UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Explorando Espaços Métricos Correlacionados em Consultas por Similaridade sobre Bases de Dados Incompletos

Lucas Santiago Rodrigues

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Lucas Santiago Rodrigues

Explorando Espaços Métricos Correlacionados em Consultas por Similaridade sobre Bases de Dados Incompletos

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Caetano Traina Júnior

USP – São Carlos
Setembro de 2021

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

R696e Rodrigues, Lucas Santiago
 Explorando Espaços Métricos Correlacionados em
Consultas por Similaridade sobre Bases de Dados
Incompletos / Lucas Santiago Rodrigues; orientador
Caetano Traina Júnior. -- São Carlos, 2021.
 111 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2021.

 1. Banco de dados relacionais. 2. Consultas por
similaridade. 3. Correlação de espaços métricos. 4.
Dados Faltantes. I. Traina Júnior, Caetano,
orient. II. Título.

Lucas Santiago Rodrigues

Exploiting Correlated Metric Spaces in Similarity Queries over Incomplete Databases

Dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Master in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Caetano Traina Júnior

USP – São Carlos
September 2021

AGRADECIMENTOS

À toda minha família, em especial aos meus pais, Divaldo e Silvana e minha irmã Lara, por todo apoio e incentivo incondicional em todos os momentos e decisões. Obrigado por presenciarem toda a caminhada da minha vida acadêmica até aqui, sei que sempre poderei contar com vocês.

Ao meu orientador, Prof. Dr. Caetano Traina Júnior e sua esposa Prof. Dra. Agma Juci Machado Traina por serem grandes pessoas e pesquisadores em suas áreas de trabalho, proporcionando grandes conhecimentos durante os momentos de reuniões, aulas e discussões realizados durante o período deste Mestrado. Tenho certeza que toda a colaboração de vocês é de grande valor para o meu aprendizado profissional e pessoal.

Todos os membros do grupo de pesquisa de Bases de Dados e Imagens (GBdI) que sempre cultivaram um ambiente pró-discussão de diversos assuntos pertinentes à pesquisa, além de sempre ajudarem durante todo este Mestrado. Em especial, agradeço a contribuição da Mirela Teixeira Cazzolato, que participou ativamente das tarefas deste projeto com suas ideias, conhecimentos, leituras, questionamentos e revisões, participando sempre com muita dedicação e paciência em todos os momentos.

Agradeço aos meus amigos Thábata, Patrícia, José Maria Clementino, André Toshio, Lucas Scabora e Maxwell Sampaio pelos momentos de diversão durante esse tempo, colaborando para fazer caminhada mais agradável. Meu muito obrigado!

Por fim, ao Conselho Nacional de Pesquisa Científica (CNPq)¹, a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP)² pelo apoio financeiro para a execução deste trabalho.

¹Processo: 132698/2019-6.

²Processos: 2016/17078-0, 2020/07200-9, 2020/10902-5.

*“Um pessimista vê uma dificuldade em cada oportunidade,
um otimista vê uma oportunidade em cada dificuldade.”*
(Winston Churchill)

RESUMO

RODRIGUES, L. S. **Explorando Espaços Métricos Correlacionados em Consultas por Similaridade sobre Bases de Dados Incompletos**. 2021. 111 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

O crescente avanço na geração de dados advindos de várias fontes, tornou necessário o desenvolvimento de métodos de apoio aos processos de gerenciamento de grandes quantidades de dados complexos, como imagens, vídeos e áudios. Entretanto, a ocorrência de falhas durante os processos de coleta e armazenamento dos dados devido a diversas causas resultam na incompletude dos bancos de dados, afetando negativamente a execução de consultas por similaridade em inúmeras tarefas de recuperação de informação. Especificamente em consultas por similaridade, as funções de distâncias tradicionais, como a Euclidiana, não medem a dissimilaridade entre pares de atributos com valores faltantes. Abordagens existentes na literatura lidam com o problema de bases de dados incompletos por meio do descarte de tuplas com valores faltantes, a imputação de valores por meio de várias heurísticas e a indexação de bases de dados incompletos. No entanto, a aplicação do descarte de dados pode ocasionar a redução significativa do conjunto de dados, imputações de valores podem introduzir distorções no conjunto de dados e a indexação de dados oferece tratamentos específicos ao tratamento de dados faltante e muitas vezes custoso. O objetivo deste trabalho consiste em extrair informações intrínsecas dos dados para auxiliar na execução de consultas por similaridade sobre bases incompletas, sem o descarte de dados e nem a utilização de um método de imputação de valores. Nesse contexto, este trabalho de mestrado propõe o método SOLID (*Search Over Correlated and Incomplete Data*), que utiliza a correlação entre pares de espaços métricos definidos por um conjunto de representações para gerar fatores de compatibilidade a partir da identificação de atributos complexos mais correlacionados com relação às ocorrências de objetos com valores faltantes. As consultas por similaridade são executadas por meio de uma função de distância, cuja construção inclui propriedades propostas neste trabalho. Ela aplica os fatores de compatibilidade de acordo com o cenário da falta de dados ocorrida e consolida as distâncias resultantes, reduzindo a influência dos dados faltantes. Uma análise experimental realizada com o SOLID mostra que, para diferentes bases de dados de dimensionalidades e cardinalidades distintas, a correlação entre espaços métricos altamente correlacionados pode auxiliar na redução da influência de dados faltantes ao executar consultas por similaridade. O SOLID é mais de 55% mais preciso do que métodos de imputação ao recuperar tuplas sobre bases que podem até mesmo conter grandes quantidades de dados faltantes (50%), além de executar consultas até 100x mais rápido do que seus concorrentes.

Palavras-chave: Dados faltantes, Consultas por similaridade, Correlação de espaços métricos, Dados complexos.

ABSTRACT

RODRIGUES, L. S. **Exploiting Correlated Metric Spaces in Similarity Queries over Incomplete Databases**. 2021. 111 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

The growing advances in data generation from various data sources make it necessary to develop methods to support the management processes of large amounts of complex data, such as images, videos, and audio. However, the occurrence of failures during the data collection and storage processes leads to incompleteness, which negatively affects the execution of similarity queries in numerous information retrieval tasks. Specifically, in similarity queries, traditional distance functions, such as Euclidean, do not measure the dissimilarity between pairs of attributes with missing values. Traditional approaches from literature deal with incomplete databases by discarding tuples with missing values, imputing values using several heuristics, and indexing incomplete databases. However, the application of data deletion can cause a significant reduction of the dataset, and imputations of values can introduce distortions in the dataset. This work aims to extract intrinsic information from the data to help execute similarity queries on incomplete databases without discarding data or using a value imputation method. In this context, this master's dissertation proposes the SOLID (*Search Over Correlated and Incomplete Data*) method, which uses the correlation between pairs of metric spaces defined by a set of data representations to generate compatibility factors from identifying complex attributes correlated concerning occurrences of objects with missing values. Similarity queries are performed when the distance function proposed in this work is employed, which applies the compatibility factors according to the missing data scenario and consolidates the resulting distances, reducing the missing data's influence. Experimental analysis performed with SOLID shows that, for several databases, the correlation between highly correlated metric spaces can reduce the influence of missing data when executing similarity queries. Thus, SOLID is more than 55% better than imputation methods in accurately retrieving tuples over databases even with large amounts of missing data, in addition to executing queries by up to 100x faster than SOLID's competitors.

Keywords: Missing Data, Similarity Searches, Metric spaces correlation, Complex data.

LISTA DE ILUSTRAÇÕES

Figura 1 – Consultas por similaridade com a comparação entre vetores de características.	26
Figura 2 – Problema na execução de consultas por similaridade em bases incompletas.	28
Figura 3 – Aplicação de operadores de comparação em imagens.	33
Figura 4 – Representação de um CBIR para consultas por similaridade.	34
Figura 5 – Extração de características por um histograma de cor RGB.	35
Figura 6 – Representação do espaço das métricas L_p .	37
Figura 7 – Representação de uma R_q , dado um centro de busca s_q e um raio de distância ε .	40
Figura 8 – Execução de uma R_q sobre uma imagem e $\varepsilon = 50m$.	41
Figura 9 – Representação de uma kNN_q com $k = 5$ objetos mais próximos.	41
Figura 10 – Execução de uma kNN_q sobre uma imagem e $k = 3$.	42
Figura 11 – Exemplos de ocorrências de dados faltantes.	47
Figura 12 – Problema de consultas por similaridade sobre objetos com atributos faltantes.	49
Figura 13 – Comparação entre a base incompleta e a base de dados com registros faltantes excluídos.	50
Figura 14 – Consultas por similaridade com tratamento de descarte de tuplas.	52
Figura 15 – Arquitetura de um <i>autoencoder</i> com três camadas ocultas.	55
Figura 16 – Consultas por similaridade com tratamento de imputação de valores.	56
Figura 17 – Sequência de processos do método SOLID.	63
Figura 18 – Base de dados complexos constituída de diferentes exames médicos.	64
Figura 19 – Definição do conjunto de espaços métrico $\mathfrak{M}$.	65
Figura 20 – Geração do mapa de correlação.	68
Figura 21 – Aplicação da função de compatibilidade.	69
Figura 22 – Associação de diâmetro para cada espaço métrico $\mathbb{M}_i$.	72
Figura 23 – Solid-Dist em consultas por similaridade.	76
Figura 24 – Quantidade de representações possíveis.	80
Figura 25 – Seleção de representações utilizando o método CorDiS sobre o conjunto <i>DS-Spine</i> .	81
Figura 26 – Mapa de correlação extraído do conjunto de dados <i>DS-Spine</i> .	82
Figura 27 – Mapa de correlação extraído do conjunto de dados <i>DS-Letters</i> .	83
Figura 28 – Atributos complexos compatíveis gerados por faixas de correlação.	84
Figura 29 – Comparativo entre limiares de correlação em relação a efetividade em consultas por vizinhança.	85

Figura 30 – Intervalo de valores de distância para cada FEM utilizado, quando aplicada a função Euclidiana para os conjuntos <i>DS-Spine</i> e <i>DS-HandPD</i>	87
Figura 31 – Consultas por Vizinhaça considerando centros de consultas incompletos	88
Figura 32 – Consultas por Abrangência considerando centros de consultas incompletos	91
Figura 33 – Comparativo do erro do SOLID para consultas <i>kNN</i>	93
Figura 34 – Comparativo do erro do SOLID para consultas R_q	94

LISTA DE ALGORITMOS

Algoritmo 1 – CorDiS: Mapeamento de Espaços Métricos Correlacionados	67
Algoritmo 2 – SOLID (Fatores de Compatibilidade Consolidados de uma Tupla)	70

LISTA DE TABELAS

Tabela 1	–	Visão Geral de abordagens selecionadas para tratamento de dados faltantes.	58
Tabela 2	–	Símbolos empregados na abordagem SOLID.	62
Tabela 3	–	FEMs utilizados nos experimentos.	78
Tabela 4	–	Bases de dados utilizadas nos experimentos do SOLID.	78
Tabela 5	–	Consolidação de ganhos médios e máximos do SOLID executando consultas por vizinhança (kNN_q) com centros de consulta incompletos.	89
Tabela 6	–	Consolidação de ganhos médios e máximos do SOLID executando consultas por abrangência (R_q) com centros de consulta incompletos.	92
Tabela 7	–	Tempo médio de execução dos métodos em consultas por vizinhança (kNN_q).	95
Tabela 8	–	Tempo médio de execução dos métodos em consultas por abrangência (R_q).	96

LISTA DE ABREVIATURAS E SIGLAS

L_p	Métricas Minkowski
CorDiS	<i>Correlation of Distance Spaces</i>
SGBDRs	Sistemas de Gerenciamento de Bases de Dados Relacionais
SGBDs	Sistemas de Gerenciamento de Bases de Dados
SOLID	<i>Search Over Correlated and Incomplete Data</i>
FEM	<i>Feature Extraction Methods</i>
MAM	Métodos de Acesso Métrico
RC	Relação de Continência
RI	Relação de Identidade
RO	Relação de Ordem
RS	Relação de Similaridade
SQL	<i>Structured Query Language</i>

LISTA DE SÍMBOLOS

$\mathcal{D}$ — Conjuntos de Dados

n — Cardinalidade de $\mathcal{D}$

d — Dimensionalidade de $\mathcal{D}$

T — Relação em um SGBDR

A — Atributo de uma relação T

$\mathbb{A}$ — Domínio do atributo A

$\mathbb{S}$ — Domínio dos dados complexos

s — Objeto Complexo

v — Vetor de Características

δ — Função de Distância

$\mathbb{M}$ — Espaço Métrico

$\mathfrak{M}$ — Conjunto de Espaços Métricos

s_q — Centro de busca

R_q — *Range Queries*

ε — Raio de Distância de uma R_q

kNN_q — *k-Nearest Neighbor Queries*

k — Quantidade de elementos a recuperar em kNN_q

σ — Operador de Seleção por Similaridade

θ — Operador de Comparação por Similaridade

ϕ — Função de Correlação

ψ — Função de Distância Solid-Dist

F — Conjunto de Extratores de Características

G — Conjunto de Funções de Distância

$mCorr$ — Mapa de Correlação

$\mathbb{A}$ — Distâncias Médias

γ — Limiar de Compatibilidade

ω — Fator de Compatibilidade

Ω — Conjunto de ω

ϖ — Fator de Compatibilidade Consolidado

Υ — Conjunto de ω Consolidados

β — Diâmetro de Normalização de Distâncias

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Motivação	25
1.2	Definição do Problema	27
1.3	Contribuições do Trabalho	28
1.4	Organização da Dissertação	29
2	CONSULTAS POR SIMILARIDADE	31
2.1	Considerações Iniciais	31
2.2	Recuperação de Informação em SGBDRs	32
2.2.1	<i>Dados Complexos em SGBDRs</i>	33
2.2.2	<i>Extração de Características</i>	34
2.3	Funções de Distâncias	35
2.4	Espaços Métricos	37
2.4.1	<i>Métodos de Acesso Métrico - MAMs</i>	38
2.5	Consultas por Similaridade em SGBDRs	39
2.5.1	<i>Consultas por Abrangência</i>	39
2.5.2	<i>Consultas por Vizinhança</i>	40
2.5.3	<i>Operadores de Seleção por Similaridade em SGBDRs</i>	41
2.6	Correlação de Dados	42
2.7	Considerações Finais	43
3	DADOS FALTANTES E TRABALHOS RELACIONADOS	45
3.1	Considerações Iniciais	45
3.2	Teoria de Dados Faltantes	45
3.2.1	<i>Mecanismos de Dados Faltantes</i>	47
3.3	Consultas por Similaridade em Bases Incompletas	48
3.4	Trabalhos Relacionados	50
3.4.1	<i>Descarte de Dados</i>	51
3.4.2	<i>Imputação de Valores</i>	52
3.4.3	<i>Indexação de Bases de Dados Incompletas</i>	56
3.4.4	<i>Discussão de Trabalhos Relacionados</i>	58
3.5	Considerações Finais	60

4	SOLID: BUSCAS SOBRE DADOS CORRELACIONADOS E IN-COMPLETOS	61
4.1	Considerações Iniciais	61
4.2	O Método SOLID	62
4.3	Processamento de Conjuntos de Dados Complexos	63
4.4	CorDiS: Correlações entre Espaços Métricos	65
4.5	Geração de Fatores de Compatibilidade	67
4.6	Consultas por Similaridade com o SOLID	70
4.6.1	<i>Normalização de Distâncias entre Espaços Métricos Distintos</i>	72
4.6.2	<i>Consultas por Similaridade SOLID: Cenários de aplicação</i>	73
4.6.3	<i>Exemplo de Consultas por Similaridade com o SOLID</i>	74
4.7	Considerações Finais	75
5	EXPERIMENTOS E RESULTADOS	77
5.1	Considerações Iniciais	77
5.2	Materiais e Métodos	78
5.3	Correlações entre Espaços Métricos	79
5.3.1	<i>Seleção de Representações Correlacionadas</i>	80
5.3.2	<i>Compatibilidade entre Atributos Complexos</i>	81
5.4	Análise Quantitativa em Consultas por Similaridade	85
5.4.1	<i>Cenário I: Centro de Consulta Incompleto</i>	87
5.4.2	<i>Cenário II: Bases Incompletas</i>	92
5.5	Análise de Tempo em Execução de Consultas	95
5.6	Discussão de Resultados	96
5.7	Considerações Finais	97
6	CONCLUSÕES	99
6.1	Considerações Iniciais	99
6.2	Principais Contribuições	100
6.3	Trabalhos Publicados	101
6.3.1	<i>Publicação Principal</i>	101
6.3.2	<i>Publicações em colaboração</i>	101
6.4	Trabalhos Futuros	103
	REFERÊNCIAS	105

INTRODUÇÃO

1.1 Motivação

O constante avanço dos sistemas de informação em tarefas de coleta e compartilhamento de dados, combinados com a disseminação de redes sociais, aumentam substancialmente a quantidade de dados disponíveis. Com isso, é cada vez mais comum a necessidade de gerenciamento de **dados complexos** como imagens, vídeos, áudios, dados georreferenciados, dados genéticos e séries temporais em Sistemas de Gerenciamento de Base de Dados (SGBDs). Os SGBDs entregam confiança e eficiência ao gerenciamento de inúmeros tipos de dados e em grandes quantidades, sendo amplamente aplicados em várias áreas da atividade humana além da computação, como medicina, aviação e atividades bancárias. A partir da Teoria Relacional, os SGBDs foram implementados, resultando nos Sistemas de Gerenciamento de Base de Dados Relacionais (SGBDRs) ([SILBERSCHATZ; KORTH; SUDARSHAN, 2020](#); [CODD, 2002](#)).

Os primeiros SGBDRs foram desenvolvidos para armazenar, manipular e recuperar dados, como valores numéricos, datas e pequenas cadeias de caracteres, conhecidos por **dados escalares**. A recuperação de dados em SGBDRs é baseada em relações, sendo uma relação composta por um conjunto de tuplas (registros) e cada tupla tem uma lista de atributos ([SILBERSCHATZ; KORTH; SUDARSHAN, 2020](#); [DATE, 2012](#)). Tradicionalmente, a recuperação de dados escalares é realizada a partir de operadores de comparação baseados em **Relações de Identidade (RI)** e **Ordem (RO)**. Entretanto, para dados complexos, os operadores baseados em RI geralmente não são adequados e RO usualmente não são aplicáveis, uma vez que possuem pouca utilidade prática sobre a maioria dos domínios de dados complexos existentes ([TRAINA-JR. et al., 2019](#)). Por exemplo, a consulta “*Imagem A é maior ou igual a Imagem B?*” mostra a inviabilidade da aplicação dos operadores de RO na recuperação de dados complexos (imagens) em SGBDRs. Portanto, a recuperação de dados complexos geralmente não utiliza operadores baseados em RI e RO.

A recuperação de dados complexos é frequentemente realizada a partir de comparações baseadas em **Relações de Similaridade - RS**, e são amplamente exploradas em diferentes tipos de domínios e em diversos trabalhos de **Consultas por Similaridade** sobre dados complexos armazenados em SGBDRs (RODRIGUES *et al.*, 2020; ZABOT *et al.*, 2019; CAZZOLATO *et al.*, 2019a; VASCONCELOS *et al.*, 2018). Tradicionalmente, consultas por similaridade executam comparações entre pares de representações extraídas sobre um conjunto de dados complexos, sobre os quais são frequentemente aplicados extratores de características, resultando em vetores de características. Com isso, é possível medir a dissimilaridade entre um par de vetores de características usando uma função de distância, para a qual quanto menor o valor resultante da comparação, mais similares são os objetos. Existem vários extratores de características apropriados ao contexto de imagens, como também várias funções de distâncias adequadas ao contexto de vetores de características, como as funções da família Minkowsky (TRAINA-JR. *et al.*, 2019; ZEZULA; BATKO; DOHNAL, 2018). A execução de uma consulta de seleção necessita de um elemento central a ser comparado e um critério de seleção. Dessa forma, a função de distância escolhida mede a dissimilaridade do elemento central para todos os elementos do conjunto dados, resultando na seleção dos elementos mais similares, de acordo com o critério escolhido. A Figura 1 ilustra o processamento de dados complexos para execução de consultas por similaridade.

Figura 1 – Execução de consultas por similaridade a partir da comparação entre vetores de características, com o cálculo da dissimilaridade por uma função de distância.

Fonte: Elaborada pelo autor.

Consultas por Similaridade são amplamente exploradas no contexto de recuperação de informação sobre imagens. Existem vários sistemas baseados em **Recuperação de Imagens por Conteúdo, do inglês Content-Based Image Retrieval (CBIR)**, que utilizam diversas representações obtidas a partir da aplicação de um conjunto de extratores de características, a fim de recuperar as imagens mais similares a partir de um determinado centro de busca (CHINO *et al.*, 2020; CAZZOLATO *et al.*, 2019b).

Com o grande volume de dados gerados por diversas fontes, a ocorrência da falta de dados em processos de gerenciamento dos dados também se tornou mais frequente, resultando em bases incompletas. Este problema é conhecido por **Dados Faltantes**, do inglês *Missing Data* (RODRIGUES *et al.*, 2020; VENUGOPALAN *et al.*, 2019; LITTLE; RUBIN, 2019). Dados faltantes podem ocorrer a partir de muitas situações durante os processos de coleta, tratamento e armazenamento em SGBDRs, considerando vários problemas durante a observação de fenômenos, medições, transmissão de dados, organização de registros, formatação e armazenamento (MCNEISH, 2017; BELL *et al.*, 2014).

Em aplicações médicas, são rotineiros os problemas na coleta e armazenamento dos dados, devido a diversas causas, como erro humano na geração de imagens, mal funcionamento de equipamentos, falha na geração de mídias ou de transmissão, ou mesmo pacientes que perdem uma consulta ou um exame (BRINIS *et al.*, 2019; DALCA *et al.*, 2019). Consequentemente, o resultado é a incompletude dos bancos de dados, o que pode afetar negativamente o processamento, a recuperação e análise de informação, prejudicando a tomada de decisão por parte dos especialistas. Nesse contexto, a execução de consultas por similaridade sobre bases de dados complexos incompletos é prejudicada, sendo este problema o alvo desta Dissertação de Mestrado.

1.2 Definição do Problema

Consultas por similaridade não lidam com tuplas ou atributos com valores faltantes. Portanto, qualquer tupla ou atributo faltante não é considerado, reduzindo drasticamente o volume de dados disponíveis, e afetando negativamente a eficiência das consultas por similaridade (BRINIS *et al.*, 2019; TRAINA *et al.*, 2019). O principal problema é que as funções de distância tradicionais (como a Euclidiana) utilizadas nas consultas não medem a dissimilaridade entre objetos complexos com valores faltantes.

A Figura 2 ilustra uma consulta por similaridade utilizando a função de distância Euclidiana. A base de dados contém imagens de letras do alfabeto de fontes distintas, sendo que algumas podem não estar definidas. Essa base de dados armazena os d atributos com características extraídas das imagens de letras do alfabeto (Aa - Zz) e as tuplas correspondem às n diferentes fontes (por exemplo, Times New Roman, Verdana, Arial) armazenadas.

Nesse cenário, são executadas consultas por similaridade para recuperar as tuplas referentes às fontes mais similares a uma tupla centro de busca. Na prática, uma consulta neste conjunto pode ser representada por: *“Quão similares são as letras das fontes 1 e 2, sendo medida a dissimilaridade pela função euclidiana?”*. Em um cenário na qual as tuplas 1 e 2 são completas, a função Euclidiana calcula a dissimilaridade entre cada par de objetos dos d atributos de cada tupla, resultando em um valor real positivo R^+ . Por outro lado, com a ocorrência de dado faltante, representada na Figura 2 pela letra c da tupla 1, a função de distância não calcula a dissimilaridade entre este par de imagens, resultando em uma indefinição. Portanto, a tupla 1

Figura 2 – Problema na execução de consultas por similaridade em bases incompletas.

tupla	a	b	c	...	z	Atributos Complexos (imagens de letras com diferentes fontes)
1	A	B	???	...	Z	
2	a	b	c	...	z	
...	...	...	...	...	...	
n	<i>A</i>	???	<i>C</i>	...	<i>Z</i>	

Qual a distância entre as tuplas 1 e 2?

$$\delta = \sqrt{\sum_{i=1}^d (a_{1i} - a_{2i})^2 + (b_1 - b_2)^2 + \underbrace{(\text{???} - c_2)^2}_{\text{valor indefinido}} + \dots + (z_1 - z_2)^2}$$

δ não mede a dissimilaridade entre tuplas com valores indefinidos.

Fonte: Adaptada de [Rodrigues et al. \(2020\)](#).

não é considerada no conjunto de resposta desta consulta por similaridade.

Em diversas aplicações, o tratamento de dados faltantes se resume em **descartar as tuplas ou atributos com valores faltantes**, reduzindo consideravelmente os dados disponíveis, e impactando negativamente a qualidade e o desempenho das consultas por similaridade ([CISMONDI et al., 2013](#)). Também existem **métodos de imputação de valores**, com o objetivo de inferir prováveis valores para cada dado faltante, por meio de várias heurísticas como média, regressão, *k*NN, árvores de decisão, entre outras ([BUUREN, 2018](#); [BATISTA; MONARD, 2003](#)). Por fim, foram exploradas **estruturas de indexação** para bases incompletas, propondo tanto heurísticas de indexação em si quanto adaptações em consultas por similaridade para a recuperação de tuplas ([BRINIS et al., 2019](#); [OOI; GOH; TAN, 1998](#)). Porém, a aplicação de métodos de imputação sobre grandes proporções de dados faltantes pode introduzir distorções no conjunto, prejudicando a qualidade das consultas por similaridade. No contexto médico, a aplicação de métodos de imputação de valores pode ser altamente prejudicial, uma vez que a inserção de um valor distorcido ou incorreto em um exame pode ocasionar a redução da qualidade ou mesmo inviabilizar as análises pelos especialistas.

A escolha de métodos eficientes e adequados para lidar com dados faltantes é um desafio relevante no contexto da recuperação de informação ([TRAINA et al., 2019](#); [CISMONDI et al., 2013](#)). Este trabalho de mestrado foca na recuperação de informação executando consultas por similaridade sobre conjuntos de dados complexos incompletos armazenados em SGBDRs. Como premissa, o trabalho aborda o problema de dados faltantes sem a exclusão de dados e sem a imputação valores em bases de dados. Ao invés disso, este projeto propõe uma abordagem baseada na utilização de informações extraídas das bases de dados complexos para auxiliar o cálculo por funções de distâncias adequadas aos domínios dos dados complexos.

