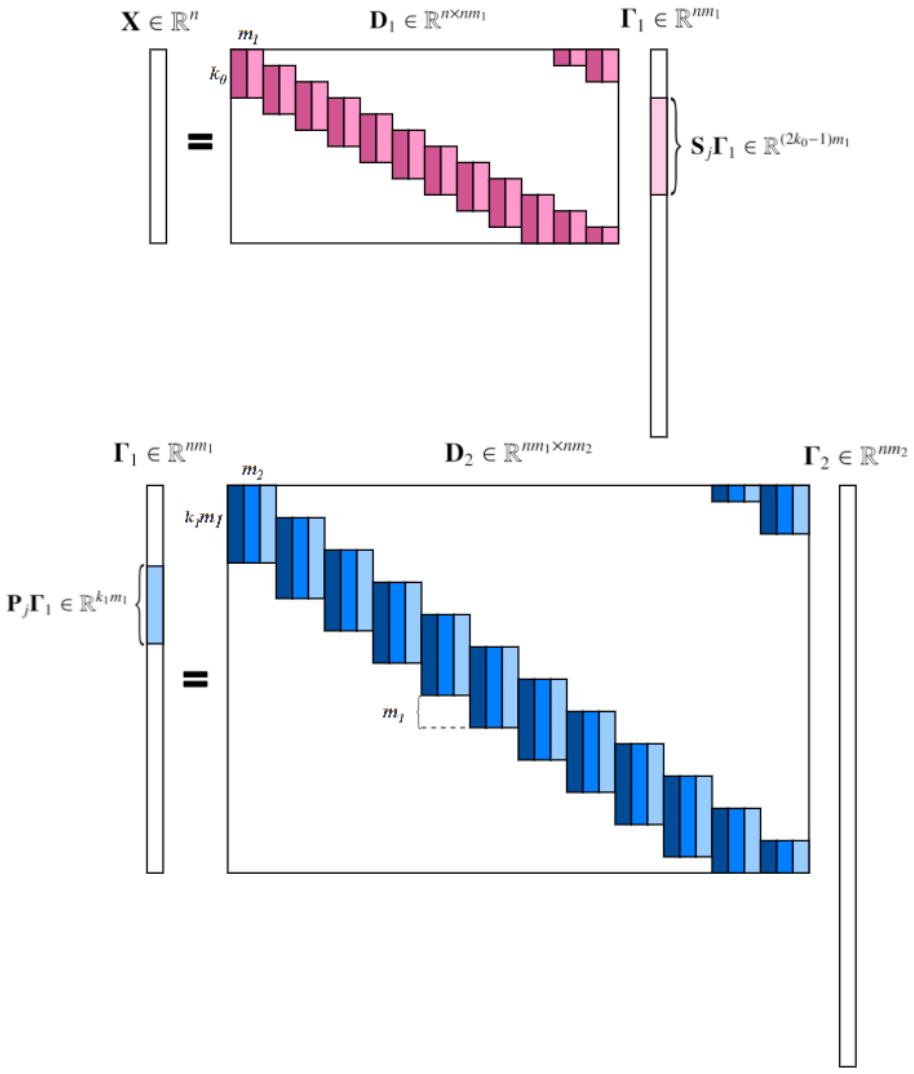
$$\mathbf{X} = \mathbf{D}_1\mathbf{D}_2\mathbf{\Gamma}_2, \tag{4.23}$$

gdje je $\mathbf{\Gamma}_2$ rijedak vektor, a matrica $\mathbf{D}_1\mathbf{D}_2$ predstavlja novi rječnik. U zapisu (4.21) $\mathbf{X}$ je dobiven kombiniranjem atoma rječnika $\mathbf{D}_1$, dok je u (4.23) dobiven kombiniranjem stupaca iz rječnika $\mathbf{D}_1\mathbf{D}_2$. Elemente tog rječnika nazivamo molekulama.

