UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

TACILDO DE SOUZA ARAÚJO

Métodos para completamento de matrizes de posto conhecido

Campinas

2023

Tacildo de Souza Araújo

Métodos para completamento de matrizes de posto conhecido

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Matemática Aplicada.

Orientador: Cristiano Torezzan

Coorientador: Douglas Soares Gonçalves

Este trabalho corresponde à versão final da Tese defendida pelo aluno Tacildo de Souza Araújo e orientada pelo Prof. Dr. Cristiano Torezzan.

Campinas

2023

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Araújo, Tacildo de Souza, 1988-
Ar15m Métodos para completamento de matrizes de posto conhecido / Tacildo de Souza Araújo. – Campinas, SP : [s.n.], 2023.

Orientador: Cristiano Torezzan.
Coorientador: Douglas Soares Gonçalves.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Matrizes (Matemática). 2. Método de gradiente proximal. 3. Método de gradiente projetado. 4. Sistemas de recomendação (Filtragem da informação). I. Torezzan, Cristiano, 1976-. II. Gonçalves, Douglas Soares, 1982-. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Informações Complementares

Título em outro idioma: Methods for known-rank matrix completion
Palavras-chave em inglês:
Matrices
Proximal gradient method
Projected gradient method
Recommender systems (Information filtering)
Área de concentração: Matemática Aplicada
Titulação: Doutor em Matemática Aplicada
Banca examinadora:
Cristiano Torezzan [Orientador]
Carlile Campos Lavor
Francisco de Assis Magalhães Gomes Neto
Luís Felipe Cesar da Rocha Bueno
Andres David Baez Sanchez
Data de defesa: 17-01-2023
Programa de Pós-Graduação: Matemática Aplicada

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-7743-6748
- Currículo Lattes do autor: http://lattes.cnpq.br/2603366101703772

Tese de Doutorado defendida em 17 de janeiro de 2023 e aprovada pela banca examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). CRISTIANO TOREZZAN

Prof(a). Dr(a). CARLILE CAMPOS LAVOR

Prof(a). Dr(a). FRANCISCO DE ASSIS MAGALHÃES GOMES NETO

Prof(a). Dr(a). LUÍS FELIPE CESAR DA ROCHA BUENO

Prof(a). Dr(a). ANDRES DAVID BAEZ SANCHEZ

A Ata da Defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-Graduação do Instituto de Matemática, Estatística e Computação Científica.

Dedico este trabalho à minha mãe e à minha esposa.

Agradecimentos

Agradeço a Deus por me dar forças para concluir este trabalho.

À minha esposa Naira, pelo apoio, companhia e compreensão ao longo dessa jornada.

À minha mãe pelas orações e pelo seu apoio incondicional.

Ao meu amigo e orientador professor Cristiano Torezzan, por sua orientação, paciência, amizade e pelas conversas ao longo dessa jornada. Obrigado por sempre ter me motivado nos momentos difíceis e sobretudo por me ajudar a tornar este sonho uma realidade.

Ao meu coorientador professor Douglas Soares Gonçalves, pelas valiosas contribuições para o desenvolvimento deste trabalho. Aprendi muito com você em nossas reuniões semanais e durante o processo de escrita dessa tese.

Aos amigos que o IMECC me deu: Gesivaldo, Luiz, Samuel, Carlos, Michele, Walter, Ivelton, Maycon. Um agradecimento especial ao meu amigo Fabiano, por ter me apresentado ao meu orientador e pelas longas conversas que tivemos durante o tempo em que estivemos em Campinas.

À Unicamp e ao IMECC pela excelente qualidade de ensino e pela oportunidade de cursar o doutorado.

A todos os professores do Doutorado em Matemática Aplicada.

A todos os funcionários do Departamento de Matemática Aplicada que sempre me receberam com muita cortesia e atenção.

Ao Instituto Federal do Amazonas (IFAM) pode ter me concedido o afastamento para cursar o doutorado.

"Talvez não tenha conseguido fazer o melhor,
mas lutei para que o melhor fosse feito.
Não sou o que deveria ser,
mas Graças a Deus, não sou o que era antes".
Marthin Luther King.

Resumo

O Problema de Completamento de Matrizes (PCM) consiste em estimar as entradas em falta de uma matriz, a partir de um subconjunto de entradas conhecidas. Esse problema pode ser formulado como um problema de otimização com a condição de que a matriz a ser completada tenha posto reduzido. Além disso, em algumas aplicações, o posto da matriz alvo é conhecido a priori e esta informação pode ser útil para o processo de completamento da matriz. Esta tese concentra-se no estudo de métodos de otimização para o completamento de matrizes de posto conhecido. Inicialmente, propomos um método que utiliza a informação do posto alvo e uma decomposição SVD truncada em cada iteração. Estabelecemos uma condição sob a qual a sequência gerada pelo método é quasi-Fejér convergente para o conjunto solução do problema. Em seguida, incluímos um mecanismo de aceleração semelhante à aceleração de Nesterov para obter uma heurística que, embora não tenha garantia de convergência, pode ser usada para a obtenção de um bom ponto inicial, bem como um valor para o parâmetro de regularização para um método de gradiente proximal acelerado que visa resolver um problema de quadrados mínimos regularizado pela norma nuclear. Uma segunda contribuição da tese consiste na aplicação de um método de gradiente projetado para resolver um problema de otimização com restrição de posto que modela o PCM. O principal desafio dessa abordagem está na escassez de resultados de convergência para o gradiente projetado (GP) em conjuntos não-convexos, que sejam baseados em hipóteses razoáveis na prática. Utilizando a propriedade de isometria restrita, bem como, assumindo conhecido o posto da matriz alvo, mostramos que a sequência gerada pelo algoritmo de GP proposto converge para a solução do problema. Todos os métodos propostos foram testados, tanto com dados sintéticos, quanto em conjuntos de dados reais usualmente utilizados na área. Os resultados computacionais obtidos mostram que os métodos propostos têm desempenho similar aos principais métodos da literatura, com a vantagem de controlar o posto da matriz resultante.

Palavras-chave: Completamento de Matrizes. Matrizes de posto reduzido. Matrizes de posto conhecido. Gradiente Proximal. Gradiente Projetado.

Abstract

The Matrix Completion Problem (MCP) consists in estimating the missing entries of a matrix from a subset of known entries. This problem can be formulated as an optimization problem with the condition that the matrix to be completed has a low rank. Furthermore, in some applications, the rank of the target matrix is known in advance; this information can be helpful during the matrix completion process. This thesis focuses on the study of optimization methods for the completion of matrices that have known rank. Initially, we propose a method that uses the target rank information and a truncated SVD decomposition at each iteration. We establish a condition under which the generated sequence is quasi-Fejér convergent to the solution set of the problem. We then include an acceleration mechanism similar to Nesterov's acceleration to obtain a heuristic that, although the convergence is not guaranteed, can be used to obtain a good starting point, as well as a value for the regularization parameter for an accelerated proximal gradient method that aims to solve a nuclear norm regularized least squares problem. A second contribution of the thesis consists of the application of a projected gradient method to solve a rank-constrained optimization problem that models the PCM. The main challenge of this approach lies in the scarcity of convergence results for the projected gradient (GP) on non-convex sets that are based on reasonable assumptions in practice. Using the restricted isometry property, as well as, assuming that the rank of the target matrix is known, we show that the sequence generated by the proposed GP algorithm converges to the solution of the problem. All the proposed methods have been tested, both with synthetic data and on real data sets commonly used in the field. The computational results obtained show that the proposed methods perform similarly to the main methods in the literature, with the advantage of controlling the rank of the resulting matrix.

Keywords: Matrix Completion. Low-rank matrices. Known-rank matrices. Proximal Gradient. Projected Gradient.

Lista de ilustrações

Lista de tabelas

Lista de símbolos

$\mathbb{R}^{m\times n}$	Conjunto das matrizes de ordem $m \times n$ com entradas reais
I	Matriz identidade
$A^\top$	Transposta da matriz A
$\mathrm{Tr}(A)$	Traço da matriz A
$\|A\|_F$	Norma de Frobenius da matriz A
$\|A\|_2$	Norma espectral da matriz A
$\|A\|_*$	Norma nuclear da matriz A
$\|v\|_p$	Norma-p do vetor v
S_λ	Operador Soft-Thresholding
∇f	Gradiente da função f
Ω	Conjunto dos índices dos valores observados
P_Ω	Operador projeção
$P_\Omega^\perp$	Operador projeção complementar
f^*	Conjugado da função f
f^{**}	Biconjugado da função f
epi f	Epígrafo da função f
$\partial f(x)$	Subdiferencial de f em x
prox_f	Operador proximal da função
$\lceil x \rceil$	Menor inteiro maior que x.
$\lambda_{\max}(A)$	Maior autovalor da matriz A.

Sumário

Introdução

O problema de recuperar uma matriz a partir de uma fração de suas entradas é conhecido como Problema de Completamento de Matrizes (PCM) e tem despertado o interesse de muitos pesquisadores nas últimas décadas. Em geral, o PCM é um problema mal-posto [16], uma vez que, dada uma amostra aleatória de entradas conhecidas de uma matriz, haverá infinitas possibilidades de completamento das entradas em falta. Nesse sentido, Candés e Recht [15] dão uma contribuição importante para a área ao mostrarem que é possível recuperar matrizes de posto reduzido a partir de um conjunto relativamente pequeno de entradas conhecidas. Assim, o PCM geralmente assume a hipótese de que a matriz alvo tem posto reduzido.

Há diversas aplicações que podem ser analisadas sob a ótica de completamento de matrizes. Na área de filtragem colaborativa [13, 36], podemos citar os sistemas de recomendação [33], que são ferramentas utilizadas por empresas para recomendar novos produtos a seus clientes utilizando o histórico de informações dos usuários armazenado no sistema. Neste caso, acredita-se que poucos fatores colaboram para a preferência dos usuários, resultando em uma matriz de posto baixo. Outras aplicações importantes incluem o problema de predição de link [59, 9], machine learning [21] e alguns problemas em processamento de imagem [24].

Esta tese tem enfoque no estudo de métodos de otimização para completamento de matrizes de posto conhecido a priori. A motivação advém de aplicações reais cuja matriz associada possui posto conhecido, tais como o problema de localização em rede de sensores [45]. Neste problema, o objetivo é determinar as posições de sensores na rede a partir de uma amostra de distâncias entre alguns pares de sensores [43]. Um passo intermediário para sua resolução é completar as entradas (distâncias) faltantes em uma Matriz de Distâncias. Tais matrizes possuem posto reduzido e conhecido [20].

As contribuições desse trabalho estão concentradas nos dois últimos capítulos e são sintetizadas a seguir:

- Propomos um algoritmo de duas fases, a primeira das quais é uma heurística baseada em uma iteração de ponto fixo que usa a informação do posto da matriz alvo para estimar um parâmetro de regularização e um bom ponto inicial para um método de gradiente proximal acelerado [46] (segunda fase), que visa resolver um problema de quadrados mínimos regularizado pela norma nuclear (mais detalhes na Seção 3.2). Parte dos resultados deste estudo foi apresentada em [3, 4] e publicada no artigo [2].

- Estudamos um método de gradiente projetado para determinar uma matriz de posto

no máximo r que melhor se ajuste às entradas conhecidas de uma matriz alvo de posto r. Mostramos que a sequência gerada converge para uma solução do problema sob uma condição relaxada do parâmetro de isometria restrita. Além disso, incluímos um mecanismo de aceleração similar à aceleração de Nesterov, que fornece bons resultados numéricos quando comparado com outros métodos para completamento estudados neste trabalho. Um resultado desse estudo foi apresentado em [1].

Essa tese está organizada da seguinte forma: no Capítulo 1, apresentamos uma revisão de alguns resultados importantes que são comumente explorados em estudos de métodos de completamento de matrizes. O Capítulo 2 é dedicado ao estudo do Problema de Completamento de Matrizes de Posto Reduzido e à revisão de alguns métodos para completamento que serão usados como benchmark nos próximos capítulos. No Capítulo 3, propomos um algoritmo de duas fases para recuperação de matrizes de posto conhecido a priori, o qual é inspirado em uma heurística para completamento de matrizes de posto conhecido, e comparamos os resultados numéricos com os métodos estudados no capítulo anterior. No Capítulo 4, estudamos um método de gradiente projetado para completamento de matrizes. Mais especificamente, aplicamos o GP para resolver um problema de otimização com restrição de posto e utilizando a função específica do problema mostramos a convergência do método sob uma condição de isometria restrita. Além disso, incluímos um mecanismo de aceleração de Nesterov e obtivemos uma heurística com bons resultados numéricos. No Capítulo 5, apresentamos as conclusões, uma síntese das contribuições dessa tese e elencamos alguns tópicos que serão considerados em pesquisas futuras.

1 Fundamentos teóricos

Neste capítulo revisitamos algumas definições e resultados que consideramos essenciais para o desenvolvimento deste trabalho. Começamos apresentando a classe das funções convexas, as quais possuem importantes propriedades que são comumente exploradas em formulações de problemas de completamento de matrizes. Em seguida, apresentamos alguns resultados de álgebra matricial, dando ênfase a conceitos importantes no contexto deste trabalho, como a Decomposição em Valores Singulares (SVD[1]), a norma nuclear, entre outros. Finalmente, apresentamos o método do gradiente proximal, o qual está diretamente relacionado com uma de nossas contribuições. Algumas referências para este capítulo são [6, 14, 22, 38, 46, 49, 50, 55, 56, 57].

1.1 Funções convexas

Convexidade é um conceito fundamental em otimização e os conjuntos convexos constituem um domínio natural para as funções convexas. Geometricamente, um conjunto S é dito convexo se todo ponto do segmento de reta que une dois pontos quaisquer de S pertence a S.

Definição 1.1 (Conjunto convexo)**.** *O conjunto $S \subset \mathbb{R}^n$ é chamado um conjunto convexo se para quaisquer $x, y \in S$ e para todo $\lambda \in [0,1]$, tem-se $\lambda x + (1-\lambda)y \in S$.*

Definição 1.2 (Função convexa)**.** *Seja $S \subset \mathbb{R}^n$ um conjunto convexo. Dizemos que a função $f : \mathbb{R}^n \to \mathbb{R}$ é convexa em S quando*

$$f(\lambda x + (1-\lambda)y) \leqslant \lambda f(x) + (1-\lambda)f(y), \tag{1.1}$$

para todos $x, y \in S$ e $\lambda \in [0,1]$.

Quando a desigualdade (1.1) é satisfeita estritamente para quaisquer $x, y \in S$ e $\lambda \in (0,1)$, diz-se que f é uma função *estritamente convexa*. Além disso, dizemos que $f : \mathbb{R}^n \to \mathbb{R}$ é *fortemente convexa* no conjunto convexo $S \subset \mathbb{R}^n$, com módulo c, quando existe uma constante $c > 0$ tal que

$$f(\lambda x + (1-\lambda)y) \leqslant \lambda f(x) + (1-\lambda)f(y) - \frac{c}{2}\lambda(1-\lambda)\|x-y\|_2^2, \tag{1.2}$$

quaisquer que sejam $x, y \in S$ e $\lambda \in [0,1]$. Uma consequência imediata de (1.2) é que toda função fortemente convexa é estritamente convexa. Entretanto, a recíproca não é

[1] Do inglês Singular Value Decomposition.

verdadeira. Por exemplo, a função $f(x) = e^x$, definida em $\mathbb{R}$, é estritamente convexa, mas não é fortemente convexa.

Embora a definição de função convexa pareça simples, nem sempre é fácil mostrar que uma função f é convexa usando a desigualdade (1.1). No entanto, quando f é diferenciável, sua convexidade pode ser caracterizada pelo teorema a seguir.

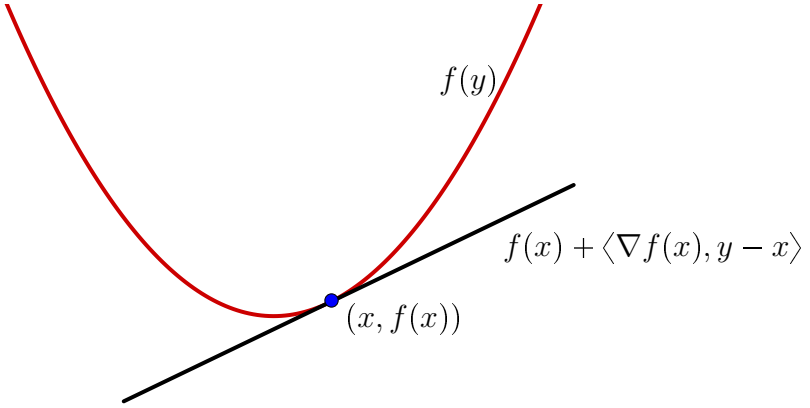
Teorema 1.1. *[38, Teorema 3.1.7] Sejam $f : \mathbb{R}^n \to \mathbb{R}$ uma função diferenciável e $S \subset \mathbb{R}^n$ um conjunto convexo. A função f é convexa em S se, e somente se,*

$$f(y) \geqslant f(x) + \langle \nabla f(x), y - x \rangle, \tag{1.3}$$

para todos $x, y \in S$.

A Figura 1 fornece a interpretação geométrica para a desigualdade (1.3). Observe que a expressão do lado direito de (1.3) é uma aproximação de Taylor de primeira ordem de f em torno de x. A desigualdade nos diz que, para uma função convexa, a aproximação de Taylor de primeira ordem é de fato um limitante inferior global de f. Outra propriedade imediata de (1.3), é que, se $\langle \nabla f(x), y - x \rangle \geqslant 0, \forall y \in \mathcal{C}$, então $f(y) \geqslant f(x)$, $\forall y \in \mathcal{C}$, isto é, x é um minimizador global de f em S.

Figura 1 – Aproximação linear de f



Com efeito, o resultado a seguir mostra que todo minimizador local de f convexa é minimizador global. Aqui lembramos a definição de minimizador local. Dados uma função $f : \mathbb{R}^n \to \mathbb{R}$ e $a \in S \subset \mathbb{R}^n$, dizemos que a é um *minimizador local* de f em S quando existe $\delta > 0$, tal que $f(a) \leqslant f(x)$, para todo $x \in B(a, \delta) \cap S$, em que $B(a, \delta) = \{x \in \mathbb{R}^n : \|x - a\|_2 < \delta\}$. Quando $f(a) \leqslant f(x)$ para todo $x \in S$, dizemos que a é um *minimizador global* de f em S.

Teorema 1.2. *[49, Teorema 3.12] Seja $S \subset \mathbb{R}^n$ e $f : S \to \mathbb{R}$ uma função convexa. Se $a \in S$ é um minimizador local de f, então a é um minimizador global de f.*

Há muitos problemas nos quais a função que se deseja minimizar é não-convexa. Nestes casos, pode ser razoável tentar encontrar uma função convexa que se aproxima de tal função e tentar minimizá-la em seu lugar. Isto nos leva ao conceito de envelope convexo de uma função. Para definirmos tal conceito, precisamos de algumas definições e resultados preliminares.

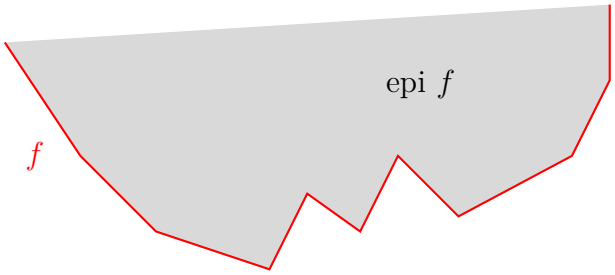
Definição 1.3 (Envelope convexo de um conjunto)**.** *Seja S um subconjunto do $\mathbb{R}^n$. O conjunto formado pela interseção de todos os conjuntos convexos que contém S é chamado de envelope convexo de S e será denotado por* conv S*.*

Definição 1.4. *Seja $S \subset \mathbb{R}^n$. O epígrafo de uma função $f : S \to [-\infty, +\infty]$, denotado por* epi f*, é definido como*

$$\text{epi } f = \{(x,t) : x \in S,\, t \in \mathbb{R},\, f(x) \leqslant t\}\,.$$

O epígrafo de uma função f é composto pelo gráfico de f e os pontos que estão acima do gráfico. Uma função f é convexa em S quando seu epígrafo é um subconjunto convexo de $\mathbb{R}^{n+1}$. Na Figura 2 temos uma ilustração do epígrafo de uma função f.

Figura 2 – O epígrafo da função f representado pela cor cinza. A fronteira inferior, em vermelho, é o gráfico de f.



O *domínio efetivo* de uma função $f : \mathbb{R}^n \supset S \to [-\infty, +\infty]$, que toma valores na reta real estendida $[-\infty, +\infty] = \mathbb{R} \cup \{\pm\infty\}$, é a projeção do epígrafo de f no espaço $\mathbb{R}^n$, isto é,

$$\text{dom } f = \{x : \text{existe}\, t,\, (x,t) \in \text{epi } f\} = \{x \in \mathbb{R}^n : f(x) < +\infty\}\,.$$

Teorema 1.3. *[28, Proposição 2.5.1] Seja a função $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ tal que $f \not\equiv +\infty$, e existe uma função afim minorante de f em $\mathbb{R}^n$, isto é, para algum $(s,b) \in \mathbb{R}^n \times \mathbb{R}$, tem-se $f(x) \geqslant \langle s, x\rangle + b$ para todo $x \in \mathbb{R}^n$. Então, as funções*

$$g_1(x) = \inf\{t : (x,t) \in \text{conv epi } f\}\,, \tag{1.4}$$

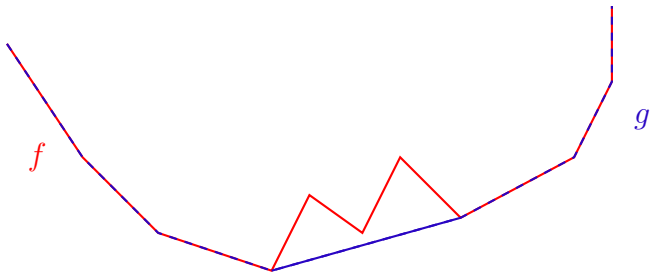
$$g_2(x) = \sup\{h(x), h \text{ é convexa}, h \leqslant f\}\,, \tag{1.5}$$

são convexas e coincidem no $\mathbb{R}^n$.

Definição 1.5 (Envelope convexo de uma função)**.** *O envelope convexo de uma função $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ que satisfaz as hipóteses do Teorema 1.3 é qualquer uma das funções $g_1(x)$ e $g_2(x)$ definidas nas equações (1.4)-(1.5).*

Intuitivamente, o envelope convexo de uma função f é a maior função g tal que $g(x) \leqslant f(x)$, para todo $x \in \mathbb{R}^n$. A Figura 3 apresenta uma ilustração geométrica para a envelope convexo da função f da figura anterior.

Figura 3 – Interpretação geométrica do envelope convexo de uma função. A função g (azul) é o envelope convexo de f (vermelho).



Dado um conjunto $S \subset \mathbb{R}^n$, diz-se que um ponto $a \in \mathbb{R}^n$ é aderente a S quando a for limite de uma sequência de pontos $x^n \in S$. O *fecho* do conjunto S, denotado por $\bar{S}$, é o conjunto de todos os pontos aderentes a S. Obviamente, $S \subset \bar{S}$. No caso de ser $S = \bar{S}$, diz-se que S é um conjunto *fechado.*

Definição 1.6 (Função fechada)**.** *Uma função $f : \mathbb{R}^n \to [-\infty, +\infty]$ é fechada se, e somente se, seu epígrafo é um conjunto fechado.*

Definição 1.7 (Função própria)**.** *Uma função $f : \mathbb{R}^n \to [-\infty, +\infty]$ é dita própria se ela não atinge o valor $-\infty$ e existe pelo menos um $x \in \mathbb{R}^n$ tal que $f(x) < +\infty$, isto é,* dom f *é não-vazio.*

Definição 1.8 (Conjugado convexo)**.** *Seja a função $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ tal que $f \not\equiv +\infty$, e existe uma função afim minorante de f em $\mathbb{R}^n$. O conjugado convexo f^*, da função f, é definido por*

$$f^*(y) = \sup_{x \in \mathbb{R}^n} \left(\langle y, x \rangle - f(x) \right), \tag{1.6}$$

em que $y \in \mathbb{R}^n$ é um ponto arbitrário.

Definição 1.9 (Biconjugado)**.** *O biconjugado, f^{**}, de uma função f satisfazendo as hipóteses da Definição 1.8 é definido por: para todo $x \in \mathbb{R}^n$,*

$$f^{**}(x) = (f^*)^*(x) = \sup_{y \in \mathbb{R}^n} \left(\langle y, x \rangle - f^*(y) \right). \tag{1.7}$$

Exemplo 1.1. *Considere a função $f : \mathbb{R} \to \mathbb{R}$, definida por $f(x) = |x^2 - 2x|$.*

O conjugado de f é dado por

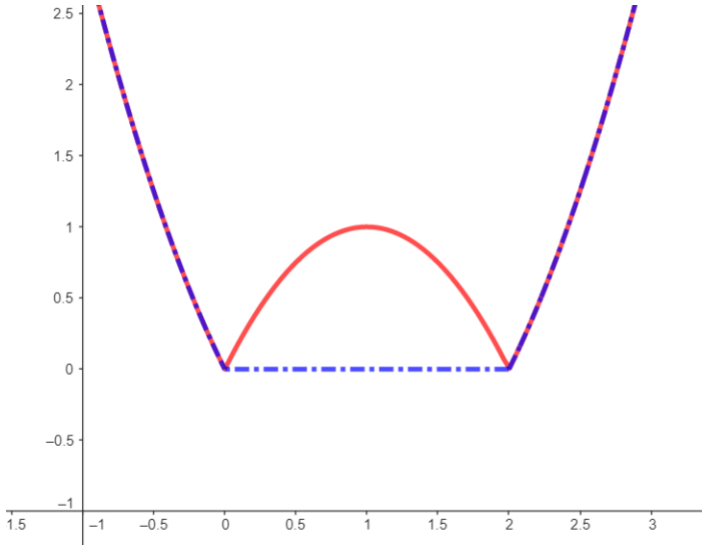
$$f^*(y) = \begin{cases} \dfrac{(y+2)^2}{4}, & \text{se } y \leqslant -2 \text{ ou } y \geqslant 2 \\ 0, & \text{se } -2 < y \leqslant 0 \\ 2y, & \text{se } 0 < y < 2. \end{cases}$$

e o biconjugado de f é dado por

$$f^{**}(x) = \begin{cases} x^2 - 2x, & \text{se } x \leqslant 0 \text{ ou } x \geqslant 2 \\ 0, & \text{se } 0 < x < 2. \end{cases}$$

A figura, a seguir, mostra os gráficos de f e f^{**}.

