



Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Matemática

Algoritmo de fatoração QR para geração de animações faciais

por

Elenilson de Vargas Fortes

Brasília
2007

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Matemática

Algoritmo de fatoração QR para geração de animações faciais

por

Elenilson de Vargas Fortes*

Dissertação apresentada ao Departamento de Matemática da Universidade de Brasília, como parte dos requisitos para obtenção do grau de

MESTRE EM MATEMÁTICA

Brasília, 13 de Dezembro de 2007

Comissão Examinadora:

Prof. Dr. Jorge Carlos Lucero - MAT/UnB (Orientador)

Prof. Dr. Carlos Maber Carrión Riveros - MAT/UnB
(Membro)

Prof. Dr. Pledson Guedes de Medeiros - EST/UFRN
(Membro)

*Este trabalho contou com apoio financeiro parcial do CNPq.

“Somos o que pensamos. Tudo o que somos surge com nossos pensamentos. Com nossos pensamentos, fazemos o nosso mundo”, (Buda).

Aos meus pais

Jonas de Vargas Fortes e Ozilia Loureti Fortes

Agradecimentos

À Deus pela vida e sabedoria concedida ao longo da minha caminhada estudantil.

Aos meus pais, Jonas de Vargas Fortes e Ozilia Loureti Fortes, aos meus queridos irmãos, Emerson, Jonas (Junior), Jaqueline e Janielli.

Agradeço ao povo brasileiro, que através do pagamento de impostos permitiu ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), financiar esta pesquisa.

Ao meu Orientador, Dr. Jorge Carlos Lucero, pela orientação e paciência que teve durante a elaboração deste trabalho.

Aos professores da Universidade Federal do Espírito Santo (UFES): Aldo Vignatti, Alex sander, Eder Machado, Gilvan, Jamil, Joccitiel, Domingos, Ana Claudia e Rocha, pelos conselhos e incentivos.

Aos professores da banca examinadora Dr. Pledson Guedes de Medeiros e Dr. Carlos Maber Carrión Riveros.

Aos colegas de graduação que ainda mantenho contato: Meu grande amigo

Edinelço Dalcumune, Wagner, Riedson, Sérgio e Vanessa. Aos colegas do curso de verão que foram muito importantes para que eu tomasse a decisão de vir para Brasília. E a todos os meus colegas do Departamento de Matemática da Universidade de Brasília que sempre me apoiaram nos bons e maus momentos. Alguns em especial como Hailton pelo suporte técnico do site que criamos, Susanne, Ricardo, Enio, Sérgio, Luciana, Igor, Euro, Evander, Nilton, Walter e claro, Eliane Ferreira, pelo carinho e atenção a mim concedidos.

Aos professores do Departamento de Matemática da Universidade de Brasília (UnB), Dr. Helmar Numes Moreira e em especial Dr. Angel Rodolfo Baigorri, pelos conselhos e conversas que foram de grande valia durante este período.

Aos Professores do Ensino Fundamental e Médio que de alguma forma acabaram influenciando-me nesta conquista, em especial Lenice, Cecília e Lena.

À todos que, de alguma forma alimentaram meus sonhos e contribuíram para esta grande conquista de minha vida. Obrigado!

Resumo

Nesta dissertação, consideramos o problema da seleção de um subconjunto de colunas independentes de uma matriz de dados, e sua resolução por meio da fatoração QR com pivoteamento de colunas, (Lucero et al., [12]). Mostraremos como este problema pode ser aplicado à identificação de padrões de deformação facial durante a fala, para a construção de um modelo empírico da cinemática facial. O modelo pode ser utilizado para a geração de animações da fala, sob controle de sinais colhidos experimentalmente.

Palavras-chaves: Seleção de Subconjunto, Fatoração QR, Animação Facial.

Abstract

In this work, we study the problem of the selection of a subset of independent columns in matrix of data, and this resolution through the factorization QR with pivoted of columns, (Lucero et al., [12]). We will show how this problem can be applied to the identification of patterns of facial deformation during speech, for the construction of an empiric model of the facial kinematics. The model can be used for generating speech animations, if we control the signals choosen experimentally.

Key Words: Subset Selection, QR Factorization , Facial Animation.

Sumário

Introdução	1
1 Conceitos Básicos e Decomposição em Valores Singulares	5
1.1 Introdução	5
1.2 Norma de Vetores	5
1.3 Norma de Matrizes	7
1.4 Imagem, Espaço Nulo e Posto	9
1.5 Ortogonalidade	11
1.6 Decomposição em Valores Singulares	12
1.7 Algumas Propriedades da SVD	14
1.7.1 SVD e sua Relação com Normas	14
1.7.2 Relações entre SVD e o Posto de uma Matriz	16
1.7.3 Posto Numérico	19
1.8 Análise e Discussão	21
2 A Fatoração QR	24
2.1 Introdução	24
2.2 Definição e Propriedades da Fatoração QR	24
2.3 Reflexão de Householder	27
2.3.1 Reflexão	27
2.3.2 Definição e Propriedades da Reflexão de Householder	30
2.3.3 Escolha do Vetor de Reflexão	31
2.4 Cálculo da Fatoração QR	33
2.5 Relações da Fatoração QR com o Posto de uma Matriz	34
2.5.1 Unicidade da Fatoração QR	35
2.5.2 Matriz de Permutação	35

2.5.3	Fatoração QR com Pivoteamento de Colunas	36
3	Aplicações da Fatoração QR	39
3.1	Introdução	39
3.2	Sistemas Lineares	40
3.3	Problema de Mínimos Quadrados	41
3.3.1	Matrizes de Posto Completo	42
3.3.2	Matrizes de Posto Incompleto	46
3.4	O Problema da Seleção de Subconjunto	48
3.4.1	Propriedades da Matriz A_{II}	48
3.4.2	Seleção de Subconjunto	50
3.4.3	Seleção de Subconjunto e Mínimos Quadrados	51
3.5	Discussão	54
4	Aplicação a Animação Facial	55
4.1	Introdução	55
4.2	Dados	55
4.3	Pré-processamento	56
4.4	Posto Numérico da Matriz de Dados	57
4.5	Alguns Resultados da Fatoração QR	59
4.6	Escolha dos Marcadores Principais e suas Respektivas Regiões de In- fluência	61
4.7	Geração de Animações Faciais	66
4.8	Análise de Erros nas Trajetórias Computadas	69
4.9	Análise para Seleção de Marcadores	72
	Conclusões	76
	Referências Bibliográficas	78
	Apêndice A	81
	Algoritmo A	81
	Algoritmo B	83
	Algoritmo C	85
	Algoritmo D	86
	Algoritmo E	87

Algoritmo F	88
Algoritmo G	89
Algoritmo H	90
Algoritmo I	93
Algoritmo J	95
Algoritmo K	98
Algoritmo L	100
Apêndice B	104
Tabela 4.6	104
Anexos	106
Teorema 2.6	106
Teorema 2.7	109
Teorema 2.9	111
Teorema 2.10	112
Teorema 2.13	114

Lista de Figuras

2.1	<i>Reflexão em torno da reta ξ.</i>	28
2.2	<i>Possibilidades de Reflexão.</i>	31
3.1	<i>Propriedades da Matriz A_{II}.</i>	49
4.1	<i>Posição dos marcadores faciais.</i>	56
4.2	<i>Elementos da diagonal da Matriz Σ.</i>	58
4.3	<i>Elementos da diagonal da Matriz R.</i>	59
4.4	<i>Valores normalizados dos 12 primeiros elementos na diagonal de R em função da quantidade de amostras nos dados.</i>	60
4.5	<i>Valores normalizados dos 12 primeiros elementos na diagonal de R em função da quantidade de amostras nos dados permutados aleatoriamente.</i>	60
4.6	<i>Regiões faciais ilustradas para os marcadores 40, 34, 38, 02, 36 e 06.</i>	65
4.7	<i>Regiões faciais ilustradas para marcadores 20, 49, 13 e 52.</i>	66
4.8	<i>Regiões de deformação facial nas direções ortogonais, linhas 1 a 6 de R.</i>	67
4.9	<i>Regiões de deformação facial nas direções ortogonais, linhas 7 a 10 de R.</i>	68
4.10	<i>Quadro inicial das animações.</i>	69
4.11	<i>Exemplo de trajetória real (linha de traços) e reconstruída pelo algoritmo (linha cheia), para o marcador 28.</i>	70
4.12	<i>Erro médio para um conjunto de k marcadores selecionados, frases 31 e 32. A linha cheia (na horizontal) representa o erro de precisão dos dados.</i>	73
4.13	<i>Erro médio para um conjunto de k marcadores selecionados, frases 33 a 36. A linha cheia (na horizontal) representa o erro de precisão dos dados.</i>	74

4.14 *Erro médio para um conjunto de k marcadores selecionados, para as frases 37 a 40. A linha cheia (na horizontal) representa o erro de precisão dos dados.* 75

Lista de Tabelas

4.1	<i>Os 12 primeiros marcadores selecionados pelo algoritmo com k linhas na matriz de dados.</i>	62
4.2	<i>Os 12 primeiros marcadores selecionados pelo Algoritmo com k linhas permutadas aleatoriamente.</i>	62
4.3	<i>Os 12 primeiros marcadores selecionados pelo algoritmo.</i>	63
4.4	<i>Erro obtido para as trajetórias dos marcadores que foram reconstituídas através algoritmo para a sentença 39.</i>	71
4.5	<i>Erro médio para trajetórias reconstituídas pelo algoritmo para as sentenças 31 a 40.</i>	72
4.6	<i>Conjunto de 40 sentenças em inglês: As 30 primeiras frases foram usadas para construir o modelo e as 10 últimas para realizar testes e validar o modelo.</i>	105

Introdução

Alguns problemas da Álgebra Linear, tais como resolução de sistemas lineares, mínimos quadrados e seleção de subconjunto (“*subset selection problem*”) podem ser convenientemente tratados por meio da fatoração QR.

A fatoração QR é um método de decomposição em que a matriz $A \in \mathbb{R}^{m \times n}$ é decomposta em um produto $A = QR$, onde $Q \in \mathbb{R}^{m \times m}$ e $R \in \mathbb{R}^{m \times n}$ são matrizes ortogonal e triangular superior, respectivamente.

Existem outros métodos para resolver os problemas acima. Por exemplo, para o problema de mínimos quadrados, podem ser usadas as equações normais (ver [7, 20]). Os sistemas lineares podem ser solucionados através da fatoração LU ou eliminação de Gauss (ver [10]).

Focalizaremos, doravante, no tema de *seleção de subconjuntos* e suas variações. Suponha que temos uma matriz de dados $A \in \mathbb{R}^{m \times n}$ com $m > n$, a partir da qual queremos prever um vetor de observações $b \in \mathbb{R}^{m \times 1}$, isto é, queremos encontrar um vetor x que minimize $\|Ax - b\|_2$. No entanto, ao invés de usar todas as colunas da matriz de dados A , desejamos prever b a partir de apenas um subconjunto de k colunas, eliminando aquelas que sejam redundantes e possam ser desconsideradas (por exemplo, colunas linearmente dependentes, [7]). Veremos que a seleção de dito subconjunto de colunas não-redundantes de A é feita através da fatoração QR da matriz A com pivoteamento de colunas, (ver [7, 6, 12]).

Mostraremos que a seleção de subconjunto pode ser aplicado a geração de animações faciais, onde queremos identificar um subconjunto de marcadores independentes para ser utilizado posteriormente, como base para prever o movimento de pontos faciais arbitrários, obtendo dessa forma geração de animações, para ser utilizado como ferramenta computacional em estudos sobre percepção e produção da fala. Os estudos baseiam-se principalmente em Lucero et al., (ver [12]).

Em artigo recente, a seleção de subconjuntos foi examinado por Hoog e Mattheij (ver [8]), onde consideram uma matriz $A \in \mathbb{R}^{m \times n}$ em que se deseja eliminar $m - k$ linhas de A de tal forma que a matriz resultante $B \in \mathbb{R}^{k \times n}$ possua um subconjunto de linhas linearmente independentes, isto é, para $A \in \mathbb{R}^{m \times n}$ encontrar uma matriz de permutação $\Pi \in \mathbb{R}^{m \times m}$ tal que

$$\Pi A = \begin{pmatrix} B \\ C \end{pmatrix}, \quad (1)$$

sendo $B \in \mathbb{R}^{k \times n}$ a matriz pretendida. Neste artigo, os autores mostram uma técnica de seleção de linhas baseado na norma de Frobenius e na Pseudo-Inversa (ver [7]).

Outros métodos como os algoritmos de tipo Backward Greedy (ver [2]) podem ser utilizados para abordar o problema de seleção de subconjunto, cuja essência consiste na determinação reiterada, de $\rho = \min_{x \in \mathbb{R}^n} \|Ax - b\|_2$ para uma dada matriz $A \in \mathbb{R}^{m \times n}$ e um vetor de observações $b \in \mathbb{R}^n$.

Especificamente, retira-se uma coluna de A e calcula-se $\min_{x_1 \in \mathbb{R}^{n-1}} \|A_1 x_1 - b\|_2$. Reinserida esta coluna na matriz A novamente retiramos outra coluna e calcula-se $\min_{x_1 \in \mathbb{R}^{n-1}} \|A_1 x_1 - b\|_2$. Repete-se este processo para todas as colunas de A e elimina-se a coluna que forneça $\min_{x_1 \in \mathbb{R}^{n-1}} \|A_1 x_1 - b\|_2$ menor possível, onde $A_1 \in \mathbb{R}^{m \times (n-1)}$. Em cada etapa do algoritmo, elimina-se uma coluna de A , sempre seguindo os passos para eliminação da primeira coluna. Desta forma, o algoritmo elimina $n - r$ colunas de A e portanto, tem-se uma matriz $A_r \in \mathbb{R}^{m \times r}$, ou seja, o algoritmo seleciona um subconjunto de r colunas de A .

Do mesmo modo pode ser utilizadas para examinar o problema de seleção de subconjunto, diferente técnica (ver [7]) denominada decomposição de valores singulares (SVD), que adiaremos temporariamente e prontamente será apresentada no Capítulo 1.

Em outro artigo (ver [18]), a seleção de subconjunto é feita usando recursos estatísticos e algébricos. Inicia-se com a aplicação de scanner a laser a 8 expressões faciais estáticas pré-determinadas originando igual número de matrizes com 71.900 pixels e 141.900 polígonos, que seguidamente são adaptadas a uma malha facial deformável genérica constituída por 576 nodos e 844 polígonos. A partir dessas 8 malhas faciais são gerados 8 (oito) vetores de 3×576 componentes, que representam

as coordenadas espaciais $\{x, y, z\}$ de cada um dos nodos e constituem os vetores colunas da matriz de dados aleatórios $A \in \mathbb{R}^{m \times n}$, onde $m = 3 \times 576$ e $n = 8$. Após calculada a matriz de covariância e via SVD obtem-se as componentes principais (PCA) dos vetores colunas de A dos quais seleciona-se o subconjunto dos mais representativos no que se refere à variância dos dados. Note-se que as componentes principais são os autovetores de A e que pelo significado decorrentes do modelo representam rostos discretizados na forma de malhas facias, o que os leva a serem chamados de autofaces (“*eigenface*”). No presente caso, gostaríamos de obter um modelo em termos de pouco marcadores faciais, ao invés de vetores que representam imagem da face.

Outra alternativa interessante, foi proposta pelo trabalho de modelagem articulatória de Badin et al., (ver [4]). Neste artigo, o PCA é usado para determinar parâmetros articulatorios para controlar a forma de uma região vocal em 3D. Para uma melhor relação à biomecânica subjacente, alguns dos parâmetros (por exemplo, altura do maxilar, etc.) são definidos a priori, e suas contribuições são subtraídas dos dados antes de computar os componentes restantes. Em nosso trabalho, nós propomos confiar inteiramente nos dados para predizer o comportamento dinâmico da face, com poucas suposições prévias como possível.

Retornando à decomposição QR com pivoteamento de colunas como procedimento de seleção de subconjunto, é mister comentar algumas recentes aplicações. Foi aplicada à matriz $A \in \mathbb{R}^{m \times n}$, com $m < n$ cujas colunas se correspondem com a sequência finita de profundidades do nível da água medida em um dado poço, em intervalos arbitrários de tempo. O objetivo deste estudo foi a identificação de subconjunto de poços independentes para posteriormente serem utilizados como base para a predição do deslocamento do nível de água em poços aleatoriamente escolhidos (ver [16]).

Como visto, existem várias técnicas para se resolver o problema de seleção de subconjunto e aplicações. Conforme mostraremos, o algoritmo de fatoração QR com pivoteamento de colunas permite identificar um subconjunto de marcadores faciais independentes, e desta forma explorar sua aplicação à construção de modelos da biomecânica facial (ver [12]) e conseqüentemente, movimentos arbitrários e animações da fala podem ser logo geradas, controlando esse modelo com sinais colhidos experimentalmente (animação facial “*data driven*”).

Os estudos sobre animações faciais deste presente trabalho está baseado em Lucero et al. (ver [12]) e é resultado da análise de pesquisas anteriores, na geração de animações faciais realísticas durante a fala. Nosso trabalho segue uma linha de pesquisas computacionais para a geração deste tipo de animação. Dentre esses trabalhos, podemos citar um outro artigo recente de Lucero et al. (ver [13]). O artigo propõe uma análise dos registros da posição 3D de um conjunto de marcadores colocados no rosto de um sujeito, enquanto este fala, indentificando grupos de marcadores com padrões de movimentos similares. Esses grupos definem regiões cinemáticas independentes, que constituem uma base para expressar o movimento total da superfície facial.

No que segue, esta dissertação terá a seguinte estrutura:

No primeiro Capítulo são introduzidos conceitos básicos de álgebra linear, definições e resultados sobre a SVD e sua relação com o posto de uma matriz e finalizamos com uma rápida discussão sobre a SVD.

No Capítulo seguinte, definimos fatoração QR e algumas propriedades desta decomposição. Em seguida, apresentamos como obter a fatoração QR por Reflexão de Householder, concluindo com alguns resultados que relacionam a fatoração QR com o posto da matriz.

Já no terceiro Capítulo, apresentamos algumas aplicações da fatoração QR, tal como na resolução de sistemas lineares, problema de mínimos quadrados e uma solução para o problema de seleção de subconjunto, finalizando com uma breve discussão deste Capítulo.

Por fim, no quarto Capítulo, mostramos como a fatoração QR pode ser útil na obtenção de um modelo para geração de animações faciais e alguns resultados acerca da fatoração QR, além da análise dos erros. O Matlab foi utilizado em todos os gráficos e tabelas que aparecem neste Capítulo.

Por último, apresentamos algumas Conclusões, Anexos e o Apêndice, onde está disposto o código fonte em Matlab da maioria dos programas que foram implementados nesta dissertação.

Capítulo 1

Conceitos Básicos e Decomposição em Valores Singulares

1.1 Introdução

Esse capítulo tem como objetivo introduzir alguns conceitos de álgebra linear que serão usados nos capítulos seguintes. Começaremos definindo normas, posto, ortogonalidade e alguns resultados sobre o mesmo.

Em seguida, focalizaremos nossos estudos na Decomposição de Valores singulares (SVD) e provaremos alguns dos principais resultados que envolvem este tipo de decomposição, principalmente naqueles que relacionam SVD e o posto de uma matriz, mostrando que a SVD pode ser usado para reduzir a dimensão de uma matriz de dados.

Por fim, usaremos o SVD para determinar o posto numérico de uma matriz A e concluiremos esse capítulo, fazendo uma análise desta decomposição.

1.2 Norma de Vetores

Uma norma é uma função

$$\begin{aligned} \|\cdot\| : \mathbb{R}^n &\rightarrow \mathbb{R} \\ x &\rightarrow \|x\| \end{aligned}$$

que faz uma correspondência de um valor real ('comprimento') a cada vetor.

Para que corresponda a uma idéia razoável de comprimento, uma norma deve satisfazer três axiomas. Para quaisquer $x, y \in \mathbb{R}^n$ e escalar $\alpha \in \mathbb{R}$,

1. $\|x\| \geq 0$, e $\|x\| = 0$ se, e só se, $x = 0$;
2. $\|x + y\| \leq \|x\| + \|y\|$;
3. $\|\alpha x\| = |\alpha| \|x\|$.

Em palavras, estas condições expressam o seguinte: Por 1. a norma de um vetor não-nulo é positiva, 2. é a famosa desigualdade do triângulo, 3. é a propriedade de homogeneidade.

Pode-se provar (ver [7]) que para quaisquer $x, y \in \mathbb{R}^n$ temos

$$\|x \cdot y\| \leq \|x\| \cdot \|y\|.$$

Esta inequação é conhecida como desigualdade de *Cauchy-Schwarz*.

As normas mais utilizadas em Análise Numérica são as denominadas p -normas.

Definição 1.1 p -norma de um vetor x é dada por

$$\|x\|_p = \left(\sum_{i=1}^m |x_i|^p \right)^{\frac{1}{p}}, \quad (1.1)$$

para todo $x \in \mathbb{R}^n$.

Para os casos particulares em que p é igual a 1, 2 ou ∞ , temos as seguintes normas:

1. 1-norma: $\|x\|_1 = \sum_{i=1}^m |x_i|$;
2. 2-norma: $\|x\|_2 = \left(\sum_{i=1}^m |x_i|^2 \right)^{\frac{1}{2}}$;
3. ∞ -norma: $\|x\|_\infty = \max_{i=1, \dots, n} |x_i|$.

O ítem 1. é conhecido como norma da *Soma*, 2. como norma *Euclidiana* e 3. é a norma do *Máximo*.

Todas as normas em \mathbb{R}^n são equivalentes (ver [7]), isto é, se $\|\cdot\|_\alpha$ e $\|\cdot\|_\beta$ são normas em \mathbb{R}^n , então existem constantes positivas c_1 e c_2 tais que

$$c_1\|x\|_\alpha \leq \|x\|_\beta \leq c_2\|x\|_\alpha \quad (1.2)$$

para todo $x \in \mathbb{R}^n$.

Pode-se provar (ver [7]) que, para todo $x \in \mathbb{R}^n$, valem as seguintes desigualdades:

1. $\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2$;
2. $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$;
3. $\|x\|_\infty \leq \|x\|_1 \leq \sqrt{n}\|x\|_\infty$.

Em geral, os valores das p -normas, $p = 1, 2$ e ∞ são distintos.

1.3 Norma de Matrizes

Uma matriz $A \in \mathbb{R}^{m \times n}$ pode ser vista como um vetor no espaço de dimensão mn , cada um dos elementos considerados como coordenada independente. Portanto, qualquer norma vetorial pode ser utilizada para medir o ‘tamanho’ de uma matriz.

Uma norma matricial é uma função

$$\begin{array}{lcl} \|\cdot\| : \mathbb{R}^{m \times n} & \rightarrow & \mathbb{R} \\ A & \rightarrow & \|A\| \end{array}$$

que satisfaz 3 condições. Para quaisquer $A, B \in \mathbb{R}^{m \times n}$ e $\alpha \in \mathbb{R}$ temos

1. $\|A\| \geq 0$, e $\|A\| = 0$ se, e só se, $A = 0$;
2. $\|A + B\| \leq \|A\| + \|B\|$;
3. $\|\alpha A\| = |\alpha|\|A\|$.

Em alguns casos, podemos ter ainda uma propriedade adicional, que relaciona a operação de multiplicação de matriz, isto é, $\|A.C\| \leq \|A\|.\|C\|$ para quaisquer $A \in \mathbb{R}^{m \times n}$ e $C \in \mathbb{R}^{n \times m}$. Essa propriedade é chamada de *consistência*, (ver [7]).

Por simplicidade, utilizaremos a mesma norma $\|\cdot\|_p$ no domínio e na imagem, a p -norma de A , e denotaremos por $\|A\|_p$. Assim,

Definição 1.2 p -norma de uma matriz A é dada por $\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$, para todo $x \in \mathbb{R}^n$.

Equivalentemente à Definição 1.2 temos que

$$\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{x \neq 0} \left\| A \left(\frac{x}{\|x\|_p} \right) \right\|_p = \max_{\|x\|_p=1} \|Ax\|_p.$$

Em particular, se $p = 1, 2$ ou ∞ temos, respectivamente, que

1. $\|A\|_1 = \max_{x \neq 0} \frac{\|Ax\|_1}{\|x\|_1}$;
2. $\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$;
3. $\|A\|_\infty = \max_{x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty}$.

Pode-se provar (ver [7]) que se $p = 1$ e $p = \infty$ temos, respectivamente, que

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}| \tag{1.3}$$

$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}| \tag{1.4}$$

Uma norma matricial, frequentemente utilizada, é chamada de *Norma de Frobenius* ou *Hilbert-Schmidt* e é definida a partir de

Definição 1.3 Norma de Frobenius de uma matriz $A \in \mathbb{R}^{m \times n}$ é dada por

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

Seja B uma matriz com elementos b_{ij} onde $1 \leq i \leq m$ e $1 \leq j \leq n$. Denotemos por b_l , com $1 \leq l \leq n$, as respectivas colunas da matriz B . Então

$$\begin{aligned} \sum_{l=1}^n \|b_l\|_2^2 &= \|b_1\|_2^2 + \|b_2\|_2^2 + \cdots + \|b_n\|_2^2 \\ &= (b_{11}^2 + b_{21}^2 + \cdots + b_{m1}^2) + \cdots + (b_{1n}^2 + b_{2n}^2 + \cdots + b_{mn}^2) \\ &= \|B\|_F^2. \end{aligned} \tag{1.5}$$

Seja $C = AB$ com entradas c_{ik} , denotamos a_i^T a i -ésima linha de A e b_j a j -ésima coluna de B . Então $c_{ij} = a_i^T b_j$ e por Cauchy-Schwartz $|c_{ij}| \leq \|a_i\| \cdot \|b_j\|$. Assim,

$$\begin{aligned} \|AB\|_F^2 &= \sum_{i=1}^n \sum_{j=1}^m |c_{ij}|^2 \\ &\leq \sum_{i=1}^n \sum_{j=1}^m (\|a_i\|_2 \|b_j\|_2)^2 \\ &= \sum_{i=1}^n (\|a_i\|_2)^2 \sum_{j=1}^m (\|b_j\|_2)^2 \\ &= \|A\|_F^2 \|B\|_F^2. \end{aligned} \tag{1.6}$$

Como as normas vetoriais, as normas matriciais A_p com $p = 1$, $p = \infty$ e A_F também possuem relações de equivalência. Podemos provar que (ver [7]) dada uma matriz $A \in \mathbb{R}^{m \times n}$ temos

1. $\|A\|_2 \leq \|A\|_F \leq \sqrt{n} \|A\|_2$;
2. $\max_{i,j} |a_{ij}| \leq \|A\|_2 \leq \sqrt{mn} \max_{i,j} |a_{ij}|$;
3. $\frac{1}{\sqrt{n}} \|A\|_\infty \leq \|A\|_2 \leq \sqrt{m} \|A\|_\infty$;
4. $\frac{1}{\sqrt{m}} \|A\|_1 \leq \|A\|_2 \leq \sqrt{n} \|A\|_1$.

