CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

UM MÉTODO BASEADO EM PARTICIONAMENTO HORIZONTAL PARA MINERAÇÃO DE PADRÕES FREQUENTES EM TABELAS DE HORÁRIOS

Claudio Marcio da Silva Teixeira

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Produção e Sistemas, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de mestre.

Orientadores:
Eduardo Soares Ogasawara

Co-orientadores:
Glauco Fiorott Amorim

Rio de Janeiro,

Março de 2021

Um método baseado em particionamento horizontal para mineração de padrões frequentes em tabelas de horários

Dissertação de Mestrado em Engenharia de Produção e Sistemas, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Claudio Marcio da Silva Teixeira

Aprovada por:

Presidente, Prof. Eduardo Soares Ogasawara, D.Sc. (orientador)

Prof. Glauco Fiorott Amorim, D.Sc. (coorientador)

DIEGO MOREIRA DE ARAUJO CARVALHO:02066727725
2021.03.08 08:55:58 -03'00'

Prof. Diego Moreira de Araujo Carvalho, D.Sc.

Prof. Eduardo Bezerra da Silva, D.Sc.

Prof[a]. Marta Lima de Queiros Mattoso, D.Sc.
(COPPE/UFRJ - Universidade Federal do Rio de Janeiro)

Rio de Janeiro,
Março de 2021

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

T266 Teixeira, Claudio Marcio da Silva
Um método baseado em particionamento horizontal para mineração de padrões frequentes em tabelas de horários / Claudio Marcio da Silva Teixeira — 2021.
55f : il. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca , 2021.
Bibliografia : f. 52-55
Orientador: Eduardo Soares Ogasawara
Coorientador: Glauco Fiorott Amorim

1. Transporte aéreo. 2. Transporte – Tabela de horário. 3. Mineração de dados (Computação). 4. Aeronáutica – Voos. 5. Aeronáutica comercial. I. Ogasawara, Eduardo Soares (Orient.). II. Amorim, Glauco Fiorott (Coorient.). III. Título.

CDD 387.7

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

DEDICATÓRIA

"Quero agradecer a Deus, ao Mestre Jesus, aos Mentores, Guias, Amigos e Familiares espirituais, que sempre me deram força, me apoiando principalmente nos momentos de maior dificuldade, lembrando-me de viver no Amor, na Paz e no Conhecimento. Quero agradecer também às minhas filhas Gabriela e Maitê, que iluminam minha vida de forma muito especial e me dão motivos para continuar sempre buscando dar o melhor de mim e agradeço ainda, aos meus pais Joaquim Teixeira (*In memoriam*) e Gessy da Silva Teixeira, a quem eu agradeço minha existência e a forma como eles me ensinaram a ver a vida."

AGRADECIMENTOS

Em função desta pesquisa tive a oportunidade de conhecer muitas pessoas interessantes com quem além de me oferecer ajuda tive o prazer de compartilhar também de seus saberes e, em alguns casos, também da amizade. A essas pessoas ofereço minha gratidão, pois este trabalho é também fruto das nossas trocas e por isso, também é obra de todos nós.

Ao meu orientador Eduardo Ogasawara, meu coorientador Glauco Amorim, e demais professores, pelos dedicados ensinamentos, orientações e amizade, que me transmitiu incansáveis informações.

Por fim, agradeço aos meus amigos, que direta ou indiretamente estiveram presentes na minha vida, me incentivando e ajudando em mais esta etapa.

RESUMO

Um método baseado em particionamento horizontal para mineração de padrões frequentes em tabelas de horários

Claudio Marcio da Silva Teixeira

Orientadores:
Eduardo Soares Ogasawara

Co-orientadores:
Glauco Fiorott Amorim

Resumo da Dissertação submetida ao Programa de Pós-graduação em Engenharia de Produção e Sistemas do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do título de mestre.

Tabelas de horários associam o inicio e o término de atividades. A essas atividades estão associados diversos atributos que podem ter conhecimento quanto a alocação de horários da tabela. As tabelas de horários que representam origem e destino ou tempo de duração de determinadas atividades podem ser melhoradas a partir de regras de associação (RAs) entre os atributos da tabela. Isso posto, analisar tabelas de horários é uma tarefa importante, pois traz a oportunidade de descobrir informações valiosas que aprimoram processos de gestão, previsão da rede de transporte, dentre outros exemplos. Enquanto isso, a mineração de padrões frequentes é uma técnica usada para apoiar esse tipo de descoberta. Entretanto, existem propriedades intrínsecas presentes nas tabelas de horários. São as dependências funcionais entre o atributo que caracteriza primariamente a atividade planejada com os seus demais atributos. Tais dependências funcionais comprometem a busca por padrões nas tabelas de horários tanto no número de RAs geradas, quanto no custo computacional. Várias dessas RAs referem-se às mesmas informações. As técnicas de remoção de redundância podem reduzir o número de RAs. No entanto, essas técnicas são projetadas para serem usadas somente após o término da mineração, o que aumenta o custo computacional de encontrar RAs úteis. Este trabalho apresenta o *Timetable Pattern Mining* (T-mine), um novo método para mineração de padrões frequentes que melhora a descoberta de conhecimento em tabelas de horários. Avaliamos o T-mine usando dados de voos brasileiros e comparamos o T-mine com a aplicação direta de abordagens de mineração de padrões frequentes com e sem dependências funcionais. Nossos experimentos indicam que o T-mine apresenta um menor custo computacional (menor tempo de execução e menor alocação de memória) do que outras abordagens.

Palavras-chave: mineração de padrões frequentes; regras de associação; redundância; dependência funcional; particionamento horizontal; tabela de horários;

Rio de Janeiro,

Março de 2021

ABSTRACT

Um método baseado em particionamento horizontal para mineração de padrões frequentes em tabelas de horários

Claudio Marcio da Silva Teixeira

Advisors:
Eduardo Soares Ogasawara

Co-orientadores:
Glauco Fiorott Amorim

Abstract of dissertation submitted to Programa de Pós-graduação em Engenharia de Produção e Sistemas - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

Timetables associate the beginning and the end of activities. These activities are associated with several attributes that may be aware of an allocation of times in the table. The timetables that represent origin and destination or duration of activities can be improved from association rules (ARs) between the attributes of the table. That said, analyzing the timetables is an important task, as it brings the opportunity to discover valuable information that improves management processes, forecasting the transport network, among other examples. Meanwhile, frequent pattern mining is a technique used to support such type of discovery. However, functional dependencies of the attributes derived from the origin-destination matrix are intrinsic properties present in timetables. Such functional dependencies compromise the search for patterns in timetables in both the number of association rules (ARs) generated and the computational cost. Several of these ARs refer to the same information. Redundancy removal techniques can reduce the number of ARs. However, these techniques are designed to be used only after mining finishes, which increases the computational cost of finding useful ARs. This work presents the Timetable Pattern Mining (T-mine), a novel method for frequent pattern mining that improves knowledge discovery in timetables. We evaluated T-mine using Brazilian Flight Data and compared T-mine with the direct application of frequent pattern mining approaches with and without functional dependencies. Our experiments indicate that T-mine has a lower computational cost (less execution time and less memory allocation) than other approaches.

Key-words: frequent pattern mining; association rules; redundancy; functional dependency; horizontal partitioning; timetable;

Rio de Janeiro,

Março de 2021

Sumário

I Introdução **1**

II Referencial Teórico **8**

II.1 MPF 8

II.2 Apriori 9

II.3 Remoção de Redundância 12

II.4 Particionamento Horizontal 14

II.5 Análise estatística das RAs 15

III Trabalhos Relacionados **17**

III.1 Análise de tabelas de horários de transporte 18

III.2 MPF em tabelas de horários 19

III.3 MPF e DF 21

III.4 MPF e análise de redundância 23

III.5 Resumo 24

IV Metodologia **26**

IV.1 T-mine 26

IV.2 *Brazilian Flights Dataset* 31

V Experimentos e Resultados **34**

V.1 Configuração 35

V.2 Experimentos computacionais 36

V.3 Análise das RAs produzidas 40

V.4 Análise de sensibilidade 41

V.5 RAs não redundantes: Comparação com base na frequência 41

VI Conclusões **49**

VI.1 Contribuições e Considerações finais 49

VI.2 Trabalhos Futuros 50

Referências Bibliográficas 52

Lista de Figuras

I.1 Relação entre Matriz OD e tabela de horários. 1

I.2 Faixas de números de voos das companhias aéreas do Brasil 3

I.3 Exemplo motivador 6

II.1 Pseudo-código para geração de candidatos 10

II.2 Pseudo-código para poda de candidatos 11

II.3 Pseudo-código do algoritmo *Apriori* 12

III.1 Caracterização dos trabalhos relacionados 19

III.2 Quadro comparativo dos trabalhos relacionados 25

IV.1 Processo de mineração usando o *Timetable Pattern Mining* (T-mine) 27

IV.2 Bloco de dados para o predicado origem=SBRJ 28

IV.3 Bloco de dados para o predicado origem=SBSP 29

IV.4 Bloco de dados para o predicado origem=SBSV 29

IV.5 Bloco de dados para o predicado destino=SBBR 29

IV.6 Bloco de dados para o predicado destino=SBGO 29

IV.7 Bloco de dados para o predicado destino=SBRJ 29

IV.8 Bloco de dados para o predicado destino=SBSP 29

IV.9 Diagrama de Classe do *Brazilian Flights Dataset* (BFD) 32

IV.10 Metadados do BFD 33

V.1 Comparação entre os tempos de execução 37

V.2 Estatística das abordagens 38

V.3 Consumo de memória 39

V.4 Análise de sensibilidade variando-se suporte 42

V.5 Análise de sensibilidade variando-se confiança 42

V.6 Comparação das curvas de desvio quadrático sobre a frequência 44

V.7 Comparação entre as curvas de frequência das abordagens 46

Lista de Tabelas

I.1 Tabela de horários de voos 3

II.1 Conjuntos candidatos com 1 item 9

IV.1 Tabela de horários de voos 28

V.1 Parâmetros de entrada 36

V.2 Quantidade de regras distintas e não redundantes por ano para cada uma das abordagens 38

V.3 Comparação das RAs geradas 41

V.4 Extrato das RAs geradas 41

V.5 Probabilidade de predição das regras perenes 45

V.6 RAs distintas e não redundantes 46

V.7 Extrato das regras perenes geradas por meio da abordagem baseline 48

V.8 As 11 RAs perenes a mais geradas por meio do T-mine comparado com baseline 48

V.9 Extrato das RAs perenes geradas por meio da abordagem *without functional dependency* (wfd) 48

Lista de Abreviações

ANAC	Agência Nacional De Aviação Civil	2, 31
APIS	*Application Program Interfaces*	18
ASOS	*Automated Surface Observing Systems*	31, 32
BFD	*Brazilian Flights Dataset*	31, 32, 33, 34, 36, 49
DAL	*Data Analytics Lab*	32, 35
DF	Dependência Funcional	4, 5, 7, 8, 12, 13, 17, 19, 21, 22, 24, 25, 26, 31, 34, 38, 40, 43, 45, 46, 47, 49, 50
DFS	Dependências Funcionais	6, 8, 21, 22, 25, 51
DTA	*Diagnostic Test Approach*	22
EDA	*Exploratory Data Analysis*	32
ETL	*Extract, Transform, and Load*	31, 32
FCA	*Formal Context Analysis*	21, 22
IATA	*International Air Transport Association*	2, 3
ICAO	*International Civil Aviation Organization*	2
IQR	*Interquartil Range*	38
MPF	Mineração De Padrões Frequentes	1, 2, 4, 6, 7, 8, 9, 12, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 34, 36, 41, 43, 49, 50
OD	Origem,Destino	1, 2, 4, 5, 25, 26, 28, 40, 45, 46, 47, 50, 51
PES	Padrões Emergentes	51
RA	Regra De Associação	4, 8, 30
RAM	*Random Access Memory*	35, 38
RAS	Regras De Associação	1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 15, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 28, 30, 31, 34, 35, 40, 41, 43, 44, 45, 46, 47, 49, 50, 51
RHS	*Right Hand Side*	24
T-MINE	*Timetable Pattern Mining*	5, 6, 8, 13, 18, 19, 20, 21, 22, 25, 26, 27, 28, 30, 31, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 49, 50, 51
TLC	Teorema Do Limite Central	15, 43, 44
VRA	Voos Regulares Ativos	31
WFD	*Without functional dependency*	34, 35, 36, 37, 38, 39, 40, 41, 42, 46, 47, 48

Capítulo I Introdução

Uma tabela de horários (*Timetable*) pode ser descrita como um plano de horários em que eventos específicos devem ocorrer [Ceder, 2016]. Ela desempenha um papel fundamental no planejamento dos sistemas de transportes e integração das operações na área de transporte, pois define como os vários recursos são integrados e gerenciados, incluindo as tripulações, o material circulante e as instalações de infraestrutura [Lee et al., 2016]. Em geral, as tabelas de horários são concebidas para atender, dentre outras demandas, as de transporte público. Uma tabela de horários de transporte é normalmente composta por pares Origem,Destino (OD) e informações de horários de partida e chegada dos meios de transporte específicos que os compõem, conforme pode ser visto na Figura I.1.

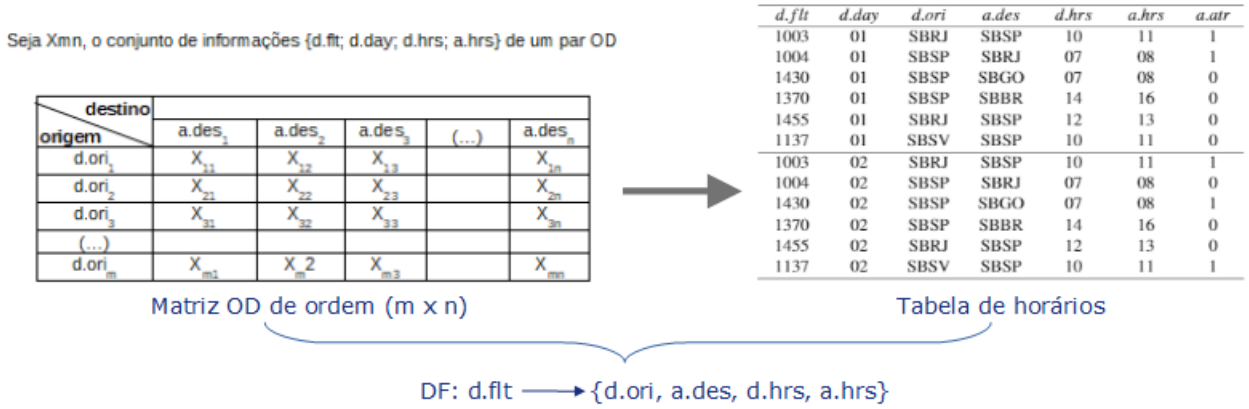


Figura I.1: Relação entre Matriz OD e tabela de horários.

Os planejadores de transporte analisam tabelas de horários para descobrir padrões de viagem e, a partir desses padrões, otimizarem os recursos disponíveis [Ceha and Ohta, 1997]. Neste cenário, a Mineração de Padrões Frequentes (MPF) é uma técnica poderosa para apoiar a descoberta de padrões em sistemas de transportes, pois esses sistemas possuem bases com grandes conjuntos de dados [Aggarwal and Han, 2014]. A descoberta de padrões que acontecem com uma certa frequência em uma base de dados possibilita uma análise diferenciada sobre o inter-relacionamento entre os dados. Os padrões são obtidos por meio das Regras de Associação (RAs), geradas durante o processo de mineração. Esses dados podem ser divididos em conjuntos distintos tendo como base alguma característica em comum ou algum atributo que queira ser melhor analisado.

As RAs destacam conjuntos frequentes de itens no antecedente, levando a conjuntos frequentes

de itens no consequente, formalmente definido por *antecedente* → *consequente* [Aggarwal and Han, 2014]. As RAs são geradas por meio da especificação de valores de interesse pelo usuário, e que caracterizam a importância daquela regra. Valores de interesse comumente utilizados em MPF são: suporte e confiança. O suporte é a frequência ou contagem de ocorrências do antecedente concomitantemente com o consequente, na base de dados. Já a confiança é a probabilidade de que, se antecedente ocorrer, então, o consequente também ocorrerá.

Sendo o dado, um fato ou valor documentado ou um resultado de uma medição, pode-se dizer que, quando um sentido semântico ou um significado é atribuído aos dados, gera-se informação. E quando esses significados se tornam conhecidos, ou seja, quando um agente os aprende, este se torna consciente e capaz de tomar decisões a partir deles, surge o conhecimento. Isso posto, apesar da MPF ser uma ferramenta muito importante para a geração de conhecimentos sobre os dados, a utilização direta da ferramenta pode levar a alguns problemas. Dependendo do tamanho da base de dados analisado, a MPF pode produzir um grande número de RAs, o que deixaria o processo com elevado custo computacional.

Considere uma amostra de uma tabela de horários que mapeia voos na rede de transporte aéreo brasileira apresentada na Tabela I.1. As colunas $d.flt$, $d.day$, $d.ori$, $a.des$, $d.hrs$, $a.hrs$ e $a.atr$ representam respectivamente: o número do voo , a data de partida, o aeroporto de partida, o aeroporto de chegada, a hora de partida, a hora de chegada e um atributo que indica se ocorreu ou não um atraso. Os valores SBSP, SBGO, SBSV, SBRJ e SBBR, indicam os aeroportos de São Paulo (Congonhas), Goiânia, Salvador, Rio de Janeiro (Santos Dumont) e Brasília, respectivamente. Os aeroportos são definidos pela *International Civil Aviation Organization* (ICAO) ou Organização da Aviação Civil Internacional.

Segundo a *International Air Transport Association* (IATA) ou Associação Internacional de Transporte Aéreo, um número de voo é um código específico que uma companhia aérea atribui a um determinado voo em sua rede. Além de ser usado para companhias aéreas, o número ajuda o Controle de Tráfego Aéreo a organizar e rastrear voos aéreos. Existem algumas regras padronizadas quando se trata de numerar voos. Para voos operando ao mesmo tempo, os números não podem ser repetidos. Além disso, os números não devem exceder quatro dígitos. Com algumas exceções, os voos geralmente são numerados com base na direção da viagem. Por exemplo, os voos para o norte e leste recebem números pares, enquanto os voos para o sul e oeste são numerados ímpares. É também comum que as rotas (OD) permaneçam os mesmos durante anos a fio, sendo modificados somente em caso de incidentes ou desastres aéreos.

Aqui no Brasil, de acordo com a Agência Nacional de Aviação Civil (ANAC), os números dos voos nacionais devem ser escolhidos a partir de uma faixa que vai de 1000 a 6999. Já os internacionais, de 7000 a 8999. Para facilitar, aqui no Brasil, cada companhia aérea recebe um

determinado intervalo entre esses números e é a partir dele que deve determinar as combinações, e isto pode ser observado na Figura I.2.

Empresa	Números - voos nacionais	Números - voos internacionais
Avianca	6000 a 6399	8500 a 8599
Azul	2400 – 2999 , 4000 - 4499, 4900 - 5299, 5700 - 5799, 6900 - 6999	8700 - 8799
Gol	1000 – 2199, 4800 – 4899, 5300-5499, 5600 – 5699, 5800 - 5899, 6600 - 6899,	7400 – 7799
Latam (TAM)	3000 – 3999, 4600 - 4799	8000 - 8299

Figura I.2: Faixas de números de voos das companhias aéreas do Brasil. Fonte: ANAC

À esquerda de um número de voo está um código de dois caracteres que identifica a companhia aérea. Todas as companhias aéreas comerciais possuem um código identificador exclusivo atribuído a elas pela IATA. Para a maioria das companhias aéreas, o código funciona como uma abreviatura do nome completo da empresa. Por exemplo, a *American Airlines* usa "AA" e a *United Airlines* usa "UA". Dado todo esse conhecimento prévio de domínio em relação a identificação primária de um voo, é que pode-se afirmar que o número do voo é o atributo que identifica primariamente um voo, e é por isso que, pode-se afirmar que a dependência funcional é caracterizada pelo atributo número do voo.

Tabela I.1: Tabela de horários de voos

$d.flt$	$d.day$	$d.ori$	$a.des$	$d.hrs$	$a.hrs$	$a.atr$
1003	01	SBRJ	SBSP	10	11	1
1004	01	SBSP	SBRJ	07	08	1
1430	01	SBSP	SBGO	07	08	0
1370	01	SBSP	SBBR	14	16	0
1455	01	SBRJ	SBSP	12	13	0
1137	01	SBSV	SBSP	10	11	0
1003	02	SBRJ	SBSP	10	11	1
1004	02	SBSP	SBRJ	07	08	0
1430	02	SBSP	SBGO	07	08	1
1370	02	SBSP	SBBR	14	16	0
1455	02	SBRJ	SBSP	12	13	0
1137	02	SBSV	SBSP	10	11	1

Na Tabela I.1, cada linha é formada por uma lista ordenada das colunas especificadas acima e representa uma transação. O termo transação indica quais itens foram consultados em uma determinada operação de consulta. As RAs têm como premissa básica encontrar elementos que implicam na presença de outros elementos em uma mesma transação, ou seja, encontrar padrões frequentes entre itens.

