Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Programa de Pós-Graduação em Ciência da Computação

Avaliação Empírica de Aprendizagem Incremental de Estruturas de Redes Bayesianas

Luiz Antonio Pereira Silva

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Inteligência Artificial

Kyller Costa Gorgônio
(Orientador)

Campina Grande, Paraíba, Brasil
©Luiz Antonio Pereira Silva, 28/02/2019

“AVALIAÇÃO EMPÍRICA DE APRENDIZAGEM INCREMENTAL DE ESTRUTURAS DE REDES BAYESIANAS”

LUIZ ANTONIO PEREIRA SILVA

DISSERTAÇÃO APROVADA EM 28/02/2019

KYLLER COSTA GORGÔNIO, Dr., UFCG
Orientador(a)

DANILO FREIRE DE SOUZA SANTOS, Dr., UFCG
Examinador(a)

MIRKO BARBOSA PERKUSICH, Dr., IFPB
Examinador(a)

CAMPINA GRANDE - PB

S586a Silva, Luiz Antonio Pereira.
Avaliação empírica de aprendizagem incremental de estruturas de redes bayesianas / Luiz Antonio Pereira Silva. – Campina Grande, 2019.
156 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2019.
"Orientação: Prof. Dr. Kyller Costa Gorgônio ".
Referências.

1. Redes bayesianas. 2. Aprendizagem incremental. 3. Adaptação estrutural. 4. Refinamento estrutural. I. Gorgônio, Kyller Costa. II. Título.

CDU 004.89:621.39(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA MARIA ANTONIA DE SOUSA CRB 15/398

Resumo

Redes Bayesianas podem ser construídas baseadas no conhecimento do especialista, nos dados históricos, ou em ambos. No entanto, alterações no domínio de aplicação, imprecisões ou alta complexidade nas informações coletadas podem resultar em produções de redes Bayesianas com baixa usabilidade e/ou baixa precisão. Diante deste problema, é essencial melhorar o modelo gerado à medida que novos conhecimentos são coletados, incorporando, continuamente, o conhecimento novo ao existente. Neste trabalho, dois estudos são realizados a partir de duas perspectivas diferentes com o objetivo de avaliar e melhor compreender o uso de algoritmos de aprendizagem incremental de estruturas de redes Bayesianas em contextos diversos de uso. No primeiro estudo, uma revisão sistemática da literatura é realizada com o intuito de identificar e avaliar soluções para o aprendizado incremental de estruturas de redes Bayesianas, bem como para delinear direções de novas pesquisas relacionadas. No segundo estudo, duas das soluções encontradas são avaliadas, experimentalmente, utilizando dados reais e sintéticos com o objetivo de testá-las em contextos diferentes e comparar suas performances quanto à qualidade da rede aprendida. Na revisão sistemática, grande parte dos estudos relevantes existentes na literatura são reunidos e é identificado que os procedimentos de aprendizagem destas soluções podem ser classificados como refinamento ou adaptação, em que a principal diferença entre eles está em como utilizam o novo conhecimento adquirido. É possível identificar com a avaliação empírica que as soluções incrementais analisadas produzem resultados com pontuação idêntica aos geradas por soluções de aprendizado em lote, mas diferem na generalização de novos dados. Nota-se também que características do contexto e fatores de restrição aplicados pelos algoritmos interferem na qualidade de generalização das redes. De modo geral, é concluído que os algoritmos de aprendizagem incremental de estruturas de redes Bayesianas analisados podem ser considerados uma solução aceitável em contextos restritos de uso.

Abstract

Bayesian networks can be constructed based on expert knowledge, historical data, or both. However, changes in the application domain, inaccuracies or high complexity in the collected information can result in Bayesian networks productions with low usability and/or precision. Faced with this problem, it is essential to improve the generated model as new knowledge is collected, continuously incorporating new knowledge to the existing one. In this work, two studies are carried out from two different perspectives in order to evaluate and better understand the use of incremental learning algorithms of Bayesian network structures in different contexts of use. In the first study, a systematic literature review is carried out in order to identify and evaluate solutions for the incremental learning of Bayesian network structures, as well as to delineate directions of new related research. In the second study, two of the solutions found are experimentally evaluated using real and synthetic data in order to test them in different contexts and compare their performances regarding the quality of the network learned. In the systematic review, most of the relevant literature studies are gathered and it is identified that the learning procedures of these solutions can be classified as refinement or adaptation, in which the main difference between them is in how they use the new knowledge acquired. It is possible to identify with the empirical evaluation that the incremental solutions analyzed produce results with scores identical to those generated by batch learning solutions, but differ in the generalization of new data. It is also noticed that the characteristics of the context and restriction factors applied by the algorithms interfere in the generalization quality of the networks. In general, it is concluded that the incremental learning algorithms of Bayesian networks can be considered an acceptable solution in restricted contexts of use.

Agradecimentos

Agradeço a Deus pela saúde e paz durante o período dedicado à realização deste trabalho.

Aos meus pais, Dionice Pereira e Ednaldo Silva, pela dedicação de grandes esforços para minha educação. Pelo apoio e incentivo necessários. Aos meus irmãos, João Victor e Júlia Vitória, pelo lazer proporcionado.

À minha namorada, Andressa Soares, pelo amor, pela compreensão e pelo companheirismo que me abasteceu ao longo de todo o caminho trilhado. Ela sempre acreditou que seria possível.

A Mirko Perkusich pela apresentação do campo de Redes Bayesianas e dos vários esforços que devem ser realizados para a aprendizagem de tais modelos. Ao meu orientador, Kyller Gorgônio, pelo apoio para a concretização deste trabalho.

Aos amigos da pós-graduação que estiveram sempre dispostos a me auxiliar durante a realização deste trabalho.

À CAPES pelos investimentos na pós-graduação.

A todos que contribuíram, direta ou indiretamente, para a conclusão deste trabalho.

Conteúdo

Lista de Símbolos

IA - Inteligência Artificial

MGP - Modelos Gráficos Probabilísticos

RB - Redes Bayesianas

DCM - Descrição de Comprimento Mínimo

TPN - Tabela de Probabilidades do Nó

HCS - *Hill-Climbing Search*

IC - Independência Condicional

RSL - Revisão Sistemática da Literatura

MPP - Máxima Probabilidade a Posteriori

OTOC - Operadores Transversais na Ordem Correta

EBR - Espaço de Busca Reduzido

CB - *City Block*

PL - Perda Logarítmica

Lista de Figuras

Lista de Tabelas

Lista de Códigos Fonte

Capítulo 1

Introdução

A modelagem de dados é um dos principais tópicos da Inteligência Artificial (IA) atualmente. Desenvolver modelos é útil para ocultar a complexidade presente nos dados cada vez mais representativos do mundo real dado o aumento das fontes de informação. Estes modelos são utilizados como base para raciocínios sobre os fenômenos coletados nos dados ou para serem obtidas previsões sobre o resultado de certos eventos. Eles também podem ser aplicados em situações diferentes daquelas em que foram treinados, devendo então aprender os padrões nos dados e reconhecer situações semelhantes.

Os contextos abordados pela IA, em sua maioria, lidam com incertezas. Neste sentido, modelos probabilísticos, como a estrutura Dempster-Shafer [55], se tornam importantes por utilizarem a teoria da probabilidade para indicar diferentes graus de certeza sobre o contexto. No entanto, gerenciar a distribuição de probabilidade conjunta completa relacionada a um problema abordado pode ser intratável, sendo necessário, por vezes, usar uma representação mais compacta dos modelos probabilísticos [16].

Os Modelos Gráficos Probabilísticos (MGP) são ferramentas naturais para lidar com a incerteza e a complexidade presentes nos diversos contextos abordados pela IA [6]. Estes modelos fornecem uma associação entre a teoria das probabilidades e a teoria dos grafos. A teoria da probabilidade serve como a cola na qual as partes do modelo são combinadas, garantindo um modelo consistente e que realize uma interface com os dados. A teoria dos grafos, por sua vez, fornece uma interface intuitiva através da qual os seres humanos podem modelar conjuntos de variáveis altamente dependentes.

Redes Bayesianas (RBs) são MGPs populares que representam variáveis aleatórias de

um determinando domínio e suas dependências condicionais. As RBs fornecem um método sistemático para estruturar informações probabilísticas sobre determinadas situações do domínio. Com estes modelos, é possível derivar muitas implicações destas informações e formar a base para conclusões e decisões importantes sobre uma determinada situação.

O uso de RBs, também conhecida como redes de crenças Bayesianas, possui várias propriedades amplamente utilizadas. Entre elas, pode-se destacar: (i) o tratamento explícito de incerteza com probabilidades, fazendo uso da teoria da probabilidade; (ii) a representação intuitiva das relações entre as variáveis do domínio utilizando modelos gráficos; e (iii) a facilidade de estimar o estado de certas variáveis dadas algumas evidências [37].

Os domínios de aplicação de RBs têm sido amplos. Um grande número de aplicações está no campo da medicina [51, 5], sendo este um dos campos onde são encontrados mais trabalhos relacionados na literatura. Há também aplicações no domínio de previsão [1], controle [20], modelagem para compreensão humana [28]. No contexto da engenharia de software, campos como o planejamento de projetos [44], gestão dos riscos [15] e gestão da qualidade [29] também são abordados.

Há uma vasta literatura dedicada à melhoria do desempenho e precisão dos procedimentos de aprendizagem de RBs baseados em dados. A elicitação de redes que representam contextos complexos tende a ser um processo caro e laborioso quando realizado utilizando somente conhecimentos de especialistas. Em geral, a aprendizagem de RBs consiste na aprendizagem da sua estrutura e de seus parâmetros. A maioria dos avanços no aprendizado de estruturas tem se concentrado em algoritmos de aprendizagem em lote, onde é assumido que todas as instâncias de dados para treinamento do algoritmo estarão disponíveis para aprender a estrutura da rede em um único momento.

No entanto, grande parte das empresas, por exemplo, armazena bancos de dados cada vez maiores sobre seus processos de negócios, sejam eles estruturados ou não estruturados. Este fenômeno, conhecido com Big Data, faz com que novos conhecimentos sejam adquirido a todo momento e estas bases de dados cresçam continuamente. É praticamente impossível obter uma descrição altamente precisa dos processos envolvidos sem novas informações sendo coletadas e com isso, o modelo obtido em lote tende a, em um certo momento, estar desatualizado. Neste caso, reexecutar o algoritmo de aprendizado em lote não só exigirá muitos recursos, como também será um procedimento oneroso.

Diante das desvantagens dos algoritmos de aprendizagem de estruturas em lote citadas, surgiu a necessidade de soluções que incorporem, eficientemente e continuamente, o conhecimento atualizado ao conhecimento prévio. Com isso, o estudo e o desenvolvimento de algoritmos de aprendizagem incremental de estruturas tomou grande importância e valor prático no processo de aprendizagem de RBs.

Este trabalho está inserido no contexto de aprendizagem incremental de estruturas de RBs. Mais especificamente, o foco deste trabalho é a mais profunda da relação entre a qualidade das estruturas geradas e algoritmos incrementais existentes, sendo este um dos principais fatores para a eficiência do processo de aprendizagem de RBs.

1.1 Problemática

Diferentes métodos foram propostos para enfrentar o problema de aprendizagem incremental de estruturas de RBs e garantir soluções ótimas [46, 50, 21]. Estes métodos são classificados em três categorias:

- métodos baseados na detecção de independências condicionais, também conhecidos como métodos baseados em restrições. Os algoritmos baseados nestes métodos utilizam um conjunto de relações de independência entre subconjuntos de variáveis para construir uma RB que represente uma grande porcentagem, ou todas, caso seja possível, dessas relações.

- métodos de busca e pontuação, também conhecidos como abordagens baseadas em pontuação. Estes métodos estão mais próximos da semântica de RBs e, por esse motivo, é onde se enquadram a maioria dos algoritmos incrementais. Os algoritmos baseados nestes métodos são resumidos à soluções de problema de otimização matemática. Dado isso, os dois principais componentes destes algoritmos são uma métrica de pontuação e um método de busca que objetiva uma solução ótima dado esta métrica;

- métodos híbridos. Os algoritmos baseados neste métodos utilizam técnicas de ambos os métodos citados anteriormente.

Em geral, a solução proposta por estes algoritmos, independente do método utilizado, é uma estrutura de RB que otimiza uma função de pontuação definida. Esta função representa,

tipicamente, uma probabilidade Bayesiana posterior da estrutura dado um conjunto de dados, como a função de verossimilhança penalizada. Sendo assim, a pontuação das estruturas, neste contexto, é um reflexo de quão bem ela modela um conjunto de dados de treinamento em um determinado passo de aprendizagem.

Levando este fato em consideração, alguns estudos avaliam a qualidade das estruturas aprendidas pela sua classificação dado uma função de pontuação. Esta avaliação é feita, geralmente, realizando uma comparação entre as estruturas aprendidas por algoritmos em lote e incrementais. Shi e Tan [50], por exemplo, avaliam a qualidade das redes geradas por seu algoritmo utilizando o valor médio da função de pontuação definida por Descrição de Comprimento Mínimo (DCM). Li et al. [38] também utiliza outra função de pontuação, Critério de Informação Bayesiano, para o mesmo objetivo.

É possível notar então que as avaliações anteriores supõem que as estruturas resultantes dos algoritmos incrementais possuem qualidade superior as resultantes dos algoritmos em lote por modelarem melhor o conjunto total de dados de treinamento, mas não necessariamente, por explicar, com precisão, possíveis novos dados. É bem sabido que um modelo pode descrever um conjunto de treinamento muito bem, mas generalizar mal a novos dados [24], provocando um *overfitting* nos dados. Assim, não há garantia de que uma rede que otimize uma pontuação para um conjunto de treinamento irá generalizar bem para novos dados.

A diferença estrutural entre as redes geradas por algoritmos incrementais e em lote também é utilizada como base para avaliação de qualidade em alguns trabalhos, seja ela em conjunto com a pontuação da rede [46] ou não [31]. Ainda que possuam valores nulos, indicando a igualdade entre as estruturas, este resultado não explica o quão qualificada está a rede para a generalização de novos dados.

Existem poucos trabalhos que além da pontuação e da diferença estrutural, avaliam a predição dos modelos gerados pelos algoritmos [2, 12]. Estas avaliações experimentais buscam medir a predição, seja do modelo final ou dos modelos gerados durante o processo de aprendizagem, utilizando classificações existentes em um conjunto de dados testes.

Alguns dos algoritmos avaliados, como no algoritmo de Alcobé [46], aprendem estruturas com características diferentes quando alimentados com dados de diferentes granularidades. A complexidade de um contexto pode ser avaliada dado o número de atributos

analisados, quantidade de instâncias, mas também pela variação das distribuições dos atributos explicadas pelo conjunto de dados a cada passo de aprendizagem. Esta complexidade influencia diretamente na qualidade de predição dos modelos aprendidos, neste caso, e este comportamento é abordado de forma superficial pelas avaliações empírica destes trabalhos.

Huang [27] apresenta em seu estudo uma pesquisa comparativa entre algoritmos de aprendizagem incremental de estruturas de RBs. Este estudo detalha os algoritmos de Buntine [8], Friedman e Goldszmidt [21] e o de Lam e Bacchus [33], principais algoritmos até o ano de sua realização, 2003. O autor também apresenta uma análise teórica sobre as complexidades computacionais destes algoritmos, mas não aborda a qualidade das redes aprendidas. Para validar sua análise, o autor realiza uma avaliação experimental onde indica diferenças entre os algoritmos abordados. No entanto, este trabalho é antigo, não abordando algoritmos mais recentes existentes na literatura. Além disso, ainda realiza uma experimentação sem uma clara descrição e análise.

Portanto, dado as avaliações citadas, é possível perceber que não há evidência empírica clara sobre a justificativa de uso dos algoritmos incrementais de aprendizagem de estruturas em termos de qualidade de generalização das RBs aprendidas. Como também não há uma explicação clara sobre a relação entre o uso de um determinado algoritmo incremental e a qualidade do modelo alcançado em determinados contextos de uso.

Além disso, como já citado, a escolha de algoritmos incrementais pode afetar a estrutura final descoberta de uma RB. De acordo com o método utilizado, os fatores adotados para cada algoritmos podem alterar o fluxo do processo de aprendizagem. O número de instâncias por passos de aprendizagem, por exemplo, pode aumentar a variação das distribuições dos dados, afetando o desempenho dos algoritmos. A restrição do número de pais a ser encontrado pelos algoritmos de busca e pontuação, o uso do procedimento de aprendizagem como refinamento (rede inicial parcialmente conhecida) ou não, dentre outros, são todos fatores que podem alterar o desempenho dos algoritmos, afetando o resultado encontrado.

A influência dessas características não possui uma explicação empírica clara na literatura. Como em Shi e Tan [50], algumas avaliações empíricas são realizadas considerando o tamanho do passo de aprendizagem, mas sua influência na qualidade final da RB é brevemente avaliada.

Malone [40] apresenta um estudo empírico para avaliar, com um melhor entendimento, as

relações entre alguns algoritmos de aprendizagem de estruturas e a qualidade de generalização das redes resultantes. Os resultados deste estudo indicam diferenças entre os resultados obtidos por métodos diferentes. No entanto, seu estudo é resumido as soluções em lote, não abordando as soluções incrementais. Até o momento deste trabalho, avaliações similares em soluções incrementais não foram encontradas. Desta forma, o problema em questão é: como se comportam os algoritmos estado da arte de aprendizagem incremental de estruturas de redes Bayesianas em domínios com diferentes complexidades e variações de seus fatores?

1.2 Objetivos

O principal objetivo deste trabalho é avaliar o comportamento de algoritmos estado da arte de aprendizagem incremental de estruturas de RBs em domínios com diferentes complexidades e variações de fatores. Nesta dissertação, a relação entre a qualidade das RBs aprendidas e as soluções de aprendizagem incremental que usam o paradigma de busca e pontuação, e informação mútua entre as variáveis é o principal ponto explorado. A eficácia na descoberta das estruturas e a generalização de novos dados são parâmetros utilizados para descrever a qualidade da rede aprendida pelos algoritmos.

O objetivo geral desse trabalho é dividido nos seguintes objetivos específicos:

- identificar as principais soluções para a aprendizagem incremental de estruturas de redes Bayesianas;
- comparar, a partir de métricas relevantes para a qualidade das estruturas geradas, os algoritmos estado da arte em aprendizagem incremental, verificando a existência de melhorias na estrutura final aprendida;
- identificar os principais fatores que interferem no processo de aprendizagem incremental dos algoritmos.

1.3 Metodologia

Três etapas são adotadas para a realização deste trabalho após a revisão literária sobre o tema e definição do problema de pesquisa.

A primeira etapa está relacionada com o objetivo específico de identificação das soluções. Nesta etapa, diante da carência de trabalhos reunindo as principais soluções incrementais, a literatura é revisada sistematicamente com o objetivo de identificar soluções representativas de aprendizagem incremental de RBs. Como resultado desta revisão, algumas soluções são identificadas e dentre elas, cinco recebem destaque, ou por apresentarem boas pontuação em avaliações experimentais dado uma função de pontuação, ou por serem trabalhos pioneiros no tema abordado.

A segunda etapa está relacionada com os objetivos específicos restantes. Avaliações experimentais empíricas em dois dos algoritmos incrementais identificados anteriormente são realizadas com o intuito de validar seu comportamento e capacidade em contextos com diferente complexidade. Os algoritmos de Alcobé [2], e Shi e Tan [50] são os abordados nesta etapa. Estes algoritmos aprendem RBs com pontuação semelhante às redes de soluções em lote, mantendo esta premissa em diferentes contextos.

Em um dos experimentos empíricos realizados nesta etapa, os princípios da aprendizagem incremental são validados através de comparações entre as redes aprendidas pelos algoritmos em lote e incrementais. Nos experimentos restantes, fatores dos algoritmos e dos contexto são alterados e os resultados alcançados pelos algoritmos incrementais são comparados.

A terceira etapa está relacionada com a análise dos resultados. Neste etapa, conclusões sobre a avaliação e validação dos achados nas etapas anteriores são elaboradas.

1.4 Contribuições e Resultados

A seguir, os principais resultados e contribuições deste trabalho são apresentados.

Uma revisão detalhada do estado da arte sobre métodos incrementais para o aprendizagem de estruturas de RBs é apresentada, onde diferentes formas de solução do problema de aprendizagem são abordadas. Algumas destas descobertas são apresentadas em Silva et al. [52].

Uma avaliação experimental para comparar o comportamento de aprendizagem dos algoritmos incrementais citados com um algoritmo de aprendizagem em lote é também realizada. É concluído que ambas as soluções produzem redes com pontuação semelhante, mas a gene-

ralização de novos dados é diferente em diversos casos. A implementação desses algoritmos está disponível para livre acesso da comunidade científica.

Uma avaliação experimental para avaliar como as características do contexto abordado pelos algoritmos afetam seus resultados também é realizada. Características como ordem das instâncias inseridas no processo de aprendizagem, tamanho do passo de aprendizagem são consideradas como influentes na qualidade final das redes produzidas.

Outra avaliação experimental é realizada para identificar as restrições dos algoritmos que afetam significativamente a qualidade das redes aprendidas pelos algoritmos em lote. É identificado que a restrição referente ao número máximo de pais utilizada pelas soluções que fazem uso de busca por pontuação possuem efeito significativo nas métricas de qualidade analisadas. Em algumas delas, o efeito é negativo. Em resumo, diretrizes para uso destes algoritmos incrementais são apresentados.

1.5 Estrutura da Dissertação

O restante do documento está estruturado da seguinte maneira:

- no Capítulo 2, a fundamentação teórica deste trabalho é apresentada. São revisados o campo de RBs e seus componentes (estruturas e parâmetros) em detalhes e apresentados alguns conceitos que subsidiam este estudo;

- no Capítulo 3, a revisão do estado da arte de algoritmos de aprendizagem incremental de RBs é apresentada. Alguns algoritmos são descritos e são apresentados detalhes dos mais representativos. Nomeadamente, estes são os algoritmos de Buntine [8], Friedman e Goldszmidt [21], Alcobé [2], Lam e Bacchus [33], e Shi e Tan [50];

- no Capítulo 4, os resultados da avaliação empírica sobre a eficiência na descoberta das estruturas e eficácia na generalização de novos dados por algoritmos detalhados no capítulo anterior são apresentados. Nomeadamente, os algoritmos de Alcobé [2], e Shi e Tan [50] são utilizados;

- no Capítulo 5, um resumo deste trabalho é apresentado, juntamente com algumas conclusões e sugestões de linhas para futuras pesquisas.

Capítulo 2

Fundamentação Teórica

Neste capítulo, uma base teórica dos temas a serem abordados nos próximos capítulos desta dissertação é apresentada, assim como alguns dos principais conceitos utilizados para subsidiar este estudo. Na Seção 2.1, é apresentado o conceito de redes Bayesianas juntamente com alguns dos conceitos básicos relacionados. Na Seção 2.2, são destacados os conceitos de conhecimento a piori e a posteriori e na Seção 2.3, o conceito de aprendizagem de RBs e algumas notações que serão utilizadas no restante do documento são definidos.

2.1 Redes Bayesianas

Na teoria da probabilidade, um domínio D e suas incertezas podem ser modelados por um conjunto de variáveis aleatórias $\mathbf{D} = \{X_1, X_2, ..., X_n\}$. Cada variável aleatória X_i possui um conjunto de valores possíveis que combinados compõem a base para a modelagem do domínio D. A ocorrência de cada combinação possível é medida usando probabilidades que são especificadas pela distribuição de probabilidade conjunta, um conceito-chave da teoria da probabilidade.

Em muitos domínios, existe um elevado número n de variáveis, exigindo o uso de MGPs para a definição da distribuição de probabilidade conjunta [6]. RBs pertencem à família desses modelos que são usados para representar um domínio e suas incertezas. Uma RB é uma descrição explícita de uma distribuição de probabilidade conjunta sobre um conjunto de variáveis aleatórias $\mathbf{D}$.

2.1.1 Definição de Redes Bayesianas

Formalmente, uma RB para D é definida pelo par $B = \{G, \mathbf{\Theta}\}$. O primeiro componente, G, é um grafo acíclico dirigido. Este grafo acíclico (também conhecido por *topologia* ou *estrutura*) consiste de um conjunto de vértices (ou nós) correspondentes ao conjunto de variáveis aleatórias $\mathbf{D}$ e de arestas (ou arcos) que representam ligações entre variáveis dependentes condicionais. Um arco partindo de X_i para X_j indica uma suposição de que há uma dependência direta causal ou influente de X_i em X_j. O nó X_i é então considerado *pai* de X_j. No grafo, também não há ciclos com o objetivo de evitar o raciocínio circular. Então, se existe um arco partindo de X_i até X_j e outro partindo de X_j até X_k, não se pode ter um arco partindo de X_k até X_i.

O segundo componente, $\mathbf{\Theta}$, representa o conjunto de parâmetros que quantifica a rede. Cada nó X_i tem uma tabela de probabilidade associada chamada de Tabela de Probabilidades do Nó (TPN). Esta é uma distribuição de probabilidade de X_i dado o conjunto de pais de X_i. Dessa forma, $\mathbf{\Theta}$ contém um parâmetro $\Theta_{ijk} = P\left(X_i = x_i^k | \mathbf{Pa}_i = \mathbf{pa}_i^j\right)$ para cada possível estado x_i^k de X_i e para cada configuração $\mathbf{pa}_i^j$ de $\mathbf{Pa}_i$. $\mathbf{Pa}_i$ é definido como o conjunto de pais da variável X_i.

Considerando ainda a RB para D, a distribuição de probabilidade conjunta completa definida pelas regras de cadeia é descrita como

$$P\left(X_1, X_2, ..., X_n\right) = P\left(X_1 | X_2, ..., X_n\right) P\left(X_2 | X_3, ..., X_n\right) P\left(X_{n-1} | X_n\right) P\left(X_n\right)$$

que pode ser reescrita como

$$P\left(X_1, X_2, ..., X_n\right) = \prod_{i=1}^{n} P\left(X_i | X_{i+1}, ..., X_n\right).$$

No entanto, é possível simplificar tal definição utilizando somente os pais representados pelas estruturas da RB. Supondo que X_1 tem somente dois pais, X_3 e X_4. Então, a seguinte parte da distribuição $P\left(X_1 | X_2, ..., X_n\right)$ pode ser representada de forma equivalente e sem perca de informação por $P\left(X_1 | X_3, X_4\right)$ (veja mais na Seção 2.1.2). Então, a distribuição de probabilidade conjunta completa da RB citada pode ser simplificada por

$$P\left(X_1, X_2, ..., X_n\right) = \prod_{i=1}^{n} P\left(X_i | \mathbf{Pa}_i\right).$$

Para a realização de uma leitura adicional sobre definições da teoria da probabilidade e RBs, a leitura de [16] é indicada. Este livro apresenta ao leitor a teoria da probabilidade e, utilizando uma linguagem pouco rebuscada, a associa com o conceito de RBs. Outro livro clássico que também introduz o princípio de RBs e é bastante importante para quem busca mais fundamentos é [43].

Um exemplo de RB é apresentado na Figura 2.1. Os nós são representados por círculos, enquanto os arcos são representados por setas indicando a direção da conexão causal, ainda que seja possível propagar informação em qualquer direção na estrutura com base nos pressupostos de independência [3].

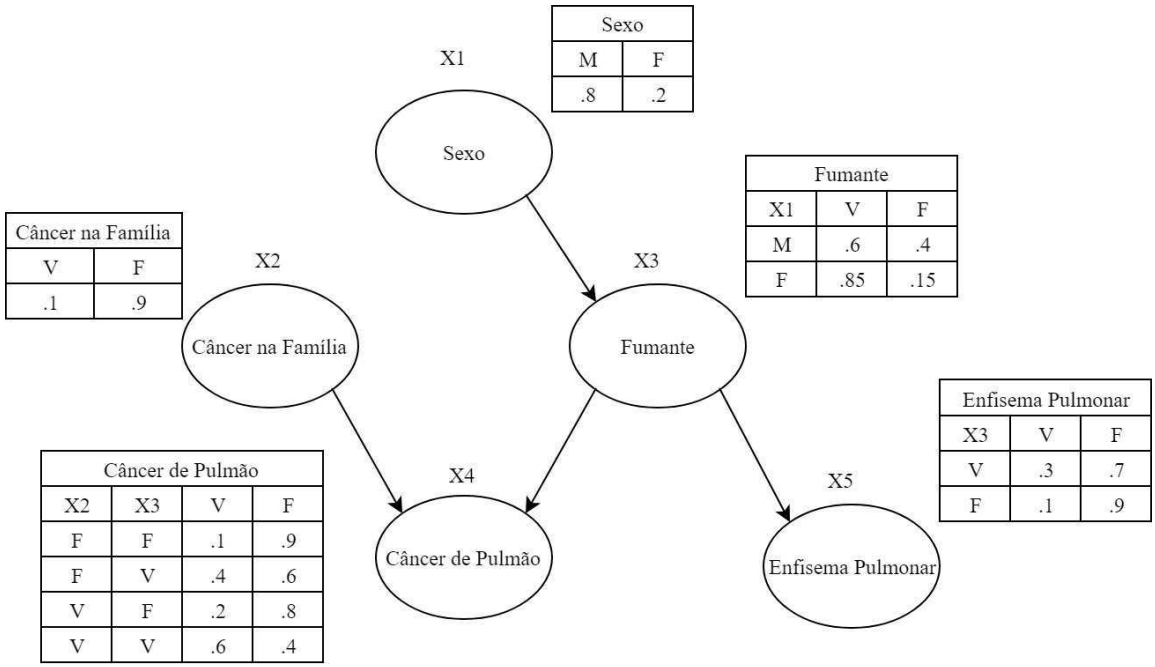

Figura 2.1: Exemplo de rede Bayesiana

Na RB exemplo apresentada, deseja-se calcular a probabilidade de uma pessoa possuir câncer de pulmão ou possuir um enfisema pulmonar dado seu sexo, o histórico de câncer de sua família e\ou sua característica de fumante. O nó X_1 indica que, considerando o conhecimento (ou experiência) existente, 80% das pessoas que procuram algum tipo de tratamento relacionado ao tabagismo são do sexo masculino. Para o nó X_2, o *conhecimento a priori* com relação à variável assume que 90% dos familiares do paciente não possuem câncer. Esta distribuição de probabilidade expressa a incerteza inicial sobre o valor de X_2. Para o nó X_3,

assume-se que em 60% dos casos, a pessoa analisada é fumante. Esse valor também pode ser atualizado com uma *evidência* caso ela seja coletada. A TPN dos nós filho representam as probabilidades do diagnóstico considerando os valores dos nós pai.

Observando então o exemplo apresentado na Figura 2.1 e definições citadas até aqui, pode-se, de fato, destacar os seguintes benefícios no uso desse modelo gráfico [16]: (i) os fatores causais são explicitamente modelados; (ii) razões do efeito para causa e vice-versa também são modelados; (iii) reduz a quantidade de valores de probabilidades e parâmetros requeridos se comparado com o modelo de probabilidade conjunta completo; (iv) faz predições com dados incompletos; (v) desconsidera crenças anteriores à luz de novas evidências; (vi) combina diversos tipos de evidências, desde crenças subjetivas à dados objetivos; entre outros.

2.1.2 Pressuposto de Independência Condicional

Como dito anteriormente, as RBs são modelos gráficos que, explicitamente, modelam as relações causais entre as suas variáveis. Com base nesse benefício, essas redes permitem a criação de pressupostos de independência mesmo que estes não tenham sido explicitamente especificados.

Analisando o exemplo de RB apresentado na Figura 2.1, pode-se perceber que, ao calcular a probabilidade marginal (veja mais em [16]) de uma determinada pessoa possuir câncer de pulmão (nó X_4), assume-se que X_4 é dependente, apenas, de X_2 e X_3. O nó X_5 não pertence ao cálculo da probabilidade marginal de X_4 porque considera-se que nenhuma das outras variáveis é diretamente dependente de X_5 devido aos arcos presentes entre os nós. No mesmo sentindo, pode-se dizer também que X_2 e X_3 são independentes entre eles.

Este tipo de pressuposto baseado na estrutura da rede é conhecido como pressuposto de Independência Condicional (IC). Este pressuposto é lido a partir da estrutura da rede utilizando-se o critério de *separação direcional*. Este critério é baseado em três propriedades estruturais de RBs que são importantes para sua compreensão: (i) conexão serial; (ii) conexão divergente; e (iii) conexão convergente.

Considere, por enquanto, apenas os nós da RB exemplo destacados na Figura 2.2. O nó X_1 tem influência em X_3, que por sua vez, tem influência em X_5. Nesse sentido, qualquer evidência inserida em X_1 será propagada para X_3 e X_5. O sentido contrário da propagação,

caso a evidência seja inserida em X_5, também ocorrerá devido às propriedades já discutidas de uma RB. No entanto, se uma evidência também for inserida em X_3, X_1 e X_5 tornam-se independentes já que o fluxo citado é bloqueado. Nesse sentido, pode-se dizer que X_1 e X_5 são independentes condicionais (ou separados direcionalmente) dado X_3.

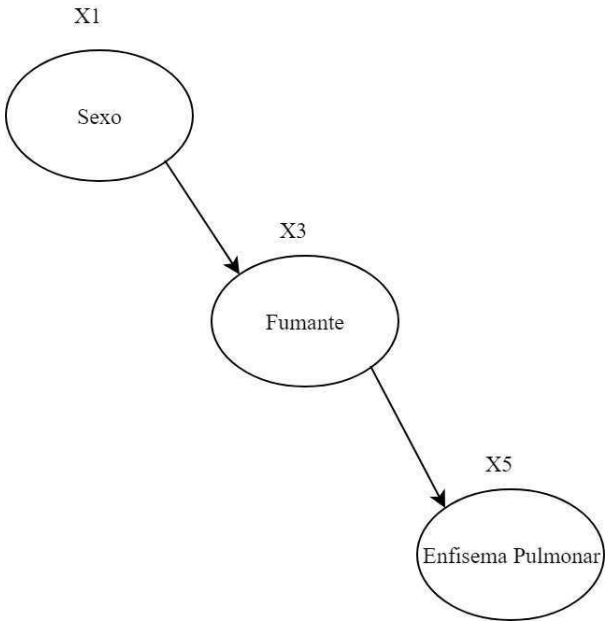

Figura 2.2: Exemplo de conexão serial

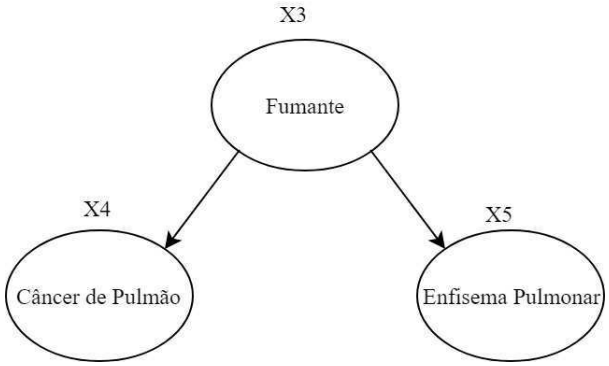

Figura 2.3: Exemplo de conexão divergente

Considere agora os nós destacados na Figura 2.3. Neste caso, a evidência inserida em X_3 tem seu valor propagado entre seus nós filho X_4 e X_5. A evidência também inserida em qualquer um dos filhos será propagada até o outro utilizando o nó X_3. No entanto, isto não

acontecerá se o estado de X_3 for conhecido pela rede. Pode-se dizer então que X_4 e X_5 são independentes condicionais (ou separados direcionalmente) dado X_3.

Por último, agora considere os nós destacados na Figura 2.4. Neste caso, a evidência inserida em qualquer um dos nós pai X_2 e X_3 não tem seu valor propagado ao outro, sendo assim independentes. No entanto, se qualquer evidência for inserida em X_4, então esse valor é propagado até os dois pais X_2 e X_3, tornando-os dependentes.

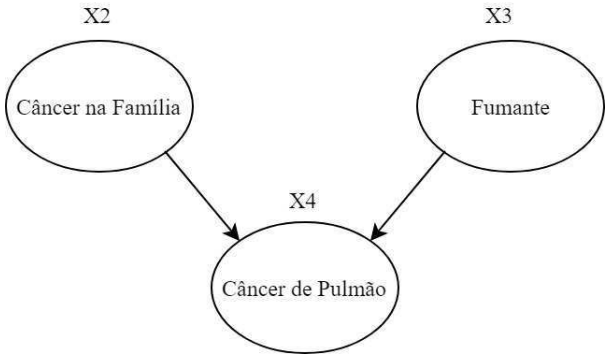

Figura 2.4: Exemplo de conexão convergente

Cobertas as três propriedades estruturais, é possível então saber, para qualquer nó em uma RB, sua dependência dado uma evidência e então definir separação direcional [16]:

Definição 1 (Separação direcional)**.** *Duas variáveis X_i e X_j em uma rede Bayesiana são separadas direcionalmente se, para todos os caminhos entre X_i e X_j, existir uma variável X_k tal que a conexão seja serial ou divergente e o estado de X_k seja conhecido, ou se a conexão é convergente e não existe entrada de evidência para X_k ou seus descendentes.*

2.2 Introdução à Inferência: Conhecimento a Priori e a Posteriori

A maioria dos casos para os quais tem-se que atribuir uma probabilidade inicial P não há uma abordagem frequentista razoável e, portanto, deve-se usar, pelo menos, algum julgamento subjetivo (veja mais em [16]). Buscando melhorias no conhecimento, evidências são observadas para que possam ajudar a revisar o conhecimento que tem-se sobre uma determinada hipótese H e sua probabilidade $P(H)$.

Mantendo-se no exemplo apresentado na Figura 2.1, considere a gerência de um hospital que trata de pessoas com possíveis doenças respiratórias originadas do seu hábito de fumar. Precisando organizar a demanda de leitos necessários, a gerência precisa saber a probabilidade de existirem pacientes que possivelmente serão diagnosticados com câncer durante os próximos dias. Baseado em sua vivência anterior, esta gerência deduz que algum paciente será diagnosticado com câncer. Então, sem os observar, a gerência atribui uma probabilidade exemplo de 0,15% de existirem pessoas com câncer.

No entanto, ao encontrar esses pacientes dia após dia, a gerência do hospital reúne observações que mudam seu conhecimento. Nota-se que, frequentemente, existem mais pacientes do sexo masculino e percebe-se que, cada vez mais, o número de pacientes que possuem familiares com câncer aumenta. Além disso, dentre os pacientes diagnosticados, existem sintomas em comum, o que leva a gerência a analisar mais uma variável do domínio, que por sua vez, não está presente no modelo. Então, ao se deparar com uma determinada evidência, a gerência atualiza o seu conhecimento sobre a hipótese analisada, revisando assim a probabilidade inicial $P(H)$. Em outras palavras, $P(H|E)$ é calculada, onde E representa cada evidência colhida.

Este exemplo permite afirmar que, sobre determinada hipótese H, existe um conhecimento $P(H)$ que é definido por *conhecimento a priori*. Usando então evidências e observações sobre H, é possível revisar $P(H)$ e calcular $P(H|E)$, que é definido por *conhecimento a posteriori*.

2.3 Aprendizagem de Redes Bayesianas

O objetivo do procedimento de aprendizagem de uma RB é encontrar uma rede que melhor explique a distribuição de probabilidade conjunta de um domínio. Aprendizagem de RBs pode ser definido como [48]:

Definicão 2 (Aprendizagem de redes Bayesianas)**.** *Dado um conjunto de dados, deve-se deduzir a topologia de uma RB que pode ter gerado o conjunto de conhecimento juntamente com a distribuição de incerteza correspondente.*

O problema de dedução definido como aprendizagem de RBs pode ser decomposto em dois subproblemas: (i) construção da estrutura; e (ii) definição das TPNs. O conhecimento

usado como base pode ser representado por meio de conjuntos de dados brutos, por meio do especialista de domínio ou ambos [31].

A seguir, alguns métodos de aprendizagem em lote de RBs são descritos de forma sucinta. O foco da próxima seção é mantido no algoritmo de Busca de Escalada em Colina (*Hill-Climbing Search - HCS*), base para alguns dos algoritmos utilizados nos capítulos seguintes deste trabalho.

2.3.1 Aprendizagem em Lote

Muito dos trabalhos de aprendizagem de RBs tem mantido seu foco nos algoritmos de aprendizagem em lote (*batch*), isto é, algoritmos que supõem ter todas as instâncias de treinamento de dados para aprender uma rede. Estes algoritmos, dado todo o conjunto de dados de treinamento, geram um modelo considerando bom (ou ótimo) depois de processá-lo.

No contexto de utilização de conjunto de dados e aprendizagem em lote, a definição dos parâmetros das TPNs é abordada por algumas soluções, como em [36], onde o algoritmo conhecido como Maximização de Expectativa é apresentado. Outras soluções para o problema utilizando as diversas formas de representação do conhecimento podem ser encontradas em [25, 59, 63].

Para construção da estrutura utilizando apenas conjuntos de dados como representação do conhecimento existente, alguns algoritmos de aprendizagem em lote podem ser destacados. K2, proposto por Cooper e Herskovits [11], e B, proposto por Buntine [8], buscam pela rede que melhor se encaixa nos dados em um conjunto de redes gerado através de um grupo de variáveis ordenadas utilizadas como entrada. Outro algoritmo proposto por Chow e Liu [9] também tenta solucionar o problema de construção da estrutura, só que utilizando uma estrutura de árvore. Tsamardinos et al. [54] utiliza técnicas de pontuação e busca, baseadas em restrições, além de uma busca local e desenvolve o algoritmo *Max-Min Hill-Climbing*. Larrañaga et al. [35], por sua vez, propôs um procedimento baseado em algoritmos genéticos. Outras soluções podem ser encontradas em [19, 22]. Dentre os algoritmos de aprendizagem em lote e antes de introduzir os algoritmos incrementais, destaca-se o HCS.

O HCS, como uma heurística de busca usada para solução de problemas de otimização matemática, busca definir um novo modelo através do ranqueamento, a cada passo da busca, de todas as alternativas de modelos possíveis. Este ranqueamento é baseado em uma função

objetiva e o novo modelo definido pelo HCS maximiza esta função. O HCS é definido, basicamente, pelos seguintes elementos:

- uma função objetiva, ou heurística, definida por $S(B, D)$ usada para medir a qualidade de um dado modelo B considerando um dado conjunto de dados D;
- um conjunto de operadores $\mathbf{OP} = \left\{op^1, ..., op^k\right\}$ que dado um argumento A e um modelo B, pode-se definir um novo modelo $B^{'} = op^i(M, A)$;
- um domínio $\mathbf{D}$ para definir os modelos legais, isto é, modelos permitidos no contexto da busca.

Estes conceitos servem como base para a definição de vizinhança [46], importante para a compreensão do HCS:

Definicão 3 (Vizinhança)**.** *A vizinhança de um modelo B, definida por $\mathbf{N}(\mathbf{B})$, é o conjunto de todos os modelos alternativos que pertencem ao domínio $\mathbf{D}$ e que podem ser construídos utilizando um único operador op^i. $\mathbf{N}(\mathbf{B})$ pode ser definido formalmente por*

$$\mathbf{N}(\mathbf{B}) = \left\{B^{'} | B^{'} = op^i(B, A) \wedge B^{'} \in \mathbf{D}\right\}$$

Seguindo a mesma ideia, pode-se também definir o conjunto de pares de operadores e argumentos permitidos com que a vizinhança $\mathbf{N}(\mathbf{B})$ é obtida [46]:

Definicão 4 (Conjunto de operadores)**.** *O conjunto de pares de operadores e argumentos usados para construir a vizinhança $\mathbf{N}(\mathbf{B})$, denotado por $\mathbf{OpA_{N(B)}}$, é o conjunto de todos os pares que se aplicados ao modelo M, será obtido um novo modelo $B^{'}$ que pertence ao domínio $\mathbf{D}$. $\mathbf{OpA_{N(B)}}$ pode ser definido formalmente por*

$$\mathbf{OpA_{N(B)}} = \left\{\left(op^i, A_i\right) | op^i(B, A_i) \in \mathbf{D}\right\}$$

O HCS inicia de um modelo inicial B_0, seja ele vazio ou não, e, iterativamente, constrói uma sequência de modelos B_i, onde i varia de $0, ..., n$. Cada modelo B_i construído em cada passo do algoritmo é o modelo com maior pontuação, baseado na função objetiva definida, entre $\mathbf{N}(\mathbf{B_{i-1}})$. Sabe-se então que $B_i = argmax_{B \in \mathbf{N}(\mathbf{B_{i-1}})} S(B, D)$. A busca é interrompida quando não é mais possível otimizar a solução encontrada, ou seja, o modelo atual B_i é

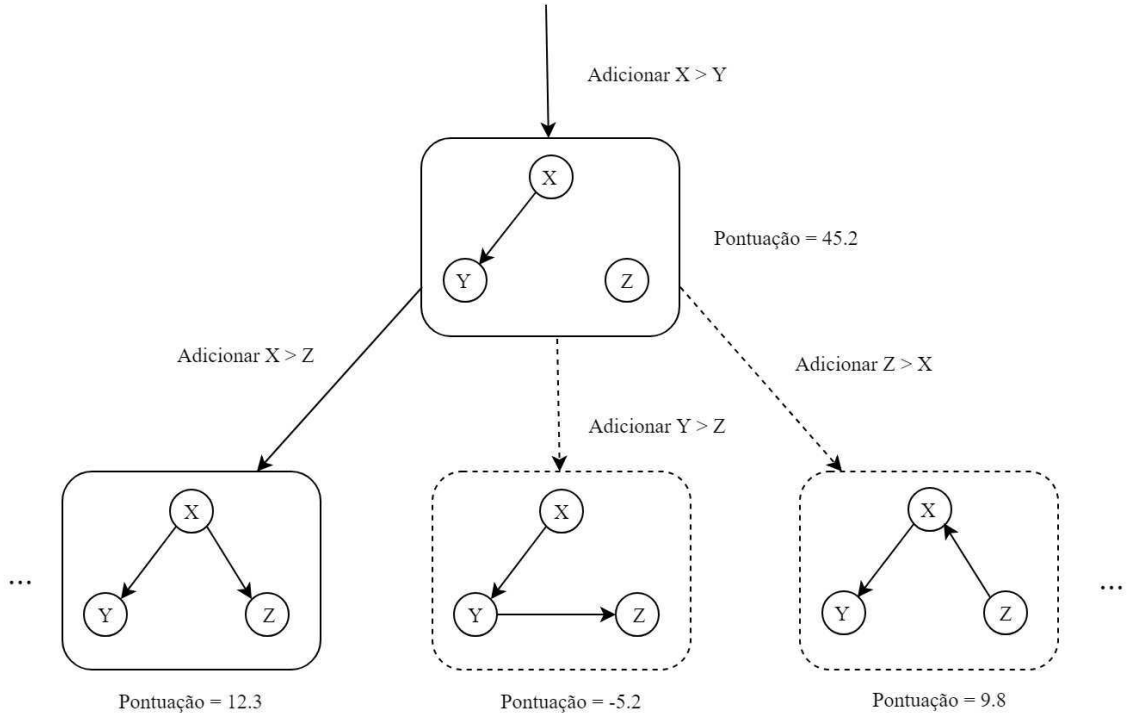

Figura 2.5: Procedimento do HCS durante aprendizagem de estrutura

o que possui maior pontuação dentre $\mathbf{N}(\mathbf{B_i})$. Este procedimento é mostrado, graficamente, na Figura 2.5.

Na Figura 2.5, pode-se notar que, após o HCS realizar a operação de inserção do arco partindo de X até Y e gerar o modelo B, a pontuação de todos os pares pertencentes ao conjunto $\mathbf{OpA_{N(B)}}$ é analisada e selecionada a de maior pontuação. No caso analisado, a próxima operação seria a inserção do arco partindo de X até Z, como destacado na Figura 2.5. E assim sucessivamente até que o HCS não consiga encontrar nenhuma operação que aumente a pontuação do modelo atual.

Diante dos casos definidos anteriormente, a definição do problema de aprendizagem de RBs permitiu que várias soluções como as descritas até aqui fossem desenvolvidas para solucioná-lo. Nas próximas seções, conceitos e notações sobre algoritmos de aprendizagem que ainda precisam ser identificados serão brevemente descritos.

2.3.2 Espaço de Busca de Redes Bayesianas

Um espaço de busca no contexto de aprendizagem pode ser descrito como um espaço de conhecimento onde se localizam estados que representam estruturas de conhecimento [46]. Geralmente, esses estados são parcialmente ordenados a fim de guiar os procedimentos de busca, como apresentado na Figura 2.6.

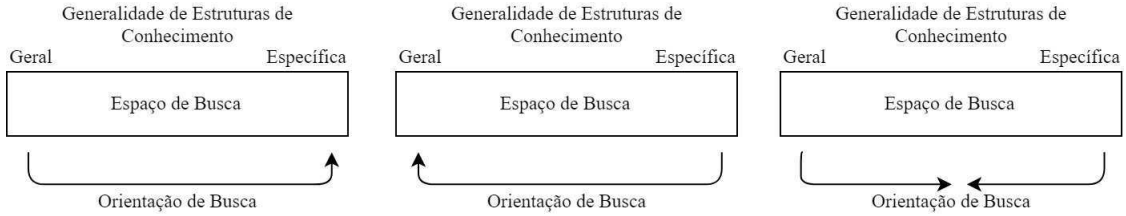

Figura 2.6: Relação entre espaço e procedimento de busca

As estruturas são distribuídas de acordo com sua generalidade: em um extremo, situam-se as estruturas mais gerais, enquanto no oposto ficam as mais específicas. As estruturas centrais possuem um grau intermediário de generalidade. A busca pode ser iniciada partindo dos estados gerais até os específicos ou vice-versa. Outro procedimento de busca pode partir de ambos os extremos até resultar na estrutura em que os dois caminhos de busca se encontram.

Frequentemente, os espaços de busca são grandes o suficiente para se tornarem inviáveis de serem percorridos e armazenados. A seguir, o espaço de busca de estruturas de RBs considerando o estudo apresentado por Robinson [45] e citado por Alcobé [46] é analisado. Nesse estudo, o tamanho do espaço $G(n)$ pode ser calculado a partir do número de nós n através de

$$G(n) = \begin{cases} 1 & \text{if } n = 0; \\ \sum_{i=1}^{n} (-1)^{i+1} \begin{pmatrix} n \\ k \end{pmatrix} 2^{i(n-1)} G(n-1) & \text{if } n > 0. \end{cases}$$

Alguns valores de $G(n)$ para alguns valores de n são apresentados na Tabela 2.1 [46]. Como observado nesta tabela, o tamanho do espaço de busca cresce exponencialmente no número de nós. Esse crescimento dificulta o desenvolvimento de soluções computacionais viáveis, sendo considerado por muitos autores como um problema *NP-hard* [2, 26, 56].

n	$G(n)$
0	1
1	1
2	3
3	25
4	543
5	29.281
.	.
8	783.702.329.343
.	.
10	4.175.098.976.430.589.143

Tabela 2.1: Tamanho de espaço de busca por número de nós

2.3.3 Descrição de Comprimento Mínimo

As RBs são desenvolvidas como um modelo que explica comportamentos presentes em um domínio representado por um conjunto de conhecimento. Por esta razão, a maioria das soluções de aprendizado de RBs busca aproximar-se da distribuição apresentada pelo conhecimento (seja em dados coletados do mundo real ou do especialista). Dado isso, pode-se entender que a qualidade do modelo está diretamente ligada a como são avaliados os passos de procedimento utilizados na solução e suas métricas base.

Considerando as medidas de qualidade utilizadas, pode-se dividir os algoritmos de aprendizagem em três grupos distintos [46]. Um deles, os algoritmos baseados em restrições, é baseado na aplicação de testes condicionais de independência entre as variáveis e a construção da rede é baseada nos resultados obtidos. Outro grupo é baseado em medidas de adequação entre a distribuição presente na base de conhecimento e nas redes alternativas. Um terceiro grupo baseia-se na intersecção entre os grupos anteriores. Dentro do segundo grupo, pode-se destacar DCM, inferência Bayesiana, entre outros.

O DCM baseia-se na ideia de que o modelo que melhor adequa-se à distribuição presente nos dados é o que minimiza a soma do *comprimento de codificação* do próprio modelo e dos dados dado o modelo [32]. Baseado nesse conceito, é preciso então codificar o modelo de

RB e os dados dado o modelo e depois, medi-los em bits.

Sabendo que uma RB é composta de uma estrutura G e de um conjunto de tabelas de probabilidades condicionais $\mathbf{\Theta}$, deve-se codificar cada uma das duas partes. Para medir a codificação de G, considera-se que, para listar os pais de um nó X_i, são necessários $|\mathbf{Pa}_i|\log_2(n)$ bits. Logo, o seguinte número de bits é necessário para codificar a estrutura G de uma RB

$$\sum_{i=n}^{n} |\mathbf{Pa}_i|\log_2(n)$$

Para medir a codificação de $\mathbf{\Theta}$, considera-se que o comprimento de codificação das probabilidades condicionais para cada nó X_i é o produto do número de bits necessários para armazenar o valor numérico de cada probabilidade e o número total de probabilidades que são necessárias. Logo, o seguinte número de bits é necessário para codificar $\mathbf{\Theta}$ de uma RB

$$\sum_{i=n}^{n} d\,(r_i - 1)\,q_i$$

onde r_i é o número de estados do nó X_i, q_i é o número de configurações de seus pais e d é o número de bits necessários para armazenar o valor numérico de cada probabilidade.