Primijetimo da u ovom konteksu vektor $\mathbf{\Gamma}_1$ ima dvije uloge:

- Iz (4.21) $\mathbf{\Gamma}_1$ je rijetka reprezentacija signala $\mathbf{X}$ uz rječnik $\mathbf{D}_1$. $\mathbf{\Gamma}_1$ se sastoji od traka $\gamma_i^1 = \mathbf{S}_i\mathbf{\Gamma}_1$ duljine $(2k_0 - 1)m_1$.
- Iz (4.22) $\mathbf{\Gamma}_1$ je signal koji ima rijetku reprezentaciju $\mathbf{\Gamma}_2$ uz rječnik $\mathbf{D}_2$. $\mathbf{\Gamma}_1$ se sastoji od dijelova $\mathbf{p}_i = \mathbf{P}_i\mathbf{\Gamma}_1$ duljine k_1m_1.

Ovaj dvojni pogled na vektor $\mathbf{\Gamma}_1$ ilustriran je na slici 4.4.



Slika 4.4: Dva različita pogleda na $\mathbf{\Gamma}_1$.

Provodeći opisani postupak više puta dolazimo do modela višeslojne konvolucijske rijetke reprezentacije dubine kojeg skraćeno nazivamo ML-CSC modelom (engl. *Multi-*

Layer Convolutional Sparse Coding Model). Neka smo postupak ponovili $K \in \mathbb{N}$ puta uz skup rječnika $\{\mathbf{D}_i\}_{i=1}^K$. ML-CSC model tada možemo zapisati kao

$$\begin{aligned} \mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1, \\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2, \\ &\vdots \\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K \end{aligned} \tag{4.24}$$

Problem pronalaska dovoljno rijetkih vektora $\{\mathbf{\Gamma}_i\}_{i=1}^K$ nazivamo problemom dubokog rijetkog kodiranja (engl. *Deep Coding Problem*) (DCP_λ) i formalno ga definiramo u nastavku.

Definicija 4.5.1. *Za globalni signal* $\mathbf{X}$*, skup konvolucijskih rječnika* $\{\mathbf{D}_i\}_{i=1}^K$ *te za vektor* $\lambda = [\lambda_1, ..., \lambda_K] \in \mathbb{R}^K$ *definiramo problem dubokog rijetkog kodiranja (*DCP_λ*) s*

$$(DCP_\lambda): \quad \textit{pronaći} \quad \{\mathbf{\Gamma}_i\}_{i=1}^K \quad \textit{t.d.} \quad \begin{aligned} \mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1, & \|\mathbf{\Gamma}_1\|_{0,\infty}^s &\leq \lambda_1 \\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2, & \|\mathbf{\Gamma}_2\|_{0,\infty}^s &\leq \lambda_2 \\ &\vdots & &\vdots \\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K, & \|\mathbf{\Gamma}_K\|_{0,\infty}^s &\leq \lambda_K \end{aligned} \tag{4.25}$$

.

Ako označimo $\mathbf{\Gamma}_0 = \mathbf{X}$, gornji problem možemo zapisati u kompaktnom obliku:

$$(DCP_\lambda): \quad \text{pronaći} \quad \{\mathbf{\Gamma}_i\}_{i=1}^K \quad \text{t.d.} \quad \mathbf{\Gamma}_{i-1} = \mathbf{D}_i\mathbf{\Gamma}_i, \quad \|\mathbf{\Gamma}_i\|_{0,\infty}^s \leq \lambda_i, \ \forall i \in \{1, ..., K\} \tag{4.26}$$

.

Definirajmo još analogno varijantu problema višeslojnog rijetkog kodiranja gdje je u signalu prisutan šum.