Em tabelas de horários, é possível encontrar uma dependência entre os atributos especificados na tabela e que pode influenciar na geração das RAs. De acordo com [Elmasri and Navathe, 2016], a informação dos horários de partida e chegada que cruza a matriz OD caracteriza uma dependência, chamada de Dependência Funcional (DF), em relação aos dados de transporte. Uma DF é apresentada como um conjunto de atributos A que determina os valores de um outro conjunto de atributos B, formalmente definida por $A \rightarrow B$. Como exemplo, as dependências funcionais são determinadas pelo número do voo na Tabela I.1.

Com base na Tabela I.1, um exemplo de DF pode ser observado na regra $r1$: $d.flt = 1003 \rightarrow d.hrs = 10$ $[2, 100\%]$, com suporte igual a dois e confiança de 100%, pois, as dependências funcionais são determinadas pelo atributo $d.flt$ que representa o número do voo nesta tabela. Logo, todas as RAs que contenham os atributos relacionados à caracterização de um mesmo voo compõem uma DF com aquele voo. No entanto, todas as RAs derivadas dessa mesma DF não enriquecem o conhecimento e não precisariam ser analisadas. Essas RAs são consideradas redundantes.

As abordagens tradicionais de MPF geram muitas RAs devido a presença da DF entre o atributo que caracteriza primariamente cada transação de uma tabela com outros atributos desta mesma tabela. A presença da DF em uma tabela gera uma regra para cada valor distinto de cada um dos atributos que são funcionalmente dependentes do atributo que caracteriza a DF. Dessa forma, várias dessas RAs são redundantes, pois se referem à mesma informação, no caso, o atributo que caracteriza a DF.

Um exemplo de redundância pode ser verificado com base na regra $r2$: $d.flt = 1003 \wedge d.ori = SBRJ \rightarrow d.hrs = 10$ $[2, 100\%]$, com suporte igual a dois e confiança de 100%. Esta regra $r2$ também foi gerada a partir da Tabela I.1 e é uma regra redundante, pois, pode-se observar que a regra $r1$ do exemplo anterior é mais geral que a regra $r2$, e refere-se à mesma informação ou conhecimento, no caso, o voo 1003 levando a horário de partida igual a 10 horas da manhã.

Uma possibilidade de reduzir o número de RAs geradas por algoritmos de MPF é remover as RAs redundantes. Uma Regra de Associação (RA) é redundante se existir uma RA mais geral com a mesma ou com maior confiança [Hahsler et al., 2020] que a regra mais específica, em que ambas referem-se a mesma informação ou conhecimento. Mesmo aplicando-se algoritmos de remoção de redundância, eles não resolvem o problema, pois as técnicas de remoção de redundância são projetadas para serem utilizadas somente após o término da mineração, o que não reduz o custo computacional. Uma solução alternativa seria remover o atributo que gera a DF (neste caso, o número do voo). Porém, ao remover-se o atributo do número do voo, algumas RAs geradas passam a considerar conjuntos de itens que não preservam a DF. Portanto, RAs não contextualizadas são geradas, como, por exemplo, a regra $r3$: $d.hrs = 07 \rightarrow a.atr = 1$ $[2, 50\%]$.

Para ilustrar a influência da DF na geração das RAs na MPF, considere a Tabela I.1. Existe

uma DF entre os valores do atributo número do voo $d.flt$ e os valores dos atributos aeroporto de origem $d.ori$, aeroporto de destino $a.des$, hora de partida $d.hrs$ e hora de chegada $a.hrs$. Se o atributo número do voo $d.flt$ for usado para mineração de padrões e criação das RAs, algumas RAs produzidas não agregarão informações em uma análise de atraso, como as duas RAs a seguir, $r4$: $d.flt = 1003 \rightarrow a.atr = 1$ e $r5$: $d.flt = 1003 \wedge a.hrs = 11 \rightarrow a.atr = 1$. Ambas as RAs $r4$ e $r5$ têm suporte igual a dois e confiança de 100%.

Suponha que seja removido o atributo número do voo $d.flt$ do conjunto de dados. Tal remoção desfaz a DF entre a origem e o destino na Tabela I.1. Pode-se observar que algumas RAs geradas não possuem os atributos de origem ou destino, ou ambos, em sua formação, logo, a DF não é preservada; portanto, é uma regra não contextualizada. Isto se deve ao fato de que é comum que as rotas (OD) permaneçam os mesmos durante anos a fio, sendo modificados somente em caso de incidentes ou desastres aéreos, conforme discutido neste capítulo.

Em razão disto, semanticamente, o atributo do número do voo pode ser substituído pelos atributos de origem, ou de destino, ou ambos. E esta combinação diz respeito a própria característica de OD, em que pese, considera uma cardinalidade N para N ou, formalmente, N:N, ou muitos para muitos, onde têm-se: N origens para N destinos, e vice-versa.

Logo, esta cardinalidade prevê que tenha-se N voos partindo de uma origem para N destinos desconhecidos, ou N voos chegando a um destino, vindo de N origens desconhecidas, ou até mesmo, um voo partindo de uma origem conhecida para um destino conhecido. Isto, garante que todas as RAs serão geradas, considerando-se este relacionamento muitos voos para muitos voos, entre as origens e os destinos.

Por exemplo, pode-se mencionar as seguintes RAs geradas com a Tabela I.1, $r6$: $d.hrs = 07 \rightarrow a.atr = 1$ $[2, 50\%]$ e $r7$: $d.day = 2 \rightarrow a.atr = 1$ $[3, 50\%]$. Estas RAs podem pertencer a pares OD diferentes. Adicionalmente, as cardinalidades aqui discutidas, especificamente, as cardinalidades 1 para N ou N para 1, permitem outras análises, tais como: análise de gargalos em determinadas origens ou destinos, por meio das RAs geradas. No Capítulo II será discutido que a técnica de particionamento horizontal utilizada no método T-mine considera estas cardinalidades ao gerar os blocos de dados na origem e no destino, por meio de fragmentação.

Assim, surge o seguinte problema de pesquisa: como produzir RAs, a partir de uma tabela de horários associada a uma matriz OD, que melhore o custo computacional, quando comparado aos métodos tradicionais. O objetivo é obter um número reduzido de RAs não redundantes e, ao mesmo tempo, possibilitar uma análise de OD, preservando-se o contexto da DF.

Para explicar a motivação que nos levou ao desenvolvimento do método, vamos considerar novamente a Tabela I.1. Conforme pode ser visto na Figura I.3, a partir desta tabela de horários, contendo o registro de 12 voos e com os sete atributos que descrevem cada voo, temos um total de

k=25 itens distintos. Cada item é representado por um par atributo e o seu correspondente valor. Como exemplo, pode-se mencionar que o atributo número de voo $d.flt = 1137$ pode ser escrito como um par $< d.flt, 1137 >$ e que este par é um item em MPF. Isto faz com que seja gerado um número enorme de M conjuntos candidatos de itens, dado por $M = 2^k$ - 1, para se encontrar os padrões. Com base nesses números, a quantidade de regras possíveis R dado por $R = 3^k$ - 2^{k+1} + 1, também é enorme [Tan et al., 2019].

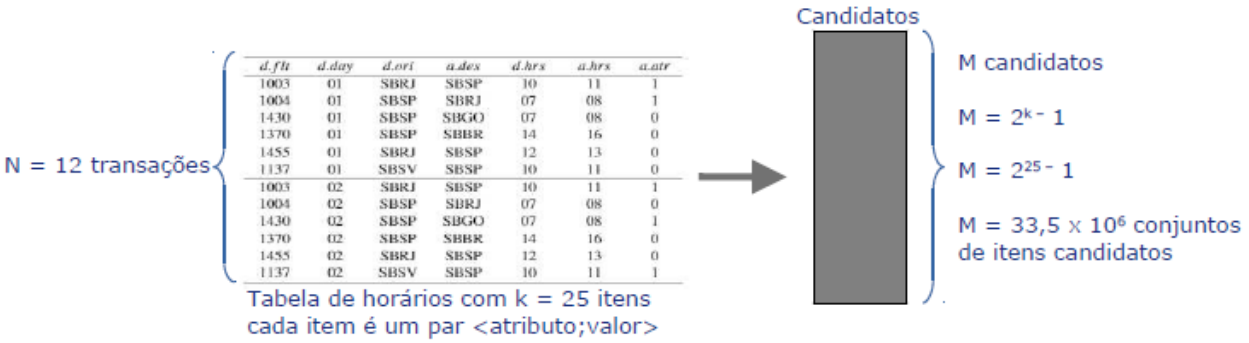


Figura I.3: Exemplo motivador para caracterização do problema

Portanto, esta pesquisa apresenta o **T-mine**, um novo método de descoberta de RAs não redundantes em tabelas de horários. O T-mine obtém uma redução significativa no número de RAs redundantes e preserva o contexto das Dependências Funcionais (DFs).

Nossos experimentos observaram que o T-mine supera as abordagens tradicionais na descoberta de RAs não redundantes. O T-mine tem dois pontos fortes: a eficiência computacional e a redução contextualizada. Isso é obtido por meio do particionamento horizontal e processamento paralelo. A estratégia de particionamento horizontal do T-mine reduz significativamente o custo de computação, melhorando o acesso à memória e o processamento paralelo. Ao carregar partições que são significativamente menores do que todo o conjunto de dados da tabela de horários, os algoritmos de MPF melhoram os movimentos de dados da memória. O particionamento horizontal também permite paralelismo de dados de forma independente. Os resultados evidenciaram que o método T-mine foi melhor em termos de tempo de execução e memória, o que responde à questão de pesquisa apresentada neste artigo. Adicionalmente, o número de RAs distintas e não redundantes do T-mine é menor do que as do baseline. Este resultado está alinhado com o objetivo de obter um número reduzido de RAs não redundantes e, ao mesmo tempo, permitir uma análise de origem e destino, preservando o contexto da DF.

Além desta introdução, este trabalho está dividido em mais cinco capítulos. O Capítulo II

descreve os principais conceitos que são utilizados no trabalho como: MPF, RAs, medidas interessantes, redundância, DF, particionamento horizontal e o algoritmo Apriori. O Capítulo III apresenta e comenta os trabalhos relacionados. O Capítulo IV, detalha a metodologia, e os resultados são apresentados e analisados no Capítulo V. E por fim, o Capítulo VI, conclui e aponta para trabalhos futuros.

Capítulo II Referencial Teórico

Neste capítulo, é apresentado o embasamento teórico e os trabalhos que mais se aproximam como referências para o desenvolvimento do método proposto. Como inicialmente contemplado por esta pesquisa, o foco principal deste trabalho é um novo método de descoberta de RAs não redundantes em tabelas de horários. O T-mine obtém uma redução significativa no número de RAs redundantes e preserva o contexto das DFs. Portanto, é apresentada uma seção sobre os formalismos que conceituam a MPF, o seu principal produto que são as RAs, as principais medidas de interesse que especificam a geração das RAs e o algoritmo de MPF usado como base na realização dos experimentos computacionais. Ressalta-se também, a importância de se considerar as definições sobre remoção de redundância, DF e a descrição sobre a técnica de particionamento horizontal.

II.1 MPF

A MPF é explorada para descobrir RAs em uma grande quantidade de dados [Aggarwal and Han, 2014]. Esta grande quantidade de dados está distribuída em transações de uma tabela. Cada par <atributo,valor> de uma transação em um conjunto de dados D corresponde a um item. Os itens são combinados em conjuntos distintos, formando-se conjuntos de itens. Um *padrão frequente* é um conjunto de itens que frequentemente aparecem em D [Han et al., 2011]. Encontrar padrões frequentes é uma parte essencial da descoberta de correlações e RAs interessantes entre itens.

As RAs mostram correlações entre conjuntos de itens e são descritos como: $A \rightarrow B$. O conjunto a esquerda da regra é conhecido como antecedente, enquanto que o conjunto a direita da regra é conhecido como consequente. Em uma RA: $A \rightarrow B$, A e B são conjuntos de itens. Para dois conjuntos de itens A e B, $A \cup B$, também é um conjunto de itens. Um item A é frequente em D se o número de ocorrências não for inferior a um limite definido pelo usuário.

Duas medidas interessantes que influenciam a geração das RAs são o suporte ($supp$) e a confiança ($conf$). Dado uma RA: $A \rightarrow B$ $[supp, conf]$ com suporte ($supp$) e confiança ($conf$), o suporte da regra $supp(A \rightarrow B)$ é a probabilidade do conjunto de itens $(A \cup B)$ em D, ou seja, $supp(A \rightarrow B) = supp(A \cup B) = \Pr(A \cup B)$. A confiança $conf(A \rightarrow B)$ é o suporte de $A \cup B$ dividido pelo suporte de A. A confiança pode ser interpretada como a probabilidade condicional de B dado A, tem-se, $conf(A \rightarrow B) = \Pr(B \mid A) = \dfrac{supp(A \cup B)}{supp(A)}$.

Existem vários métodos para a MPF e diversas maneiras para se reduzir a complexidade computacional da geração de conjuntos de itens frequentes, nos quais o algoritmo *Apriori* é um dos principais [Han et al., 2000]. Ao final de sua execução, ele produz um conjunto de RAs.

II.2 Apriori

O *Apriori* começa com conjuntos de itens de tamanho um. Este algoritmo identifica os conjuntos de itens k frequentes e os estende para conjuntos de itens $(k+1)$ até que esses itens sejam frequentes o suficiente no conjunto de dados.

Usando-se a Tabela I.1 do Capítulo I e com o intuito de simplificarmos um exemplo para demonstrar a geração dos conjuntos frequentes de itens, reduziu-se a dimensionalidade desta tabela para ficarmos somente com os seguintes itens: SBRJ, SBSP, 10 e 11, por conseguinte, tem-se a seguinte tabela com os conjuntos candidatos com apenas 1 item.

Tabela II.1: Conjuntos candidatos com 1 item

item	*contagem*
SBRJ	6
SBSP	12
10	4
11	4

Supondo um limite de suporte igual a 50%, equivalente a uma contagem mínima de suporte igual a 6, os itens 10 e 11 são descartados, pois aparecem em menos do que 6 transações. Logo, o algoritmo prosseguirá com os itens SBRJ e SBSP. Na iteração seguinte, os conjuntos candidatos com 2 itens são gerados usando-se apenas os conjuntos frequentes com 1 item, porque o princípio do Apriori, garante que todos os superconjuntos dos conjuntos infrequentes com 1 item deve ser infrequente.

Então, se existem 2 conjuntos frequentes com 1 item, o número de conjuntos candidatos com 2 itens gerado pelo algoritmo é dado por: $\binom{2}{2}=1$. Se não houvesse a poda com base no suporte, teríamos $\binom{4}{2}=6$ conjuntos candidatos com 2 itens. Têm-se, então, o único conjunto candidato com 2 itens: {SBRJ,SBSP}. Mas este conjunto tem uma contagem igual a 4, logo não atende a contagem mínima de suporte igual a 6, portanto, não é um conjunto frequente. Como resultado, o algoritmo retornaria apenas com os conjuntos com 1 item SBRJ e SBSP como sendo frequentes.

O *Apriori* possui uma propriedade importante chamada *antimonotonicidade* para melhorar a eficiência do processo de mineração de padrões. Esta propriedade define que cada subconjunto de um conjunto frequente de itens também deve ser frequente. Seja I um conjunto de itens e, $J=2^I$

um conjunto-potência de I. A medida f é monótona (ou fechado para cima) se

$$\forall\ X,Y \in J\colon (X \subseteq Y) \rightarrow f(X) \leq f(Y),$$

o que significa que, se X for um subconjunto de Y, então $f(Y)$ não excede a $f(X)$. [Tan et al., 2005]. Para ilustrar esta propriedade, considere a Tabela I.1 do Capítulo I. Supondo que o conjunto de itens {SBRJ,SBSP,10} seja frequente. Toda transação que contém {SBRJ,SBSP,10}, também deve conter {SBRJ}, {SBSP}, {10}, {SBRJ,SBSP}, {SBRJ,10}, e {SBSP,10}. Como resultado, se {SBRJ,SBSP,10} for frequente, todos os subconjuntos de {SBRJ,SBSP,10} também são frequentes.

Assim, os candidatos do conjunto de itens do espaço de busca são reduzidos. Por outro lado, se o conjunto de itens {SBRJ,SBSP} não for frequente (infrequente), como aconteceu no exemplo anterior em que este conjunto não atendeu a contagem de suporte mínimo, todos os seus super-conjuntos também não são frequentes. O pseudo-código do algoritmo Apriori para a geração de conjuntos frequentes de itens é mostrado pela Figura II.1 abaixo.

```
function candidates (F_k)
begin
   C := ∅;
   forall f_1, f_2 ∈ F_k
   with  f_1 = {i_1, ..., i_{k-1}, i_k}
   and   f_2 = {i_1, ..., i_{k-1}, i'_k}
   and   i_k < i'_k do begin
      f := f_1 ∪ f_2 = {i_1, ..., i_{k-1}, i_k, i'_k};
      if ∀i ∈ f : f − {i} ∈ F_k
      then C := C ∪ {f};
   end;
   return C;
end (* candidates *)
```

Figura II.1: Pseudo-código para geração dos conjuntos frequentes de itens do algoritmo *Apriori*. Fonte: `https://borgelt.net/docs/apriori.pdf`

Seja F_k um conjunto frequente de K itens. O algoritmo inicializa o conjunto C de candidatos. Para todos os pares $f1$, $f2$ de conjuntos de itens que difere apenas em um item e estão em uma ordem lexicográfica, realiza a união desses conjuntos que possuem $k + 1$ itens, apenas se todos os k subconjuntos de elementos forem frequentes, então, adiciona o novo conjunto de itens ao conjunto C de candidatos. Caso contrário, o conjunto de itens não é frequente e é descartado. Por fim, retorna os conjuntos de candidatos C. Para a contagem do suporte mínimo e poda dos conjuntos que não são frequentes, o algoritmo Apriori pode ser representado pelo pseudo-código mostrado pela Figura II.2.

Seja C o conjunto de candidatos frequentes gerado pela função *candidates* do algoritmo mostrado na Figura II.1 nesta subseção. Seja T o conjunto de transações que contém os itens de C e S_{min} o valor de suporte que contém a contagem mínima para considerar um conjunto como sendo um

```
function prune (C, T, s_min)
begin
    forall c ∈ C do
        s(c) := 0;
    forall t ∈ T do
        forall c ∈ C do
            if c ∈ t
            then s(c) := s(c) + 1;
    F := ∅;
    forall e ∈ E do
        if s_T(e) ≥ s_min
        then F := F ∪ {e};
    return F;
end (* prune *)
```

Figura II.2: Pseudo-código para a poda dos conjuntos candidatos que não são frequentes do algoritmo *Apriori*. Fonte: `https://borgelt.net/docs/apriori.pdf`

conjunto frequente. A função *prune* do algoritmo mostrado na Figura II.2 recebe os parâmetros C, T e S_{min} descritos e inicializa o contador c de suporte de todos os conjuntos candidatos que serão verificados. Para todas as transações t em T e para todos os conjuntos candidatos c em C, se a transação contém o conjunto candidato, incrementa o contador de suporte.

Na sequência, é inicializado o conjunto frequente F de candidatos. Para todos os candidatos, se o candidato é frequente, adiciona ele ao conjunto de conjuntos frequentes de itens. Por fim, retorna o conjunto podado de candidatos.

Em síntese, o *Apriori* emprega um limite de suporte e confiança para produzir as RAs. Para Tan et al. [2019], esta estratégia de reduzir o espaço de busca, baseado na medida de suporte é conhecido como poda baseada em suporte. As RAs são consideradas interessantes na análise se satisfizerem o suporte e a confiança mínimos estabelecidos pelo usuário. Embora estabelecer um suporte mínimo e limites de confiança evite muitas RAs desinteressantes, ele ainda pode gerar RAs que não são interessantes [Agrawal and Srikant, 1994].

O pseudo-código com o algoritmo Apriori completo e que utiliza as funções *candidates* e *prune* mostradas nos pseudo-códigos das Figuras II.1 e II.2, respectivamente, é mostrado na Figura II.3 abaixo. O algoritmo começa com conjuntos C_k de elementos únicos e determina os frequentes F_k. Enquanto houver conjuntos de itens frequentes, cria conjuntos de itens com mais um item, determina os frequentes e incrementa o contador k de itens. Na sequência, para todos os conjuntos frequentes f de itens, inicializa os cabeças de regra m, também chamado de consequentes que contém apenas um item.

