

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

JORGE PEIXOTO DE MORAIS NETO

Aceleração de uma variação do problema k-nearest neighbors

Goiânia
2014

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR AS TESES E DISSERTAÇÕES ELETRÔNICAS (TEDE) NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: ☒ [X] Dissertação ☐ [] Tese

2. Identificação da Tese ou Dissertação

Autor (a):	Jorge Peixoto de Moraes Neto		
E-mail:	jorgepeixotomoraes@gmail.com		
Seu e-mail pode ser disponibilizado na página?	<input checked="" type="checkbox"/> [X] Sim <input type="checkbox"/> [] Não		
Vínculo empregatício do autor	Nenhum (bolsista REUNI)		
Agência de fomento:	CAPES	Sigla:	CAPES
País:	Brasil	UF:	DF
		CNPJ:	00889834/0001-08
Título:	Aceleração de uma variação do problema k-nearest neighbors		
Palavras-chave:	aceleração, análise de dados multidimensionais, k-nearest neighbors, k vizinhos mais próximos, matriz circulante, processamento de imagem, programação paralela, transformada rápida de Fourier		
Título em outra língua:	Acceleration of a variation of the k-nearest neighbors problem		
Palavras-chave em outra língua:	acceleration, k-nearest neighbors, circulant matrix, image processing, fast Fourier transform, multidimensional data analysis, parallel programming		
Área de concentração:	Ciência da Computação		
Data defesa: (dd/mm/aaaa)	29/1/2014		
Programa de Pós-Graduação:	Ciência da Computação		
Orientador (a):	Prof. Dr. Wellington Santos Martins		
E-mail:	wellington@inf.ufg.br		
Co-orientador (a):*	Prof.Dr. Humberto José Longo, Prof. Dr. Leslie Richard Foulds		
E-mail:	longo@inf.ufg.br, lesfoulds@gmail.com		

*Necessita do CPF quando não constar no SisPG

3. Informações de acesso ao documento:

Concorda com a liberação total do documento ☒ [X] SIM ☐ [] NÃO¹

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF ou DOC da tese ou dissertação.

O sistema da Biblioteca Digital de Teses e Dissertações garante aos autores, que os arquivos contendo eletronicamente as teses e ou dissertações, antes de sua disponibilização, receberão procedimentos de segurança, criptografia (para não permitir cópia e extração de conteúdo, permitindo apenas impressão fraca) usando o padrão do Acrobat.

Jorge Peixoto de Moraes Neto
Assinatura do (a) autor (a)

Data: 28 / 3 / 2014

¹ Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

JORGE PEIXOTO DE MORAIS NETO

Aceleração de uma variação do problema k-nearest neighbors

Dissertação apresentada ao Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Computação.

Área de concentração: Ciência da Computação.

Orientador: Prof. Wellington Santos Martins

Co-Orientador: Prof. Humberto José Longo

Co-Orientador: Prof. Leslie Richard Foulds

Goiânia
2014

Dados Internacionais de Catalogação na Publicação (CIP)

Morais Neto, Jorge Peixoto de.

M791a Aceleração de uma variação do problema K-nearest neighbors [manuscrito] / Jorge Peixoto de Moraes Neto. – 2014. 97 f.; il., enc.

Orientador: Prof. Dr. Welington Santos Martins;
Co-orientador: Prof. Dr. Humberto José Longo; Co-orientador:
Prof. Dr. Leslie Richard Foulds.

Dissertação (Mestrado) – Universidade Federal de Goiás,
Instituto de Informática, 2014.

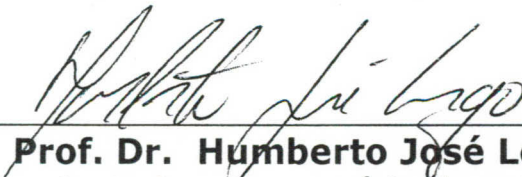
1. Transformada rápida de Fourier. 2. KNN—. I. Universidade
Federal de Goiás. II. Título

CDU: 004.421.2:517.443

Jorge Peixoto de Moraes Neto

Aceleração de uma variação do problema *K-Nearest Neighbors*

Dissertação defendida no Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Ciência da Computação, aprovada em 29 de Janeiro de 2014, pela Banca Examinadora constituída pelos professores:



Prof. Dr. Humberto José Longo
Co-orientador no exercício da orientação
Instituto de Informática – UFG
Presidente da Banca



Profa. Dra. Rosiane de Freitas Rodrigues
Instituto de Computação – UFAM



Prof. Dr. EdCarlos Domingos da Silva
Instituto de Matemática e Estatística – UFG

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador(a).

Jorge Peixoto de Moraes Neto

Graduou-se em Engenharia Eletrônica no Instituto Tecnológico de Aeronáutica. Em seu trabalho de graduação, acelerou (programando em assembly) a compactação de vídeo no formato MPEG-4 parte 2 para um processador Blackfin. Durante o mestrado foi monitor REUNI da disciplina Arquitetura de Computadores.

Dedico o trabalho a Deus.

Agradecimentos

Agradeço a Deus. Agradeço também à minha família, em particular minha mãe Maria Maia de Moraes, meu pai Orlando Peixoto de Moraes, minhas irmãs e minha namorada, Wanessa Simão Barbosa, por apoio e paciência. Agradeço também ao meu orientador, prof. Dr. Wellington Santos Martins, por fornecer artigos, revisão, orientação, aulas e apoio. Agradeço a sua paciência. Agradeço por ele ter encontrado o uso de FFT para acelerar a multiplicação de matriz circulante por vetor. Agradeço aos meus co-orientadores, professores Dr. Humberto José Longo e Dr. Leslie Richard Foulds, por orientação, apoio e paciência. Agradeço à coordenação do programa de pós-graduação em informática pela extensão de prazo. Agradeço a Ana Cristina Gomes de Jesus, pela disposição em me ajudar e por indicar o professor Edcarlos. Agradeço ao prof. Dr. Edcarlos Domingos da Silva, que me deu encorajamento e orientação e, em particular, chamou atenção para a forma canônica de Jordan. Agradeço a Humberto Silva Naves, que provou a desigualdade triangular para a distância especial. Agradeço ao Ministério da Educação pelo financiamento do trabalho através da bolsa REUNI.

And why do we fall, Bruce? So we can learn to pick ourselves up.

Thomas Wayne,
Batman Begins (2005).

Resumo

de Moraes Neto, Jorge Peixoto. **Aceleração de uma variação do problema k-nearest neighbors**. Goiânia, 2014. 98p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

Seja M um espaço métrico e P um subconjunto de M . O conhecido problema *k vizinhos mais próximos* (*k-nearest neighbors*, *KNN*) consiste em encontrar, dado $q \in M$, os k elementos de P mais próximos de q conforme a métrica de M . Abordamos uma variação do problema *KNN* para uma classe particular de espaços pseudo-métricos, descrita a seguir. Seja $m \in \mathbb{N}$ um natural e seja d a distância euclidiana em \mathbb{R}^m . Dado um vetor $p \in \mathbb{R}^m$:

$$p := (p_1, \dots, p_m)$$

seja $C^*(p)$ o conjunto das m rotações das coordenadas de p :

$$C^*(p) := \{(p_1, \dots, p_m), (p_2, \dots, p_m, p_1), \dots, (p_m, p_1, \dots, p_{m-1})\}$$

definimos a *distância especial* d_e como:

$$d_e(p, q) := \min_{p' \in C^*(p)} d(p', q).$$

d_e é uma pseudo-métrica, e (\mathbb{R}^m, d_e) é um espaço pseudo-métrico. A classe de espaços pseudo-métricos abordada é

$$(\mathbb{R}^m, d_e) \mid m \in \mathbb{N}.$$

A solução por força bruta é cara demais para instâncias de tamanho prático. Nós apresentamos uma solução mais eficiente empregando paralelismo, a *FFT* (transformada rápida de Fourier) e a eliminação rápida de vetores de treinamento desfavoráveis. Desenvolvemos um programa — chamado *CyclicKNN* — que implementa essa solução. Reportamos o *speedup* desse programa em comparação com a força bruta sequencial, processando bases de dados de referência.

Palavras-chave

aceleração, análise de dados multidimensionais, k-nearest neighbors, k vizinhos mais próximos, matriz circulante, processamento de imagem, programação paralela, transformada rápida de Fourier

Abstract

de Moraes Neto, Jorge Peixoto. **Acceleration of a variation of the k-nearest neighbors problem**. Goiânia, 2014. 98p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

Let M be a metric space and let P be a subset of M . The well known k-nearest neighbors problem (KNN) consists in finding, given $q \in M$, the k elements of P which are closest to q according to the metric of M . We discuss a variation of KNN for a particular class of pseudo-metric spaces, described as follows. Let $m \in \mathbb{N}$ be a natural number and let d be the Euclidean distance in \mathbb{R}^m . Given $p \in \mathbb{R}^m$:

$$p := (p_1, \dots, p_m)$$

let $C^*(p)$ be the set of the m rotations of p 's coordinates:

$$C^*(p) := \{(p_1, \dots, p_m), (p_2, \dots, p_m, p_1), \dots, (p_m, p_1, \dots, p_{m-1})\}$$

we define the *special distance* d_e as:

$$d_e(p, q) := \min_{p' \in C^*(p)} d(p', q).$$

d_e is a *pseudo-metric*, and (\mathbb{R}^m, d_e) is a *pseudo-metric space*. The class of pseudo-metric spaces under discussion is

$$\{(\mathbb{R}^m, d_e) \mid m \in \mathbb{N}\}$$

The brute force approach is too costly for instances of practical size. We present a more efficient solution employing parallelism, the *FFT* (fast Fourier transform) and the fast elimination of unfavorable training vectors. We describe a program — named *CyclicKNN* — which implements this solution. We report the speedup of this program over serial brute force search, processing reference datasets.

Keywords

acceleration, k-nearest neighbors, circulant matrix, image processing, fast Fourier transform, multidimensional data analysis, parallel programming

Sumário

Lista de Figuras	13
Lista de Tabelas	14
Lista de Algoritmos	15
Notação, nomes e abreviaturas	16
1 Introdução	20
1.1 Roteiro do trabalho	20
2 O KNN tradicional e uma variação	21
2.1 Espaço métrico e espaço pseudo-métrico	21
2.2 KNN	22
2.3 Variação do KNN	22
2.4 Aplicação em visão computacional	28
2.5 Estruturas de dados na literatura	29
2.5.1 Metric skip list	29
2.5.2 Cover Tree	30
3 Aceleração da solução da variação do KNN	31
3.1 Aceleração do cálculo de d_e e v_c	31
3.1.1 Matriz circulante	35
3.1.2 Matriz circulante e FFT	37
3.1.3 Custo de memória	39
3.2 Eliminação rápida de vetores de treinamento desfavoráveis	39
3.2.1 Custo de memória	44
4 Implementação e experimentação	46
4.1 Implementação	46
4.1.1 Computação paralela	46
4.2 Experimentação	47
5 Tentativa de aceleração com interpretação geométrica	50
5.1 Casos particulares $m = 1$, $m = 2$ e $m = 3$	50
5.2 Conjectura: $d_e(p, q)$ pode ser calculado em tempo $O(m)$	56
5.3 Desenvolvimento matemático	60
5.3.1 Matriz de C na forma canônica de Jordan	61
5.3.2 Base ordenada \mathcal{B} correspondente à forma canônica de Jordan	64

5.3.3	Cálculo de distâncias na base ordenada \mathcal{B}	69
5.4	Redução eficiente do produto de matriz circulante por vetor ao cálculo da distância especial	74
6	Conclusão	76
	Referências Bibliográficas	77
A	Propriedades da distância especial	79
B	Apêndice do capítulo 3	85
C	Apêndice do capítulo 5	88
C.1	Em \mathbb{R}^3 , C é uma rotação	89
C.2	A fórmula (5-58) vale também para $R_{2\pi}(\theta_q - \theta_p) \in R_4$	90
C.3	Ortonormalidade da base ordenada \mathcal{B}	91

Lista de Figuras

5.1 Círculo unitário $[-\pi; \pi]$ em quatro partições:

54

Lista de Tabelas

4.1	Bases de dados para medida de desempenho	47
4.2	Parâmetros de medida de desempenho para cada base de dados	48
4.3	<i>Speedup</i> de paralelismo para a base Mnist, $k = 7$, com o algoritmo acelerado com FFT e exclusão de vetores de treinamento	48
4.4	<i>Speedup</i> de paralelismo para a base Mnist, $k = 50$, com o algoritmo acelerado com FFT e exclusão de vetores de treinamento	48
4.5	<i>Speedup</i> de paralelismo para a base Optdigits, $k = 7$, com o algoritmo acelerado com FFT e exclusão de vetores de treinamento	49
4.6	<i>Speedup</i> de paralelismo para a base Optdigits, $k = 50$, com o algoritmo acelerado com FFT e exclusão de vetores de treinamento	49
4.7	Tempo de execução de cada técnica processando a base Mnist (limitada aos 3600 primeiros vetores de consulta), $k = 7$	49
C.1	Situações em que $d_2 \geq d_1$	89
C.2	Situações em que p e q estão no mesmo semiplano	89

Lista de Algoritmos

2.1	Cálculo de d_e e v_c por força bruta	25
2.2	Busca exaustiva pelos vizinhos mais próximos	27
3.1	Cálculo da distância especial através do produto de matriz circulante por vetor	36
3.2	Eliminação de vetores de treinamento desfavoráveis	43

Notação, nomes e abreviaturas

Observação. As seguintes definições estão em ordem alfabética. Por isso, algumas definições fazem referência a definições posteriores.

Notação

- 2^X O conjunto de todos os subconjuntos de X .
 $|X|$ Para um conjunto X , $|X|$ é o seu número de elementos.
 $|x|$ Para um número x (real ou complexo), $|x|$ é o seu módulo.
 $\|p\|$ Norma (comprimento) do vetor p :

$$\|p\| := \sqrt{\langle p, p \rangle}$$

Definido em [3.2](#), página [32](#).

$[a; b]$ Intervalo de a até b , incluindo a e b . Dependendo do contexto, pode ser um intervalo de números inteiros ou um intervalo de números reais.

$]a; b[$ Intervalo de a até b , excluindo a e b . Dependendo do contexto, pode ser um intervalo de números inteiros ou um intervalo de números reais.

$a := b$ a é definido por b .

\bar{z} Conjugado do número complexo z .

\hat{f} Transformada discreta de Fourier (DFT); definida em [3.11](#), página [37](#).

$f(x) \in O(g(x))$ f é da ordem de g .

$f(x) \in o(g(x))$ f tem “crescimento assintótico” estritamente menor que g :

$$\lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} = 0$$

$X \times Y$ Produto cartesiano dos conjuntos X e Y .

$X \setminus Y$ Diferença de conjuntos:

$$X \setminus Y := \{x \in X \mid x \notin Y\}$$

$p \cdot q$ Produto interno canônico entre os vetores $p, q \in \mathbb{R}^m$:

$$p \cdot q := p_1 q_1 + \cdots + p_m q_m$$

definido em 3.1, página 31.

p^c Para $p \in \mathbb{R}^m$ e $c \in \mathbb{Z}$, definimos

$$p^c := C^c p.$$

Essa notação é usada em algumas partes desse trabalho para abreviação.

f^j Para uma função f e um número natural j , definimos f^j como a composição de j funções iguais a f :

$$f^j := f \circ \cdots \circ f.$$

Definimos também f^{-j} como a composição de j funções iguais a f^{-1} :

$$f^{-j} := f^{-1} \circ \cdots \circ f^{-1}.$$

Definido em 2.7, página 22.

$f(X)$ Para um conjunto X e uma função f cujo domínio contém X , definimos:

$$f(X) := \{f(x) \mid x \in X\}.$$

Em particular, se D é o domínio de f então $f(D)$ é a imagem de f .

C Operador linear $\mathbb{R}^m \rightarrow \mathbb{R}^m$. Dado $p = (p_1, \dots, p_m)^T \in \mathbb{R}^m$,

$$C(p) := (p_2, \dots, p_m, p_1)^T$$

definido em 2.5, página 22. $C(p)$ também é representado sem os parênteses:

$$Cp := C(p)$$

$C^*(p)$ Para $p \in \mathbb{R}^m$, $C^*(p)$ é um conjunto de m vetores definido por:

$$C^*(p) := \{p, Cp, \dots, C^{m-1}p\}$$

Definido em 2.10, página 23.

$d(p, q)$ Distância euclidiana entre $p, q \in \mathbb{R}^m$:

$$d(p, q) := \sqrt{(q_1 - p_1)^2 + \cdots + (q_m - p_m)^2}$$

$d_e(p, q)$ distância especial entre os vetores $p, q \in \mathbb{R}^m$:

$$d_e(p, q) := \min_{p' \in C^*(p)} d(p', q)$$

d_e é uma pseudo-métrica em \mathbb{R}^m . É definida em 2.12, página 23.

\mathbb{F}_m Matriz correspondente à DFT em \mathbb{C}^m . Definida em (3-40), página 38.

k Número de vetores de treinamento mais próximos que se deseja encontrar para cada vetor de consulta.

m Dimensão do espaço vetorial real que contém todos os vetores de treinamento e de consulta. Esse espaço vetorial é \mathbb{R}^m .

n Número de vetores de treinamento:

$$n := |P|$$

norma Ver $\|p\|$.

P Conjunto dos vetores de treinamento. É subconjunto de \mathbb{R}^m .

$P_k(q)$ Conjunto dos k vizinhos mais próximos de q dentre os elementos de P . Definido em 2.17, página 25.

Q Conjunto dos vetores de consulta. É subconjunto de \mathbb{R}^m .

r Número de vetores de consulta:

$$r := |Q|$$

$\mathbb{R}_{\geq 0}$ Conjunto dos números reais não-negativos (incluindo 0).

$v_c(p, q)$ Rotação vizinha. É o elemento de $C^*(p)$ mais próximo (conforme a distância euclidiana) de q . Definido em 2.14, página 24.

ω A m -ésima raiz primitiva da unidade:

$$\omega := e^{-i \frac{2\pi}{m}}$$

Observação. Neste trabalho usamos a vírgula para separar elementos de um conjunto ou sequência:

$$\text{Divisores de } 6 : \{1, 2, 3, 6\}$$

e usamos o ponto para separar a parte inteira da parte fracionária de um número real:

$$\pi \approx 3.141592$$

Nomes e abreviaturas

C++	Uma linguagem de programação conhecida pelo excelente desempenho. Seu último padrão é o C++11, lançado em 2011.
DFT	<i>discrete Fourier transform</i> , transformada discreta de Fourier. Definida em 3.11 , página 37 .
Distância especial	Ver $d_e(p, q)$.
FFT	<i>fast Fourier transform</i> , um algoritmo eficiente para calcular a DFT e a IDFT.
IDFT	<i>inverse discrete Fourier transform</i> , a transformada inversa da DFT.
KNN	<i>k-nearest neighbor</i> , busca dos k vizinhos mais próximos.

Introdução

Neste trabalho estudamos uma variação do problema *KNN* (*k-nearest neighbor*, busca pelos k vizinhos mais próximos). Achar o vizinho mais próximo de um ponto conforme uma métrica dada é um problema matemático clássico com muitas aplicações práticas. Algumas dessas aplicações incluem buscas em banco de dados, em particular para dados complexos como dados multimídia ou estruturas biológicas, por exemplo em estruturas de proteína ou dados de genoma. Outros usos são em compressão de dados com perda, em que o dado pode ser codificado (*encoded*) pelo representante mais próximo de um conjunto fixo de representantes, e aprendizagem de máquina [2]. Comparar dois pontos pode ser caro, e deseja-se usar estruturas de dados que permitam a busca por vizinho mais próximo com um número pequeno de comparações [6] (neste trabalho, oferecemos uma solução que rapidamente exclui vetores de treinamento desfavoráveis).

A variação do KNN abordada neste trabalho tem aplicação em reconhecimento de padrão em imagem. Reconhecimento de padrão é um aspecto fundamental de muitos problemas em visão computacional, como reconhecimento de cena ou objeto, construção de estrutura 3D a partir de múltiplas imagens, e outros. [10]

1.1 Roteiro do trabalho

O capítulo 2 apresenta o contexto do trabalho na literatura, define formalmente o problema e oferece uma aplicação em visão computacional.

O capítulo 3 apresenta a nossa contribuição para a aceleração da variação do KNN. As seções 3.1 e 3.2 apresentam técnicas que, em conjunto, aceleram drasticamente o problema.

O capítulo 4 descreve uma implementação paralela em C++11 da solução apresentada no capítulo 3 e apresenta o desempenho e o *speedup* empíricos de tal programa.

O capítulo 5 apresenta uma outra tentativa de cálculo eficiente da distância especial: interpretação geométrica. Até o momento não soubemos tirar proveito dessa abordagem.

O capítulo 6 conclui o trabalho e propõe trabalhos futuros.

O KNN tradicional e uma variação

Este capítulo apresenta o contexto do trabalho na literatura, define formalmente o problema e oferece uma aplicação em visão computacional.

2.1 Espaço métrico e espaço pseudo-métrico

Definição 2.1 (Métrica). *Uma métrica em um conjunto M é uma função $d: M \times M \rightarrow \mathbb{R}$ que associa a cada par ordenado $u, v \in M$ um número real $d(u, v)$, chamado a distância de u a v , de modo a satisfazer as seguintes condições para quaisquer $u, v, w \in M$:*

1. $d(u, u) = 0$;
2. $u \neq v \implies d(u, v) > 0$;
3. $d(u, v) = d(v, u)$
4. $d(u, w) \leq d(u, v) + d(v, w)$.

A condição 4 é conhecida como desigualdade triangular ou desigualdade do triângulo. [8]

A distância especial proposta neste trabalho não satisfaz a condição 2 (há um contraexemplo no apêndice, teorema A.2) e, portanto, não é uma métrica.

Definição 2.2 (Espaço métrico). *Um espaço métrico é um par (M, d) onde M é um conjunto e d é uma métrica em M .*

Definição 2.3 (Pseudo-métrica). *Uma pseudo-métrica em um conjunto M é uma função real $d: M \times M \rightarrow \mathbb{R}$ que cumpre as condições de uma métrica, exceto que $d(u, v)$ pode ser 0 mesmo quando $u \neq v$. Mais precisamente, uma pseudo-métrica satisfaz as seguintes condições para quaisquer $u, v, w \in M$:*

1. $d(u, u) = 0$;
2. $d(u, v) \geq 0$;
3. $d(u, v) = d(v, u)$
4. $d(u, w) \leq d(u, v) + d(v, w)$.

Definição 2.4 (Espaço pseudo-métrico). *Um espaço pseudo-métrico é um par (M, d) onde M é um conjunto e d é uma pseudo-métrica em M . [8]*

A condição 1 da definição 2.3 é satisfeita pela nossa distância especial, como pode ser trivialmente verificado. As condições 2, 3 e 4 também são satisfeitas (provas em A.1, A.4 e A.6 respectivamente). Portanto a nossa distância especial é uma pseudo-métrica.