1.3 Contribuições do Trabalho

Neste trabalho de mestrado foi concebida uma nova abordagem para recuperar informação executando consultas por similaridade em bases de dados incompletas. O método desenvol-

vido, chamado de SOLID (*Search Over Correlated and Incomplete Data*), utiliza correlações calculadas sobre os dados complexos disponíveis, para viabilizar a execução de consultas por similaridade, sem aplicação de métodos de descarte nem de imputação de valores. Ao desenvolver o método SOLID, buscou-se responder às seguintes questões de pesquisa:

- I - *Como é possível distribuir importância à atributos complexos para reduzir a influência dos dados faltantes, por meio da utilização de informação intrínseca dos conjuntos armazenados em SGBDRs?*
- II - *Como executar consultas por similaridade considerando todos os dados complexos disponíveis, sem a exclusão de elementos (tuplas ou atributos) e sem imputar valores, de forma eficiente e eficaz?*

A solução proposta pelo SOLID tem as premissas de (i) auxiliar funções de distância adequadas aos domínios dos dados complexos sejam calculáveis na execução de consultas por similaridade em base de dados incompletos, e (ii) é possível reduzir a influência de dados faltantes usando a correlação entre dados complexos para possibilitar que as funções de distâncias compensem a falta de dados por meio de pesos nos cálculos de distância sobre os atributos complexos mais correlacionados. O método SOLID tem três contribuições principais:

- I - A definição de CorDiS (*Correlation of Distance Spaces*) que mede as correlações entre diversos espaços de distâncias, e também entre todos os atributos complexos disponíveis na base de dados, resultando em um mapa do espaço de distâncias que pode ser usado para substituir os atributos faltantes quando os espaços estão altamente correlacionados.
- II - A geração de fatores de compatibilidade para os casos de atributos faltantes a partir dos mapas de correlações do CorDiS, possibilitando “quantificar” a influência da falta de dados por meio da ponderação dos atributos mais correlacionados.
- III - A definição da função de distância Solid-Dist (ψ), proposta para aplicação de fatores de compatibilidade entre os atributos mais correlacionados em consultas por similaridade ponderadas, possibilitando a comparação entre tuplas de atributos complexos.

1.4 Organização da Dissertação

Os demais capítulos desta dissertação estão organizados da seguinte forma:

- O [Capítulo 2](#) apresenta os conceitos fundamentais, descrevendo como consultas por similaridade são modeladas e executadas a partir da Teoria dos Espaços Métricos.

- O [Capítulo 3](#) apresenta os conceitos necessários sobre a Teoria dos Dados Faltantes. Além disso, é discutido o problema de consultas por similaridade sobre bases de dados incompletas. Por fim, são apresentados os Trabalhos Relacionados no contexto de recuperação de informação em bases incompletas executando consultas por similaridade.
- O [Capítulo 4](#) detalha a abordagem SOLID para execução de consultas por similaridade em bases de dados incompletas.
- O [Capítulo 5](#) apresenta os experimentos realizados com o SOLID em vários cenários de consultas por similaridade em bases de dados incompletas.
- Finalmente, o [Capítulo 6](#) apresenta as conclusões dessa dissertação e os trabalhos futuros a serem realizados.

CONSULTAS POR SIMILARIDADE

2.1 Considerações Iniciais

A recuperação de informação de dados complexos armazenados em SGBDRs é comumente realizada por meio de consultas por similaridade, uma vez que comparações por identidade geralmente não são adequadas e comparações por ordem não são aplicáveis. Exemplos de dados complexos armazenados em SGBDRs são imagens, áudios e coleções de vídeos, séries temporais, etc. Consultas por similaridade envolvem objetos que são usualmente representados por vetores de características, extraídos a partir da aplicação de métodos de extração de características. Tipicamente, dois vetores são comparados usando uma função de distância, para quantificar a dissimilaridade entre eles. Na literatura existem funções de distâncias métricas e não-métricas, sendo uma métrica uma função que atende às propriedades da Teoria dos Espaços Métricos. Uma vez definidos os extratores e funções de distâncias métricas adequadas, consultas por similaridade são realizadas em espaços métricos.

Neste capítulo, são apresentados os conceitos fundamentais necessários para execução de consultas por similaridade sobre dados complexos armazenados em SGBDRs. Na [Seção 2.2](#) são apresentados os conceitos sobre dados complexos, e como eles são processados, armazenados e recuperados em SGBDRs. A [Seção 2.4](#) apresenta a teoria dos espaços métricos e também estruturas para recuperação por similaridade em espaços métricos de forma eficiente. Na [Seção 2.3](#) são apresentadas as funções de distâncias mais utilizadas em diferentes cenários na execução de consultas por similaridade em SGBDRs. Na [Seção 2.5](#) são definidas as consultas por similaridade a partir dos operadores relacionais existentes. A [Seção 2.6](#) apresenta o conceito de correlação de dados e suas aplicações. Por fim, a [Seção 2.7](#) finaliza este capítulo sobre consultas por similaridade.

2.2 Recuperação de Informação em SGBDRs

A necessidade de armazenar, recuperar e analisar dados originou o surgimento dos **SGBDRs - Sistemas de Gerenciamento de Banco de Dados Relacionais**, dedicados a armazenar dados como valores numéricos, datas e pequenas cadeias de caracteres. Na literatura, esses dados são conhecidos como **dados escalares** (SILBERSCHATZ; KORTH; SUDARSHAN, 2020; CODD, 2002).

Mais recentemente, os SGBDRs são utilizados em diversas áreas da atividade humana, como industrial e médica, com desenvolvimento de softwares bastante robustos (BOEHM, 2020; SASAK-OKON; TUDRUJ, 2020; VYAWAHARE; KARDE; THAKARE, 2018). Inicialmente, foram desenvolvidos SGBDRs a fim de oferecer métodos eficientes de armazenamento e recuperação de dados escalares. Existem vários SGBDRs que são amplamente utilizados no meio acadêmico e comercial, por exemplo o PostgreSQL, Oracle, MySQL, SQLServer, entre outros. Os SGBDRs são fortemente padronizados quanto a sua linguagem de acesso, conhecida como *Strutured Query Language - SQL* (CODD, 2002).

Um SGBDR consiste em uma coleção de relacionamentos, sendo uma relação (T) referente a uma tabela do banco de dados. Cada relação (T) possui uma coleção de n linhas, comumente chamadas de tuplas. Uma tupla é composta por uma sequência (ou lista) de d valores, conhecidos por atributos (A), sendo uma tupla de T representada por $t[T] = \{A_1, A_2, \dots, A_d\}$. Assim, é possível dizer que uma relação é representada por n tuplas, e compostas por d atributos (SILBERSCHATZ; KORTH; SUDARSHAN, 2020).

A recuperação de dados escalares em SGBDRs envolve a utilização de operadores de comparação baseados em relações entre os objetos, como:

- **Relações de Identidade - RI:** representadas pelos operadores $=$ e $\neq$ e aplicados a qualquer tipo de dado;
- **Relações de Ordem - RO:** representadas pelos operadores $<$, $\leq$, $>$ e $\geq$ e aplicados a todos os tipos, que tenham definida uma relação de ordem. Todos os tipos de dados escalares normalmente tratados pelos SGBDRs podem ser comparados com esses operadores;
- **Relações de Continência - RC:** representadas pelos operadores $\subset$, $\subseteq$, $\supset$ e $\supseteq$ e aplicados, por exemplo, sobre cadeia de caracteres, datas e conjuntos em geral.

No contexto de banco de dados, os operadores $=$, $\neq$, $<$, $\leq$, $>$ e $\geq$ são conhecidos como *the Big Six* e são amplamente utilizados como operadores de comparação entre dados escalares (DATE, 2012).

2.2.1 Dados Complexos em SGBDRs

A evolução da tecnologia da informação trouxe a necessidade de organizar, armazenar e recuperar outros tipos de dados, comumente conhecidos como **dados complexos**, por exemplo, dados georreferenciados, imagens, áudios e vídeos, e séries temporais (TRAINA *et al.*, 2019). Usualmente, cada dado complexo possui um domínio dos dados, sendo este representado por $\mathbb{S}$.

A recuperação de informação em coleções de dados complexos é realizada por operadores de comparação baseados em **Relações de Similaridade - RS**. Usualmente, comparar dados complexos utilizando operadores baseados em RI e RO não são adequadas, pois possuem pouca utilidade prática. Na Figura 3 é ilustrado o problema na aplicação de operadores por RI e RO, em que uma consulta nessas imagens não pode ser dito que “A Imagem A é maior igual ou menor que a Imagem B”. Por outro lado, pode ser aplicada uma **Função de Avaliação de Similaridade** para medir a similaridade entre dois objetos complexos, como na Figura 3 pela consulta: ‘A Imagem A é similar a Imagem B?’.

Figura 3 – Aplicação de operadores de comparação em imagens.

Fonte: Elaborada pelo autor.

No contexto de imagens, consultas por similaridade são utilizadas em técnicas de Recuperação de Imagens por Descrição (**DBIR - Description-Based Image Retrieval**), que corresponde à associação de *tags* para cada objeto complexo (UNAR *et al.*, 2019). Além disso, existe a aplicação de consultas por similaridade em técnicas de Recuperação de Informação por Conteúdo (**CBIR - Content-Based Image Retrieval**), que correspondem à associação de vetores de características obtidos a partir da própria imagem, comparação de pares de vetores e recuperar o conteúdo mais similar com base em um critério de busca (CHINO *et al.*, 2018; NESSO *et al.*, 2018). Na Figura 4 é mostrada uma generalização de técnicas baseadas em CBIR, incluindo os elementos: Extratores de Características, Funções de Distância, Métodos de Acesso Métrico e Operadores de Álgebra Relacional em SGBDRs.

Um ambiente para CBIR recupera dados usando a extração de vetores de características sobre imagens, realizadas por **Métodos de Extração de Características ou Feature Extraction Methods - FEMs**, os quais são detalhados a seguir. A comparação entre dois vetores é medida

Figura 4 – Representação de um CBIR para consultas por similaridade.

Fonte: Elaborada pelo autor.

por uma **Função de Distância**, resultando em quão similares são esses vetores, detalhadas na [Seção 2.3](#).

Ao longo desta dissertação é utilizado o símbolo $\mathbb{S}$ para especificar um domínio de dados complexos. Da mesma maneira, são utilizados os símbolos s , S e $\mathcal{S}$ para indicar que um elemento, um atributo ou um domínio de dados é complexo, respectivamente.

2.2.2 Extração de Características

Consultas por similaridade não comparam, por exemplo duas imagens, mas representações dessas imagens, usualmente obtidas por **Métodos de Extração de Características (Feature Extraction Methods - FEMs)**. A aplicação de FEMs tem como objetivo extrair características discriminativas sobre um conjunto de dados complexos, gerando um vetor de características para cada objeto do conjunto. Um vetor de características, frequentemente consiste em um arranjo (*array*) de valores reais, representado por $v = \{x_1, x_2, \dots, x_n\}$. Pode ser dito que cada vetor v é uma representação de um objeto complexo $s_i \in \mathbb{S}$, sendo $\mathbb{S}$ o domínio dos dados complexos.

Na literatura, um dos extratores de características para imagens mais utilizado é o Histograma de Cores, de extração relativamente rápida e construído por meio do agrupamento de cores similares em faixas, conhecidas por *bins*. Na prática, é feita a contagem da frequência de cada intensidade de cor presente no “espaço de cor”, como o RGB, YCbCr ou HSV ([HUGHES et al., 2014](#)). Na [Figura 5](#) ilustra um histograma de cor RGB extraído sobre uma imagem.

Existem vários extratores de características adequados para serem aplicados e representar

Figura 5 – Extração de características por um histograma de cor RGB.

Fonte: Elaborada pelo autor.

imagens por meio de cor, textura ou forma. O padrão MPEG-7 (SALEMBIER; SIKORA; MANJUNATH, 2002) define descritores que incluem diversos extratores de características, como extratores de:

- **Cor:** *Color Layout, Color Structure, Scalable Color, Color Temperature, Dominant Color, Color Correlogram e Group-of-Frame;*
- **Textura:** *Edge Histogram, Texture Browsing e Homogeneous Texture;*
- **Forma:** *Contour Shape, Shape Spectrum e Region Shape.*

Na execução de consultas por similaridade, são comparados pares de vetores de características (v_i, v_j) resultantes da extração sobre um conjunto de dados complexos. Além disso, para medir a similaridade dos pares de vetores é necessário de uma **Função de Distância**, discutida a seguir.

2.3 Funções de Distâncias

Funções de Distâncias - δ medem a distância ou (dis)similaridade entre pares de objetos complexos de um domínio complexo $\mathbb{S}$, representadas por $\delta : \mathbb{S} \times \mathbb{S}$. A execução de uma função δ resulta em um valor real $R^+ \in \mathbb{R}$ (DEZA; DEZA, 2016). Considerando dois objetos complexos

$s_i, s_j \in \mathbb{S}$, quanto menor a distância resultante de $\delta(s_i, s_j)$, maior será a similaridade entre os objetos.

A escolha de uma função δ está associada diretamente à aplicação ou ao tipo de dados utilizado. Consequentemente, existem funções dedicadas à comparação de tipo de dados variados, como dados dimensionais (como os vetores de características), conjuntos, cadeias/*strings* e grafos (DEZA; DEZA, 2016). Alguns exemplos de funções de distâncias mais utilizadas na literatura são apresentados a seguir.

Distâncias Minkowski

As distâncias Minkowski são representadas pela família de funções de distância métricas conhecidas por métricas L_p , que calculam a distância entre dois elementos de um espaço dimensional (ZEZULA, 2016). Considerando a comparação entre dois objetos complexos $s_i, s_j \in \mathbb{S}$, as métricas L_p são definidas genericamente pela Equação 2.1,

$$L_p(s_i, s_j) = \sqrt[p]{\sum_{i=1}^n |s_i - s_j|^p} \quad p \geq 1 \quad (2.1)$$

em que o valor de p determina a métrica utilizada. Mais especificamente, para $p = 1$ é definida a métrica **Manhattan** (L_1), e quando aplicada a um espaço bidimensional, o resultado de valor constante corresponde a um losango. Para $p = 2$ é definida a métrica **Euclidiana** (L_2), sendo o seu resultado representado pela região do espaço bidimensional por um círculo. E por fim, para $p = \infty$ é definida a métrica **Chebyshev** (L_∞), sendo o seu resultado representado pela região do espaço bidimensional por um quadrado. As métricas L_1, L_2 e L_∞ são definidas, respectivamente, por:

- **Manhattan:** $L_1(s_i, s_j) = \sum_{i=1}^n |s_i - s_j|$
- **Euclidiana:** $L_2(s_i, s_j) = \sqrt{\sum_{i=1}^n |s_i - s_j|^2}$
- **Chebyshev:** $L_\infty(s_i, s_j) = \max_{i=1}^n (|s_i - s_j|)$

A Figura 6 ilustra a representação de um espaço bidimensional gerado pela aplicação de cada métrica L_p . As métricas L_p são amplamente utilizadas para medir a distâncias entre objetos multidimensionais, como objetos complexos representados a partir de vetores de características.

Distância de Jaccard

A distância de Jaccard é amplamente aplicada para a comparação entre conjuntos ou multiconjuntos (KOSUB, 2019). É calculada pela razão da intersecção com a união de dois

Figura 6 – Representação do espaço das métricas L_p .

Fonte: Elaborada pelo autor.

conjuntos, resultando em um valor real positivo R^+ . Considerando dois objetos complexos $s_i, s_j \in \mathbb{S}$, a função de distância de Jaccard é definida pela [Equação 2.2](#):

$$J(s_i, s_j) = \frac{|s_i \cup s_j| - |s_i \cap s_j|}{|s_i \cup s_j|} \quad (2.2)$$

Distância Levenshtein

A distância Levenshtein - L_{edit} ([YUJIAN; BO, 2007](#)), conhecida também por distância de edição, é utilizada para medir a distância entre objetos complexos do tipo cadeia de caracteres (*string*). Considerando dois objetos complexos $s_i, s_j \in \mathbb{S}$, é definida por $L_{edit}(s_i, s_j) \geq 0$. A função calcula a quantidade mínima de operações de edição (inserções, exclusões e substituições) para transformar o objeto s_i para o objeto s_j .

Outras Distâncias

Existem outras funções de distâncias amplamente utilizadas em diversos tipos de dados como séries temporais (*times-series*), grafos entre outros. Exemplos dessas funções são: a Canberra, Bray-Curtis, Distância Quadrática, Hamming, DTW (*Dynamic Time Warping*) ([DEZA; DEZA, 2016](#)).

Existem funções de distância **métricas e não-métricas**, sendo uma métrica as que atendem às propriedades dos espaços métricos. A teoria dos espaços métricos e suas propriedades são definidas e detalhadas a seguir.

2.4 Espaços Métricos

As consultas por similaridade são executadas sobre grandes de bases dados, sendo importante que sejam devidamente projetadas para serem escaláveis. As consultas podem ser

modeladas a partir da Teoria dos Espaços Métricos, de acordo com a [Definição 1 \(BRINIS et al., 2019; TRAINA-JR et al., 2002\)](#).

Definição 1. Espaço Métrico - $\mathbb{M}$. Um espaço métrico é definido como um par $\mathbb{M} = \langle \mathbb{S}, \delta \rangle$, em que $\mathbb{S}$ é o domínio dos dados e $\delta : \mathbb{S} \times \mathbb{S} \rightarrow \mathbb{R}^+$ é uma função de distância métrica. Considerando os objetos complexos $s_i, s_j, s_k \in \mathbb{S}$, uma função de distância métrica (δ) deve atender às seguintes propriedades ([BURAGO; BURAGO; IVANOV, 2001](#)):

- **Não-Negatividade:** $\delta(s_i, s_j) \geq 0$
- **Identidade dos Indiscerníveis:** $\delta(s_i, s_j) = 0 \Rightarrow s_i = s_j$
- **Simetria:** $\delta(s_i, s_j) = \delta(s_j, s_i)$
- **Desigualdade Triangular:** $\delta(s_i, s_j) \leq \delta(s_i, s_k) + \delta(s_k, s_j)$

A execução de consultas por similaridade de acordo com as propriedades dos espaços métricos, possibilita a utilização de estruturas de indexação para dados complexos, conhecidas por **Métodos de Acesso Métrico - MAM**. Uma estrutura de indexação tem como objetivo otimizar a organização espacial dos objetos, resultando em melhor desempenho na execução de consultas por similaridade.

2.4.1 Métodos de Acesso Métrico - MAMs

Métodos de Acesso Métrico - MAMs são estruturas de indexação que organizam dados complexos, e frequentemente seguem uma estrutura hierárquica. O objetivo de um MAM é otimizar a organização dos dados complexos, por meio de heurísticas de particionamento do espaço dos dados, seguindo as propriedades dos Espaços Métricos. Com isso, é possível reduzir a quantidade de cálculos de distância entre os objetos, e também o número de acessos à memória secundária para buscar os objetos na estrutura ([BRINIS et al., 2019; TRAINA-JR. et al., 2019](#)). Quanto mais eficiente é o armazenamento de dados complexos nos SGBDRs, maior o desempenho das consultas por similaridade na recuperação de informação sobre dados complexos.

Na literatura existem vários MAMs, com inúmeras heurísticas de organização distintas, e desenvolvidas para a indexação de objetos complexos. A princípio, existem duas categorias: Estáticos e Dinâmicos.

MAMs estáticos tem como característica principal não suportar atualizações, ou seja, caso tenha alguma edição de objetos na estrutura, o índice é reconstruído totalmente. Exemplos de MAMs estáticos são a *GH-Tree* ([UHLMANN, 1991](#)), *VP-Tree* ([YIANILOS, 1993](#)) e *MVP-Tree* ([BOZKAYA; OZSOYOGLU, 1997](#)).

Por outro lado, **MAMs Dinâmicos** permitem atualizações, ou seja, a cada edição a estrutura é organizada sem precisar de uma nova reconstrução. Alguns exemplos de MAMs dinâmicos são a *M-Tree* (CIACCIA; PATELLA; ZEZULA, 1997), sendo esta a primeira estrutura de indexação dinâmica descrita na literatura, a *PM-Tree* (SKOPAL; POKORNÝ; SNÁSEL, 2004) como uma extensão da *M-Tree* com a inclusão de *pivots*. A *Slim-Tree* (TRAINA-JR et al., 2002) é uma extensão da *M-Tree* proposta para lidar com o problema de sobreposição de nós, apresentando heurísticas para medir e reduzir a sobreposição entre nós, melhorando o desempenho das consultas por similaridade. Mais recentemente, a *Hollow-Tree* (BRINIS et al., 2019) foi desenvolvida para a indexação de dados complexos armazenados em bases incompletas. Ela propõe heurísticas para indexar dados complexos incompletos mantendo a estrutura balanceada. Além disso, a *Hollow-Tree* define consultas por similaridade adaptadas para recuperar elementos incompletos junto a elementos completos no conjunto de resposta.

Nesta seção foram apresentados alguns exemplos de MAMs para otimizar a armazenagem e recuperação de dados complexos em SGBDRs. Neste trabalho de mestrado, o ponto principal é como recuperar dados complexos armazenados em base de dados incompletos. Dessa forma, o uso de MAMs para otimizar o armazenamento e recuperação se torna uma opção viável quando associados a métodos que lidem com dados faltantes em SGBDRs.

2.5 Consultas por Similaridade em SGBDRs

A execução de consultas por similaridade em SGBDRs é realizada a partir da definição de funções de distâncias (δ) e extratores de características adequados ao domínio $\mathbb{S}$ dos dados complexos. Com isso, a comparação entre pares de objetos complexos (s_i, s_j) , medida pela função de distância é representado por $\delta(s_i, s_j)$. Com a associação de um domínio dos dados complexos $\mathbb{S}$ à uma métrica δ , as consultas por similaridade são realizadas em espaços métricos (ver Definição 1).

As consultas por similaridade tem como requisito um objeto como **centro de busca** e um **critério de seleção**. Como resultado da seleção, são retornados os objetos mais similares ao centro de consulta a partir do critério de seleção. Os dois critérios de seleção mais conhecidos para consultas por similaridade são **Abrangência** e **Vizinhança**, definidos e detalhados a seguir.

2.5.1 Consultas por Abrangência

As consultas por **abrangência** ou *Range Queries* (R_q), são executadas em SGBDRs e são definidas formalmente como:

Definição 2. Consulta por Abrangência ou Range Query (R_q). Dado um centro de busca $s_q \in \mathbb{S}$ e um raio de distância ε , uma R_q em um espaço métrico $(\mathbb{S}, \delta)$ recupera cada objeto

$s_i \in \mathbb{S}$, que tem a similaridade para o objeto s_q menor ou igual ao raio ε , representado por $\delta(s_q, s_i) \leq \varepsilon$.

A Figura 7 ilustra a execução de uma **Range Query** (R_q). Considerando cada ponto como um objeto complexo e um centro de busca (ponto amarelo), a R_q retorna todos os pontos (em marrom) que estão a até no máximo à distância do raio ε .

Figura 7 – Representação de uma R_q , dado um centro de busca s_q e um raio de distância ε .

Fonte: Elaborada pelo autor.

Considerando um cenário de dados georreferenciados, um exemplo mais prático da execução de uma R_q é: **“Recupere todos os prédios que estão à distância máxima de 50 metros do Laboratório GBdI”**. A Figura 8 ilustra a R_q com o Lab. GBdI como centro de busca. O conjunto de resposta retorna os prédios “Cantina do ICMC”, “Centro de Tecnologia da Informação”, “Biblioteca Central EESC” e “Biblioteca Achille Bassi - ICMC”.

2.5.2 Consultas por Vizinhança

As consultas por vizinhança ou *k-Nearest Neighbor Queries* (kNN_q), recuperam os k vizinhos mais próximos de um objeto central e são definidas formalmente como:

Definição 3. Consulta por Vizinhança ou *k-Nearest Neighbor Query* (kNN_q). Dado um centro de busca $s_q \in \mathbb{S}$ e um valor inteiro $k > 0$, uma kNN_q em um espaço métrico $(\mathbb{S}, \delta)$ recupera os k objetos $s_i \in \mathbb{S}$ mais próximos a s_q , medido por uma função de distância δ .

A Figura 9 mostra a execução de uma **kNN Query** (kNN_q). Considerando cada ponto como um objeto complexo e um centro de busca (ponto amarelo), a consulta retorna os k pontos (em marrom) mais próximos do centro de busca.

Considerando um cenário de dados georreferenciados, um exemplo prático da aplicação de uma consulta kNN_q é: **“Recupere os três prédios mais próximos do Laboratório GBdI”**. A

Figura 8 – Execução de uma R_q com o Lab. GBdI como centro de busca (s_q) e $\epsilon = 50m$.

Fonte: Elaborada pelo autor.

Figura 9 – Representação de uma kNN_q dado um centro de busca s_q e $k = 5$ objetos mais próximos.

Fonte: Elaborada pelo autor.

Figura 10 ilustra a kNN_q com o Lab. GBdI como centro de busca. O conjunto de resposta retorna os três prédios próximos: “Cantina do ICMC”, “Biblioteca Central EESC” e “Biblioteca Achille Bassi - ICMC”.

2.5.3 Operadores de Seleção por Similaridade em SGBDRs

Na Teoria Relacional, uma seleção por similaridade é definida e executada a partir de um **operador de seleção por similaridade** (σ). O operador σ seleciona as tuplas de uma relação que atendem ao critério de seleção da consulta. Uma seleção por similaridade é expressa por $\sigma_{(S \theta s_q)} T$, onde T é uma relação, S é um objeto complexo de T , θ é um operador de

Figura 10 – Execução de uma kNN_q com o Lab. GBdI como centro de busca (s_q) e $k = 3$.

Fonte: Elaborada pelo autor.

comparação válido no domínio de S e s_q é o centro de consulta (TRAINA-JR. *et al.*, 2019). A operação de seleção retorna os elementos mais similares ao centro de consulta, entre os elementos armazenados nos atributos S da relação T .