1.4 Imagem, Espaço Nulo e Posto

Definição 1.4 Dada uma coleção de vetores a_1, a_2, \dots, a_n em \mathbb{R}^n , o conjunto de todas as combinações lineares desses vetores é um subespaço denotado por **Espaço**

de a_1, a_2, \dots, a_n , ou seja,

$$\langle a_1, a_2, \dots, a_n \rangle = \left\{ \sum_{j=1}^n \beta_j a_j; \beta_j \in \mathbb{R} \right\}.$$

Definição 1.5 *Seja $A \in \mathbb{R}^{m \times n}$. Dizemos que a **Imagem** de A é*

$$\text{Im}(A) = \{y \in \mathbb{R}^m; y = Ax \text{ para algum } x \in \mathbb{R}^n\}$$

Em relação a Definição 1.5, observe que qualquer vetor y que pertence ao espaço gerado pelas colunas de A pode ser escrito como combinação linear de suas colunas, ou seja, $y = \sum_{j=1}^n x_j a_j$, logo $y = Ax$ e portanto, $y \in \text{Im}(A)$. Reciprocamente, se $y \in \text{Im}(A)$ então $y = Ax$ para algum $x \in \mathbb{R}^n$ e portanto, Ax é combinação linear das colunas de A . Assim, se $A = [a_1, a_2, \dots, a_n]$ são as colunas particionadas de A , então

$$\text{Im}(A) = \langle a_1, a_2, \dots, a_n \rangle. \quad (1.7)$$

Definição 1.6 (Núcleo) *Seja $A \in \mathbb{R}^{m \times n}$. O **Espaço Nulo** ou **Núcleo** de A é dado por*

$$\text{N}(A) = \{x \in \mathbb{R}^n; Ax = 0\}.$$

Definição 1.7 (Posto) *Dizemos que o **Posto** de uma matriz A é*

$$\text{posto}(A) = \dim(\text{Im}(A)).$$

Definição 1.8 *Dizemos que $A \in \mathbb{R}^{m \times n}$ possui **Posto Incompleto**, se $\text{posto}(A) < \min\{m, n\}$.*

Pode-se provar que (ver [7]):

1. $\text{posto}(A) = \text{posto}(A^T)$;
2. Se $A \in \mathbb{R}^{m \times n}$, então $\dim(\text{N}(A)) + \text{posto}(A) = n$.

1.5 Ortogonalidade

Definição 1.9 Uma matriz $Q \in \mathbb{R}^{m \times n}$ é dita **Ortogonal** se, e somente se, $QQ^T = Q^TQ = I_n$.

Teorema 1.10 Se Q é uma matriz ortogonal, então

1. $\det(Q) = \pm 1$;
2. Q^T é ortogonal.
3. $\|Qx\|_2 = \|x\|_2$, para todo vetor x ;
4. $\|QA\|_2 = \|A\|_2$, para toda matriz A ;
5. $\|Q\|_2 = 1$.

Demonstração:

Omitiremos a demonstração para os itens 1, 2, 4 e podem ser encontradas em [19].

3. Se x é um vetor qualquer, então

$$\begin{aligned}
 \|Qx\|_2^2 &= (Qx)^T(Qx) \\
 &= x^T Q^T Q x \\
 &= x^T x \\
 &= \|x\|_2^2 \\
 &= \|x\|_2.
 \end{aligned} \tag{1.8}$$

5. É consequência imediata de 2. e de $\|I\|_2 = 1$.

■

Teorema 1.11 Se Q_1 e Q_2 são matrizes ortogonais, então Q_1Q_2 é ortogonal.

Demonstração: Ver [19].

■

Teorema 1.12 *Seja $A \in \mathbb{R}^{m \times n}$ e $Q \in \mathbb{R}^{m \times m}$ ortogonal, então $\|QA\|_F = \|A\|_F$, onde F é norma de Frobenius.*

Demonstração:

Suponha que a_1, a_2, \dots, a_n sejam as colunas de A , então

$$\begin{aligned} \|QA\|_F^2 &= \|(Qa_1, Qa_2, \dots, Qa_n)\|_F^2 \\ &= \sum_{i=1}^n \|Qa_i\|_2^2 \\ &= \sum_{i=1}^n \|a_i\|_2^2 \\ &= \|A\|_F^2. \end{aligned} \tag{1.9}$$

■

1.6 Decomposição em Valores Singulares

A Decomposição em Valores Singulares (SVD, “*Singular Value Decomposition*”) é uma fatoração de matrizes frequentemente utilizada em muitos algoritmos. É uma ferramenta contida na maioria dos pacotes matemáticos de computação e muitos problemas de Álgebra Linear podem ser resolvidos utilizando esta fatoração. Iniciamos esta Seção, apresentando a definição de SVD.

Definição 1.13 *Seja $A \in \mathbb{R}^{m \times n}$. A SVD de A é a fatoração*

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

onde $U \in \mathbb{R}^{m \times m}$ e $V \in \mathbb{R}^{n \times n}$ são **Ortogonais** e $\Sigma \in \mathbb{R}^{m \times n}$ é uma matriz **Diagonal**, isto é,

$$\sigma_{ij} = \begin{cases} \sigma_{ij}, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$$

Os elementos σ_{ii} , $i = 1, 2, \dots, p$, onde $p = \min\{m, n\}$, são denominados **Valores Singulares** de A e denotados por σ_i sendo escolhidos de modo que $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$.

Cada uma das m colunas da matriz U , denotadas por u_k , são chamadas *Vetores Singulares à Esquerda* de A e cada uma das n colunas da matriz V , denotadas por v_k , são chamadas *Vetores Singulares à Direita* de A . Os vetores u_k são os autovetores da matriz $A^T A$, e os vetores v_k são os autovetores da matriz AA^T .

Observações:

1. A decomposição de valores singulares definida em 1.13 recebe o nome de *SVD Completa*.
2. Seja $A \in \mathbb{R}^{m \times n}$, com $m \geq n$ e $A = U\Sigma V^T$ a SVD de A , onde U , V e Σ são dados como na Definição 1.13. Observe que para $m > n$ a matriz Σ possui todos $m - n$ valores singulares iguais a zero. De modo que apenas os n primeiros valores singulares de Σ influenciam na decomposição de A . Assim, suponha que Σ tenha n valores singulares não-nulos. Logo, a equação (1.10) definida em 1.13 pode ser reescrita como

$$A = U \begin{pmatrix} \widehat{\Sigma} \\ 0 \end{pmatrix} V^T.$$

Desta forma, o produto acima anula $m - n$ colunas de U e $\widehat{\Sigma}$ possui n valores singulares. Logo

$$A = \widehat{U} \widehat{\Sigma} V^T \tag{1.11}$$

onde $\widehat{U} \in \mathbb{R}^{m \times n}$, $V \in \mathbb{R}^{n \times n}$ e $\widehat{\Sigma} \in \mathbb{R}^{n \times n}$ é uma matriz diagonal com $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ todos não-nulos. Essa decomposição é conhecida como SVD *reduzida* de A .

Dada uma matriz $A \in \mathbb{R}^{m \times n}$ pode-se provar que:

1. Toda matriz A sempre admite uma decomposição em valores singulares;
2. Seus respectivos valores singulares são únicos;
3. Se os valores singulares forem distintos dois a dois, então os vetores singulares (à esquerda e à direita) são únicos a menos de sinal.

Demonstrações para esses resultados podem ser encontrados em [19].

1.7 Algumas Propriedades da SVD

Um dos aspectos que valorizam a SVD é sua capacidade de lidar com o conceito de posto de matrizes. Nessa Seção mostraremos que o SVD é uma chave para esse problema por caracterizar eficientemente uma aproximação de matrizes de um posto definido, (ver [6]). Iniciaremos essa Seção, introduzindo algumas relações entre SVD e normas (ver [7, 20]), que serão úteis adiante, quando falaremos de posto numérico.

1.7.1 SVD e sua Relação com Normas

Nessa Subseção apresentaremos alguns resultados que relacionam 2-norma e norma de Frobenius com os valores singulares de uma matriz A que são importantes e merecem ser destacadas. Estas relações serão mostradas nos Teoremas 1.14 e 1.15.

Teorema 1.14 *Seja $A \in \mathbb{R}^{m \times n}$ uma matriz com decomposição de valor singular $U\Sigma V^T$, então*

$$\|A\|_2 = \sigma_1 \quad (\text{maior valor singular}). \quad (1.12)$$

Demonstração:

Sabemos que $\|U\|_2 = \|V^T\|_2 = 1$ porque U e V são ortogonais. Assim,

$$\begin{aligned} \|A\|_2 &= \|U\Sigma V^T\|_2 \\ &= \|\Sigma\|_2 \\ &= \max_{x \neq 0} \frac{\|\Sigma x\|_2}{\|x\|_2} \\ &= \max_{x \neq 0} \frac{\left(\sum_{i=1}^n (\sigma_i x_i)^2 \right)^{\frac{1}{2}}}{\left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}} \\ &= \max_{x \neq 0} \frac{((\sigma_1 x_1)^2 + (\sigma_2 x_2)^2 + \cdots + (\sigma_n x_n)^2)^{\frac{1}{2}}}{(x_1^2 + x_2^2 + \cdots + x_n^2)^{\frac{1}{2}}}. \end{aligned}$$

Como σ_1 é o maior valor singular da matriz A , substituindo cada σ_i , onde $i = 1, \dots, n$, por σ_1 teremos,

$$\begin{aligned} \|A\|_2 &\leq \max_{x \neq 0} \frac{(\sigma_1^2(x_1^2 + x_2^2 + \dots + x_n^2))^{\frac{1}{2}}}{(x_1^2 + x_2^2 + \dots + x_n^2)^{\frac{1}{2}}} \\ &= \max_{x \neq 0} \sigma_1 \frac{(x_1^2 + x_2^2 + \dots + x_n^2)^{\frac{1}{2}}}{(x_1^2 + x_2^2 + \dots + x_n^2)^{\frac{1}{2}}} \\ &= \max_{x \neq 0} \sigma_1 \\ &= \sigma_1. \end{aligned}$$

Portanto,

$$\|A\|_2 \leq \sigma_1. \quad (1.13)$$

Por outro lado, escolhendo $x = e_1$, temos

$$\begin{aligned} \|A\|_2 &= \|U\Sigma V^T\|_2. \\ &= \max_{x \neq 0} \frac{((\sigma_1 x_1)^2 + (\sigma_2 x_2)^2 + \dots + (\sigma_n x_n)^2)^{\frac{1}{2}}}{(x_1^2 + x_2^2 + \dots + x_n^2)^{\frac{1}{2}}} \\ &= \max_{x \neq 0} \frac{((\sigma_1 1)^2 + (\sigma_2 0)^2 + \dots + (\sigma_n 0)^2)^{\frac{1}{2}}}{(1^2 + 0^2 + \dots + 0^2)^{\frac{1}{2}}} \\ &= \max_{x \neq 0} \sigma_1 \\ \|A\|_2 &= \sigma_1. \end{aligned} \quad (1.14)$$

De (1.13) e (1.14) segue (1.12). ■

Teorema 1.15 *Seja $A \in \mathbb{R}^{m \times n}$ uma matriz com decomposição em valores singulares $U\Sigma V^T$, então*

$$\|A\|_F = (\sigma_1^2 + \sigma_2^2 + \dots + \sigma_n^2)^{\frac{1}{2}}. \quad (1.15)$$

Demonstração:

Pelo Teorema 1.12 para qualquer matriz ortogonal Q temos que $\|QA\|_F = \|A\|_F$.

Considere $U\Sigma V^T$ a decomposição em valores singulares de A , onde $U \in \mathbb{R}^{m \times n}$, $\Sigma \in \mathbb{R}^{n \times n}$ e $V^T \in \mathbb{R}^{n \times n}$. Então

$$\begin{aligned}
 \|A\|_F &= \|U\Sigma V^T\|_F \\
 &= \|\Sigma V^T\|_F \\
 &= \|(\Sigma V^T)^T\|_F \\
 &= \|(V\Sigma^T)\|_F \\
 &= \|\Sigma^T\|_F \\
 &= (\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_n^2)^{\frac{1}{2}}.
 \end{aligned}$$

■

1.7.2 Relações entre SVD e o Posto de uma Matriz

Esta Subseção tem o propósito de relacionar a SVD e o posto de uma matriz, usando os respectivos valores singulares. Começamos por um teorema que fornece uma relação entre o posto e os valores singulares de uma matriz.

Teorema 1.16 (Posto) *Seja $A \in \mathbb{R}^{m \times n}$ e Σ a matriz de valores singulares de A . Então, $\text{posto}(A) = n$ se, e somente se, Σ têm n valores singulares não-nulos.*

Demonstração:

Suponha que Σ têm n valores singulares não-nulos então $\text{posto}(\Sigma) = n$, já que Σ é uma matriz diagonal. Por outro lado, $A = U\Sigma V^T$ e como U e V têm posto completo, então $\text{posto}(A) = \text{posto}(\Sigma)$. A recíproca é análoga.

■

O Teorema 1.16 fornece uma alternativa para se determinar o posto de uma matriz em vez de usar a Definição 1.7, porém agora, usando a SVD. Conseqüentemente, uma matriz $A \in \mathbb{R}^{m \times n}$ terá posto incompleto, isto é, $r < n$, se $\sigma_i > 0$, $i = 1, \dots, r$ e $\sigma_{r+1} = \cdots = \sigma_n = 0$.

A fórmula seguinte é uma das mais importantes propriedades da decomposição em valores singulares.

Teorema 1.17 *Uma matriz $A \in \mathbb{R}^{m \times n}$ com posto r , $r \leq n$, pode ser escrita na forma*

$$A_r = \sum_{j=1}^r \sigma_j u_j v_j^T, \quad (1.16)$$

em que u_i , $i = 1, \dots, r$, são os primeiros r vetores singulares à esquerda de A , v_i , $i = 1, \dots, r$, são os primeiros r vetores singulares à direita de A e σ_i , $i = 1, \dots, r$, são os valores singulares de A .

Demonstração:

Tem-se que

$$\begin{aligned} I &= VV^T \\ &= (v_1, v_2, \dots, v_n)(v_1^T, v_2^T, \dots, v_n^T) \\ &= v_1 v_1^T + v_2 v_2^T + \dots + v_n v_n^T, \end{aligned}$$

já que V é uma matriz ortogonal. Se multiplicarmos por A a esquerda da equação acima em ambos os membros da igualdade teremos:

$$\begin{aligned} A &= A(v_1 v_1^T + v_2 v_2^T + \dots + v_n v_n^T) \\ &= (Av_1)v_1^T + (Av_2)v_2^T + \dots + (Av_n)v_n^T \\ &= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_n u_n v_n^T. \end{aligned}$$

Se A tem posto $r = n$, segue do Teorema 1.16 que A possui n valores singulares não-nulos e portanto

$$A = \sum_{j=1}^r \sigma_j u_j v_j^T.$$

Caso contrário, se A tem posto $r < n$, então pelo Teorema 1.16 temos que $\sigma_{r+1} = \sigma_{r+2} = \dots = \sigma_n = 0$. Daí,

$$\begin{aligned} A &= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_r u_r v_r^T \\ &= \sum_{j=1}^r \sigma_j u_j v_j^T. \end{aligned}$$

■

Observação:

• Dados $u_1 = \begin{pmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{m1} \end{pmatrix}$ e $v_1 = \begin{pmatrix} v_{11} \\ v_{21} \\ \vdots \\ v_{n1} \end{pmatrix}$ então,

$$\begin{aligned} u_1 v_1^T &= \begin{pmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{m1} \end{pmatrix} (v_{11} \ v_{21} \ \cdots \ v_{n1}) \\ &= \begin{pmatrix} u_{11}v_{11} & u_{11}v_{21} & \cdots & u_{11}v_{n1} \\ u_{21}v_{11} & u_{21}v_{21} & \cdots & u_{21}v_{n1} \\ \vdots & \vdots & \cdots & \vdots \\ u_{m1}v_{11} & u_{m1}v_{21} & \cdots & u_{m1}v_{n1} \end{pmatrix}. \end{aligned}$$

Portanto, as colunas da matriz $u_1 v_1^T$ são múltiplos do vetor u_1 , e assim, a matriz $u_1 v_1^T$ tem posto 1. De maneira análoga, todas as matrizes $u_i v_i^T$, com $i = 1, \dots, r$ têm exatamente posto 1.

A partir desta observação e do Teorema 1.17 podemos afirmar que qualquer matriz (com posto r) é uma combinação linear de r matrizes de posto 1. Os coeficientes desta combinação linear são os valores singulares $\sigma_1, \sigma_2, \dots, \sigma_r$ da matriz. Em determinadas aplicações, aparecem matrizes cujos valores singulares menores deveriam ser nulos, mas não o são por determinados motivos (por exemplo, erros de arredondamento). É freqüente, nestes casos, substituir esses valores singulares por zero, desprezando as suas contribuições, e considerar uma matriz aproximada com menos termos na combinação linear das matrizes de posto 1.

Teorema 1.18 *Seja $A \in \mathbb{R}^{m \times n}$. Para qualquer ν com $0 \leq \nu \leq r$, definimos*

$$A_\nu = \sum_{j=1}^{\nu} \sigma_j u_j v_j^T, \tag{1.17}$$

se $\nu = p = \min\{m, n\}$, defina $\sigma_{\nu+1} = 0$. Então

$$\|A - A_\nu\|_2 = \min_{\text{posto}(B) \leq \nu} \|A - B\|_2 = \sigma_{\nu+1}.$$

Demonstração: Ver [19].

■

Observações:

1. É fácil ver que $U^T A_\nu V = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_\nu, 0, \dots, 0)$ e daí, $\text{posto}(A_\nu) = \nu$. Por outro lado, $U^T(A - A_\nu)V = \text{diag}(0, 0, \dots, 0, \sigma_{\nu+1}, \dots, \sigma_p)$ e pelo Teorema 1.14, temos $\|A - A_\nu\|_2 = \sigma_{\nu+1}$.
2. Observe que de acordo com o Teorema 1.17 podemos usar o SVD para expressar uma matriz A como combinação linear de uma base de vetores ortogonais. Ou seja, dada uma matriz $A \in \mathbb{R}^{m \times n}$, podemos usar os primeiros r valores singulares ($r < n$), e aproximar as colunas de A como combinação linear de uma base de r vetores. De acordo com o Teorema 1.18, quando aproximamos uma matriz A de posto $n \geq r$, para uma matriz $A_r = U \Sigma_r V^T$ com posto r , temos que essa aproximação no sentido da $\|\cdot\|_2$ é a melhor possível de A , pois o ínfimo é atingido para toda matriz B definida por

$$B = U \begin{pmatrix} \Sigma' \\ 0 \end{pmatrix} V^T, \quad (1.18)$$

onde $\Sigma' = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r, 0, \dots, 0)$.

1.7.3 Posto Numérico

Seja $A \in \mathbb{R}^{m \times n}$. Suponha que uma matriz A que originalmente tinha posto $r < n$ têm seus elementos perturbados por algum tipo de erro, por exemplo, arredondamento ou erros de medidas. Certamente, esses erros de arredondamento, não permitirá que a matriz A continue com posto exatamente igual a r . Realmente, o que é provável, é que a matriz que foi perturbada terá posto maior que r .

Suponha que ambas as matrizes acima sejam submetidas a algoritmos numéricos ou estatísticos. A proximidade de A para matriz perturbada não fornecerá interpretações erradas quando ambas as matrizes forem submetidas a esses algoritmos.

Assim, um modo de evitar possíveis problemas com a definição de posto (ver [6]), é especificar uma tolerância e dizer que a matriz A tem posto numericamente deficiente, se dentro dessa tolerância, a matriz A está próxima da matriz de posto deficiente. Em outras palavras,

Definição 1.19 *Uma matriz A possui um ϵ -posto r com norma $\|\cdot\|$ se*

$$r = \text{posto}(A, \epsilon) = \inf\{\text{posto}(B); \|A - B\| \leq \epsilon\}.$$

Entretanto, essa definição pode apresentar problemas, pois um pequeno aumento em ϵ poderia acarretar na diminuição do posto numérico de A . É necessário então, achar um limite superior para ϵ para o qual o posto numérico fica pelo menos igual a r . Tal número é fornecido por qualquer δ satisfazendo

Definição 1.20

$$\epsilon < \delta \leq \sup\{\eta; \|A - B\| < \eta \Rightarrow \text{posto}(B) \geq r\},$$

onde ϵ é dado como na Definição 1.19.

Através das Definições 1.19 e 1.20 podemos então caracterizar posto numérico de uma matriz A , ou seja,

Definição 1.21 *Uma matriz A tem **Posto Numérico** (δ, ϵ, r) com norma $\|\cdot\|$ se δ, ϵ e r satisfazem as Definições 1.19 e 1.20.*

Usando as Definições 1.19 e 1.20, podemos caracterizar o posto numérico de uma matriz. O seguinte teorema tem esse propósito. Usaremos a notação $\text{posto}(\delta, \epsilon, r)_2$ para caracterizar o posto numérico com $\|\cdot\|_2$.

Teorema 1.22 *Sejam $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ os valores singulares de uma matriz A . Então A tem posto numérico $(\delta, \epsilon, r)_2$ se, e somente se,*

$$\sigma_r \geq \delta > \epsilon \geq \sigma_{r+1}. \quad (1.19)$$

Demonstração:

Suponha (1.19). Pelo Teorema 1.18, se $\|A - B\|_2 < \delta$ logo, $\sigma_{r+1} < \delta$ e portanto $\text{posto}(B) \geq r$ e conseqüentemente satisfaz a Definição 1.18. Como a matriz B definida em (1.18), tem posto r e portanto $\|A - B\|_2 \leq \epsilon$, daí satisfaz a Definição 1.19.

Reciprocamente, suponha agora que δ, ϵ e r satisfazem as Definições 1.19 e 1.20 então $\delta > \epsilon$ e por definição temos $\sigma_r \geq \sigma_{r+1}$. Basta então mostrar que $\sigma_r \geq \delta$ e que $\epsilon \geq \sigma_{r+1}$. Suponha por contradição que $\epsilon < \sigma_{r+1}$, então, como $\|A - B\|_2 < \epsilon$, isso implicaria que $\|A - B\|_2 < \sigma_{r+1}$, contradição com o Teorema 1.18. Por outro lado, pelo Teorema 1.18 tem-se $\sigma_r \geq \delta$. Isso prova o teorema. ■

Mais detalhes acerca desta Subseção podem ser encontradas em [6].

1.8 Análise e Discussão

Como já mencionamos na introdução, alguns trabalhos usam o PCA para geração de animações faciais. Em geral, a SVD é utilizado na obtenção de uma base para o PCA. Podemos citar por exemplo o trabalho de Kuratate et al., (ver [18]).

Como vimos, a SVD é uma poderosa técnica que permite aproximar uma determinada matriz A de posto n para uma matriz de posto $r < n$ (Teorema 1.17). Essa aproximação para uma matriz A_r de posto $r < n$ é ótima no sentido da $\| \cdot \|_2$ (Teorema 1.18).

Sob a ótica do problema de seleção de subconjunto gostaríamos de selecionar r colunas de uma matriz de dados $A \in \mathbb{R}^{m \times n}$ ($m \geq n$). Tais r colunas devem ser independentes de tal forma que podemos desprezar as contribuições das outras

selecionadas sejam usadas para obter uma aproximação das outras $(n - r)$ colunas. Dessa forma, obteremos um modelo que pode ser usado para prever as $(n - r)$ colunas restantes de uma matriz de dados qualquer, bastando apenas selecionar um conjunto de r colunas independentes da matriz A . Com o SVD, isso não é possível. Primeiro, porque teríamos que encontrar todos os vetores u_i , com $i = 1, \dots, r$ para obtermos \tilde{a}_i onde $i = 1, \dots, n$, mas obviamente, esses vetores dependem de todas as colunas da matriz A e isso não nos interessa, pois frustraria todo o nosso esforço na tentativa de reduzir a dimensão dos dados da matriz A . Segundo, é verdade que a interpretação dos resultados através da utilização dessa técnica é facilitada pela redução da dimensionalidade (posto), mantendo um elevado grau de explicação, porém, quais colunas da matriz A foram usadas para obter as colunas da matriz A_r ? De fato, não sabemos e no presente caso, não queremos que isso ocorra. Gostaríamos de saber exatamente quais são essas colunas.

Como veremos nos Capítulos 2 e 3, a fatoração QR atende precisamente as nossas necessidades, selecionando um conjunto de r colunas independentes de uma matriz A . Posteriormente, essas r colunas são utilizadas para aproximar exatamente cada uma das $(n - r)$ colunas da matriz de dados A em termos de uma combinação linear de uma base de r colunas. Justificando nossa escolha de usar a fatoração QR para obtenção do modelo.

Capítulo 2

A Fatoração QR

2.1 Introdução

Nesse Capítulo começaremos por apresentar a definição de Fatoração QR e alguns resultados acerca do mesmo. Em seguida, discutimos as chamadas matrizes de Householder e como podemos obter a decomposição QR usando essas matrizes.

Estudaremos as relações que existem entre a fatoração QR e o posto de uma determinada matriz. Provaremos, que se uma matriz A tem posto completo, implica na unicidade de sua decomposição QR e conseqüentemente, uma base para a $\text{Im}(A)$.

Por final, analisaremos o caso em que A não tem posto completo e como obter neste caso, uma base para a $\text{Im}(A)$.

2.2 Definição e Propriedades da Fatoração QR

Sejam a_1, a_2, \dots, a_n as colunas da matriz A e a sequência de subspaços gerados por elas:

$$\langle a_1 \rangle \subseteq \langle a_1, a_2 \rangle \subseteq \dots \subseteq \langle a_1, a_2, \dots, a_n \rangle.$$

Assim, $\langle a_1 \rangle$ é um espaço 1-dimensional gerado por a_1 , $\langle a_1, a_2 \rangle$ é o espaço 2-dimensional gerado por a_1 e a_2 e assim por diante. A idéia da fatoração QR está na construção de uma sequência de vetores q_1, q_2, \dots, q_n ortonormais que geram essa mesma sequência de subspaços, (ver [7]).