2.2 KNN

Seja M um espaço métrico e $P \subseteq M$ o conjunto de treinamento; os elementos de P são os vetores de treinamento. O conhecido problema k vizinhos mais próximos (*k-nearest neighbors*, *KNN*) consiste em, dado um vetor de consulta $q \in M$, encontrar os k vetores de treinamento mais próximos de q conforme a métrica de M . Como vimos acima, a distância especial abordada neste trabalho é uma pseudo-métrica, não uma métrica, e, portanto, convém ter cuidado ao aplicar soluções propostas na literatura para espaços métricos.

2.3 Variação do KNN

Esta seção define formalmente o problema abordado neste trabalho.

Definição 2.5 (C). *Seja m um natural. Seja*

$$C: \mathbb{R}^m \rightarrow \mathbb{R}^m \quad (2-1)$$

a função tal que, dado $p = (p_1, \dots, p_m)^T \in \mathbb{R}^m$,

$$C(p) := (p_2, \dots, p_m, p_1)^T. \quad (2-2)$$

O nome “C” vem de “circular”, pois C desloca as coordenadas do vetor uma posição para cima (considerando o vetor como vetor-coluna) mantendo a ordem circular.

Observação 2.6. *Quando não houver perigo de confusão representaremos $C(p)$ sem os parênteses:*

$$Cp := C(p) \quad (2-3)$$

Definição 2.7 (Número sobrescrito como índice de composição). *Sejam D, C dois conjuntos, e $f: D \rightarrow C$ uma função. Seja f^0 a função identidade:*

$$\forall x \in D, f^0(x) = x. \quad (2-4)$$

Para $j \in \mathbb{N}$ seja

$$f^j := f \circ f^{j-1} \quad (2-5)$$

Seja f^{-1} a função inversa de f (se existir):

$$\forall x \in D, f^{-1}(f(x)) = x. \quad (2-6)$$

Para $j \in \mathbb{N}$ seja

$$f^{-j} := f^{-1} \circ f^{-(j-1)}. \quad (2-7)$$

Exemplo 2.8.

$$f^2 = f \circ f \quad (2-8)$$

$$f^{-2} = f^{-1} \circ f^{-1} \quad (2-9)$$

Observação 2.9. Para uma função $f: D \rightarrow C$:

- Para $j \geq 2$, f^j só está bem definida se $f(D) \subseteq D$.
- f^{-1} só existe se f for injetiva.
- Para $j \geq 2$, f^{-j} só existe se $D \subseteq f(D)$.

Definição 2.10 (C^*). Dado um ponto $p \in \mathbb{R}^m$, seja

$$C^*(p) := \{p, Cp, \dots, C^{m-1}p\}. \quad (2-10)$$

$C^*(p)$ é um conjunto de m elementos. Ou seja, $C^*(p)$ é o conjunto de todos os pontos de \mathbb{R}^m que têm as mesmas coordenadas que p na mesma ordem circular. Cada elemento de $C^*(p)$ é chamado uma rotação de p .

Exemplo 2.11. Se

$$m = 3 \quad (2-11)$$

$$p = (1, 2, 3)^T \quad (2-12)$$

então

$$C^*(p) = \{(1, 2, 3)^T, (2, 3, 1)^T, (3, 1, 2)^T\} \quad (2-13)$$

Definição 2.12 (d_e). $d_e: \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ é a função tal que, para $p, q \in \mathbb{R}^m$:

$$d_e(p, q) := \min_{p' \in C^*(p)} d(p', q) \quad (2-14)$$

onde $d(p', q)$ é a distância euclidiana entre p' e q . Chamamos $d_e(p, q)$ de distância especial entre p e q .

Exemplo 2.13. Se

$$m = 3 \quad (2-15)$$

$$p = (1, 2, 3)^T \quad (2-16)$$

$$q = (2, 3, 1)^T \quad (2-17)$$

então

$$d_e(p, q) = 0 \quad (2-18)$$

Definição 2.14 (Rotação vizinha). *Seja*

$$v_c: \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^m \quad (2-19)$$

uma função tal que, dados $p, q \in \mathbb{R}^m$

$$d(v_c(p, q), q) = d_e(p, q) \quad (2-20)$$

Ou seja, $v_c(p, q)$ é um elemento de $C^(p)$ que minimiza a distância euclidiana a q . Se houver mais de um elemento de $C^*(p)$ que minimiza a distância euclidiana a q , a escolha é arbitrária.*

Exemplo 2.15. Se

$$m = 2 \quad (2-21)$$

$$p = (0, 1)^T \quad (2-22)$$

$$q = (0, 0)^T \quad (2-23)$$

então:

$$Cp = (1, 0)^T \quad (2-24)$$

$$C^*(p) = \{p, Cp\} \quad (2-25)$$

$$d(p, q) = 1 \quad (2-26)$$

$$d(Cp, q) = 1 \quad (2-27)$$

$$d_e(p, q) = 1 \quad (2-28)$$

$$v_c(p, q) = p \quad (2-29)$$

Exemplo 2.16. Se

$$m = 3 \quad (2-30)$$

$$p = (0, 1, 2)^T \quad (2-31)$$

$$q = (3, 1, 2)^T \quad (2-32)$$

então:

$$Cp = (1, 2, 0)^T \quad (2-33)$$

$$C^2p = (2, 0, 1)^T \quad (2-34)$$

$$C^*(p) = \{p, Cp, C^2p\} \quad (2-35)$$

$$d(p, q) = 3 \quad (2-36)$$

$$d(Cp, q) = 3 \quad (2-37)$$

$$d(C^2p, q) = \sqrt{3} \quad (2-38)$$

$$d_e(p, q) = \sqrt{3} \quad (2-39)$$

$$v_c(p, q) = C^2p \quad (2-40)$$

Calcular d_e e v_c por força bruta sequencial tem complexidade de tempo $O(m^2)$. Veja o algoritmo 2.1.

Algoritmo 2.1: Cálculo de d_e e v_c por força bruta

Entrada: Vetores $p, q \in \mathbb{R}^m$

Saída: $d_e(p, q)$ e $v_c(p, q)$

```

1   $c\_min \leftarrow 0$ 
2   $distancia\_min \leftarrow d(p, q)$ 
3  para cada inteiro  $c$  tal que  $1 \leq c < m$  faça
4       $d\_c \leftarrow d(C^c p, q)$ 
5      se  $d\_c < distancia\_min$  então
6           $distancia\_min \leftarrow d\_c$ 
7           $c\_min \leftarrow c$ 
8      fim
9  fim
10  $d_e(p, q) \leftarrow distancia\_min$ 
11  $v_c(p, q) \leftarrow C^{c\_min} p$ 

```

O passo 4 demora $O(m)$ pois os vetores têm m dimensões. O condicional 5 demora $O(1)$. Portanto cada iteração do laço 3 demora $O(m)$ e as $m - 1$ iterações demoram $O(m^2)$. O passo 2 demora $O(m)$. Portanto o algoritmo 2.1 demora $O(m^2)$.

Definição 2.17 (P_k , as k rotações mais próximas). *Sejam P e Q dois subconjuntos de \mathbb{R}^m . Seja k um natural. Seja*

$$P_{vc}: Q \rightarrow 2^{\mathbb{R}^m} \quad (2-41)$$

a função tal que, dado $q \in Q$:

$$P_{vc}(q) := \{v_c(p, q) \mid p \in P\} \quad (2-42)$$

Seja

$$P_k: Q \rightarrow 2^{\mathbb{R}^m} \quad (2-43)$$

a função tal que, dado $q \in Q$, $P_k(q)$ é o conjunto dos k elementos de $P_{vc}(q)$ que estão mais próximos de q . Esses k elementos são chamados os k vizinhos mais próximos de q . Ou seja: para $q \in Q$, $P_k(q)$ é um conjunto tal que:

$$P_k(q) \subseteq P_{vc}(q) \quad (2-44)$$

$$|P_k(q)| = k \quad (2-45)$$

$$\forall v, u \in P_{vc}(q), v \in P_k(q) \wedge u \notin P_k(q) \implies d(v, q) \leq d(u, q) \quad (2-46)$$

Exemplo 2.18. Se

$$m = 2 \quad (2-47)$$

$$P = \{(0, 1)^T, (1, -1)^T, (1, 2)^T\} \quad (2-48)$$

$$q = (0, 1)^T \quad (2-49)$$

$$k = 2 \quad (2-50)$$

$$(2-51)$$

então

$$P_k(q) = \{(0, 1)^T, (-1, 1)^T\} \quad (2-52)$$

O problema abordado neste trabalho consiste em calcular $P_k(q)$ para cada $q \in Q$. A solução intuitiva é a busca exaustiva, conforme o algoritmo [2.2](#):

Algoritmo 2.2: Busca exaustiva pelos vizinhos mais próximos

Entrada: Natural k , conjuntos $P, Q \subset \mathbb{R}^m$

Saída: Para cada $q \in Q$ retorna $P_k(q)$ (conforme a definição 2.17).

```

1  para cada  $q \in Q$  faça
2      para cada  $p \in P$  faça
3           $d' \leftarrow d_e(p, q)$ 
4           $v' \leftarrow v_c(p, q)$ 
5           $vizinhos \leftarrow vizinhos \cup \{v'\}$ 
6      fim
7      Dentre os elementos de  $vizinhos$ , selecione os  $k$  mais próximos de  $q$ 
8      Construa o conjunto  $P_k(q)$  com os  $k$  vizinhos selecionados no passo 7
9      Adicione  $P_k(q)$  à saída
10 fim
  
```

Sejam

$$n := |P| \quad (2-53)$$

$$r := |Q| \quad (2-54)$$

e seja c_d a complexidade de tempo combinada dos passos 3 e 4. Se a complexidade de tempo do passo 5 (uma simples inserção em conjunto) for $O(c_d)$ então a complexidade de tempo de cada iteração do laço 2 é $O(c_d)$ e as n iterações têm complexidade total $O(nc_d)$. Para selecionar os k vizinhos mais próximos de q no passo 7, pode-se construir uma heap de mínimo — complexidade $O(n)$ — e extrair k vezes o mínimo da heap — complexidade $O(k \log n)$. Assim a complexidade de tempo do passo 7 é $O(n + k \log n)$. Os passos 8 e 9 têm complexidade $O(k)$. Assim cada iteração do laço 1 tem complexidade $O(nc_d + n + k \log n + k) = O(nc_d + k \log n)$. As r iterações têm complexidade total $O(rnc_d + rk \log n)$.

Se d_e e v_c forem calculados pelo algoritmo 2.1 então $c_d = O(m^2)$ e o algoritmo 2.2 tem complexidade $O(rnm^2 + rk \log n)$.

Desejamos resolver rapidamente instâncias em que $k = 50$, $m = 200$, $r = 100$ e $n = 10^7$. Por isso a complexidade $O(rnm^2 + rk \log n)$ é muito ruim.

Neste trabalho abordamos duas contribuições para acelerar o algoritmo 2.2:

- Calcular $P_k(q)$ sem calcular $d_e(p, q)$ e $v_c(p, q)$ para todo $p \in P$.
- Acelerar o cálculo de d_e e v_c .

Ambas contribuições são descritas no capítulo 3.

2.4 Aplicação em visão computacional

Reconhecimento de padrão em imagem é um aspecto fundamental de muitos problemas em visão computacional, como reconhecimento de cena ou objeto e reconstrução de estrutura 3D a partir de múltiplas imagens. [10]

Uma operação importante em processamento de imagem é extrair características (*features*) de um objeto em uma imagem de referência e procurar características semelhantes em um conjunto de imagens de treinamento. O número de características extraídas da imagem de referência costuma ser pequeno (da ordem de 10 ou 100) mas o número de características extraídas do conjunto de treinamento pode ser muito maior. Assim, deseja-se métodos eficientes.

O artigo [10] apresenta um método eficiente e eficaz para extrair características distintivas invariantes de imagens. Tais características são chamadas *SIFT features* (SIFT: *Scale-Invariant Feature Transform*) e são úteis para casamento confiável entre vistas diferentes de um objeto ou cena. Elas são invariantes a rotação e mudança de escala, e proveem casamento robusto a distorção afim, mudança de ponto de vista 3D, ruído, e mudança de iluminação. Elas são fortemente distintivas: uma SIFT feature pode ser corretamente casada — com alta probabilidade — com um grande banco de dados de SIFT features extraídas de muitas imagens. O artigo descreve brevemente uma abordagem (usando busca aproximada por vizinho mais próximo) para usar essas características para reconhecimento de objeto, e apresenta dois exemplos bem-sucedidos.

Uma medida de similaridade — baseada em *dual trees* e em transformada wavelet orientada complexa — desenvolvida recentemente mostrou-se benéfica para processamento multi-dimensional de imagem. [11] Essa medida tem alguns aspectos vantajosos em comparação com a SIFT proposta em [10], e usa a distância especial abordada neste trabalho.

A principal razão do uso da distância especial nesse contexto é que objetos podem ser representados rotacionados. Depois de translação, rotação e mudança simples de escala são os dois graus de liberdade mais importante para objetos estáticos. Depois disso vêm *shear distortions*, inclusive mudanças diferenciais de escala nas duas coordenadas cartesianas, e efeitos de perspectiva. Todas essas transformações, exceto efeitos de perspectiva, podem ser acomodados por um modelo de distorção afim 2D, com seis graus de liberdade, dos quais os dois graus de translação e o grau de rotação e o grau de mudança de escala tendem a ser os mais importantes para mudanças moderadas de ponto de vista.

Há duas abordagens gerais para lidar com rotações em um descritor. A primeira, usada pelo SIFT, alinha o descritor com uma direção dominante de gradiente no ponto de consulta. Para essa abordagem ter sucesso, o ponto de treinamento correspondente deve ter uma direção dominante de gradiente similar. A segunda abordagem, usada

nos descritores introduzidos recentemente por [11], tenta casar o ponto de consulta com o ponto de treinamento usando todas as possíveis rotações (dentro de um limite de resolução) do ponto de consulta através de um processo de casamento eficiente. A abordagem SIFT tende a falhar se o ponto de consulta não tem uma direção dominante de gradiente.

2.5 Estruturas de dados na literatura

Como vimos no capítulo 1, o KNN é um problema matemático clássico com muitas aplicações práticas. Na literatura (pelo menos até 2002) a maior parte da pesquisa em algoritmos de busca de vizinhos mais próximos focou no caso euclidiano, que é particularmente importante porque muitas aplicações práticas lidam com vetores de características (*feature vectors*) que são naturalmente embutidos em um espaço euclidiano. Porém, em muitos problemas práticos de busca, assim como no nosso caso, a métrica é não-euclidiana. Algoritmos exatos de busca de vizinho mais próximo para espaços métricos gerais são bastante lentos, o que motiva a busca por outras classes de espaços métricos onde a busca é tratável. [6] Também há soluções aproximadas [10].

Foram desenvolvidas muitas estruturas de dados com muito bom desempenho (tempo logarítmico por operação) para espaços euclidianos de baixa dimensão. [6] No problema abordado pelo presente trabalho, o espaço métrico não é euclidiano, e a dimensão é alta.

2.5.1 Metric skip list

O artigo [6], desenvolve uma estrutura de dados eficiente para buscas de vizinho mais próximo em uma classe especial de métricas chamada *growth-constrained*. Seja $\mathcal{M} = (M, d_M)$ um espaço métrico, seja $P \subseteq M$ e seja

$$B_p(x) := \{p' \in P \mid d_M(p, p') \leq x\} \quad (2-55)$$

Dizemos que P é *growth-constrained* se existem ρ, c tais que para qualquer $x > 0$ e $p \in M$ vale

$$|B_p(x)| \geq \rho \implies |B_p(2x)| \leq c |B_p(x)| \quad (2-56)$$

A estrutura de dados apresentada, a *metric skip list*, permite busca de vizinho mais próximo em tempo logarítmico em n (n é o número de vetores de treinamento) e polinomial na taxa de expansão c (que infelizmente pode ser exponencial na dimensão do espaço métrico) com alta probabilidade. O artigo não menciona uma implementação disponível. A estrutura de dados consome espaço $c^{O(1)} n \log n$ [2] o que é muito desencorajador

quando $n \approx 10^7$ e $m = 200$ (c pode ser exponencial em m !). Nesse quesito, essa estrutura de dados perde para a *cover tree*, que gasta apenas espaço $O(n)$. O tempo de construção é $c^{O(1)}n \log n$ e o tempo de consulta é $c^{O(1)} \log n$; nesses quesitos, não fica claro se ela ganha ou perde da *cover tree*. A implementação seria difícil; o algoritmo supõe algumas premissas não-triviais, que não fomos capazes de verificar para o nosso problema.

2.5.2 Cover Tree

O artigo [2], apresenta uma estrutura de dados em árvore para operações de vizinho mais próximo rápidas em espaços métricos gerais de n pontos (onde o conjunto de dados consiste de n pontos). A estrutura de dados requer espaço apenas $O(n)$ independente da estrutura da métrica — de fato, é focada em economia de espaço — mas mantém todas as propriedades de desempenho de uma *navigating net*. Se o conjunto de pontos tem uma constante de expansão limitada c , que é uma medida da dimensionalidade intrínseca (como definida em [6]) a estrutura de dados *cover tree* pode ser construída em tempo $O(c^6 n \log n)$. Buscas de vizinho mais próximo são feitas em tempo apenas logarítmico em n : $O(c^{12} \log n)$. Esses limites são preocupantes pois c pode ser exponencial em m [6], e queremos resolver um problema em que $m = 200$. Os resultados experimentais do artigo mostram *speedups* (em relação à busca exaustiva) entre uma e várias ordens de grandeza em bases de dados naturais de aprendizado de máquina.

Aceleração da solução da variação do KNN

Este capítulo apresenta as nossas contribuições para a aceleração da variação do KNN abordada neste trabalho. A seção 3.1 apresenta uma técnica (empregando a FFT) para acelerar drasticamente o cálculo de d_e e v_c ; a seção 3.2 apresenta uma técnica para calcular $P_k(q)$ sem calcular d_e e v_c para cada elemento de $P \times Q$. Desenvolvemos um programa, chamado CyclicKNN, implementando essa solução em computação paralela. Ele é apresentado em 4.1.

3.1 Aceleração do cálculo de d_e e v_c

d_e e v_c podem ser calculados muito mais rapidamente (do que por força bruta) por um algoritmo que tira proveito da estrutura de *matrizes circulantes*. Para chegar lá precisamos de algumas definições e teoremas:

Definição 3.1 (Produto interno). *Seja E um espaço vetorial cujo corpo é \mathbb{R} ou \mathbb{C} . Um produto interno em E é uma função que atribui a cada par de vetores $u, v \in E$ um escalar $\langle u, v \rangle$ e satisfaz as seguintes propriedades para quaisquer vetores $u, v, w \in E$ e para qualquer escalar x :*

1. $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$
2. $\langle xu, v \rangle = x\langle u, v \rangle$
3. $\langle u, v \rangle = \overline{\langle v, u \rangle}$
4. $u \neq 0 \implies \langle u, u \rangle > 0$

Se o corpo é \mathbb{R} , a condição (3) pode ser simplificada para

$$\langle u, v \rangle = \langle v, u \rangle \quad (3-1)$$

[5]

Dado um espaço vetorial E , pode haver várias funções que satisfazem as condições acima. No espaço euclidiano \mathbb{R}^m , define-se o produto interno canônico dos

vetores

$$u := (u_1, \dots, u_m)^T \quad (3-2)$$

$$v := (v_1, \dots, v_m)^T \quad (3-3)$$

por

$$u_1 v_1 + \dots + u_m v_m. \quad (3-4)$$

[9]

Dados $u, v \in \mathbb{R}^m$, denotamos o seu produto interno canônico por $u \cdot v$.¹ Portanto:

$$u \cdot v := u_1 v_1 + \dots + u_m v_m \quad (3-5)$$

Definição 3.2 (Norma). *Dado um espaço vetorial E munido de produto interno, então para $u \in E$ o número real não-negativo*

$$\|u\| := \sqrt{\langle u, u \rangle} \quad (3-6)$$

chama-se a norma ou o comprimento de u .² [9]

Definição 3.3 (Distância definida pela norma). *Em um espaço vetorial E munido de produto interno, definimos a distância entre os vetores $u, v \in E$ como*

$$d(u, v) := \|v - u\|. \quad (3-7)$$

[9]

Observação 3.4. *Num espaço vetorial real a distância definida em 3.3 satisfaz a propriedade:*

$$\forall u, v \in E, d^2(u, v) = \|u\|^2 + \|v\|^2 - 2\langle u, v \rangle \quad (3-8)$$

Para verificá-la, bastam manipulações algébricas simples:

$$\begin{aligned} d^2(u, v) &= \\ \|v - u\|^2 &= \langle (v - u), (v - u) \rangle = \langle v, v \rangle - \langle v, u \rangle - \langle u, v \rangle + \langle u, u \rangle = \\ &= \|v\|^2 + \|u\|^2 - 2\langle u, v \rangle \end{aligned} \quad (3-9)$$

¹A notação $u \cdot v$ é comumente usada para o produto interno canônico de \mathbb{R}^m . [5]

²Aqui adotamos a notação de [5] e fazemos uma escolha diferente de [9], que denota a norma de u por $|u|$.

Observação 3.5. No domínio $\mathbb{R}_{\geq 0}$ a função $f(x) = x^2$ é estritamente crescente:

$$\forall x, y \in \mathbb{R}_{\geq 0}, y > x \implies y^2 > x^2 \quad (3-10)$$

Portanto, dado um conjunto $X \in \mathbb{R}_{\geq 0}$ e sendo $X' := \{x^2 \mid x \in X\}$ ³, temos

$$\min(X') = (\min(X))^2. \quad (3-11)$$

Ou seja, o quadrado do mínimo é o mínimo dos quadrados. Resultado análogo vale para o máximo.

Teorema 3.6. Sejam E um espaço vetorial real munido de produto interno, $t \in \mathbb{R}_{\geq 0}$, $m \in \mathbb{N}$ e $U \subseteq E$ tais que

$$u \in U \implies \|u\| = t \quad (3-12)$$

Ou seja, U é um subconjunto de E cujos elementos têm todos a mesma norma.

Seja $q \in E$ e seja $u_{\min} \in U$ tal que:

$$d(u_{\min}, q) = \min_{u \in U} d(u, q) \quad (3-13)$$

Então

$$\langle u_{\min}, q \rangle = \max_{u \in U} \langle u, q \rangle \quad (3-14)$$

Ou seja, quando maximizamos o produto interno $\langle u, q \rangle$ (para $u \in U$) nós minimizamos a respectiva distância.

Prova. Como vimos na observação 3.4,

$$d^2(u, q) = \|u\|^2 + \|q\|^2 - 2\langle u, q \rangle \quad (3-15)$$

Para $u \in U$, por hipótese temos $\|u\| = t$, portanto

$$d^2(u, q) = (t^2 + \|q\|^2) - 2\langle u, q \rangle. \quad (3-16)$$

Na última expressão acima (o lado direito da última equação), colocamos entre parênteses os termos que independem de u . Note que $d^2(u, q)$ é uma função estritamente decrescente de $\langle u, q \rangle$. Portanto $d^2(u, q)$ é mínimo quando $\langle u, q \rangle$ é máximo. Como \sqrt{x} é uma função estritamente crescente de x então o mínimo de $d^2(u, q)$ corresponde ao mínimo de $d(u, q)$. \square

³Ou seja, X' é o conjunto dos quadrados dos elementos de X .