Figura 4 – Os gráficos da função f (vermelho) e seu biconjugado f^{**} (azul tracejado).



Observação 1.1. *Se uma função f satisfaz as hipóteses do Teorema 1.3, então usando as equações (1.6) e (1.7) mostra-se que $f^{**}(x) \leqslant f(x)$ para todo $x \in \mathbb{R}^n$. Além disso, f^{**} é a maior função convexa tal que $f^{**}(x) \leqslant f(x)$, de modo que se uma função convexa h é tal que $h \leqslant f$, para todo $x \in \mathbb{R}^n$, então $h \leqslant f^{**}$ (para mais detalhes consulte [28, Capítulo E]).*

Definição 1.10 (Subgradiente)**.** *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função convexa. Dizemos que $g \in \mathbb{R}^n$ é um subgradiente de f no ponto $x_0 \in \mathbb{R}^n$, se*

$$f(x) \geqslant f(x_0) + \langle g, x - x_0 \rangle, \tag{1.8}$$

para todo $x \in \mathbb{R}^n$. O conjunto de todos os subgradientes de f em x_0 é chamado de subdiferencial *de f em x_0 e é denotado por $\partial f(x_0)$.*

O subdiferencial pode ser utilizado para estabelecer uma condição necessária e suficiente para um minimizador de uma função convexa f, da seguinte maneira:

Lema 1.1. *[42, Teorema 3.1.20] Um ponto $x^\star$ é um minimizador de uma função convexa f se, e somente se, f é subdiferenciável em $x^\star$ e $0 \in \partial f(x^\star)$.*

A próxima seção trata dos conceitos e resultados de álgebra matricial que serão usados ao longo dos próximos capítulos.

1.2 Noções de álgebra matricial

Definição 1.11 (Combinação linear)**.** *Sejam o conjunto de vetores $V = \{v_1, v_2, \ldots, v_n\} \subset \mathbb{R}^m$ e o conjunto de escalares $\{\alpha_1, \alpha_2, \ldots, \alpha_n\} \subset \mathbb{R}$. A expressão dada por*

$$\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n$$

chamamos de combinação linear dos elementos de V.

Definição 1.12 (Dependência e independência linear)**.** *Dizemos que o conjunto $V = \{v_1, v_2, \ldots, v_n\} \subset \mathbb{R}^m$ é linearmente independente (LI), ou que os vetores $v_1, v_2, \ldots, v_n$ são LI, se a equação*

$$\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_n v_n = 0, \tag{1.9}$$

implica $\alpha_1 = \alpha_2 = \cdots = \alpha_n = 0$. No caso em que exista algum $\alpha_i \neq 0$, dizemos que o conjunto V é linearmente dependente (LD).

Definição 1.13 (Posto de uma matriz)**.** *Seja a matriz $A \in \mathbb{R}^{m\times n}$. O posto de A, denotado por* posto(A)*, é o número máximo de linhas ou colunas linearmente independentes.*

Dois vetores $u, v \in \mathbb{R}^n$ são ditos *ortogonais* quando $\langle u, v\rangle = 0$. Um conjunto de vetores $\{v_1, v_2, \ldots, v_m\} \subset \mathbb{R}^n$ é dito *ortornormal* quando seus vetores são ortogonais dois a dois e $\|v_i\| = 1$ para cada i; isto é, $\langle v_i, v_j\rangle = 0$, quando $i \neq j$ e $\langle v_i, v_j\rangle = 1$, quando $i = j$.

Uma matriz $Q \in \mathbb{R}^{m\times m}$ é dita *ortogonal* se $Q^\top Q = QQ^\top = I$, em que I é a matriz identidade de ordem m. Em uma matriz ortogonal as linhas (colunas) são vetores ortonormais. Além disso, se Q é uma matriz ortogonal, então $Q^{-1} = Q^\top$.

Teorema 1.4. *(SVD [23, Teorema 2.4.1]) Seja $A \in \mathbb{R}^{m\times n}$ uma matriz de posto r. Então, A pode ser escrita na forma*

$$A = U\Sigma V^\top, \tag{1.10}$$

em que $U \in \mathbb{R}^{m\times m}$ e $V \in \mathbb{R}^{n\times n}$ são matrizes ortogonais e $\Sigma \in \mathbb{R}^{m\times n}$ é dada por

$$\Sigma = \begin{bmatrix} \overline{\Sigma}_{r\times r} & 0_{r\times(n-r)} \\ 0_{(m-r)\times r} & 0_{(m-r)\times(n-r)} \end{bmatrix}, \tag{1.11}$$

em que $\overline{\Sigma}_{r\times r} = diag(\sigma_1, \sigma_2, \ldots, \sigma_r)$, com $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0$.

Na decomposição (1.10), as colunas de U e V são vetores ortonormais e são chamados de vetores singulares à esquerda de A e vetores singulares à direita de A, respectivamente. Os números reais σ_i são as raízes quadradas positivas dos autovalores de $A^\top A$ ou $AA^\top$ e são chamados de valores singulares da matriz A.

O próximo resultado segue imediatamente do Teorema 1.4, escrevendo-se a SVD da matriz A em blocos adequados e eliminando-se os blocos desnecessários.

Teorema 1.5. *(SVD compacta [57, Teorema 4.1.10]) Seja $A \in \mathbb{R}^{m\times n}$ uma matriz de posto r. Então, existem matrizes $U \in \mathbb{R}^{m\times r}$, $\Sigma \in \mathbb{R}^{r\times r}$ e $V \in \mathbb{R}^{n\times r}$, em que U e V são matrizes com colunas ortonormais e Σ é uma matriz diagonal com as entradas da diagonal principal dadas por $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0$, tais que*

$$A = U\Sigma V^\top. \tag{1.12}$$

Uma forma interessante de apresentar a SVD de uma matriz A e que segue escrevendo-se a equação (1.12) na forma

$$A = \begin{bmatrix} u_1 & u_2 & \dots & u_r \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_r \end{bmatrix} \begin{bmatrix} v_1^\top \\ v_2^\top \\ \dots \\ v_r^\top \end{bmatrix},$$

é dada por

$$A = \sum_{j=1}^{r} \sigma_j u_j v_j^\top. \tag{1.13}$$

Uma outra consequência imediata da SVD é que o posto de um matriz A é dado pela quantidade de valores singulares não-nulos. Este fato está formalizado no teorema a seguir, cujo resultado advém do fato que U e V são matrizes de posto completo.

Teorema 1.6. *[56, Teorema 5.1] Seja $A \in \mathbb{R}^{m\times n}$. O posto de A é dado pela quantidade de valores singulares não-nulos.*

Definição 1.14 (Norma matricial)**.** *Uma norma matricial é uma função que associa a cada matriz $A \in \mathbb{R}^{m\times n}$ um número real $\|A\|$ e satisfaz as seguintes propriedades:*

1. $\|A\| \geqslant 0 \quad e \quad \|A\| = 0 \iff A = 0.$
2. $\|\alpha A\| = |\alpha|\|A\|$, *para todo escalar* α.
3. $\|A + B\| \leqslant \|A\| + \|B\|.$

São exemplos de normas matriciais a norma de *Frobenius* de uma matriz $A \in \mathbb{R}^{m\times n}$, que é definida por

$$\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2},$$

e a *norma-2*, definida por

$$\|A\|_2 = \max_{\|x\|_2} \|Ax\|_2.$$

Teorema 1.7. *[23, Corolário 2.4.3] Seja $A \in \mathbb{R}^{m\times n}$ e $q = \min(m,n)$. Então,*

$$\|A\|_2 = \sigma_1 \quad e \quad \|A\|_F^2 = \sigma_1^2 + \cdots + \sigma_q^2.$$

Dada uma matriz $X \in \mathbb{R}^{m\times n}$ e seja $q = \min\{m,n\}$, define-se a *norma nuclear* de X como a soma dos valores singulares de X, isto é, $\|X\|_* = \sum_{k=1}^{q} \sigma_k(X)$. O Teorema 1.8, que apresentaremos a seguir, garante que a função $\|\cdot\|_*$ é de fato uma norma. Antes, precisaremos de um resultado auxiliar.

Lema 1.2. *[50, Corolário 3.4.3] Sejam $A, B \in \mathbb{R}^{m\times n}$ matrizes tais que $\sigma_1(A) \geqslant \sigma_2(A) \geqslant \cdots \geqslant \sigma_q(A) \geqslant 0$ e $\sigma_1(B) \geqslant \sigma_2(B) \geqslant \cdots \geqslant \sigma_q(B) \geqslant 0$, $q = \min(m,n)$ e sejam $\sigma_1(A+B) \geqslant \sigma_2(A+B) \geqslant \cdots \geqslant \sigma_q(A+B) \geqslant 0$ os valores singulares ordenados de $A+B$. Então,*

$$\sum_{i=1}^{k} \sigma_i(A+B) \leqslant \sum_{i=1}^{k} \sigma_i(A) + \sum_{i=1}^{k} \sigma_i(B), \quad k = 1,\ldots,q. \tag{1.14}$$

A desigualdade (1.14) garante que a soma dos k maiores valores singulares é sub-aditiva. Entretanto, exceto para o maior valor singular σ_1, valores singulares individuais em geral não satisfazem tal desigualdade. Para ver isto, considere as matrizes $A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ e $B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$, e repare que $\sigma_2(A+B) = 1$, enquanto que $\sigma_2(A) + \sigma_2(B) = 0 + 0 = 0$.

Teorema 1.8. *[50, Corolário 3.4.4] Para uma matriz $X \in \mathbb{R}^{m\times n}$, sejam $q = \min\{m,n\}$ e $S_k(X) = \sigma_1(X) + \sigma_2(X) + \cdots + \sigma_k(X)$ a soma dos k maiores valores singulares de X. Então, $S_k(\cdot)$ satisfaz as propriedades 1, 2 e 3 da Definição 1.14, para $k = 1,\ldots,q$.*

Demonstração. Como $\sigma_i(X) \geqslant 0$, $\forall\ i = 1,\ldots,k$, segue que $S_k(X) \geqslant 0$. Se $X = 0$, então $\sigma_i(X) = 0$, $\forall\ i = 1,\ldots,q$, logo $S_k(X) = 0$. Por outro lado, $0 = S_k(X) \geqslant \sigma_1(X) = \|X\|_2 \geqslant 0$, implica $X = 0$, o que prova a propriedade 1. Para provar a propriedade 2, considere um escalar α e note que $(\alpha X)^\top(\alpha X) = \alpha^2 X^\top X$, logo $\sigma_i(\alpha X) = \sqrt{\lambda_i(\alpha^2 X^\top X)} =$

$\sqrt{\alpha^2\lambda_i(X^\top X)} = |\alpha|\sqrt{\lambda_i(X^\top X)} = |\alpha|\sigma_i(X)$, em que $\lambda_i(X^\top X) \geqslant 0$ é o i-ésimo autovalor de $X^\top X$. Portanto, $S_k(\alpha X) = |\alpha| S_k(X)$. A propriedade 3 segue do Lema 1.2. □

A norma nuclear desempenha um papel importante na área de completamento de matrizes. Parte disso, deve-se ao teorema seguinte, o qual relaciona tal conceito com o posto de uma matriz A.

Teorema 1.9. *[22, Teorema 1] No conjunto* $\mathcal{S} = \left\{X \in \mathbb{R}^{m\times n} \mid \|X\|_2 \leqslant 1\right\}$*, o envelope convexo da função* $\phi(X) = \text{posto}(X)$ *é a função* $f(X) = \|X\|_*$.

Demonstração. A prova é feita em duas partes e consiste em mostrar que $\phi^{**} = f$ no conjunto $\mathcal{S}$. O resultado segue da Observação 1.1 e do Teorema 1.3.

Parte 1 (Cálculo de ϕ^*): Sejam $\phi(X) = \text{posto}(X)$ e $\mathcal{S} = \left\{X \in \mathbb{R}^{m\times n} \mid \|X\|_2 \leqslant 1\right\}$. Pela equação (1.6) tem-se

$$\phi^*(Y) = \sup_{X\in\mathcal{S}} \left(\langle Y, X\rangle - \phi(X)\right), \tag{1.15}$$

em que $\langle Y, X\rangle = \text{Tr}\left(Y^\top X\right)$ e $Tr(A) = \sum a_{ii}$ é o traço da matriz $A \in \mathbb{R}^{m\times n}$. Seja $q = \min(m, n)$, pelo teorema do traço de von Neumann (veja a Seção 3.3 em [50]), tem-se

$$\text{Tr}\left(Y^\top X\right) \leqslant \sum_{i=1}^{q} \sigma_i(Y)\sigma_i(X), \tag{1.16}$$

em que $\sigma_i(Y)$ e $\sigma_i(X)$ são os $i-$ésimos maiores valores singulares de Y e X, respectivamente. Dada uma matriz $Y \in \mathbb{R}^{m\times n}$, pode-se mostrar, usando as propriedades do traço de uma matriz, que a igualdade em (1.16) é atingida fazendo-se $U_X = U_Y$ e $V_X = V_Y$, em que $Y = U_Y\Sigma_Y V_Y^\top$ e $X = U_X\Sigma_X V_X^\top$ são as SVDs de Y e X, respectivamente. O termo $\phi(X)$ em (1.15) não depende de U_X e V_X. Portanto, o supremo pode ser encontrado maximizando-se o primeiro termo, isto é, fazendo-se $U_X = U_Y$ e $V_X = V_Y$. Dessa forma, a equação (1.15) pode ser reescrita como

$$\phi^*(Y) = \sup_{X\in\mathcal{S}} \left(\sum_{i=1}^{q} \sigma_i(Y)\sigma_i(X) - \text{posto}(X)\right). \tag{1.17}$$

Agora, vamos considerar apenas as matrizes $X \in \mathcal{S}$ cujo posto r é no máximo q. Note que, se $\text{posto}(X) = 0$, então $\phi^*(Y) = 0$, para todo $Y \in \mathbb{R}^{m\times n}$. Por outro lado, se $\text{posto}(X) = r$, com $1 \leqslant r \leqslant q$, tomamos $\sigma_i(X) = 1$ para $i = 1, \ldots, r$ e $\sigma_i(X) = 0$ para $i = r+1, \ldots, q$, e então a expressão em (1.17) se reduz a $\phi^*(Y) = \sum_{i=1}^{r} \sigma_i(Y) - r$. Como temos que considerar todos os casos para $r = 0, \ldots, q$, a expressão de $\phi^*(Y)$ fica

$$\phi^*(Y) = \max\left(0, \sigma_1(Y) - 1, \ldots, \sum_{i=1}^{r} \sigma_i(Y) - r, \ldots, \sum_{i=1}^{q} \sigma_i(Y) - q\right). \tag{1.18}$$

Repare que, se $\|Y\|_2 \leqslant 1$, então $\sigma_i(Y) \leqslant 1$ e, portanto, segue de (1.18) que $\phi^*(Y) = 0$. Por outro lado, para $\|Y\|_2 > 1$, o maior termo de (1.18) é aquele que soma todos os termos positivos $(\sigma_i(Y) - 1)$ e isto ocorre quando $\sigma_r(Y) > 1$ e $\sigma_{r+1}(Y) \leqslant 1$. Portanto, temos

$$\phi^*(Y) = \begin{cases} 0, & \text{se } \|Y\|_2 \leqslant 1 \\ \sum_{i=1}^{r} \sigma_i(Y) - r, & \sigma_r(Y) > 1 \text{ e } \sigma_{r+1}(Y) \leqslant 1, \quad \text{se} \quad \|Y\|_2 > 1 \end{cases} \tag{1.19}$$

$$= \sum_{i=1}^{q} (\sigma_i(Y) - 1)_+ , \tag{1.20}$$

em que $t_+ = \max(t, 0)$.

Parte 2 (Cálculo de ϕ^{**}): Usando o raciocínio anterior, podemos escrever o conjugado de ϕ^*, como

$$\phi^{**}(Z) = \sup_Y \left(\sum_{i=1}^{q} \sigma_i(Z)\sigma_i(Y) - \phi^*(Y) \right). \tag{1.21}$$

Vamos considerar dois casos, $\|Z\|_2 > 1$ e $\|Z\|_2 \leqslant 1$:

Se $\|Z\|_2 > 1$, podemos escolher $\sigma_1(Y)$ grande o bastante para que $\phi^{**}(Z) \to \infty$. Para ver isso, note que, em

$$\phi^{**}(Z) = \sup_Y \left(\sum_{i=1}^{q} \sigma_i(Z)\sigma_i(Y) - \sum_{i=1}^{q} (\sigma_i(Y) - 1)_+ \right) \tag{1.22}$$

$$= \sup_Y \left((\sigma_1(Z) - 1)\sigma_1(Y) + \sum_{i=2}^{q} \sigma_i(Z)\sigma_i(Y) - \sum_{i=2}^{q} (\sigma_i(Y) - 1)_+ \right), \tag{1.23}$$

o coeficiente de $\sigma_1(Y)$ é igual ao número positivo $(\sigma_1(Z) - 1)$ e, como Y é irrestrito, pode-se tomar $\sigma_1(Y)$ tão grande quanto se queira.

Agora, seja $\|Z\|_2 \leqslant 1$. Se $\|Y\|_2 \leqslant 1$, então segue que $\phi^*(Y) = 0$ e o supremo é atingido para $\sigma_i(Y) = 1$, $i = 1, \ldots, q$, fornecendo

$$\phi^{**}(Z) = \sum_{i=1}^{q} \sigma_i(Z) = \|Z\|_*.$$

Agora mostraremos que, se $\|Y\|_2 > 1$, o argumento do supremo é sempre menor ou igual que o valor acima. Adicionando e subtraindo $\sum_{i=1}^{q} \sigma_i(Z) = \|Z\|_*$ e rearranjando os termos, temos, segundo (1.19)

$$\begin{aligned} \sum_{i=1}^{q} \sigma_i(Z)\sigma_i(Y) - \sum_{i=1}^{q} (\sigma_i(Y) - 1)_+ &= \sum_{i=1}^{q} \sigma_i(Z)\sigma_i(Y) - \sum_{i=1}^{r} (\sigma_i(Y) - 1) \\ &= \sum_{i=1}^{q} \sigma_i(Z)\sigma_i(Y) - \sum_{i=1}^{r} (\sigma_i(Y) - 1) - \sum_{i=1}^{q} \sigma_i(Z) + \|Z\|_* \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{r}(\sigma_i(Z)-1)(\sigma_i(Y)-1) + \sum_{i=r+1}^{q}(\sigma_i(Y)-1)\sigma_i(Z) + \|Z\|_* \\
&\leqslant \|Z\|_*
\end{aligned}
$$

A desigualdade acima segue dos fatos: $\|Z\|_2 \leqslant 1$ implica $\sigma_i(Z) - 1 \leqslant 0$ para $i = 1, \ldots, r$, $\sigma_r(Y) > 1$ e $\sigma_i(Y) \leqslant 1$ para $i = r+1, \ldots, q$. Assim

$$
\sum_{i=1}^{r}(\sigma_i(Z)-1)(\sigma_i(Y)-1) + \sum_{i=r+1}^{q}(\sigma_i(Y)-1)\sigma_i(Z) \leqslant 0.
$$

□

Os resultados do teorema acima podem ser estendidos [22] para o conjunto limitado $S = \left\{X \in \mathbb{R}^{m\times n} \mid \|X\|_2 \leqslant M\right\}$. Neste caso, o envelope convexo da função posto(X) no conjunto S é a função $\frac{1}{M}\|X\|_*$, isto é, posto$(X) \geqslant \frac{1}{M}\|X\|_*$ para todo $X \in S$.

Definição 1.15 (Soft-thresholding)**.** *Seja $A \in \mathbb{R}^{m\times n}$ uma matriz de posto r, cuja SVD é dada por*

$$
A = U\Sigma V^\top,
$$

em que $U \in \mathbb{R}^{m\times r}$, $\Sigma \in \mathbb{R}^{r\times r}$ e $V \in \mathbb{R}^{n\times r}$. Para cada $\lambda > 0$, define-se o operador soft-thresholding $S_\lambda(\cdot)$ da seguinte forma:

$$
S_\lambda(A) := U\Sigma_\lambda V^\top, \tag{1.24}
$$

em que $\Sigma_\lambda = \operatorname{diag}[(\sigma_1 - \lambda)_+, \cdots, (\sigma_r - \lambda)_+]$, com $t_+ = \max(0, t)$.

Teorema 1.10. *[14, Teorema 2.1] Seja $A \in \mathbb{R}^{m\times n}$ uma matriz de posto r e $\lambda \geqslant 0$. A solução do problema de otimização*

$$
\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|X - A\|_F^2 + \lambda\|X\|_*, \tag{1.25}
$$

é dada por $S_\lambda(A)$.

Demonstração. Seja $f(X) = \frac{1}{2}\|X - A\|_F^2 + \lambda\|X\|_*$. Como f é a soma de uma função fortemente convexa e uma função convexa, ela é uma função estritamente convexa e, portanto, possui um único minimizador, digamos $\hat{X}$. Resta mostrar que $\hat{X} = S_\lambda(A)$. Para $\lambda = 0$, o resultado segue fazendo-se $\hat{X} = A = S_0(A)$. Suponha agora que $\lambda > 0$. Pelo Lema 1.1 basta mostrar que

$$
0 \in \partial f(\hat{X}) = \hat{X} - A + \lambda\partial\|\hat{X}\|_*,
$$

em que $\partial\|\hat{X}\|_*$ é o subdiferencial da norma nuclear em $\hat{X}$. Seja $X \in \mathbb{R}^{m\times n}$ uma matriz arbitrária e $U\Sigma V^\top$ sua SVD. É possível mostrar [58] que

$$
\partial\|X\|_* = \left\{UV^\top + W : W \in \mathbb{R}^{m\times n},\ U^\top W = 0,\ WV = 0,\ \|W\|_2 \leqslant 1\right\}.
$$

Tomemos $\hat{X} := S_\lambda(A)$ e consideremos a SVD de A como $A = U_0\Sigma_0 V_0^\top + U_1\Sigma_1 V_1^\top$, em que U_0, V_0 (U_1, V_1) são os vetores singulares associados com os valores singulares maiores que λ (menores ou iguais que λ). Assim, $\hat{X} = U_0(\Sigma_0 - \lambda I)V_0$ e, portanto,

$$A - \hat{X} = \lambda\left(U_0 V_0^\top + W\right), \quad W = \lambda^{-1} U_1 \Sigma_1 V_1^\top.$$

Por construção de U_0, V_0, U_1, V_1, tem-se que $U_0^\top W = 0$ e $WV_0 = 0$, e uma vez que os elementos da diagonal de Σ_1 têm magnitudes limitadas por λ, segue que $\|W\|_2 \leqslant 1$. Consequentemente, $A - \hat{X} \in \lambda\partial\|\hat{X}\|_* \Rightarrow 0 \in \hat{X} - A + \lambda\partial\|\hat{X}\|_*$, o que conclui a prova. □

Encerramos esta seção com o resultado que mostra que o operador soft-thresholding é não-expansivo e, consequentemente, uniformemente contínuo.

Lema 1.3. *[39, Lema 3] Para todos $X, Y \in \mathbb{R}^{m\times n}$, tem-se*

$$\|S_\lambda(X) - S_\lambda(Y)\|_F^2 \leqslant \|X - Y\|_F^2.$$

1.3 O método do gradiente proximal

Dada uma função convexa, própria e fechada $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, o operador proximal $\text{prox}_f : \mathbb{R}^n \to \mathbb{R}^n$ de f é definido por

$$\text{prox}_f(v) = \arg\min_x \left\{ \frac{1}{2}\|x - v\|_2^2 + f(x) \right\}, \tag{1.26}$$

em que $\|\cdot\|_2$ é a norma Euclidiana usual. Note que a expressão que está sendo minimizada do lado direito (1.26) é fortemente convexa e não é identicamente igual a $+\infty$ (pela definição de f), de modo que ela possui um único minimizador para todo $v \in \mathbb{R}^n$.

Exemplo 1.2. *Seja $f : \mathbb{R}^n \to \mathbb{R}$ definida por $f(x) = \frac{1}{2}x^\top Ax + b^\top x + c$ em que $A \in \mathbb{R}^{n\times n}$ é simétrica definida positiva, $b \in \mathbb{R}^n$ e $c \in \mathbb{R}$. O vetor $\text{prox}_f(v)$ é o minimizador do problema*

$$\min_x \left\{ \frac{1}{2}\|x - v\|_2^2 + \frac{1}{2}x^\top Ax + b^\top x + c \right\} \tag{1.27}$$

A solução ótima de (1.27) é atingida quando o gradiente da função objetivo é igual a zero:

$$Ax + b + x - v = 0,$$

isto é, quando

$$(A + I)x = v - b,$$

logo,

$$\text{prox}_f(v) = (A + I)^{-1}(v - b).$$

Neste trabalho, utilizaremos o operador proximal da função escalada tf, com $t > 0$, que é definido por

$$\text{prox}_{tf}(v) = \arg\min_x \left\{ \frac{1}{2t}\|x - v\|_2^2 + f(x) \right\}. \tag{1.28}$$

Geometricamente, $\text{prox}_{tf}(v)$ é um ponto que minimiza f e ainda se mantém próximo de v. O fator de escala t controla a extensão com que o operador proximal mapeia pontos na direção do mínimo de f.