Definicija 4.5.2. *Za globalni signal* $\mathbf{Y}$*, skup konvolucijskih rječnika* $\{\mathbf{D}_i\}_{i=1}^K$ *te za vektore* $\lambda = [\lambda_1, ..., \lambda_K], \varepsilon = [\varepsilon_0, ..., \varepsilon_{K-1}] \in \mathbb{R}^K$ *definiramo problem dubokog rijetkog kodiranja sa šumom (*DCP_λ^ε*) s*

$$(DCP_\lambda^\varepsilon): \quad \textit{pronaći} \quad \{\mathbf{\Gamma}_i\}_{i=1}^K \quad \textit{t.d.} \quad \begin{aligned} \|\mathbf{Y} - \mathbf{D}_1\mathbf{\Gamma}_1\|_2 &\leq \varepsilon_0, & \|\mathbf{\Gamma}_1\|_{0,\infty}^s &\leq \lambda_1 \\ \|\mathbf{\Gamma}_1 - \mathbf{D}_2\mathbf{\Gamma}_2\|_2 &\leq \varepsilon_2, & \|\mathbf{\Gamma}_2\|_{0,\infty}^s &\leq \lambda_2 \\ &\vdots & &\vdots \\ \|\mathbf{\Gamma}_{K-1} - \mathbf{D}_K\mathbf{\Gamma}_K\|_2 &\leq \varepsilon_{K-1}, & \|\mathbf{\Gamma}_K\|_{0,\infty}^s &\leq \lambda_K \end{aligned} \tag{4.27}$$

.

Sada možemo definirati problem traženja parametara modela dubokog učenja (DLP_λ) (engl. *Deep Learning Problem*). Označimo $\text{DCP}^*_\lambda(\mathbf{X}, \{\mathbf{D}_i\}_{i=1}^K) = \mathbf{\Gamma}_K$ reprezentaciju dobivenu rješavanjem problema (DCP_λ). Sada proširujemo problem učenja rječnika na višeslojni slučaj.

Definicija 4.5.3. *Za skup označenih globalnih signala $\{\mathbf{X}_j, h(\mathbf{X}_j)\}_j$, funkciju gubitka ℓ te vektor $\lambda \in \mathbb{R}^K$ definiramo problem (DLP_λ) s*

$$(DLP_\lambda): \quad \min_{\{\mathbf{D}_i\}_{i=1}^K,\, U} \sum_j \ell\Big(h(\mathbf{X}_j), \mathbf{U}, \mathbf{\Gamma}_K\Big). \tag{4.28}$$

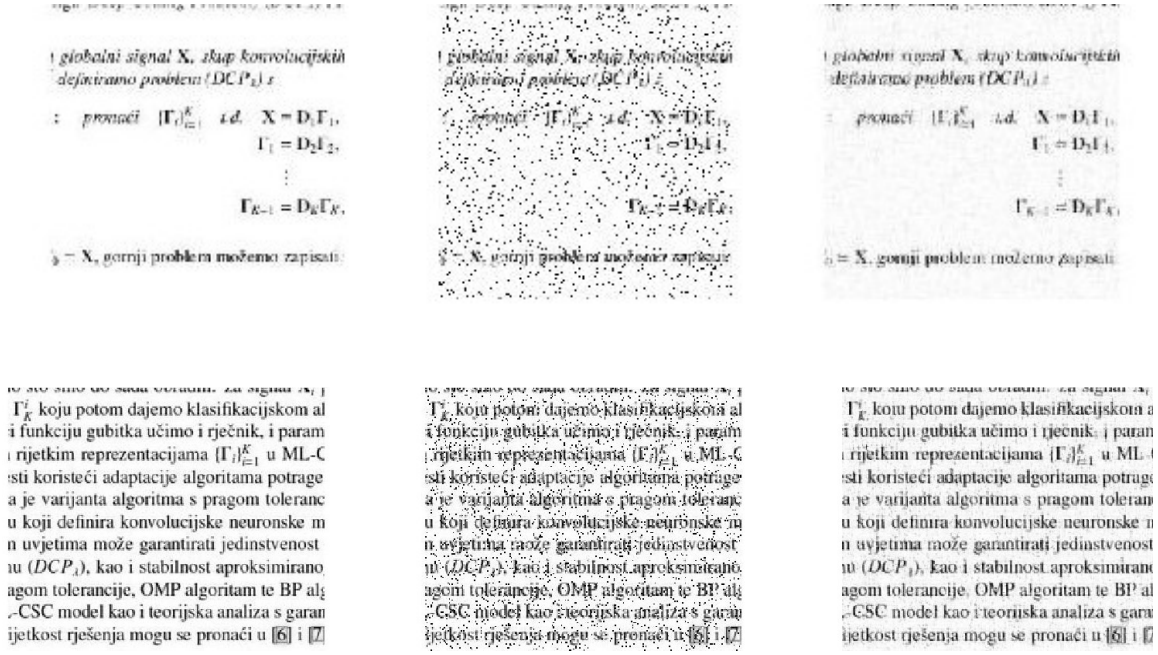
Analogno se može definirati i problem (DLP^ε_λ). Primijetimo kako je u ovim problemima ugrađeno što smo do sada obradili: za signal $\mathbf{X}_i$ pronalazimo konvolucijski rijetku reprezentaciju $\mathbf{\Gamma}^i_K$ koju potom dajemo klasifikacijskom algoritmu uz poznate oznake $h(\mathbf{X}_i)$. Minimizirajući funkciju gubitka učimo i rječnik, i parametre klasifiktora $\mathbf{U}$.