Lema 3.7. *Os quadrados da norma de todos os vetores $p \in P$ e de todos os vetores $q \in Q$ podem ser computados em tempo $O(m(n+r))$.*

Prova. A complexidade de tempo de calcular o quadrado da norma de um vetor de \mathbb{R}^m é $O(m)$:

$$\|u\|^2 = u_1^2 + \dots + u_m^2 \quad (3-17)$$

São n vetores em P e r vetores em Q ; $n+r$ vetores no total. Portanto a complexidade de tempo total de pré-computar todos os quadrados das normas é $O(m(n+r))$. \square

Observação 3.8. *Os quadrados das normas de todos os vetores $p \in P$ e $q \in Q$ podem ser pré-computados antes do laço mais interno.*

Teorema 3.9. *Suponhamos que os quadrados das normas de todos os vetores $p \in P$ e $q \in Q$ são conhecidos (vide 3.7 e 3.8).*

Para $p \in P$ e $q \in Q$, seja $\max p(p, q)$ o máximo do produto interno canônico entre as rotações de p e q :

$$\max p(p, q) := \max_{0 \leq c < m} C^c p \cdot q \quad (3-18)$$

Seja c_m a complexidade de tempo média do cálculo de $\max p(p, q)$. Então o cálculo de $d_e(p, q)$ tem complexidade média $c_m + O(1)$.

Prova. Pela definição de v_c , temos

$$d(v_c(p, q), q) = \min_{p' \in C^*(p)} d(p', q) \quad (3-19)$$

Do corolário 5.7 e do teorema 3.6 segue que

$$v_c(p, q) \cdot q = \max p(p, q) \quad (3-20)$$

Disso e da prova do teorema 3.6 segue que

$$d_e(p, q) = d(v_c(p, q), q) = \sqrt{\|q\|^2 - 2v_c \cdot q + \|p\|^2} = \sqrt{\|q\|^2 - 2\max p(p, q) + \|p\|^2} \quad (3-21)$$

Assim:

$$d_e(p, q) = \sqrt{\|q\|^2 - 2\max p(p, q) + \|p\|^2} \quad (3-22)$$

Portanto o cálculo de $d_e(p, q)$ tem complexidade $c_m + O(1)$. \square

3.1.1 Matriz circulante

Dados $p \in P, q \in Q$, queremos calcular

$$d_e(p, q) := \min_{r \in C^*(p)} d(r, q) \quad (3-23)$$

conforme as definições de 2.3. Pelo teorema 3.9, se soubermos o máximo do produto interno

$$\max p(p, q) := \max_{0 \leq c < m} (C^c p) \cdot q \quad (3-24)$$

então podemos calcular $d_e(p, q)$ com um custo adicional de apenas $O(1)$.

Se $p = (p_1, \dots, p_m)^T$, seja M_p a matriz

$$M_p := \begin{pmatrix} p_m & p_1 & \dots & p_{m-1} \\ p_{m-1} & p_m & \dots & p_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ p_1 & p_2 & \dots & p_m \end{pmatrix} \quad (3-25)$$

Ou seja, a j -ésima linha de M_p é igual a $C^{m-j}p$. Então podemos calcular todos os m produtos internos $(C^c p) \cdot q$ (para $0 \leq c < m$) através de um produto de matriz por vetor:

$$M_p q = \begin{pmatrix} p_m & p_1 & \dots & p_{m-1} \\ p_{m-1} & p_m & \dots & p_{m-2} \\ \vdots & \vdots & \ddots & \vdots \\ p_1 & p_2 & \dots & p_m \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_m \end{pmatrix} = \begin{pmatrix} (C^{m-1}p) \cdot q \\ (C^{m-2}p) \cdot q \\ \vdots \\ p \cdot q \end{pmatrix} \quad (3-26)$$

Portanto as m coordenadas do vetor-coluna $M_p q$ são os m produtos internos $(C^c p) \cdot q$ para $0 \leq c < m$. Portanto $\max p(p, q)$ é o máximo de tais coordenadas; o cálculo do máximo de m números reais custa $m - 1$ comparações. Assim, a partir de $M_p q$ podemos calcular $d_e(p, q)$ em tempo $O(m)$. Portanto se pudermos calcular o produto $M_p q$ em tempo $o(m^2)$ (possivelmente após um pré-processamento) então podemos calcular $d_e(p, q)$ em tempo $o(m^2)$ através do algoritmo 3.1:

Algoritmo 3.1: Cálculo da distância especial através do produto de matriz circulante por vetor

Entrada: Vetores $p, q \in \mathbb{R}^m$

Saída: $d_e(p, q)$ e $v_c(p, q)$ (conforme a definição 2.14)

1 $produto \leftarrow M_p q$

2

$$max \leftarrow \arg \max_{j \in [1; m]} (produto[j]) \quad (3-27)$$

3 $d_e(p, q) \leftarrow \|p\|^2 + \|q\|^2 - 2 \cdot produto[max]$

4 $c \leftarrow m - max$

5 $v_c(p, q) \leftarrow C^c p$

De fato o produto $M_p q$ pode ser calculado em tempo $o(m^2)$ — especificamente, em tempo $O(m \log m)$ — se aproveitarmos a propriedade de M_p ser uma *matriz circulante*.

Definição 3.10 (Matriz circulante). Segundo [4] e [7] uma matriz circulante (circulant matrix) é uma matriz quadrada em que cada vetor-coluna é uma versão “deslocada ciclicamente uma posição pra baixo” do seu predecessor. Precisamente: em \mathbb{R}^m seja

$$S_m: \mathbb{R}^m \rightarrow \mathbb{R}^m \quad (3-28)$$

o operador linear que desloca ciclicamente uma posição para baixo as coordenadas de um vetor. Ou seja, se (e_1, \dots, e_m) é a base canônica de \mathbb{R}^m então:

$$S_m e_1 := e_2 \quad (3-29)$$

$$\vdots \quad (3-30)$$

$$S_m e_{m-1} := e_m \quad (3-31)$$

$$S_m e_m := e_1 \quad (3-32)$$

Então a matriz circulante que tem v como o primeiro vetor-coluna é

$$(v, S_m v, \dots, S_m^{m-1} v) \quad (3-33)$$

Por exemplo uma matriz circulante 4×4 tem a forma:

$$\begin{pmatrix} v_1 & v_4 & v_3 & v_2 \\ v_2 & v_1 & v_4 & v_3 \\ v_3 & v_2 & v_1 & v_4 \\ v_4 & v_3 & v_2 & v_1 \end{pmatrix} \quad (3-34)$$

O produto de uma matriz circulante $m \times m$ por um vetor $m \times 1$ pode ser feito em tempo $O(m \log m)$ através da *FFT* (*fast Fourier transform, transformada rápida de Fourier*).

3.1.2 Matriz circulante e FFT

FFT (fast Fourier transform) é um algoritmo eficiente para calcular a *DFT* (*discrete Fourier transform, transformada discreta de Fourier*) e sua inversa, a *IDFT* (*inverse discrete Fourier transform*), com apenas $O(m \log m)$ multiplicações complexas. A seguir apresentamos a DFT e como o produto de matriz circulante por vetor pode ser calculado em tempo $O(m \log m)$ através da FFT.[7]

Na definição seguinte, contamos as coordenadas dos vetores a partir da posição 0 (e não da posição 1 como no resto deste trabalho). Seja

$$\omega := e^{-\frac{j2\pi}{m}} \quad (3-35)$$

Definição 3.11 (DFT). *Seja $f \in \mathbb{C}^m$:*

$$f := (f_0, \dots, f_{m-1})^T \quad (3-36)$$

Segundo [7] e [3] a transformada discreta de Fourier em uma dimensão de f é um vetor $\hat{f} \in \mathbb{C}^m$:

$$\hat{f} := (\hat{f}_0, \dots, \hat{f}_{m-1})^T \quad (3-37)$$

definido por:

$$\hat{f}_l := \sum_{j=0}^{m-1} \omega^{jl} f_j \quad (3-38)$$

para $l \in [0; m-1]$.

Essa transformada é conhecida como *DFT* (*Discrete Fourier Transform*). Ela pode ser representada em notação matricial. A matriz de DFT de dimensão m é dada por:

$$\forall j, l \in [0; m-1], (\mathbb{F}_m)_{jl} = \omega^{jl} \quad (3-39)$$

ou seja:

$$\mathbb{F}_m := \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega^1 & \dots & \omega^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{m-1} & \dots & \omega^{(m-1)(m-1)} \end{pmatrix}. \quad (3-40)$$

Então

$$\hat{f} = \mathbb{F}_m f. \quad (3-41)$$

A inversa de \mathbb{F}_m é dada por

$$\mathbb{F}_m^{-1} = \frac{1}{m} \mathbb{F}_m^* \quad (3-42)$$

onde \mathbb{F}_m^* é a conjugada transposta de \mathbb{F} .

Definição 3.12 (diag). *Se c é um vetor de m coordenadas, então denotamos por $\text{diag}(c)$ a matriz diagonal $m \times m$ cuja diagonal principal é igual a c :*

$$\text{diag}(v_1, \dots, v_m) := \begin{pmatrix} v_1 & 0 & \dots & 0 \\ 0 & v_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & v_m \end{pmatrix} \quad (3-43)$$

Se M_p é uma matriz circulante $m \times m$, então seus autovetores são as colunas de \mathbb{F}_m . Portanto uma matriz circulante é diagonalizada por \mathbb{F}_m — sendo c a primeira coluna de M_p temos:

$$M_p = \mathbb{F}_m^{-1} \text{diag}(\mathbb{F}_m c) \mathbb{F}_m \quad (3-44)$$

[7]

Definição 3.13 (Produto de Hadamard). *Dados $p, q \in \mathbb{R}^m$, denotamos por $p \odot q$ o produto de Hadamard (Hadamard product) entre p e q :*

$$p \odot q := (p_1 q_1, \dots, p_m q_m)^T \quad (3-45)$$

Essa fatoração de M_p propicia multiplicação eficiente de M_p por vetor. Seja $q \in \mathbb{R}^m$ e seja M_p uma matriz circulante $m \times m$ cuja primeira coluna é c . Então:

$$M_p q = \mathbb{F}_m^{-1} \text{diag}(\mathbb{F}_m c) \mathbb{F}_m q = \mathbb{F}_m^{-1} (\mathbb{F}_m c \odot \mathbb{F}_m q) \quad (3-46)$$

No nosso caso as transformadas $\mathbb{F}_m c$ e $\mathbb{F}_m q$ podem ser pré-computadas e armazenadas. No laço mais interno, só precisamos calcular o produto de Hadamard (que custa tempo $O(m)$) e a IDFT.

3.1.3 Custo de memória

Devemos saber o custo de memória de armazenar a DFT de todos os $n + r$ vetores de $P \cup Q$. A DFT \hat{p} de um vetor real $p \in \mathbb{R}^m$ satisfaz a *simetria hermitiana* [3]:

$$\hat{p}_l = \overline{\hat{p}_{m-l}} \quad (3-47)$$

Portanto os valores $\{\hat{p}_{\frac{m}{2}+1}, \dots, \hat{p}_{m-1}\}$ são redundantes e não precisam ser computados nem armazenados, reduzindo o custo — tanto de memória quanto de processamento — praticamente pela metade. A biblioteca FFTW 3, usada no software CyclicKNN apresentado nesse trabalho, tira proveito dessa simetria. Assim, para cada vetor são armazenados $\frac{m}{2} + 1$ números complexos. O CyclicKNN armazena cada número complexo com parte real e imaginária em precisão simples (4 bytes para cada parte). Assim, o custo de memória é $8(\frac{m}{2} + 1)$ bytes por vetor. Aproximadamente, $4m$ bytes por vetor, ou $4m(n + r)$ bytes para todos os vetores em $P \cup Q$. Após a transformada DFT, as coordenadas reais dos vetores não são mais usadas. O CyclicKNN libera a memória das coordenadas reais de cada vetor imediatamente após calcular a DFT.

3.2 Eliminação rápida de vetores de treinamento desfavoráveis

Mesmo com a grande aceleração exposta na seção 3.1.2, a solução por busca exaustiva ainda demora tempo $O(nrm \log m + rk \log n)$. Em um Laptop com processador Intel Core i3 com clock máximo de 2.27GHz, o programa demorou 3.4s para processar 512 vetores de consulta e 4096 vetores de treinamento de dimensão 200. Como o tempo de execução é proporcional ao produto nr , então estimamos que para processar 100 vetores de consulta com 10^7 vetores de treinamento levaríamos 27 minutos. Por isso queremos uma solução que não precise calcular a distância especial (mesmo através da IDFT) para todos os vetores de treinamento. Isso é possível se eliminarmos vetores de treinamento desfavoráveis — que não podem estar entre os k mais próximos.

Em um espaço vetorial E munido de produto interno, o produto interno define uma norma, e essa norma define uma distância que satisfaz a propriedade:

$$\forall u, v \in E, d^2(u, v) = \|u\|^2 + \|v\|^2 - 2u \cdot v \quad (3-48)$$

conforme a observação 3.4. Isso vale em particular para \mathbb{R}^m com o produto interno canônico e a distância euclidiana. Como vimos anteriormente, as normas podem facilmente ser pré-computadas, e o único desafio restante para o cálculo da distância especial é o cálculo do produto interno máximo.

Se soubermos um limite superior para o produto interno máximo, então sabemos um limite inferior para a distância especial. Dados $q \in Q$ e $p \in P$, se o limite inferior de $d_e(p, q)$ for muito alto então sabemos que q não está entre os k vizinhos mais próximos de q . Procuramos então uma função

$$t: \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R} \quad (3-49)$$

tal que:

$$\forall p, q \in \mathbb{R}^m, \forall c \in [0; m-1], (C^c p) \cdot q \leq t(p, q) \quad (3-50)$$

Sabemos que

$$d^2(C^c p, q) = \|p\|^2 + \|q\|^2 - 2(C^c p) \cdot q \quad (3-51)$$

portanto:

$$d^2(C^c p, q) \geq \|p\|^2 + \|q\|^2 - 2t(p, q) \quad (3-52)$$

portanto:

$$d_e^2(p, q) \geq \|p\|^2 + \|q\|^2 - 2t(p, q) \quad (3-53)$$

Para que essa abordagem seja frutífera, t deve ser fácil de calcular. Calcular todos os m produtos internos da forma $(C^c p) \cdot q$ custa tempo $O(m \log m)$ com a FFT; só vale a pena calcular t se for significativamente mais rápido do que a FFT. Por outro lado, t precisa ser um limite superior “justo”, que elimine muitos vetores de treinamento. Não adianta definir que $t(p, q) = 10^{100} \|p\| \|q\|$.

Uma primeira ideia é a *desigualdade de Cauchy-Schwarz* [5]. Ela diz que para qualquer espaço vetorial E (real ou complexo) munido de produto interno e para quaisquer $p, q \in E$, vale:

$$|p \cdot q| \leq \|p\| \|q\| \quad (3-54)$$

A partir dela e algumas manipulações algébricas simples, concluímos que

$$d_e(p, q) \geq |||p| - |q||| \quad (3-55)$$

conforme o teorema B.1 do apêndice B.

Esse primeiro resultado pode ser útil. Como já observamos, as normas de todos os vetores podem ser pré-computadas e armazenadas, a custo baixo. Porém esse limite superior é “frouxo” e elimina poucos vetores de treinamento. Queremos eliminar muitos vetores de treinamento sem prejudicar a exatidão da resposta. Queremos um limite inferior mais justo para d_e . Uma tentativa é analisar mais cuidadosamente a DFT dos nossos vetores.

Seja

$$\mathbb{F}'_m := \frac{1}{\sqrt{m}} \mathbb{F}_m \quad (3-56)$$

\mathbb{F}'_m é uma matriz unitária [12] e portanto [5] preserva o produto interno hermitiano de um espaço vetorial complexo E :

$$\forall p, q \in E, \langle p, q \rangle = \langle \mathbb{F}'_m p, \mathbb{F}'_m q \rangle \quad (3-57)$$

Substituindo a equação (3-56):

$$\langle p, q \rangle = \langle \frac{1}{\sqrt{m}} \mathbb{F}_m p, \frac{1}{\sqrt{m}} \mathbb{F}_m q \rangle \quad (3-58)$$

portanto:

$$m \langle p, q \rangle = \langle \mathbb{F}_m p, \mathbb{F}_m q \rangle \quad (3-59)$$

Essa última fórmula vale em particular para vetores de \mathbb{R}^m , e em particular quando um dos vetores é $C^c p$:

$$\forall c \in \mathbb{Z}, \forall p, q \in \mathbb{R}^m, m(C^c p) \cdot q = m \langle C^c p, q \rangle = \langle \mathbb{F}_m C^c p, \mathbb{F}_m q \rangle \quad (3-60)$$

Se definirmos

$$\hat{p} := \mathbb{F}_m p \quad (3-61)$$

$$\hat{q} := \mathbb{F}_m q \quad (3-62)$$

$$p^c := C^c p \quad (3-63)$$

$$\hat{p}^c := \mathbb{F}_m p^c \quad (3-64)$$

então:

$$mp^c \cdot q = \langle \hat{p}^c, \hat{q} \rangle \quad (3-65)$$

Podemos expressar \hat{p}^c em termos de \hat{p} . Sejam p_0, \dots, p_{m-1} as coordenadas de p :

$$p := (p_0, \dots, p_{m-1}) \quad (3-66)$$

Então

$$\hat{p}_l^c = \omega^{-lc} \hat{p}_l \quad (3-67)$$

conforme o teorema B.2 do apêndice B.

Das equações (3-65), (3-67) e da definição de produto interno hermitiano segue que:

$$mp^c \cdot q = \sum_{l=0}^{m-1} \omega^{-lc} \hat{p}_l \overline{\hat{q}_l} \quad (3-68)$$

portanto:

$$p^c \cdot q = \frac{1}{m} \sum_{l=0}^{m-1} \omega^{-lc} \widehat{p}_l \overline{\widehat{q}_l} \quad (3-69)$$

Agora é fácil achar um limite superior para o produto interno $p^c \cdot q$. Para qualquer número real $x \in \mathbb{R}$ temos $x \leq |x|$, portanto

$$p^c \cdot q \leq |p^c \cdot q| \quad (3-70)$$

Por sua vez:

$$|p^c \cdot q| = \frac{1}{m} \left| \sum_{l=0}^{m-1} \omega^{-lc} \widehat{p}_l \overline{\widehat{q}_l} \right| \quad (3-71)$$

portanto

$$p^c \cdot q \leq \frac{1}{m} \left| \sum_{l=0}^{m-1} \omega^{-lc} \widehat{p}_l \overline{\widehat{q}_l} \right| \quad (3-72)$$

Para qualquer sequência de números complexos $(x_j)_{j \in J}$, o módulo da soma é menor ou igual à soma dos módulos:

$$\left| \sum_{j \in J} x_j \right| \leq \sum_{j \in J} |x_j| \quad (3-73)$$

Isso vale em particular para a sequência $(\omega^{-lc} \widehat{p}_l \overline{\widehat{q}_l})_{l \in [0; m-1]}$:

$$\left| \sum_{l=0}^{m-1} \omega^{-lc} \widehat{p}_l \overline{\widehat{q}_l} \right| \leq \sum_{l=0}^{m-1} \left| \omega^{-lc} \widehat{p}_l \overline{\widehat{q}_l} \right| = \sum_{l=0}^{m-1} \left| \omega^{-lc} \right| |\widehat{p}_l| |\widehat{q}_l| \quad (3-74)$$

como $|\omega| = 1$ então

$$\sum_{l=0}^{m-1} \left| \omega^{-lc} \right| |\widehat{p}_l| |\widehat{q}_l| = \sum_{l=0}^{m-1} |\widehat{p}_l| |\widehat{q}_l| \quad (3-75)$$

Portanto:

$$\left| \sum_{l=0}^{m-1} \omega^{-lc} \widehat{p}_l \overline{\widehat{q}_l} \right| \leq \sum_{l=0}^{m-1} |\widehat{p}_l| |\widehat{q}_l| \quad (3-76)$$

Combinando com a desigualdade (3-72) temos:

$$p^c \cdot q \leq \frac{1}{m} \sum_{l=0}^{m-1} |\widehat{p}_l| |\widehat{q}_l| \quad (3-77)$$

Conforme a desigualdade (3-53), isso significa que:

$$d_e^2(p, q) \geq \|p\|^2 + \|q\|^2 - \frac{2}{m} \sum_{l=0}^{m-1} |\widehat{p}_l| |\widehat{q}_l| \quad (3-78)$$

Assim chegamos ao algoritmo 3.2, que é uma melhora do algoritmo 2.2 (busca

exaustiva). Nesse algoritmo, a partir da sua inicialização o conjunto $P'_k(q)$ descreve os k vizinhos mais próximos dentre os vetores de treinamento analisados até então. Cada elemento de $P'_k(q)$ contém o índice do respectivo vetor, o índice da rotação que minimiza a distância, e o valor de $d_e^2(p_j, q)$. Para maior eficiência, $P'_k(q)$ é implementado como uma heap de máximo, permitindo inserção e remoção de um elemento em tempo $O(\log k)$, e leitura do máximo em tempo $O(1)$. Denotamos por d^{*2} o valor de $d_e^2(p, q)$ para o pior vizinho em $P'_k(q)$. Ou seja:

$$d^{*2} := \max_{p \in P'_k(q)} d_e^2(p, q) \quad (3-79)$$

Algoritmo 3.2: Eliminação de vetores de treinamento desfavoráveis

Entrada: Natural k , conjuntos $P, Q \subset \mathbb{R}^m$

Saída: Para cada $q \in Q$ retorna $P_k(q)$.

```

1  Pré-compute e armazene a norma e a DFT dos elementos de  $P$  e  $Q$ 
2  Compute e armazene os  $(\frac{m}{2} + 1)$  módulos não-redundantes de cada DFT
   computada na linha 1
3  para cada  $q \in Q$  faça
4      para cada  $j \in [1; k]$  faça
5          Compute  $d_e^2(p_j, q)$ 
6          Adicione a  $P'_k(q)$  os respectivos índices e as respectivas distâncias
7      fim
8      Transforme  $P'_k(q)$  em uma heap de máximo
9       $d^{*2} \leftarrow \max_{p \in P'_k(q)} d_e^2(p, q)$ 
10     para cada  $j \in [k + 1; n)$  faça
11         se  $(\|p_j\| - \|q\|)^2 < d^{*2}$  então
12             se  $(\|p\|^2 + \|q\|^2 - \frac{2}{m} \sum_{l=0}^{m-1} |\hat{p}_l| |\hat{q}_l|) < d^{*2}$  então
13                  $d'^2 \leftarrow d_e^2(p_j, q)$ 
14                 se  $d'^2 < d^{*2}$  então
15                     Insira  $(j, d'^2)$  em  $P'_k(q)$ 
16                     Remova de  $P'_k(q)$  o pior vizinho
17                      $d^{*2} \leftarrow \max_{p \in P'_k(q)} d_e^2(p, q)$ 
18                 fim
19             fim
20         fim
21     fim
22      $P_k(q) \leftarrow P'_k(q)$ 
23 fim
```

Observações:

- As normas são pré-computadas antes do laço da linha 3.
- As linhas 15-17 custam apenas tempo $O(\log k)$ pois $P'_k(q)$ é uma heap de máximo.
- A linha 12 é conceitual; o programa otimiza o cálculo sabendo que praticamente a metade dos módulos é redundante.