É possível então definir que o comprimento de codificação de uma RB pode ser representado pelo seguinte número de bits

$$\sum_{i=n}^{n} |\mathbf{Pa}_i|\log_2(n) + d\,(r_i - 1)\,q_i$$

Seguindo o conceito para aplicação do DCM, deve-se codificar os dados dado um modelo e depois medi-lo em bits. Para a comparação desse comprimento de codificação, será utilizado, por enquanto, o método conhecido como *códigos de caracteres*. Esse método atribui, a cada configuração em um conjunto de dados B, uma string binária única. Logo, se B possui m casos, sua codificação será dada pela concatenação de m strings binárias.

O comprimento da string binária final pode ser minimizado com a substituição dos casos com maiores frequências por um código mais curto. Diante disso, Alcobé [46] utiliza o algoritmo de *Huffman* para computar o comprimento da string que codifica B dado um modelo. Esse algoritmo precisa que seja imputado a probabilidade de ocorrência de cada configuração c_i que aparece no conjunto de dados.

A probabilidade de cada configuração c_b ocorrer no domínio real é denotada por p_i. As probabilidades apresentadas pela RB utilizada são denotadas por θ_i. Essas probabilidades

são os valores mais próximos (pelo menos, teoricamente) do que acontece no domínio. Para cada configuração então, o algoritmo de Huffman atribui o código de comprimento aproximado $-\log_2(\theta_i)$. Logo, o comprimento da codificação string do conjunto de dados é, aproximadamente,

$$-m \sum_i p_i \log_2(\theta_i)$$

No entanto, existem dois problemas para a utilização desta medida. Um deles é a falta de conhecimento sobre o valor de p_i. O outro é a alta quantidade de configurações que precisam ser cobertas pela equação, dado o exponencial número de variáveis destacado na Seção 2.3.2. Friedman e Goldszmidt [21] e Lam e Bacchus [33] tentam solucionar tais problemas usando diferentes abordagens apresentadas nas Seções 3.4 e 3.6, respectivamente.

2.3.4 Informação Mútua Condicional

Como citado na seção anterior, alguns algoritmos de aprendizagem de RBs baseiam a construção da estrutura na aplicação de testes de IC entre as variáveis existentes na base de dados. Um dos conceitos bastante utilizados pelos algoritmos é o de *informação mútua*.

De maneira informal, pode-se definir que informação mútua é a medida da quantidade de informação que uma variável contém acerca de outra variável. Para a compreensão de como essa medida é calculada, é preciso entender a base sobre a divergência de Kullback-Leibler (ou entropia relativa).

Em resumo, a divergência de Kullback-Leibler $D_{KL}(P||Q)$ mede a ineficiência de assumir que uma distribuição é Q quando a distribuição verdadeira é P. Por exemplo, dado que existem informações sobre a distribuição de probabilidade verdadeira de uma variável, pode-se afirmar que o comprimento de codificação de uma configuração será igual a $H(P)$ (veja mais na Seção 2.3.3). No entanto, caso a distribuição Q presente no modelo de RB (próximo, mas não igual ao real) seja utilizada, por exemplo, haveria uma desadequação do comprimento codificação sendo necessários $H(P)$ + $D_{KL}(P||Q)$ bits.

Alcobé [46] apresenta uma definição para $D_{KL}(P||Q)$:

Definicão 5 (Divergência de Kullback-Leibler)**.** *A divergência de Kullback-Leibler é a seguinte medida de proximidade entre duas distribuições diferentes P e Q definidas sobre o*

mesmo espaço de evento

$$D_{KL}\left(P||Q\right) = \sum_{i}^{r_X} P\left(x_i\right) log \frac{P\left(x_i\right)}{Q\left(x_i\right)}$$

Seguindo as definições de entropia descritas em [4], pode-se afirmar que

$$D_{KL}\left(P||Q\right) = E_{P(x_i)}\left[\log \frac{P\left(X_i\right)}{Q\left(X_i\right)}\right]$$

Agora, conhecendo a definição da divergência de Kullback-Leibler, é possível definir, formalmente, informação mútua [4]:

Definicão 6 (Informação mútua)**.** *Considere duas variáveis aleatórias X_i e X_j com distribuição conjunta $P\left(x_i, x_j\right)$ e distribuições marginais $P\left(x_i\right)$ e $P\left(x_j\right)$. A informação mútua $I\left(X_i; X_j\right)$ é a divergência de Kullback-Leibler entre a distribuição conjunta e o produto das distribuições marginais.*

Logo,

$$\begin{aligned} I\left(X_i; X_j\right) &= \sum_{x_i \in X_i} \sum_{x_j \in X_j} P\left(x_i, x_j\right) \log \frac{P\left(x_i, x_j\right)}{P\left(x_i\right) P\left(x_j\right)} \\ &= D_{KL}\left(P\left(x_i, x_j\right) || P\left(x_i\right) P\left(x_j\right)\right) \\ &= E_{P(x_i, x_j)}\left[\log \frac{P\left(X_i, X_j\right)}{P\left(X_i\right) P\left(X_j\right)}\right] \end{aligned}$$

A condicionalidade entre variáveis também reflete na definição acima. Quando busca-se saber a informação mútua de duas variáveis dado uma terceira, defini-se esta informação mútua como *informação mútua condicional*. Barao [4] define como:

Definicão 7 (Informação mútua condicional)**.** *A informação mútua condicional das variáveis aleatórias X_i e X_j dado X_k é definida por*

$$I\left(X_i; X_j | X_k\right) = H\left(X_i | X_k\right) - H\left(X_i | X_j, X_k\right)$$

Considerando as definições de entropia descritas por Barao [4], $H\left(X_i | X_k\right)$ e $H\left(X_i | X_j, X_k\right)$ representam entropias condicionais. Logo,

$$I\left(X_i; X_j | X_k\right) = E_{P(x_i, x_j, x_k)}\left[\log \frac{P\left(X_i, X_j | X_k\right)}{P\left(X_i | X_k\right) P\left(X_j | X_k\right)}\right]$$

2.3.5 Estatísticas Suficientes

Até este ponto, sabe-se que o conhecimento pode ser representado de algumas formas, dentre elas, por conjuntos de dados. No entanto, existem domínios em que é necessário um grande número n de variáveis para representá-lo e armazenar informações sobre todas as relações seria altamente custoso [61]. Apesar disso, as RBs não modelam as relações entre instâncias (e sim, variáveis), reduzindo a quantidade de valores de probabilidades e parâmetros requeridos, como já citado na Seção 2.1. São armazenadas apenas informações suficientes para reproduzir, de forma aproximada, a distribuição de probabilidade abordada, isto é, fazer inferência sobre $\mathbf{\Theta}$. Essas informações são chamadas por *estatísticas suficientes*.

O conceito de estatísticas suficientes pode ser definido como [46]:

Definicão 8 (Estatísticas suficientes)**.** *Dado* $\mathbf{D}$ *para denotar um conjunto de variáveis randômicas cuja distribuição depende de um parâmetro* $\mathbf{\Theta}$*. Uma função vetorial* T *de* $\mathbf{D}$ *é dita suficiente se a distribuição condicional de* $\mathbf{D}$*, dado* $T = t$*, é independente de* $\mathbf{\Theta}$*.*

X_1	X_2	X_3
V	V	V
F	F	F
V	F	V
V	F	V
V	V	V
V	V	V

Tabela 2.2: Conjunto de dados exemplo

Na tentativa de descrever as estatísticas suficientes que um algoritmo precisa para aprender uma RB a partir de um base de dados B, denota-se $N^B_{\mathbf{D}}(\mathbf{d})$ para ser o número de instâncias em B onde $\mathbf{D} = \mathbf{d}$, dado que $\mathbf{D} = \mathbf{d}$ define o valor x_i assumido por cada variável X_i presente em D. Denota-se também $\hat{N}^B_{\mathbf{D}}(\mathbf{d})$ (ou, somente, $\hat{N}_{\mathbf{D}}$) para ser o vetor de números $N^B_{\mathbf{D}}(\mathbf{d})$ para todos os valores de $\mathbf{D}$. Logo, as estatísticas suficientes que são necessárias, inclusive para a medida padrão de qualidade de uma estrutura, resumem-se a $\hat{N}_{X,\mathbf{Pa}_X}$ para todo $X \in \mathbf{D}$ e o conjunto de seus possíveis pais, $\mathbf{Pa}_X$. As estatísticas suficientes para aprender uma estrutura de RB G serão denotadas então pelo conjunto $\mathbf{Suff}(\mathbf{G}) = \left\{\hat{N}_{X_i,\mathbf{Pa}_i} : 1 \leq i \leq n\right\}$.

O conjunto de dados descrito na Tabela 2.2 é usado como exemplo. Esses dados contêm informações sobre um domínio D, modelado pelo conjunto de variáveis randômicas $\mathbf{D} = \{X_1, X_2, X_3\}$, onde X_1 e X_2 são nós pai de X_3. Cada variável aleatória X_i pode assumir qualquer valor dentro do conjunto $\{"V", "F"\}$. No exemplo apresentado na Tabela 2.2, nota-se que ao ser feito o uso das estatísticas suficientes, não é preciso armazenar todos os dados apresentados. Tem-se então $\mathbf{Suff}(\mathbf{D}) = \left\{\hat{N}_{X_1}, \hat{N}_{X_2}, \hat{N}_{X_3,\mathbf{Pa}_3}\right\}$. Logo, $\hat{N}_{X_2} = \{N_{X_2} = "V", N_{X_2} = "F"\}$, por exemplo. Portanto, $\mathbf{Suff}(\mathbf{D}) = \{\{5,1\}, \{3,3\}, \{3,2,1\}\}$.

Capítulo 3

Aprendizagem Incremental de Estruturas de Redes Bayesianas

O conceito de aprendizagem incremental surgiu baseado na tentativa de reprodução do processo gradual de formação de conceito e de aquisição de conhecimento, considerando as novas experiências existente, nos humanos [46]. O desenvolvimento e aplicação desse método de aprendizagem foi impulsionado com a mudança de paradigma que forneceu aos vários domínios a capacidade de gerar e armazenar cada vez mais dados. Neste capítulo, a teoria utilizada na definição do processo de aprendizado incremental é discutida. Além disso, diversas soluções para aprendizado incremental de estruturas de RBs são identificadas e avaliadas através de um estudo sistemático. O foco é mantido na estrutura pela ineficiência dos parâmetros precisos se a estrutura não for representativa do domínio [46].

A organização do restante do capítulo é a seguinte. Na Seção 3.1, o conceito de aprendizagem incremental é definido. Na Seção 3.2, os métodos utilizados para a realização do estudo sistemático são descritos, e da Seção 3.3 até a Seção 3.8, as soluções encontradas na literatura são apresentadas como resultado do estudo. Na Seção 3.9, alguns dos principais resultados são detalhados.

3.1 Definição de Aprendizagem Incremental

A vasta aplicação de aprendizado incremental em alguns campos de pesquisa, como aprendizado de máquina [18] [23], permitiu o desenvolvimento de definições amplamente aceitas

pela comunidade científica.

Langley [34] define aprendiz incremental como:

Definicão 9 (Aprendiz incremental)**.** *Um aprendiz L é incremental somente se L insere uma experiência de treinamento de cada vez, não reprocessa experiências anteriores e mantém apenas uma estrutura de conhecimento na memória.*

Nesta definição, existem três restrições para que um algoritmo possa ser classificado como incremental. As duas primeiras objetivam dar a capacidade ao algoritmo de aprender a qualquer momento diante da inserção de um novo conhecimento. A terceira restrição possui foco na quantidade de memória necessária para o funcionamento do algoritmo. Friedman e Goldszmidt [21] apresentam outra definição com uma maneira diferente de manutenção do conhecimento:

Definicão 10 (Procedimento incremental)**.** *Um procedimento de aprendizagem de redes Bayesianas é incremental se a cada iteração l, ele recebe uma nova instância de dados u_l e produz uma próxima hipótese S_{l+1}. Essa estimativa é utilizada para executar a tarefa necessária na próxima instância u_{l+1}, que por sua vez é utilizada para atualizar a rede e assim por diante. O procedimento pode gerar um novo modelo depois que um número de k instâncias for coletado.*

Essa definição relaxa as restrições impostas pela definição de Langley [34]. Para Friedman e Goldszmidt [21], um algoritmo incremental pode processar, pelo menos, k instâncias anteriores após encontrar uma nova instância de treinamento ou manter k bases de conhecimento alternativas na memória.

No estudo de Domingos e Hulten [13], existe também outra definição baseada na definição de Langley [34]:

Definicão 11 (Algoritmo incremental)**.** *Um algoritmo incremental deve atender às seguintes restrições:*

- *deve exigir um tempo pequeno constante por registro;*
- *deve ser capaz de construir um modelo usando, no máximo, uma varredura dos dados;*
- *deve usar apenas uma quantidade fixa de memória principal, independentemente do número total de registros que tenha sido utilizado;*

- *deve disponibilizar um modelo utilizável em qualquer momento, em oposição a apenas quando é feito o processamento dos dados;*
- *deve produzir um modelo que seja equivalente (ou quase idêntico) àquele que seria obtido pelo algoritmo de lote correspondente.*

Como nas definições anteriores, esta definição impõe restrições relacionadas ao tempo, memória e conhecimento abordados. Ela incrementa a restrição relacionada à disponibilidade de um modelo útil imposto pela definição de Friedman e Goldszmidt [21]. Agora, muito devido à sua aplicação em fluxos de dados, é necessário disponibilizar um modelo utilizável em qualquer ponto do tempo, sendo o contrário válido somente quando é feito o processamento dos dados.

Com base nas definições acima de aprendizado incremental, são encontradas soluções que apresentam diferentes metodologias. Dois grupos separam essas soluções. A principal diferença entre eles está na forma em que o conhecimento adquirido é utilizado. Em um desses grupos, denotado por *procedimentos de refinamento incremental*, os dados são utilizados de acordo com o conhecimento já possuído. Esse conhecimento é mantido no gráfico probabilístico já desenvolvido, sendo apenas refinado com os novos dados.

Definicão 12 (Procedimento de refinamento incremental)**.** *Um procedimento incremental de aprendizagem de redes Bayesianas é considerado de refinamento se a cada iteração l, ele recebe uma nova instância de dados u_l, baseado no conhecimento codificado na estrutura G, e produz uma próxima hipótese S_{l+1}. Essa estimativa é utilizada para melhorar pontos específicos de G, produzindo um conhecimento semelhante ao existente.*

No outro grupo, indicado por soluções de adaptação estrutural, as soluções mantêm uma ou mais estruturas candidatas e aplicam, a estas estruturas, as observações recebidas. Este novo conjunto de dados é usado para atualizar as estatísticas suficientes necessárias para criar essas estruturas de candidatos.

Definicão 13 (Procedimento de adaptação incremental)**.** *Um procedimento incremental de aprendizagem de redes Bayesianas é considerado de adaptação se a cada iteração l, ele recebe uma nova instância de dados u_l, atualiza o conhecimento existente u_{l-1} e produz uma próxima hipótese S_{l+1}. Novas estruturas são desenvolvidas e selecionada a que melhor representa S_{l+1}.*

Os conceitos e informações sobre os tipos de soluções encontrados nesta pesquisa são mapeados, de forma esquemática, na Figura 3.1

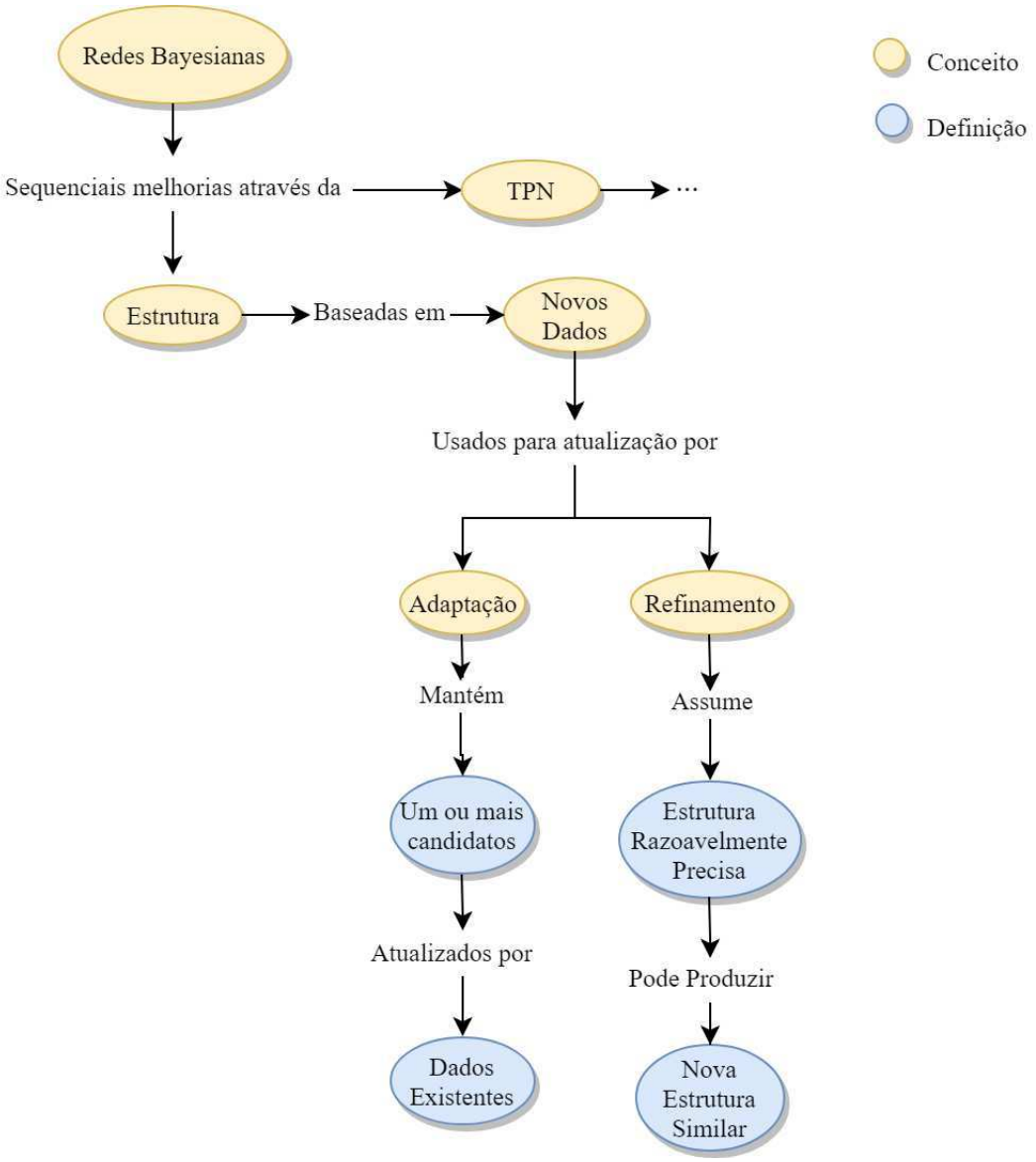

Figura 3.1: Mapa conceitual sobre definição de aprendizado incremental

3.2 Metodologia da Revisão Sistemática da Literatura

A Revisão Sistemática da Literatura (RSL) é realizada com o objetivo de identificar e avaliar soluções para o aprendizado incremental de estruturas de RBs, bem como para delinear direções de pesquisas futuras relacionadas (veja mais no Capítulo 5).

Durante as pesquisas de diretrizes para executar a RSL, vários estudos foram encontrados e, apesar de métodos semelhantes, existem diferenças quanto à ordem de execução de suas tarefas. Nesse sentido, o estudo realizado por Budgen e Brereton [7] é utilizado como base para a realização da RSL proposta. As etapas da revisão sistemática são resumidas a: (i) planejar; (ii) conduzir; e (iii) reportar. Nesta seção, o planejamento do estudo sistemático é detalhado.

Na fase de planejamento, o desenvolvimento do protocolo de revisão foi uma das tarefas realizadas. Ter um protocolo pré-definido é necessário para reduzir o viés do pesquisador, entre outros problemas relacionados à execução do estudo [7]. O protocolo desenvolvido inclui, principalmente, quatro elementos: (i) as questões de pesquisa que o estudo pretende responder; (ii) a estratégia adotada para a busca de soluções primárias; (iii) os procedimentos de seleção de soluções; e (iv) os métodos para a avaliação da qualidade de uma solução. A sequência de definição dos elementos citados é apresentada na Figura 3.2.

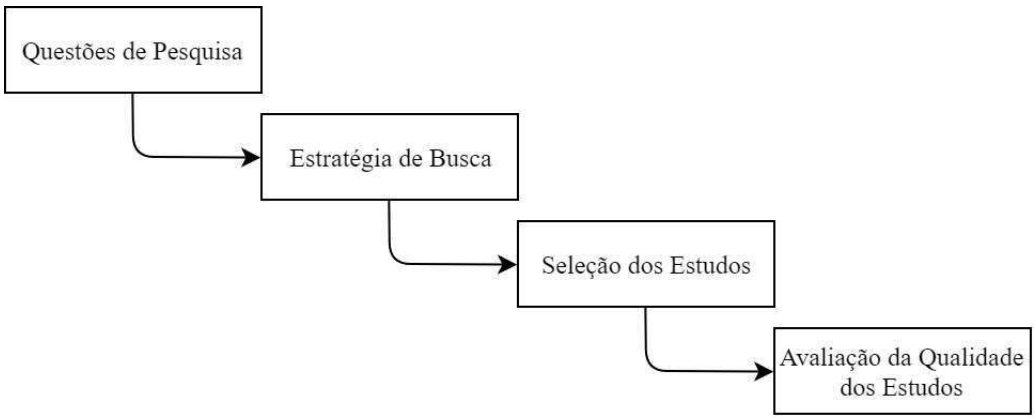

Figura 3.2: Metodologia para a definição do protocolo de revisão

Uma combinação de strings baseada nas perguntas de pesquisa foi aplicada somente no Scopus[1] para o processo de busca. Inicialmente, a busca foi realizada apenas em títulos e palavras-chave. Não foram utilizadas mais restrições neste processo.

Na busca inicial, 4.150 itens foram encontrados, mas apenas os primeiros 400 resultados foram verificados. Baseado no método utilizado por Zhou e Mäntylä [62], essa parada foi realizada porque, dentre os primeiros 400 resultados, classificados por relevância, foram encontrados uma sequência de 150 artigos totalmente não relacionados à busca.

[1] https://www.scopus.com/home.uri

Para verificar essa relação, três etapas de leitura foram realizadas. Inicialmente, os itens foram selecionados considerando apenas o título e o resumo. Uma leitura superficial dos itens restantes foi então realizada. Esta etapa consistiu em ler e interpretar títulos de seções, figuras, gráficos, conclusões e outros elementos. Nos itens que ainda mantinham um grau de relação com o tema, foi realizada uma leitura crítica buscando interpretar e analisar o texto completo. *Snowballing* e seu reverso também foram usados para complementar o processo de seleção de itens e reduzir a ameaça de validade de soluções relevantes em falta. Para serem considerados adequados para a aplicação desta técnica, as três etapas de leitura citadas anteriormente foram aplicadas novamente nos novos itens encontrados.

3.2.1 Ameaças à validade

Uma das principais ameaças à validade deste estudo sistemático é um possível viés durante a seleção de documentos. Idealmente, este processo deve ser realizado por dois ou mais pesquisadores. No entanto, este estudo foi realizado apenas por um pesquisador, abrindo a possibilidade de tomada de decisões distorcidas.

Outra ameaça possível é a imprecisão na extração de dados. A combinação de strings foi construída com base nas questões de pesquisa. Entretanto, itens relevantes podem não usar os termos relacionados às questões em seus títulos ou palavras-chave, sendo excluídos do escopo do estudo. *Snowballing* foi usado para tentar diminuir o número desses itens, no entanto, entende-se que alguns podem ter sido deixados de fora.

Uma descrição destes esforços é apresentada nas próximas seções.

3.3 Algoritmo de Buntine (B)

Buntine [8] propõe uma abordagem baseada em refinamento teórico. A principal tarefa do refinamento teórico é atualizar a teoria parcial inicial, geralmente o conhecimento prévio do especialista, na medida em que novos casos produzem um conhecimento posterior sobre o espaço das teorias possíveis. Tal tarefa é, como discutido, é um dos fundamentos da aprendizagem contínua. Dado um conjunto de novos dados e ordenação total das variáveis de domínio, a solução atualiza tanto o conhecimento sobre a estrutura, quanto os parâmetros, usando diferentes RBs.

Para estender e modificar a estrutura, Buntine [8] propôs um algoritmo de lote que usa a abordagem Bayesiana baseada em pontuação e pesquisa. No entanto, usando algumas diretrizes apresentadas pelo autor, é possível converter a aprendizagem em lote em processo de aprendizagem incremental.

O algoritmo de lote requer um conjunto de variáveis ordenadas de acordo com o conhecimento a priori do domínio, $\mathbf{X} = \{X_1, ..., X_n | X_i \prec X_{i+1}, ..., X_n\}$, onde, para o especialista, as variáveis que vêm primeiro têm influência sobre as outras. Para cada variável X_i, um conjunto de conjuntos de pais alternativos $\prod_{\mathbf{i}} = \{\mathbf{Pa}_{i1}, ..., \mathbf{Pa}_{im}\}$ é mantido de acordo com alguns critérios de razoabilidade. Cada conjunto de pais $\mathbf{Pa}_{ij}$ é um subconjunto de $\{Y | Y \prec X_i\}$.

Um conjunto de conjuntos de pais alternativos $\prod_{\mathbf{i}}$ para a variável X_i é denotado por *estrutura pai* para X_i. Essa estrutura pai é uma estrutura de rede em que subconjuntos e superconjuntos de conjuntos de pais são interligados. Para facilitar o acesso a todos conjuntos de pais alternativos $\mathbf{Pa}_j \in \prod_{\mathbf{i}}$ eficientemente, apenas os conjuntos de pais com probabilidades a posteriori significativas são armazenados na estrutura pai para X_i. O nó raiz da estrutura pai para X_i é um conjunto vazio e as folhas são os conjuntos $\mathbf{Pa}_j$ que não possuem superconjuntos contidos em $\prod_{\mathbf{i}}$.

O algoritmo em lote também requer três parâmetros $1 > C > D > E$. Estes são usados para variar e guiar a pesquisa. O algoritmo usa esses parâmetros como base para classificar os conjuntos de pais como *Alive*, *Asleep* ou *Dead*. O parâmetro C é usado para separar os conjuntos de pais que finalmente participam do espaço de redes alternativas. O parâmetro D é usado para selecionar as alternativas razoáveis para os conjuntos de pais *Alive*. As estruturas alternativas do conjuntos de pais *Alive* são os *beams* pesquisados pelo algoritmo. O parâmetro E é usado para selecionar as alternativas razoáveis para os conjuntos de pais *Dead*. Conjuntos de pais *Dead* são alternativas que são exploradas e decididas, inevitavelmente, como alternativas não razoáveis e não devem ser exploradas. Por outro lado, os conjuntos de pais *Asleep* são semelhantes, mas só são desconsiderados por enquanto e podem se tornar *Alive* mais tarde.

Definir os três parâmetros C, D e E com o valor 1 fará com que o algoritmo seja reduzido para o algoritmo K2. Para isso, muitos pesquisadores citam esse algoritmo como uma generalização do algoritmo K2 [46, 34, 47]. No final, uma estrutura de redes alternativas

resulta do conjunto de conjuntos de pais e dos parâmetros de rede, denotada por uma RB combinada. O pseudo-código do algoritmo em lote definido por Buntine [8] é apresentado no Código Fonte 3.1 e no Código Fonte 3.2.

Código Fonte 3.1: B em Lote

1: **função** BUNTINE(base de dados D sobre $\mathbf{X} = \{X_1, ..., X_n\}$, uma ordem $\prec$ entre as variáveis, parâmetros C, D e E)
2: Calcule $\mathbf{Suff}\ (\tau)$
3: **para** i $\leftarrow$ 1 **até** n **faça**
4: *Melhor-posterior* = $P\left(X_i | Pa_i = \oslash, D, \prec\right)$
5: *Lista-aberta* = $\{\oslash\}$
6: *Lista-alive* = $\{\oslash\}$
7: **enquanto faça**
8: Pegue o conjunto de pai Pa_i do topo da *Lista-aberta*
9: **se** $P\left(X_i | Pa_i, D, \prec\right) < E \cdot Melhor - posterior$ **então**
10: Marque Pa_i como *dead*
11: **senão**
12: **se** $P\left(X_i | Pa_i, D, \prec\right) > D \cdot Melhor - posterior$ **então**
13: Marque todos os filhos ch e calcule a probabilidade posterior
14: Chame *MarkChildren* $(\mathbf{CH_{Pa_i}})$
15: **senão**
16: Ignore Pa_i
17: **fim se**
18: **fim se**
19: **fim enquanto**
20: Lista-aberta = $\oslash$
21: **fim para**
22: **fim função**

Código Fonte 3.2: MarckChildren (B)

1: **função** MARKCHILDREN(base de dados D sobre $\mathbf{X} = \{X_1, ..., X_n\}$, parâmetros C, D

e E e um conjunto de filhos $\mathbf{CH_{Pa_i}}$)

para i $\leftarrow$ 1 **até** tamanho de $\mathbf{CH_{Pa_i}}$ **faça**

  ch on position i on $\mathbf{CH_{Pa_i}}$

  se $P\left(X_i|Pa_i, D, \prec\right) >$ *Melhor-posterior* **então**

    $Melhor - posterior = P\left(X_i|Pa_i, D, \prec\right)$

    Modifique a lista-alive para refletir o novo máximo

  senão

  fim se

  se $P\left(X_i|Pa_i, D, \prec\right) < E\cdot$ *Melhor-posterior* **então**

    Marque ch como $dead$

  senão

  fim se

  se $P\left(X_i|Pa_i, D, \prec\right) > D\cdot$ *Melhor-posterior* **então**

    Adicione ch para a lista-aberta

  senão

  fim se

  se $P\left(X_i|Pa_i, D, \prec\right) > C\cdot$ *Melhor-posterior* **então**

    Marque ch como $alive$ e adicione ch para a lista-alive

  senão

  fim se

fim para

fim função

Para converter esse algoritmo em lote em um algoritmo de aprendizado incremental, Buntine [8] descreve duas situações que variam de acordo com o tempo disponível para a atualização. No caso em que há um curto período de tempo para atualizar as RBs, o algoritmo apenas atualiza as probabilidades a posteriori das estruturas pai. Para isso, é necessário armazenar probabilidades posteriores e os contadores N_{ijk} para cada conjunto alternativo de conjuntos de pais *Alive*.

Por outro lado, tanto a estrutura quanto as probabilidades a posteriori são atualizadas de acordo com novos dados. Para cada variável X_i da RB combinada, o algoritmo precisa: (i) atualizar as probabilidades a posteriori de todos os conjuntos *Alive* da estrutura pai; (ii)

calcular o novo melhor-posterior; e (iii) expandir os nós da *Lista-Aberta* e continuar com a pesquisa.

A geração de diferentes redes precisa atualizar as probabilidades a posteriori de todos os conjuntos *Alive* da estrutura pai. Essa solução usa estatísticas suficientes de dados, exigindo tempo constante para atualizar estas estatísticas apenas quando novos registros chegarem. Além disso, Buntine [8] realiza uma pesquisa adicional sobre o espaço de RBs alternativas.

3.4 Algoritmo de Friedman-Goldszmidt (FG)

Como a solução anterior, Friedman e Goldszmidt [21] também abordaram o problema da atualização sequencial do conhecimento do domínio anterior. Através do uso de estatísticas suficientes mantidas na memória para cada estrutura de rede em uma fronteira definida, o conhecimento é continuamente incrementado. Deste modo, esta solução proporciona um método que possui em *tradeoff* entre a precisão, isto é, a qualidade da estrutura, e o armazenamento, isto é, a quantidade de informação sobre as observações anteriores.

Friedman e Goldszmidt [21] propõem três soluções diferentes para aprender sequencialmente RBs. Entre elas, dois extremos. A *abordagem ingênua*, como é chamada, armazena todos os dados vistos anteriormente e invoca repetidamente um procedimento de aprendizado em lote após cada nova observação ser registrada. No entanto, apesar de usar o máximo de informações possível, aumentando assim a qualidade da estrutura gerada, essa abordagem tem um alto custo de armazenamento. Além disso, a reutilização da aprendizagem em lote aumenta a quantidade de tempo e o processamento gasto.

Por outro lado, a abordagem Máxima Probabilidade a Posteriori (MPP) usa um modelo para armazenar todas as informações consideradas úteis para as próximas etapas da atualização de conhecimento. No entanto, o uso de um único modelo pode influenciar fortemente a aprendizagem contínua do modelo e perder informações.

Consciente das desvantagens das abordagens anteriores, Friedman e Goldszmidt [21] apresentam uma nova abordagem, denominada *incremental*, que propõe um *tradeoff* entre os extremos. A abordagem incremental não armazena todos os dados, ao contrário da abordagem ingênua, e não usa uma rede para representar o conhecimento a priori, ao contrário da abordagem MPP. Além disso, permite escolhas flexíveis no *tradeoff* entre espaço e qualidade

das redes induzidas.

O componente básico deste procedimento é um módulo que mantém um conjunto S de registros de estatísticas suficientes. O conjunto de estatísticas suficientes para G, denotado por $\mathbf{Suff}(\mathbf{G})$, pode ser alimentado por $\mathbf{Suff}(\mathbf{G}) = \{N_{X_i,\mathbf{Pa}_i} : 1 \leq i \leq n\}$. Da mesma forma, dado um conjunto S de registros de estatísticas suficientes, o conjunto de estruturas de rede, denotado por $\mathbf{Nets}(\mathbf{S})$, pode ser avaliado usando os registros em S por $\mathbf{Nets}(\mathbf{S}) = \{G : \mathbf{Suff}(\mathbf{G}) \subseteq S\}$.

Duas estruturas podem ser acompanhadas facilmente mantendo um conjunto de estatísticas ligeiramente maior. Por exemplo, supondo a escolha deliberada entre duas estruturas G e $G^{'}$. Para avaliar G utilizando uma função de pontuação, é necessário manter o conjunto $\mathbf{Suff}(\mathbf{G})$. Por outro lado, para avaliar $G^{'}$, é necessário manter o conjunto $\mathbf{Suff}(\mathbf{G}^{'})$. Agora, supondo que G e $G^{'}$ diferem apenas por um arco de X_i a X_j, nota-se que existe uma grande sobreposição entre $\mathbf{Suff}(\mathbf{G})$ e $\mathbf{Suff}(\mathbf{G}^{'})$. Ou seja, $\mathbf{Suff}(\mathbf{G}) \cup \mathbf{Suff}(\mathbf{G}^{'}) = \mathbf{Suff}(\mathbf{G}) \cup \left\{N_{X_j,\mathbf{Pa}_j}\right\}$, onde $\mathbf{Pa}_j$ é o conjunto pai de X_j em $G^{'}$. Esse argumento pode ser útil quando se considera o uso do procedimento de busca HCS, por exemplo. Observa-se também que é possível avaliar o conjunto de vizinhos de S mantendo um conjunto limitado de estatísticas suficientes.

Generalizando essa discussão, a abordagem incremental pode ser aplicada a qualquer procedimento de pesquisa que possa definir uma fronteira de pesquisa. Esta fronteira, denotada por F, consiste em todas as redes que serão comparadas na próxima iteração. A escolha de F determina quais estatísticas suficientes são mantidas na memória. Depois que uma nova instância é recebida (ou, em geral, após um número de novas instâncias recebidas), o procedimento usa a estatística suficiente em S para avaliar e selecionar a rede de melhor pontuação na fronteira F (ou em $\mathbf{Nets}(\mathbf{S})$). O pseudo-código para abordagem incremental descrito por Friedman e Goldszmidt [21] é apresentado no Código Fonte 3.3.

Código Fonte 3.3: FG

1: Atribua a rede inicial à variável S
2: Dado F para sera fronteira de S
3: Dado $ST = Suff\,(S) \cup U_{S^{'} \in F} Suff\left(S^{'}\right)$
4: **para** i $\leftarrow$ 1 **até** ∞ **faça**

5: Leia a base de dados u_l

6: Atualize ST usando u_l

7: **se** *n mod k = 0* **então**

8: $S = argmax_{S' \in Redes(S')} Score\left(S'|ST\right)$

9: Atualize a fronteira F utilizando alguma heurística de busca

10: $S = Suff\left(S\right) \cup U_{S' \in F} Suff\left(S'\right)$

11: **senão**

12: **fim se**

13: Compute valores para S usando ST

14: **fim para**

Muitas funções de pontuação podem ser usadas para avaliar a adequação das redes em relação aos dados de treinamento (veja mais na Seção 2.3.3) e, em seguida, procurar a melhor rede. No entanto, esta solução incremental coleta estatísticas suficientes em diferentes momentos do processo de aprendizagem. Assim, é preciso comparar as RBs em relação aos diferentes conjuntos de dados. Este problema acontece porque, ao contrário de Buntine [8], Friedman e Goldszmidt [21] consideraram aquelas estruturas que, anteriormente, eram consideradas não promissoras (as que estavam fora da fronteira).

As duas principais funções de pontuação comumente usadas para aprender RBs, pontuação Bayesiana [11] e DCM [32] são inadequadas para esse problema. A fim de superar este problema, Friedman e Goldszmidt [21] propuseram uma medida DCM média $S'_{DCM}(G|D) = S_{DCM}(G|D)/N$, onde N é o número de instâncias do conjunto de dados. Essa pontuação mede o comprimento médio de codificação por instância.

Para superar o problema na utilização do DCM para calcular o comprimento de codificação dos dados descrito na Seção 2.3.3, Friedman e Goldszmidt [21] realizaram algumas extensões na equação presente na seção citada. A extensão pode ser descrita como

$$-m \sum_i p_i \log_2 \left(\theta_i\right)$$

$$-m \sum_i p_i \log_2 \prod_u P\left(x_u|\mathbf{Pa}_u\right)$$

$$-\sum_i \sum_{x_i,\mathbf{Pa}_i} N\left(x_i|\mathbf{Pa}_i\right) \log_2 P\left(x_i|\mathbf{Pa}_i\right)$$

onde $N\left(x_i|\mathbf{Pa}_i\right)$ representa o número de casos no conjunto de dados B em que $X_i = x_i$ e $\mathbf{Pa}_i = \mathbf{pa}_i$.

Analisando tanto a solução de Buntine [8], quanto a de Friedman e Goldszmidt [21], como soluções que utilizam *hill climbing*, elas executam, para cada nó, operações para aumentar a pontuação da estrutura resultante, sem introduzir um ciclo na rede e com base na suposição de que eles partem de uma rede sem arco, como pode ser observado em seus pseudocódigos. Ambos param a busca quando não conseguem realizar uma única operação que aumente a pontuação da rede. A diferença entre as duas soluções citadas é a composição da vizinhança. Enquanto Buntine [8] usa apenas o operador de adição para construir vizinhos, Friedman e Goldszmidt [21] usam a adição, reversão e exclusão de um arco.

3.5 Algoritmo de Alcobé (R)

As conversões das abordagens de aprendizado em lote realizadas por Buntine [8] e Friedman e Goldszmidt [21] em abordagens de aprendizado incremental abriram o caminho para algoritmos de lote populares como os algoritmos B, K2 [11] e HCMC [30] serem transformados em algoritmos incrementais [46].

Alcobé [2] propõe duas heurísticas para alterar o algoritmo de busca HCS em um algoritmo incremental baseado na combinação de estatísticas suficientes com espaço de pesquisa reduzido. Na versão em lote do algoritmo HCS, uma pesquisa no espaço de busca é realizada para examinar todas as possíveis alterações locais que podem ser feitas para maximizar a função de pontuação.

Relembrando a definição citada em 2.3.1, uma vizinhança de um modelo B, semelhante à fronteira apresentada por Friedman e Goldszmidt [21], consiste em todos os modelos que podem ser construídos usando pares de um determinado argumentos A e um único operador de um conjunto de operadores $\mathbf{OP} = \{"B", "C", "D"\}$, onde B, C e D podem ser a adição, remoção ou reversão de um arco, respectivamente. Levando isso em conta, a sequência de pares de operadores e argumentos adicionados para obter o modelo G_f é denotada por caminho de pesquisa. Seja G_0 um modelo inicial, um modelo final obtido por um HCS pode ser descrito por $G_f = op_n\left(...\left(op_1, A_1\right), ...\right), A_n$, onde o caminho de busca $O_{op} = \{\left(op_1, A_1\right), ..., \left(op_n, A_n\right)\}$ é usado para construir G_f.

As heurísticas apresentadas por Alcobé [2] baseiam-se em dois problemas principais: (i) quando e qual parte atualizar; e (ii) como calcular e armazenar estatísticas suficientes. A

primeira heurística desenvolvida para solucionar o primeiro problema é chamada por Operadores de Percurso na Ordem Correta (Operadores Transversais na Ordem Correta - OTOC).

Em resumo, OTOC verifica quando uma estrutura necessita ser atualizada através da análise do caminho de pesquisa do modelo. Sempre que novas instâncias de dados estão disponíveis, OTOC calcula a pontuação de cada par de operação de argumento presente no caminho de busca e verifica se ainda estão na ordem correta de classificação. Nota-se que, durante a geração do caminho de pesquisa pelo HCS utilizando a função de pontuação, $S\left(op_1^k\left(B_0, A_1\right), D\right) < S\left(op_2^k\left(B_1, A_2\right), D\right) < \ldots < S\left(op_i^k\left(B_{i-1}, A_i\right), D\right)$. Caso esta ordem não se mantenha, OTOC entende que a estrutura precisa ser revisada e permite que HCS realize sua busca. Formalmente, OTOC pode ser definido por [46]:

Definicão 14 (Heurística OTOC)**.** *Dado D para ser um conjunto de dados, B para ser um modelo aprendido utilizando o HCS. Dado $D^{'}$ para ser um novo conjunto de dados, a heurística OTOC afirma que HCS aprenda um novo modelo $B^{'}$ correspondente a $\left\{D \cup D^{'}\right\}$ como*

$$B^{'} = op_{n^{'}}^{k^{'}{}_{n^{'}}}\left(op_{n^{'}-1}^{k^{'}{}_{n^{'}-1}} \ldots \left(op_{1^{'}}^{k^{'}{}_{1^{'}}}\left(B_{ini}, A_{1^{'}}\right) \ldots, A_{n^{'}-1}\right) A_{n^{'}}\right)$$

onde os operadores e argumentos são obtidos por

$$\forall i \in \left[1, n^{'}\right] : \left(op_i^{k_i^{'}}, A_i\right) = argmax_{\left(op_i^{k_i^{'}}, A_i\right) \in \mathbf{OpA_N(B_{i-1})}} S\left(op_i^{k_i^{'}}\left(B_{i-1}, A_i\right), \left\{D \cup D^{'}\right\}\right)$$

e onde o modelo inicial B_{ini} é

$$B_{ini} = op_j^{k_j}\left(op_{j-1}^{k_{j-1}} \ldots \left(op_1^{k_1}\left(B_0, A_1\right) \ldots, A_{j-1}\right) A_j\right)$$

onde $\left(op_j^{k_j}, A_j\right)$ é o último par de operador e argumento corretamente ordenado. Isto é, o último par que produz o modelo com maior pontuação entre os pares do caminho de busca Oop.

A segunda heurística é chamada de Espaço de Busca Reduzido (Espaço de Busca Reduzido - EBR). EBR aplica-se quando a estrutura atual precisa ser revisada. Em cada etapa do caminho de busca, esta heurística armazena os $nRss$ modelos com a pontuação mais próxima ao modelo de maior pontuação naquela etapa do processo de aprendizagem em um conjunto denotado por **B**. O conjunto B então reduz o espaço de pesquisa, evitando explorar as partes onde modelos de baixa qualidade foram encontrados durante as etapas de pesquisa anteriores. EBR pode ser definida formalmente como [46]:

Definição 15 (Heurística EBR)**.** *Dado D para ser um conjunto de dados, B para ser um modelo aprendido utilizando o HCS. Dado $\forall i \in [1,n] : \mathbf{B}_{\left(op_i^{k_i}, A_i\right)}$ seja um conjunto de k pares de operadores e argumentos com a pontuação mais próxima de $S\left(op_i^{k_i}\left(B_{i-1}, A_i\right), D\right)$. Dado $D^{'}$ para ser um novo conjunto de dados, a heurística EBR afirma que HCS aprenda um novo modelo $B^{'}$ correspondente a $\left\{D \cup D^{'}\right\}$ como*

$$B^{'} = op_{n^{'}}^{k^{'}_{n^{'}}}\left(op_{n^{'}-1}^{k^{'}_{n^{'}-1}} ... \left(op_{1^{'}}^{k^{'}_{1^{'}}}\left(B_{ini}, A_{1^{'}}\right) ..., A_{n^{'}-1}\right) A_{n^{'}}\right)$$

onde os operadores e argumentos são obtidos usando $\mathbf{B}_{\left(op_i^{k_i}, A_i\right)}$

$$\forall i \in \left[1, n^{'}\right] : \left(op_i^{k^{'}_i}, A_i\right) =$$

$$argmax_{\left(op_i^{k^{'}_i}, A_i\right) \in \mathbf{OpA_N(B_{i-1}) \cap B}_{\left(op_i^{k_i}, A_i\right)}} S\left(op_i^{k^{'}_i}\left(B_{i-1}, A_i\right), \left\{D \cup D^{'}\right\}\right)$$

O pseudo-código para o algoritmo prosposto por Alcobé [2] é apresentado no Código Fonte 3.4.

Código Fonte 3.4: IHCS

1: **função** IHCS(um domínio D, um modelo inicial B_0, um conjunto de operadores $\mathbf{OP}$, um conjunto de dados $\mathbf{D}$, uma função de pontuação $S\left(B, \mathbf{D}\right)$, o caminho de pesquisa Oop usado para gerar o modelo anterior B_{for}, um conjunto de pares de operadores e argumentos com a k melhor pontuação no i_{th} passo $\mathbf{B}_{\left(\mathbf{op_i^{k^{'}}, A_i}\right)}$ e um inteiro $q \leq k$ que afirma o número de pares que serão usado de $\mathbf{B}$)

2: Dado $op_j^{k_j}, A_j$ para ser o último par ordenado corretamente em Oop

3: $B_{ini} = op_j^{k_j}\left(op_{j-1}^{k_{j-1}} ... \left(op_1^{k_1}\left(B_0, A_1\right) ..., A_{j-1}\right) A_j\right)$

4: Dado $ST = Suff\left(S\right) \cup U_{S^{'} \in F} Suff\left(S^{'}\right)$

5: **se** $B_{ini} = B_{for}$ **então**

6: Use $q = k$

7: **senão**

8: **fim se**

9: $i = 0$; $B_i = B_{ini}$

10: **enquanto** pontuaçãoAntiga $< S = (B_i, \mathbf{D})$ **faça**

11: pontuaçãoAntiga **=** $S = (B_i, \mathbf{D})$

12: $i = i + 1$

13: $B_i = op_i^{k_i}\left(B_{i-1}, A_i\right)$ onde

14: $\left(op_i^{k_i}, A_i\right) = argmax_{\left(op^k, A\right) \cap \mathbf{B}_{\left(op_i^{k_i}, A_i\right)}} S\left(op_i^{k_i}\left(B_{i-1}, A_i\right), \mathbf{D}\right)$

15: **se** $B_{ini} \neq B_{for}$ **então**

16: Calcular $\mathbf{B}_{\left(\mathbf{op_i^{k'}, A_i}\right)}$

17: **senão**

18: **fim se**

19: **fim enquanto**

20: **fim função**

3.6 Algoritmo de Lam-Bacchus (LB)

Lam e Bacchus [33] apresentam outra solução de aprendizado incremental baseada em uma extensão de solução em lote. Esta solução em lote é apresentada por Lam e Bacchus [32], no entanto, não será apresentada aqui porque a nova solução não está acoplada ao seu algoritmo de lote. A extensão proposta visa realizar uma revisão da estrutura da RB incrementalmente, pois novos dados sobre um subconjunto de variáveis passam a estar disponíveis.

Essa revisão é feita usando a estrutura da RB como probabilidade a priori sob a suposição implícita de que a rede existente já é um modelo razoavelmente preciso do banco de dados. Essa suposição é uma maneira de incorporar o conhecimento de domínio no problema, no entanto, a nova estrutura de rede refinada deve ser semelhante à existente, distorcendo o processo.

Lam e Bacchus [33] também provam, como Friedman e Goldszmidt [21] e com base no DCM, que se a estrutura de rede parcial obtiver, mudando sua topologia, melhor pontuação para a função de pontuação, toda a estrutura de rede será melhorada se nenhum ciclo é introduzido. Com base nisso, Lam e Bacchus [33] desenvolveram um algoritmo para atualizar a RB, melhorando partes a cada vez ao invés do todo. Este algoritmo produz uma nova estrutura de rede parcial baseada em um novo conjunto de dados e a rede existente usando uma extensão do DCM. Depois, modifica localmente a estrutura antiga comparando e alterando a parte correspondente de acordo com a nova rede parcial.

A fonte de dados para o algoritmo consiste de dois componentes: (i) os novos dados;

e (ii) a estrutura de rede existente. Então é preciso encontrar uma estrutura parcial G_p que minimize a soma do comprimento da codificação:

- da estrutura parcial G_p;
- dos novos dados dado a estrutura G_p e;
- da estrutura existente dada a rede G_p.

Para calcular o comprimento de codificação do primeiro item, Lam e Bacchus [33] utilizam a mesma ideia descrita na Seção 2.3.3. Para calcular o segundo item, na tentativa de superar o problema descrito na mesma seção, Lam e Bacchus [33] relacionam o comprimento da codificação dos dados à divergência de Kullback-Leibler e depois mostraram que esta divergência pode ser substituída pela soma da informação mútua de cada variável e seu conjunto de pais.

Na tentativa de descrever a relação citada acima, Lam e Bacchus [33] desenvolveram o seguinte teorema:

Teorema 1. *O comprimento de codificação dos dados é uma função monotonicamente crescente da divergência de Kullback-Leibler entre a distribuição definida pelo modelo e a distribuição verdadeira.*

Baseado no teorema de Gibbs, Lam e Bacchus [33] explicam como a divergência de Kullback-Leibler pode substituir a equação apresentada na Seção 2.3.3 (veja mais em [33]). Ao analisar a definição da divergência de Kullback-Leibler na Seção 2.3.4, pode-se notar que o problema não é resolvido somente com a substituição.

Lam e Bacchus [33] então estendem o teorema que relaciona informação mútua entre dois nós e a divergência de Kullback-Leibler apresentada por Chow e Liu [9] através da especificação para cada nó X_i da informação mútua entre ele e o seu conjunto de pais $\mathbf{Pa}_i$ dado por

$$I\left(X_i; \mathbf{Pa}_i\right) = \sum_{x_i \in X_i, \mathbf{pa}_i \in \mathbf{Pa}_i} P(x_i, \mathbf{pa}_i) \log \frac{P\left(x_i, \mathbf{pa}_i\right)}{P\left(x_i\right) P\left(\mathbf{pa}_i\right)}$$

E apresentam o seguinte teorema, solucionando o problema citado na Seção 2.3.3:

Teorema 2. *A divergência de Kullback-Leibler $D_{KL}(P||Q)$ é uma função monotonicamente decrescente de*

$$\sum_{i=1,\mathbf{Pa}_i\neq\emptyset}^{n} I(X_i;\mathbf{Pa}_i)$$

Consequentemente, ela será minimizada se, e somente se, a soma acima for maximizada.

Para calcular o comprimento da codificação do terceiro item, é necessário calcular a descrição da estrutura existente completa G dada a rede G_p, isto é, descrever as diferenças entre G e G_p. Essas diferenças são descritas por uma lista de arcos: (i) invertidos; (ii) adicionais de G; e (iii) ausentes de G.

Uma maneira simples de codificar um arco é descrever o nó de origem e o nó de destino. $2\log n$ bits são necessários para descrever um arco, uma vez que é necessário $\log n$ para identificar apenas um, desde que existam n nós. Seja r, a e m, respectivamente, o número de arcos invertidos, adicionados e ausentes em G com respeito a G_p, o comprimento de descrição G, dado que a rede G_p é $(r+a+m)\,2\log n$.

Para aprender a estrutura local, o algoritmo de lote proposto por Lam e Bacchus [33] ou outro algoritmo pode usar a seguinte função de pontuação para cada nó da estrutura parcial:

$$DL_i = |\mathbf{Pa}_i| + \log n + \sum_{X_i\in\mathbf{Pa}_i} I(X_i,X_j) + (r_i+a_i+m_i)\,2\log n$$

Com o terceiro termo da equação, Lam e Bacchus [33] evitam usar a estatística suficiente dos dados antigos.

Depois que a nova estrutura parcial é aprendida, o processo de revisão continua com a tentativa de obter uma estrutura refinada de menor comprimento total de descrição com a ajuda da estrutura existente G e da estrutura parcial G_p. O problema de revisão agora é reduzido à escolha de subgráficos apropriados, denotados pelo subgráfico marcado, para o qual deve-se realizar a substituição do pai para obter uma estrutura refinada de menor comprimento de descrição total.

