

2 Análise de Componentes Principais Robusta

A RPCA surgiu com o objetivo de dar robustez a Análise de Componentes Principais (PCA, *Principal Component Analysis*) clássica frente a algumas de suas limitações. Existem várias estratégias que buscam tal robustez, no entanto, uma obordagem clássica consiste na formulação que busca decompor a matriz de entrada de dados em duas matrizes: matriz de posto reduzido e matriz esparsa. Neste capítulo, apresenta-se uma formulação amplamente utilizada para resolver esse problema de decomposição, conhecida como PCP. Por fim, apresentam-se alguns algoritmos para resolvê-la.

2.1 Análise de Componentes Principais - PCA

A PCA, introduzida por Karl Pearson em 1901, tornou-se uma metodologia amplamente utilizada nas áreas de reconhecimento de padrões e visão computacional (PEARSON, 1901; MAATEN *et al.*, 2009). A PCA é uma técnica clássica de extração de recursos e representação de dados, a qual visa, principalmente, reduzir a dimensão dos dados, explorar relações entre variáveis, ou ainda, permitir o emprego de outros métodos estatísticos, ortogonalizando os dados e eliminando redundâncias (CHAMUNDEESWARI *et al.*, 2009; DEMŠAR *et al.*, 2013; KARAMIZADEH *et al.*, 2013).

A PCA gera combinações lineares a partir de variáveis que caracterizam os dados. Essas combinações, chamadas de componentes principais, representam variações dos dados em qualquer projeção, de modo que a primeira componente será aquela com maior variação, a segunda componente com a segunda maior variação e assim sucessivamente (DEMŠAR *et al.*, 2013; SRIVASTAVA *et al.*, 2013). Portanto, torna-se possível eliminar dados que contribuem com pouca informação e, consequentemente, obter uma aproximação com dimensão reduzida (XU *et al.*, 2010). Sabe-se que essa aproximação de dimensão reduzida é um problema chave em análise de dados e aprendizado de máquina.

A PCA pode ser estimada em termos dos mínimos quadrados, em que busca-se minimizar o erro entre um conjunto de dados e sua versão aproximada (UDELL *et al.*, 2016).

Nesse cenário, a busca pela melhor aproximação de dimensão reduzida para uma determinada matriz $\mathbf{X} \in \mathbf{R}^{n \times m}$ pode ser definida sob a norma de Frobenius ($\|\cdot\|_F$) da seguinte forma (UDELL *et al.*, 2016):

$$\begin{aligned} \min \quad & \|\mathbf{X} - \mathbf{L}\|_F^2 \\ \text{sujeito a } & \text{posto}(\mathbf{L}) \leq k \end{aligned}, \quad (2.1)$$

em que $\mathbf{L} \in \mathbf{R}^{n \times m}$ representa a aproximação de posto reduzido de \mathbf{X} , com o posto de \mathbf{L} sujeito a ser no máximo k , esse que é geralmente menor que o posto de \mathbf{X} . Nessa abordagem, a PCA pode ser vista como sendo a minimização da norma de Frobenius de um termo de ruído \mathbf{N}_0 , em que $\mathbf{X} = \mathbf{L} + \mathbf{N}_0$.

Devido ao critério quadrático apresentado na Equação 2.1, observa-se que a PCA amostral é claramente sensível à valores discrepantes, em que o seu desempenho reduz significativamente mesmo diante de poucas amostras corrompidas (FENG *et al.*, 2012). Ainda, um único *outlier*, i.e., valor atípico, pode fazer com que as componentes principais sejam distorcidas para se ajustar ao *outlier*, resultando em uma inadequada interpretação dos resultados (SERNEELS; VERDONCK, 2008). Sabendo que a PCA é sensível a ruídos, seu desempenho e sua aplicabilidade são limitados, uma vez que ruídos são normalmente introduzidos na aquisição de dados (ZHAO *et al.*, 2014).