A partir do operador de seleção por similaridade (σ), uma R_q é expressa por $\sigma_{(S \ R_q(\delta, \epsilon) \ s_q)} T$, em que recupera as tuplas de T cujo o valor de s_i do atributo S satisfaz a distância de $\delta(s_1, s_2) \leq \epsilon$. Por outro lado, uma kNN_q é expressa por $\sigma_{(S \ kNN_q(\delta, k) \ s_q)} T$, recuperando as tuplas em que o valor de s_i do atributo S é um dos k -valores mais próximos a s_q .

2.6 Correlação de Dados

A correlação de dados pode ser definido como o grau de relacionamento entre duas variáveis, por exemplo dois atributos. Comumente, é aplicado um coeficiente de correlação como medida do grau de relacionamento, ou correlação (r), entre as variáveis, independente das unidades ou termos em que foram originalmente expressas (CROXTON; COWDEN, 1939; RODGERS; NICEWANDER, 1988).

O coeficiente de correlação é amplamente explorado na literatura de Bases de Dados, sendo um número que varia no intervalo de $[+1, -1]$, no qual o sinal indica a linha de relacionamento como positiva ou negativa, enquanto o grau medido indica o quanto relacionadas são as duas variáveis (ASUERO; SAYAGO; GONZÁLEZ, 2006). Em termos práticos, valores negativos são interpretados como “*negativamente correlacionados*” e valores positivos como “*positivamente correlacionados*”. Em relação ao grau de correlação, valores próximos a $+1$ e -1 podem ser interpretados como fortemente correlacionados, enquanto valores próximos a 0 não

apresentam relacionamento entre as variáveis.

O coeficiente mais comum é o **Coeficiente de Correlação de Pearson**, que mede o relacionamento **linear** entre duas variáveis (BENESTY *et al.*, 2009). É calculado a partir da razão da covariância das duas variáveis analisadas, normalizado pela raiz quadrada de suas variâncias. Na prática, dado s_i e s_j como dois atributos, é dividido a covariância de ambos atributos pelo produto de seus desvios-padrão, conforme a [Equação 2.3](#):

$$\text{corr}(s_i, s_j) = \frac{\text{cov}(s_i, s_j)}{\sigma_{s_i} \sigma_{s_j}} \quad (2.3)$$

Outros coeficientes, como **Spearman** e **Kendall**, são conhecidos por Coeficientes de correlação de classificação e medem **relacionamentos não lineares** entre duas variáveis (ASUERO; SAYAGO; GONZÁLEZ, 2006). Por exemplo, conforme uma variável se aumenta, uma outra variável relacionada tende a seguir o mesmo comportamento, mesmo que o comportamento não seja por relacionamento linear. Caso o comportamento entre as variáveis seja oposto, por exemplo uma aumenta e a outra diminui, isso é interpretado como um relacionamento não linear negativo, com valores de correlação menores que zero.

Neste trabalho de Mestrado, a correlação é utilizada para auxiliar consultas por similaridade em bases de dados incompletas. Maiores detalhes serão discutidos no [Capítulo 4](#), relacionado à proposta desta Dissertação.

2.7 Considerações Finais

Neste capítulo foram apresentados os conceitos fundamentais sobre **consultas por similaridade**. Mostrou-se que a representação de dados complexos por meio de vetores de características, associados a uma função de distância métrica, possibilita a execução de consultas por similaridade. Ainda, foi apresentada a teoria dos **Espaços Métricos**, que define as propriedades de uma métrica, e que também propicia meios de executar consultas com eficiência com o auxílio de **Métodos de Acesso Métrico - MAMs**. Por fim, foram definidas formalmente as duas principais consultas por similaridade: **Range Queries** (R_q) e **kNN Queries** (kNN_q).

DADOS FALTANTES E TRABALHOS RELACIONADOS

3.1 Considerações Iniciais

Um dos desafios recentes no contexto da recuperação de dados complexos em SGBDRs é como lidar com bases de dados incompletos, ou seja, parte dos dados com valores faltantes. Dados faltantes são ocasionados por inúmeras situações durante os processos de gerenciamento de dados em SGBDRs. Independente dos motivos da falta de dados, o processamento e a recuperação de informação pode ser afetado negativamente, uma vez que as consultas por similaridade não lidam adequadamente com incompletude dos dados. Vários trabalhos exploram métodos de descarte de dados e imputação de valores para lidar com o problema de dados faltantes. O descarte de dados consiste na exclusão de tuplas incompletas, os métodos de imputação de valores inferem prováveis valores aos dados faltantes.

Neste capítulo, são apresentados os conceitos sobre a Teoria dos Dados Faltantes, com a definição do problema abordado neste trabalho de mestrado e também abordagens existentes na literatura para lidar com dados faltantes. Na [Seção 3.2](#) é apresentada formalmente a Teoria dos Dados Faltantes, os mecanismos e as consequências dos casos de dados faltantes. A [Seção 3.3](#) contextualiza o problema de dados faltantes em consultas por similaridade sobre dados complexos armazenados em SGBDRs. Na [Seção 3.4](#) são apresentados os trabalhos relacionados a este trabalho de mestrado para o tratamento de dados faltantes em bases de dados incompletos. Por fim, a [Seção 3.5](#) conclui este capítulo.

3.2 Teoria de Dados Faltantes

A geração de grandes volumes de dados por diversas atividades tornou mais frequente a ocorrência de falhas em processos de gerenciamento de dados, destacando o problema de

dados faltantes em SGBDRs, sendo este um desafio abordado por vários trabalhos (RAJA; THANGAVEL, 2020; BRINIS *et al.*, 2019; CISMONTI *et al.*, 2013).

Dados faltantes ocorrem em muitas situações durante os processos onde os dados são coletados, tratados e armazenados (TRAINA *et al.*, 2019; CISMONTI *et al.*, 2013). De fato, é possível que problemas na observação de fenômenos, aquisição de medidas em sensores, transmissão, registro, formatação e armazenamento podem causar a perda de dados.

No contexto médico, são rotineiros os problemas na coleta e armazenamento, que podem ter diversas causas, como o erro humano na geração de imagens ou coleta de medidas, mal funcionamento de equipamentos, falha na geração de mídias, falhas de transmissão ou mesmo um paciente que não compareça a uma consulta médica (BRINIS *et al.*, 2019; DALCA *et al.*, 2019). Com isso, os SGBDRs podem apresentar uma quantidade relativamente alta de dados faltantes, afetando negativamente o processamento e recuperação de informação, sendo prejudicial em várias análises por parte dos especialistas.

A ocorrência de dados faltantes é alvo de análises que visam determinar quais os melhores meios para se processar e recuperar informação em bases de dados incompletas. A princípio, existem dois tipos de dados faltantes em SGBDRs (LITTLE; RUBIN, 2019; ALLISON, 2001):

- I - O dado pode ser faltante para alguns, mas não todos os atributos de um objeto, conhecido por **tupla com atributos faltantes**. Caso todos os atributos de uma tupla sejam faltantes, é dito que a tupla é um **registro faltante**.
- II - O dado pode ser faltante para alguns, mas não para todos os dados de um mesmo atributo, conhecido por **atributo com valores faltantes**. Caso todos os valores de um mesmo atributo sejam faltantes, é dito que o atributo em questão é **latente** ou **não-observado**.

Ocorrências de **registros faltantes** usualmente reduzem a quantidade de tuplas recuperadas nos SGBDRs, uma vez que tais registros não são utilizados em análises. Por outro lado, **atributos com valores faltantes** podem afetar negativamente a qualidade dos dados, e consequentemente a eficiência e eficácia das análises de dados.

A Figura 11 ilustra um exemplo de uma base de dados na qual as colunas representam os atributos e as tuplas representam os objetos. Na Figura 11a é mostrado que todos os atributos são observados, ou seja, com valores não nulos para todos os objetos. Por outro lado, na Figura 11b são apresentados os dois tipos comuns de dados faltantes, em que o objeto “Ob₄” é um caso de **registro faltante** e o atributo “A₃” é um caso de **atributo não-observado**.

Em SGBDRs, devido à forte padronização do Modelo Relacional, dados faltantes são representados por NULL (SILBERSCHATZ; KORTH; SUDARSHAN, 2020; DATE, 2012). Porém, uma preocupação recorrente é a falta de padronização na representação de dados faltantes em diversas aplicações ou mesmo linguagens de programação, ocasionando divergências quanto

Figura 11 – Exemplos de ocorrências de dados faltantes.

	A_1	A_2	A_3	A_4	A_5
Obj_1					
Obj_2					
Obj_3					
Obj_4					
Obj_5					

(a) Todos os valores dos atributos são observados para todos os objetos.

	A_1	A_2	A_3	A_4	A_5
Obj_1	?		?		
Obj_2			?		?
Obj_3			?		
Obj_4	?	?	?	?	?
Obj_5			?		

(b) Atributos com valores faltantes em alguns objetos, com registro faltante e atributo não-observado.

Fonte: Elaborada pelo autor.

ao modo de lidar com a incompletude das bases, podendo resultar em erros nas análises de dados (PEARSON, 2006).

3.2.1 Mecanismos de Dados Faltantes

Um ponto importante é a escolha de como lidar com bases de dados com dados faltantes. Considerando que não existe uma solução geral e definitiva, lidar com dados faltantes pode considerar uma abordagem distinta para cada aplicação ou estudo, sendo importante considerar as razões para que os dados sejam faltantes. (BRINIS *et al.*, 2019; CISMONTI *et al.*, 2013). Vários mecanismos foram propostos na literatura para classificar a falta de dados, sendo discutido como a falta pode influenciar na estratégia ao lidar com o conjunto de dados, de acordo com o cenário de falta de dados. Os mecanismos mais importantes para classificar o tipo da falta de dados são (LITTLE; RUBIN, 2019):

Falta Completamente Aleatória

Casos de Falta Completamente Aleatória ou *Missing Completely at Random (MCAR)* são dados que não existem uma relação para cada ocorrência e a falta satisfaz o nível mais alto de aleatoriedade. Este tipo de caso consiste na probabilidade de os valores faltantes não dependerem dos dados observados, ou seja, a possibilidade de falta é a mesma para todos os atributos. Nesses casos, tanto o conjunto de dados observados quanto o de dados faltantes não são tendenciosos, possibilitando a aplicação de qualquer método para lidar com dados faltantes sem enviesar o conjunto de dados. Exemplos de MCAR são geralmente relacionados a erros no processamento ou gestão de dados.

Falta Aleatória

A Falta Aleatória ou **Missing at Random (MAR)** são casos em que a probabilidade de valores faltantes está relacionada aos dados observados, mas não aos dados faltantes. Por exemplo, considere um conjunto de dados composto de imagens de tumor e informações relacionadas, obtidas de pacientes com câncer. O atributo de tamanho do tumor geralmente está ausente em exames de certos tipos de tumor, obtidos na fase inicial da doença. Neste caso, a probabilidade do tamanho do tumor estar ausente depende apenas do tipo de tumor, portanto, este caso é considerado MAR (IBRAHIM; CHU; CHEN, 2012).

Falta Não Aleatória

A Falta Não Aleatória ou **Missing Not at Random (MNAR)** são casos em que a probabilidade da falta está diretamente relacionada aos dados faltantes. Por exemplo, considere que o tamanho do tumor está faltando porque o tumor está em um estágio inicial, dependendo somente do tamanho que é o dado faltante (IBRAHIM; CHU; CHEN, 2012).

3.3 Consultas por Similaridade em Bases Incompletas

A recuperação de informação em SGBDRs é amplamente realizada por meio de consultas por similaridade, mas a ocorrência de dados faltantes em bases de dados impossibilita a execução das consultas. O principal problema consiste em que as funções de distância, essenciais para responder as consultas, não calculam a dissimilaridade entre objetos com valores faltantes. Com isso, ao executar as consultas sobre bases de dados incompletos, as tuplas incompletas não são consideradas.

A Figura 12 ilustra uma base de dados médicos em que alguns exames por imagem são faltantes. Nesta base de dados, os atributos são imagens (ou características extraídas) de exames médicos, como cérebro, coração, pulmão, estômago e olho. Cada tupla é composta por vários atributos complexos, definidos a partir de cada domínio $\mathbb{S}$ dos dados complexos.

Suponha-se que o alvo da execução de consultas por similaridade nesta base de dados é a comparação entre pacientes com exames mais similares, utilizando a função de distância Euclidiana para medir a dissimilaridade dos objetos. A partir da consulta: “*Quão similares são os pacientes 1 e 4?*”, e considerando ambas as tuplas como completas, é executada a comparação de cada par de atributos, como na Equação 3.1:

$$\delta = \sqrt{\sum_{i=1}^d (ce_1, ce_4)^2 + (co_1, co_4)^2 + (Pu_1, Pu_4)^2 + (Ol_1, Ol_4)^2}, d' \geq 0 \quad (3.1)$$

Neste caso, cada comparação resulta em uma distância ≥ 0 , e ao final é possível obter uma distância d' entre as tuplas 1 e 4. Ao invés disso, com a ocorrência de dados faltantes,

Figura 12 – Problema de consultas por similaridade sobre objetos com atributos faltantes.

	Cérebro	Coração	Pulmão	Estômago	Olho
1
2	NULL				NULL
3
4			NULL	NULL
5		NULL			NULL

Qual é a distância entre as tuplas 1 e 4?

$$\delta = \sqrt{\sum_{i=1}^d (Ce_1 - Ce_4)^2 + (Co_1 - Co_4)^2 + \underbrace{(Pu_1 - ???)^2 + (Es_1 - ???)^2}_{\text{indeterminado}} + (Ol_1 - Ol_4)^2}$$

δ não pode medir a distância de tuplas com valores indefinidos.

Fonte: Elaborada pelo autor.

apresentados na [Figura 12](#) pelos atributos de *Pulmão* e *Estômago* da tupla n , a função de distância é impossibilitada de calcular a distância quando existe objeto faltante, resultando em uma indefinição. Portanto, não é possível medir a distância d' entre ambas as tuplas.

Outro problema de dados faltantes é a redução de dados quando se considera somente tuplas sem atributos faltantes em consultas por similaridade em base de dados incompletas. Este problema é relevante no contexto de tratamentos de dados faltantes, pois a redução de grandes quantidades de dados afeta negativamente a recuperação de informação em diversas aplicações.

A [Figura 13](#) ilustra a redução de dados durante uma consulta por similaridade somente sobre tuplas completas. Considerando o cenário em que o paciente 3 é o centro de busca, e utilizada a função Euclidiana, a finalidade é retornar os pacientes com exames mais similares. Na [Figura 13a](#), a base de dados médicos possui alguns atributos com valores faltantes, mas nenhuma tupla com todos os atributos faltantes. Neste cenário, a consulta em questão é impossibilitada de ser executada, uma vez que existem tuplas incompletas. Por outro lado, na [Figura 13b](#) é ilustrado a execução desta consulta, considerando a aplicação do descarte de registros incompletos. Neste novo cenário, foram excluídas as tuplas referentes aos pacientes 2, 4 e 5, possibilitando a execução da consulta em questão, mas retornando somente o paciente 1. Analisando a estratégia aplicada neste último cenário, não considerar pacientes potencialmente similares pode ocasionar a perda de informações úteis em análises de dados, prejudicando os trabalhos dos especialistas ([TRAINA et al., 2019](#); [CISMONDI et al., 2013](#)).

Figura 13 – Comparação entre a base incompleta e a base de dados com registros faltantes excluídos.

	Cérebro	Coração	Pulmão	Estômago	Olho		Cérebro	Coração	Pulmão	Estômago	Olho
1						1
2	NULL				NULL	2	—	—	—	—	—
3						3
4			NULL	NULL		4	—	—	—	—	—
5		NULL			NULL	5	—	—	—	—	—

(a) Base de dados médicos com exames por imagem são (b) Base de dados com a exclusão de registros na execução de consultas por similaridade.

Fonte: Elaborada pelo autor.

Em grandes bases incompletas, a execução de consultas por similaridade na recuperação de informação se torna um desafio, quando são considerados os problemas apresentados ao longo desta seção. Neste projeto de mestrado, a procura de uma solução é abordada a partir da extração de informação adicional, como a correlação entre objetos complexos, a fim de manter a cardinalidade original de dados, sem manipulação da base em casos de dados faltantes.

As principais técnicas relacionadas para lidar com dados faltantes são baseadas em duas abordagens principais: a remoção de dados e a imputação de valores (RAJA; THANGAVEL, 2020; LITTLE; RUBIN, 2019; BATISTA; MONARD, 2003). Além disso, existem alguns trabalhos que lidam com o problema de dados por meio da indexação de bases incompletas por meio de estruturas especializadas para dados faltantes. Algumas abordagens baseadas na exclusão de dados, a imputação de valores e a indexação de dados faltantes serão apresentadas e discutidas a seguir.

3.4 Trabalhos Relacionados

Com a necessidade de recuperar informação sobre bases incompletas, vários trabalhos foram desenvolvidos ao longo dos anos considerando cenários de aplicação e conjuntos de dados diversos. Os primeiros trabalhos desenvolvidos surgiram com estudos para analisar as ocorrências de dados faltantes, e consequentemente, o impacto que tal problema causa ao recuperar os dados (LITTLE; RUBIN, 2019; ALLISON, 2001; DIXON, 1979). Nessa época, surgiram os primeiros métodos para o tratamento de dados faltantes, a princípio representados por duas abordagens: o **Descarte de Dados** e a **Imputação de Valores**. No geral, as abordagens de descarte de dados não provêm heurísticas para indicar o impacto do descarte no conjunto de dados, porém algumas técnicas de descarte mais robustas realizam análises prévias para selecionar os melhores casos a serem excluídos a fim de causar o menor impacto possível no conjunto de dados.

Os métodos imputação de valores são amplamente explorados e desenvolvidos a partir

de diversas heurísticas. Inicialmente, as abordagens de imputação de valores desenvolvidas são baseadas em heurísticas estatísticas, como a imputação por média e regressão linear (VENUGOPALAN *et al.*, 2019; LITTLE; RUBIN, 2019). Além disso, surgiram vários métodos baseados em Aprendizado de Máquina (*Machine Learning*), com a aplicação de heurísticas de árvores de decisão, imputação por vizinhança e agrupamentos de dados (NIKFALAZAR *et al.*, 2020; RAJA; THANGAVEL, 2020; BATISTA; MONARD, 2003). Mais recentemente, são apresentados métodos baseados em Aprendizado Profundo (*Deep Learning*), com a utilização de redes neurais convolucionais, modelo generativos e autoencoders (ABIRI *et al.*, 2019; YOON *et al.*, 2018).

Além de trabalhos baseados no descarte de dados e na imputação de valores, foram propostas algumas **Estruturas de Indexação** para lidar com bases de dados incompletos no contexto de consultas por similaridade (BRINIS *et al.*, 2019; OOI; GOH; TAN, 1998). A seguir são apresentados alguns trabalhos relacionados ao tratamento de dados faltantes por meio de métodos baseados em: descarte de dados, imputação de valores e indexação de bases de dados incompletas.

3.4.1 Descarte de Dados

O descarte de dados em SGBDRs consiste em excluir as tuplas com um ou mais atributos com valores faltantes. Esta abordagem é utilizada em análises estatísticas e comum em várias áreas de aplicação (CISMONDI *et al.*, 2013). A princípio, existem dois tipos de descarte de dados (LITTLE; RUBIN, 2019; BRINIS *et al.*, 2019):

- **Listwise**: remove a(s) tupla(s) para quaisquer casos em que há pelo menos um atributo com valor faltante.
- **Pairwise**: remove a(s) tupla(s) que causam menos impacto no conjunto de dados quando descartadas, a partir de uma análise para cada ocorrência de atributo com valor faltante.

A aplicação de qualquer um dos tipos de descarte de dados pode resultar na considerável redução da cardinalidade da base de dados. O principal impacto é a perda de informação potencialmente útil em análises e em tarefas de recuperação de dados complexos nas consultas por similaridade.

A Figura 14 ilustra a execução de uma consulta por vizinhança (kNN_q) sobre um conjunto de dados. Seja s_q um centro de busca, $k = 5$ a quantidade elementos a serem retornados, cada ponto verde representa uma tupla completa e cada ponto vermelho representa uma tupla com atributos com valores faltantes. A Figura 14a apresenta o cenário de uma consulta por vizinhança considerando que as tuplas incompletas são removidas da base de dados original. Neste cenário hipotético, o conjunto resposta é composto por 5 tuplas, mas não necessariamente são as desejadas. Por outro lado, na Figura 14b é apresentado o cenário considerando todas as

tuplas (completas e incompletas), retornando de fato as k tuplas mais próximas a s_q . Neste caso, as tuplas incompletas (pontos vermelhos) são mais próximas do que outras tuplas escolhidas inicialmente na Figura 14a. O descarte de tuplas incompletas pode gerar distorções no espaço de busca, retornando elementos que originalmente não são desejados no conjunto resposta, prejudicando a eficácia da consulta. Portanto, considerar todo o conjunto de dados na consulta por vizinhança evita distorções no espaço de busca, não alterando o resultado da busca, e com maior corretude na execução da consulta.

Figura 14 – Consultas por similaridade com tratamento de descarte de tuplas.

Fonte: Elaborada pelo autor.

3.4.2 Imputação de Valores

Os trabalhos relacionados apresentados nesta seção estão organizados em três categorias, de acordo com o tipo de heurística que o método utiliza: métodos baseados em modelos Estatísticos, em Aprendizado de Máquina e em Aprendizado Profundo.

Métodos Estatísticos

Os métodos estatísticos foram os primeiros a serem desenvolvidos a fim de reduzir ou solucionar o problema de dados faltantes em bases de dados. O método mais simples é a **Imputação por Média** (LITTLE; RUBIN, 2019), que infere o valor médio dos atributos observados da amostra de dados ou mesmo entre todos os atributos do conjunto de dados. Além disso, este método é um dos mais utilizados em aplicações pois possui custo computacional baixo, por outro lado pode gerar imputações imprecisas, introduzindo distorções no conjunto de dados.

A **Imputação por Hot-Deck** (VENUGOPALAN *et al.*, 2019) envolve o particionamento do conjunto de dados em vários grupos, associando cada atributo incompleto a um determinado

grupo. Dessa forma, o método calcula a média ou moda baseado na amostra de cada grupo gerado, utilizando o valor resultante para inferir o valor faltante.

A **Imputação por Regressão** (ZHANG, 2016) infere valores faltantes a partir da predição de valores baseado em um modelo de regressão linear sobre os dados observados. Além disso, o modelo de regressão pode utilizar a correlação sobre os dados observados para refinar as predições realizadas no processo. A imputação baseada no método *Expectation-Maximization* (EM) (DALCA *et al.*, 2019) consiste em um processo iterativo a partir da distribuição probabilística dos dados para a predição de valores.

A **Imputação Múltipla** (BUUREN, 2018) é uma técnica que gera uma imputação de dados mais sofisticada. Os valores faltantes são imputados k vezes para produzir k conjunto de dados completos. Em seguida, são produzidas estimativas gerais e erros padrão que espelham os dados faltantes do conjunto (LITTLE; RUBIN, 2019). O método gera imputações melhores, com distorção menor no conjunto de dados. Por outro lado, devido as inúmeras iterações necessárias, o custo computacional é alto quando comparado a métodos mais efetivos e recentes, como os métodos baseados em Aprendizado de Máquina (BATISTA; MONARD, 2003; GARCIARENA; SANTANA, 2017; BATISTA; MONARD, 2003).

Métodos baseados em Aprendizado de Máquina

A **Imputação por k NN ou k -Nearest Neighbors Imputation (kNNI)** é o método mais representativo entre os métodos baseados em heurísticas supervisionadas (GARCIARENA; SANTANA, 2017; BATISTA; MONARD, 2003). O k NNI consiste em procurar, entre os dados observados, a quantidade k de vizinhos completos mais próximos de um objeto com valores faltantes. Com isso, um objeto com valores faltantes é inferido a partir do valor médio entre os k vizinhos próximos. Analisando o desempenho do k NNI, o método apresenta limitações quando aplicado sobre bases com altas proporções de dados faltantes e poucos dados observados, uma vez que é baseado em buscar vizinhos mais próximos sobre o conjunto de dados observados. Além disso, o método é bastante sensível ao valor de k vizinhos escolhido, considerando que não existe um valor preciso a ser definido, o que pode impactar o desempenho da imputação de valores.

A imputação de dados faltantes também é realizada por meio de heurísticas não supervisionadas como o **agrupamento**, mais especificamente o *k-means* (KMI) (RAJA; THANGAVEL, 2020) e o *Fuzzy c-means* (FCMI) (ZHANG *et al.*, 2016). O objetivo é a imputação por meio de agrupamento, dividindo o conjunto de dados em grupos ou *clusters*, em que cada grupo tem um representante central ou centroide. Cada *cluster* é representado por um conjunto de objetos (observados e faltantes) similares. Dessa forma, é calculada a distância dos objetos incompletos para o centroide do seu grupo, sendo utilizado os valores dos objetos mais próximos ao centroide para inferir o provável valor faltante ao objeto incompleto.

Existem alguns métodos de imputação baseados em **Árvores de Decisão (DTI)** (NIKFA-

LAZAR *et al.*, 2020; BURGETTE; REITER, 2010). A aplicação de árvores de decisão consiste em organizar os dados em várias folhas, de modo que os registros de cada folha tenham valores próximos. Dessa forma, quando houver casos de valores faltantes em algum dos registros de determinada folha, o grupo de registros da mesma folha se torna a melhor opção para estimar o valor ausente.

Métodos baseados em Aprendizado Profundo

Os métodos de imputação de valores baseados em aprendizado profundo (*deep learning*) são os desenvolvidos mais recentemente. No geral, os métodos são baseados em arquiteturas de redes neurais, e exploram modelos de redes mais especializadas no problema de dados faltantes, como redes adversárias generativas e *autoencoders*.

Em Yoon *et al.* (2018) os autores propõem a imputação de valores utilizando **Redes Adversárias Generativas ou Generative Adversarial Networks (GAN)**. Uma GAN é composta por duas redes que são competitivas entre si: (i) a rede generativa e a (ii) rede discriminativa. A premissa da **rede generativa** é aprender sobre o conjunto de dados de entrada na etapa de treinamento para imputar os elementos faltantes com precisão. Por outro lado, a rede generativa é treinada para maximizar a taxa de classificação da rede discriminativa. Os autores propõem uma adaptação no modelo original da **rede discriminativa** para que receba informação adicional. Tradicionalmente, a saída da rede generativa é classificada entre *true* ou *false* para cada imputação, em comparação com os elementos observados. No modelo de Yoon *et al.* (2018), a rede discriminativa é modificada para distinguir se a saída da rede generativa é observada ou imputada, a fim de melhorar a classificação dos elementos. Com isso, a rede discriminativa é treinada para minimizar o erro da geração de valores imputados pela rede generativa. Os experimentos realizados pelos autores demonstram que o método realiza imputações eficazes mesmo em altas taxas de dados faltantes, como também a classificação da rede discriminativa é realizada com qualidade.