Para sermos mais precisos, assumimos por um momento que $A \in \mathbb{R}^{m \times n}$ com $m \geq n$ e que $\text{posto}(A) = n$ (nesse caso, dizemos que se A possui posto máximo, então as n colunas de Q formam uma base ortonormal da imagem de A). Gostaríamos que a sequência q_1, q_2, \dots, q_n tivesse a seguinte propriedade

$$\langle a_1, a_2, \dots, a_i \rangle = \langle q_1, q_2, \dots, q_i \rangle, \quad i = 1, \dots, n. \quad (2.1)$$

Em particular, tomando $i = 2$ na igualdade acima temos que $\langle a_1, a_2 \rangle = \langle q_1, q_2 \rangle$. Assim, para que a_1 seja combinação linear de q_1 , devemos ter um escalar r_{11} tal que $a_1 = r_{11}q_1$. Por definição, para que $\langle a_1, a_2 \rangle = \langle q_1, q_2 \rangle$ devem existir escalares t_i , $i = 1, \dots, 4$ tais que $t_1 a_1 + t_2 a_2 = t_3 q_1 + t_4 q_2$. Substituindo a_1 por $r_{11}q_1$, teremos:

$$\begin{aligned} t_1 a_1 + t_2 a_2 &= t_3 q_1 + t_4 q_2 \\ t_1(r_{11}q_1) + t_2 a_2 &= t_3 q_1 + t_4 q_2 \\ t_2 a_2 &= -t_1(r_{11}q_1) + t_3 q_1 + t_4 q_2 \\ t_2 a_2 &= q_1(-t_1 r_{11} + t_3) + t_4 q_2 \\ a_2 &= \left(\frac{-t_1 r_{11} + t_3}{t_2}\right)q_1 + \frac{t_4}{t_2}q_2 \\ a_2 &= r_{12}q_1 + r_{22}q_2 \end{aligned}$$

onde, r_{12} e r_{22} escalares tais que, $r_{12} = \frac{-t_1 r_{11} + t_3}{t_2}$ e $r_{22} = \frac{t_4}{t_2}$. Analogamente, temos:

$$\begin{aligned} a_1 &= r_{11}q_1 \\ a_2 &= r_{12}q_1 + r_{22}q_2 \\ &\vdots \\ a_n &= r_{1n}q_1 + r_{2n}q_2 + \dots + r_{nn}q_n. \end{aligned}$$

Observando as equações acima, vemos que elas podem ser escritas utilizando matrizes Q ortogonal e R triangular superior, onde $A = QR$.

Definição 2.1 *Seja $A \in \mathbb{R}^{m \times n}$. A decomposição de uma matriz A do tipo $A = QR$, onde $Q \in \mathbb{R}^{m \times m}$ é uma matriz ortogonal e $R \in \mathbb{R}^{m \times n}$ é triangular superior, é dita **Fatoração QR** de A .*

Observe que dada uma matriz $A \in \mathbb{R}^{m \times n}$, temos duas possibilidades para as linhas de A , $m \geq n$ ou $m < n$. Daremos em seguida, alguns detalhes (ver [20, 7]) sobre como obter a fatoração QR de uma matriz A para esses casos.

1º Caso: $m \geq n$

Teorema 2.2 Dada uma matriz $A \in \mathbb{R}^{m \times n}$, com $m \geq n$, existe uma matriz ortogonal $Q \in \mathbb{R}^{m \times m}$ e uma matriz triangular superior $R \in \mathbb{R}^{m \times n}$ tal que

$$A = QR. \quad (2.2)$$

Demonstração:

A prova desse teorema será dada na Seção 2.4.

■

Observações:

1. Seja $A \in \mathbb{R}^{m \times n}$, $m \geq n$. A decomposição QR onde existem matrizes $Q \in \mathbb{R}^{m \times m}$

e $R \in \mathbb{R}^{m \times n}$, onde Q é ortogonal e $R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}$, $\hat{R} \in \mathbb{R}^{n \times n}$ é triangular superior é chamada de *fatoração QR completa*.

2. Suponha que $A \in \mathbb{R}^{m \times n}$, $m \geq n$. Se $m > n$ então pelo Teorema 2.2 e pela observação 1, existem $Q \in \mathbb{R}^{m \times m}$ e $R \in \mathbb{R}^{m \times n}$, onde Q é ortogonal e $R =$

$\begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}$, $\hat{R} \in \mathbb{R}^{n \times n}$ é triangular superior e $A = QR$. Seja $\hat{Q} \in \mathbb{R}^{m \times n}$ uma matriz que consiste das n primeiras colunas de Q , logo \hat{Q} também é ortogonal. Seja $\tilde{Q} \in \mathbb{R}^{m \times (m-n)}$ uma matriz ortogonal que representa as outras $m - n$

colunas restantes de Q . Então $A = QR = [\hat{Q} \tilde{Q}] \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} = \hat{Q}\hat{R} + \tilde{Q}0$, isto é, $A = \hat{Q}\hat{R}$.

A fatoração QR de $A \in \mathbb{R}^{m \times n}$ com $m \geq n$ onde, $Q \in \mathbb{R}^{m \times n}$ é ortogonal e $R \in \mathbb{R}^{n \times n}$ é triangular superior é denominada de *fatoração QR reduzida*.

2º Caso: $m < n$

Teorema 2.3 *Seja $A \in \mathbb{R}^{m \times n}$, $m < n$. Então existem matrizes $Q \in \mathbb{R}^{m \times m}$ e $R = (R_{11} \ R_{12}) \in \mathbb{R}^{m \times n}$, onde Q é ortogonal, $R_{11} \in \mathbb{R}^{m \times m}$ triangular superior, $R_{12} \in \mathbb{R}^{m \times (n-m)}$ é retangular e $A = QR$.*

Demonstração:

A prova desse teorema será dada na Seção 2.4. ■

Basicamente, existem três métodos para se obter a fatoração QR, são eles:

1. Reflexão de Householder;
2. Rotação de Givens;
3. Processo de Ortogonalização de Gram-Schmidt.

Descreveremos a seguir um método para se obter a fatoração QR. Esse método é baseado na *reflexão de Householder*.

2.3 Reflexão de Householder

Para melhor entendimento da reflexão de Householder, é conveniente definir reflexão (ver [20]). Entretanto, não entraremos em detalhes, trataremos de um caso simples de reflexão no \mathbb{R}^2 , apenas para podermos ter uma idéia geométrica.

2.3.1 Reflexão

Seja ξ uma reta em \mathbb{R}^2 que passa pela origem. O operador que reflete cada vetor em \mathbb{R}^2 pela reta ξ é uma transformação linear, e portanto, pode ser representado por uma matriz. Gostariamos de determinar esta matriz.

Seja v um vetor não nulo pertencente a ξ . Logo, cada vetor não-nulo que pertence a ξ só pode ser múltiplo de v . Considere u um vetor não-nulo ortogonal a ξ . Segue

que, $\{u, v\}$ é uma base de \mathbb{R}^2 , logo cada $x \in \mathbb{R}^2$ pode ser expressado como combinação linear de u e v . Então $x = \alpha u + \beta v$, com $\alpha, \beta \in \mathbb{R}$. A reflexão de x por ξ é $x = -\alpha u + \beta v$ (ver Figura 2.1), logo, a matriz Q de reflexão deve satisfazer $Q(\alpha u + \beta v) = -\alpha u + \beta v$. Assim, para que isso ocorra, é necessário e suficiente que

$$Qu = -u \text{ e } Qv = v. \quad (2.3)$$

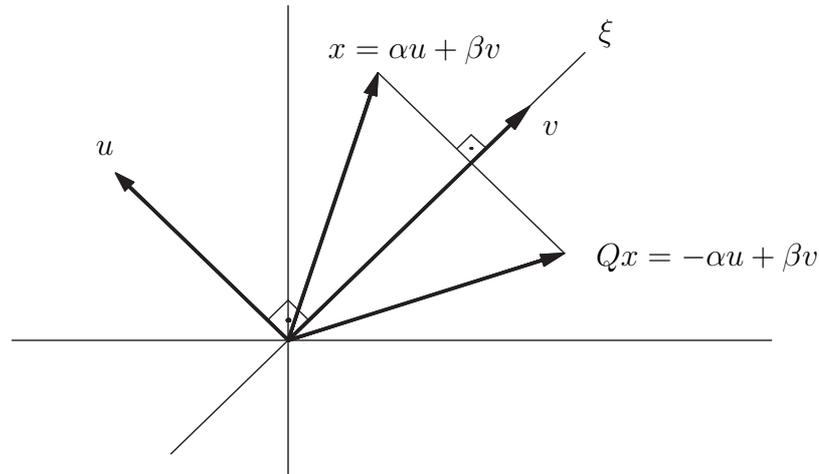


Figura 2.1: Reflexão em torno da reta ξ .

Sem perda de generalidade, podemos supor que u foi escolhido de tal forma que $\|u\|_2 = 1$.

Considere agora a matriz $Q = uu^T \in \mathbb{R}^{2 \times 2}$. Então Q possui a seguinte propriedade

$$\begin{aligned} Qu &= (uu^T)u \\ &= u(u^T u) \\ &= u\|u\|_2 \\ &= u \end{aligned}$$

e

$$\begin{aligned} Qv &= (uu^T)v \\ &= u(u^T v) \\ &= u(u, v) \\ &= 0 \end{aligned}$$

já que u e v são ortogonais.

Portanto, a matriz Q não é uma reflexão. Por outro lado, se definimos $W = I - 2Q$ temos

$$\begin{aligned} Wu &= u - 2Qu \\ &= u - 2u \\ &= -u \end{aligned}$$

e

$$\begin{aligned} Wv &= v - 2Qv \\ &= v - 0 \\ &= v. \end{aligned}$$

Assim, a matriz $W \in \mathbb{R}^{2 \times 2}$ que é dada por $W = I - 2Q$ reflete o vetor que passa pela reta ξ , onde u é um vetor unitário e ortogonal a ξ . Logo, W é uma matriz de reflexão.

Teorema 2.4 *Seja $v \in \mathbb{R}^n$ um vetor não nulo e $P = I - 2\frac{vv^T}{v^T v}$. Então P é uma reflexão.*

Demonstração:

$$\begin{aligned} P &= I - 2\frac{vv^T}{v^T v} \\ &= I - 2\frac{vv^T}{\|v\|_2^2} \\ &= I - 2\frac{vv^T}{\|v\|_2 \cdot \|v\|_2} \\ &= I - 2\widehat{v}\frac{v^T}{\|v\|} \\ &= I - 2\widehat{v}\widehat{v}^T. \end{aligned} \tag{2.4}$$

Logo, P é uma reflexão.

■

2.3.2 Definição e Propriedades da Reflexão de Householder

A definição seguinte é chamada de reflexão de Householder devido ao matemático e psicólogo norte americano Alston Householder.

Definição 2.5 *Seja $v \in \mathbb{R}^n$ um vetor não nulo. Uma matriz P da forma*

$$P = I - 2 \frac{vv^T}{v^T v} \quad (2.5)$$

*é chamada de **reflexão de Householder**, ou em outros casos de **matriz de Householder** ou ainda **transformação de Householder**. O vetor v é chamado de **vetor de Householder**.*

Para os nossos propósitos, como veremos a seguir, é necessário obtermos algumas propriedades relacionadas com a reflexão de Householder (ver [19, 10, 20]). Uma reflexão de Householder goza das seguintes propriedades:

Teorema 2.6 *Seja $P \in \mathbb{R}^{m \times n}$ uma reflexão de Householder e $A \in \mathbb{R}^{m \times n}$. Então*

1. *P é simétrica;*
2. *P é ortogonal;*
3. *$\|Px\|_2 = \|x\|_2$;*
4. *$\|PA\|_2 = \|A\|_2$;*
5. *$P^2 = I$.*

Demonstração: Ver Anexo.

■

Um das propriedades mais importantes da reflexão de Householder se consiste em como determinar um vetor v que defina tal reflexão. O seguinte resultado mostra como encontrar o vetor v de Householder de tal forma que Px seja múltiplo de $e_1 = (1, 0, \dots, 0)^T$.

Teorema 2.7 *Dado um vetor não-nulo $x = (x_1, x_2, \dots, x_n)^T$, podemos encontrar um vetor $v = (v_1, v_2, \dots, v_n)^T$ de maneira que Px é múltiplo de e_1 onde a matriz de householder P é definida pelo vetor $v = x \pm \|x\|_2 e_1$ e $Px = \pm \|x\|_2 e_1$.*

Demonstração: Ver Anexo. ■

Portanto, todas as componentes do vetor x à exceção da primeira são anuladas por multiplicação da matriz P .

De acordo com o Teorema 2.7 temos duas possibilidades de reflexão para o vetor x , $Px = \|x\|_2 e_1$ ou $Px = -\|x\|_2 e_1$. A figura 2.2 representa geometricamente essa situação, onde H^+ e H^- representam hiperplanos no espaço n -dimensional. Aliás, um hiperplano, representa um conjunto de pontos no \mathbb{R}^n , tais que $ax_1 + \dots + ax_n = b$, com $a_i, i = 1, \dots, n$ e b são números reais. Em particular, num espaço tridimensional um hiperplano é o plano habitual. Num espaço bidimensional, um hiperplano é uma reta. Num espaço monodimensional, um hiperplano é um ponto. Assim, um hiperplano divide o espaço em que está definido em duas partes. Cada uma delas é chamada de semi-espaço.

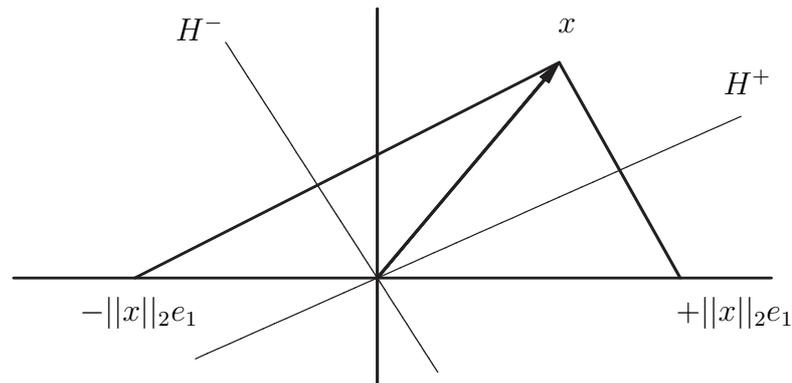


Figura 2.2: Possibilidades de Reflexão.

2.3.3 Escolha do Vetor de Reflexão

Para evitar problemas no cálculo da reflexão, é importante determinarmos uma boa escolha para o vetor v . Observe que através da Definição 2.5 e das equações

(4.9) e (4.10) conclui-se que é indiferente escolher o sinal + ou - no vetor v para anular todas as componentes de Px . Por outro lado, se $\text{sign}(x_1)$ representar o sinal da componente x_1 do vetor x e se o mesmo difere pouco de e_1 então $v = x - \text{sign}(x_1)\|x\|_2 e_1$ tem norma bastante reduzida. Isso implica que o produto escalar $v^T v$ possa ser uma quantidade muito pequena, o que acarretariam problemas no cálculo da matriz P . Por essa razão o vetor v é habitualmente definido a partir de

$$v = x + \text{sign}(x_1)\|x\|_2 e_1,$$

onde

$$\text{sign}(y) = \begin{cases} -1, & \text{se } y < 0, \\ 1, & \text{se } y > 0, \\ 0, & \text{se } y = 0. \end{cases} \quad (2.6)$$

Além disso, como todos os elementos da matriz vv^T são produtos de duas componentes do vetor x , então valores grandes dessas componentes podem implicar a ocorrência de fenômenos de *overflow* (o erro de “overflow” ocorre quando o resultado de uma operação aritmética excede o valor de 3.4028235×10^{38} , ou seja, erro por estouro de memória). Esse problema é resolvido considerando o vetor $x \frac{1}{\|x\|_\infty}$ em vez de x na definição de v . Assim na definição da matriz P , o vetor v é definido por

$$v = x \frac{1}{\|x\|_\infty} + \text{sign}(x_1) \left\| x \frac{1}{\|x\|_\infty} \right\|_2 e_1 \quad (2.7)$$

e portanto, como $Px = \pm\|x\|_2 e_1$, então

$$Px = \text{sign}(x_1) \left\| \frac{x}{\|x\|_\infty} \right\|_2 e_1 = \begin{pmatrix} -\text{sign}(x_1)\alpha \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (2.8)$$

onde

$$\alpha = \left\| \frac{x}{\|x\|_\infty} \right\|_2 = \sqrt{\sum_{i=1}^n \left(\frac{x_i}{v_{max}} \right)^2} \quad (2.9)$$

e

$$v_{max} = \max_i |x_i|.$$

2.4 Cálculo da Fatoração QR

Como visto na Seção 2.2, provaremos a existência da decomposição QR para uma matriz $A \in \mathbb{R}^{n \times n}$. A prova do próximo teorema mostra a existência da fatoração QR quando $m \geq n$, usando reflexão de Householder.

Teorema 2.8 *Seja P a matriz de Householder como na Definição 2.5 e seja $A \in \mathbb{R}^{m \times n}$, com $m \geq n$. Então existem matrizes $Q \in \mathbb{R}^{m \times m}$ e $R \in \mathbb{R}^{m \times n}$ tal que $A = QR$.*

Demonstração:

De fato, suponha que $A \in \mathbb{R}^{m \times n}$ e seja $P_1 = I - \beta_1 v^{(1)} v^{(1)T}$, onde $\beta_1 = \frac{2}{v^{(1)T} v^{(1)}}$. Como $m \geq n$, temos que a matriz de Householder é uma matriz quadrada $m \times m$. Então,

$$v^{(1)} = \left[\frac{a_{11}}{v_{max}} + \text{sign}(a_{11})\alpha, \frac{a_{21}}{v_{max}}, \dots, \frac{a_{n1}}{v_{max}} \right]^T,$$

onde

$$\alpha = \sqrt{\sum_{i=1}^n \left(\frac{x_i}{v_{max}} \right)^2}, \beta_1 = \frac{2}{v^{(1)T} v^{(1)}},$$

assim

$$P_1 A = A^{(2)} = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{m2}^{(2)} & \cdots & a_{mn}^{(2)} \end{pmatrix}.$$

De maneira geral, cada matriz P_k de Householder é dada por $P_k = I - \beta_k v^{(k)} v^{(k)T}$,

$$v^{(k)} = \left[0 \quad \cdots \quad 0 \quad \frac{a_{kk}^{(k)}}{v_{max}} + \text{sign}(a_{kk}^{(k)})\alpha \quad \frac{a_{k+1,k}^{(k)}}{v_{max}} \quad \cdots \quad \frac{a_{nk}^{(k)}}{v_{max}} \right]^T$$

com

$$v_{max} = \max_{i=k, \dots, n} |a_{ik}^{(k)}|,$$

onde

$$\alpha = \sqrt{\sum_{i=k}^n \left(\frac{a_{ik}^{(k)}}{v_{max}} \right)^2}, \beta_k = \frac{2}{v^{(k)T} v^{(k)}}.$$

Assim,

$$P_{n-1}P_{n-2} \cdots P_2P_1A = R$$

donde segue,

$$\begin{aligned} A &= P_1^{-1}P_2^{-1} \cdots P_{n-2}^{-1}P_{n-1}^{-1}R \\ &= P_1P_2 \cdots P_{n-2}P_{n-1}R. \end{aligned}$$

Observe agora que todas as matrizes P_k são ortogonais e portanto, o produto $P_1P_2 \cdots P_{n-2}P_{n-1}$ também é ortogonal, além disso, para cada $k = 1, 2, \dots, n$, P_k é uma matriz simétrica, assim, $P_k^T = P_k$. Além disso, P_k é ortogonal, logo $P_kP_k^T = I$. Isso implica que $P_k^T = P_k^{-1}$ e portanto, $P_k^{-1} = P_k$. Assim tomando $Q = P_1P_2 \cdots P_{n-2}P_{n-1}$ podemos escrever

$$A = QR, \tag{2.10}$$

onde $Q \in \mathbb{R}^{m \times m}$ e $R \in \mathbb{R}^{m \times n}$.

■

Portanto, o Teorema 2.8 prova a existência da decomposição QR quando $m \geq n$. Em relação ao Teorema 2.3 (quando $m < n$), sua demonstração é totalmente análoga ao Teorema 2.8, basta considerarmos para esse caso, uma matriz de Householder $P \in \mathbb{R}^{m \times m}$, onde $m < n$ e obteremos uma matriz $Q \in \mathbb{R}^{m \times m}$ ortogonal e $R = (R_{11} \ R_{12}) \in \mathbb{R}^{m \times n}$, $R_{11} \in \mathbb{R}^{m \times m}$ triangular superior e $R_{12} \in \mathbb{R}^{m \times (n-m)}$ retangular.

Nos limitaremos a partir das próximas Seções a obter resultados somente quando $A \in \mathbb{R}^{m \times n}$ com $m \geq n$, pois essa matriz vai ser nosso objeto de estudo no Capítulo 4.

2.5 Relações da Fatoração QR com o Posto de uma Matriz

Essa Seção objetiva estudar a unicidade da fatoração QR e analisar em que circunstâncias ela ocorre. Veremos que o fato de uma matriz A posuir posto completo, implica em sua unicidade, caso contrário, a fatoração QR não produz necessariamente uma base para $\text{Im}(A)$. Assim, antes de darmos detalhes da unicidade, daremos uma

prova para a suposição (2.1), pois a idéia da fatoração QR, está na construção para uma base para $\text{Im}(A)$.

Teorema 2.9 *Seja $A \in \mathbb{R}^{m \times n}$ e sua respectiva fatoração QR. Suponha $\text{posto}(A) = n$ e considere as seguintes partições $A = [a_1, a_2, \dots, a_n]$ e $Q = [q_1, q_2, \dots, q_m]$, onde cada a_i e q_j são as respectivas colunas de A e Q , $1 \leq i \leq n$ e $1 \leq j \leq m$. Então*

$$\langle a_1, a_2, \dots, a_n \rangle = \langle q_1, q_2, \dots, q_n \rangle, \quad k = 1, \dots, n. \quad (2.11)$$

Demonstração: Ver Anexo. ■

2.5.1 Unicidade da Fatoração QR

A unicidade da fatoração QR pode ser muito útil em certas aplicações, (ver [11, 7]). Portanto, poder determinar quando ela ocorre, pode ser muito importante.

Teorema 2.10 *Seja $A \in \mathbb{R}^{m \times n}$, com $m \geq n$ e suponha que $\text{posto}(A) = n$. Então existe uma única matriz $Q \in \mathbb{R}^{m \times n}$ e $R \in \mathbb{R}^{n \times n}$ tal que Q têm colunas ortogonais e R é triangular superior com todas as entradas da diagonal positiva e*

$$A = QR. \quad (2.12)$$

Demonstração: Ver Anexo. ■

2.5.2 Matriz de Permutação

Se uma matriz A não tem posto completo, então a fatoração QR não necessariamente produz uma base para $\text{Im}(A)$. Esse problema pode ser corrigido implementando o que chamamos de fatoração QR com pivoteamento de colunas. Antes falarmos sobre pivoteamento de colunas, daremos algumas definições úteis que serão necessárias ao entendimento desse método.

Definição 2.11 *Uma Matriz de Permutação Π_{ij} é uma matriz quadrada que se obtém da matriz identidade por troca das suas linhas e colunas i e j .*

Uma matriz de permutação Π tem as seguintes propriedades:

1. $\Pi_{ij} = \Pi_{ij}^T$
2. $\Pi_{ij} = \Pi_{ij}^{-1}$.

Uma matriz de permutação (ver [10]) é o produto de um número finito de matrizes da forma Π_{ij} . Se Π é uma matriz de permutação, então ΠA ($A\Pi$) é uma matriz que se obtém trocando as linhas (colunas) de A correspondentes aos fatores Π_{ij} que constituem Π . Assim, $\Pi^T A \Pi$ é a matriz que difere de A na ordem de um certo número de linhas e colunas de A .

Definição 2.12 *Uma matriz $\Pi^T A \Pi$ é denominada **Permutação Principal** de A .*

2.5.3 Fatoração QR com Pivoteamento de Colunas

Vimos que quando uma matriz A possui posto completo, a fatoração QR é única. Porém, o que dizer quando o posto de uma matriz $A \in \mathbb{R}^{m \times n}$ é incompleto? Essa Subseção tem o propósito de responder a essa questão.

Descrevemos a seguir, como esse problema pode ser resolvido calculando-se a fatoração QR de A com suas colunas permutadas (isto é, com pivoteamento de colunas, [7, 20]). Essa fatoração é dada por $A\Pi = QR$, onde Q e R são matrizes usuais da fatoração QR e Π é uma matriz permutação.

Basicamente, a fatoração QR com pivoteamento de colunas permuta as colunas de matriz A de tal forma os elementos da diagonal da matriz R que são ‘pequenos’ ou nulos, são movidos para a parte inferior da diagonal da matriz R . A idéia dessa fatoração segue abaixo dividida em dois passos.

Primeiro Passo:

Para cada coluna da matriz A , calculamos a $\| \cdot \|_2$ da j -ésima coluna, com $1 \leq j \leq n$. Se a j -ésima coluna tem maior comprimento, isto é, maior $\| \cdot \|_2$, então

as colunas 1 e j são permutadas. Dessa forma, a primeira coluna da matriz $A\Pi$ satisfaz $\|(A\Pi)_1\|_2 = \max_j \|a_j\|_2$, com $1 \leq j \leq n$. Após a determinação da coluna da matriz A com maior 2-norma, a reflexão de householder transforma a primeira coluna da matriz A em $[r_{11} \ 0 \ 0 \ \cdots \ 0]^T$. Como a primeira coluna é que tem maior 2-norma e $Px = \pm \|x\|_2 e_1 = [|r_{11}| \ 0 \ 0 \ \cdots \ 0]^T$ então $|r_{11}|$ têm maior valor absoluto. Em particular, $r_{11} \neq 0$, a menos que A seja identicamente nula.

Segundo Passo:

Terminado o primeiro passo, operamos em uma nova submatriz que denotaremos por $A_{22}^{(k)}$, onde o índice k indica o número de interações e como estamos na segunda iteração, $k = 2$. Essa matriz é obtida por ignorar primeira linha e coluna da matriz A . De maneira totalmente análoga, prosseguimos de acordo com o que foi feito no primeiro passo, apenas com uma ressalva, quando trocamos a coluna de $A_{22}^{(k)}$, a coluna completa deve ser trocada e não apenas a coluna que pertence a $A_{22}^{(k)}$.