Agora que já definimos o conceito de operador proximal e fornecemos sua interpretação geométrica, podemos apresentar o método do gradiente proximal.

Considere o problema de otimização da soma de duas funções dado por

$$\min_{x\in\mathbb{R}^n} f(x) \equiv g(x) + h(x), \tag{1.29}$$

em que assumimos as seguintes hipóteses:

H1: $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ é uma função própria, fechada e convexa, não necessariamente diferenciável;

H2: $g : \mathbb{R}^n \to \mathbb{R}$ é uma função convexa e continuamente diferenciável, com gradiente Lipschitz contínuo com constante $L > 0$:

$$\|\nabla g(x) - \nabla g(y)\|_2 \leqslant L\|x - y\|_2, \quad \forall\, x, y \in \mathbb{R}^n;$$

H3: O conjunto ótimo de (1.29) é não-vazio, isto é, $\mathcal{X}^\star = \arg\min_x f(x) \neq \varnothing$.

Repare que, se f fosse diferenciável, então com o intuito de resolver (1.29) poderíamos considerar o método clássico do gradiente:

$$x^0 \in \mathbb{R}^n, \quad x^{k+1} = x^k - t^k \nabla f(x^k). \tag{1.30}$$

Como f não é diferenciável (já que h pode não o ser), o que se faz é minimizar a aproximação quadrática $Q_t(x, y)$ de f em torno de um ponto arbitrário y: para $t > 0$

$$Q_t(x, y) = g(y) + \langle \nabla g(y), x - y\rangle + \frac{1}{2t}\|x - y\|_2^2 + h(x). \tag{1.31}$$

Note que $Q_t(\cdot, y)$ é fortemente convexa, pois é a soma de uma função fortemente convexa e uma função convexa. Consequentemente, $Q_t(x, y)$ admite um único minimizador, digamos $\text{p}_{th}(y)$. Assim,

$$\text{p}_{th}(y) := \arg\min_x \{Q_t(x, y)\} \tag{1.32}$$

$$= \arg\min_x \left\{ g(y) + \langle \nabla g(y), x - y\rangle + \frac{1}{2t}\|x - y\|_2^2 + h(x) \right\} \tag{1.33}$$

$$= \arg\min_x \left\{ \frac{1}{2t}\|x - (y - t\nabla g(y))\|_2^2 + h(x) + g(y) - \frac{t}{2}\|\nabla g(y)\|_2^2 \right\} \tag{1.34}$$

$$= \arg\min_x \left\{ \frac{1}{2t}\|x - (y - t\nabla g(y))\|_2^2 + h(x) \right\} \tag{1.35}$$

$$= \text{prox}_{th}\left(y - t\nabla g(y)\right), \tag{1.36}$$

em que a equivalência entre (1.34) e (1.35) se dá eliminando-se o termo constante $g(y) - \frac{t}{2}\|\nabla g(y)\|_2^2$. Portanto, o método do gradiente proximal é descrito como,

$$x^0 \in \mathbb{R}^n, \quad x^{k+1} = \text{prox}_{th}\left(x^k - t\nabla g(x^k)\right), \tag{1.37}$$

em que $t > 0$ e $\text{prox}_{th}(\cdot)$ é o operador proximal. A equação (1.37) é calculada em dois passos: primeiro se computa a atualização pelo gradiente de g e, em seguida, o operador proximal. Com isto, garante-se que o ponto x^{k+1} se move na direção do mínimo de h e ao mesmo tempo permanece próximo da atualização do gradiente de g.

Uma das vantagens do método do gradiente proximal é que $\text{prox}_{th}(\cdot)$ pode ser calculado analiticamente para algumas funções importantes. Por outro lado, a implementação pode ser computacionalmente cara dependendo da função h, o que pode comprometer a performance do método.

A convergência do método descrito em (1.37) é dada pelo teorema:

Teorema 1.11. *[6, Teorema 3.1] Sejam f uma função satisfazendo as hipóteses do problema (1.29) e $\{x^k\}$ a sequência gerada por (1.37). Então, para um tamanho de passo fixo $t \in (0, 1/L]$ e qualquer $k \geqslant 1$ tem-se*

$$f(x^k) - f(x^\star) \leqslant \frac{\|x^0 - x^\star\|_2^2}{2tk}, \ \forall x^\star \in \mathcal{X}^\star. \tag{1.38}$$

Uma versão do Teorema 1.11 para o caso em que o tamanho de passo é encontrado em uma busca linear pode ser vista em [46]. A desigualdade (1.38) nos diz que, dado um número real $\varepsilon > 0$, o número k de iterações necessárias para se obter um $\hat{x}$, tal que $f(\hat{x}) - f(x^\star) \leqslant \varepsilon$, é dado por $\lceil C/\varepsilon \rceil$, com $C = \frac{\|x^0 - x^\star\|_2^2}{2t}$. Isto é, o método do gradiente proximal tem taxa de convergência de $O(1/k)$ ou complexidade de iteração $O(1/\varepsilon)$.

Há, ainda, a versão acelerada do método descrito em (1.37), conhecida como gradiente proximal acelerado, no qual se considera o sistema de equações [6, 46] dado por

$$\begin{aligned} x^{k+1} &= \text{prox}_{th}\left(z^k - t\nabla g(z^k)\right) \\ z^{k+1} &= x^{k+1} + \omega^k\left(x^{k+1} - x^k\right), \end{aligned} \tag{1.39}$$

em que ω^k é um parâmetro de extrapolação, o qual deve ser escolhido de modo adequado para que haja convergência e aceleração do método. Neste trabalho, usaremos $\omega^k = (k-1)/(k+2)$, com o qual a taxa de convergência de (1.39) é dada pelo teorema, a seguir.

Teorema 1.12. *[6, Teorema 4.4] Sejam f uma função satisfazendo as hipóteses do problema (1.29) e $\{x^k\}$ e $\{z^k\}$ sequências geradas por (1.39). Então, para um tamanho de passo fixo $t \in (0, 1/L]$ e qualquer $k \geqslant 1$ tem-se*

$$f(x^k) - f(x^\star) \leqslant \frac{2\|x^0 - x^\star\|_2^2}{t(k+1)^2}, \ \forall x^\star \in \mathcal{X}^\star. \tag{1.40}$$

Encerramos esta seção apresentando um exemplo importante, no qual conhecemos uma fórmula fechada para o operador proximal.

Exemplo 1.3 (LASSO [54])**.** *Dados um vetor de resultados $b \in \mathbb{R}^n$ e uma matriz de variáveis preditoras $M \in \mathbb{R}^{n\times p}$. Considere o problema de otimização dado por*

$$\arg\min_{\beta\in\mathbb{R}^p} \left\{ \frac{1}{2}\|b - M\beta\|_2^2 + \lambda\|\beta\|_1 \right\}, \tag{1.41}$$

onde $\lambda \geqslant 0$ é um parâmetro de regularização do modelo. O problema (1.41) é conhecido como LASSO[2] *e é uma ferramenta bastante usada para se encontrar estimativas para os coeficientes em um modelo de regressão linear com dados esparsos.*

Definindo-se $g(\beta) = \frac{1}{2}\|b - M\beta\|_2^2$ e $h(\beta) = \lambda\|\beta\|_1$, tem-se $\nabla g(\beta) = -M^\top(b - M\beta)$ e o operador proximal dado por

$$\begin{aligned} prox_{th}(\beta) &= \arg\min_x \left\{ \frac{1}{2t}\|x - \beta\|_2^2 + \lambda\|x\|_1 \right\} \\ &= S_{\lambda t}(\beta), \end{aligned}$$

em que $S_{\lambda t}(\beta)$ é o operador soft-thresholding aplicado ao vetor β, cuja expressão, neste caso (vetorial), é dada por

$$[S_{\lambda t}(\beta)]_i = \begin{cases} \beta_i - \lambda t, & \text{se} \quad \beta_i > \lambda t \\ 0, & \text{se} \quad -\lambda t \leqslant \beta_i \leqslant \lambda t \\ \beta_i + \lambda t, & \text{se} \quad \beta_i < -\lambda t. \end{cases}$$

Portanto, a atualização do gradiente proximal é dada por

$$\beta^{k+1} = S_{\lambda t}\left(\beta^k + tM^\top(b - M\beta^k)\right).$$

No Capítulo 2, apresentaremos o Problema de Completamento de Matrizes e formulações envolvendo o posto de uma matriz e a norma nuclear como sua relaxação convexa. A Decomposição em Valores Singulares será usada para explorar a noção de (in)coerência de uma matriz, a qual é usada para medir o "espalhamento" de suas entradas não-nulas. Por fim, alguns algoritmos para o problema de completamento serão apresentados e os resultados das Seções 1.2 e 1.3 serão usados na Seção 2.3.3 para mostrar que o algoritmo Soft-Impute [39] é um método de gradiente proximal.

[2] Acrônimo para Least Absolute Shrinkage and Selection Operator.

2 Completamento de Matrizes de Posto Reduzido

Este capítulo introduz o Problema de Completamento de Matrizes de Posto Reduzido, bem como as principais definições e resultados pertinentes ao tema. Discutimos, detalhadamente, o papel do posto para o problema de completamento e fornecemos um estudo sobre as hipóteses comumente usadas na literatura de completamento de matrizes, no qual conceitos importantes como incoerência e isometria restrita são explorados. Em seguida, apresentamos uma breve revisão de alguns algoritmos bem estabelecidos na literatura e que servirão de benchmark em capítulos posteriores. Na última seção, mostramos algumas aplicações práticas importantes. As principais referências para este capítulo são [15, 16, 31, 44, 47, 48].

2.1 O Problema de Completamento de Matrizes

O Problema de Completamento de Matrizes consiste em completar/recuperar uma matriz a partir de uma amostra de suas entradas. Em geral, obter uma recuperação precisa é uma tarefa impossível. No entanto, assumir que a matriz desconhecida tem posto baixo é um passo importante, pois torna a busca por soluções uma tarefa possível. O exemplo, a seguir, nos dá uma noção do papel do posto no contexto de completamento.

Exemplo 2.1. *Considere as matrizes M e N dadas por*

$$M = \begin{bmatrix} 1 & ? & -1 & 2 \\ ? & 9 & ? & 6 \\ ? & ? & 7 & ? \\ 5 & ? & ? & ? \end{bmatrix} \quad \text{e} \quad N = \begin{bmatrix} -3 & ? & 2 & -7/3 \\ ? & 5 & ? & ? \\ ? & 10 & ? & ? \\ 9 & ? & -6 & 7 \end{bmatrix},$$

em que "?" representa as entradas em falta de M e N. Está claro que sem hipóteses adicionais haveriam infinitas possibilidades de completar as entradas desconhecidas de M e N. Entretanto, assumindo-se que $\text{posto}(M) = 1$*, pode-se facilmente completar as entradas em falta de M, de modo a obter-se uma única solução*

$$M = \begin{bmatrix} 1 & 3 & -1 & 2 \\ 3 & 9 & -3 & 6 \\ -7 & -21 & 7 & -14 \\ 5 & 15 & -5 & 10 \end{bmatrix}.$$

Por outro lado, mesmo assumindo-se que a matriz N tem posto igual 1, verifica-se que é impossível obtermos uma recuperação única para tal matriz. Alguns completamentos possíveis para N são

$$N_1 = \begin{bmatrix} -3 & 1 & 2 & -7/3 \\ -15 & 5 & 10 & -35/3 \\ -30 & 10 & 20 & -70/3 \\ 9 & -3 & -6 & 7 \end{bmatrix} \quad \text{e} \quad N_2 = \begin{bmatrix} -3 & -10/3 & 2 & -7/3 \\ 4.5 & 5 & -3 & 3.5 \\ 9 & 10 & -6 & 7 \\ 9 & 10 & -6 & 7 \end{bmatrix}.$$

Assim, em virtude da importância do posto para o Problema de Completamento, uma maneira comum de abordá-lo é em termos do problema de minimização do posto [15, 44], dado por

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & \text{posto}(X) \\ \text{sujeito a} \quad & P_\Omega(X) = P_\Omega(A), \end{aligned} \tag{2.1}$$

em que X é a variável de decisão e Ω é o conjunto de índices das entradas conhecidas da matriz A e $P_\Omega(\cdot)$ é o operador projeção, definido como

$$[P_\Omega(X)]_{ij} := \begin{cases} X_{ij}, & \text{se } (i,j) \in \Omega \\ 0, & \text{caso contrário,} \end{cases} \tag{2.2}$$

com $P_\Omega^\perp(\cdot)$ dado por $P_\Omega^\perp(X) = X - P_\Omega(X)$.

O raciocínio subjacente à formulação acima é o seguinte: se a informação (entradas conhecidas) está concentrada nas $r := \text{posto}(A) \ll \min(m,n)$ colunas (linhas) linearmente independentes da matriz A, então o restante das colunas (linhas) são combinações lineares de tais colunas (linhas) de A e, portanto, podem ser recuperadas, desde que algumas condições sejam satisfeitas. Apesar de sua importância teórica, o problema (2.1) é não-convexo e combinatoriamente difícil para conjuntos Ω gerais [44, 51]. Além disso, todos os algoritmos que fornecem soluções exatas para este problema requerem um tempo exponencial na dimensão n (supondo que $m = n$) da matriz [15, 18].

Uma forma de driblar as dificuldades da formulação (2.1), é considerar a norma nuclear como uma relaxação convexa para a função posto(X). Assim, temos

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & \|X\|_* \\ \text{sujeito a} \quad & P_\Omega(X) = P_\Omega(A). \end{aligned} \tag{2.3}$$

Antes de prosseguirmos, cabe ressaltar que o conjunto viável do problema (2.3) não é limitado. Logo, as hipóteses do Teorema 1.9 e da discussão subsequente não

são satisfeitas e, portanto, a norma nuclear não é o envelope convexo da função posto. Entretanto, o ponto-chave, neste caso, é que a norma nuclear é uma função convexa que pode ser minimizada eficientemente usando algumas técnicas de otimização como a programação semidefinida [16].

Outros modelos consideram que as entradas da matriz são obtidas por meio de processos ruidosos. Nesses casos, pode ser útil considerar uma tolerância associada ao nível de ruído nas entradas conhecidas da matriz. Assim, uma formulação alternativa ao problema (2.3) pode ser dada por

$$\begin{aligned} \min_{X \in \mathbb{R}^{m \times n}} \quad & \|X\|_* \\ \text{sujeito a} \quad & \|P_\Omega(X - A)\|_F^2 \leqslant \delta, \end{aligned} \tag{2.4}$$

em que $\delta \geqslant 0$ controla o nível de ruído nas entradas conhecidas. Note que, se $\delta = 0$, então (2.4) se reduz a (2.3).

O Exemplo 2.1 deixa claro que somente a hipótese de que a matriz desconhecida tenha posto baixo não é suficiente para garantir que o problema de completamento tenha solução única. Com efeito, veremos que condições adicionais sobre a quantidade mínima de entradas conhecidas e também sobre a posição de tais entradas são hipóteses fundamentais para o sucesso do completamento.

2.2 Hipóteses comuns no problema de completamento

Já vimos, no Exemplo 2.1, que a hipótese do posto reduzido não é suficiente para garantir o sucesso do completamento. Nesta seção, discorreremos sobre as propriedades intrínsecas do problema de completamento.

2.2.1 Quantidade mínima de entradas conhecidas

A quantidade mínima de entradas conhecidas que uma matriz alvo deve possuir, para que possa ser completada eficientemente, pode ser determinada a partir de uma noção de grau de liberdade, que se trata do número de variáveis que são escolhidas livremente em uma matriz [16, 44].

Vamos analisar o número de graus de liberdade de uma matriz. Para isto, seja $A \in \mathbb{R}^{m \times n}$ uma matriz de posto r. Como o posto de A é r, podemos escolher livremente os valores para as entradas das r colunas linearmente independentes de A, resultando em mr graus de liberdade para essas r colunas. Uma vez que r colunas linearmente independentes, digamos $A_1, \ldots, A_r$, são construídas, cada uma das $n - r$ colunas restantes é escrita como uma combinação linear de $A_1, \ldots, A_r$. Assim, para cada coluna $A_i = \alpha_{1i}A_1 + \cdots + \alpha_{ri}A_r$, com $i = r + 1, \ldots, n$, pode-se escolher livremente os valores dos r coeficientes $\alpha_{1i}, \ldots, \alpha_{ri}$.

Assim, o número de graus de liberdade para as $n-r$ colunas é $(n-r)r$. Portanto, o número de graus de liberdade de uma matriz $A \in \mathbb{R}^{m\times n}$ de posto r é dado pela fórmula [44]

$$(m+n)r - r^2. \tag{2.5}$$

Agora, podemos usar a fórmula (2.5) para calcular o número de graus de liberdade das matrizes M e N do Exemplo 2.1. Com efeito, supondo que tais matrizes tenham posto igual a 1, então o número de graus de liberdade de M e N é 7. Por outro lado, foi possível completar M de maneira única com apenas 7 entradas conhecidas, enquanto que N não tem recuperação única, embora possamos observar 8 entradas na matriz (quantidade maior que o número de graus de liberdade de N).

Assim, o número de graus de liberdade é fundamental, pois ele dá a quantidade mínima de entradas conhecidas que uma matriz deve possuir (supondo que não se estabeleça condições sobre as posições das entradas) para que possamos completá-la. Se a quantidade de entradas observadas $|\Omega|$ é tal que $|\Omega| < (m+n)r - r^2$, então não importa quais entradas de A estão disponíveis, haverá infinitas possibilidades de completamento e a obtenção de uma solução única é impossível [16].

Observação 2.1. *Apesar de alguns trabalhos afirmarem [16, 20, 44] que o número de graus de liberdade estabelece uma quantidade mínima de entradas conhecidas que uma matriz deve ter para que possa ser recuperada, veremos, a seguir, que há outros resultados com limitantes inferiores que dependem da forma como as entradas conhecidas são amostradas.*

2.2.2 Posições das entradas conhecidas

Como vimos na discussão acima, dispor de uma quantidade suficiente de entradas conhecidas é importante, mas, para evitar que o problema seja mal-posto, caso em que admite infinitas soluções, é preciso estarmos atentos às posições de tais entradas na matriz.

Uma situação crítica para o problema é quando a matriz tem uma linha ou coluna inteira de entradas desconhecidas. Nesse caso, é impossível recuperamos a matriz. Para ver isso, considere a matriz de posto unitário $B = uv^\top$, com $u \in \mathbb{R}^m$ e $v \in \mathbb{R}^n$. Assim,

$$B = \begin{bmatrix} u_1v_1 & u_1v_2 & \dots & u_1v_j & \dots & u_1v_n \\ u_2v_1 & u_2v_2 & \dots & u_2v_j & \dots & u_2v_n \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ u_iv_1 & u_iv_2 & \dots & u_iv_j & \dots & u_iv_n \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ u_mv_1 & u_mv_2 & \dots & u_mv_j & \dots & u_mv_n \end{bmatrix}.$$

Note que u_i só aparece na linha i e v_j só aparece na coluna j. Assim, é impossível determinar o valor de u_i, se a linha i for desconhecida. O mesmo vale para v_j. Portanto, para que uma matriz possa ser completada, é necessário que ela tenha pelo menos uma entrada observada em cada linha ou coluna.

2.2.3 Espalhamento das entradas da matriz

Seja $M \in \mathbb{R}^{m\times n}$ uma matriz de posto r. Se M tiver suas entradas não-nulas concentradas em uma determinada região da matriz, é possível que tenhamos uma amostra de entradas pertencente ao espaço nulo de P_Ω, inviabilizando sua recuperação. O exemplo (clássico) a seguir ilustra essa situação.

Exemplo 2.2. *Considere a seguinte matriz $M \in \mathbb{R}^{n\times n}$ (com n grande) de posto unitário*

$$M = \begin{bmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}.$$

Note que, se todas as entradas dessa matriz forem conhecidas, exceto as quatro entradas não-nulas do canto superior esquerdo, então é impossível recuperar essa matriz, mesmo que tenhamos quase 100% *de entradas conhecidas.*

Uma maneira de medir a concentração das entradas não-nulas de uma matriz é através da noção de coerência.

Definição 2.1 (Coerência [15])**.** *Seja U um subespaço de $\mathbb{R}^n$ de dimensão r e P_U a projeção ortogonal sobre U. A coerência de U (com relação à base canônica (e_i)) é definida como*

$$\mu(U) = \frac{n}{r} \max_{1\leqslant i\leqslant n} \|P_U e_i\|_2^2. \tag{2.6}$$

Pode-se mostrar que $1 \leqslant \mu(U) \leqslant \dfrac{n}{r}$ (veja [15, p. 6]). Quanto maior o valor de $\mu(U)$ mais coerente o subespaço U é em relação base canônica (e_i), no sentido de que os vetores da base de U podem estar quase alinhados com os vetores canônicos, resultando em um valor de $\mu(U)$ grande. Como P_U é um projetor ortogonal, segue que $\|P_U e_i\|_2^2 \leqslant \|e_i\|_2^2$, logo $\mu(U)$ atinge seu valor máximo quando U contém algum vetor canônico.

No caso matricial, a análise é feita nos vetores singulares da matriz. Para isso, seja $M \in \mathbb{R}^{n_1\times n_2}$ uma matriz de posto r, com sua Decomposição em Valores Singulares compacta dada por

$$M = U\Sigma V^\top = \sum_{j=1}^{r} \sigma_j u_j v_j^\top. \tag{2.7}$$

Agora, considere as seguintes matrizes

$$P_U = UU^\top, \quad P_V = VV^\top \quad \text{e} \quad E = UV^\top, \tag{2.8}$$

em que P_U e P_V são os projetores ortogonais no espaço coluna e espaço linha de M, respectivamente.

Usando a expressão de P_U definida em (2.8) podemos escrever $\|P_U e_i\|_2^2$ em (2.6) como

$$\begin{aligned}
\|P_U e_i\|_2^2 &= (P_U e_i)^\top (P_U e_i) \\
&= e_i^\top P_U^\top P_U e_i \\
&= e_i^\top UU^\top UU^\top e_i \\
&= e_i^\top UU^\top e_i, \quad U^\top U = I \\
&= \|U_i\|_2^2,
\end{aligned}$$

em que U_i é a i-ésima linha de U.

Assim, podemos escrever (2.6) como

$$\mu(U) = \frac{n_1}{r} \max_{1 \leqslant i \leqslant n_1} \|U_i\|_2^2. \tag{2.9}$$

Por outro lado, U é uma matriz com r colunas ortonormais, o que implica que $0 \leqslant |U_{ij}|^2 \leqslant 1$ e $\|U_i\|_2^2 \leqslant r$. Assim, numa situação de baixa coerência (ideal), ou seja, quando $\mu(U)$ é próximo de 1, teremos que ter $\max_{1 \leqslant i \leqslant n_1} \|U_i\|_2^2$ próximo de r/n_1 (relembre que $r \ll n_1$) e isto impõe que as colunas de U não tenham muitas entradas iguais a zero.

Em geral, a coerência é maximizada quando há uma concentração de entradas não-nulas em poucas linhas ou em poucas colunas da matriz. Então, é importante garantir que ambos, o espaço linha e o espaço coluna da matriz, possuam baixa coerência. Veja o exemplo a seguir:

Exemplo 2.3. *Considere a matriz* $M \in \mathbb{R}^{1000 \times 1000}$ *dada por*

$$M = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix},$$

e sejam $R(M)$ o espaço coluna de M e $R(M^\top)$ o espaço linha de M. Tem-se que $\mu(R(M)) = 1$ e $\mu(R(M^\top)) = 1000$. Note que, neste caso, $R(M)$ tem coerência mínima (ideal), enquanto que $R(M^\top)$ tem coerência alta.

Agora que já fornecemos os requisitos do problema de completamento, vamos apresentar os principais teoremas da literatura sobre a garantia de recuperação de uma matriz em termos de probabilidades. Para isto, considere as seguintes hipóteses para as matrizes descritas em (2.7) e (2.8).

A1: A coerência obedece $\max(\mu(U), \mu(V)) \leqslant \mu_0$, para algum μ_0.

A2: Existe $\mu_1 > 0$ tal que $\forall (i,j) \in \{1,\dots,n_1\}\times\{1,\dots,n_1\}$ e $(k,l) \in \{1,\dots,n_2\}\times\{1,\dots,n_2\}$ tem-se

$$\langle e_i, P_U e_j\rangle - \frac{r}{n_1} 1_{i=j} \leqslant \mu_1 \frac{\sqrt{r}}{n_1} \tag{2.10}$$

$$\langle e_k, P_V e_l\rangle - \frac{r}{n_2} 1_{k=l} \leqslant \mu_1 \frac{\sqrt{r}}{n_2}, \tag{2.11}$$

em que $1_{i=j} = 1$ se $i = j$ e 0, caso contrário.

A3: Existe $\mu_2 > 0$ tal que $\forall (i,j) \in \{1,\dots,n_1\} \times \{1,\dots,n_2\}$

$$|E_{ij}| \leqslant \mu_2 \frac{\sqrt{r}}{\sqrt{n_1 n_2}}. \tag{2.12}$$

Note que, se **A1** é satisfeita, então desigualdade (2.12) sempre vale para $\mu_2 = \mu_0\sqrt{r}$. Com efeito, pela Desiguldade de Cauchy-Schwarz temos

$$|E_{ij}| = \left|\sum_{k=1}^{r} u_{ik} v_{jk}\right| \leqslant \sqrt{\sum_{k=1}^{r} |u_{ik}|^2} \cdot \sqrt{\sum_{k=1}^{r} |v_{jk}|^2} \leqslant \frac{\mu_0 r}{\sqrt{n_1 n_2}}. \tag{2.13}$$

Assim, para valores de r pequenos, μ_2 e μ_0 são comparáveis. Na prática, isso permite que μ_2 seja maior que μ_0, permitindo um limitante superior maior para os valores absolutos das entradas da matriz E.