Outra abordagem explorada no contexto de imputação baseada em *deep learning* é a aplicação de **Autoencoders** (ABIRI *et al.*, 2019). Um *autoencoder* é uma rede neural não supervisionada que consiste em (i) um *encoder* e (ii) um *decoder*. O **encoder** tem o objetivo de aprender sobre o conjunto de entrada para que cada camada de pesos seja treinada. Com isso, é gerada em uma camada intermediária da rede a representação dimensional de baixo nível do conjunto original de dados, comumente chamada de representação de espaço latente. Por outro lado, o **decoder** é treinado para reconstruir os dados de entrada a partir da representação do espaço latente que comprime os dados de entrada. Com isso, a saída de um *autoencoder* é o elemento imputado. Na Figura 15 é ilustrado um *autoencoder* com a arquitetura de rede com três camadas ocultas. Nos resultados de Abiri *et al.* (2019), o modelo de *autoencoder* proposto foi testado com várias bases de dados de imagens, apresentando baixas taxas de erros ou introdução de ruídos nas imputações realizadas. Por outro lado, a etapa de treinamento do método proposto

Figura 15 – Arquitetura de um *autoencoder* com três camadas ocultas.

Fonte: Elaborada pelo autor.

necessita de grandes quantidades de amostras para apresentar resultados satisfatórios.

Imputação de Valores em Consultas por Similaridade

Os métodos de Imputação de Valores são amplamente explorados, a partir de várias heurísticas eficientes e eficazes para lidar com o problema de dados faltantes. Por outro lado, a escolha de métodos de imputação adequados a cada aplicação é um desafio, uma vez que a análise sobre os tipos de falta de dados (MCAR, MAR, MNAR) é importante na tomada de decisão de qual método é o mais adequado (BRINIS *et al.*, 2019; TRAINA *et al.*, 2019; CISMONTI *et al.*, 2013). Nesse contexto, o principal ponto a ser considerado é que a aplicação de métodos de imputação pode introduzir distorções ao conjunto de dados, prejudicando negativamente a qualidade da recuperação de informação.

A Figura 16 ilustra a execução de uma consulta por similaridade por abrangência sobre um conjunto de dados complexos armazenados SGBDRs. Seja s_q um **centro de busca com valores faltantes**, ε um raio de consulta, cada ponto verde representa uma tupla completa e cada ponto vermelho representa uma tupla incompleta. A Figura 16a mostra o cenário em que a consulta com o centro de busca incompleto. Considerando todos as tuplas da base de dados, são retornadas 5 tuplas, incluindo 3 tuplas incompletas. Em outro cenário, é ilustrado na Figura 16b o resultado da aplicação de uma imputação de valores no centro de busca (ponto cinza), o que pode introduzir distorções na tupla em questão. Neste caso, com a distorção desta tupla, e repetindo a mesma consulta do caso 16a, é possível observar o deslocamento o raio de busca, uma vez que é baseado na posição do centro de consulta. Consequentemente, o conjunto resposta neste cenário pode ser diferente do caso anterior, retornando elemento não esperados no conjunto resposta.

Figura 16 – Consultas por similaridade com tratamento de imputação de valores.

Fonte: Elaborada pelo autor.

3.4.3 Indexação de Bases de Dados Incompletas

Outra maneira de lidar com dados faltantes e desenvolvida por alguns trabalhos é a **indexação de dados**, por meio de estruturas dedicadas ao armazenamento e recuperação de informação em bases de dados incompletas. O objetivo do tratamento de dados faltantes por meio de índices é incorporar heurísticas que lidem com o problema, de modo que a estrutura de indexação seja balanceada, recuperando informação de forma eficiente. O desenvolvimento de estruturas de indexação é especializado para cada aplicação. Portanto, é importante levar em consideração o contexto dos dados, e consequentemente as análises a serem realizadas.

Em Ooi, Goh e Tan (1998), os autores propõem duas estruturas de indexação para bases de dados incompletas: **BR-Tree** (*Bitstring-augmented R-Tree*) e **MOSAIC** (*Multiple One-attribute One-Dimension*). A **BR-Tree** foi proposta para reduzir o espaço de busca por meio da indexação aleatória de objetos com valores faltantes, dispersando os elementos e evitando o agrupamento de objetos incompletos. Para cada tupla incompleta, a estrutura atribui a média dos outros valores do mesmo atributo, melhorando a eficiência das consultas por similaridade. Os experimentos utilizando a BR-Tree mostram que a estrutura consegue processar de forma eficiente consultas com espaço de busca reduzido em altas taxas de dados faltantes. Por outro lado, o custo de execução cresce proporcionalmente ao número de 2^k sub-consultas que a estrutura necessita, sendo k o número de atributos faltantes. Portanto, a estrutura é prejudicada de acordo com o aumento da dimensionalidade dos conjuntos de dados.

Outra estrutura proposta em Ooi, Goh e Tan (1998) é a **MOSAIC**. Ela consiste na construção de um índice B^+ -Tree para cada atributo da base de dados, e os valores ausentes são substituídos por indicadores (-1) . Uma consulta é processada para cada B^+ -Tree, e desmembrada em $2k$ sub-consultas, onde cada uma retorna um conjunto de elementos candidatos incluindo

elementos completos e incompletos. Ao final, é realizada a intersecção entre os conjuntos de candidatos recuperados em cada B^+ -Tree, resultando no conjunto resposta final. Os resultados experimentais mostram que o MOSAIC é mais eficiente em consultas pontuais, enquanto a BR-Tree é mais eficiente em consultas por abrangência (OOI; GOH; TAN, 1998).

Em Canahuate *et al.* (2006), os autores propuseram estruturas de indexação baseadas em *Bitmap* e *VA-Files*, tratando bases incompletas em casos de falta não aleatória (MNAR). Com isso, as técnicas consideram que os atributos faltantes ocorridos não são relevantes para o objeto com a falta. Ambas as técnicas marcam os elementos com valores faltantes, sendo os atributos faltantes indexados separadamente. Dessa forma, para cada consulta realizada, os elementos marcados são verificados e caso atendam ao critério da consulta, são incluídos no conjunto de resposta. Os resultados experimentais (CANAHUATE *et al.*, 2006) mostram que a estrutura possui um bom desempenho na execução de consultas, mesmo em casos de crescente cardinalidade e dimensionalidade, devido a indexação independente de cada atributo.

Em Brinis *et al.* (2019) foi proposta a estrutura de indexação **Hollow-Tree**. Este é um dos primeiros trabalhos desenvolvidos para o problema de dados faltantes utilizando um método de acesso métrico (MAM) baseado em raio de cobertura, e não aplica métodos de imputação de valores. A Hollow-Tree propõe dois métodos para indexar e recuperar informação em banco de dados multidimensionais. O método **Observed Attribute Distance - (ObAD)** consiste em um conjunto de funções de distâncias métricas, sendo cada função definida para comparar elementos complexos de acordo com cada tipo de caso de dados faltantes (MCAR, MAR e MNAR). A finalidade do método é estimar um *score* de similaridade, considerando elementos completos e incompletos, resultando em um ranqueamento das distâncias para auxiliar na etapa de indexação dos elementos.

O outro método proposto por Brinis *et al.* (2019) é o **Complete First and Missing Last Insert - (CFMLI)**, e apresenta heurísticas para a indexação de elementos com valores faltantes para manter um alto grau de consistência da estrutura. O CFMLI usa heurísticas mais rígidas para a seleção de representantes para construção da estrutura, considerando apenas os elementos completos. A indexação é realizada em duas etapas: (i) o conjunto de elementos completos são indexados, formando o conjunto de representantes; (ii) os elementos incompletos são inseridos em nós folhas. Além disso, o CFMLI propõe uma heurística de quebra de nós considerando os elementos incompletos, para manter a estrutura balanceada e consistente.

A Hollow-Tree define consultas por similaridade adaptadas para a recuperação de dados complexos em base de dados incompletas. **Consultas por Abrangência (RM_q Query)** recuperam duas listas como conjuntos de candidatos, sendo uma lista de elementos completos e outra de elementos incompletos. **Consultas por Vizinhança ($k - NNF M_q$ Query)** recuperam os k elementos baseados na teoria de fractais, estimando a quantidade de objetos relevantes a serem recuperados a partir da dimensão intrínseca do conjunto de dados indexado. A consulta é mais precisa ao considerar elementos incompletos no conjunto resposta, diminuindo a distorção do

espaço causada pelos elementos incompletos.

3.4.4 Discussão de Trabalhos Relacionados

Nesta seção são discutidos os trabalhos relacionados apresentados em relação aos aspectos pertinentes à abordagem proposta desta dissertação:

- I - Suporte a **Consultas por Similaridade** por Abrangência e Vizinhança;
- II - **Não descarta dados**, sendo estes dados as tuplas ou atributos do conjunto;
- III - **Não Aplica Imputação de Valores** as tuplas ou atributos do conjunto;
- IV - **Utilização de Informação Intrínseca**, em específico a correlação de dados, para reduzir influência de dados faltantes em consultas.

Na [Tabela 1](#) são sumarizados os trabalhos relacionados de acordo com os aspectos selecionados, incluindo trabalhos de **descarte de dados**, **imputação de valores** e **indexação de dados**. Além disso, são discutidos os trabalhos de acordo com cada aspecto, em comparação a abordagem desta dissertação, nomeada por SOLID.

Tabela 1 – Visão Geral de abordagens selecionadas para tratamento de dados faltantes.

Abordagens	Consultas por Similaridade	Não Descarta Dados	Não Aplica Imputação de Valores	Utiliza Informação Intrínseca
Descarte (LITTLE; RUBIN, 2019)	×	×	✓	×
FCMI (ZHANG et al., 2016)	×	✓	×	×
kNNI (BATISTA; MONARD, 2003)	×	✓	×	×
DTI (NIKFA LAZAR et al., 2020)	×	✓	×	×
GAIN (YOON et al., 2018)	×	✓	×	×
SDAi (ABIRI et al., 2019)	×	✓	×	×
BR-Tree (OOI; GOH; TAN, 1998)	✓	✓	×	×
MOSAIC (OOI; GOH; TAN, 1998)	✓	✓	×	×
BEE (CANAHUATE et al., 2006)	✓	✓	×	×
Hollow-Tree (BRINIS et al., 2019)	✓	✓	✓	×
SOLID (RODRIGUES et al., 2020)	✓	✓	✓	✓

Consultas por Similaridade

Tarefa fundamental desta dissertação, a recuperação de informação é realizada por meio da execução de consultas por similaridade em dados complexos. Usualmente, os trabalhos que

realizam a imputação de valores, como o FCMI, kNNI, DTI GAIN e SDAi, lidam somente com a incompletude de dados em si, não propondo o suporte a recuperação de dados executando consultas por similaridade.

Em relação as estruturas de indexação, usualmente são desenvolvidas para indexar e recuperar informação. Com isso, as estruturas BR-Tree, MOSAIC, BEE e Hollow-Tree oferecem suporte a similaridade. Por fim, da mesma maneira que o método proposto nesta dissertação, as estruturas seguem as premissas dos espaços métricos para suporte a similaridade.

Não descarta dados

Analisando os trabalhos listados, somente o trabalho dedicado ao descarte utiliza tal método para lidar com a incompletude. Com isso, pode ser destacado que tal abordagem é rejeitada como solução no contexto de dados faltantes, uma vez que reduz a quantidade de tuplas/atributos do conjunto.

Em especial para esta dissertação, não descartar dados pode oferecer apoio ao lidar com a incompletude, uma vez que o método proposto utiliza a correlação dos dados disponíveis. Com isso, utilizar a originalidade dos dados se torna útil ao modo proposto de se lidar com dados faltantes.

Não Aplica Imputação de Valores

O aspecto de utilização de métodos de imputação de valores ao lidar com dados faltantes é outro ponto importante, pois sua aplicação pode introduzir distorções no conjunto de dados. Abordagens que não utilizam tais heurísticas, como a Hollow-Tree e a abordagem desta dissertação, podem se beneficiar ao lidar com a incompletude.

Em especial para a abordagem desta dissertação, não inferir valores a tuplas/atributos é vantajoso para evitar inserir distorções no espaços de busca em consultas por similaridade, como ilustrado na [Figura 16](#). Com isso, trabalhar com os dados originais favorece a abordagem proposta ao realizar consultas por similaridade.

Utilização de Informação Intrínseca

Ponto diferencial deste trabalho, é a utilização de informação intrínseca, em especial a correlação de dados. Com isso, parte do principio de utilizar a originalidade dos dados para lidar com a incompletude sem descarte ou inferir valores para executar consultas.

É possível destacar que todos os trabalhos relacionados exploram meios de lidar com os casos de dados faltantes, seja por descarte, imputação ou indexação, e posteriormente realizar a recuperação de informação. O método proposto tira vantagem da correlação para reduzir a influência da falta de dados durante a execução de consultas por similaridade.

3.5 Considerações Finais

Neste capítulo foi apresentada a Teoria dos Dados Faltantes e também os Trabalhos Relacionados desta dissertação. A Teoria dos dados faltantes define mecanismos para identificar tipos de dados faltantes, auxiliando a escolha de métodos adequados ao problema de dados faltantes em SGBDRs. Além disso, foi apresentado o problema de consultas por similaridade sobre bases incompletas, em que funções de distâncias tradicionais são impossibilitadas de medir a dissimilaridade entre objetos complexos.

Analisando os métodos existentes na literatura para lidar com o problema de dados faltantes, fica destacado que é um tema relevante e que ainda não está adequadamente resolvido. Além disso, a literatura apresenta uma quantidade significativa de abordagens, mostrando a preocupação em desenvolver métodos eficazes no tratamento e recuperação de informação em bases incompletas.

Um importante desafio ao lidar com dados faltantes é a escolha de métodos eficazes e eficientes que contribuam na recuperação de informação, considerando todos os dados disponíveis e sem introduzir distorções. No [Capítulo 4](#) é apresentada formalmente a abordagem, proposta nesta dissertação para a execução de consultas por similaridade sobre conjuntos de dados complexos em base de dados incompletas. O método lida com dados faltantes a partir da correlação dos dados complexos para se gerar fatores de compatibilidade considerando os atributos mais correlacionados sobre cada ocorrência de dados faltante. Naquele capítulo são apresentadas as etapas da abordagem com a formalização de consultas por similaridade utilizando o método desenvolvido.

SOLID: BUSCAS SOBRE DADOS CORRELACIONADOS E INCOMPLETOS

4.1 Considerações Iniciais

Neste trabalho de Mestrado é proposta e desenvolvida a abordagem SOLID - *Search Over Correlated and Incomplete Data*, que tem como objetivo a execução de consultas por similaridade sobre bases incompletas. O SOLID lida com o problema de consultas por similaridade sobre bases incompletas por meio do cálculo da correlação entre atributos complexos representados por representações extraídas de conjunto de métodos de extração de características (FEMs). O método gera fatores de compatibilidade a partir dos atributos complexos mais correlacionados em relação a cada ocorrência de dado faltante no conjunto. A execução de consultas por similaridade é realizada explorando correlações entre funções de distância, seguindo a proposta deste trabalho, a fim de aplicar os fatores de compatibilidade gerados, reduzindo a influência dos dados faltantes. O SOLID permite a utilização de múltiplas funções de distâncias, como as da família Minkowski, que comparam os objetos de cada um dos atributos separadamente, e pode ser aplicado um conjunto de extratores de características, adequado ao domínio dos dados complexos. Vale destacar que o método proposto não utiliza o descarte de elementos ou mesmo a imputação de valores em bases de dados.

Este capítulo está organizado da seguinte maneira: a [Seção 4.2](#) apresenta os conceitos principais e os processos que compõem o SOLID. Na [Seção 4.3](#) é abordada a etapa de pré-processamento de dados complexos com aplicação de FEMs e funções de distâncias. A [Seção 4.4](#) apresenta o método CorDiS, que calcula e armazena a correlação sobre os múltiplos espaços métricos. Na [Seção 4.5](#) é apresentado o processamento das correlações para a geração dos fatores de compatibilidade entre espaços métricos correlacionados a partir das ocorrências de dados faltantes no conjunto de dados. Por fim, a [Seção 4.6](#) apresenta a função de distância Solid-Dist para aplicar e consolidar os fatores de compatibilidade em consultas por similaridade sobre

bases de dados incompletos. Por fim, a [Seção 4.7](#) finaliza este capítulo, com considerações sobre a principal contribuição desta Dissertação de Mestrado. A [Tabela 2](#) apresenta os símbolos empregados neste capítulo.

Tabela 2 – Símbolos empregados na abordagem SOLID.

Símbolo	Descrição	Símbolo	Descrição
$\mathcal{D}$	Base de Dados	ψ	Função Solid-Dist
n	Cardinalidade de $\mathcal{D}$	F	Quantidade de FEMs
d	Dimensionalidade de $\mathcal{D}$	G	Quantidade de δ
S	Conjunto de Objetos Complexos	$mCorr$	Mapa de Correlação
$\mathbb{S}$	Domínio dos Dados Complexos	$\mathbb{A}$	Distâncias Médias
s_i	Atributo Complexo	γ	Limiar de Compatibilidade
$\mathbb{M}$	Espaço Métrico	ω	Fator de Compatibilidade
$\mathfrak{M}$	Conjunto de $\mathbb{M}$	Ω	Conjunto de ω
δ	Função de Distância Métrica	ϖ	ω Consolidado
v	Vetor de Características	Υ	Conjunto de ω Consolidados
ϕ	Função de Correlação	β	Diâmetro de Normalização de Distâncias

Fonte: Elaborada pelo autor.

4.2 O Método SOLID

A abordagem SOLID assume que é possível executar consultas por similaridade em um conjunto de dados $\mathcal{D}$ com tuplas com valores faltantes, utilizando pares de espaços métricos $\langle \mathbb{M}_i, \mathbb{M}_j \rangle \in \mathfrak{M}$ correlacionados entre si, considerando $1 < i \leq d$ como os espaços métricos existentes em $\mathcal{D}$, quando calculada por uma função de correlação (ϕ). Com isso, é formado um conjunto de espaços métricos $\mathfrak{M}$. Nesse contexto, o SOLID gera fatores de compatibilidade a partir da identificação de atributos complexos compatíveis utilizando as correlações de $\mathfrak{M}$. Além disso, é proposta uma função de distância global para aplicação dos fatores de compatibilidade, chamada de Solid-Dist - (ψ). Ela é utilizada na execução de consultas por similaridade ponderadas a partir dos fatores gerados a fim de reduzir a influência da falta de dados.

A principal vantagem desta abordagem é explorar informações extraídas do conjunto de dados complexos armazenados em SGBDRs para a execução de consultas por similaridade. Diferentemente dos métodos tradicionais, como a imputação de valores, esta abordagem não visa inferir um valor ao objeto faltante, evitando inserir distorções no conjunto de dados. Além disso, o método não descarta nenhum elemento do conjunto de dados, pois informações úteis e relevantes podem ser perdidas. Com isso, o SOLID tira vantagem da originalidade da base de dados de dados para extração de informação intrínseca dos dados.

A [Figura 17](#) apresenta o método SOLID para executar consultas por similaridade sobre bases de dados incompletos, composto pelos seguintes processos, detalhados nas seções a seguir:

- I - Processamento e Armazenamento de bases de dados complexos $\mathfrak{D}$, extraindo características a partir da aplicação de um conjunto de FEMs e de um conjunto de funções de distâncias, constituindo um conjunto de espaços métricos $\mathfrak{M}$.
- II - Mapeamento de Correlações entre Espaços de Distâncias, no qual é proposto o método CorDiS, que calcula a correlação entre pares de espaços métricos de $\mathfrak{M}$, resultando em um mapa de correlação sobre $\mathfrak{M}$.
- III - Geração de Fatores de Compatibilidade sobre Espaços de Distâncias, calculados a partir do mapa de correlação gerado no processo anterior (2).
- IV - A aplicação dos fatores de compatibilidade para ponderar consultas por similaridade com o SOLID, a partir da função de distância Solid-Dist, proposta para utilizar os espaços métricos mais correlacionados para reduzir a influência dos objetos faltantes nas consultas.

Figura 17 – Sequência de processos do método SOLID.

Fonte: Elaborada pelo autor.

4.3 Processamento de Conjuntos de Dados Complexos

O SOLID executa consultas por similaridade sobre um conjunto de espaços métricos $\mathfrak{M}$, constituídos a partir de um conjunto de representações, a partir de extratores de característi-

cas (FEMs) e funções de distância (δ), quando aplicados sobre bases de dados complexos $\mathcal{D}$, de acordo com a Teoria dos Espaços Métricos. Nesse contexto, a primeira etapa do método é o processamento de todos os elementos. A Figura 18 ilustra um conjunto de imagens médicas com exames distintos em que ocorrem alguns atributos com valores faltantes.

Figura 18 – Base de dados complexos constituída de diferentes exames médicos.

	Cérebro	Coração	Pulmão	Estômago	Olho
1
2	NULL				NULL
3
4			NULL	NULL
5		NULL			NULL

Fonte: Elaborada pelo autor.

Considerando d como a dimensionalidade de $\mathcal{D}$ e os atributos Cérebro, Coração, Pulmão, Estômago e Olho, é dito que cada atributo complexo (por exemplo Cérebro) possui um conjunto de imagens de cardinalidade n , e constitui cada conjunto de dados $S \in \mathbb{S}$. Casos em que um objeto complexo $s_i \in S$ de determinado atributo é incompleto, o elemento é indicado por NULL. As tuplas com atributos faltantes não são excluídas e nem são inferidos valores aos objetos faltantes, sendo que cada objeto NULL é ignorado no processo de extração de características.

Na extração de características, o SOLID necessita de um conjunto de F extratores FEM_f , tal que $1 \leq f \leq F$, e um conjunto de G funções de distância δ_g , tal que $1 \leq g \leq G$. O método possibilita a utilização de combinações entre FEMs e funções de distâncias (δ), gerando representações distintas para cada atributo de $\mathcal{D}$. A combinação $\langle FEM_f, \delta_g \rangle$ escolhida para cada atributo gera um espaço métrico $\mathbb{M}_i = \langle S_f, \delta_g \rangle$, na qual S_f é o conjunto de vetores de características resultantes do extrator FEM_f , pertencentes a um domínio dos dados complexos $\mathbb{S}$, e δ_g uma função de distância que mede a dissimilaridade entre os elementos de S_f . Portanto, é gerado um espaço métrico $\mathbb{M}_i$ sobre cada atributo do conjunto de d atributos de $\mathcal{D}$, formando o conjunto de espaços métricos $\mathfrak{M}$, de acordo com a Definição 4:

Definição 4. Conjunto de Espaços Métricos - $\mathfrak{M}$: Dado um conjunto de atributos de $\mathcal{D}$, é definido para cada atributo um $\mathbb{M}_i$ espaço métrico por $\mathbb{M} = \langle \mathbb{S}, \delta \rangle \in \mathfrak{M}$, $1 \leq i \leq d$, na qual δ é uma função de distância, resultando em $\mathfrak{M} = \{\mathbb{M}_1, \mathbb{M}_2, \dots, \mathbb{M}_d\}$.

Na Figura 19 é ilustrado um conjunto de espaços métricos ($\mathfrak{M}$) resultante do processamento de uma coleção de dados complexos. No exemplo, o conjunto de d atributos é representado

por vetores de características resultantes da aplicação de um conjunto de extratores de características (FEMs). Especificamente, foram aplicados os seguintes extratores: Textura, Forma, Cor, Forma e Textura, sendo cada extrator (FEM) associado à função euclidiana. Considerando o atributo S_1 , cada objeto complexo $s_i \in S_1$ é definido por um par $\langle FEM, \delta \rangle$, por exemplo $\langle Textura, Euclidiana \rangle$. Com isso, o $\mathbb{M}_1$ é constituído para o atributo S_1 .

Figura 19 – Definição do conjunto de espaços métrico $\mathfrak{M}$.

FEMs					
	Textura	Forma	Cor	Forma	Textura
	S_1	S_2	S_3	S_4	S_d
1	$\{x_1, x_2, \dots, x_i\}$	$\{x_1, x_2, \dots, x_i\}$	$\{x_1, x_2, \dots, x_i\}$	$\{x_1, x_2, \dots, x_i\}$	$\{x_1, x_2, \dots, x_i\}$
2	NULL	$\{x_1, x_2, \dots, x_i\}$	$\{x_1, x_2, \dots, x_i\}$	$\{x_1, x_2, \dots, x_i\}$	NULL
...					
n	$\{x_1, x_2, \dots, x_i\}$	$\{x_1, x_2, \dots, x_i\}$	NULL	NULL	$\{x_1, x_2, \dots, x_i\}$
	$\mathbb{M}_1$		$\mathbb{M}_3$		$\mathbb{M}_d$

Fonte: Elaborada pelo autor.

Este processamento é realizado sobre cada atributo complexo $S_i \in \mathfrak{D}$, constituindo o respectivo espaço métrico $\mathbb{M}_i \in \mathfrak{M}$. Ao final, esta etapa armazena o conjunto de espaços métricos $\mathfrak{M}$ de $\mathfrak{D}$. A seguir, é discutido o método CorDiS, para o mapeamento de correlações sobre espaços de distâncias, considerando o conjunto de espaços métricos $\mathfrak{M}$ armazenado.

4.4 CorDiS: Correlações entre Espaços Métricos

Uma das premissas do SOLID é utilizar informações extraídas de conjuntos de dados complexos a fim de auxiliar a execução de consultas por similaridade sobre base de dados incompletos. O SOLID propõe o método CorDiS - *Correlation of Distance Spaces*, que mapeia as correlações entre pares de espaços métricos $\langle \mathbb{M}_i, \mathbb{M}_j \rangle$, de acordo com a [Definição 5](#).

Definição 5. CorDiS. Seja um par de espaços métricos $\langle \mathbb{M}_i, \mathbb{M}_j \rangle$, uma função de distância δ e uma função de correlação ϕ . O método CorDiS calcula as distâncias médias para cada conjunto de vetores de características de $\mathbb{M}_i$ e de $\mathbb{M}_j$ quando aplicada δ , resultando em vetores de distâncias médias $\mathbb{A}_i$ e $\mathbb{A}_j$, que definem um espaço de distâncias sobre cada espaço métrico. É retornada a correlação entre o par de espaços de distâncias, a partir da função de correlação $\phi(\mathbb{A}_i, \mathbb{A}_j)$.