Podemos melhorar esse algoritmo. O laço da linha 4 inicializa $P'_k(q)$ com os k primeiros vetores de treinamento. Na linha 12, nós comparamos o limite inferior da distância especial atual com o pior vizinho de $P'_k(q)$. Se todos os elementos de $P'_k(q)$ forem horríveis, então a linha 12 não eliminará o vetor. No pior caso, P está ordenado em ordem decrescente de $d_e(p, q)$; nesse caso a linha 12 não eliminará um único vetor.

Então uma melhoria simples para o algoritmo 3.2 é tratar primeiro os vetores mais favoráveis. Para tal, uma tática simples é ser otimista: ordenar P pelo limite inferior de $d_e(p, q)$ (calculado conforme (3-78)) e supor que, provavelmente, os primeiros vetores de P (conforme a nova ordenação) serão de fato bons vizinhos. Isso em nada altera a resposta; apenas muda o tempo de execução do programa (presumivelmente para melhor).

Nós não queremos, porém, modificar P para cada vetor de consulta. Para obter bom desempenho, nós queremos paralelizar o programa, e a maneira mais natural de fazê-lo é designar threads diferentes para analisar vetores de consulta diferentes. Para minimizar o custo de memória, queremos que todas as threads compartilhem o mesmo conjunto de treinamento P . Então ao invés de modificar P , criamos registros com dois números

1. O índice j do vetor de treinamento correspondente;
2. O limite inferior de $d_e^2(p_j, q)$ (conforme a equação (3-78)).

Criamos um vetor com n tais registros (um para cada vetor de P), e ordenamos esse vetor antes do laço da linha 3.

3.2.1 Custo de memória

Para que essa última desigualdade seja proveitosa, queremos calcular o lado direito muito rapidamente. Calcular o módulo de um número complexo é significativamente mais custoso do que fazer uma multiplicação real. No nosso programa, se calculássemos os módulos no laço mais interno, sem otimização, teríamos que calcular o módulo de nrm números complexos. Para reduzir esse custo, podemos pré-computar os valores

$$(|\hat{p}_0|, \dots, |\hat{p}_{m-1}|) \quad (3-80)$$

para cada vetor em $P \cup Q$. Aparentemente isso consumiria memória demais. Porém, a DFT \hat{p} de um vetor real $p \in \mathbb{R}^m$ satisfaz a simetria hermitiana:

$$\hat{p}_l = \overline{\hat{p}_{m-l}} \quad (3-81)$$

portanto:

$$|\hat{p}_l| = |\hat{p}_{m-l}| \quad (3-82)$$

Portanto os valores $\{|\hat{p}_{\frac{m}{2}+1}|, \dots, |\hat{p}_{m-1}|\}$ são redundantes e não precisam ser computados nem armazenados. Assim, precisamos armazenar $\frac{m}{2} + 1$ valores reais para cada vetor em $P \cup Q$. Como o CyclicKNN usa precisão simples (4 bytes por número real), o custo de memória em questão é $4(\frac{m}{2} + 1)$ bytes por vetor, ou aproximadamente $2m$ bytes por vetor. Como são $n + r$ vetores, o custo de armazenar os módulos em questão é $2m(n + r)$. Como vimos em 3.1.3, o custo de memória para armazenar a DFT de todos os vetores de $P \cup Q$ é aproximadamente $4m(n + r)$. Assim, o custo total de memória do CyclicKNN em bytes é aproximadamente

$$4m(n + r) + 2m(n + r) = 6m(n + r) \quad (3-83)$$

Para $m = 200$, $n = 10^5$ e $r = 100$, isso é aproximadamente 115MB.

Implementação e experimentação

Este capítulo descreve uma implementação paralela em C++11 da solução apresentada no capítulo 3 e apresenta o desempenho e o *speedup* empíricos de tal programa, chamado CyclicKNN.

4.1 Implementação

Para implementar a FFT e sua inversa usamos a biblioteca FFTW 3.3.3. A arquitetura da FFTW é descrita no artigo [3], publicado no IEEE e citado por mais de 800 artigos. A FFTW se adapta ao hardware para acelerar o desempenho. Tira proveito inclusive de instruções SIMD, como SSE e até AVX. É código aberto, muito rápida, muito flexível e largamente utilizada. Seu desempenho é competitivo até com programas otimizados por fabricantes de hardware, mas com a vantagem de rodar bem em várias máquinas diferentes. A biblioteca decompõe recursivamente um problema em subproblemas mais simples do mesmo tipo. Problemas suficientemente simples são resolvidos por código *straight-line* otimizado.

Para maior desempenho, o CyclicKNN usa ponto flutuante de precisão simples. Foi desenvolvido em C++11 com o compilador GCC 4.8. Foi desenvolvido em Linux (Ubuntu 13.10) mas não depende de funcionalidades específicas de Linux.

4.1.1 Computação paralela

Para reduzir o tempo de execução, implementamos paralelismo (usando a biblioteca de threads do C++11) particionando o conjunto de consulta. Cada partição exceto a última tem o mesmo número de vetores de consulta (a última partição pode ser menor, se o número de vetores de consulta não for múltiplo do tamanho de partição desejado). Cada partição é processada em sua própria thread.

Em computação paralela, um problema difícil e trabalhoso é a comunicação e sincronização entre tarefas, que deve ser correta e eficiente. No nosso problema com a tática de paralelização descrita acima nós não enfrentamos esse problema potencial.

Base de dados	Vetores de consulta	Vetores de teste	Dimensão (m)
Pendigits	3498	7494	16
Optdigits	1797	3823	64
Mnist	10000	60000	784

Tabela 4.1: Bases de dados para medida de desempenho

Threads diferentes acessam o mesmo conjunto de treinamento, mas esse acesso é somente leitura e facilmente gerenciado. As diferentes threads escrevem sua saída em posições de memória diferentes, o que também é fácil de gerenciar. A única “sincronização” foi a thread mestre esperar o término de todas as outras, para ter certeza de que o resultado está pronto. Isso é trivial com a biblioteca de threads C++11.

Em conclusão, a paralelização do nosso programa foi fácil e resultou em um *speedup* excelente. A limitação dessa tática é que não podemos usar mais threads do que o número de vetores de consulta.

A implementação de paralelismo teve boa relação benefício x custo — o benefício foi uma excelente aceleração, e o custo foi um moderado tempo de desenvolvimento. Por esse motivo, nós implementamos paralelismo antes mesmo de ampliar os esforços na busca de uma estrutura de dados otimizada para o problema abordado — que fica como trabalho futuro. Além disso, deseja-se combinar a aceleração advinda da estrutura de dados com a aceleração do paralelismo; para comparar tal solução com um algoritmo mais simples, desprovido de estrutura de dados especializada — para determinar se tal estrutura é proveitosa — é necessário que esse algoritmo mais simples seja paralelo, para a comparação ser justa.

4.2 Experimentação

Para medir o desempenho, nós compilamos e executamos o programa em um computador com as seguintes características:

- Dois processadores Intel Xeon E5506 @ 2.13GHz, cada um com 4 núcleos
- Sistema operacional Linux, openSUSE 12.3 (rodando em 64 bits)
- Compilador g++ 4.7.2
- Biblioteca FFTW 3.3.3

Nós processamos três bases de dados, descritas na tabela 4.1.

A base MNIST está disponível em <http://yann.lecun.com/exdb/mnist/> e as outras duas estão disponíveis no *UCI Machine Learning Repository* (<http://archive.ics.uci.edu/ml/>).

O processamento da base MNIST com o algoritmo força bruta é muito lento; por isso, o processamento com o força bruta foi limitado aos primeiros 3600 vetores de

k	Número de threads	Algoritmo
<ul style="list-style-type: none"> • 1 • 7 • 50 	<ul style="list-style-type: none"> • 1 • 2 • 4 • 8 	<ul style="list-style-type: none"> • Força bruta • Acelerado com FFT • Acelerado com FFT e exclusão de vetores de treinamento

Tabela 4.2: *Parâmetros de medida de desempenho para cada base de dados*

Número de threads	Tempo de execução	Speedup
1	1293s	-
2	798s	1.6
4	462s	2.8
8	365s	3.5

Tabela 4.3: *Speedup de paralelismo para a base Mnist, $k = 7$, com o algoritmo acelerado com FFT e exclusão de vetores de treinamento*

Número de threads	Tempo de execução	Speedup
1	2319s	-
2	1286s	1.8
4	709s	3.3
8	500s	4.6

Tabela 4.4: *Speedup de paralelismo para a base Mnist, $k = 50$, com o algoritmo acelerado com FFT e exclusão de vetores de treinamento*

consulta. Mesmo com esse limite, o força bruta com uma thread levou 20h para cada valor de k .

O programa foi medido com todas as combinações dos parâmetros descritos na tabela 4.2 (mas apenas alguns dos resultados foram expostos nas tabelas abaixo).

Os resultados estão nas tabelas abaixo:

Número de threads	Tempo de execução	<i>Speedup</i>
1	2.3s	-
2	1.2s	2
4	0.62s	3.6
8	0.33s	7.1

Tabela 4.5: *Speedup de paralelismo para a base Optdigits, $k = 7$, com o algoritmo acelerado com FFT e exclusão de vetores de treinamento*

Número de threads	Tempo de execução	<i>Speedup</i>
1	3.7s	-
2	1.9s	2
4	1s	3.8
8	0.51s	7.3

Tabela 4.6: *Speedup de paralelismo para a base Optdigits, $k = 50$, com o algoritmo acelerado com FFT e exclusão de vetores de treinamento*

Força bruta sequencial	Força bruta (8 threads)	FFT (8 threads)	FFT e exclusão (8 threads)
71340s	8967s	252s	179s

Tabela 4.7: *Tempo de execução de cada técnica processando a base Mnist (limitada aos 3600 primeiros vetores de consulta), $k = 7$*

Tentativa de aceleração com interpretação geométrica

Na seção 3.1.1 o produto $M_p q$ (usado para calcular a distância especial) calcula m produtos internos, e só aproveita o maior deles. Isso sugere que há computação desnecessária e oportunidade para uma aceleração maior.

Esse capítulo apresenta uma outra abordagem para o cálculo eficiente da distância especial: interpretação geométrica do operador C . Vemos que para $m \leq 3$ podemos resolver o problema em tempo $O(m)$ e tentamos estender esse resultado para o caso geral. Até o momento não conseguimos tirar proveito dessa abordagem, e não conseguimos provar a conjectura 5.2.

5.1 Casos particulares $m = 1$, $m = 2$ e $m = 3$

Para $m \leq 3$, $d_e(p, q)$ pode ser calculada em tempo $O(m)$, assumindo (para simplificar) que o computador faz aritmética em \mathbb{R} em tempo $O(1)$ por operação.

Para $m = 1$ o problema é trivial.

Para $m = 2$, \mathbb{R}^2 está dividido em dois semiplanos — acima e abaixo da reta $y = x$. Dado $p \in \mathbb{R}^2$ a operação C consiste em refletir p em torno dessa reta. O conjunto $C^*(p)$ tem exatamente um ponto acima dessa reta e um ponto abaixo dela. Dado $q \in \mathbb{R}^2$, o ponto de $C^*(p)$ mais próximo de q é aquele que estiver no mesmo semiplano (provado no teorema C.1, apêndice C). Podemos computar o semiplano de cada p com a comparação $p_2 > p_1$ e podemos computar o semiplano de cada q com a comparação $q_2 > q_1$. Assim $d_e(p, q)$ pode ser calculada em tempo $O(m)$.

Para $m = 3$, a operação C é uma rotação de $\frac{-2\pi}{3}$ rad em torno do eixo $(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$. Para verificá-lo basta substituir os valores adequados na matriz de rotação em torno de um eixo em \mathbb{R}^3 , como foi feito na seção C.1 do apêndice C.

Definição 5.1 (Base ordenada). *Dado um espaço vetorial de dimensão finita E , uma base ordenada de E é uma sequência finita de vetores de E que é linearmente independente e gera E . [5]*

Podemos mudar a base ordenada do espaço \mathbb{R}^3 para que novo eixo z coincida com a reta $x = y = z$. Isso pode ser feito com uma rotação de 0.96 rad em torno do eixo dado por $(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0)$ ¹. Essa mudança de base pode ser feita antes do laço principal e não custaria muito tempo de processamento². Nessa base ordenada, a transformação C é uma rotação de $\frac{-2\pi}{3}$ rad em torno do eixo z . Essa rotação não muda a coordenada z de um vetor. Portanto temos

$$p = (p_x, p_y, p_z) \quad (5-1)$$

$$Cp = (p'_x, p'_y, p_z) \quad (5-2)$$

$$C^2p = (p''_x, p''_y, p_z) \quad (5-3)$$

Logo

$$d_0 := d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2} \quad (5-4)$$

$$d_1 := d(Cp, q) = \sqrt{(p'_x - q_x)^2 + (p'_y - q_y)^2 + (p_z - q_z)^2} \quad (5-5)$$

$$d_2 := d(C^2p, q) = \sqrt{(p''_x - q_x)^2 + (p''_y - q_y)^2 + (p_z - q_z)^2} \quad (5-6)$$

Queremos o mínimo dentre d_0 , d_1 e d_2 . Conforme a observação 3.5 podemos comparar os respectivos quadrados das distâncias pois o mínimo dos quadrados corresponde ao quadrado do mínimo.

$$d_0^2 = (p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2 \quad (5-7)$$

$$d_1^2 = (p'_x - q_x)^2 + (p'_y - q_y)^2 + (p_z - q_z)^2 \quad (5-8)$$

$$d_2^2 = (p''_x - q_x)^2 + (p''_y - q_y)^2 + (p_z - q_z)^2 \quad (5-9)$$

O termo $(p_z - q_z)^2$ é o mesmo em todas as três equações acima. Podemos passá-lo para o lado esquerdo da respectiva equação. Assim, para $c \in \{0, 1, 2\}$ seja

$$D_c := \sqrt{d_c^2 - (p_z - q_z)^2} \quad (5-10)$$

¹0.96 rad é $\arccos\left(\frac{1}{\sqrt{3}}\right)$. $\frac{1}{\sqrt{3}}$ é o cosseno do ângulo entre a reta $x = y = z$ e o eixo z , como pode ser calculado pelo produto interno $(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}) \cdot (0, 0, 1) = \frac{1}{\sqrt{3}}$. O eixo $(\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0)$ está contido no plano $z = 0$ e é perpendicular à reta $x = y = z$.

²A matriz de mudança de base seria $m \times m$. A mudança de base seria feita multiplicando essa matriz por uma outra matriz cujas colunas são os vetores de treinamento e consulta. O custo de multiplicar uma matriz $m \times m$ por uma matriz $m \times (n + r)$ é $O(m^2(n + r))$, pequeno em comparação com $O(nrm^2)$. Além disso a multiplicação de matrizes é uma operação básica e altamente otimizada, inclusive em computação paralela.

portanto

$$D_c^2 = d_c^2 - (p_z - q_z)^2 \quad (5-11)$$

Assim:

$$D_0^2 = d_0^2 - (p_z - q_z)^2 = (p_x - q_x)^2 + (p_y - q_y)^2 \quad (5-12)$$

$$D_1^2 = d_1^2 - (p_z - q_z)^2 = (p'_x - q_x)^2 + (p'_y - q_y)^2 \quad (5-13)$$

$$D_2^2 = d_2^2 - (p_z - q_z)^2 = (p''_x - q_x)^2 + (p''_y - q_y)^2 \quad (5-14)$$

Queremos o mínimo dentre D_0, D_1, D_2 .

Seja $F: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ a função tal que, dado $u \in \mathbb{R}^3$:

$$F(u) := (u_x, u_y) \quad (5-15)$$

Intuitivamente, $F(u)$ é a projeção ortogonal de u no plano $z = 0$. Temos então:

$$D_0 = d(F(p), F(q)) \quad (5-16)$$

$$D_1 = d(F(Cp), F(q)) \quad (5-17)$$

$$D_2 = d(F(C^2p), F(q)) \quad (5-18)$$

Em coordenadas polares:

$$F(q) = (r_q, \theta_q) \quad (5-19)$$

$$F(p) = (r_p, \theta_p) \quad (5-20)$$

$$F(Cp) = (r_p, \theta_p - \frac{2\pi}{3}) \quad (5-21)$$

$$F(C^2p) = (r_p, \theta_p - \frac{4\pi}{3}) \quad (5-22)$$

Ou seja,

$$\forall c \in \{0, 1, 2\}, F(C^c p) = (r_p, \theta_p - \frac{2\pi}{3}c) \quad (5-23)$$

Os pontos $F(C^c p)$ têm todos a mesma norma, portanto pelo teorema 3.6 o mínimo da distância corresponde ao máximo do produto interno. Temos

$$F(C^c p) \cdot F(q) =$$

$$r_p r_q \cos(\theta_q - (\theta_p - \frac{2\pi}{3}c)) = r_p r_q \cos(\theta_q - \theta_p + \frac{2\pi}{3}c) \quad (5-24)$$

Portanto queremos achar $c_{max} \in \{0, 1, 2\}$ que maximiza

$$r_p r_q \cos(\theta_q - \theta_p + \frac{2\pi}{3}c) \quad (5-25)$$

Precisamente:

$$c_{max} = \arg \max_{c \in \{0;1;2\}} (r_p r_q \cos(\theta_q - \theta_p + \frac{2\pi}{3}c)) \quad (5-26)$$

Se houver mais de um c_{max} que satisfaz a equação (5-26) então escolhemos arbitrariamente qualquer um deles.

Lembre que, pelo teorema 3.6, $d_e(p, q) = d(C^{c_{max}}p, q)$.

Seja

$$a: \mathbb{R} \rightarrow \mathbb{Z} \quad (5-27)$$

a função que arredonda um número real para o inteiro mais próximo (e se a parte fracionária do número for 0.5, arredonda para o inteiro *par* mais próximo). Por exemplo:

$$a(-1.5) = -2 \quad (5-28)$$

$$a(0.5) = 0 \quad (5-29)$$

$$a(0.6) = 1 \quad (5-30)$$

$$a(1.5) = 2 \quad (5-31)$$

Seja $R_{2\pi}: \mathbb{R} \rightarrow [-\pi; \pi]$ a função definida por

$$R_{2\pi}(\theta) = \theta - 2\pi a(\frac{\theta}{2\pi}) \quad (5-32)$$

Note que $R_{2\pi}$ é uma função periódica, de período 2π . Note também que $\cos(R_{2\pi}(\theta)) = \cos \theta$.

Particionemos o intervalo $[-\pi; \pi]$ em quatro conjuntos:

$$R_1 :=]-\frac{\pi}{3}; \frac{\pi}{3}[\quad (5-33)$$

$$R_2 :=]\frac{\pi}{3}; \pi[\quad (5-34)$$

$$R_3 :=]-\pi; -\frac{\pi}{3}[\quad (5-35)$$

$$R_4 := \{-\pi; -\frac{\pi}{3}; \frac{\pi}{3}; \pi\} \quad (5-36)$$

Essa partição é ilustrada pela figura 5.1

Vamos supor por enquanto que

$$R_{2\pi}(\theta_q - \theta_p) \in R_1 \cup R_2 \cup R_3 \quad (5-41)$$

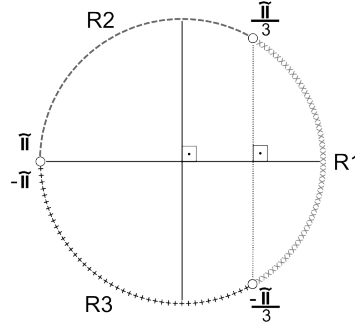


Figura 5.1: Círculo unitário $[-\pi; \pi]$ em quatro partições:

$$R_1 := \left] -\frac{\pi}{3}; \frac{\pi}{3} \right[\quad (5-37)$$

$$R_2 := \left] \frac{\pi}{3}; \pi \right[\quad (5-38)$$

$$R_3 := \left] -\pi; -\frac{\pi}{3} \right[\quad (5-39)$$

$$R_4 := \{-\pi; -\frac{\pi}{3}; \frac{\pi}{3}; \pi\} \quad (5-40)$$

Depois verificaremos que o resultado obtido também vale se $R_{2\pi}(\theta_q - \theta_p) \in R_4$.

Note que R_1 , R_2 e R_3 têm, cada um, comprimento $\frac{2\pi}{3}$. Note também que R_1 é a região de maior cosseno dentre R_1, R_2, R_3 :

$$\theta_1 \in R_1, \theta_2 \in R_2, \theta_3 \in R_3 \implies \cos \theta_1 = \max_{\theta \in \{\theta_1, \theta_2, \theta_3\}} \cos \theta \quad (5-42)$$

Retomemos a equação (5-26). Se $R_{2\pi}(\theta_q - \theta_p) \in R_1$ então

$$R_{2\pi}(\theta_q - \theta_p + \frac{2\pi}{3}) \in R_2 \quad (5-43)$$

$$R_{2\pi}(\theta_q - \theta_p + \frac{4\pi}{3}) \in R_3 \quad (5-44)$$

portanto

$$R_{2\pi}(\theta_q - \theta_p) \in R_1 \implies \cos(\theta_q - \theta_p) = \max_{c \in \{0,1,2\}} \cos(\theta_q - \theta_p + \frac{2\pi}{3}c) \implies c_{max} = 0 \quad (5-45)$$

Se $R_{2\pi}(\theta_q - \theta_p) \in R_2$ então

$$R_{2\pi}(\theta_q - \theta_p + \frac{2\pi}{3}) \in R_3 \quad (5-46)$$

$$R_{2\pi}(\theta_q - \theta_p + \frac{4\pi}{3}) \in R_1 \quad (5-47)$$

portanto

$$R_{2\pi}(\theta_q - \theta_p) \in R_2 \implies \cos(\theta_q - \theta_p + \frac{4\pi}{3}) = \max_{c \in \{0,1,2\}} \cos(\theta_q - \theta_p + \frac{2\pi}{3}c) \implies c_{max} = 2 \quad (5-48)$$

Se $R_{2\pi}(\theta_q - \theta_p) \in R_3$ então

$$R_{2\pi}(\theta_q - \theta_p + \frac{2\pi}{3}) \in R_1 \quad (5-49)$$

$$R_{2\pi}(\theta_q - \theta_p + \frac{4\pi}{3}) \in R_2 \quad (5-50)$$

portanto

$$R_{2\pi}(\theta_q - \theta_p) \in R_3 \implies \cos(\theta_q - \theta_p + \frac{2\pi}{3}) = \max_{c \in \{0,1,2\}} \cos(\theta_q - \theta_p + \frac{2\pi}{3}c) \implies c_{max} = 1 \quad (5-51)$$

Podemos determinar em tempo $O(1)$ se $R_{2\pi}(\theta_q - \theta_p) \in R_1$ ou $R_{2\pi}(\theta_q - \theta_p) \in R_2$ ou $R_{2\pi}(\theta_q - \theta_p) \in R_3$:

$$R_{2\pi}(\theta_q - \theta_p) \in R_1 \iff \frac{3}{2\pi}R_{2\pi}(\theta_q - \theta_p) \in]-0.5; 0.5[\iff a\left(\frac{3}{2\pi}R_{2\pi}(\theta_q - \theta_p)\right) = 0 \quad (5-52)$$

$$R_{2\pi}(\theta_q - \theta_p) \in R_2 \iff \frac{3}{2\pi}R_{2\pi}(\theta_q - \theta_p) \in]0.5; 1.5[\iff a\left(\frac{3}{2\pi}R_{2\pi}(\theta_q - \theta_p)\right) = 1 \quad (5-53)$$

$$R_{2\pi}(\theta_q - \theta_p) \in R_3 \iff \frac{3}{2\pi}R_{2\pi}(\theta_q - \theta_p) \in]-1.5; -0.5[\iff a\left(\frac{3}{2\pi}R_{2\pi}(\theta_q - \theta_p)\right) = -1 \quad (5-54)$$

Das equações (5-45) e (5-52) concluímos que

$$a\left(\frac{3}{2\pi}R_{2\pi}(\theta_q - \theta_p)\right) = 0 \implies c_{max} = 0 \quad (5-55)$$

Analogamente, das equações (5-48) e (5-53) concluímos que

$$a\left(\frac{3}{2\pi}R_{2\pi}(\theta_q - \theta_p)\right) = 1 \implies c_{max} = 2 \quad (5-56)$$

E das equações (5-51) e (5-54) concluímos que

$$a\left(\frac{3}{2\pi}R_{2\pi}(\theta_q - \theta_p)\right) = -1 \implies c_{max} = 1 \quad (5-57)$$

Portanto

$$c_{max} = \left(3 - a\left(\frac{3}{2\pi}R_{2\pi}(\theta_q - \theta_p)\right)\right) \mod 3 \quad (5-58)$$

Até aqui supomos que $R_{2\pi}(\theta_q - \theta_p) \in R_1 \cup R_2 \cup R_3$. A fórmula (5-58) satisfaz a equação (5-26) também quando $R_{2\pi}(\theta_q - \theta_p) \in R_4$. Para prová-lo basta verificar cada caso, como foi feito na seção C.2 do apêndice C.