Na sequência, para todas as cabeças de regra (consequentes da regra) [Agrawal and Srikant, 1994] de tamanho crescente H_m e para todas as possíveis cabeças de regra h, se a confiança da regra $\frac{s(f)}{s(f-h)}$ é alta o suficiente, adiciona-a ao resultado R, caso contrário, descarta a cabeça da

```
function apriori (I, T, s_min, c_min, k_max)
begin
    k  := 1;
    C_k := ⋃_{i∈I} {i};
    F_k := prune(C_k, T, s_min);
    while F_k ≠ ∅ and k ≤ k_max do begin
        C_{k+1} := candidates(F_k);
        F_{k+1} := prune(C_{k+1}, T, s_min);
        k       := k + 1;
    end;
    R := ∅;
    forall f ∈ ⋃_{j=2}^{k} F_j do begin
        m  := 1;
        H_m := ⋃_{i∈f} {i};
        repeat
            forall h ∈ H_m do
                if s(f) / s(f−h) ≥ c_min
                then R   := R ∪ {[(f − h) → h]};
                else  H_m := H_m − {h};
            H_{m+1} := candidates(H_m);
            m       := m + 1;
        until H_m = ∅ or m ≥ |f|;
    end;
    return R;
end (* apriori *)
```

Figura II.3: Pseudo-código do algoritmo *Apriori* para encontrar as RAs. Fonte: `https://borgelt.net/docs/apriori.pdf`

regra $\{h\}$. Seguindo, cria as cabeças de regra H_{m+1} com mais um item, incrementa o contador de cabeças de regra m até não existir mais cabeças de regra H_m ou ou o antecedente se tornar vazio $m \geqslant |f|$. Por fim, retorna as RAs R encontradas.

II.3 Remoção de Redundância

A redução do espaço de busca dos conjuntos frequentes e, por conseguinte, a geração de regras com a opção de remoção das redundantes, está diretamente relacionado com o problema de redução do custo computacional na MPF. Devido a este fato, é de grande interesse identificar pequenas representações de conjuntos de itens, os quais a partir de sua derivação, são capazes de gerar todos os outros conjuntos de itens frequentes. Sob este aspecto, as RAs podem ser filtradas de acordo com a sua redundância. Dados duas RAs $(A \rightarrow B)$ e $(\hat{A} \rightarrow B)$, de modo que $\hat{A} \subset A$ e $supp(A \rightarrow B) \cong supp(\hat{A} \rightarrow B)$. É possível não apresentar a regra $(\hat{A} \rightarrow B)$, visto que $(A \rightarrow B)$ é uma regra mais geral [Bayardo Jr. et al., 2000].

Ainda no contexto de remoção de redundância, um assunto importante que deve ser considerado, é a presença da DF. Uma DF é um relacionamento entre os atributos de uma relação. Este relacionamento é válido se o valor de um atributo é determinado exclusivamente pelos valores de

alguns outros atributos [Huhtala et al., 1999]. Formalmente, uma DF $X \rightarrow Y$, entre dois conjuntos de atributos X e Y que são subconjuntos de uma relação R especifica uma restrição nas possíveis tuplas que formam o estado da relação r de R. A restrição é, para qualquer duas tuplas t_1 e t_2 em r que tenha $t_1[X] = t_2[X]$, também tem que ter $t_1[Y] = t_2[Y]$ [Elmasri and Navathe, 2016].

Considere o seguinte exemplo, onde têm-se duas regras geradas, no contexto da presença da DF, caracterizada pelo atributo $d.flt$ que primariamente identifica um voo. Seja a regra r_1: $\{d.flt$=3820, $d.ori$=SBRJ, $d.hrs$=10$\} \rightarrow \{a.atr$=1$\}$ e a regra r_2: $\{d.flt$=3820, $d.ori$=SBRJ, $a.hrs$=12$\} \rightarrow \{a.atr$=1$\}$, estas duas regras referem-se a mesma informação, ou seja, número de voo igual a 3820, com partida às 10h da manhã no aeroporto Santos Dumont no Rio de Janeiro e que chegou com atraso às 12h no seu destino não informado. Logo, são regras redundantes e poderiam ser descartadas ou não fazer parte da análise das RAs.

A princípio, para implementar a estratégia de remoção de redundância, teríamos as seguintes alternativas: (i) remover as duas regras, o que não seria bom, pois perderíamos toda a informação por completo; (ii) remover uma das duas regras, o que também não é aconselhável, pois, perderia-se uma parte da informação; ou (iii) unir duas regras em uma só, tendo-se como resultado, por exemplo, uma regra r_3 dada por $\{d.flt$=3820, $d.ori$=SBRJ, $d.hrs$=10, $a.des$=12$\} \rightarrow \{a.atr$=1$\}$. Porém, esta estratégia de fusão ou *merge* de duas regras redundantes em uma só regra não redundante não ocorre por meio da remoção de redundância proposta por [Bayardo Jr. et al., 2000].

E como o método T-mine aborda esse problema? Primeiro, por meio da remoção do atributo que causa a DF que, tomando-se o mesmo exemplo descrito acima, as seguintes regras são geradas: r_4: $\{d.ori$=SBRJ, $d.hrs$=10$\} \rightarrow \{a.atr$=1$\}$ e r_5: $\{d.ori$=SBRJ, $a.des$=12$\} \rightarrow \{a.atr$=1$\}$ e estas regras não são redundantes, pois, mesmo preservando-se a DF por meio do atributo de origem na formação destas regras, como não há a informação do destino, não há como afirmar que são regras que tratam do mesmo voo, como ocorreu nos exemplos anteriores em que, há o atributo que identifica aquele voo.

Por outro lado, pode ocorrer o seguinte cenário de regras redundantes: r_6: $\{d.ori$=SBRJ, $a.des$=SBSP, $d.hrs$=10$\} \rightarrow \{a.atr$=1$\}$ e r_7: $\{d.ori$=SBRJ, $a.des$=SBSP, $a.hrs$=12$\} \rightarrow \{a.atr$=1$\}$, neste exemplo, estas duas regras também se referem a mesma informação e não há a presença da DF, a DF está preservada pela presença dos atributos de origem e de destino na formação destas regras. Em outras palavras, temos duas regras onde o voo parte às 10h da manhã do Rio de Janeiro-Santos Dumont com destino São Paulo-Congonhas, chegando com atraso às 12h. Neste caso, o método T-mine faz a junção ou *merge* destas duas regras em uma só, gerando a regra r_8: $\{d.ori$=SBRJ, $a.des$=SBSP, $d.hrs$=10, $a.des$=12$\} \rightarrow \{a.atr$=1$\}$, e esta regra não é redundante, pois não existe nenhuma outra regra mais geral que esta que apresenta esta informação.

II.4 Particionamento Horizontal

Uma partição é uma divisão de um banco de dados lógico ou seus elementos constituintes em partes independentes distintas. O particionamento de banco de dados normalmente é feito por razões de gerenciabilidade, desempenho ou disponibilidade. O processo de particionamento divide uma tabela em blocos de dados para fins de armazenamento físico (em dispositivos externos) ou lógicos (em memória). Permite melhorar o desempenho no acesso aos dados, pois eles podem ser manipulados em memória com mais rapidez, devido a utilização de blocos de dados menores [Pajwani and Bosamia, 2013]. Existem duas maneiras de se realizar particionamento de bancos de dados: (i) Particionamento Vertical, onde as colunas de uma tabela são divididas e armazenadas em tabelas separadas (ou bancos de dados separados); e (ii) Particionamento Horizontal, onde as linhas de uma tabela são divididas e armazenadas em tabelas distintas no bancos de dados. O particionamento horizontal também é conhecido como *sharding*. Portanto, o particionamento serve para melhorar o desempenho e o custo computacional [Özsu and Valduriez, 2019].

Em banco de dados, a fragmentação é um tipo de particionamento horizontal que divide bancos de dados grandes em componentes menores, que são mais rápidos e fáceis de serem gerenciados. Faz-se necessário encontrar um nível adequado de fragmentação que mantenha um compromisso entre os dois extremos: (i) Nenhuma fragmentação; e (ii) Fragmentação até o nível de tuplas ou de atributos. Este equilíbrio pode ser determinado de acordo com as aplicações que farão uso do banco de dados e com um conjunto de parâmetros que caracterizam essas aplicações.

A fragmentação horizontal de uma relação R, geram fragmentos que são subconjuntos horizontais de R, ou seja, dado o conjunto de n tuplas de R, F_1 é um fragmento contendo um subconjunto de tuplas de R. A fragmentação horizontal usa um critério de fragmentação por meio de um predicado de seleção aplicado sobre as tuplas da relação R. Esse predicado é uma expressão atributo-valor. Logo o que fragmenta a relação são os valores de um ou mais atributos de uma tupla.

A fragmentação possui algumas propriedades. A primeira é a completeza ou completude. Se uma instância da relação R é decomposta em fragmentos F_1, F_2, ..., F_n, cada item de dados que pode ser encontrado em R também pode ser encontrado em um ou mais fragmentos F_i. Esta propriedade é idêntica a decomposição sem perdas da normalização em banco de dados. Os dados em uma relação são mapeados em fragmentos sem perdas. Outra propriedade é a disjunção. Se uma relação R é decomposta horizontalmente em fragmentos F_1, F_2, ..., F_n, e o item de dados d_1 está em R_j, então, este item de dados d_1 não está em qualquer outro fragmento $R_k(k \neq j)$.

Os blocos de dados, após a fragmentação horizontal, podem ser mais rapidamente movimentados em memória, devido a menor quantidade de dados, e processados em paralelo. O paralelismo processa vários blocos de dados distintos e assim, se beneficia da fragmentação inicial realizada. E

isto faz com que se tenha uma menor quantidade de memória alocada e, posteriormente, menor tempo de execução, reduzindo-se o custo computacional. A fragmentação no modelo relacional discutida acima é perfeitamente aplicável para o conjunto de transações de uma tabela de horários que será usada para se gerar as RAs.

II.5 Análise estatística das RAs

Com o intuito de garantir que as RAs geradas sejam estatisticamente significantes, faz-se necessário realizar uma análise estatística destas RAs. De acordo com Lallich et al. [2007], uma abordagem inspirada no método *Bootstrap* pode ser realizada, gerando-se um conjunto de RAs estatisticamente significantes R', onde $R' \subset R$ (conjunto de RAs geradas). Esta abordagem propõe, repetir l vezes, a coleta de uma amostra aleatória, com reposição, e de mesmo tamanho com m regras, a partir do conjunto de regras geradas T. Algumas regras $t \in T$ podem não estar nas amostras T', mas outras poderão se repetir muitas vezes.

Outro conceito importante de estatística utilizado neste trabalho refere-se aos erros de aceitar padrões sem significância estatística e rejeitar padrões verdadeiros, padrões em que não ocorrem os conhecidos erros do tipo I (quando a hipótese nula é verdadeira e rejeitada) e tipo II (quando a hipótese nula é falsa e não é rejeitada). Para ilustrar a influência das hipóteses estatísticas e os erros associados, pode-se tomar como exemplo, uma hipótese nula H_0 que representa as regras contextualizadas e que são estatisticamente significantes, conforme discutido na Seção V.5 do Capítulo V. Quando H_0 é verdadeira e não rejeitada, a decisão é correta com probabilidade (1 - α), onde α é o nível de significância e (1 - α) é a confiança definida para a geração das regras. Ainda no cenário de H_0 ser verdadeira e for rejeitada, ocorre o erro do tipo I com probabilidade igual a α ou chance pequena de ocorrer este erro. Por outro lado, quando ocorre de H_0 ser falsa e não rejeitada, ocorre o erro do tipo II com probabilidade igual a β e caso contrário, sendo rejeitada, a decisão é correta com probabilidade (1 - β). Como solução, o uso de técnicas estatísticas ou outras medidas objetivas são usadas para classificar as regras ou para orientar a pesquisa [Morishita and Sese, 2000]. Dentre estas técnicas estatísticas, destaca-se a técnica de re-amostragem, também conhecida como técnica de replicação, pois é simples e muito eficiente para obter o intervalo de predição.

Por meio desta técnica, executa-se n replicações independentes da população, onde aqui, nesta pesquisa, é representada pelo conjunto de RAs distintas e não redundantes geradas, onde cada replicação produz uma amostra ou saída x_1, ..., até x_n [Cumming, 2006]. Esta técnica de replicação somente é possível graças ao conceito do Teorema do Limite Central (TLC). O TLC estabelece que, quando variáveis aleatórias independentes são adicionadas, sua distribuição tende a ser uma distribuição normal (informalmente sob a forma de uma curva de sino) mesmo se as próprias variáveis originais (da população) não sejam distribuídas normalmente, conforme pode ser visto em Islaqm

[2018]. Neste trabalho, a análise das RAs distintas e não redundantes sob o ponto de vista da frequência, utilizou-se o cálculo de escore ou pontuação padrão (*z-score*). Conforme nos mostra o trabalho de Behboodian and Asgharzadeh [2008], o escore padrão pode ser definido como sendo uma sequência de amostras aleatórias X_1, ..., X_n com uma distribuição com média $\bar{X}$ e variância S^2.

Capítulo III Trabalhos Relacionados

No tocante aos trabalhos relacionados, vislumbrou-se a busca pelos trabalhos mais recentes, associados a análise de tabelas de horários. Mais especificamente, buscou-se trabalhos que utilizavam técnicas de MPF para auxiliar na construção de regras de associação e a partir delas geração de conhecimento aplicadas em tabelas de horários. Inicialmente, a busca não foi restringida a trabalhos que utilizavam técnicas de particionamento, como o apresentado nesta dissertação.

Em razão da fonte de dados ser uma tabela de horários e, pelo fato do conhecimento de domínio prévio de uma tabela de horários aplicada a área de transporte, ter como propriedade intrínseca, a presença da DF, buscou-se também por obras de MPF com a presença da DF. E analisou-se como a DF influenciou ou não, o resultado destas pesquisas correlacionadas. Complementarmente, buscou-se também por trabalhos que utilizam tabelas de horários de outras áreas, como por exemplo, a área educacional.

Além dos temas mencionados, buscou-se ainda trabalhos de MPF associados ao tema de remoção de redundância. Estas duas temáticas DF e remoção de redundância são importantes neste trabalho, em razão de um dos objetivos desta pesquisa ser a redução do número de RAs não redundantes. Então, foram encontrados trabalhos que tratam de problemas similares. Não foram encontrados trabalhos que tratam do mesmo problema específico. Todos os trabalhos encontrados foram classificados (agrupados) em temas (taxonomia) e são apresentados nas respectivas seções deste capítulo seguido de considerações, limitações, como os trabalhos diferem da nossa proposta, identificação de lacunas e oportunidades exploradas por esta pesquisa.

Em relação à literatura disponível sobre MPF e *tabelas de horários*, duas *queries* foram realizadas na base Scopus em dezembro de 2020 para encontrar trabalhos relacionados. Ambas as *queries* consideraram títulos, palavras-chave e resumos e foram consideradas apenas artigos (Jornais e Conferências) inteiramente escritos em inglês. A primeira *query* foi: ("*data analytics*" ∨ "*data mining*") ∧ ("*timetable*") ∧ ("*transport*" ∨ "*flight*" ∨ "*rail*"). O objetivo era encontrar trabalhos relacionados que usassem técnicas de mineração de dados para estudar *tabelas de horários* de transporte. A segunda *query* foi: ("*association rules*" ∨ "*frequent pattern*") ∧ ("*timetable*" ∨ "*functional dependency*" ∨ "*redundancy*"). Teve como objetivo encontrar trabalhos relacionados de mineração de padrões frequentes associados a *tabelas de horários*, dependência funcional ou redundância. Ambas as *queries* juntas retornaram 305 artigos. Todos os resumos foram lidos e 19

foram selecionados para posterior análise de trabalhos relacionados.

III.1 Análise de tabelas de horários de transporte

Em quase duas décadas de literatura, é possível observar um crescente interesse por técnicas de Ciência de Dados na investigação de atrasos em voos [Carvalho et al., 2020]. Em estudos de aviação, a ciência de dados e a análise de dados estimularam novas ideias de pesquisa, conhecimento, teorias, técnicas e aplicações. Isso resultou em muitas novas intuições e soluções para problemas anteriormente não resolvidos [Chung et al., 2020]. Ainda de acordo com Chung et al. [2020], o uso de análise de *big data*, gerenciamento de rede de transporte aéreo, previsão e aprendizado de máquina são alternativas na área analítica de dados. A partir dessas opções podemos citar que os processos de gestão e previsão da rede de transporte aéreo estão sendo aprimorados com o uso da análise de tabelas de horários como ferramenta de apoio a esses processos de fato.

A título de exemplo, é possível verificar que, no transporte aéreo, um atraso de chegada previsto leva em consideração tanto as informações do voo (aeroporto de origem, aeroporto de destino, horário de partida e chegada programados) e as condições meteorológicas no aeroporto de origem e aeroporto de destino de acordo com a tabela de horários dos voos [Belcastro et al., 2016]. O nosso trabalho utiliza um conjunto de dados integrados, denominado de *Brazilian Flight Dataset* que integra informações de voos com as condições meteorológicas. E adicionalmente, as análises realizadas, utilizando-se o nosso método, foi no contexto de atrasos na chegada, ao fixarmos o lado direito da regra, indicando que houve atraso. Na busca pela melhoria contínua dos serviços de transporte, um modelo de reescalonamento das tabelas de horários visa melhorar a pontualidade desses serviços. Neste modelo, a causa raiz do atraso pode ser descoberta por filtragem de frequência nos registros de operação [Lee et al., 2016]. Comparando-se com o nosso método, o T-mine não filtra a frequência das transações de entrada para o algoritmo de MPF e, sim, por meio de técnicas estatísticas, realizou-se uma análise das regras distintas e não redundantes com base na frequência de cada uma delas, ao encontrarmos as regras que mais se repetiram ao longo do período de 2000 a 2019 e provando-se que estas regras são estatisticamente significantes.

Outros trabalhos levam à análise dos tempos de viagem. Cristóbal et al. [2018] apresentam uma metodologia, baseada em mineração de dados, para análise do tempo de viagem em um sistema de transporte de massa planejado por tabela de horários. Os tempos de viagem e sua relação com a tabela de horários também foram abordados por Carrel et al. [2015]. Jäppinen et al. [2013] usam rotas abertas e bancos de dados de tabelas de horários e ferramentas desenvolvidas internamente para realizar mineração de dados extensiva por meio de *Application Program Interfaces* (APIs) ou Interfaces de Programação de Aplicativos. Por meio do T-mine também é possível realizar análise dos tempos de viagem e descobrir em quais rotas pode estar havendo gargalos.

Em geral, para o desenvolvimento da abordagem do T-mine, escolheu-se tabelas de horários como conjunto de dados a ser minerado, devido a ampla gama de questões que podem ser resolvidas por meio da análise de RAs geradas a partir da MPF, apoiando áreas de transporte, complexa pela sua própria natureza logística, educacional, com a ampliação da oferta de turmas/disciplinas em horários melhores distribuídos com base nas necessidades históricas dos discentes. E esta diversidade de questões associada a DF, característica intrínseca das tabelas de horários, dificulta a análise destes padrões, principalmente nos casos em que, no espaço de busca, se encontrem regras redundantes ou regras que nos levam a mesma informação.

Os trabalhos descritos nesta subseção estão diretamente associados ao método T-mine, pois o T-mine pode ser utilizado como ferramenta para apoiar a responder as questões apresentadas por estes trabalhos, conforme pode ser visto na Figura III.1. As questões que podem ser respondidas por meio da utilização do método T-mine são as questões propostas por Carvalho et al. [2020] e Belcastro et al. [2016] ao analisar atrasos no transporte aéreo, o uso de *data analytics* em tabelas de horários para apoiar os processos de gestão e previsão da rede de transporte aéreo preconizado por Chung et al. [2020], o foco na melhoria contínua dos serviços de transporte relatado por Lee et al. [2016], análise de fatos relevantes, como por exemplo, a análise dos tempos de viagens e posterior refinamento dos horários previstos das tabelas de horários visto em Cristóbal et al. [2018], Carrel et al. [2015] e Jäppinen et al. [2013].

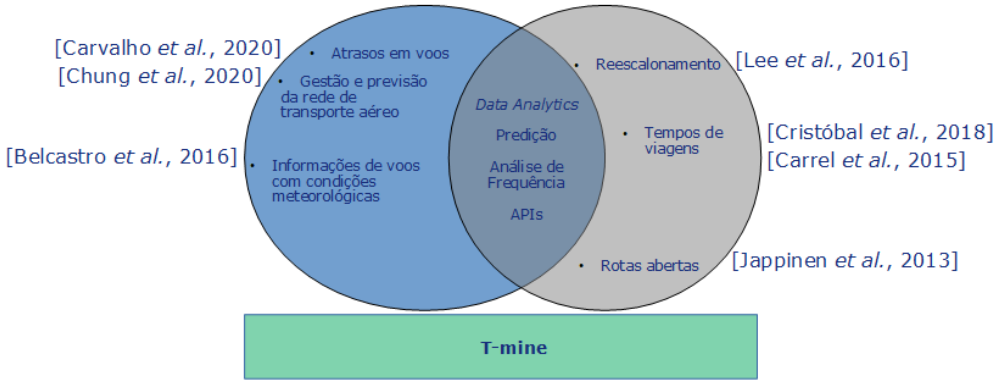


Figura III.1: Caracterização dos trabalhos relacionados

III.2 MPF em tabelas de horários

Nossa revisão sistemática mostra que manter tabelas de horários atualizadas e que atendam a ampla maioria das necessidades, tais como, (i) na área de transportes: minimizar meios ociosos, desequilíbrio entre a oferta e demanda e otimização de rotas. Ou (ii) na área educacional: a eliminação de janelas (horários ociosos entre duas aulas de um mesmo turno) e aulas isoladas (aquelas presentes isoladamente em um dia), e independente da área de aplicação da tabela de horários, seja em qual for, têm sido um desafio. Por exemplo, podem-se citar trabalhos em que, com base em

dados históricos de preferência de horários em escolas, faculdades ou qualquer estabelecimento de ensino, seja possível prever padrões de frequência e, com base nesses padrões, refinar a elaboração de uma nova e atualizada tabela de horários. Isso pode ser visto no trabalho de Albalooshi and Shatnawi [2021].