Potraga za rijetkim reprezentacijama $\{\mathbf{\Gamma}_i\}_{i=1}^K$ u ML-CSC modelu definiranom s (4.24) može se provesti koristeći adaptacije algoritama potrage koje smo obradili u 3. poglavlju. Pokazuje se da je varijanta algoritma s pragom tolerancije ekvivalentna prolasku unaprijed - algoritmu koji definira konvolucijske neuronske mreže. Nadalje, pokazuje se da se pod određenim uvjetima može garantirati jedinstvenost aproksimirane rijetke reprezentacije u problemu (DCP_λ), kao i stabilnost aproksimiranog rješenja u problemu (DCP^ε_λ) za algoritam s pragom tolerancije, OMP algoritam te BP algoritam. Opis varijanti algoritama potrage za ML-CSC model kao i teorijska analiza s garancijama jedinstvenosti i potrebnim ocjenama na rijetkost rješenja mogu se pronaći u [6] i [7].

4.6 Primjena rijetke reprezentacije

U ovom ćemo dijelu rada ilustrirati efikasnost rijetke reprezentacije u primjeni na nekoliko primjera koristeći kôd iz [11].

Primjer 4.6.1. *Jedan od problema koje rijetka reprezentacija rješava je uklanjanje šuma iz podataka (signala, slike, videa...). Ilustrirat ćemo primjenu rijetke reprezentacije na primjeru uklanjanja šuma iz fotografija. Fotografije koje promatramo su fotografije isječaka teksta ovog rada na koje smo dodali šum – tzv. salt and pepper noise. Učenje rječnika provodi se na 16 fotografija isječaka teksta, a potom se testira na neviđenim primjerima, tj. na primjerima koji nisu bili korišteni za učenje rječnika. Na slici 4.5 prikazujemo dva primjera originalne fotografije, fotografije sa šumom te rekonstruirane fotografije. Vidimo kako je tekst na fotografiji sa šumom teško prepoznatljiv, dok je na rekostruiranoj fotografiji vrlo lako čitljiv.*



Slika 4.5: Lijevo: originalne fotografije isječaka teksta. Sredina: fotografije isječaka teksta sa šumom. Desno: rekonstruirane fotografije.