Para maior detalhamento do método CorDiS, o [Algoritmo 1](#) mostra a implementação do mapeamento da correlação entre pares de espaços métricos $\langle \mathbb{M}_i, \mathbb{M}_j \rangle$, pertencentes ao conjunto de espaços métricos $\mathfrak{M}$. O método necessita somente de uma amostra $\mathfrak{D}'$ de $\mathfrak{D}$ para ser executado, sendo $\mathfrak{D}'$ um par de espaços métricos $\langle \mathbb{M}_i, \mathbb{M}_j \rangle$, por exemplo um par de atributos complexos de $\mathfrak{D}$. Consideram-se os conjuntos de vetores de características $S_i \in \mathbb{M}_i$ e $S_j \in \mathbb{M}_j$, descritos na linha 3. Nas linhas 4 e 5, o CorDiS inicia a execução a partir da chamada da Função `CalculaDistanciasMedias`, calculando as distâncias médias sobre os conjuntos de vetores de característica S_i e S_j .

A função `CalculaDistanciasMedias` inicializa na linha 11 um vetor de distâncias parciais D_j que armazenará todas as distâncias calculadas por δ . Nas linhas 10 a 17 a função compara cada objeto complexo s_i com cada outro objeto s_j do conjunto de vetores de características S . Na linha 14, a função concatena o valor de distância média de D_j ao vetor final de distâncias médias $\mathbb{A}$. Na linha 18, a função retorna o vetor de distâncias médias $\mathbb{A}$. Após obter a distância de cada objeto para todos os elementos de $\mathfrak{D}'$, na linha 6 o CorDiS calcula a correlação entre ambos os espaços de distâncias utilizando uma função de correlação ϕ , e devidamente armazenada na matriz *mCorr*. Por fim, na linha 7 a função retorna o valor de correlação entre o par de espaços métricos, sendo adicionado na matriz de correlação $\langle \mathbb{M}_i, \mathbb{M}_j \rangle$, nomeada de **mapa de correlação**. Após calcular a correlação entre todos os pares de espaços métricos disponíveis, temos um mapa de correlação (*mCorr*) do conjunto de espaços métricos ($\mathfrak{M}$).

Vale notar que o CorDiS tem como requisito uma função de correlação. Mais especificamente, o método pode utilizar qualquer função de correlação que retorne um valor no intervalo $[\mathbb{R}, \mathbb{R}]$, como os coeficientes de correlação de **Pearson** ou **Spearman**. Além disso, a etapa de cálculo de correlação, apesar de computacionalmente cara, é realizada somente uma vez, e de modo *offline*, ou seja, como uma etapa de pré-processamento do SOLID. Dessa forma, não impacta o tempo final de execução das consultas por similaridade sobre bases de dados incompletos, pois o custo computacional desta etapa não é considerado.

A [Figura 20](#) ilustra o mapa de correlação, gerado a partir do coeficiente de correlação de Pearson sobre a base de dados de exames médicos ilustrada na [Figura 19](#)). A escala de correlação é válida para todos valores de $0 < corr < 1$, $corr \in \mathbb{R}$, representada visualmente por tons em vermelho para valores de alta correlação, e tons em azul para valores de baixa correlação.

Em um mapa de correlação, os valores são apresentados pelas combinações entre os diversos atributos existentes e seus respectivos FEMs. O CorDiS foi implementado para possibilitar o mapeamento de correlações entre atributos representados por qualquer tipo de FEM, incluindo FEMs distintos, por exemplo $\langle \text{Textura}, \text{Forma} \rangle$ e $\langle \text{Textura}, \text{Cor} \rangle$.

A partir de um mapa de correlação, são gerados fatores de compatibilidade em relação aos atributos mais correlacionados identificados para cada atributo com valores faltantes. Os fatores de compatibilidade têm como objetivo reduzir a influência da falta de um ou mais atributos com valores faltantes por meio de um conjunto de outros atributos correlacionados a ele(s). Este

Algoritmo 1 – CorDiS: Mapeamento de Espaços Métricos Correlacionados**Entrada:** $\langle \mathbb{M}_i, \mathbb{M}_j \rangle$: Par de espaços métricos, δ_i : Função de Distância de $\mathbb{M}_i$, δ_j : Função de Distância de $\mathbb{M}_j$, ϕ : Função de Correlação.**Saída:** $mCorr$: Correlação.**1 início****2** Inicialização; // Gerar conjuntos de distâncias médias $\mathbb{A}_i$ e $\mathbb{A}_j$ **3** Sejam S_i e S_j os conjuntos de vetores de características de $\mathbb{M}_i$ e $\mathbb{M}_j$, respectivamente;**4** $\mathbb{A}_i \leftarrow \text{CalculaDistanciasMedias}(S_i, \delta_i)$;**5** $\mathbb{A}_j \leftarrow \text{CalculaDistanciasMedias}(S_j, \delta_j)$;**6** $mCorr_{i,j} \leftarrow \phi(\mathbb{A}_i, \mathbb{A}_j)$; // Calcula a correlação e armazena na matriz**7** **retorna** $mCorr_{i,j}$;**8 fim****9 Função** $\text{CalculaDistanciasMedias}(S, \delta)$ **10** **para cada** objeto s_i em S **faça****11** $D_i \leftarrow []$; // Inicializar vetor de distâncias**12** **para cada** objeto s_j em S // Obter distâncias para s_i **13** **faça****14** $D_i \leftarrow D_i \cup [\delta(s_i, s_j)]$;**15** **fim****16** $\mathbb{A} \leftarrow \mathbb{A} \cup \text{MEAN}(D_i)$; // Calcular a distância média**17** **fim****18** **retorna** $\mathbb{A}$;

processamento é descrito a seguir na etapa de Geração de Fatores de Compatibilidade.

4.5 Geração de Fatores de Compatibilidade

O SOLID define o conceito de compatibilidade entre espaços métricos, a partir das correlações extraídas pelo CorDiS, armazenadas no mapa de correlação. Mais especificamente, o objetivo é estimar o quanto o espaço métrico $\mathbb{M}_a$ (correspondente ao atributo a) é compatível em relação a outro o espaço métrico $\mathbb{M}_b$ (correspondente ao atributo b), sendo esses atributos pertencentes a $\mathcal{D}$. Formalmente, um fator de compatibilidade é definido de acordo com a [Definição 6](#).

Definição 6. Fator de Compatibilidade (ω). Sejam a e b dois atributos de $\mathcal{D}$, tal que $0 \leq a \leq d, 1 \leq b \leq d$, sendo d a dimensionalidade de $\mathcal{D}$. O fator de compatibilidade $\omega_{a,b}$ é o complemento da correlação normalizada considerando o par de espaços métricos $\langle \mathbb{M}_a, \mathbb{M}_b \rangle$, sendo calculado pela função de compatibilidade, conforme a [Equação 4.1](#).

Figura 20 – Geração do mapa de correlação sobre o conjunto de espaços métricos $\mathfrak{M}$.

Fonte: Elaborada pelo autor.

$$\omega(a, b) = |1 - mCorr(\mathbb{M}_a, \mathbb{M}_b)| \quad 0 \leq \omega \leq 1, \quad \omega \in \mathbb{R} \quad (4.1)$$

Mais detalhes sobre a aplicação dos mesmos é discutida na [Seção 4.6](#). Assumindo que uma base $\mathfrak{D}$ tem d atributos, todos os fatores de compatibilidade calculados sobre d atributos constituem o **Conjunto de Fatores de Compatibilidade**, conforme a [Definição 7](#).

Definição 7. Conjunto de Fatores de Compatibilidade (Ω). São os fatores de compatibilidade gerados entre um atributo a sobre cada atributo b de $\mathfrak{D}$. Cada $\omega_{a,b} \in \Omega_a$ é o fator de compatibilidade de a sobre o atributo b .

A função de compatibilidade recebe o parâmetro γ , sendo o limiar de compatibilidade mínimo entre dois atributos, selecionando somente valores de correlação maiores ou iguais a γ . Considerando que γ são valores reais $\mathbb{R}^+$ no intervalo de $[0, 1]$, γ restringe a seleção de atributos com baixa ou nenhuma correlação para o conjunto de fatores de compatibilidade Ω . A aplicação do limiar γ tem como hipótese que valores de correlação mais altos, por exemplo, valores maiores ou iguais que $\gamma \geq 0,5$, resultam em atributos mais correlacionados à atributos com valores faltantes.

Os fatores de compatibilidade do conjunto Ω são normalizados para que sejam devidamente aplicados sobre as distâncias resultantes das comparações realizadas nas consultas por

similaridade. Dado o mapa de correlação do conjunto de dados, todos os fatores de compatibilidade do conjunto Ω são normalizados no intervalo de $[0, 1]$. A normalização é feita a partir de cada linha dos valores de correlação da matriz, que corresponde aos fatores de compatibilidade de determinado atributo, calculados pela função de compatibilidade (ver [Equação 4.1](#)). Esta normalização é justificada pelo fato de que os valores de correlação são incompatíveis quando aplicados diretamente sobre distâncias resultantes de uma função de distância δ . Na prática, serão atribuídos pesos (fatores de compatibilidade) a partir dos valores de correlação sobre os atributos compatíveis, de modo que as distâncias resultantes diminuam, compensando os atributos faltantes que não podem ser considerados. Com isso, a aplicação dos pesos sobre as distâncias mensuradas por uma δ , resultaria em distâncias maiores que as originais, uma vez que o fator de compatibilidade multiplica a distância resultante. Portanto, a normalização torna os valores de correlação aplicáveis para a ponderação de distâncias em consultas por similaridade.

A [Figura 21](#) ilustra o processo do cálculo de fatores de compatibilidade utilizando o mapa de correlação de uma base de dados $\mathcal{D}$, resultando no conjunto de fatores de compatibilidade Ω . Considerando que o CorDiS fornece o mapa de correlação de $\mathcal{D}$ e o limiar de compatibilidade $\gamma = 0,5$, a função de compatibilidade calcula todos os fatores ω entre as combinações dos atributos. Por exemplo, a geração de fatores para o **atributo Cérebro** é executada com a aplicação da função de compatibilidade, selecionando os respectivos atributos com valores de correlação maiores ou iguais ao limiar γ : $\{\text{Pulmão} : 0,8, \text{Estômago} : 0,8, \text{Olho} : 0,5\}$, sendo não incluído o atributo $\{\text{Coração} : 0,2\}$ por não atender ao mínimo de γ . Logo, para cada **atributo compatível** é gerado um fator de compatibilidade ω , devidamente armazenado no conjunto de fatores de compatibilidade Ω . Esse processo é realizado sobre todos os atributos de $\mathcal{D}$.

Figura 21 – Aplicação da função de compatibilidade na geração do conjunto Ω de fatores utilizando valores de correlação superiores ao limiar $\gamma = 0,5$.

Fonte: Elaborada pelo autor.

Considerando que esta etapa é executada *offline*, ou seja, é uma etapa de pré-processamento do SOLID, não há impacto no desempenho na execução de consultas por similaridade sobre base de dados incompletas. A seguir, é detalhado como são executadas as consultas por similaridade sobre bases de dados incompletos com o SOLID.

4.6 Consultas por Similaridade com o SOLID

O SOLID aborda o problema de recuperação de informação em bases de dados incompletos sem a exclusão de elementos ou imputação de valores. Para isso, o método tem como premissa utilizar o máximo de dados disponíveis nas bases de dados, considerando que quando há casos de dados faltantes, as funções de distâncias tradicionais não consideram tuplas com atributos faltantes.

Nesse contexto, o SOLID propõe consultas por similaridade que recuperam as **tuplas mais similares**, comparando atributos complexos que podem estar incompletos. Primeiramente, o método utiliza os **Fatores de Compatibilidade Consolidados** para distribuir mais importância aos atributos mais correlacionados em relação a cada caso de dado faltante, de acordo com [Definição 8](#).

Definição 8. Fatores de Compatibilidade Consolidados (ϖ). Seja t_j uma tupla em $\mathcal{D}$ e P_j o conjunto de atributos faltantes de t_j . Cada ϖ é atribuído dinamicamente durante a execução da consulta, de acordo com cada atributo faltante $p \in P_j$, e calculado com base no Fator de Compatibilidade ω pelo [Algoritmo 2](#).

Algoritmo 2 – SOLID (Fatores de Compatibilidade Consolidados de uma Tupla)

Entrada: t_j : Tupla, Ω : Conjunto de Fatores de Compatibilidade de $\mathcal{D}$.

Saída: Υ : Conjunto de Fatores de Compatibilidade Consolidados da t_j .

```

1 início
2   Seja  $P_j$  o conjunto de atributos faltantes em  $t_j$ ;
3    $\varpi_{a,p} \leftarrow []$ ;
4   para cada atributo  $a$  em  $t_j$  faça
5     para cada atributo faltante  $p$  em  $P_j$  faça
6        $\varpi_{a,p}.append(\omega_{a,p})$ ; // Obter fatores de compatibilidade de  $p$ 
7     fim
8      $\Upsilon_j.add(\varpi_{a,p})$ ; // Adicionar fatores consolidados ao conjunto de  $t_j$ 
9   fim
10  retorna  $\Upsilon_j$ ;
11 fim

```

O [Algoritmo 2](#) calcula cada Fator de Compatibilidade Consolidado (ϖ), procurando pelos atributos com valores ausentes em cada tupla t_j existente em $\mathcal{D}$. Seja P_j um conjunto de atributos faltantes que podem ser identificados em t_j , inicializado como vazio. Na linha 3, é inicializada a variável de um fator consolidado $\varpi_{a,p}$ de t_j . Nas linhas 4 a 9, é feita a busca pelos fatores de compatibilidade correspondentes aos atributos faltantes p de t_j . Na linha 6, cada fator de compatibilidade $\omega_{a,p}$ identificado para p é concatenado em $\varpi_{a,p}$. Na linha 8, os fatores $\varpi_{a,p}$ obtidos são adicionados ao conjunto de fatores consolidados Υ_j da tupla t_j . Ao final (linha 10), é retornado o conjunto de fatores de compatibilidade consolidados Υ_j a fim de ser aplicado na ponderação da consulta por similaridade.

Os fatores de compatibilidade consolidados (ϖ) são gerados somente para cada atributo faltante identificado, ou seja, o processo é realizado pontualmente a cada caso. Desse modo, o cálculo e a distribuição dos fatores consolidados sobre outros atributos correlacionados são realizados dinamicamente quando um atributo faltante é encontrado durante a execução da consulta. Esse processo auxilia na comparação entre tuplas com atributos faltantes, pois os atributos mais importantes serão ponderados quando calculadas as distâncias entre pares de tuplas. O cálculo de distâncias e a aplicação do conjunto de fatores de compatibilidade consolidado - Υ é realizada pela função proposta no SOLID e apresentada a seguir.

O SOLID propõe a **Função Solid-Dist** que possibilita o cálculo das distâncias entre tuplas de atributos complexos para diferentes cenários de dados faltantes. Para mensurar as distâncias entre duas tuplas, a Solid-Dist compara os pares de atributos não nulos existentes, aplicando os fatores de compatibilidade consolidados nos atributos mais correlacionados, de acordo com cada caso de dado faltante identificado no [Algoritmo 2](#), conforme a [Definição 9](#).

Definição 9. Solid-Dist (ψ). Seja δ_a uma função de distância utilizada para comparar um par de vetores de características $\langle v_{a,q}, v_{a,j} \rangle$, $1 \leq a \leq d$, $1 \leq j \leq n$, de um atributo a . Dado Υ_q, Υ_j como os conjuntos de fatores de compatibilidade consolidados de t_q e t_j , respectivamente. A distância entre o par de tuplas $\langle t_q, t_j \rangle$ é medida pela [Equação 4.2](#):

$$\psi(t_q, t_j) = \sum_{a=1}^d \varpi_{q,a} \times \varpi_{j,a} \times \delta(v_{a,q}, v_{a,j}), \quad (4.2)$$

onde $\varpi_{q,a} \in \Upsilon_q$ e $\varpi_{j,a} \in \Upsilon_j$ são os fatores de compatibilidade consolidados do atributo a .

Nas subseções a seguir são apresentados conceitos adicionais e exemplos sobre a execução de consultas por similaridade com o SOLID. Em específico, na [Subseção 4.6.1](#) é apresentada a normalização de distâncias entre vários espaços métricos distintos realizada durante a aplicação da função Solid-Dist, consolidando a distância entre pares de tuplas. Na [Subseção 4.6.2](#) são apresentados os cenários de aplicação em que o SOLID executa consultas por similaridade sobre bases incompletas. Por fim, na [Subseção 4.6.3](#) é apresentado um exemplo prático da execução de consultas por similaridade com SOLID, destacando a associação de fatores de compatibilidade e os cálculos e normalização de distâncias com a função Solid-Dist.

4.6.1 Normalização de Distâncias entre Espaços Métricos Distintos

O SOLID executa consultas por similaridade sobre um conjunto de espaços métricos $\mathfrak{M}$, na qual os elementos são organizados em diferentes distribuições de distância, de acordo com cada espaço métrico $\mathbb{M}_i$. Desse modo, as distâncias resultantes sobre cada par de atributos, quando medidas pela Solid-Dist- ψ , podem variar amplamente, uma vez que cada espaço métrico tem um diâmetro espacial variável. Neste cenário, é dito que as distâncias resultantes de ψ não são normalizadas entre si. O SOLID realiza a consolidação de espaços de distâncias por meio da normalização de distâncias, estimando o diâmetro de cada espaço métrico $\mathbb{M}_i \in \mathfrak{M}$, seguindo a [Definição 10](#).

Definição 10. Diâmetro de Normalização de Distâncias (β). Dado um espaço métrico $\mathbb{M}_i$, o SOLID calcula as distâncias entre todos os elementos e recupera a maior distância do espaço de distâncias, sendo um valor real ≥ 0 e definido como o diâmetro β_i de $\mathbb{M}_i$. O SOLID normaliza qualquer distância d resultante de δ para o intervalo $[0, 1]$, selecionando um $\beta_i \in \mathbb{M}_i$, e calculando como d/β_i .

A normalização de distâncias é baseada na técnica de normalização *Min-max* ([HAN; KAMBER; PEI, 2011](#)), resultando na transformação linear das distâncias originais. No caso do SOLID, cada distância d resultante é normalizada com base no diâmetro de cada espaço métrico $\mathbb{M}_i \in \mathfrak{M}$ durante a consulta por similaridade, ou seja, o processo é realizado em tempo de execução. Esta normalização é justificada para se compatibilizar as distâncias resultantes entre os diferentes espaços métricos de uma tupla quando utilizada na consulta, permitindo mensurar a distância final entre tuplas da base de dados. A [Figura 22](#) ilustra como é associado um diâmetro para cada espaço métrico existente no conjunto $\mathfrak{M}$.

Figura 22 – Associação de diâmetro para cada espaço métrico $\mathbb{M}_i$.

	$\mathbb{M}_1$		$\mathbb{M}_3$		$\mathbb{M}_d$
	S_1	S_2	S_3	S_4	S_d
1	$\{x_1, x_2, \dots, x_d\}$	$\{x_1, x_2, \dots, x_d\}$	$\{x_1, x_2, \dots, x_d\}$	$\{x_1, x_2, \dots, x_d\}$	$\{x_1, x_2, \dots, x_d\}$
2	NULL	$\{x_1, x_2, \dots, x_d\}$	$\{x_1, x_2, \dots, x_d\}$	$\{x_1, x_2, \dots, x_d\}$	NULL
...					
n	$\{x_1, x_2, \dots, x_d\}$	$\{x_1, x_2, \dots, x_d\}$	NULL	NULL	$\{x_1, x_2, \dots, x_d\}$
	β_1		β_3		β_d

Fonte: Elaborada pelo autor.

A função Solid-Dist (ψ) aplica a normalização de distâncias baseada no diâmetro β_i de determinado espaço métrico $\mathbb{M}_i$ quando calculada a distância entre pares de vetores de características. Na prática, as distâncias resultantes são normalizadas quando calculadas de acordo com a [Equação 4.3](#):

$$\psi(t_q, t_j) = \sum_{a=1}^d \varpi_{q,a} \times \varpi_{j,a} \times (\delta(v_{a,q}, v_{a,j}) / \beta_i), \quad (4.3)$$

sendo β_i o diâmetro de normalização definido sobre o respectivo espaço métrico $\mathbb{M}_i$ de $v_{a,q}$ e $v_{a,j}$, quando mensurados por δ . Vale notar que a [Equação 4.3](#) é uma adaptação da função Solid-Dist original (ver [Equação 4.2](#)). Com isso, é possível a comparação de tuplas e consolidação de distâncias entre espaços métricos distintos, quando executadas consultas por similaridade com o SOLID.

4.6.2 Consultas por Similaridade SOLID: Cenários de aplicação

A partir das definições de como SOLID aplica a correlação dos dados para mensurar a distância entre tuplas em bases de dados incompletas, o método explora três cenários distintos de ocorrência de dados faltantes para executar consultas por similaridade:

- I - a tupla de centro de consulta (t_q) e a base de dados **não possuem** atributos com valores faltantes.
- II - a tupla de centro de consulta (t_q) possui um ou mais atributos com valores faltantes, mas a base de dados tem somente tuplas completas;
- III - a tupla de centro de consulta (t_q) e/ou a base de dados possuem atributos com valores faltantes;

Seja t_q uma tupla de centro de consulta e t_j uma tupla da base de dados $\mathcal{D}$. No cenário (i) em que a tupla de centro de consulta (t_q) e a base de dados não têm atributos com valores faltantes, a consulta é realizada normalmente, considerando todas as respectivas comparações entre atributos e sem a aplicação da ponderação de distâncias.

Nos cenários (ii) e (iii), se t_q e/ou t_j possuem um ou mais atributos com valores faltantes, não é possível executar a comparação entre os respectivos vetores de características do par de atributos. Sendo $v_{a,q}$ e $v_{a,j}$ os vetores de um par de atributo com valores faltantes, a distância resultante é $\delta_a(v_{a,q}, v_{a,j}) = 0$. Essa regra atende a todas as comparações entre pares de vetores que envolvem valores faltantes da base de dados.

Em consultas com atributos incompletos, como nos cenários (ii) e (iii), o SOLID compensa cada elemento faltante aplicando os fatores de compatibilidade gerados, distribuindo o peso da dimensão ausente entre os espaços métricos mais correlacionados. Na prática, o método atribui

mais importância a outros atributos correlacionados, dito como atributos compatíveis. Com isso, são aplicados os fatores de compatibilidade consolidados ϖ sobre as distâncias resultantes das comparações dos atributos compatíveis, aproximando a distância deste(s) atributo(s). O foco deste processo é reduzir a influência da falta de dados na execução das consultas por similaridade.

O SOLID suporta a execução de consultas por similaridade por vizinhança (kNN_q) e por abrangência (R_q). Em **consultas** kNN_q , o método realiza o ranqueamento do conjunto resposta final, retornando as k tuplas mais similares a t_q , conforme a quantidade requerida pelo usuário.

A execução de **consultas** R_q o método realiza o ranqueamento de forma similar a consultas kNN_q , porém a seleção de tuplas é baseada no raio ε definido pelo usuário. Mais especificamente, os raios aplicados em consultas com o SOLID são estipulados com base nos diâmetros de cada espaço de distância, identificados na etapa de processamento dos dados e realizado de forma *offline*. Na prática, as distâncias do conjunto de resposta são normalizadas pelo diâmetro (ver [Subseção 4.6.1](#)). Desse modo, o raio aplicado em uma consulta com o SOLID significa buscar pela proporção do espaço de distância, por exemplo, $\varepsilon = 0,2$ equivale a 20% do diâmetro, e assim para cada raio aplicado. Com isso, dado um centro de consulta t_q , são retornadas as tuplas mais similares com distância menor ou igual a ε , de acordo com $\delta < \varepsilon$.

Vale destacar que o SOLID necessita de pouca parametrização para a execução de consultas por similaridade. É necessário definir os requisitos mínimos de cada tipo de operador por similaridade, como o valor de k e raio de consulta ε , e o limiar de correlação γ a fim de filtrar atributos mais correlacionados na geração de fatores de compatibilidade.

4.6.3 Exemplo de Consultas por Similaridade com o SOLID

O exemplo detalhado a seguir representa o cenário (iii), em que na tupla de centro de consulta (t_q) e na base de dados (t_j) ocorrem atributos com valores faltantes. O processo de execução de consultas por similaridade executada pelo SOLID é dividido em três etapas:

- I - Associação de Fatores de Compatibilidade Consolidados ϖ ;
- II - Cálculos de distâncias com a Solid-Dist- ψ ;
- III - Normalização de distâncias entre diferentes atributos.

A [Figura 23](#) ilustra uma consulta por similaridade com o SOLID. Mais especificamente, são detalhados a aplicação de fatores de compatibilidade consolidados ϖ , os cálculos de distâncias realizados pela Solid-Dist (ψ) sobre diferentes atributos, e por fim a normalização de distâncias e consolidação do resultado final da consulta. O processo é realizado para cada par de tuplas $\langle t_q, t_j \rangle \in \mathcal{D}$, os respectivos conjuntos de fatores de compatibilidade consolidados Υ_q e Υ_j e o diâmetro referente ao espaço métrico de cada atributo.

Na Figura 23-I é mostrada a primeira etapa em que cada ocorrência de dado faltante nas tuplas é identificada. Com isso, são associados os respectivos fatores de compatibilidade consolidados ϖ de cada atributo, para serem utilizados sobre os atributos compatíveis. No exemplo, são associados os fatores de compatibilidade consolidados $\varpi \in Y$ do atributo *Pulmão* = $\{0.1\}$ em t_q e *Cérebro* = $\{0, 2, 0.2, 0.5\}$ em t_j , respectivamente.