Para Rahim et al. [2013], pesquisas contínuas e trabalhos na implementação do algoritmo Apriori em ambiente educacional comprovaram que as RAs têm evoluído. Eles analisam o padrão de matrícula dos alunos do último semestre de uma universidade e descobrem conhecimentos interessantes que podem ajudar a equipe gestora da tabela de horários na redução de conflitos de horários e para futuras tomadas de decisão ao longo do processo gradual. As regras fortes baseadas nos dados extraídos podem ser aplicadas no processo de preparação da tabela de horários existente para agregar novo valor a fim de aumentar o desempenho do sistema enquanto reduz o número de conflitos de classe. Os autores propõem um algoritmo de árvore de padrões frequentes (*FP-tree*) adaptado para a predição de informações. Por meio do T-mine, também vislumbra-se a sua utilização para reescalonamento das tabelas de horários, como sendo tema de trabalho futuro e previsto na Seção VI.2 do Capítulo VI.

Susan and Bhutani [2020] propuseram o uso de Algoritmo Genético e Algoritmo de Arrefecimento Simulado para geração e otimização automatizada de tabelas de horários. Os horários gerados são analisados em relação à eficiência de tempo e otimização da função de custo. Aqui trata-se do mesmo caso de reescalonamento de tabelas de horários, sendo um componente adicional a geração automática da tabela. Uma vez percebido o uso do T-mine para o reescalonamento, fica como passo adicional, um estudo para geração automática de tabelas de horários.

Soni et al. [2015] desenvolveram um método para mineração de conjuntos de itens frequentes. Segundo eles, uma tabela de contagem de suporte atualizada contém a contagem de frequência de cada conjunto de itens. Para encontrar o conjunto de itens frequentes para qualquer valor limite, ele executou uma varredura na tabela, não no banco de dados. Esta abordagem tem apenas uma única varredura do banco de dados e nenhuma geração de candidato é necessária. Esta metodologia pode manter a transação na tabela de horários previsível e, em seguida, fornecer três zonas de espessura exclusivas centralizadas em torno da transação e suporte mínimo que é baixo (L), Médio (M), Alto (H). Considerando o suporte zonal, ele classifica o conjunto para poda. Em relação ao T-mine têm-se duas diferenças: (i) utilização da própria tabela de horários para realização da contagem do suporte; e (ii) não gerar conjuntos frequentes de itens candidatos a MPF. Existe então, uma oportunidade de se realizar uma análise ortogonal entre os métodos, ou seja, planejar experimentos em que serão estudados os efeitos das diferenças entre os métodos, medindo-se os custos e os benefícios de cada um deles.

Jorge et al. [2012] estudaram o desvio da duração da viagem de ônibus e suas causas. Os desvios

são obtidos comparando os horários programados com a duração real da viagem e são atrasos ou chegadas antecipadas. Os autores utilizam regras de distribuição, uma espécie de RAs que podem ter distribuições contínuas no consequente. Segundo eles, as RAs permitem a identificação sistemática de condições particulares, que chamamos de contextos, nas quais a distribuição dos desvios dos tempos de viagem difere significativamente da distribuição geral dos desvios. Após identificar as causas específicas do atraso, os gerentes operacionais da empresa de ônibus podem fazer ajustes nas tabelas de horários, aumentando a pontualidade, sem interromper o serviço. O T-mine também gera regras contextualizas, basta que o contexto das RAs a serem geradas reflita a regra de negócio especificada que, neste caso, são os desvios de duração das viagens.

Os trabalhos de Albalooshi and Shatnawi [2021], Rahim et al. [2013] e Susan and Bhutani [2020] utilizam os algoritmos *Apriori* (mesmo algoritmo de MPF do nosso trabalho), *FP-tree* e o algoritmo genético, respectivamente, sendo todos estes algoritmos pertencentes a categoria de algoritmos de MPF, utilizados, nestes trabalhos específicos, para a geração de padrões que são analisados para reescalonamento de tabelas de horários.

Em geral, todos estes trabalhos buscam aumentar o desempenho da MPF, assim como o nosso, por meio de melhoria do custo computacional. Futuramente é possível a realização de uma pesquisa de comparação, após a implementação do método T-mine por meio da generalização do algoritmo, conforme mencionado na Seção VI.2 do Capítulo VI.

O trabalho de Rahim et al. [2013] propõe a redução do número de conflitos de classes (turmas) que, em uma análise mais detalhada, é realizada por meio da análise das RAs em que uma turma/disciplina leva a outra e estes padrões são comparados com os padrões da tabela de horários original. Já o trabalho de Soni et al. [2015] propõe um método para mineração de conjuntos de itens frequentes fechado, enquanto o nosso método utiliza como inspiração para remoção de redundância o algoritmo *Dense-Miner* proposto por Bayardo Jr. et al. [2000] e descrito na Seção II.3 do Capítulo II.

III.3 MPF e DF

Nesta subseção, a revisão buscou trabalhos de MPF que abordaram a DF. Baixeries et al. [2014] consideram a descoberta de dependências funcionais como a relação binária de um contexto formal, cujas implicações são equivalentes a essas dependências. No entanto, isso leva a uma nova representação de dados que é quadrática no número de objetos com respeito aos dados originais. Os autores propuseram uma alternativa para evitar essa representação de dados e mostram como caracterizar DFs usando o formalismo de estruturas de padrões, uma extensão de *Formal Context Analysis* (FCA) ou Análise Formal de Conceito é clássica para lidar com dados complexos. No T-mine remove-se o atributo que causa primariamente a DF, porém, ela é preservada por meio da

geração das RAs contextualizadas.

Adicionalmente, o trabalho de Baixeries et al. [2014] propôs o uso de FCA para caracterizar a descoberta dos padrões. Este trabalho se relaciona com o nosso, no sentido de que existe uma relação de DF entre os conjuntos e os superconjuntos de itens identificados e estruturados em árvore, sendo que, especificamente os superconjuntos podem ser eliminados, desde que não sejam frequentes e, os conjuntos imediatamente anteriores a este superconjunto eliminado, serem considerados conjuntos candidatos à geração de regras não redundantes.

Papenbrock et al. [2015] descreveram, avaliaram e compararam os sete algoritmos mais citados e mais importantes para descoberta de DFs em tabelas. Os algoritmos são: *TANE*, *Fun*, *FD_Mine*, *Dfd*, *DepMiner*, *FastFDs* e *Fdep*. Os autores mostram que todos os algoritmos de busca de DF otimizam para certas características de dados e fornecem dicas sobre quando escolher qual algoritmo. Ainda de acordo com os autores, todas as abordagens atuais escalam surpreendentemente mal, mostrando potencial para pesquisas futuras. Apenas com base em uma análise preliminar teórica, entende-se que *TANE* e *FD_Mine* não podem ser usados para a MPF de tabelas de horários com DF, mas, como trabalho futuro, supõem-se que estes algoritmos possam ser utilizados na descoberta das DFs e auxiliar o método T-mine. Em relação aos outros algoritmos faz-se necessário um estudo mais detalhado.

Papenbrock et al. [2015] ao compararem os sete algoritmos clássicos de MPF, concluíram que tais abordagens não tiveram um bom desempenho e que existe uma lacuna para a melhoria do custo computacional, objetivo geral do nosso método. Uma das propostas descritas na Seção de VI.2 do Capítulo VI é a generalização de um algoritmo com base no nosso método e realizar uma pesquisa de comparação com outros algoritmos de MPF e verificar se de fato, existe uma melhoria do custo computacional. Já a abordagem proposta por Naidenova [2013] pode ser usada para comprovar as RAs geradas pelo nosso método que, é um subconjunto das regras lógicas definidas por estes autores.

Um levantamento analítico das abordagens atuais para minerar RAs é apresentado por Naidenova [2013], incluindo regras lógicas (implicativas, RAs e DFs), as definições equivalentes dos conceitos críticos na mineração de todos os tipos de regras lógicas: item, conjunto de itens, conjunto de itens frequentes, conjunto de itens máximo, conjunto de itens maximamente redundantes, gerador, gerador mínimo (conjunto de itens não redundantes ou irredundantes), conjunto de itens fechados, suporte e confiança, a base da mesma linguagem matemática (a teoria da rede) para analisar as interconexões entre eles e apresenta o *Diagnostic Test Approach* (DTA) ou Abordagem de Teste de Diagnóstico para minerar regras lógicas. T-mine gera RAs distintas e não redundantes, oriundas de conjuntos frequentes de itens, logo é um tipo de regras lógicas.

III.4 MPF e análise de redundância

Ainda é um desafio na MPF e considerado como o principal fator que atrapalha as aplicações de geração das RAs, o grande número de RAs retornadas pelo processo de mineração, conforme destacamos no Capítulo I. Como alternativa de solução, está a remoção de regras redundantes. Considerando-se este contexto, nossa revisão buscou trabalhos que tentassem solucionar esse problema.

Cheng et al. [2008] definiram uma solução que apresenta resultados de mineração de regras concisas eliminando a redundância no conjunto de RAs. Esses autores adotam o conceito de σ-*tolerance* para definir o conjunto de RAs σ-*tolerance*(σ-TARs), que é uma representação concisa do conjunto de RAs. A noção de σ-*tolerance* é um relaxamento no fechamento definido no suporte de conjuntos de itens frequentes, permitindo eliminar as RAs redundantes de forma eficaz.

Ainda no contexto de regras concisas, Xu and Li [2007] apresentaram uma definição relaxante para redundância sob a qual a base exata *Min-max* ainda contém regras redundantes; em seguida, eles propuseram uma representação condensada chamada Base exata confiável para regras exatas de associação. As regras na base exata confiável não são apenas não redundantes, mas também mais sucintas do que as regras na base exata *Min-max*. Uma base de regra de associação condensada a partir da qual regras não redundantes mais concisas podem ser extraídas. Os autores adotaram uma definição de RAs não redundantes que têm antecedentes mínimos e consequentes máximos.

Durante a revisão sistemática da nossa pesquisa, pode-se verificar que alguns trabalhos buscaram associações baseadas no conceito de conjuntos de itens frequentes fechados. De acordo com Zaki [2000], o conjunto de todos os conjuntos de itens frequentes fechados pode ser ordens de magnitude menor do que o conjunto de todos os conjuntos de itens frequentes, especialmente para conjuntos de dados reais (densos).

Os conjuntos de itens fechados determinam exclusivamente o conjunto de todos os conjuntos de itens frequentes e sua frequência exata. Os autores observam que o uso de conjuntos frequentes de itens máximo resulta em perda de informações, uma vez que a frequência de subconjunto não está disponível. Conjuntos de itens máximos não podem ser usados para geração de regras, uma vez que o suporte de subconjuntos é necessário para o cálculo de confiança. Considerando-se esta abordagem, este tipo de algoritmo está interessado em encontrar todas as regras de alto suporte e alta confiança. A complexidade da geração de regras nos métodos tradicionais é $O(f \cdot 2^l)$, exponencial no comprimento l do conjunto de itens frequente mais longo (f é o número total de conjuntos de itens frequentes) . Por outro lado, usando a estrutura de conjunto de itens fechados, o número de regras não redundantes é linear no número de conjuntos de itens fechados Zaki [2004].

Outro tipo de abordagem foi proposta por Jeyapal and Ganesan [2016] através da aplicação de

um algoritmo linear onde um conjunto de regras não redundante é inicializado como um conjunto vazio. Então, todo o conjunto de itens da primeira regra é armazenado em *intial_node* e a próxima regra é armazenada em *present_node*. Se *present_node* não for nulo, os dois nós serão comparados. Se o conjunto de itens *present_node* apontar para o conjunto de itens *intial_node*, o algoritmo eliminará essa regra do conjunto de regras, caso contrário, essa regra permanecerá no conjunto de regras. De forma concreta, se houver uma regra r: $item_1 \wedge item_2 \rightarrow item_3$, todas as regras geradas pela combinação de $item_1$, $item_2$ e $item_3$ apresentadas na regra r mesmo no antecedente ou consequente das outras regras são consideradas redundantes e são eliminadas.

As regras concisas abordadas por Cheng et al. [2008] e Xu and Li [2007] são semelhantes as regras não redundantes que são geradas pelo nosso método. No caso específico do trabalho de Xu and Li [2007] foi definido RAs não redundantes que têm antecedentes mínimos e consequentes máximos, o que difere do nosso trabalho que foca na geração de regras não redundantes do tipo *"then-part"* com a fixação do lado direito da regra ou *Right Hand Side* (RHS), até porque, entende-se que regras com muitos consequentes poderiam não ajudar muito nas análises, devido a complexidade de semântica nesta formação e que, pode ser uma restrição.

Os trabalhos de Zaki [2000] e Zaki [2004] propõem a busca pelas regras formadas com base nos conjuntos de itens frequentes fechado ou máximo, como sendo as regras não redundantes. Talvez sejam estes trabalhos sejam os principais rivais do nosso método. Existe uma oportunidade de se realizar uma comparação, executando-se ambos os métodos e verificar qual deles gera a menor quantidade de regras máximas ou regras não-redundantes, com base na definição proposta por Zaki et al. [1997]. Já em relação ao trabalho de Jeyapal and Ganesan [2016] entende-se, a princípio que, somente com base na descrição do algoritmo proposto, muitas regras não redundantes deixariam de ser geradas, incorrendo em perda de informações. De qualquer forma, pode-se propor experimentos comparativos com o nosso método e que comprovem essa suspeita.

III.5 Resumo

Como se pode verificar em todas obras selecionadas e analisadas neste capítulo, nenhuma delas está associada a MPF em tabelas de horários na área dos transportes. Em geral, têm-se que as obras de [Lee et al., 2016], [Susan and Bhutani, 2020] e [Albalooshi and Shatnawi, 2021] tem em comum a reprogramação de uma tabela de horários com base em padrões frequentes encontrados em um conjunto de dados. No caso de tabela de horários na área de transporte, há DF que dificulta a análise das regras, deviso a grande quantidade de RAs que são geradas.

Nenhum dos trabalhos está preocupado com o problema do elevado número de RAs devido a presença da DF. A consequência disto é que todas as RAs derivadas dessa mesma DF não enriquecem o conhecimento e não precisariam ser analisadas. Essas RAs são consideradas redundantes.

E este é um dos objetivos do método T-mine, apresentar uma quantidade reduzida de RAs não redundantes. Os trabalhos de [Zaki, 2000] e [Zaki, 2004] voltados a reduzir RAs desconsideraram DFs. Isso limita muito o potencial de redução das RAs, uma vez que as DFs permitem descartar inúmeras RAs redundantes.

Uma DF em uma tabela de horários gera uma regra para cada atributo que faz parte da DF, que deriva uma regra para cada valor de atributo distinto que caracteriza esta DF. Portanto, há uma oportunidade de explorar e analisar padrões em tabelas de horários com DFs intrínsecas. Também ficou claro: (i) não foram encontrados estudos que relatassem problemas com as DFs em tabelas de horários; e (ii) não há estudos que enfoquem a análise de padrões frequentes em tabelas de horários caracterizadas pela presença da DF.

Em geral, tem-se uma análise comparativa dos algoritmos que buscam DF nos conjuntos de dados, conforme visto em [Papenbrock et al., 2015]. Considerando o contexto de MPF e análise de redundância, os trabalhos de [Zaki, 2000] e [Zaki, 2004] tratam da remoção de regras redundantes pelo uso de conjuntos de itens frequentes fechados como método na geração de RAs. Por fim, pode-se afirmar por meio desta revisão sistemática, que o método T-mine proposto neste trabalho é inovador, pois em nenhum dos trabalhos descritos nesta seção, descreve-se como produzir um número reduzido de RAs não redundantes e estatisticamente significantes, a partir de tabelas de horários associadas a OD. Por fim, apresentamos na Figura III.2, um quadro comparativo entre os trabalhos relacionados e o método T-mine para facilitar a compreensão e o posicionamento do método perante os outros trabalhos.

Análise de Tabelas de Horários de Transporte

Autores	Ano	Temática	Principais diferenças ou similaridades com o método T-mine
Jappinen *et al.*	2013	Rotas abertas	Pode-se realizar esta análise.
Carrel *et al.*	2015	Análise de tempos de viagens	É possível realizar esta análise.
Lee *et al.*	2016	Reescalonamento de tabela de horários	As RAs geradas pelo T-mine apoiam o reescalonamento da tabela de horários.
Belcastro *et al.*	2016	Informações de voos com condições meteorológicas	Permite análise de vários atributos combinados.
Cristóbal *et al.*	2018	Análise de tempos de viagens	É possível realizar esta análise.
Carvalho et. al.	2020	Atrasos em voos	Permite este tipo de análise.
Chung *et al.*	2020	Gestão e previsão da rede de transporte aéreo	Pode ser usado para melhorar os horários de uma rede de transporte aéreo.

MPF em Tabelas de Horários

Autores	Ano	Temática	Principais diferenças ou similaridades com o método T-mine
Rahim *et al.*	2013	Predição por meio de *FP-tree*	Método T-mine também pode ser usado para predição.
Soni *et al.*	2015	Abordagem com uma única varredura	Duas varreduras: conjuntos candidatos e regras.
Susan and Bhutani	2020	Reescalonamento de tabela de horários	As RAs geradas pelo T-mine apoiam o reescalonamento da tabela de horários.
Albalooshi and Shatnawi	2020	Reescalonamento de tabela de horários	As RAs geradas pelo T-mine apoiam o reescalonamento da tabela de horários.

MPF e DF

Autores	Ano	Temática	Principais diferenças ou similaridades com o método T-mine
Naidenova	2013	Regras Lógicas	AS RAs geradas pelo T-mine são um tipo de regras lógicas.
Baixeries *et al.*	2014	*Formal Context Analysis e padrões frequentes*	Padrões com base nos conjuntos frequentes de itens.
Papenbrock *et al.*	2015	Comparação algoritmos que buscam DF	Remove DF. Pode ser generalizado para descobrir DF.

MPF e Análise de Redundância

Autores	Ano	Temática	Principais diferenças ou similaridades com o método T-mine
Zaki	2000	Conjuntos frequentes de itens fechados	Remoção de redundância com base na proposição de Bayardo, 2000
Zaki	2004	Conjuntos frequentes de itens fechados	Remoção de redundância com base na proposição de Bayardo, 2000
Xu and Li	2007	Regras Concisas	Mínimo de itens no consequente
Cheng *et al.*	2008	Regras Concisas	Mínimo de itens no consequente
Jeyapal and Ganesan	2016	Regras não redundantes	Remoção de redundância com base na proposição de Bayardo, 2000

Figura III.2: Quadro comparativo dos trabalhos relacionados

Capítulo IV Metodologia

Este capítulo descreve a metodologia empregada no presente trabalho. Conforme descrito na Seção II.1 do Capítulo III, ao longo do tempo surgiram diversas abordagens para MPF que visavam melhorar o custo computacional. Mas como se pode verificar em todas elas, nenhuma delas está associada a utilização de MPF em tabelas de horários, com o objetivo de encontrar um número reduzido de RAs não redundantes. A abordagem usada neste trabalho enquadra-se no contexto de geração das RAs e que permitiu a realização de análises de informações sobre OD em uma tabela de horários, sem que estas informações fossem redundantes.

IV.1 T-mine

A abordagem T-mine é projetada para descobrir RAs em tabelas de horários. O T-mine foi conduzido para gerar um número reduzido de RAs contextualizadas (com a presença mínima do atributo que indica a origem ou o destino, ou ambos, na formação das regras) e não redundantes. Para atingir esses objetivos, o atributo que causa a DF é removido do conjunto de dados. Então, este conjunto de dados é fragmentado horizontalmente de acordo com as localidades de origem e de destino. As localidades de origem e de destino são, de forma isolada, o predicado de seleção para a realização da fragmentação horizontal na origem e no destino. O algoritmo de padrão frequente é aplicado para gerar as RAs, em paralelo, considerando-se a premissa de ter os atributos de origem ou destino, ou ambos, no antecedente das RAs para preservar a DF, pois o atributo de origem ou destino, ou ambos, presente na regra, equivale ao atributo do número de voo que foi removido para não gerar uma enorme quantidade de regras redundantes e ou regras não contextualizadas. Finalmente, o algoritmo para remoção de redundância é aplicado, e as regras duplicadas são removidas. O resumo da metodologia pode ser visualizada na Figura IV.1.