Primjer 4.6.2. *Pri slanju podataka često dolazi do njihovog potpunog ili djelomičnog gubitka. Primjerice, umjesto originalnog signala pri primitku dobivamo verziju gdje su pojedine vrijednosti izgubljene ili nepoznate. U tom slučaju potrebno je rekonstruirati originalni signal što je točnije moguće, a pokazalo se da je moćan alat u rekonstrukciji originalnog signala upravo rijetka reprezentacija. Pogledajmo problem nestalih piksela u fotografiji, kojeg obrađujemo u ovom primjeru. Promatramo dvije crno-bijele fotografije kojima smo izbrisali vrijednosti na slučajno odabranim pikselima. Slika 4.6 prikazuje originalne fotografije, nakon čega slijede fotografije u kojima smo na slučajan način odabrali piksele čiju smo vrijednost izbrisali, a nakon toga rekonstruirane fotografije. Fotografije smo promatrali kroz njihove* 8×8 *dijelove te kroz 81 sloj mreže učili rječnik i tražili rijetku reprezentaciju. Uz poznat rječnik i rijetku reprezentaciju mogli smo vrlo precizno rekonstruirati polaznu fotografiju što pokazuje da je za spremanje fotografije dovoljno spremiti samo rječnik i rijetku reprezentaciju.*

Slika 4.6: Lijevo: originalne fotografije. Sredina: fotografije s izgubljenim pikselima. Desno: rekonstruirane fotografije.

Bibliografija

[1] S. Arora, M. Khodak, N. Saunshi i K. Vodrahalli, *A Compressed Sensing View of Unsupervised Text Embeddings, Bag-of-n-Grams, and LSTMs*, Proceedings of the 6th International Conference on Learning Representations (ICLR), 2018.

[2] I. S. Dhillon, Jr. R. W. Heath, T. Strohmer i J. A. Tropp, *Constructing Packings in Grassmannian Manifolds via Alternating Projection*, Experimental Mathematics **17** (2008), br. 1, 9–35, http://dx.doi.org/10.1080/10586458.2008.10129018.

[3] M. Elad, *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing*, Springer, 2010.

[4] I. F. Gorodnitsky i B. D. Rao, *Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm*, IEEE Transactions on Signal Processing **45** (1997), br. 3, 600–616.

[5] T. T. Nguyen, C. Soussen, J. Idier i E. Djermoune, *NP-hardness of l_0 minimization problems: revision and extension to the non-negative setting*, (2019), https://hal.archives-ouvertes.fr/hal-02112180.

[6] V. Papyan, Y. Romano i M. Elad, *Convolutional Neural Networks Analyzed via Convolutional Sparse Coding*, 2016.

[7] V. Papyan, Y. Romano, J. Sulam i M. Elad, *Theoretical Foundations of Deep Learning via Sparse Representations: A Multilayer Sparse Model and Its Connection to Convolutional Neural Networks*, IEEE Signal Processing Magazine **35** (2018), br. 4, 72–89.

[8] V. Papyan, J. Sulam i M. Elad, *Working Locally Thinking Globally: Theoretical Guarantees for Convolutional Sparse Coding*, CoRR **abs/1707.06066** (2017), http://arxiv.org/abs/1707.06066.

[9] R. Rubinstein, A. M. Bruckstein i M. Elad, *Dictionaries for Sparse Representation Modeling*, Proceedings of the IEEE **98** (2010), br. 6, 1045–1057.

[10] A. M. Tillmann i M. E. Pfetsch, *The Computational Complexity of the Restricted Isometry Property, the Nullspace Property, and Related Concepts in Compressed Sensing*, (2012), https://arxiv.org/pdf/1205.2081.pdf.

[11] E. Zisselman, J. Sulam i M. Elad, *A Local Block Coordinate Descent Algorithm for the CSC Model*, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