2.2 PCA Robusta - RPCA

Dante das limitações da PCA, variantes buscando dar robustez a PCA foram surgindo de modo que a expressão RPCA foi utilizada para diferentes abordagens desse problema (JOLLIFFE; CADIMA, 2016). Em Huber (1977), Campell (1980) e Huber (1981), foram apresentadas as primeiras alternativas robustas para estimação de matrizes de covariância e correlação, as quais são utilizadas para computar as componentes principais. Uma segunda abordagem visando PCA robusta utilizou técnicas de Busca por Projeções (PP, *Projection Pursuit*) (LI; CHEN, 1985; CROUX; RUIZ-GAZEN, 1996). Conforme Hubert *et al.* (2005), essas técnicas maximizam uma medida robusta de propagação com o intuito de obter direções consecutivas nas quais os dados são projetados. Mesmo com o surgimento de diversas alternativas robustas, todas as estratégias iniciais se limitavam a dados com dimensão relativamente baixa ou com um custo computacional elevado, tornando-se, assim, técnicas limitadas em aplicações de visão computacional (HUBERT *et al.*, 2002; BOUWMANS *et al.*, 2018).

A necessidade de técnicas para lidar com conjuntos de dados maiores, bem como dados corrompidos e *outliers*, em áreas de pesquisa como processamento de imagem, aprendizado de máquina, ou bioinformática, fomentou o surgimento de variantes robustas da PCA

(JOLLIFFE; CADIMA, 2016). A ideia de decompor a matriz de dados de entrada em uma matriz de posto reduzido e uma matriz esparsa visando uma PCA robusta, tornou-se uma abordagem adequada para uma grande variedade de problemas, principalmente em aplicações de processamento de vídeo, processamento de imagem e visão computacional 3D (CHANDRASEKARAN *et al.*, 2009; BOUWMANS *et al.*, 2018).

Essa decomposição se motiva no fato de que, em muitas aplicações, a matriz de posto reduzido está associada a um padrão (e.g., rostos em aplicações de reconhecimento facial, imagem de fundo em aplicações de vídeo vigilância), enquanto que a matriz esparsa está associada a um distúrbio como, por exemplo, valores de dados corrompidos, um objeto em movimento, ou até mesmo uma pessoa no primeiro plano de imagens em aplicações de vídeo vigilância (JOLLIFFE; CADIMA, 2016). Nesse contexto, os dados de entrada $\mathbf{X} \in \mathbf{R}^{n \times m}$ são decompostos buscando satisfazer a seguinte condição:

$$\mathbf{X} = \mathbf{L} + \mathbf{S}, \quad (2.2)$$

em que $\mathbf{L} \in \mathbf{R}^{n \times m}$ representa uma versão de \mathbf{X} de posto reduzido, e $\mathbf{S} \in \mathbf{R}^{n \times m}$ representa uma matriz esparsa. Além da RPCA, essa decomposição também é usada em formulações de problemas semelhantes, como por exemplo, em métodos de Fatoração de Matrizes Não-Negativas Robustos (RNNMF, *Robust Non-Negative Matrix Factorization*), Preenchimento de Matrizes Robustos (RMC, *Robust Matrix Completion*), Recuperação de Subespaço Robusto (RSR, *Robust Subspace Recovery*), Rastreamento de Subespaço Robusto (RST, *Robust Subspace Tracking*) e Minimização de Posto Reduzido Robusta (RLRM, *Robust Low-Rank Minimization*) (BOUWMANS *et al.*, 2017).