Portanto c_{max} pode ser calculado em $O(m)$. Como $d_e(p, q) = d(C^{c_{max}}(p), q)$, e $d(C^{c_{max}}(p), q)$ pode ser calculado em $O(m)$, então podemos calcular $d_e(p, q)$ em tempo $O(m)$ como queríamos demonstrar.

5.2 Conjectura: $d_e(p, q)$ pode ser calculado em tempo $O(m)$

Nesta seção apresentamos duas conjecturas que, se verdadeiras, podem levar a uma aceleração maior da solução do problema abordado neste trabalho. Até o momento não conseguimos provar essas conjecturas.

Conjectura 5.2. Dados $P, Q \subset \mathbb{R}^m$, sejam:

$$n := |P| \tag{5-59}$$

$$r := |Q| \tag{5-60}$$

Após um pré-processamento de complexidade $O((n+r)m^2)$, para cada $p \in P$ e $q \in Q$, $d_e(p, q)$ pode ser calculado em tempo $O(m)$.

Já provamos isso para $m \leq 3$. Para m geral a veracidade dessa conjectura é sugerida pela conjectura 5.3 em combinação com o raciocínio empregado para o caso $m = 3$.

Conjectura 5.3. Em \mathbb{R}^m , a transformação C é uma espécie de “rotação” de $-\frac{2\pi}{m}$ rad em torno da reta $x = y = z = \dots$

Essa conjectura é verdade para $m \leq 3$. Para m geral, sabemos que:

- C é um operador linear bijetivo $\mathbb{R}^m \rightarrow \mathbb{R}^m$ (teorema 5.4).
- $C^m p = p$ (prova trivial).
- C preserva distâncias entre pontos (teorema 5.5).
- Os pontos de $C^*(p)$ têm todos a mesma norma (corolário 5.7).
- Seja u a reta $p_1 = p_2 = \dots = p_m$. Então

$$\forall p \in \mathbb{R}^m, Cp = p \iff p \in u \tag{5-61}$$

Ou seja, u é o conjunto de todos os pontos fixos de C (prova trivial).

- Os pontos de $C^*(p)$ são equidistantes da reta $p_1 = p_2 = \dots = p_m$ (teorema 5.9).
- C preserva o produto interno (teorema 5.10)

- C preserva ângulos (corolário 5.11)
- Dado um inteiro c tal que $0 \leq c < m$, o ângulo entre $C^c p$ e $C^{c+1} p$ é igual ao ângulo entre p e Cp (segue imediatamente do corolário 5.11 por indução finita).
- Dado $p \in \mathbb{R}^m$, todos os pontos de $C^*(p)$ pertencem a um hiperplano de dimensão $m - 1$ que é perpendicular à reta $p_1 = p_2 = \dots = p_m$ (teorema 5.12).

Teorema 5.4. *A transformação C é um operador linear bijetivo $\mathbb{R}^m \rightarrow \mathbb{R}^m$.*

Prova. Dado $p \in \mathbb{R}^m$, se interpretarmos p como um vetor coluna, temos $Cp = Mp$, onde M é a matriz $m \times m$ dada por:

$$M = \begin{pmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ 1 & 0 & \dots & 0 \end{pmatrix} \quad (5-62)$$

A primeira coluna de M tem 1 na última linha, e para $2 \leq j \leq m$ a j -ésima coluna de M tem 1 na $(j - 1)$ -ésima linha. Todas as outras entradas são 0.

Como C é uma multiplicação de matriz por vetor, segue imediatamente que C é um operador linear. E como os vetores-coluna da matriz M são linearmente independentes segue que C é bijetiva. \square

Teorema 5.5. *Se $u, v \in \mathbb{R}^m$ então $d(Cu, Cv) = d(u, v)$.*

Prova. Temos

$$d(u, v) = \sqrt{\sum_{j=1}^m (v_j - u_j)^2} \quad (5-63)$$

Temos também

$$Cu = (u_2, \dots, u_m, u_1) \quad (5-64)$$

$$Cv = (v_2, \dots, v_m, v_1) \quad (5-65)$$

$$\begin{aligned} d(Cu, Cv) &= \sqrt{(v_2 - u_2)^2 + \dots + (v_m - u_m)^2 + (v_1 - u_1)^2} = \\ &= \sqrt{\left(\sum_{j=2}^m (v_j - u_j)^2\right) + (v_1 - u_1)^2} = \\ &= \sqrt{\sum_{j=1}^m (v_j - u_j)^2} = d(u, v) \end{aligned} \quad (5-66)$$

\square

Corolário 5.6. $\forall p \in \mathbb{R}^m, \|Cp\| = p$.

Corolário 5.7. Dado $p \in \mathbb{R}^m$, todos os pontos de $C^*(p)$ têm a mesma norma.

Prova. Segue trivialmente do corolário 5.6 por indução finita. \square

Observação 5.8. Como $C: \mathbb{R}^m \rightarrow \mathbb{R}^m$ é uma função bijetiva (conforme o teorema 5.4) e preserva distâncias (teorema 5.5) então C é uma isometria [8].

Teorema 5.9. Dado $p \in \mathbb{R}^m$, os pontos de $C^*(p)$ são equidistantes da reta $p_1 = \dots = p_m$.

Prova. Seja u a reta $p_1 = \dots = p_m$. Ou seja, u é o conjunto de todos os pontos de \mathbb{R}^m que têm a forma (t, \dots, t) para $t \in \mathbb{R}$. Note que $v \in u \implies Cv = v$. Dado $o \in C^*(p)$ temos, pela definição de distância entre ponto e reta,

$$d(o, u) = \min_{v \in u} d(o, v) \quad (5-67)$$

Temos

$$\begin{aligned} d(Co, u) &= \\ \min_{v \in u} d(Co, v) &= \min_{v \in u} d(Co, Cv) = \min_{v \in u} d(o, v) = \\ &= d(o, u) \end{aligned} \quad (5-68)$$

(a penúltima igualdade segue do fato de que C preserva distâncias entre pontos). Provamos que $o \in C^*(p) \implies d(Co, u) = d(o, u)$; por indução finita segue trivialmente que todos os pontos de $C^*(p)$ são equidistantes de u . \square

Teorema 5.10. Para quaisquer $u, v \in \mathbb{R}^m$, $(Cu) \cdot (Cv) = u \cdot v$.

Prova.

$$u = (u_1, u_2, \dots, u_m) \quad (5-69)$$

$$Cu = (u_2, u_3, \dots, u_m, u_1) \quad (5-70)$$

$$v = (v_1, v_2, \dots, v_m) \quad (5-71)$$

$$Cv = (v_2, v_3, \dots, v_m, v_1) \quad (5-72)$$

$$u \cdot v = u_1 v_1 + u_2 v_2 + \dots + u_m v_m = \quad (5-73)$$

$$u_2 v_2 + \dots + u_m v_m + u_1 v_1 = \quad (5-74)$$

$$(Cu) \cdot (Cv) \quad (5-75)$$

□

Corolário 5.11. Para quaisquer $u, v \in \mathbb{R}^m$ o ângulo entre u e v é igual ao ângulo entre Cu e Cv .

Prova. Seja θ_{uv} o ângulo entre u e v . Temos

$$u \cdot v = \|u\| \|v\| \cos \theta_{uv} \quad (5-76)$$

Portanto

$$\theta_{uv} = \arccos \left(\frac{u \cdot v}{\|u\| \|v\|} \right) \quad (5-77)$$

onde $\arccos \theta$ é a função de $[-1; 1]$ em $[0; \pi]$ tal que

$$\forall \theta \in [0; \pi], \arccos(\cos \theta) = \theta \quad (5-78)$$

Analogamente

$$\theta'_{uv} = \arccos \left(\frac{Cu \cdot Cv}{\|Cu\| \|Cv\|} \right) \quad (5-79)$$

onde θ'_{uv} é o ângulo entre Cu e Cv . Pelo corolário 5.6 temos

$$\|Cu\| = \|u\| \quad (5-80)$$

$$\|Cv\| = \|v\| \quad (5-81)$$

portanto

$$\begin{aligned} \theta'_{uv} &= \arccos \left(\frac{Cu \cdot Cv}{\|u\| \|v\|} \right) = \\ &= \arccos \left(\frac{u \cdot v}{\|u\| \|v\|} \right) = \theta_{uv} \end{aligned} \quad (5-82)$$

□

Teorema 5.12. Dado $p \in \mathbb{R}^m$ todos os pontos de $C^*(p)$ pertencem a um hiperplano de dimensão $m - 1$ que é perpendicular à reta $p_1 = p_2 = \dots = p_m$.

Prova. Seja $U: \mathbb{R}^m \rightarrow \mathbb{R}$ o operador linear dado por

$$U((p_1, \dots, p_m)) := p_1 + \dots + p_m \quad (5-83)$$

Todos os pontos de $C^*(p)$ têm o mesmo conjunto de coordenadas; portanto

$$\forall v \in C^*(p), U(v) = U(p) \quad (5-84)$$

pois a adição é comutativa e associativa. Dado $p \in \mathbb{R}^m$ seja

$$T_p := \{v \in \mathbb{R}^m \mid U(v) = U(p)\} \quad (5-85)$$

De (5-84) e (5-85) segue que $C^*(p) \subseteq T_p$. Provemos que T_p é uma variedade afim de dimensão $m - 1$ e perpendicular à reta $p_1 = \dots = p_m$. Seja O a origem de \mathbb{R}^m . Note que $U(O) = 0$. Seja

$$T_O := \{v \in \mathbb{R}^m \mid U(v) = 0\} \quad (5-86)$$

Da definição do operador U segue trivialmente que

$$\forall s \in T_O, s \cdot \left(\frac{1}{\sqrt{m}}, \dots, \frac{1}{\sqrt{m}}\right) = 0 \quad (5-87)$$

Portanto T_O é perpendicular à reta $p_1 = \dots = p_m$. Temos

$$v \in T_p \iff U(v) = U(p) \iff U(v) - U(p) = 0 \iff U(v - p) = 0 \quad (5-88)$$

portanto

$$v \in T_p \iff v - p \in T_O \quad (5-89)$$

portanto

$$v \in T_p \iff \exists w \in T_O \mid v = p + w \quad (5-90)$$

Portanto T_p é a translação de T_O por p . Como T_O é um espaço vetorial de dimensão $m - 1$ (pelo teorema do núcleo e da imagem [9]) então T_p é uma variedade afim de dimensão $m - 1$. □

5.3 Desenvolvimento matemático

Nessa seção apresentamos uma tentativa de provar as conjecturas 5.3 e 5.2.

5.3.1 Matriz de C na forma canônica de Jordan

Como vimos na prova do teorema 5.4, o operador $C: \mathbb{R}^m \rightarrow \mathbb{R}^m$ corresponde à matriz

$$M = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ 1 & 0 & \cdots & 0 \end{pmatrix} \quad (5-91)$$

A primeira coluna de M tem 1 na última linha, e para $2 \leq j \leq m$ a j -ésima coluna de M tem 1 na $(j-1)$ -ésima linha. Todas as outras entradas são 0.

Para descrevê-la como uma rotação m -dimensional (na verdade uma sequência de várias rotações bidimensionais, mais uma reflexão unidimensional se m é par) é útil colocá-la na forma canônica de Jordan. A matriz na forma canônica de Jordan é uma matriz muito simples dentre as várias matrizes (referentes a diferentes bases ordenadas) para um dado operador linear. Ela é útil no estudo de questões que envolvem potências sucessivas de um operador [9] como é o nosso caso.

Para calcular a forma canônica de Jordan do operador C , comecemos calculando os seus autovalores. Para calcular seus autovalores, calculemos o seu polinômio característico $p(\lambda) := \det(M - \lambda I)$, onde I é a matriz identidade. Temos

$$M - \lambda I = \begin{pmatrix} -\lambda & 1 & 0 & 0 & \cdots & 0 \\ 0 & -\lambda & 1 & 0 & \cdots & 0 \\ 0 & 0 & -\lambda & 1 & \cdots & 0 \\ 0 & 0 & 0 & -\lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \cdots & -\lambda \end{pmatrix} \quad (5-92)$$

Fazendo o desenvolvimento de Laplace ao longo da primeira coluna, temos

$$\det(M - \lambda I) = -\lambda \det(A) + (-1)^{m+1} \det(B) \quad (5-93)$$

onde

$$A = \begin{pmatrix} -\lambda & 1 & 0 & \cdots & 0 \\ 0 & -\lambda & 1 & \cdots & 0 \\ 0 & 0 & -\lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\lambda \end{pmatrix} \quad (5-94)$$

$$B = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -\lambda & 1 & 0 & \cdots & 0 \\ 0 & -\lambda & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\lambda & 1 \end{pmatrix} \quad (5-95)$$

Como A é uma matriz triangular superior, então $\det(A)$ é o produto dos elementos da diagonal principal. Ou seja, $\det(A) = (-\lambda)^{m-1}$. Logo

$$\det(M - \lambda I) = -\lambda \det(A) + (-1)^{m+1} \det(B) = (-\lambda)^m + (-1)^{m+1} \det(B) \quad (5-96)$$

Como B é uma matriz triangular inferior então $\det(B)$ é o produto dos elementos da diagonal principal. Ou seja, $\det(B) = 1^{m-1} = 1$. Logo

$$\begin{aligned} \det(M - \lambda I) &= \\ (-\lambda)^m + (-1)^{m+1} \det(B) &= (-\lambda)^m + (-1)^{m+1} 1 = \\ (-\lambda)^m + (-1)^{m+1} &= (-1)^m \lambda^m + (-1)^m (-1) = \\ &= (-1)^m (\lambda^m - 1) \end{aligned} \quad (5-97)$$

Para achar os autovalores, igualamos o polinômio característico a 0:

$$\begin{aligned} p(\lambda) = 0 &\iff \\ (-1)^m (\lambda^m - 1) = 0 &\iff (\lambda^m - 1) = 0 \iff \\ \lambda^m &= 1 \end{aligned} \quad (5-98)$$

Portanto os m autovalores são os m números complexos da forma

$$\lambda_j = \omega^j \quad (5-99)$$

onde $\omega := e^{ij\frac{2\pi}{m}}$ e $1 \leq j \leq m$. Note que $\lambda_m = 1$.

Dado um natural j tal que $1 \leq j < m$, temos

$$\lambda_j = \omega^j \quad (5-100)$$

$$\lambda_{m-j} = \omega^{m-j} \quad (5-101)$$

Portanto

$$|\lambda_j| = |\lambda_{m-j}| = 1 \quad (5-102)$$

e

$$\lambda_j \lambda_{m-j} = \omega^j \omega^{m-j} = \omega^{j+m-j} = \omega^m = 1 \quad (5-103)$$

Ou seja,

$$\lambda_j \lambda_{m-j} = 1 \quad (5-104)$$

As equações (5-102) e (5-104) significam que $\lambda_{m-j} = \overline{\lambda_j}$.

Se m é ímpar, podemos então colocar os autovalores na ordem

$$(1, \lambda_1, \lambda_{m-1}, \lambda_2, \lambda_{m-2}, \dots, \lambda_{\frac{m-1}{2}}, \lambda_{m-\frac{m-1}{2}}) \quad (5-105)$$

Como $\lambda_{m-j} = \overline{\lambda_j}$ então a lista acima é igual a

$$(1, \lambda_1, \overline{\lambda_1}, \lambda_2, \overline{\lambda_2}, \dots, \lambda_{\frac{m-1}{2}}, \overline{\lambda_{\frac{m-1}{2}}}) \quad (5-106)$$

Se m é par então $\lambda_{\frac{m}{2}} = -1$ e podemos colocar os autovalores na ordem

$$(1, \lambda_1, \lambda_{m-1}, \lambda_2, \lambda_{m-2}, \dots, \lambda_{\frac{m-2}{2}}, \lambda_{m-\frac{m-2}{2}}, -1) \quad (5-107)$$

Como $\lambda_{m-j} = \overline{\lambda_j}$ então a lista acima é igual a

$$(1, \lambda_1, \overline{\lambda_1}, \lambda_2, \overline{\lambda_2}, \dots, \lambda_{\frac{m-2}{2}}, \overline{\lambda_{\frac{m-2}{2}}}, -1) \quad (5-108)$$

Para todo operador linear A em um espaço vetorial real de dimensão finita, existe uma base ordenada na qual a matriz de A tem a forma canônica de Jordan. No nosso caso particular, em que a multiplicidade algébrica de todos os autovalores é 1, a matriz na forma canônica de Jordan é ainda mais simples do que no caso geral. Cada autovalor real aparecerá na diagonal principal. Para cada par de autovalores complexos conjugados $\lambda_j = \alpha_j + i\beta_j$ e $\overline{\lambda_j} = \alpha_j - i\beta_j$, o bloco

$$\begin{pmatrix} \alpha & \beta \\ -\beta & \alpha \end{pmatrix} \quad (5-109)$$

aparecerá ao longo diagonal principal. No nosso caso, $\lambda_j = \omega^j$, portanto

$$\alpha_j = \cos\left(j\frac{2\pi}{m}\right) \quad (5-110)$$

$$\beta_j = \sin\left(j\frac{2\pi}{m}\right) \quad (5-111)$$

Seja

$$m' = \begin{cases} \frac{m-1}{2} & \text{se } m \text{ é ímpar} \\ \frac{m-2}{2} & \text{se } m \text{ é par} \end{cases} \quad (5-112)$$

Portanto para m ímpar a matriz da operação C na forma canônica de Jordan é:

$$J = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cos\left(\frac{2\pi}{m}\right) & \sin\left(\frac{2\pi}{m}\right) & \cdots & 0 & 0 \\ 0 & -\sin\left(\frac{2\pi}{m}\right) & \cos\left(\frac{2\pi}{m}\right) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \cos\left(m'\frac{2\pi}{m}\right) & \sin\left(m'\frac{2\pi}{m}\right) \\ 0 & 0 & 0 & \cdots & -\sin\left(m'\frac{2\pi}{m}\right) & \cos\left(m'\frac{2\pi}{m}\right) \end{pmatrix} \quad (5-113)$$

E para m par, a matriz da operação C na forma canônica de Jordan é:

$$J = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \cos\left(\frac{2\pi}{m}\right) & \sin\left(\frac{2\pi}{m}\right) & \cdots & 0 & 0 & 0 \\ 0 & -\sin\left(\frac{2\pi}{m}\right) & \cos\left(\frac{2\pi}{m}\right) & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \cos\left(m'\frac{2\pi}{m}\right) & \sin\left(m'\frac{2\pi}{m}\right) & 0 \\ 0 & 0 & 0 & \cdots & -\sin\left(m'\frac{2\pi}{m}\right) & \cos\left(m'\frac{2\pi}{m}\right) & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & -1 \end{pmatrix} \quad (5-114)$$

$$J = J =$$

5.3.2 Base ordenada \mathcal{B} correspondente à forma canônica de Jordan

Precisamos agora conhecer a base ordenada \mathcal{B} de \mathbb{R}^m que deixa a matriz de C na forma canônica de Jordan. Para isso precisamos calcular os autovetores complexos de C . Como C têm exatamente m autovalores complexos, e todos têm multiplicidade 1, então C tem exatamente um autovetor para cada autovalor.

Sejam v_1 o autovetor correspondente ao autovalor 1, v_{-1} o autovetor correspondente ao autovalor -1 (se m é par) e v_j e v'_j os vetores correspondentes ao par conjugado $\lambda_j = \omega^{-j}$ e $\overline{\lambda_j} = \omega^j$. Aqui a notação v_1 significa “o autovetor correspondente ao autovalor 1” e não “a primeira coordenada do vetor v ”; dado um natural h tal que $1 \leq h \leq m$, seja $v_{j,h}$ a h -ésima coordenada do vetor v_j .

Assim a base ordenada \mathcal{B} é, para m ímpar:

$$\mathcal{B} = \{v_1, v_1, v'_1, \dots, v_{m'}, v'_{m'}\} \quad (5-115)$$

³ E para m par:

$$\mathcal{B} = \{v_1, v_1, v'_1, \dots, v_{m'}, v'_{m'}, v_{-1}\} \quad (5-116)$$

³Lembre que m' é definido pela equação (5-112).