Sažetak

U ovom radu bavili smo se rijetkim reprezentacijama i njihovom primjenom u dubokom strojnom učenju. Definirali smo rijetku reprezentaciju vektora uz rječnik kao linearnu kombinaciju samo nekoliko atoma iz rječnika. Cilj nam je pronaći rijetku reprezentaciju vektora koja ga dovoljno dobro opisuje uz korištenje što je manje moguće atoma iz rječnika, tj. uz što manje ne-nul komponenti. Formalno smo opisali optimizacijski problem pronalaska rijetke reprezentacije za dani vektor uz pogodnu mjeru rijetkosti i linearne uvjete. Za taj se problem pokazalo da je nekonveksan, stoga nismo mogli koristiti standardne alate konveksne optimizacije niti garantirati jedinstvenost rješenja.
Iz tog razloga uveli smo funkcije *sparka*, međusobne koherencije i kumulativne koherencije koje opisuju matricu rječnika te pomoću njih dokazali jedinstvenost rijetke reprezentacije pod određenim uvjetima na rijetkost rješenja.
Nakon teorijske analize, uz garanciju jedinstvenosti rješenja optimizacijskog problema pronalaska rijetke reprezentacije, bavili smo se opisom algoritama potrage koji aproksimiraju rijetku reprezentaciju vektora uz matricu rječnika. Promatrali smo pohlepne algoritme kojima je glavna ideja lokalno optimalno djelovanje u potrazi za globalno optimalnim rješenjem kao što su OMP (engl. *Orthogonal-Matching Pursuit*) i algoritam s pragom tolerancije. Bavili smo se i metodama relaksacije, kao što je BP algoritam (engl. *Basis Pursuit*), koje problem pronalaska rijetke reprezentacije svode na konveksan problem, a potom koriste metode konveksne optimizacije za aproksimaciju rješenja. Nakon opisa algoritama potrage dali smo njihovu numeričku usporedbu na primjerima.
Posljednji dio rada usmjeren je na rijetku reprezentaciju u dubokom strojnom učenju, točnije u konvolucijskim neuronskim mrežama. Dali smo kratak uvod u konvolucijske neuronske mreže i učenje rječnika, a zatim formalno opisali primjenu rijetke reprezentacije u konvolucijskim neuronskim mrežama za razne probleme nadziranog i nenadziranog učenja. Na kraju smo ilustrirali moć rijetke reprezentacije na nekoliko primjera iz primjene.

Summary

This thesis deals with sparse representations and their applications in deep learning. Firstly, we defined the sparse representation of a vector as a linear combination of only a few atoms from a dictionary. Our goal is to find the sparse representation of a vector which has as many zero components as possible and still describes the vector with sufficient accuracy. This task, when described formally, is a non-convex optimization problem. Due to its non-convexity, we could not use standard convex optimization tools and therefore could not guarantee the uniqueness of such solution, i.e., sparse representation.
For that reason, we introduced functions which describe the dictionary, such as spark, mutual coherence and cumulative coherence. We managed to state and prove the theorems which guarantee the uniqueness of sparse representation under certain conditions by using these functions.
With the uniqueness of the solution guaranteed, we could move on to pursuit algorithms, the practical methods for finding the sparse representation of a vector given a dictionary matrix. We described greedy methods, such as the Orthogonal-Matching Pursuit (OMP) algorithm and the thresholding algorithm, which perform locally-optimal updates in search of the optimal solution. We also described relaxation methods, such as Basis Pursuit (BP), which replace the non-convex sparsity measure with a convex approximation and then find the sparse representation using convex optimization algorithms. After that, we compared these pursuit algorithms on a couple of examples.
The last part of this thesis deals with sparse representation in deep learning, precisely in convolutional neural networks (CNNs). We briefly described CNNs and the dictionary learning problem, followed by a description of the Convolutional Sparse Coding model and the Multi-Layer Convolutional Sparse Coding Model, coined CSC and ML-CSC respectfully. Lastly, we gave a couple of examples to illustrate the power and efficiency of sparse representation models.

Životopis

Rođena sam 24. lipnja 1995. godine u Karlovcu. Završila sam Osnovnu školu "Ivan Goran Kovačić" u Dugoj Resi. Nakon toga sam upisala Gimnaziju Karlovac, opći smjer, koju sam završila 2014. godine. Iste sam godine upisala preddiplomski studij Matematike na Prirodoslovno-matematičkom fakultetu u Zagrebu kojeg sam završila 2017. godine. Potom sam upisala diplomski studij Primijenjena matematika na istome fakultetu kojeg završavam ovim radom.