Essa decomposição do problema sugere uma solução que encontre o posto mais baixo possível que os dados possam assumir, sujeito a uma restrição de que o conteúdo em \mathbf{S} seja esparsa. A formulação idealizada desse problema é dada por (WRIGHT *et al.*, 2009):

$$\begin{aligned} & \min \text{posto}(\mathbf{L}) + \gamma \|\mathbf{S}\|_0 \\ & \text{sujeito a } \mathbf{L} + \mathbf{S} = \mathbf{X} \end{aligned}, \quad (2.3)$$

em que $\text{posto}(\mathbf{L})$ é o posto da matriz \mathbf{L} , $\|\mathbf{S}\|_0$ representa uma medida de cardinalidade da matriz esparsa, ou seja, uma medida que encontre o número de valores não nulos presentes na matriz esparsa e γ é um parâmetro que estabelece uma relação de compromisso (*trade-off*) entre a matriz de posto reduzido e os *outliers* esparsos. No entanto, essa reformulação além de ser não-convexa, apresenta soluções exatas que requerem um tempo duplamente exponencial na dimensão n da matriz, ou seja, soluções intratáveis em grandes bases de dados (CANDÈS; RECHT, 2009).

Diversas abordagens foram desenvolvidas buscando encontrar as matrizes \mathbf{L} e \mathbf{S} sob certas condições de uma maneira eficiente (tratável). Essas estratégias diferem-se entre si

nas medidas de erro, no problema de otimização ou até mesmo nos métodos de solução usados (BOUWMANS *et al.*, 2018). A formulação idealizada, mais difundida e considerada como *baseline* para muitos outros trabalhos, foi proposta por Candès *et al.* (2011). Candès *et al.* (2011) garantem, sob hipóteses fracas, que uma matriz de posto reduzido \mathbf{L} pode ser recuperada perfeitamente mesmo com medições altamente corrompidas ($\mathbf{X} = \mathbf{L} + \mathbf{S}$), a partir de um problema de otimização convexa, conhecido como PCP.

É válido ressaltar que, além dessa formulação clássica proposta por Candès *et al.* (2011), outras formulações foram propostas. Xu *et al.* (2012) propuseram uma RPCA via Busca de *Outliers* (RPCA-OP, *Outlier Pursuit*), visando recuperar \mathbf{L} e \mathbf{S} quando entradas corrompem colunas inteiras nos dados como, por exemplo, em aplicações financeiras e de bioinformática. Em Goes *et al.* (2014), pode-se encontrar uma RPCA via otimização estocástica. Os autores em Guyon *et al.* (2012a) e em Guyon *et al.* (2012b) propõem um esquema iterativo, baseado no Método dos Mínimos Quadrados Reponderados Iterativamente (IRLS, *Iteratively Reweighted Least Squares*), para resolver o problema da RPCA com uma convergência rápida e com um baixo custo computacional. Além disso, existem trabalhos que visam estimar \mathbf{L} e \mathbf{S} de forma eficiente e robusta, decompondo \mathbf{X} em, aproximadamente, $\mathbf{L} + \mathbf{S} + \mathbf{E}$, em que \mathbf{E} representa um termo de ruído gaussiano (ZHOU *et al.*, 2010; ZHOU; TAO, 2011). Métodos Bayseanos também são utilizados para inferir uma representação aproximada nas estatísticas de um termo de ruído \mathbf{E} , enquanto estimam-se, simultaneamente, as matrizes \mathbf{L} e \mathbf{S} (DING *et al.*, 2011; BABACAN *et al.*, 2012). Mais detalhes sobre estratégias para resolver esse problema são apresentadas em Bouwmans *et al.* (2017). Dentre as metodologias propostas, nenhuma delas produz um algoritmo de tempo polinomial com fortes garantias de desempenho como a RPCA via PCP (ZHOU *et al.*, 2010; CANDÈS *et al.*, 2011). Nessa dissertação de mestrado, a RPCA via PCP é apresentada de forma mais detalhada.