Começemos calculando v_1 .

$$\begin{aligned}
 Cv_1 = 1v_1 &\iff \\
 \begin{pmatrix} v_{1,2} \\ v_{1,3} \\ \vdots \\ v_{1,m} \\ v_{1,1} \end{pmatrix} &= \begin{pmatrix} v_{1,1} \\ v_{1,2} \\ \vdots \\ v_{1,m-1} \\ v_{1,m} \end{pmatrix} \iff \\
 v_{1,1} &= \cdots = v_{1,m}
 \end{aligned} \tag{5-117}$$

Temos um grau de liberdade para escolher $v_{1,1}$ (que é igual a todas as outras coordenadas). Escolhamos $v_{1,1} = \frac{1}{\sqrt{m}}$, para que $\|v_1\| = 1$. Assim,

$$v_1 = \begin{pmatrix} \frac{1}{\sqrt{m}} \\ \frac{1}{\sqrt{m}} \\ \vdots \\ \frac{1}{\sqrt{m}} \end{pmatrix} \tag{5-118}$$

Agora descubramos v_{-1} (aqui vamos supor que m é par).

$$\begin{aligned}
 Cv_{-1} = -1v_{-1} &\iff \\
 \begin{pmatrix} v_{-1,2} \\ v_{-1,3} \\ \vdots \\ v_{-1,m} \\ v_{-1,1} \end{pmatrix} &= \begin{pmatrix} -v_{-1,1} \\ -v_{-1,2} \\ \vdots \\ -v_{-1,m-1} \\ -v_{-1,m} \end{pmatrix} \iff \\
 -v_{-1,1} &= v_{-1,2} = -v_{-1,3} = \cdots = -v_{-1,m-1} = v_{-1,m}
 \end{aligned} \tag{5-119}$$

Temos um grau de liberdade para escolher $v_{-1,1}$ (que determina todas as outras coordenadas). Escolhamos $v_{-1,1} = \frac{1}{\sqrt{m}}$, para que $\|v_{-1}\| = 1$. Assim,

$$v_{-1} = \begin{pmatrix} \frac{1}{\sqrt{m}} \\ -\frac{1}{\sqrt{m}} \\ \frac{1}{\sqrt{m}} \\ \vdots \\ -\frac{1}{\sqrt{m}} \end{pmatrix} \tag{5-120}$$

Agora descubramos \mathbf{v}_j e \mathbf{v}'_j (aqui, m pode ser par ou ímpar). Seja \mathbf{w}_j o autovetor complexo correspondente ao autovalor λ_j . Temos

$$\mathbf{w}_j = \mathbf{v}_j + i\mathbf{v}'_j \quad (5-121)$$

$$\begin{aligned} C\mathbf{w}_j &= \lambda_j \mathbf{w}_j \iff \\ \begin{pmatrix} \mathbf{w}_{j,2} \\ \mathbf{w}_{j,3} \\ \vdots \\ \mathbf{w}_{j,m} \\ \mathbf{w}_{j,1} \end{pmatrix} &= \begin{pmatrix} \lambda_j \mathbf{w}_{j,1} \\ \lambda_j \mathbf{w}_{j,2} \\ \vdots \\ \lambda_j \mathbf{w}_{j,m-1} \\ \lambda_j \mathbf{w}_{j,m} \end{pmatrix} \end{aligned} \quad (5-122)$$

A igualdade acima é equivalente ao conjunto das duas seguintes equações⁴:

1. $\mathbf{w}_{j,1} = \lambda_j \mathbf{w}_{j,m}$
2. $\forall k \in \mathbb{N} \mid 1 < k \leq m, \mathbf{w}_{j,k} = \lambda_j^{k-1} \mathbf{w}_{j,1}$

Seja $\rho_j = |\mathbf{w}_{j,1}|$ e seja $\theta_j = \arg(\mathbf{w}_{j,1})$, de modo que $\mathbf{w}_{j,1} = \rho_j e^{i\theta_j}$. Como $\lambda_j = \omega^j$ então da equação (2) concluímos que, para todo natural k tal que $1 \leq k \leq m$:

$$\begin{aligned} \mathbf{w}_{j,k} &= \\ \lambda_j^{k-1} \mathbf{w}_{j,1} &= (\omega^j)^{k-1} \rho_j e^{i\theta_j} = \rho_j e^{i((k-1)j\frac{2\pi}{m} + \theta)} = \\ \rho_j \cos((k-1)j\frac{2\pi}{m} + \theta) &+ i\rho_j \sin((k-1)j\frac{2\pi}{m} + \theta) \end{aligned} \quad (5-124)$$

Das equações (5-121) e (5.3.2) segue que, para todo natural k tal que $1 \leq k \leq m$:

$$v_{\lambda_j,k} = \rho_j \cos((k-1)j\frac{2\pi}{m} + \theta) \quad (5-125)$$

$$v'_{\lambda_j,k} = \rho_j \sin((k-1)j\frac{2\pi}{m} + \theta) \quad (5-126)$$

Para que $\|\mathbf{v}_j\| = 1$ temos

$$\sqrt{\sum_{k=1}^m (\rho_j \cos((k-1)j\frac{2\pi}{m} + \theta))^2} = 1 \quad (5-127)$$

⁴ Note que a equação (1) combinada com a equação (2) significa

$$\mathbf{w}_{j,1} = \lambda_j \mathbf{w}_{j,m} = \lambda_j \lambda_j^{m-1} \mathbf{w}_{j,1} = \lambda_j^m \mathbf{w}_{j,1}$$

portanto, como $\mathbf{w}_{j,1} \neq 0$, (pois se fosse $\mathbf{w}_{j,1} = 0$ então todas as coordenadas de \mathbf{w}_j seriam nulas e \mathbf{w}_j não seria um autovetor) é necessário que $\lambda_j^m = 1$. Isso de fato ocorre, pois

$$\lambda_j^m = (\omega^j)^m = (\omega^m)^j = 1 \quad (5-123)$$

$$\begin{aligned}
& \sum_{k=1}^m \frac{1 - \cos(2((k-1)j\frac{2\pi}{m} + \theta))}{2} = \\
& \frac{m}{2} - \frac{1}{2} \sum_{k=1}^m \cos(2((k-1)j\frac{2\pi}{m} + \theta)) = \\
& \frac{m}{2} - \frac{1}{2} \sum_{k=1}^m \operatorname{Re}\{e^{i2((k-1)j\frac{2\pi}{m} + \theta)}\} = \\
& \frac{m}{2} - \frac{1}{2} \operatorname{Re}\left\{\sum_{k=1}^m e^{i2((k-1)j\frac{2\pi}{m} + \theta)}\right\} \tag{5-134}
\end{aligned}$$

Das equações (5-128) e (5-132) temos

$$\frac{m}{2} + \frac{1}{2} \operatorname{Re}\left\{\sum_{k=1}^m e^{i2((k-1)j\frac{2\pi}{m} + \theta_j)}\right\} = \frac{1}{\rho_j^2}. \tag{5-135}$$

Das equações (5-133) e (5-134) temos

$$\frac{m}{2} - \frac{1}{2} \operatorname{Re}\left\{\sum_{k=1}^m e^{i2((k-1)j\frac{2\pi}{m} + \theta_j)}\right\} = \frac{1}{\rho_j^2}. \tag{5-136}$$

Seja

$$\sigma_j := \sum_{k=1}^m e^{i2((k-1)j\frac{2\pi}{m} + \theta_j)} \tag{5-137}$$

de modo que

$$\frac{m}{2} + \frac{1}{2} \operatorname{Re}\{\sigma_j\} = \frac{1}{\rho_j^2}, \tag{5-138}$$

$$\frac{m}{2} - \frac{1}{2} \operatorname{Re}\{\sigma_j\} = \frac{1}{\rho_j^2}. \tag{5-139}$$

Note que σ_j é uma progressão geométrica. Portanto

$$\begin{aligned}
\sigma_j &= e^{i2((k-1)j\frac{2\pi}{m} + \theta_j)} \frac{(e^{i2((k-1)j\frac{2\pi}{m} + \theta_j)})^m - 1}{e^{i2((k-1)j\frac{2\pi}{m} + \theta_j)} - 1} = \\
& e^{i2((k-1)j\frac{2\pi}{m} + \theta_j)} \frac{e^{i2((k-1)j2\pi + m\theta_j)} - 1}{e^{i2((k-1)j\frac{2\pi}{m} + \theta_j)} - 1} = \\
& e^{i2((k-1)j\frac{2\pi}{m} + \theta_j)} \frac{e^{i2m\theta_j} - 1}{e^{i2((k-1)j\frac{2\pi}{m} + \theta_j)} - 1}. \tag{5-140}
\end{aligned}$$

As contas ficam muito mais simples se $\theta_j = 0$. Procuremos uma solução com $\theta_j = 0$ (se

for impossível, retomamos os cálculos a partir daqui). Com $\theta_j = 0$, temos

$$\sigma_j = e^{i2((k-1)j\frac{2\pi}{m})} \frac{e^0 - 1}{e^{i2((k-1)j\frac{2\pi}{m})} - 1} = 0 \quad (5-141)$$

portanto, pela equação (5-138), temos

$$\frac{m}{2} = \frac{1}{\rho_j^2} \quad (5-142)$$

portanto

$$\begin{aligned} \frac{2}{m} &= \rho_j^2 \\ \rho_j &= \pm \sqrt{\frac{2}{m}}. \end{aligned} \quad (5-143)$$

Entre as duas alternativas, escolhemos (arbitrariamente)

$$\rho_j = \sqrt{\frac{2}{m}} \quad (5-144)$$

Note que a equação (5-139) também é satisfeita.

Sabendo que $\theta_j = 0$, então das equações (5-125) e (5-144) segue que

$$\mathbf{v}_{j,k} = \sqrt{\frac{2}{m}} \cos((k-1)j\frac{2\pi}{m}) \quad (5-145)$$

E das equações (5-126) e (5-144) segue que

$$\mathbf{v}'_{j,k} = \sqrt{\frac{2}{m}} \sin((k-1)j\frac{2\pi}{m}) \quad (5-146)$$

Para expressar a distância entre dois vetores em termos das suas \mathcal{B} -coordenadas, desejamos que \mathcal{B} seja ortonormal (ou seja, que todos os vetores de \mathcal{B} tenham norma 1 e que cada par de vetores distintos seja ortogonal). Já sabemos que os vetores de \mathcal{B} têm norma 1, por construção. Só falta verificar que são mutuamente ortogonais. Isso foi feito detalhadamente na seção C.3 do apêndice C.

5.3.3 Cálculo de distâncias na base ordenada \mathcal{B}

Lembre que para m ímpar:

$$\mathcal{B} = \{\mathbf{v}_1, \mathbf{v}'_1, \dots, \mathbf{v}_{m'}, \mathbf{v}'_{m'}, \mathbf{v}_1\} \quad (5-147)$$

onde m' é definido pela equação (5-112). E para m par:

$$\mathcal{B} = \{\mathbf{v}_1, \mathbf{v}'_1, \dots, \mathbf{v}_{m'}, \mathbf{v}'_{m'}, v_1, v_{-1}\} \quad (5-148)$$

Para cada $j \in \mathbb{N}$ tal que $1 \leq j \leq m'$ seja H_j o subespaço de \mathbb{R}^m gerado pelos vetores \mathbf{v}_j e \mathbf{v}'_j . H_j é um hiperplano de dimensão 2. Seja I o subespaço de \mathbb{R}^m gerado por v_1 ; I é o autoespaço correspondente ao autovalor 1. Se m é par, seja J o subespaço de \mathbb{R}^m gerado por v_{-1} ; J é o autoespaço correspondente ao autovalor -1. Todos esses subespaços (I, J , e todos os H_j) são ortogonais entre si e a soma direta deles é \mathbb{R}^m . Se m é ímpar:

$$\left(\bigoplus_{j=1}^{m'} H_j \right) \oplus I = \mathbb{R}^m \quad (5-149)$$

Se m é par:

$$\left(\bigoplus_{j=1}^{m'} H_j \right) \oplus I \oplus J = \mathbb{R}^m \quad (5-150)$$

Analisando a matriz de C na forma canônica de Jordan (equações (5-113) e (5-114)) vemos que todos esses subespaços são invariantes por C .

Se m é ímpar então para cada $p \in \mathbb{R}^m$ temos pelo teorema 5.14:

$$p = \left(\sum_{j=1}^{m'} p_{\lambda_j} \mathbf{v}_j + p'_{\lambda_j} \mathbf{v}'_j \right) + p_1 v_1 \quad (5-151)$$

onde

$$p_{\lambda_j} = p \cdot \mathbf{v}_j \quad (5-152)$$

$$p'_{\lambda_j} = p \cdot \mathbf{v}'_j \quad (5-153)$$

$$p_1 = p \cdot v_1 \quad (5-154)$$

Teorema 5.14. *Dado um espaço vetorial E munido de produto interno, se \mathcal{D} é uma base ortonormal, as \mathcal{D} -coordenadas de um vetor qualquer podem ser obtidas pelo respectivo produto interno. Ou seja, se*

$$\mathcal{D} = \{d_1, \dots, d_m\} \quad (5-155)$$

é uma base ortonormal de E então

$$\forall u \in E, u = \sum_{j=1}^m (u \cdot d_j) d_j \quad (5-156)$$

Prova. Como \mathcal{D} é uma base de E então para qualquer $u \in E$ existem escalares $\alpha_1, \dots, \alpha_m$ tais que

$$u = \alpha_1 d_1 + \dots + \alpha_m d_m \quad (5-157)$$

Dado $j \in \mathbb{N}$ tal que $1 \leq j \leq m$ podemos fazer o produto interno de ambos os lados da equação (5-157) por d_j :

$$\begin{aligned} u \cdot d_j &= (\alpha_1 d_1 + \dots + \alpha_m d_m) \cdot d_j = \\ &= \alpha_1 (d_1 \cdot d_j) + \dots + \alpha_m (d_m \cdot d_j) \end{aligned} \quad (5-158)$$

Como \mathcal{D} é ortonormal, então para todo natural l tal que $1 \leq l \leq m$ e $l \neq j$, $d_l \cdot d_j = 0$, portanto $\alpha_l (d_l \cdot d_j) = 0$. Portanto a equação (5-158) fica

$$u \cdot d_j = \alpha_j d_j \cdot d_j = \alpha_j \quad (5-159)$$

Como \mathcal{D} é ortonormal, $d_j \cdot d_j = 1$, portanto

$$u \cdot d_j = \alpha_j (1) = \alpha_j \quad (5-160)$$

□

Se m é par então para cada $p \in \mathbb{R}^m$ temos

$$p = \left(\sum_{j=1}^{m'} p_{\lambda_j} \mathbf{v}_j + p'_{\lambda_j} \mathbf{v}'_j \right) + p_{-1} \mathbf{v}_{-1} + p_1 \mathbf{v}_1 \quad (5-161)$$

Em ambos os casos (m par e m ímpar) definimos $P_{H_j}(p)$ como a projeção ortogonal de p sobre H_j :

$$P_{H_j}(p) := p_{\lambda_j} \mathbf{v}_j + p'_{\lambda_j} \mathbf{v}'_j \quad (5-162)$$

Dados $p, q \in \mathbb{R}^m$ seja $d(p, q)$ a distância euclidiana entre p e q e seja

$$d_{H_j}(p, q) := d(P_{H_j}(p), P_{H_j}(q)) \quad (5-163)$$

Claramente temos

$$d_{H_j}(p, q) = \sqrt{(p_{\lambda_j} - q_{\lambda_j})^2 + (p'_{\lambda_j} - q'_{\lambda_j})^2} \quad (5-164)$$

Analogamente, seja $P_I(p) := p_1 \mathbf{v}_1$ (a projeção ortogonal de p sobre I) e para $p, q \in \mathbb{R}^m$ seja $d_I(p, q) = d(P_I(p), P_I(q))$. Temos

$$d_I(p, q) = |q_1 - p_1| \quad (5-165)$$

Para m par seja

$$P_J(p) := p_{-1}v_{-1} \quad (5-166)$$

(a projeção ortogonal de p sobre J) e para $p, q \in \mathbb{R}^m$ seja

$$d_J(p, q) := d(P_J(p), P_J(q)) \quad (5-167)$$

Temos

$$d_J(p, q) = |q_{-1} - p_{-1}| \quad (5-168)$$

Se m é ímpar então para cada $p, q \in \mathbb{R}^m$ temos

$$\begin{aligned} d(p, q) = & \sqrt{\left(\sum_{j=1}^{m'} (q_{\lambda_j} - p_{\lambda_j})^2 + (q'_{\lambda_j} - p'_{\lambda_j})^2\right) + (q_1 - p_1)^2} = \\ & \sqrt{\left(\sum_{j=1}^{m'} d_{H_j}^2(p, q)\right) + d_I^2(p, q)} \end{aligned} \quad (5-169)$$

Analogamente se m é par então para cada $p, q \in \mathbb{R}^m$ temos

$$d(p, q) = \sqrt{\left(\sum_{j=1}^{m'} d_{H_j}^2(p, q)\right) + d_I^2(p, q) + d_J^2(p, q)} \quad (5-170)$$

Nosso objetivo é achar a distância mínima. Pela observação 3.5 podemos analisar os respectivos quadrados da distâncias pois o mínimo dos quadrados corresponde ao quadrado do mínimo. Temos então para quaisquer $p, q \in \mathbb{R}^m$ e m ímpar:

$$d^2(p, q) = \left(\sum_{j=1}^{m'} d_{H_j}^2(p, q)\right) + d_I^2(p, q) \quad (5-171)$$

E para m par:

$$d^2(p, q) = \left(\sum_{j=1}^{m'} d_{H_j}^2(p, q)\right) + d_I^2(p, q) + d_J^2(p, q) \quad (5-172)$$

Analisemos a matriz da operação C na forma canônica de Jordan. Observe as equações (5-113) e (5-114). Lembre que em \mathbb{R}^2 o operador linear que rotaciona por ângulo θ é descrito pela matriz

$$\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (5-173)$$

[5] Então

- Dado $p \in \mathbb{R}^m$, C não muda a projeção ortogonal de p sobre I :

$$P_I(Cp) = P_I(p) \quad (5-174)$$

Para todo $c \in \mathbb{Z}$:

$$P_I(C^c p) = P_I(p) \quad (5-175)$$

portanto para quaisquer $p, q \in \mathbb{R}^m$ e qualquer $c \in \mathbb{Z}$:

$$d_I^2(C^c p, q) = d_I^2(p, q) = (q_1 - p_1)^2 \quad (5-176)$$

- Dado $p \in \mathbb{R}^m$, C reflete a projeção ortogonal de p sobre J em torno da origem:

$$P_J(Cp) = -P_J(p) = P_J(-p) \quad (5-177)$$

Para todo $c \in \mathbb{Z}$:

$$P_J(C^c p) = P_J((-1)^c p) \quad (5-178)$$

portanto para quaisquer $p, q \in \mathbb{R}^m$ e qualquer $c \in \mathbb{Z}$:

$$d_J^2(C^c p, q) = d_J^2((-1)^c p, q) = (q_{-1} - (-1)^c p_{-1})^2 \quad (5-179)$$

portanto

$$d_J^2(C^c p, q) = q_{-1}^2 + p_{-1}^2 - 2q_{-1}(-1)^c p_{-1} \quad (5-180)$$

- Dado $p \in \mathbb{R}^m$, C rotaciona $P_{H_j}(p)$, $-j\frac{2\pi}{m}$ rad. Se representarmos $P_{H_j}(p)$ em coordenadas polares

$$P_{H_j}(p) = (r_{pj}, \theta_{pj}) \quad (5-181)$$

então

$$P_{H_j}(Cp) = (r_{pj}, \theta_{pj} - j\frac{2\pi}{m}) \quad (5-182)$$

Para todo $c \in \mathbb{Z}$:

$$P_{H_j}(C^c p) = (r_{pj}, \theta_{pj} - cj\frac{2\pi}{m}) \quad (5-183)$$

Pela lei dos cossenos sabemos que para quaisquer $p, q \in \mathbb{R}^2$:

$$d^2(p, q) = \|p\|^2 + \|q\|^2 - 2\|p\|\|q\|\cos(\theta(p, q)) \quad (5-184)$$

onde $\theta(p, q)$ é o ângulo entre os vetores p e q . Sejam (r_{qj}, θ_{qj}) as coordenadas polares de $P_{H_j}(q)$. Então

$$d_{H_j}^2(C^c p, q) = r_{pj}^2 + r_{qj}^2 - 2r_{pj}r_{qj}\cos(\theta_{qj} - \theta_{pj} + cj\frac{2\pi}{m}) \quad (5-185)$$

Se definirmos $\Delta_j := \theta_{qj} - \theta_{pj}$ temos então

$$d_{H_j}^2(C^c p, q) = r_{pj}^2 + r_{qj}^2 - 2r_{pj}r_{qj} \cos(\Delta_j + cj\frac{2\pi}{m}) \quad (5-186)$$

Combinando as equações (5-172), (5-176), (5-180) e (5-186) concluímos que, para $c \in \mathbb{Z}$ e m par:

$$\begin{aligned} d^2(C^c p, q) = & \\ & \left(\sum_{j=1}^{m'} r_{pj}^2 + r_{qj}^2 - 2r_{pj}r_{qj} \cos(\Delta_j + cj\frac{2\pi}{m}) \right) + (q_1 - p_1)^2 + q_{-1}^2 + p_{-1}^2 - 2q_{-1}(-1)^c p_{-1} \end{aligned} \quad (5-187)$$

Aglutinando todos os termos que independem de c temos, para m par:

$$\begin{aligned} d^2(C^c p, q) = & \\ & \left(\sum_{j=1}^{m'} r_{pj}^2 + r_{qj}^2 \right) + (q_1 - p_1)^2 + q_{-1}^2 + p_{-1}^2 - \\ & 2 \left(\sum_{j=1}^{m'} r_{pj}r_{qj} \cos(\Delta_j + cj\frac{2\pi}{m}) \right) - 2q_{-1}p_{-1}(-1)^c = \\ & \|p\|^2 + \|q\|^2 - 2q_1p_1 - 2 \left(\sum_{j=1}^{m'} r_{pj}r_{qj} \cos(\Delta_j + cj\frac{2\pi}{m}) + q_{-1}p_{-1}(-1)^c \right). \end{aligned} \quad (5-188)$$

Analogamente para m ímpar:

$$d^2(C^c p, q) = \|p\|^2 + \|q\|^2 - 2q_1p_1 - 2 \left(\sum_{j=1}^{m'} r_{pj}r_{qj} \cos(\Delta_j + cj\frac{2\pi}{m}) \right). \quad (5-189)$$

5.4 Redução eficiente do produto de matriz circulante por vetor ao cálculo da distância especial

Como vimos na seção 3.1.1 o cálculo da distância especial pode ser eficientemente reduzido ao cálculo do produto entre matriz circulante e vetor. Queremos saber se a redução eficiente também ocorre no sentido oposto: podemos reduzir eficientemente o produto entre matriz circulante e vetor ao cálculo da distância especial? Se essa redução for possível em tempo $o(m \log m)$ e se soubermos como calcular a distância especial em tempo $o(m \log m)$ então podemos multiplicar matriz circulante por vetor em tempo $o(m \log m)$. Portanto, se tal redução eficiente for possível então calcular a distância especial em tempo $o(m \log m)$ é pelo menos tão “difícil” quanto multiplicar matriz circulante

por vetor em tempo $o(m \log m)$, o que seria desencorajador.

Essa redução, se existir, não é óbvia. A partir de M_{pq} o algoritmo 3.1, usa apenas uma de suas coordenadas (a máxima) para construir $d_e(p, q)$. Ocorre “perda de informação” e não é fácil reconstruir M_{pq} tendo como base apenas $d_e(p, q)$ e $v_c(p, q)$.