Em uma tentativa de evitar a criação de ciclos durante cada substituição de subgráficos, Lam e Bacchus [33] usam a busca *best-first* para encontrar o conjunto de unidades de subgráficos que produz a melhor redução no comprimento de descrição sem gerar nenhum ciclo. Além disso, uma lista $S = \{S_1,...,S_n\}$ contém uma classificação de todos os subgráficos em ordem crescente do benefício obtido.

3.7 Algoritmo de Shi-Tan (ST)

Shi e Tan [50] apresentam um eficiente algoritmo de aprendizado incremental híbrido. Todas as soluções apresentadas até agora são soluções baseadas em pontuação e pesquisa. Esta solução consiste em uma técnica baseada em restrições de tempo polinomial e o procedimento de busca HCS com um domínio D baseado nessas restrições. Dessa forma, essa solução fornece um algoritmo híbrido que oferece consideráveis economias de complexidade computacional e precisão do modelo ligeiramente melhor em dados complexos, como os dados sobre um contexto real.

A primeira parte da solução é composta de uma técnica baseada em restrições. O objetivo desta técnica é selecionar os pais candidatos para cada variável. Para cada variável X_i, um conjunto de pais candidatos $\mathbf{S_{X_i}}$ é, inicialmente, configurado contendo todas as outras variáveis do modelo. Se a variável X_j for independente de X_i condicionada em algum conjunto de variáveis $\mathbf{C}$ em procedimentos de aprendizado anteriores, o algoritmo fará o teste de IC e removerá X_j de $\mathbf{S_{X_i}}$ se a IC ainda persistir.

Depois disso, um procedimento heurístico chamado HeuristicIND é proposto para reduzir ainda mais $\mathbf{S_{X_i}}$. Este procedimento tenta descobrir um conjunto de variáveis para separar X_i e X_j condicionalmente usando a estrutura de rede atual e um esqueleto não direcionado em forma de árvore gerado baseado nos dados e usando Chow e Liu [9].

Shi e Tan [50] desenvolvem um novo método para realizar os testes de IC e medir a associação entre as variáveis ao construir o *skeleton*. Esse novo método faz uso de uma das propriedades fundamentais de informação mútua (veja mais na Seção 2.3.4) que é expressado no seguinte teorema [50]:

Teorema 3. *Dado um conjunto de dados D com n instâncias, se a hipótese de que $\mathbf{X}$ e $\mathbf{Y}$ são condicionalmente independentes dado $\mathbf{Z}$ é verdadeira, então a estatística $2nI(\mathbf{X}, \mathbf{Y}|\mathbf{Z})$ está próxima a uma distribuição $\chi^2(l)$ (qui-quadrado) com $l = (r_X - 1)(r_Y - 1) r_Z$ graus de liberdade, onde r_X, r_Y e r_Z representam o número de configurações entre os conjuntos de variáveis $\mathbf{X}$, $\mathbf{Y}$ e $\mathbf{Z}$, respectivamente. Se $\mathbf{Z} = \emptyset$, a estatística $2nI(\mathbf{X}, \mathbf{Y})$ está próxima a uma distribuição $\chi^2(l)$ com $l = (r_X - 1)(r_Y - 1)$ graus de liberdade.*

A informação mútua $I(X_i; X_j)$ ainda é usada para medir o grau de associação entre duas variáveis X_i e X_j no novo método. Ao mesmo tempo, no entanto, um termo penalizante

relacionado à distribuição do χ^2 também é adicionado. Shi e Tan [50] afirmam que X_i e X_j somente serão consideradas independentes se $2nI(\mathbf{X}, \mathbf{Y}) \leq \chi_{\alpha,l}$. Em seus experimentos, Shi e Tan [50] usam valores tabelados para $\chi_{\alpha,l}$ quando $l \leq 100$. Já quando $l > 100$, a aproximação de *Wilson-Hilferty* é utilizada. Portanto,

$$\chi^2(l) \approx \chi_{\alpha,l} = l \left[1 - \frac{2}{9l} + z_\alpha \sqrt{\frac{2}{9l}} \right]^3$$

onde z_α pode ser definido como um ponto crítico, dado $0 < \alpha < 1$, de uma distribuição normal [42] e pode ser encontrado por

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z_\alpha} e^{\frac{-x^2}{2}} dx = \alpha$$

Utilizando os novos conceitos, o novo método, *InfoChi*, pode ser descrito como

$$InfoChi(\mathbf{X}, \mathbf{Y}) = 2nI(\mathbf{X}, \mathbf{Y}) - \chi_{\alpha,l}$$

$$InfoChi(\mathbf{X}, \mathbf{Y}|\mathbf{Z}) = 2nI(\mathbf{X}, \mathbf{Y}|\mathbf{Z}) - \chi_{\alpha,l}$$

3.8 Outros Algoritmos

Algumas das soluções apresentadas até agora supõem que um processo estocástico estacionário produz todo o conhecimento, ou seja, a ordenação do conjunto de dados é irrelevante. No entanto, em muitos domínios de aplicação de RBs, tais como problemas financeiros [49], os processos variam de acordo com o tempo e os dados são não-estacionários ou parcialmente estacionários, o que reduziria a adequação das soluções já mencionadas. Nielsen e Nielsen [41] relaxam a suposição sobre dados estacionários e desenvolvem uma RB de aprendizagem incremental baseada em domínios de dados não-estacionários.

Tian et al. [53] apresentam uma melhoria no processo de refinamento de Friedman e Goldszmidt [21]. Neste estudo, desenvolveu-se um método incremental de aprendizado de RBs baseado em computação evolucionária, denotado pelo IEMA. Yasin e Leray [58] propõem um algoritmo incremental para a aprendizagem de estruturas de RB que lida com domínios de alta dimensionalidade.

Chunsheng e Qiquan [10] apresentam uma abordagem para otimizar incrementalmente estruturas de RBs. Com base em um método específico, essa abordagem decompõe a rede

inicial em várias sub-redes criadas a partir de uma árvore de junção desenvolvida usando informações sobre a probabilidade conjunta da rede. Com algumas adaptações, pode ser usado como um algoritmo de aprendizado incremental.

Liu et al. [39] usam o conceito de *grau de influência* para descrever a influência de novos dados na RB existente. Propõe-se um algoritmo baseado em pontuação para revisar uma RB de forma iterativa utilizando HCS para adicionar, reverter ou excluir bordas. Essas revisões acontecem somente onde os dados possuem influência e explicam alguma mudança de conhecimento.

Yue et al. [60] abordam a previsão de dados de *streaming*. Uma solução paralela e incremental para o aprendizado de RBs a partir de dados massivos, distribuídos e dinamicamente variáveis. Essa solução realiza adaptações em um algoritmo clássico de pontuação e busca e utiliza MapReduce. Cooper e Herskovits [11] também apresentam um algoritmo para dados de *streaming*, mais precisamente, um dado que é privado e horizontalmente compartilhado entre duas ou mais partes. Este algoritmo é baseado em uma versão eficiente de estatísticas suficientes para o aprendizado de RBs que preservam a privacidade.

3.9 Comentários sobre Algoritmos

Alguns pontos importantes para a compreensão dos métodos utilizados pelas principais soluções descritas nesse capítulo são apresetados na Tabela 3.1, 3.2 e na Tabela 3.3. Estas comparações fogem do escopo já abordado em comparações feitas nas seções anteriores.

Solução	Especialista de Domínio	Validação
Buntine [8]	Priori	Não possui
Friedman e Goldszmidt [21]	Priori	Dados sintéticos
Alcobé [2]	Priori	Dados sintéticos e reais
Lam e Bacchus [33]	Priori	Dados sintéticos
Shi e Tan [50]	Priori	Dados sintéticos e reais
Nielsen e Nielsen [41]	Priori	Dados sintéticos

Tabela 3.1: Tabela de comparação de metodologias

Solução	Processo Estocástico
Buntine [8]	Estacionário
Friedman e Goldszmidt [21]	Estacionário
Alcobé [2]	Não-estacionário
Lam e Bacchus [33]	Estacionário
Shi e Tan [50]	Não-estacionário
Nielsen e Nielsen [41]	Não-estacionário

Tabela 3.2: Continuação de tabela de comparação de metodologias

Algumas características da metodologia utilizada são apresentadas na Tabela 3.1. Dentre as soluções destacadas, nenhuma propõe a utilização do especialista de domínio como fonte de conhecimento a posteriori. Todas as soluções permitem usá-lo apenas como fonte de conhecimento a priori. Esse conhecimento pode ser incrementado ao longo do tempo e ser usado para melhorar a acurácia da rede construída. Além disso, mudanças que não podem ser identificadas apenas com o uso de dados, como a adição de nós e a incompreensibilidade do modelo, podem ser identificadas.

Ao verificar o processo estocástico, observa-se que Nielsen e Nielsen [41], Shi e Tan [50], Alcobé [2] adotam um domínio não-estacionário, já considerando a crescente diversidade de domínios. As outras soluções citadas mantiveram seu foco em domínios estacionários, mais comuns no momento em que foram desenvolvidas.

Nota-se também que, em geral, as metodologias utilizadas são validadas apenas em ambientes controlados utilizando dados sintéticos. A maioria realiza o experimento com conhecimento de onde chegará, realizando percussos controlados. Por exemplo, utilizando a rede *Alarm* [5], o procedimento padrão é a remoção de arcos existentes na rede original e a busca, através de um conjunto de dados gerados pelo procedimento, desses arcos removidos durante o procedimento. Como pode ser notado, ainda há casos em que não existe qualquer tipo de validação.

A Tabela 3.3 compara alguns conceitos relacionados ao algoritmo desenvolvido pelas soluções.

Na Tabela 3.3, a adição de arcos é indicado por "A", a remoção de arcos é indicada por

Solução	Alterações	Função de Pontuação	Busca	IC
Buntine [8]	"A"	Qualquer	Qualquer	-
Friedman e Goldszmidt [21]	"ABC"	DCM	Qualquer	-
Alcobé [2]	"ABC"	Qualquer	HCS	-
Lam e Bacchus [33]	"ABC"	DCM	Best-first	-
Shi e Tan [50]	"ABC"	-	HCS	InfoChi
Nielsen e Nielsen [41]	"ABC"	-	Qualquer	-

Tabela 3.3: Tabela de comparação de algoritmos

"B"e a reversão de arcos é indicado por "C". A busca local é um padrão presente entre os procedimentos de pesquisa usados. Apesar de sua alta complexidade computacional, os métodos são desenvolvidos para que isso não seja uma restrição e eles continuem sendo usados. Apenas uma solução fez uso de testes de IC. Shi e Tan [50] desenvolveram uma nova técnica que é usada como base para testes de IC. Alcobé [2] também desenvolveu duas heurísticas para minimizar a complexidade produzida pelo HCS em alguns casos. Considerando as funções de pontuação, algumas soluções deixam em aberto o uso de qualquer função. No entanto, uma adaptação para suas aplicações em conjunto com estatísticas suficientes é necessária. Algumas soluções usam o DCM com adaptações, seja para reduzir a complexidade computacional ou para alcançar melhores resultados em diferentes dados do conjunto de dados. Outros recursos podem ser abordados em revisões futuras, como complexidade computacional, foco de procedimento, tipo de domínio de aplicação, entre outros.

Os conceitos-chave das principais soluções apresentadas nesse capítulo podem ser encontrados em um mapa conceitual incluído no Apêndice A.

Capítulo 4

Avaliação Experimental

Neste capítulo, diferentes experimentos empíricos para avaliar o comportamento de dois algoritmos de aprendizagem incremental apresentados no Capítulo 3 são realizados diante de ambientes com complexidades divergentes. Um dos algoritmos é o algoritmo híbrido ST apresentado por Shi e Tan[50]. O outro é a versão incremental do HCS, o IHCS, apresentado por Alcobé[46]. Dados sintéticos e sobre o mundo real são utilizados na tentativa de reprodução de complexidade diferentes.

O restante do capítulo é organizado da seguinte maneira. Na Seção 4.1, uma descrição dos experimentos a serem realizados é apresentada. Na Seção 4.2, o comportamento dos algoritmos de aprendizagem em lote são avaliados em uma comparação com os incrementais. Na Seção 4.3, o comportamento dos algoritmos incrementais diante de contextos diversos são avaliados. Na Seção 4.4, a maneira como as restrições dos algoritmos incrementais afetam a qualidade dos seus resultados também é avaliada e na Seção 4.5, comentários sobre o comportamento dos algoritmos diante dos contextos abordados são descritos.

4.1 Protocolo Experimental

A estratégia de investigação empírica utilizada neste trabalho é uma pesquisa do tipo experimental em que são testadas hipóteses que avaliam a qualidade das estruturas aprendidas pelos algoritmos incrementais. Nesta seção, os principais pontos do planejamento dos experimentos são abordados. Os objetivos e questões de pesquisa são descritos, assim como os fatores e métricas de qualidade adotadas nos experimentos.

4.1.1 Objetivos de Pesquisa

O principal objetivo dos experimentos deste trabalho é avaliar algoritmos de aprendizagem incremental de estruturas com a intenção de comparar as RBs resultantes com respeito à eficácia na descoberta de suas estruturas e na generalização de novos dados no contexto de domínios com diferentes complexidades e variação de fatores destes algoritmos.

A eficácia na descoberta das estruturas e a generalização de novos dados são parâmetros utilizados para descrever a qualidade da rede final descoberta pelos algoritmos. Estas métricas são resumidas: (i) à pontuação DCM; (ii) à diferença entre a estrutura aprendida pela solução incremental e a estrutura original da rede ou a gerada pela versão em lote; (iii) à acurácia de predição; e (iv) ao desempenho de classificação, medido pela perda logarítmica. Uma melhor descrição de cada métrica é apresentada na Subseção 4.1.3.

Os fatores utilizados para alterar a complexidade do domínio são resumidos: (i) ao conjunto de dados; (ii) ao tamanho de passo de aprendizagem; (iii) à ordenação das instâncias; e (iv) à rede inicial. Os fatores utilizados para alterar as restrições impostas pelos algoritmos são: (i) número máximo de pais; (ii) número de operações com pontuação mais próxima à operação de melhor pontuação no algoritmo IHCS; e (iii) nível de confiança associado ao teste de IC no algoritmo ST. Uma melhor descrição de cada fator e seus níveis é apresentada na subseção 4.1.2.

As seguintes questões de pesquisa são utilizadas neste trabalho, baseadas no objetivo principal e na problemática abordada:

- **QP1:** A qualidade é significativamente semelhante nas redes aprendidas pelas soluções incrementais e pela solução em lote abordada?
- **QP2:** A qualidade das redes aprendidas é afetada significativamente pela complexidade do domínio?
- **QP3:** A qualidade das redes aprendidas é afetada significativamente pelas restrições presentes nos algoritmos incrementais?
- **QP4:** A qualidade é significativamente semelhante nas redes aprendidas pelas diferentes soluções incrementais?

4.1.2 Fatores do Experimento

Os fatores utilizados neste experimento são divididos em grupos dado a sua finalidade. Inicialmente, os fatores que indicam alterações no contexto são descritos. Posteriormente, os algoritmos utilizados e o grupo dos fatores que altera restrições realizadas pelos algoritmos são descritos.

Conjunto de Dados

Os contextos são baseados em conjuntos de dados com informações sintéticas e sobre o mundo real. Esta alternância é realizada com o objetivo de reprodução de contextos com diferentes complexidades. As seguintes bases de dados sintéticas são utilizadas na avaliação experimental:

- *Alarm*[1]: base de dados sobre monitoramento de pacientes em terapia intensiva;
- *Asia*[2]: base de dados usada para diagnósticos de doenças pulmonares;

As bases de dados sintéticas estão disponíveis no *bnlearn*[3], um pacote R para inferência e aprendizagem de RBs. Como bases de dados com informações reais, as seguintes são utilizadas:

- *Nursery*[4]: base de dados usada para classificação de pedidos de ingresso em creches;
- *Car*[5]: base de dados contendo informações para avaliação de modelos de carros;

As bases de dados reais, por sua vez, são coletadas no repositório de aprendizagem de máquina UCI[6]. A utilização de dados sobre o mundo real objetiva apresentar, aos algoritmos, dados com características diferentes aos dados sintéticos. Pretende-se, com o uso dos dados sobre o mundo real, apresentar alta complexidade aos algoritmos, mantendo altas divergências entre distribuições das variáveis. Esta variação será melhor abordada na descrição do fator *Tamanho do Passo*.

[1] http://www.bnlearn.com/documentation/man/alarm.html
[2] http://www.bnlearn.com/documentation/man/asia.html
[3] http://bnlearn.com/
[4] https://archive.ics.uci.edu/ml/machine-learning-databases/nursery/nursery.data
[5] https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.data
[6] https://archive.ics.uci.edu

Todas as bases de dados são dividas em dois conjuntos. Um conjunto para treinamento no algoritmo e outro, para teste da rede aprendida. A divisão é baseada na quantidade de instâncias das bases de dados. O conjunto de treinamento possui 75% da quantidade de instâncias, enquanto o restante é separado para testes de generalização.

As características dos conjuntos de dados sintéticos e reais analisados até aqui são detalhadas na Tabela 4.1. Todas as bases de dados tem classificação como uma de suas tarefas associadas e possuem somente atributos do tipo categóricos.

Base de Dados	#Atributos	#Instâncias Totais	#Treinamento	#Testes
Alarm	37	20000	15000	5000
Asia	8	5000	3250	1750
Nursery	9	12960	9720	3240
Car	7	1728	1296	432

Tabela 4.1: Descrição de conjunto de dados usados nos experimentos

É possível notar que, além da complexidade exercida pelas divergências nas distribuições, algumas outras características dos conjuntos de dados tornam um contexto mais ou menos complexo que outros. Na Tabela 2.2, há diferentes números de atributos em cada base de dados e também diferentes números de instâncias totais, por exemplo. Quanto maior o número de atributos, maior a complexidade da explicação do contexto. Unir esta complexidade a um pequeno número de instâncias, por exemplo, pode fazer com que haja uma descrição incorreta de cada instâncias de dados ou que classes sejam classificadas erroneamente. Estes ruídos, causado nos passos iniciais, podem ser maiores em bases de dados com o número de variáveis maior. O ruído citado também depende de dificuldade de coleta dessas variáveis, ou seja, ele pode ser maior em atributos reais do que em sintéticos.

Como citado anteriormente na Seção 2.3.2, o número de atributos influencia diretamente no espaço de busca. Quanto maior o espaço de busca, maior é a probabilidade de erros estruturais acontecerem.

Tamanho de Passo de Aprendizagem

Cada base de dados citada na subseção anterior é dividida em grupos de subconjuntos de dados para cada procedimento de aprendizagem i, onde $i = \{1, 2, 3\}$. Cada grupo, de tamanho k_i, é criado de acordo com o número de instância de cada base para que fosse possível a criação de um cenário incremental considerando a alimentação de dados aos algoritmos. Os valores de k são descritos na Tabela 4.2.

Base de Dados	k_1	k_2	k_3
Alarm	100	1000	4000
Asia	100	500	1000
Nursery	100	1000	2000
Car	100	200	400

Tabela 4.2: Quantidade de instâncias em subconjuntos de bases de dados

Todos os valores adotados para k, também definidos aqui como tamanho do passo de aprendizagem, são encontrados na literatura em avaliações experimentais [2, 50]. Os tamanhos do segundo e terceiro grupo de dados para a base de dados *Asia* e *Car* são menores que as demais bases, assim como é o terceiro grupo de dados para a base de dados *Nursery* em relação à base de dados *Alarm* por conta de seus tamanhos totais.

Subconjuntos menores tendem a explicar menos sobre o contexto e a cada novo passo de aprendizagem realizado, novas informações são incorporadas aos algoritmos. Esta nova informação pode causar grandes alterações nas distribuições de probabilidade já conhecidas e representadas pelos modelos aprendidos e, consequentemente, provocar perca de desempenho em algum dos algoritmos analisados. Quanto maior o conjunto de dados, maior é a quantidade de informação sobre o contexto e, em alguns casos, menor o custo de adaptação diante novos conjuntos de dados.

Considerando o tamanho dos passos adotados, as variações entre informações dos atributos em cada grupo de subconjuntos das bases de dados *Alarm*, *Asia*, *Nursery* e *Car* são apresentadas nas Figuras 4.1, 4.2, 4.3 e 4.4, respectivamente.

Pode-se notar, inicialmente, que a variação entre distribuições de probabilidade dos atributos é inversamente proporcional ao tamanho do passo adotado. Independente da base de

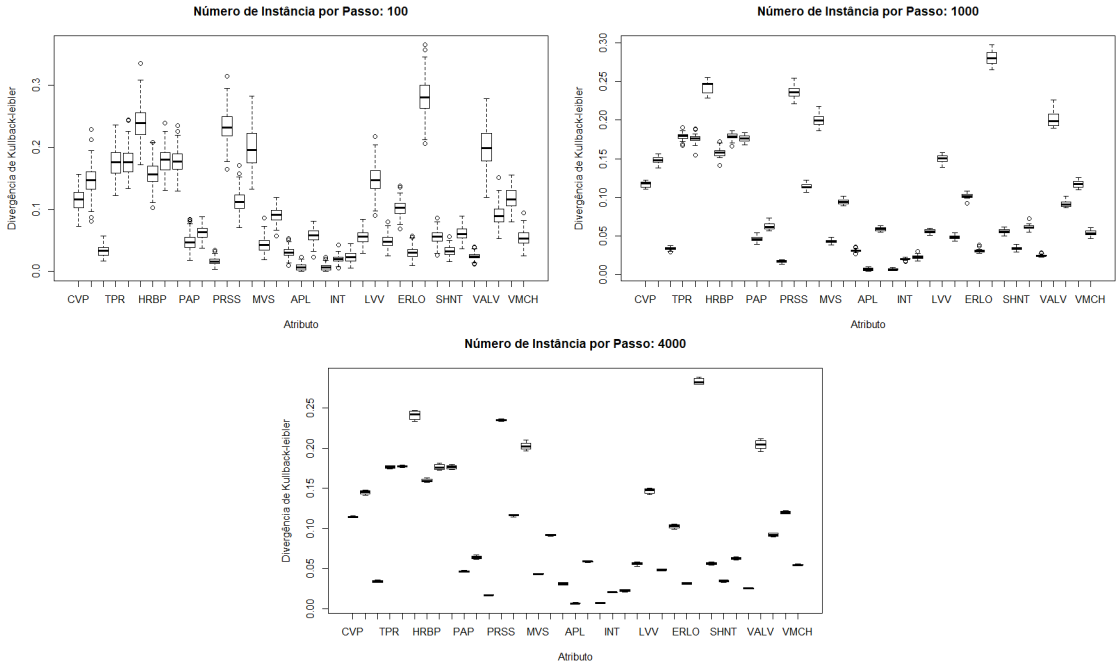

Figura 4.1: Variação entre distribuições dos atributos da base de dados *Alarm*

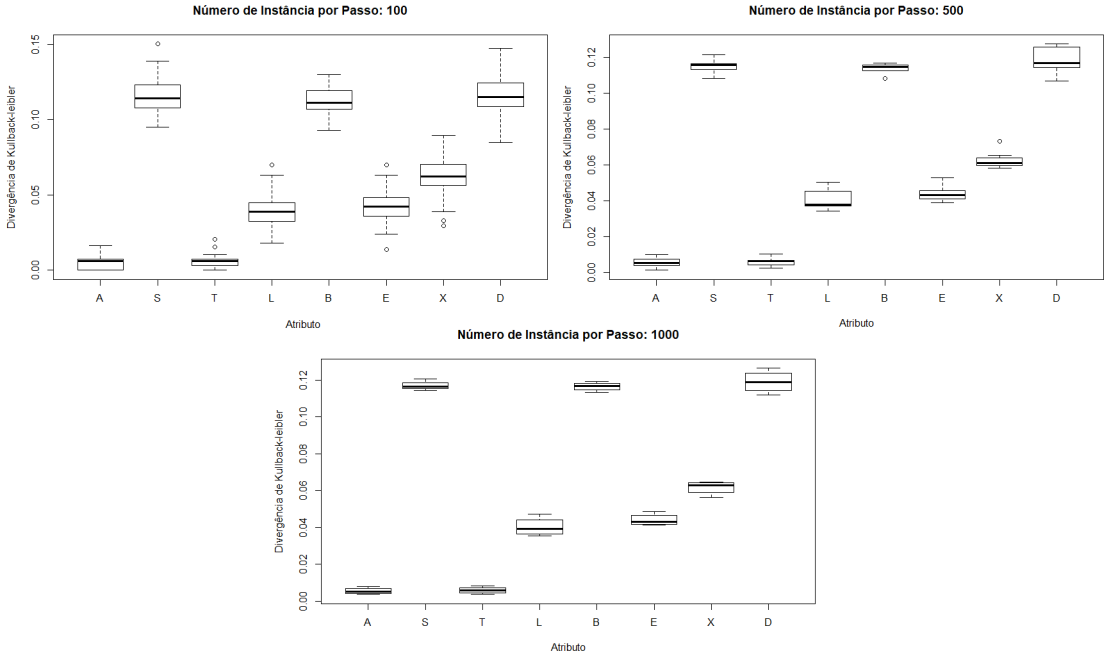

Figura 4.2: Variação entre distribuições dos atributos da base de dados *Asia*

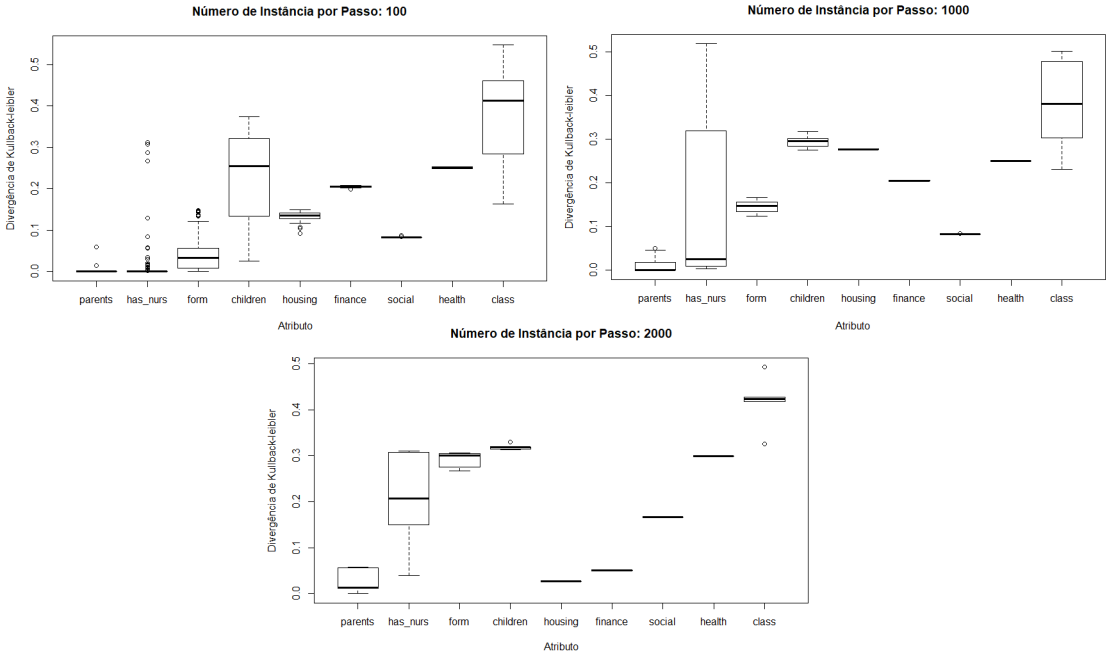

Figura 4.3: Variação entre distribuições dos atributos da base de dados *Nursery*

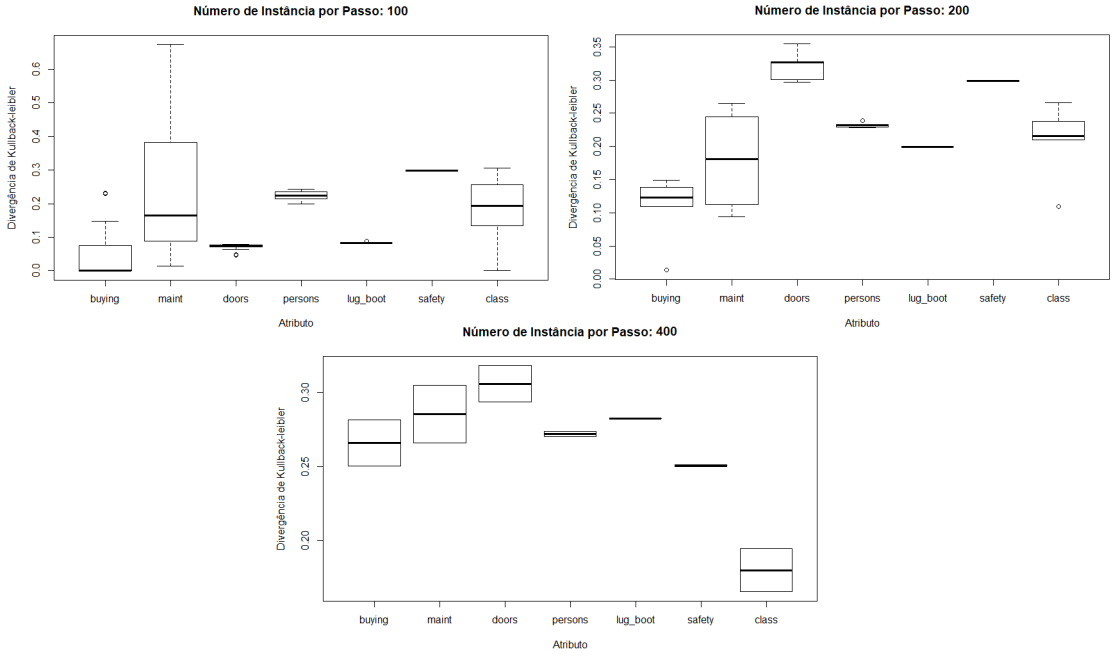

Figura 4.4: Variação entre distribuições dos atributos da base de dados *Car*

dados, quanto maior o passo, menor é a diferença entre as distribuições dos atributos.

Adotando a Figura 4.1 para a base de dados *Alarm* como exemplo, nota-se que nesta base de dados sintética, quando o passo de aprendizagem é 100, a maior diferença entre as variações dos atributos é 0,17 na escala de divergência de Kullback-Leibler. Já quando o passo de aprendizagem é aumentado para 4000, a maior diferença entre as variações é 0,02, bem menor do que nos passos anteriores. Esse comportamento também se repete nas bases de dados reais, como na Figura 4.3 para a base de dados *Nursery*. Neste caso, a maior diferença entre as variações é de 0,38 nos passos iniciais e cai para 0,28 no maior passo.

Percebe-se também que há um caso onde a variação aumenta na base de dados *Nursery*. Este fato ocorre, algumas vezes, em bases de dados com informações reais por conta da complexidade presente no domínio real de coleta de informações. Isto também explica outro padrão encontrado nas bases de dados reais. É possível notar que nessas bases, há divergências significativas entre distribuições, com um valores maiores que 0,6. Nas bases de dados sintéticas, a maior variação não passa de 0,34.

Apesar da alta variação identificada nas bases de dados, ainda sim, a variação média das distribuições de probabilidade dos atributos é bem parecida entre os passos, o que permite aos algoritmos produzirem RBs semelhantes.

Ordenação de Instâncias

Alguns algoritmos incrementais, como citado por Fisher et al. [17], são sensível à ordem em que as instâncias são usadas para alimentar o modelo. Ou seja, dado dois conjunto de dados D_1 e D_2 com ordens O_1 e O_2, o modelo gerado pelo mesmo algoritmo incremental para D_1 pode ser diferente quando gerado para D_2.

Caso O_1 ordene os dados para serem apresentados ao algoritmo de forma similar, ou seja, a próxima instância possui maior similaridade possível com a anterior, o resultado pode ser um modelo altamente enviesado com as primeiras instâncias. Por outro lado, se as instâncias forem ordenadas para que expliquem de forma uniforme o espaço de busca, as que refletem com precisão a variação presente nos dados poderão evoluir [17]. Portanto, a ordenação das instâncias de cada grupo também é utilizada como um fator a alterar a complexidade do domínio.

A similaridade entre as instâncias são medidas utilizando a distância definida como City

Block (CB). Esta medida basicamente indica o número de valores de atributos compartilhados entre as instâncias. Dado duas instâncias Z_i e Z_j, pode-se definir a distância CB como

$$CB(Z_i, Z_j) = \sum_{k=1}^{n} |z_{ik} - z_{jk}|$$

onde n é o número de atributos presentes no conjunto de dados e z_{ik} é o valor do atributo k na instância i. Caso os valores das instâncias sejam discretos, se $z_{ik} = z_{jk}$, o valor adotado é 1. Caso contrário, o valor adotado é 0. O valor máximo desta pontuação é igual ao número de atributos e o valor mínimo é 0.

Três níveis para esse fator são utilizados: (i) randômica; (ii) ordem similar; e (iii) ordem dissimilar. Quando um conjunto de dados $D = \{Z_1, ..., Z_i\}$ é ordenado similarmente, isto indica que

$$CB(Z_n, Z_{n-1}) <= ... <= CB(Z_2, Z_1).$$

Já quando conjunto de dados D é ordenado de forma dissimilar, isto indica que

$$CB(Z_2, Z_1) <= ... <= CB(Z_n, Z_{n-1}).$$

Z_1 é definido de forma randômica. Quando o nível para esse fator é randômico, isto indica que não há uma ordem específica a ser adotada na base de dados. A variação da pontuação CB de cada instância para as bases de dados *Alarm*, *Asia*, *Nursery* e *Car* são apresentadas nas Figuras 4.5, 4.6, 4.7 e 4.8, respectivamente.

Nota-se na Figura 4.5, no gráfico que exibe a pontuação na ordem similar, que a base de dados *Alarm* possui uma grande quantidade de grupos de instâncias similares com tamanho praticamente uniformes, fato que é acentuado pela sua alta quantidade de atributos. No entanto, a variação entre esses grupos é baixa, apresentando uma distância de apenas 10 pontos de pontuação em praticamente toda a base de dados (como citado, a maior pontuação é igual ao número de atributos, neste caso, 37). Isto indica que novas instâncias, quando ordenadas similarmente ou de forma inversa, acrescentam pouco conhecimento a cada passo do processo de aprendizagem. Este comportamento é acentuado na base de dados *Asia*, onde o gráfico que exibe a pontuação para a mesma ordem na Figura 4.6 apresenta uma alta quantidade de instâncias com a mesma pontuação ou com diferença mínima.

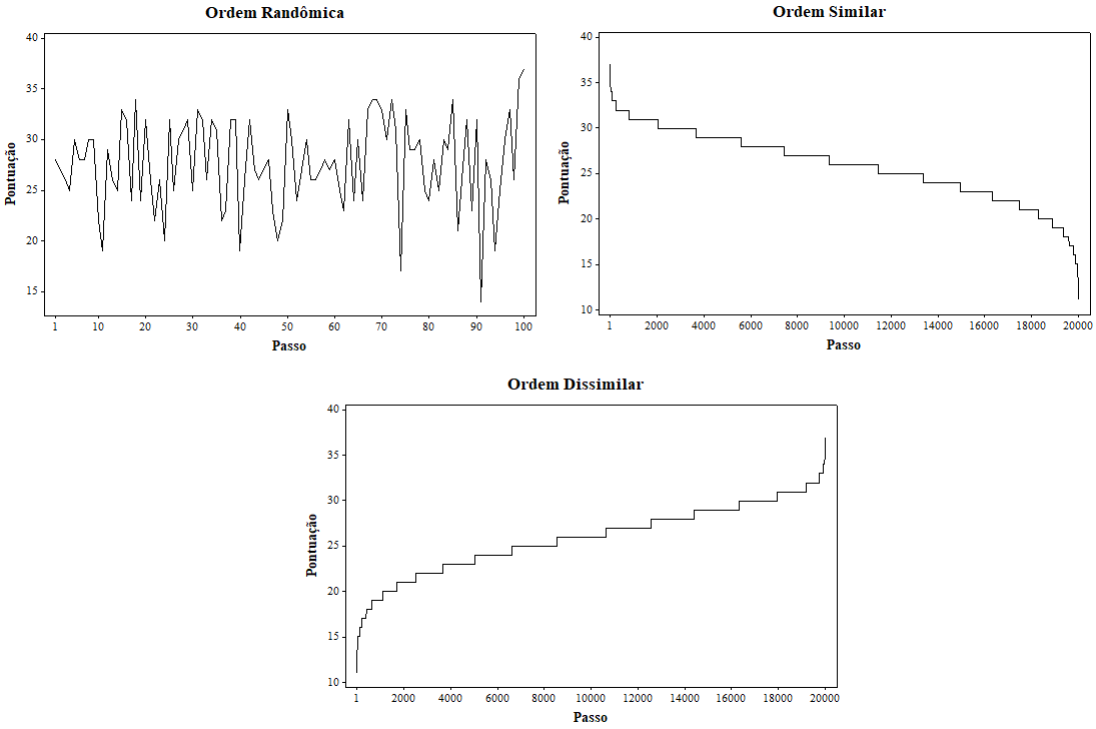

Figura 4.5: Variação entre pontuação CB de cada instância da base de dados *Alarm*

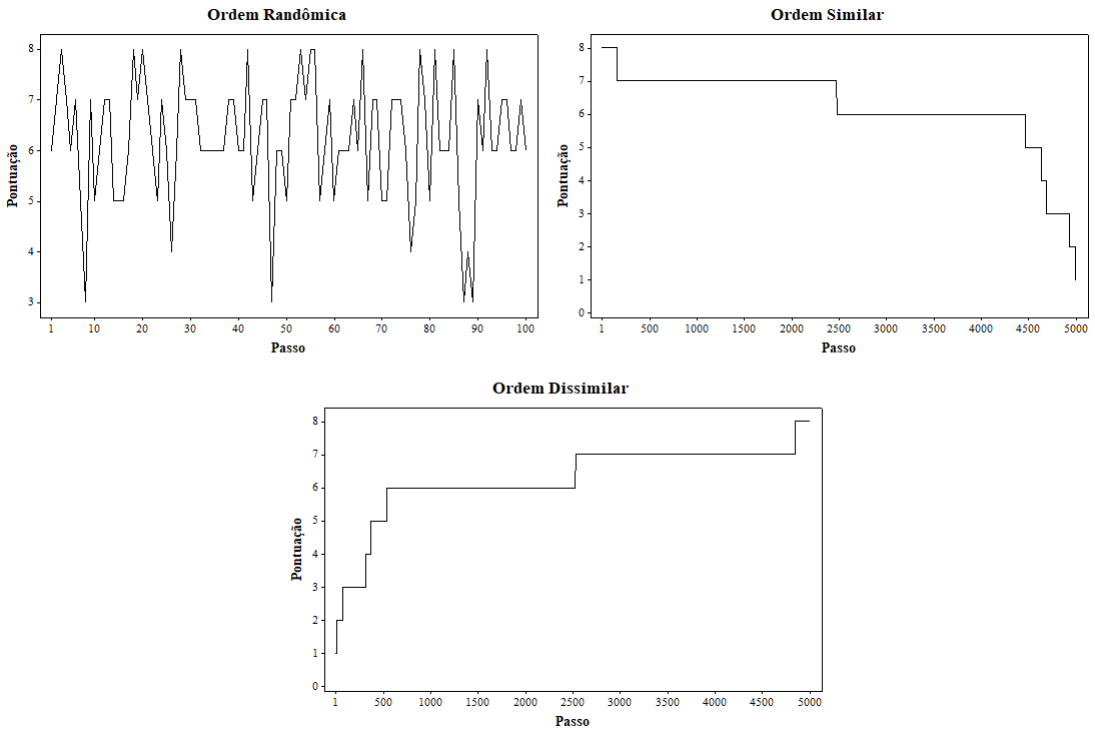

Figura 4.6: Variação entre pontuação CB de cada instância da base de dados *Asia*

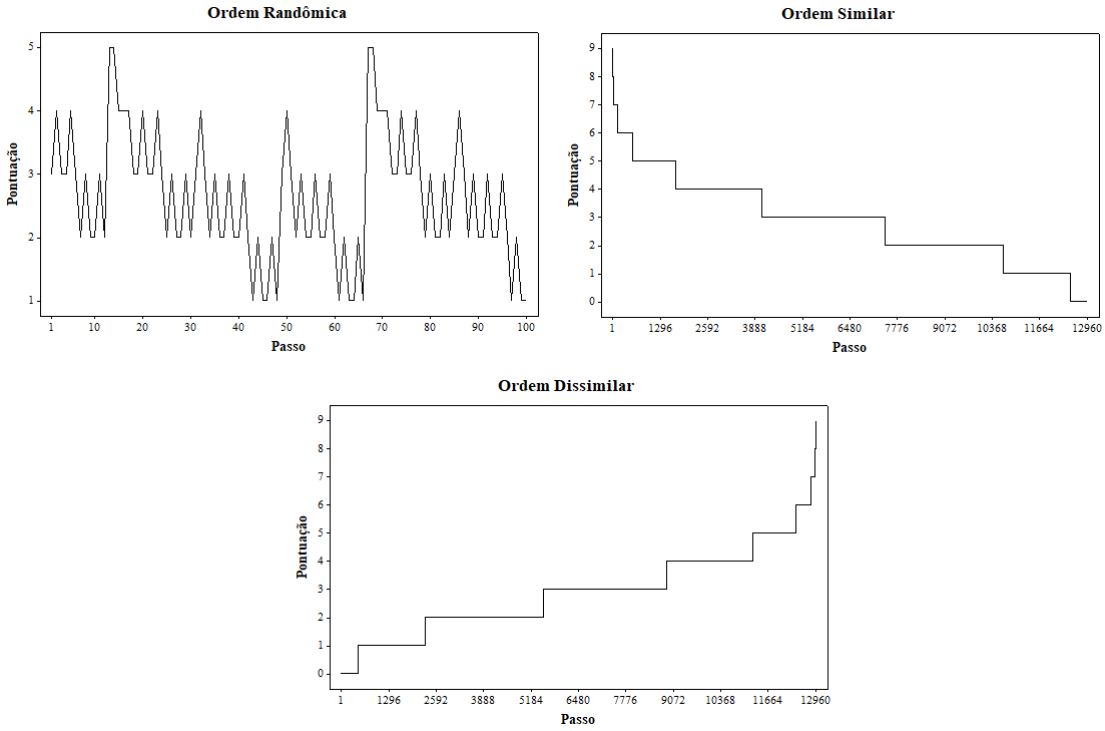

Figura 4.7: Variação entre pontuação CB de cada instância da base de dados *Nursery*

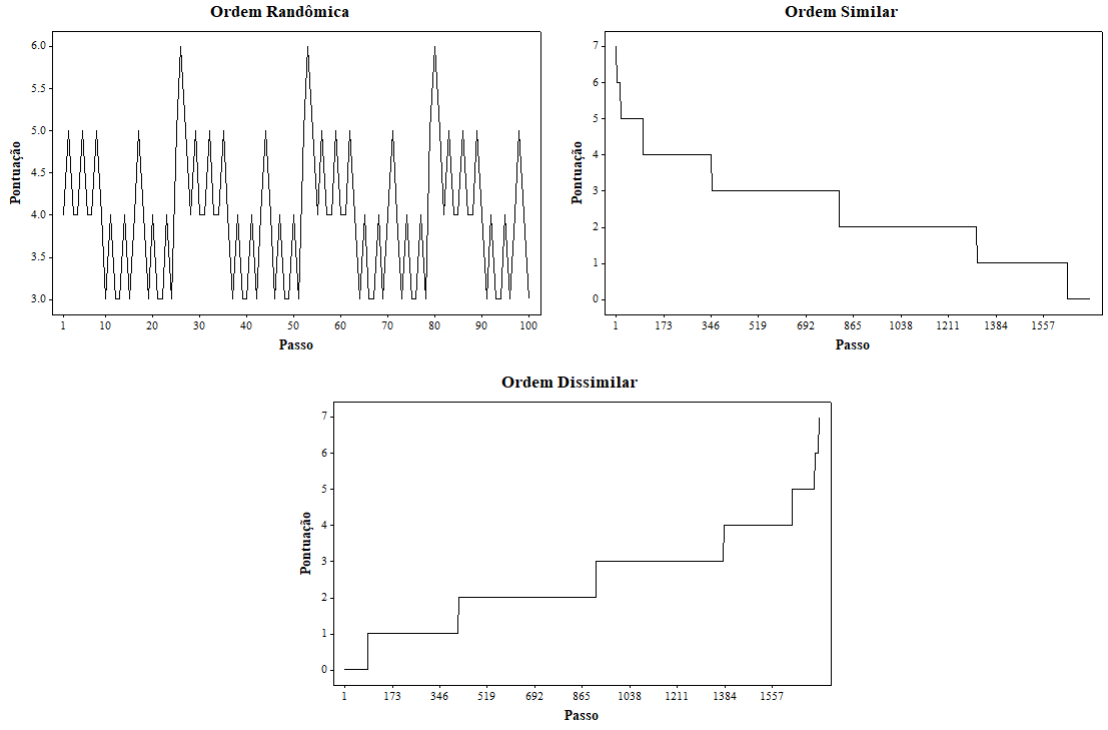

Figura 4.8: Variação entre pontuação CB de cada instância da base de dados *Car*

Nota-se também nas Figuras 4.7 e 4.8, no gráfico que exibe a pontuação na ordem similar, que as bases de dados reais *Nursery* e *Car* apresentam uma variação maior das pontuações dos grupos de instâncias similares. Também em ambos os gráficos, a linha que indica a pontuação possui uma variação maior em relação ao seu eixo horizontal. Esta variação não é identificada nas bases de dados sintéticas *Alarm* e *Asia*.

É possível identificar também um modelo aprendido e mantido possivelmente enviesado por mais tempo nas bases de dados sintéticas, onde os primeiros passos do procedimento de aprendizagem nos conjuntos de dados ordenados similarmente possuem baixa variância e essa variância diminui por boa parte do restante das instâncias como já citado. Este fato não se repete nos conjuntos de dados reais, onde a variância é maior nestes passos iniciais.

Já quando a ordem dos conjuntos de dados é invertida, ordenados agora de forma dissimilar, a variação nos primeiros passos é maior nas bases de dados sintéticas, o que permite uma explicação maior sobre o contexto, causando um efeito possivelmente contrário ao encontrado na ordem similar.

Rede Inicial

Outro fator utilizado nos experimentos é a RB que é usada para alimentar os algoritmos incrementais no início do procedimento de aprendizagem. Ambos os algoritmos incrementais precisam ser alimentados com uma rede no início do procedimento, seja ela vazia ou parcial.

Quando uma rede parcial é adotada, um conhecimento prévio é agregado ao processo de aprendizagem, tornando-o agora um processo que possui o refinamento do conhecimento como principal tarefa. A diferença entre as distribuições representadas pela rede inicial parcial e o conjunto de dados é menor à medida que os arcos que existem na rede parcial estejam corretamente adicionados. Durante o processo de refinamento então, neste caso, o contexto tende a ser menos complexo com relação à diferença entre as distribuições das variáveis.

Quando uma rede vazia é adotada, os algoritmos iniciam a aprendizagem do zero, partindo do pressuposto que nenhuma informação anterior ao procedimento é conhecida. Portanto, a cada novo conhecimento inserido no processo de aprendizagem a partir de novos conjuntos de dados, a tendência é uma variância maior nas informações já adquiridas pelos algoritmos do que nos processos de refinamento, aumentando o espaço de busca e, por vezes,

diminuindo a qualidade da estrutura aprendida.

Para este fator, redes iniciais vazias e parciais são adotadas. Nas Figuras 4.9 e 4.10, as estruturas parciais para os conjuntos de dados *Alarm* e *Nursery*, respectivamente, são apresentadas.

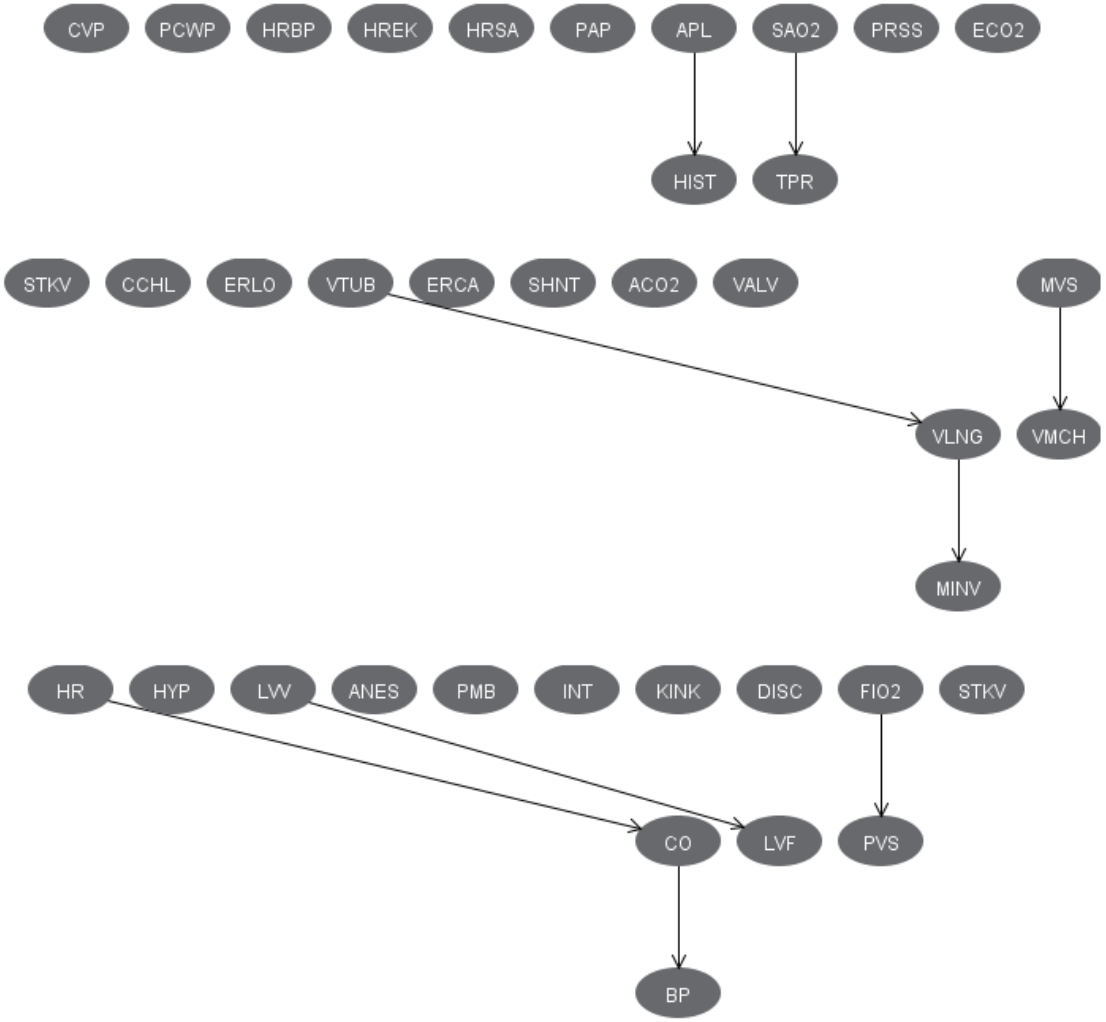

Figura 4.9: Rede parcial utilizada para o procedimento de aprendizagem do conjunto de dados *Alarm*

Nota-se na Figura 4.9 que apenas 9 arcos são adicionados na estrutura inicial da rede utilizada no experimento em que a base de dados *Alarm* é utilizada. Já na Figura 4.10, identifica-se a inserção de apenas 2 arcos na estrutura utilizada juntamente com a base de dados *Nursery*. Os arcos de ambas as estruturas são escolhidos aleatoriamente entre os arcos

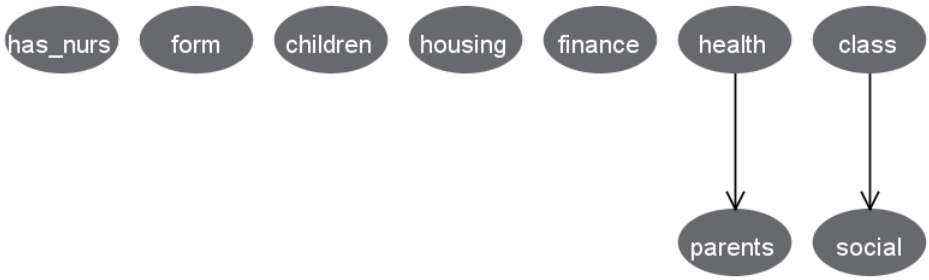

Figura 4.10: Rede parcial utilizada para o procedimento de aprendizagem do conjunto de dados *Nursery*

descobertos pelo algoritmo em lote HCS.

Uma alta quantidade de arcos é evitada para não enviesar o procedimento. Com este fator, pretende-se apenas avaliar o comportamento dos algoritmos na utilização destes conhecimentos prévios. Nota-se também que redes parciais são geradas para apenas duas redes. Isto acontece porque este fator não é adotado em todos os experimentos, sendo utilizado apenas nos experimento com redução da quantidade de ensaios (veja mais na Subseção 4.1.5).