2.3 RPCA via Busca de Componentes Principais (PCP)

Conforme mencionado anteriormente, a RPCA via PCP oferece uma separação cega de dados na forma: $\mathbf{L} + \mathbf{S}$. Existem muitas aplicações importantes em que os dados podem ser modelados naturalmente dessa forma. Dentre elas, pode-se citar a área de processamento de sinais, incluindo problemas na área de processamento de imagens SAR, de imagens médicas (e.g., imagens cerebrais, cardíacas e até retiniais). Além disso, essa decomposição também pode ser utilizada como uma ferramenta em vídeo vigilância, seja para separar o primeiro ou o segundo plano da imagem, detectar alvos em movimento, codificar vídeo e rastrear objetos visuais (BOUWMANS *et al.*, 2018).

Um exemplo de aplicação da RPCA via PCP em vídeo vigilância pode ser encontrado

na Figura 2.1. Nesse exemplo, a sequência de *background* é modelada por um subespaço de posto reduzido que pode mudar gradualmente ao longo do tempo, enquanto que os objetos de primeiro plano em movimento constituem os *outliers* esparsos (BOUWMANS; ZAHZAH, 2014). Portanto, a RPCA via PCP mostra ser uma ferramenta adequada para separar objetos em movimento do fundo da imagem (BOUWMANS *et al.*, 2018).

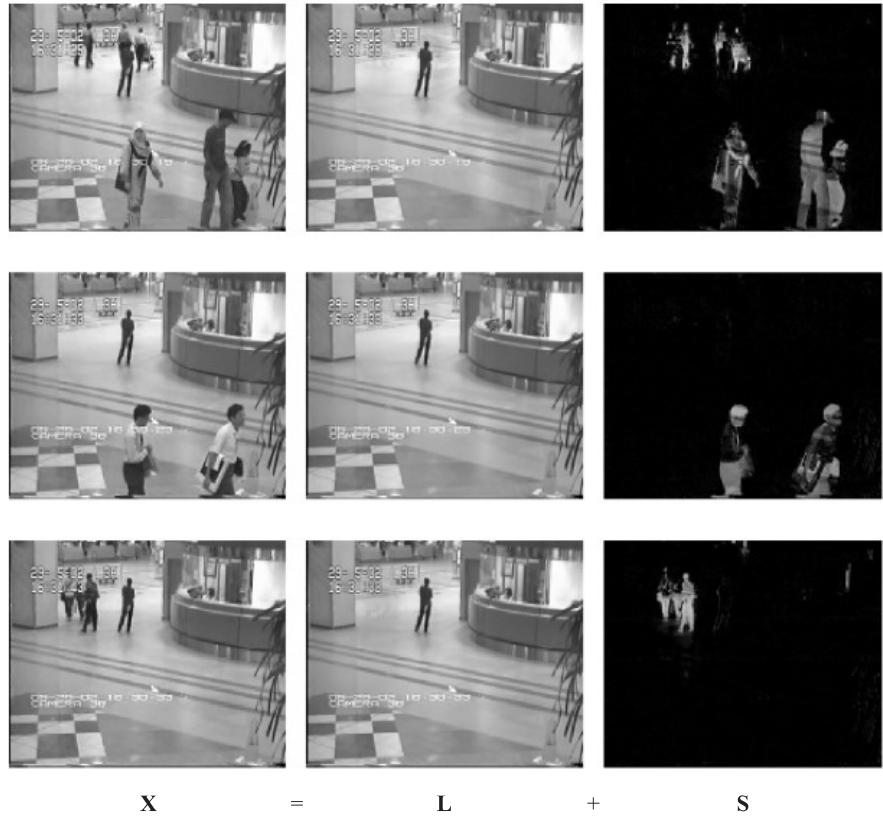


FIGURA 2.1 – RPCA via PCP em aplicações de vídeo vigilância (CANDÈS *et al.*, 2011).