Cada passo, opera em uma submatriz $A_{22}^{(k)}$, obtida com uma coluna e uma linha a menos que no passo anterior, justamente como antes, exceto que as colunas trocadas são colunas completas. Observe que, o efeito de troca de colunas corresponde a troca de colunas em A antes de efetuarmos a fatoração QR.

Teorema 2.13 *Seja $A \in \mathbb{R}^{m \times n}$ com $\text{posto}(A) = r \leq n$. Então existem matrizes \widehat{A} , Q e R , tal que $\widehat{A} = A\Pi$ é obtida de A por permutação de colunas, $Q \in \mathbb{R}^{m \times m}$*

é ortogonal, $R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times n}$, $R_{11} \in \mathbb{R}^{r \times r}$ é não-singular e triangular superior.

Demonstração: Ver Anexo. ■

De forma análoga ao teorema acima obtemos decomposição QR com permutação de colunas para uma matriz $A \in \mathbb{R}^{m \times n}$ com $m \geq n$, onde $Q \in \mathbb{R}^{m \times n}$ e $R \in \mathbb{R}^{n \times n}$.

Observações:

1. Seja $A \in \mathbb{R}^{m \times n}$, com $m \geq n$. Considere a decomposição $A\Pi = QR$ onde, $R \in \mathbb{R}^{m \times n}$ é uma matriz triangular superior, $Q \in \mathbb{R}^{m \times m}$ é uma matriz ortogonal e $\Pi \in \mathbb{R}^{n \times n}$ uma matriz de permutação. Com o esquema de pivoteamento de colunas descrito acima podemos escrever

$$A\Pi = QR = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} \quad (2.13)$$

onde os elementos da diagonal da matriz R estão todos dispostos em ordem não-crescente, isto é, $r_{11} \geq r_{22} \geq r_{33} \geq \dots \geq r_{nn}$ e $\|R_{22}\|_2$ pequena, (ver [3]);

2. Se uma matriz $A \in \mathbb{R}^{m \times n}$ e $\text{posto}(A) = r < n$, então a fatoração QR de A não necessariamente produz uma base ortonormal para a $\text{Im}(A)$. Felizmente, a fatoração QR com pivoteamento de colunas pode ser usada para produzir uma base ortogonal para $\text{Im}(A)$.

Pelo Teorema 2.13, existe uma matriz de permutação Π tal que $A\Pi = QR$ e

$$Q^T A\Pi = R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix}$$

onde, $R_{11} \in \mathbb{R}^{r \times r}$ triangular superior, $R_{12} \in \mathbb{R}^{r \times (n-r)}$, Q é uma matriz ortogonal e $\text{posto}(A) = r$. Sabemos que a matriz $A\Pi$ têm suas colunas reordenadas de tal forma que $A\Pi = [a_1, a_2, \dots, a_r, \dots, a_n]$ e $Q = [q_1, q_2, \dots, q_r, \dots, q_m]$. Assim, para $k = 1, \dots, n$ temos que

$$a_i = \sum_{j=1}^{\min\{r,k\}} r_{ij} q_j \in \langle q_1, \dots, q_r \rangle, \quad (2.14)$$

implicando que $\text{Im}(A) = \langle q_1, q_2, \dots, q_r \rangle$.

Capítulo 3

Aplicações da Fatoração QR

3.1 Introdução

Iniciaremos esse Capítulo resolvendo sistema de equações lineares $Ax = b$, onde $A \in \mathbb{R}^{n \times n}$ com $\text{posto}(A) = n$ e veremos que a fatoração QR fornece uma boa solução para esse sistema.

Em seguida, estudaremos o problema de mínimos quadrados (3.5). Mostraremos sob quais condições ocorre a unicidade deste problema. Provaremos que a solução do problema de mínimos quadrados está relacionado com o posto da matriz. Primeiro, trataremos do caso em que uma matriz A tem posto completo e em seguida, quando A não tem posto completo, analisando as duas soluções.

Definiremos o problema de seleção de subconjunto de uma matriz de dados A e forneceremos uma solução para este problema, por meio da fatoração QR com pivoteamento de colunas. Por fim, daremos uma solução aproximada para o sistema sobredeterminado $Ax = b$ usando um subconjunto de k colunas *independentes* da matriz A através de mínimos quadrados e finalizaremos este capítulo com algumas conclusões.

3.2 Sistemas Lineares

Considere o sistema de equações lineares

$$\begin{cases} Ax = b \\ A \in \mathbb{R}^{m \times n} \\ b \in \mathbb{R}^m, m = n. \end{cases} \quad (3.1)$$

e $\text{posto}(A) = n$.

Existem vários métodos disponíveis na literatura para resolver o sistema (3.1), por exemplo, eliminação de Gauss, fatoração LU, dentre outros (ver [7]). Porém, a fatoração QR também é uma boa alternativa para encontrar a solução do sistema (3.1). De fato, seja A como definida em (3.1). Como A tem posto completo existem únicas matrizes Q e R tais que $A = QR$. Portanto, substituindo QR por A temos que $QRx = b$. Multiplicando por Q^T em ambos os membros da igualdade teremos $Q^TQRx = Q^Tb$ e como a matriz Q é uma matriz ortogonal ($Q^TQ = I$), temos $Rx = d$, onde $d = Q^Tb \in \mathbb{R}^{n \times n}$.

Assim, para encontrar a solução para o sistema (3.1), basta resolver o sistema triangular superior $Rx = d$. Como a matriz R é triangular superior, temos $r_{ij} = 0$ para $i > j$, portanto temos um sistema da forma

$$\begin{cases} r_{11}x_1 + r_{12}x_2 + r_{13}x_3 + \cdots + r_{1n}x_n = d_1 \\ r_{22}x_2 + r_{23}x_3 + \cdots + r_{2n}x_n = d_2 \\ r_{33}x_3 + \cdots + r_{3n}x_n = d_3 \\ \vdots \\ r_{nn}x_n = d_n, \end{cases}$$

onde cada x_i e d_i , com $i = 1, \dots, n$ são componentes dos vetores x e d respectivamente.

Tais sistemas são resolvidos por substituições retroativas, através de equações da forma:

$$x_i = \frac{b_i - \sum_{j=i+1}^n r_{ij}x_j}{r_{ii}} \quad (3.2)$$

para todo $i = 1, \dots, n$.

Observe que para encontrar a solução para o sistema (3.1), podemos através da fatoração, de forma equivalente, resolver um sistema triangular superior, fácil de

resolver. E de fato, essa é a grande vantagem de se resolver o sistema (3.1) usando a fatoração QR.

3.3 Problema de Mínimos Quadrados

O método dos mínimos quadrados tem sido o procedimento padrão para a análise de dados a partir do início de 1800. Um famoso exemplo deste método foi quando Gauss com sucesso predisse a órbita do asteroide Ceres, em 1801. Mais de duzentos anos depois, mínimos quadrados continua sendo amplamente utilizado em áreas computacionais, engenharia, etc (ver [1]).

Considere um sistema (*sobredeterminado*) de m equações com n incógnitas, isto é, $m > n$. Simbolicamente, gostaríamos de encontrar um vetor x que satisfaz $Ax = b$, onde $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$. Normalmente, este problema não tem solução. Pois, tal vetor x só existe se $b \in \text{Im}(A)$, mas $b \in \mathbb{R}^m$ e $\text{Im}(A)$ é no máximo de dimensão n . Assim, devemos encontrar um vetor $\hat{x} \in \mathbb{R}^n$ tal que $A\hat{x}$ é a melhor aproximação para b .

Considere o vetor $r = Ax - b$. Supondo que r não é nulo, devemos fazê-lo tão pequeno quanto possível. Desse modo, é natural a escolha de uma norma para medir o tamanho de r , isto é, devemos encontrar um vetor \hat{x} tal que $\|r\|_p = \|A\hat{x} - b\|_p$ seja mínimo, para alguma escolha de p . Porém, diferentes normas produzem diferentes soluções. Normalmente, a norma escolhida é a 2-norma pois possui algumas propriedades importantes em relação as outras, 1-norma e ∞ -norma, (ver [19, 7]). A 2-norma é preservada sob uma transformação ortogonal. Isto significa que podemos encontrar uma matriz Q ortogonal tal que o problema se torna equivalente a minimização $\|Q^T Ax - Q^T b\|_2$ e como veremos, fácil de resolver.

De fato, seja $Q \in \mathbb{R}^{m \times m}$ uma matriz ortogonal e considere o sistema sobredeterminado

$$Q^T Ax = Q^T b, \tag{3.3}$$

obtido através de (3.1) por multiplicação de Q^T em ambos os membros da igualdade. Então:

$$s = Q^T Ax - Q^T b = Q^T (Ax - b) = Q^T r. \tag{3.4}$$

Como Q^T é ortogonal, temos $\|s\|_2 = \|r\|_2$. Logo, $x \in \mathbb{R}^n$ minimiza $\|s\|_2$ se, e

somente se, minimiza $\|r\|_2$, isto é, os dois sistemas sobre-determinados tem a mesma solução.

A técnica que consiste em encontrar um vetor \hat{x} que minimize $\|Ax - b\|_2$, isto é,

Definição 3.1 *Dados uma matriz $A \in \mathbb{R}^{m \times n}$, com $m > n$ e $b \in \mathbb{R}^m$, o **Problema de Mínimos Quadrados** (“Least Squares Problem”) consiste em encontrar $\hat{x} \in \mathbb{R}^n$, tal que, $\|A\hat{x} - b\|_2$ é mínima.*

Usaremos a notação

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2 \quad (3.5)$$

para denotar o problema de encontrar o tal vetor \hat{x} . O vetor

$$r = A\hat{x} - b \quad (3.6)$$

é conhecido como *vetor residual* (ver [1]). Usaremos a notação

$$\|r\|_2 = \|A\hat{x} - b\|_2 \quad (3.7)$$

e denotaremos por *resíduo*. Note que se $\|r\|_2$ é pequeno, dependendo de uma escolha para o vetor \hat{x} (mas em geral, $\|r\|_2$ não é nulo), então podemos prever b através das colunas de A .

Nas Subseções seguintes, estudaremos a solução para o problema de mínimos quadrados através da fatoração QR. Dividiremos esse estudo em dois casos. Primeiro trataremos do caso em que uma matriz A possui posto completo e em seguida, para o caso contrário.

3.3.1 Matrizes de Posto Completo

Teorema 3.2 *Seja $A \in \mathbb{R}^{m \times n}$ e suponha que $\text{posto}(A) = n$, então o problema de mínimos quadrados para o sistema sobre-determinado $Ax = b$ têm uma única solução, o qual pode ser resolvido pelo sistema triangular superior $\hat{R}x = c$, onde $c \in \mathbb{R}^{n \times 1}$,*

$$Q^T b = \begin{pmatrix} c \\ d \end{pmatrix}, \text{ e } Q \text{ e } \hat{R} \text{ são obtidas pela decomposição QR de } A.$$

Demonstração:

Queremos encontrar

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2. \quad (3.8)$$

Temos que $A = QR$, com $Q \in \mathbb{R}^{m \times m}$ e $R \in \mathbb{R}^{m \times n}$. Logo

$$\begin{aligned} \|Ax - b\|_2^2 &= \|QRx - b\|_2^2 \\ &= \|Q^T(QRx - b)\|_2^2 \\ &= \|Rx - Q^Tb\|_2^2. \end{aligned} \quad (3.9)$$

Fazendo $Q^Tb = \begin{pmatrix} c \\ d \end{pmatrix}$ e $R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}$, temos que

$$\|Ax - b\|_2^2 = \|\hat{R}x - c\|_2^2 + \|d\|_2^2, \quad (3.10)$$

onde $c \in \mathbb{R}^{n \times 1}$, $d \in \mathbb{R}^{(m-n) \times 1}$ e $\hat{R} \in \mathbb{R}^{n \times n}$.

Como o termo $\|d\|_2^2$ é independente de x , temos que $\|\hat{R}x - c\|_2^2 + \|d\|_2^2$ será minimizado exatamente quando $\|\hat{R}x - c\|_2^2$ for mínimo. Obviamente, $\|\hat{R}x - c\|_2^2 \geq 0$, com igualdade se, e somente se, $\hat{R}x = c$. Como A tem posto completo, \hat{R} é não-singular. Assim, o sistema $\hat{R}x = c$ tem uma única solução. ■

Portanto, a solução para o problema de mínimos quadrados para o caso em que a matriz A possui posto completo é dada por $\hat{R}x = c$, e seu resíduo é dado por $\|d\|_2^2$.

Vimos na Seção 3.2 deste mesmo capítulo, que o sistema de equações $\hat{R}x = c$ é fácil de se resolver, ou seja, podem ser resolvidos por substituições retroativas. Dessa forma, a fatoração QR se torna uma ferramenta bastante eficaz para encontrar uma solução para o problema de mínimos quadrados no caso em que A possui posto completo.

Observe que através do sistema sobredeterminado $Ax = b$, podemos aplicar a fatoração QR e encontrar $Rx = Q^Tb$. Como $\text{posto}(A) = n$, temos R não-singular e portanto podemos multiplicar R^T em ambos os membros do sistema triangular superior $Rx = Q^Tb$, obtendo $R^T Rx = R^T Q^T b$. Daí, como Q é ortogonal, isto é, $Q^T Q = I$, segue

$$R^T Q^T Q R x = R^T Q^T b.$$

Como $A = QR$ e $A^T = R^T Q^T$, temos que

$$A^T Ax = A^T b. \quad (3.11)$$

Podemos provar os seguintes resultados (ver [1]) para o sistema de equações (3.11):

Afirmção 1:

$A^T A$ é positiva definida, isto é, $x^T (A^T A)x \geq 0$ para todo x , dado que $\text{posto}(A) = n$.

De fato

$$\begin{aligned} x^T A^T Ax &= (Ax)^T (Ax) \\ &= \|Ax\|_2^2 \\ &\geq 0. \end{aligned}$$

Além disso, como A tem posto n temos que se $Ax = 0$, então $x = 0$. Logo, se $x \neq 0$ segue que $Ax \neq 0$. Isto significa que

$$x^T A^T Ax > 0$$

para todo $x \neq 0$, isto é, $A^T A$ é positiva definida.

Afirmção 2:

A matriz $A^T A$ é não-singular. De fato,

$$\begin{aligned} A^T Ax = 0 &\Rightarrow x^T A^T Ax = 0 \\ &\Rightarrow \|Ax\|_2^2 = 0 \\ &\Rightarrow Ax = 0 \\ &\Rightarrow x = 0 \end{aligned}$$

assim, a inversa $(A^T A)^{-1}$ na expressão $(A^T A)^{-1} A^T b$ existe.

Afirmção 3:

Mostremos que \hat{x} satisfaz

$$\|A\hat{x} - b\|_2^2 < \|Ax - b\|_2^2, \quad (3.12)$$

para todo $x \neq \hat{x}$.

De fato,

$$\begin{aligned} \|Ax - b\|_2^2 &= \|(Ax - A\hat{x}) + (A\hat{x} - b)\|_2^2 \\ &= \|Ax - A\hat{x}\|_2^2 + \|A\hat{x} - b\|_2^2 + 2(Ax - A\hat{x})^T (A\hat{x} - b), \end{aligned} \quad (3.13)$$

onde nesta segunda igualdade foi usado que

$$\begin{aligned} \|u + v\|_2^2 &= (u + v)^T(u + v) \\ &= u^T u + 2u^T v + v^T v \\ &= \|u\|_2^2 + \|v\|_2^2 + 2u^T v. \end{aligned}$$

Afirmção 4:

$$(Ax - A\hat{x})^T(A\hat{x} - b) = 0 \tag{3.14}$$

De fato,

$$\begin{aligned} 2(Ax - A\hat{x})^T(A\hat{x} - b) &= (x - \hat{x})^T A^T(A\hat{x} - b) \\ &= (x - \hat{x})^T(A^T A\hat{x} - A^T b) \\ &= (x - \hat{x})^T[A^T A(A^T A)^{-1}A^T b] - A^T b \\ &= (x - \hat{x})^T(A^T b - A^T b) \\ &= (x - \hat{x})^T 0 \\ &= 0, \end{aligned}$$

pois, $\hat{x} = (A^T A)^{-1}A^T b$. Com essa simplificação,

$$\|Ax - b\|_2^2 = \|A(x - \hat{x})\|_2^2 + \|A\hat{x} - b\|_2^2.$$

O primeiro termo é não negativo para todo x , e conseqüentemente

$$\|Ax - b\|_2^2 \geq \|Ax - A\hat{x}\|_2^2.$$

Além disso, temos a igualdade somente se $A(x - \hat{x}) = 0$, isto é, $x = \hat{x}$ (pois, o $\text{posto}(A) = n$). Portanto, concluímos que

$$\|Ax - b\|_2^2 > \|A\hat{x} - b\|_2^2, \forall x \neq \hat{x}.$$

As equações (3.11) são conhecidas como *Equações Normais* e como foi provado através das afirmações acima, também podem ser utilizadas para encontrar uma única solução para o problema de mínimos quadrados para o caso em que A tem posto completo.

3.3.2 Matrizes de Posto Incompleto

Vimos anteriormente que quando A tem posto completo a solução do problema de mínimos quadrados $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$ é única. Por outro lado, se uma matriz A não possui posto completo, provaremos que existem infinitas soluções para o problema de mínimos quadrados, (ver [7]).

Teorema 3.3 *Seja $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$ com $m > n$. Então o problema de mínimos quadrados para o sistema sobredeterminado $Ax = b$ sempre tem uma solução. Se $\text{posto}(A) < n$, então há infinitas soluções.*

Demonstração:

Suponha que $A \in \mathbb{R}^{m \times n}$ e $\text{posto}(A) = r < n$, pois se $r = n$, estamos no caso em que A tem posto completo e portanto já existiria uma única solução para o problema de mínimos quadrados (Teorema 3.2). Usando a fatoração QR com pivoteamento de colunas, temos $A\Pi = QR$, onde $\Pi \in \mathbb{R}^{n \times n}$ é uma matriz de permutação, $Q \in \mathbb{R}^{m \times m}$ com colunas ortogonais e $R \in \mathbb{R}^{m \times n}$ é triangular superior tal que

$$R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix}$$

com $R_{11} \in \mathbb{R}^{k \times k}$ e $R_{12} \in \mathbb{R}^{n \times (n-k)}$.

Assim, podemos aplicar mínimos quadrados para minimizar $\|Ax - b\|_2^2$. Portanto:

$$\begin{aligned} \|Ax - b\|_2^2 &= \|Q^T Ax - Q^T b\|_2^2 \\ &= \|Q^T A(\Pi\Pi^T)x - Q^T b\|_2^2 \\ &= \|(Q^T A\Pi)\Pi^T x - Q^T b\|_2^2 \\ &= \|R\Pi^T x - Q^T b\|_2^2. \end{aligned} \tag{3.15}$$

Seja $\Pi^T x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ e $Q^T b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$, onde $x_1, b_1 \in \mathbb{R}^{k \times 1}$, $x_2 \in \mathbb{R}^{(n-k) \times 1}$ e $b_2 \in \mathbb{R}^{(m-k) \times 1}$. Substituindo em (3.15) teremos:

$$\left\| \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right\|_2^2 = \left\| \begin{pmatrix} R_{11}x_1 - R_{12}x_2 - b_1 \\ b_2 \end{pmatrix} \right\|_2^2. \text{ Assim,}$$

$$\left\| \begin{pmatrix} R_{11}x_1 + R_{12}x_2 - b_1 \\ b_2 \end{pmatrix} \right\|_2^2 = \|R_{11}x_1 + R_{12}x_2 - b_1\|_2^2 + \|b_2\|_2^2. \text{ Logo o mínimo é}$$

atingido quando $\|R_{11}x_1 + R_{12}x_2 - b_1\|_2^2 = 0$, cujo resíduo é dado por $\|b_2\|_2$. Como R_{11} possui inversa temos que $R_{11}x_1 + R_{12}x_2 - b_1 = 0$ ou $x_1 = R_{11}^{-1}(b_1 - R_{12}x_2)$.

Lembrando que $\Pi^T x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ segue que $x = \Pi \begin{pmatrix} R_{11}^{-1}(b_1 - R_{12}x_2) \\ x_2 \end{pmatrix}$.

Portanto a solução depende x_2 , ou seja, para cada escolha de x_2 temos uma única solução para o problema de mínimos quadrados. Logo, infinitas soluções. ■

Observe que o mínimo é atingido quando $R_{11}x_1 + R_{12}x_2 - b_1 = 0$, isto é, $R_{11}x_1 = -R_{12}x_2 + b_1$, onde R_{11} é não-singular. Escolhendo $x_2 = 0$ teremos $R_{11}x_1 = b_1$ e portanto

$$x = \Pi \begin{pmatrix} R_{11}^{-1}b_1 \\ 0 \end{pmatrix}. \quad (3.16)$$

Qualquer solução, onde Ax só envolve no máximo r colunas de A , é chamada de *solução básica*. A solução básica é usada freqüentemente em várias aplicações, por exemplo, quando as colunas de A representam fatores redundantes em um modelo linear, e queremos predizer um vetor de observações b usando r colunas de A , (ver [1]).

A solução básica não é a solução de 2-norma mínima a menos que a submatriz R_{12} é nula, (ver [7]).

Observe que (3.16) é novamente um sistema triangular superior e fácil de se resolver por substituição retroativa.

3.4 O Problema da Seleção de Subconjunto

Suponha que ao invés de usar todas as colunas da matriz de dados A para prever b , gostaríamos de predizê-lo a partir de apenas um subconjunto de suas colunas, eliminando aquelas que sejam redundantes, ou seja, colunas que contém informações supérfluas, que podem ser desconsideradas (por exemplo, colunas linearmente dependentes, ver [7]). O problema é então como selecionar um subconjunto de colunas não-redundantes de A . Como escolher estas colunas é o conhecido como *problema de seleção de subconjunto* e é o assunto desta Seção. Veremos que esse problema pode ser resolvido através da fatoração QR da matriz A , com pivoteamento de colunas, (ver [7, 12]).

Na subseção seguinte, trataremos de comentar algumas importantes propriedades da matriz $A\Pi$ que serão úteis mais adiante, quando discutiremos uma solução para o problema de seleção de subconjunto.

3.4.1 Propriedades da Matriz $A\Pi$

Seja $A \in \mathbb{R}^{m \times n}$, com $m \geq n$. Ao usarmos fatoração QR com pivoteamento de colunas, obtemos a seguinte decomposição para A ,

$$A\Pi = QR, \tag{3.17}$$

onde $\Pi \in \mathbb{R}^{n \times n}$ é uma matriz permutação, $Q \in \mathbb{R}^{m \times n}$ uma matriz ortogonal e $R \in \mathbb{R}^{n \times n}$ é uma matriz triangular superior.

Para entendermos melhor as propriedades da matriz $A\Pi$, trataremos de entender o caso particular em que $m = n = 3$. A fatoração QR com pivoteamento de colunas para esse caso nos fornece uma matriz $Q \in \mathbb{R}^{3 \times 3}$ e $R \in \mathbb{R}^{3 \times 3}$ tal que $A\Pi = QR$. Denotemos por a_1, a_2, a_3 as colunas de $A\Pi$ e q_1, q_2, q_3 as colunas de Q então,

$$\begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} = \begin{pmatrix} q_1 & q_2 & q_3 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{pmatrix}$$

conseqüentemente,

$$\begin{aligned} a_1 &= r_{11}q_1 \\ a_2 &= r_{12}q_1 + r_{22}q_2 \\ a_3 &= r_{13}q_1 + r_{23}q_2 + r_{33}q_3. \end{aligned}$$

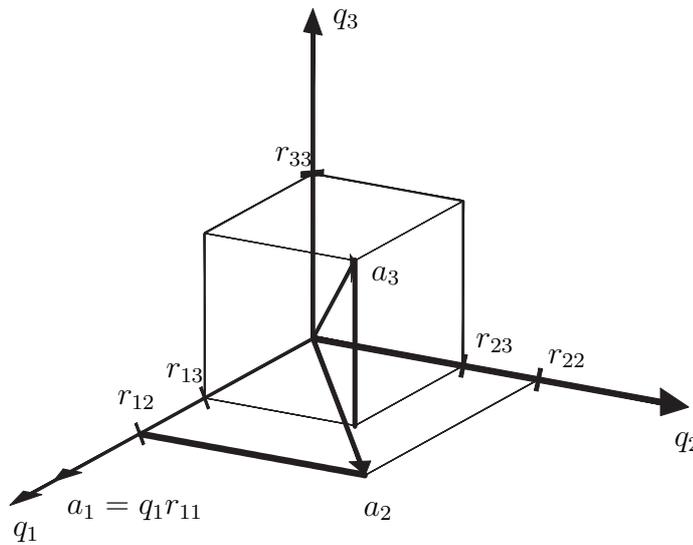


Figura 3.1: *Propriedades da Matriz AII.*

A figura 3.1 representa geometricamente o caso particular de pivoteamento de colunas para uma matriz $A \in \mathbb{R}^{3 \times 3}$, onde os elementos da matriz R foram tomados em módulo.

De acordo com a Subseção 2.5.3 do Capítulo 2, foi observado que quando efetuamos pivoteamento de colunas na matriz de dados A , obtemos uma matriz R com os elementos da diagonal todos não-crescente, ou seja, $r_{11} \geq r_{22} \geq r_{33} \geq \dots \geq r_{nn}$.

Na Figura 3.1, a primeira coluna da matriz AII é escolhida dentre todas as colunas de A que possuem maior 2-norma, (ver Subseção 2.5.3 do Capítulo 2) de forma que q_1 possui a mesma direção de a_1 . A próxima coluna para a matriz AII foi escolhida dentre as colunas restantes a_2 e a_3 , como a coluna que maximiza r_{22} e esta substitui a segunda coluna de AII . Desta forma, a última coluna da matriz AII é justamente a coluna que minimiza r_{33} de R .

De maneira geral, temos portanto que os elementos da diagonal de R , ou seja,

r_{kk} , medem a componente ortogonal de cada coluna k relativa as primeiras $k - 1$ colunas da matriz $A\Pi$, e sempre aparecem em ordem decrescente para $k = 1, \dots, n$. Portanto:

1. A primeira coluna da matriz permutada $A\Pi$ é justamente a coluna de A que possui maior $\| \cdot \|_2$;
2. A segunda coluna de $A\Pi$ é a coluna de A que tem a maior componente em direção ortogonal à primeira coluna da matriz $A\Pi$;
3. Em geral, k -ésima coluna de $A\Pi$ é a coluna de A que tem a maior componente em direção ortogonal às $k - 1$ colunas da matriz $A\Pi$.