Definição 2.2. *Dizemos que uma matriz M é* incoerente *relativa aos parâmetros $\mu_0 > 0$ e $\mu_2 > 0$ se **A1** e **A3** são satisfeitas. Por outro lado, M é dita* fortemente incoerente *com relação ao parâmetro $\mu_3 > 0$, se **A1** e **A2** são satisfeitas, com $\max(\mu_0, \mu_1) \leqslant \mu_3$.*

O teorema a seguir é considerado um dos principais resultados da área e talvez tenha sido um dos primeiros a estabelecer condições para a recuperação de uma matriz de forma única:

Teorema 2.1. *[15, Teorema 1.3] Seja $M \in \mathbb{R}^{n_1 \times n_2}$ uma matriz incoerente de posto r, defina $n := \max(n_1, n_2)$. Suponha que observamos m entradas de M com posições amostradas aleatoriamente com distribuição uniforme. Então existem constantes positivas C e c tais que, se*

$$m \geqslant C \max(\mu_2^2, \mu_0^{1/2}\mu_2, \mu_0 n^{1/4}) nr(\beta \log n),$$

para algum $\beta > 2$, então o minimizador para o problema (2.3) é único e igual a M com probabilidade no mínimo de $1 - cn^{-\beta}$. Para $r \leqslant \mu_0^{-1} n^{1/5}$ essa estimativa melhora para

$$m \geqslant C\mu_0 n^{6/5} r\beta \log n,$$

com a mesma probabilidade de sucesso.

O próximo teorema melhora o resultado anterior, mas sob a condição de incoerência forte, a qual é mais restritiva que a hipótese `A1` [47].

Teorema 2.2. *[16, Teorema 1.1] Seja $M \in \mathbb{R}^{n_1 \times n_2}$ uma matriz de posto r, obedecendo a propriedade de incoerência forte relativa ao parâmetro μ_3 e defina $n := \max(n_1, n_2)$. Suponha que observamos m entradas de M com posições amostradas aleatoriamente com distribuição uniforme. Então existe uma constante positiva C tal que, se*

$$m \geqslant C\mu_3^4 n(\log n)^2,$$

então o minimizador para o problema (2.3) é único e igual a M com probabilidade no mínimo de $1 - n^{-3}$.

Por fim, apresentamos o Teorema 2.3 que melhora os resultados anteriores.

Teorema 2.3. *[47, Teorema 2] Seja $M \in \mathbb{R}^{n_1 \times n_2}$ uma matriz de posto r satisfazendo `A1` e `A3`. Imponha, sem perda de generalidade, que $n_1 \leqslant n_2$. Suponha que m entradas de M são observadas com posições amostradas aleatoriamente com distribuição uniforme. Se*

$$m \geqslant 32 \max(\mu_0, \mu_2^2) r(n_1 + n_2)\beta \log^2(2n_2),$$

para algum $\beta > 1$, então o minimizador para o problema (2.3) é único e igual a M com probabilidade no mínimo de $1 - 6\log(n_2)(n_1 + n_2)^{2(1-\beta)} - n_2^{2(1-\sqrt{\beta})}$.

2.2.4 Propriedade de Isometria Restrita (RIP)

Outra propriedade utilizada por alguns métodos de completamento de matrizes é a isometria restrita (ou RIP, do inglês Restricted Isometry Property), cuja definição segue.

Definição 2.3 (RIP [48]). *Uma transformação afim $\mathscr{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^d$ satisfaz a propriedade de isometria restrita de ordem r (r-RIP) com constante $\delta_r \in [0, 1)$, se, para todas as matrizes $X \in \mathbb{R}^{m\times n}$ de posto no máximo r, vale*

$$(1 - \delta_r)\|X\|_F^2 \leqslant \|\mathscr{A}(X)\|_2^2 \leqslant (1 + \delta_r)\|X\|_F^2. \tag{2.14}$$

A condição (2.14) geralmente é analisada a partir de uma noção de incoerência mais restritiva que a apresentada na Definição 2.2, a qual chamaremos de μ-incoerência e cuja definição segue.

Definição 2.4 (μ-incoerência [31]). *Uma matriz $X \in \mathbb{R}^{n_1\times n_2}$ com SVD $X = U\Sigma V^\top$ é μ-incoerente se*

$$\max_{i,j} |U_{ij}| \leqslant \sqrt{\frac{\mu}{n_1}}, \quad \max_{i,j} |V_{ij}| \leqslant \sqrt{\frac{\mu}{n_2}}. \tag{2.15}$$

Note que as desigualdades em (2.15) implicam que

$$\frac{n_1}{r} \max_{1\leqslant i\leqslant n_1} \|U_i\|_2^2 \leqslant \mu \quad \text{e} \quad \frac{n_2}{r} \max_{1\leqslant i\leqslant n_2} \|V_i\|_2^2 \leqslant \mu, \tag{2.16}$$

em que U_i e V_i são i-ésimas linhas de U e V, respectivamente. Assim, a condição `A1` vale com $\mu_0 = \mu$. Por outro lado, definindo a matriz E como em (2.8), usando a desigualdade triangular e as desigualdades em (2.15), temos

$$|E_{ij}| = \left|\sum_{k=1}^{r} u_{ik}v_{jk}\right| \leqslant \sum_{k=1}^{r} |u_{ik}v_{jk}| \leqslant \frac{\mu r}{n_1 n_2}. \tag{2.17}$$

Assim, a μ-incoerência pede que as entradas da matriz E sejam limitadas em valor absoluto por um limitante menor que o da Definição 2.2.

Conforme discutido anteriormente, se μ for um número muito pequeno, então as desigualdades em (2.15) garantem que as entradas não-nulas de U e V estão "bem espalhadas" (em outras palavras, as entradas não-nulas de X não estão concentradas em um número pequeno de linhas e/ou colunas). Logo, uma amostra aleatória de entradas de X deve fornecer informação suficiente para a reconstrução da matriz inteira [31].

Em geral, o operador $P_\Omega(\cdot)$ não satisfaz a desigualdade (2.14). Contudo, para matrizes $X \in \mathbb{R}^{n_1\times n_2}$ de posto reduzido e μ-incoerentes, tal operador satisfaz a desigualdade

(quase isometria),

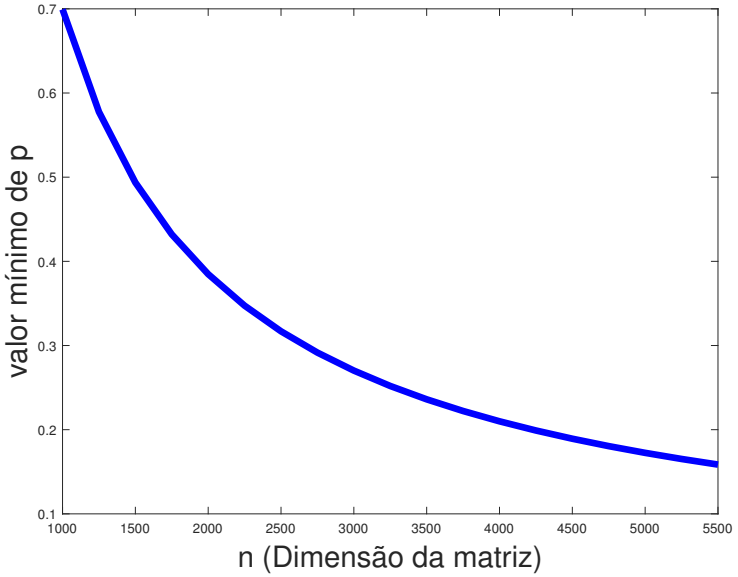
$$(1-\delta)p\|X\|_F^2 \leqslant \|P_\Omega(X)\|_F^2 \leqslant (1+\delta)p\|X\|_F^2, \tag{2.18}$$

com $p := |\Omega|/(n_1 n_2)$, em que $|\Omega|$ é a cardinalidade de Ω. Esse resultado está formalizado no teorema a seguir, adaptado de [31] para matrizes quadradas de ordem n.

Teorema 2.4. *[31, Teorema 4.2] Para $\mu \geqslant 1$, $n \geqslant 3$, existe $C \geqslant 0$ e $\delta \in (0,1)$, tal que se $\Omega \subset \{1,\dots,n\} \times \{1,\dots,n\}$ é escolhido de acordo com o modelo de Bernoulli, com densidade $p \geqslant C\mu^2 r^2 \log n/\delta^2 n$, então P_Ω obedece a desigualdade (2.18) com probabilidade no mínimo de $1-\exp(-n\log n)$, para todas as matrizes $\mu-$incoerentes X de posto no máximo r.*

Fixados os parâmetros μ, r, δ e C, o teorema acima estabelece o valor mínimo de p para que P_Ω satisfaça a desigualdade (2.18) para matrizes $\mu-$incoerentes X de posto no máximo r, com alta probabilidade. A Figura 5, mostra o valor mínimo de p para $C = 0.05, \mu = 3, r = 5$ e $\delta = 1/3$.

Figura 5 – Valor mínimo para p de acordo com o Teorema 2.4.



2.3 Alguns métodos de completamento de matrizes

Nesta seção apresentaremos alguns algoritmos bem estabelecidos na literatura e que serão usados como benchmark nos próximos capítulos.

2.3.1 Fixed Point Continuation (FPC)

O FPC é um algoritmo de primeira ordem que explora o seguinte problema de otimização,

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}} \quad & \|X\|_* \\ \text{sujeito a} \quad & \|\mathscr{A}(X) - b\|_F^2 \leqslant \delta, \end{aligned} \tag{2.19}$$

em que a transformação afim $\mathscr{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^d$ e o vetor $b \in \mathbb{R}^d$ são dados e $\delta \geqslant 0$ é um parâmetro que controla o nível de ruído nas entradas de b.

Observação 2.2. *No problema de completamento $\mathscr{A}(X) = vec(P_\Omega(X))$ e $b = vec(P_\Omega(A))$, em que $vec(B)$ é o vetor resultante ao se empilhar as colunas da matriz B. Assim, o Problema (2.4) pode ser visto como um caso particular do Problema (2.19).*

O problema (2.19) tem a seguinte formulação Lagrangiana

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|\mathscr{A}(X) - b\|_F^2 + \lambda\|X\|_* =: f_\lambda(X), \tag{2.20}$$

em que $\lambda > 0$ é um parâmetro de regularização.

Seja $g(X) = \frac{1}{2}\|\mathscr{A}(X) - b\|_F^2$. Em [37] os autores propõem o processo iterativo

$$\begin{aligned} Y^k &= X^k + t\nabla g(X^k) \\ X^{k+1} &= S_{\lambda t}(Y^k), \end{aligned} \tag{2.21}$$

que converge para uma solução do problema (2.20).

Inspirados em [26], os autores ainda propõem uma técnica de continuação para acelerar a convergência de (2.21), a qual consiste em estabelecer valores decrescentes para o parâmetro de regularização λ da seguinte forma,

$$\lambda_{i+1} = \max(\eta\lambda_i, \lambda_0), \quad i = 1, \ldots, j,$$

em que $\eta \in (0, 1)$ controla a taxa de decrescimento de λ_k. Assim, na k-ésima iteração, o FPC resolve o problema (2.20) para $\lambda = \lambda_k$ fixo. Este processo é descrito no Algoritmo 1 abaixo:

Por fim, apresentamos o teorema que garante a convergência de (2.21). Para isso, considere $\mathscr{A}(X) = A\text{vec}(X)$, em que $\text{vec}(X)$ é o vetor resultante do empilhamento das colunas da matriz X e A é a matriz correspondente à transformação $\mathscr{A}$.

Teorema 2.5. *[37, Teorema 4] A sequência $\{X^k\}$ gerada em (2.21) com $0 < t < 2/\lambda_{\max}(A^\top A)$ converge para uma solução do problema (2.20).*

Algoritmo 1 – Fixed Point Continuation (FPC) [37]

Entrada: $\lambda_0 > 0$, o tamanho de passo t e a tolerância $\varepsilon > 0$.

1. Inicialize X^0.
2. Escolha escalares $\lambda_1 > \lambda_2 > \cdots > \lambda_j = \lambda_0$.
3. **para** $\ell = 1, 2, \ldots, j$ **faça**
4. **para** $k = 0, 1, 2, \ldots$ **faça**
5. $Y^k = X^k + t\nabla g(X^k)$
6. $X^{k+1} = S_{\lambda_\ell t}(Y^k)$
7. **se** o critério de parada for atingido, **termine**.
8. **fim**
9. $X^0 = X^{k+1}$.
10. **fim**

Saída: X^k

2.3.2 Singular Value Thresholding (SVT)

Em [14], baseado no método de Uzawa (veja Seção 2.4 em [14]) para encontrar pontos de sela do Lagrangiano, os autores apresentam um algoritmo, chamado *Singular Value Thresholding* (SVT), e provam que a sequência gerada por

$$\begin{aligned} X^k &= S_\tau(Y^{k-1}) \\ Y^k &= Y^{k-1} + t_k P_\Omega(A - X^k), \end{aligned} \tag{2.22}$$

em que t_k é o tamanho de passo, converge para a única solução do seguinte problema de otimização

$$\begin{aligned} \min_{X \in \mathbb{R}^{m\times n}} \quad & \tau\|X\|_* + \frac{1}{2}\|X\|_F^2 \\ \text{sujeito a} \quad & P_\Omega(X) = P_\Omega(A), \end{aligned} \tag{2.23}$$

onde $\tau > 0$ é um parâmetro de regularização. O primeiro termo $\tau\|X\|_*$ na função objetivo é a relaxação convexa para o posto(X), enquanto que $\frac{1}{2}\|X\|_F^2$ é um termo fortemente convexo que garante que (2.23) tem uma única solução. Devido a sua importância teórica e computacional, o algoritmo SVT é uma importante referência em completamento de matrizes e é frequentemente comparado com outros métodos.

A convergência do Algoritmo 2 é garantida pelo seguinte teorema.

Teorema 2.6. *[14, Teorema 4.2] Suponha que a sequência de tamanho de passos obedeça* $0 < \inf t_k \leqslant \sup t_k < 2$. *Então, a sequência* $\{X^k\}$ *gerada por* (2.22) *converge para a única solução do problema* (2.23).

Algoritmo 2 – Singular Value Thresholding (SVT) [14]

Entrada: Matriz com entradas em falta $A \in \mathbb{R}^{m\times n}$, $\tau > 0$, o tamanho de passo t_k, a tolerância $\varepsilon > 0$.

1. Inicialize $Y^0 = 0$.
2. **para** $k = 1, 2, \ldots$ **faça**
3. $\quad X^k = S_\tau(Y^{k-1})$
4. $\quad Y^k = Y^{k-1} + t_k P_\Omega\left(A - X^k\right)$
5. $\quad$ **se** o critério de parada for atingido, **termine**.
6. **fim**

Saída: X^k

É importante ressaltar que apesar da garantia de convergência do Algoritmo 2 ser restrita ao tamanho de passo $t_k \in (0, 2)$, na prática os autores usam a fórmula

$$t_k = 1.2\frac{mn}{|\Omega|},$$

em que $|\Omega|$ é a cardinalidade do conjunto Ω e m e n são as dimensões da matriz A. Repare que se a quantidade de entradas conhecidas for muito pequena, (como ocorre em muitas situações reais), o valor de t_k pode ser muito maior que 2, resultando na perda de garantia da convergência teórica do método. Por outro lado, para um tamanho de passo $t_k \in (0, 2)$ o Algoritmo 2 apresenta uma redução na velocidade de convergência empírica.

Outro fato interessante é que, apesar da formulação (2.23) ser diferente da relaxação nuclear, é possível mostrar que para τ suficientemente grande, a sequência $\left\{X^k\right\}$ gerada em (2.22) converge para a solução do problema (2.3). O teorema a seguir formaliza estes fatos:

Teorema 2.7. *[14, Teorema 3.1] Sejam X_τ^* a solução do problema (2.23) e X_∞ a solução de norma de Frobenius mínima do problema (2.3). Então,*

$$\lim_{\tau\to\infty} \|X_\tau^* - X_\infty\|_F = 0.$$

2.3.3 Soft-Impute (SI)

Fazendo $\mathscr{A}(X) = P_\Omega(X)$ e $P_\Omega(A) = b$, o problema (2.20) e o processo iterativo em (2.21) se reduzem, respectivamente a

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|P_\Omega(X) - P_\Omega(A)\|_F^2 + \lambda\|X\|_* =: f_\lambda(X), \tag{2.24}$$

e

$$\begin{aligned} Y^k &= X^k + tP_\Omega(A - X^k) \\ X^{k+1} &= S_{\lambda t}(Y^k). \end{aligned} \tag{2.25}$$

O Algoritmo Soft-Impute foi proposto por Mazumder et al [39] e sua iteração é obtida a partir das equações em (2.25) , fixando-se o tamanho de passo $t = 1$. Assim, o processo iterativo do SI é descrito como

$$\begin{aligned} Y^k &= X^k + P_\Omega(A - X^k) = P_\Omega(A) + P_\Omega^\perp(X^k) \\ X^{k+1} &= S_\lambda(Y^k), \end{aligned} \tag{2.26}$$

em que a sequência $\{X^k\}$ converge para uma solução do problema (2.24).

Como ocorre com o FPC, o SI também depende de uma sequência pré-especificada de valores do parâmetro de regularização $\lambda_1 > \cdots > \lambda_j$.

Algoritmo 3 – Soft-Impute (SI) [39]

Entrada: Matriz com entradas em falta $A \in \mathbb{R}^{m\times n}$ e a tolerância $\varepsilon > 0$.

1. Inicialize $X^0 = 0$.
2. Escolha escalares $\lambda_1 > \lambda_2 > \cdots > \lambda_j = \lambda_0$.
3. **para** $\ell = 1, 2, \ldots, j$ **faça**
4. **para** $k = 0, 1, 2, \ldots$ **faça**
5. $Y^k = P_\Omega(A) + P_\Omega^\perp(X^k)$
6. $X^{k+1} = S_{\lambda_\ell t}(Y^k)$
7. **se** $\dfrac{\|X^{k+1} - X^k\|_F^2}{\|X^k\|_F^2} < \epsilon$, **termine**.
8. **fim**
9. $X^0 = X^{k+1}$.
10. **fim**

Saída: X^k

Para analisar a convergência de (2.26), considere a função Q_λ definida da seguinte forma: dada uma matriz $\tilde{X}$ e $\lambda > 0$ fixo

$$Q_\lambda(X, \tilde{X}) = \frac{1}{2}\|P_\Omega(A) + P_\Omega^\perp(\tilde{X}) - X\|_F^2 + \lambda\|X\|_*. \tag{2.27}$$

Agora defina a sequência $\{X_\lambda^k\}$ como

$$X_\lambda^{k+1} := \arg\min_X Q_\lambda(X, X_\lambda^k), \tag{2.28}$$

em que X_λ^0 é um ponto inicial. Tem-se os seguintes teoremas que garantem a convergência de (2.26).

Teorema 2.8. *[39, Teorema 1] Para todo $\lambda > 0$ fixo, a sequência $\{X_\lambda^k\}$ gerada em (2.28) converge para um ponto limite X_λ^∞ que resolve (2.24).*

O teorema a seguir estabelece a taxa de convergência não-assintótica de (2.26).

Teorema 2.9. *[39, Teorema 2] Seja $\lambda > 0$ um número fixo. A sequência $\{X_\lambda^k\}$ gerada em (2.28), no pior caso, tem a seguinte taxa de convergência não-assintótica:*

$$f_\lambda(X_\lambda^k) - f_\lambda(X_\lambda^\infty) \leqslant \frac{2\|X_\lambda^0 - X_\lambda^\infty\|_2^2}{k+1}. \tag{2.29}$$

2.3.4 O Soft-Impute como um método de gradiente proximal

Nesta seção mostraremos que o Soft-Impute é um caso particular do método de gradiente proximal aplicado ao problema (2.24), com um tamanho de passo fixo (esse resultado será usado no próximo capítulo). De fato, repare que o problema de otimização dado pela equação (2.24) é um caso particular do problema (1.29).

Para função $h(X) = \lambda\|X\|_*$, o operador proximal é definido como

$$\text{prox}_{th}(M) = \arg\min_X \left\{ \frac{1}{2t}\|M - X\|_F^2 + \lambda\|X\|_* \right\},$$

cuja solução é dada pelo Teorema 1.10 (com λt no lugar de λ) por

$$\text{prox}_{th}(M) = S_{\lambda t}(M). \tag{2.30}$$

Agora, considere a iteração do Soft-Impute descrita em (2.26). Para o problema (2.24), observe que $g(X) = \frac{1}{2}\|P_\Omega(A) - P_\Omega(X)\|_F^2$, e assim $L = 1$. Como

$$X^k - \nabla g(X^k) = X^k - (P_\Omega(X^k) - P_\Omega(A)) = P_\Omega(A) + P_\Omega^\perp(X^k) =: Y^k,$$

segue de (1.37) e (2.30), com $t = 1$, que $X^{k+1} = \text{prox}_h(Y^k) = S_\lambda\left(Y^k\right)$, o que mostra que de fato (2.26) corresponde a iteração de um método de gradiente proximal aplicado a (2.24).

2.3.5 Fixed-Rank Soft-Impute (FRSI)

O algoritmo Fixed-Rank Soft-Impute (FRSI) [41] tem como motivação o fato de que em alguns problemas o posto da matriz a ser completada é conhecido a priori. O algoritmo também considera a formulação (2.24) e foi inspirado no SI. Mais especificamente, considere a iteração do Soft-Impute dada por

$$X^k = S_\lambda\left(P_\Omega(A) + P_\Omega^\perp(X^{k-1})\right). \tag{2.31}$$

Como foi visto anteriormente, tanto o FPC quanto o SI se utilizam de uma sequência pré-especificada de valores decrescentes do parâmetro de regularização. A proposta do FRSI é baseada em experimentos numéricos e consiste em atualizar o valor de λ em (2.31) através da expressão

$$\lambda = \beta\sigma_{r+1}\left(P_\Omega(A) + P_\Omega^\perp(X^{k-1})\right), \quad \text{com} \quad \beta \in (0,1), \tag{2.32}$$

e usá-lo como o thresholding para o calculo de X^{k+1}. Como em geral a sequência gerada por (2.32) é decrescente (pelo menos empiricamente), a partir de um certo número de iterações, segue que cada iterada X^{k+1} tem posto aproximadamente igual a r.

Algoritmo 4 – Fixed-Rank Soft-Impute (FRSI)

Entrada: Matriz com entradas em falta $A \in \mathbb{R}^{m\times n}$, o posto de A, a tolerância ϵ, $\beta \in (0,1)$, $\lambda > 0$ e o número máximo de iterações $k_{\max}$.

1. Inicialize $X^0 = 0$.
2. **para** $k = 1, 2, \ldots, k_{\max}$ **faça**
3. $\quad X^k = S_\lambda\left(P_\Omega(A) + P_\Omega^\perp(X^{k-1})\right)$.
4. $\quad \lambda = \beta\sigma_{r+1}\left(P_\Omega(A) + P_\Omega^\perp(X^{k-1})\right)$.
5. $\quad$ **se** o critério de parada for atingido, **termine**.
6. **fim**

Saída: X^k.

Esta nova forma de calcular o valor do parâmetro de regularização garantiu bons resultados numéricos ao FRSI quando comparado com os outros métodos até aqui apresentados. Repare que diferentemente do SI, que resolve o problema (2.24) para cada λ_k fixo até a convergência das iterações internas, o Algoritmo 4 resolve o problema (2.24) com $\lambda = \lambda_k$ apenas uma vez, para cada k. O método proposto no Algoritmo 4 ainda não possui prova de convergência e, portanto, é considerado apenas uma heurística.

Encerramos este capítulo com alguns problemas que podem ser analisados sob a ótica de completamento de matrizes.

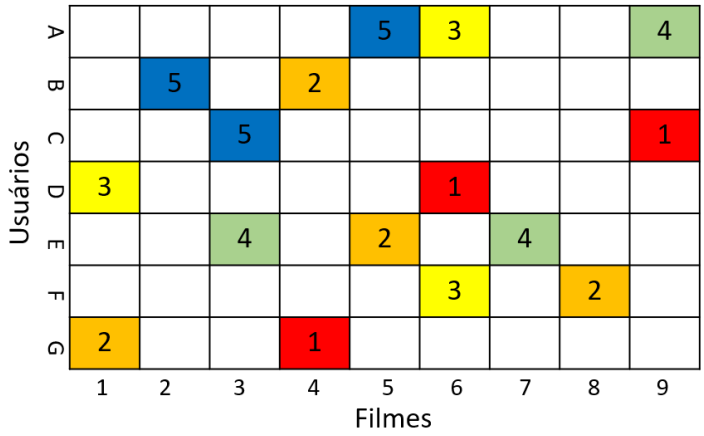
2.4 Aplicações do problema de completamento

Existem muitas situações práticas nas quais a matriz associada possui entradas em falta e, felizmente, em algumas dessas situações tal matriz possui posto baixo, o que torna possível a recuperação das entradas em falta através de técnicas de completamento de matrizes. A seguir, apresentaremos alguns cenários em que isto ocorre.

2.4.1 Sistemas de recomendação

Um sistema de recomendação usa o histórico de informações fornecidas por diversos usuários e que estão armazenadas em um sistema para gerar uma lista de opções das quais o usuário possa se interessar [33]. Como exemplo, podemos citar os bancos de dados da Netflix [8, 7] e o MovieLens [40], que são frequentemente usados em experimentos de completamento de matrizes. Neles, cada usuário avalia diversos filmes, atribuindo notas variando no conjunto $\{1, 2, 3, 4, 5\}$, que são armazenadas em uma matriz de avaliações do tipo usuário-item, como na Figura 6. Podemos ver que cada linha representa um usuário e cada coluna representa um filme. Em geral, há centenas ou milhares de filmes e o usuário avalia alguns poucos filmes (aqueles que já assistiu), resultando em uma matriz com pouquíssimas entradas observadas (coloridas). A tarefa consiste em estimar as entradas em falta (em branco).