Esta seção detalha as etapas para aplicar o método T-mine, conforme apresentado no Algoritmo 1. Por meio da chamada a função *t_mine* (linha 1), o método T-mine recebe uma tabela de horários de transporte por meio do parâmetro *dataset*, os valores limites estabelecidos pelo usuário, de suporte, por meio do parâmetro *supp* e de confiança, por meio do parâmetro *conf* e o parâmetro *attr* que especifica o atributo a ser removido e que causa a DF. Na sequência, a função *removeDF* é chamada (linha 2), recebendo o atributo identificado pelo parâmetro *attr* que causa a DF, e este

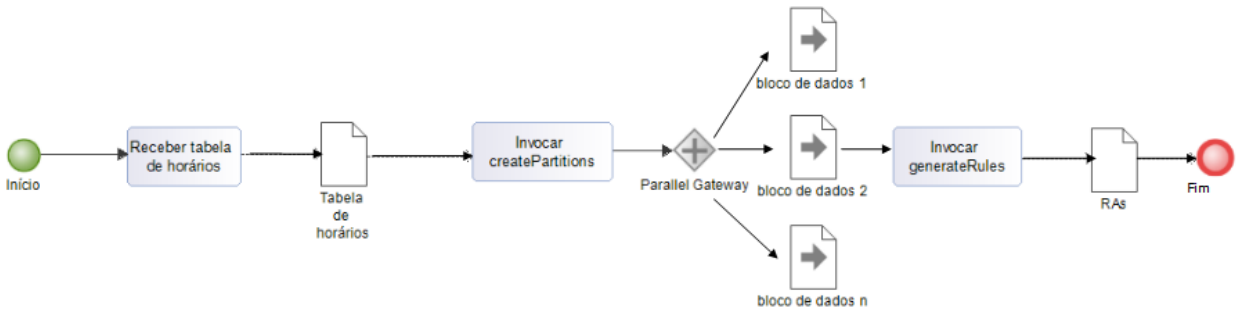


Figura IV.1: Processo de mineração usando o T-mine

atributo é removido do *dataset*.

A seguir, a função *createPartitions* é invocada (linha 3), recebendo como parâmetro, o *dataset* que representa a tabela de horários. Todo o *dataset* é fragmentado horizontalmente, tendo-se como predicado de seleção, todos os valores do atributo de origem encontrados no *dataset*. Todos os blocos de dados de origem criados p_{ori} são armazenadas em P. Então, todo o *dataset* também é fragmentado horizontalmente, tendo-se como predicado de seleção, todos os valores do atributo de destino encontrados no *dataset*. Todos os blocos de dados de destino criados p_{des} são armazenados em P. Neste caso, tem-se, então, uma fragmentação horizontal primária, pois é executada com o uso de predicados definidos. Em P encontram-se, então, todos os blocos de dados gerados pela fragmentação horizontal, tanto na origem quanto no destino. P então é retornado pela função. Assim, pode ocorrer que, um mesmo voo pode apresentado no bloco de dados relativo à origem e em outro bloco de dados relativo ao destino. Em outras palavras, tem-se uma fragmentação horizontal dupla, aplicada ao *dataset*. A expressão "dupla" se dá porque a fragmentação horizontal ocorre duas vezes, uma de acordo com a origem e a outra, de acordo com o destino.

Algorithm 1 T-mine Approach

1: **function** `t_mine`($dataset$, $supp$, $conf$, $attr$)
2: $dataset \leftarrow$ `removeDF`($attr$)
3: $P \leftarrow$ `createPartitions`($dataset$)
4: $rules \leftarrow$ `generateRules`($dataset$, P, $supp$, $conf$)
5: $rules \leftarrow$ `mergeDuplicates`($rules$)

1: **function** `createPartitions`($dataset$)
2: $P \leftarrow$ `unique`(p_{ori}) $\cup$ `unique`(p_{des})
3: return P

1: **function** `generateRules`($dataset$, P, $supp$, $conf$)
2: $rules \leftarrow \emptyset$
3: **do in parallel**
4: **for each** $p \in P$ **do**
5: $lsupp \leftarrow \frac{nrow(dataset)}{nrow(p)} \cdot supp$
6: $rules \leftarrow$ `FPM`($p, lsupp, conf$)
7: $rules \leftarrow$ `removeRedundancy`($rules$)
8: return $rules$

Para ilustrar o funcionamento da fragmentação horizontal, considere a Tabela IV.1. O método T-mine, por meio da chamada à função *createPartitions*, cria uma lista com blocos de dados todas as localidades (no caso, aeroportos) de origem e de destino, encontradas na tabela de horários. Neste exemplo brinquedo, uma lista P é formada pelos aeroportos de origem que são: SBRJ, SBSP e SBSV, e pelos aeroportos de destino que são: SBSP, SBRJ, SBGO e SBBR. Então, têm-se os blocos de dados p em P que podem ser observados nas Figuras IV.2 a IV.7. Neste caso, usou-se um predicado simples, que envolve apenas operadores relacionais. Por exemplo: $d.ori$=SBRJ. Pode se observar nas figuras dos blocos de dados do exemplo, que eles possuem diferentes seletividades. Por exemplo, a seletividade do bloco de dados, onde $d.ori$=SBRJ é igual a quatro, pois retornou-se quatro transações da Tabela usada como exemplo. No caso específico de OD, os blocos de dados não são disjuntos, pois existem transações em blocos de dados na origem, que podem aparecer em outros blocos de dados no destino. Na próxima etapa a função *generateRules* (linha 4) é executada. Ela recebe como parâmetros: o dataset com a tabela de horários *dataset*, a lista P contendo todos os blocos de dados p das origens e destinos, o valor do suporte global *supp* (calculado sobre todo o *dataset*) e o valor da confiança $conf$ especificado pelo usuário.

Tabela IV.1: Tabela de horários de voos

$d.flt$	$d.day$	$d.ori$	$a.des$	$d.hrs$	$a.hrs$	$a.atr$
1003	01	SBRJ	SBSP	10	11	1
1004	01	SBSP	SBRJ	07	08	1
1430	01	SBSP	SBGO	07	08	0
1370	01	SBSP	SBBR	14	16	0
1455	01	SBRJ	SBSP	12	13	0
1137	01	SBSV	SBSP	10	11	0
1003	02	SBRJ	SBSP	10	11	1
1004	02	SBSP	SBRJ	07	08	0
1430	02	SBSP	SBGO	07	08	1
1370	02	SBSP	SBBR	14	16	0
1455	02	SBRJ	SBSP	12	13	0
1137	02	SBSV	SBSP	10	11	1

d.flt	***d.day***	***d.ori***	***a.des***	***d.hrs***	***a.hrs***	***a.atr***
1003	01	SBRJ	SBSP	10	11	1
1455	01	SBRJ	SBSP	12	13	0
1003	02	SBRJ	SBSP	10	11	1
1455	02	SBRJ	SBSP	12	13	0

Figura IV.2: Bloco de dados para o predicado origem=SBRJ

Na função *generateRules* é inicializado o vetor *rules*, e este será retornado, com todas as RAs geradas, ao término da execução desta função. Os blocos de dados das origens e destinos são processados em paralelo por meio do laço *foreach*. O T-mine permite a geração paralela de RAs para os blocos de dados, respeitando-se o limite de processadores alocados. Isto é possível, graças

d.flt	d.day	d.ori	a.des	d.hrs	a.hrs	a.atr
1004	01	SBSP	SBRJ	07	08	1
1430	01	SBSP	SBGO	07	08	0
1370	01	SBSP	SBBR	14	16	0
1004	02	SBSP	SBRJ	07	08	0
1430	02	SBSP	SBGO	07	08	1
1370	02	SBSP	SBBR	14	16	0

Figura IV.3: Bloco de dados para o predicado origem=SBSP

d.flt	d.day	d.ori	a.des	d.hrs	a.hrs	a.atr
1137	01	SBSV	SBSP	10	11	0
1137	02	SBSV	SBSP	10	11	1

Figura IV.4: Bloco de dados para o predicado origem=SBSV

d.flt	d.day	d.ori	a.des	d.hrs	a.hrs	a.atr
1430	01	SBSP	SBGO	07	08	0
1430	02	SBSP	SBGO	07	08	1

Figura IV.5: Bloco de dados para o predicado destino=SBBR

d.flt	d.day	d.ori	a.des	d.hrs	a.hrs	a.atr
1430	01	SBSP	SBGO	07	08	0
1430	02	SBSP	SBGO	07	08	1

Figura IV.6: Bloco de dados para o predicado destino=SBGO

d.flt	d.day	d.ori	a.des	d.hrs	a.hrs	a.atr
1004	01	SBSP	SBRJ	07	08	1
1004	02	SBSP	SBRJ	07	08	0

Figura IV.7: Bloco de dados para o predicado destino=SBRJ

d.flt	d.day	d.ori	a.des	d.hrs	a.hrs	a.atr
1003	01	SBRJ	SBSP	10	11	1
1455	01	SBRJ	SBSP	12	13	0
1003	02	SBRJ	SBSP	10	11	1
1455	02	SBRJ	SBSP	12	13	0
1137	01	SBSV	SBSP	10	11	0
1137	02	SBSV	SBSP	10	11	1

Figura IV.8: Bloco de dados para o predicado destino=SBSP

ao pacote *doParallel*, que é um *backend* paralelo para o pacote *foreach*. Este *backend* fornece uma outra sessão de ambiente para executar laços *foreach* em paralelo. Isto pode ser observado no Algoritmo 1, linha 3 da função *generateRules*.

O objetivo é registrar um ambiente que permite o paralelismo dos trechos de código que serão

executados. Quando se registra o ambiente paralelo, define-se também a quantidade de processadores que serão utilizados, limitado ao número especificado pela arquitetura da máquina. Na sequência, o laço *foreach* é executado, otimizando-se a geração das RAs em paralelo. O pacote *foreach* deve ser usado em conjunto com o pacote *doParallel* para executar código em paralelo. O pacote *doParallel* atua como uma interface entre o pacote *foreach* e o pacote *parallel* da linguagem R.

O pacote *parallel* possui uma funcionalidade *multicore* que oferece suporte a vários processadores nos sistemas operacionais que suportam a chamada de sistema em modo *fork*. Este modo copia a sessão do R para que esta cópia seja o ambiente paralelo em que os códigos serão executados. A funcionalidade *multicore* discutida aqui, só executa em uma única máquina e não em um *cluster* de máquinas. De fato, o que garante a melhoria de velocidade, é a execução em uma máquina com vários processadores (neste trabalho, especificado em 15), ou vários núcleos ou ambos. Considerando-se o exemplo brinquedo apresentado, em paralelo, as RAs são geradas para cada um dos blocos de dados mostrados nas Figuras IV.2 a IV.7, e todas, em paralelo, no limite dos processadores alocados.

Para cada p em P (linha 4 da função *generateRules*), o suporte local $lsupp$ dos blocos de dados é calculado (linha 5), o valor do suporte local é proporcional a quantidade de transações que o bloco de dados conter, e as RAs são geradas em paralelo, para os blocos de dados, usando-se o algoritmo *Apriori* (linhas 6). O algoritmo *MPF* requer o bloco de dados p de P, o limite de suporte $lsupp$ calculado o respectivo bloco de dados, e o limite de confiança $conf$ especificado pelo usuário.

O método T-mine calcula o suporte local $lsupp$ que cada bloco de dados p em P deve conter para que as RAs descobertas também sejam frequentes em todo o *dataset*. Após gerar as RAs, as regras redundantes são eliminadas da seguinte forma: dadas duas RAs que se referem a mesma informação, sendo uma RA mais geral e outra mais específica, ou seja, a RA mais específica foi gerada por meio de um superconjunto de itens do conjunto de itens da regra mais geral, se ambas as regras possuem o mesmo valor de suporte e o valor da confiança da regra mais geral for maior ou igual que a confiança da regra mais específica, a regra mais específica não é apresentada, e ficamos somente com a regra mais geral, de acordo com a definição de remoção de redundância descrito na Seção II.3, do Capítulo II. A função *removeRedundancy* é chamada na linha 7 da função *generateRules*, a medida que as RAs são geradas. Ao término do processamento em paralelo, têm-se as RAs geradas para todos os blocos de dados. As RAs então são armazenadas no vetor $rules$ que é retornado.

Finalmente, as RAs duplicadas, e geradas por blocos de dados de origem e destino distintos, são mescladas na função *mergeDuplicates* (linha 5 da função *t_mine*). Esta duplicação ocorre, pois, como as regras são mineradas em blocos de dados de origem e de destino distintos, pode ocorrer de

termos uma mesma regra gerada nestes blocos de dados, como por exemplo, a regra $\{d.ori$=SBRJ, $d.hrs$=10, $a.des$=SBSP$\} \rightarrow \{a.atr$=1$\}$. Esta regra é gerada no bloco de dados cujo predicado de seleção foi $d.ori$=SBRJ e no bloco de dados cujo predicado de seleção foi $a.des$=SBSP.

Vislumbra-se que o método T-mine possa ser aplicado a qualquer tabela de horários. Para isto ocorrer, a premissa é que a tabela de horários possua um identificador único para cada entrada, caracterizando-se uma DF. Uma vez conhecendo-se a DF desta tabela de horários, o método T-mine poderá ser aplicado, pois, de acordo com o método, somente o atributo ou o conjunto de atributos que causa a DF é removido do *dataset*. O método mantém os atributos que preservam esta DF, nesta tabela de horários, para que seja possível a geração das RAs contextualizadas.

IV.2 *Brazilian Flights Dataset*

Antes de discorrermos sobre a implementação do T-mine e os experimentos realizados, faz-se necessário descrever o conjunto de dados usado para avaliação experimental. O conjunto de dados de *tabela de horários* adotado é o BFD ou Conjunto de Dados de Voos Brasileiros [Teixeira et al., 2020].

O BFD é um conjunto de dados integrado contendo dados de voos com partida e chegada em aeroportos brasileiros no período de 2000 a 2019. É composto por 15.505.922 registros de dados de voos, cada um contendo 45 atributos. Os atributos incluem dados de companhia aérea, voo, aeroportos, condições meteorológicas, horários programados e decorridos para embarque e desembarque. O metadados com a descrição de todos os atributos pode ser vista nas Figura IV.10. Vale observar que os atributos com identificador de 40 a 45 são atributos discretizados. E o diagrama de classes do BFD pode ser visto na Figura IV.9.

O BFD também faz parte das principais contribuições deste trabalho. É resultado de um processo de Extração, Transformação e Carga ou *Extract, Transform, and Load* (ETL), tendo como fontes de dados: o conjunto de dados de Voos Regulares Ativos (VRA) e o conjunto de dados meteorológicos oriundos do *Automated Surface Observing Systems* (ASOS).

O VRA é disponibilizado pela ANAC e contém dados de todos os voos brasileiros. Já o ASOS é um sistema administrado pela *IOWA University* (EUA). Ele contém dados meteorológicos coletados por sensores instalados em aeroportos de todo o mundo. Ressalta-se o BFD como uma das principais contribuições deste trabalho, devido as melhorias no processo de ETL a fim de: (i) Melhorar a qualidade dos dados (foi realizado um trabalho de curadoria do conjunto de dados); (ii) Tornar o BFD fácil de usar em qualquer pesquisa; e (iii) Contemplar o período de 2000 a 2019 em um único conjunto integrado de dados. Por meio da página de histórico [1] de voos da ANAC é possível consultar e obter dados do VRA no período de 1º de janeiro de 2000 a 31 de dezembro de 2019. O

[1] http://www.anac.gov.br/assuntos/dados-e-estatisticas/historico-de-voos

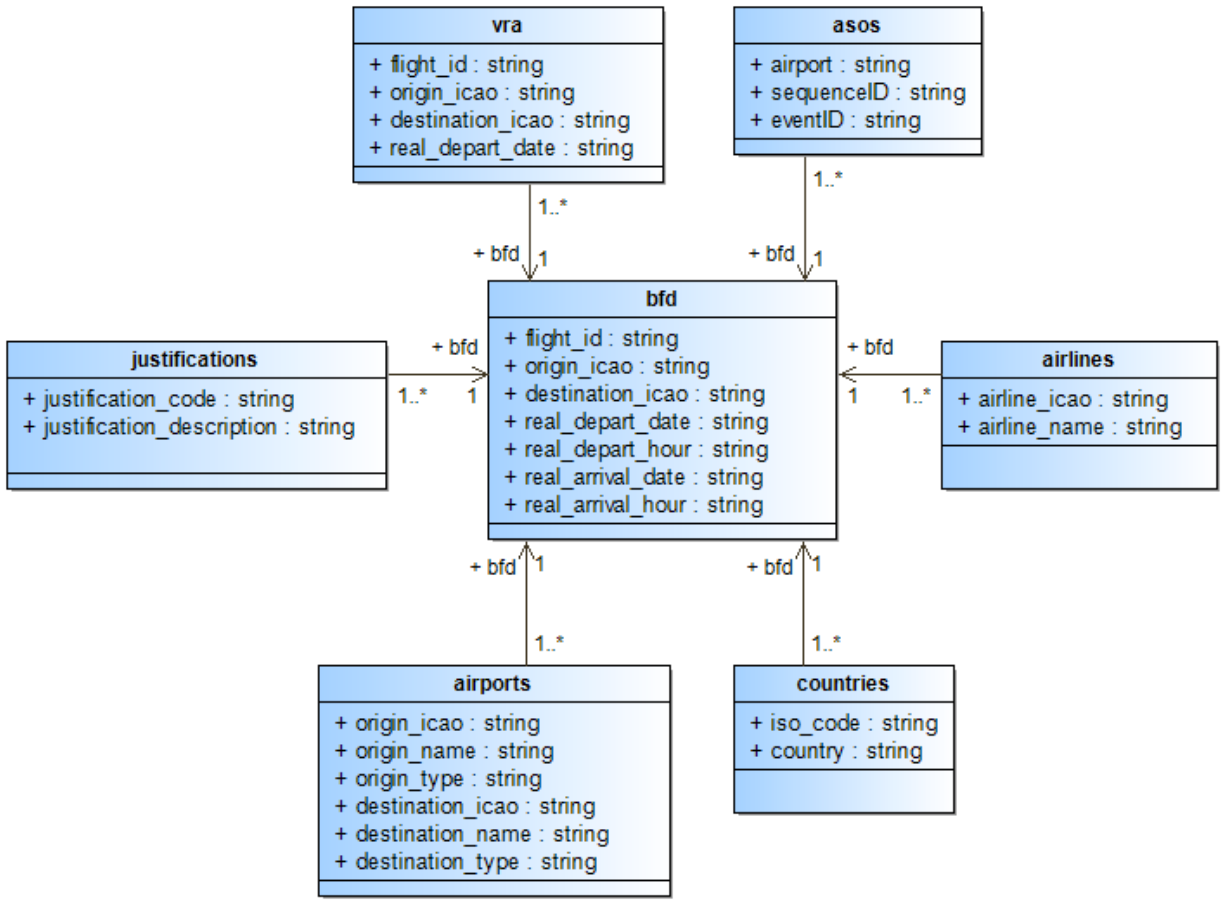


Figura IV.9: Diagrama de Classe do BFD

ASOS além de ser um sistema de informação e de apoio, é um programa que envolve várias agências governamentais americanas. Foi criado para formar uma rede oficial de informações meteorológicas para apoiar principalmente entidades de aviação. Inclui informações sobre pesquisas meteorológicas, climatológicas e hidrológicas.

O Departamento de Agronomia da *Iowa State University*, nos Estados Unidos, compila diariamente não apenas informações do sistema US ASOS, mas também de diversas entidades ligadas à aviação civil e militar de todo o planeta. Esta universidade disponibiliza gratuitamente esses dados para download [2] no seu sítio.

O processo de ETL pode ser acessado na página do *Data Analytics Lab* (DAL) [3], clicando-se no arquivo VRA_ASOS_Datasets_Integration_Report.ipynb, no sítio do *Github*. Os gráficos de análise exploratória sobre os dados do BFD ou *Exploratory Data Analysis* (EDA) podem ser acessados também na página do DAL [4] no sítio do *Github*. Em relação ao próprio BFD e sua documentação, estes estão disponíveis no sítio do *IEEE Data Port* [5].