3.4.2 Seleção de Subconjunto

A solução para o problema de seleção de subconjunto, depende do método com que a fatoração QR com pivoteamento de colunas é computada. Provavelmente, o melhor algoritmo numérico é o que está baseado na reflexão de Householder, (ver [6]). Como vimos no Capítulo 2, este algoritmo baseia-se no seguinte: Suponha que no k -ésimo passo a fatoração QR com pivoteamento de colunas foi computada e temos $A^{(k)}\Pi^{(k)} = Q^{(k)}R^{(k)}$, onde $k = 1, 2, \dots, n$. No k -ésimo passo antes que $Q^{(k)}$ e $R^{(k)}$ sejam computados, há possibilidade de substituir a k -ésima coluna de $A\Pi$ por uma das colunas restantes $a_{k+1}, a_{k+2}, \dots, a_n$. Se a coluna que maximiza o (k, k) -elemento de R , isto é, r_{kk} , é escolhida para substituir a_k , então haverá uma tendência para que as colunas independentes de A sejam escolhidas de tal forma que estejam representadas nas primeiras k colunas de $A\Pi$, deixando as últimas colunas da matriz $A\Pi$ justamente com as colunas dependentes de A , (ver [6]).

Podemos justificar tal critério de independência das k primeiras colunas da matriz $A\Pi$ fazendo uma análise das propriedades citadas na Subseção anterior. Lá, foi visto que cada coluna da matriz $A\Pi$ é escolhida de tal forma que sua k -ésima coluna é justamente a coluna de A que possui a maior componente em direção ortogonal às $(k - 1)$ colunas da matriz $A\Pi$. Desta forma, tal critério de independência, se equivale na escolha das k colunas de A que possuem as maiores componentes em direção ortogonal. Em outras palavras, quando efetuamos a fatoração QR com pivoteamento de colunas na matriz A , esta será decomposta da seguinte forma

$$A\Pi = QR = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}, \quad (3.18)$$

onde $R \in \mathbb{R}^{n \times n}$ é uma matriz triangular superior, $Q \in \mathbb{R}^{m \times n}$ é uma matriz ortogonal e $\Pi \in \mathbb{R}^{n \times n}$ uma matriz de permutação. Logo, a submatriz R_{22} de R contém exatamente as $(n - k)$ menores (ou nulos) componentes em direção ortogonal. Se os elementos da diagonal de R_{22} são identicamente nulos, certamente as k primeiras colunas de R_{11} fornecem um subconjunto de k colunas da matriz $A\Pi$ que são justamente as colunas de A que possuem as maiores componentes em direção ortogonal. Caso contrário, como todos os elementos da diagonal da matriz R estão todos em ordem não-crescente e R_{22} tem norma reduzida (ver [3]), podemos desprezar suas contribuições e desta forma, selecionamos novamente, um conjunto de k colunas de A que possuem as maiores componentes em direção ortogonal, dados pelos elementos da diagonal da matriz R_{11} que são fornecidos pelas k primeiras colunas de $A\Pi$.

Assim, as primeiras k colunas da matriz $A\Pi$ fornecem uma base para $\text{Im}(A)$ e conseqüentemente as primeiras k colunas da matriz Q são uma base ortogonal para $\text{Im}(A)$.

3.4.3 Seleção de Subconjunto e Mínimos Quadrados

Até o momento, vimos que o problema sobredeterminado $Ax = b$ pode ser resolvido usando a técnica de mínimos quadrados, este por sua vez, possui uma única solução quando a matriz A possui posto completo e infinitas soluções para o caso em que a matriz A possui posto incompleto. Em alguns casos, pode ser vantajoso reduzir a dimensão da matriz de dados para resolver o sistema $Ax = b$, por exemplo, para a geração de animações faciais (ver Capítulo 4).

Selecionado um subconjunto de k colunas independentes da matriz A , dadas por $A\Pi_1$, obtidas através da fatoração QR com pivoteamento de colunas, gostaríamos de

$$\min_{x \in \mathbb{R}^n} \|A\Pi_1 x - b\|_2^2. \quad (3.19)$$

De fato, a matriz A pode ser fatorada na forma $A\Pi = QR$, que pode ser reescrita particionando a matriz $\Pi = [\Pi_1 \ \Pi_2]$, onde $\Pi_1 \in \mathbb{R}^{m \times k}$ e $\Pi_2 \in \mathbb{R}^{m \times (n-k)}$. Assim, $A[\Pi_1 \ \Pi_2] = QR$. Daí segue,

$$A[\Pi_1 \ \Pi_2] = Q \left(\begin{array}{c|c} R_{11} & R_{12} \\ \hline 0 & R_{22} \end{array} \right),$$

tal que $A\Pi_1 \in \mathbb{R}^{m \times k}$ representa um subconjunto de k colunas independentes de A onde, $R_{11} \in \mathbb{R}^{k \times k}$, $R_{12} \in \mathbb{R}^{k \times (n-k)}$ e $R_{22} \in \mathbb{R}^{(n-k) \times (n-k)}$. Conseqüentemente, $A\Pi_1 = QR_{11}$ e $A\Pi_2 \in \mathbb{R}^{m \times (n-k)}$.

Podemos resolver (3.19), que é um problema de mínimos quadrados usando fatoração QR. De fato, substituindo a matriz A por $A\Pi_1$ no Teorema 3.2 e repetindo de forma análoga a demonstração do teorema encontraremos

$$\|R_{11}x - b_1\|_2^2 + \|b_2\|_2^2,$$

onde

$$Q^T b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

e $b_1 \in \mathbb{R}^{k \times 1}$ e $b_2 \in \mathbb{R}^{(m-k) \times 1}$. Logo, o mínimo é atingido quando

$$R_{11}x = b_1. \tag{3.20}$$

Assim, a solução para o problema de mínimos quadrados usando um subconjunto de colunas independentes da matriz A é dada pelo sistema triangular superior $R_{11}x = b_1$ e resíduo $\|b_2\|_2$.

Dessa forma, podemos selecionar um subconjunto de k colunas independentes da matriz A para prever um vetor de observações b , utilizando a técnica de mínimos quadrados.

Gostaríamos de aproximar as $(n - k)$ colunas que não foram selecionadas pelo algoritmo de fatoração QR com pivoteamento de colunas, dadas por $A\Pi_2$, em função de uma combinação linear das colunas independentes de A , dadas pelas primeiras k colunas da matriz $A\Pi$, isto é, $A\Pi_1$. Desse modo, obteríamos uma matriz A_k de posto k , pois estamos predizendo $(n - k)$ colunas da matriz A em termos de k colunas independentes de A e portanto, reduziremos a dimensão (posto) da matriz A . Como veremos no próximo capítulo, essa redução de posto está relacionada com a obtenção do modelo para geração de animações faciais.

Podemos expressar esse problema com a minimização de

$$E_i = \|A\Pi_1 x_i - (A\Pi_2)_i\|_2^2,$$

com $i = 1, \dots, (n - k)$. Como cada $E_i \geq 0$ e cada parcela E_i possui norma mínima, logo $\sum_{i=1}^{n-k} E_i$ terá norma mínima. Isto é equivalente a minimização de

$$\begin{aligned} E &= \sum_{i=1}^{n-k} \|A\Pi_1 x_i - (A\Pi_2)_i\|_2^2 \\ &= \|A\Pi_1 x_1 - (A\Pi_2)_1\|_2^2 + \dots + \|A\Pi_1 x_{n-k} - (A\Pi_2)_{n-k}\|_2^2, \end{aligned}$$

onde cada índice i representa cada uma das $n - k$ colunas de $A\Pi_2$. Assim, pela definição de norma de Frobenius e por (1.5) teremos

$$E = \|A\Pi_1 X - A\Pi_2\|_F^2, \quad (3.21)$$

onde $X \in \mathbb{R}^{k \times (n-k)}$.

Observe que (3.21) é equivalente a minimização de

$$E = \|Q^T(A\Pi_1 X - A\Pi_2)\|_F^2,$$

visto que $Q \in \mathbb{R}^{m \times n}$ é uma matriz ortogonal. Assim,

$$\begin{aligned} E &= \|Q^T(A\Pi_1 X - A\Pi_2)\|_F^2 \\ &= \|Q^T A\Pi_1 X - Q^T A\Pi_2\|_F^2 \\ &= \left\| Q^T Q R_{11} X - Q^T Q \begin{pmatrix} R_{12} \\ R_{22} \end{pmatrix} \right\|_F^2 \end{aligned}$$

pois, $A[\Pi_1 \ \Pi_2] = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$. Então,

$$\begin{aligned} E &= \left\| Q^T Q R_{11} X - Q^T Q \begin{pmatrix} R_{12} \\ R_{22} \end{pmatrix} \right\|_F^2 \\ &= \left\| R_{11} X - \begin{pmatrix} R_{12} \\ R_{22} \end{pmatrix} \right\|_F^2 \\ &= \left\| \begin{pmatrix} R_{11} X - R_{12} \\ -R_{22} \end{pmatrix} \right\|_F^2 \\ &= \|R_{11} X - R_{12}\|_F^2 + \|R_{22}\|_F^2. \end{aligned} \quad (3.22)$$

Portanto, para que (3.22) seja mínimo devemos ter

$$R_{11} X = R_{12}, \quad (3.23)$$

onde sua parte residual é dada por $\|R_{22}\|_F$.

3.5 Discussão

A fatoração QR é de fato é uma ferramenta eficiente. Através da fatoração QR, deduzimos as equações normais, encontramos uma solução para o sistema de equações lineares $Ax = b$, onde $A \in \mathbb{R}^{n \times n}$ e para o problema de mínimos quadrados $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$, com $A \in \mathbb{R}^{m \times n}$, onde A possui ou não posto completo. Em ambos os casos a solução é fornecida com a resolução de um sistema triangular superior, que é fácil de resolver.

Mostrou-se eficaz na solução para o problema de seleção de subconjunto, fornecendo um subconjunto de colunas independentes de uma matriz A , através de pivoteamento de colunas.

Entretanto, é claro que existem outros métodos para encontrar soluções para os problemas propostos nesse capítulo (ver [7]), porém, para os propósitos do capítulo 4, a fatoração QR mostrou-se eficaz, permitindo resolver os principais problemas para se chegar em um modelo para geração de animações faciais.

Capítulo 4

Aplicação a Animação Facial

4.1 Introdução

Neste Capítulo, mostraremos como a fatoração QR pode ser utilizada na obtenção de um modelo e produzir animações computadorizadas da fala com um nível aceitável de realismo. O algoritmo permite identificar um subconjunto de marcadores faciais independentes. Esse subconjunto pode ser utilizado posteriormente, como uma base para prever o movimento de pontos faciais arbitrários.

Investigaremos as trajetórias reconstituídas pelo algoritmo, mostrando que o erro médio obtido é relativamente pequeno e geraremos computacionalmente animações faciais. A maioria dos algoritmos que foram implementados nessa dissertação estão disponíveis no Apêndice A. O software escolhido para os cálculos foi o Matlab.

4.2 Dados

Os dados consistem em registros da posição 3D de 57 marcadores distribuídos na face de um sujeito (Figura 4.1). Foram colhidos com uma frequência de amostragem de 120 Hz, enquanto o sujeito pronunciava uma sequência de 40 frases (ver Tabela 4.6 no Apêndice B) em inglês [12, 13]. As primeiras 30 frases foram usadas para construir o modelo, e as 10 últimas para validar o modelo e gerar animações. Antes de pronunciar cada uma das 40 frases, o sujeito adotou uma posição de descanso (neutra), para que fosse obtido o deslocamento (separado) de cada marcador.

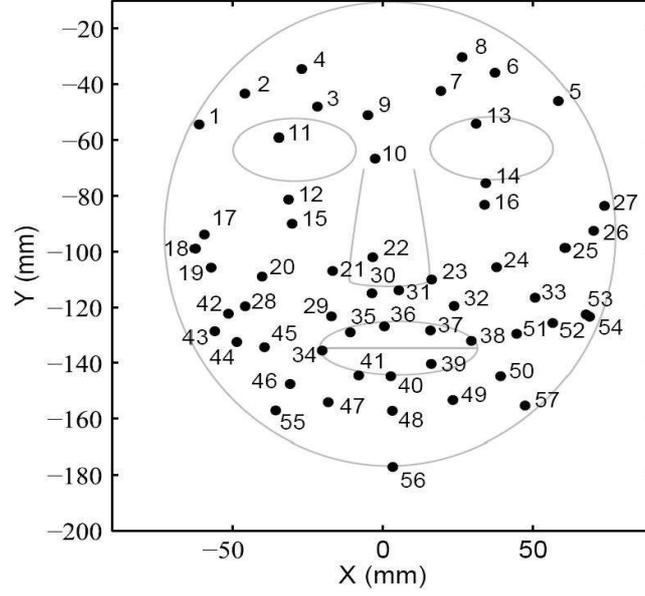


Figura 4.1: Posição dos marcadores faciais.

4.3 Pré-processamento

Primeiramente, os deslocamentos de cada marcador foram calculados relativos à posição neutra inicial e foram armazenados em uma matriz $B_{m \times n}$, onde $m = 16106$ e $n = 171$. A matriz B possui $57 \times 3 = 171$ colunas, onde o fator 3 representa as coordenadas espaciais $\{x, y, z\}$ de cada marcador.

Parte dos dados da matriz B foram reagrupados em uma nova matriz $\hat{B} \in \mathbb{R}^{\hat{m} \times n}$ com as 30 primeiras frases, onde $\hat{m} = 11975$ e $n = 171$ (ver Seção 4.5).

Podemos representar a matriz \hat{B} como segue

$$\hat{B} = \begin{pmatrix} x_{1,1} & y_{1,2} & z_{1,3} & x_{1,4} & y_{1,5} & z_{1,6} & \cdots & x_{1,169} & y_{1,170} & z_{1,171} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ x_{m,1} & y_{m,2} & z_{m,3} & x_{m,4} & y_{m,5} & z_{m,6} & \cdots & x_{m,169} & y_{m,170} & z_{m,171} \end{pmatrix}$$

onde $(x_{11} \cdots x_{m1})^T$, $(y_{12} \cdots y_{m2})^T$ e $(z_{13} \cdots z_{m3})^T$ representam os deslocamentos do primeiro marcador, $(x_{14} \cdots x_{m4})^T$, $(y_{15} \cdots y_{m5})^T$ e $(z_{16} \cdots z_{m6})^T$ para o segundo marcador e de forma análoga para o restante dos marcadores.

Como veremos na Seção 4.7, devemos aproximar o deslocamento de cada marcador que não foi selecionado pelo algoritmo de fatoração QR como uma combinação

linear de k marcadores selecionados. Seja P_s o deslocamento obtido por combinação linear desses k deslocamentos. Então

$$P_s = a_{1s}P_1 + a_{2s}P_2 + \cdots + a_{ks}P_k, \quad (4.1)$$

a_{ks} são escalares. Por simplicidade, para obtermos os deslocamentos de cada marcador, vamos concatenar os deslocamentos dos marcadores em único marcador. Dessa forma teremos

$$\begin{pmatrix} x_s \\ y_s \\ z_s \end{pmatrix} = a_{1s} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + a_{2s} \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} + \cdots + a_{ks} \begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix}. \quad (4.2)$$

Portanto, os deslocamentos de cada marcador foram representados em um única coluna da matriz de dados como segue

$$x = \begin{cases} x_{1r} \\ \vdots \\ x_{mr} \end{cases} \\ y = \begin{cases} y_{1j} \\ \vdots \\ y_{mj} \end{cases} \\ z = \begin{cases} z_{1l} \\ \vdots \\ z_{ml} \end{cases},$$

onde $r = \{1, 4, 7, \dots, (n-2)\}$, $j = \{2, 5, 8, \dots, (n-1)\}$ e $l = \{3, 6, 9, \dots, n\}$.

Assim, deslocamentos dos marcadores nas 30 primeiras frases foram concatenados e arranjadas em uma matriz de dados $A \in \mathbb{R}^{M \times N}$, onde $N = \frac{n}{3}$ é o número de marcadores (57) e $M = 3 \times m$. A fatoração QR com pivoteamento de colunas foi aplicada a matriz A , usando o software Matlab.

4.4 Posto Numérico da Matriz de Dados

Lucero et al. no seu artigo [12] obteve o posto numérico da matriz de dados com análise baseada na decomposição QR com pivoteamento de colunas. Mostraremos

nesta Seção que a mesma análise pode ser feita com decomposição de valores singulares.

Como visto no Capítulo 1, o posto numérico de uma matriz pode ser calculado analisando os seus respectivos valores singulares. A Figura 4.2 mostra os elementos da diagonal da matriz Σ . O pequeno valor entre os elementos $\sigma_{57,57} = 0.07$ em relação ao $\sigma_{56,56} = 18.88$ indica posto 56 para matriz de dados A . De fato, pelo Teorema 1.22 temos que A tem posto numérico $(\delta, \epsilon, r)_2$ se, e somente se,

$$\sigma_r \geq \delta > \epsilon \geq \sigma_{r+1}. \quad (4.3)$$

Em determinadas aplicações (ver Capítulo 1), aparecem matrizes cujos valores singulares menores deveriam ser nulos, mas não o são devido erros de arredondamentos, etc (ver [7]). No que se refere aos dados, houve uma pequena margem de erro que chega a aproximadamente 0.5mm (ver [14]). Desse modo, se escolhermos $r = 56$ temos $\sigma_r = \sigma_{56} = 18.88$ e $\sigma_{r+1} = \sigma_{57} = 0.07$. Portanto, $\delta = 1.0$ satisfaz a Definição 1.20 e daí

$$18.88 \geq 1.0 > 0.5 \geq 0.07. \quad (4.4)$$

Portanto, podemos supor posto numérico 56 e identificar a dimensão dos dados.

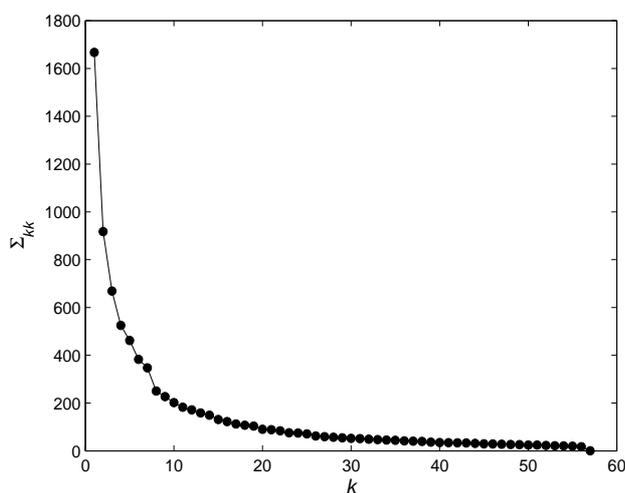


Figura 4.2: Elementos da diagonal da Matriz Σ .

A figura 4.3 mostra os elementos da diagonal da matriz R , r_{kk} , que medem a componente ortogonal de cada coluna k relativo as primeiras $k - 1$ anteriores, e

aparecem em ordem decrescente para $k = 1, \dots, 57$. Observe que há uma semelhança entre os elementos da diagonal das matrizes R e Σ . Ambas as têm diagonal em ordem decrescente e apresentando uma abertura entre os elementos $r_{56,56} = 25.41$ e o $r_{57,57} = 0.01$ que pode ser usado para determinar o posto numérico da matriz de dados, (ver [6]).

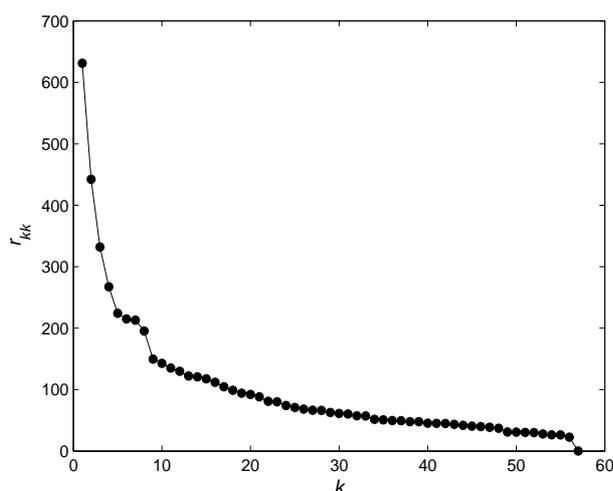


Figura 4.3: *Elementos da diagonal da Matriz R.*

4.5 Alguns Resultados da Fatoração QR

Em seu artigo [12], Lucero et al. observou que os dados estabilizavam para os primeiros 10 valores de r_{kk} , normalizados relativos a norma de R . Os valores estabilizam para um conjunto de aproximadamente 15 sentenças concluindo que qualquer conjunto de dados maior, é bastante seguro para construir um modelo, o que justificou adoção de 30. Usaremos esta mesma metodologia aqui.

A escolha da quantidade de sentenças pode ser justificada através de uma análise na Figura 4.4. Lá, os primeiros 12 valores de r_{kk} estão normalizados relativos à norma de R , em função da quantidade de amostras nos dados. Os valores estabilizam antes de atingir o total de amostras disponíveis (aproximadamente 15^a frase, já que cada frase ocupa aproximadamente em média 500 linhas da matriz de dados), o que justifica a adoção de 30 frases para construir o modelo. Dessa forma, sobram 10 frases para testar e validar o modelo.

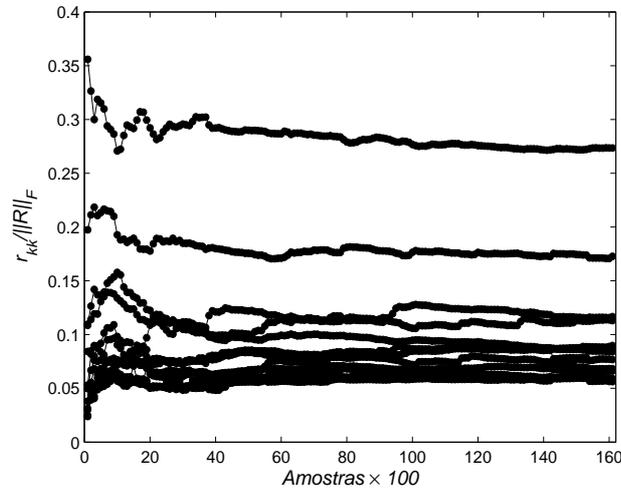


Figura 4.4: Valores normalizados dos 12 primeiros elementos na diagonal de R em função da quantidade de amostras nos dados.

Na Figura 4.4, ocorre estabilidade dos dados aproximadamente a partir da 15ª frase. Porém, quando permutamos as linhas de A aleatoriamente (Figura 4.5) observamos que a estabilidade é alcançada mais rapidamente, ou seja, aproximadamente entre a linha 1000 e 2000, o que nós dá aproximadamente 3 sentenças.

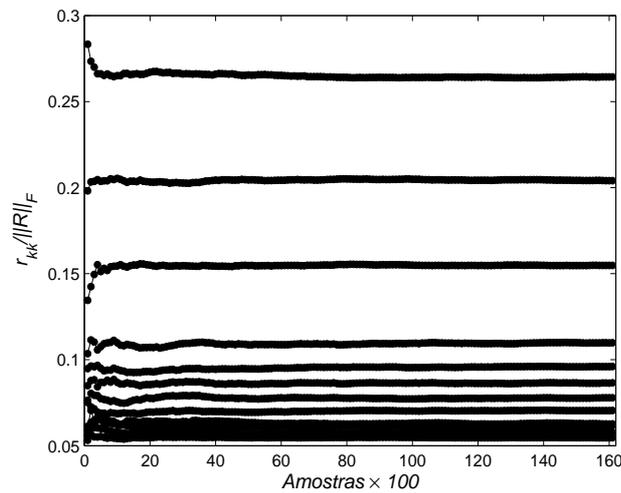


Figura 4.5: Valores normalizados dos 12 primeiros elementos na diagonal de R em função da quantidade de amostras nos dados permutados aleatoriamente.

De fato, em cada linha da matriz estão representados os deslocamentos de todos os marcadores. Assim, em um determinado momento, a matriz de dados poderá ter

ou não todos os detalhes dos movimentos manifestados na face do indivíduo através dos marcadores: Por exemplo, a fala, piscar dos olhos, sorriso, etc. Isto é, nem sempre em determinadas frases, o indivíduo manifesta todas estas expressões faciais. Portanto, em alguns casos, há necessidade de mais de uma frase até que todas estas características dos movimentos sejam incorporadas. Logo, quando dizemos que os r_{kk} estão se estabilizando, na verdade estamos dizendo que o conjunto de linhas que foram selecionadas para aplicar a fatoração QR, é suficiente, no sentido de que todos os movimentos da face capturados pelos marcadores estão agregadas nesse conjunto de sentença. Daí, um conjunto 30 sentenças é suficiente para que isso aconteça (Figura 4.4). Por outro lado, quando permutamos aleatoriamente as linhas da matriz de dados, temos um cruzamento de informações, isto é, expressões faciais que antes eram manifestas em frases posteriores, por exemplo, agora se misturam. Assim, movimentos faciais que antes eram obtidos com um conjunto de 30 frases por exemplo, não é mais necessário e portanto, a estabilidade ocorre bem antes (ver Figura 4.5) em comparação com a Figura 4.4.

4.6 Escolha dos Marcadores Principais e suas Res- pectivas Regiões de Influência

Em [12] é apresentado uma Tabela com 10 colunas. Cada coluna possui 12 marcadores selecionados pelo algoritmo. Um conjunto de dados de 30 sentenças foi usado em todas as tentativas. Na primeira tentativa, as primeiras 30 sentenças do conjunto foram usadas, e nas 9 tentativas restantes, as sentenças foram aleatoriamente selecionadas. As colunas selecionadas tiveram uma pequena mudança a cada nova tentativa.

Neste trabalho, faremos um outro tipo de análise. A Tabela 4.1 mostra experimentos feitos com k linhas da matriz de dados, onde k varia de 1000 a 11975 linhas na matriz A . Cada coluna da tabela representa os marcadores selecionados pelo algoritmo na k -ésima linha. E como já tínhamos observado na Figura 4.4, os dados se estabilizam a partir da 7^a coluna, que corresponde a 15^a sentença aproximadamente.