Figura 6 – Matriz de avaliações de um sistema de recomendação



Uma hipótese importante aqui é assumir que essa matriz tem posto reduzido. A justificativa para essa hipótese é que se acredita que há poucos fatores que controlam as preferências dos usuários, como por exemplo, o gênero do filme e os personagens que estão atuando.

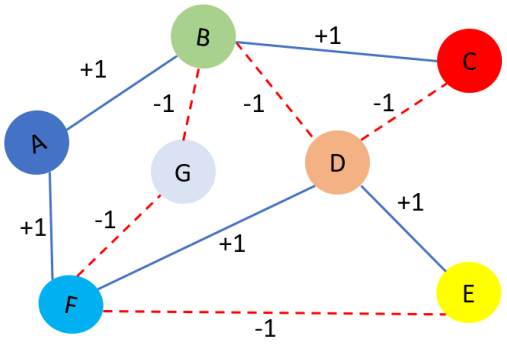
2.4.2 O problema de predição de link

Outro tópico que pode ser visto sob a ótica do completamento de matrizes é o problema de predição de link [59, 9]. Em redes sociais como Facebook, Instagram ou Twitter, um usuário pode seguir ou não seguir (bloquear) outros usuários. A Figura 7a ilustra esta interação. Nela, $+1$ indica que dois usuários estão seguindo um ao outro, enquanto -1 indica eles não estão se seguindo. Assim como no problema anterior, essas informações podem ser apresentadas em uma matriz de adjacência, ilustrada na Figura 7b. Como o número de interações de cada usuário é muito limitado, essa matriz possui várias entradas em falta. O objetivo da predição de link é identificar, com base nas conexões

observadas (conhecidas), se dois usuários, que a princípio não têm conexão entre si (por exemplo, os usuários C e E), irão formar um link ou não no futuro [35]. Como as pessoas tendem a formar grupos, é comum assumir que a matriz de adjacência possui posto reduzido [17, 59].

Figura 7 – Ilustração do problema de predição de link

(a) Interação na rede social



(b) Matriz de adjacência

Usuários	A	B	C	D	E	F	G
A		+1				+1	
B	+1		+1	-1			-1
C		+1		-1			
D		-1	-1		+1	+1	
E				+1		-1	
F	+1			+1	-1		-1
G		-1				-1	

2.4.3 Modelagem com Matrizes de Distâncias Euclidianas

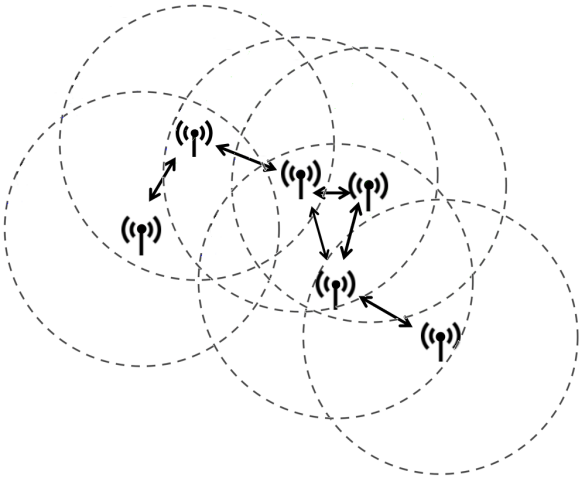
Uma matriz de distâncias $D = (d_{ij}^2) \in \mathbb{R}^{n\times n}$ é dita Euclidiana se existem $x_i \in \mathbb{R}^K$, $i = 1, \ldots, n > K$, tais que $D_{ij} = \|x_i - x_j\|^2$.

Um resultado importante acerca dessa classe de matrizes é que o posto de uma EDM é no máximo $K + 2$ [20], em que K é a dimensão do espaço ao qual os pontos pertencem. Repare que este resultado nos diz que o posto de uma EDM independe da quantidade de pontos. Por exemplo, se tomarmos 1.000 pontos no espaço $\mathbb{R}^3$, então teremos uma EDM de dimensão 1.000×1.000 cujo posto é no máximo 5. Assim, o posto de uma EDM é reduzido e conhecido a priori.

Há problemas práticos que podem ser modelados por meio de Matrizes de Distâncias Euclidianas (EDM). A título de exemplificação, podemos citar o problema de localização em rede de sensores [45], no qual se assume que cada sensor tem um raio de comunicação limitado e, portanto, um sensor qualquer só conhece as posições de outros sensores que estão dentro do seu raio de comunicação. Nesse caso, a tarefa é determinar as posições de todos os sensores na rede, a partir do conhecimento de uma amostra de posições de alguns sensores e informações de conectividade entres os sensores próximos [43]. A Figura 8 ilustra o problema. As setas indicam as distâncias conhecidas e as circunferências representam o raio de comunicação de cada sensor. Outras aplicações práticas podem ser consultadas com mais detalhes em [20, 27, 53].

Neste capítulo, introduzimos o Problema de Completamento e exploramos algumas de suas propriedades intrínsecas. Também apresentamos algumas aplicações

Figura 8 – Ilustração do problema de localização em rede de sensores



práticas do problema, bem como, alguns algoritmos de completamento, entre os quais o FRSI. No próximo capítulo, propomos um algoritmo de duas fases inspirado no FRSI e que considera a versão acelerada do Soft-Impute, apresentada na Seção 2.3.3.

3 Um algoritmo de duas fases para completamento de matrizes de posto conhecido

No capítulo anterior, vimos algumas aplicações nas quais o posto da matriz a ser completada é conhecido a priori. Vimos, por exemplo, que o problema de localização de sensores em redes pode ser modelado por meio de matrizes de distâncias Euclidianas, que são matrizes de posto reduzido e conhecido a priori.

Neste capítulo, inspirados no FRSI, apresentamos uma heurística baseada na informação do posto (o qual supomos ser conhecido a priori) e estabelecemos uma condição sob a qual a sequência gerada é quasi-Fejér convergente [19] ao conjunto solução do problema.

Em seguida, incluímos um mecanismo de aceleração semelhante à aceleração de Nesterov [6] e obtemos uma nova heurística. Embora a convergência dessa nova heurística não possa ser garantida em geral, ela pode ser muito útil como uma fase de "warm-start" (Fase 1), fornecendo uma estimativa adequada para o parâmetro de regularização do problema (2.24) e um bom ponto de partida para um algoritmo de gradiente proximal acelerado (Fase 2).

Finalmente, apresentamos alguns resultados numéricos com dados sintéticos e reais e comparamos com alguns algoritmos para completamento de matrizes estudados no capítulo anterior.

Parte dos resultados deste capítulo foi publicada no artigo [2].

3.1 Revisitando o Fixed-Rank Soft-Impute (FRSI)

Seja $A \in \mathbb{R}^{m\times n}$ uma matriz com entradas em falta e posto r, o qual assumimos ser conhecido a priori. A iteração do FRSI [41], com a atualização de λ dada pela equação (2.32), pode ser escrita como

$$\begin{aligned} \lambda &= \beta\sigma_{r+1}\left(P_\Omega(A) + P_\Omega^\perp(X^{k-1})\right) \\ X^{k+1} &= S_\lambda\left(P_\Omega(A) + P_\Omega^\perp(X^k)\right). \end{aligned} \tag{3.1}$$

Note que na expressão (3.1) não é possível garantir que X^{k+1} terá posto r ao longo das iterações. De fato, nos estudos feitos em [41], os melhores resultados do processo iterativo (3.1) são obtidos quando $\beta = 0,85$. Para este valor de β, observa-se que os valores de λ são decrescentes ao longo das iterações, de modo que em geral se tem

$\sigma_{r+1}\left(P_\Omega(A) + P_\Omega^\perp(X^k)\right)$ diferente do valor de λ e pela definição de S_λ a matriz X^{k+1} terá um posto $\hat{r}$ diferente de r (em geral $\hat{r}$ é maior que r).

Uma forma de garantir que X^{k+1} tenha posto no máximo r e ainda manter os bons resultados da iteração (3.1) é considerar a seguinte expressão para λ

$$\lambda = \sigma_{r+1}\left(P_\Omega(A) + P_\Omega^\perp(X^k)\right) \tag{3.2}$$

isto é, λ é igual ao $(r+1)$-ésimo maior valor singular de $P_\Omega(A) + P_\Omega^\perp(X^k)$. Assim, segue da definição de S_λ em (1.24) que cada iterada X^{k+1} terá posto no máximo r.

Deste modo, ao invés de (3.1), iremos considerar o seguinte processo iterativo:

$$\begin{aligned} \lambda &= \sigma_{r+1}\left(P_\Omega(A) + P_\Omega^\perp(X^k)\right) \\ X^{k+1} &= S_\lambda\left(P_\Omega(A) + P_\Omega^\perp(X^k)\right). \end{aligned} \tag{3.3}$$

Definindo $g(X) = \dfrac{1}{2}\|P_\Omega(X - A)\|_F^2$, o processo iterativo (3.3) pode ser escrito como

$$X^{k+1} = S_{\sigma_{r+1}(X^k - \nabla g(X^k))}\left(X^k - \nabla g(X^k)\right). \tag{3.4}$$

A fim de estudar o comportamento da sequência gerada por (3.4), vamos considerar o operador

$$T(X) := S_{\sigma_{r+1}(X - \nabla g(X))}\left(X - \nabla g(X)\right).$$

Proposição 3.1. *Seja $T : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ o operador definido acima e assuma que posto$(A) = r$. Então, as seguintes propriedades valem:*

(i) $T(A) = A$.

(ii) *Se $B \in \mathbb{R}^{m\times n}$ é tal que posto$(B) \leqslant r$ e $P_\Omega(B) = P_\Omega(A)$, então $T(B) = B$.*

(iii) $T(P_\Omega^\perp(A)) = A$.

(iv) $T(0) = T(P_\Omega(A))$.

Demonstração. Observe que $\nabla g(X) = P_\Omega(X - A)$. (i) Assim, como $\nabla g(A) = 0$, e posto$(A) = r$, segue da definição do operador soft-thresholding e do fato de $\sigma_{r+1}(A) = 0$, que $T(A) = A$, isto é, A é um ponto fixo de T. O mesmo raciocínio se aplica à matriz B tal que $P_\Omega(B) = P_\Omega(A)$ e posto$(B) \leqslant r$, provando (ii).

Note também que

$$P_\Omega^\perp(A) - \nabla g(P_\Omega^\perp(A)) = P_\Omega^\perp(A) - P_\Omega\left(P_\Omega^\perp(A) - A\right) = P_\Omega^\perp(A) + P_\Omega(A) = A,$$

e assim $T(P_\Omega^\perp(A)) = A$, mostrando (iii). Finalmente, uma vez que

$$P_\Omega(A) - \nabla g(P_\Omega(A)) = P_\Omega(A) - P_\Omega\left(P_\Omega(A) - A\right) = P_\Omega(A) = 0 - P_\Omega\left(0 - A\right) = 0 - \nabla g(0),$$

concluímos (iv): $T(0) = T(P_\Omega(A))$. □

Portanto, não apenas a matriz alvo A é um ponto fixo de T, mas qualquer outra matriz B, de posto no máximo r, tal que $P_\Omega(B) = P_\Omega(A)$ também o é. Talvez mais surpreendente é o fato de que T também admite pontos fixos X tais que $P_\Omega(X) \neq P_\Omega(A)$, como mostrado na próxima proposição.

Proposição 3.2. *Seja $X \in \mathbb{R}^{m\times n}$ uma matriz de posto no máximo r, com $X = U\Sigma V^\top$ sua SVD truncada no posto $r+1$. Se $\nabla g(X) = -\gamma UV^\top - U_\perp \Sigma_\perp V_\perp^\top$, em que as colunas de $U_\perp$ e $V_\perp$ são bases ortonormais para o complemento ortogonal da imagem de U e V, respectivamente, e $\gamma > 0$ com $\sigma_i^\perp < \gamma$, para $i = r+2, \ldots, \min\{m,n\}$, então $X = T(X)$.*

Demonstração. Observe que

$$X - \nabla g(X) = U(\Sigma + \gamma \mathrm{I}_{r+1})V^\top + U_\perp \Sigma_\perp V_\perp^\top,$$

em que I_{r+1} é a matriz identidade de ordem $r+1$. Então, como $\sigma_{r+1}(X - \nabla g(X)) = \gamma > \sigma_i^\perp$, segue que $T(X) = U\Sigma V^\top = X$. □

Das proposições acima, vemos que, embora a matriz alvo A seja um ponto fixo de T, o que é desejável, o operador T geralmente não possui um único ponto fixo e, mais ainda, podem existir pontos fixos X tais que $P_\Omega(X) \neq P_\Omega(A)$.

No entanto, mostraremos que, se a sequência dos valores singulares $\sigma_{r+1}(X^k - \nabla g(X^k))$ vai a zero suficientemente rápido, então podemos provar que a sequência $\{X^k\}$ é quasi-Fejér convergente para o conjunto

$$\mathcal{X}^* = \{X \in \mathbb{R}^{m\times n} \mid \mathrm{posto}(X) \leqslant r, P_\Omega(X) = P_\Omega(A)\}.$$

Definição 3.1. *[19, Definição 1.1] Uma sequência $\{X^k\}$ em $\mathbb{R}^{m\times n}$ é quasi-Fejér convergente para $C \subset \mathbb{R}^{m\times n}$ se, para cada $X^\star \in C$, existe uma sequência somável não-negativa $\{\varepsilon_k\}$ tal que*

$$\|X^k - X^\star\|_F \leqslant \|X^{k-1} - X^\star\|_F + \varepsilon_k, \quad k = 1, 2, \ldots$$

Proposição 3.3. *[52, Proposição 2.1] Seja $C \subset \mathbb{R}^{m\times n}$ um conjunto não vazio e $\{X^k\}$ uma sequência quasi-Fejér convergente para C. Então,*

1. *$\{X^k\}$ é limitada.*
2. *Se $\{X^k\}$ tem um ponto de acumulação $\bar{X} \in C$, então a sequência $\{X^k\}$ converge para $\bar{X}$.*

Teorema 3.1. *Seja $\{X^k\}$ uma sequência gerada por $X^k = T(X^{k-1})$, com $X^0 \in \mathbb{R}^{m\times n}$. Se a sequência $\{\sigma_{r+1}(X^k - \nabla g(X^k))\}$ é somável, então $\{X^k\}$ é uma sequência quasi-Fejér convergente para o conjunto $\mathcal{X}^*$.*

Demonstração. Seja $X^\star \in \mathcal{X}^*$. Considere a notação $\sigma_{r+1}^k = \sigma_{r+1}(X^k - \nabla g(X^k))$ e $\sigma_i^\star = \sigma_i(X^\star)$. Então,

$$\begin{aligned}
\|X^{k+1} - X^\star\|_F &= \|T(X^k) - T(X^\star)\|_F \\
&= \|S_{\sigma_{r+1}(X^k-\nabla g(X^k))}(X^k - \nabla g(X^k)) - S_{\sigma_{r+1}(X^\star)}(X^\star)\|_F \\
&\leqslant \|S_{\sigma_{r+1}(X^k-\nabla g(X^k))}(X^k - \nabla g(X^k)) - S_{\sigma_{r+1}(X^k-\nabla g(X^k))}(X^\star)\|_F \\
&\quad + \|S_{\sigma_{r+1}(X^k-\nabla g(X^k))}(X^\star) - S_{\sigma_{r+1}(X^\star)}(X^\star)\|_F \\
&\leqslant \|X^k - \nabla g(X^k) - X^\star\|_F + \left(\sum_{i=1}^{r}((\sigma_i^\star - \sigma_{r+1}^k)_+ - \sigma_i^\star)^2\right)^{1/2} \\
&\leqslant \|P_\Omega^\perp(X^k - X^\star)\|_F + \sqrt{r}\sigma_{r+1}^k \leqslant \|X^k - X^\star\|_F + \sqrt{r}\sigma_{r+1}^k, \qquad (3.5)
\end{aligned}$$

em que usamos o fato de $X^\star$ ser um ponto-fixo de T, a desigualdade triangular e a não-expansividade do operador $S_\lambda(\cdot)$ (Lema 1.3), com $\lambda = \sigma_{r+1}(X^k - \nabla g(X^k))$ fixo. Consequentemente, se a sequência $\{\sigma_{r+1}^k\}$ é somável, então a sequência $\{X^k\}$ é quasi-Fejér convergente para $\mathcal{X}^*$. □

Portanto, desde que

$$\sum_{k=0}^{\infty} \sigma_{r+1}(X^k - \nabla g(X^k)) < \infty, \qquad (3.6)$$

a sequência $\{X^k\}$ gerada por (3.4) será quasi-Fejér convergente para $\mathcal{X}^*$ e, de acordo com a Proposição 3.3, se ela tem um ponto de acumulação neste conjunto, então toda a sequência irá convergir para ele. Infelizmente, a condição (3.6) é admitidamente forte e não vale em geral (falha para uma porcentagem de entradas conhecidas muito baixa e matrizes de posto mais alto, por exemplo, $n = m = 1000, r = 50$ e porcentagem de apagamento de 90%). Por esta razão, a iteração (3.4), deve ser tratada como uma heurística.

3.2 Um algoritmo de duas fases baseado na informação do posto

Embora o processo iterativo em (3.4) possa não convergir para uma matriz em $\mathcal{X}^*$, aqui nós propomos usá-lo por um número fixo de iterações, como uma fase de "warm-start" para obtermos um bom ponto inicial e ao mesmo tempo estimar o parâmetro de regularização λ do problema (2.24) antes de usarmos um método de gradiente proximal para resolver (2.24).

Isso é motivado por nossa experiência numérica com o processo iterativo (3.4). Observamos que quando $\{X^k\}$ não converge para um elemento em $\mathcal{X}^*$, ele normalmente

converge para um $\tilde{X}$ como na Proposição 3.2 que, embora não satisfaça $P_\Omega(\tilde{X}) = P_\Omega(A)$, é tal que $\|\tilde{X}\|_* < \|A\|_*$, sugerindo $\tilde{X}$ como um minimizador de $\frac{1}{2}\|P_\Omega(X) - P_\Omega(A)\|_F^2 + \lambda\|X\|_*$ para um valor apropriado de $\lambda > 0$.

Primeiro, inspirados na versão acelerada do método do gradiente proximal vista no Capítulo 1, nós incluímos uma aceleração para a heurística (3.4), resultando no Algoritmo 5. Esta fase de warm-start será chamada de *Fase 1*.

Algoritmo 5 – Fase 1: Warm-Start

Entrada: Entradas conhecidas de $A \in \mathbb{R}^{m\times n}$ indexadas por Ω, o posto r, $\varepsilon > 0$, $w \in \mathbb{N}$, e $\beta > 0$.

1. Inicialize $X^0 = 0$, $Z^1 = 0$ e $\rho_0 = \infty$
2. **para** $j = 1, 2, \ldots, w$ **faça**:
3. Compute a $(r+1)$-SVD truncada de $P_\Omega(A) + P_\Omega^\perp(Z^j)$
4. Defina $\rho_j = \sigma_{r+1}(P_\Omega(A) + P_\Omega^\perp(Z^j))$
5. **se** $|\rho_j - \rho_{j-1}|/(1+\rho_{j-1}) < \varepsilon$ **então** saia.
6. Compute $X^j \leftarrow S_{\rho_j}\left(P_\Omega(A) + P_\Omega^\perp(Z^j)\right)$
7. $Z^{j+1} \leftarrow X^j + \frac{j-1}{j+\beta}\left(X^j - X^{j-1}\right)$
8. **fim**

Saída: X^j, ρ_j.

A primeira fase roda por um número pré-especificado w de iterações ou até o critério de parada ser atingido. O último valor de ρ_j da Fase 1 é usado como o parâmetro de regularização λ para a segunda fase, que consiste de um método de gradiente proximal acelerado para o problema (2.24), iniciando de X^{j+1}. A *Fase 2* é descrita no Algoritmo 6.

Algoritmo 6 – Fase 2: Soft-Impute Acelerado

Entrada: Entradas conhecidas de $A \in \mathbb{R}^{m\times n}$ indexadas por Ω, o posto r, $\varepsilon > 0$, $it_{\max} \in \mathbb{N}$, $\lambda > 0$ e $X^0 \in \mathbb{R}^{m\times n}$.

1. Inicialize $Z^1 = X^0$
2. **para** $k = 1, \ldots, it_{\max}$ **faça**
3. Compute $X^k \leftarrow S_\lambda\left(P_\Omega(A) + P_\Omega^\perp(Z^k)\right)$
4. **se** algum critério de parada é verificado **pare**
5. $Z^{k+1} \leftarrow X^k + \frac{k-1}{k+2}\left(X^k - X^{k-1}\right)$
6. **fim**

Saída: X^k

Observação 3.1. *Diferentemente da Fase 1, na qual uma SVD truncada no posto* $(r+1)$ *foi suficiente para calcular o operador soft-thresholding (porque o valor limiar era exatamente o* $(r+1)$*-ésimo maior singular de* $P_\Omega(A) + P_\Omega^\perp(Z^k)$*), na Fase 2, o valor de* λ *é fixo e pode ser diferente de* $\sigma_{r+1}(P_\Omega(A) + P_\Omega^\perp(Z^k))$*. Como resultado, nós precisamos manter uma estimativa do posto* r_k*, que é atualizada em cada iteração (começando com* $r_1 = r$*). Para isso, computamos uma SVD truncada com os* $(r_k + 1)$ *maiores valores singulares de* $P_\Omega(A) + P_\Omega^\perp(Z^k)$*. Se o* $(r_k + 1)$*-ésimo valor singular já está abaixo do valor limiar* λ*, nós mantemos a estimativa do posto* r_k*. Caso contrário, aumentamos* r_k *(para* $r_k + 5$*, como fizemos em nossos experimentos numéricos, por exemplo) e repetimos a SVD truncada. Finalmente,* r_{k+1} *é definido para o número de valores singulares positivos deslocados após a última SVD truncada (note que, neste caso, o posto recuperado pode ser maior que o posto alvo). Um esquema semelhante foi usado em [14].*

Algoritmo 7 – Algoritmo de Duas Fases baseado no posto

Entrada: Entradas conhecidas de $A \in \mathbb{R}^{m\times n}$ indexadas por Ω, o posto r, $\varepsilon > 0$, $w, it_{\max} \in \mathbb{N}$, e $\beta > 0$.

1. Chame o Algoritmo 5 fornecendo A, Ω, r, $\varepsilon > 0$, w e $\beta > 0$ ▷ *Fase 1*
2. Defina $\lambda = \rho_j$, $X^0 = X^j$
3. Chame o Algoritmo 6 fornecendo A, Ω, r, $\varepsilon > 0$, $it_{\max}$, λ e X^0 ▷ *Fase 2*

Saída: X^k

O Algoritmo 7 resume o algoritmo de duas fases baseado na informação do posto, o qual usa o Algoritmo 5 como uma fase de warm-start (Fase 1) e, então, chama um Algoritmo de Gradiente Proximal Acelerado (Algoritmo 6) na segunda fase. Como será visto nos experimentos numéricos da Seção 3.3, o Algoritmo 7 não apenas supera o Fixed-Rank Soft-Impute[41], como também é competitivo com outros algoritmos para completamento de matrizes de posto reduzido que revisamos no Capítulo 2.

Além disso, o Algoritmo 7 tem garantia de convergência para uma solução de

$$\min_{X\in\mathbb{R}^{m\times n}} \quad \frac{1}{2}\|P_\Omega(X) - P_\Omega(A)\|_F^2 + \rho_j\|X\|_* \tag{3.7}$$

(em que ρ_j é a saída da Fase 1), porque a Fase 1 roda por um número finito de iterações e a Fase 2 é um método de gradiente proximal acelerado aplicado ao problema (3.7). Vale ressaltar que embora os iterados X^j da Fase 1 tenham sempre o posto alvo r, não temos esta garantia para os iterados da Fase 2, uma vez que neste o valor de λ é fixo.

3.2.1 Complexidade do algoritmo proposto

Cada fase do algoritmo proposto requer o cálculo de uma decomposição em valores singulares a cada iteração. O cálculo da SVD de uma matriz $X \in \mathbb{R}^{m\times n}$ requer $O(mn^2)$

flops e isso torna alguns métodos baseados em SVD computacionalmente inadequados para problemas de grande porte.

Uma vantagem importante do método proposto é que em cada iteração a matriz da qual será obtida a SVD possui a estrutura especial "esparsa + posto baixo" e podemos usar o pacote PROPACK [34] (mais especificamente, uma rotina `lansvd` que implementa uma variante do algoritmo de Lanczos, projetada para grandes matrizes com estrutura esparsa + posto baixo) para calcular apenas os principais valores/vetores singulares da matriz.

Faremos a análise da complexidade da Fase 1 do algoritmo proposto. Primeiro, note que em cada iteração dessa fase é necessário obter uma SVD truncada no posto $r+1$ da matriz $Y^j := P_\Omega(A) + P_\Omega^\perp(Z^j)$ (veja o passo 3). Mostraremos que essa matriz possui estrutura esparsa + posto baixo. Com efeito, fazendo $X^0 = X^{-1} = 0$ e $Z^j = (1+c_j)X^{j-1} - c_j X^{j-2}$, a matriz Y^j pode ser escrita como

$$\begin{aligned} Y^j &= P_\Omega(A) + P_\Omega^\perp(Z^j) \\ &= P_\Omega(A - Z^j) + Z^j \\ &= P_\Omega(A - Z^j) + (1+c_j)X^{j-1} - c_j X^{j-2}, \end{aligned} \tag{3.8}$$

em que $c_j = \dfrac{j-2}{j+\beta-1}$ e a equação (3.8) segue da atualização de Z^{j+1} no passo 7.

Segue da definição de P_Ω que $P_\Omega(A - Z^j)$ é uma matriz esparsa e, como as matrizes X^{j-1} e X^{j-2} têm postos menores ou iguais a $r \ll \min(m,n)$, segue que a matriz $(1+c_j)X^{j-1} - c_j X^{j-2}$ tem posto menor ou igual a $2r$, o que ainda é considerado um posto baixo, quando comparado com as dimensões da matriz.