Inicialmente, pode-se presumir que este problema seja impossível de ser resolvido, uma vez que o número de incógnitas a inferir em \mathbf{L} e \mathbf{S} é o dobro de incógnitas em \mathbf{X} . Ainda, é mais difícil pensar que possamos recuperar de forma confiável a matriz de posto reduzido \mathbf{L} com erros em \mathbf{S} de valores arbitrariamente grande (CANDÈS *et al.*, 2011). No entanto, conforme mencionado na Seção 2.2, Candès *et al.* (2011) apresentaram um problema de otimização convexa tratável, em que é possível recuperar exatamente as matrizes \mathbf{L} e \mathbf{S} sem conhecimento prévio da matriz de posto reduzido e do padrão de dispersão da parte esparsa. Assim, esse problema conhecido como PCP busca resolver:

$$\begin{aligned} & \min \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \\ & \text{sujeito a } \mathbf{L} + \mathbf{S} = \mathbf{X} \end{aligned}, \quad (2.4)$$

em que $\|\mathbf{L}\|_* = \sum_i \sigma_i(\mathbf{X})$ representa a norma nuclear de \mathbf{L} (i.e., a soma dos valores singulares, σ_i) como uma aproximação de posto reduzido, $\|\mathbf{S}\|_1 = \sum_{ij} |\mathbf{X}_{ij}|$ representa a norma

l_1 (i.e., a soma das componentes de todas as entradas, em absoluto) como uma heurística para encontrar representações esparsas — também conhecida como *Basis Pursuit* — e $\lambda > 0$ é um parâmetro de regularização que equilibra ambos os termos (CANDÈS *et al.*, 2011).

Conforme descrito na Equação 2.4, a PCP busca, por meio do processo de otimização, convergir para uma decomposição que minimize a combinação ponderada entre a norma nuclear e a norma l_1 , a um custo computacional não muito superior ao do PCA clássico. Através desse problema convexo, Candès e Recht (2009) demonstraram que \mathbf{L} e \mathbf{S} podem ser recuperados perfeitamente, desde que o posto de \mathbf{L} não seja muito grande e que \mathbf{S} seja razoavelmente esparso, respeitando a seguinte desigualdade:

$$\text{posto}(\mathbf{L}) \leq \frac{\rho_r \max(n, m)}{\mu(\log \min(n, m))^2}, \quad \|\mathbf{S}\|_0 \leq \rho_s mn, \quad (2.5)$$

em que $\rho_r > 0$ e $\rho_s > 0$ são constantes numéricas e $\mu > 0$ é um parâmetro constante das condições de incoerência impostas à Decomposição em Valores Singulares (SVD, *Singular Value Decomposition*) para considerar \mathbf{L} como um conteúdo não esparso e, consequentemente, conseguir separar \mathbf{L} e \mathbf{S} .

A separação do conteúdo de \mathbf{L} para \mathbf{S} é realizada durante as iterações do algoritmo, até atingir uma determinada medida de esparsidade para \mathbf{S} . De acordo com a Equação 2.4, o valor de esparsidade a ser atingido depende da escolha de λ , de modo que baixos valores resultam em mais informação em \mathbf{S} . Conforme Candès *et al.* (2011), um valor de λ , adotado para diversas aplicações, pode ser encontrado da seguinte forma:

$$\lambda = \frac{1}{\sqrt{\max(n, m)}}. \quad (2.6)$$

É interessante salientar que esta escolha de λ segue como uma sugestão. Dessa forma, com um conhecimento prévio sobre a solução, aumentar ou diminuir λ em muitos casos pode apresentar um melhor resultado (CANDÈS *et al.*, 2011).

Existem inúmeras abordagens possíveis para resolver a Equação 2.4 (PCP). Em um cenário onde o problema pode ter uma solução fechada, a PCP pode ser reformulada por meio de uma Programação Semidefinida (SDP, *Semidefinite Programming*) e resolvida por Métodos de Ponto Interior (*Interior-Point Methods*). Apesar desses métodos normalmente apresentarem poucas iterações para convergirem, a utilização em matrizes grandes pode ser inviável, uma vez que sua complexidade computacional a cada iteração é de $O((nm \min(n, m))^2)$ (LIN *et al.*, 2010; BOUWMANS; ZAHZAH, 2014).