A segunda etapa, ilustrada pela Figura 23-II, detalha os cálculos de distâncias realizados pela Solid-Dist - ψ na comparação entre os atributos das tuplas $\langle t_q, t_j \rangle$. A função ψ somente considera a distância entre os pares de vetores de características $\langle v_{a,q}, v_{a,j} \rangle$ não nulos. Neste cenário, a função ignora os pares referentes aos atributos nulos, atribuindo valor de distância zero, uma vez que $\delta_a(v_{a,q}, v_{a,j}) = 0$, como nos atributos *cérebro* e *pulmão*. Por outro lado, nos atributos não nulos, como *Coração*, *Estômago* e *Olho*, são calculadas de acordo com $d = \delta_a(v_{a,q}, v_{a,j})$ sobre cada par de vetores de características, resultando nas distâncias d_2, d_4 e d_5 , respectivamente.

Por fim, na última etapa é realizada a normalização das distâncias resultantes entre os diferentes atributos e a consolidação do resultado final, o que é ilustrado na Figura 23-III. A normalização é justificada pois os múltiplos espaços métricos referentes aos atributos possuem diâmetros distintos. Logo, as distâncias resultantes são divergentes entre si para expressar a distância final entre tuplas. Para permitir as comparações, as distâncias são ponderadas conforme os respectivos fatores de compatibilidade consolidados ϖ que foram associados, por exemplo em $(0, 1 \times 0, 2 \times d_4)$ e $(0, 5 \times d_5)$. Na sequência, a distância (ponderada ou não) é dividida pelo diâmetro β_i do respectivo espaço métrico, definido sobre cada atributo da base de dados, de acordo com d_j/β_i . Com isso, as distâncias normalizadas são consolidadas, resultando em um valor real positivo como a distância final entre as tuplas calculadas, de acordo com $\psi(t_q, t_j) \geq 0$.

O resultado das consultas por similaridade em bases de dados incompletos com o SOLID retorna o conjunto de tuplas mais similares, conforme medido pela função Solid-Dist - ψ . Vale destacar que durante todo o processo da consulta, em nenhuma etapa é realizado o descarte de dados, como também não ocorre nenhuma aplicação de métodos de imputação de valores aos atributos com valores faltantes. Portanto, o SOLID é executado sobre a base de dados original sem qualquer manipulação dos elementos.

4.7 Considerações Finais

Este capítulo apresentou os conceitos que embasam a principal contribuição desta dissertação, que se constitui no método SOLID (*Search Over Correlated and Incomplete Data*) para consultas por similaridade sobre bases de dados correlacionados e incompletos. O SOLID implementa o método CorDiS (*Correlation of Distance Spaces*) para o mapeamento da correlação entre espaços métricos, gerando um mapa de correlação para os distintos atributos, considerando o conjunto de características extraídas por FEMs e funções de distância δ .

O método utiliza o mapa de correlação resultante do CorDiS para a geração de fatores

Figura 23 – Solid-Dist em consultas por similaridade.

Fonte: Elaborada pelo autor.

de compatibilidade ω para definir outros atributos compatíveis na execução de consultas por similaridade sobre bases de dados incompletos. Os fatores de compatibilidade são aplicados e consolidados pela função Solid-Dist que distribui importância entre os atributos compatíveis mais correlacionados a cada dado faltante, reduzindo a influência da falta de dados na recuperação de informação executando consultas por similaridade. Além disso, foi apresentada uma técnica para a normalização de distâncias sobre múltiplos espaços métricos na consolidação de consultas por similaridade com a função Solid-Dist para os cenários de aplicação em que o método pode ser utilizado. Por fim, foi exemplificada e detalhada a execução das etapas do SOLID em consultas por similaridade sobre bases de dados incompletos.

O Capítulo 5 apresenta experimentos, resultados e análises obtidos a partir da avaliação do SOLID em diferentes cenários explorados, que validaram sua aplicabilidade.

EXPERIMENTOS E RESULTADOS

5.1 Considerações Iniciais

Este capítulo detalha o processo de avaliação de qualidade e desempenho do método SOLID na execução de consultas por similaridade sobre conjuntos de dados complexos incompletos. São apresentados os experimentos a partir da execução do SOLID sobre várias coleções de imagens do mundo real. Foi utilizado um conjunto de FEMs de diferentes dimensionalidades a fim de que o método pudesse ser avaliado sobre diferentes representações dos dados complexos, e considerando diferentes cenários de dados faltantes nesses conjuntos.

O mapeamento de correlações sobre dados complexos pelo método CorDiS é realizado a partir da utilização de dois coeficientes de correlação distintos (Pearson e Spearman), analisados considerando diferentes limiares de correlação. O objetivo é analisar a efetividade da geração e aplicação de fatores de compatibilidade em consultas por similaridade, considerando a identificação de atributos correlacionados a partir de atributos com representações distintas.

O SOLID é comparado com cinco outros métodos que lidam com dados faltantes, como métodos de descarte de dados e imputações de valores. O foco é analisar a efetividade do método proposto e dos competidores na recuperação de tuplas de atributos complexos, considerando cada conjunto de dados completo e incompleto. Além disso, todos os métodos foram avaliados em relação ao tempo de execução nos dois tipos de consultas por similaridade.

Este capítulo está organizado nas seguintes seções: a [Seção 5.2](#) detalha os materiais e métodos utilizados nos experimentos. Na [Seção 5.3](#) é analisado o método CorDiS no mapeamento da correlação entre espaços métricos e a geração de fatores compatibilidade aplicação em consultas com o SOLID. A [Seção 5.4](#) detalha os experimentos e cenários em que o SOLID foi executado. Na [Seção 5.5](#) são apresentadas as análises de tempo dos métodos conforme os cenários de consultas executados com SOLID. A [Seção 5.6](#) discute os resultados dos experimentos com o método proposto. Por fim, na [Seção 5.7](#) são feitas as considerações finais deste capítulo.

5.2 Materiais e Métodos

Os extratores de características utilizados nos experimentos deste capítulo obtêm informações sobre cor, forma e textura, e foram aplicados para diversificar as representações de um mesmo objeto durante o processamento e armazenamento de dados complexos. A [Tabela 3](#) mostra os 13 FEMs utilizados e as respectivas dimensionalidades (d) geradas para representar um objeto complexo. Para o *Normalized Color Histogram* (NCH), foram utilizadas 5 variações de dimensionalidade. Com isso, os experimentos foram realizados utilizando 18 diferentes representações de características para cada imagem.

Tabela 3 – FEMs utilizados nos experimentos.

Acrônimo	d	Nome do FEM	Acrônimo	d	Nome do FEM
CT	3	Color Texture	LBP	177	Local Binary Patterns
TS	8	Texture Spectrum	SC	256	Scalable Color
CL	16	Color Layout	TCH	768	Total Color Histogram
Hr	24	Haralick	MH	506	Metric Histogram
Zr	36	Zernike	BIC	512	BIC Histogram
CS	128	Color Structure	NCH	8, 16, 32, 64, 128, 256	Normalized Color Histogram
EH	150	Edge Histogram			

Fonte: Dados da pesquisa.

Os experimentos foram executados usando seis conjuntos de imagens do mundo real. A [Tabela 4](#) apresenta as coleções de imagens selecionadas, organizadas por quantidade de objetos, cardinalidade e dimensionalidade. Todas as coleções selecionadas têm como característica comum a possibilidade de organização por tuplas compostas por uma lista de atributos complexos.

Tabela 4 – Bases de dados utilizadas nos experimentos do SOLID detalhados pela quantidades de objetos n e d sendo as cardinalidades e dimensionalidades, respectivamente.

Base de dados	Quantidade de Objetos	n	d
<i>DS-Spine</i> (BURIAN et al., 2019)	540	54	10
<i>DS-HandPD</i> (PEREIRA et al., 2016)	736	92	8
<i>DS-LibraGestures</i> (BASTOS; ANGELO; LOULA, 2015)	4800	120	40
<i>DS-Coil100</i> (NENE; MURASE, 1996)	7200	100	72
<i>DS-Cars</i> (YANG et al., 2015)	11544	1443	8
<i>DS-Letters</i> (HAJDER, 2020)	15340	295	52

A primeira coleção (*DS-Spine*) possui 540 imagens de ressonância magnética de múscu-

los lombares a partir de 54 pacientes com 10 exames cada (BURIAN *et al.*, 2019). O segundo conjunto (*DS-HandPD*) é composto por 736 imagens para diagnóstico da doença de Parkinson, envolvendo 92 pacientes (entre saudáveis e com Parkinson) com 8 de testes de caligrafia, considerando formas geométricas distintas (PEREIRA *et al.*, 2016). A terceira coleção (*DS-LibraGestures*) possui 4800 imagens de diferentes sinais de Libras do alfabeto, um conjunto de palavras e números selecionados, compondo de 120 testes com 40 sinais cada (BASTOS; ANGELO; LOULA, 2015). O quarto conjunto (*DS-Coil100*) envolve 7200 imagens de objetos distintos obtidas a partir da rotação sistemática em 360° de cada objeto, com uma imagem a cada cinco graus, composto por 100 objetos com 72 ângulos distintos (NENE; MURASE, 1996). A quinta coleção (*DS-Cars*) é composta por 11544 imagens entre diversas marcas e modelos de carros, organizados por partes de um carro (farol, farol traseiro, farol de nevoeiro, entrada de ar, console, volante, painel e alavanca de câmbio), envolvendo 1443 variações de modelos com 8 partes distintas para cada registro (YANG *et al.*, 2015). Por fim, o conjunto (*DS-Letters*) possui 15340 imagens de letras do alfabeto representadas por um conjunto de 295 fontes distintas (*Times New Roman*, *Verdana*, *Arial*, entre outras) com 52 letras cada (alfabeto maiúsculo e minúsculo). Nos experimentos de consultas por similaridade com o SOLID (Seção 5.4), foi utilizada a função de distância Euclidiana quando associada aos FEMs definidos para cada conjunto de dados durante a etapa de pré-processamento de dados complexos.

Todos os experimentos foram realizados em um computador com a seguinte especificação: processador Intel Core i7-8700@3.20GHz, memória RAM de 16GB@2.400MHz, HD 2TB SATA III 6.0Gb/s 7200RPM / SSD 256GB SATA III 6.0Gb/s e sistema operacional Linux Ubuntu 20.04 LTS 64-bit.

5.3 Correlações entre Espaços Métricos

O método CorDiS para extração de correlações entre espaços métricos foi utilizado neste trabalho em dois processos do SOLID, sumarizados pelos seguintes experimentos:

1. **Seleção de Representações Correlacionadas:** aplicação do método CorDiS para selecionar e utilizar um conjunto de representações correlacionados a partir das combinações entre os FEMs e função de distância utilizada.
2. **Compatibilidade de Atributos Complexos:** analisa a geração de atributos complexos compatíveis baseada na correlação extraída de um conjunto de espaços métricos.

Ambos os experimentos foram executados com o método CorDiS utilizando a função de distância **Euclidiana**, combinada aos coeficientes de correlação de **Pearson** e **Spearman**, respectivamente.

5.3.1 Seleção de Representações Correlacionadas

Este experimento tem como objetivo investigar sobre as correlações calculadas sobre várias representações distintas. Além disso, o experimento visa auxiliar na escolha de um conjunto de representações correlacionadas utilizando o CorDiS. Este processo foi repetido para cada conjunto de dados a partir da escolha de d FEMs para avaliar a correlação entre os espaços métricos resultantes. Analisando todas as possíveis representações, dado um conjunto de F extratores de características e d como a dimensionalidade de cada conjunto de dados, o número máximo de combinações é calculado pela [Equação 5.1](#):

$$\frac{(d + F - 1)!}{(d - 1)! \times F!} \quad (5.1)$$

Figura 24 – Quantidade de representações possíveis, de acordo com a [Equação 5.1](#), considerando 18 FEMs e dimensionalidades no intervalo de $[1, 100]$, passo 1.

Fonte: Dados da pesquisa.

No exemplo apresentado na [Figura 24](#), nota-se que a quantidade final de representações é extremamente alta, conforme a dimensionalidade e do tamanho do conjunto de FEMs aumentam. Isso pode ser visto com os FEMs utilizados nesse trabalho ($F = 18$). Na prática, considerando que a dimensionalidade do maior conjunto utilizado neste trabalho é 72, o custo computacional desta tarefa se torna bastante elevado. Desso modo, para escolher uma boa combinação foram realizados 10 testes, sendo que para cada teste, foram selecionados d FEMs aleatoriamente, e avaliado o valor médio de correlação entre as representações resultantes de cada combinação. Ao final de todos os testes, foi escolhida a combinação com o maior valor médio de correlação sobre os atributos complexos.

A Figura 25 apresenta o gráfico com o valor de correlação dos testes de seleção de representações correlacionadas realizado sobre o conjunto de dados *DS-Spine*. Durante os testes, a melhor combinação gerada apresentou valor médio de correlação de 0,495, resultante do conjunto de d FEMs: {NCH32, NCH128, NCH256, Zr, NCH128, NCH8, NCH16, NHC256, CL, NCH128}. Vale notar que para todos os conjuntos de dados, o CorDiS não explorou todas as combinações possíveis para se encontrar a melhor combinação. Porém, o alvo aqui é realizar a escolha de uma boa combinação com representações correlacionadas entre si, consequentemente, apresentando atributos complexos correlacionados.

Figura 25 – Seleção de representações utilizando o método CorDiS sobre o conjunto *DS-Spine*.

Fonte: Dados da pesquisa.

5.3.2 Compatibilidade entre Atributos Complexos

Esta seção avalia o segundo passo do método SOLID, ou seja, o método CorDiS. É analisado o mapeamento de correlações sobre o conjunto de espaços métricos de cada conjunto de dados, além de analisar o comportamento na identificação de atributos compatíveis por níveis de valores de correlação.

As Figuras 26 e 27 ilustram os mapas de correlação resultante do CorDiS sobre os conjuntos *DS-Spine* e *DS-Letters*, respectivamente. A escala de correlação é feita de maneira que atributos com valores maiores de correlação são apresentados em cores mais escuras, enquanto pares com menores correlações em cores mais claras. Vale notar que em ambos os conjuntos de dados existem vários pares de atributos correlacionados em casos de alta correlação (valores próximo a 1).

A Figura 28 apresenta os gráficos sobre a identificação de atributos complexos compatíveis e a geração de fatores de compatibilidade sobre os conjuntos de dados, respectivamente. O eixo horizontal de ambos os gráficos indica os limiares de correlação utilizados no experimento

Figura 26 – Mapa de correlação extraído do conjunto de dados *DS-Spine*.

Fonte: Dados da pesquisa.

e o eixo vertical indica as porcentagens proporcionais de atributos compatíveis em relação ao total de atributos que podem ser identificados em cada conjunto.

Na Figura 28a são apresentadas as porcentagens relativas aos atributos compatíveis por cada faixa de correlação, variando o limiar de correlação em incrementos de 0,1 para o intervalo $[0, 1]$. Com isso, é mapeada a quantidade sobre cada intervalo considerando somente atributos com valores de correlação até cada limiar definido. É possível observar que na maioria dos conjuntos de dados foram identificados atributos compatíveis em vários limiares de correlação, em especial, até o limiar de correlação $\gamma = 0,5$. Em faixas com limiares $\gamma \geq 0,5$, a identificação de atributos é mais restritiva, porém são selecionados os atributos complexos mais correlacionados, como nos conjuntos *DS-Spine*, *DS-Coil100* e *DS-Letters* para os limiares de correlação 0,7, 0,8 e 0,9.

Na Figura 28b é apresentada a quantidade acumulada de atributos compatíveis quando definido um limiar de correlação γ . São agrupados todos os atributos que apresentam valores

Figura 27 – Mapa de correlação extraído do conjunto de dados *DS-Letters*.

Fonte: Dados da pesquisa.

de correlação maiores ou iguais ao γ , sendo estes os atributos selecionados e que geram os fatores de compatibilidade a serem aplicados e consolidados em consultas por similaridade neste trabalho. Os conjuntos *DS-Spine*, *DS-Coil100* e *DS-Letters* apresentam para todos os limiares um ou mais atributos, mesmo para os limiares mais altos, como $\gamma \geq 0,7$. Além disso, os conjuntos *DS-HandPD* e *DS-LibraGestures* apresentam atributos com valores de correlação para qualquer limiar até $\gamma = 0,7$. Por outro lado, o conjunto *DS-Cars* apresenta atributos com valores de correlação para qualquer $\gamma \leq 0,2$, sendo o conjunto com atributos complexos menos correlacionados entre si.

A [Figura 29](#) mostra um comparativo sobre limiares de correlação para o intervalo $[0,5, 1,0]$ com incremento 0,1, avaliados com a execução de consultas por similaridade com o SOLID nos conjuntos *DS-Spine* e *DS-Letters*. Foram executadas consultas por vizinhança (kNN_q)

(a) Atributos Compatíveis por faixa de correlação.

(b) Acumulado de atributos compatíveis para cada conjunto de dados.

Figura 28 – Quantidades de atributos complexos compatíveis gerados por faixas de correlação.

Fonte: Dados da pesquisa.

utilizando valores de k de 1 a 21, passo 2, e variando a quantidade de dados faltantes entre 10% a 50%, com incremento de 10% em cada teste. Além disso, a efetividade é medida pelo coeficiente de Jaccard entre o resultado da consulta sobre o conjunto completo e o resultado com SOLID. Em ambos os conjuntos são gerados atributos compatíveis para o maior limiar de correlação $\gamma = 0,9$ (em torno de 20% e 6,2%), como observado na Figura 28.

Analisando a qualidade dos fatores de compatibilidade gerados, é possível observar a boa aplicabilidade dos limiares de correlação $\gamma \geq 0,5$. Ao observar a quantidade de atributos compatíveis em relação ao limiar mais alto ($\gamma = 0,9$ - linhas vermelhas da Figura 29), pode-se afirmar que é o limiar mais efetivo, uma vez que gera-se menos atributos compatíveis mas realizam as consultas com qualidade similar ou superior aos limiares inferiores. Esse comportamento mostra que o SOLID é especialmente útil quando aplicado sobre bases de dados mais correlacionadas, pois são identificados atributos compatíveis que geram fatores de compatibilidade mais efetivos em reduzir a influência dos dados faltantes em consultas.

Figura 29 – Comparativo entre limiares de correlação em relação a efetividade em consultas por vizinhança.

Fonte: Dados da pesquisa.

Nos experimentos de consultas por similaridade apresentados a seguir, os limiares de correlação são selecionados de acordo com a quantidade de atributos compatíveis no limiar mais alto válido, ou seja, que apresentem um ou mais atributos compatíveis, para cada conjunto de dados. Com isso, o limiar escolhido para cada o conjunto de dados é: *DS-Spine* e *DS-Letters* com $\gamma = 0,9$, *DS-HandPD*, *DS-LibraGestures* e *DS-Coil100* com $\gamma = 0,5$, respectivamente. Especificamente para o conjunto *DS-Cars*, que é o que possui menos atributos correlacionados entre si, as consultas serão realizadas utilizando o limiar em $\gamma \geq 0,2$, pois em limiares maiores não são identificados atributos compatíveis.

5.4 Análise Quantitativa em Consultas por Similaridade

Neste experimento é analisada a efetividade da aplicação dos fatores de compatibilidade pelo SOLID quando comparados com os seguintes competidores:

- **Descarte de Dados:** realiza consultas considerando todas as tuplas completas, descartando casos de atributos faltantes durante os cálculos de distâncias;
- **Imputação por Média:** infere valores médios utilizando todos os objetos de determinado atributo;
- **Imputação por kNN:** infere valores baseado nos 5 vizinhos mais próximos de um atributo;
- **Imputação por Regressão:** infere valores baseado na predição de valores a partir da interpolação dos 5 vizinhos mais próximos;

- **Imputação por Árvores de Decisão:** infere valores baseado em uma árvore de decisão considerando a quantidade máxima de objetos de acordo com $\sqrt{n}$, sendo n a cardinalidade do conjunto de dados.

Para analisar a qualidade do SOLID e seus competidores na recuperação de informação, foram executadas consultas por similaridade a partir de dois cenários distintos para a falta de dados:

- **Cenário I - Centro de Consulta Incompleto:** é analisado o quão bem SOLID recupera informação sobre conjuntos de dados completos quando executado com um **centro de consulta com valores faltantes** (s_q);
- **Cenário II - Bases Incompletas:** é analisado se o SOLID reduz o erro em consultas por similaridade quando executado sobre **bases de dados com grandes proporções de dados faltantes**.

Os experimentos foram realizados de forma padronizada para ambos os cenários quando executadas consultas por vizinhança (kNN_q) e consultas por abrangência (R_q). A execução de consultas foi realizada sobre as bases de dados quando inseridas aleatoriamente as seguintes taxas de dados faltantes: 10%, 20%, 30%, 40% e 50%. Em específico para o Cenário I, são inseridas as respectivas taxas de dados faltantes de acordo com a quantidade de atributos do centro de consulta, por exemplo s_q com 10 atributos e taxa de 50% corresponde a metade da tupla incompleta, e assim para outras taxas de dados faltantes.

Para consultas por vizinhança (kNN_q) foram executadas para diferentes valores de k : 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 e 21. Os valores de raios definidos são equivalentes à proporção do espaço de distâncias da consulta, uma vez que todas as distâncias resultantes são normalizadas. Para valores de raio $\varepsilon > 0,55$ do diâmetro, em geral, é retornado mais de aproximadamente 92% das tuplas do conjunto de dados.

A normalização de distâncias aplicada neste trabalho não altera o espaço de distâncias de cada atributo da consulta. Portanto, a aplicação de fatores de compatibilidade gerados a partir da correlação entre os espaços métricos não normalizados não modifica o relacionamento linear dos atributos, pois não altera a distribuição dos dados (SCHÖBER; BOER; SCHWARTE, 2018; MUKAKA, 2012).

A Figura 30 apresenta os intervalos de distâncias sobre o conjunto de FEMs para os conjuntos *DS-Spine* e *DS-HandPD*. É possível analisar a necessidade da normalização de cada FEM com seu respectivo diâmetro (β) de modo que cada distância resultante de um espaço de distância seja equivalente para cada outro espaço de distância existente na consulta, ao serem consolidados pela função Solid-Dist (ψ).

Figura 30 – Intervalo de valores de distância para cada FEM utilizado, quando aplicada a função Euclidiana para os conjuntos *DS-Spine* e *DS-HandPD*, respectivamente.

Fonte: Dados da pesquisa.

5.4.1 Cenário I: Centro de Consulta Incompleto

No primeiro cenário é avaliado como o SOLID e seus competidores executam consultas por similaridade com **centros de consulta incompletos**. Cada consulta é executada com um centro de consulta distinto utilizando todos as tuplas do conjunto de dados, sendo comparada com o restante do conjunto, ou seja, é uma execução de todas para todas as tuplas. Ao final os resultados são sumarizados para medir a qualidade das consultas neste cenário. A qualidade dos métodos foi medida utilizando o **coeficiente de Jaccard** entre o conjunto resposta de consultas por vizinhança (kNN_q) e abrangência R_q) quando executadas sobre a base de dados completos (*ground truth*) em comparação com o resultado de cada método. Utilizando esta métrica, é possível quantificar a semelhança entre o conjunto resposta do *ground truth* em comparação cada método testado.

As Figuras 31 e 32 mostram a avaliação deste experimento para consultas por vizinhança e abrangência, respectivamente. Para todos os gráficos, quanto maior o valor de Jaccard resultante, maior a qualidade do método em questão para o valor de k ou *raio* (ϵ) utilizado durante a execução da consulta.

Consultas por Vizinhança (kNN_q)

Analisando os resultados executando consultas kNN (Figura 31), é possível observar que o SOLID (linhas vermelhas) é melhor que todos seus competidores na maioria dos casos. Além disso, é destacado que o método é melhor, especialmente quando se aumenta

Figura 31 – Consultas por Vizinhança (kNN_q) considerando **centros de consultas incompletos** em que **quanto maior o valor de Jaccard, melhor o resultado**.

Fonte: Dados da pesquisa.

a proporção de dados faltantes nas consultas. Dada uma tupla como centro de consulta (t_i), e considerando as distintas dimensionalidades entre os conjuntos selecionados (8 a 72), o SOLID necessita aplicar fatores de acordo com a proporção de dimensões faltantes. Visto que o método obtém bons resultados na maioria dos casos, **é possível concluir que os fatores de compatibilidade aplicados reduzem a influência de cada dimensão faltante da tupla,**

apontando que os limiares (γ) definidos são significativos nesses conjuntos de dados. Com isso, é observado que a utilização da correlação entre atributos complexos é viável em consultas por vizinhança, independente da variação do valor de k (1 a 21) neste cenário de aplicação. Por fim, o SOLID executou consultas por vizinhança de forma estável em todos os conjuntos de dados em relação a variação de cardinalidade (54 a 1443), independentemente do valor de k e da taxa de dados faltantes aplicada.

A Tabela 5 apresenta a consolidação do ganho médio e máximo do SOLID em relação aos seus competidores. Esta consolidação é calculada a partir de todos os valores de Jaccard entre todas as porcentagens de dados faltantes utilizadas anteriormente, organizada para cada método competidor em relação ao SOLID, e por cada conjunto de dados. Além disso, é destacado em verde o maior de ganho médio do SOLID, em cada conjunto de dados avaliado.

Tabela 5 – Consolidação de ganhos de qualidade (em %) do SOLID em relação a todos os competidores em consultas por vizinhança (kNN_q) com centros de consulta incompletos.

Base de Dados	Descarte de Dados		Imp. por Média		Imp. por kNN		Imp. por Regressão		Imp. por Árvores de Decisão	
	Ganho Médio	Ganho Máximo	Ganho Médio	Ganho Máximo	Ganho Médio	Ganho Máximo	Ganho Médio	Ganho Máximo	Ganho Médio	Ganho Máximo
<i>DS-Spine</i>	200,30	219,13	1,31	4,59	6,91	13,39	7,11	13,576	5,25	9,85
<i>DS-HandPD</i>	262,74	326,30	262,74	326,30	11,83	20,70	13,94	24,50	12,83	23,74
<i>DS-LibraGestures</i>	447,07	496,07	447,07	496,07	1,87	8,92	1,95	10,28	2,09	11,00
<i>DS-Coil100</i>	628,41	671,36	9,41	25,24	4,59	14,59	4,56	13,95	5,01	15,49
<i>DS-Cars</i>	2607,41	3760,56	-18,82	-15,52	4,15	9,84	3,71	8,50	2,73	6,16
<i>DS-Letters</i>	756,07	818,64	3,55	24,68	0,09	15,07	0,14	14,89	-0,56	14,51

Fonte: Dados da pesquisa.