Agora, assuma que X^{j-1} e X^{j-2} tenham postos r_{j-1} e r_{j-2} e suas SVDs sejam dadas por $U_{j-1}\Sigma_{j-1}V_{j-1}^\top$ e $U_{j-2}\Sigma_{j-2}V_{j-2}^\top$, respectivamente. Um passo importante para o cálculo da SVD, por meio da rotina `lansvd`, são multiplicações da forma $Y^j v$ e $(Y^j)^\top u$. Para reduzir custo computacional dessas operações, o que se faz é aproveitar a estrutura das matrizes envolvidas, por meio das fórmulas

$$Y^j v = P_\Omega(A - Z^j)v + (1+c_j)U_{j-1}\Sigma_{j-1}(V_{j-1}v) - c_j U_{j-2}\Sigma_{j-2}(V_{j-2}^\top v), \tag{3.9}$$

$$(Y^j)^\top u = P_\Omega(A - Z^j)^\top u + (1+c_j)V_{j-1}\Sigma_{j-1}^\top(U_{j-1}^\top u) - c_j V_{j-2}\Sigma_{j-2}^\top(U_{j-2}^\top u). \tag{3.10}$$

Para computar $Y^j v$ em (3.9), precisamos construir $P_\Omega(A - Z^j)$ e calcular produto $P_\Omega(A-Z^j)v$ e ambos resultam em $O(|\Omega|)$ flops. Por outro lado, $(1+c_j)U_{j-1}\Sigma_{j-1}(V_{j-1}v)$ e $c_j U_{j-2}\Sigma_{j-2}(V_{j-2}^\top v)$ necessitam de $O((m+n)(r_{j-1}+r_{j-2}))$ operações. O mesmo vale para a equação (3.10). Assim, para obter cada valor/vetor singular é necessário um total de $O(|\Omega|) + O((m+n)(r_{j-1}+r_{j-2}))$ flops. Como em cada iteração da Fase 1 são computados exatamente $r+1$ valores/vetores singulares, segue que o número total de operações para se obter uma SVD truncada no posto $r+1$ é dado por

$$O(|\Omega|(r+1)) + O((m+n)(r_{j-1}+r_{j-2})(r+1)) \tag{3.11}$$

No Algoritmo 6 (Fase 2) a estimativa do número de operações pode ser maior que em (3.11), em função do exposto na Observação 3.1.

Na Tabela 1 comparamos as complexidades para uma iteração de cada um dos algoritmos apresentados no capítulo anterior[2]. Relembre que os algoritmos SI, SVT e FPC não necessitam da informação do posto, então $\tilde{r}$ é uma estimativa para o posto da matriz resultante em cada iteração. Para o FPC estamos considerando $\mathscr{A} = P_\Omega$ e isso não destrói a estrutura esparsa + posto baixo da matriz. Já no caso do SVT, a matriz resultante em cada iteração tem a estrutura esparsa + esparsa e a SVD é obtida de uma matriz de mesma esparsidade que $P_\Omega(A)$, com complexidade $O(|\Omega|\tilde{r})$.

Tabela 1 – Comparação da complexidade de uma iteração do algoritmo proposto com os algoritmos SI, SVT e FPC.

Algoritmo	Complexidade de uma iteração
Alg. 5	$O(\|\Omega\|(r+1)) + O((m+n)(r_{j-1}+r_{j-2})(r+1))$
FRSI	$O(\|\Omega\|(r+1)) + O((m+n)r(r+1))$
Soft-Impute	$O(\|\Omega\|\tilde{r}) + O((m+n)\tilde{r}^2)$
SVT	$O(\|\Omega\|\tilde{r})$
FPC	$O(\|\Omega\|\tilde{r}) + O((m+n)\tilde{r}^2)$

Podemos observar que o SVT possui a menor complexidade por iteração. Isto se deve ao fato da matriz resultante em cada iteração ter a estrutura esparsa + esparsa. Por outro lado, o Soft-Impute e o FPC possuem a mesma complexidade por iteração, enquanto que o FRSI e o Algoritmo 5 são os mais caros computacionalmente. Vale ressaltar que, a estimativa do número de operações por iteração da Fase 2 (Algoritmo 6) pode ser um pouco maior que o número de operações da Fase 1 (nos casos em que o posto recuperado for maior que o posto alvo) . No entanto, a Fase 2 roda um número pequeno de iterações e o custo computacional total do algoritmo proposto é mais do que compensado pela aceleração na taxa de convergência, como será visto nos experimentos numéricos.

3.3 Resultados numéricos

Nesta seção, realizamos experimentos de completamento de matrizes com dados sintéticos e com o banco de dados *MovieLens*[3]. Além disso, fornecemos um estudo empírico para a escolha do parâmetro de aceleração β na Fase 1 (Algoritmo 5).

[2] Não inclui o número de operações internas da rotina `lansvd`, o qual supomos ser aproximadamente o mesmo para todos os algoritmos.

[3] Um banco de dados que é largamente utilizado em experimentos de completamento de matrizes e está disponível em https://grouplens.org/datasets/movielens/.

Todos os algoritmos foram implementados em linguagem Matlab e todos os resultados numéricos foram realizados em um PC com processador Intel Core i7-7500U e 16 GB RAM.

O Algoritmo 7 proposto é comparado com os métodos mencionados na Seção 2.3: FRSI, SVT e FPC. Todos estes métodos usam o pacote PROPACK para calcular apenas um número pré-determinado de valores/vetores singulares da matriz.

Como critério de parada para o Algoritmo 6, nós definimos

$$\min\left\{\frac{|f_\lambda(X^k)-f_\lambda(X^{k+1})|}{f_\lambda(X^k)}, \frac{\|X^{k+1}-X^k\|_F}{\|X^k\|_F}\right\} \leqslant \varepsilon_\lambda,$$

para uma tolerância dada $\varepsilon_\lambda > 0$, em que f_λ é a função objetivo do problema (2.24). Para o algoritmo FRSI usamos

$$\min\left\{\frac{\|P_\Omega\left(X^k-A\right)\|_F}{\|P_\Omega\left(A\right)\|_F}, \frac{\|X^{k+1}-X^k\|_F}{\|X^k\|_F}\right\} \leqslant \varepsilon_1,$$

como critério de parada.

Para algoritmo SVT nós seguimos a recomendação em [14] e usamos

$$\frac{\|P_\Omega\left(X^k-A\right)\|_F}{\|P_\Omega\left(A\right)\|_F} \leqslant \varepsilon_2.$$

Por outro lado, para o FPC foi usado o critério de parada

$$\frac{\|X^{k+1}-X^k\|_F}{\max\left\{1, \|X^k\|_F\right\}} \leqslant \varepsilon_3,$$

conforme recomendado em [37].

O procedimento a seguir foi usado para gerar o conjunto de dados sintéticos: nós geramos matrizes $n \times n$ de posto $r \ll n$ da forma $A = MN \in \mathbb{R}^{n\times n}$, em que as entradas de $M \in \mathbb{R}^{n\times r}$ e $N \in \mathbb{R}^{r\times n}$ são amostradas independentes e identicamente distribuídas da distribuição normal padrão. Então, deletamos, aleatoriamente, uma porcentagem p_- de entradas (entradas desconhecidas) de A.

Antes de apresentarmos alguns resultados numéricos para dados sintéticos e o MovieLens, daremos uma visão geral de como definir o parâmetro β no Algoritmo 5.

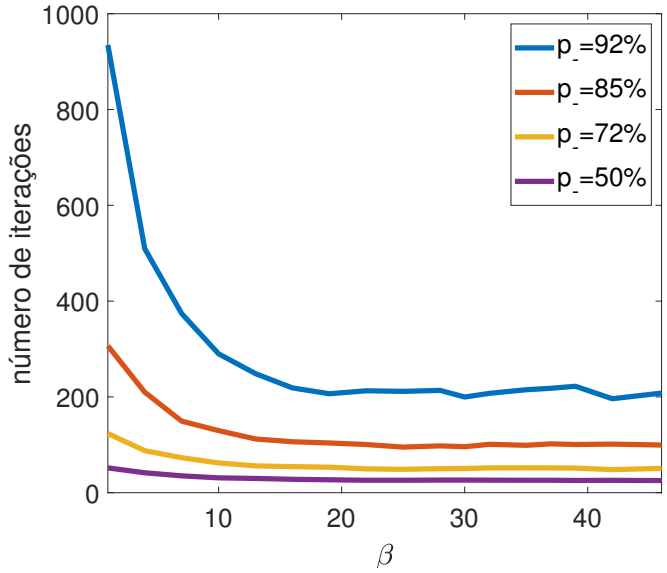
3.3.1 Ajustando o parâmetro β

Para avaliar a sensibilidade do Algoritmo 5 ao parâmetro β, realizamos extensos experimentos numéricos no conjunto de dados sintéticos. Definimos um limite de $w = 1.000$ iterações e variamos a dimensão do problema n, o posto r, a porcentagem de dados faltantes p_- e a tolerância ε. Os resultados são uma média de cinco simulações. Os experimentos

mostram que o número (médio) de iterações do Algoritmo 5 (Fase 1) pode ser altamente reduzido por uma escolha adequada do parâmetro β, principalmente quando o número de entradas observadas é muito pequeno.

A Figura 9 mostra o número de iterações como uma função do parâmetro β considerando a porcentagem de dados faltantes $p_- \in \{92\%, 85\%, 72\%, 50\%\}$, $n = 1000$, $r = 5$, and $\varepsilon = 10^{-8}$. Como pode ser visto, para $\beta \geqslant 19$, Algoritmo 5 atinge o número mínimo de iterações nos quatro cenários. Além disso, para $p_- = 92\%$ o número de iterações é reduzido em 79% com respeito a escolha de $\beta = 2$ (valor padrão).

Figura 9 – Número de iterações versus β para $n = 1000$, $r = 5$, $\varepsilon = 10^{-8}$ e $p_- \in \{92\%, 85\%, 72\%, 50\%\}$.



Também fizemos experimentos variando a dimensão n da matriz no conjunto $\{500, 1000, 2000, 4000\}$ fixando $r = 5$, $\varepsilon = 10^{-8}$, e $p_- = 40\%$. A Figura 10 mostra que o número mínimo de iterações em todos os cenários ocorreu para $\beta \geqslant 20$.

A Figura 11 nos dá uma intuição de como definir o valor de β quando o posto r varia. Para este experimento, nós fixamos $(n, \varepsilon, p_-) = \left(1000, 10^{-5}, 50\%\right)$ e variamos $r \in \{3, 5, 10, 30, 50, 80, 100\}$. Como pode ser visto, quanto maior o posto da matriz alvo, menor será o valor "ótimo" de β.

3.3.2 Experimentos com dados sintéticos

Agora, voltemos nossa atenção para experimentos com dados sintéticos, gerados conforme descrito no início da Seção 3.3. Para esses experimentos, definimos $n = 1000$, $p = 40\%$ e o posto r assume valores no conjunto $\{10, 15, 20, 40, 80, 100\}$. Nos critérios de parada, usamos as tolerâncias $\varepsilon = \varepsilon_1 = \varepsilon_2 = 10^{-4}$, $\varepsilon_3 = 10^{-3}$, $\varepsilon_\lambda = 10^{-6}$ e para o SVT, seguindo [14], fixamos $\tau = 5n$ e $t_k = 1.2n^2/|\Omega|$, em que $|\Omega|$ é a cardinalidade de Ω. Para o FPC usamos a estratégia padrão para atualizar o parâmetro de regularização:

Figura 10 – Número de iterações versus β para $n \in \{500, 1000, 2000, 4000\}$, $r = 5$, $\varepsilon = 10^{-8}$, e $p_- = 40\%$.

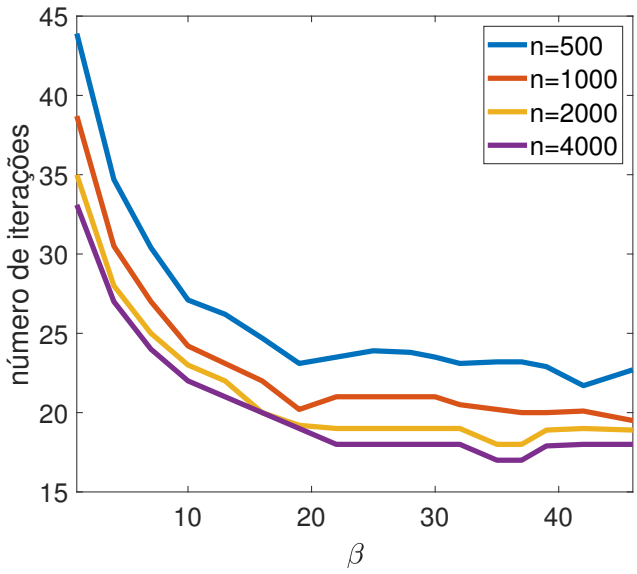
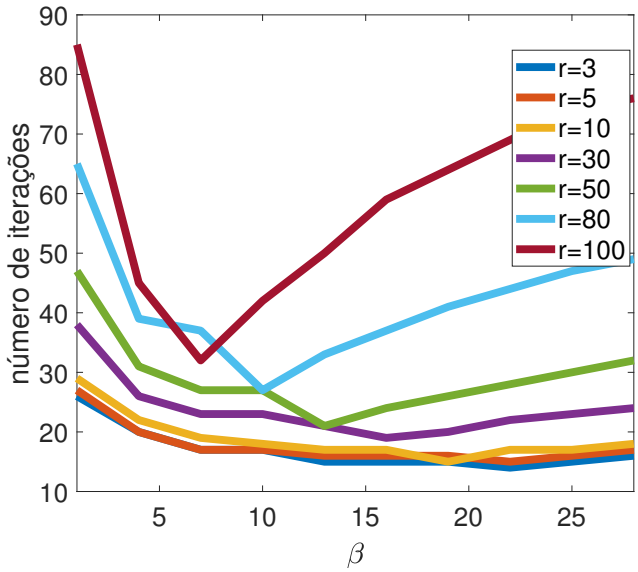


Figura 11 – Valor ótimo de β com $n = 1000$, $r \in \{3, 5, 10, 30, 50, 80, 100\}$, $\varepsilon = 10^{-5}$ e $p_- = 50\%$.



$\lambda_0 = \|P_\Omega(A)\|_2$, $\lambda_k = \max\{0.25\lambda_{k-1}, 0.01\}$ e $t_k = 1.99$, conforme recomendado em [37]. No Algoritmo 7, definimos o número máximo de iterações da Fase 1 como $w = 500$, bem como o limite de iteração para a Fase 2 igual a $it_{\max} = 500$. Para o parâmetro de aceleração β usamos $\{13, 13, 12, 10, 5, 5\}$, respectivamente (seguindo o estudo da Seção 3.3.1).

Para a avaliação de desempenho, usamos o erro relativo, definido por $\text{Rer} = \|A - \tilde{A}\|_F / \|A\|_F$, em que $\tilde{A}$ é a matriz recuperada e A é a matriz alvo.

Os resultados são mostrados na Tabela 2, na qual r denota o posto alvo, IT é o número total de iterações (para o Algoritmo 7, o número de iterações da Fase 2 está apresentado entre parênteses) e $t(s)$ o tempo em segundos. Neste primeiro conjunto de

experimentos, destacamos que todos os algoritmos recuperaram corretamente o posto subjacente. Como pode ser visto, nosso algoritmo converge mais rápido que os outros algoritmos. Além disso, quanto maior o posto r da matriz desejada, melhor é o desempenho do Algoritmo 7, quando comparado com FRSI que, por sua vez, é consistentemente mais rápido que SVT e FPC. Em termos de erro relativo, o Algoritmo 7 foi sempre o primeiro ou o segundo melhor.

Ressaltamos que a maioria das iterações do Algoritmo 7 corresponde às iterações da Fase 1 (warm-start). Depois de mudar para a segunda fase, apenas algumas iterações são necessárias para atingir o critério de parada.

Tabela 2 – Comparação do Algortimo 7 com FRSI, SVT e FPC. Avaliação de performance para $n = 1,000$, $p_- = 40\%$, r toma valor no conjunto $\{10, 15, 20, 40, 80, 100\}$ e $\beta \in \{13, 13, 12, 10, 5, 5\}$, respectivamente.

r	método	IT	t(s)	Rer
10	Alg. 7	**16** (1)	**1.80**	**5.84e-06**
	FRSI	18	2.04	1.68e-04
	SVT	43	5.78	1.09e-04
	FPC	74	9.68	1.70e-05
15	Alg. 7	**18** (1)	**1.77**	**6.90e-06**
	FRSI	20	2.15	1.49e-04
	SVT	47	5.82	1.07e-04
	FPC	81	13.04	1.72e-05
20	Alg. 7	**18** (2)	**1.80**	**1.12e-06**
	FRSI	21	2.89	1.95e-04
	SVT	51	6.60	1.13e-04
	FPC	91	15.62	1.78e-05
40	Alg. 7	**25** (1)	**2.77**	**1.63e-06**
	FRSI	28	3.85	2.90e-04
	SVT	64	11.15	1.26e-04
	FPC	125	56.49	1.83e-05
80	Alg. 7	**31** (3)	**7.08**	4.76e-05
	FRSI	42	10.03	5.71e-04
	SVT	93	34.44	1.47e-04
	FPC	212	165.98	**2.04e-05**
100	Alg. 7	**38** (2)	**12.26**	5.42e-05
	FRSI	46	20.51	1.21e-04
	SVT	144	68.41	1.76e-04
	FPC	361	415.25	**2.38e-05**

Nós também realizamos experimentos em matrizes maiores com pouquíssimas entradas observadas. Os experimentos foram conduzidos sob os mesmos parâmetros de antes e estabelecemos um limite de tempo de uma hora. Comparamos os resultados apenas com o SVT, pois, neste caso, ele é mais rápido que os algoritmos FRSI e FPC. Os resultados são exibidos na Tabela 3. Pode-se observar que ambos os algoritmos apresentam

desempenho competitivo para os casos testados. Nessa tabela, fornecemos uma coluna adicional com o posto recuperado $\hat{r}$.

O Algoritmo 7 geralmente supera o SVT em termos de erro relativo e é mais rápido para matrizes com posto maior. Ressaltamos que o tempo do Algoritmo 7 nunca é maior que 2,15 vezes o tempo do SVT, mas pode ser muito menor nos casos em que r é grande. Além disso, observamos que o posto $\hat{r}$ da matriz recuperada pelo SVT pode ser maior que o posto da matriz original, enquanto que o Algoritmo 7 recuperou uma matriz com o posto correto para este conjunto de experimentos. Por outro lado, o SVT tende a apresentar melhor desempenho para valores menores do posto e quando o número de entradas faltantes não é muito alto. No entanto, torna-se consideravelmente lento quando o posto aumenta e a porcentagem de entradas conhecidas diminui. Em alguns casos, como (1000,20,90%) e (10000,40,97%), tivemos que mudar o tamanho de passo para a escolha conservadora de $t_k = 1,99$, para o qual SVT tem garantias de convergência teórica, ao invés de $t_k = 1.2n^2/|\Omega|$, para evitar exceder o limite de tempo.

Tabela 3 – Comparação do Algoritmo 7 com o SVT para diferentes valores de (n, r, p_-) e $\beta \in \{13, 12, 19, 12, 19, 12, 19, 10\}$, respectivamente.

(n, r, p_-)	método	IT	t(s)	Rer	$\hat{r}$
(1000,10,90%)	Alg. 7	**116** (3)	5,45	**1.36e-04**	10
	SVT	174	**5.26**	1.44e-04	10
(1000,20,90%)	Alg. 7	**102** (1)	**7.34**	3.25e-01	20
	SVT	500	279.56	**2.23e-01**	168
(2000,10,90%)	Alg. 7	86 (2)	12.54	**3.68e-05**	10
	SVT	**83**	**9.55**	1.39e-04	10
(2000,20,92%)	Alg. 7	**147** (5)	**28.11**	1.59e-04	20
	SVT	262	168.62	**1.51e-04**	31
(5000,10,90%)	Alg. 7	69 (1)	63.4	**2.36e-05**	10
	SVT	**53**	**33.7**	1.18e-04	10
(5000,25,96%)	Alg. 7	**215** (4)	**149.89**	**1.62e-04**	25
	SVT	297	1355.23	2.34e-04	50
(10000,10,90%)	Alg. 7	65 (3)	245.13	**8.27e-06**	10
	SVT	**43**	**113.84**	1.07e-04	10
(10000,40,97%)	Alg. 7	**256** (12)	**1018.56**	**8.01e-04**	40
	SVT	677	3600	4.13e-02	95

3.3.3 Experimentos com o banco de dados *MovieLens*

O banco de dados *MovieLens* é um sistema de recomendação bem conhecido que é frequentemente usado em experimentos de completamento de matrizes [59]. Ele contém avaliações (variando no conjunto $\{1, 2, 3, 4, 5\}$) de diferentes usuários em filmes. A Tabela 4 apresenta as características dos conjuntos de dados utilizados nos experimentos.

Tabela 4 – Bancos de dados do *MovieLens* utilizados nos experimentos

banco de dados	# usuários	# filmes	# avaliações
MovieLens-100k	943	1,682	100,000
MovieLens-1M	6,040	3,952	1,000,209

Seja $\hat{\Omega}$ o conjunto de índices de entradas observadas (avaliações) em um banco de dados. Dividimos aleatoriamente $\hat{\Omega}$ em dois conjuntos disjuntos Ω e Ω' de modo que $\hat{\Omega} = \Omega \cup \Omega'$ e $|\Omega| = \frac{9}{10}|\hat{\Omega}|$. As entradas observadas correspondentes a Ω são passadas como entrada para os algoritmos, enquanto que as de Ω' são usadas para avaliar o desempenho pela raiz do erro quadrático médio (RMSE):

$$RMSE = \sqrt{\|P_{\Omega'}(A - \tilde{A})\|_F^2/|\Omega'|}.$$

Como o Algoritmo 7 e o FRSI precisam da informação do posto, nós realizamos alguns experimentos para diferentes valores de r e definimos $r = 130$ para MovieLens-100k e $r = 340$ para MovieLens-1M porque essas opções fornecem o menor RMSE para ambos os métodos. Observe que, neste caso, r é um valor grande e usando a Figura 11 nós fixamos o parâmetro de aceleração do Algoritmo 7 em $\beta = 2$. Nesses experimentos, definimos as tolerâncias $\varepsilon = \varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 10^{-3}$, $\varepsilon_\lambda = 10^{-2}$, e para SVT fixamos $t_k = 1.99$ e, como $m \neq n$, definimos $\tau = 8\sqrt{mn}$ como sugerido em [14] . Além disso, estabelecemos um limite de tempo de uma hora para todos os algoritmos.

Os resultados (média de cinco execuções) são mostrados na Tabela 5. Como podemos ver, o Algoritmo 7 requer muito menos tempo de computação para atingir um valor de RMSE razoável (em comparação com os algoritmos considerados). Observamos que, para o banco de dados MovieLens-1M, o Algoritmo 7 foi o único capaz de atingir os critérios de parada em menos de uma hora.

Tabela 5 – Resultados numéricos no banco de dados *MovieLens*

método	100k			1M		
	IT	t(s)	RMSE	IT	t(s)	RMSE
Alg. 7	**35** (2)	**16.50**	**0.9852**	**40** (3)	**565.91**	0.91833
FRSI	272	196.95	0.9898	176	3600	0.95236
SVT	2000	2186	1.0037	929	3600	**0.9088**
FPC	557	1042	1.0163	183	3600	0.94424

Encerramos este capítulo com algumas considerações acerca dos resultados nele apresentados. Os experimentos numéricos mostram que a Fase 1 (Algoritmo 5) foi responsável pela maior parte das iterações e, além disso, o número de iterações dessa fase pode ser bastante reduzido pela escolha adequada do parâmetro β (veja o passo 7), conforme mostramos na Seção 3.3.1. Assim, duas linhas de trabalho futuro são (a) provar

a convergência da Fase 1, evitando possíveis casos em que o posto recuperado pela Fase 2 é diferente do posto alvo (veja Observação 3.1) e (b) investigar outras expressões para usar no lugar do parâmetro de extrapolação $\omega^j = (j-1)/(j+\beta)$.

4 Gradiente projetado para completamento de matrizes

Neste capítulo, consideramos um método de Gradiente Projetado (GP) aplicado ao problema de otimização com restrição de posto dado por

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}} \quad & \frac{1}{2}\|\mathscr{A}(X)-b\|_2^2 \\ \text{sujeito a} \quad & \operatorname{posto}(X)\leqslant r, \end{aligned} \tag{4.1}$$

em que $\mathscr{A}:\mathbb{R}^{m\times n}\to\mathbb{R}^d$ é uma transformação afim, $b\in\mathbb{R}^d$ e r é o posto da matriz alvo, o qual supomos ser conhecido a priori. No problema de completamento, podemos tomar $\mathscr{A}=\operatorname{vec}(P_\Omega)$ em virtude da Observação 2.2.

Como o conjunto viável do problema (4.1) é não-convexo, muitos dos resultados clássicos de teoria de convergência do Gradiente Projetado para o caso convexo não se aplicam a este problema.

A seguir, apresentamos uma revisão do método de gradiente projetado, primeiro em conjuntos convexos e depois em conjuntos não-convexos. Em seguida, aplicamos um método de gradiente projetado ao problema (4.1) e mostramos que, se $\mathscr{A}=\operatorname{vec}(P_\Omega)$ satisfaz $2r$-RIP (veja a Definição 2.3) com parâmetro $\delta_{2r}\in(0,1/2)$, então, a sequência gerada pelo método converge para a solução do problema. Finalmente, apresentamos alguns resultados numéricos do método e também considerando um mecanismo de aceleração similar ao estudado no método de gradiente proximal acelerado.

As principais referências para este capítulo são [25, 30, 31, 32].

4.1 Gradiente projetado em conjuntos convexos

O método do gradiente projetado é uma maneira simples e padrão para resolver problemas de otimização com restrição e utiliza o conceito de projeção, que consiste em resolver o problema

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}\|z-x\|_2^2 \\ \text{sujeito a} \quad & x\in\mathcal{X}\subset\mathbb{R}^n, \end{aligned} \tag{4.2}$$

que é conhecido como o problema de projeção de z em $\mathcal{X}$.