A Tabela 4.2 mostra os marcadores que o algoritmo selecionou com as linhas permutadas aleatoriamente na matriz de dados A , onde k varia 1000 a 11975. Cada coluna da tabela representa os marcadores selecionados pelo algoritmo na k -ésima

Ordem	Marcadores Seleccionados											
	1	2	3	4	5	6	7	8	9	10	11	12
1	40	48	48	40	40	40	40	40	40	40	40	40
2	34	34	34	34	34	34	02	34	34	34	34	34
3	51	51	51	38	38	38	38	38	38	38	38	38
4	20	20	20	2	2	2	2	2	2	2	2	2
5	36	02	03	36	36	20	20	20	36	36	36	36
6	32	07	36	20	20	36	08	36	36	36	36	36
7	11	36	39	49	49	08	36	36	20	08	08	20
8	56	39	11	07	07	49	49	49	49	49	49	49
9	12	57	06	52	52	11	13	13	13	13	13	13
10	44	11	44	13	13	52	52	12	12	52	52	52
11	39	32	32	23	54	12	12	54	54	54	54	54
12	16	47	47	57	23	54	54	52	47	56	47	47
Amostras ×100	10	20	30	40	50	60	70	80	90	100	110	119.75

Tabela 4.1: Os 12 primeiros marcadores seleccionados pelo algoritmo com k linhas na matriz de dados.

linha. Como observamos na Figura 4.5, os dados se estabilizam a partir da 2ª coluna ou aproximadamente 5ª sentença.

Ordem	Marcadores Seleccionados											
	1	2	3	4	5	6	7	8	9	10	11	12
1	40	40	40	40	40	40	40	40	40	40	40	40
2	34	34	34	34	34	34	34	34	34	34	34	34
3	38	38	38	38	38	38	38	38	38	38	38	38
4	2	2	2	2	2	2	2	2	2	2	2	2
5	36	36	36	36	36	36	36	36	36	36	36	36
6	20	08	06	06	06	06	06	20	20	06	06	06
7	49	20	20	20	20	20	20	06	06	20	20	20
8	06	49	49	49	49	49	49	49	49	49	49	49
9	11	13	13	13	13	13	13	13	13	13	13	13
10	47	52	52	52	52	52	52	52	52	52	52	52
11	54	54	54	54	54	54	54	54	54	54	54	54
12	47	47	47	47	47	47	47	47	47	47	47	47
Amostras ×100	10	20	30	40	50	60	70	80	90	100	110	119.75

Tabela 4.2: Os 12 primeiros marcadores seleccionados pelo Algoritmo com k linhas permutadas aleatoriamente.

A partir do momento em que os dados começam a estabilizar, há poucas mudanças entre as colunas seleccionadas pelo algoritmo. Assim, usando os resultados

das Tabelas 4.1 e 4.2 selecionamos um conjunto de 12 marcadores. A tabela 4.3 representa essa seleção.

Ordem	Marcador
1	40
2	34
3	38
4	2
5	36
6	06
7	20
8	49
9	13
10	52
11	54
12	47

Tabela 4.3: *Os 12 primeiros marcadores selecionados pelo algoritmo.*

Nos itens a seguir, estão relacionados os 12 marcadores que foram selecionados pelo algoritmo e suas respectivas regiões de influência para cada um dos marcadores.

- O primeiro marcador selecionado é o 40, no centro do lábio inferior (Figura 4.1), que tem o maior deslocamento;
- Os próximos dois marcadores são o 34 e o 38, em ambos cantos dos lábios;
- O quarto marcador é o 02, próximo da sobrancelha direita;
- Na 5ª posição aparece o marcador 36 e situá-se no centro do lábio superior. O marcador 06 aparece em seguida e está situado acima do olho esquerdo.
- O marcador 20 aparece na 7ª posição e se encontra ao lado direito do nariz;
- O marcador 49 aparece em seguida e encontra-se na parte inferior dos lábios, a esquerda;
- O marcador 13 representa a ação de piscar do olho esquerdo;
- Na 10ª posição aparece o marcador 52 e este se encontra ao lado esquerdo do rosto;

- O marcador 54, na pálpebra esquerda, aparece na 11^a posição, e encontra-se bem próximo do marcador 52;
- Por último, aparece o marcador 47 e encontra-se na parte inferior dos lábios, a direita.

Os valores r_{kk} associados aos 4 primeiros marcadores são bem maiores em comparação com os outros (ver Figura 4.2), o que sugere que seus movimentos determinam grande parte da cinemática facial geral.

Após selecionar os marcadores (colunas) principais, calculamos com o auxílio de mínimos quadrados das colunas restantes resolvendo

$$R_{11}X = R_{12}, \quad (4.5)$$

como na Seção 3.4 do Capítulo 3. Como um exemplo numérico, usamos os resultados da Tabela 4.3. Em [12], Lucero et al. adotou uma base constituída de 09 marcadores. Nesta dissertação, adotamos base de 12 marcadores (ver Seção 4.9) que inclui até o marcador 47. As Figuras 4.6 e 4.7 mostram os resultados desse ajuste. Lá, os coeficientes apropriados computados para os marcadores secundários foram estendidos a outros pontos faciais por interpolação cúbica.

As regiões associadas a cada um dos marcadores principais incluem as sub-regiões positivas (vermelho) e negativas (azuis), onde o movimento está no mesmo sentido e no oposto, respectivamente, ao movimento do marcador principal. As regiões aparecem no mesmo número e posição similar em ambos os lados do rosto, embora tenham uma grande assimetria. A respeito das pálpebras, note que embora um dos dois marcadores da pálpebra seja um marcador principal, ambos têm os pesos similares, indicando padrões quase iguais do movimento.

O gráficos das Figuras 4.6 e 4.7 representam as linhas da matriz X obtidas pela equação (4.5), interpoladas a toda a superfície facial. Cada gráfico mostra a região controlada pelo marcador i , que corresponde a linha i de X .

Do mesmo modo, podemos fazer gráficos similares (Figuras 4.8 e 4.9), porém nesse caso, considerando a matriz R . Sabemos que a primeira linha de R contém as projeções de cada coluna da matriz A (com as colunas já permutadas) na direção da primeira coluna da matriz Q (que tem maior norma). A segunda linha de R

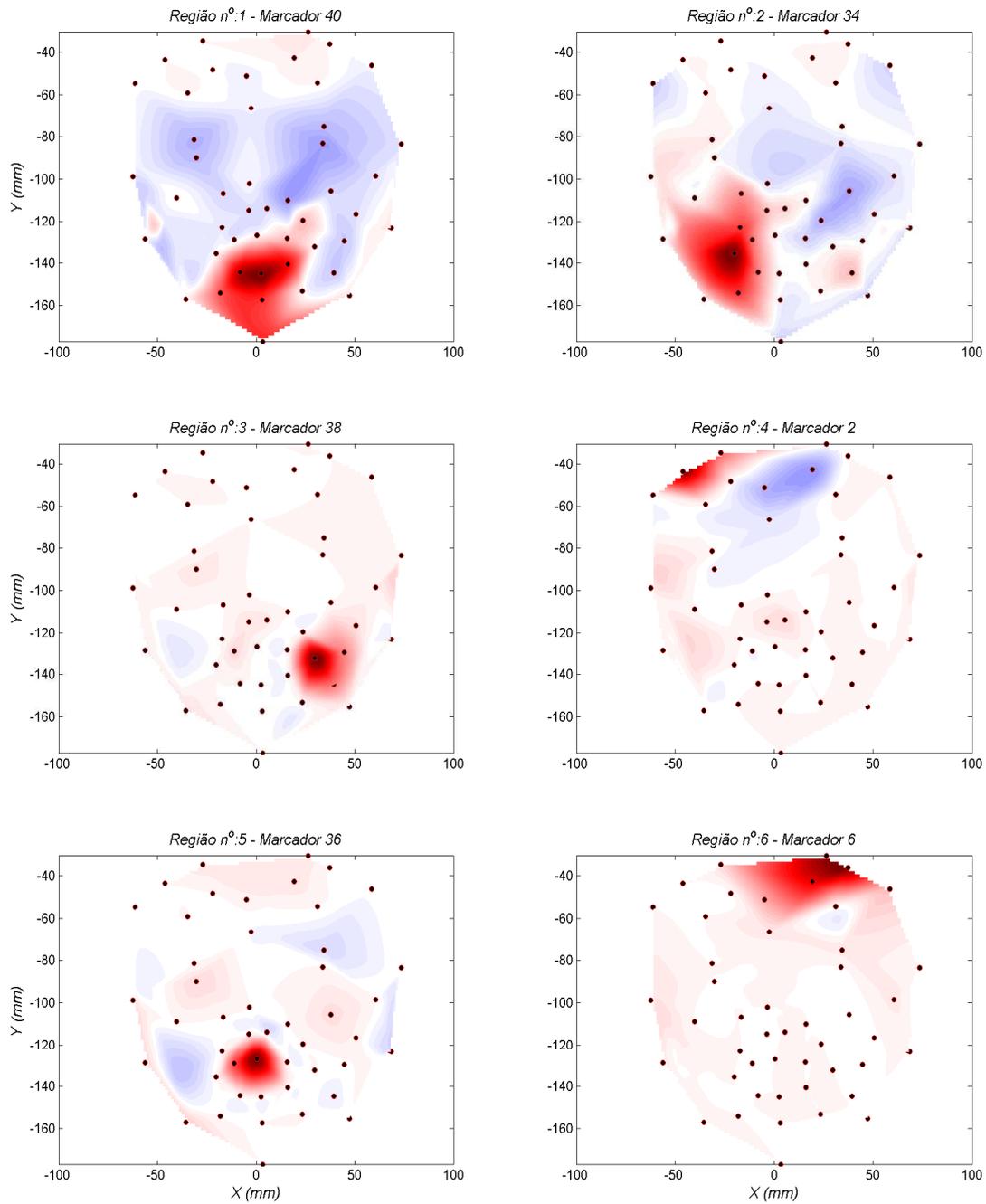


Figura 4.6: Regiões faciais ilustradas para os marcadores 40, 34, 38, 02, 36 e 06.

contém as projeções de cada coluna de A em direção ortogonal à primeira coluna da matriz Q . Em geral, a k -ésima linha de R contém as projeções de cada coluna de A em direção ortogonal à $k - 1$ colunas da matriz Q . Então, gráficos das linhas de R

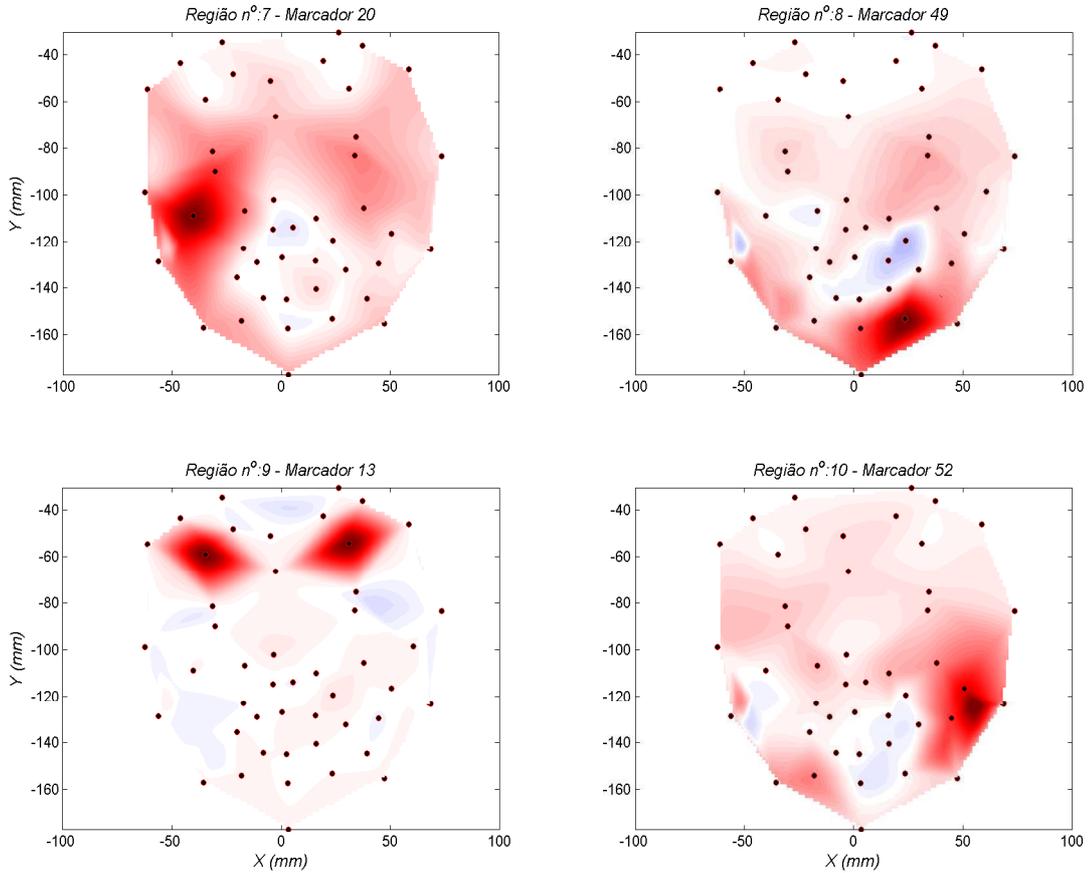


Figura 4.7: Regiões faciais ilustradas para marcadores 20, 49, 13 e 52.

mostram a projeção do deslocamento de cada ponto facial nas direções ortogonais definidas pela matriz Q . Isto é, os gráficos (regiões mais escuras) mostram quais os padrões regionais de deformação facial nessas direções ortogonais.

4.7 Geração de Animações Faciais

Como exemplo de animação facial, usaremos a frase 39 do conjunto de sentenças disponível no Apêndice B (ver Tabela 4.6). As animações faciais de movimentos arbitrários podem ser produzidas, controlando o movimento dos marcadores principais com sinais apropriados.

Seja $P_1 \in \mathbb{R}^{m \times k}$ a matriz com os deslocamentos dos marcadores principais relativos à posição neutra inicial. Uma vez computados todos os coeficientes da matriz

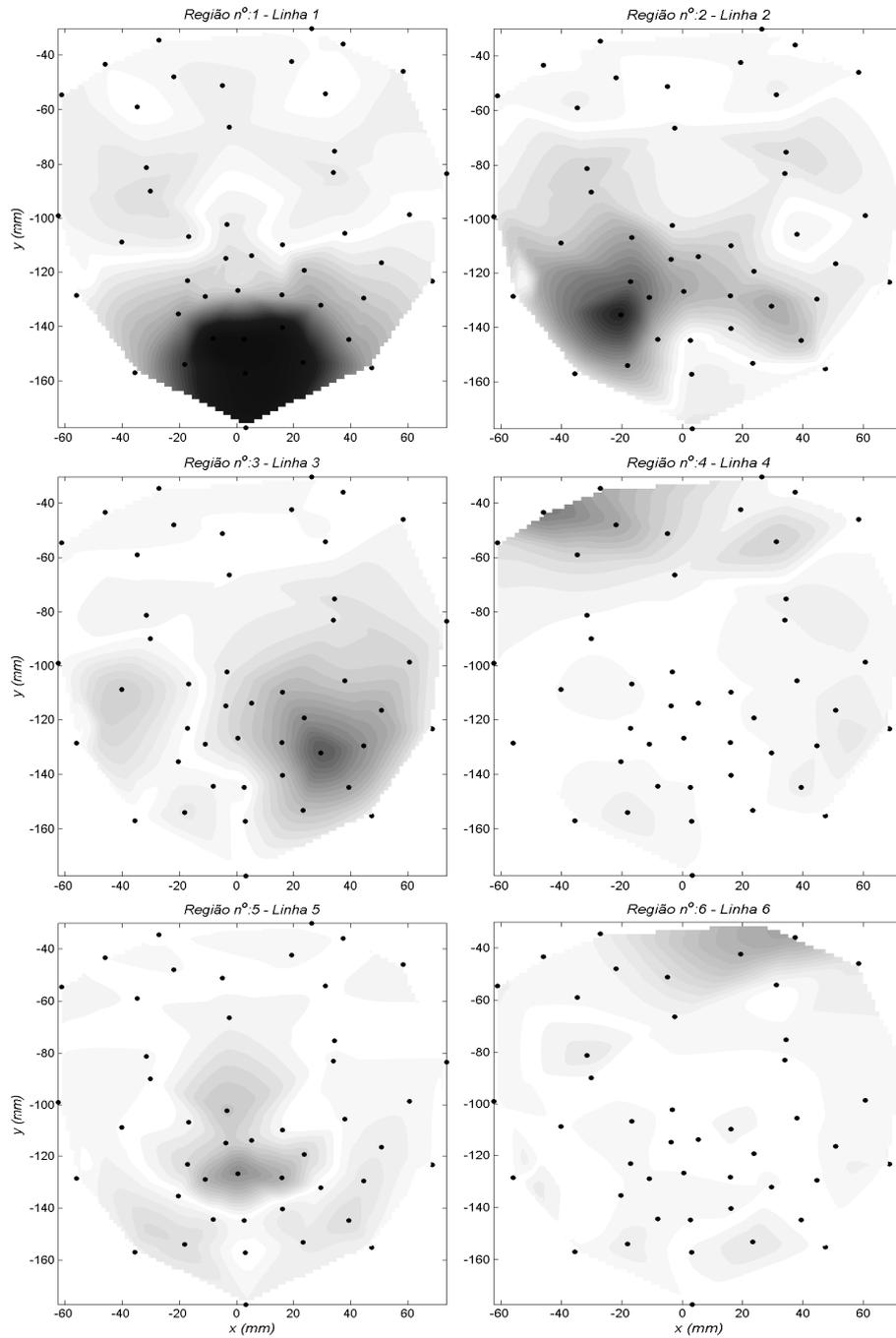


Figura 4.8: Regiões de deformação facial nas direções ortogonais, linhas 1 a 6 de R .

X , calculamos o deslocamento para os marcadores secundários, resolvendo

$$P_2 = P_1 X. \quad (4.6)$$

Calculados os deslocamento para os marcadores secundários, a posição inicial de

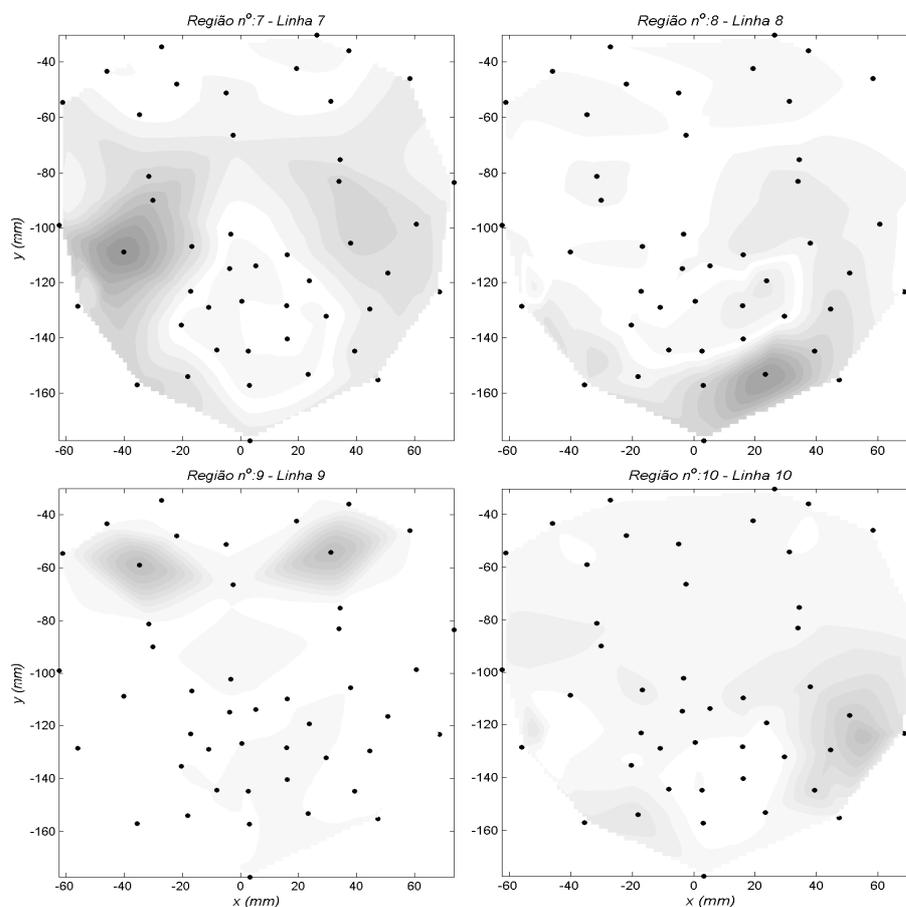


Figura 4.9: *Regiões de deformação facial nas direções ortogonais, linhas 7 a 10 de R.*

todos os marcadores é introduzida para obter sua posição no espaço (3D). Desse modo, criamos um malha facial equiespaçada contendo 2500 pontos faciais arbitrários. Intercalamos esses pontos faciais através da interpolação *cúbica*, isto é, cada curva entre dois pontos adjacentes consiste num polinómio de terceira ordem cujos coeficientes foram obtidos resolvendo sistema de equações lineares montadas a partir condições preestabelecidas.

Finalmente, nós produzimos animações faciais para a sentença 39, em formato AVI (ver Algoritmo L do Apêndice A). Outros exemplos de animações que usam esta técnica podem ser visualizadas em <http://vargas.mat.unb.br/>. A Figura 4.10 mostra o quadro inicial das animações geradas usando esse método.

Em geral, as animações são visualmente realistas, sem nenhuma distorção perceptível nos padrões de movimento.

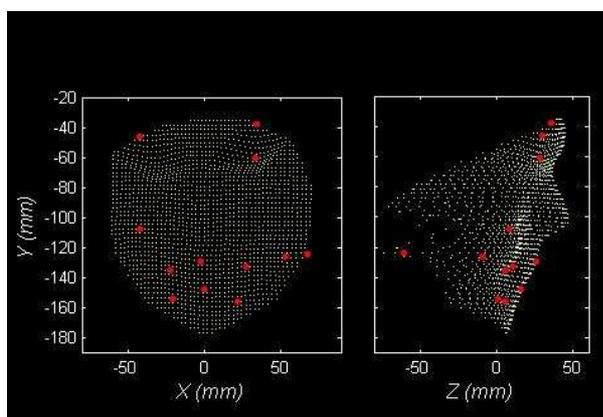


Figura 4.10: Quadro inicial das animações.

4.8 Análise de Erros nas Trajetórias Computadas

Como já comentado, deixamos 10 frases para testar e validar o modelo e como exemplo numérico, escolhemos a frase 39 do conjunto de sentenças.

Calculada a matriz P_2 que contém os deslocamentos para os marcadores secundários, como em (4.6), podemos medir o erro, comparando os deslocamentos dos marcadores secundários que foram reconstituídos pelo algoritmo, com os deslocamentos originais dos marcadores. Para calcular o erro, adotamos o *Erro Médio Quadrático* dado pela seguinte fórmula

$$E = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2 + (z_i - \bar{z}_i)^2}, \quad (4.7)$$

onde x_i, y_i, z_i são componentes dos deslocamentos dos marcadores originais, \bar{x}_i, \bar{y}_i e \bar{z}_i componentes dos deslocamentos obtidos através da reconstituição dos marcadores pelo algoritmo.

A Figura 4.11 mostra a comparação entre a trajetória real (linha em traços) e a nova trajetória que foi reconstituída pelo algoritmo (linha cheia), no presente caso, para o marcador 28. Observe que o marcador 28 encontra-se na parte inferior do rosto (ver Figura 4.1), uma região onde há muito deslocamento, ainda assim, há grande semelhança entre as trajetórias originais e as reconstituídas pelo algoritmo, mostrando que a fatoração QR é de fato um bom algoritmo para reconstrução das trajetórias.

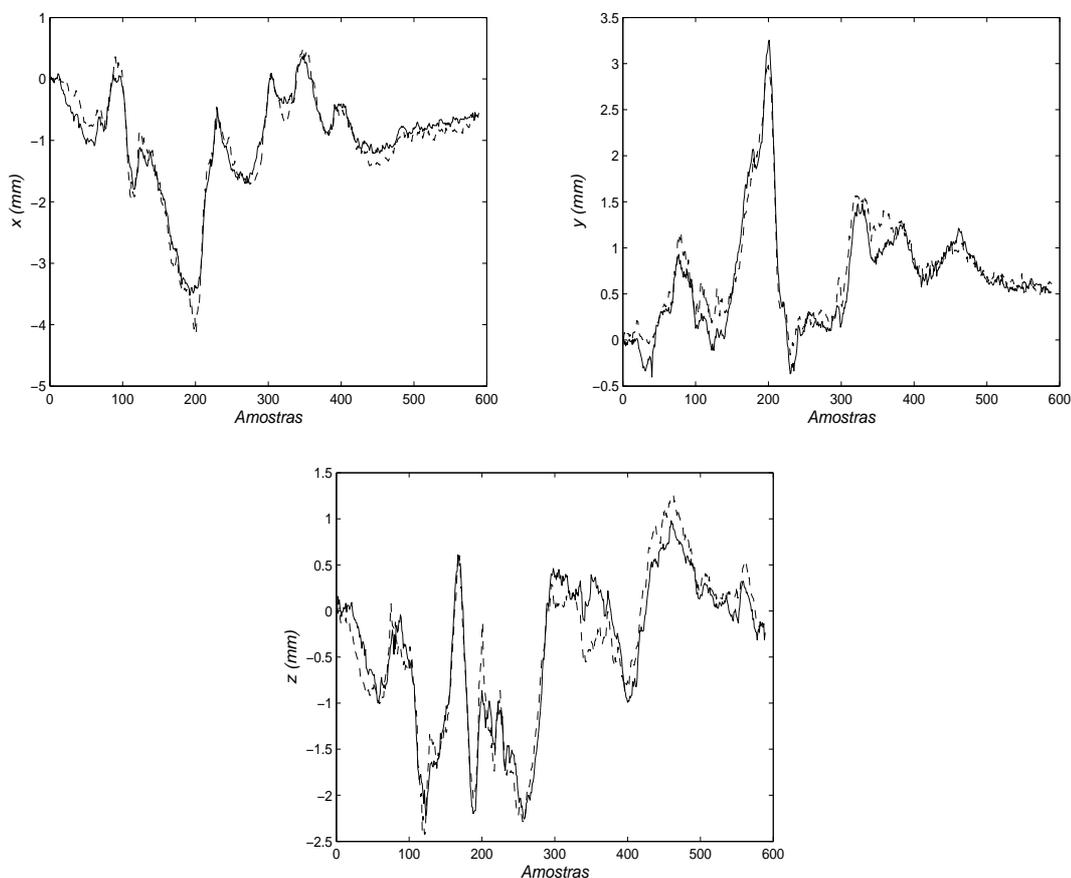


Figura 4.11: *Exemplo de trajetória real (linha de traços) e reconstruída pelo algoritmo (linha cheia), para o marcador 28.*

A Tabela 4.4 contém os erros obtidos através da reconstituição dos deslocamentos dos marcadores pelo algoritmo. Em relação a Tabela 4.4 destacamos:

1. O erro calculado para o 11º marcador é quase nulo, na ordem de 1.07×10^{-5} mm. Se analisarmos a Figura 4.1 veremos que o 11º marcador é simétrico ao 13º marcador, que é um marcador selecionado pelo algoritmo, portanto ambos possuem pesos similares, e como o 13º possui erro nulo, isso justifica o erro do 11º marcador ser tão pequeno;
2. Além do marcador 11 que é simétrico ao marcador 13, o menor erro é justamente para o marcador 10, aproximadamente 0.30mm e de fato já era esperado, pois esse marcador encontra-se justamente na parte superior na sombrancelha direita, local onde há pouco deslocamento dos marcadores;

Marcadores	Erro	Marcadores	Erro	Marcadores	Erro
01	0.80	20	0.00	39	1.05
02	0.00	21	0.76	40	0.00
03	0.40	22	0.66	41	0.68
04	0.50	23	0.93	42	0.93
05	0.66	24	0.61	43	0.74
06	0.00	25	0.49	44	0.82
07	0.70	26	0.38	45	0.58
08	0.55	27	0.56	46	0.65
09	0.49	28	0.35	47	0.00
10	0.30	29	0.70	48	0.84
11	1.07×10^{-5}	30	0.75	49	0.00
12	0.94	31	0.97	50	0.61
13	0.00	32	0.95	51	0.72
14	0.88	33	0.54	52	0.00
15	0.44	34	0.00	53	0.51
16	0.63	35	0.61	54	0.00
17	0.65	36	0.00	55	0.80
18	0.67	37	0.78	56	0.70
19	0.39	38	0.00	57	0.84

Tabela 4.4: Erro obtido para as trajetórias dos marcadores que foram reconstituídas através algoritmo para a sentença 39.