Para superar o problema apresentado pelos métodos de ponto inferior, outros métodos foram propostos utilizando informações de primeira ordem da função objetivo. Nesse

cenário, Cai *et al.* (2010) propuseram uma técnica conhecida como Limiar de Valores Singulares (SVT, *Singular Value Thresholding*) para minimizar a norma nuclear em problemas de Preenchimento de Matrizes (MC, *Matrix Completation*). Em frente a necessidade da PCP em minimizar a norma nuclear e a norma l_1 , Wright *et al.* (2009) apresentaram um algoritmo conhecido como Limiar Iterativo (IT, *Iterative Threshold*). No entanto, essa técnica, apesar de ser extremamente simples e comprovadamente correta, requer um grande número de interações e converge muito lentamente para a solução. Portanto, o seu uso é limitado (LIN *et al.*, 2010; BOUWMANS; ZAHZAH, 2014).

Para contornar o problema de convergência apresentado no algoritmo IT, dois algoritmos foram propostos por Lin *et al.* (2009) para resolver o problema dual da PCP. Esses algoritmos baseiam-se na otimização por gradiente, sendo um deles conhecido como Gradiente Próximo Acelerado (APG, *Accelerated Proximal Gradient*) e outro, como Gradiente Ascendente (*Gradient-ascent*). No entanto, como esses algoritmos apresentam, teoricamente, apenas taxa de convergência sublinear, esses ainda são considerados lentos para aplicações reais (BOUWMANS; ZAHZAH, 2014). É importante salientar que, em processos de otimização convexa, as convergências podem ser classificadas na ordem da mais lenta para mais rápida como: sublinear, linear, superlinear, Q-quadrática, Q-cúbica, e assim sucessivamente (NOCEDAL; WRIGHT, 1999).

Lin *et al.* (2010) apresentaram dois algoritmos escalonáveis e rápidos para resolver o problema de otimização convexa da PCP, utilizando o método dos Multiplicadores de Lagrange Aumentado (ALM, *Augmented Lagrange Multipliers*). Esse método funciona de forma estável e eficiente em diferentes problemas, apresentando uma precisão muito maior que o algoritmo APG, sem ajuste de parâmetros e com uma taxa de convergência linear (CANDÈS *et al.*, 2011; BOUWMANS; ZAHZAH, 2014).

Em geral, os métodos ALM são utilizados para resolver problemas de otimização com restrições gerais. Para isso, o método ALM busca substituir o problema restrito original por uma sequência de subproblemas sem restrições ou com restrições simples, que podem ser simplesmente resolvidos alcançando soluções exatas ou aproximadas (BIRGIN; MARTÍNEZ, 2008; DEB; SRIVASTAVA, 2012). Para a RPCA via PCP, a função Lagrangiana aumentada (L) é definida da seguinte maneira:

$$L(\mathbf{L}, \mathbf{S}, \Lambda, \mu) \doteq \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 + \langle \Lambda, \mathbf{X} - \mathbf{L} - \mathbf{S} \rangle + \frac{\mu}{2} \|\mathbf{X} - \mathbf{L} - \mathbf{S}\|_F^2, \quad (2.7)$$

em que Λ representa a matriz multiplicadora de Lagrange, e $\langle \cdot, \cdot \rangle$ denota o produto interno. A cada iteração, o método ALM procura resolver o seguinte subproblema:

$$\begin{cases} (\mathbf{L}^{(t+1)}, \mathbf{S}^{(t+1)}) = \arg \min_{\mathbf{L}, \mathbf{S}} L(\mathbf{L}, \mathbf{S}, \Lambda^{(t)}), \\ \Lambda^{(t+1)} = \Lambda^{(t)} + \mu (\mathbf{L}^{(t+1)} + \mathbf{S}^{(t+1)} - \mathbf{X}) . \end{cases} \quad (2.8)$$

Note que a abordagem ALM é baseada em uma única etapa, o que significa que a otimização é realizada tratando as variáveis \mathbf{L} e \mathbf{S} em conjunto. Assim, a aplicação direta do método ALM considera a Equação 2.4 como um problema genérico e ignora sua estrutura separável (YUAN; YANG, 2013; BOUWMANS; ZAHZAH, 2014).