Algoritmos HCS, IHCS e ST

Algoritmos de aprendizagem em lote e de aprendizagem incremental também são utilizados como fatores dos experimentos. Os algoritmos incrementais utilizados são os de Alcobé e de Shi e Tan, descritos nas Seção 3.5 e 3.7 e, no restante do trabalho, citados como IHCS e ST, respectivamente. Ambos são escolhidos dentre os algoritmos encontrados e destacados pela RSL realizada. Em experimentos na literatura, estes algoritmos incrementais possuem estruturas aprendidas com pontuações semelhantes às versões de algoritmo de aprendizagem em lote em algumas bases de dados, superando, em alguns casos [50], os algoritmos de Friedman-Goldszmidt [21] e Buntine [8].

Estes algoritmos são escolhidos também por utilizarem técnicas diferentes para atividades semelhantes quanto à restrição de possíveis operações entre variáveis antes de iniciarem suas buscas pela resolução ótima do problema de aprendizagem. O IHCS faz uso da heurística EBR, que baseia esta restrição na inclusão de apenas um grupo de k operações mais próximos à operação de melhor pontuação já obtida no passo anterior de aprendizagem. ST,

por sua vez, baseia sua restrição apenas em operações de adição e reversão de arcos entre nós utilizando o conhecimento obtido no cálculo da informação mútua existente entre os atributos.

Há diferenças também no uso de informações coletadas anteriormente. Enquanto ST utiliza o conhecimento sobre dependência condicionais anteriores nas suas buscas, IHCS, por sua vez, remove este conhecimento embutido na estrutura anterior e inicia uma nova busca em um determinado momentos.

Como o principal foco deste experimento é a avaliação dos procedimentos de aprendizagem, o HCS, descrito na 2.3.1, também é utilizado como solução em lote para a resolução ótima do problema de aprendizagem sem restrições de operações derivada de outros métodos. Os três algoritmos se assemelham na busca pela solução ótima, utilizando uma busca por pontuação como parte da resolução.

Os próximos três últimos fatores se referem à alterações nas restrições impostas pelos algoritmos.

Número Máximo de Pais

O número máximo de pais é um dos fatores usados para restrições presentes no algoritmo HCS, e consequentemente, nos algoritmos incrementais ST e IHCS. Este fator basicamente restringe a quantidade de pais possíveis para uma determinada variável durante a busca por otimização realizada pelo HCS. Nos algoritmos incrementais, essa é mais um etapa de restrição a ser considerada.

Como nível padrão, esse fator não adota restrição sobre os número de pais. Nos experimentos onde é necessário a alteração deste e dos fatores de restrições seguintes, são utilizados os valores conhecidos como *high* e *low*. Estes valores estão relacionados aos maiores e menos valores possíveis. Neste caso, como o padrão é sem restrição, o valor alternativo é restrição máxima de 1 pai por nó.

nRSS

O algoritmo IHCS utiliza algumas heurísticas para adaptação ao contexto incremental de aprendizagem. Dentre elas, a heurística EBR, em cada etapa do caminho de busca, armazena

os $nRSS$ modelos com a pontuação mais próxima ao modelo de maior pontuação naquela etapa do processo de aprendizagem em um conjunto denotado por **B** (veja mais em 3.5).

Alcobé [46] cita que 2 é o valor ideal para o parâmetro $nRSS$. Se houverem incrementações ou diminuições neste valor, a qualidade das estruturas não melhora e o tempo gasto aumenta. Em alguns casos, alterações no $nRSS$ provocaram redução drástica na qualidade das estruturas resultantes.

Diante da falta de evidências sobre este parâmetro e buscando entender melhor seu funcionamento, alguns valores referentes à qualidade das estruturas aprendidas para diferentes valores de $nRSS$ foram obtidos. Diferentes valores para o parâmetro em dois conjuntos de dados a serem experimentados foram utilizados. Diante desta análise, o parâmetro $nRSS$ é mantido com seu valor padrão de 2. Nos experimentos que é necessário alterar o nível desse fator, a remoção da restrição é adotada como novo nível.

Nível de Confiança de Testes Estatísticos (α)

O algoritmo ST é um algoritmo híbrido para solucionar o problema de aprendizagem. Ele possui técnicas de busca por pontuação e restrições realizadas por testes de IC. Para as restrições realizadas, Shi e Tan[50] propuseram um novo método, conhecido como InfoCHI, para realizar os testes condicionais e medir a associação entre variáveis (veja mais na Seção 3.7).

Um dos parâmetros do método InfoChi é α, que representa o nível de confiança associado aos testes estatísticos realizados para validar a associação entre as variáveis. O valor padrão adotado nos experimentos realizados pelos autores é 0,99. Este também é o valor padrão para os experimentos deste trabalho. No entanto, quando é necessária a variação de níveis deste fator, os valores são alternados entre 0,9 e 0,99. Estes valores são citados pelos autores como os valores padrões para α, juntamente com 0,95 [50].

4.1.3 Métricas de Avaliação

Nesta seção, as métricas utilizadas para avaliar a qualidade do modelos produzidos pelos algoritmos incrementais são discutidas.

Pontuação de Estruturas

A função de pontuação DCM, explicada na Seção 2.3.3, é usada para medir a adequação da distribuição explicada pelo modelo à contida na base de dados. Como citado na Seção 2.3.3, a qualidade do modelo está diretamente ligada à explicação, pela distribuição por ele representada, da distribuição de probabilidade do contexto a ser abordado.

O cálculo do DCM para este experimento é baseado na função de verossimilhança definida como

$$LL(B|D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} log \left(\frac{N_{ijk}}{N_{ij}} \right)$$

Assim, a função para o cálculo do DCM pode ser definida por

$$DCM\,(B|D) = LL\,(B|D) - \frac{1}{2} log\,(n)\,|B|$$

onde n é o número de instância da base de dados D e

$$|B| = \sum_{i=1}^{n} (r_i - 1)\, q_i$$

Quanto menor a pontuação do modelo na função DCM, maior a qualidade da explicação da distribuição dos dados. A razão entre a pontuação dos modelos produzidos também é calculada. P i/l é a pontuação das redes produzidas pelas abordagens incrementais dividida pela pontuação das redes produzidas pela abordagem em lote. Nota-se que quando a pontuação toma valores maiores que um, isto significa que a qualidade dos resultados dos algoritmos incrementais é melhor do que aqueles obtidos com as abordagens de lote. Pelo contrário, os valores menores que um favorecem os resultados da abordagem de lote. A razão entre a pontuação das redes produzidas pelo IHCS e a pontuação das redes produzidas por ST também é adotada. Esta razão é denotada por P r/s.

Perda Logarítmica

A Perda Logarítmica (PL) é utilizada como mais uma métrica para a medida de qualidade do modelo. Esta métrica é frequentemente utilizada para medir a performance de um modelo de classificação onde sua predição é um valor de probabilidade entre 0 e 1. A perda logarítmica é definida como

$$PL(B|D) = \sum_{d \in D_{teste}} logP_B(d)$$

onde D_{teste} é uma base de dados com instâncias selecionadas aleatoriamente da base de dados usada para treinamento. O tamanho de D_{teste} é 25% do tamanho da base de dados original (veja mais em 4.1.2). Um modelo perfeito teria uma perda logarítmica igual a 0. Quanto mais distante de 0 esteja o valor da perda, pior é a performance de predição do modelo analisado.

A razão entre a perda dos modelos produzidos também é adotada. PL i/l é a perda logarítmica das redes produzidas pelas abordagens incrementais dividida pela logarítmica das redes produzidas pela abordagem em lote. Nota-se que quando a pontuação toma valores maiores que um, isto significa que a qualidade dos resultados dos algoritmos incrementais é pior do que aqueles obtidos com as abordagens de lote. Pelo contrário, os valores menores que um favorecem os resultados das abordagens incrementais. A razão entre a perda das redes produzidas pelo IHCS e a pontuação das redes produzidas por ST também é calculada. Esta razão é denotada por PL r/s.

Acurácia de Predição

A acurácia das predições também é utilizada como mais uma métrica a ser avaliada. A acurácia é adotada como a porcentagem de acertos na predição realiza pelos modelos aprendidos. A acurácia é definida por

$$A(B) = \frac{TP}{TP + FP}$$

onde TP é a quantidade de acertos na predição e FP é a quantidade de erros. Nos conjuntos de dados onde há mais que um atributo como classe, a média da acurácia de cada classe é utilizada. Portanto, caso um modelo B possua n classes, sua acurácia média será

$$AverageA(B) = \frac{1}{n} \sum_{i=1}^{n} A(B_i)$$

onde $A(B_n)$ é a valor da acurácia para a classe i do modelo B.

A base de testes utilizada para validação da predição possui 25% do tamanho da base de dados original (veja mais em 4.1.2). Um modelo perfeito tem uma acurácia igual a 1. Quanto mais próximo de 0 esteja o valor da acurácia, pior é o modelo analisado.

Diferença Estrutural

A diferença estrutural entre dois modelos aprendidos, B_i e B_j, é mais uma métrica utilizada. Esta diferença é descrita por seis medidas: (i) número de arcos extras; (ii) número de arcos perdidos; (iii) número de arcos invertidos; (iv) precisão entre estruturas; (v) cobertura entre estruturas; e (vi) valor F.

O número de arcos extras entre B_i e B_j refere-se à quantidade de arcos presentes em B_i, mas ausentes em B_j. O número de arcos perdidos refere-se à quantidade de arcos presentes em B_j, mas ausentes em B_i. O número de arcos invertidos refere-se à quantidade de arcos que propagam o efeito de uma variável X_i a uma outra variável X_j presentes em B_i, mas que propagam de X_j a X_i em B_j.

A precisão P entre estruturas é medida por

$$P(B_i, B_j) = \frac{TP}{TP + FP}$$

onde TP indica a quantidade de arcos que estão em B_i e em B_j. FP indica a quantidade de arcos que não estão presentes em B_i, mas estão em B_j. A cobertura C entre estruturas é medida por

$$C(B_i, B_j) = \frac{TP}{TP + FN}$$

onde FN indica a quantidade de arcos que estão em B_i, mas não estão em B_j. O Valor F é calculado utilizando a precisão P e a cobertura C apresentadas anteriormente

$$F(B_i, B_j) = \frac{P(B_i, B_j)C(B_i, B_j)}{P(B_i, B_j) + C(B_i, B_j)} \times 2$$

4.1.4 Instrumentação

Todos os algoritmos incrementais utilizados nos experimentos são implementados usando a linguagem de programação Java[7] com auxílio da IDE Eclipse[8]. No mesmo ambiente, a execução dos experimentos é codificada objetivando a coleta, filtragem e extração das métricas de qualidade analisadas e dos modelos aprendidos.

[7] https://www.java.com/

[8] https://www.eclipse.org/

A versão em lote do HCS utilizada é implementada pela biblioteca Weka de mineração de dados do Java. Aproveitando o uso da biblioteca, o pacote *ibn* é então desenvolvido como uma camada extra da biblioteca que contém algumas das funcionalidades necessárias para manipular o aprendizado incremental utilizando esta biblioteca de mineração de dados. O código-fonte deste pacote pode ser acessado no GitHub[9].

A execução da análise estatística adotada neste trabalho, assim como a definição e execução dos *designs* dos experimentos é realizada com o auxílio RStudio 1.1.463[10], ambiente de desenvolvimento para R[11], e do Minitab 18.1[12], ferramenta de análise estatística.

4.1.5 Design de Experimentos

É possível perceber que, dado a descrição dos grupos de fatores e métricas, a quantidade de fatores a ser utilizados nos experimentos muda de acordo com a questão de pesquisa. Enquanto a principal avaliação é feita sobre os algoritmos na questões de pesquisa **QP1** e **QP4**, nas questões **QP2** e **QP3**, os fatores que alteram o processo de aprendizagem agora são o ponto principal.

Entende-se que seria possível a realização de um só experimento para explicar todas as questões. No entanto, a quantidade de ensaios necessários é muito alta, chegando a ser preciso realizar 2592 ensaios, excluindo possíveis repetições. Mesmo que designs alternativos ao fatorial completo fossem utilizados, como o design 2^k e o fatorial fracionário, este número ainda permanece alto ou o design possui uma resolução muito baixa, confundindo efeitos dos fatores principais que seriam importantes para a análise.

Consequentemente, três experimentos são definidos para responder todas as quatro questões de pesquisa. Seus designs são então desenvolvidos objetivando reduzir o erro experimental. Para a **QP1** e **QP2**, um design fatorial completo é utilizado, onde combina-se cada nível de um fator com todos os níveis dos demais fatores, possibilitando a análise completa dos efeitos nas métricas de avaliação. Para a **QP3**, um design fatorial fracionário 2^{6-2} é adotado, onde algumas iterações entre fatores são confundidas para diminuir o número de ensaios, mas ainda sim, sendo possível realizar uma análise dos efeitos principais nas métri-

[9] https://github.com/LuizAntonioPS/IBN

[10] https://www.rstudio.com/products/rstudio/

[11] https://www.r-project.org/

[12] https://www.minitab.com/

cas de avaliação. A questão de pesquisa **QP4** é então respondida baseando-se nos resultados dos experimentos anteriores.

No experimento para **QP1**, os fatores referentes aos algoritmos em lote e incrementais, ao conjunto de dados, ao tamanho k dos passos de aprendizagem e a ordem das instâncias são utilizados. O fator sobre a rede inicial não é utilizado porque os algoritmos em lote não utilizam qualquer rede como entrada no seu procedimento de aprendizagem. Para o algoritmo em lote, o número de instâncias é incrementado de acordo com o passo de aprendizagem e então disponibilizados para serem usados no processo de aprendizagem. Para os incrementais, os dados são disponibilizados aos algoritmos com o mesmo tamanho k. Todas as métricas de qualidade são utilizadas e as restrições impostas pelos algoritmos são usadas com os valores padrões descritos em suas respectivas seções.

No experimento para **QP2**, todos os fatores que compõe o grupo que altera o contexto (veja mais na Subseção 4.1.2) são utilizados. Neste experimento, os conjunto de dados também são utilizados como fatores. No entanto, apenas dois níveis são abordados: (i) *Alarm*; e (ii) *Nursery*. Esses dois níveis são diversos o suficiente para explicar alterações no contexto. Como abordado na Subseção 4.1.2, o conjunto de dados *Alarm* possui dados sintéticos e características bastantes comuns a esse cenário, como a baixa variação na distribuição dos dados dos atributos. Além disso, apesar da alta quantidade de variáveis, possui um grande número de instâncias, o que permite uma boa explicação sobre contexto. A baixa variação permite que um grande conhecimento sobre a base de dados já ocorra nos primeiros passos. A base de dados *Nursery* segue um caminho oposto. Enquanto possui uma alta variação, até entre passos de aprendizagem, possui poucas variáveis descrevendo o contexto real, o que aumenta a probabilidade de ruído nos dados. Todas as métricas de qualidade são utilizadas e as restrições impostas pelos algoritmos são usadas com os valores padrões descritos em suas respectivas seções.

No experimento para **QP3**, ainda sim, todos os fatores descritos na Subseção 4.1.2 são utilizados. No entanto, apenas dois níveis de cada fator são adotados. Estes níveis são conhecidos como *high* e *low* e definem os níveis que possuem os maiores e menores efeitos nas métricas de qualidade. Para o fator referente ao conjunto de dados, *Alarm* e *Nursery* são adotados pelo mesmo motivo já descrito anteriormente. Para o tamanho do passo, os maiores e menores passos possíveis para cada conjunto de dados são utilizados: (i) 100 e

	Ordem de Instâncias	P i/l	PL i/l	A i/l
IHCS	Randômica	1	1	1
	Similar	1	1	1
	Dissimilar	1	1	1
ST	Randômica	0.98959	1.011404	0.999974
	Similar	0.989812	1.049708	0.987628
	Dissimilar	0.993316	1.018286	1.000025

Tabela 4.3: Sumarização de resultados usando *Alarm*

4000 para *Alarm*; e (ii) 100 e 2000 para *Nursery*. A ordem das instâncias são definidas seguindo os resultados alcançados por Fisher et al. [17] em seus experimentos: similar como *low* e dissimilar como *high*. Para os fatores referentes às restrições, os valores já descritos em seus seções são os utilizados. Todas as métricas de qualidade também são utilizadas.

Dois experimentos, para cada algoritmos incremental analisado, são realizados já que algumas das restrições são únicas do algoritmo. Os experimentos para a questão **QP3** são fracionados. A Resolução IV é utilizada, onde os efeitos dos principais fatores não são confundidos com interações entre os efeitos de dois fatores, apenas com três fatores ou mais.

Os ensaios definidos para cada experimento são apresentados no Apêndice B.

4.2 Comparação entre Soluções Incrementais e em Lote

Uma sumarização dos resultados obtidos sobre a aplicação dos algoritmos nos conjuntos de dados *Alarm*, *Asia*, *Car* e *Nursery* é apresentada nas Tabelas 4.3, 4.4, 4.5 e 4.6, respectivamente. Nestas tabelas, somente a razão entre a pontuação DCM (P i/l), perda logarítmica (PL i/l) e acurácia (A i/l) dos algoritmos incrementais (i) e dos algoritmos em lote (l) são descritas. A descrição dos dados completos tomaria bastante espaço e explicaria, basicamente, as mesmas divergências. Uma noção melhor dos valores absolutos será abordada nas próximas seções.

É possível ver nas Tabelas 4.3, 4.4, 4.5 e 4.6 para todos os conjuntos de dados que tanto os algoritmos incrementais, como os algoritmos em lote produzem redes com métricas bastante

	Ordem de Instâncias	P i/l	PL i/l	A i/l
IHCS	Randômica	0.997151	1.001385	1
	Similar	1.060271	0.77014	1.08073
	Dissimilar	1.409869	1.830761	0.707646
ST	Randômica	1.091399	1.393762	0.946032
	Similar	0.990698	0.981787	1.08073
	Dissimilar	1.101865	1.08591	1

Tabela 4.4: Sumarização de resultados usando *Asia*

	Ordem de Instâncias	P i/l	PL i/l	A i/l
IHCS	Randômica	1.012566	2.397834	0.761062
	Similar	1	1	1
	Dissimilar	0.999089	2.272694	0.671053
ST	Randômica	0.988037	1.787487	0.864307
	Similar	0.998062	0.818922	1.157233
	Dissimilar	0.993659	0.551637	1.076316

Tabela 4.5: Sumarização de resultados usando *Car*

	Ordem de Instâncias	P i/l	PL i/l	A i/l
IHCS	Randômica	1	1	1
	Similar	0.990041	2.401422	0.826814
	Dissimilar	0.975076	0.815985	1.012418
ST	Randômica	0.988031	0.827838	0.930591
	Similar	0.991407	0.828198	1.022404
	Dissimilar	0.975204	0.967362	1.006985

Tabela 4.6: Sumarização de resultados usando *Nursery*

similares. A razão entre elas sempre está próximo a 1, o que configura suas similaridades. No entanto, há alguns padrões que podem ser observados nos dados. A pontuação DCM do algoritmo ST é quase sempre ligeiramente inferior à da rede gerada pelo HCS, enquanto que, na comparação entre o IHCS e HCS, uma também pequena inferioridade alterna em contextos sintéticos que não possuem ordem randômica ou em contextos reais que possuem esta ordem.

Pode-se notar também que a acurácia e perda dos algoritmos incrementais se comporta de maneira diferente quando comparada as mesmas métricas do HCS. IHCS produz redes com acurácia inferior e perda superior às redes geradas pelo IHCS em contexto reais complexos, enquanto que o ST inverte esses resultados, apresentando acurácias e perdas ligeiramente melhores que o HCS.

Nota-se algumas exceções onde IHCS, entre seus resultados, apresenta inferioridade, além da média das outras observações, em comparação ao HCS. A maioria desses resultados encontram-se no conjunto de dados *Car* e outros em *Nursery*, onde a complexidade com número de instâncias e a alternância de conhecimento é maior.

Nas próximas seções, uma avaliação sobre cada métrica de qualidade ao longo dos diferentes processos de aprendizagem é realizada. Todas as ordens das instâncias são adotadas. Um teste hipótese é utilizado para avaliar as amostras de cada processo entre si. Baseado na presença de variâncias desiguais entre as amostras, o teste T de Welch é realizado, com significância de 5%, para avaliar a igualdade entre as médias das amostras. A hipótese nula adotada é que a diferença entre as médias das amostras é 0. A alternativa é que a diferença entre as médias é diferente de 0.

4.2.1 Pontuação Estrutural

Na Figura 4.11, são apresentados os intervalos, com confiança de 95%, para a média da pontuação atribuídas as redes geradas durante os processos de aprendizagem no conjunto de dados *Alarm*. Todos os intervalos apresentados no restante deste trabalho também possuem confiança de 95%.

Pode-se notar que as pontuações das estruturas geradas por todos os algoritmos parecem semelhantes em todas as ordem adotadas. A única diferença é a pontuação DCM inferior no caso onde a ordem utilizada é a dissimilar. Nota-se, ao analisar o gráfico, que os intervalos

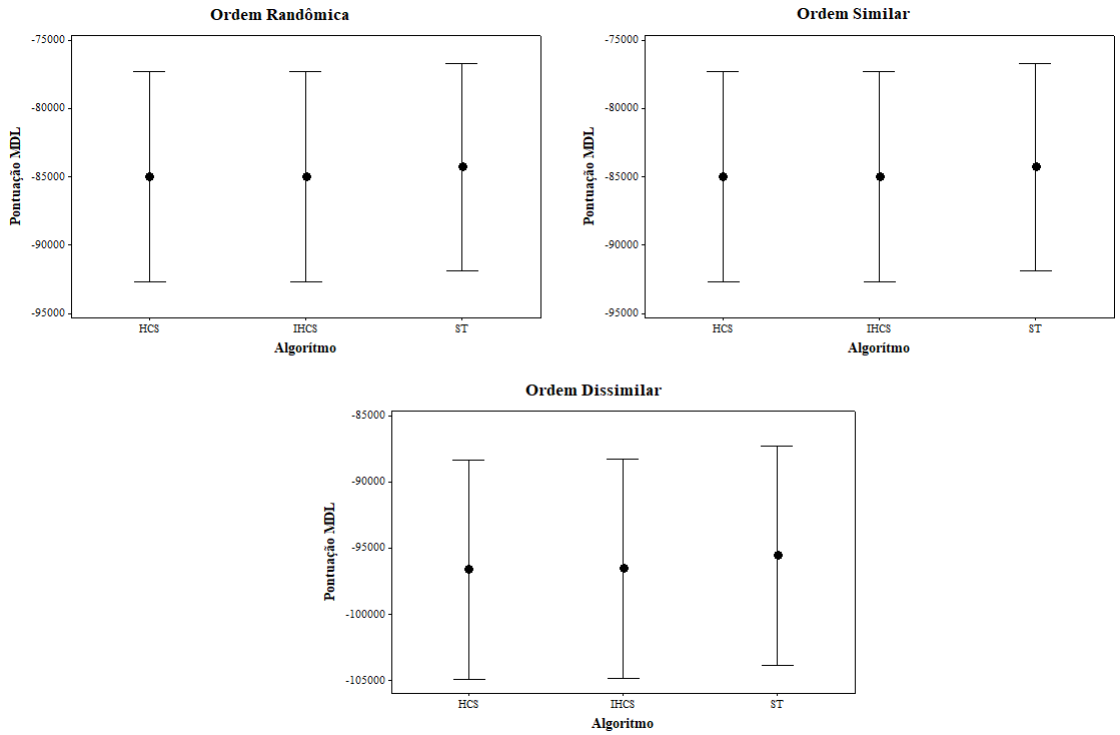

Figura 4.11: Pontuação DCM dos modelos para o conjunto de dados *Alarm*

se sobrepõem, indicando que há semelhança entre os algoritmos, tanto na ordem randômica, quanto similar e dissimilar, sem a necessidade de realização do teste T.

O comportamento acima descrito também pode ser observado nas Figuras 4.12 e 4.13. Nestas figuras, são apresentados os intervalos de confiança para a média da pontuação atribuídas as redes geradas durante os processos de aprendizagem no conjunto de dados *Car* e *Nursery*.

Para os intervalos de confiança sobre a média alcançada pelos algoritmos no conjunto de dados *Asia*, no entanto, há um comportamento diferente. Estes intervalos podem ser observados na Figura 4.14.

Na Figura 4.14, pode-se notar que quando a ordem das instâncias é randômica, o algoritmo ST produz redes com a média da pontuação DCM mais baixa que os outros dois algoritmos. Utilizando o teste T de Welch, o valor de p de 0,544 é obtido para as amostras dos algoritmos ST e HCS, obtendo significância estatística que não indicam diferença entre as médias das amostras. Quando a ordem das instâncias é similar, o algoritmo IHCS produz redes com a média da pontuação DCM mais baixa que os outros dois algoritmos. Utilizando

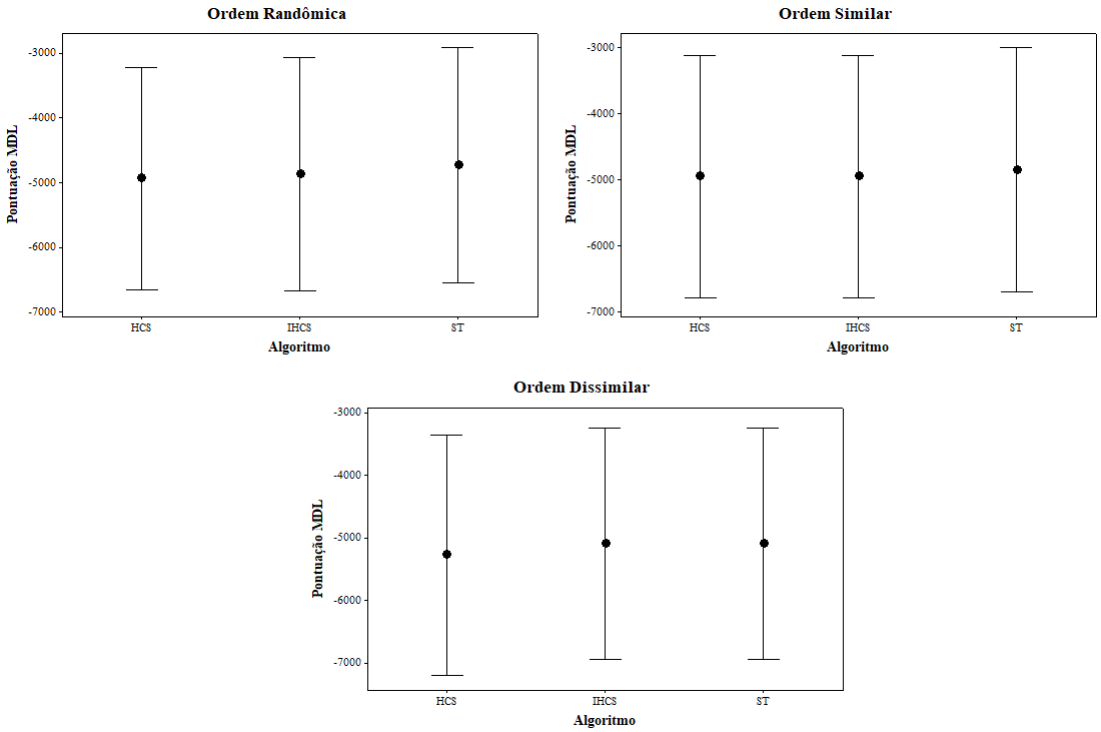

Figura 4.12: Pontuação DCM dos modelos para o conjunto de dados *Car*

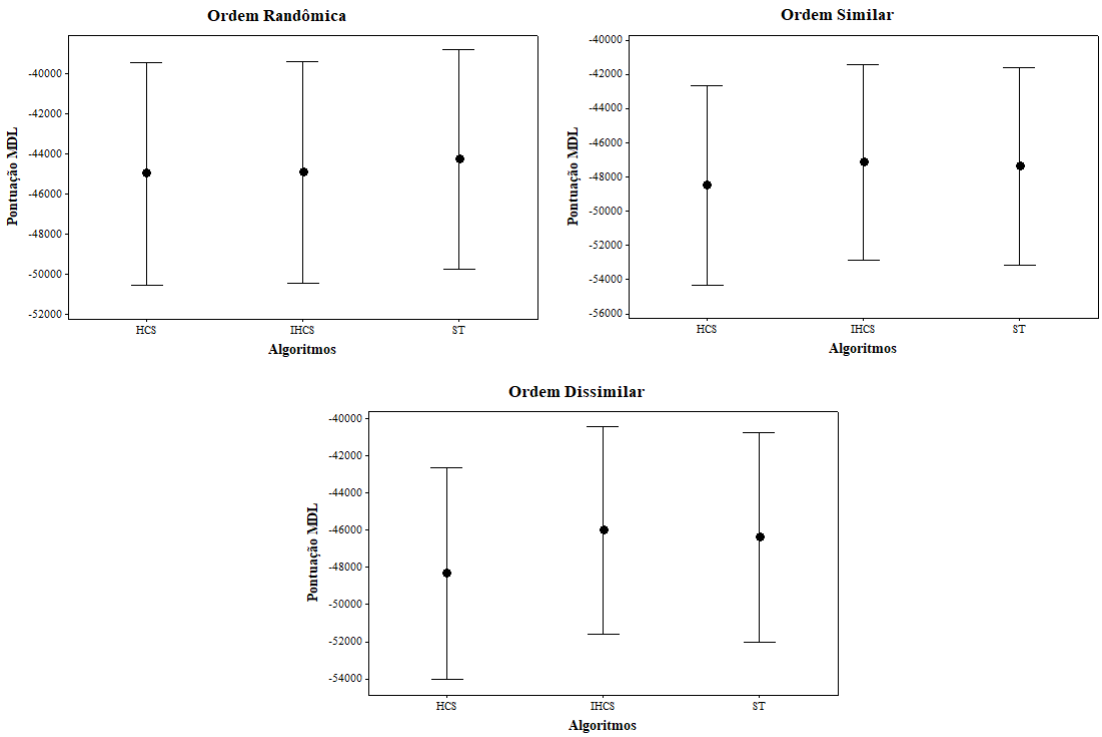

Figura 4.13: Pontuação DCM dos modelos para o conjunto de dados *Nursery*

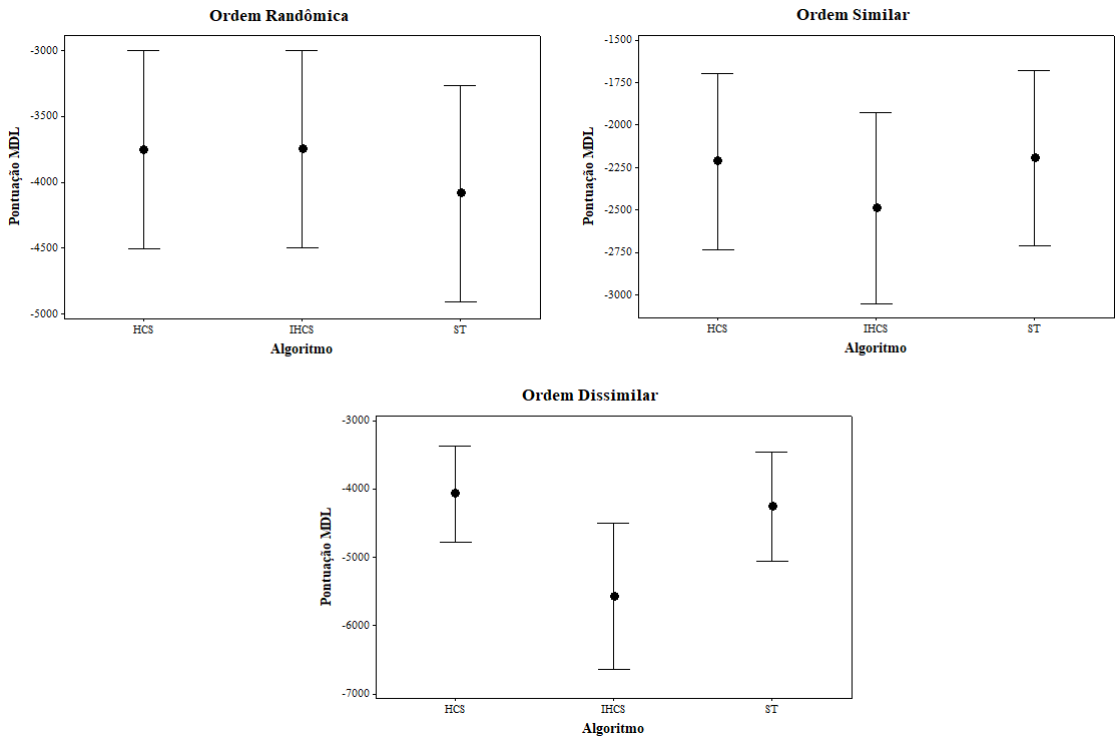

Figura 4.14: Pontuação DCM dos modelos para o conjunto de dados *Asia*

o teste T de Welch, o valor de p de 0,470 é obtido para as amostras dos algoritmos IHCS e HCS, obtendo significância estatística para supor indiferença entre as médias das amostras.

Ainda na Figura 4.14, percebe-se que quando a ordem é dissimilar, novamente o IHCS produz pontuações com média menor do que os demais. Utilizando o teste T de Welch, o valor de p de 0,027 é obtido, refutando a hipótese nula sobre a igualdade entre as médias das amostras.

Em resumo, pode-se verificar que os algoritmos produzem estruturas com pontuação DCM semelhantes, em todas as ordens analisadas e em todas os conjuntos de dados, exceto no conjunto *Asia*, onde, na ordem dissimilar, IHCS produz redes com pontuação DCM média estatisticamente diferente.

4.2.2 Curva de Aprendizagem

Na Figura 4.15, são apresentados os intervalos de confiança para a média da perda logarítmica atribuídas as redes geradas durante os processos de aprendizagem no conjunto de dados *Alarm*.

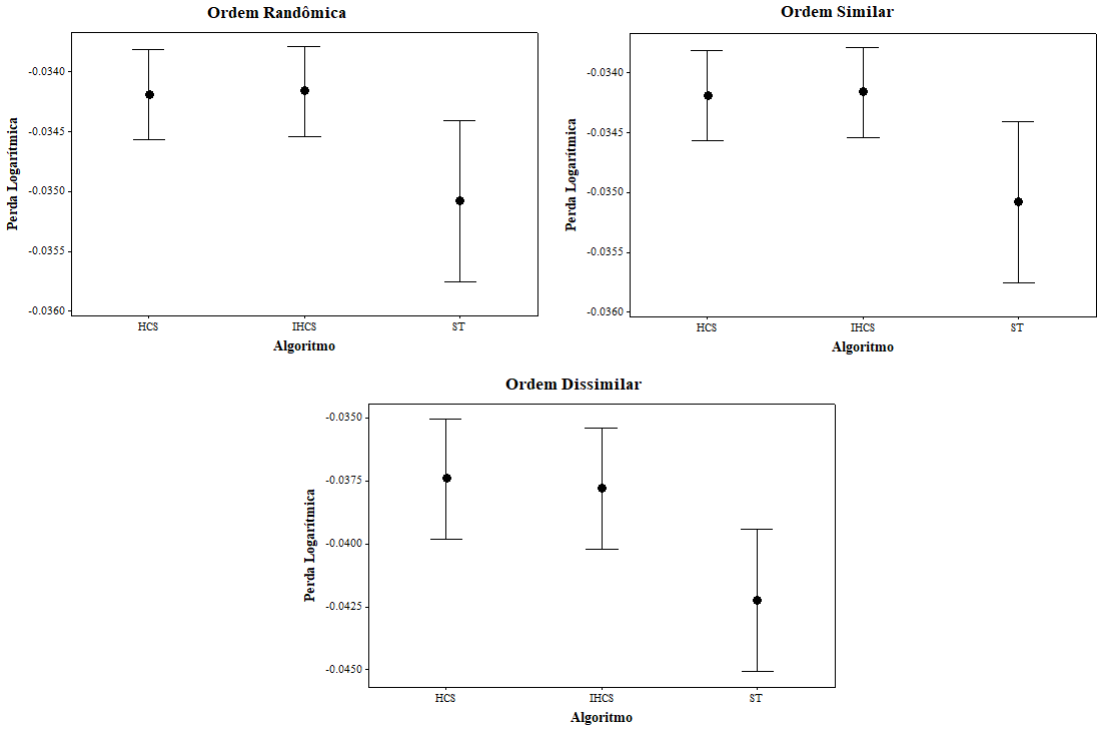

Figura 4.15: Perda logarítmica dos modelos para o conjunto de dados *Alarm*

Pode-se notar que na Figura 4.15, o algoritmo IHCS produz redes com perda logarítmica semelhante às redes produzidas por HCS, independente da ordem das instâncias. Este fato não se repete com ST. Este algoritmo produz redes com perda média maior que o HCS. Utilizando o teste T, os valores de p iguais a 0,024, 0,02 e 0,010 são obtidos. Logo, é suposto, com significância de 5%, que o algoritmo ST produz redes com perda logarítmica diferente das redes produzidas por HCS. Isto indica que ST produz redes com performance pior que HCS.

Seguindo a análise, na Figura 4.16, são apresentados os intervalos de confiança para a média da perda logarítmica atribuídas as redes geradas durante os processos de aprendizagem no conjunto de dados *Asia*. Pode-se notar na Figura 4.16 que, para a ordem similar e dissimilar, os algoritmos HCS e IHCS produzem redes com qualidade diferentes. Nota-se que seus intervalos não se sobrepõem, havendo diferença estatística entre as redes produzidas por IHCS e HCS quando a ordem é similar. No entanto, possuem semelhança quando a ordem é randômica. Quando as redes produzidas por ST são analisadas, há similaridade com HCS quando a ordem é dissimilar, mas há diferença entre elas quando a ordem é randômica

e similar. Quando randômica, a diferença visual é aparente. Quando similar, o teste T é adotado e um valor p de 0,009 é obtido, indicando diferença estatística e refutando a hipótese nula.

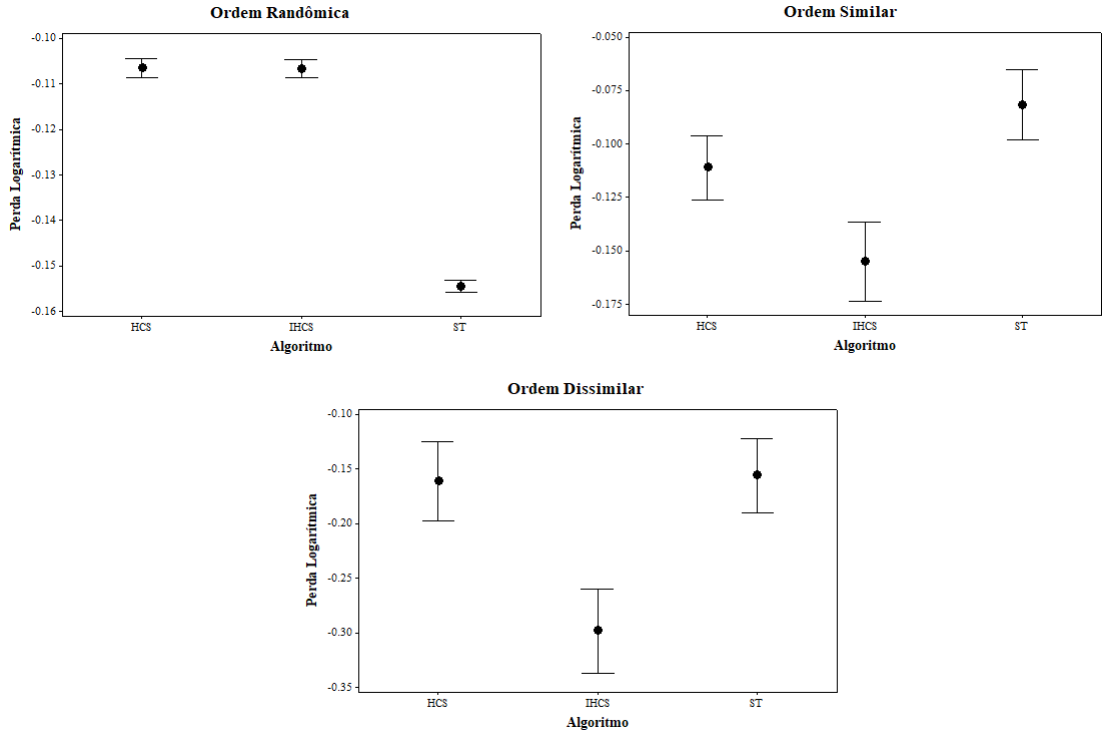

Figura 4.16: Perda logarítmica dos modelos para o conjunto de dados *Asia*

A Figura 4.17 é utilizada para apresentar os intervalos de confiança para a média da perda logarítmica atribuídas as redes geradas durante os processos de aprendizagem no conjunto de dados *Car*. Nota-se na Figura 4.17 que não há diferença entre as redes produzias pelos algoritmos quando a ordem é similar, nem dissimilar. No entanto, quando a ordem é randômica, não é possível avaliar a diferença visualmente. Usando o teste T, os valores de p igual a 0,005 e 0,024 são obtidos para a diferença entre HCS e IHCS e HCS e ST, respectivamente, indicando que nenhum dos algoritmos incrementais produzem redes com perda semelhante ao HCS.

A Figura 4.18 é utilizada para apresentar os intervalos de confiança para a média da perda logarítmica atribuídas as redes geradas durante os processos de aprendizagem no conjunto de dados *Nursery*.

Nota-se na Figura 4.18 que não há diferença entre as redes produzias pelos algoritmos

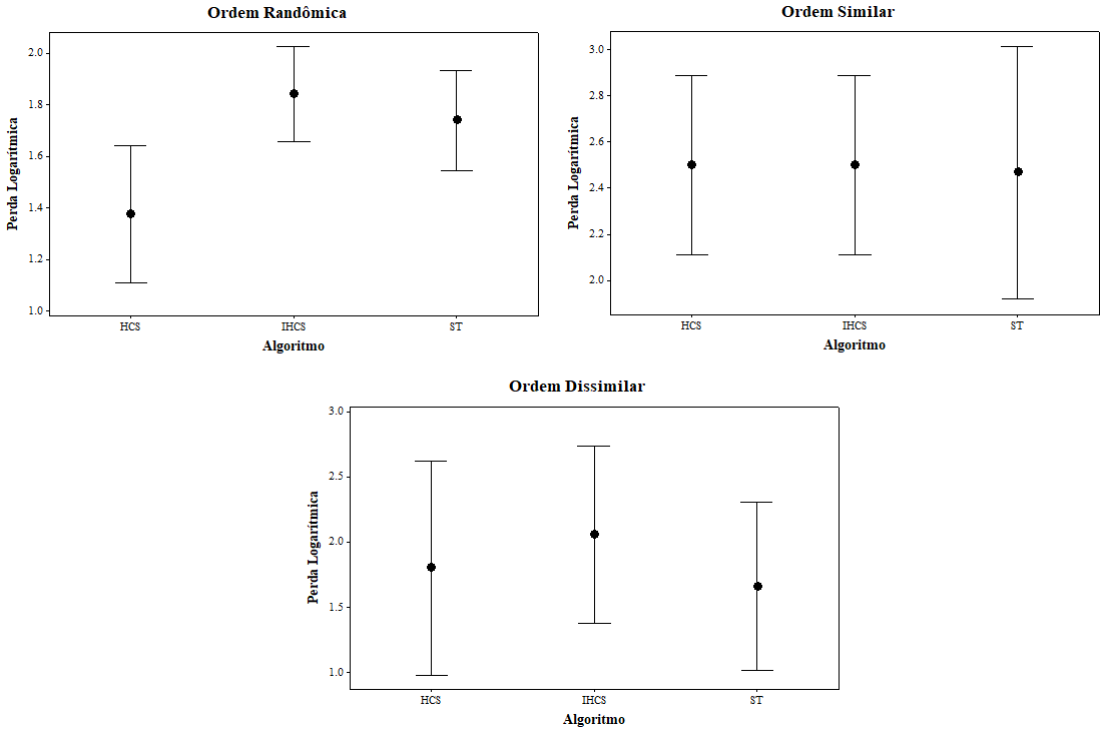

Figura 4.17: Perda logarítmica dos modelos para o conjunto de dados *Car*

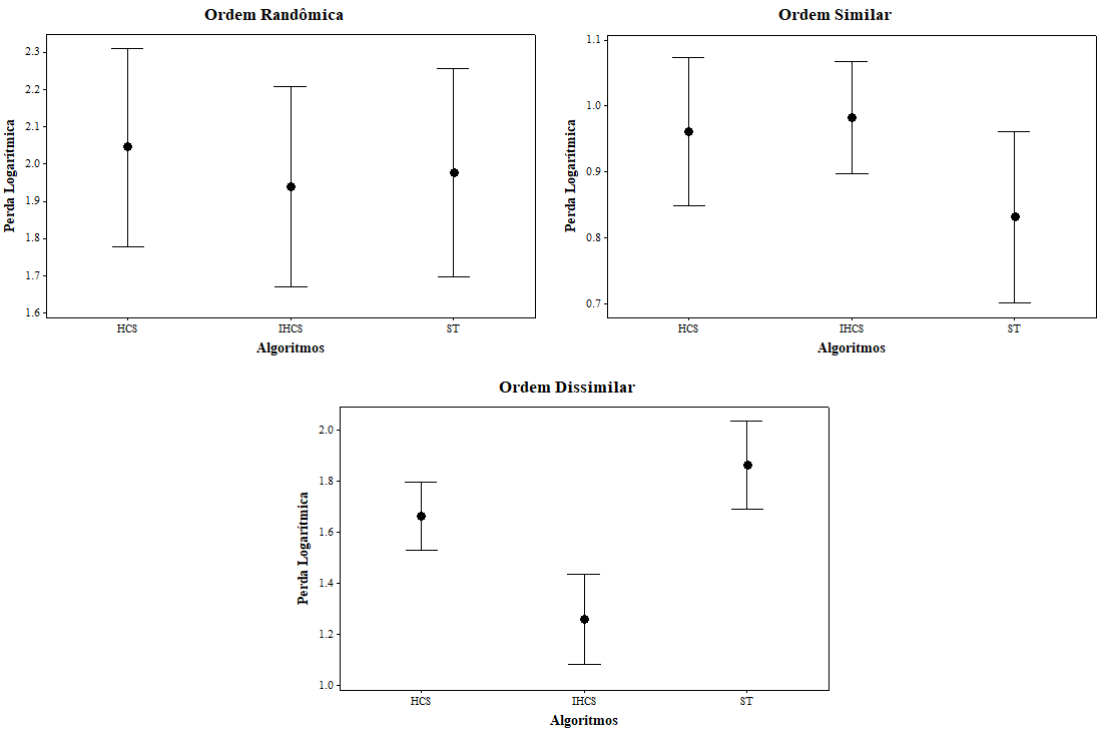

Figura 4.18: Perda logarítmica dos modelos para o conjunto de dados *Nursery*

quando a ordem é randômica. Quando a ordem é similar, o IHCS e HCS produzem redes semelhantes. No entanto, há uma aparente diferença entre o ST e o HCS. O valor de p de 0,135 é obtido utilizando o teste T. Portanto, não exitem evidências suficientes para refutar a hipótese nula e a igualdade entre as médias das amostras é aceita. Pode-se notar uma diferença clara entre as redes aprendidas por IHCS e HCS quando a ordem é dissimilar. No entanto, há uma aparente diferença entre o ST e o HCS novamente. O valor de p de 0,07 é obtido, indicando a falta de evidências suficientes para refutar a hipótese nula e a igualdade entre as médias das amostras é aceita.

Portanto, é possível notar que os algoritmos tendem a produzir redes com perda semelhante quando a ordem dos dados é similar. O contrário pode ser observado quando as instâncias são dispostas em ordem dissimilar. Nota-se também que em bases de dados sintéticas, os algoritmos tendem a produzir redes com perda diferente quando a ordem é randômica. Já para dados reais, nesta ordem, os algoritmos tendem a produzir redes mais similares.

4.2.3 Curva de Acurácia

Na Figura 4.19, são apresentados os intervalos de confiança para a média da perda logarítmica atribuídas as redes geradas durante os processos de aprendizagem no conjunto de dados *Alarm*. Nota-se que os algoritmos produzem redes com acurácia semelhante tanto na ordem randômica, quando na ordem similar. Quando dissimilar, HCS e IHCS produzem redes semelhantes, mas ST produz redes diferentes em relação ao HCS.

Na Figura 4.20, são apresentados os intervalos da média para o conjunto de dados *Asia*. Nota-se que todas as redes produzidas são divergentes. Com exceção para as redes produzidas por HCS e IHCS que possuem valor p de 0,965.

Na Figura 4.21, são apresentados os intervalos da média para o conjunto de dados *Car*. Nota-se que há uma alternância nas médias entre as redes produzidas pelos algoritmos na mesma ordem de dados. Na ordem randômica, apesar da alternância, todos os dois algoritmos incrementais produzem redes semelhantes às redes produzias por HCS. Quando a ordem é similar, IHCS, HCS e ST também possuem semelhantes, com um valor de p de 0,141 entre as amostras dos dois últimos algoritmos. Quando a ordem é dissimilar, ST, IHCS e HCS produzem redes com acurácia semelhantes, com um valor de p igual a 0,119 entre os dois últimos.