Analisando o SOLID em relação ao método que não considera atributos faltantes em centros de consulta, é observado que **o método é até três ordens de magnitude melhor, além de ganhar em todos os casos do Descarte de Dados.** Em relação aos métodos de imputação de valores, o SOLID é em média 13,94% e 12,83% e no máximo 24,50% e 23,74% melhor do que a Imputação por Regressão e Imputação por Árvores de Decisão, respectivamente, sendo estes os melhores competidores.

Na maioria dos casos consolidados, o SOLID **apresenta ganhos médios maiores do que seus competidores, concluindo que o método lida bem com consultas por vizinhança**, considerando a variação de k e proporção de dados faltantes aplicadas, como também a dimensionalidade e a cardinalidade dos conjuntos de dados. Em relação aos ganhos máximos, o SOLID apresenta resultados que variam entre 4,59% até três ordens de magnitude superior aos competidores.

O SOLID executa consultas por vizinhança (kNN_q) **melhor sobre conjuntos com atributos mais correlacionados na maioria dos casos**, como em *DS-Spine*, *DS-HandPD*, *DS-*

LibraGestures, *DS-Coil100* e *DS-Letters*. No entanto, o conjunto *DS-Cars* é o que apresenta menos atributos compatíveis, sendo que neste caso o SOLID perde na maioria dos casos. Analisando o limiar de correlação utilizado neste conjunto ($\gamma \geq 0,2$) em relação aos outros conjuntos, é possível notar que o método **SOLID lida melhor com o problema de dados faltantes quando executado em conjuntos mais correlacionados**, sendo este conjunto de dados não indicado, por ser composto por atributos pouco correlacionados entre si. Por fim, o conjunto *DS-Letters* apresenta a tendência de melhora dos resultados do SOLID conforme o aumento considerável proporção de dados faltantes, em particular, empatando e ganhando nas faixas de (40% e 50%) em comparação com as abordagens concorrentes. Com isso, é possível observar que o **método proposto lida melhor com maiores quantidades de dados faltantes do que os métodos de descarte e imputação**, analisados neste estudo.

Consultas por Abrangência (R_q)

Os resultados sobre as consultas R_q (Figura 32) mostram que o SOLID **recupera informação com qualidade superior na maioria dos casos em relação aos seus competidores**, considerando a variação de raio (ϵ) aplicado e as diferentes proporções de dados faltantes. O SOLID é melhor em relação aos seus competidores para todos os casos, considerando $\epsilon = 0,05$, sendo este o menor raio. Nos outros raios aplicados, **o método ganha na maioria dos casos em relação a todos os competidores**, mostrando que a aplicação de fatores de compatibilidade em consultas por abrangência também reduz a influência dos dados faltantes neste cenário.

Analisando os resultados para o conjunto *DS-Cars*, o SOLID ganha em todas as faixas de dados faltantes para $\epsilon = 0,05$ e $\epsilon = 0,15$. Em particular para este conjunto, para $\epsilon = 0,25$ o método apresenta perda em relação a seus competidores, sendo que para os raios restantes, no geral o método é próximo ou superior aos métodos de imputação, apresentando esta tendência em todas as faixas de dados faltantes. Para o conjunto *DS-Letters* é apresentada uma tendência específica para $\epsilon = 0,15$, em que SOLID perde para métodos de imputação nas faixas de 10%, 20% e 30%, mas empata e ganha nas faixas posteriores, mostrando maior aplicabilidade dos fatores de compatibilidade em 40% e 50% de dados faltantes. Para os outros conjuntos (*DS-Spine*, *DS-HandPD*, *DS-LibraGestures* e *DS-Coil100*), o SOLID é melhor em quase todos os raios e faixas de dados faltantes, mostrando que a geração de fatores de compatibilidade a partir do *threshold* de correlação definido ($\gamma \geq 0,5$) é significativa.

A Tabela 6 apresenta a consolidação do ganho médio e máximo do SOLID em relação aos seus competidores para consultas por abrangência (R_q) neste cenário, e calculada de modo similar à apresentada na Tabela 5. Além disso, é destacado em verde o maior de ganho médio do SOLID, em cada conjunto de dados avaliado.

Neste cenário, o SOLID executa consultas R_q **melhor que todos seus competidores para todos os conjuntos de dados, considerando ganho médio e máximo**. Em relação aos ganhos máximos, o método é até duas ordens de magnitude melhor que Imputação por Árvores

Figura 32 – Consultas por Abrangência (R_q) considerando **centros de consultas incompletos** em que **quanto maior o valor de Jaccard, melhor o resultado**.

Fonte: Dados da pesquisa.

de Decisão, Regressão e KNN e até três ordens de magnitude em relação a Imputação por Média e o Descarte de Dados. Com isso, é possível afirmar que o **SOLID reduz consideravelmente a influência de dados faltantes neste cenário, independente da proporção de dados faltantes aplicada**.

Tabela 6 – Consolidação de ganhos de qualidade (em %) do SOLID em relação a todos os competidores executando **consultas por abrangência (R_q) com centros de consulta incompletos**.

Base de Dados	Descarte de Dados		Imp. por Média		Imp. por kNN		Imp. por Regressão		Imp. por Árvores de Decisão	
	Ganho Médio	Ganho Máximo	Ganho Médio	Ganho Máximo	Ganho Médio	Ganho Máximo	Ganho Médio	Ganho Máximo	Ganho Médio	Ganho Máximo
<i>DS-Spine</i>	233,21	247,22	233,21	247,22	85,09	157,57	92,91	170,10	89,13	158,83
<i>DS-HandPD</i>	280,17	309,98	280,17	309,98	48,36	80,09	47,10	87,94	46,18	87,70
<i>DS-LibraGestures</i>	2826,00	2907,00	2826,00	2907,00	82,36	173,74	83,74	176,08	81,29	156,46
<i>DS-Coil100</i>	980,97	1034,04	41,86	82,14	83,57	170,78	81,66	165,24	81,93	161,17
<i>DS-Cars</i>	167,36	201,93	7,00	16,32	18,83	33,65	19,57	36,02	18,70	33,40
<i>DS-Letters</i>	74,45	81,92	-5,88	10,47	8,33	28,90	8,76	31,26	8,56	28,70

Fonte: Dados da pesquisa.

5.4.2 Cenário II: Bases Incompletas

No segundo cenário é avaliado como o SOLID e seus competidores reduzem o erro em consultas por similaridade sobre conjuntos de dados com grandes proporções de dados faltantes. A taxa de erro foi avaliada com coeficiente de Jaccard, mas neste cenário é calculada como um menos o valor de Jaccard resultante ($1 - \text{Coef. Jaccard}$) entre o conjunto resposta de consultas sobre dados completos e o conjunto resposta de consultas sobre incompletos para cada método, de acordo com as proporções de dados faltantes e valores de k e ϵ utilizados.

As Figuras 33 e 34 ilustram a avaliação da taxa de erro do SOLID em relação aos seus competidores. Para ambas as figuras, o eixo x é referente a taxa de erro dos competidores e o eixo y é a taxa de erro do SOLID. Cada ponto é referente a um k ou ϵ utilizado na respectiva consulta e cada cor diferente corresponde a um competidor avaliado em relação ao SOLID, como mostrado na legenda da figura. A região em amarelo (diagonal inferior) corresponde a todos os casos em que o SOLID reduz mais o erro em relação a cada competidor avaliado.

Analisando os resultados das consultas neste cenário, é possível afirmar que o SOLID **consegue reduzir o erro de forma similar ou superior que seus competidores na maioria dos casos e para diferentes proporções de dados faltantes**. Além disso, é observado **a aplicabilidade do SOLID em grandes proporções de dados faltantes (em torno de 50%), sendo que nesta faixa o método lida melhor que os concorrentes com a recuperação de tuplas em bases incompletas com maior precisão**. Uma vantagem do SOLID considerando essa faixa de falta de dados é que o método não introduz distorções no conjunto de dados quando comparado a qualquer dos concorrentes que imputam valores.

Em termos de ganhos sobre seus concorrentes, o SOLID é até 26,89% melhor em reduzir o erro em relação ao descarte de dados. Em relação aos métodos de imputação de valores, o SOLID é até 15,71%, 5,82%, 1,74% e 2,03% melhor do que os métodos de Imputação por Média, kNN, Regressão e Árvores de Decisão, respectivamente. Portanto, **o método desenvolvido**

Figura 33 – Comparativo do erro do SOLID em relação aos seus competidores para consultas k NN.

apresenta maior precisão do que seus concorrentes em recuperar informação sobre bases com grandes proporções de dados faltantes neste cenário.

Os resultados das consultas R_q sobre bases incompletas (Figura 34) mostram que o SOLID reduz o erro na maioria dos casos, de modo similar ou melhor do que em consultas k NN. Neste cenário, é possível identificar que **o método executa consultas com maior precisão conforme se aumenta a proporção de dados faltantes**, uma vez que há uma tendência de

haver mais pontos na região em amarelo. Além disso, quando os competidores perdem, eles tendem a perder significativamente mais do que o SOLID quando este perde. Com isso, é observado que o **SOLID recupera informação com maior precisão em grandes proporções de dados faltantes melhor do que seus concorrentes, mostrando a aplicabilidade dos fatores de compatibilidade baseados na correlação entre atributos complexos.**

Figura 34 – Comparativo do erro do SOLID em relação aos seus competidores para consultas R_q .

Fonte: Dados da pesquisa.

Vale notar, casos em que o SOLID perde em relação aos seus competidores, o principal adversário é o descarte dados. Em particular, é possível afirmar que o descarte ganha somente nos casos no qual é possível calcular a resposta, ou seja, sem valores faltantes durante a consulta.

Analisando o ganho do SOLID em relação a seus concorrentes, é possível destacar que o método é até 51,42% melhor em reduzir o erro em relação ao descarte de dados. Considerando os métodos de imputação de valores, SOLID é até 56,30%, 45,10%, 48,13% e 22,85% melhor do que inferir valores por Média, kNN, Regressão e Árvores de Decisão, respectivamente.

5.5 Análise de Tempo em Execução de Consultas

As Tabelas 7 e 8 mostram a média de tempo de execução do SOLID e seus competidores quando executadas consultas por vizinhança (kNN_q) e abrangência (R_q), respectivamente, sobre todos os conjuntos de dados selecionados. Foram coletados os tempos médios de cada método a partir de 100 execuções de consultas com centro de consultas aleatórios, considerando cada conjunto de dados com 50% de dados faltantes. Além disso, foi selecionado o valor de $k = 21$ para consultas por vizinhança (kNN_q) e valor de raio $\varepsilon = 0,55$ para consultas por abrangência (R_q). Para ambas as tabelas (7 e 8), destaques em verde representam o método mais rápido e destaques em vermelho representam o método mais lento em cada caso, desconsiderando o descarte de dados, uma vez que não calcula casos de atributos incompletos, consequentemente realiza menos cálculos de distância.

Tabela 7 – Tempo médio de execução (em segundos) dos métodos em **consultas por vizinhança** (kNN_q).

Base de Dados	SOLID	Descarte de Dados	Imp. por Média	Imp. por kNN	Imp. por Regressão	Imp. por Árvores de Decisão
DS-Spine	0,680	0,606	0,719	0,766	5,626	7,274
DS-HandPD	0,643	0,605	0,715	0,794	1,764	2,938
DS-LibraGestures	0,893	0,664	1,185	3,134	32,163	58,343
DS-Coil100	1,024	0,668	1,411	3,642	78,538	102,45
DS-Cars	1,153	0,791	3,102	2,788	92,281	12,455
DS-Letters	0,986	0,663	1,484	3,337	33,277	20,045

Fonte: Dados da pesquisa.

O SOLID é mais rápido que todos os outros métodos competidores, considerando somente as abordagens que lidam com a falta de dados. Vale notar que **o método executa consultas por vizinhança (kNN_q) e abrangência (R_q), respectivamente, em até 100 e 98,5 vezes mais rápido do que Imputação por Árvores de Decisão, sendo este o método mais lento**, no geral. Em relação a Imputação por Regressão, o SOLID é até **80 e 76,8 vezes mais rápido** para as consultas por vizinhança (kNN_q) e abrangência (R_q), respectivamente. Além disso, **o SOLID**

Tabela 8 – Tempo médio de execução (em segundos) dos métodos em **consultas por abrangência** (R_q).

Base de Dados	SOLID	Descarte de Dados	Imp. por Média	Imp. por kNN	Imp. por Regressão	Imp. por Árvores de Decisão
<i>DS-Spine</i>	0.689	0.613	0.710	0.771	5.569	7.284
<i>DS-HandPD</i>	0.661	0.606	0.718	0.803	1.774	2.957
<i>DS-LibraGestures</i>	0.912	0.662	1.176	3.162	32.160	58,302
<i>DS-Coil100</i>	1,042	0,670	1,406	3,648	78,005	102,59
<i>DS-Cars</i>	1,201	0,796	3,117	2,807	92,241	12,517
<i>DS-Letters</i>	1,004	0,667	1,480	3,222	33,247	20,018

Fonte: Dados da pesquisa.

mantém o tempo de execução menor e estável em todos os casos quando considerado as cardinalidades e dimensionalidades distintas dos conjuntos de dados, podendo-se concluir que o método lida com o problema de consultas por similaridade em bases de dados incompletos com eficiência.

5.6 Discussão de Resultados

Analisando as tendências e os comportamentos do SOLID sobre a aplicabilidade da correlação entre espaços métricos em relação aos resultados executando consultas por similaridade, é possível afirmar que:

- I - Em conjuntos com atributos mais correlacionados ($\gamma \geq 0,5$) o método é melhor pois são gerados fatores por compatibilidade significativos como nos conjuntos *DS-HandPD*, *DS-LibraGestures*, *DS-Coil100*, em especial para os conjuntos *DS-Spine* e *DS-Letters* que foi aplicado ($\gamma = 0,9$);
- II - Analisando em específico o conjunto *DS-Cars*, o método perde em vários casos, uma vez que existem poucos atributos compatíveis, além de pouco correlacionados, destacando que fatores por compatibilidade gerados a partir de limiares baixos, como $\gamma = 0,2$, resultam em pouca efetividade em reduzir a influência de dados faltantes;
- III - O método apresenta maior qualidade em recuperar informação em relação aos seus competidores quando executada consultas sobre bases com grandes proporções de dados faltantes (atingindo até 40 a 50% dos dados faltantes). **Esta tendência é destacada na maioria dos casos e em ambos os cenários de consultas de vizinhança e abrangência analisados;**

IV - O método é mais rápido que todos os métodos competidores de imputação de valor, em todos os cenários analisados e em todos os tipos de consulta.

A partir dos aspectos apresentados, é possível afirmar que o SOLID é um método que lida muito bem com o problema de dados faltantes em bases incompletas. Analisando e validando o método desenvolvido, é observado que o método é efetivo em vários cenários executados, como também em conjuntos com variações em relação a correlação dos dados (a partir de fatores de correlação $\gamma \geq 0,5$), além de ser mais rápido que seus competidores diretos.

5.7 Considerações Finais

Neste capítulo foi apresentada a análise experimental para validação do método SOLID. Foram utilizados seis conjuntos de dados reais referentes a imagens de diversas aplicações, apresentando distintas cardinalidades e dimensionalidades. Todos os conjuntos foram processados a partir da escolha dos FEMs mais correlacionados entre 18 possíveis combinações distintas, quando associados a função Euclidiana. Desse modo, foi possível avaliar a qualidade e o desempenho do SOLID em vários cenários de consultas por similaridade, como também definir as características dos conjuntos de dados para os quais o SOLID é adequado, e com isso estabelecer as condições de efetividade dos fatores de compatibilidade que são dependentes dos fatores de correlação entre atributos complexos.

Os resultados mostram que o SOLID no cenário em que se executa consultas por vizinhança e abrangência com centros de consulta incompletos, o método é de 4,59% a até três ordens de magnitude melhor do que métodos de imputação. Para outro cenário analisado, na execução de consultas sobre bases de dados incompletos, o SOLID consegue reduzir o erro em mais de 55%, destacando que o método consegue lidar com grandes proporções de dados faltantes (até em torno de 50%) com maior precisão do que seus competidores. Além disso, o SOLID consegue executar consultas até 100x mais rápido que métodos de imputação. Portanto, o método proposto neste trabalho é capaz de executar consultas por similaridade utilizando a correlação entre atributos complexos com eficácia e eficiência, uma vez que reduz a influência dos dados faltantes na maioria dos casos em relação a seus competidores.

CONCLUSÕES

6.1 Considerações Iniciais

As tarefas de recuperação de informação sobre bases incompletas demandam métodos eficientes e escaláveis. Neste cenário, novos métodos que lidam com a execução de consultas por similaridade sobre conjuntos de dados incompletos se tornam extremamente relevantes, sendo este tópico recentemente explorado em diversas aplicações e pesquisas recentes no contexto de Bases de Dados.

Consultas por similaridade em bases incompletas são executadas somente quando são descartados os objetos incompletos, aplicados métodos de imputação de valores ou mesmo a indexação de dados incompletos, uma vez que funções de distâncias tradicionais não calculam a distância entre pares de objetos com valores faltantes. Este trabalho explorou uma abordagem para o problema de dados faltantes que não é baseada nas estratégias citadas anteriormente, devido a limitações como perda de informações úteis e inserção de distorções no conjunto de dados.

Neste trabalho foi proposto o método SOLID - *Search Over Correlated and Incomplete Data*, que executa consultas por similaridade sobre bases incompletas. O SOLID implementa o método CorDiS - *Correlation of Distance Spaces*, que mapeia a correlação entre os espaços métricos do conjunto de dados para identificar atributos complexos correlacionados em relação aos casos de dados faltantes. Com isso, é possível gerar fatores de compatibilidade utilizando sub-conjuntos de atributos compatíveis com o intuito de reduzir a influência de objetos faltantes em consultas por similaridade. A aplicação dos fatores e consolidação das distâncias resultantes é realizada pela função Solid-Dist (ψ), que aplica uma normalização dos espaços de distâncias, possibilitando medir a distância entre tuplas do conjunto de dados.

A análise experimental apresentada no [Capítulo 5](#) mostra a aplicação do SOLID sobre diferentes conjuntos de dados complexos do mundo real. A execução do CorDiS mostra que a

geração de fatores de compatibilidade quando utilizado limiares maiores de correlação, como $\gamma \geq 0.7$, é mais efetiva mesmo que se utilize menos atributos compatíveis. Vale destacar que em conjuntos com atributos complexos mais correlacionados, o SOLID é mais efetivo em reduzir a influência de dados faltantes em consultas. A análise quantitativa realizada utilizando o coeficiente de Jaccard indica que o SOLID é mais de 55% melhor do que métodos de imputação ao recuperar informação, considerando consultas sobre grandes proporções de dados faltantes (até 50%). Além disso, o método executa consultas por similaridade até 100x mais rápido que seus concorrentes, pois não realiza processos adicionais em tempo de execução, como a imputação de valores, além de considerar todos os objetos da base de dados sem descartá-los. Por fim, o presente trabalho colabora com a oportunidade de tópicos de pesquisas adicionais, para a utilização do SOLID em outros tipos de dados ou áreas de aplicação, descritos a seguir na Seção 6.4.

6.2 Principais Contribuições

O SOLID difere dos métodos concorrentes e contribui ao estado da arte no contexto de Recuperação de Informação em Bases de Dados Incompletos nos seguintes aspectos:

- I - Propõe, implementa e valida um método para execução de consultas por similaridade por vizinhança e abrangência sobre bases de dados complexos em que ocorrem dados faltantes, sem imputação de valores e nem descarte de objetos. O método recupera informação em menor tempo de execução que seus concorrentes, e com menores taxas de erros quando executa consultas sobre bases de dados, mesmo com grandes proporções de dados faltantes em vários cenários;
- II - Desenvolvimento do método CorDiS para o mapeamento da correlação entre espaços métricos, que mede o quanto dois espaços de distâncias são correlacionados, permitindo a identificação de atributos correlacionados a atributos incompletos, considerando conjuntos de dados com baixa e alta correlação;
- III - Geração de fatores de compatibilidade entre atributos complexos baseada no conjunto de atributos compatíveis identificados. Com isso, é possível distribuir pesos entre os atributos mais importantes, reduzindo a influência de dados faltantes em consultas por similaridade;
- IV - O desenvolvimento de uma função de distância, chamada Solid-Dist (ψ), para medir a distância entre tuplas em bases de dados incompletos. A função consolida as distâncias resultantes entre o conjunto de espaços de distâncias, e aplica a normalização de distâncias baseados no diâmetro de cada espaço de distância.

6.3 Trabalhos Publicados

Nesta seção é apresentada a principal publicação realizada durante o desenvolvimento deste trabalho de mestrado. Além disso, são apresentados outros trabalhos publicados resultantes de outras atividades realizadas em colaboração com o grupo de pesquisa, que contribuíram com o aprendizado do mestrando ao longo do desenvolvimento de seu projeto.

6.3.1 *Publicação Principal*

- **Taking Advantage of Highly-Correlated Attributes in Similarity Queries with Missing Values.** *RODRIGUES, L. S., CAZZOLATO, M. T., TRAINA, A. J. M. e TRAINA-Jr, C.. In 13th International Conference on Similarity Search and Applications. SISAP 2020. Lecture Notes in Computer Science, vol 12440. Springer, Cham. DOI: 10.1007/978-3-030-60936-8_13. [QUALIS-CC A4].*

Neste trabalho foi proposto o método SOLID, que permite consultas por similaridade sobre bases de dados complexos sem a imputação de valores e nem descarte de dados. O método explora espaços métricos altamente correlacionados para a geração de fatores de compatibilidade a partir da identificação de atributos compatíveis. Além disso, o SOLID utiliza uma função de distância ponderada para aplicar os fatores de compatibilidade em atributos compatíveis, diminuindo a influência de casos de dados faltantes, quando executadas consultas por similaridade em tuplas da base de dados. Os resultados mostraram que a abordagem é melhor que métodos de imputação em consultas sobre várias taxas de dados faltantes, além de ser mais rápida que o concorrente mais próximo.

6.3.2 *Publicações em colaboração*

- **The UTrack Framework for Segmenting and Measuring Dermatological Ulcers through Telemedicine.** *CAZZOLATO, M. T., RAMOS, J. S., RODRIGUES, L. S., SCARBORA, L. C., CHINO, D. Y., JORGE, A. E., MARQUES, P. M. A, TRAINA-Jr, C. e TRAINA, A. J. M.. In: Elsevier Journal - Computers in Biology and Medicine (CIBM) DOI: 10.1016/j.compbiomed.2021.104489. [QUALIS-CC A1].*

Neste trabalho foi proposto o UTrack, um *framework* para auxiliar na aquisição de imagens, segmentação e medição de feridas, como também possibilitar o acompanhamento da evolução da cicatrização de úlceras. Como aplicação prática, é proposto o UTrack-App, como um aplicativo móvel Android que processa localmente imagens capturadas por câmeras de dispositivos móveis padrão. O Aplicativo processa as imagens a partir da inserção de marcações manuais do usuário sobre a região de interesse da ferida e do objeto de medição (por exemplo, réguas ou fitas) a fim de segmentar a imagem com o método proposto, o UTrack-Seg. Além disso, o UTrack-app permite que o usuário refine a medição

automática, melhorando a estimativa da área de ferida. O UTrack Framework supera seus concorrentes em tarefas de segmentação de úlceras, como também o UTrack-App realiza todos as etapas de avaliação em imagens de alta resolução em no máximo 30 segundos, sendo adequado para analisar úlceras no apoio em rotinas clínicas e médicas.

- **VD-Tree: How to build an efficient and fit metric access method using Voronoi Diagrams.** MORIYAMA, A., RODRIGUES, L. S., SCABORA, L. C., CAZZOLATO, M. T., TRAINA, A. J. M. e TRAINA-Jr, C.. In: *36th ACM/SIGAPP Symposium on Applied Computing (SAC 2021)*, 2021. *Proceedings of the 36th ACM SAC'2021*. New York: ACM Press, 2021. v. 1. p. 1-4. DOI: 10.1145/3412841.3441915. [QUALIS-CC A2].

Neste trabalho foi proposto o VD-Tree, um método de acesso métrico (MAM) baseado na combinação da estratégia de raio de cobertura com uma organização de nós utilizando diagramas de Voronoi. O foco deste MAM é realizar a indexação de conjuntos de objetos complexos de forma eficiente e reduzindo o problema de sobreposição de nós. Na prática, o método impõe novas heurísticas para atualização de nós, com trocas de elementos entre nós utilizando o diagrama de Voronoi para a divisão de regiões em sobreposição. Os resultados realizados sobre conjuntos de dados complexos do mundo real mostram que a VD-Tree reduz a sobreposição de nós de forma efetiva, como também realiza consultas por similaridade mais rápidas, quando comparada com seus concorrentes.

- **Semi-Automatic Ulcer Segmentation and Wound Area Measurement Supporting Telemedicine.** CAZZOLATO, M. T., RAMOS, J. S., RODRIGUES, L. S., SCABORA, L. C., CHINO, D. Y., JORGE, A. E., MARQUES, P. M. A, TRAINA-Jr, C. e TRAINA, A. J. M.. In *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)* (pp. 356-361). DOI: 10.1109/CBMS49503.2020.00073. [Qualis-CC A3].

Neste trabalho foi proposto o *framework* URule, composto por heurísticas de segmentação e medição de úlceras cutâneas. A motivação é oferecer soluções práticas de apoio ao acompanhamento sobre o estado clínico do paciente para profissionais de saúde e cuidadores. Com isso é apresentado o URule-App, um aplicativo móvel do *framework* que analisa imagens capturadas por câmeras em dispositivos móveis. O foco é oferecer meios práticos para a segmentação de imagens baseada em marcações manuais da região externa da ferida e da ferramenta de medição, realizadas pelo próprio usuário de forma fácil e rápida, resultando na medição (em centímetros ou polegadas) da ferida. Os resultados mostram que o URule segmenta e mede as feridas com precisão e de forma semiautomática quando avaliados em vários cenários.

- **DBMS-based framework for content-based retrieval and analysis of skin ulcer Images in medical practice.** CAZZOLATO, M. T., RODRIGUES, L. S., SCABORA, L. C., ZABOT, G. F., VASCONCELOS, G. Q., CHINO, D. Y., JORGE, A. E., CORDEIRO, R. L.

F., TRAINA-Jr, C. e TRAINA, A. J. M.. In Anais do XXXIV Simpósio Brasileiro de Banco de Dados (pp. 109-120), 2019. SBC. DOI:10.5753/sbbd.2019.8812. [Qualis-CC A4].