[2] https://mesonet.agron.iastate.edu/request/download.phtml
[3] https://github.com/cefet-rj-dal/BFD
[4] https://github.com/cefet-rj-dal/BFD/blob/master/graphsDataPaper.ipynb
[5] http://dx.doi.org/10.21227/k10b-qn21

1. **flight_id** - ANAC flight identifier;
2. **airline_icao** - ICAO acronym of the airline. Ex: GLO, TAM, ONE;
3. **airline_name** - Full name of the airline;
4. **lynetype_code** - (N) National, (I) Intenational, (R) Regional, (H) Sub-regional, (E) Special, (C) Freight / Cargo, (G) International Freight / Cargo and (L) Postal Network;
5. **origin_icao** - ICAO code of the flight origin airport;
6. **origin_country** - ISO Country Name where the origin airport is located;
7. **origin_name** - Full name of the airport of origin of the flight;
8. **expected_depart_date** - Expected flight departure date in YYYYMMDD format where YYYY = year, MM = month and DD = day;
9. **expected_depart_hour** - Estimated flight departure hour in HH format where HH = full hour;
10. **real_depart_date** - Flight departure date in YYYYMMDD format where YYYY = year, MM = month and DD = day;
11. **real_depart_hour** - Flight departure in HH format where HH = full hour;
12. **expected_duration** - Difference in minutes between expected departure and arrival datetime;<br**>
13. **departure_delay** - Difference in minutes between expected and real departure datetime;
14. **depart_temperature** - Temperature in degrees Celsius at the airport of origin, at the time of flight departure;
15. **depart_dew_point** - Dew point, in degrees Celsius at the airport of origin, at the time of flight departure;
16. **depart_humidity** - Percentage of relative humidity in the airport of origin;
17. **depart_pressure** - Atmospheric pressure, in mbar, at the airport of origin;
18. **depart_visibility** - Visibility, in miles, at the airport of origin;
19. **depart_wind_speed** - Wind speed, in knots, at the airport of origin;
20. **depart_wind_direction** - Wind direction, based on Wind Rose, at the airport of origin;
21. **destination_icao** - ICAO code of the destination airport of the flight;
22. **destination_name** - Full name of the airport of destination of the flight;
23. **destination_country** - ISO country name where the destination airport is located;
24. **expected_arrival_date** - Estimated date and time of flight arrival in YYYYMMDD format where YYYY = year, MM = month and DD = day;
25. **expected_arrival_hour** - Estimated flight arrival hour in HH format where HH = full hour;
26. **real_arrival_date** - Flight arrival date in YYYYMMDD format where YYYY = year, MM = month and DD = day;
27. **real_arrival_hour** - Flight arrival hour in HH format where HH = full hour;
28. **real_duration** - Difference in minutes between real departure and arrival datetime;
29. **arrival_delay** - Difference in minutes between expected and real arrival datetime;
30. **arrival_temperature** - Temperature in degrees Celsius at the destination airport, at the time of flight arrival;
31. **arrival_dew_point** - Dew point, in degrees Celsius at the destination airport, at the time of flight arrival.
32. **arrival_humidity** - Percentage of relative humidity in the destination airport;
33. **arrival_pressure** - Atmospheric pressure, in mbar, at the destination airport;
34. **arrival_visibility** - Visibility, in miles, at the destination airport;
35. **arrival_wind_speed** - Wind speed in knots at the destination airport;
36. **arrival_wind_direction** - Wind direction, based on Wind Rose, at the airport of destination;
37. **situation_type** - Flight status;
38. **justification_code** - Flight delay or cancellation justification code;
39. **justification_description** - Description of justification for flight delay or cancellation;

Discretized fields (ds_):

40. **ds_depart_wind_speed** - The wind classification based on Belfourt Scale at the origin airport;
41. **ds_arrival_wind_speed** - The wind classification based on Belfourt Scale at the destination airport;
42. **ds_depart_wind_direction** - The acronymn for the wind direction at destination airport;
43. **ds_arrival_wind_direction** - The acronymn for the wind direction at destination airport;
44. **ds_depart_day_period** - Period of the day on real departure; and
45. **ds_arrival_day_period** - Period of the day on real arrival.

Figura IV.10: Metadados do BFD

Capítulo V Experimentos e Resultados

Este capítulo descreve os experimentos realizados neste trabalho e que permitem uma melhor compreensão sobre o método, mostrando as principais situações em que o emprego da proposta melhora os resultados obtidos pelo *baseline*. A partir dos resultados apresentados, também é possível verificar a aderência do método ao problema de pesquisa definido, apresentado e discutido diversas vezes ao longo deste trabalho, ou seja, a dificuldade em gerar uma menor quantidade de RAs não redundantes e com eficiência a partir de uma grande quantidade de dados. Com o objetivo de avaliar o T-mine na geração de RAs, a configuração experimental foi conduzida para medir o desempenho do T-mine contra as abordagens tradicionais.

Para efeito de comparação, conforme descrito no Capítulo III, uma abordagem tradicional para analisar *tabelas de horários* é aplicar diretamente a MPF. Esta abordagem foi denominada de abordagem baseline neste trabalho. Outra abordagem intuitiva é remover o atributo que causa a DF no conjunto de dados. Esta abordagem foi denominada sem DF, com origem do inglês *Without Functional Dependency* ou *wfd*. Considerando o BFD utilizado, o atributo de número de voo que caracteriza a DF foi removido para a abordagem *wfd*.

Para todas as abordagens (T-mine, baseline e *wfd*), o algoritmo de MPF usado é o *Apriori*. Com base nisto, também entende-se que o uso do mesmo algoritmo de MPF permite uma comparação justa entre as abordagens. Escolheu-se o *Apriori* por se tratar de um dos algoritmos mais utilizados na literatura para mineração de padrões frequentes. Após uma análise cuidadosa do seu método, representado aqui neste trabalho pelos pseudo-códigos descritos na Seção II.2 do Capítulo II vislumbrou-se que poderia existir um gargalo em relação a geração e teste de conjuntos candidatos. Adicionalmente, suspeitou-se que aplicação direta do *Apriori* em situações de padrões longos e limites mínimos de suporte, possam ocorrer dois tipos de custos computacionais não-triviais: (i) obtenção de um enorme número de conjuntos candidatos; e (ii) repetidamente varrer o conjunto de dados com esta enorme quantidade de conjuntos candidatos e checar a contagem de cada um deles com o limite mínimo do suporte de contagem e uma vez atendido, com a confiança estabelecida. Entende-se que todos estes problemas afetam o custo computacional como um todo e cria uma oportunidade de obter resultados significativos, mesmo usando o algoritmo *Apriori* sem modificação. A combinação das técnicas de particionamento horizontal, paralelismo, contextualização das RAs, remoção do atributo que causa a DF e remoção de redundância, abre espaço para futuramente,

incorporar estas técnicas, generalizando-se o algoritmo na busca de uma evolução do mesmo. O *Apriori* está presente no pacote *arules* disponível na linguagem R.

Para avaliar o comportamento do T-mine e das outras abordagens, decidiu-se realizar um teste de sensibilidade. Durante este teste, a confiança foi fixada em 95% e variou-se o valor de suporte 1.289×10^{-7} entre 1 a 6 vezes, e denominamos estes multiplicadores de fator de suporte. Adicionalmente, vislumbrou-se a realização de um outro passo, com a variação do valor da confiança entre 0,8 a 1, com incremento de 0,05. Em ambos os casos, o lado direito das regras foi fixado em $atraso = 1$, o que significa que foram analisados apenas voos atrasados.

Uma análise ortogonal é aplicar ou não um algoritmo de remoção de redundância. Em resumo, um experimento planejado é ortogonal se os efeitos de qualquer fator compensarem (sem remoção de redundância) entre os efeitos dos outros fatores (com remoção de redundância). A ortogonalidade garante que o efeito de um fator ou interação possa ser estimado separadamente a partir do efeito de qualquer outro fator ou interação no modelo. Nesta pesquisa, analisar o comportamento das abordagens ajuda no entendimento para que uma análise comparativa entre eles possa ser realizada de uma forma justa sob os mesmos parâmetros. Desta forma, todos os métodos (T-mine, baseline, e wfd) foram avaliados com e sem aplicação de algoritmo de remoção de redundância. Usamos o proposto por Bayardo Jr. et al. [2000] no algoritmo *Dense-Miner* descrito na Seção II.3 do Capítulo II. Está presente no pacote *arules* disponível em R.

V.1 Configuração

O T-mine e toda a avaliação experimental foram implementados na linguagem R. A avaliação foi executada em um computador i7 com 16 núcleos com 128GB de memória de acesso aleatório ou *Random Access Memory* (RAM) e sistema operacional Ubuntu 20.04. Os códigos em R e os dados para criação dos gráficos estão disponíveis no repositório do autor [1] na página do DAL no sítio do *Github*.

O algoritmo *Apriori* requer os parâmetros *supp* e *conf*. Adicionalmente, os parâmetros *minlen*, *maxlen*, *maxtime*, *target* e *rhs* foram usados durante a análise. Esses parâmetros influenciam a qualidade dos resultados e o tempo decorrido de cálculo. A descrição desses parâmetros e a faixa explorada estão resumidas na Tabela V.1. O valor de suporte foi calculado como sendo a menor quantidade de voos entre as localidades de origem e de destino, divido pela quantidade total de tuplas do conjunto de dados. A ideia aqui é restringir o mínimo possível a utilização de todas as transações presentes no conjunto de dados e, ao mesmo, realizar um teste de estresse para verificar se é exequível ou não, a geração de RAs com este valor de suporte. Porém, durante a execução de testes preliminares, constatou-se a inviabilidade de usar o valor de suporte encontrado, devido

[1] https://github.com/cefet-rj-dal/cteixeira/tree/master/t-mine

ao elevado consumo de memória. Logo, o valor de suporte foi corrigido para 1.289×10^{-7} que representa dois voos no mínimo por localidade de origem e de destino.

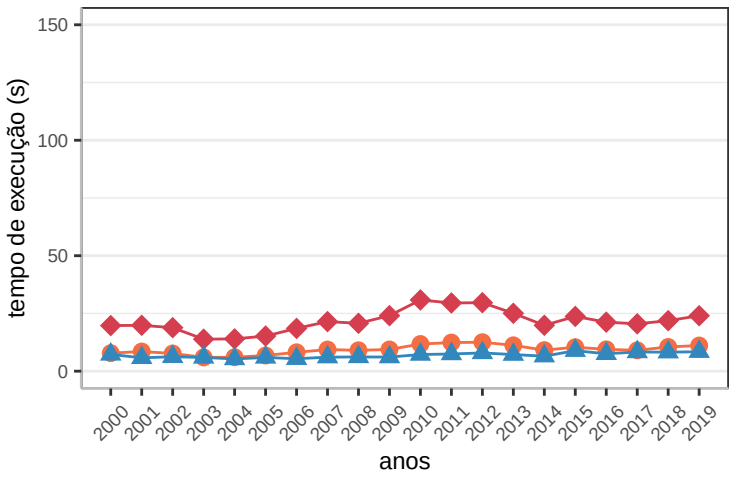
Tabela V.1: Parâmetros de entrada

Parâmetro	**Descrição (valores explorados)**
supp	Valor do suporte. O valor do suporte global foi 1.289×10^{-7}. Durante a análise de sensibilidade foi usado fatores de suporte equivalentes a 1x, 30x, 60x, 120x, 240x, e 480x respectivamente com base no valor do suporte global
conf	Valor da confiança. O valor básico foi 0,95. Durante a análise de sensibilidade variou-se de 0,8 a 1.
minlen	Número mínimo de itens nas RAs. O valor básico foi 2.
maxlen	Número máximo de itens nas RAs. O valor básico foi 10.
maxtime	Limite de tempo em segundos para análise dos subconjuntos. Foi definido como zero para desabilitar o limite de tempo.
rhs	Estabelece os itens que aparecerão no lado direito da regra. Fixado em $\{delay = 1\}$.

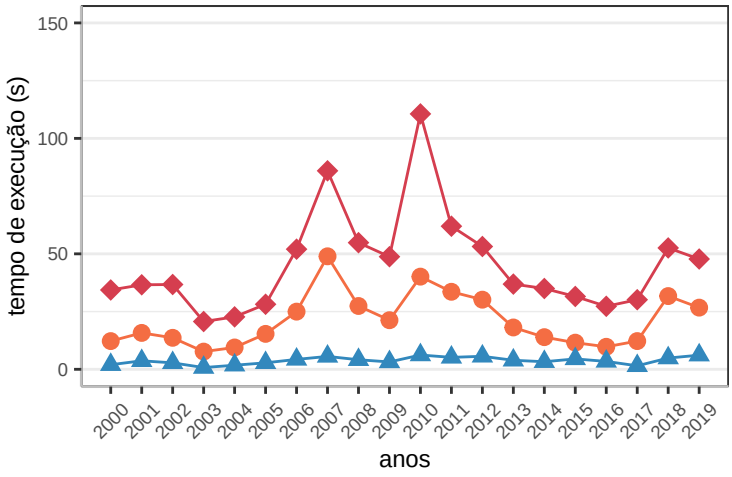
V.2 Experimentos computacionais

Para entender as vantagens de desempenho do método proposto, conduzimos uma série de experimentos computacionais, comparando o tempo de execução e o consumo de memória dos três métodos principais: T-mine, baseline e wfd. Para cada ano do BFD, examinamos o tempo decorrido da MPF antes e durante a remoção da redundância, bem como, o tempo total decorrido. As Figuras V.1a, V.1b, e V.1c mostram o tempo de execução comparativo para cada método em execução ao longo dos 20 anos de BFD com os seguintes cenários: (a) antes da remoção da redundância, (b) na remoção da redundância, e (c) tempo total de execução. O T-mine supera os métodos baseline e wfd com uma dispersão comprimida. Além disso, deve-se notar que o T-mine diminui significativamente o seu tempo de execução em relação a remoção da redundância.

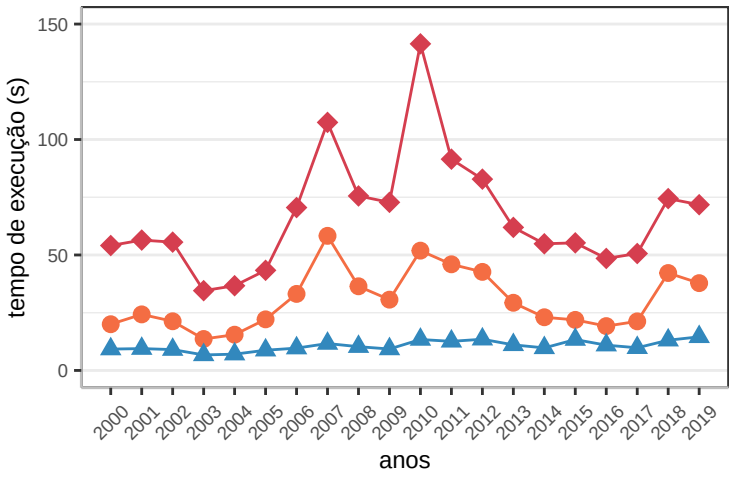
Uma das razões do T-mine ser mais rápido que outros métodos é a habilidade de permitir a execução paralela do laço *foreach* (apresentado no Capítulo IV). Outro resultado importante está relacionado a uma perda significativa de desempenho de *baseline* e wfd em algum ano específico (como 2007 e 2010) devido às características intrínsecas dos dados, confirmando a baixa oscilação de tempo do T-mine. Isto ocorre, pois, analisando-se estatisticamente os dados (vide a Tabela V.2 e a Figura V.2), pode-se observar que, as abordagens *baseline* e wfd não desempenharam bem quando a quantidade de regras não redundantes geradas foi maior do que a estatística correspondente ao terceiro quartil. E isto se deve a maior quantidade total de regras geradas redundantes nestes anos,



(a)



(b)



(c)

Figura V.1: Comparação entre os tempos de execução das abordagens T-mine (▲), baseline (◆) and wfd (●). Tempo decorrido antes da remoção de redundância (a); Tempo decorrido durante a remoção de redundância (b); Total do tempo decorrido (c)

para os métodos *baseline* e wfd, o que demanda um maior tempo de execução para a remoção da redundância. Esses pontos fora da curva ou *outliers* ocorreram também nos anos de 2008, 2011 e 2012.

Tabela V.2: Quantidade de regras distintas e não redundantes por ano para cada uma das abordagens

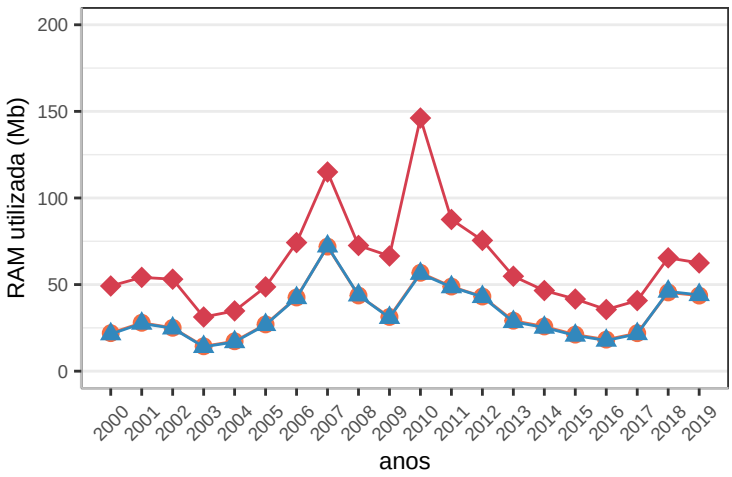
ano	baseline	wfd	T-mine
2000	91.053	47.989	54.514
2001	114.636	62.843	71.222
2002	102.649	55.047	62.932
2003	60.929	32.480	36.573
2004	75.624	40.596	44.695
2005	107.060	62.024	71.032
2006	160.303	95.315	106.932
2007	256.998	164.517	183.534
2008	164.658	104.806	116.211
2009	119.282	72.209	79.792
2010	201.553	120.648	132.355
2011	197.943	121.527	129.778
2012	169.400	107.364	116.211
2013	112.694	73.466	79.868
2014	91.454	63.101	68.906
2015	77.468	52.024	57.050
2016	64.349	46.236	50.162
2017	77.844	56.429	61.324
2018	144.115	119.231	129.322
2019	138.264	115.471	125.310

```
   baseline          wfd             tmine
Min.   : 60929   Min.   : 32480   Min.   : 36573
1st Qu.: 87751   1st Qu.: 54291   1st Qu.: 60256
Median :113665   Median : 67655   Median : 75507
Mean   :126414   Mean   : 80666   Mean   : 88886
3rd Qu.:161392   3rd Qu.:109391   3rd Qu.:118486
Max.   :256998   Max.   :164517   Max.   :183534
```

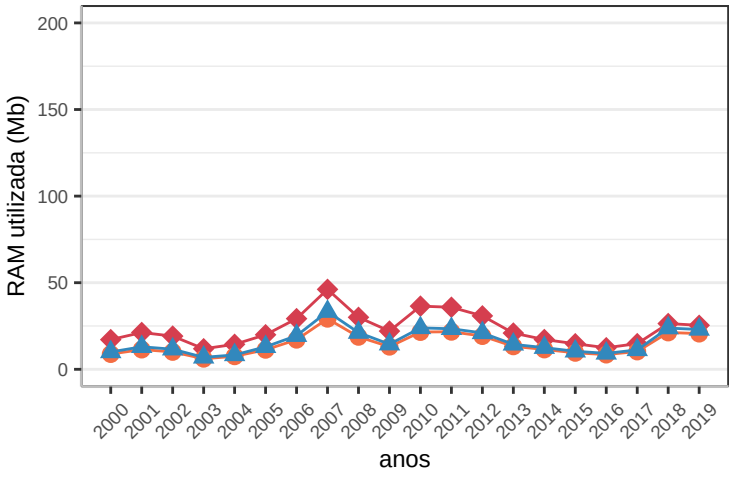
Figura V.2: Estatística das abordagens na geração de regras distintas e não redundantes

Na comparação do uso de memória, os resultados são mostrados nas Figuras V.3a, V.3b, e V.3c, considerando os seguintes cenários: (a) memória usada antes da remoção da redundância, (b) memória usada após a remoção da redundância e (c) memória total usada. É possível observar que o T-mine e o wfd usam menos RAM do que o baseline, e indica que a remoção da DF leva a uma redução na quantidade de padrões candidatos. Em termos de T-mine e wfd, eles são comparáveis no uso de memória.

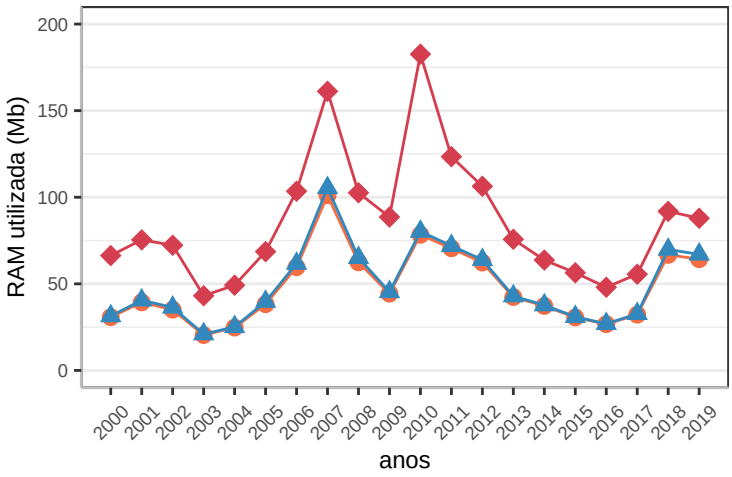
Em geral, todas as medições de tempo de execução e alocação de memória encontram-se no intervalo entre o limite inferior (-1,5IQR) e o limite superior (+1,5IQR), onde *Interquartil Range* (IQR) ou Amplitude Interquartil diz respeito aos valores que se encontram entre o primeiro quartil (Q1) e o terceiro quartil (Q3). Existe a presença de *outliers* ou pontos fora da curva, apenas em alguns anos específicos, conforme discutido acima. De qualquer forma, a grande maioria das



(a)



(b)



(c)

Figura V.3: Comparação entre alocação de memória das abordagens T-mine (▲), baseline (◆) and wfd (●). (a) RAM utilizada antes da remoção de redundância; (b) RAM utilizada após a remoção de redundância; and (c) Total de RAM utilizada.

medições concentram-se em torno da média. Adicionalmente, também é possível observar pelos gráficos das Figuras V.1 e V.3 que, os valores das medianas referente a tempo de execução e alocação de memória do T-mine é menor que as das outras abordagens. Por fim, destaca-se que o comportamento do T-mine é mais estável do que as outras abordagens, pois, a variação dos valores encontram-se no intervalo entre os quartis (Q1 e Q3). Em geral, a abordagem T-mine foi melhor em termos de tempo de execução e memória, o que responde à questão de pesquisa apresentada neste artigo.