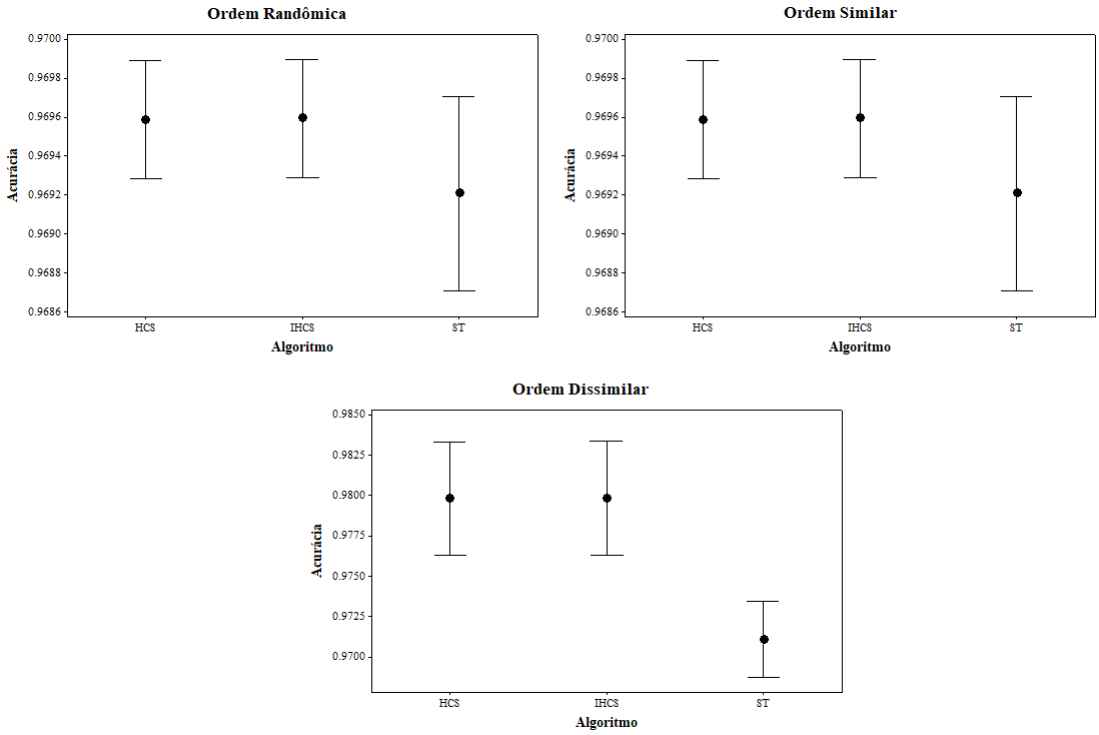

Figura 4.19: Acurácia dos modelos para o conjunto de dados *Alarm*

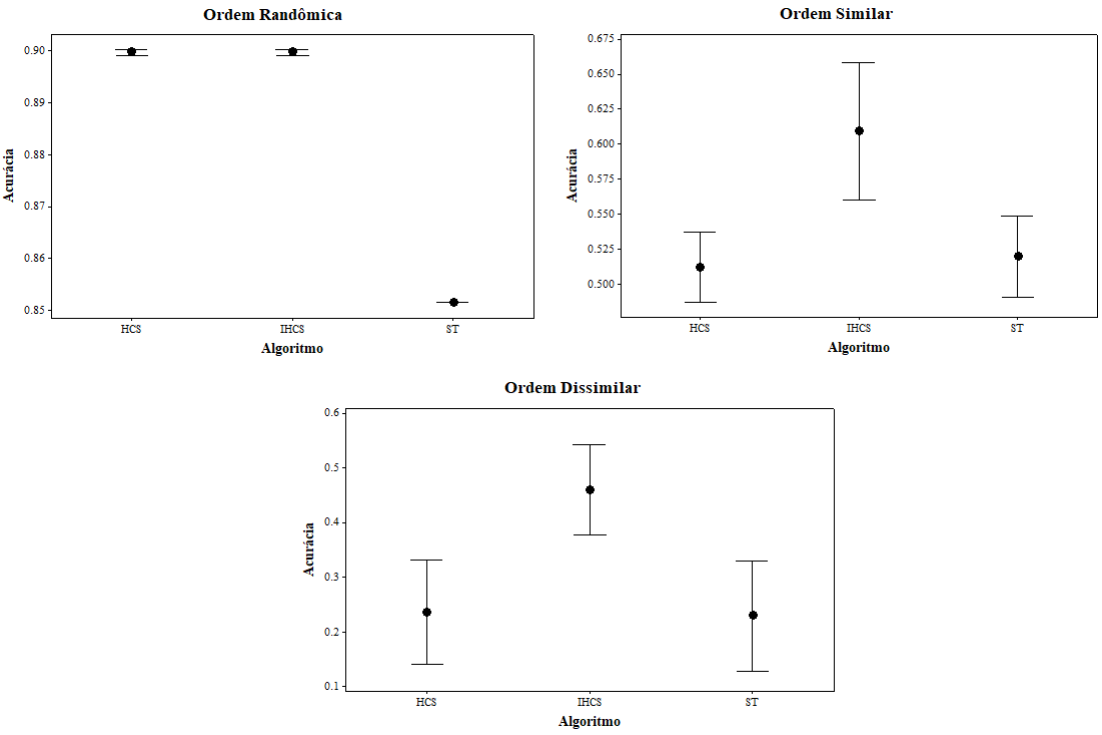

Figura 4.20: Acurácia dos modelos para o conjunto de dados *Asia*

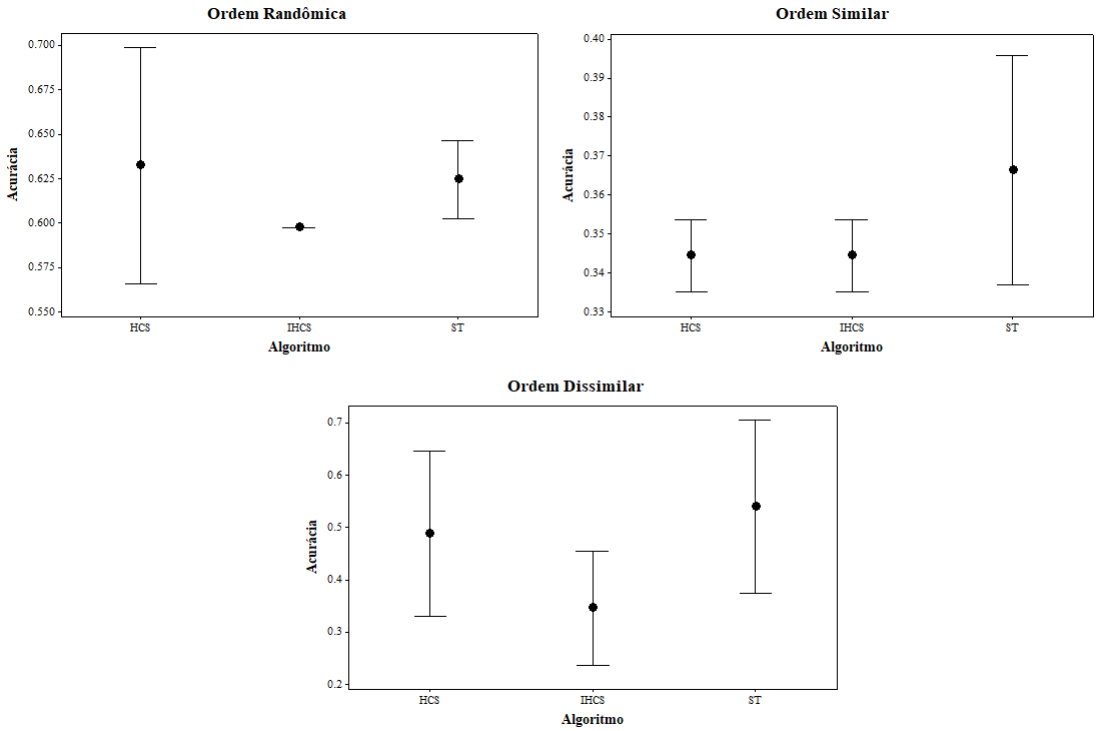

Figura 4.21: Acurácia dos modelos para o conjunto de dados *Car*

Na Figura 4.22, são apresentados os intervalos da média para o conjunto de dados *Nursery*. Percebe-se então uma semelhança entre as redes produzidas na ordem randômica e uma diferença significativa quando as redes são produzidas pelos algoritmos na ordem similar. Nesta ordem, HCS e ST produzem, apesar das médias diferentes, redes com acurácia semelhantes, atingindo um valor de p de 0,135. Quando a ordem é dissimilar, HCS e ST produzem redes semelhantes, com valor de p igual a 0,07. Já HCS e IHCS produzem redes com acurácia diferentes, dado que a hipótese nula é refutada com um p valor de 0,001.

Nota-se que quando a ordem é randômica, os algoritmos incrementais produzem acurácia semelhante ao algoritmo em lote, com exceção de ST na base de dados *Asia*. Quando a ordem é dissimilar, os algoritmos incrementais tendem a produzir redes diferentes quando às suas acurácias, alternando semelhança com o HCS entre o IHCS e ST.

4.2.4 Ameaças à Validade

A análise da validade dos experimentos tem como objetivo examinar a relação entre as conclusões alcançadas e a realidade com o objetivo de mitigar prováveis ameaças que possam

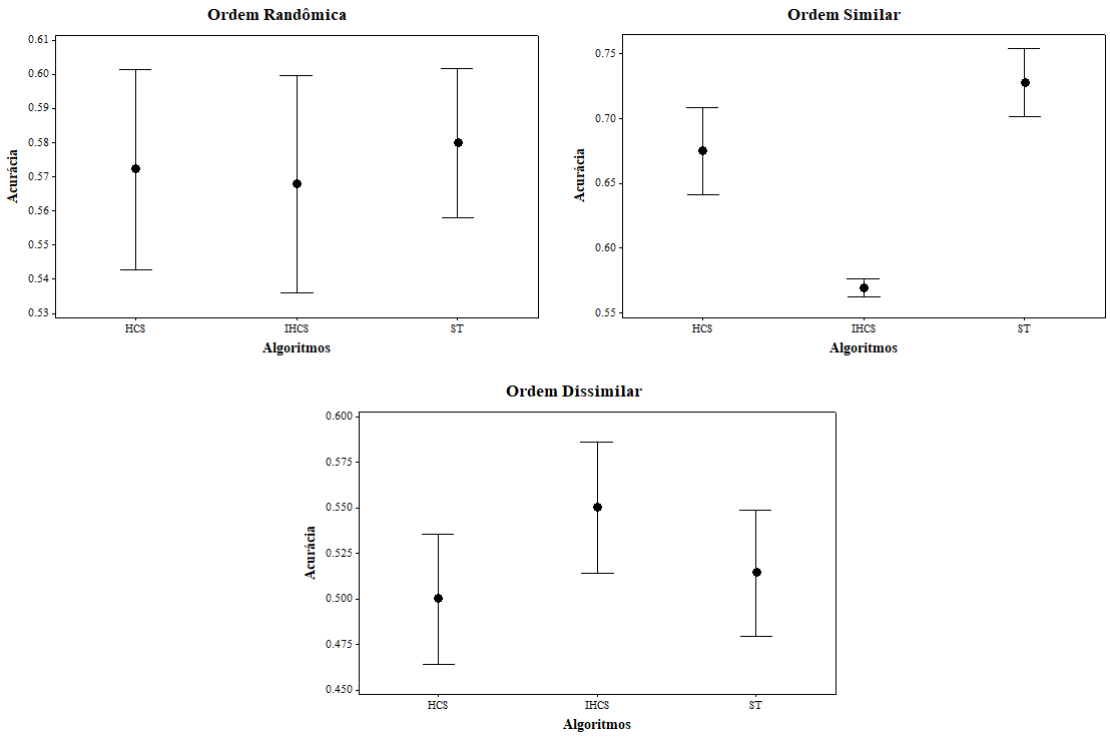

Figura 4.22: Acurácia dos modelos para o conjunto de dados *Nursery*

afetar os resultados. A seguir, os principais tipos de ameaças à validade detectados de acordo com a classificação de Wohlin et al. [57] são detalhados.

Como ameaça à validade interna, pode-se citar a implementação dos algoritmos abordados neste experimento. Apenas o código-fonte do HCS foi encontrado disponível na literatura. Os algoritmos incrementais foram implementados baseados nos pseudocódigos disponibilizados pelos autores em seus trabalhos. Ambos foram implementados e executados em ambientes idênticos e técnicas de algoritmos foram utilizadas para tornar a execução destas soluções mais ágil. Para validação da implementação, buscou-se a replicação dos resultados encontrados pelos autores, mas a maioria é impossível de ser replicado dado a falta da base de dados citada nos experimentos.

Como ameaça à validade de constructo, são citadas a falta de replicações dos ensaios e a explicação dos dados contendo informações reais que, apesar de complexos, ainda sim, são simples se comparados aos dados coletados no cotidiano real. Apesar da utilização de dados com informações sobre o mundo real, há uma diferença entre as características desses dados e vários outros conjuntos de dados do mundo real, como dados faltantes e com

altos ruídos. Isto também afeta a generalização dos resultados, constituindo uma ameaça à validade externa.

4.3 Avaliação de Adaptação das Soluções Incrementais às Complexidades de Domínio

Um experimento fatorial completo é realizado para avaliar o impacto do contexto na qualidade da rede aprendida pelas soluções. Todas as combinações são testadas neste design experimental e assim, é possível descobrir o efeito isolado de cada fator e de suas combinações. O conjunto de ensaios para este experimento é descrito no Apêndice B. Para avaliar estatisticamente e validar as hipóteses de influência das características abordadas nas soluções aprendidas, uma análise de variância dos dados é realizada.

Nos próximos tópicos, busca-se a compreensão de influência das variações e complexidades do contexto, referentes ao conjunto de dados, ao tamanho do passo, à ordenação das instâncias e à rede inicial, nas métricas de qualidade deste trabalho. Também é avaliado se os algoritmos explicam variações diferentes nestas métricas com o objetivo de encontrar alguma diferença significativa entre eles.

Para a análise dos resultados, gráficos de pareto são utilizados para destacar os valores absolutos dos efeitos padronizados, desde o maior até o menor efeito. Os efeitos padronizados são estatísticas T que testam a hipótese nula de que o efeito não é significante, considerando a métrica resposta. O gráfico também exibe uma linha de referência para indicar quais efeitos são estatisticamente significativos. O nível de significância de 5% é utilizado.

4.3.1 Pontuação Estrutural

Na Figura 4.23, é apresentado o gráfico de pareto contendo os valores padronizados dos efeitos na pontuação DCM. De todos os efeitos possíveis, os que podem ser considerados significantes são os efeitos principais referentes ao conjunto de dados, ao tamanho do passo, à ordem dos dados e aos algoritmos. Além desses efeitos principais, pode-se destacar os efeitos de interações entre o conjunto de dados, a ordem das instâncias e o tamanho do passo.

Nota-se na Figura 4.23 que a rede inicial não possui qualquer efeito significativo na pontuação, seja ele principal ou referente a alguma interação deste com qualquer outro fator. Sendo assim, o efeito da rede inicial na pontuação das redes finais obtidas é descartado.

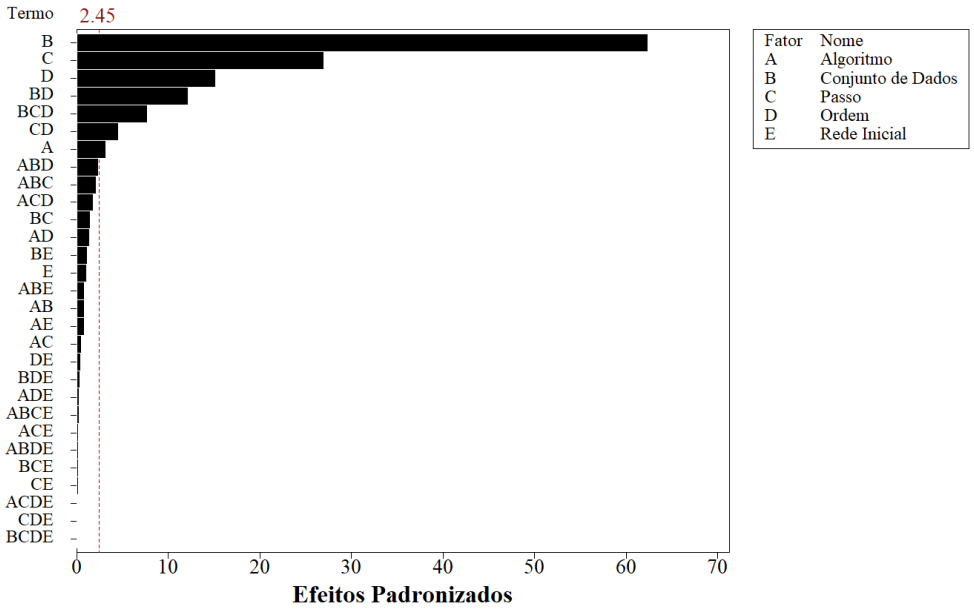

Figura 4.23: Gráfico de pareto dos efeitos padronizados na pontuação estrutural

Na Figura 4.24, é apresentada a ordem e sentido dos efeitos principais. Nota-se que a alteração nos níveis da rede inicial causam pouca alteração na média do DCM. O fator algoritmo causa um pouco mais e por isso, dado o efeito dos outros fatores, já é considerado significante. Nota-se também que este efeito é positivo, ou seja, quando altera-se de IHCS para ST, a média DCM aumenta, o que indica piores estruturas. Como foi visto, quanto mais baixo o DCM, mais a rede explica sobre a distribuição contida nos dados.

Ainda na Figura 4.24, nota-se que o conjunto de dados e o tamanho do passo causam uma influência grande e positiva nos dados. Alterar de *Alarm* para *Nursery* ou aumentar os passos de 100 para 1000 e para 4000, consequentemente, produz redes com maior DCM. Já a ordem segue o sentido inverso. Quando a ordem é randômica, em média, são produzidas redes com um DCM mais alto, enquanto quando dissimilar, são produzidas as redes com menor DCM.

Na Figura 4.25, verifica-se a normalidade, variação e independência dos erros produzidos pelo modelo gerado pela análise de variância para análise dos efeitos. Analisando o gráfico

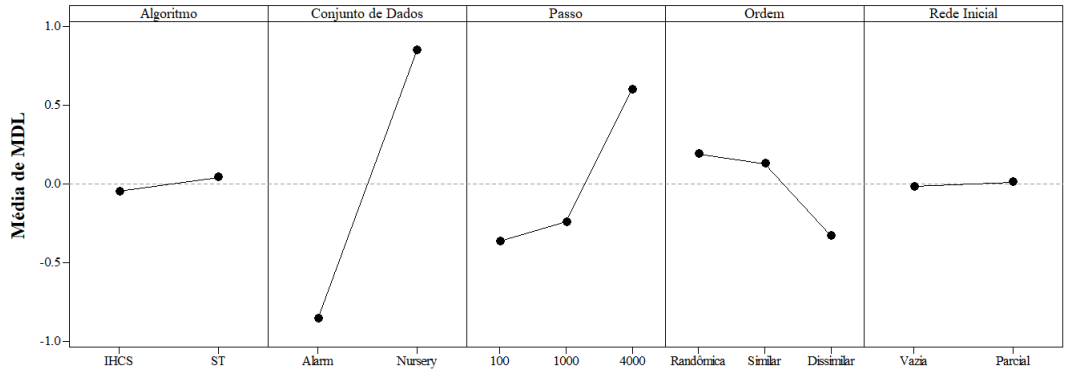

Figura 4.24: Gráfico de efeitos significantes de fatores na pontuação estrutural

de probabilidade normal e o histograma, nota-se que os resíduos seguem uma distribuição normal. Usando o teste de normalidade de Ryan-Joiner, similar ao Shapiro-Wilk, um valor de p maior que 1 é obtido, indicando a falta de evidências para refutar a hipótese nula de que a distribuição analisada possui uma distribuição normal.

Ainda na Figura 4.25, pode-se verificar que, quando os resíduos são plotados junto com os valores ajustados da métrica, não há nenhum padrão quanto à sua variância, mantendo a suposição de homocedasticidade necessária. Quando o gráfico em que são plotados os resíduos na ordem que foram obtidos também é verificado, nota-se que não há relação entre eles, mantendo a suposição de independência dos resíduos.

Dado a validade do modelo usado para a análise de variância, supõem-se que, com significância de 5%, todos os fatores analisados explicam alguma variação nas pontuações DCM das redes obtidas pelos algoritmos, com exceção da rede inicial. Além disso, supõem-se também que o algoritmo ST produz pontuações DCM maiores que o IHCS.

4.3.2 Diferença Estrutural

A seguir, são descritos os resultados experimentais sobre as variações do contexto abordadas nas métricas predição, cobertura e valor F da estrutura gerada pelos algoritmos incrementais sobre as estruturas geradas pelos algoritmos em lote.

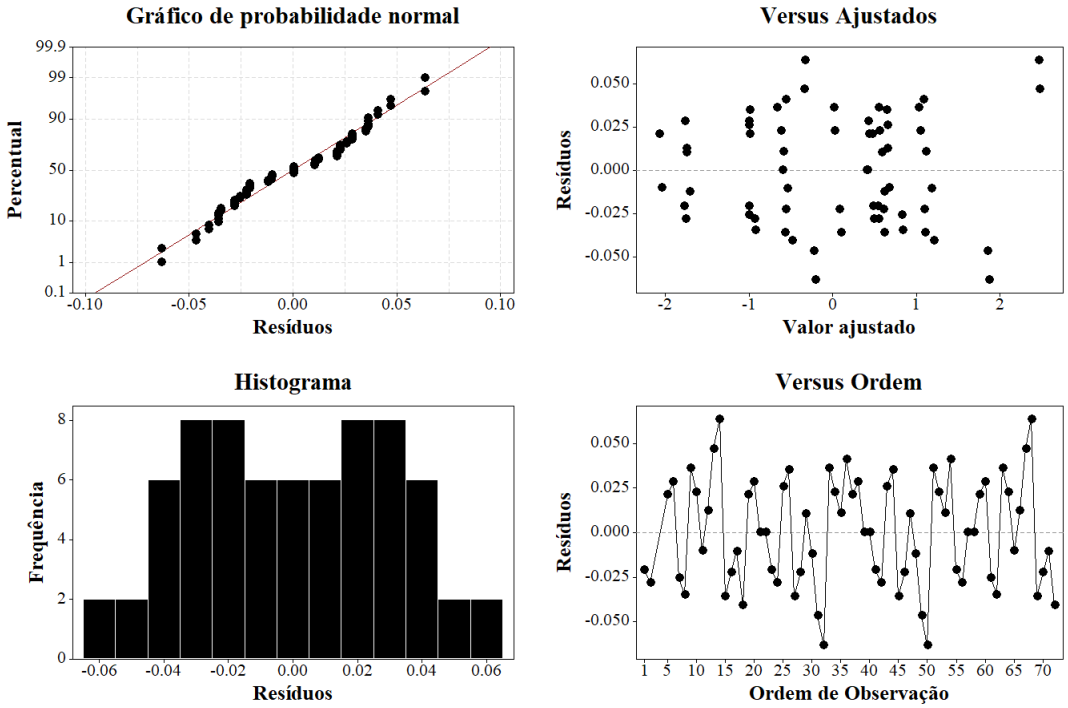

Figura 4.25: Gráfico de resíduos do modelo sobre efeitos significantes na pontuação estrutural

Precisão

O gráfico de pareto contendo os valores padronizados dos efeitos na pontuação DCM é apresentado na Figura 4.26. Nota-se que todos os efeitos principais, isto é, referentes unicamente aos fatores, podem ser considerados significantes. Ao contrário dos efeitos na pontuação DCM, os algoritmos agora possuem um efeito ainda mais significante.

Nota-se na Figura 4.26 que poucos são os efeitos que não são significantes para a precisão estrutural. A maioria deles, efeitos de interações entre 3 ou 4 fatores. No entanto, a maioria dos efeitos que consideram a rede inicial como um fatores da interação é insignificante.

A ordem e sentido dos efeitos principais é apresentado na Figura 4.27. Nota-se que a alteração nos níveis da rede inicial causam pouca alteração na média da precisão, mas ainda sim é considerado significativo. O fator algoritmo, também considerado significante, possui um efeito negativo, ou seja, quando altera-se de IHCS para ST, a precisão cai, o que indica estruturas menos precisas com relação às geradas pelos algoritmos em lote.

Ainda na Figura 4.27, nota-se que o tamanho do passo e a ordem das instâncias causam

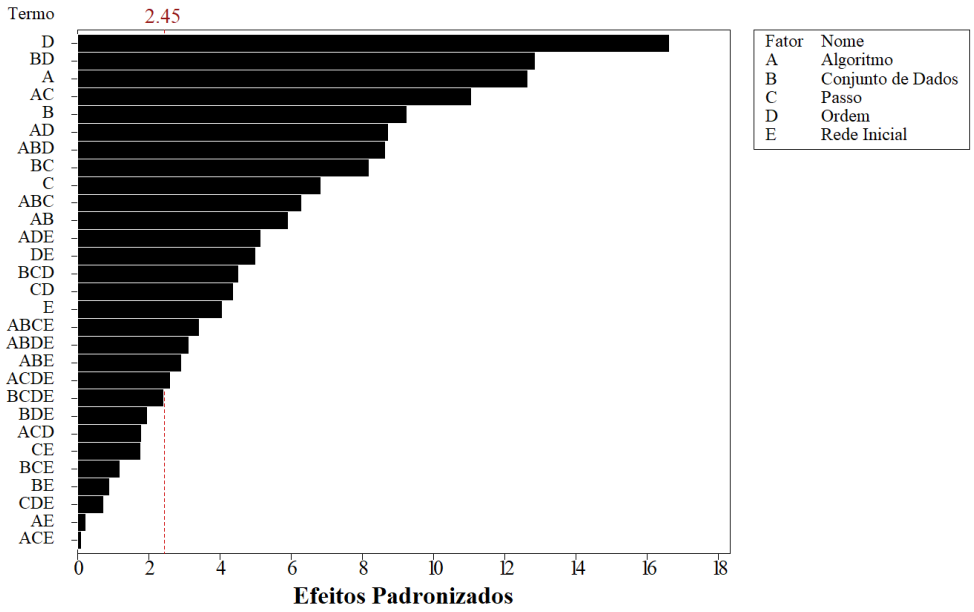

Figura 4.26: Gráfico de pareto dos efeitos padronizados na precisão estrutural

uma influência similar nos dados. Alterar os nível iniciais para os intermediários e logo após, para os finais, causa um efeito em forma de "V"nos dados, indicando que os fatores intermediários produzem, em média, as redes com piores precisão. O conjunto de dados, por sua vez, enquanto *Alarm*, produz redes piores que quando *Nursery*.

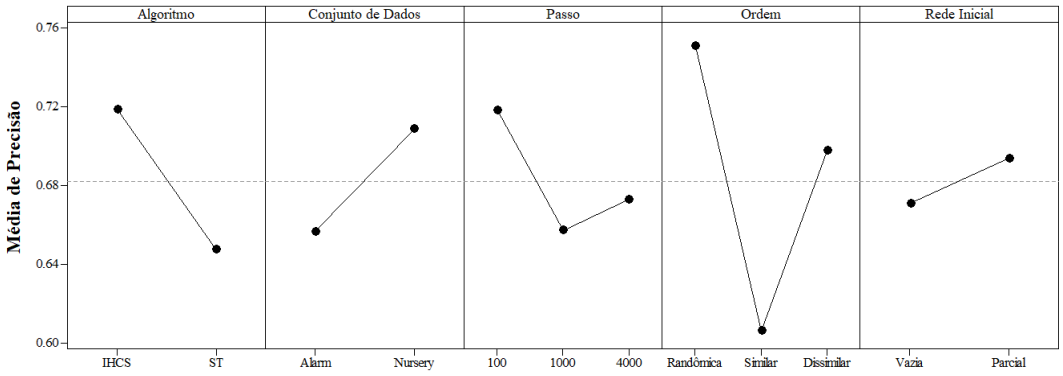

Figura 4.27: Gráfico de efeitos significantes de fatores na precisão estrutural

A normalidade, variação e independência dos erros produzidos pelo modelo gerado pela anova para análise dos efeitos pode ser verificada analisando a Figura 4.28. Verificando o gráfico de probabilidade normal e o histograma, entende-se que os resíduos seguem uma

distribuição normal. Usando o teste de normalidade de Ryan-Joiner, similar ao Shapiro-Wilk, um valor de p maior que 1 é obtido, indicando a falta de evidências para refutar a hipótese nula de que a distribuição analisada possui uma distribuição normal.

Ainda na Figura 4.28, nota-se que, quando os resíduos são plotados junto com os valores ajustados da métrica, não há nenhum padrão quanto à sua variância, mantendo a suposição de homocedasticidade. Quando verifica-se também o gráfico em que são plotados os resíduos na ordem que foram obtidos, nota-se que não há relação entre eles, mantendo a suposição de independência dos resíduos.

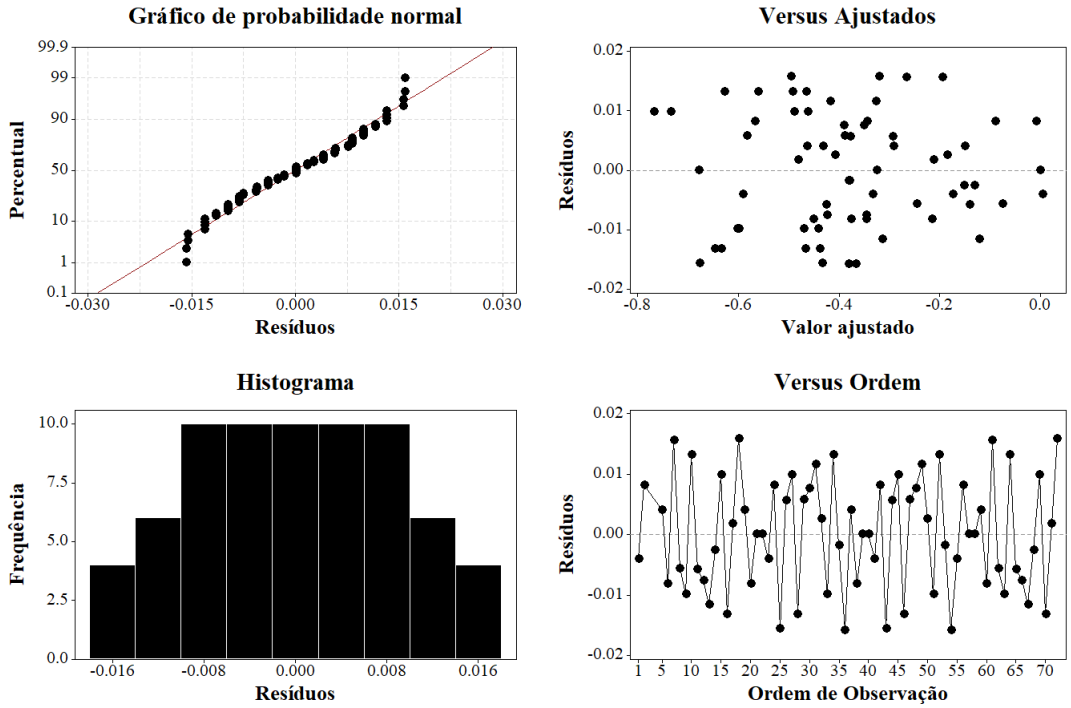

Figura 4.28: Gráfico de resíduos do modelo sobre efeitos significantes na precisão estrutural

Com o modelo válido, supõem-se que, com significância de 5%, todos os fatores analisados explicam alguma variação na precisão estrutural das redes obtidas pelos algoritmos. Além disso, também supõem-se que o algoritmo ST produz redes com precisão estrutural piores que o IHCS.

Cobertura

O gráfico de pareto contendo os valores padronizados dos efeitos na cobertura estrutural é apresentado na Figura 4.29. Nota-se que, de todos os efeitos possíveis, os que podem ser

considerados significantes são apenas os efeitos principais referentes à ordem dos dados e aos algoritmos. Além desses efeitos principais, destaca-se os efeitos de interações entre o conjunto de dados, a ordem das instâncias e os algoritmos.

Analisando a Figura 4.29, nota-se que a rede inicial, o conjunto de dados e o tamanho do passo não possuem qualquer efeito significativo na cobertura. Para o fator relativo ao tamanho do passo e a rede inicial, seus efeitos são considerados insignificantes, sejam eles principal ou referentes a alguma interação destes com quaisquer outros fatores. Sendo assim, os efeito do tamanho do passo e a rede inicial na cobertura estrutural das redes finais obtidas são descartados.

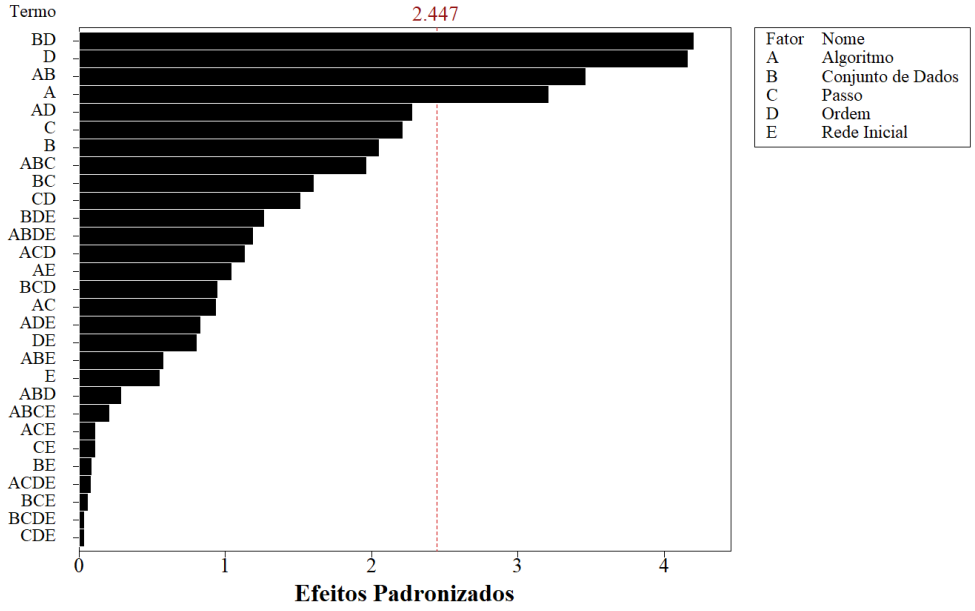

Figura 4.29: Gráfico de pareto dos efeitos padronizados na cobertura estrutural

Na Figura 4.30, são apresentados a ordem e sentido dos efeitos principais considerados significativos. Nota-se que a alteração nos níveis do fator algoritmo causa um efeito bastante notável na cobertura. Este efeito é negativo, ou seja, quando altera-se de IHCS para ST, a média DCM diminui, o que indica piores estruturas com relação às geradas pelos algoritmos em lote.

Ainda na Figura 4.30, percebe-se que o conjunto de dados e a ordem dos dados causam efeitos contrários nos dados. Enquanto o conjunto de dados causa um efeito menor e negativo, a ordem dos dados causa o maior efeito e positivo na cobertura estrutural das redes

aprendidas.

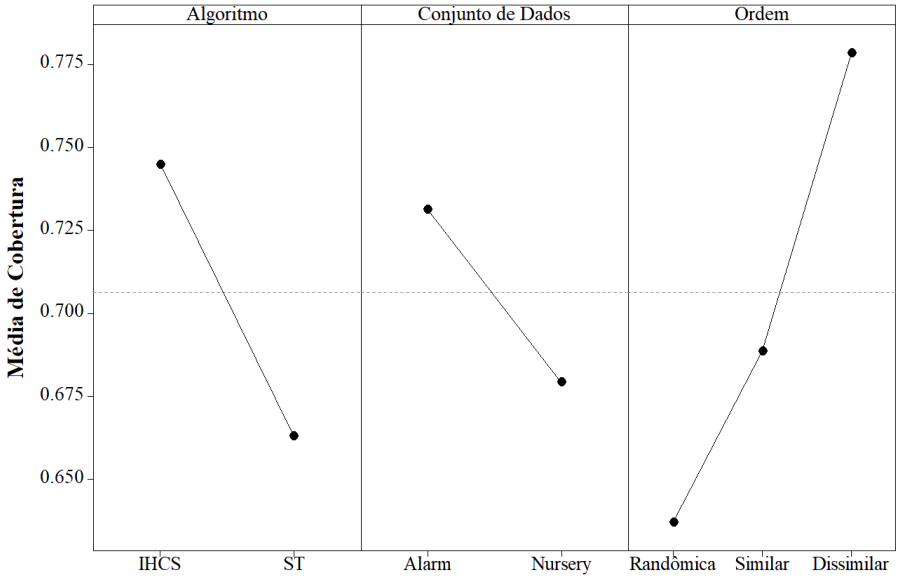

Figura 4.30: Gráfico de efeitos significantes de fatores na cobertura estrutural

Na Figura 4.31, a normalidade, variação e independência dos erros produzidos pelo modelo gerado pela anova para análise dos efeitos podem ser verificados. Nota-se pelo gráfico de probabilidade normal e pelo histograma que os resíduos seguem uma distribuição normal. Usando o teste de normalidade de Ryan-Joiner, similar ao Shapiro-Wilk, um valor de p maior que 1 é obtido, indicando a falta de evidências para refutar a hipótese nula de que a distribuição analisada possui uma distribuição normal.

Ainda analisando a Figura 4.31, nota-se que, quando os resíduos são plotados junto com os valores ajustados da métrica, não há nenhum padrão quanto à sua variância, mantendo a suposição de homocedasticidade. Verificando também o gráfico em que são plotados os resíduos na ordem que foram obtidos, nota-se que não há relação entre eles, mantendo a suposição de independência dos resíduos.

Dado a validade do modelo, supõem-se que, com significância de 5%, os fatores algoritmo, conjunto de dados e ordem de instâncias explicam alguma variação nas coberturas estruturais das redes obtidas pelos algoritmos. Pode também supor que os fatores tamanho do passo e rede inicial não explicam qualquer variação na métrica analisada. Além disso, supõem-se também que o algoritmo ST produz pontuações de redes com cobertura estrutural pior que o IHCS.

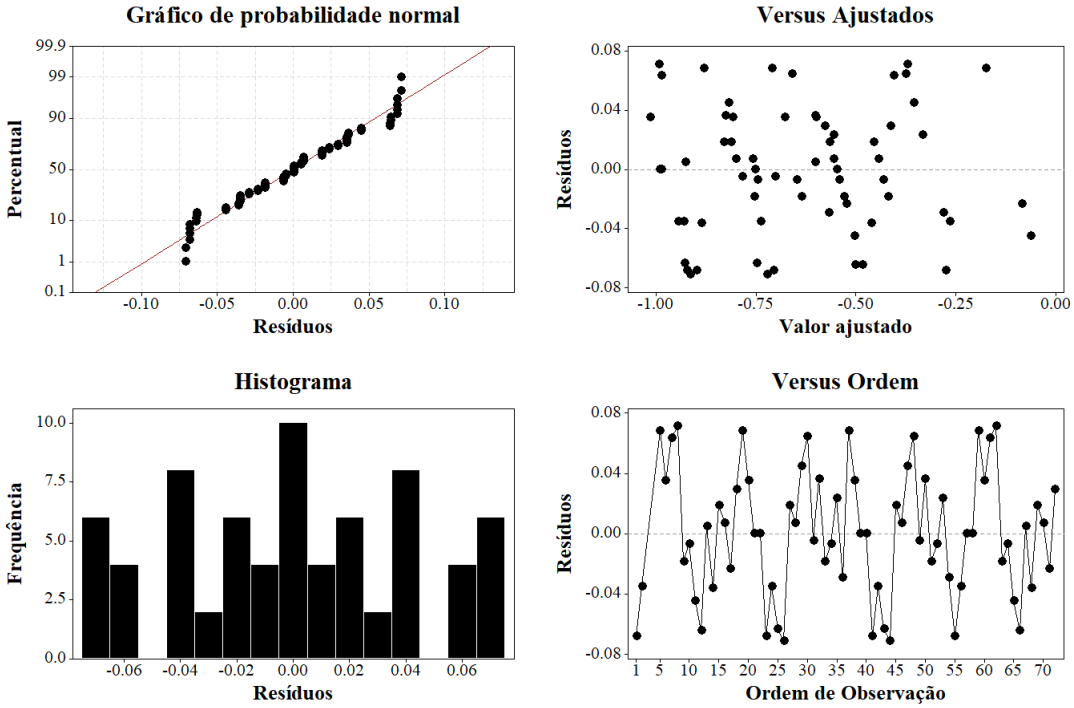

Figura 4.31: Gráfico de resíduos do modelo sobre efeitos significantes na cobertura estrutural

Valor F

Na Figura 4.32, é apresentado o gráfico de pareto contendo os valores padronizados dos efeitos no valor F. Nota-se que de todos os efeitos possíveis, apenas um efeito é significativo, sendo explicativo de uma interação entre o efeito do conjunto de dados e a ordem das instâncias. Os fatores sobre o algoritmo, o tamanho do passo e rede inicial não possuem qualquer efeito significativo no valor F, seja ele principal ou referente a alguma interação deste com qualquer outro fator, sendo assim, descartados.

A normalidade, variação e independência dos erros produzidos pelo modelo gerado pela anova podem ser verificados na Figura 4.33. Nota-se, pelo gráfico de probabilidade normal e pelo histograma, que os resíduos seguem uma distribuição normal. Usando o teste de normalidade de Ryan-Joiner, um valor de p maior que 1 é obtido, indicando a falta de evidências para refutar a hipótese nula de que a distribuição analisada possui uma distribuição normal.

Quando os resíduos são plotados junto com os valores ajustados da métrica, não há nenhum padrão quanto à sua variância, mantendo a suposição de homocedasticidade. Verificando também o gráfico em que são plotados os resíduos na ordem que foram obtidos,

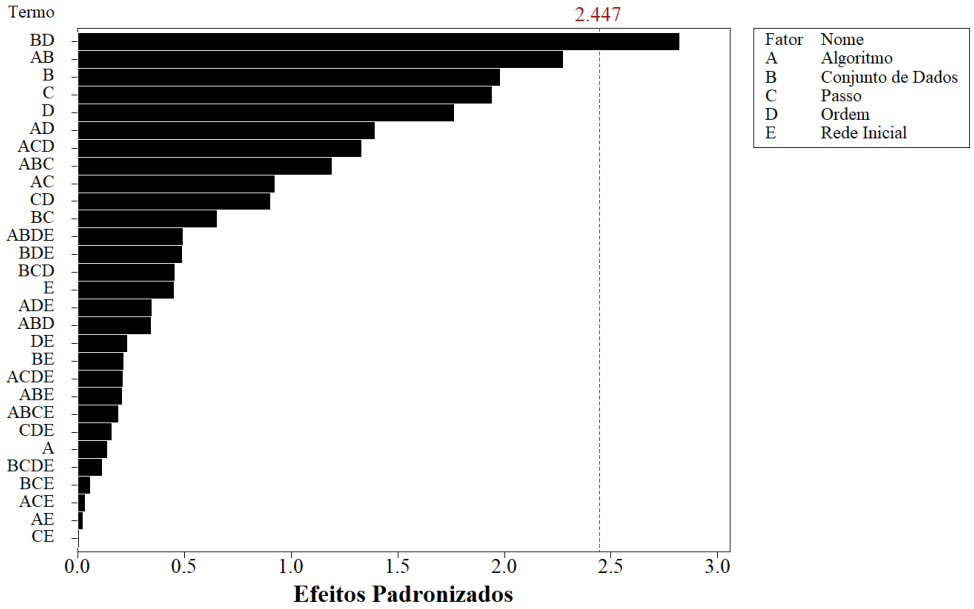

Figura 4.32: Gráfico de pareto dos efeitos padronizados no valor F

nota-se que não há relação entre eles, mantendo a suposição de independência dos resíduos.

Dado a validade do modelo, supõem-se que, com significância de 5%, apenas os fatores sobre o conjunto de dados e a ordem de instâncias explicam alguma variação nos valores F das redes obtidas pelos algoritmos. Todos os outros efeitos são insignificantes. Além disso, não pode-se afirmar que o algoritmo produz a melhor rede, mas supõem-se que nenhum dos algoritmos causa influência significativa.

4.3.3 Curva de Aprendizagem

Na Figura 4.34, é apresentado o gráfico de pareto contendo os valores padronizados dos efeitos na perda logarítmica das redes. Percebe-se que de todos os efeitos possíveis, os que podem ser considerados significantes são os efeitos principais referentes ao conjunto de dados, ao tamanho do passo, à ordem dos dados e aos algoritmos. Além desses efeitos principais, destacam-se os efeitos de várias interações entre esses fatores.

Analisando a Figura 4.34, entende-se que a rede inicial não possui qualquer efeito significativo na perda logarítmica das redes, seja ele principal ou referente a alguma interação deste com qualquer outro fator, sendo assim, descartada como fator significativo.

Na Figura 4.35, são apresentadas a ordem e sentido dos efeitos principais. Nota-se que

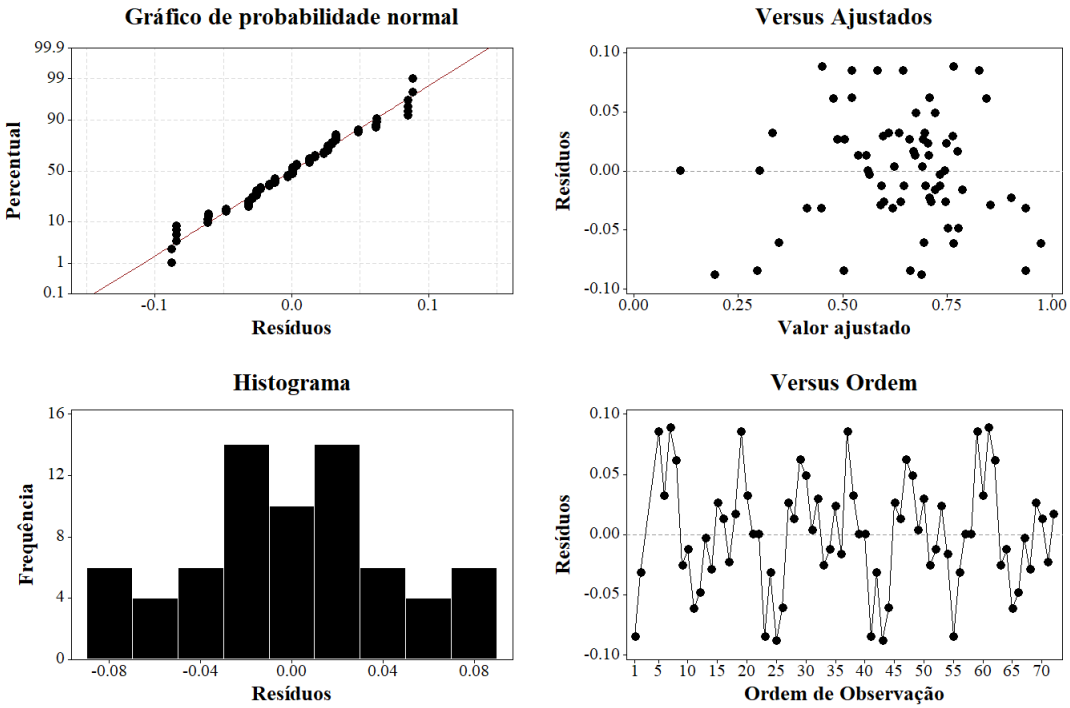

Figura 4.33: Gráfico de resíduos do modelo sobre efeitos significantes no valor F

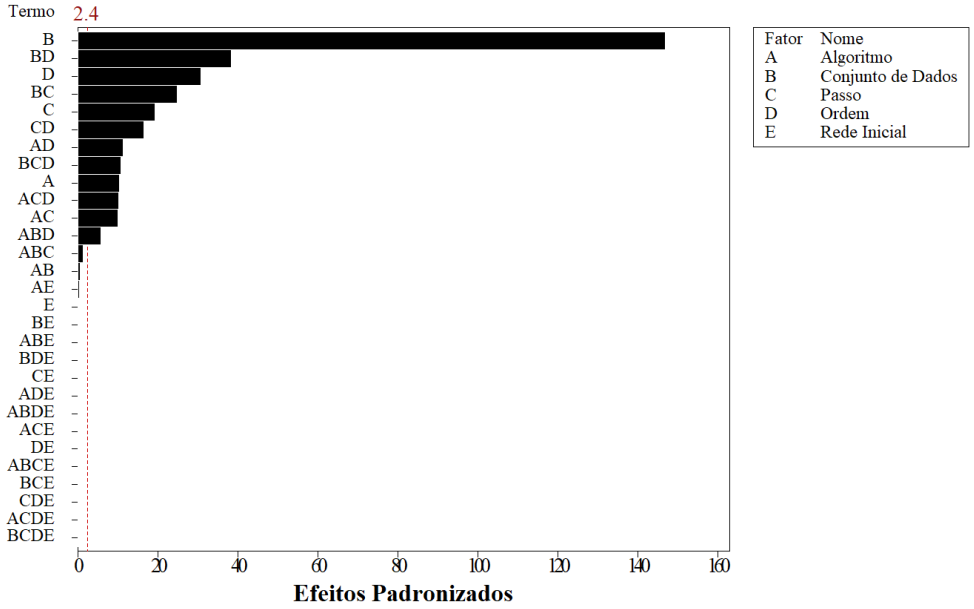

Figura 4.34: Gráfico de pareto dos efeitos padronizados na perda logarítmica

a alteração nos níveis da rede inicial causam pouca alteração na média da perda. O fator algoritmo causa um pouco mais e por isso, dado o efeito dos outros fatores, já é considerado significante. Nota-se também que este efeito é negativo, ou seja, quando altera-se de IHCS para ST, a média da perda diminui, o que indica melhores generalizações. Como já visto, quanto menor a perda, melhor é o desempenho de classificação do modelo gerado.

Analisando a Figura 4.35, nota-se que o conjunto de dados causa uma influência grande e positiva nos dados. Alterar de *Alarm* para *Nursery*, a perda aumenta significantemente, provavelmente explicada pela complexidade dos dados. Aumentar os passos de 100 para 1000 e para 4000, consequentemente, também produz redes com maior perda. A ordem das instâncias produz, em média, redes similares quando é randômica ou dissimilar, melhorando quando a ordem é similar.

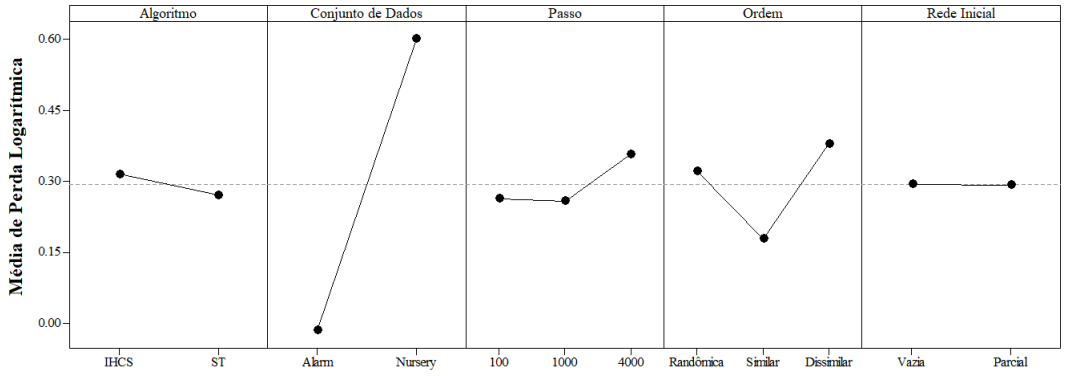

Figura 4.35: Gráfico de efeitos significantes de fatores na perda logarítmica

Interpretando a Figura 4.36, pode-se verificar a normalidade, variação e independência dos erros produzidos pelo modelo gerado pela anova para análise dos efeitos. Nota-se, pelo gráfico de probabilidade normal e pelo histograma, que os resíduos não seguem perfeitamente uma distribuição normal. No entanto, usando o teste de normalidade de Kolmogorov-Smirnov, um valor de p de 0,03 é obtido, indicando, com significância de 1%, a falta de evidências para refutar a hipótese nula de que a distribuição analisada possui uma distribuição normal.

Pode-se notar também que, quando os resíduos são plotados junto com os valores ajustados da métrica, não há nenhum padrão quanto à sua variância, mantendo a suposição de homocedasticidade. Há apenas um padrão de distribuição do lado esquerdo. Isto é resultado

da anormalidade da variável resposta. No entanto, anova também é robusta para a análise de dados anormais. Quando verifica-se também o gráfico em que são plotados os resíduos na ordem que foram obtidos, nota-se que não há relação entre eles, mantendo a suposição de independência dos resíduos.

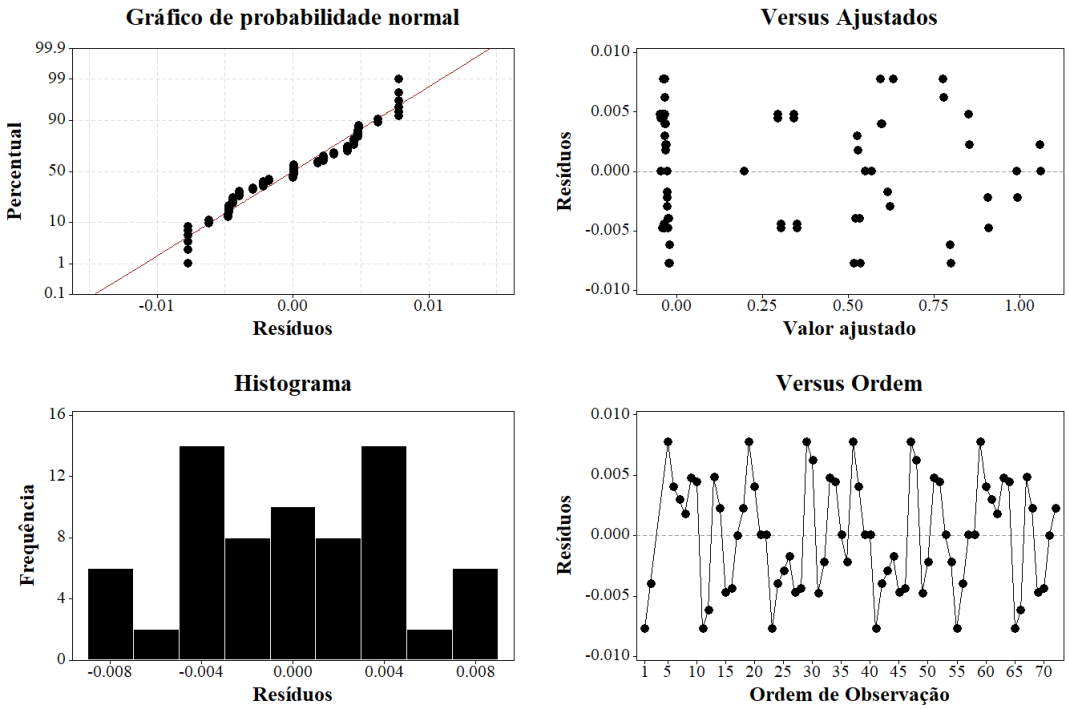

Figura 4.36: Gráfico de resíduos do modelo sobre efeitos significantes na perda logarítmica

Dado a validade do modelo, supõem-se que, com significância de 1%, todos os fatores analisados explicam alguma variação nas perdas logarítmicas das redes obtidas pelos algoritmos, com exceção da rede inicial. Além disso, supõem-se também que o algoritmo ST produz perdas menores que o IHCS.

4.3.4 Curva de Acurácia

O gráfico de pareto contendo os valores padronizados dos efeitos na acurácia dos modelos aprendidos é apresentado na Figura 4.37. Nota-se que de todos os efeitos possíveis, os que podem ser considerados significantes são os efeitos principais referentes ao conjunto de dados, à ordem dos dados e aos algoritmos. Além desses efeitos principais, destacam-se os efeitos de interações entre estes fatores e o tamanho do passo, dando significância a este

último fator.

Nota-se, na Figura 4.37, que a rede inicial não possui qualquer efeito significativo na acurácia, seja ele principal ou referente a alguma interação deste com qualquer outro fator. Sendo assim, descarta-se o efeito da rede inicial na acurácia das redes finais obtidas.

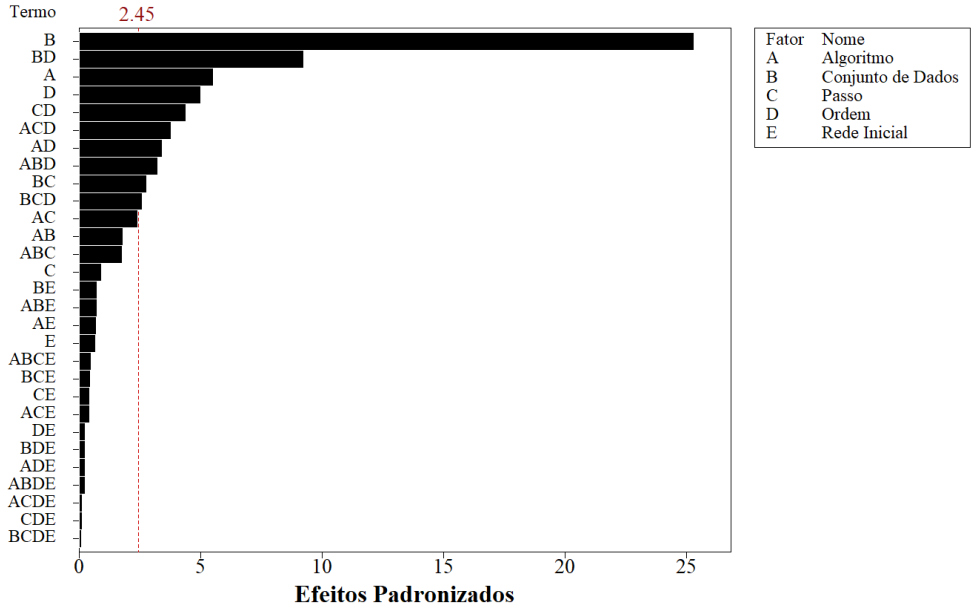

Figura 4.37: Gráfico de pareto dos efeitos padronizados na acurácia

A ordem e sentido dos efeitos principais são analisados interpretando a Figura 4.38. Nota-se que a alteração nos níveis da rede inicial causam pouca alteração na média da acurácia. O fator algoritmo, por sua vez, causa um efeito positivo, ou seja, quando altera-se de IHCS para ST, a acurácia aumenta, o que indica melhor desempenho de generalização de novos dados.