- O erro calculado para os 12 marcadores primários (marcadores selecionados pelo algoritmo de fatoração QR com pivoteamento de colunas) é nulo, uma vez que $x_i = \bar{x}_i$, $y_i = \bar{y}_i$ e $z_i = \bar{z}_i$;
- De todos os erros calculados, os marcadores 39, 42, 48 e 57 são os que possuem os maiores erros e chegam a 1.05mm para o marcador 39, 0.93mm para o marcador 42, 0.84mm para o marcador 48 e 57. Se observarmos que esses marcadores (ver figura 4.1) se encontram na parte inferior do rosto, que é a região onde se têm os maiores deslocamento de marcadores, era natural que tivéssemos nessa região os maiores erros.

A Tabela 4.5 mostra o erro médio para cada uma das sentenças usadas para validar o modelo. As sentenças 39 e 31,37 são justamente as que apresentam o menor e o maior erro médio respectivamente. Comparado com o erro de precisão dos dados, cerca de 0.5mm (ver [14]), o erro para as trajetórias reconstruídas para os

marcadores faciais é pequeno, já que em média, o erro é de 0.64mm (ver Algoritmo I do Apêndice A).

Sentenças	Amostras	Erro Médio
31	468	1.10
32	323	0.84
33	424	1.06
34	416	0.93
35	474	1.08
36	454	0.68
37	315	0.71
38	308	0.97
39	589	0.64
40	364	1.01

Tabela 4.5: *Erro médio para trajetórias reconstituídas pelo algoritmo para as sentenças 31 a 40.*

Em [12], Lucero et al. obteve seu modelo adotando uma base de 09 marcadores. Neste capítulo, calculamos os erros, reconstruímos as trajetórias, obtivemos animações faciais através de uma base constituída de 12 marcadores com erro médio de 0.90mm e como já era esperado, menor que o erro médio que Lucero, de 1.05mm, pois em seu artigo, a base adotada contém 03 marcadores a menos.

4.9 Análise para Seleção de Marcadores

Esta Seção objetiva justificar a escolha dos 12 marcadores na Seção 4.6 para análise de erros, trajetórias e geração de animações faciais através de uma frase predefinida do conjunto de sentenças, como nas Seções 4.7 e 4.8.

Sabemos que as primeiras k colunas da matriz $A\Pi$ fornecem uma solução para o problema de seleção de subconjunto (ver Capítulo 3). No nosso caso, as colunas desta matriz são representadas por deslocamentos de 57 marcadores.

O modelo (matriz X) é obtido a partir da seleção de k marcadores na matriz de deslocamentos, resolvendo $R_{11}X = R_{12}$ (ver Capítulo 3). Em seguida, conseguimos os marcadores secundários resolvendo $P_1X = P_2$ (ver Seção 4.7). Para cada conjunto de k marcadores selecionados temos uma matriz X (modelo) e uma matriz P_2

(deslocamentos secundários). Fazendo k variar de 1 a 57 (número de marcadores) e escolhendo uma frase predefinida do conjunto de sentenças, logramos para cada subconjunto de k marcadores um erro médio. Sabemos a priori que o erro de precisão dos dados é de 0.5mm (ver [14]), então qualquer subconjunto de k marcadores em que o erro médio é aproximadamente do mesmo valor é apropriado para construir o modelo. As Figuras 4.12, 4.13 e 4.14, representam os gráficos para as frases 31 a 40 do conjunto de sentenças (a esquerda, frases ímpares e a direita as pares).

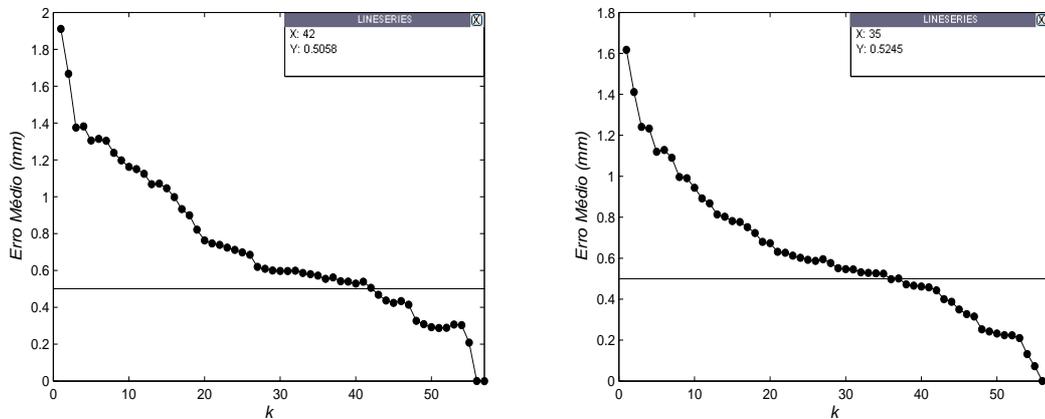


Figura 4.12: Erro médio para um conjunto de k marcadores selecionados, frases 31 e 32. A linha cheia (na horizontal) representa o erro de precisão dos dados.

Nas Figuras 4.12, 4.13 e 4.14, a linha horizontal representa aproximadamente o erro de precisão dos dados e observamos que as frases 31 e 35, são as que possuem os maiores subconjuntos de marcadores necessários, aproximadamente 40, para se obter um erro médio próximo de 0.5mm. Por outro lado, as frases 36 e 39, possuem os menores subconjunto de marcadores, em torno de 20, para conseguir erro médio similar.

Através de uma análise das Figuras 4.12, 4.13 e 4.14 concluímos que o erro médio diminui quando se aumenta o número de marcadores e pelo observado acima, uma quantidade razoável para se construir o modelo deveria ser de pelo menos 40. Por outro lado, não é interessante uma base muito numerosa, já que propomos desde o início deste trabalho um modelo em termos de ‘poucos’ marcadores faciais. Na verdade, não há procedimento que estime a priori o número de marcadores a serem escolhidos. Porém certamente, podemos afirmar que a base com 12 marcadores mostrou bons resultados, com erro médio relativamente pequeno e uma boa ani-

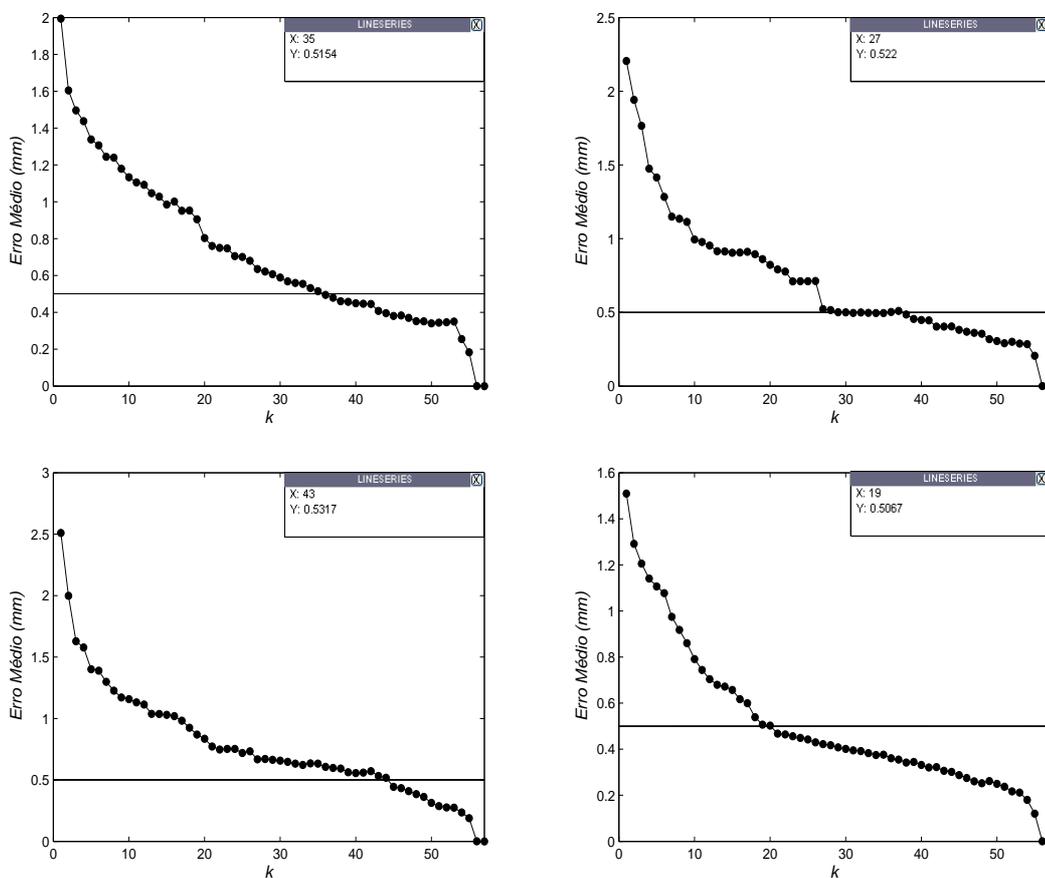


Figura 4.13: Erro médio para um conjunto de k marcadores selecionados, frases 33 a 36. A linha cheia (na horizontal) representa o erro de precisão dos dados.

mação facial sem distorções perceptíveis.

Os algoritmos utilizados para fazer as Figuras 4.12, 4.13 e 4.14 estão disponíveis em <http://vargas.mat.unb.br/>.

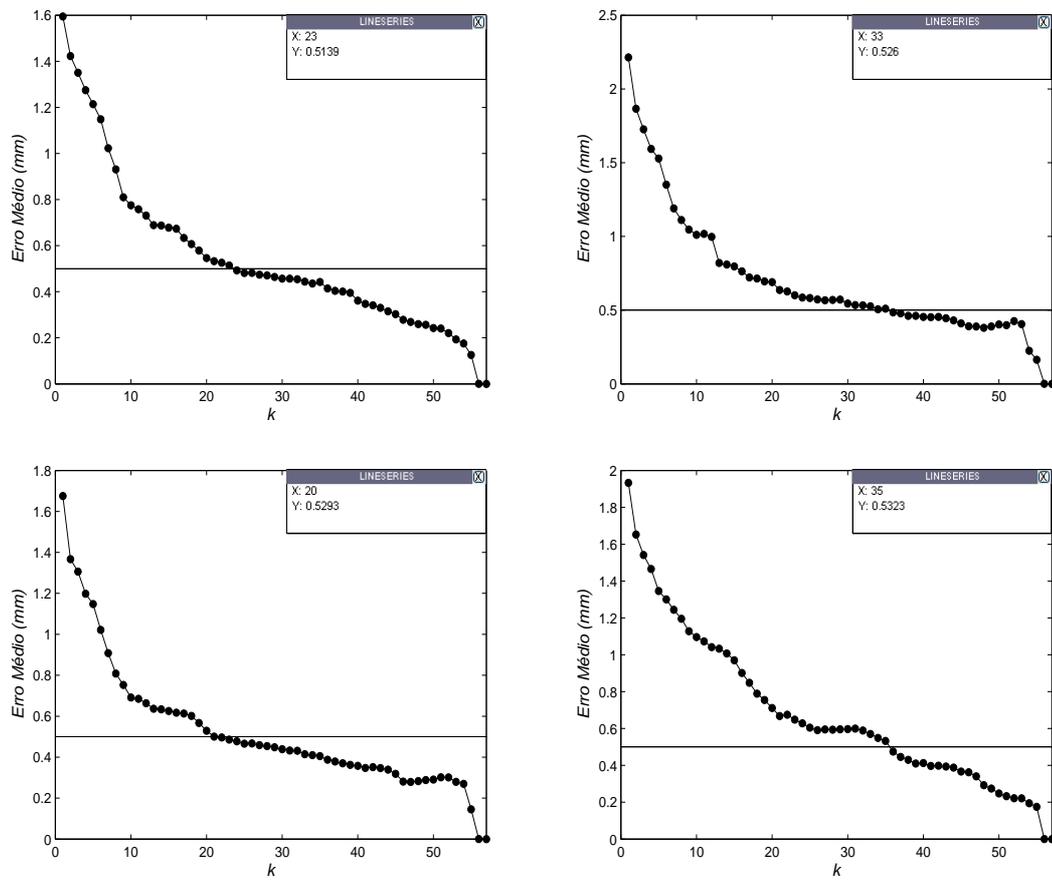


Figura 4.14: Erro médio para um conjunto de k marcadores selecionados, para as frases 37 a 40. A linha cheia (na horizontal) representa o erro de precisão dos dados.

Conclusões

Neste trabalho, estudamos a fatoração QR e suas aplicações. Através desta decomposição, encontramos uma solução para o sistema de equações lineares $Ax = b$, onde $A \in \mathbb{R}^{n \times n}$ e para o problema de mínimos quadrados $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$, com $A \in \mathbb{R}^{m \times n}$ e $m > n$, tal que A possui ou não posto completo. Em ambos os casos a solução é fornecida com a resolução de um sistema triangular superior. Tais sistemas são resolvidos por substituições retroativas, através de equações da forma

$$x_i = \frac{b_i - \sum_{j=i+1}^n r_{ij}x_j}{r_{ii}}$$

para todo $i = 1, \dots, n$.

A fatoração QR com pivoteamento de colunas mostrou-se eficaz, fornecendo uma solução para o problema de seleção de subconjunto, extraindo um subconjunto de colunas linearmente independentes de uma matriz A .

Entretanto, é claro que existem outros métodos para encontrar soluções para os problemas propostos nesta dissertação (ver [7]), porém, para os propósitos deste trabalho, fatoração QR mostrou bons resultados, permitindo resolver os principais problemas para se chegar em um modelo para geração de animações faciais.

Tal procedimento mostrou-se eficiente, pois permitiu identificar um subconjunto base de marcadores faciais independentes, posteriormente utilizados para prognosticar o movimento de pontos faciais arbitrários.

Através da Figura 4.11 podemos observar que as trajetórias reconstituídas pelo algoritmo estão bem próximas das originais, mostrando a eficácia da fatoração QR na reconstrução das trajetórias dos marcadores. As animações computadorizadas produzidas através da fatoração QR com pivoteamento de colunas não possuem nenhuma distorção perceptível nos padrões movimento faciais.

Em [12], Lucero et al. obteve seu modelo adotando uma base de 09 marcadores e um erro de aproximadamente 1.05mm. Neste trabalho, reconstruímos as trajetórias, obtivemos animações faciais através de uma base constituída de 12 marcadores com erro médio de 0.90mm e como já era esperado, houve uma diminuição do erro.

Salientamos que os marcadores faciais selecionados são utilizados na obtenção de um modelo linear individualizado da cinemática facial, pois cada indivíduo possui suas próprias propriedades da pele, tecidos, peso ou estruturas musculares diferentes.

Possivelmente, alguns aspectos desta técnica requerem algumas melhorias. Por exemplo, um melhor critério para determinar uma dimensão apropriada na seleção dos marcadores utilizados para construir o modelo poderiam inferir melhores resultados. Por exemplo, na geração de animações faciais. Mas estas e outras questões afins, deixamos como assunto para futuros trabalhos de pesquisas.

Referências Bibliográficas

- [1] Björck, A., *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia (1996).
- [2] Couvreur, C.; Breler, Y., *On the optimality of the backward greedy algorithm for the subset selection problem*, SIAM Journal on Matrix Analysis and Applications 21, 797 – 808 (2006).
- [3] Chan, T. F.; Hansen P. C., *Low-rank revealing QR factorizations*, *Numerical Linear Algebra with Applications* 1, 33 – 44 (1994).
- [4] Beautemps, D.; Badin, P.; Bailly, G., *Linear degrees of freedom in speech production: Analysis of cineradio-and labio-film data and articulatory-acoustic modeling*, *Journal Acoustical Society of America* 109, 2165 – 2180 (2001).
- [5] Fukuda, H. E., *Recuperação de Informações por Álgebra Linear Computacional*, Universidade de São Paulo - USP, Brasil (2004).
- [6] Golub, G.; Klema, V.; Stewart, G. W., *Rank degeneracy and least squares problems*, Computer Science Department Stanford University, STAN – CS – 76 – 559 (1976).
- [7] Golub, G. H.; Van Loan, C. F., *Matrix Computations*, The John Hopkins University Press, Baltimore (1996).
- [8] Hoog, F. R.; Mattheij, R. M. M., *Subset selection for matrices*, *Journal of the Linear Algebra and Applications* 422, 349 – 459 (2007).
- [9] Hong, Y. P.; Pan, C. T., *Rank - revealing QR factorizations and the singular value decomposition*, *Mathematics of Computation* 58, 213 – 232 (1992).

- [10] Júdice, J. J.; Patrício, J. M., *Sistemas de Equações Lineares*, Departamento de Matemática da Universidade de Coimbra, Portugal (**1996**).
- [11] Leon S. J., *Linear Algebra with Applications*, Prentice-Hall International Editions, New York (**1994**).
- [12] Lucero, J. C.; Baigorri, R. A.; Munhall K. G., *Data-driven facial animation of speech using a QR factorization algorithm*, Proceedings of the 7th International Seminar on Speech Production, pp. 135 – 142, Ubatuba - SP (**2006**).
- [13] Lucero, J. C.; Maciel, S. T. R.; Johns, D. A.; Munhall, K. G., *Empirical modeling of human face kinematics during speech using motion clustering*, Journal Acoustical Society of America 118, 405 – 409 (**2005**).
- [14] Munhall, K. G., informação privada via e-mail: <munhallk@post.queensu.ca>.
- [15] Alexa, M.; Müller, W., *Representing animations by principal components*, EUROGRAPHICS 2000 (Gross M.; Hopgood F. R. A., eds.), pp. 1 – 8, Blackwell Publishers, Malden (**2000**).
- [16] Maciel, S. T. R.; Silva, A. M.; Lucero, J. C., *Fatoração QR para o problema de seleção de subconjuntos aplicado a uma rede de monitoramento da variação da piezometria*, XXX Congresso Nacional de Matemática Aplicada Computacional - CNMAC. Anais do XXX Congresso Nacional de Matemática Aplicada e Computacional 1, 1 – 7 (**2007**).
- [17] Setnes, M.; Babuška, R., *Rule base reduction: some comments on the use of orthogonal transforms*, IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews 31, 199 – 206 (**2001**).
- [18] Kuratate, T.; Yehia, H.; Vatikiotis-Bateson, E., *Kinematics-based synthesis of realistic talking faces*, International Conference on Auditory - Visual Speech Processing (AVSP'98) (Burnham, D.; Robert-Ribes, J.; Vatikiotis-Bateson E., eds.), pp. 185 – 190, Causal Productions, Terrigal-Sydney (**1998**).
- [19] Trefethen, L. N.; Bau, D., *Numerical Linear Algebra*, SIAM, Philadelphia (**1997**).

- [20] Watkins, D. S., *Fundamentals of Matrix Computations*, John Wiley & Sons, New York (**1991**).

Apêndice A

Estão relacionados abaixo o código fonte dos programas implementados em Matlab para gerar os gráficos e tabelas que aparecem no texto. Alguns programas, Algoritmos J, K e L foram adaptados de [12] e os outros foram elaborados neste trabalho.

Algoritmo A

Código fonte para gerar a Figura 4.4.

```
-----  
  
clear all  
load Matriz  
MARCADORES = 57;  
[k,l] = size(trM5);  
c = 2;  
l = 0;  
n = 12;  
m = 0;  
while (k >= 57) & (c > 1)  
    for j = 1:k  
        if mod(j,100) == 0  
            D = trM5(1:j,:);  
            C = reshape(D,3*j,MARCADORES);  
            c = 0;  
            [q,r,e] = qr(C,0);
```

```
        a = norm(r, 'fro');
        for i = 1:n
            l = l + 1;
            R(l) = r(i,i);
            b(l) = R(l)/a;
        end
    end
end
for i = 1:(l/n)
    for j = 1:n
        m = m + 1;
        M(i,j) = b(m);
    end
end
plot(abs(M), '-o', 'MarkerSize', 4, 'MarkerFaceColor', 'black');
xlabel('Amostras \times 100')
ylabel('r_{kk}/||R||_F')
end
if (k < 57)
    display('k tem que ser maior ou igual a 57')
end
```

Algoritmo B

Código fonte para gerar a Figura 4.5.

```
clear all
load MatrizHexa
[M,N] = size(A);
P = randperm(M);
C = A(P,:);
k = 16106;
c = 2;
l = 0;
n = 10;
m = 0;
while (k >= 57) & (c > 1)
    for j = 1:k
        if mod(j,100) == 0
            H = C(1:j,:);
            W = reshape(H,6*j,N/6);
            c = 0;
            [q,r,e] = qr(W,0);
            a = norm(r,'fro');
            for i = 1:n
                l = l + 1;
                R(l) = r(i,i);
                b(l) = R(l)/a;
            end
        end
    end
end
for i = 1:(l/10)
    for j = 1:10
```

```
        m = m + 1;
        D(i,j) = b(m);
    end
end
plot(abs(D),'-o','MarkerSize',4,'MarkerFaceColor','black');
xlabel('Amostras \times 100')
ylabel('R_{kk}/||R||_F')
end
if (k < 57)
    display('n tem que ser maior ou igual a 57')
end
```

Algoritmo C

Código fonte para gerar a Figura 4.2.

```
clear all
load DataW
[m,n]= size(trM);
N = n/3;
trM3 = reshape(trM,3*m,N);
[U,T,V] = svd(trM3,0);
for i = 1:57
    R(i) = T(i,i);
    k(i) = i;
end
plot(k,R,'-ro','MarkerSize',5,'MarkerFaceColor','black');
xlabel('k')
ylabel('\Sigma_{kk}')
```

Algoritmo D

Código fonte para gerar a Figura 4.3.

```
clear all
load DataW
[m,n]= size(trM);
N = n/3;
trM3 = reshape(trM,3*m,N);
[q,r,e] = qr(trM3,0);
SIGMA = diag(r);
SIGMA = SIGMA';
for i = 1:57
    if SIGMA(i) < 0
        M(i,i) = - 1;
    else
        M(i,i) = 1;
    end
end
D = M*r;
for i = 1:57
    R(i) = D(i,i);
    k(i) = i;
end
plot(k,R,'-ro','MarkerSize',5,'MarkerFaceColor','black');
xlabel('k')
ylabel('r_{kk}')
```

Algoritmo E

Código fonte para gerar as Tabela 4.1.

```
clear all
load DataW
[m,n] = size(trM);
j = 0;
for i = 1:12
    j = j + 1000;
    if j < 11975
        D = trM(1:j,:);
        C = reshape(D,3*j,57);
        [q,r,e] = qr(C,0);
        M(:,i) = e(1:12);
    else
        j = j - 25;
        D = trM(1:j,:);
        C = reshape(D,3*j,57);
        [q,r,e] = qr(C,0);
        M(:,i) = e(1:12);
    end
end
end
```

Algoritmo F

Código fonte para gerar a Tabela 4.2.

```
clear all
load DataW
[m,n] = size(trM);
P = randperm(m);
E = trM(P,:);
j = 0;
for i = 1:12
    j = j + 1000;
    if j < 11975
        D = E(1:j,:);
        C = reshape(D,3*j,57);
        [q,r,e] = qr(C,0);
        M(:,i) = e(1:12);
    else
        j = j - 25;
        D = E(1:j,:);
        C = reshape(D,3*j,57);
        [q,r,e] = qr(C,0);
        M(:,i) = e(1:12);
    end
end
end
```

Algoritmo G

Código fonte para gerar a Tabela 4.3.

```
clear all
load DataW
[m,n] = size(trM);
C = reshape(B,3*m,57);
[q,r,e] = qr(C,0);
M = e(1:12);
```