Neste trabalho foi proposto um *framework* baseado em um SGBDR para o processamento de imagens de úlceras. O foco é permitir a utilização de dispositivos móveis no gerenciamento de informações e recuperação por conteúdo no apoio ao tratamento de pacientes. O ULEARN-App é uma aplicação *mobile* que permite a coleta de imagens a partir da captura por câmeras comuns, realizando a segmentação e análise de úlceras, armazenando as informações geradas, como também a execução de consultas por similaridade para a recuperação dos casos mais próximos. Os resultados mostram que a ferramenta pode responder análises de úlceras e recuperação por conteúdo de forma rápida e com precisão.

- **G-FranC: a dataset of criminal activities mapped as a complex network in a relational DBMS.** SCABORA, L. de C., SPADON, G., RODRIGUES, L. S., CAZZOLATO, M. T., ARAÚJO, M. V. dos S., SOUSA, E. P. M. de, TRAINA, A. J. M., RODRIGUES-Jr, J. F E TRAINA-Jr, C.. In Anais do XXXIV Simpósio Brasileiro de Banco de Dados, Dataset Showcase Workshop (DSW), 2019. [QUALIS-CC A4].

Neste trabalho foi proposto o G-FranC, uma base de dados mapeada a partir de uma rede complexa das vias urbanas combinada com estatísticas de várias atividades criminais da cidade de São Francisco - CA, EUA, período de 2003 a 2018. O pré-processamento realizado permite que o conjunto de dados seja analisado em vários cenários de aplicações a partir dos dados crime e da topologia da rede, por exemplo o estudo de atividades ilegais por regiões com a aplicação de algoritmos de detecção de comunidades. O G-Franc contribui por combinar dois tipos de dados e de forma estruturada, sendo armazenado no PostgreSQL a partir de *scripts* que gerenciam a coleta e atualização dos dados de origem.

6.4 Trabalhos Futuros

O trabalho desenvolvido possibilita a realização de pesquisa sobre diferentes aspectos em trabalhos futuros. As principais sugestões de tópicos a serem explorados são:

- I - Aplicação do SOLID sobre outros tipos de dados complexos, como também a combinação de dados, por exemplo de imagens e textos, considerando bases de dados advindas de diversas fontes;
- II - Desenvolvimento de uma heurística para selecionar a melhor combinação de FEMs correlacionados, funções de distâncias disponíveis utilizando o método CorDiS para serem aplicados sobre bases de dados complexos;
- III - Desenvolvimento de heurísticas alternativas, mais adequadas ao processo usado no CorDiS, mais adequadas para dados complexos mais específicos, ou que considerem outras formas

de correlação para a geração dos fatores de compatibilidade entre atributos complexos, por exemplo, utilizando métodos baseadas em Aprendizado Profundo (*Deep Learning*);

- IV - Explorar alternativas para a aplicação de outras funções de distâncias na consolidação de distâncias, em comparação com a função Solid-Dist;
- V - Explorar a aplicação do método SOLID combinado à estruturas de indexação de dados complexos.

REFERÊNCIAS

- ABIRI, N.; LINSE, B.; EDÉN, P.; OHLSSON, M. Establishing strong imputation performance of a denoising autoencoder in a wide range of missing data problems. **Neurocomputing**, v. 365, p. 137 – 146, 2019. ISSN 0925-2312. Disponível em: <<https://doi.org/10.1016/j.neucom.2019.07.065>>. Citado nas páginas 51, 54 e 58.
- ALLISON, P. D. **Missing data**. Thousand Oaks: Sage publications, 2001. v. 136. 07-136 p. ISBN 978-0-7619-1672-7. Citado nas páginas 46 e 50.
- ASUERO, A. G.; SAYAGO, A.; GONZÁLEZ, A. G. The correlation coefficient: An overview. **Critical Reviews in Analytical Chemistry**, Taylor & Francis, v. 36, n. 1, p. 41–59, 2006. Citado nas páginas 42 e 43.
- BASTOS, I. L.; ANGELO, M. F.; LOULA, A. C. Recognition of static gestures applied to brazilian sign language (libras). In: IEEE. **2015 28th SIBGRAPI Conference on Graphics, Patterns and Images**. Salvador, Brazil, 2015. p. 305–312. Disponível em: <<https://doi.org/10.1109/SIBGRAPI.2015.26>>. Citado nas páginas 78 e 79.
- BATISTA, G. E. A. P. A.; MONARD, M. C. An analysis of four missing data treatment methods for supervised learning. **Applied Artificial Intelligence**, Taylor & Francis, v. 17, n. 5-6, p. 519–533, 2003. Disponível em: <<https://doi.org/10.1080/713827181>>. Citado nas páginas 28, 50, 51, 53 e 58.
- BELL, M. L.; FIERO, M.; HORTON, N. J.; HSU, C.-H. Handling missing data in rcts; a review of the top medical journals. **BMC Medical Research Methodology**, Springer, v. 14, n. 1, p. 118, 2014. Disponível em: <<https://doi.org/10.1186/1471-2288-14-118>>. Citado na página 27.
- BENESTY, J.; CHEN, J.; HUANG, Y.; COHEN, I. Pearson correlation coefficient. In: **Noise reduction in speech processing**. Springer Berlin Heidelberg, 2009. p. 1–4. ISBN 978-3-642-00295-3. Disponível em: <https://doi.org/10.1007/978-3-642-00296-0_5>. Citado na página 43.
- BOEHM, M. Technical perspective: Declarative recursive computation on an rdbms. **ACM SIGMOD Record**, Association for Computing Machinery, New York, NY, USA, v. 49, n. 1, p. 42, set. 2020. ISSN 0163-5808. Disponível em: <<https://doi.org/10.1145/3422648.3422658>>. Citado na página 32.
- BOZKAYA, T.; OZSOYOGLU, M. Distance-based indexing for high-dimensional metric spaces. In: **ACM SIGMOD Record**. Tucson, Arizona, USA: ACM SIGMOD, 1997. v. 26, n. 2, p. 357–368. Disponível em: <<https://doi.org/10.1145/253260.253345>>. Citado na página 38.
- BRINIS, S. *et al.* Hollow-tree: a metric access method for data with missing values. **Journal of Intelligent Information Systems**, v. 53, n. 3, p. 481–508, 2019. Disponível em: <<https://doi.org/10.1007/s10844-019-00567-8>>. Citado nas páginas 27, 28, 38, 39, 46, 47, 51, 55, 57 e 58.

BURAGO, D.; BURAGO, Y.; IVANOV, S. **A course in metric geometry**. Providence, RI-USA: American Mathematical Society, 2001. v. 33. ISBN 9780821821299. Citado na página 38.

BURGETTE, L. F.; REITER, J. P. Multiple imputation for missing data via sequential regression trees. **American Journal of Epidemiology**, Oxford University Press, v. 172, n. 9, p. 1070–1076, 2010. Disponível em: <<https://doi.org/10.1093/aje/kwq260>>. Citado na página 54.

BURIAN, E.; ROHRMEIER, A.; SCHLAEGER, S.; DIECKMEYER, M.; DIEFENBACH, M. N.; SYVÄRI, J.; KLUPP, E.; WEIDLICH, D.; ZIMMER, C.; RUMMENY, E. J. *et al.* Lumbar muscle and vertebral bodies segmentation of chemical shift encoding-based water-fat mri: the reference database myosegmentum spine. **BMC Musculoskeletal Disorders**, BioMed Central, v. 20, n. 1, p. 1–7, 2019. Disponível em: <<https://doi.org/10.1186/s12891-019-2528-x>>. Citado nas páginas 78 e 79.

BUUREN, S. van. **Flexible Imputation of Missing Data**. Boca Raton, FL, USA: CRC Press LLC, 2018. (Chapman & Hall/CRC Interdisciplinary Statistics). ISBN 9781138588318. Citado nas páginas 28 e 53.

CANAHUATE, G. *et al.* Indexing incomplete databases. In: SPRINGER. **International Conference on Extending Database Technology**. Berlin, Heidelberg, 2006. p. 884–901. Disponível em: <https://doi.org/10.1007/11687238_5>. Citado nas páginas 57 e 58.

CAZZOLATO, M.; RODRIGUES, L.; SCABORA, L.; ZABOTI, G.; VASCONCELOS, G.; CHINO, D.; JORGE, A.; CORDEIRO, R.; TRAINA-JR, C.; TRAINA, A. A dbms-based framework for content-based retrieval and analysis of skin ulcer images in medical practice. In: **Anais Principais do XXXIV Simpósio Brasileiro de Banco de Dados**. Porto Alegre, RS, Brasil: SBC, 2019. p. 109–120. Disponível em: <<https://doi.org/10.5753/sbbd.2019.8812>>. Citado na página 26.

CAZZOLATO, M. T.; SCABORA, L. C.; NESSO, M. R.; MILANO-OLIVEIRA, L. F.; COSTA, A. F.; KASTER, D. S.; KOENIGKAM-SANTOS, M.; AZEVEDO-MARQUES, P. M. de; TRAINA-JR, C.; TRAINA, A. J. dp-breath: Heat maps and probabilistic classification assisting the analysis of abnormal lung regions. **Computer Methods and Programs in Biomedicine**, v. 173, p. 27 – 34, 2019. ISSN 0169-2607. Disponível em: <<https://doi.org/10.1016/j.cmpb.2019.01.014>>. Citado na página 26.

CHINO, D. Y.; SCABORA, L. C.; CAZZOLATO, M. T.; JORGE, A. E.; TRAINA-JR., C.; TRAINA, A. J. Segmenting skin ulcers and measuring the wound area using deep convolutional networks. **Computer Methods and Programs in Biomedicine**, v. 191, p. 105376, 2020. ISSN 0169-2607. Disponível em: <<https://doi.org/10.1016/j.cmpb.2020.105376>>. Citado na página 26.

CHINO, D. Y. T.; SCABORA, L. C.; CAZZOLATO, M. T.; JORGE, A. E. S.; CAETANO, T.-J.; TRAINA, A. J. M. Icarus: Retrieving skin ulcer images through bag-of-signatures. In: **2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)**. Los Alamitos: IEEE, 2018. v. 1, p. 82–87. ISSN 2372-9198. Disponível em: <<https://doi.org/10.1109/CBMS.2018.00022>>. Citado na página 33.

CIACCIA, P.; PATELLA, M.; ZEZULA, P. M-tree: An efficient access method for similarity search in metric spaces. In: JARKE, M.; CAREY, M. J.; DITTRICH, K. R.; LOCHOVSKY, F. H.; LOUCOPOULOS, P.; JEUSFELD, M. A. (Ed.). **VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases**. Athens, Greece: Morgan Kaufmann, 1997. p. 426–435. Disponível em: <<http://www.vldb.org/conf/1997/P426.PDF>>. Citado na página 39.

- CISMONDI, F.; FIALHO, A. S.; VIEIRA, S. M.; RETI, S. R.; SOUSA, J. M. da C.; FINKELSTEIN, S. N. Missing data in medical databases: Impute, delete or classify? **Artificial Intelligence in Medicine**, Elsevier, v. 58, n. 1, p. 63–72, 2013. Disponível em: <<https://doi.org/10.1016/j.artmed.2013.01.003>>. Citado nas páginas 28, 46, 47, 49, 51 e 55.
- CODD, E. F. A relational model of data for large shared data banks. In: BRODY, M.; DENERT, E. (Ed.). **Software Pioneers**. Springer Berlin Heidelberg, 2002. p. 263–294. Disponível em: <https://doi.org/10.1007/978-3-642-59412-0_16>. Citado nas páginas 25 e 32.
- CROXTON, F. E.; COWDEN, D. J. Applied general statistics. Prentice-Hall, Inc, 1939. Disponível em: <<https://doi.org/10.1037/13608-000>>. Citado na página 42.
- DALCA, A. V.; BOUMAN, K. L.; FREEMAN, W. T.; ROST, N. S.; SABUNCU, M. R.; GOLLAND, P. Medical image imputation from image collections. **IEEE Transactions on Medical Imaging**, v. 38, n. 2, p. 504–514, 2019. Disponível em: <<https://doi.org/10.1109/TMI.2018.2866692>>. Citado nas páginas 27, 46 e 53.
- DATE, C. J. **SQL and Relational Theory - How to Write Accurate SQL Code, Second Edition**. O'Reilly, 2012. (Theory in practice). ISBN 978-1-449-31640-2. Disponível em: <<http://www.oreilly.de/catalog/9781449316402/index.html>>. Citado nas páginas 25, 32 e 46.
- DEZA, M. M.; DEZA, E. **Encyclopedia of Distances**. 4th. ed. Heidelberg: Springer, 2016. ISBN 978-3-662-52843-3. Disponível em: <<https://doi.org/10.1007/978-3-662-52844-0>>. Citado nas páginas 35, 36 e 37.
- DIXON, J. K. Pattern recognition with partly missing data. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, n. 10, p. 617–621, 1979. Disponível em: <<https://doi.org/10.1109/TSMC.1979.4310090>>. Citado na página 50.
- GARCIARENA, U.; SANTANA, R. An extensive analysis of the interaction between missing data types, imputation methods, and supervised classifiers. **Expert Systems with Applications**, v. 89, p. 52 – 65, 2017. ISSN 0957-4174. Disponível em: <<https://doi.org/10.1016/j.eswa.2017.07.026>>. Citado na página 53.
- HAJDER, S. **Letters organized by typefaces - Standard Windows fonts with each letters organized in classes by typeface**. 2020. Disponível em: <<https://www.kaggle.com/killen/bw-font-typefaces>>. Citado na página 78.
- HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques, 3rd edition**. Morgan Kaufmann, 2011. ISBN 978-0123814791. Disponível em: <<http://hanj.cs.illinois.edu/bk3/>>. Citado na página 72.
- HUGHES, J. F.; DAM, A. van; MCGUIRE, M.; SKLAR, D. F.; FOLEY, J. D.; FEINER, S. K.; AKELEY, K. **Computer Graphics - Principles and Practice, 3rd Edition**. Upper Saddle River, NJ, USA: Addison-Wesley, 2014. ISBN 978-0-321-39952-6. Citado na página 34.
- IBRAHIM, J. G.; CHU, H.; CHEN, M.-H. Missing data in clinical studies: issues and methods. **Journal of Clinical Oncology**, American Society of Clinical Oncology, v. 30, n. 26, p. 3297, 2012. Disponível em: <<https://doi.org/10.1200/JCO.2011.38.7589>>. Citado na página 48.
- KOSUB, S. A note on the triangle inequality for the jaccard distance. **Pattern Recognition Letters**, v. 120, p. 36–38, 2019. ISSN 0167-8655. Citado na página 36.

LITTLE, R. J.; RUBIN, D. B. **Statistical analysis with missing data**. 3th. ed. Hoboken, NJ, USA: John Wiley & Sons, 2019. v. 793. ISBN 9781118595695. Citado nas páginas 27, 46, 47, 50, 51, 52, 53 e 58.

MCNEISH, D. Missing data methods for arbitrary missingness with small samples. **Journal of Applied Statistics**, Taylor & Francis, v. 44, n. 1, p. 24–39, 2017. Disponível em: <<https://doi.org/10.1080/02664763.2016.1158246>>. Citado na página 27.

MUKAKA, M. M. A guide to appropriate use of correlation coefficient in medical research. **Malawi medical journal**, v. 24, n. 3, p. 69–71, 2012. Citado na página 86.

NENE, S. K. N. S. A.; MURASE, H. **Columbia Object Image Library (COIL-100)**. New York, USA, 1996. Citado nas páginas 78 e 79.

NESSO, M. R.; CAZZOLATO, M. T.; SCABORA, L. C.; OLIVEIRA, P. H.; SPADON, G.; SOUZA, J. A. de; OLIVEIRA, W. D.; CHINO, D. Y. T.; RODRIGUES, J. F.; TRAINA, A. J. M.; CAETANO, T.-J. Rafiki: Retrieval-based application for imaging and knowledge investigation. In: **2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)**. Karlstad: IEEE, 2018. p. 71–76. ISSN 2372-9198. Disponível em: <<https://doi.org/10.1109/CBMS.2018.00020>>. Citado na página 33.

NIKFALAZAR, S.; YEH, C.; BEDINGFIELD, S. E.; KHORSHIDI, H. A. Missing data imputation using decision trees and fuzzy clustering with iterative learning. **Knowledge and Information Systems**, Springer, v. 62, n. 6, p. 2419–2437, 2020. Disponível em: <<https://doi.org/10.1007/s10115-019-01427-1>>. Citado nas páginas 51, 54 e 58.

OOI, B. C.; GOH, C. H.; TAN, K.-L. Fast high-dimensional data search in incomplete databases. In: **Proceedings of the 24rd International Conference on Very Large Data Bases**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998. (VLDB '98), p. 357–367. ISBN 1558605665. Disponível em: <<http://www.vldb.org/conf/1998/p357.pdf>>. Citado nas páginas 28, 51, 56, 57 e 58.

PEARSON, R. K. The problem of disguised missing data. **ACM SIGKDD Explorations Newsletter**, ACM New York, NY, USA, v. 8, n. 1, p. 83–92, 2006. Disponível em: <<https://doi.org/10.1145/1147234.1147247>>. Citado na página 47.

PEREIRA, C. R.; WEBER, S. A. T.; HOOK, C.; ROSA, G. H.; PAPA, J. P. Deep learning-aided parkinson's disease diagnosis from handwritten dynamics. In: **29th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2016**. Sao Paulo, Brazil: IEEE Computer Society, 2016. p. 340–346. Disponível em: <<https://doi.org/10.1109/SIBGRAPI.2016.054>>. Citado nas páginas 78 e 79.

RAJA, P. S.; THANGAVEL, K. Missing value imputation using unsupervised machine learning techniques. **Soft Computing**, Springer, v. 24, n. 6, p. 4361–4392, 2020. Disponível em: <<https://doi.org/10.1007/s00500-019-04199-6>>. Citado nas páginas 46, 50, 51 e 53.

RODGERS, J. L.; NICEWANDER, W. A. Thirteen ways to look at the correlation coefficient. **The American Statistician**, Taylor & Francis, v. 42, n. 1, p. 59–66, 1988. Disponível em: <<https://doi.org/10.1080/00031305.1988.10475524>>. Citado na página 42.

RODRIGUES, L. S.; CAZZOLATO, M. T.; TRAINA, A. J. M.; TRAINA-JR., C. Taking advantage of highly-correlated attributes in similarity queries with missing values. In: **Similarity**

Search and Applications - 13th International Conference, SISAP 2020. Copenhagen, Denmark: Springer, 2020. (Lecture Notes in Computer Science, v. 12440), p. 168–176. Disponível em: <https://doi.org/10.1007/978-3-030-60936-8_13>. Citado nas páginas 26, 27, 28 e 58.

SALEMBIER, P.; SIKORA, T.; MANJUNATH, B. **Introduction to MPEG-7: Multimedia Content Description Interface.** USA: John Wiley & Sons, 2002. ISBN 0471486787. Citado na página 35.

SASAK-OKON, A.; TUDRUJ, M. Speculative query execution in RDBMS based on analysis of query stream multigraphs. In: **IDEAS 2020: 24th International Database Engineering & Applications Symposium.** Seoul, Republic of Korea: ACM, 2020. p. 23:1–23:10. Disponível em: <<https://doi.org/10.1145/3410566.3410604>>. Citado na página 32.

SCHOBER, P.; BOER, C.; SCHWARTE, L. A. Correlation coefficients: appropriate use and interpretation. **Anesthesia & Analgesia**, Wolters Kluwer, v. 126, n. 5, p. 1763–1768, 2018. Disponível em: <<https://doi.org/10.1213/ANE.0000000000002864>>. Citado na página 86.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Database System Concepts, Seventh Edition.** McGraw-Hill Book Company, 2020. ISBN 9780078022159. Disponível em: <<https://www.db-book.com/db7/index.html>>. Citado nas páginas 25, 32 e 46.

SKOPAL, T.; POKORNÝ, J.; SNÁSEL, V. PM-tree: Pivoting metric tree for similarity search in multimedia databases. In: CITESEER. **Advances in Databases and Information Systems, 8th East European Conference, ADBIS 2004.** Budapest, Hungary: Local Proceedings, 2004. Disponível em: <<http://www.sztaki.hu/conferences/ADBIS/9-Skopal.pdf>>. Citado na página 39.

TRAINA, A. J. M.; BRINIS, S.; PEDROSA, G. V.; AVALHAIS, L. P. S.; TRAINA-JR., C. Querying on large and complex databases by content: Challenges on variety and veracity regarding real applications. **Information Systems**, v. 86, p. 10–27, 2019. Disponível em: <<https://doi.org/10.1016/j.is.2019.03.012>>. Citado nas páginas 27, 28, 33, 46, 49 e 55.

TRAINA-JR., C.; MORIYAMA, A.; ROCHA, G. M. da; CORDEIRO, R. L. F.; CIFERRI, C. D. de A.; TRAINA, A. J. M. The SimilarQL framework: similarity queries in plain SQL. In: **Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019.** ACM, 2019. p. 468–471. Disponível em: <<https://doi.org/10.1145/3297280.3299736>>. Citado nas páginas 25, 26, 38 e 42.

TRAINA-JR, C.; TRAINA, A.; FALOUTSOS, C.; SEEGER, B. Fast indexing and visualization of metric data sets using Slim-Trees. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 14, n. 2, p. 244–260, 2002. Disponível em: <<https://doi.org/10.1109/69.991715>>. Citado nas páginas 38 e 39.

UHLMANN, J. K. Satisfying general proximity/similarity queries with metric trees. **Information Processing Letters**, Elsevier, v. 40, n. 4, p. 175–179, 1991. Disponível em: <[https://doi.org/10.1016/0020-0190\(91\)90074-R](https://doi.org/10.1016/0020-0190(91)90074-R)>. Citado na página 38.

UNAR, S.; WANG, X.; ZHANG, C.; WANG, C. Detected text-based image retrieval approach for textual images. **IET Image Processing**, v. 13, n. 3, p. 515–521, 2019. Disponível em: <<https://doi.org/10.1049/iet-ipr.2018.5277>>. Citado na página 33.

VASCONCELOS, G. Q.; ZABOT, G. F.; LIMA, D. M. de; RODRIGUES-JR., J. F.; TRAINA-JR., C.; KASTER, D. S.; CORDEIRO, R. L. F. TendeR-Sims - similarity retrieval system for public tenders. In: **Proceedings of the 20th International Conference on Enterprise Information Systems, ICEIS 2018**. Funchal, Madeira, Portugal: SciTePress, 2018. Volume 1, p. 143–150. Disponível em: <<https://doi.org/10.5220/0006697601430150>>. Citado na página 26.

VENUGOPALAN, J.; CHANANI, N.; MAHER, K. O.; WANG, M. D. Novel data imputation for multiple types of missing data in intensive care units. **IEEE Journal of Biomedical and Health Informatics**, v. 23, n. 3, p. 1243–1250, 2019. Disponível em: <<https://doi.org/10.1109/JBHI.2018.2883606>>. Citado nas páginas 27, 51 e 52.

VYAWAHARE, H. R.; KARDE, P. P.; THAKARE, V. M. A hybrid database approach using graph and relational database. In: **Proceedings of the 2018 IEEE International Conference on Research in Intelligent and Computing in Engineering (RICE 2018)**. San Salvador, El Salvador: IEEE, 2018. p. 1–4. Disponível em: <<https://doi.org/10.1109/RICE.2018.8509057>>. Citado na página 32.

YANG, L.; LUO, P.; LOY, C. C.; TANG, X. A large-scale car dataset for fine-grained categorization and verification. In: **IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015**. Boston, MA, USA: IEEE Computer Society, 2015. p. 3973–3981. Disponível em: <<https://doi.org/10.1109/CVPR.2015.7299023>>. Citado nas páginas 78 e 79.

YIANILOS, P. N. Data structures and algorithms for nearest neighbor search in general metric spaces. In: RAMACHANDRAN, V. (Ed.). **Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms**. Austin, Texas, USA: ACM/SIAM, 1993. p. 311–321. Disponível em: <<http://dl.acm.org/citation.cfm?id=313559.313789>>. Citado na página 38.

YOON, J. *et al.* GAIN: missing data imputation using generative adversarial nets. In: **Proceedings of the 35th International Conference on Machine Learning, ICML 2018**. Stockholmsmässan, Stockholm, Sweden: PMLR, 2018. (Proceedings of Machine Learning Research, v. 80), p. 5675–5684. Disponível em: <<http://proceedings.mlr.press/v80/yoon18a.html>>. Citado nas páginas 51, 54 e 58.

YUJIAN, L.; BO, L. A normalized levenshtein distance metric. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 29, n. 6, p. 1091–1095, 2007. Citado na página 37.

ZABOT, G. F.; CAZZOLATO, M. T.; SCABORA, L. C.; TRAINA, A. J. M.; TRAINA-JR., C. Efficient indexing of multiple metric spaces with Spectra. In: **IEEE International Symposium on Multimedia, ISM 2019**. San Diego, CA, USA: IEEE, 2019. p. 169–176. Disponível em: <<https://doi.org/10.1109/ISM46123.2019.00038>>. Citado na página 26.

ZEZULA, P. Similarity searching for database applications. In: **Advances in Databases and Information Systems - 20th East European Conference, ADBIS 2016**. Prague, Czech Republic: Springer, 2016. (Lecture Notes in Computer Science, v. 9809), p. 3–10. Disponível em: <https://doi.org/10.1007/978-3-319-44039-2_1>. Citado na página 36.

ZEZULA, P.; BATKO, M.; DOHNAL, V. Indexing metric spaces. In: **Encyclopedia of Database Systems, Second Edition**. Springer, 2018. Disponível em: <https://doi.org/10.1007/978-1-4614-8265-9_199>. Citado na página 26.

ZHANG, L.; LU, W.; LIU, X.; PEDRYCZ, W.; ZHONG, C. Fuzzy c-means clustering of incomplete data based on probabilistic information granules of missing values. **Knowledge-Based Systems**, Elsevier, v. 99, p. 51–70, 2016. Disponível em: <<https://doi.org/10.1016/j.knosys.2016.01.048>>. Citado nas páginas 53 e 58.

ZHANG, Z. Missing data imputation: focusing on single imputation. **Annals of Translational Medicine**, AME Publications, v. 4, n. 1, 2016. Disponível em: <<https://doi.org/10.3978/j.issn.2305-5839.2015.12.38>>. Citado na página 53.