A partir de agora, usaremos a notação $\boldsymbol{P}_{\mathcal{X}}(z)$ para denotar a projeção de um ponto z no conjunto $\mathcal{X}$.

Nesta seção, consideramos o caso em que $\mathcal{X}$ é não-vazio, fechado e convexo, o que permite provar alguns resultados importantes para projeção e que estão resumidos na proposição a seguir.

Proposição 4.1. *[10, Proposição 2.1.3] Seja $\mathcal{X}$ um subconjunto não-vazio, fechado e convexo de $\mathbb{R}^n$.*

(a) Para cada $z \in \mathbb{R}^n$, existe um único $x^\star \in \mathcal{X}$ que minimiza $\frac{1}{2}\|z - x\|_2^2$ em $\mathcal{X}$. O vetor $x^\star$ é chamado de projeção de z no conjunto $\mathcal{X}$ e é denotado por $\boldsymbol{P}_{\mathcal{X}}(z)$.

(b) Dado algum $z \in \mathbb{R}^n$, um vetor $x^\star \in \mathcal{X}$ é igual a $\boldsymbol{P}_{\mathcal{X}}(z)$ se, e somente se,

$$\langle z - x^\star, x - x^\star \rangle \leqslant 0, \quad \forall x \in \mathcal{X}.$$

(c) A projeção $\boldsymbol{P}_{\mathcal{X}}(\cdot)$ é contínua e não-expansiva, isto é,

$$\|\boldsymbol{P}_{\mathcal{X}}(x) - \boldsymbol{P}_{\mathcal{X}}(y)\|_2 \leqslant \|x - y\|_2, \quad \forall x, y \in \mathbb{R}^n.$$

Sejam $f : \mathbb{R}^n \to \mathbb{R}$ uma função continuamente diferenciável com gradiente Lipschitz contínuo de constante $L > 0$ e $\mathcal{X} \subset \mathbb{R}^n$ um conjunto não-vazio, fechado e convexo. Considere o seguinte problema de otimização

$$\begin{aligned} \min \quad & f(x) \\ \text{sujeito a} \quad & x \in \mathcal{X}. \end{aligned} \tag{4.3}$$

A iteração do método de gradiente projetado é descrita como

$$x^0 \in \mathcal{X}, \quad x^{k+1} = \boldsymbol{P}_{\mathcal{X}}\left(x^k - \eta^k \nabla f(x^k)\right), \tag{4.4}$$

em que η^k é o tamanho de passo e $\boldsymbol{P}_{\mathcal{X}}(z)$ é a projeção do ponto z no conjunto $\mathcal{X}$, dada por

$$\boldsymbol{P}_{\mathcal{X}}(z) = \arg\min_{x \in \mathcal{X}} \frac{1}{2}\|x - z\|_2^2. \tag{4.5}$$

Quando a função f é convexa e considerando $i(x)$ a função indicadora de conjunto $\mathcal{X}$ definida como

$$i(x) = \begin{cases} 0, & \text{se } x \in \mathcal{X} \\ +\infty, & \text{se } x \notin \mathcal{X}, \end{cases} \tag{4.6}$$

tem-se

$$\text{prox}_i(z) = \arg\min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}\|z-x\|_2^2 + i(x) \right\} = \arg\min_{x\in\mathcal{X}} \left\{ \frac{1}{2}\|z-x\|_2^2 \right\} = P_{\mathcal{X}}(z),$$

e podemos escrever (4.3) como

$$\min_{x\in\mathbb{R}^n} f(x) + i(x), \tag{4.7}$$

o que mostra que, neste caso, o gradiente projetado é um caso especial de gradiente proximal.

Métodos de gradiente projetado têm sido aplicados com sucesso em problemas de otimização convexa, por exemplo, em [11, 12, 29]. Para esses casos, a continuidade e unicidade da projeção permitem demonstrar que a sequência gerada por (4.4)-(4.5) converge para uma solução do problema (4.3), mediante uma escolha adequada do tamanho de passo, o que pode ser feita por meio de uma busca de Armijo [29].

No entanto, a aplicação do gradiente projetado ao problema (4.1) torna-se um desafio, pois seu conjunto viável $C = \{X \in \mathbb{R}^{m\times n} \mid \text{rank}(X) \leqslant r\}$ é não-convexo e a continuidade e unicidade da projeção não valem, em geral. Esse problema será abordado na próxima seção.

4.2 Gradiente projetado em conjuntos não-convexos

Os resultados da Proposição 4.1 permitem derivar a convergência do processo iterativo (4.4)-(4.5) no cenário convexo. Entretanto, tais resultados não valem quando $\mathcal{X}$ é apenas fechado, o que torna a análise da convergência do gradiente projetado em conjuntos não-convexos mais complexa.

Nesta seção, fornecemos alguns resultados preliminares envolvendo a projeção e o método de gradiente projetado em conjuntos não-convexos. Note que, neste caso, a projeção $\boldsymbol{P}_{\mathcal{X}}(z)$ não é única. A Figura 12 ilustra essa situação - neste caso, $\boldsymbol{P}_{\mathcal{X}}(z) = \{u_1, u_2, u_3\}$.

Agora, vamos considerar o problema (4.3) para o caso em que o conjunto $\mathcal{X}$ é fechado e não-convexo. Como a projeção pode não ser única, assumiremos que o método do gradiente projetado tomará qualquer elemento do conjunto $\boldsymbol{P}_{\mathcal{X}}(z)$. Ou seja, a iteração do GP em conjuntos não-convexos pode ser dada por:

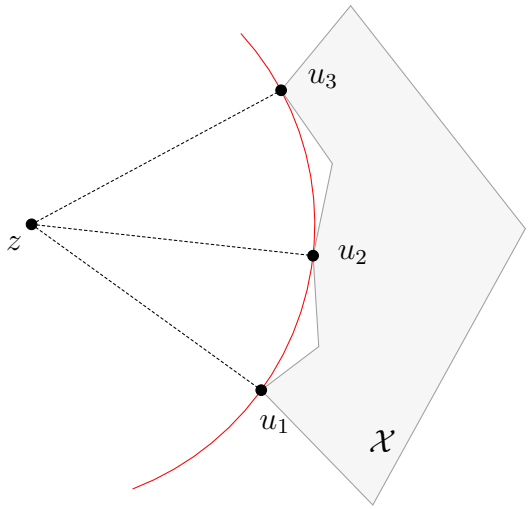
$$x^0 \in \mathcal{X}, \quad x^{k+1} \in \boldsymbol{P}_{\mathcal{X}}\left(x^k - \eta^k \nabla f(x^k)\right), \tag{4.8}$$

com $\boldsymbol{P}_{\mathcal{X}}$ definido em (4.5).

De (4.8) e (4.5) pode-se mostrar que x^{k+1} é solução do seguinte problema

$$\min_{x\in\mathcal{X}} \quad \langle \nabla f(x^k), x - x^k \rangle + \frac{1}{2\eta^k}\|x - x^k\|_2^2.$$

Figura 12 – Ilustração do conjunto $\boldsymbol{P}_{\mathcal{X}}(z)$. Os pontos u_1, u_2 e u_3 são as projeções do ponto z no conjunto $\mathcal{X}$.



Desse modo, segue que

$$\langle \nabla f(x^k), x^{k+1} - x^k \rangle \leqslant -\frac{1}{2\eta^k} \|x^{k+1} - x^k\|_2^2. \tag{4.9}$$

Usando a desigualdade acima e o fato de f ter gradiente Lipschitz contínuo, obtemos

$$\begin{aligned} f(x^{k+1}) &\leqslant f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L}{2} \|x^{k+1} - x^k\|_2^2 \\ &\leqslant f(x^k) - \frac{1}{2\eta^k} \|x^{k+1} - x^k\|^2 + \frac{L}{2} \|x^{k+1} - x^k\|_2^2 \\ &= f(x^k) + \frac{1}{2} \left(L - \frac{1}{\eta^k} \right) \|x^{k+1} - x^k\|_2^2. \end{aligned} \tag{4.10}$$

Isso mostra que, para $\eta^k \leqslant 1/L$, a sequência $\{f(x^k)\}$ é monótona não-crescente. Mais ainda, desde que $\|x^{k+1} - x^k\|_2 \neq 0$ e $\eta_k < 1/L$, temos que $f(x^{k+1}) < f(x^k)$.

Agora, assuma que $x^\star \in \mathcal{X}$ é uma solução para o problema (4.3), com $\mathcal{X}$ não-convexo, ou seja, $f(x) \geqslant f(x^\star), \forall x \in \mathcal{X}$. Como f é limitada inferiormente em $\mathcal{X}$ por $f(x^\star)$, temos então que $\{f(x^k)\}$ é convergente. Mas, novamente por (4.10), para $\eta^k = \eta \in (0, 1/L]$, temos que

$$\frac{1}{2} \left(\frac{1}{\eta} - L \right) \|x^{k+1} - x^k\|_2^2 \leqslant f(x^k) - f(x^{k+1}), \tag{4.11}$$

que implica

$$\lim_{k \to \infty} \|x^{k+1} - x^k\|_2 = 0. \tag{4.12}$$

Em outras palavras

$$\lim_{k \to \infty} \operatorname{dist} \left(x_k, \boldsymbol{P}_{\mathcal{X}} \left(x^k - \eta \nabla f(x_k) \right) \right) = 0,$$

em que

$$\operatorname{dist}(x, C) = \min\{\|x - y\|_2 \mid y \in C\},$$

é a distância do ponto $x \in \mathbb{R}^n$ ao conjunto fechado C.

Se $\mathcal{X}$ fosse convexo, pela Proposição 4.1 (a) $\boldsymbol{P}_{\mathcal{X}}(\cdot)$ seria única, e de (4.12) teríamos

$$\lim_{k\to\infty} \|x^k - \boldsymbol{P}_{\mathcal{X}}\left(x^k - \eta\nabla f(x^k)\right)\|_2 = 0.$$

Mais ainda, caso $x^k \to \bar{x}$, poderíamos usar a continuidade de $\boldsymbol{P}_{\mathcal{X}}(\cdot)$ (Proposição 4.1 (c)) para obter

$$\|\boldsymbol{P}_{\mathcal{X}}\left(\bar{x} - \eta\nabla f(\bar{x})\right) - \bar{x}\|_2 = 0, \tag{4.13}$$

que é uma condição necessária de otimalidade.

A propriedade (b) da Proposição 4.1 também pode não valer quando $\mathcal{X}$ é não-convexo, isto é, o ângulo formado pelos vetores $z - \boldsymbol{P}_{\mathcal{X}}(z)$ e $x - \boldsymbol{P}_{\mathcal{X}}(z)$ pode ser menor que o ângulo reto. Neste caso, temos o seguinte resultado.

Lema 4.1. *Seja $\mathcal{X}$ um conjunto não-vazio, fechado e não-convexo. Então, $\boldsymbol{P}_{\mathcal{X}}(z)$ é tal que*

$$\langle z - \boldsymbol{P}_{\mathcal{X}}(z), x - \boldsymbol{P}_{\mathcal{X}}(z)\rangle \leqslant \frac{1}{2}\|x - \boldsymbol{P}_{\mathcal{X}}(z)\|_2^2, \quad \forall x \in \mathcal{X}. \tag{4.14}$$

Demonstração. Primeiro note que

$$2\langle z - \boldsymbol{P}_{\mathcal{X}}(z), x - \boldsymbol{P}_{\mathcal{X}}(z)\rangle = \|z - \boldsymbol{P}_{\mathcal{X}}(z)\|_2^2 + \|x - \boldsymbol{P}_{\mathcal{X}}(z)\|_2^2 - \|z - x\|_2^2.$$

Agora, como $\boldsymbol{P}_{\mathcal{X}}(z)$ é minimizador global de $\|z - x\|_2^2$ em $\mathcal{X}$, temos que

$$\|z - \boldsymbol{P}_{\mathcal{X}}(z)\|_2^2 \leqslant \|z - x\|_2^2, \qquad \forall x \in \mathcal{X},$$

logo

$$\langle z - \boldsymbol{P}_{\mathcal{X}}(z), x - \boldsymbol{P}_{\mathcal{X}}(z)\rangle \leqslant \frac{1}{2}\|x - \boldsymbol{P}_{\mathcal{X}}(z)\|_2^2, \qquad \forall x \in \mathcal{X}.$$

□

Lema 4.2. *[25, Lema 2.5] Seja $\mathcal{X}$ não-vazio e fechado. Se $z^k \to z^*$ e $y^k \in \boldsymbol{P}_{\mathcal{X}}(z^k)$ é tal que $y^k \to y^*$, então $y^* \in \boldsymbol{P}_{\mathcal{X}}(z^*)$.*

Teorema 4.1. *Seja $\{x^k\}$ a sequência gerada pelo GP. Suponha que $x^k \to \bar{x}$. Então*

$$dist(\bar{x}, \boldsymbol{P}_{\mathcal{X}}(\bar{x} - \eta\nabla f(\bar{x}))) = 0. \tag{4.15}$$

Demonstração. Seja $\{x^k\} \subset \mathcal{X}$ sequência gerada pelo GP e assuma que $x^k \to \bar{x}$. Claramente $\bar{x} \in \mathcal{X}$ já que $\mathcal{X}$ é fechado. Como ∇f é contínua, temos que

$$z^k = x^k - \eta\nabla f(x^k) \to \bar{x} - \eta\nabla f(\bar{x}) =: \bar{z}.$$

Perceba que $x^{k+1} \in \boldsymbol{P}_{\mathcal{X}}(z^k)$ e como $x^{k+1} \to \bar{x}$, pelo Lema 4.2 temos que

$$\bar{x} \in \boldsymbol{P}_{\mathcal{X}}(\bar{z}) = \boldsymbol{P}_{\mathcal{X}}(\bar{x} - \eta\nabla f(\bar{x})),$$

e, portanto,

$$\text{dist}(\bar{x}, \boldsymbol{P}_{\mathcal{X}}(\bar{x} - \eta\nabla f(\bar{x}))) = 0.$$

□

Observação 4.1. *Mesmo com f convexa, quando $\mathcal{X}$ não é convexo, infelizmente a condição da equação (4.15) não configura uma condição suficiente de otimalidade. Por exemplo, considere o problema de minimizar $f(x) = c^T x$, com $c \neq 0$, sujeito a $\mathcal{X} = \{x \in \mathbb{R}^n \mid \|x\|_2 = 1\}$. Veja que qualquer constante $L > 0$ serve como constante de Lipschitz e com uma escolha adequada de $\eta < 1/L$ é possível mostrar que um maximizador cumpre tal condição, i.e, $\bar{x} - \eta\nabla f(\bar{x}) = 0$, com $c^T\bar{x} > c^T x, \forall x \in \mathcal{X}$. Com efeito, temos $\nabla f(x) = c$ e $f(x) \leqslant \|c\|_2$ para todo $x \in \mathcal{X}$. Assim, escolhendo $\eta = 1/\|c\|_2$, segue que $\bar{x} = \eta\nabla f(\bar{x}) = c/\|c\|_2 \in \mathcal{X}$ é o maximizador de f em $\mathcal{X}$, com $\bar{x} - \eta\nabla f(\bar{x}) = 0$. Por outro lado, como $\bar{x} \in \boldsymbol{P}_{\mathcal{X}}(\bar{x} - \eta\nabla f(\bar{x}))$, verifica-se que $\bar{x}$ cumpre a condição (4.15).*

Em relação a complexidade de iteração, temos o seguinte resultado.

Teorema 4.2. *Para $\eta^k = \eta < 1/L$, em no máximo*

$$\left\lceil \frac{2(f(x^0) - f(x^\star))}{(1/\eta - L)\varepsilon^2} \right\rceil$$

iterações do gradiente projetado, teremos $\|x^{k+1} - x^k\| < \varepsilon$.

Demonstração. Segue de (4.11) que

$$\|x^{k+1} - x^k\|_2^2 \leqslant \frac{2}{1/\eta - L}(f(x^k) - f(x^{k+1})).$$

Seja k o primeiro índice tal que $\|x^{k+1} - x^k\|_2 < \varepsilon$. Então para $j = 0, 1, \ldots, k-1$, temos que

$$\varepsilon^2 < \|x^{j+1} - x^j\|_2^2 \leqslant \frac{2}{1/\eta - L}(f(x^j) - f(x^{j+1})),$$

que implica em

$$\begin{aligned} k\varepsilon^2 &\leqslant \frac{2}{1/\eta - L}\sum_{j=0}^{k-1}(f(x^j) - f(x^{j+1})) = \frac{2}{1/\eta - L}(f(x^0) - f(x^k)) \\ &\leqslant \frac{2}{1/\eta - L}(f(x^0) - f(x^\star)). \end{aligned}$$

□

4.3 Completamento de matrizes usando gradiente projetado

Seja A uma matriz de posto r. Nessa seção, vamos considerar o problema (4.1) para o caso específico em que $\mathscr{A} = \text{vec}\,(P_\Omega)$ e $b = \text{vec}\,(P_\Omega(A))$. Assim, o problema (4.1) se reduz a

$$\begin{aligned} \min_{X \in \mathbb{R}^{m\times n}} \quad & f(X) := \frac{1}{2}\|P_\Omega(X) - P_\Omega(A)\|_F^2 \\ \text{sujeito a} \quad & \text{posto}(X) \leqslant r. \end{aligned} \tag{4.16}$$

Embora o conjunto viável de (4.16) seja não-convexo, *uma* projeção de uma matriz Y sobre tal conjunto é dada por [32]

$$\mathrm{SVD}_r(Y) = \sum_{i=1}^{r} \sigma_i u_i v_i^\top. \tag{4.17}$$

De agora em diante vamos considerar a seguinte iteração de gradiente projetado

$$\begin{aligned} Z^{k+1} &= X^k - \nabla f(X^k) \\ X^{k+1} &= \boldsymbol{P}_C(Z^{k+1}), \end{aligned} \tag{4.18}$$

em que $\boldsymbol{P}_C(Z^{k+1}) = \mathrm{SVD}_r(Z^{k+1})$ e $\nabla f(X) = P_\Omega(X - A)$ é o gradiente da função f definida no problema (4.16).

Proposição 4.2. *Sejam $X, Y \in \mathbb{R}^{m\times n}$. As seguintes propriedades valem para o operador P_Ω:*

(i) $\|P_\Omega(X)\|^2 \leqslant \|X\|^2$.

(ii) $P_\Omega^2(X) = P_\Omega * P_\Omega(X) = P_\Omega(P_\Omega(X)) = P_\Omega(X)$.

(iii) $\langle P_\Omega(X), Y\rangle = \langle X, P_\Omega(Y)\rangle$.

Demonstração. Consequência imediata da definição de P_Ω. □

Jain e Kar [30] propõem um algoritmo de gradiente projetado que visa resolver o problema geral (4.1) e mostram que se o operador $\mathscr{A}$ satisfaz RIP de ordem $2r$ com um parâmetro $\delta_{2r} \in (0, 1/3)$, então o método converge para uma solução do problema.

O próximo resultado mostra que o valor de δ_{2r} pode ser maior no caso específico em que $\mathscr{A} = \mathrm{vec}(P_\Omega)$ e f é a função objetivo do problema (4.16). Com efeito, o teorema a seguir mostra que se P_Ω satisfaz RIP de ordem $2r$ com um parâmetro $\delta_{2r} \in (0, 1/2)$, então a sequência $\{X^k\}$ gerada por (4.18) converge para a solução do problema (4.16).

Teorema 4.3. *Suponha que o operador P_Ω satisfaz RIP de ordem $2r$ com parâmetro $0 < \delta_{2r} < 1/2$. Sejam $X^\star$ uma solução ótima do problema (4.16) tal que $P_\Omega(X^\star) = P_\Omega(A)$ e $\{X^k\}$ a sequência gerada por (4.18). Então, o processo iterativo (4.18) gera uma matriz $\tilde{X}$ de posto no máximo r, tal que $\|\tilde{X} - X^\star\|_F^2 \leqslant \varepsilon$, em no máximo*

$$\left\lceil \frac{1}{\log\left(\dfrac{1-\delta_{2r}}{\delta_{2r}}\right)} \log\left(\frac{\|P_\Omega(X^0 - X^\star)\|_F^2}{(1-\delta_{2r})\varepsilon}\right) \right\rceil$$

iterações.

Demonstração. Sejam $f(X) = \frac{1}{2}\|P_\Omega(X - X^\star)\|_F^2$ e $Z^{k+1} = X^k - P_\Omega(X^k - X^\star)$. Temos,

$$\begin{aligned}
f(X^{k+1}) &= \frac{1}{2}\|P_\Omega(X^{k+1} - X^\star)\|_F^2 \\
&= \frac{1}{2}\|P_\Omega(X^{k+1} - X^k + X^k - X^\star)\|_F^2 \\
&= \frac{1}{2}\|P_\Omega(X^k - X^\star)\|_F^2 + \frac{1}{2}\|P_\Omega(X^{k+1} - X^k)\|_F^2 + \langle P_\Omega(X^{k+1} - X^k), P_\Omega(X^k - X^\star)\rangle \\
&\leqslant f(X^k) + \langle P_\Omega(X^{k+1} - X^k), P_\Omega(X^k - X^\star)\rangle + \frac{1}{2}\|X^{k+1} - X^k\|_F^2, \qquad (4.19)
\end{aligned}$$

em que a desigualdade (4.19) segue aplicando a propriedade (i) da Proposição 4.2 à matriz $X^{k+1} - X^k$. Agora, escrevendo $\frac{1}{2}\|X^{k+1} - X^k\|_F^2 = \frac{1}{2}\|X^{k+1} - Z^{k+1} + Z^{k+1} - X^k\|_F^2$ e expandindo o lado direito de (4.19), obtemos

$$\begin{aligned}
f(X^{k+1}) &\leqslant f(X^k) + \langle P_\Omega(X^{k+1} - X^k), P_\Omega(X^k - X^\star)\rangle + \frac{1}{2}\|X^{k+1} - Z^{k+1} + Z^{k+1} - X^k\|_F^2 \\
&= f(X^k) + \frac{1}{2}\|X^{k+1} - Z^{k+1}\|_F^2 + \frac{1}{2}\|Z^{k+1} - X^k\|_F^2 + \langle P_\Omega(X^{k+1} - X^k), P_\Omega(X^k - X^\star)\rangle \\
&+ \langle X^{k+1} - Z^{k+1}, Z^{k+1} - X^k\rangle \\
&= f(X^k) + \frac{1}{2}\|X^{k+1} - Z^{k+1}\|_F^2 + \frac{1}{2}\|Z^{k+1} - X^k\|_F^2 + \langle P_\Omega(X^{k+1} - X^k), P_\Omega(X^k - X^\star)\rangle \\
&+ \langle X^{k+1} - X^k + P_\Omega(X^k - X^\star), X^k - P_\Omega(X^k - X^\star) - X^k\rangle \\
&= f(X^k) + \frac{1}{2}\|X^{k+1} - Z^{k+1}\|_F^2 + \frac{1}{2}\|Z^{k+1} - X^k\|_F^2 + \langle X^{k+1} - X^k, P_\Omega(X^k - X^\star)\rangle \\
&+ \langle X^{k+1} - X^k + P_\Omega(X^k - X^\star), X^k - P_\Omega(X^k - X^\star) - X^k\rangle \qquad (4.20) \\
&= f(X^k) + \frac{1}{2}\|X^{k+1} - Z^{k+1}\|_F^2 - \frac{1}{2}\|P_\Omega(X^k - X^\star)\|_F^2 \\
&\leqslant f(X^k) + \frac{1}{2}\|X^\star - Z^{k+1}\|_F^2 - \frac{1}{2}\|P_\Omega(X^k - X^\star)\|_F^2. \qquad (4.21)
\end{aligned}$$

A desigualdade (4.20) segue de (ii) e (iii) da Proposição 4.2 e da expressão de Z^{k+1}. A desigualdade (4.21) segue do fato de que X^{k+1} é a projeção de Z^{k+1} e, portanto, $\|X^{k+1} - Z^{k+1}\|_F^2 \leqslant \|X^\star - Z^{k+1}\|_F^2$. Usando novamente a expressão de Z^{k+1} no lado direito da desigualdade (4.21) e, em seguida, as propriedades (ii) e (iii) da Proposição 4.2, temos

$$\begin{aligned}
f(X^{k+1}) &\leqslant f(X^k) + \frac{1}{2}\|X^\star - X^k + P_\Omega(X^k - X^\star)\|_F^2 - \frac{1}{2}\|P_\Omega(X^k - X^\star)\|_F^2 \\
&= f(X^k) + \frac{1}{2}\|X^\star - X^k + P_\Omega(X^k - X^\star)\|_F^2 - \frac{1}{2}\|P_\Omega(X^k - X^\star)\|_F^2 \\
&= f(X^k) + \frac{1}{2}\|X^\star - X^k\|_F^2 + \frac{1}{2}\|P_\Omega(X^k - X^\star)\|_F^2 - \frac{1}{2}\|P_\Omega(X^k - X^\star)\|_F^2 \\
&+ \langle X^\star - X^k, P_\Omega(X^k - X^\star)\rangle \\
&= f(X^k) + \frac{1}{2}\|X^\star - X^k\|_F^2 - \langle X^k - X^\star, P_\Omega(X^k - X^\star)\rangle \\
&= f(X^k) + \frac{1}{2}\|X^\star - X^k\|_F^2 - \|P_\Omega(X^k - X^\star)\|_F^2. \qquad (4.22)
\end{aligned}$$

Finalmente, aplicando RIP à matriz $X^k - X^\star$ em (4.22), obtemos

$$\begin{aligned} f(X^{k+1}) &\leqslant f(X^k) + \frac{1}{2(1-\delta_{2r})}\|P_\Omega(X^k - X^\star)\|_F^2 - 2f(X^k) \\ &= \left(\frac{1}{(1-\delta_{2r})} - 1\right) f(X^k) \\ &= \frac{\delta_{2r}}{(1-\delta_{2r})} f(X^k). \end{aligned}$$

Como $0 < \delta_{2r} < 1/2$, segue que $0 < \delta_{2r}/(1-\delta_{2r}) < 1$. Assim, a sequência $\left\{f(X^k)\right\}$ converge geometricamente e podemos escrever

$$f(X^k) \leqslant \left(\frac{\delta_{2r}}{1-\delta_{2r}}\right)^k f(X^0). \tag{4.23}$$

Agora, da RIP aplicada à matriz $X^k - X^\star$, temos $(1-\delta_{2r})\|X^k - X^\star\|_F^2 \leqslant 2f(X^k)$, resultando na desigualdade

$$\|X^k - X^\star\|_F^2 \leqslant \frac{1}{(1-\delta_{2r})}\left(\frac{\delta_{2r}}{1-\delta_{2r}}\right)^k \|P_\Omega(X^0 - X^\star)\|_F^2, \tag{4.24}$$

completando a prova. □

Observação 4.2. *A análise do Teorema 4.3 se aplica a qualquer matriz $X^\star$ do conjunto $\mathcal{X}^\star = \{X, posto(X) \leqslant r \, e \, P_\Omega(X) = P_\Omega(A)\}$. Assim, usando este fato e notando que a sequência $\left\{Z^{k+1}\right\}$ é a mesma para qualquer elemento de $\mathcal{X}^\star$, pode-se mostrar que a sequência $\left\{X^k\right\}$ converge para a solução única do problema (4.16).*

O método estudado nesta seção está sintetizado no Algoritmo 8.