Conclusão

Foram apresentadas técnicas de aceleração da variação do KNN:

- Aceleração do cálculo da distância especial através da FFT
- Eliminação rápida de vetores de treinamento desfavoráveis
- Paralelismo

Foi descrita uma aplicação em visão computacional. Foi descrito um programa implementado em C++11, de desempenho drasticamente melhor do que o força bruta. O desempenho foi medido e apresentado. Para a base de dados Mnist, o algoritmo mais otimizado e rodando em 8 threads foi 398 vezes mais rápido do que o força bruta sequencial.

Quanto a trabalhos futuros, a primeira sugestão é adaptar uma estrutura de dados na literatura de KNN à variação abordada neste trabalho. Outra sugestão é obter uma implementação em software da medida de similaridade descrita em [11], acoplá-la ao CyclicKNN, e testar o desempenho e eficácia da solução em instâncias reais de processamento de imagem.

Referências Bibliográficas

- [1] BELONGIE, S. **Rodrigues' rotation formula**. From MathWorld—A Wolfram Web Resource, created by Eric W. Weisstein. Acessado em 2013-09-19.
- [2] BEYGELZIMER, A.; KAKADE, S.; LANGFORD, J. **Cover trees for nearest neighbor**. In: *Proceedings of the 23rd international conference on Machine learning*, p. 97–104. ACM, 2006.
- [3] FRIGO, M.; JOHNSON, S. G. **The design and implementation of fftw3**. *Proceedings of the IEEE*, 93(2):216–231, 2005.
- [4] GOLUB, G. H.; VAN LOAN, C. F. **Matrix computations**. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore, MD, USA, third edition, 1996.
- [5] HOFFMAN, K.; KUNZE, R. **Linear algebra, Second edition**. Prentice-Hall, Inc, Englewood Cliffs, N.J., second edition, 1971.
- [6] KARGER, D. R.; RUHL, M. **Finding nearest neighbors in growth-restricted metrics**. In: *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, p. 741–750. ACM, may 2002.
- [7] KOROBEYNIKOV, A. **Computation- and space-efficient implementation of SSA**. *Statistics and Its Interface*, p. 357–368, 2010.
- [8] LIMA, E. L. **Espaços métricos**. Instituto de Matemática Pura e Aplicada, CNPq, 1977.
- [9] LIMA, E. L. **Álgebra linear, 6ª edição**. *Coleção Matemática Universitária*. IMPA, 2003.
- [10] LOWE, D. G. **Distinctive image features from scale-invariant keypoints**. *International journal of computer vision*, 60(2):91–110, 2004.
- [11] NELSON, J. D.; KINGSBURY, N. G. **Enhanced shift and scale tolerance for rotation invariant polar matching with dual-tree wavelets**. *Image Processing, IEEE Transactions on*, 20(3):814–821, 2011.

- [12] WEISSTEIN, E. W. **Fourier matrix**. From MathWorld—A Wolfram Web Resource. Acessado em 2014-01-06.

Propriedades da distância especial

Conforme vimos na definição 2.12, para $p, q \in \mathbb{R}^m$ a nossa distância especial é definida como

$$d_e(p, q) := \min_{p' \in C^*(p)} d(p', q) \quad (\text{A-1})$$

Teorema A.1. *A distância especial é não-negativa:*

$$\forall p, q \in \mathbb{R}^m, d_e(p, q) \geq 0 \quad (\text{A-2})$$

Prova. Seja $q \in \mathbb{R}^m$. A distância euclidiana é não-negativa, portanto

$$\forall p' \in \mathbb{R}^m, d(p', q) \geq 0 \quad (\text{A-3})$$

$C^*(p) \in \mathbb{R}^m$, portanto:

$$\forall p' \in C^*(p), d(p', q) \geq 0 \quad (\text{A-4})$$

Portanto, $d_e(p, q)$ é o mínimo de um conjunto de números reais não negativos, portanto

$$d_e(p, q) \geq 0 \quad (\text{A-5})$$

□

Teorema A.2. *A distância especial fere a propriedade*

$$p \neq q \implies d_e(p, q) > 0 \quad (\text{A-6})$$

Prova. Contraexemplo: $p = (1, \dots, m)$ e $q = (2, \dots, m, 1)$. Note que $q = Cp$. Claramente $p \neq q$, porém $d_e(p, q) = 0$. De fato:

$$d_e(p, q) = \min_{r \in C^*(p)} d(r, q) \quad (\text{A-7})$$

Temos $q = Cp$, portanto $q \in C^*(p)$, portanto o conjunto

$$\{d(p', q) \mid p' \in C^*(p)\} \quad (\text{A-8})$$

inclui o elemento $d(q, q)$, que é 0. Todos os outros elementos desse conjunto são reais não-negativos, portanto o mínimo é 0. \square

Lema A.3. $\forall c \in \mathbb{Z} \forall p, q \in \mathbb{R}^m, d(C^c p, C^c q) = d(p, q)$

Prova. C é um operador linear que preserva distâncias (teorema 5.5), ou seja,

$$\forall p, q \in \mathbb{R}^m, d(Cp, Cq) = d(p, q) \quad (\text{A-9})$$

Segue imediatamente que

$$\forall p, q \in \mathbb{R}^m, d(C^2 p, C^2 q) = d(p, q) \quad (\text{A-10})$$

e por indução finita segue que

$$\forall c \in \mathbb{N}, \forall p, q \in \mathbb{R}^m, d(C^c p, C^c q) = d(p, q) \quad (\text{A-11})$$

Por convenção, C^0 é a função identidade, portanto

$$\forall p, q \in \mathbb{R}^m, d(C^0 p, C^0 q) = d(p, q) \quad (\text{A-12})$$

Das equações (A-11) e (A-12) segue que

$$\forall c \in \mathbb{Z}_{\geq 0} \forall p, q \in \mathbb{R}^m, d(C^c p, C^c q) = d(p, q) \quad (\text{A-13})$$

Portanto para quaisquer $p, q \in \mathbb{R}^m$ e qualquer $c \in \mathbb{Z}_{\geq 0}$ temos

$$d(p, q) = d(C^c C^{-c} p, C^c C^{-c} q) = d(C^{-c} p, C^{-c} q) \quad (\text{A-14})$$

ou seja:

$$\forall c \in \mathbb{Z}_{\geq 0} \forall p, q \in \mathbb{R}^m, d(C^{-c} p, C^{-c} q) = d(p, q) \quad (\text{A-15})$$

portanto

$$\forall c \in \mathbb{Z}_{\leq 0} \forall p, q \in \mathbb{R}^m, d(C^c p, C^c q) = d(p, q). \quad (\text{A-16})$$

Das afirmações (A-13) e (A-16) segue o que queremos demonstrar. \square

Teorema A.4. *A distância especial é simétrica:*

$$\forall p, q \in \mathbb{R}^m, d_e(p, q) = d_e(q, p) \quad (\text{A-17})$$

Prova. Temos

$$d_e(p, q) := \min_{p' \in C^*(p)} d(p', q) \quad (\text{A-18})$$

Repetimos aqui a definição 2.10: dado um ponto $p \in \mathbb{R}^m$,

$$C^*(p) := \{p, C^0 p, \dots, C^{m-1} p\} \quad (\text{A-19})$$

ou seja:

$$C^*(p) = \{C^c p \mid c \in \{0, \dots, m-1\}\} \quad (\text{A-20})$$

¹ Portanto da equação (A-18) temos

$$d_e(p, q) = \min_{c \in \{0, \dots, m-1\}} d(C^c p, q) \quad (\text{A-21})$$

Analogamente,

$$d_e(q, p) = \min_{l \in \{0, \dots, m-1\}} d(C^l q, p) \quad (\text{A-22})$$

Pelo lema A.3,

$$d_e(q, p) = \quad (\text{A-23})$$

$$\min_{l \in \{0, \dots, m-1\}} d(C^l q, p) = \quad (\text{A-24})$$

$$\min_{l \in \{0, \dots, m-1\}} d(C^{m-l} C^l q, C^{m-l} p) = \quad (\text{A-25})$$

$$\min_{l \in \{0, \dots, m-1\}} d(C^m q, C^{m-l} p) = \quad (\text{A-26})$$

$$\min_{l \in \{0, \dots, m-1\}} d(q, C^{m-l} p) = \quad (\text{A-27})$$

$$\min_{j \in \{1, \dots, m\}} d(q, C^j p) = \quad (\text{A-28})$$

$$\min_{j \in \{1, \dots, m\}} d(C^j p, q) = \quad (\text{A-29})$$

$$\min_{j \in \{0, \dots, m-1\}} d(C^j p, q) = \quad (\text{A-30})$$

$$d_e(p, q) \quad (\text{A-31})$$

¹ $C^*(p)$ é o conjunto dos pontos $C^c p$ para c inteiro entre 0 e $m-1$, inclusive.

A penúltima igualdade segue do fato de que $C^m p = C^0 p$. □

Lema A.5. $\forall n \in \mathbb{Z}_{\geq 0}, d_e(p, q) \leq d(C^n p, q)$

Prova. Para qualquer $n \in \mathbb{Z}_{\geq 0}$, temos $C^n p \in C^*(p)$, portanto

$$d(C^n p, q) \in \{d(p', q) \mid p' \in C^*(p)\} \quad (\text{A-32})$$

Portanto

$$d_e(p, q) = \min_{p' \in C^*(p)} d(p', q) \leq d(C^n p, q) \quad (\text{A-33})$$

□

Teorema A.6. d_e satisfaz a desigualdade triangular: para quaisquer $p, q, s \in \mathbb{R}^m$ vale

$$d(p, s) \leq d(p, q) + d(q, s) \quad (\text{A-34})$$

Prova. Dados $p, q, s \in \mathbb{R}^m$ queremos provar que $d_e(p, q) + d_e(q, s) \geq d_e(p, s)$. Pela definição de d_e , existem inteiros c e c' tais que

$$d_e(p, q) = d(C^c p, q) \quad (\text{A-35})$$

$$d_e(q, s) = d(C^{c'} q, s). \quad (\text{A-36})$$

Pelo lema A.3,

$$d(C^c p, q) = d(p, C^{-c} q) \quad (\text{A-37})$$

$$d(C^{c'} q, s) = d(C^{-c} q, C^{-c-c'} s). \quad (\text{A-38})$$

Combinando as equações (A-35), (A-36), (A-37) e (A-38) obtemos

$$\begin{aligned} d_e(p, q) + d_e(q, s) &= d(p, C^{-c} q) + d(C^{-c} q, C^{-c-c'} s) \geq \\ &= d(p, C^{-c-c'} s) \end{aligned} \quad (\text{A-39})$$

onde a última desigualdade decorre da desigualdade triangular em \mathbb{R}^m para a distância euclidiana d . Mas pelo lema A.5,

$$d(p, C^{-c-c'} s) \geq d_e(p, s). \quad (\text{A-40})$$

Das equações (A-39) e (A-40) segue que

$$d_e(p, q) + d_e(q, s) \geq d_e(p, s). \quad (\text{A-41})$$

□

Teorema A.7. *A distância especial não é invariante quanto a translação. Ou seja, é falso que*

$$\forall p, q, s \in \mathbb{R}^m, d_e(p, q) = d_e(p + s, q + s). \quad (\text{A-42})$$

Prova. Contraexemplo: $p = (0, 1)$, $q = (1, 0)$, $s = (0, -1)$. Para esses valores temos:

$$d_e(p, q) = 0 \quad (\text{A-43})$$

$$d_e(p + s, q + s) = d_e((0, 0), (1, -1)) = \sqrt{2} \quad (\text{A-44})$$

Portanto

$$d_e(p, q) \neq d_e(p + s, q + s). \quad (\text{A-45})$$

□

Teorema A.8. *A distância especial é invariante quanto a translação por vetores múltiplos de $(1, \dots, 1)$*

Prova. Seja $u = (t, \dots, t)$ um vetor múltiplo de $(1, \dots, 1)$. Dado $c \in \mathbb{N}$ temos $C^c u = u$. Por outro lado,

$$C^c(p + u) = C^c p + C^c u = C^c p + u, \quad (\text{A-46})$$

portanto

$$C^*(p + u) = \{r + u \mid r \in C^*(p)\}, \quad (\text{A-47})$$

portanto

$$\begin{aligned} d_e(p + u, q + u) &= \\ \min_{r \in C^*(p+u)} d(r, q + u) &= \\ \min_{s \in C^*(p)} d(s + u, q + u) &= \\ \min_{s \in C^*(p)} d(s, q) &= \\ d_e(p, q). \end{aligned} \quad (\text{A-48})$$



Apêndice do capítulo 3

Teorema B.1. Para quaisquer $p, q \in \mathbb{R}^m$:

$$d_e(p, q) \geq ||p|| - ||q|| \quad (\text{B-1})$$

Prova. Para qualquer espaço vetorial E (real ou complexo) munido de produto interno e para quaisquer $p, q \in E$, vale a *desigualdade de Cauchy-Schwarz* [5]:

$$|p \cdot q| \leq ||p|| ||q|| \quad (\text{B-2})$$

Vale em particular para $E = \mathbb{R}^m$. Em \mathbb{R}^m o produto interno canônico é real. Para qualquer $x \in \mathbb{R}$ vale $x \leq |x|$, portanto

$$p \cdot q \leq |p \cdot q| \quad (\text{B-3})$$

portanto, pela desigualdade de Cauchy-Schwarz:

$$p \cdot q \leq ||p|| ||q|| \quad (\text{B-4})$$

Combinando com a desigualdade (3-53) temos:

$$d_e^2(p, q) \geq ||p||^2 + ||q||^2 - 2 ||p|| ||q|| \quad (\text{B-5})$$

portanto:

$$d_e^2(p, q) \geq (||p|| - ||q||)^2 \quad (\text{B-6})$$

portanto:

$$d_e(p, q) \geq ||p|| - ||q|| \quad (\text{B-7})$$

□

Teorema B.2. Seja $p \in \mathbb{R}^m$ e sejam p_0, \dots, p_{m-1} as coordenadas de p :

$$p := (p_0, \dots, p_{m-1}) \quad (\text{B-8})$$

Aqui indexamos as coordenadas a partir de 0 e não de 1 como no resto desse trabalho. Seja $\hat{p} = (\hat{p}_0, \dots, \hat{p}_{m-1})$ a DFT de p . Sejam¹

$$\begin{aligned} p^c &:= C^c p, \\ \hat{p}^c &:= \mathbb{F}_m p^c \end{aligned} \quad (\text{B-9})$$

Então:

$$\hat{p}_l^c = \omega^{-lc} \hat{p}_l \quad (\text{B-10})$$

Prova. Pela definição da DFT (equação (3-38)):

$$\hat{p}_l = \sum_{j=0}^{m-1} \omega^{jl} p_j \quad (\text{B-11})$$

Temos:

$$p^c = (p_c, \dots, p_{m-1}, p_0, \dots, p_{c-1}) \quad (\text{B-12})$$

Mais precisamente:

- $\forall j \in \{0, \dots, m-1-c\}, p_j^c = p_{j+c}$
- $\forall j \in \{m-c, \dots, m-1\}, p_j^c = p_{j+c-m}$

A transformada de p^c é:

$$\begin{aligned} \hat{p}_l^c &= \sum_{j=0}^{m-1} \omega^{jl} p_j^c = \\ &= \omega^{-lc} \left(\sum_{j=0}^{m-1} \omega^{(j+c)l} p_j^c \right) = \\ &= \omega^{-lc} \left(\sum_{j=0}^{m-1-c} \omega^{(j+c)l} p_{j+c} + \sum_{j=m-c}^{m-1} \omega^{(j+c)l} p_{j+c-m} \right). \end{aligned} \quad (\text{B-13})$$

Fazendo $h = j + c$ no somatório $\sum_{j=0}^{m-1-c}$:

$$\hat{p}_l^c = \omega^{-lc} \left(\sum_{h=c}^{m-1} \omega^{hl} p_h + \sum_{j=m-c}^{m-1} \omega^{(j+c)l} p_{j+c-m} \right). \quad (\text{B-14})$$

Fazendo $h = j - m + c$ no somatório $\sum_{j=m-c}^{m-1}$:

$$\hat{p}_l^c = \omega^{-lc} \left(\sum_{h=c}^{m-1} \omega^{hl} p_h + \sum_{h=0}^{c-1} \omega^{(h+m)l} p_h \right) = \quad (\text{B-15})$$

¹Nesse teorema e na sua prova, o subscrito c em um vetor indica rotação e não exponenciação.

$$\omega^{-lc} \left(\sum_{h=c}^{m-1} \omega^{hl} p_h + \omega^{ml} \sum_{h=0}^{c-1} \omega^{hl} p_h \right). \quad (\text{B-16})$$

Como $\omega^{ml} = 1$ então:

$$\begin{aligned} \hat{p}_l^c &= \omega^{-lc} \left(\sum_{h=c}^{m-1} \omega^{hl} p_h + \sum_{h=0}^{c-1} \omega^{hl} p_h \right) = \\ &\omega^{-lc} \sum_{h=0}^{m-1} \omega^{hl} p_h = \omega^{-lc} \hat{p}_l. \end{aligned} \quad (\text{B-17})$$

□

Apêndice do capítulo 5

Teorema C.1. *Dividamos \mathbb{R}^2 em dois semiplanos: acima e abaixo da reta $p_2 = p_1$. Ou seja, o semiplano 1 é*

$$(p_1, p_2) \in \mathbb{R}^2 \mid p_2 \leq p_1 \quad (\text{C-1})$$

e o semiplano 2 é

$$(p_1, p_2) \in \mathbb{R}^2 \mid p_2 \geq p_1 \quad (\text{C-2})$$

A intersecção dos dois semiplanos é a reta $p_1 = p_2$. Dado $q = (q_1, q_2) \in \mathbb{R}^2$, o ponto de $C^(p)$ mais próximo de q é aquele que estiver no mesmo semiplano que q .*

Prova. Sejam

$$d_1 := d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \quad (\text{C-3})$$

$$d_2 := d(Cp, q) = \sqrt{(q_1 - p_2)^2 + (q_2 - p_1)^2} \quad (\text{C-4})$$

Pela observação 3.5 podemos comparar d_1^2 com d_2^2 pois o mínimo dos quadrado corresponde ao quadrado do mínimo.

$$d_2 \geq d_1 \iff d_2^2 \geq d_1^2 \iff d_2^2 - d_1^2 \geq 0 \quad (\text{C-5})$$

Temos

$$\begin{aligned} d_2^2 - d_1^2 &= (q_1 - p_2)^2 + (q_2 - p_1)^2 - ((q_1 - p_1)^2 + (q_2 - p_2)^2) = \\ &= q_1^2 - 2q_1p_2 + p_2^2 + q_2^2 - 2q_2p_1 + p_1^2 - (q_1^2 - 2q_1p_1 + p_1^2 + q_2^2 - 2q_2p_2 + p_2^2) = \\ &= 2(q_1p_1 + q_2p_2 - q_1p_2 - q_2p_1) \end{aligned} \quad (\text{C-6})$$

logo

$$d_2 \geq d_1 \iff 2(q_1p_1 + q_2p_2 - q_1p_2 - q_2p_1) \geq 0 \quad (\text{C-7})$$

portanto

$$d_2 \geq d_1 \iff q_1p_1 + q_2p_2 - q_1p_2 - q_2p_1 \geq 0. \quad (\text{C-8})$$

	$p_2 < p_1$	$p_2 = p_1$	$p_2 > p_1$
$q_2 < q_1$	V	V	F
$q_2 = q_1$	V	V	V
$q_2 > q_1$	F	V	V

Tabela C.1: Situações em que $d_2 \geq d_1$

	$p_2 < p_1$	$p_2 = p_1$	$p_2 > p_1$
$q_2 < q_1$	V	V	F
$q_2 = q_1$	V	V	V
$q_2 > q_1$	F	V	V

Tabela C.2: Situações em que p e q estão no mesmo semiplano

Temos

$$q_1 p_1 + q_2 p_2 - q_1 p_2 - q_2 p_1 = (q_2 - q_1)(p_2 - p_1) \quad (\text{C-9})$$

Disso e da equação (C-8) segue:

$$d_2 \geq d_1 \iff (q_2 - q_1)(p_2 - p_1) \geq 0. \quad (\text{C-10})$$

Da equação (C-10) segue a tabela C.1. Dessa tabela se vê que $d_2 \geq d_1$ se e somente se p e q estão no mesmo semiplano (tabela C.2).

□

C.1 Em \mathbb{R}^3 , C é uma rotação

A matriz de rotação de um ângulo θ ao longo do vetor unitário $u = (u_x, u_y, u_z)$ é dada por

$$R = \begin{pmatrix} \cos \theta + u_x^2(1 - \cos \theta) & u_x u_y(1 - \cos \theta) - u_z \sin \theta & u_x u_z(1 - \cos \theta) + u_y \sin \theta \\ u_x u_y(1 - \cos \theta) + u_z \sin \theta & \cos \theta + u_y^2(1 - \cos \theta) & u_y u_z(1 - \cos \theta) - u_x \sin \theta \\ u_x u_z(1 - \cos \theta) - u_y \sin \theta & u_y u_z(1 - \cos \theta) + u_x \sin \theta & \cos \theta + u_z^2(1 - \cos \theta) \end{pmatrix} \quad (\text{C-11})$$

[1]

Substituindo $\theta = \frac{-2\pi}{3}$ rad e $u_x = u_y = u_z = \frac{1}{\sqrt{3}}$, temos

$$\cos \theta = -0.5$$

$$\sin \theta = -\frac{\sqrt{3}}{2}$$

$$1 - \cos \theta = 1.5$$

$$u_x^2 = u_y^2 = u_z^2 = u_x u_y = u_x u_z = u_y u_z = \frac{1}{3}$$

$$\begin{aligned} -u_z \sin \theta &= -u_x \sin \theta = -u_y \sin \theta = 0.5 \\ u_y \sin \theta &= u_z \sin \theta = u_x \sin \theta = -0.5. \end{aligned} \quad (\text{C-12})$$

Assim

$$R = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad (\text{C-13})$$

que é a matriz da transformação C .

C.2 A fórmula (5-58) vale também para $R_{2\pi}(\theta_q - \theta_p) \in R_4$

Aqui terminamos o trabalho da seção 5.1.

Lembremos que

$$R_4 = \{-\pi; -\frac{\pi}{3}; \frac{\pi}{3}; \pi\} \quad (\text{C-14})$$

Se $R_{2\pi}(\theta_q - \theta_p) = -\pi$ então o valor calculado para c_{max} pela fórmula (5-58) é

$$c_{max} = (3 - a(-\frac{3}{2})) \mod 3 = 2 \quad (\text{C-15})$$

E para $R_{2\pi}(\theta_q - \theta_p) = -\pi$ temos:

$$\begin{aligned} \cos(\theta_q - \theta_p) &= \cos(-\pi) = -1 \\ \cos(\theta_q - \theta_p + \frac{2\pi}{3}) &= \cos(-\frac{\pi}{3}) = 0.5 \\ \cos(\theta_q - \theta_p + \frac{4\pi}{3}) &= \cos(\frac{\pi}{3}) = 0.5 \end{aligned} \quad (\text{C-16})$$

portanto a fórmula (5-58) satisfaz a equação (5-26).