Ainda na Figura 4.38, nota-se que o conjunto de dados causa uma influência grande e negativa nos dados. Alterar de *Alarm* para *Nursery* produz redes com acurácia média bastante piores, consequentemente, com pior generalização de novos dados. Já a ordem das instâncias e o tamanho do passo seguem o sentido inverso. Quando a ordem é randômica ou o passo é 100, em média, são produzidas redes com uma acurácia mais baixa, enquanto que quando dissimilar ou o passo é 4000, são produzidas as redes com melhor acurácia.

Analisando a Figura 4.39, é possível verificar a normalidade, variação e independência dos erros produzidos pelo modelo gerado pela anova para análise dos efeitos. Nota-se pelo

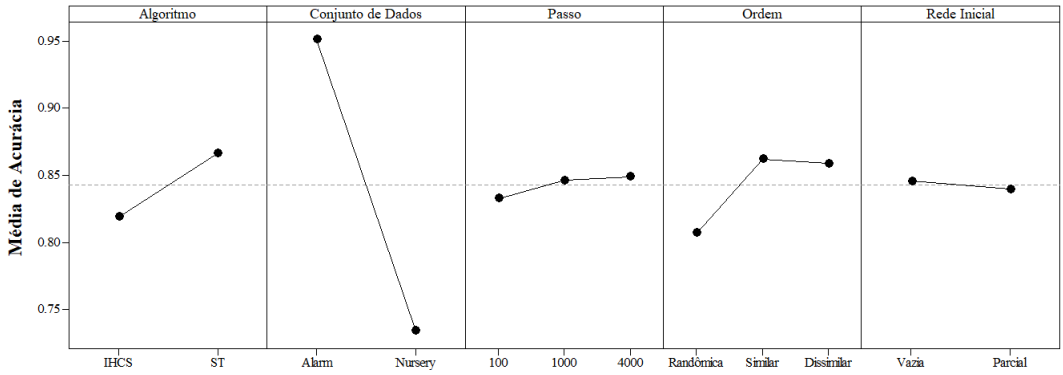

Figura 4.38: Gráfico de efeitos significantes de fatores na acurácia

gráfico de probabilidade normal e pelo histograma que os resíduos não seguem perfeitamente uma distribuição normal. No entanto, usando o teste de normalidade de Kolmogorov-Smirnov, o valor de p de 0,038 é obtido, indicando, com significância de 1%, a falta de evidências para refutar a hipótese nula de que a distribuição analisada possui uma distribuição normal.

Ainda verificando a Figura 4.39, nota-se que, quando os resíduos são plotados junto com os valores ajustados da métrica, não há nenhum padrão quanto à sua variância, mantendo a suposição de homocedasticidade. Há apenas um padrão de distribuição do lado direito. Isto indica a anormalidade da variável resposta. No entanto, anova adequa-se bem a dados anormais. Quando verifica-se também o gráfico em que são plotados os resíduos na ordem que foram obtidos, nota-se que não há relação entre eles, mantendo a suposição de independência dos resíduos.

Dado a validade do modelo, supõem-se que, com significância de 1%, todos os fatores analisados explicam alguma variação na acurácia das redes obtidas pelos algoritmos, com exceção da rede inicial. Além disso, supõem-se também que o algoritmo ST produz redes, em média, com acurácia melhor que o IHCS.

4.3.5 Ameaças à Validade

Nesta seção, as ameças à validade dos experimentos são analisadas. A seguir, os principais tipos de ameaças à validade detectados de acordo com a classificação de Wohlin et al. [57] são detalhados.

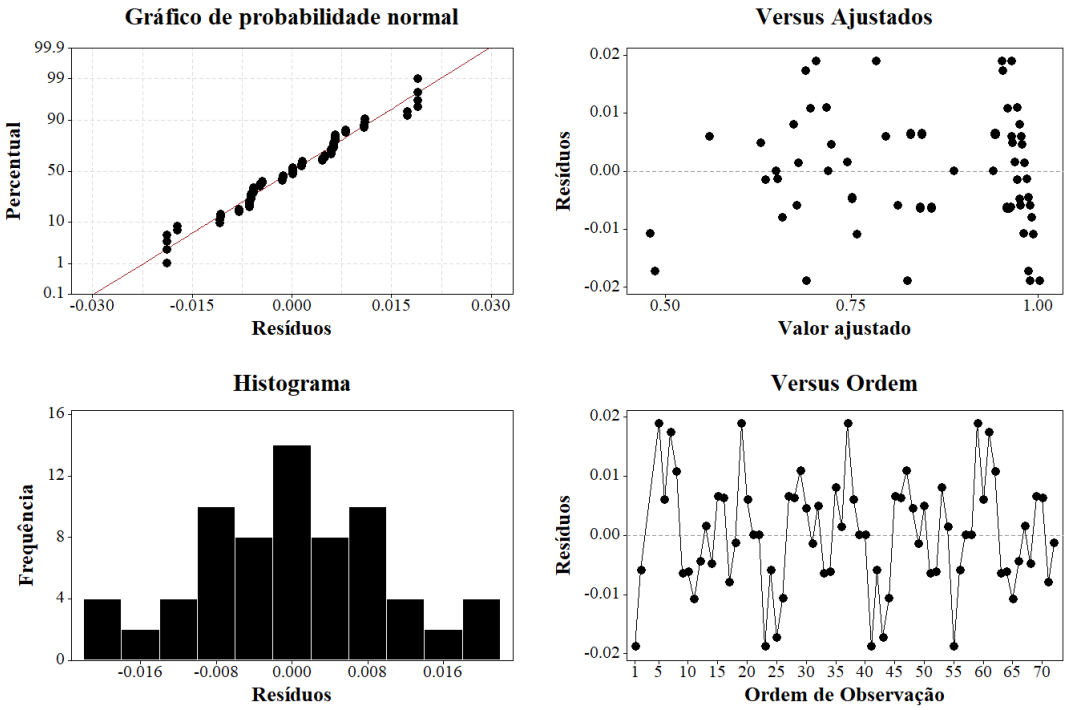

Figura 4.39: Gráfico de resíduos do modelo sobre efeitos significantes na acurácia

Como ameaça à validade interna, pode-se citar a implementação dos algoritmos abordados neste experimento. O IHCS e o ST foram implementados baseados nos pseudocódigos disponibilizados pelos autores em seus trabalhos. Ambos foram implementados e executados em ambientes idênticos e técnicas de algoritmos foram utilizadas para tornar a execução destas soluções mais ágil. Buscou-se a replicação dos resultados encontrados pelos autores para validação da implementação, mas a maioria é impossível de ser replicado dado a falta da base de dados usada nos experimentos.

Como ameaça à validade de conclusão, é possível citar a baixa significância dos resultados sobre os efeitos que explicam alguma variância nas métricas de acurácia e de perda logarítmica. A anormalidade dos dados, considerando um nível de significância de 5%, é o principal motivo desta ameça.

Como ameaça à validade de constructo, cita-se a falta de replicações dos ensaios e a explicação dos dados contendo informações reais que, apesar de complexos, ainda sim, são simples se comparados aos dados coletados no cotidiano real. Apesar de utilizados dados com informações sobre o mundo real, há uma diferença entre as características desses dados

e vários outros conjuntos de dados do mundo real, como dados faltantes e com altos ruídos. Isto também afeta a generalização dos resultados, constituindo uma ameaça à validade externa.

4.4 Avaliação de Restrições de Soluções Incrementais

Para avaliar o impacto das restrições das soluções na qualidade da rede, um experimento para cada algoritmo incremental com um design fatorial fracionado 2^{6-2} é realizado. Este design tem resolução IV, o que permite identificar os efeitos principais confundidos apenas com efeitos de interações entre 3 fatores. Estas interações geralmente tem efeitos baixos e a confusão com os efeitos principais não altera o resultado de forma significativa.

Cada experimento fracionado realizado tem apenas 16 combinações de fatores exclusivas, então só é possível usá-lo para estimar 16 parâmetros no modelo de regressão linear ao realizar a análise de variância nos dados. A estrutura de confusão para cada experimento desta seção é descrita no Apêndice B.

Não é possível realizar diretamente a análise de variância dos dados dos experimentos porque não há ensaios que considerem todos os efeitos. Portanto, não há como estimar o termo de erro no modelo de regressão, fazendo com que a soma dos quadrados dos erros, componente importante para o cálculo da anova, seja igual a 0.

No entanto, para situações como essa, o método de Lenth, juntamente com o gráfico de probabilidade normal dos efeitos, pode ser usado para identificar efeitos significativos. O método de Lenth assume que todos os efeitos devem ser normalmente distribuídos com uma média de 0, dada a hipótese de que eles não são significativos. Se quaisquer efeitos forem significativamente diferentes de 0, eles devem ser considerados significativos para a variável resposta analisada.

Nos próximos tópicos, busca-se compreender a influência da restrição no número de pais, o nível de confiança dos testes estatísticos utilizados em ST e a quantidade de conjuntos com melhor pontuação sendo armazenados e utilizado por IHCS nas métricas de qualidade deste trabalho.

4.4.1 Pontuação Estrutural

Os gráficos de probabilidade normal dos efeitos na pontuação estrutural para os experimentos com IHCS e ST são apresentados na Figura 4.40 e 4.41, respectivamente.

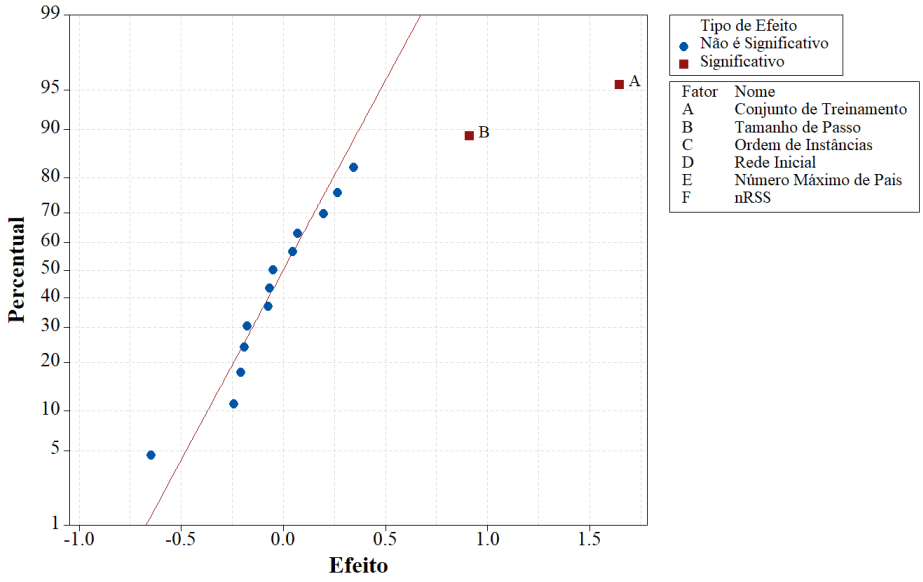

Figura 4.40: Gráficos de probabilidade normal dos efeitos na pontuação estrutural para ST

Na Figura 4.40, nota-se que, no experimento com IHCS, os efeitos significativos são apenas os efeitos principais do conjunto de dados e do tamanho do passo. Todos esses dois são positivos, ou seja, incrementam a variável resposta. Já na Figura 4.41, nota-se que, no experimento com ST, além dos efeitos principais do conjunto de dados e do tamanho do passo, o efeito da interação entre o conjunto de dados e o número de pais também é considerado significativo.

Uma comparação entre o impacto dos efeitos para os experimentos com IHCS e ST é exibida na Figura 4.42 e 4.43, respectivamente. Nota-se que, no experimento descrito na Figura 4.43, os efeitos principais são mais significativos do que o efeito da interação. Além disso, em ambos os experimentos, a efeito do conjunto de dados é maior que o efeito do tamanho do passo.

Mantendo-se agora somente no experimento com ST, o modo como a interação entre os efeitos dos conjuntos de treinamento e o número máximo de pais usado no ST influenciam

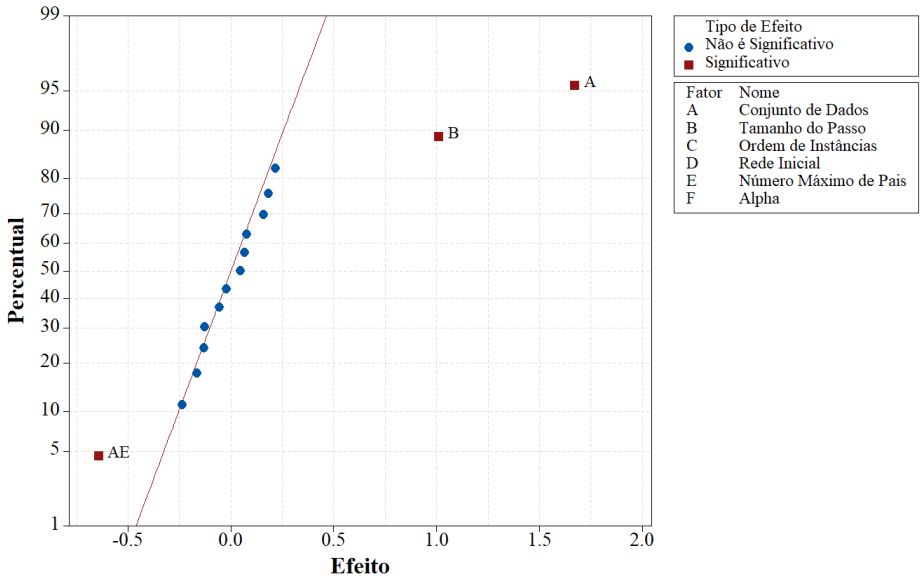

Figura 4.41: Gráficos de probabilidade normal dos efeitos na pontuação estrutural para IHCS

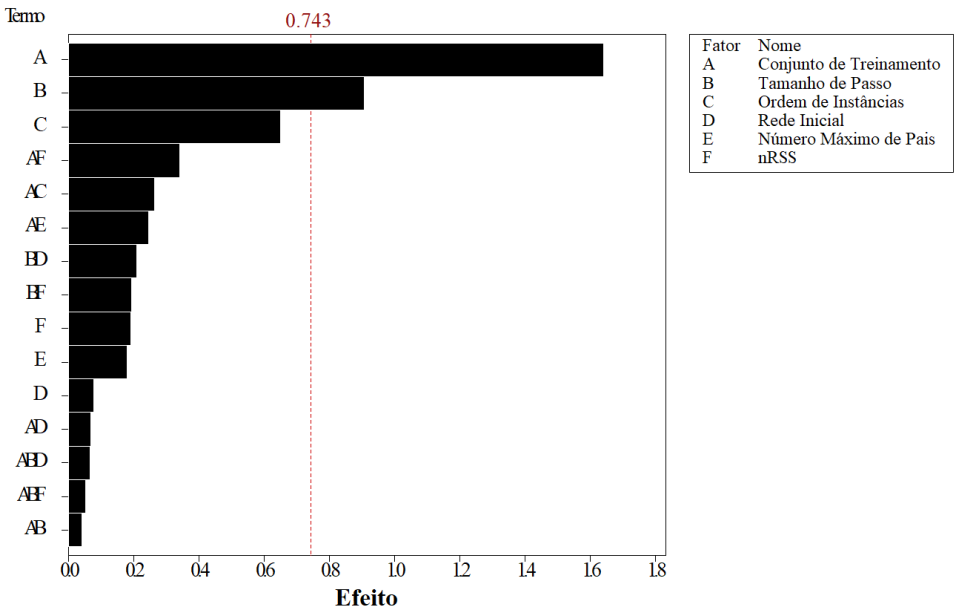

Figura 4.42: Gráfico de pareto dos efeitos padronizados na pontuação estrutural para IHCS

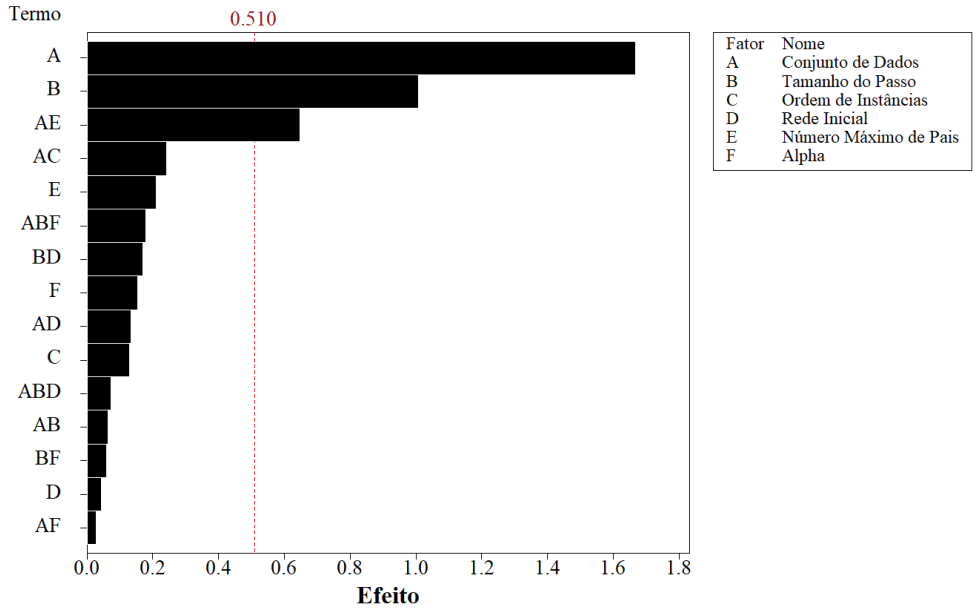

Figura 4.43: Gráfico de pareto dos efeitos padronizados na pontuação estrutural para IHCS

a variável resposta é apresentada na Figura 4.44. Os efeitos principais não são abordados porque não são o foco destes experimentos.

Nota-se na interação entre efeitos que usar o número máximo de pais igual a 1 nas duas bases de dados faz produzir redes melhores do que não realizar esta restrição. No caso da base de dados *Alarm*, essa diferença é bastante significativa.

A análise de variância também é utilizada para validar estatisticamente o modelo com os termos significativos encontrados e dar significância aos resultados. Todos os outros efeitos insignificantes fora tratados como resíduos. É uma prática comum agrupar efeitos não significativos em resíduos [14]. A análise sobre os resíduos do modelo é apresentada na Figura 4.45.

Nota-se na Figura 4.45 que, pelo histograma e o gráfico de probabilidade normal, os resíduos parecem seguir uma distribuição normal. Para validar esta suposição, o teste de Ryan-Joiner, similar ao Shapiro-Wilk é utilizado. Adotando um nível de significância de 5%, o valor de p maior que 1 é obtido, significando a ausência de evidências que indicam que a distribuição não é normal. Assim, pode-se adotar a distribuição dos resíduos como uma distribuição normal.

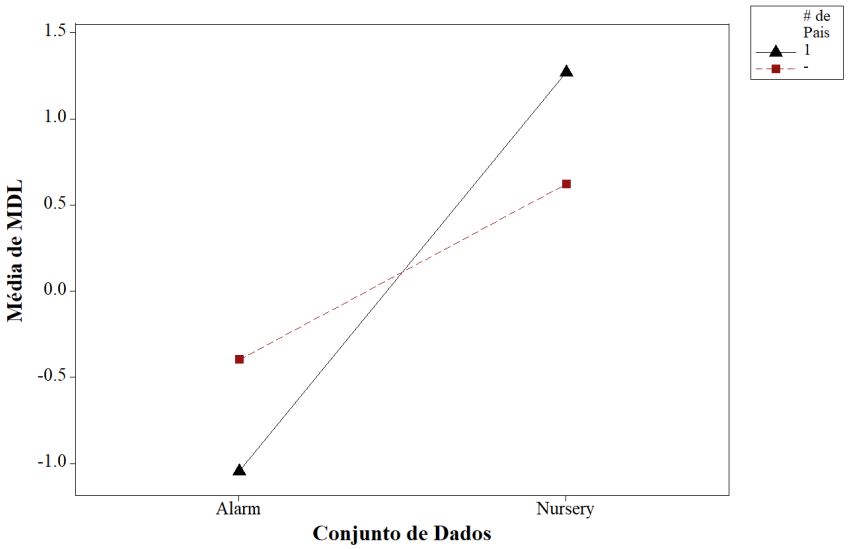

Figura 4.44: Gráfico de efeitos de interações na pontuação estrutural para experimento com ST

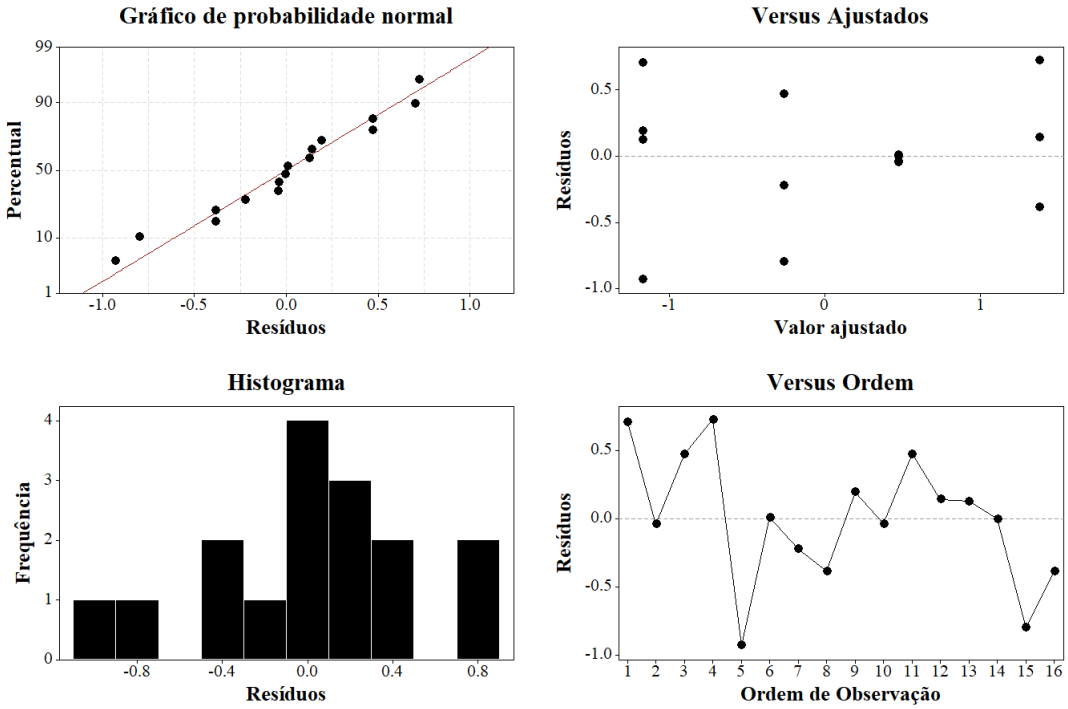

Figura 4.45: Gráficos de resíduos da pontuação estrutural para ST

É possível verificar a variação constante dos resíduos no gráfico em que são comparados com os valores ajustados da variável resposta. Neste gráfico, pode-se considerar válida a homocedasticidade dos resíduos dada a ausência de padrões. Pode-se também verificar a independência dos resíduos entre si no gráfico onde estes são exibidos dado sua ordem de coleta. Nota-se a ausência de padrões, como círculos, e a suposição de independência dos resíduos é considerada válida.

Com o modelo validado, é possível adotar o valor de p como estatística suficiente para validar a significância dos efeitos analisados. Adotando um nível de significância de 5%, todos os três valores de p foram bem próximos a 0. Dado estes valores de p, supõem-se que, com 5% de significância, o número de pais tem efeito significativo na pontuação DCM das redes aprendidas pelo algoritmo ST. Já para o valor de α, referente ao nível de confiança dos testes estatísticos utilizados em ST e a quantidade de conjuntos com melhor pontuação, *nRSS*, sendo armazenados e utilizados por IHCS, não é possível concluir sobre sua significância estatística na pontuação DCM das redes aprendidas, pois o valor de alguma de suas interações com outros fatores é confundido com alguns efeitos considerados significantes.

4.4.2 Diferença Estrutural

A seguir, são apresentados os resultados experimentais sobre as restrições abordadas nas métricas predição, cobertura e valor F da estrutura gerada pelos algoritmos incrementais sobre as estruturas geradas pelos algoritmos em lote.

Precisão

Os gráficos de pareto dos efeitos na precisão estrutural nos experimentos com ST e IHCS são apresentados nas Figuras 4.46 e 4.47, respectivamente.

Nota-se que no experimento para ST, há dois efeitos significativos de fatores, sem interação, e um efeito da interação entre dois fatores. Os efeitos principais destacados são do conjunto de dados e do número máximo de pais que o algoritmo de busca e pontuação pode utilizar. O efeito de interação engloba os mesmo dois fatores. Já no experimento para o IHCS, apenas a interação entre os dois fatores é adotada como significativa. Nota-se também que os gráficos estão diferentes, onde existem barras cinzas na Figura 4.47. Estes são fatores

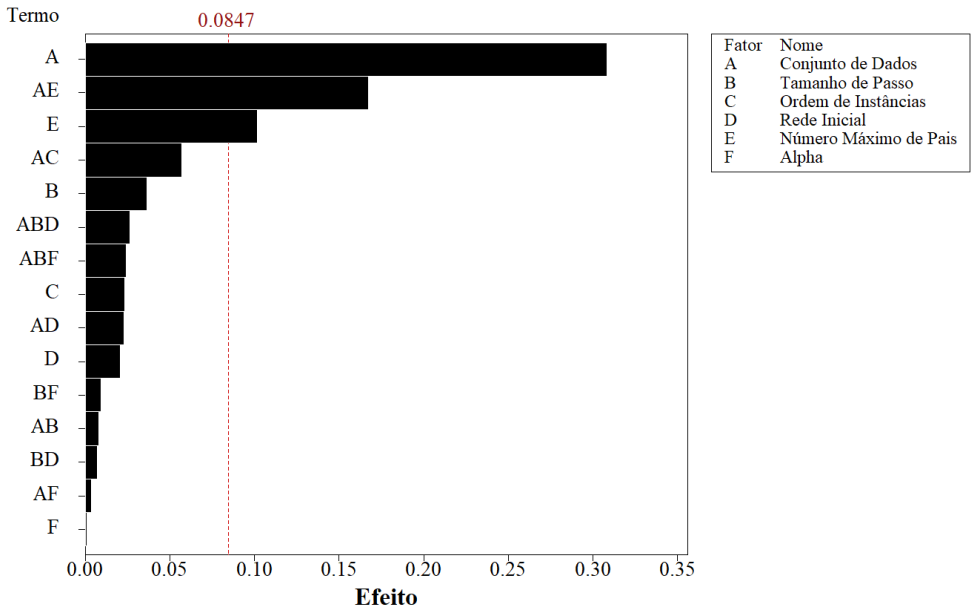

Figura 4.46: Gráfico de pareto dos efeitos padronizados na precisão para ST

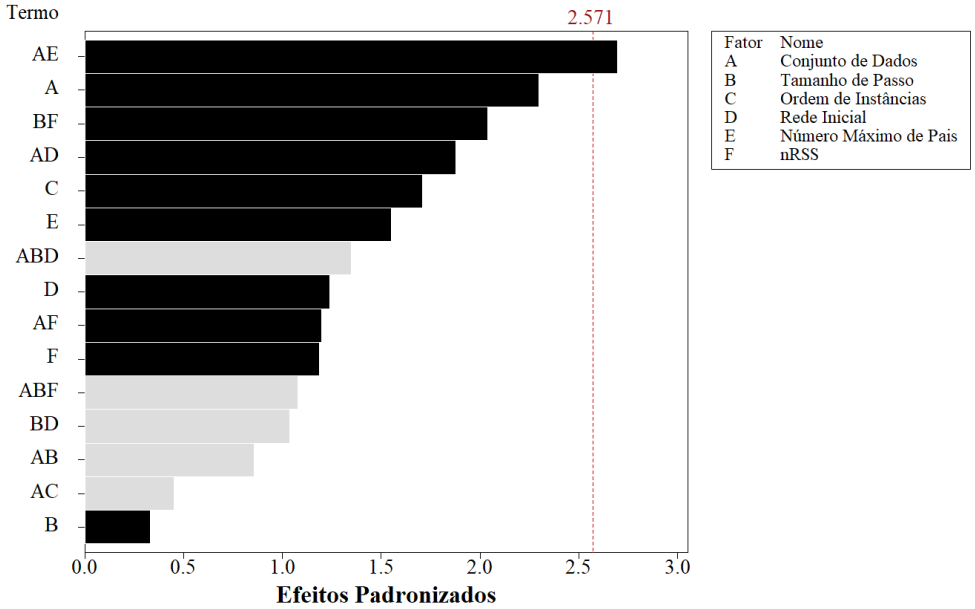

Figura 4.47: Gráfico de pareto dos efeitos padronizados na precisão para IHCS

removidos da análise sobre os efeitos porque, inicialmente, o método de Lenth não identificou nenhum efeito significativo. Nestes casos, o indicado é a remoção, gradativa, dos efeitos de menor significância, até que existam efeitos considerados significativos no modelo.

Os gráficos de probabilidade normal dos efeitos na precisão estrutural nos experimentos com ST e IHCS são apresentados nas Figuras 4.48 e 4.49, respectivamente.

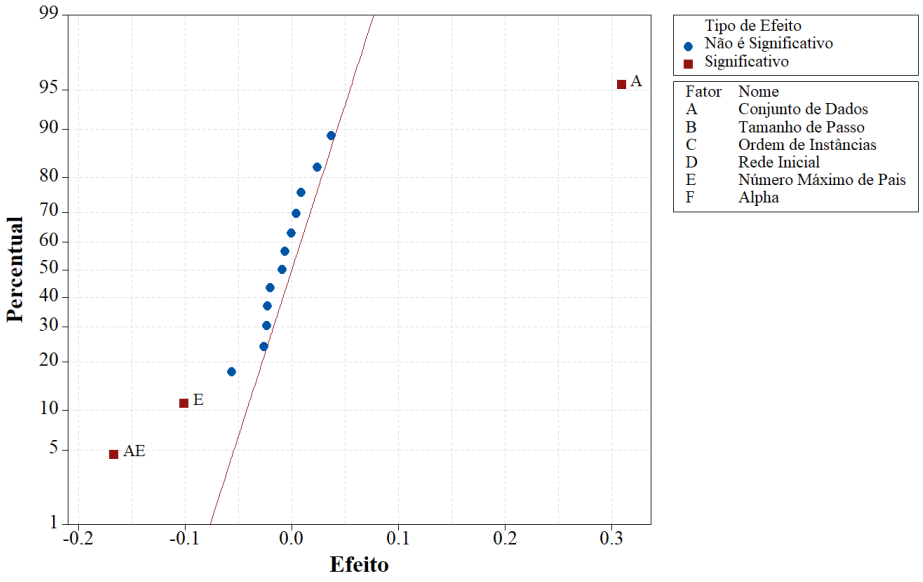

Figura 4.48: Gráficos de probabilidade normal dos efeitos na precisão para ST

Nota-se que, nos dois experimentos, o efeito da interação entre os fatores é negativo, ou seja, quando se altera o nível do fator do *low* para o *high*, a precisão é afetada negativamente. Percebe-se que, no experimento usando ST, o efeito do conjunto de dados afeta positivamente a precisão, já a interação entre os fatores e o fator número de pais afeta negativamente. Estes efeitos são descritos nas Figuras 4.50, 4.51 e 4.52.

Nota-se que, na Figura 4.50, que descreve os efeitos no experimento com ST, ao manter a restrição dos pais em 1, a predição média é de, aproximadamente, 70%. Quando remove-se esta restrição, a precisão média cai para, aproximadamente, 60%. Agora nota-se que na Figura 4.51, que descreve os efeitos das interações para o mesmo experimento, que ao manter as restrições, o conjunto de dados *Alarm* possui um valor médio de precisão de, aproximadamente, 45%, e o conjunto *Nursery* de, aproximadamente, 63%. Quando restringe-se para 1 pai, a precisão para *Alarm* cai pouco menos que, aproximadamente, 10%, mas para *Nursery*, ela sobe quase 40%. Este comportamento se mantém na Figura 4.52, que descreve os

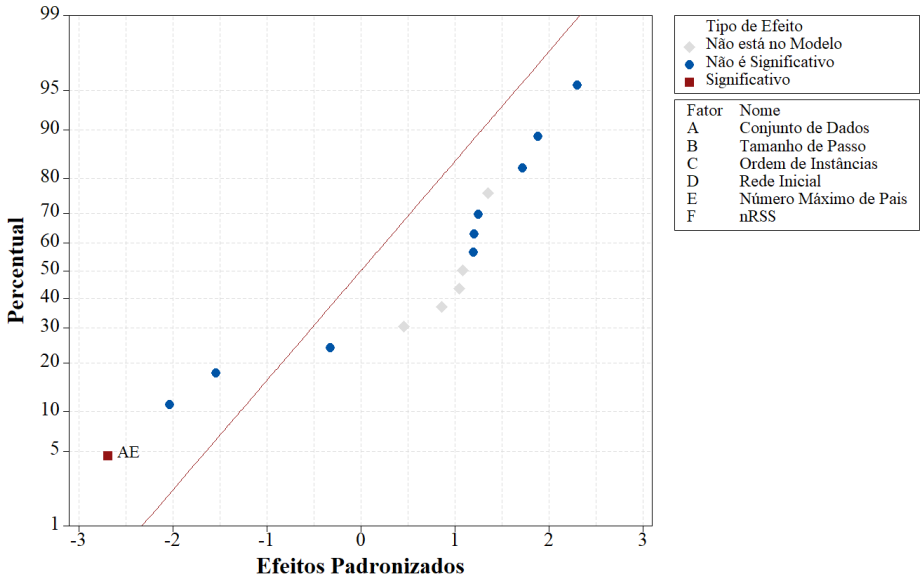

Figura 4.49: Gráficos de probabilidade normal dos efeitos na precisão para IHCS

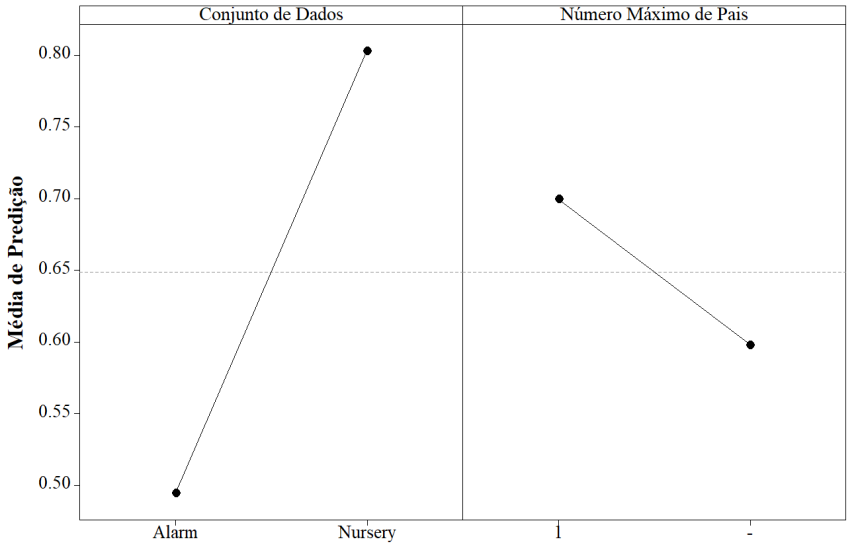

Figura 4.50: Gráfico de efeitos significantes de fatores na precisão para ST

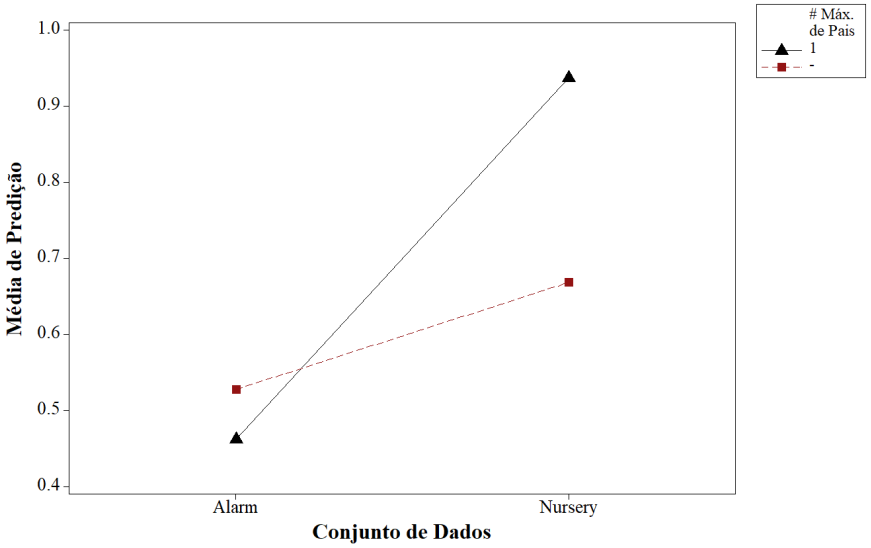

Figura 4.51: Gráfico de efeitos significantes de interações na precisão para ST

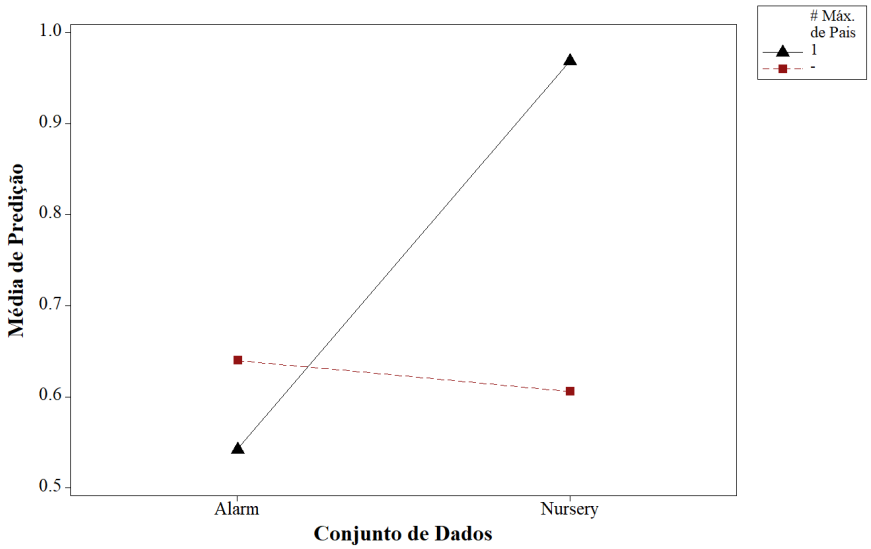

Figura 4.52: Gráfico de efeitos significantes de interações na precisão para IHCS

resultados para o uso do IHCS.

Percebe-se então uma diferença notável entre os resultados dado os níveis dos fatores citados como influentes. Mas, para existir significância estatística, a variância dos efeitos deve ser analisada. Os gráficos para análise dos resíduos para o experimento usando ST e IHCS são apresentados nas Figuras 4.53 e 4.54, respectivamente.

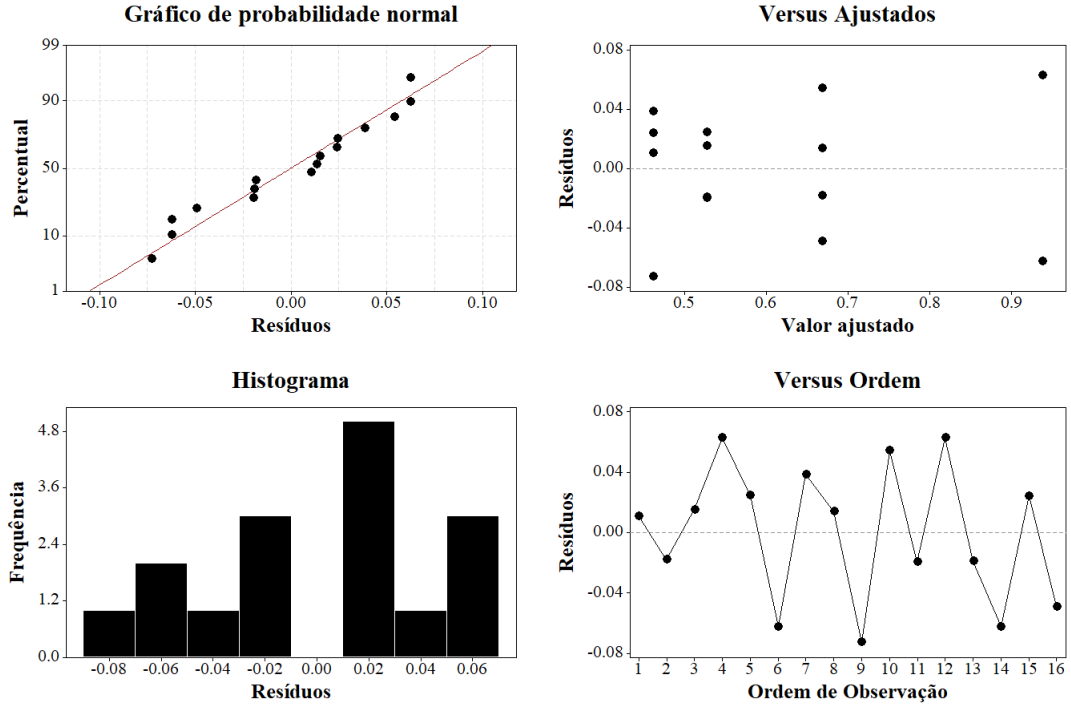

Figura 4.53: Gráficos de resíduos do modelo com efeitos significativos na precisão para ST

Nota-se que, dado os histogramas e os gráficos de probabilidade normal, ambos os resíduos parecem seguir uma distribuição normal. Para validar esta suposição, o teste de normalidade de Ryan-Joiner é aplicado. Adotando um nível de significância de 5%, um valor de p maior que 1 é obtido, significando a ausência de evidências que indicam que a distribuição não é normal.

Pode-se também validar as suposições de independência dos erros e de homocedasticidade dos resíduos analisando os gráficos restantes. Não há padrões entre os resíduos, o que valida as suposições citadas. Com os modelos validados, pode-se adotar o valor de p como estatística suficiente para validar a significância dos efeitos analisados. Adotando um nível de significância de 5%, todos os três valores de p foram bem próximos a 0, então supõem-se que o número de pais tem efeito significativo na precisão das redes aprendidas pelo algo-

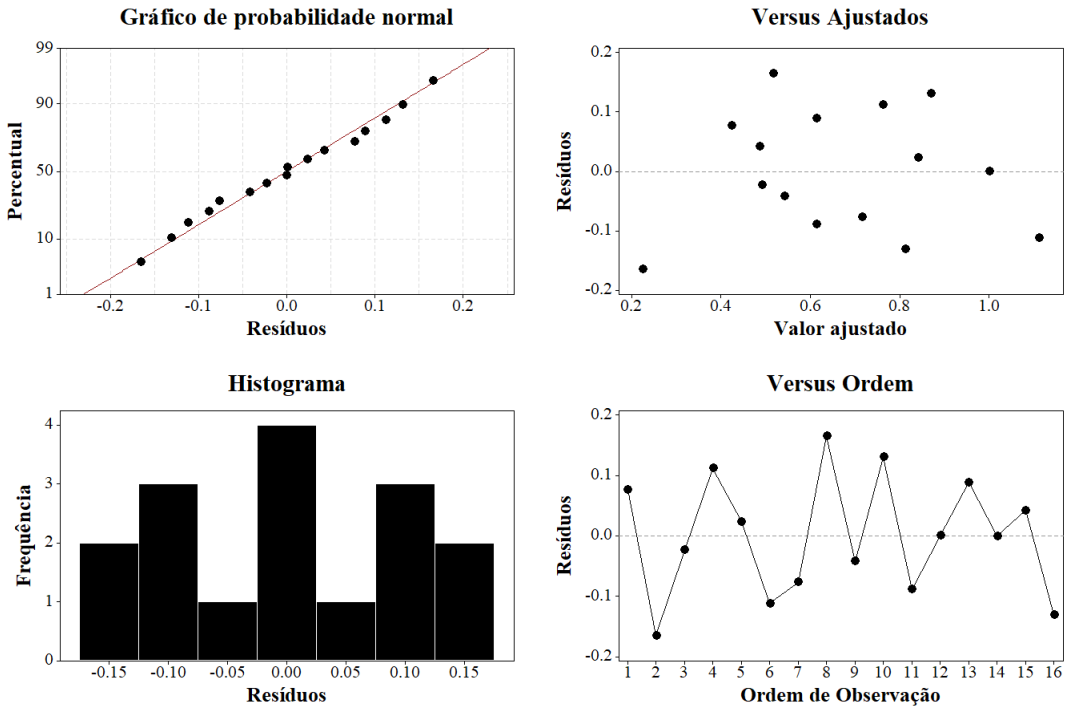

Figura 4.54: Gráficos de resíduos do modelo com efeitos significativos na precisão para IHCS

ritmo ST. Esse comportamento também é mantido para o algoritmo IHCS. Já para o valor de α para o ST e o valor de $nRSS$ para o IHCS, não é possível concluir sobre sua significância estatística, pois, apesar de serem considerados insignificantes, o valor de alguma de suas interações com outros fatores é confundido com alguns efeitos considerados significantes.

Cobertura

Os gráficos de pareto dos efeitos na cobertura estrutural nos experimentos com ST e IHCS são apresentados nas Figuras 4.55 e 4.56, respectivamente.

Nota-se que no experimento para ST, há somente dois efeitos significativos de fatores, sem interação. Os efeitos principais destacados são do conjunto de dados e do número máximo de pais. Já no experimento para o IHCS, há vários efeitos, tanto de fatores, como de interações entre fatores, dentre eles, o número máximo de pais e o $nRSS$. Os gráficos na Figura 4.56 seguem o mesmo padrão de cores da 4.47. Como já citado, os efeitos em cinza foram removidos da análise. Os gráficos de probabilidade normal dos efeitos na cobertura estrutural nos experimentos com ST e IHCS são apresentados nas Figuras 4.57 e 4.58,

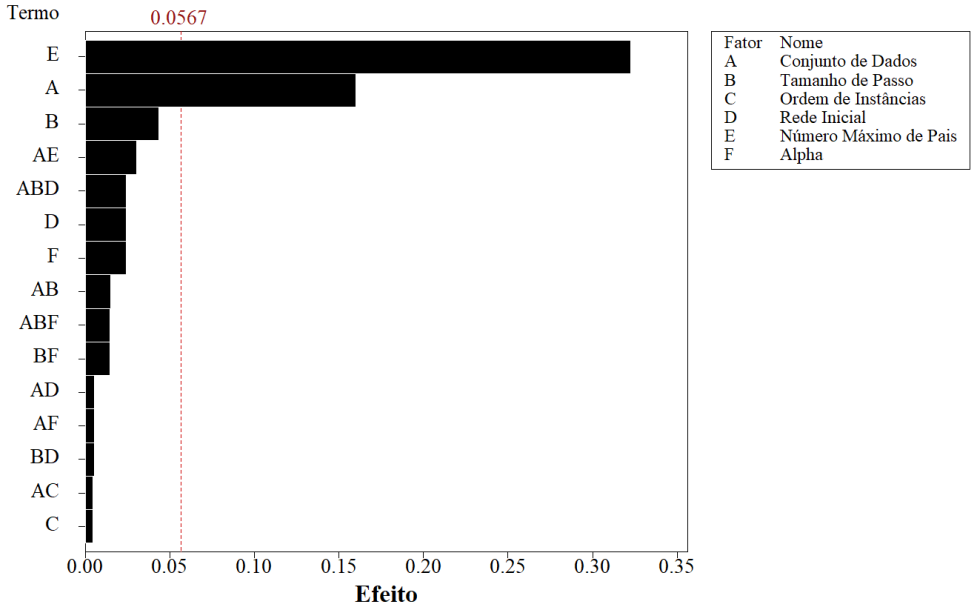

Figura 4.55: Gráfico de pareto dos efeitos padronizados na cobertura para ST

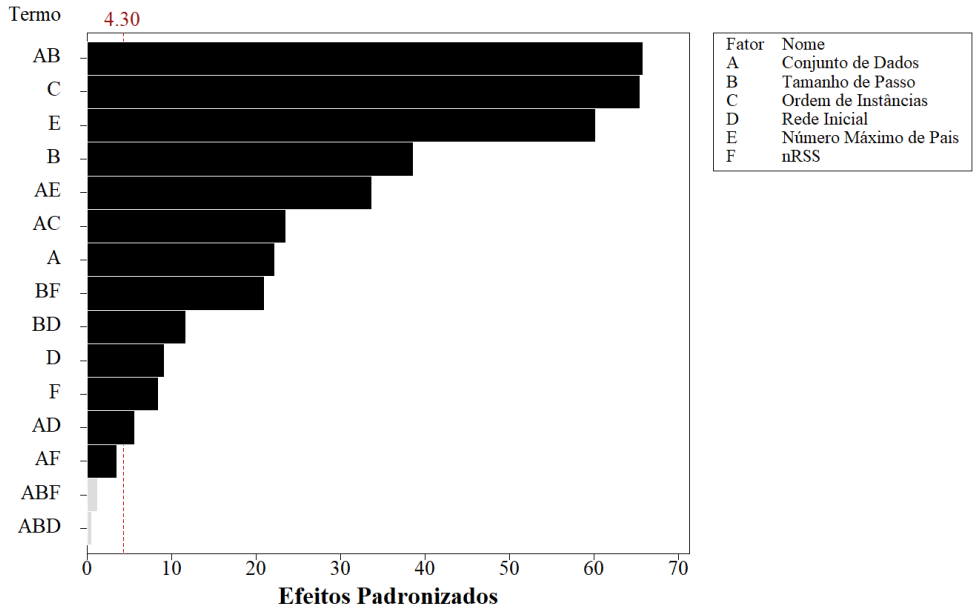

Figura 4.56: Gráfico de pareto dos efeitos padronizados na cobertura para IHCS

respectivamente.

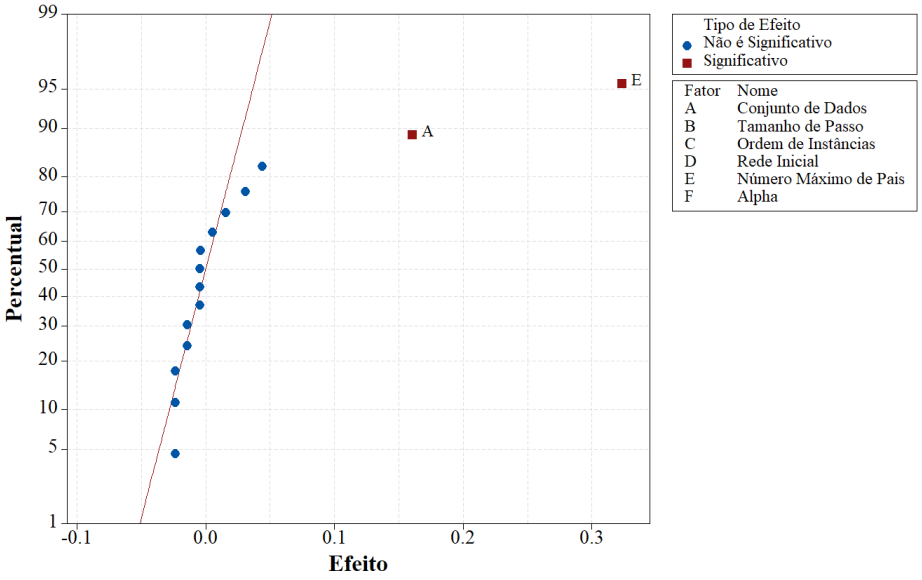

Figura 4.57: Gráficos de probabilidade normal dos efeitos na cobertura para ST

Na Figura 4.57, nota-se que os efeitos significantes causam um efeito positivo na cobertura. Ou seja, alterar de seu estado *low* para *high* melhora a média de cobertura estrutural. Na Figura 4.57, nota-se que os efeitos significantes principais do número máximo de pais permitidos e de $nRSS$ causam um efeito positivo na cobertura. No entanto, os efeitos de suas interações com outros fatores causam efeitos negativos. Para uma melhor compreensão, os gráficos nas Figuras 4.59, 4.60 e 4.61 são analisados a seguir.