Algoritmo 8 – Gradiente projetado (GP)

Entrada: Entradas conhecidas de $A \in \mathbb{R}^{m\times n}$ indexadas por Ω, o posto r, $\varepsilon > 0$.

1. Inicialize $X^1 = 0$.
2. **para** $k = 1, 2, \ldots$ **faça**:
3. $\quad Z^{k+1} \leftarrow X^k - P_\Omega(X^k - A)$.
4. $\quad$ Compute $X^{k+1} \leftarrow \mathrm{SVD}_r(Z^{k+1})$.
5. $\quad$ **Se** algum critério de parada for atingido **então** termine.
6. **fim**

Saída: X^k.

4.4 Experimentos numéricos

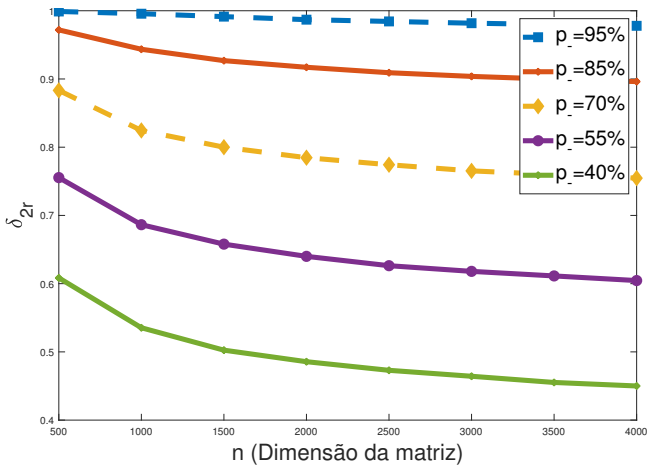
Nesta seção, apresentamos os resultados numéricos para o método estudado. Para isso, geramos matrizes de dados sintéticos utilizando o mesmo protocolo descrito na Seção 3.3.

Com o intuito de compararmos os resultados numéricos desse capítulo com os resultado do Algoritmo 7, que foram apresentados no Capítulo 3, nós realizamos os experimentos dessa seção considerando os mesmos parâmetros utilizados nas simulações daquele capítulo e utilizamos o critério de parada

$$\frac{\|P_\Omega\left(X^k - A\right)\|_F}{\|P_\Omega\left(A\right)\|_F} \leqslant \varepsilon.$$

Primeiro nós avaliamos quão restritiva é a propriedade de isometria restrita no contexto de completamento de matrizes. Para isso, geramos matrizes com $r = 10$ fixo, $n \in \{500, 1000, 1500, 2000, \ldots, 4000\}$, $p_- \in \{95\%, 85\%, 70\%, 55\%, 40\%\}$ e $\epsilon = 10^{-4}$ e estimamos o valor mínimo de δ_{2r} para o qual a matriz $X^k - X^\star$ satisfaz RIP de ordem $2r$. O resultado pode ser visto na Figura 13.

Figura 13 – Valor de δ_{2r} para o qual matriz $X^k - X^\star$ satisfaz $2r-$RIP. Simulação para $r = 10$ fixo, $n \in \{500, 1000, 1500, 2000, \ldots, 4000\}$, $p_- \in \{95\%, 85\%, 70\%, 55\%, 40\%\}$ e $\epsilon = 10^{-4}$.



Como podemos observar, a condição é considerada forte, uma vez que ela requer um apagamento pequeno para que o valor de δ_{2r} fique abaixo de $1/2$. Por outro lado, o resultado do Teorema 4.3 representa um avanço, uma vez que relaxa o valor de δ_{2r} de $1/3$ para $1/2$, fornecendo garantia de convergência teórica para os casos em há uma quantidade relativamente grande de entradas conhecidas.

Nas próximas seções apresentamos os resultados numéricos com dados sintéticos por meio de tabelas nas quais r denota o posto alvo, IT é o número total de iterações (para o Algoritmo 7, o número de iterações da Fase 2 está apresentado entre parênteses), $t(s)$ o tempo em segundos e Rer é o erro relativo (o mesmo usado no Capítulo 3).

4.4.1 Gradiente projetado sem aceleração

Nessa seção, recuperamos matrizes de dados sintéticos gerados de acordo com as tabelas da Seção 3.3.2. Para um primeiro experimento, nós geramos matrizes com $n = 1000$, $p_- = 40\%$, $\varepsilon = 10^{-4}$ e r variando no conjunto $\{10, 15, 20, 40, 80, 100\}$. Os resultados estão descritos na Tabela 6. Como pode ser visto, neste cenário, no qual a porcentagem de entradas conhecidas é relativamente alta, o Algoritmo 8 possui resultados levemente inferiores aos obtidos pelo Algoritmo 7.

Tabela 6 – Resultados numéricos para o Gradiente Projetado (GP) e o Algoritmo 7. Avaliação de performance para $n = 1000$, $p_- = 40\%$, $\varepsilon = 10^{-4}$, e r variando no conjunto $\{10, 15, 20, 40, 80, 100\}$.

r	método	IT	t(s)	Rer
10	Alg. 7	16 (1)	1.80	5.84e-06
	GP	13	1.89	1.01e-04
15	Alg. 7	18 (1)	1.77	6.90e-06
	GP	14	2.15	1.03e-04
20	Alg. 7	18 (2)	1.80	1.12e-06
	GP	15	3.09	9.84e-05
40	Alg. 7	25 (1)	2.77	1.63e-06
	GP	19	4.28	9.99e-05
80	Alg. 7	31 (3)	7.08	4.76e-05
	GP	27	10.01	1.56e-04
100	Alg. 7	38 (2)	12.26	5.42e-05
	GP	33	16.74	1.56e-04

Encerramos essa seção com os resultados numéricos para matrizes geradas com $r = 10$, $p_- = 90\%$, $\varepsilon = 10^{-4}$, e n variando no conjunto $\{1000, 2000, 5000, 10000\}$. Os resultados são mostrados na Tabela 7. Podemos observar que neste cenário o Algoritmo 7 tem resultados melhores que o GP, tanto em tempo quanto em número de iterações e erro relativo.

Tabela 7 – Resultados numéricos para o Gradiente Projetado (GP) e Algoritmo 7. Avaliação de performance para $r = 10$, $p_- = 90\%$, $\varepsilon = 10^{-4}$, e n variando no conjunto $\{1000, 2000, 5000, 10000\}$

n	método	IT	t(s)	Rer
1000	Alg. 7	116 (3)	5,45	1.36e-04
	GP	239	7.40	1.91e-04
2000	Alg. 7	86 (2)	12.54	3.68e-05
	GP	156	18.82	1.45e-04
5000	Alg. 7	69 (1)	63.40	2.36e-05
	GP	116	90.81	1.18e-04
10000	Alg. 7	65 (3)	245.13	8.27e-06
	Gp	103	359.23	1.08e-04

4.4.2 Gradiente projetado com aceleração

Nessa seção, apresentamos resultados numéricos incluindo no gradiente projetado o mesmo mecanismo de aceleração de Nesterov utilizado no passo 7 do Algoritmo 5, isto é,

$$Z^{k+1} = X^k + \frac{k-1}{k+\beta}\left(X^k - X^{k-1}\right), \quad \text{com} \quad \beta \geqslant 2.$$

Dessa vez, geramos matrizes com posto $r = 10$, $p_- = 90\%$, $\varepsilon = 10^{-4}$, n variando no conjunto $\{1000, 2000, 5000, 10000\}$ e obtivemos resultados para diferentes valores de β. Podemos observar, na Tabela 8, que o método converge mais rápido para valores maiores de β, convergindo para a solução mais rápido que o Algoritmo 7, quando usamos o mesmo valor de β para ambos.

Tabela 8 – Comparação dos resultados do Gradiente Projetado Acelerado (GPA) e do Algoritmo 7. Simulação para $r = 10$, $p_- = 90\%$, $\varepsilon = 10^{-4}$, n variando no conjunto $\{1000, 2000, 5000, 10000\}$ e para diferentes valores de β.

n	método	β	IT	t(s)	Rer
1000	Alg. 7	13	116 (3)	5,45	1.36e-04
	GP	2	157	4.90	1.76e-04
		13	67	2.05	1.77e-04
2000	Alg. 7	19	86 (2)	12.54	3.68e-05
	GP	2	118	14.25	1.36e-04
		19	53	6.33	1.37e-04
5000	Alg. 7	19	69 (1)	63.40	2.36e-05
	GP	2	97	74.95	1.16e-04
		19	44	34.08	9.49e-05
10000	Alg. 7	19	65 (3)	245.13	8.27e-06
	GP	2	90	311.63	1.07e-04
		19	41	141.93	1.07e-04

Tabela 9 – Comparação dos resultados do Gradiente Projetado Acelerado (GPA) e do Algoritmo 7. Tabela gerada para matrizes geradas com $r \in \{20, 25, 40\}$, $p_- \in \{92\%, 96\%, 97\%\}$, $\varepsilon = 10^{-4}$, n variando no conjunto $\{2000, 5000, 10000\}$ e β variando no conjunto $\{12, 12, 10\}$.

(n, r, p_-)	método	β	IT	t(s)	Rer
$(2000, 20, 92\%)$	Alg. 7	12	147 (5)	28.11	1.59e-04
	GP	2	204	24.43	1.83e-04
		12	81	9.89	1.86e-04
$(5000, 25, 96\%)$	Alg. 7	12	215 (4)	149.89	1.62e-04
	GP	2	359	190.74	1.82e-04
		12	120	62.77	1.82e-04
$(10000, 40, 97\%)$	Alg. 7	10	256 (12)	1018.56	8.01e-04
	GP	2	462	1266.43	1.85e-04
		10	151	410.43	1.86e-04

Também realizamos experimentos variando a porcentagen de apagamento e o posto da matriz. Para isso, geramos matrizes como $r \in \{20, 25, 40\}$, $p_- \in \{92\%, 96\%, 97\%\}$, $\varepsilon = 10^{-4}$, n variando no conjunto $\{2000, 5000, 10000\}$ e β variando no conjunto $\{12, 12, 10\}$. Os resultados da Tabela 9 mostram que o gradiente projetado acelerado consegue recuperar matrizes de posto mais alto e ainda possui resultados melhores que o Algoritmo 7.

5 Conclusões e trabalhos futuros

Neste trabalho, estudamos a aplicação de métodos de gradiente proximal e de gradiente projetado para completamento de matrizes de posto conhecido a priori. As principais contribuições originais desta tese concentram-se nos capítulos 3 e 4 e são sintetizadas na sequência.

No Capítulo 3, definimos uma iteração tipo ponto fixo inspirada na heurística Fixed-rank Soft-Impute (FRSI) para resolver um problema de quadrados mínimos regularizado (QMR) pela norma nuclear que modela o completamento de matrizes.

Uma decomposição SVD parcial é utilizada a cada iteração e leva em conta a informação do posto da matriz resultante. Mostramos que, se a sequência dos valores singulares subjacentes é somável, então a sequência gerada pelo processo iterativo é quasi-Fejér convergente para a solução do problema de QMR. Independente dessa condição, uma versão acelerada da heurística pode ser utilizada como uma fase de *warm-start* para um algoritmo de gradiente proximal acelerado.

Essa ideia deu origem a um algoritmo de duas fases, descrito na Seção 3.2. Experimentos numéricos realizados com dados sintéticos e reais indicam que o algoritmo proposto supera a heurística FRSI [41] e tem desempenho comparável com algoritmos bem estabelecidos na literatura de completamento de matrizes, como o SVT e FPC. Estes resultados estão publicados no artigo [2].

No Capítulo 4, propomos a aplicação de um método de gradiente projetado para resolver um problema de otimização com restrição de posto, como uma formulação alternativa para completamento de matrizes. O principal desafio nessa abordagem refere-se ao uso de uma abordagem de gradiente projetado em conjuntos não-convexos, cuja literatura é limitada e não há garantia de convergência para o caso geral.

Não obstante, seguindo algumas referências da literatura, utilizamos a propriedade de isometria restrita para mostrar que a sequência gerada pelo algoritmo proposto converge para a solução do problema. Em particular, no Teorema 4.3 mostramos que é possível relaxar o parâmetro da isometria restrita (δ_{2r}) de 1/3, como usado em trabalhos anteriores [30], para 1/2. Apesar de pequena, essa diferença pode ser bastante relevante nos experimentos numéricos, conforme mostramos nas simulações. Além disso, propomos também um mecanismo similar à aceleração de Nesterov para o método de gradiente projetado que, embora não tenha garantia de convergência, possui bons resultados numéricos.

Além das contribuições destacadas neste capítulo e dos demais resultados que compõem esta tese, diversas ideias surgiram ao longo do desenvolvimento deste trabalho

e, por limitação de tempo e escopo, não foi possível explorá-las. Algumas dessas ideias constituem desdobramentos dos resultados aqui apresentados e serão consideradas em termos de trabalhos futuros.

Embora a primeira fase do Algoritmo 7 seja considerada uma heurística, nos resultados numéricos apresentados na Seção 3.3, ela foi responsável pela maioria das iterações. Assim, uma linha de investigação futura consiste em estudar a convergência teórica da fase 1 sob condições mais fracas.

Na Seção 4.3 aplicamos um método de gradiente projetado para completamento de matrizes e mostramos que a sequência gerada converge para uma solução do problema sob uma condição de isometria restrita admitidamente forte nesse contexto. Diante disso, uma outra linha de estudo futuro seria investigar as propriedades do conjunto viável do problema (4.16) e tentar derivar resultados para a projeção que permitam analisar a convergência do método abrindo mão da propriedade de isometria restrita.

Referências

[1] Araujo, T. d. S., Gonçalves, D. S., and Torezzan, C. Completamento de matrizes de posto reduzido usando gradiente projetado. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics 9*, 1 (2022). Cited on page 16.

[2] Araújo, T. d. S., Gonçalves, D. S., and Torezzan, C. A two-phase rank-based algorithm for low-rank matrix completion. *Optimization Letters* (2022), 1–17. Cited 3 times on pages 15, 51 e 79.

[3] Araújo, T. d. S., and Torezzan, C. Using rank information to estimate missing values in data matrices. In *LII Simpósio Brasileiro de Pesquisa Operacional (SBPO)* (2021). Cited on page 15.

[4] Araújo, T. d. S., and Torezzan, C. Using rank information to improve performance of matrix completion algorithms. In *22nd Conference of the International Federation of Operational Research Societies (IFORS)* (2021). Cited on page 15.

[5] Attouch, H., and Peypouquet, J. The rate of convergence of nesterov's accelerated forward-backward method is actually faster than 1/k^2. *SIAM Journal on Optimization 26*, 3 (2016), 1824–1834. Nenhuma citação no texto.

[6] Beck, A., and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences 2*, 1 (2009), 183–202. Cited 4 times on pages 17, 30, 31 e 51.

[7] Bell, R. M., and Koren, Y. Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter 9*, 2 (2007), 75–79. Cited on page 48.

[8] Bennett, J., Lanning, S., et al. The netflix prize. In *Proceedings of KDD cup and workshop* (2007), vol. 2007, Citeseer, p. 35. Cited on page 48.

[9] Berg, R. v. d., Kipf, T. N., and Welling, M. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017). Cited 2 times on pages 15 e 48.

[10] Bertsekas, D. P. *Nonlinear Programming 2nd.* (Belmont, MA: Athena Scientific), 1999. Cited on page 67.

[11] Birgin, E. G., Martínez, J. M., and Raydan, M. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization 10*, 4 (2000), 1196–1211. Cited on page 68.

[12] Birgin, E. G., Martínez, J. M., and Raydan, M. Inexact spectral projected gradient methods on convex sets. *IMA Journal of Numerical Analysis 23*, 4 (2003), 539–559. Cited on page 68.

[13] Bokde, D., Girase, S., and Mukhopadhyay, D. Matrix factorization model in collaborative filtering algorithms: A survey. *Procedia Computer Science 49* (2015), 136–146. Cited on page 15.

[14] Cai, J.-F., Candès, E. J., and Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization 20*, 4 (2010), 1956–1982. Cited 8 times on pages 17, 27, 43, 44, 56, 59, 60 e 64.

[15] Candès, E. J., and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational mathematics 9*, 6 (2009), 717–772. Cited 5 times on pages 15, 32, 33, 36 e 39.

[16] Candès, E. J., and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory 56*, 5 (2010), 2053–2080. Cited 5 times on pages 15, 32, 34, 35 e 39.

[17] Chiang, K.-Y., Hsieh, C.-J., Natarajan, N., Dhillon, I. S., and Tewari, A. Prediction and clustering in signed networks: a local to global perspective. *The Journal of Machine Learning Research 15*, 1 (2014), 1177–1213. Cited on page 49.

[18] Chistov, A. L., and Grigor'Ev, D. Y. Complexity of quantifier elimination in the theory of algebraically closed fields. In *International Symposium on Mathematical Foundations of Computer Science* (1984), Springer, pp. 17–31. Cited on page 33.

[19] Combettes, P. L. Quasi-fejérian analysis of some optimization algorithms. In *Studies in Computational Mathematics*, vol. 8. Elsevier, 2001, pp. 115–152. Cited 2 times on pages 51 e 53.

[20] Dokmanic, I., Parhizkar, R., Ranieri, J., and Vetterli, M. Euclidean distance matrices: essential theory, algorithms, and applications. *IEEE Signal Processing Magazine 32*, 6 (2015), 12–30. Cited 3 times on pages 15, 35 e 49.

[21] Fan, J., Tian, Z., Zhao, M., and Chow, T. W. Accelerated low-rank representation for subspace clustering and semi-supervised classification on large-scale data. *Neural Networks 100* (2018), 39–48. Cited on page 15.

[22] Fazel, M. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002. Cited 3 times on pages 17, 25 e 27.

[23] Golub, G. H., and Van Loan, C. F. *Matrix computations*. JHU press, 2013. Cited 2 times on pages 22 e 24.

[24] Gu, S., Xie, Q., Meng, D., Zuo, W., Feng, X., and Zhang, L. Weighted nuclear norm minimization and its applications to low level vision. *International journal of computer vision 121*, 2 (2017), 183–208. Cited on page 15.

[25] Hager, W. W., Phan, D. T., and Zhu, J. Projection algorithms for nonconvex minimization with application to sparse principal component analysis. *Journal of Global Optimization 65*, 4 (2016), 657–676. Cited 2 times on pages 66 e 70.

[26] Hale, E. T., Yin, W., and Zhang, Y. A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. *CAAM TR07-07, Rice University 43* (2007), 44. Cited on page 42.

[27] Havel, T. F., and Wüthrich, K. An evaluation of the combined use of nuclear magnetic resonance and distance geometry for the determination of protein conformations in solution. *Journal of molecular biology 182*, 2 (1985), 281–294. Cited on page 49.

[28] Hiriart-Urruty, J.-B., and Lemaréchal, C. *Fundamentals of convex analysis*. Springer Science & Business Media, 2004. Cited 2 times on pages 19 e 21.

[29] Iusem, A. N. On the convergence properties of the projected gradient method for convex optimization. *Computational and Applied Mathematics 22*, 1 (2003), 37–52. Cited on page 68.

[30] Jain, P., Kar, P., et al. Non-convex optimization for machine learning. *Foundations and Trends in Machine Learning 10*, 3-4 (2017), 142–363. Cited 3 times on pages 66, 72 e 79.

[31] Jain, P., Meka, R., and Dhillon, I. Guaranteed rank minimization via singular value projection. *Advances in Neural Information Processing Systems 23* (2010). Cited 4 times on pages 32, 40, 41 e 66.

[32] Jain, P., Tewari, A., and Kar, P. On iterative hard thresholding methods for high-dimensional m-estimation. *Advances in neural information processing systems 27* (2014). Cited 2 times on pages 66 e 72.

[33] Jannach, D., Resnick, P., Tuzhilin, A., and Zanker, M. Recommender systems—beyond matrix completion. *Communications of the ACM 59*, 11 (2016), 94–102. Cited 2 times on pages 15 e 48.

[34] Larsen, R. M. Lanczos bidiagonalization with partial reorthogonalization. *DAIMI Report Series*, 537 (1998). Cited on page 57.

[35] Liben-Nowell, D., and Kleinberg, J. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management* (2003), pp. 556–559. Cited on page 49.

[36] Luo, X., Zhou, M., Xia, Y., and Zhu, Q. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics 10*, 2 (2014), 1273–1284. Cited on page 15.

[37] Ma, S., Goldfarb, D., and Chen, L. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming 128*, 1 (2011), 321–353. Cited 4 times on pages 42, 43, 59 e 61.

[38] Martinez, J. M., and Santos, S. A. Métodos computacionais de otimização. *Colóquio Brasileiro de Matemática, Apostilas 20* (1995). Cited 2 times on pages 17 e 18.

[39] Mazumder, R., Hastie, T., and Tibshirani, R. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research 11* (2010), 2287–2322. Cited 4 times on pages 28, 31, 45 e 46.

[40] Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., and Riedl, J. Movielens unplugged: experiences with an occasionally connected recommender system. In *Proceedings of the 8th international conference on Intelligent user interfaces* (2003), pp. 263–266. Cited on page 48.

[41] Moreira, N. J., Duarte, L. T., Lavor, C., and Torezzan, C. A novel low-rank matrix completion approach to estimate missing entries in euclidean distance matrix. *Computational and Applied Mathematics 37*, 4 (2018), 4989–4999. Cited 4 times on pages 46, 51, 56 e 79.

[42] Nesterov, Y., et al. *Lectures on convex optimization*, vol. 137. Springer, 2018. Cited on page 22.

[43] Nguyen, L. T., Kim, J., Kim, S., and Shim, B. Localization of iot networks via low-rank matrix completion. *IEEE Transactions on Communications 67*, 8 (2019), 5833–5847. Cited 2 times on pages 15 e 49.

[44] Nguyen, L. T., Kim, J., and Shim, B. Low-rank matrix completion: A contemporary survey. *IEEE Access 7* (2019), 94215–94237. Cited 4 times on pages 32, 33, 34 e 35.

[45] Nguyen, T. L., and Shin, Y. Matrix completion optimization for localization in wireless sensor networks for intelligent iot. *Sensors 16*, 5 (2016), 722. Cited 2 times on pages 15 e 49.

[46] Parikh, N., and Boyd, S. Proximal algorithms. *Foundations and Trends in optimization 1*, 3 (2014), 127–239. Cited 3 times on pages 15, 17 e 30.

[47] Recht, B. A simpler approach to matrix completion. *Journal of Machine Learning Research 12*, 12 (2011). Cited 2 times on pages 32 e 39.

[48] Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review 52*, 3 (2010), 471–501. Cited 2 times on pages 32 e 40.

[49] Ribeiro, A. A., and Karas, E. W. Otimização contínua: aspectos teóricos e computacionais. *São Paulo: Cengage Learning* (2013), 9. Cited 2 times on pages 17 e 18.

[50] Roger, H., and Charles, R. J. *Topics in matrix analysis*. Cambridge University Press, 1991. Cited 3 times on pages 17, 24 e 25.

[51] Srebro, N., and Jaakkola, T. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)* (2003), pp. 720–727. Cited on page 33.

[52] Svaiter, B. F. A class of Fejér convergent algorithms, approximate resolvents and the hybrid proximal-extragradient method. *Journal of Optimization Theory and Applications 162*, 1 (2014), 133–153. Cited on page 53.

[53] Tasissa, A., and Lai, R. Exact reconstruction of euclidean distance geometry problem using low-rank matrix completion. *IEEE Transactions on Information Theory 65*, 5 (2019), 3124–3144. Cited on page 49.

[54] Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological) 58*, 1 (1996), 267–288. Cited on page 31.

[55] Tibshirani, R. Proximal gradient descent and acceleration. *Lecture Notes* (2010). Cited on page 17.

[56] Trefethen, L. N., and Bau III, D. *Numerical linear algebra*, vol. 50. Siam, 1997. Cited 2 times on pages 17 e 23.

[57] Watkins, D. S. *Fundamentals of matrix computations*. John Wiley & Sons, 2004. Cited 2 times on pages 17 e 23.

[58] Watson, G. A. Characterization of the subdifferential of some matrix norms. *Linear algebra and its applications 170*, 0 (1992), 33–45. Cited on page 27.

[59] Yao, Q., and Kwok, J. T. Accelerated and inexact soft-impute for large-scale matrix and tensor completion. *IEEE Transactions on Knowledge and Data Engineering 31*, 9 (2018), 1665–1679. Cited 4 times on pages 15, 48, 49 e 63.