Se $R_{2\pi}(\theta_q - \theta_p) = \pi$ então o valor calculado para c_{max} pela fórmula (5-58) é

$$c_{max} = (3 - a(\frac{3}{2})) \mod 3 = 1 \quad (\text{C-17})$$

E para $R_{2\pi}(\theta_q - \theta_p) = \pi$ temos:

$$\begin{aligned} \cos(\theta_q - \theta_p) &= \cos \pi = -1 \\ \cos(\theta_q - \theta_p + \frac{2\pi}{3}) &= \cos(\frac{5\pi}{3}) = 0.5 \\ \cos(\theta_q - \theta_p + \frac{4\pi}{3}) &= \cos(\frac{\pi}{3}) = 0.5 \end{aligned} \quad (\text{C-18})$$

portanto a fórmula (5-58) satisfaz a equação (5-26).

Se $R_{2\pi}(\theta_q - \theta_p) = -\frac{\pi}{3}$ então o valor calculado para c_{max} pela fórmula (5-58) é

$$c_{max} = (3 - a(-0.5)) \mod 3 = 0 \quad (C-19)$$

E para $R_{2\pi}(\theta_q - \theta_p) = -\frac{\pi}{3}$ temos:

$$\begin{aligned} \cos(\theta_q - \theta_p) &= \cos(-\frac{\pi}{3}) = 0.5 \\ \cos(\theta_q - \theta_p + \frac{2\pi}{3}) &= \cos(\frac{\pi}{3}) = 0.5 \\ \cos(\theta_q - \theta_p + \frac{4\pi}{3}) &= \cos\pi = -1 \end{aligned} \quad (C-20)$$

portanto a fórmula (5-58) satisfaz a equação (5-26).

Se $R_{2\pi}(\theta_q - \theta_p) = \frac{\pi}{3}$ então o valor calculado para c_{max} pela fórmula (5-58) é

$$c_{max} = (3 - a(0.5)) \mod 3 = 0 \quad (C-21)$$

e para $R_{2\pi}(\theta_q - \theta_p) = \frac{\pi}{3}$ temos:

$$\begin{aligned} \cos(\theta_q - \theta_p) &= \cos(\frac{\pi}{3}) = 0.5 \\ \cos(\theta_q - \theta_p + \frac{2\pi}{3}) &= \cos\pi = -1 \\ \cos(\theta_q - \theta_p + \frac{4\pi}{3}) &= \cos(\frac{5\pi}{3}) = 0.5 \end{aligned} \quad (C-22)$$

portanto a fórmula (5-58) satisfaz a equação (5-26).

C.3 Ortonormalidade da base ordenada \mathcal{B}

Como vimos na seção 5.3.2, todos os vetores da base ordenada \mathcal{B} correspondente à forma canônica de Jordan têm norma 1, por construção. Para verificar que \mathcal{B} é ortonormal, falta verificar que cada par de vetores distintos é ortogonal. Para isso usaremos o

Lema C.2. *Seja $\mathbb{Z}_{\neq 0}$ o conjunto dos inteiros não-nulos. Seja $l \in \mathbb{Z}$ e $m \in \mathbb{Z}_{\neq 0}$. Então para todo $j \in \mathbb{Z}$ vale:*

$$\sum_{l=0}^{m-1} \sin(jl \frac{2\pi}{m}) = 0 \quad (C-23)$$

E para todo $j \in \mathbb{Z}_{\neq 0}$ vale:

$$\sum_{l=0}^{m-1} \cos(jl \frac{2\pi}{m}) = 0 \quad (C-24)$$

Prova. Seja

$$\omega := e^{-i \frac{2\pi}{m}} \quad (C-25)$$

Temos

- $\omega = \cos\left(\frac{2\pi}{m}\right) - i\sin\left(\frac{2\pi}{m}\right)$
- $\omega^\theta = \cos\left(\theta\frac{2\pi}{m}\right) - i\sin\left(\theta\frac{2\pi}{m}\right)$
- $\omega^m = 1$

Seja agora:

$$\sigma(j) := \sum_{l=0}^{m-1} \omega^{jl} \quad (\text{C-26})$$

Temos

$$\begin{aligned} \sum_{l=0}^{m-1} \cos\left(jl\frac{2\pi}{m}\right) &= \sum_{l=0}^{m-1} \operatorname{Re}\{\omega^{jl}\} = \\ \operatorname{Re}\left\{\sum_{l=0}^{m-1} \omega^{jl}\right\} &= \operatorname{Re}\{\sigma(j)\}. \end{aligned} \quad (\text{C-27})$$

Também temos

$$\begin{aligned} \sum_{l=0}^{m-1} \sin\left(jl\frac{2\pi}{m}\right) &= -\sum_{l=0}^{m-1} \operatorname{Im}\{\omega^{jl}\} = \\ -\operatorname{Im}\left\{\sum_{l=0}^{m-1} \omega^{jl}\right\} &= -\operatorname{Im}\{\sigma(j)\}. \end{aligned} \quad (\text{C-28})$$

Portanto basta provar que:

1. $\forall j \in \mathbb{Z}, \operatorname{Im}\{\sigma(j)\} = 0$
2. $\forall j \in \mathbb{Z}_{\neq 0}, \operatorname{Re}\{\sigma(j)\} = 0$

Para $j = 0$:

$$\sigma(0) = \sum_{l=0}^{m-1} \omega^0 = m \quad (\text{C-29})$$

E se $j \in \mathbb{Z}_{\neq 0}$ então $\sigma(j)$ é o somatório de uma progressão geométrica de m termos cujo termo inicial é 1 e cuja razão é $\omega^j \neq 1$. Portanto vale a fórmula tradicional para somatório de progressão geométrica:

$$\sigma(j) = \frac{\omega^m - 1}{\omega - 1} = \frac{1 - 1}{\omega - 1} = 0 \quad (\text{C-30})$$

De (C-30) segue imediatamente (2). E de (C-30) e (C-29) segue (1). \square

Agora provaremos a nulidade do produto interno de qualquer par de vetores distintos de \mathcal{B} .

Lembre que para m ímpar:

$$\mathcal{B} = \{\mathbf{v}_1, \mathbf{v}'_1, \dots, \mathbf{v}_{m'}, \mathbf{v}'_{m'}, \mathbf{v}_1\} \quad (\text{C-31})$$

onde m' é definido pela equação (5-112). E para m par:

$$\mathcal{B} = \{\mathbf{v}_1, \mathbf{v}'_1, \dots, \mathbf{v}_{m'}, \mathbf{v}'_{m'}, v_1, v_{-1}\} \quad (\text{C-32})$$

Teorema C.3. Se $j \neq l$ então $\mathbf{v}_j \cdot \mathbf{v}_l = 0$.

Prova.

$$\mathbf{v}_j \cdot \mathbf{v}_l = \frac{2}{m} \sum_{l=1}^m \cos\left((l-1)j\frac{2\pi}{m}\right) \cos\left((l-1)l\frac{2\pi}{m}\right) = \quad (\text{C-33})$$

Fazendo $o = l - 1$, temos

$$\mathbf{v}_j \cdot \mathbf{v}_l = \frac{2}{m} \sum_{o=0}^{m-1} \cos\left(oj\frac{2\pi}{m}\right) \cos\left(ol\frac{2\pi}{m}\right) \quad (\text{C-34})$$

Sabemos que $\forall p \in \mathbb{R} \forall q \in \mathbb{R}, \cos p \cos q = \frac{\cos(p+q) + \cos(p-q)}{2}$. Portanto

$$\begin{aligned} \mathbf{v}_j \cdot \mathbf{v}_l &= \frac{2}{m} \sum_{o=0}^{m-1} \frac{\cos\left((j+l)o\frac{2\pi}{m}\right) + \cos\left((j-l)o\frac{2\pi}{m}\right)}{2} = \\ &= \frac{1}{m} \sum_{o=0}^{m-1} \left(\cos\left((j+l)o\frac{2\pi}{m}\right) + \cos\left((j-l)o\frac{2\pi}{m}\right) \right) = \\ &= \frac{1}{m} \sum_{o=0}^{m-1} \cos\left((j+l)o\frac{2\pi}{m}\right) + \frac{1}{m} \sum_{o=1}^m \cos\left((j-l)o\frac{2\pi}{m}\right) \end{aligned} \quad (\text{C-35})$$

j e l são naturais tais que $j \neq l$, portanto $j+l \neq 0$ e $j-l \neq 0$. Portanto pelo lema C.2 o valor da expressão C-35 é

$$\frac{1}{m}0 + \frac{1}{m}0 = 0 \quad (\text{C-36})$$

portanto $\mathbf{v}_j \cdot \mathbf{v}_l = 0$. □

Teorema C.4. $v_{\lambda_j} \cdot v'_{\lambda_l} = 0$

Prova.

$$v_{\lambda_j} \cdot v'_{\lambda_l} = \frac{2}{m} \sum_{h=1}^m (\cos\left((h-1)j\frac{2\pi}{m}\right) \sin\left((h-1)l\frac{2\pi}{m}\right)) \quad (\text{C-37})$$

Fazendo $o = h - 1$:

$$v_{\lambda_j} \cdot v'_{\lambda_l} = \frac{2}{m} \sum_{o=0}^{m-1} (\cos\left(oj\frac{2\pi}{m}\right) \sin\left(ol\frac{2\pi}{m}\right)) \quad (\text{C-38})$$

Sabemos que $\forall a, b \in \mathbb{R}, \cos a \sin b = \frac{\sin(b+a) + \sin(b-a)}{2}$ portanto

$$v_{\lambda_j} \cdot v'_{\lambda_l} = \frac{2}{m} \sum_{o=0}^{m-1} \left(\frac{\sin\left(ol\frac{2\pi}{m} + oj\frac{2\pi}{m}\right) + \sin\left(ol\frac{2\pi}{m} - oj\frac{2\pi}{m}\right)}{2} \right) =$$

$$\frac{1}{m} \sum_{o=0}^{m-1} \sin(o(l+j)\frac{2\pi}{m}) + \frac{1}{m} \sum_{o=0}^{m-1} \sin(o(l-j)\frac{2\pi}{m}) \quad (\text{C-39})$$

pelo lema C.2, a expressão acima é 0. \square

Teorema C.5. Para m par, $v_{\lambda_j} \cdot v_{-1} = 0$

Prova.

$$v_{\lambda_j} \cdot v_{-1} = \frac{\sqrt{2}}{m} \sum_{h=1}^m (\cos((h-1)j\frac{2\pi}{m})(-1)^{h+1}) = \quad (\text{C-40})$$

Fazendo $o = h - 1$:

$$\begin{aligned} v_{\lambda_j} \cdot v_{-1} &= \frac{\sqrt{2}}{m} \sum_{o=0}^{m-1} (\cos(oj\frac{2\pi}{m})(-1)^{o+2}) = \\ &= \frac{\sqrt{2}}{m} \sum_{o=0}^{m-1} (\cos(oj\frac{2\pi}{m})(-1)^o) = \\ &= \frac{\sqrt{2}}{m} \sum_{t=0}^{\frac{m}{2}-1} (\cos(2tj\frac{2\pi}{m})(-1)^{2t} + \cos((2t+1)j\frac{2\pi}{m})(-1)^{2t+1}) = \\ &= \frac{\sqrt{2}}{m} \sum_{t=0}^{\frac{m}{2}-1} (\cos(2tj\frac{2\pi}{m}) - \cos((2t+1)j\frac{2\pi}{m})) = \\ &= \frac{\sqrt{2}}{m} \sum_{t=0}^{\frac{m}{2}-1} (\cos(2tj\frac{2\pi}{m}) + \cos((2t+1)j\frac{2\pi}{m}) - 2\cos((2t+1)j\frac{2\pi}{m})) = \\ &= \frac{\sqrt{2}}{m} \sum_{t=0}^{\frac{m}{2}-1} (\cos(2tj\frac{2\pi}{m}) + \cos((2t+1)j\frac{2\pi}{m})) - \frac{2\sqrt{2}}{m} \sum_{t=0}^{\frac{m}{2}-1} \cos((2t+1)j\frac{2\pi}{m}) = \\ &= \frac{\sqrt{2}}{m} \sum_{x=0}^{m-1} (\cos(xj\frac{2\pi}{m}) - \frac{2\sqrt{2}}{m} \sum_{x=0}^{\frac{m}{2}-1} \cos((2x+1)j\frac{2\pi}{m})) \end{aligned} \quad (\text{C-41})$$

$j \in \mathbb{Z}_{\neq 0}$, portanto pelo lema C.2 temos

$$\frac{\sqrt{2}}{m} \sum_{x=0}^{m-1} (\cos(xj\frac{2\pi}{m})) = 0 \quad (\text{C-42})$$

Portanto

$$\begin{aligned} v_{\lambda_j} \cdot v_{-1} &= -\frac{2\sqrt{2}}{m} \sum_{x=0}^{\frac{m}{2}-1} \cos((2x+1)j\frac{2\pi}{m}) = \\ &= -\frac{2\sqrt{2}}{m} \sum_{x=0}^{\frac{m}{2}-1} \text{Re}\{e^{i((2x+1)j\frac{2\pi}{m})}\} = \end{aligned}$$

$$-\frac{2\sqrt{2}}{m} \operatorname{Re}\left\{ \sum_{x=0}^{\frac{m}{2}-1} e^{i((2x+1)j\frac{2\pi}{m})} \right\} \quad (\text{C-43})$$

Temos $j \neq 0$, e o somatório acima é o somatório de uma progressão geométrica de termo inicial $e^{i(j\frac{2\pi}{m})}$, $\frac{m}{2}$ termos e razão $e^{i(2j\frac{2\pi}{m})} \neq 1$. Portanto

$$\begin{aligned} \sum_{x=0}^{\frac{m}{2}-1} e^{i((2x+1)j\frac{2\pi}{m})} &= \\ e^{i(j\frac{2\pi}{m})} \frac{(e^{i(2j\frac{2\pi}{m})})^{\frac{m}{2}} - 1}{e^{i(2j\frac{2\pi}{m})} - 1} &= \\ e^{i(j\frac{2\pi}{m})} \frac{(e^{i(j2\pi)}) - 1}{e^{i(2j\frac{2\pi}{m})} - 1} &= \\ e^{i(j\frac{2\pi}{m})} \frac{1 - 1}{e^{i(2j\frac{2\pi}{m})} - 1} &= 0 \end{aligned} \quad (\text{C-44})$$

Portanto $v_{\lambda_j} \cdot v_{-1} = -\frac{2\sqrt{2}}{m} \operatorname{Re}\{0\} = 0$. □

Teorema C.6. $v_{\lambda_j} \cdot v_1 = 0$

Prova.

$$v_{\lambda_j} \cdot v_1 = \frac{\sqrt{2}}{m} \sum_{h=1}^m \cos((h-1)j\frac{2\pi}{m}) = \quad (\text{C-45})$$

Fazendo $n = h - 1$, temos

$$v_{\lambda_j} \cdot v_1 = \frac{\sqrt{2}}{m} \sum_{n=0}^{m-1} \cos(nj\frac{2\pi}{m}) = \quad (\text{C-46})$$

$j \in \mathbb{Z}_{\neq 0}$ portanto, pelo lema C.2

$$\sum_{n=0}^{m-1} \cos(nj\frac{2\pi}{m}) = 0. \quad (\text{C-47})$$

□

Teorema C.7. Se $j \neq l$ então $v'_{\lambda_j} \cdot v'_{\lambda_l} = 0$

Prova.

$$v'_{\lambda_j} \cdot v'_{\lambda_l} = \frac{2}{m} \sum_{h=1}^m (\sin((h-1)j\frac{2\pi}{m}))(\sin((h-1)l\frac{2\pi}{m})) \quad (\text{C-48})$$

Fazendo $n = h - 1$ temos

$$v'_{\lambda_j} \cdot v'_{\lambda_l} = \frac{2}{m} \sum_{n=0}^{m-1} (\sin(nj \frac{2\pi}{m})) (\sin(nl \frac{2\pi}{m})) \quad (\text{C-49})$$

Para todo $\theta \in \mathbb{R}$ temos

$$\begin{aligned} \cos(a - b) - \cos(a + b) &= \\ (\cos a \cos b + \sin a \sin b) - (\cos a \cos b - \sin a \sin b) &= \\ 2 \sin a \sin b \end{aligned} \quad (\text{C-50})$$

portanto

$$\sin a \sin b = \frac{1}{2} (\cos(a - b) - \cos(a + b)). \quad (\text{C-51})$$

Combinando essa igualdade com a equação (C-49) temos

$$\begin{aligned} v'_{\lambda_j} \cdot v'_{\lambda_l} &= \frac{2}{m} \sum_{n=0}^{m-1} \frac{1}{2} (\cos(n(j-l) \frac{2\pi}{m}) - \cos(n(j+l) \frac{2\pi}{m})) = \\ \frac{1}{m} \sum_{n=0}^{m-1} \cos(n(j-l) \frac{2\pi}{m}) - \frac{1}{m} \sum_{n=0}^{m-1} \cos(n(j+l) \frac{2\pi}{m}) \end{aligned} \quad (\text{C-52})$$

j e l são naturais tais que $j \neq l$, portanto $j - l \in \mathbb{Z}_{\neq 0}$ e $j + l \in \mathbb{Z}_{\neq 0}$. Portanto pelo lema C.2

$$v'_{\lambda_j} \cdot v'_{\lambda_l} = \frac{1}{m} m 0 - \frac{1}{m} m 0 = 0. \quad (\text{C-53})$$

□

Teorema C.8. Para m par, $v'_{\lambda_j} \cdot v_{-1} = 0$

Prova.

$$v'_{\lambda_j} \cdot v_{-1} = \frac{\sqrt{2}}{m} \left(\sum_{h=1}^m \sin((h-1)j \frac{2\pi}{m}) (-1)^{h+1} \right). \quad (\text{C-54})$$

Fazendo $n = h - 1$:

$$\begin{aligned} v_{\lambda_j} \cdot v_{-1} &= \frac{\sqrt{2}}{m} \sum_{n=0}^{m-1} (\sin(nj \frac{2\pi}{m}) (-1)^{n+2}) = \\ \frac{\sqrt{2}}{m} \sum_{n=0}^{m-1} (\sin(nj \frac{2\pi}{m}) (-1)^n) &= \\ \frac{\sqrt{2}}{m} \sum_{o=0}^{\frac{m}{2}-1} (\sin(2oj \frac{2\pi}{m}) (-1)^{2o} + \sin((2o+1)j \frac{2\pi}{m}) (-1)^{2o+1}) &= \end{aligned}$$

$$\begin{aligned}
& \frac{\sqrt{2}}{m} \sum_{o=0}^{\frac{m}{2}-1} (\sin(2oj\frac{2\pi}{m}) - \sin((2o+1)j\frac{2\pi}{m})) = \\
& \frac{\sqrt{2}}{m} \sum_{o=0}^{\frac{m}{2}-1} (\sin(2oj\frac{2\pi}{m}) + \sin((2o+1)j\frac{2\pi}{m}) - 2\sin((2o+1)j\frac{2\pi}{m})) = \\
& \frac{\sqrt{2}}{m} \sum_{o=0}^{\frac{m}{2}-1} (\sin(2oj\frac{2\pi}{m}) + \sin((2o+1)j\frac{2\pi}{m})) - \frac{2\sqrt{2}}{m} \sum_{o=0}^{\frac{m}{2}-1} \sin((2o+1)j\frac{2\pi}{m}) = \\
& \frac{\sqrt{2}}{m} \sum_{g=0}^{m-1} (\sin(gj\frac{2\pi}{m}) - \frac{2\sqrt{2}}{m} \sum_{o=0}^{\frac{m}{2}-1} \sin((2o+1)j\frac{2\pi}{m})) \tag{C-55}
\end{aligned}$$

$j \in \mathbb{Z}_{\neq 0}$, portanto pelo lema C.2 temos

$$\frac{\sqrt{2}}{m} \sum_{g=0}^{m-1} \cos(gj\frac{2\pi}{m}) = 0 \tag{C-56}$$

Portanto

$$\begin{aligned}
v'_{\lambda_j} \cdot v_{-1} &= -\frac{2\sqrt{2}}{m} \sum_{o=0}^{\frac{m}{2}-1} \sin((2o+1)j\frac{2\pi}{m}) = \\
&= -\frac{2\sqrt{2}}{m} \sum_{o=0}^{\frac{m}{2}-1} \text{Im}\{e^{i((2o+1)j\frac{2\pi}{m})}\} = \\
&= -\frac{2\sqrt{2}}{m} \text{Im}\left\{\sum_{o=0}^{\frac{m}{2}-1} e^{i((2o+1)j\frac{2\pi}{m})}\right\} \tag{C-57}
\end{aligned}$$

Temos $j \neq 0$, e o somatório acima é o somatório de uma progressão geométrica de termo inicial $e^{i(j\frac{2\pi}{m})}$, razão $e^{i(2j\frac{2\pi}{m})} \neq 1$ e $\frac{m}{2}$ termos. Portanto

$$\begin{aligned}
& \sum_{o=0}^{\frac{m}{2}-1} e^{i((2o+1)j\frac{2\pi}{m})} = \\
& e^{i(j\frac{2\pi}{m})} \frac{(e^{i(2j\frac{2\pi}{m})})^{\frac{m}{2}} - 1}{e^{i(2j\frac{2\pi}{m})} - 1} = \\
& e^{i(j\frac{2\pi}{m})} \frac{(e^{i(j2\pi)}) - 1}{e^{i(2j\frac{2\pi}{m})} - 1} = \\
& e^{i(j\frac{2\pi}{m})} \frac{1 - 1}{e^{i(2j\frac{2\pi}{m})} - 1} = 0 \tag{C-58}
\end{aligned}$$

Portanto $v_{\lambda_j} \cdot v_{-1} = -\frac{2\sqrt{2}}{m} \text{Im}\{0\} = 0$. □

Teorema C.9. $v'_{\lambda_j} \cdot v_1 = 0$

Prova.

$$v'_{\lambda_j} \cdot v_1 = \frac{\sqrt{2}}{m} \sum_{h=1}^m \sin((h-1)j \frac{2\pi}{m}) \quad (\text{C-59})$$

Fazendo $o = h - 1$, temos

$$v_{\lambda_j} \cdot v_1 = \frac{\sqrt{2}}{m} \sum_{o=0}^{m-1} \sin(oj \frac{2\pi}{m}) \quad (\text{C-60})$$

Pelo lema [C.2](#)

$$\sum_{o=0}^{m-1} \sin(oj \frac{2\pi}{m}) = 0 \quad (\text{C-61})$$

□

Teorema C.10. Para m par, $v_{-1} \cdot v_1 = 0$

Prova.

$$\begin{aligned} v_{-1} \cdot v_1 &= \sum_{j=1}^m \frac{1}{\sqrt{m}} \left(\frac{1}{\sqrt{m}} (-1)^{j+1} \right) = \\ &= \frac{1}{m} \sum_{j=1}^m (-1)^{j+1} = \\ &= \frac{1}{m} (1 - 1 + \dots + 1 - 1) = 0. \end{aligned} \quad (\text{C-62})$$

□