Essa forma de tratar o problema pode ser contornada pelo Método de Direção Alternada de Multiplicadores (ADMM, *Alternating Direction Method of Multipliers*) (YUAN; YANG, 2013). O método ADMM divide a tarefa de minimização apresentada na Equação 2.8 em dois subproblemas menores e mais simples, em que \mathbf{L} e \mathbf{S} são minimizados alternadamente. Logo, o método ADMM busca resolver os seguintes subproblemas (BOYD *et al.*, 2011):

$$\begin{cases} \mathbf{L}^{(t+1)} = \arg \min_{\mathbf{L}} L(\mathbf{L}, \mathbf{S}^{(t)}, \Lambda^{(t)}), \\ \mathbf{S}^{(t+1)} = \arg \min_{\mathbf{S}} L(\mathbf{L}^{(t+1)}, \mathbf{S}, \Lambda^{(t)}), \\ \Lambda^{(t+1)} = \Lambda^{(t)} + \mu (\mathbf{L}^{(t+1)} + \mathbf{S}^{(t+1)} - \mathbf{X}). \end{cases} \quad (2.9)$$

O método ADMM pode ser escrito de uma forma mais conveniente, combinando os termos lineares e quadráticos da função Lagrangiana aumentada. Assim, pode-se reescrever a Equação 2.9 como (YUAN; YANG, 2013; BOYD *et al.*, 2011):

$$\begin{cases} \mathbf{L}^{(t+1)} = \arg \min_{\mathbf{L}} \left\{ \|\mathbf{L}\|_* + \frac{\mu}{2} \|\mathbf{L} + \mathbf{S}^{(t)} - \mathbf{X} + \mu^{-1} \Lambda^{(t)}\|_F^2 \right\}, \\ \mathbf{S}^{(t+1)} = \arg \min_{\mathbf{S}} \left\{ \|\mathbf{S}\|_1 + \frac{\mu}{2} \|\mathbf{L}^{(t+1)} + \mathbf{S} - \mathbf{X} + \mu^{-1} \Lambda^{(t)}\|_F^2 \right\}, \\ \Lambda^{(t+1)} = \Lambda^{(t)} + \mu (\mathbf{L}^{(t+1)} + \mathbf{S}^{(t+1)} - \mathbf{X}). \end{cases} \quad (2.10)$$

Um pseudocódigo que descreve as etapas de execução do algoritmo é apresentado no Algoritmo 1. Para fins de implementação, um critério de parada para a convergência pode ser adotado, como por exemplo, um número específico de iterações ou um número de erros com determinada tolerância.

Inicialmente, a ideia do método ADMM é minimizar a função Lagrangeana com relação a \mathbf{L} enquanto mantém \mathbf{S} fixo. Em seguida, o método minimiza \mathbf{S} com \mathbf{L} fixo, e por fim atualiza Λ , conforme o resíduo ($\mathbf{X} - \mathbf{L} - \mathbf{S}$). Note que as etapas alternadas podem ser resolvidas simplesmente definindo o operador de retração ($\mathcal{S}_{1/\mu}$) e o operador de retração para valores singulares ($\mathcal{D}_{1/\mu}$), da seguinte maneira (CAI *et al.*, 2010; CANDÈS *et al.*, 2011):

$$\mathcal{S}_{1/\mu}(x) = \text{sgn}(x) \max(|x| - 1/\mu, 0), \quad (2.11)$$

$$\mathcal{D}_{1/\mu}(X) = U \mathcal{S}_{1/\mu}(\Sigma) V^*, \text{ com } X = U \Sigma V^*. \quad (2.12)$$

Algoritmo 1: Etapas do método ADMM para resolver a PCP (BOYD *et al.*, 2011).