Na Figura 4.59, pode-se notar os efeitos positivos dos fatores. Enquanto *Alarm* e restrição de pais está ativa, os valores são abaixo dos valores médios de cobertura das redes. Quando o conjunto de dados é invertido e não há mais restrições, a cobertura mantém-se melhor que a média dos valores de cobertura do experimento. Na Figura 4.60, nota-se que o efeito para o número de pais e, apesar de inferiores a outros efeitos, o efeito para $nRSS$ tem efeitos abaixo e acima da média quando alternado seus níveis. Ambos ultrapassam a linha média de cobertura do experimento no seu nível *high*.

Já na Figura 4.61, percebe-se, ainda sim, pontos que não explicam a inversão de efeitos quando há interação. Quando interagindo com o tamanho do passo, o $nRSS$ mais cresce a taxa de cobertura das redes quando o é 100 do que quando 2000 ou 4000, dependendo da base de dados. Isto também se repete entre o conjunto de dados e o número máximo de pais.

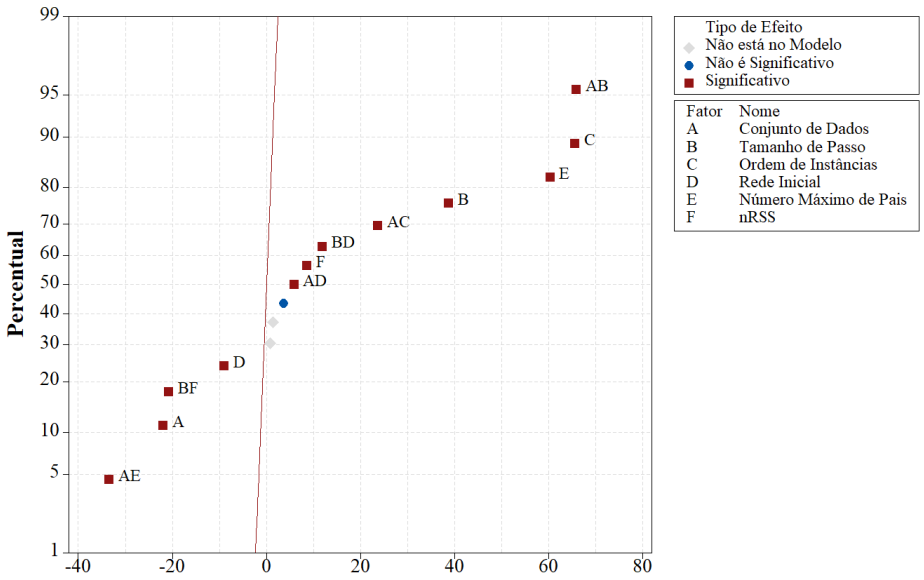

Figura 4.58: Gráficos de probabilidade normal dos efeitos na cobertura para IHCS

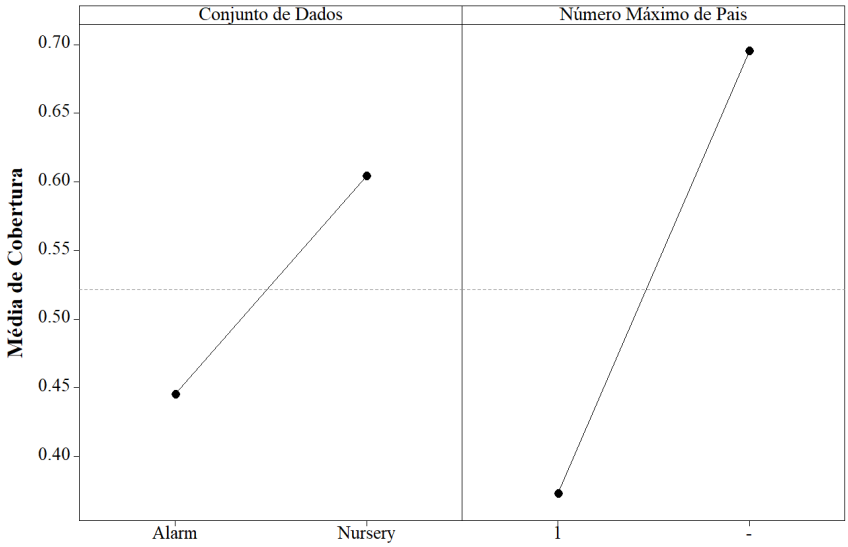

Figura 4.59: Gráfico de efeitos significantes de fatores na cobertura para ST

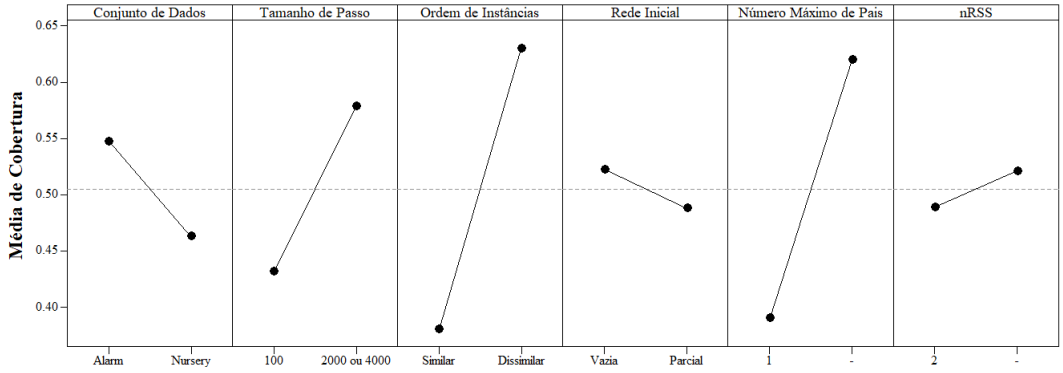

Figura 4.60: Gráfico de efeitos significantes de fatores na cobertura para IHCS

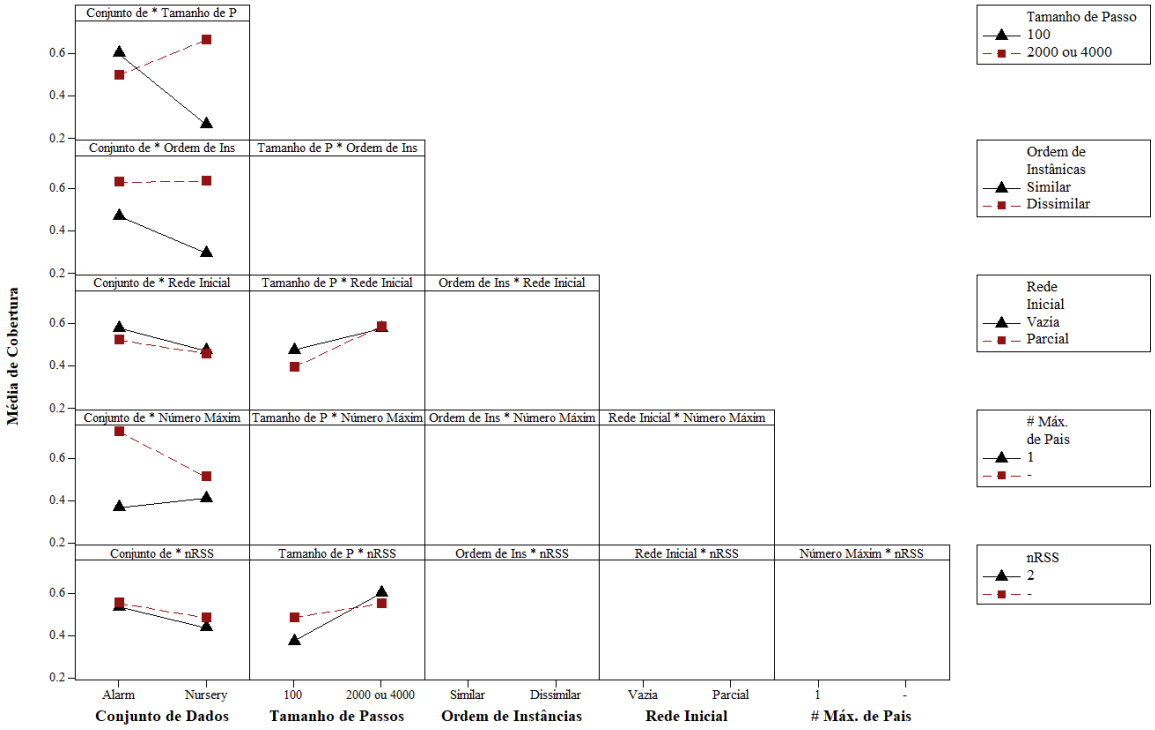

Figura 4.61: Gráfico de efeitos significantes de interações na cobertura para IHCS

Neste caso, existe a necessidade de um experimento mais completo para que estes efeitos sejam melhor entendidos já que podem estar confundidos com todos os outros explicados na estrutura de confusão deste experimento (veja no Apêndice A). Nas Figuras 4.62 e 4.63, os gráficos para análise dos resíduos do modelo gerado são apresentados.

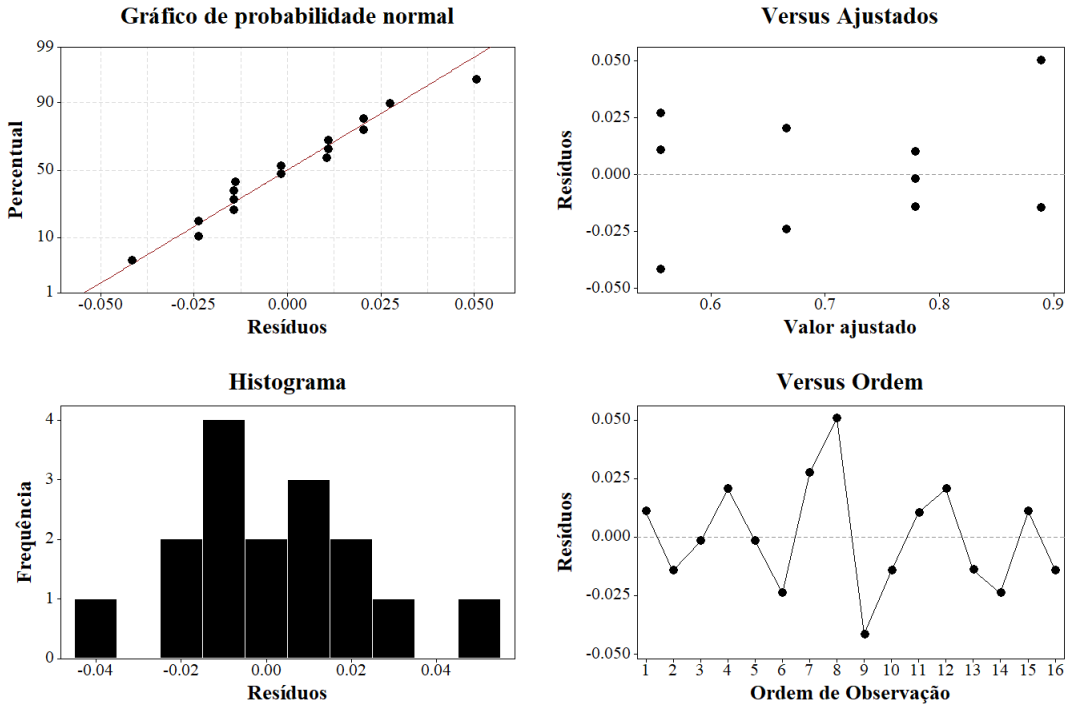

Figura 4.62: Gráficos de resíduos do modelo com efeitos significativos na cobertura para ST

Nota-se que, dado os histogramas e os gráfico de probabilidade normal da Figura 4.62, os resíduos parecem seguir uma distribuição normal, possuem homocedasticidade e independência. Para validar esta suposição, o teste de normalidade de Ryan-Joiner é aplicado. Adotando um nível de significância de 5%, um valor de p maior que 1 é obtido, significando a ausência de evidências que indicam que a distribuição não é normal.

Já na Figura 4.62, os resíduos seguem uma distribuição, aparentemente, um pouco diferente de uma distribuição normal. No entanto, usando o teste de normalidade de Ryan-Joiner e dotando um nível de significância de 5%, um valor de p maior que 1 é obtido, significando a ausência de evidências que indicam que a distribuição não é normal. Neste caso, o teste de Kolmogorov-Smirnov também é aplicado e um valor de $p > 0{,}150$ é obtido, também atestando a normalidade. Dado os gráficos dos resíduos e das variáveis ajustadas, e da ordem de obtenção dos resíduos, percebe-se que não há heterocedasticidade e interação entre os

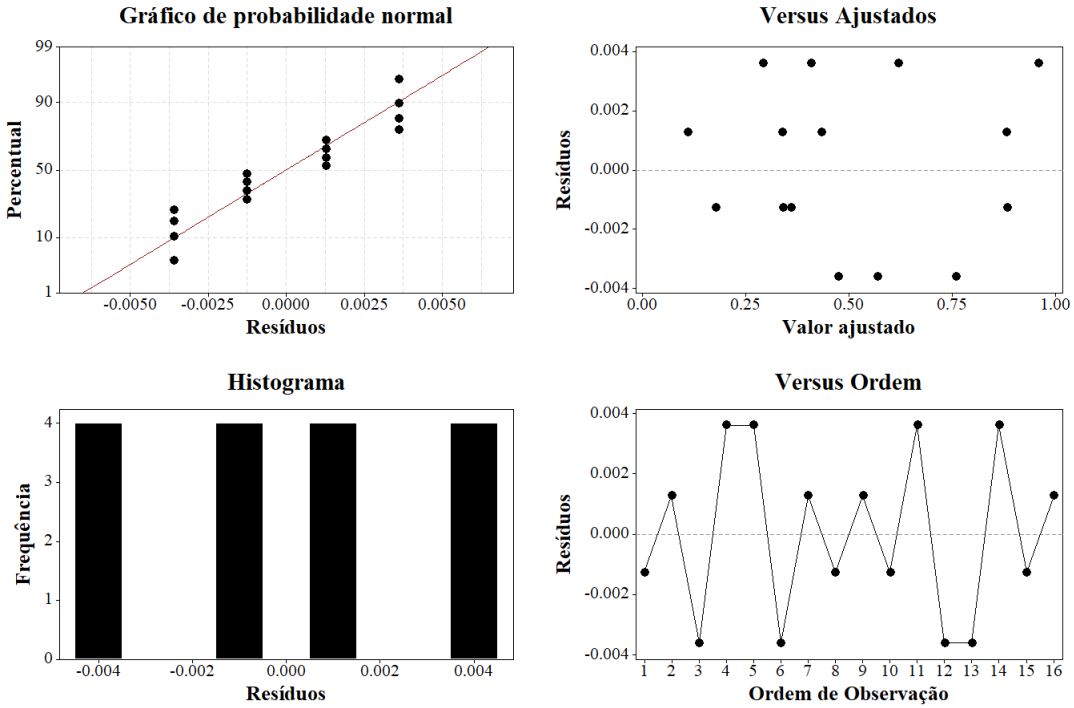

Figura 4.63: Gráficos de resíduos do modelo com efeitos significativos na cobertura para IHCS

resíduos.

Com o modelo válido, supõem-se então que, com significância de 5%, o número máximo de pais possui influência na qualidade de cobertura das redes aprendidas pelo ST. Já para α, não é possível supor que há ou não relação entre fator e cobertura. Considerando agora o experimento utilizando o IHCS, supõem-se, com 5% de significância que há efeito do número de pais e do $nRSS$ na métrica de cobertura das redes aprendidas.

Valor F

os gráficos de pareto dos efeitos no valor F nos experimentos com ST e IHCS são apresentados nas Figuras 4.64 e 4.65, respectivamente.

Nota-se que no experimento para ST, há somente dois efeitos significativos de fatores, sem interação. Os efeitos principais destacados são do conjunto de dados e do número máximo de pais. Já no experimento para o IHCS, há três efeitos, tanto de fatores, como de interações entre fatores, dentre eles, o número máximo de pais. Nota-se que o gráfico na Figura 4.65 segue o mesmo padrão de cores da 4.47. Os gráficos de probabilidade normal

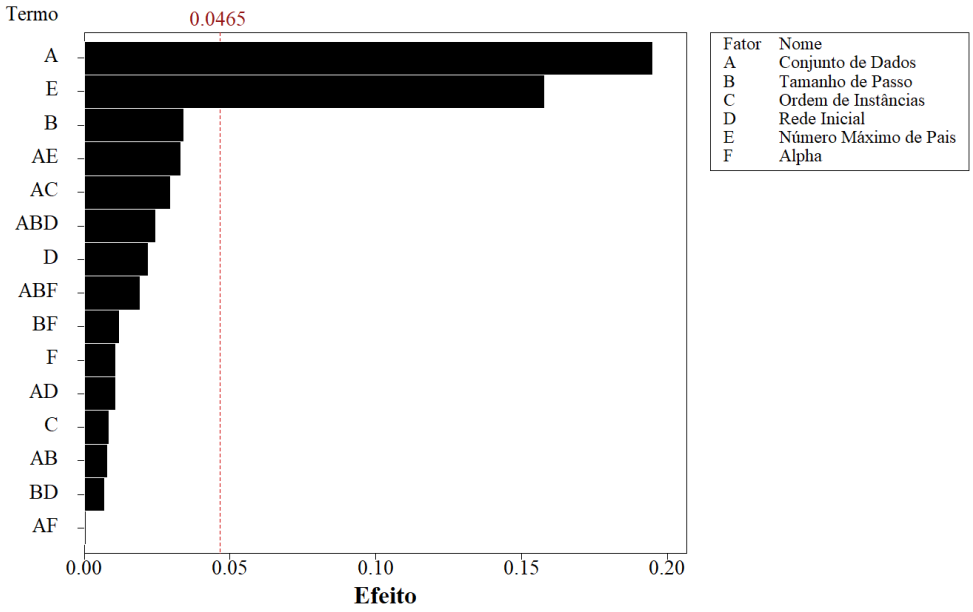

Figura 4.64: Gráfico de pareto dos efeitos padronizados no valor F para ST

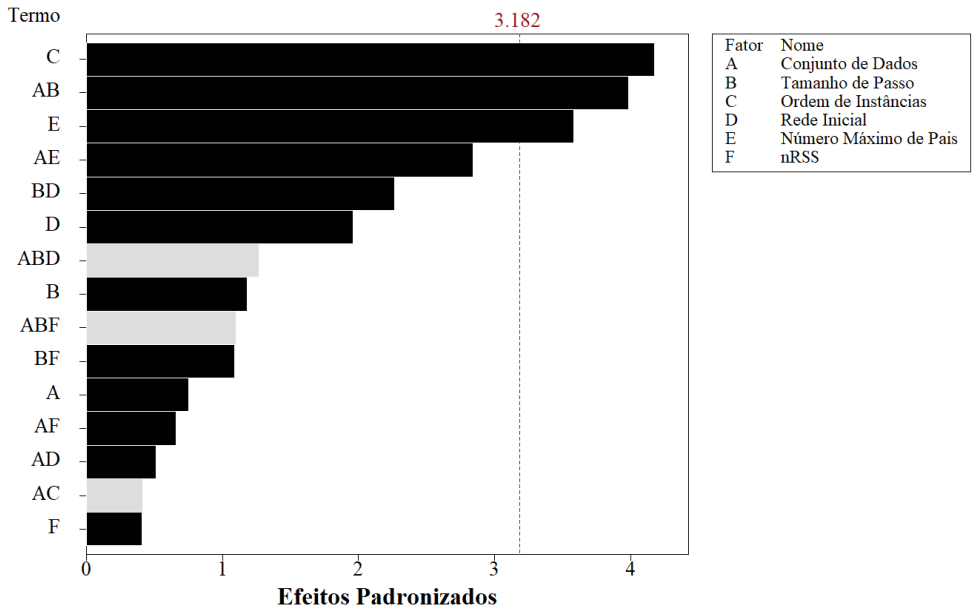

Figura 4.65: Gráfico de pareto dos efeitos padronizados no valor F para IHCS

dos efeitos na cobertura estrutural nos experimentos com ST e IHCS são apresentados nas Figuras 4.66 e 4.67, respectivamente.

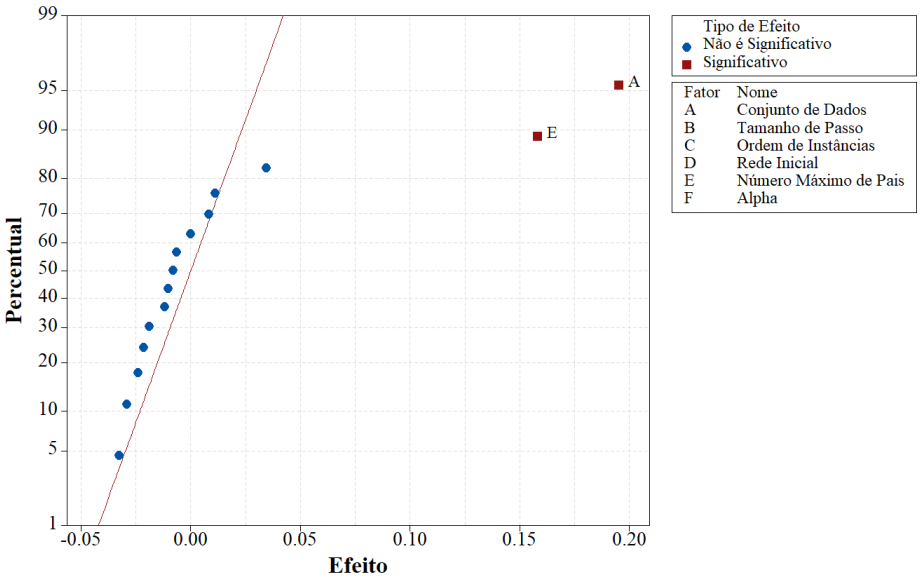

Figura 4.66: Gráficos de probabilidade normal dos efeitos no valor F para ST

Nas Figuras 4.66 e 4.67, nota-se que todos fatores significantes possuem efeitos positivos. Essa relação é melhor compreendida nas Figuras 4.68 e 4.69, que descrevem os efeitos dos níveis de cada fator significante nos experimentos com ST e IHCS, respectivamente.

Nota-se que o valor médio de F cresce de forma bastante similar entre os experimentos. Para o experimento com ST, descrito na Figura 4.68, o crescimento do valor F é maior para o conjunto de dados, mas ainda sim, o efeito do número de pais é significante. Mesmo comportamento se repete na Figura 4.69, mas com os fatores referentes às ordens de instâncias e o número máximo de pais. O efeito da interação não engloba nenhum dos fatores que abordados.

O modelo contendo os efeitos significativos é então avaliado para existir significância estatística nos resultados. Os gráficos dos resíduos são apresentados nas Figuras 4.70 e 4.71. Nota-se que ambos os resíduos seguem uma distribuição normal. Esta suposição é confirmada com o teste de normalidade de Ryan-Joiner. Os resíduos também não possuem padrão de variância, nem independência.

Com isso, supõem-se que, com 5% de significância, o número máximo de pais tem efeito significativo no valor de F sobre as estruturas geradas, tanto por ST, quanto por IHCS. Não

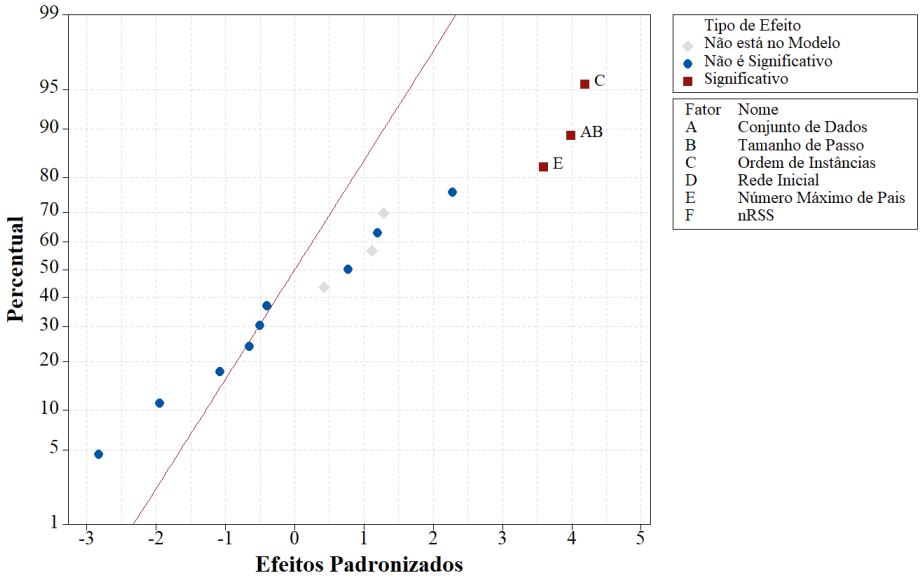

Figura 4.67: Gráficos de probabilidade normal dos efeitos no valor F para IHCS

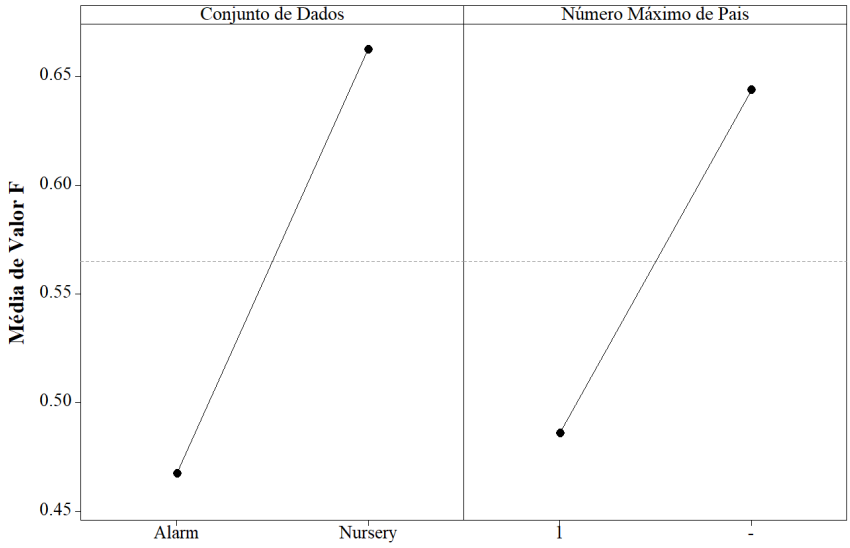

Figura 4.68: Gráfico de efeitos significantes de fatores no valor F para ST

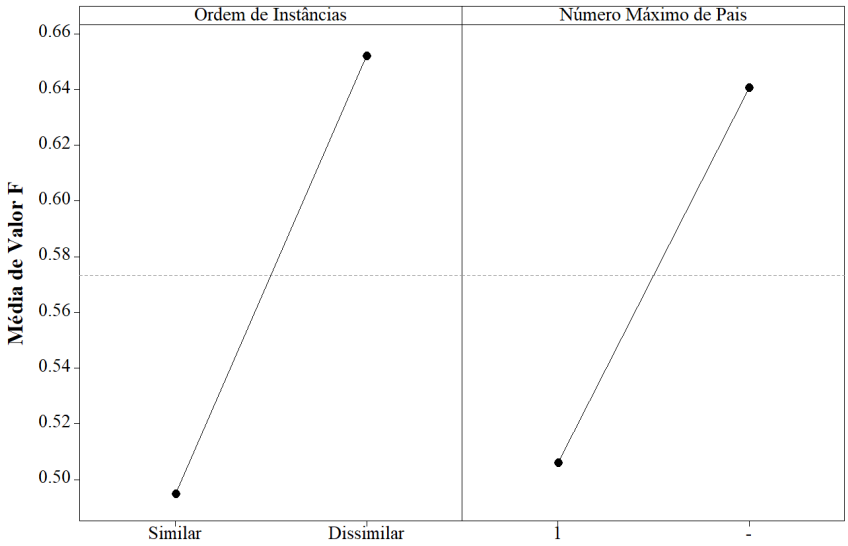

Figura 4.69: Gráfico de efeitos significantes de fatores no valor F para IHCS

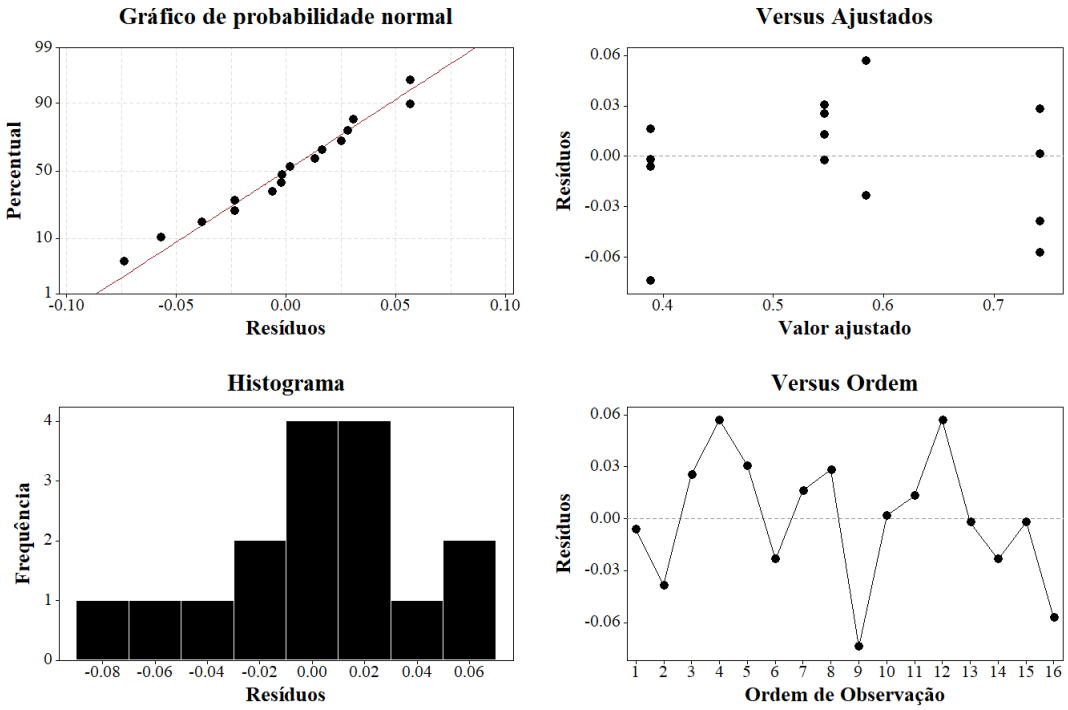

Figura 4.70: Gráficos de resíduos do modelo com efeitos significativos no valor F para ST

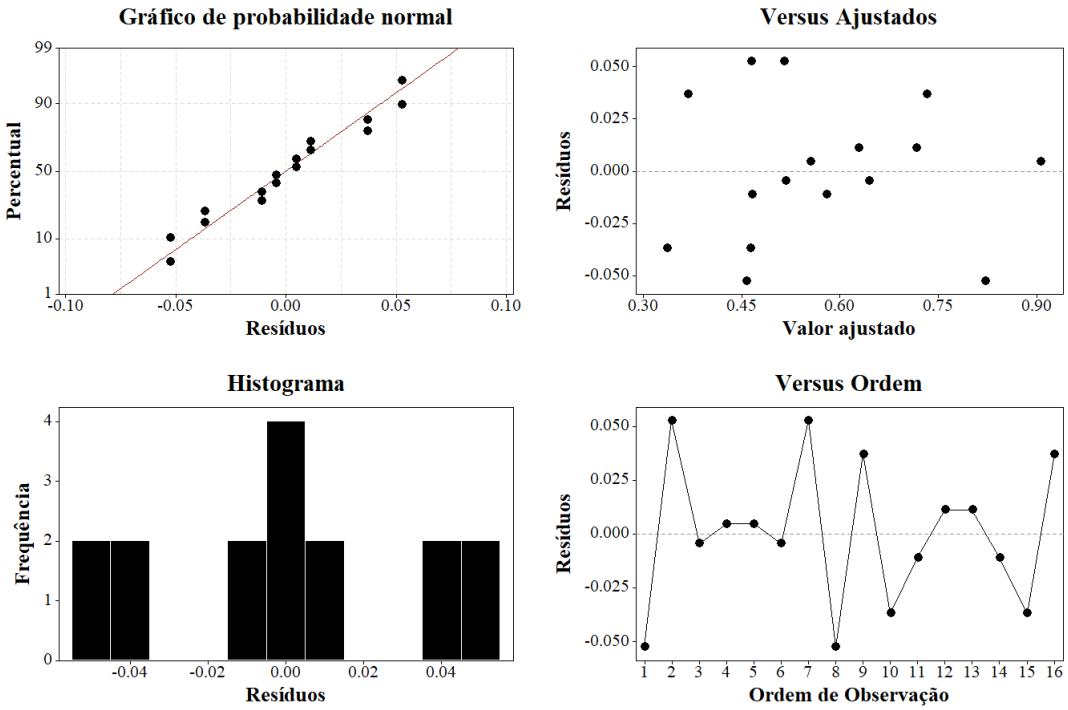

Figura 4.71: Gráficos de resíduos do modelo com efeitos significativos no valor F para IHCS

é possível supor o contrário sobre a significância sobre os efeitos principais de $nRSS$ para o IHCS e α para o ST, pois, apesar de serem pequenos, o valor de alguma de suas interações com outros fatores pode ser confundido com alguns efeitos considerados significantes.

4.4.3 Curva de Aprendizagem

os gráficos de pareto dos efeitos na perda logarítmica nos experimentos com ST e IHCS são apresentados nas Figuras 4.72 e 4.73, respectivamente.

Nota-se na Figura 4.72, referente aos experimentos com ST, que existem alguns efeitos principais significativos sobre a métrica analisada. Estes efeitos são dos fatores referentes ao conjunto de dados, ao tamanho do passo, à ordem das instâncias e ao número de pais. Há também efeitos significativos de interações entre o conjunto de dados, tamanho do passo e a ordem das instâncias. Já na Figura 4.73, referente aos experimentos com IHCS, nota-se que existem efeitos principais significativos sobre a métrica analisada. Estes efeitos são de fatores similares ao que acontece nos experimentos com ST e são referentes ao conjunto de dados, ao tamanho do passo, à ordem das instâncias e ao número de pais.

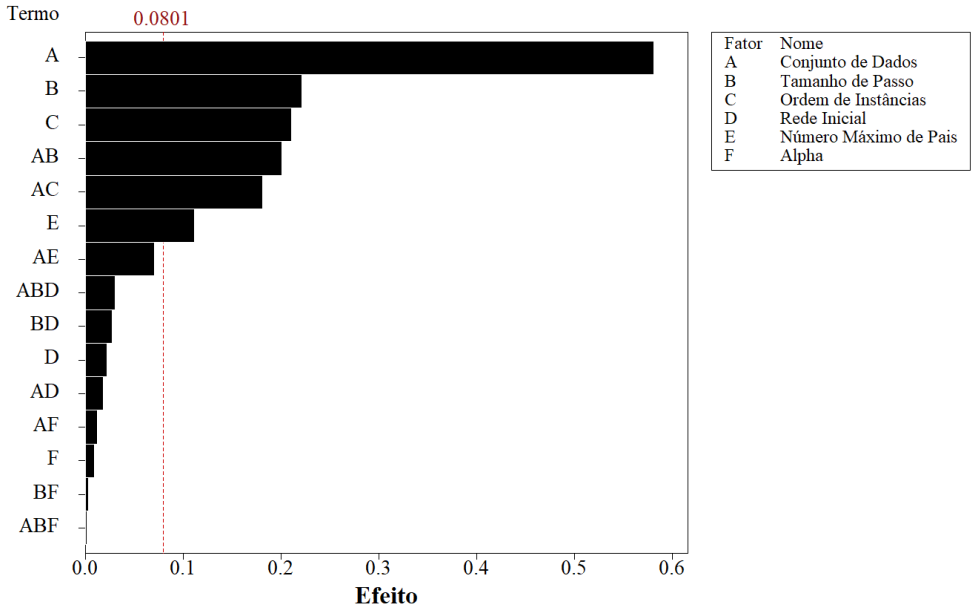

Figura 4.72: Gráfico de pareto dos efeitos padronizados na perda logarítmica para ST

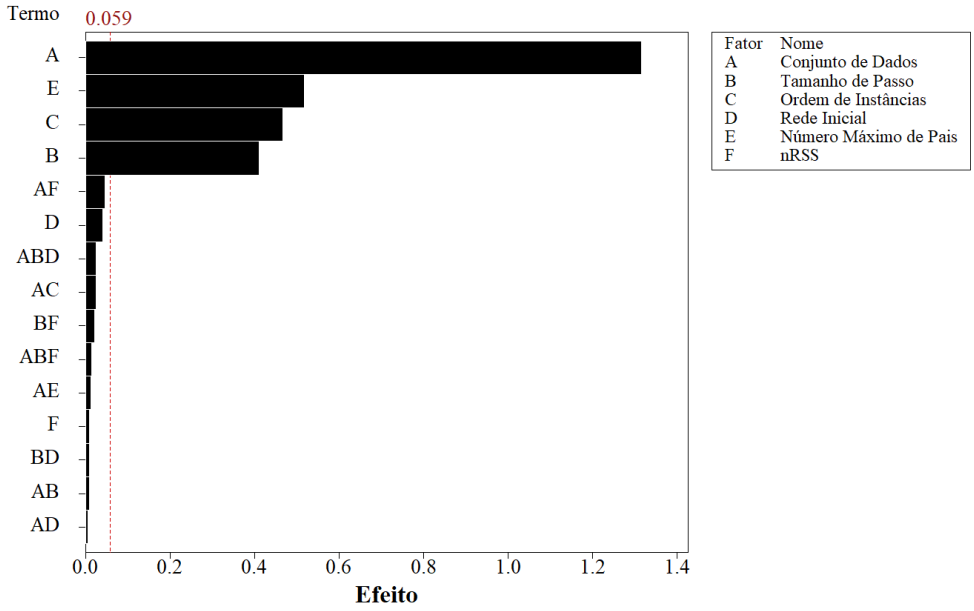

Figura 4.73: Gráfico de pareto dos efeitos padronizados na perda logarítmica para IHCS

Nós próximos gráficos, as direções dos efeitos significativos dentro dos experimentos com ST e IHCS são apresentadas nas Figuras 4.74 e 4.75, respectivamente.

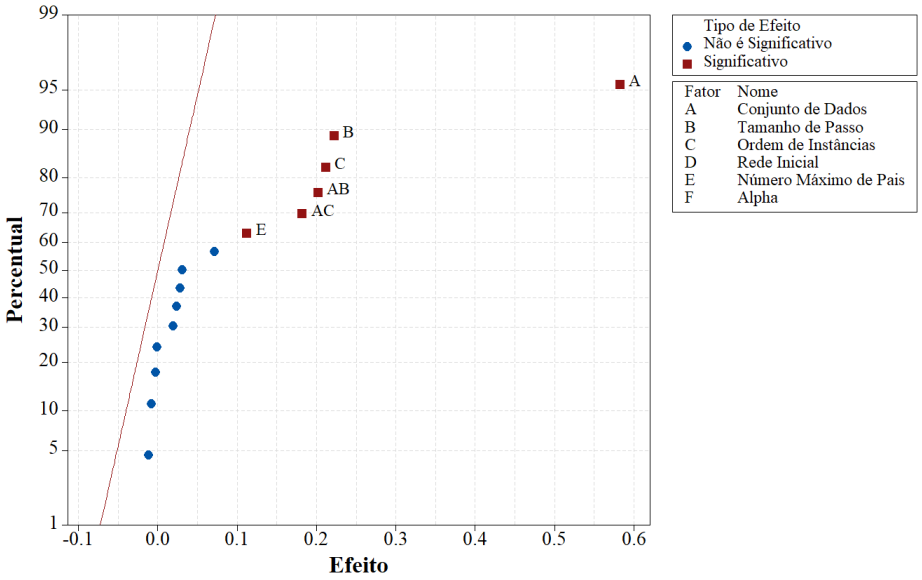

Figura 4.74: Gráficos de probabilidade normal dos efeitos na perda logarítmica para ST

Todos os efeitos analisados, independente do algoritmo experimentado, possuem efeito positivo no valor da perda. Até mesmo o número máximo de pais, foco de análise nestes experimentos. Nas Figuras 4.76 e 4.77, são apresentadas as influências de cada efeito principal nos experimentos com ST e IHCS, respectivamente. Os efeitos secundários não são abordados.

Percebe-se que, apesar de menor que os demais, a influência do número de pais ainda é considerável. Nota-se que ao restringir o número máximo de pais para 1, o valor médio da perda é de, aproximadamente, 0,15. Já ao remover esta restrição, o valor sobe para quase 0,3. Este comportamento também ocorre no IHCS e pode ser notado na Figura 4.77. O comportamento dos resíduos do modelo encontrado são apresentados nas Figuras 4.78 e 4.79.

Nota-se uma tendência de resíduos com distribuição anormal na Figura 4.78. No entanto, utilizando o teste de Ryan-Joiner e Kolmogorov-Smirnov, os valores de p de 0,28 e 0,53 são obtidos, respectivamente. Portanto, a hipótese de normalidade dos dados é aceita dada uma significância de 1%. A homocedasticidade e independência dos resíduos também são mantidas nos modelos.

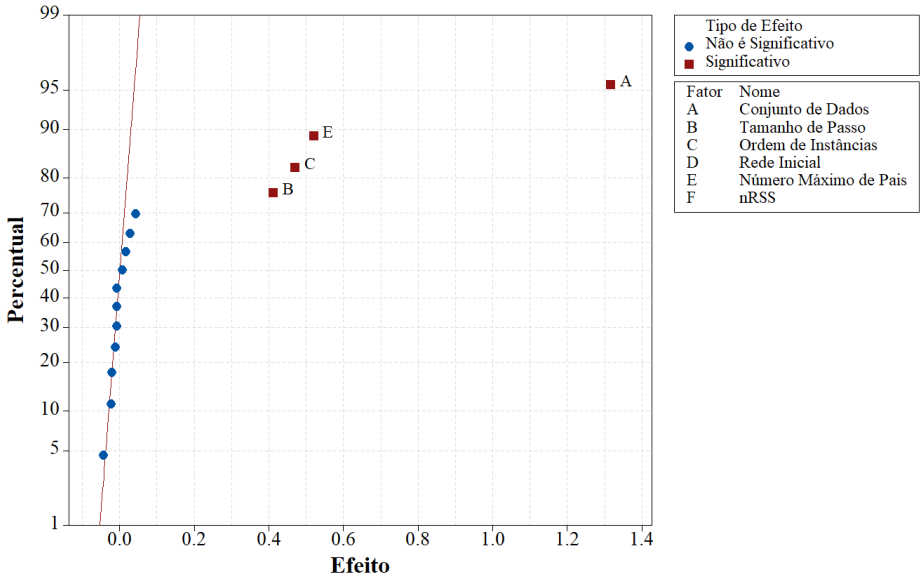

Figura 4.75: Gráficos de probabilidade normal dos efeitos na perda logarítmica para IHCS

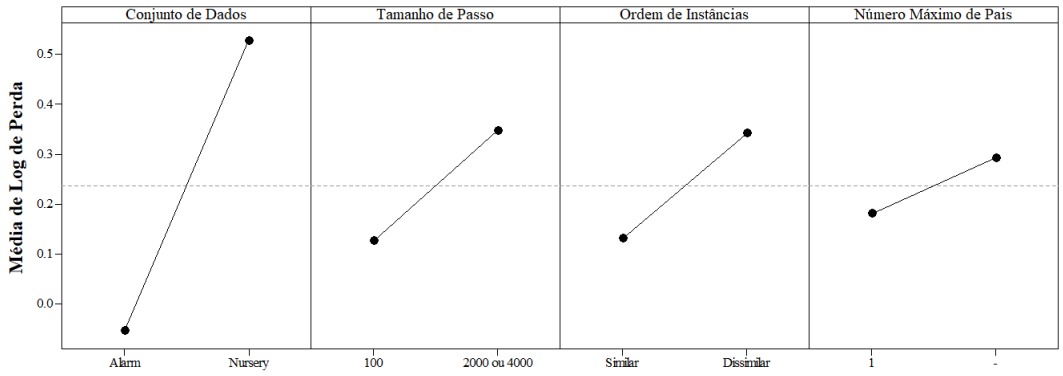

Figura 4.76: Gráfico de efeitos significantes de fatores na perda logarítmica para ST

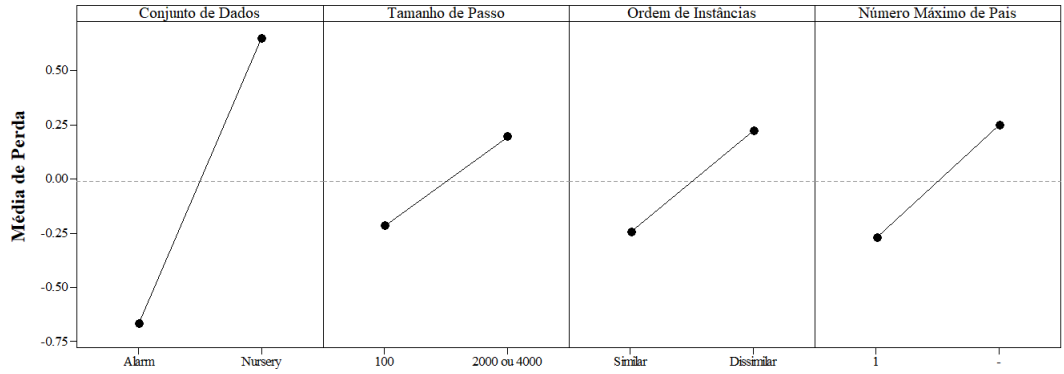

Figura 4.77: Gráfico de efeitos significantes de fatores na perda logarítmica para IHCS

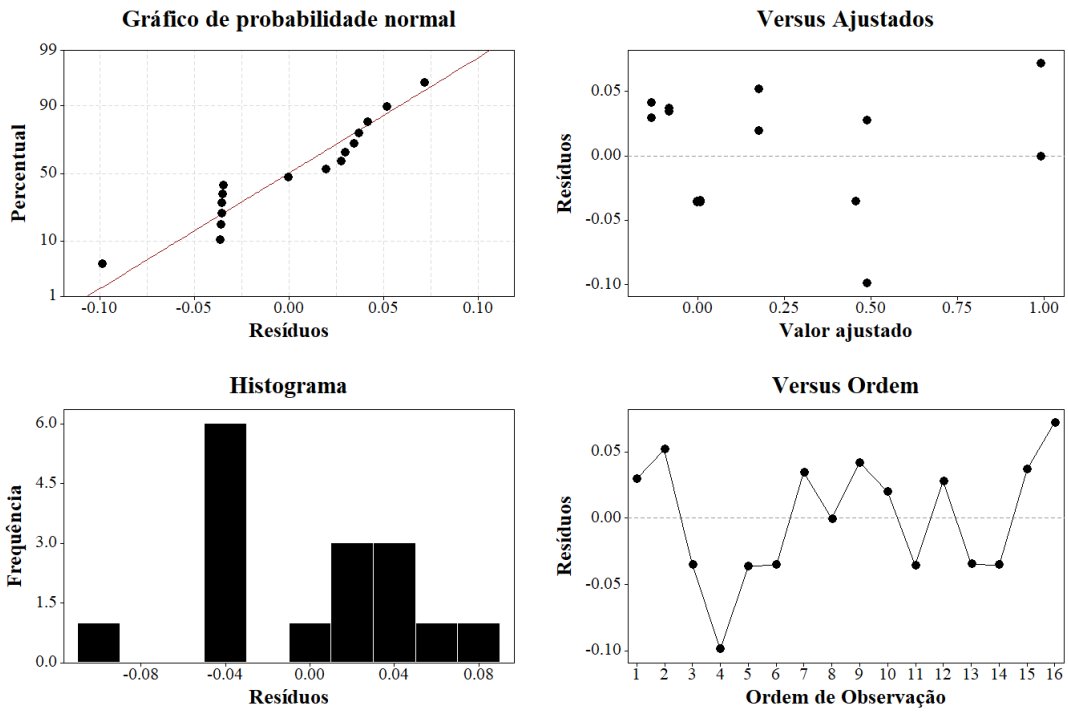

Figura 4.78: Gráficos de resíduos do modelo com efeitos significativos na perda logarítmica para ST

Com os modelos validados, o valor de p é adotado como estatística suficiente para validar a significância dos efeitos analisados. Adotando um nível de significância de 1%, todos os valores de p foram bem próximos ou igual a 0. Dado estes valores, pode-se supor que, com 1% de significância, o número de pais tem efeito significativo na perda logarítmica das redes aprendidas pelo algoritmo ST. Já o valor de α e $nRSS$, não é possível chegar a uma conclusão, pois, apesar de serem pequenos, o valor de alguma de suas interações com outros fatores é confundido com alguns efeitos considerados significantes.

4.4.4 Curva de Acurácia

Os gráficos de pareto dos efeitos na acurácia de predição nos experimentos com ST e IHCS são apresentados nas Figuras 4.80 e 4.81, respectivamente.

Nota-se na Figura 4.80, referente aos experimentos com ST, que existem alguns efeitos principais significativos sobre a métrica analisada. Estes efeitos são dos fatores referentes ao conjunto de dados e ao tamanho do passo. Há também efeitos significativos de interações

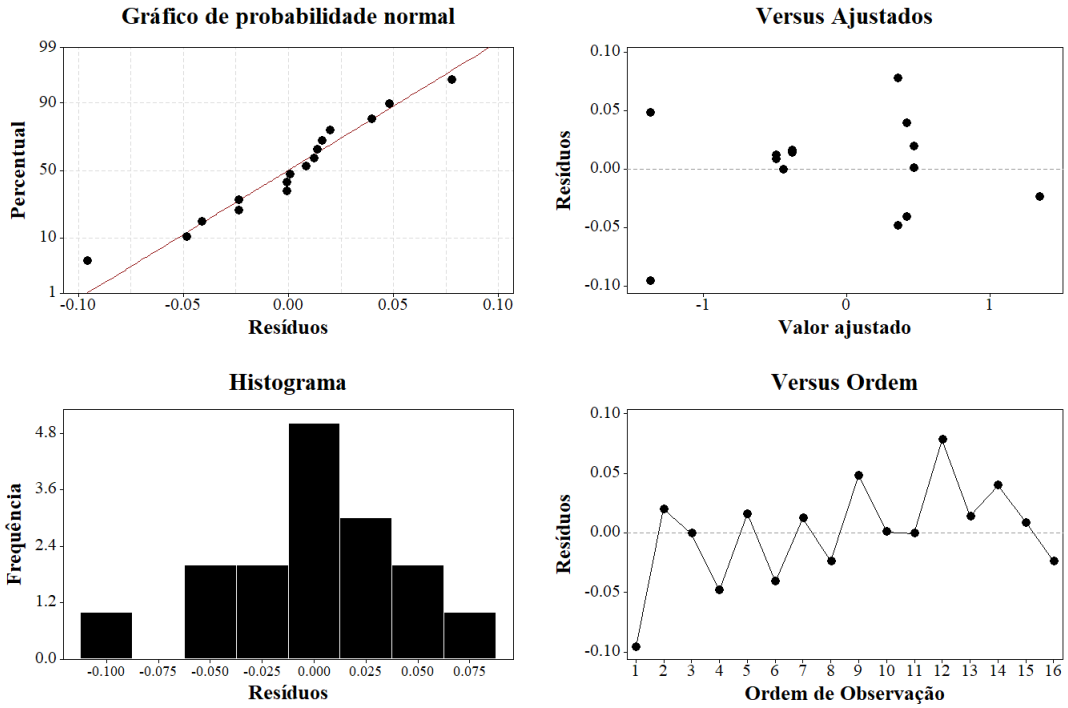

Figura 4.79: Gráficos de resíduos do modelo com efeitos significativos na perda logarítmica para IHCS

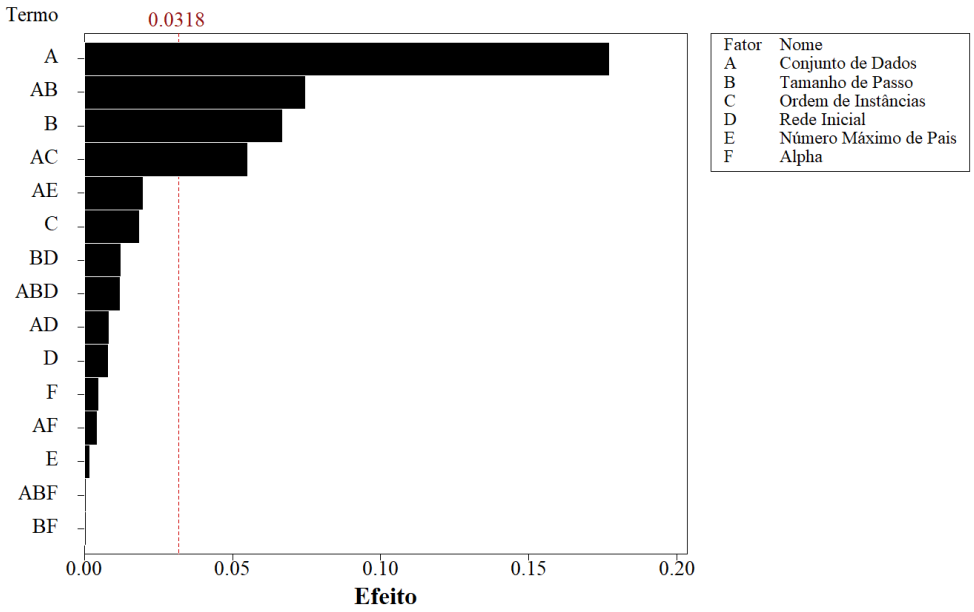

Figura 4.80: Gráfico de pareto dos efeitos padronizados na acurácia para ST

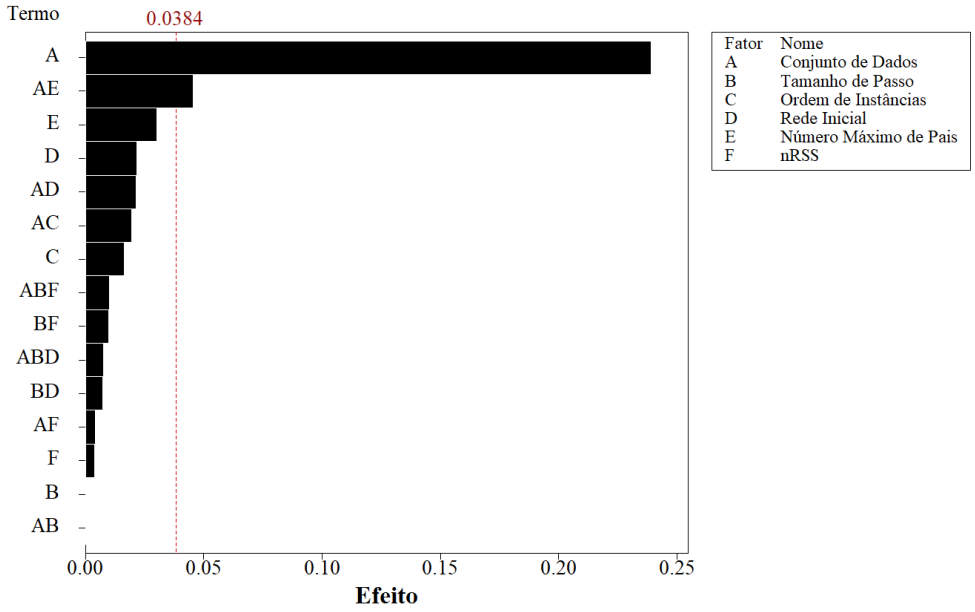

Figura 4.81: Gráfico de pareto dos efeitos padronizados na acurácia para IHCS

entre o conjunto de dados, tamanho do passo e a ordem das instâncias. Já na Figura 4.81, referente aos experimentos com IHCS, são descritos os efeitos do conjunto de dados e de sua interação com o número de pais como significativos.