V.3 Análise das RAs produzidas

O objetivo desta seção é avaliar as RAs produzidas pelas três abordagens em relação à redundância. Observa-se na Tabela V.3, que a quantidade de RAs não redundantes (*non-red*) (1.777.723) geradas pela abordagem T-mine é 42,21% menor que o número de RAs *non-red* (2.528.276) geradas pela abordagem baseline. Isso ocorre porque a abordagem baseline considera a presença da DF na formação das RAs. Isso gera um número muito grande de RAs que não agregam conhecimento.

Por outro lado, T-mine é 10,19% maior que o número de RAs *non-red* (1.613.323) gerados por wfd. Por que esse fenômeno ocorre? No intuito de explicar esse fato, vamos tomar como exemplo, voos, em dias distintos, com origem no aeroporto Londres-Heathrow (EGLL) e seus destinos, aeroportos São Paulo-Guarulhos (SBGR) e Rio de Janeiro-Galeão (SBGL), respectivamente, conforme mostra a Tabela V.4.

Observa-se que, usando wfd, essas RAs r1: $origin = \text{EGLL} \wedge dest = \text{SBGL} \wedge r_arriv_h = 08 \rightarrow delay = 1$ e r2: $origin = \text{EGLL} \wedge dest = \text{SBGR} \wedge r_arriv_h = 08 \rightarrow delay = 1$ são redundantes e são substituídos por r3: $origin = \text{EGLL} \wedge r_arriv_h = 08 \rightarrow delay = 1$, que é uma regra mais geral, mas perde a semântica OD e pode generalizar mais do que seria esperado para a análise de transporte. Porém, por meio da abordagem T-mine, as RAs r1 e r2 tornam-se não redundantes, pois a regra r1 foi gerada na partição de origem e a regra r2 foi gerada na partição de destino, e não há mais RAs gerais, considerando o par OD.

Os valores mostram que o número de RAs distintos e não redundantes do T-mine é menor do que as do baseline. Este resultado está alinhado com o objetivo de obter um número reduzido de RAs não redundantes e, ao mesmo tempo, permitir uma análise de origem e destino, preservando o contexto da DF. Por outro lado, T-mine é superior que o wfd. RAs consideradas redundantes produzidas pelo wfd, por meio da aplicação do T-mine, essas RAs redundantes tornam-se não redundantes, pois não existem outras RAs mais gerais do que estas, considerando-se o par OD.

Tabela V.3: Comparação das RAs geradas

baseline		wfd		T-mine	
total (k)	non-red (k)	total (k)	non-red (k)	total (k)	non-red (k)
7,176	2,528	3,868	1,613	3,809	1,778

Tabela V.4: Extrato das RAs geradas

origin	*dest*	*r_arriv_h*	*delay*
EGLL	SBGL	08	1
EGLL	SBGR	08	1
EGLL	SBGL	22	0
EGLL	SBGR	16	0
EGLL	SBGL	08	1
EGLL	SBGR	08	1
EGLL	SBGL	22	0
EGLL	SBGR	16	0

V.4 Análise de sensibilidade

Realizou-se uma análise de sensibilidade para quantificar o efeito dos métodos na geração das RAs não redundantes enquanto variamos o suporte e a confiança, e apresenta-se a faixa de valores explorados na Tabela V.1. Pode-se observar a sensibilidade do suporte na Figura V.4, onde o valor de confiança foi definido em 95%, e os valores usados para o suporte foram 1x, 30x, 60x, 120x, 240x e 480x respectivamente com base no valor do suporte global (1.289×10^{-7}). Na segunda análise (Figura V.5, usou-se o valor do suporte global (1.289×10^{-7}) como o valor do suporte, e a confiança variou de 80% a 100%. Enquanto os valores tornam o experimento mais restritivo (aumentando-se o suporte ou aumentando-se a confiança), o número de RAs distintas e não redundantes diminui para todas as abordagens.

V.5 RAs não redundantes: Comparação com base na frequência

Conforme já discutido anteriormente, as regras foram geradas com base nos limites mínimos de suporte e confiança (vide Capítulo IV). E com o intuito de garantir que estas regras sejam estatisticamente significantes, faz-se necessário uma análise estatística destas RAs geradas. As RAs são estatisticamente significantes, se cada uma delas estiver associada uma pequena chance de ocorrer ao acaso.

Isso significa que, ao coletar-se algumas amostras aleatórias de RAs geradas e algumas destas RAs estiverem contidas em diversas amostras, tem-se que estas RAs se repetem em diversas amostras aleatórias. E estas RAs que mais se repetem (frequência das RAs), são consideradas como sendo as RAs mais gerais e que, por isso, são RAs estatisticamente significantes.

Também com base na literatura, sabe-se que os métodos tradicionais de MPF que, prelimi-

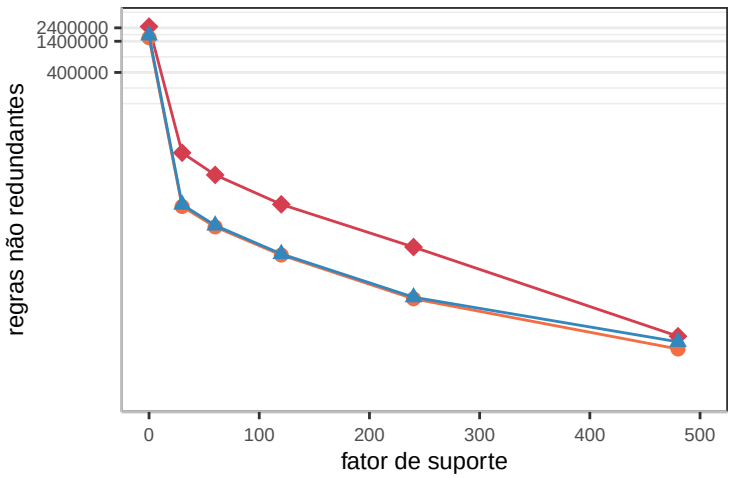


Figura V.4: RAs distintas e não redundantes das abordagens T-mine (▲), baseline (◆) and wfd (●) considerando 95% de confiança e fator de suporte equivalente a 1x, 30x, 60x, 120x, 240x, e 480x respectivamente com base no valor de suporte global

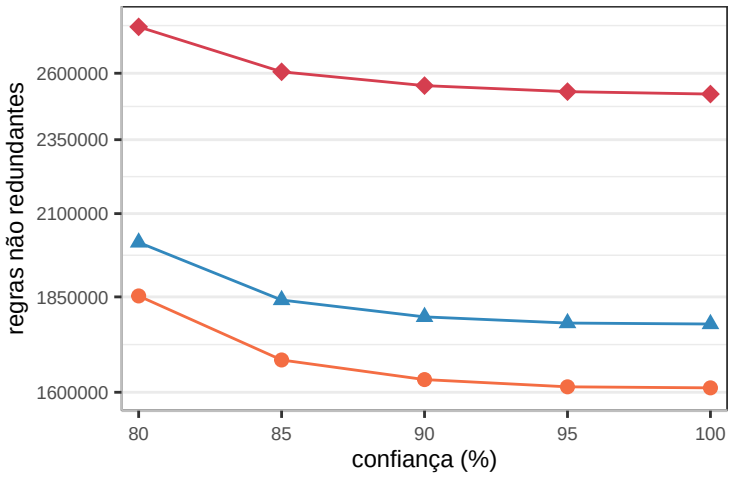


Figura V.5: RAs distintas e não redundantes das abordagens T-mine (▲), baseline (◆) and wfd (●) considerando 1.289×10^{-7} de suporte e 80%, 85%, 90%, 95%, e 100% como confiança

narmente, geram regras somente por meio das medidas de interesse de suporte e confiança não produzem regras estatisticamente significantes. Os métodos tradicionais podem gerar regras que não são estatisticamente significantes e rejeitar regras verdadeiras, regras que não contém erros do tipo 1 ou 2, conforme definição na Seção II.5 do Capítulo II.

Este mesmo problema ocorre também comumente em outras medidas de interesse e heurísticas de poda. Na prática, a significância estatística depende de duas coisas: frequência da regra, $P(X = \overline{x}; Y = \overline{y})$ e o grau da dependência $\frac{P(X=x;Y=y)}{P(X=x)P(Y=y)}$. Segundo Hamalainen [2014], quanto mais forte a dependência entre os conjuntos de itens do antecedente e do consequente na regra, menor será a frequência da regra, e vice-versa.

Este problema da dependência entre o antecedente e o consequente de uma regra é bem conhecido na literatura, e autores, como por exemplo, [Webb, 2007] não recomendam o modelo

[*supp*,*conf*] dos métodos tradicionais para a geração de RAs. Pois, quase todos os algoritmos de MPF utilizam a propriedade da antimonotonicidade do suporte: se o conjunto X não for frequente (dado algum suporte mínimo), então nenhum de seus os superconjuntos $Y \supset X$ podem ser frequentes. A partir desta propriedade, ocorre que se o suporte mínimo definido for muito alto, várias regras estatisticamente significantes podem ser perdidas. Por outro lado, se o suporte mínimo definido for muito baixo, um grande número de regras que não são estatisticamente significantes são geradas e o problema pode se tornar inviável computacionalmente. Vale destacar que, o nosso método ao aplicar as técnicas combinadas de remoção da DF, particionamento horizontal contextualizado em paralelo e com remoção de redundância, resolve estes dois problemas: (i) dependência entre o antecedente e o consequente; e (ii) geração de RAs que não sejam estatisticamente significantes, ou seja, gerando apenas regras distintas, não redundantes, sem a dependência discutida e estatisticamente significantes.

Isso posto, optou-se por realizar uma análise estatística dos dados, denominada por análise das RAs não redundantes: Comparação com base na frequência. Esta análise foi realizada por meio de uma simulação utilizando-se a técnica de re-amostragem ou *resampling*, inspirado no método *Bootstrap* (discutido na Seção II.5 do Capítulo II), seguido do cálculo da frequência das regras, ou seja, quantas vezes a regra aparece em um período de tempo associado a um intervalo de predição onde a regra está inserida.

Por meio da técnica de re-amostragem, amostras aleatórias (replicações) foram criadas a partir da população de regras distintas e não redundantes geradas por meio do T-mine. De uma forma mais geral, cada replicação inteira é independente de qualquer outra replicação e as observações de cada replicação têm a mesma distribuição. Essa independência entre as execuções das replicações é a chave da análise estatística dos dados, conforme preconizado por [Law, 2015].

Considerando a técnica de replicação, nossa análise proporciona uma comparação dos desvios quadráticos das distribuições pelas frequências das regras, tanto na população (conjunto de RAs distintas e não redundantes geradas) quanto na simulação. A técnica de replicação somente é possível graças ao TLC, conforme descrito na Seção II.5 do Capítulo II. Adicionalmente, nesta análise das RAs distintas e não redundantes sob o ponto de vista da frequência, utilizou-se o cálculo de escore ou pontuação padrão (ou *z-score*), onde o valor absoluto de z representa a distância entre esse escore e a média populacional em unidades de desvio padrão.

De acordo com Cumming [2006], o escore padrão pode ser usado no cálculo de intervalos de predição. E para [Geisser, 2019], dada uma amostra de uma distribuição normal, cujos parâmetros são desconhecidos, é possível calcular intervalos de predição no sentido frequencista, ou seja, um intervalo $[a, b]$ baseado na estatística da amostra, tal que em experimentos repetidos, $X_n + 1$ cai no intervalo com o percentual previsto no tempo; e pode-se chamar esses intervalos de "intervalos de

confiança preditivos"ou "intervalos de predição".

Um intervalo de predição em um nível *lambda* (λ) ou confiança, é um intervalo em que podemos calcular observando uma realização de X_1, ..., até X_n e outros que, com probabilidade λ, existem probabilidades iguais de ocorrência ou não de repetição da regra, seja na população (conjunto de RAs distintas e não redundantes geradas) ou na simulação.

Nesta análise, buscou-se comprovar como aceitável, a obtenção dos mesmos resultados quando a simulação é repetida. A simulação foi definida com (k=30) réplicas independentes ou amostras aleatórias com tamanho (s) definido pela divisão do número total de tuplas do conjunto de dados por k e foram descartadas observações com frequência $(fr) = 1$, regras transientes conforme será discutido mais abaixo.

A Figura V.6 mostra uma simulação com 30 amostras aleatórias. Considerando-se o TLC discutido anteriormente, pode-se observar que ambas as distribuições tendem a ser uma distribuição normal. Como as curvas mostradas são aproximadamente uma distribuição normal, então a probabilidade de uma regra se repetir ou não na população e essa mesma regra se repetir ou não na simulação é a mesma.

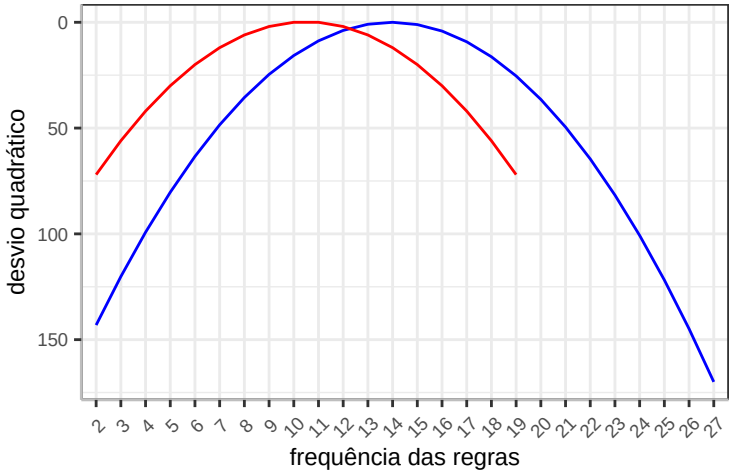


Figura V.6: Comparação das curvas de desvio quadrático sobre a frequência entre a população (—) e a simulação (—)

Assim, quando normalizamos a distribuição da frequência das regras, por exemplo, calculando-se os escores padrão, encontramos as probabilidades de ocorrência iguais para a repetição da regra, tanto na população quanto na simulação. Por exemplo, pode ser visto na Tabela V.5 que, a probabilidade de predição para as regras perenes (vide conceito mais abaixo) está na faixa entre 76,3% e 94,95% tanto na população quanto na simulação. Com base nesse resultado, essas regras são consideradas estatisticamente significantes e assim está provado a qualidade das regras geradas por meio de análise estatística de dados, saindo do conceito comum que é a utilização apenas das medidas de interesse.

Tabela V.5: Probabilidade de predição das regras perenes

$z-score$	freq. (fr)	média (μ)	$(fr\text{-}\mu)$	$(fr\text{-}\mu)/dp$	z-table
z1	14	10,50	3,50	0,66	0,7673
z2	19	10,50	8,50	1,59	0,9495

Uma vez provado que as regras geradas pelo nosso método são estatisticamente significantes e considerando o cenário dos sistemas em evolução, é possível classificar as RAs descobertas com base no número de vezes que ocorrem repetidamente ao longo do tempo em: (i) *Perenes*, RAs que permanecem ou existem por muito tempo ou aparentemente infinitas, durando, em períodos recorrentes e continuamente ou que se repetem ao longo do tempo; (ii) *Intermediárias*, RAs que permanecem ou existem por um período médio ou que se repetem muitas vezes; (iii) *Efêmeras*, RAs que permanecem, existem por pouco tempo ou que se repetem algumas vezes; e (iv) *Transientes*, RAs que ocorrem apenas uma vez.

Para avaliar o aspecto da geração da RAs não redundantes sob a perspectiva da frequência dessas RAs, foi realizada uma análise entre o T-mine e abordagens tradicionais quanto à frequência da RAs distintas entre 2000 e 2019. Considerando cada ano em nosso conjunto de dados como um lote para análise de mineração de padrões, é possível classificar as RAs descobertas pela frequência em que ocorrem repetidamente entre o período mencionado:

i *Perenes*, RAs com frequência $14 \leq \text{fr} \leq 19$;

ii *Intermediárias*, RAs com frequência $8 \leq \text{fr} \leq 13$;

iii *Efêmeras*, RAs com frequência $2 \leq \text{fr} \leq 7$; e

iv *Transientes*, RAs com frequência igual a 1.

A Figura V.7 compara o número de RAs produzidas para as três abordagens conforme a frequência das RAs aumenta. A frequência das regras geradas por meio do T-mine é menor do que a da abordagem do baseline, o que significa que as RAs não redundantes geradas pela abordagem baseline são mais frequentes do que aquelas geradas por meio do T-mine. No entanto, esta comparação entre T-mine com a abordagem baseline tem a seguinte ressalva: devido à presença da DF na abordagem baseline, a grande maioria das RAs distintas e não redundantes geradas pela abordagem baseline são RAs caracterizadas pelo número de voo.

O número do voo é um atributo que define aquele voo. Todas as RAs que contenham os atributos relacionados à caracterização do voo atual formam uma DF com o voo. Em seguida, ocorre a geração da RAs, não obstante, em suas formações, existe o atributo que identifica primariamente a DF (número do voo) juntamente com outros atributos distintos que não fazem parte de uma OD.

Por exemplo, pode-se mencionar as regras $\{d.flt{=}3820, d.hrs{=}10\} \rightarrow \{a.atr{=}1\}$ e $\{d.flt{=}3820, a.hrs{=}11\} \rightarrow \{a.atr{=}1\}$. Observa-se por este exemplo que, ambas as regras referem-se ao mesmo

voo e, em cada uma delas, na sua formação, não existem atributos que fazem parte de um par OD. Portanto, essas RAs são redundantes, pois se referem ao mesmo número de voo.

Por outro lado, a curva de frequência de T-mine é maior do que wfd denotando que as RAs do T-mine são mais frequentes do que aquelas geradas pelo wfd. Devido à grande quantidade de RAs transientes ou RAs com frequência igual a um, gerado por todos os métodos, estes não foram considerados aqui. A quantidade de RAs não redundantes produzidos pelo T-mine também é maior que as produzidas pelo wfd, no que diz respeito à geração da RAs não redundantes em todos as RAs classificadas, conforme pode ser visto na Tabela V.6

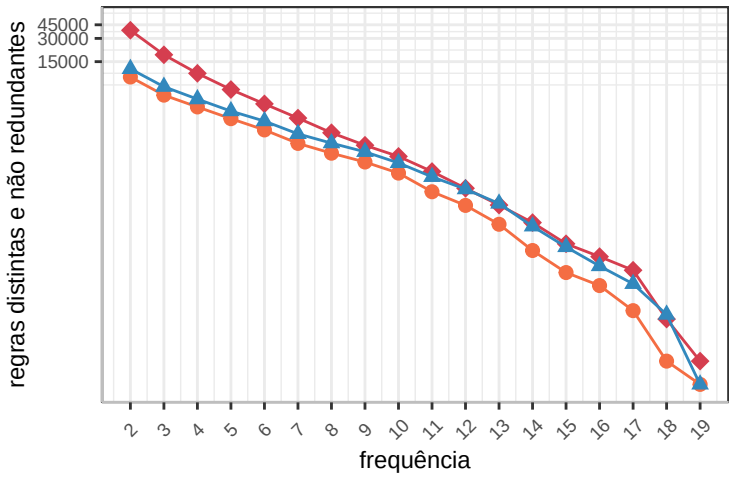


Figura V.7: Comparação entre as curvas de frequência das abordagens: T-mine (▲), baseline (◆) and wfd (●) . RAs distintas e não redundantes no eixo das ordenadas apresentadas na escala logarítmica e os resultados foram interpolados para uma melhor visualização.

Tabela V.6: RAs distintas e não redundantes

classe	faixa da freq.	baseline	wfd	T-mine
Perenes	$14 \leq fr \leq 19$	292	113	234
Intermediárias	$8 \leq fr \leq 13$	5.058	2.913	4.119
Efêmeras	$2 \leq fr \leq 7$	80.825	24.936	31.857
Transientes	fr *equal* 1	2.360.054	1.544.087	1.684.179

A tabela de frequência V.6 mostra que a abordagem baseline é superior ao T-mine para todas as classificações das RAs (coluna *classe*), e isso se deve à presença da DF na abordagem baseline e, por isso, gera uma grande quantidade de RAs que se referem às mesmas informações. A tabela V.6 também mostra que quanto maior a frequência (variável *fr* na coluna *faixa da freq.*) ou generalidade das RAs, maior será a porcentagem de superioridade ou aumento das RAs geradas usando o T-mine em comparação com a abordagem wfd. Pode-se observar que, com exceção das RAs transientes, a abordagem T-mine é maior na geração das RAs distintas e não redundantes em mais de 10% do que a abordagem wfd.