Algoritmo H

Código fonte para gerar a Figura 4.11.

```
clear all
load DataW
load Resultados
load Posicao
IC = 11;
[m,n] = size(sntc_39);
N = n/3;
%Cálculo do Deslocamento
for i = 1:m
    for j = 1:n
        D(i,j) = sntc_39(i,j) - Pos9(1,j);
    end
end
[m,n] = size(D);
%Matriz Deslocamento dos Marcadores Principais
P3 = reshape(D,3*m,N);
for j = 1:IC
    P1(:,j) = P3(:,e(j));
end
% Deslocamento dos Marcadores Secundários
P2 = P1*AA2;
for i = 1:m
    for j = 1:N
        x1(i,j) = P3(i,j);
        y1(i,j) = P3(m + i,j);
        z1(i,j) = P3(2*m + i,j);
    end
end
```

```
        x2(i,j) = P2(i,j);
        y2(i,j) = P2(m + i,j);
        z2(i,j) = P2(2*m + i,j);
    end
end
%Comparação dos Marcadores Originais com secundários
for i = 28:28
    X = [x1(:,i),x2(:,i)];
    Y = [y1(:,i),y2(:,i)];
    Z = [z1(:,i),z2(:,i)];
    plot(X(:,1),'k--');
    hold on
    plot(X(:,2),'k-');
    xlabel('Amostras','FontSize',12);
    ylabel('x (mm)','FontSize',12);
    %legend('Marcador primário x_i', 'Marcador secundário x_i');
    pause
    hold off
    clear X
    clf
    plot(Y(:,1),'K--');
    hold on
    plot(Y(:,2),'K-');
    xlabel('Amostras','FontSize',12);
    ylabel('y (mm)','FontSize',12);
    %legend('Marcador primário y_i', 'Marcador secundário y_i');
    pause
    hold off
    clear Y
    clf
    plot(Z(:,1),'k--');
    hold on
    plot(Z(:,2),'k-');
    xlabel('Amostras','FontSize',12);
```

```
ylabel('z (mm)', 'FontSize', 12);  
%legend('Marcador primário z_i', 'Marcador secundário z_i');  
pause  
hold off  
clear Z  
clf  
end
```

Algoritmo I

Código fonte para gerar as Tabelas 4.4 e 4.5.

```
clear all
load DataW
load Resultados
load Posicao
IC = 10;
[m,n] = size(sntc_39);
N = n/3;
T = m*3;
k = T*N;
M = 0;
S = 0;
%Cálculo do Deslocamento
for i = 1:m
    for j = 1:n
        D(i,j) = sntc_39(i,j) - Pos9(1,j);
    end
end
%Matriz Deslocamento dos Marcadores Principais
P3 = reshape(D,3*m,N);
for j = 1:IC
    P1(:,j) = P3(:,e(j));
end
% Deslocamento dos Marcadores Secundários
P2 = P1*AA2;
%Erro de cada Marcador Primário e Secundário Respectivamente
for j = 1:N
    for i = 1:m
```

```
M = M + (P3(i,j) - P2(i,j)).^2 + (P3(m + i,j) -  
P2(m + i,j)).^2 + (P3(2*m + i,j) - P2(2*m + i,j)).^2 ;  
end  
erro(j) = sqrt((1/m)*M);  
M = 0;  
l(j) = j;  
end  
erro_medio = sum(erro)/46;
```

Algoritmo J

Código fonte para gerar a Figuras 4.6 e 4.7.

```
function QRanalysis(dfile)
if nargin<1
    [filename, pathname] = uigetfile( ...
        { '*.mat','MAT files (*.mat)'; ...
          '*.*', 'All Files (*.*)'}, ...
        'Select data file', ...
        'MultiSelect', 'off');
    dfile=fullfile(pathname,filename);
end
load(dfile);
load DataW
[m,n] = size(trM);
IC = 10;
N = n/3;
C = reshape(trM,3*m,N);
% QR
[q,r,e] = qr(C,0);
% Solve least squares
AA = [eye(IC),r(1:IC,1:IC)\r(1:IC,IC+1:N)];
% Put columns back into original positions
[a,b] = sort(e);
AA2 = AA(:,b);
% Plot facial regions
load cmap
colormap(cmap);
LGRID = 100;
% Grid dimension
```

```

for i = 1:N;
    x(i) = mPos(1,1+(i-1)*3);
    y(i) = mPos(1,2+(i-1)*3);
    z(i) = mPos(1,3+(i-1)*3);
end
xlin = linspace(min(x),max(x),LGRID);
ylin = linspace(min(y),max(y),LGRID);
[X,Y] = meshgrid(xlin,ylin);
Z = griddata(x,y,z,X,Y,'cubic');
ZPos(1,:) = reshape(X,1,LGRID*LGRID);
ZPos(2,:) = reshape(Y,1,LGRID*LGRID);
ZPos(3,:) = reshape(Z,1,LGRID*LGRID);
ZPos = reshape(ZPos,1,3*LGRID*LGRID);
for iC = 1:IC
    w = AA2(iC,:);
    W = griddata(x,y,w,X,Y,'cubic');
    %ZW(iC,:) = reshape(W,1,LGRID*LGRID);
    C = W + ones(size(W));
    surf(X,Y,W,C,'FaceColor','interp','EdgeColor','none')
    caxis([0 2]);
    daspect([5 5 1])
    view(0,90)
    axis tight
    axis equal
    xlabel('X (mm)','FontSize',12);
    ylabel('Y (mm)','FontSize',12);
    title(['Região n^{o}:',int2str(iC),'-Marcador',int2str(e(iC))])
    hold on
    plot3(x,y,z,'ro','MarkerSize',4,'MarkerFaceColor','black');
    pause
    hold off
    %print('-deps',int2str(iC));
end
% Save results

```

```
[filename2, pathname2] = uiputfile( ...  
    {'*.mat', 'MAT-files (*.mat)'}; ...  
    '*.*', 'All Files (*.*)'}, ...  
    'Save as', 'Resultados.mat');  
save(fullfile(pathname2,filename2),'e','AA2','IC','mPos');
```

Algoritmo K

Código fonte para gerar as Figuras 4.8 e 4.9.

```
clear all
load DataW
% Reshape trajectory matrix
[m,n] = size(trM);
trM3 = reshape(trM,3*m,n/6);
%QR
[q,r,e] = qr(trM3,0);
% Plot facial regions
load cmap
colormap(cmap);
b = max(abs(r(:,1)));
RR = r/b;
[a,b] = sort(e);
RR2 = RR(:,b);
LGRID = 100;
% Grid dimension
for i = 1:n/6;
    x(i) = mPos(1,1+(i-1)*3);
    y(i) = mPos(1,2+(i-1)*3);
    z(i) = mPos(1,3+(i-1)*3);
end
xlin = linspace(min(x),max(x),LGRID);
ylin = linspace(min(y),max(y),LGRID);
[X,Y] = meshgrid(xlin,ylin);
Z = griddata(x,y,z,X,Y,'cubic');
ZPos(1,:) = reshape(X,1,LGRID*LGRID);
ZPos(2,:) = reshape(Y,1,LGRID*LGRID);
```

```
ZPos(3,:) = reshape(Z,1,LGRID*LGRID);
ZPos = reshape(ZPos,1,3*LGRID*LGRID);
for i = 1:n/6
    w = RR2(i,:);
    W = griddata(x,y,w,X,Y,'cubic');
    C = W + ones(size(W));
    surf(X,Y,W,C,'FaceColor','interp','EdgeColor','none')
    caxis([0 2]);
    daspect([5 5 1]);
    view(0,90)
    axis tight
    xlabel('x (mm)','FontSize',12);
    ylabel('y (mm)','FontSize',12);
    title(['Região n^{o}:',int2str(i),' - Linha ',int2str(i)])
    hold on
    plot3(x,y,z,'ko','MarkerSize',4,'MarkerFaceColor','black');
    pause
    hold off
    %print('-deps',int2str(i));
end
```

Algoritmo L

Código fonte para gerar animações faciais utilizadas na seção 4.7.

```
clear all
NMARKERS = 57;
IC = 10;
load Resultados
load DataW
iFile = 39;
sntc_iFile = eval(['sntc_',int2str(iFile)]);
sentti = CID_31to40{iFile-30};
LGRID = 30;
ccm = [0.9 0.9 0.7];
for i = 1:NMARKERS;
    x(i) = mPos(1,1+(i-1)*3);
    y(i) = mPos(1,2+(i-1)*3);
    z(i) = mPos(1,3+(i-1)*3);
end
xlin = linspace(min(x),max(x),LGRID);
ylin = linspace(min(y),max(y),LGRID);
[X,Y] = meshgrid(xlin,ylin);
Z = griddata(x,y,z,X,Y,'cubic');
ZPos(1,:) = reshape(X,1,LGRID*LGRID);
ZPos(2,:) = reshape(Y,1,LGRID*LGRID);
ZPos(3,:) = reshape(Z,1,LGRID*LGRID);
ZPos=reshape(ZPos,1,3*LGRID*LGRID);
for iC = 1:IC
    w = AA2(iC,:);
    W = griddata(x,y,w,X,Y,'cubic');
    ZW(iC,:) = reshape(W,1,LGRID*LGRID);
```

```
end
nS = length(sntc_iFile);
trM = sntc_iFile-ones(nS,1)*mPos;
trM = round(10*trM)/10;
% Reshape trajectory matrix
[m,n] = size(trM);
trM3 = reshape(trM,3*m,NMARKERS);
bb1 = trM3(:,e(1:IC))*ZW;
bb1 = reshape(bb1,m,3*LGRID*LGRID);
pos2 = reshape(sntc_iFile,3*m,NMARKERS);
position2 = pos2(:,e(1:IC));
position2 = reshape(position2,m,3*IC);
clear pos2 trM trM3 Data AA2 X Y Z x y z w W
clf
colordef black;
h1 = figure(1);
pf = get(h1,'Position');
pf = [10 200 480 332];
set(h1,'Position',pf);
a1=axes;
aviobj = avifile(['QR_',int2str(iFile),'_',int2str(IC),'_huge.avi'],
'FPS',120,'COMPRESSION','none');
[m,n] = size(bb1);
position = bb1+ones(m,1)*ZPos;
clear bb1
initF=1;
for iT = initF:m
    disp(['Now processing frame ',int2str(iT),'/',int2str(m)]);
    for iM = 1:n/3
        x0(iM) = position(iT,1+(iM-1)*3);
        y0(iM) = position(iT,2+(iM-1)*3);
        z0(iM) = position(iT,3+(iM-1)*3);
    end
end
zz0=reshape(z0,30,30);
```

```
for iM = 1:IC
    x2(iM) = position2(iT,1+(iM-1)*3);
    y2(iM) = position2(iT,2+(iM-1)*3);
    z2(iM) = position2(iT,3+(iM-1)*3);
end
if iT == initF
    p1 = subplot(1,2,1);
    set(p1,'FontSize',10,'LineWidth',1);
    p11 = plot(x0,y0,'o','MarkerSize',1,...
        'Color',ccm,'MarkerFaceColor',ccm);
    hold on;p11b=plot(x2,y2,'o','MarkerSize',4,...
        'Color','red','MarkerFaceColor','red');hold off;
    axis equal
    xlabel('X (mm)','FontSize',10);
    ylabel('Y (mm)','FontSize',10);
    axis([-80 90 -190 -20]);
    set(p1,'Position', [0.124 -0.007 0.427 0.975]);
    set(p1,'NextPlot','add','DrawMode','fast')
    p2 = subplot(1,2,2);
    set(p2,'FontSize',10,'LineWidth',1);
    p12 = plot(z0,y0,'o','MarkerSize',1,...
        'Color',ccm,'MarkerFaceColor',ccm);
    hold on; p12b=plot(z2,y2,'o','MarkerSize',4,...
        'Color','red','MarkerFaceColor','red');hold off;
    axis equal
    xlabel('Z (mm)','FontSize',10);
    axis([-80 60 -190 -20]);
    set(p2,'YTickLabel',[]);
    set(p2,'Position',[0.604 0.068 0.352 0.826]);
    set(p2,'NextPlot','add','DrawMode','fast');
    t1=title(sentti);
    set(t1,'Position',[-124.194 -2.097 1101.181],'FontSize',10);
    else
        set(p11,'XData',x0,'yData',y0);
```

```
        set(p12,'XData',z0,'yData',y0);
        set(p11b,'XData',x2,'yData',y2);
        set(p12b,'XData',z2,'yData',y2);
    end
    drawnow
    aviobj = addframe(aviobj,h1);
end
clear x0 y0 z0
aviobj = close(aviobj);
close(h1);
```

Apêndice B

Tabela contendo o conjunto de 40 frases no inglês usadas para construir o modelo.

01	"It's time to go."
02	"If you don't want these old magazines, throw them out."
03	"Do you want to wash up?"
04	"It's a real dark night so watch your driving."
05	"I'll carry the package for you."
06	"Did you forget to shut off the water?"
07	"Fishing in a mountain stream is my idea of a good time."
08	"Fathers spend more time with their children than they used to."
09	"Be careful not to break your glasses!"
10	"I'm sorry."
11	"You can catch the bus across the street."
12	"Call her on the phone and tell her the news."
13	"I'll catch up with you later."
14	"I'll think it over."
15	"I don't want to go to the movies tonight."
16	"If your tooth hurts that much you ought to see a dentist."
17	"Put that cookie back in the box!"
18	"Stop fooling around! "
19	"Time's up."
20	" How do you spell your name?"
21	"Music always cheers me up."
22	"My brother's in town for a short while on business."
23	"We live a few miles from the main road."
24	"This suit needs to go to the cleaners."
25	"They ate enough green apples to make them sick for a week."
26	"Where have you been all this time?"
27	"Have you been working hard lately? "
28	" There's not enough room in the kitchen for a new table."
29	"Where is he?"

30	“Look out!”
31	“I’ll see you right after lunch.”
32	“See you later.”
33	“White shoes are awful to keep clean.”
34	“Stand there and don’t move until I tell you!”
35	“There’s a big piece of cake left over from dinner.”
36	“Wait for me at the corner in front of the drugstore.”
37	“It’s no trouble at all.”
38	“Hurry up!”
39	“The morning paper didn’t say anything about rain this afternoon or tonight.”
40	“The phone call’s for you.”

Tabela 4.6: *Conjunto de 40 sentenças em inglês: As 30 primeiras frases foram usadas para construir o modelo e as 10 últimas para realizar testes e validar o modelo.*

Anexos

Encontram-se abaixo as demonstrações dos Teoremas 2.6, 2.7, 2.9, 2.10 e 2.13 que aparecem no Capítulo 2.

Teorema 2.6

Teorema 2.6 *Seja $P \in \mathbb{R}^{m \times n}$ uma reflexão de Householder. Então*

1. P é simétrica;
2. P é ortogonal;
3. $\|Px\|_2 = \|x\|_2$;
4. $\|PA\|_2 = \|A\|_2$;
5. $P^2 = I$.

onde $A \in \mathbb{R}^{m \times n}$.

Demonstração de 1.:

$$\begin{aligned} P^T &= \left(I - 2\frac{vv^T}{v^T v}\right)^T \\ &= I^T - \left(2\frac{vv^T}{v^T v}\right)^T \\ &= I - 2\frac{vv^T}{v^T v} \\ &= P \end{aligned}$$

Portanto, P é simétrica. ■

Demonstração de 2.:

Pelo ítem 1. temos que P é simétrica, então $P = P^T$, assim

$$\begin{aligned} PP^T &= P^T P \\ &= \left(I - 2\frac{vv^T}{v^T v}\right)^2 \\ &= I - 4\frac{vv^T}{v^T v} + 4\frac{v(v^T v)v^T}{(v^T v)(v^T v)} \end{aligned}$$

Como $\|v\|^2 = v^T v$, então

$$\begin{aligned} I - 4\frac{vv^T}{v^T v} + 4\frac{v(v^T v)v^T}{(v^T v)(v^T v)} &= I - 4\frac{vv^T}{v^T v} + 4\frac{v(\|v\|_2^2)v^T}{\|v\|_2^4} \\ &= I - 4\frac{vv^T}{v^T v} + 4\frac{vv^T}{\|v\|_2^2} \\ &= I - 4\frac{vv^T}{v^T v} + 4\frac{vv^T}{v^T v} \\ &= I. \end{aligned}$$

Portanto, P é ortogonal. ■

Demonstração de 3.:

Usando 1. e 2. temos,

$$\begin{aligned} \|Px\|_2^2 &= \|P^T x\|_2^2 \\ &= (P^T x)^T (P^T x) \\ &= (x^T P)(P^T x) \\ &= x^T x \\ &= \|x\|_2^2 \end{aligned}$$

portanto, $\|Px\|_2 = \|x\|_2$. ■

Demonstração de 4.:

Segue imediatamente do Teorema 1.10.

**Demonstração de 5.:**

Como $P = P^T$ e multiplicando por P em ambos os membros, teremos:

$$\begin{aligned} P^2 &= P^T P \\ &= I \end{aligned}$$



Teorema 2.7

Teorema 2.7 *Dado um vetor não-nulo $x = (x_1, x_2, \dots, x_n)^T$, podemos encontrar um vetor $v = (v_1, v_2, \dots, v_n)^T$ de maneira que Px é múltiplo de e_1 onde a matriz de householder P é definida pelo vetor $v = x \pm \|x\|_2 e_1$ e $Px = \pm \|x\|_2 e_1$.*

Demonstração:

Notemos que:

$$\begin{aligned} Px &= \left(I - 2 \frac{vv^T}{v^T v} \right) x \\ &= x - 2 \frac{vv^T x}{v^T v} \end{aligned}$$

onde $Px \in \langle e_1 \rangle$. Este último implica que $v \in \langle x, e_1 \rangle$, com isso, v pode ser escrito como $v = x + \alpha e_1$. Assim,

$$\begin{aligned} v^T x &= (x + \alpha e_1)^T x \\ &= x^T x + \alpha \beta \end{aligned}$$

e

$$\begin{aligned} v^T v &= (x + \alpha e_1)^T (x + \alpha e_1) \\ &= x^T x + 2\alpha\beta + \alpha^2. \end{aligned}$$

Portanto

$$\begin{aligned} Px &= x - 2 \frac{x^T x + \alpha\beta}{x^T x + 2\alpha\beta + \alpha^2} (x + \alpha e_1) \\ &= x - 2 \frac{x^T x + \alpha\beta}{x^T x + 2\alpha\beta + \alpha^2} x - 2 \frac{x^T x + \alpha\beta}{x^T x + 2\alpha\beta + \alpha^2} \alpha e_1 \\ &= \left(1 - 2 \frac{x^T x + \alpha\beta}{x^T x + 2\alpha\beta + \alpha^2} \right) x - \left(2 \frac{x^T x + \alpha\beta}{x^T x + 2\alpha\beta + \alpha^2} \right) \alpha e_1. \end{aligned} \quad (4.8)$$

Como $Px \in \langle e_1 \rangle$, seque que o componente em x deve ser nulo, logo

$$\left(1 - 2 \frac{x^T x + \alpha\beta}{x^T x + 2\alpha\beta + \alpha^2} \right) = 0,$$

mas isso só se verifica se, e somente se,

$$x^T x + 2\alpha\beta + \alpha^2 - 2x^T x - 2\alpha\beta = 0$$

e daí, $\Rightarrow \alpha = \pm\|x\|_2$ e portanto,

$$v = x \mp \|x\|_2 e_1. \quad (4.9)$$

Sem perda de generalidade, podemos tomar $\alpha = \|x\|_2$ e substituindo em (4.8) obtemos:

$$\begin{aligned} Px &= - \left(2 \frac{x^T x + \|x\|_2 \beta}{x^T x + 2\|x\|_2 \beta + \|x\|_2^2} \right) \|x\|_2 e_1 \\ &= - \left(2 \frac{(\|x\|_2)^2 + \|x\|_2 \beta}{\|x\|_2^2 + 2\|x\|_2 \beta + \|x\|_2^2} \right) \|x\|_2 e_1 \\ &= - \left(2 \frac{\|x\|_2^2 + \|x\|_2 \beta}{2\|x\|_2^2 + 2\|x\|_2 \beta} \right) \|x\|_2 e_1 \\ &= - \left(2 \frac{\|x\|_2^2 + \|x\|_2 \beta}{2(\|x\|_2^2 + \|x\|_2 \beta)} \right) \|x\|_2 e_1 \\ &= -\|x\|_2 e_1. \end{aligned} \quad (4.10)$$

Analogamente, se $\alpha = -\|x\|_2$ então, $Px = \|x\|_2 e_1$.

■

Teorema 2.9

Teorema 2.9 *Seja $A \in \mathbb{R}^{m \times n}$ e sua respectiva fatoração QR . Suponha $\text{posto}(A) = n$ e considere as seguintes partições $A = [a_1, a_2, \dots, a_n]$ e $Q = [q_1, q_2, \dots, q_m]$, onde cada a_i e q_j são as respectivas colunas de A e Q , $1 \leq i \leq n$ e $1 \leq j \leq m$. Então*

$$\langle a_1, a_2, \dots, a_n \rangle = \langle q_1, q_2, \dots, q_n \rangle, \quad k = 1, \dots, n. \quad (4.11)$$

Demonstração:

Comparando a k -ésima coluna de $A = QR$, concluimos que

$$a_k = \sum_{i=1}^k r_{ik} q_i \in \langle q_1, \dots, q_k \rangle. \quad (4.12)$$

Assim, $\langle a_1, \dots, a_n \rangle \subseteq \langle q_1, \dots, q_k \rangle$. Como $\text{posto}(A) = n$ segue que a matriz R é não singular, portanto possui inversa, logo $AR^{-1} = Q$ e daí,

$$q_k = \sum_{i=1}^k \bar{r}_{ik} a_i, \quad k = 1, \dots, n, \quad (4.13)$$

logo, $\langle a_1, \dots, a_n \rangle \supseteq \langle q_1, \dots, q_k \rangle$. Daí, $\langle a_1, a_2, \dots, a_n \rangle = \langle q_1, q_2, \dots, q_n \rangle$ onde $k = 1, \dots, n$.

■

Teorema 2.10

Teorema 2.10 *Seja $A \in \mathbb{R}^{m \times n}$, com $m \geq n$ e suponha que $\text{posto}(A) = n$. Então existe uma única matriz $Q \in \mathbb{R}^{m \times n}$ e $R \in \mathbb{R}^{n \times n}$ tal que Q têm colunas ortogonais e R é triangular superior com todas as entradas da diagonal positiva e*

$$A = QR. \quad (4.14)$$

Demonstração:

Vimos na Seção 2.2 que existem matrizes $\widehat{Q} \in \mathbb{R}^{m \times n}$ e $\widehat{R} \in \mathbb{R}^{n \times n}$, \widehat{Q} com colunas ortogonais e \widehat{R} triangular superior (não necessariamente com todas as entradas da diagonal positiva) tal que

$$A = \widehat{Q}\widehat{R}.$$

Como $\text{posto}(A) = n$ então a matriz \widehat{R} é não singular, ou seja, $\det(\widehat{R}) \neq 0$. Seja D uma matriz diagonal com elementos d_{ii} definida por

$$d_{ii} = \begin{cases} 1, & \text{se } \widehat{r}_{ii} > 0 \\ -1, & \text{se } \widehat{r}_{ii} < 0. \end{cases} \quad (4.15)$$

Então $D = D^T = D^{-1}$ e portanto, $DD^T = I$ logo, D é ortogonal. Segue que $Q = \widehat{Q}D^{-1}$ possui todas as colunas ortogonais e $R = D\widehat{R}$ é triangular superior com $r_{ii} = d_{ii}\widehat{r}_{ii} > 0$ e $A = QR$. Isso prova a existência.

Para provar a unicidade usaremos *Decomposição de Cholesky* (ver [7]). Esta decomposição consiste no seguinte: Dada uma matriz $A \in \mathbb{R}^{n \times n}$ positiva e definida, então existe uma única matriz triangular inferior $G \in \mathbb{R}^{n \times n}$ com entradas da diagonal todas positivas tal que $A = GG^T$.

Suponha que existissem matrizes Q_1, Q_2, R_1 e R_2 tais que $A = Q_1R_1 = Q_2R_2$ onde Q_i e R_i são matrizes com colunas ortogonais e triangular superior respectivamente e $i = 1, 2$.

$$AA^T = (Q_1R_1)^T(Q_1R_1) = R_1^TQ_1^TQ_1R_1 = R_1^TR_1$$

$$AA^T = (Q_2R_2)^T(Q_2R_2) = R_2^T Q_2^T Q_2 R_2 = R_2^T R_2$$

daí, $R_1^T R_1 = R_2^T R_2$. Pela unicidade da decomposição de Cholesky, devemos ter $R_1^T = R_2^T$ e portanto, $R_1 = R_2$, logo $Q_1 = Q_2$. Isso prova a unicidade. ■

Teorema 2.13

Teorema 2.13 *Seja $A \in \mathbb{R}^{m \times n}$ com $\text{posto}(A) = r \leq n$. Então existem matrizes \widehat{A} , Q e R , tal que $\widehat{A} = A\Pi$ é obtida de A por permutação de colunas, $Q \in \mathbb{R}^{m \times m}$*

é ortogonal, $R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times n}$, $R_{11} \in \mathbb{R}^{r \times r}$ é não-singular e triangular superior.

Demonstração:

Suponha que $r = n$, então a matriz A tem posto completo e portanto todas as entradas da diagonal da matriz R são todas não-nulas. Portanto, o resultado é uma decomposição $\widehat{A} = QR$, onde \widehat{A} é a matriz obtida através de A por permutação de

suas colunas. $R = \begin{pmatrix} \widehat{R} \\ 0 \end{pmatrix}$, onde \widehat{R} é uma matriz triangular superior e não-singular.

Suponha agora que $r < n$, portanto a matriz A possui posto incompleto. Então, pelo o que vimos anteriormente, em algum passo durante a decomposição QR com pivoteamento de colunas, temos que $r_{ii} = 0$, $r < i \leq n$. Observe que isso só ocorre se, e somente se, todas as entradas da submatriz restante é nula.

Suponha que isso ocorra depois de r passos computados. Sejam $P_i \in \mathbb{R}^{m \times m}$ denotando a reflexão obtida no passo i , então temos $P_r P_{r-1} \cdots P_1 \widehat{A} = \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix} = R$, onde $R_{11} \in \mathbb{R}^{r \times r}$ é triangular superior não singular com as entradas da diagonal $r_{11}, r_{22}, \dots, r_{rr}$ todas não-nulas e $R_{12} \in \mathbb{R}^{r \times (n-r)}$.

Seja $Q = P_1 P_2 \cdots P_r$, daí $Q^T = P_r P_{r-1} \cdots P_1$ e portanto $Q^T \widehat{A} = R$ donde temos que $\widehat{A} = QR$. Assim, $\text{posto}(A) = \text{posto}(\widehat{A}) = \text{posto}(R) = r$ onde $Q \in \mathbb{R}^{m \times m}$ e $R \in \mathbb{R}^{m \times n}$.

■