Entrada: $\mathbf{S}^{(0)} = \Lambda^{(0)} = 0$, $\mu > 0$

Saída: \mathbf{L} , \mathbf{S} .

início

repita

Passo 1. Gerar $\mathbf{L}^{(t+1)}$:

$$\mathbf{L}_{(t+1)} = \mathcal{D}_{1/\mu} (\mathbf{X} - \mathbf{S}^{(t)} + \mu^{-1} \Lambda^{(t)}) ;$$

Passo 2. Gerar $\mathbf{S}^{(t+1)}$:

$$\mathbf{S}^{(t+1)} = \mathcal{S}_{\lambda/\mu} (\mathbf{X} - \mathbf{L}^{(t+1)} + \mu^{-1} \Lambda^{(t)}) ;$$

Passo 3. Atualizar $\Lambda^{(t+1)}$:

$$\Lambda^{(t+1)} = \Lambda^{(t)} + \mu (\mathbf{S}^{(t+1)} + \mathbf{L}^{(t+1)} - \mathbf{X}) ;$$

até convergir para um erro aproximadamente igual a zero, com uma tolerância;

fim

Por fim, é possível encontrar outras estratégias muito semelhantes para desenvolver algoritmos ALM de forma simples para o problema da PCP. O trabalho de Yang e Yuan (2013) propõem linearizar os subproblemas dos métodos ALM e ADMM para quando esses não tiverem uma solução fechada. Além disso, Lin *et al.* (2011) propõem melhorar a convergência e reduzir a complexidade do método ADMM linearizado por meio de uma penalidade adaptativa.

Todos esses métodos requerem a SVD como ferramenta de fatoração de matrizes. Sua computação geralmente resulta em uma complexidade $\mathcal{O}(M \times N \times \min(M, N))$. Nesse cenário, alguns estudos adotam o uso de SVDs parciais, visando reduzir a complexidade de cada interação para $\mathcal{O}(kMN)$, em que k é geralmente menor do que N e M . No entanto, tal complexidade ainda é considerada alta para grandes conjuntos de dados. Dessa forma, pesquisas recentes na área visam reduzir essa complexidade, evitando o cálculo de SVDs. Como exemplo, Liu *et al.* (2011) por meio de um algoritmo chamado de l_1 filtering, apresentaram um método com uma complexidade $\mathcal{O}(k^2(M + N))$, o qual é altamente paralelizável. Esse método é considerado o primeiro algoritmo que pode resolver o problema de minimização da norma nuclear em tempo linear. Além desse método, outras estratégias podem ser encontradas em (BOUWMANS; ZAHZAH, 2014).

2.4 Considerações Finais

Considerando a relativa estabilidade de *background* do sistema SAR UWB VHF, conforme apresentado no Capítulo 3, espera-se que o *background* da imagem possa ser modelado em \mathbf{L} , enquanto que o conteúdo de natureza esparsa nas imagens (i.e., alvos) possa

ser apresentado em \mathbf{S} . Uma vez que nessa dissertação têm-se o interesse em detectar alvos encontrado em uma certa imagem de interesse, é plausível decompor a matriz de dados em $\mathbf{X} = \mathbf{L} + \mathbf{S}$. Desse modo, a RPCA via PCP (CANDÈS *et al.*, 2011) por ser o método clássico, mais representativo e por adequar-se ao contexto de aplicação, foi a formulação considerada para tratar o problema discutido nessa dissertação. Por fim, por ser um método de implementação relativamente simples e apresentar um bom desempenho em diversos problemas de otimização, utilizou-se o algoritmo ADMM para resolver a PCP.