No caso das RAs perenes, a abordagem T-mine gerou 234 RAs, enquanto a abordagem wfd

gerou 113 RAs, ou seja, um aumento de 121 RAs ou 107,07%. Por outro lado, o resultado mostra 292 RAs perenes geradas na abordagem baseline e 234 na abordagem T-mine. Das 292 RAs perenes geradas pela abordagem baseline, 179 contém o atributo *flight_id*, o atributo primário que causa a DF. Analisando essas RAs, observa-se que em todas elas, além do atributo *flight_id*, possuem em sua formação o atributo *r_depart_h* ou *r_arriv_h*. Diante desse fato, analisamos a possibilidade das RAs que digam respeito às mesmas informações e, observou-se que, destas 179 RAs, 68 são RAs que se repetem, por possuírem o mesmo número de voo, variando apenas o horário do voo na partida ou chegada.

O algoritmo de remoção de redundância usado não leva em consideração a DF no contexto de remoção de redundância e isso pode ser visto nas RAs com *tuple_id* 21 e 22, 87 e 88, 107 e 108 na Tabela V.7. O foco da análise acima descrita foi em relação as RAs perenes, por serem estas as RAs mais interessantes devido à sua generalidade (RAs que se repetem mais ao longo do tempo), mas é importante destacar que os problemas mencionados também ocorrem na geração das outras RAs.

Além disso, uma regra com *tuple_id* 205 na Tabela V.7 não tem o atributo do número do voo, mas também não tem nem o atributo de origem nem de destino em sua formação, caracterizado por ser uma regra não contextualizada. Portanto, descartando as 69 RAs caracterizadas pelos problemas descritos acima, permanecem 223 RAs válidas. Um extrato das RAs perenes geradas pela abordagem baseline pode ser visto na Tabela V.7.

Comparando a quantidade de RAs perenes geradas pela abordagem T-mine (234) com as 223 RAs com semântica correta gerada pela abordagem baseline, a abordagem T-mine gera 11 RAs perenes a mais, sendo um pouco maior (4,93%) que a abordagem baseline e pode ser visto na Tabela V.8. Essas RAs na abordagem baseline são consideradas redundantes devido à presença da DF produzir RAs mais gerais com maior confiança, mas com o atributo *flight_id* em sua formação. Quando a DF é eliminada pela abordagem T-mine, isso não acontece e essas RAs são as mais gerais possuindo em sua formação um par OD.

RAs não contextualizadas também acontecem na abordagem wfd. Das 113 RAs perenes geradas, uma não é contextualizada e isso pode ser visto na (*tuple_id* 21) na Tabela V.9. Também é possível notar que esta regra é a mesma regra perene e não contextualizada que aparece na abordagem baseline (*tuple_id* 205 na Tabela V.7).

Este problema se repete em todos as outras classificações de RAs. A geração de RAs não contextualizadas é um problema geral nas abordagens tradicionais aplicadas a *tabelas de horários*. Isto porque é uma característica semântica de qualquer *tabela de horários* em que possa existir a previsão de transportes entre n origens e m destinos, derivando-se (n x m) transportes realizados. Com base nas duas explicações acima, temos para a regra não contextualizada: *r_depart_h*=08

Tabela V.7: Extrato das regras perenes geradas por meio da abordagem baseline

#	regra	freq.
1	*flight_id*=3910 ∧ *r_depart_h*=09	19
2	*origin*=LFPG ∧ *r_arriv_h*=09	19
3	*flight_id*=3245 ∧ *r_depart_h*=21	18
(...)		
10	*dest*=SBAT ∧ *r_arriv_h*=16	17
(...)		
21	*flight_id*=**3706** ∧ *r_arriv_h*=**15**	**17**
22	*flight_id*=**3706** ∧ *r_depart_h*=**13**	**17**
(...)		
58	*dest*=LIMC ∧ *r_arriv_h*=12	**16**
(...)		
87	*flight_id*=**3945** ∧ *r_arriv_h*=**19**	**16**
88	*flight_id*=**3945** ∧ *r_arriv_h*=**18**	**16**
(...)		
103	*dest*=SBMQ ∧ *r_arriv_h*=04	**15**
(...)		
107	*flight_id*=**3015** ∧ *r_arriv_h*=**19**	**15**
108	*flight_id*=**3015** ∧ *r_depart_h*=**18**	**15**
(...)		
205	*r_depart_h*=**08** ∧ *r_arriv_h*=**21**	**14**
(...)		

Tabela V.8: As 11 RAs perenes a mais geradas por meio do T-mine comparado com baseline

#	regra	freq.
1	*origin*=SBSP ∧ *dest*=SBBR ∧ *r_arriv_h*=00	18
2	*origin*=EGLL ∧ *dest*=SBGR ∧ *r_arriv_h*=08	18
3	*origin*=LFPG ∧ *dest*=SBGR ∧ *r_arriv_h*=09	18
4	*origin*=SBSP ∧ *dest*=SBGO ∧ *r_arriv_h*=00	18
5	*origin*=SBGR ∧ *dest*=EDDF ∧ *r_arriv_h*=12	17
6	*origin*=SBSP ∧ *r_depart_h*=08 ∧ *dest*=SBRJ ∧ *r_arriv_h*=10	17
7	*origin*=SBSP ∧ *r_depart_h*=10 ∧ *dest*=SBRJ ∧ *r_arriv_h*=12	17
8	*origin*=EGLL ∧ *r_depart_h*=21 ∧ *dest*=SBGR	17
9	*origin*=LFPG ∧ *r_depart_h*=21 ∧ *dest*=SBGR	17
10	*origin*=SBLO ∧ *dest*=SBSP ∧ *r_arriv_h*=22	17
11	*origin*=SBLO ∧ *r_depart_h*=21 ∧ *dest*=SBSP	17

∧ *r_arriv_h*=21 encontrada nas RAs perenes para as abordagens baseline e wfd que há vários voos saindo às 8h e chegando às 21h, tornando a análise em torno desse tipo de regra imprecisa e até inviável, portanto, esse tipo de regra não poderia trazer muitas informações.

Tabela V.9: Extrato das RAs perenes geradas por meio da abordagem wfd

#	regra	freq.
1	*origin*=LFPG ∧ *r_arriv_h*=09	19
2	*origin*=EGLL ∧ *r_arriv_h*=08	18
3	*r_depart_h*=20 ∧ *dest*=SBBE ∧ *r_arriv_h*=00	18
4	*dest*=SBAT ∧ *r_arriv_h*=16	17
(...)		
21	*r_depart_h*=**08** ∧ *r_arriv_h*=**21**	**14**
(...)		
113	*origin*=SBRB ∧ *r_arriv_h*=09	14

Capítulo VI Conclusões

O estudo na área da mineração de dados engloba diferentes conhecimentos dentro da própria ciência da computação, tornando-se uma área bastante completa e responsável por fornecer diversas ferramentas para a resolução dos mais variados problemas encontrados atualmente. Dentro desse contexto, encontra-se a MPF que vem sendo amplamente estudada na literatura devido a sua inúmera aplicações e variados problemas como agrupamento ou *clustering* e classificação. Além disso, a MPF é utilizada em diversos domínios como por exemplo, na análise de padrões em séries espaço-temporais. Por fim, a evolução dos métodos e os algoritmos que vêm sendo propostos com base nestes métodos, tornam a MPF uma técnica a ser explorada amplamente. Neste capítulo buscou-se resumir as principais contribuições desta pesquisa e oportunidades à serem exploradas por meio de apontamento para trabalhos futuros.

VI.1 Contribuições e Considerações finais

Conforme discutido na Seção IV.2 do Capítulo IV, antes mesmo do desenvolvimento da abordagem do T-mine, fez-se necessário o desenvolvimento do BFD.

As principais contribuições do BFD são: (i) Enriquecido com voos nacionais e internacionais que chegam e partem de aeroportos brasileiros. Isso permite realizar diversos estudos para auxiliar o processo de tomada de decisão; e (ii) Permite que companhias aéreas e governos façam perguntas sobre atrasos de voos para revisar seus protocolos e otimizar seus serviços. As medidas de interesse disponíveis nos métodos tradicionais e seguidas ou não de remoção de redundância após o procedimento de mineração não são suficientes para reduzir o número de RAs na presença de DF (característica presente em qualquer tabela de horários e ou para eliminar RAs redundantes, bem como remover RAs não contextualizadas sem aumentar o custo computacional.

Isso posto, ressalta-se a importância deste trabalho, por meio da proposição de um novo método, denominado T-mine. A principal contribuição do T-mine é **a redução do custo computacional, quando comparado às abordagens tradicionais, possui um menor de tempo de execução e com menor alocação de memória**.

Adicionalmente, mais quatro contribuições secundárias ou objetivos da contribuição principal foram alcançados por meio do T-mine são elas:

(i) **redução do espaço de busca das RAs na presença de DF em qualquer tabela de horários**;
(ii) **eliminação das RAs redundantes por meio de um merge das RAs que fornecem a mesma informação**;
(iii) **eliminação das RAs duplicadas em que a ordem dos atributos está invertida**; e
(iv) **remoção das RAs não contextualizas, RAs que não possuem na sua formação os atributos de um par OD**.

O T-mine tem dois pontos fortes: a eficiência computacional e a redução contextualizada. Isso é obtido por meio do particionamento horizontal e processamento paralelo. A estratégia de particionamento horizontal do T-mine reduz significativamente o custo de computação, melhorando o acesso à memória e o processamento paralelo. Ao carregar partições que são significativamente menores do que todo o conjunto de dados da tabela de horários, os algoritmos de MPF melhoram os movimentos de dados da memória. O particionamento horizontal também permite paralelismo de dados de forma independente.

O particionamento horizontal também contribui para o paralelismo de manipulação dos dados de forma independente. Neste último, a estratégia de dividir e conquistar, particionando os dados na origem e no destino, descartando as RAs não contextualizadas que não trouxeram valor durante a análise. Adicionalmente, a técnica de particionamento do T-mine permite a geração de RAs por localidade de origem e destino.

Isto facilita a remoção das RAs redundantes com todo o processamento em paralelo, reduzindo o tempo total de execução e o consumo de memória alocada. O T-mine é capaz de **gerar um maior número de RAs distintas, não redundantes e contextualizadas do que a abordagem tradicional sem DF**. Os algoritmos clássicos de remoção de redundância não levam em consideração a questão da DF. Dessa forma, eles produzem RAs não redundantes que, no entanto, referem-se às mesmas informações. Finalmente, essas RAs distintas e não redundantes geradas por meio do T-mine **são as que mais se repetem ao longo de todo o período, portanto, essas RAs são as mais gerais**, **possuem significância estatística** conforme discutido na Seção V.5 do Capítulo V e com a sua **semântica contextualizada, ou seja, sua formação contendo minimamente o atributo de origem ou destino**.

VI.2 Trabalhos Futuros

Uma questão interessante a ser abordada é o quanto o método T-mine é específico para tabelas de horários versus outras abordagens de DF. Foi discutido no Capítulo I que a DF nesta pesquisa foi identificada a partir do conhecimento prévio do domínio da aplicação. No caso específico, o conhecimento prévio do atributo que causa a DF em uma tabela de horários de voos, em um sistema

de transporte aéreo. Por outro lado, também foi visto no Capítulo II que existem algoritmos com foco na descoberta das DFs. Então, possíveis perguntas de pesquisa seriam: (i) Pode-se usar estes algoritmos para a descoberta das DFs e a partir daí, ser aplicado o método T-mine, considerando-se estas DFs descobertas? (ii) o quanto esta descoberta de RAs, de fato representam fielmente o conhecimento sobre o domínio a ser aplicado? Diante disso, sugere-se que um trabalho futuro mais próximo que os demais, encontra-se no entorno dessas questões que necessitam de uma análise mais detalhada.

Além disso, todos os resultados encontrados e contribuições destacadas na seção anterior, abrem caminho para um estudo aprofundado dos padrões mais frequentes entre aqueles que foram gerados, como, por exemplo, as RAs perenes que foram descobertas neste trabalho. Existem pesquisas promissoras para trabalhos futuros: primeiro, o uso combinado das técnicas de particionamento horizontal contextualizado paralelo, remoção da RAs redundantes e poda de RAs não contextualizadas, pode ser generalizada e implementada por meio de um algoritmo que permitirá a comparação sob os aspectos de custo computacional e qualidade das RAs com os algoritmos discutidos no Capítulo III. Em segundo lugar, a abordagem do T-mine pode ser útil para outros estudos usando a DF. Terceiro, uma tabela de horários como uma série de espaço-tempo, pode-se descobrir Padrões Emergentes (PEs) para analisar fenômenos não estacionários.

Também é visto o desenvolvimento de um modelo não paramétrico onde a análise é feita no domínio da frequência. Adiante, melhorias e possíveis novas técnicas de paralelização no T-mine devem ser consideradas. Além disso, o teste em conjuntos de dados de tabelas de horários de diferentes transportes pode levar a intuições que não foram consideradas devido a algum teste específico sobre a natureza dos dados. Em geral, o T-mine pode ser usado para melhorar a estimativa de qualquer tabela de horários de uma matriz OD usada para o planejamento da demanda de transporte.

Referências Bibliográficas

Avishai Ceder. *Public Transit Planning and Operation: Modeling, Practice and Behavior*. CRC Press, March 2016. ISBN 978-1-4665-6392-6. 1

W.-H. Lee, L.-H. Yen, and C.-M. Chou. A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services. *Transportation Research Part C: Emerging Technologies*, 73:49–64, 2016. 1, 18, 19, 24

R. Ceha and H. Ohta. Prediction of future origin destination matrix of air passengers by fratar and gravity models. *Computers and Industrial Engineering*, 33(3-4):845–848, 1997. 1

Charu C. Aggarwal and Jiawei Han. *Frequent Pattern Mining*. Springer, August 2014. ISBN 978-3-319-07821-2. 1, 2, 8

Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems*. Pearson Education Limited, September 2016. ISBN 978-1-292-09762-6. 4, 13

Michael Hahsler, Christian Buchta, Bettina Gruen, and Kurt Hornik. arules: Mining Association Rules and Frequent Itemsets. Technical report, https://cran.r-project.org/package=arules, 2020. 4

P-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar. *Introduction to Data Mining*. Pearson, 2019. ISBN 9780133128901,. 6, 11

Jiawei Han, Jian Pei, and Micheline Kamber. *Data Mining: Concepts and Techniques*. Elsevier, June 2011. ISBN 978-0-12-381480-7. 8

J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 29(2):1–12, 2000. 9

P. Tan, M. Steinbacm, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., 2005. ISBN 0321321367. 10

Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB

'94, pages 487–499, San Francisco, CA, USA, September 1994. Morgan Kaufmann Publishers Inc. ISBN 978-1-55860-153-6. 11

R.J. Bayardo Jr., R. Agrawal, and D. Gunopulos. Constraint-based rule mining in large, dense databases. *Data Mining and Knowledge Discovery*, 4(2-3):217–240, 2000. 12, 13, 21, 35

Y. Huhtala, J. Kärkkäinen, P. Porkka, and H. Toivonen. TANE: An efficient algorithm for discovering functional and approximate dependencies. *Computer Journal*, 42(2):100–111, 1999. 13

Meeta Pajwani and Mansi Bosamia. Overview of horizontal partitioning and vertical partitioning. 04 2013. 14

M. Tamer Özsu and Patrick Valduriez. *Principles of Distributed Database Systems*. Springer Nature, December 2019. ISBN 978-3-030-26253-2. 14

S. Lallich, O. Teytaud, and E. Prudhomme. Association rule interestingness: Measure and statistical validation. *Studies in Computational Intelligence (SCI)*, 43:251–275, 2007. 15

S. Morishita and J. Sese. Transversing itemset lattices with statistical metric pruning. *PODS '00: Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, page 226–236, 05 2000. 15

Geoff Cumming. Understanding replication: Confidence intervals, p-values, and what's likely to happen next time. *Cumming (Refereed)*, 1, 01 2006. 15, 43

Mohammad Islaqm. Sample size and its role in central limit theorem (clt). *Computational and Applied Mathematics Journal*, 4:1–7, 02 2018. 15

J Behboodian and Akbar Asgharzadeh. On the distribution of z-scores. *Iranian Journal of Science and Technology*, 32, 12 2008. 16

L. Carvalho, A. Sternberg, L. Maia Gonçalves, A. Beatriz Cruz, J.A. Soares, D. Brandão, D. Carvalho, and E. Ogasawara. On the relevance of data science for flight delay research: a systematic review. *Transport Reviews*, 2020. 18, 19

S.-H. Chung, H.-L. Ma, M. Hansen, and T.-M. Choi. Data science and analytics in aviation. *Transportation Research Part E: Logistics and Transportation Review*, 134, 2020. 18, 19

L. Belcastro, F. Marozzo, D. Talia, and P. Trunfio. Using scalable data mining for predicting flight delays. *ACM Transactions on Intelligent Systems and Technology*, 8(1), 2016. 18, 19

T. Cristóbal, G. Padrón, A. Quesada-Arencibia, F. Alayón, and C.R. García. Systematic Approach to Analyze Travel Time in Road-Based Mass Transit Systems Based on Data Mining. *IEEE Access*, 6:32861–32873, 2018. 18, 19

A. Carrel, P.S.C. Lau, R.G. Mishalani, R. Sengupta, and J.L. Walker. Quantifying transit travel experiences from the users' perspective with high-resolution smartphone and vehicle location data: Methodologies, validation, and example analyses. *Transportation Research Part C: Emerging Technologies*, 58:224–239, 2015. 18, 19

S. Jäppinen, T. Toivonen, and M. Salonen. Modelling the potential effect of shared bicycles on public transport travel times in Greater Helsinki: An open data approach. *Applied Geography*, 43:13–24, 2013. 18, 19

F.A. Albalooshi and S.M. Shatnawi. Timetable generation: Applying a modified FP-tree algorithm on mined students' and faculty preferences. *International Journal of Applied Metaheuristic Computing*, 12(1):20–40, 2021. 20, 21, 24

N.F.A. Rahim, S.M. Taib, and S. Mahamad. Courses enrollment pattern analysis. *Lecture Notes in Electrical Engineering*, 152 LNEE:283–292, 2013. 20, 21

S. Susan and A. Bhutani. Data Mining with Association Rules for Scheduling Open Elective Courses Using Optimization Algorithms. *Advances in Intelligent Systems and Computing*, 941:770–778, 2020. 20, 21, 24

K.K. Soni, J. Agrawal, S. Sharma, and S. Agrawal. Association rule mining based on density and regional minimum support. In *Proceedings - 2015 5th International Conference on Communication Systems and Network Technologies, CSNT 2015*, pages 968–975, 2015. 20, 21

A.M. Jorge, J. Mendes-Moreira, J.F. De Sousa, C. Soares, and P.J. Azevedo. Finding interesting contexts for explaining deviations in bus trip duration using distribution rules. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7619 LNCS:139–149, 2012. 20

J. Baixeries, M. Kaytoue, and A. Napoli. Characterizing functional dependencies in formal concept analysis with pattern structures. *Annals of Mathematics and Artificial Intelligence*, 72(1-2): 129–149, 2014. 21, 22

T. Papenbrock, J. Ehrlich, J. Marten, T. Neubert, J.-P. Rudolph, M. Schönberg, J. Zwiener, and F. Naumann. Functional dependency discovery: An experimental evaluation of seven algorithms. *Proceedings of the VLDB Endowment*, 8:1082–1093, 2015. 22, 25

Xenia Naidenova. An Analytical Survey of Current Approaches to Mining Logical Rules from Data. In *Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems*, pages 71–101. IGI Global, 2013. ISBN 978-1-4666-1901-2. 22

J. Cheng, Y. Ke, and W. Ng. Effective elimination of redundant association rules. *Data Mining and Knowledge Discovery*, 16(2):221–249, 2008. 23, 24

Y. Xu and Y. Li. Generating concise association rules. In *International Conference on Information and Knowledge Management, Proceedings*, pages 781–789, 2007. 23, 24

M.J. Zaki. Generating non-redundant association rules. In *Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 34–43, 2000. 23, 24, 25

M.J. Zaki. Mining non-redundant association rules. *Data Mining and Knowledge Discovery*, 9(3): 223–248, 2004. 23, 24, 25

A. Jeyapal and J. Ganesan. Elimination of redundant association rules—an efficient linear approach. *Advances in Intelligent Systems and Computing*, 412:171–180, 2016. 23, 24

M.J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel algorithms for discovery of association rules. *Data Mining and Knowledge Discovery*, 1(4):343–373, 1997. 24

Claudio Teixeira, Lucas Teixeira, Joel dos Santos, Glauco Amorim, Jorge Soares, and Eduardo Ogasawara. Integrated Brazilian Flight Datasets Description. Technical report, http://dx.doi.org/10.21227/k10b-qn21, 2020. 31

Wilhelmiina Hamalainen. Assessing the statistical significance of association rules. *ArXiv*, 05 2014. 42

Geoffrey Webb. Discovering significant patterns. *Machine Learning*, 71, 07 2007. 42

A. M. Law. Statistical analysis of simulation output data: The practical state of the art. In *2015 Winter Simulation Conference (WSC)*, pages 1810–1819, 2015. doi: 10.1109/WSC.2015.7408297. 43

S. Geisser. *Predictive Inference*. Chapman and Hall/CRC, 2019. ISBN 9780367449919. 43