As direções dos efeitos significativos dentro dos experimentos com ST e IHCS são apresentadas nas Figuras 4.82 e 4.83. Diferente do que aconteceu com a métrica referente à perda do modelo, pode-se perceber que, todos os efeitos analisados, independente do algoritmo experimentado, possuem efeito negativo no valor da acurácia. Na Figura 4.84, é descrita a influência do efeito da interação entre o conjunto de dados e o número máximo de pais no experimento com IHCS.

Nota-se na Figura 4.84 que o conjunto de dados possui uma influência negativa alta da acurácia. Além disto, quando o número máximo de pais é restrito, a acurácia na base de dados *Alarm* é menor que quando o número de pais não é restrito, com uma diferença de, aproximadamente, 5%. No entanto, esse comportamento é contrário no conjunto de dados *Nursery*. Para validar as avaliações sobre o efeito dos fatores citados na acurácia, anova é adotada.

Os comportamentos dos resíduos do modelo gerado são descritos nas Figuras 4.85 e 4.86. Pode-se notar a distribuição normal dos resíduos em ambos os histogramas das Figu-

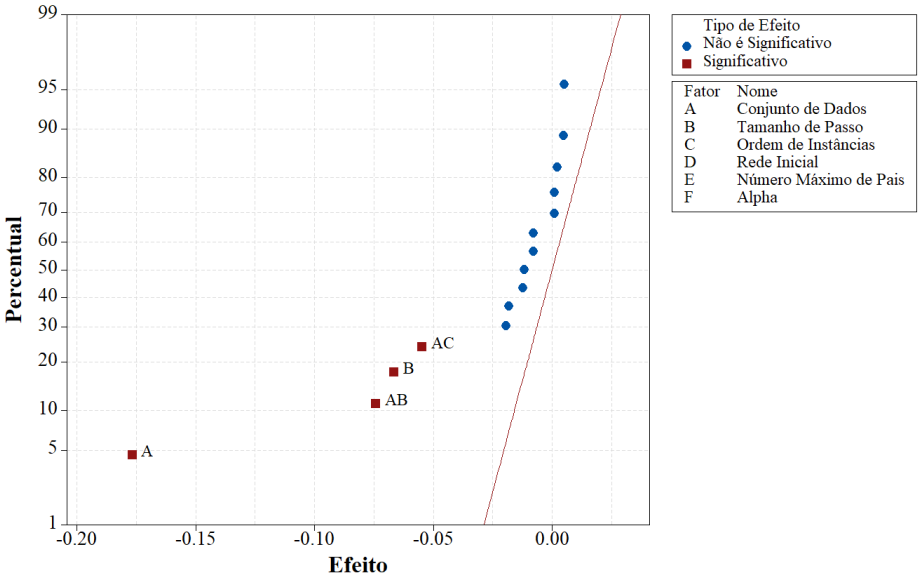

Figura 4.82: Gráficos de probabilidade normal dos efeitos na acurácia para ST

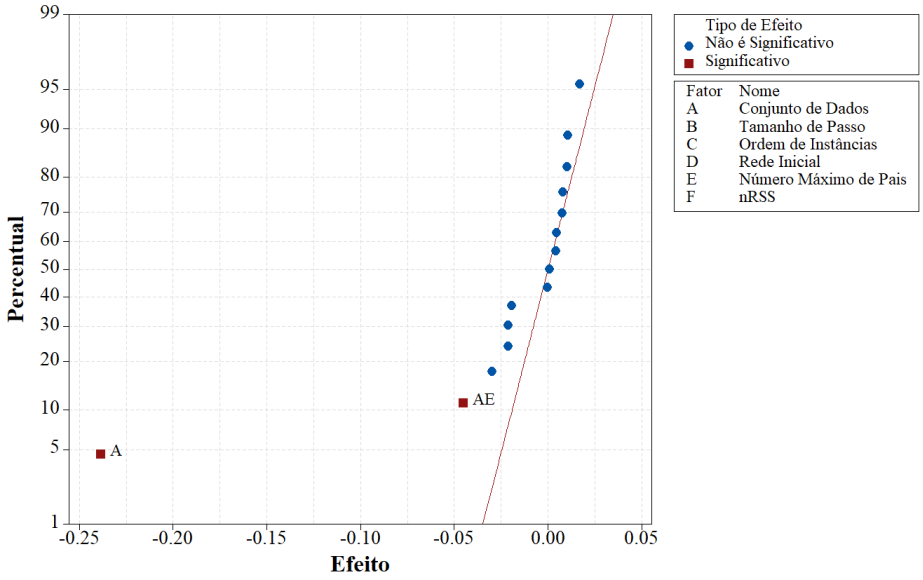

Figura 4.83: Gráficos de probabilidade normal dos efeitos na acurácia para IHCS

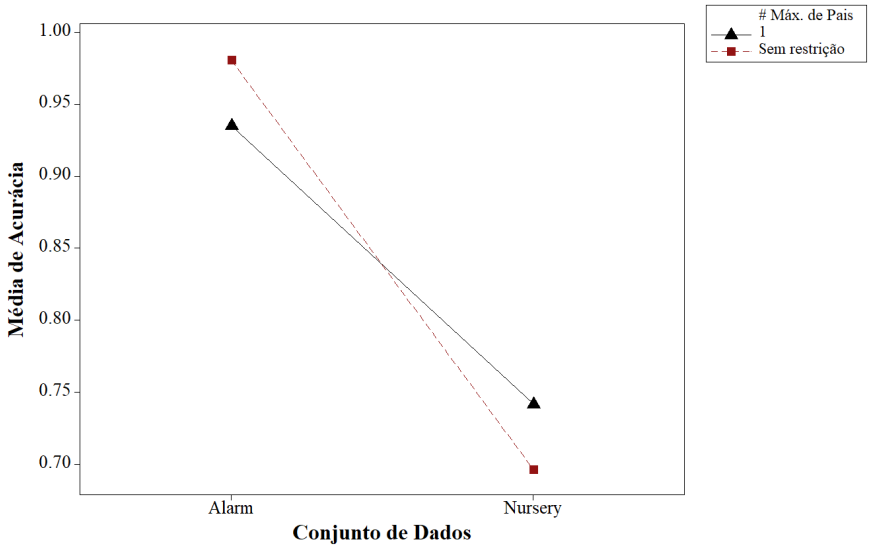

Figura 4.84: Gráfico de efeitos significantes de interações na acurácia para IHCS

ras 4.85 e 4.86. Ambas as distribuições dos resíduos foram testadas e, dado um valor de p maior que 1 nos teste de normalidade de Ryan-Joiner, supõem-se que a distribuição é normal. Como notado nos gráficos que exibe os resíduos na ordem que foram encontrados e plotados juntamente com o valor ajustado da variável resposta, não há presença de dependência dos resíduos entre eles e heterocedasticidade. Com isso, supõem-se que, com significância de 5%, há efeitos significativos entre o número de pais e a acurácia das redes analisadas. Já para o valor de α e $nRSS$, não é possível chegar a uma conclusão, pois, apesar de serem considerados insignificantes, o valor de alguma de suas interações com outros fatores é confundido com alguns efeitos considerados significantes.

4.4.5 Ameaças à Validade

Nesta seção, as ameças à validade dos experimentos são analisadas. A seguir, os principais tipos de ameaças à validade detectados de acordo com a classificação de Wohlin et al. [57] são detalhados.

Como ameaça à validade interna, pode-se citar a implementação dos algoritmos utilizados neste experimento. Os algoritmos incrementais foram implementados baseados nos pseudocódigos disponibilizados pelos autores em seus trabalhos. Ambos foram implementados e executados em ambientes idênticos e técnicas de algoritmos foram utilizadas para

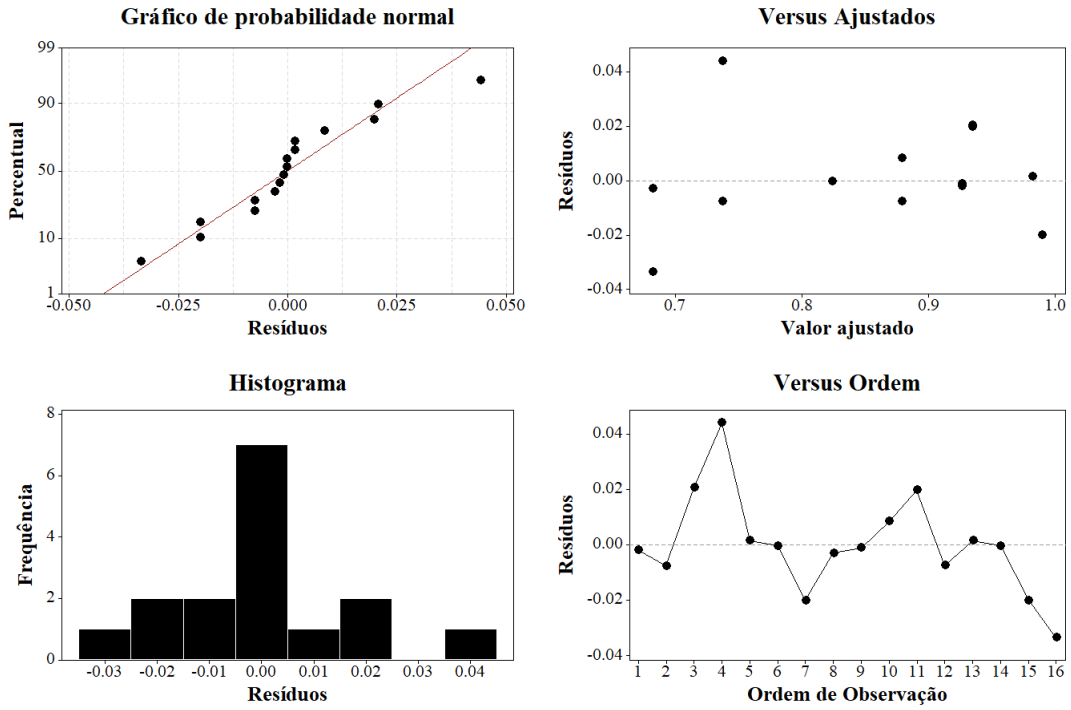

Figura 4.85: Gráficos de resíduos do modelo com efeitos significativos na acurácia para ST

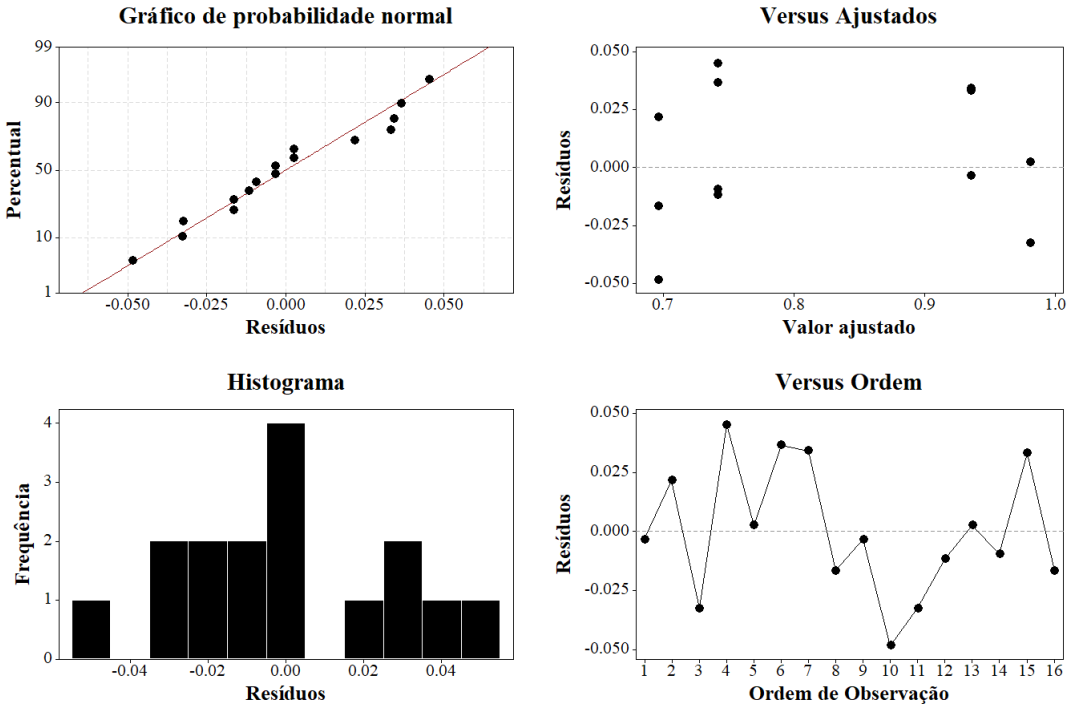

Figura 4.86: Gráficos de resíduos do modelo com efeitos significativos na acurácia para IHCS

tornar a execução mais ágil. Para validação da implementação, buscou-se a replicação dos resultados encontrados pelos autores, mas a maioria é impossível de ser replicado dado a falta da base de dados utilizada nos experimentos.

Como ameaça à validade de conclusão, pode-se citar a baixa significância dos resultados sobre os efeitos que explicam alguma variância nas métricas de acurácia e de perda logarítmica. A anormalidade dos dados, considerando um nível de significância de 5% é o principal motivo desta ameça. Como ameaça à validade interna, pode-se citar as confusões entre os fatores causados pelo design experimental utilizado. Como um design fracionário foi utilizado, os efeitos dos fatores são confundidos com possíveis efeitos de outras interações de fatores, não permitindo afirmar que o efeito citado descreve somente o efeito do fator analisado.

Como ameaça à validade de constructo, pode-se citar a falta de replicações dos ensaios e a explicação dos dados contendo informações reais que, apesar de complexos, ainda sim, são simples se comparados aos dados coletados no cotidiano real. Apesar de dados com informações sobre o mundo real serem utilizados, há uma diferença entre as características desses dados e vários outros conjuntos de dados do mundo real, como dados faltantes e com altos ruídos. Isto também afeta a generalização dos resultados, constituindo uma ameaça à validade externa.

4.5 Comentários sobre Comportamentos de Algoritmos

Em parte dos experimentos, o algoritmo ST atinge pontuações DCM melhores quando os passos são menores. No entanto, essa situação se inverte assim que o tamanho do passo é incrementado. Esse comportamento ocorre devido ao cenário projetado, iniciando os experimentos com uma rede vazia. Dado que sua representação da distribuição de probabilidades do domínio é nula, a rede atual (neste caso, a rede vazia) necessita de grandes ajustes de acordo com as novas instâncias de dados. Nota-se na Subseção 4.1.2 que, quanto maior o passo, menor é a variação entre as distribuições. Isto faz com que grandes instâncias de dados também causem grandes variações na RB gerada.

Para melhor compreensão desse *tradeoff*, a diferença entre a pontuação dos modelos gerados durante o processo de aprendizagem com $k = 1000$ na base de dados *Nursery* utilizando os dois algoritmos incrementais é apresentada na Figura 4.87.

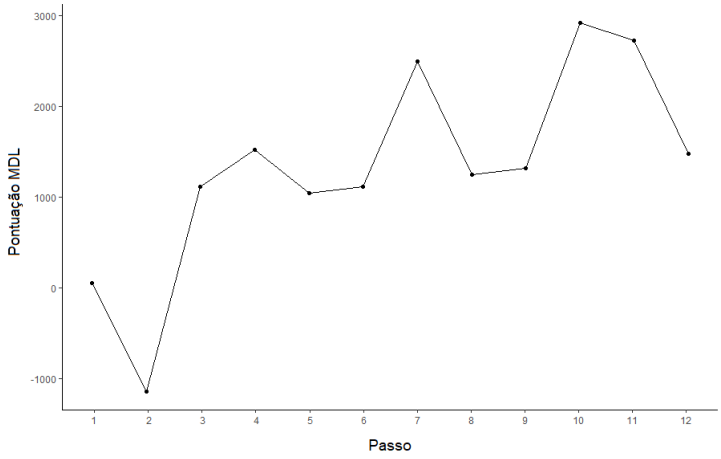

Figura 4.87: Variação entre pontuação de modelos em *Nursery*

Nota-se que, nos primeiros passos, a pontuação DCM do modelo gerado pelo algoritmo ST é menor do que a do modelo gerado pelo IHCS, indicando um melhor modelo. Essa melhor pontuação ocorre no contexto de maior mudança no modelo gerado. O IHCS possui a suposição de não realizar grandes alterações no modelo, apenas refiná-lo. Enquanto que ST reinicia o processo de busca a cada novo conjunto de dados, o IHCS apenas estende as buscas entre os vizinhos de um determinado nó que, além de possuir um conjunto de possíveis pais controlado por k, mantém toda a estrutura considerada corretamente ordenada sem alterações. Esse comportamento permite que ST não necessite da suposição citada para o IHCS.

Esta configuração no comportamento de operações nos modelos pode ser vista nas figuras a seguir. Como exemplo, nas Figuras 4.88 e 4.89, as alterações no comportamento das operações durante o procedimento de aprendizagem utilizando a base de dados *Alarm* são apresentadas. Os comportamentos das operações utilizando a base de dados *Nursery* também são apresentados nas Figuras 4.90 e 4.91.

Em geral, percebe-se que IHCS promove mais alterações nas redes. Vendo o número de arcos extras adicionados pelos algoritmos, nota-se que as operações de adição de novos arcos são mais restritas no algoritmo ST. Isso deve-se à técnica baseada em restrição que o ST faz uso. Esta técnica utiliza melhor as informações disponíveis, não somente as presentes na rede atual ou na nova base de dados, mas também em procedimentos de aprendizagem anteriores.

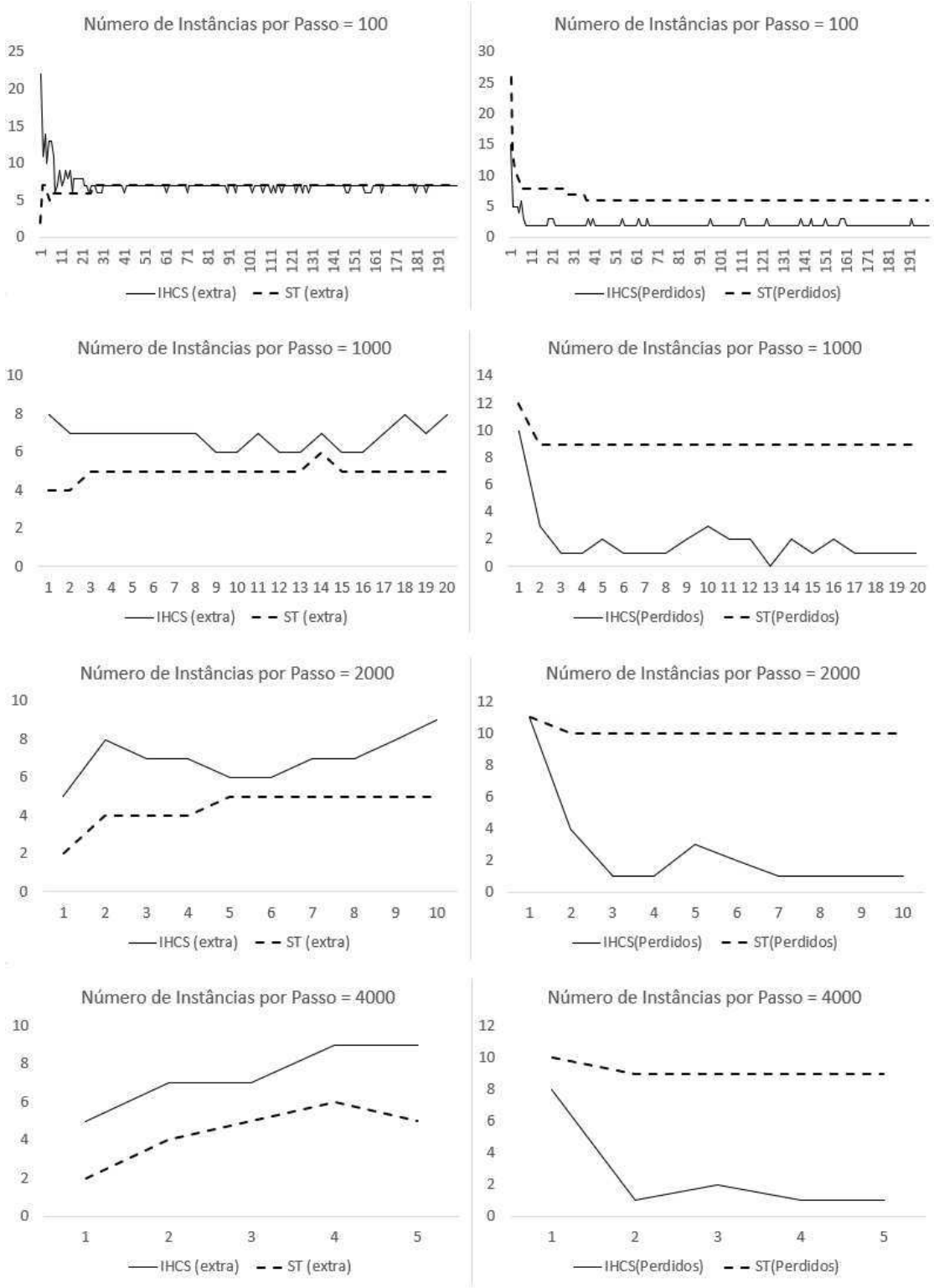

Figura 4.88: Evolução no número de arcos extras e perdidos em aprendizagem utilizando *Alarm*

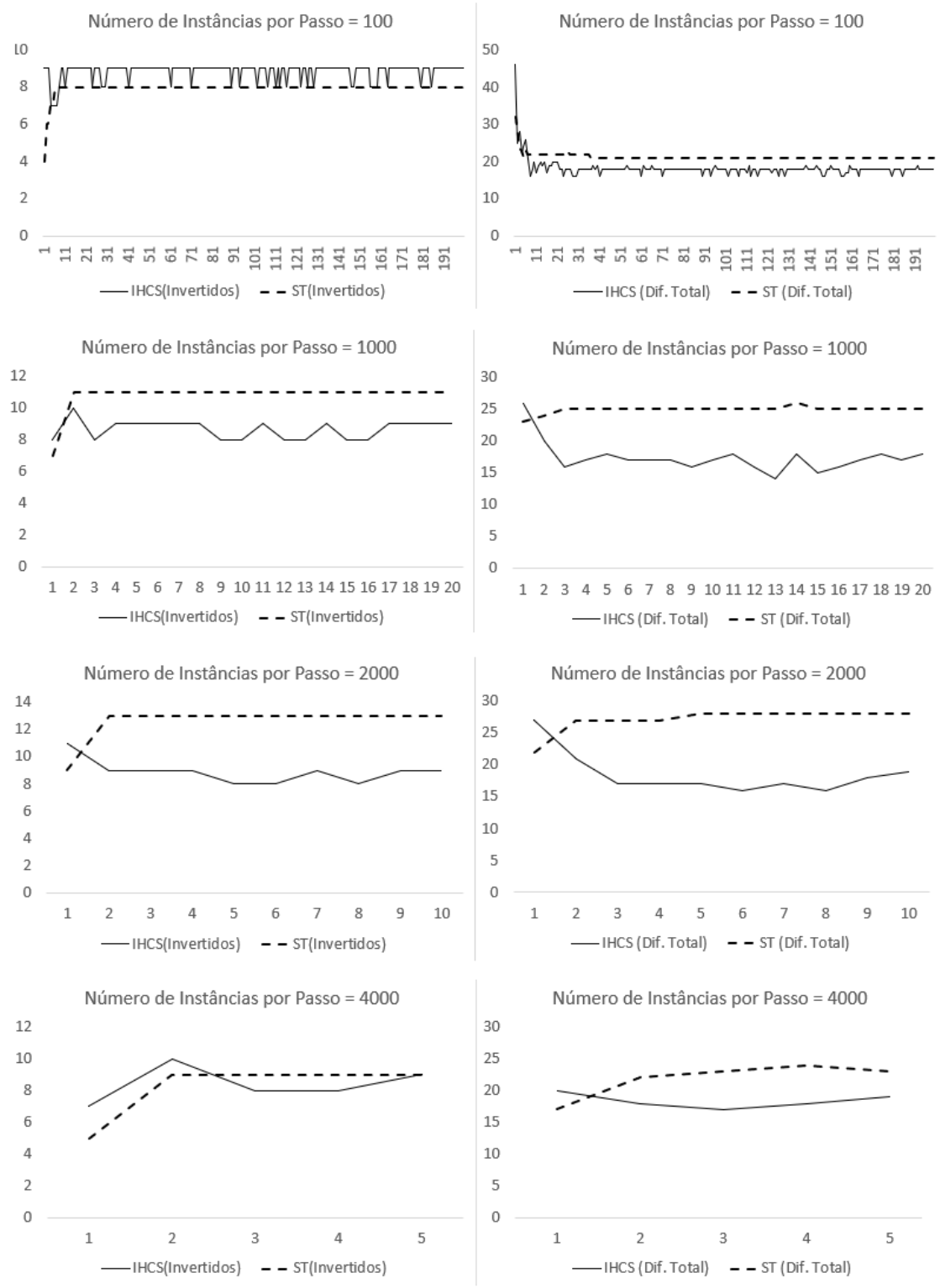

Figura 4.89: Evolução no número de arcos invertidos e diferentes em aprendizagem utilizando *Alarm*

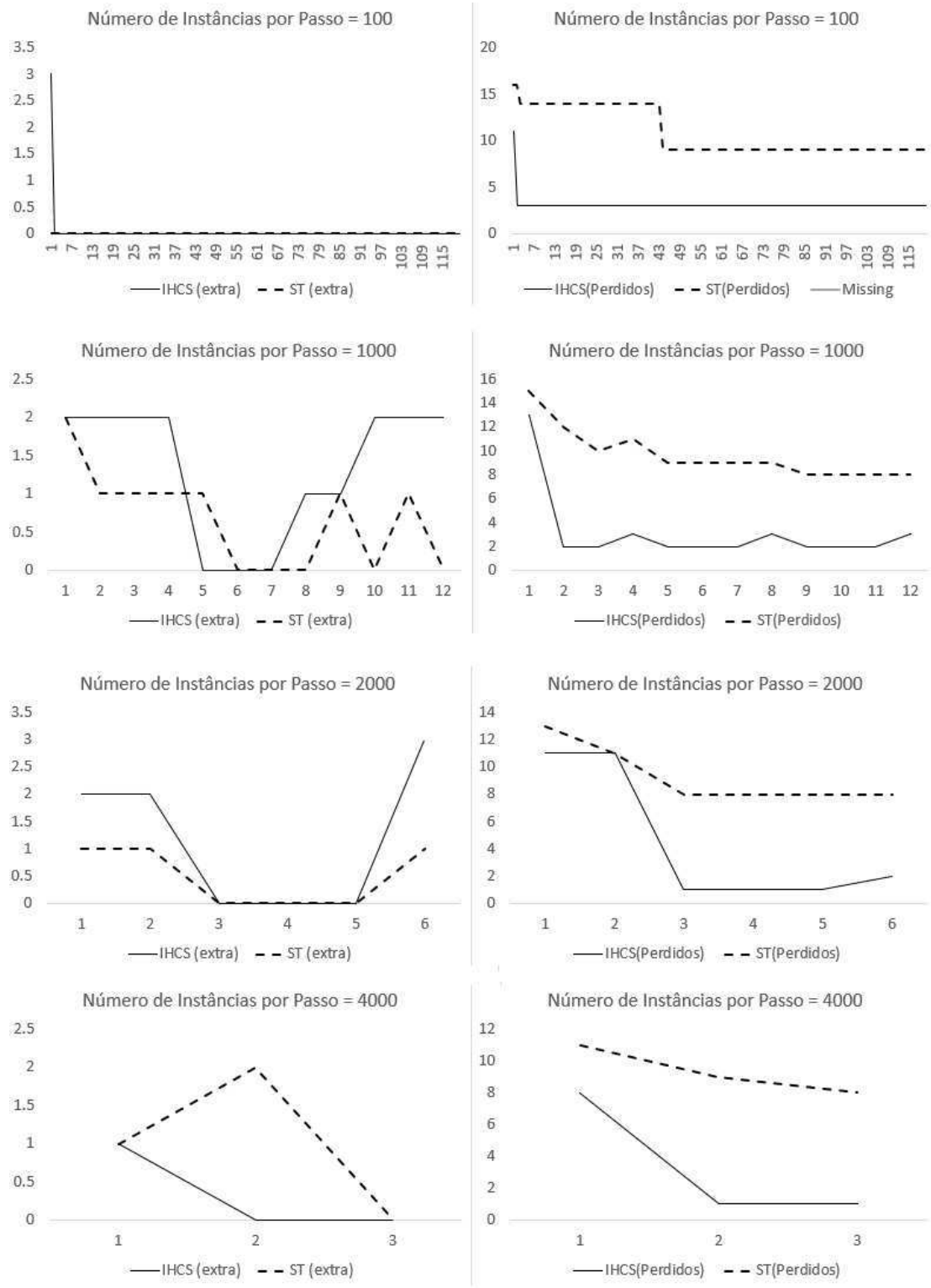

Figura 4.90: Evolução no número de arcos extras e perdidos em aprendizagem utilizando *Nursery*

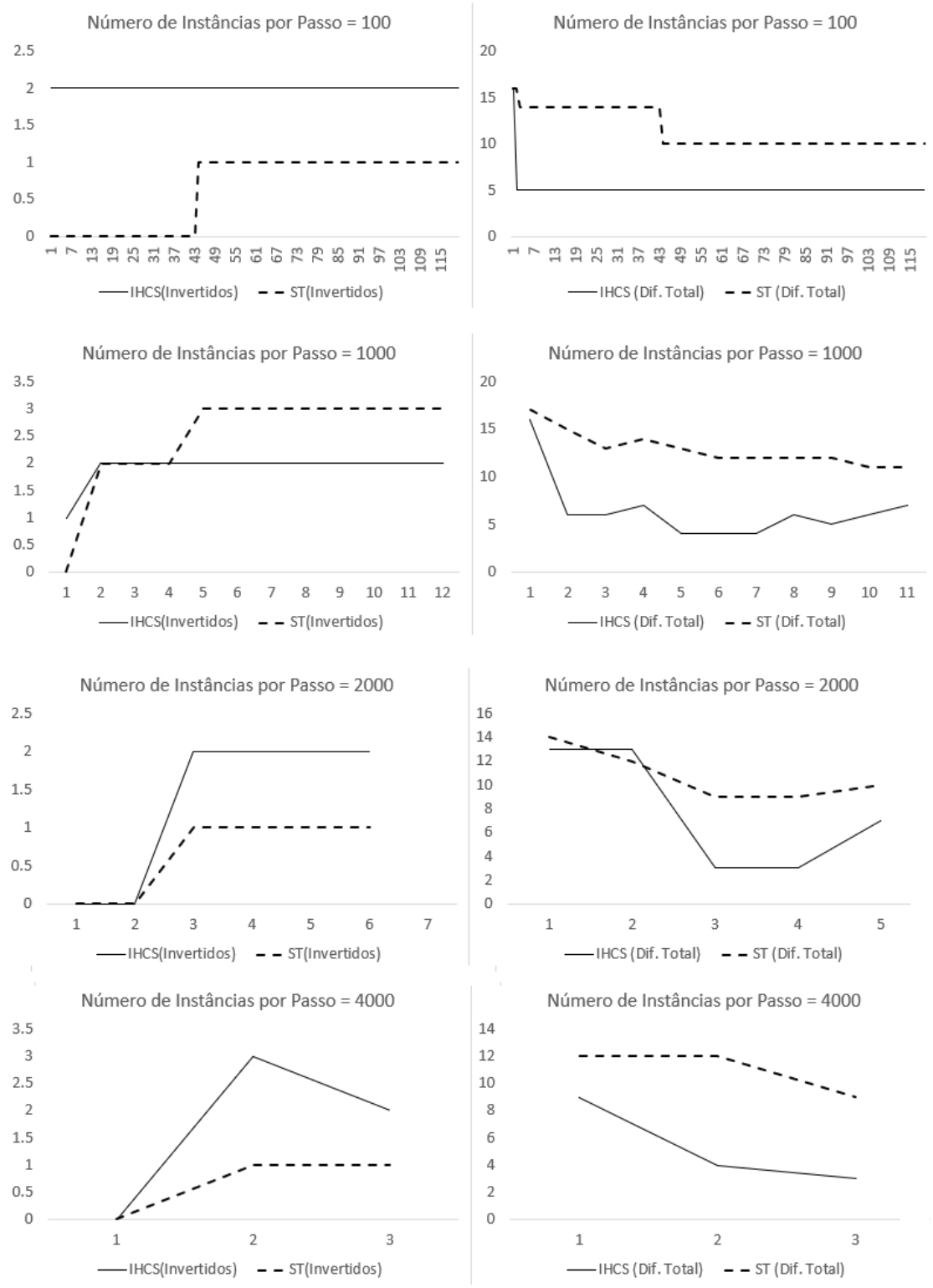

Figura 4.91: Evolução no número de arcos invertidos e diferentes em aprendizagem utilizando *Nursery*

A alta taxa de arcos perdidos pode também ser notada. Esse número é proporcionalmente maior nas bases de dados reais. Isso acontece porque os algoritmos possuem uma certa dificuldade em encontrar relação entre os atributos em bases de dados mais complexas, ou seja, que possuem alta divergência entre as informações dos atributos. Nota-se que, quanto maior o passo, menor é a tendência de existirem arcos perdidos ou extras já que mais informação sobre as variáveis é inserida nos algoritmos. Já o número de arcos revertidos tende a aumentar dado a dificuldade dos algoritmos de indicarem a direção correta dos arcos.

Agora, o desempenho das RBs aprendidas com os diferentes algoritmos é analisado através da métrica citada como perda logarítmica. Nas Figuras 4.92, 4.93 e 4.94, são exibidas as curvas de aprendizagem dos modelos gerados com as bases de dados *Alarm*, *Asia* e *Nursery*, respectivamente. Nestas figuras, o desempenho do algoritmo em lote e as abordagens incrementais são apresentados à medida que são alimentados com os subconjuntos de dados já citados. Com esses gráficos, o desempenho do lote com as abordagens incrementais pode ser comparado.

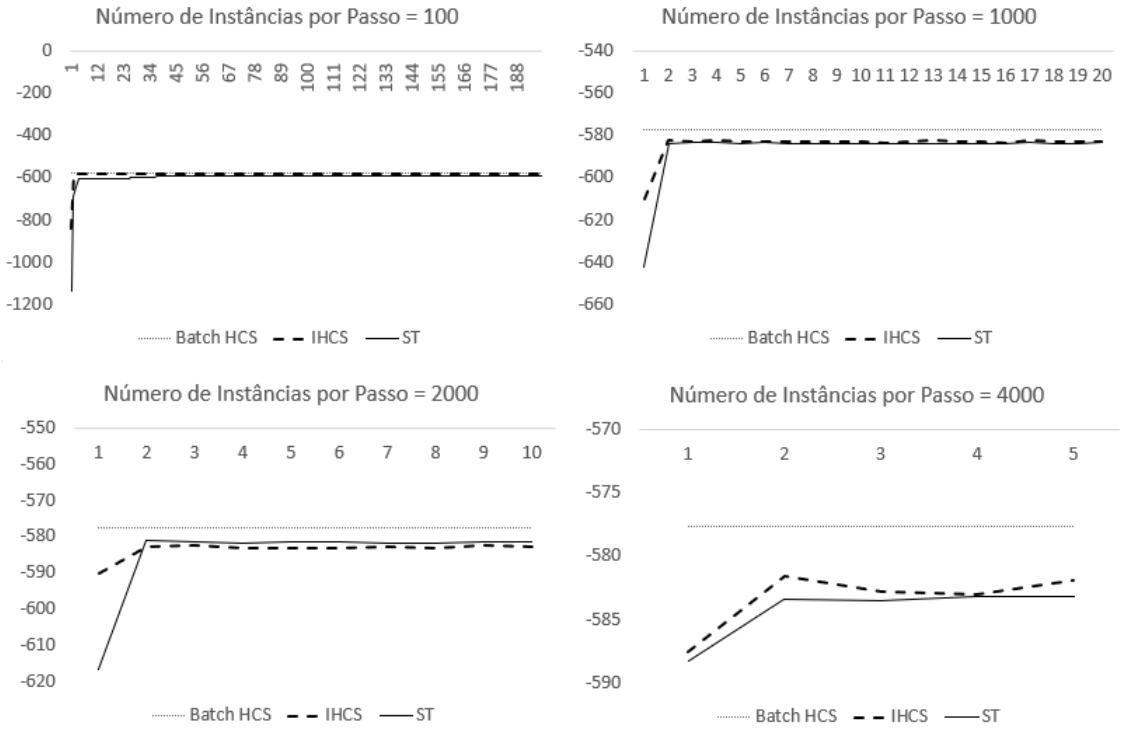

Figura 4.92: Evolução na perda logarítmica dos modelos gerados em *Alarm*

Nestas figuras, nota-se que os dois algoritmos incrementais tem comportamentos como

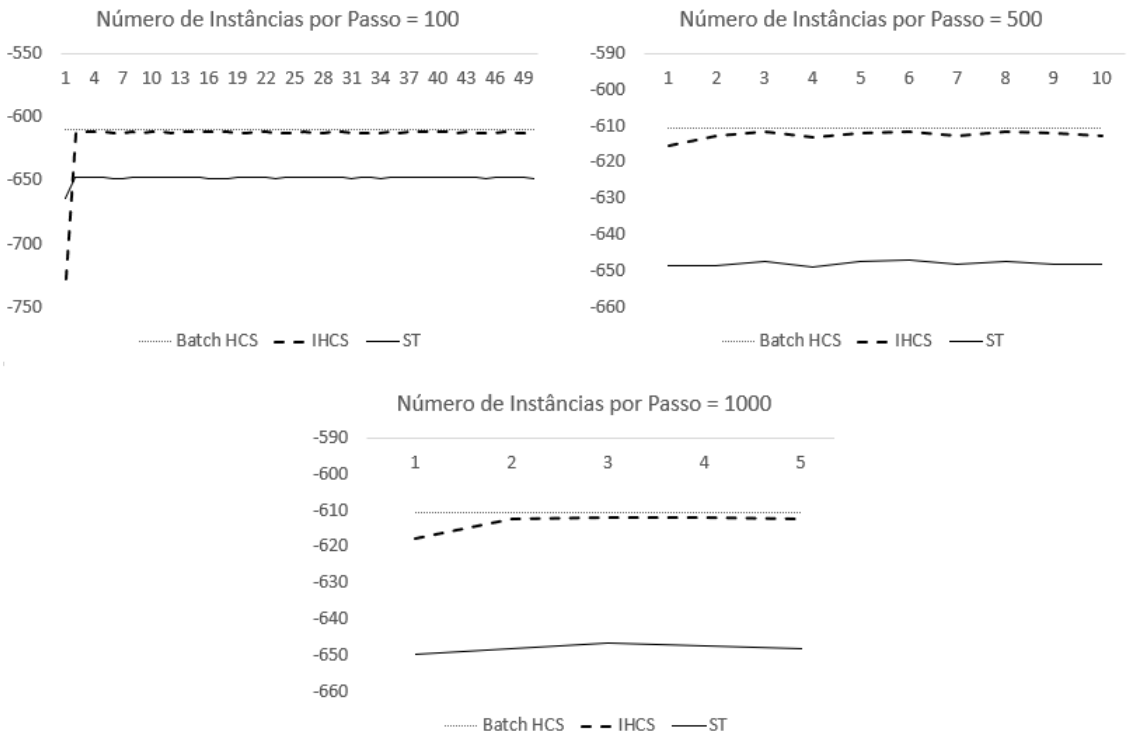

Figura 4.93: Evolução na perda logarítmica dos modelos gerados em *Asia*

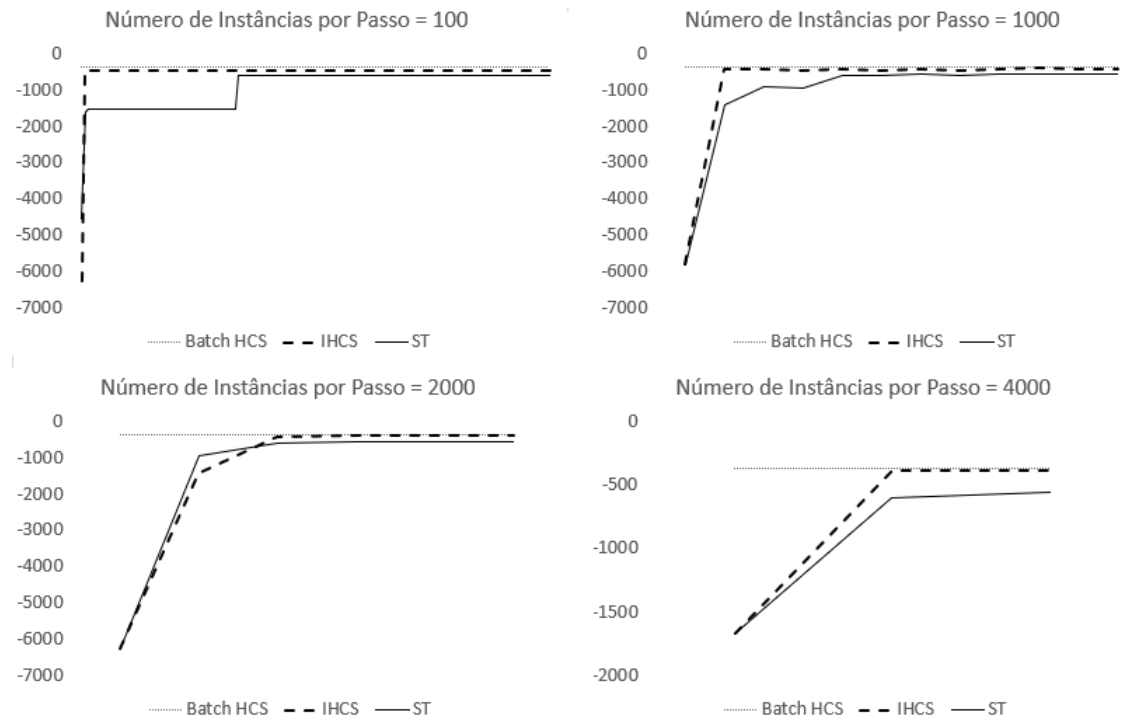

Figura 4.94: Evolução na perda logarítmica dos modelos gerados em *Nursery*

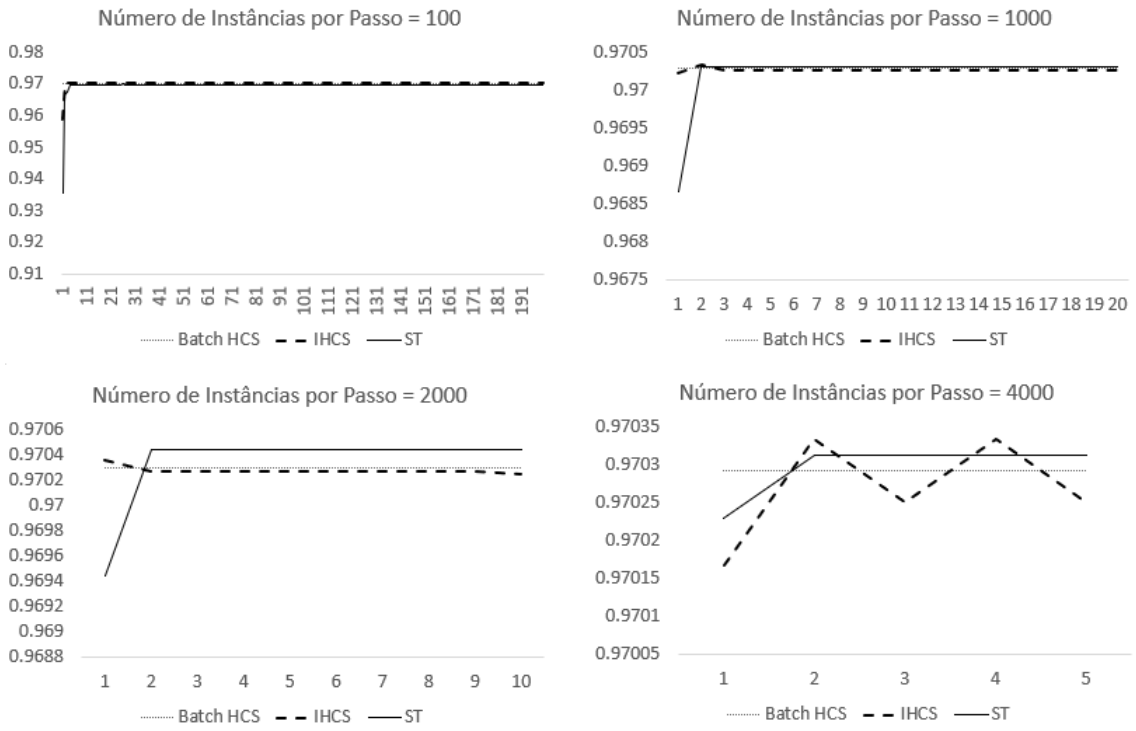

Figura 4.95: Evolução na acurácia dos modelos gerados em *Alarm*

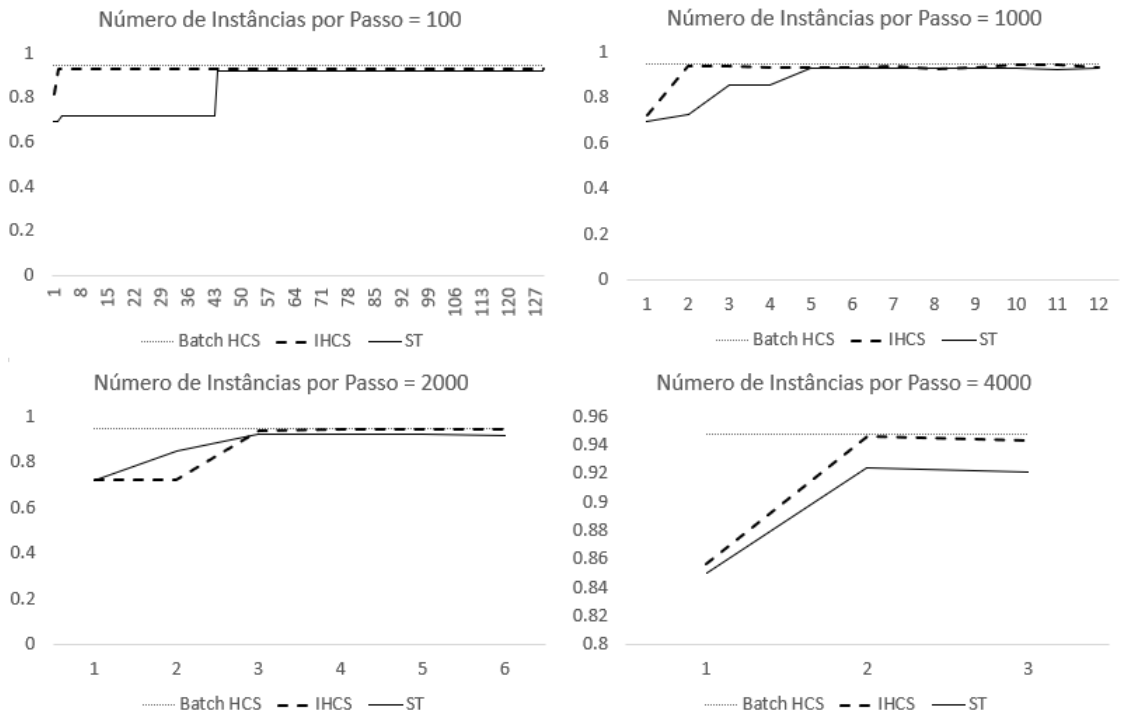

Figura 4.96: Evolução na acurácia dos modelos gerados em *Nursery*

esperado, isto é, as curvas de aprendizado convergem rapidamente para o valor de desempenho das RBs obtidas em lote. No entanto, pode-se notar uma exceção no desempenho de ST na base de dados *Asia*, como visto na Figura 4.93. Esse comportamento ocorre devido às dificuldades do algoritmo encontrar as dependências condicionais corretas do atributo E (tuberculose vs bronquite). Quando comparado o desempenho dos algoritmos dentro de processos de aprendizagem com dados de tamanhos idênticos, nota-se que ST demora mais para convergir ao valor desempenhado por HCS.

Nas Figura 4.95 e 4.96, são apresentadas as evoluções da acurácia para os conjuntos de dados *Alarm* e *Nursery*, respectivamente. Nas duas figuras, o desempenho do algoritmo em lote e as abordagens incrementais são apresentados à medida que são alimentados com os subconjuntos de dados. Nota-se que os algoritmos incrementais se comportam bem diante de mesmo contexto quando comparados ao HCS, o que é esperado. Quando comparado o desempenho dos algoritmos dentro de processos de aprendizagem com dados de tamanhos idênticos, nota-se também que ST demora mais para convergir ao valor desempenhado por HCS.

Capítulo 5

Conclusão e Futuras Pesquisas

Nesta dissertação, o principal objetivo foi avaliar algoritmos de aprendizagem incremental de estruturas que incorporam continuamente as instâncias de treinamento à medida que chegam, aplicando-os em contextos complexos.

O estudo realizado neste trabalho se apresenta como uma alternativa promissora para auxiliar os avanços das pesquisas no campo de aprendizagem incremental de estruturas de RBs. Neste trabalho, a literatura sobre os algoritmos de aprendizagem incremental foi analisada a partir de dados e um resumo de publicações relevantes foi apresentado com o objetivo de que este seja usado como uma referência pronta para novos trabalhos. Tópicos específicos foram focados em detalhes, onde parte dos principais campos da área foram abordados. Atualmente, na literatura, há uma falta de trabalhos relacionados na área.

Dentre as soluções encontradas na literatura, foram destacados dois algoritmos que possuem metodologias idênticas, utilizando restrições de pais e buscas baseadas em pontuação. Nomeadamente, são os algoritmos ST, apresentado por Shi e Tan [50], e IHCS, apresentado por Alcobé [46]. Foram realizados estudos empíricos para avaliar o comportamento dos métodos de aprendizagem incremental em contextos com diferentes complexidades. Considerando a falta de estudo similares na literatura, foi concluído que este trabalho também pode ser usado como subsídio no sentido de diminuir a imprecisão sobre algoritmos incrementais e justificar, de forma empírica mais clara, a sua importância no processo de aprendizagem. Além disto, este estudo empírico foi expandido para auxiliar a definição sobre a escolha de algoritmos incrementais e seus fatores. Este estudo abordou questões importantes na definição da qualidade do modelo que deixam de ser analisadas em estudos similares.

Dentre os resultados dos estudos empíricos, são destacados:

- ambos os algoritmos incrementais apresentaram modelos idênticos, em pontuação, àqueles que seriam obtidos por algoritmos de lote semelhantes;
- ambos os algoritmos incrementais apresentaram modelos diferentes, estruturalmente e em acurácia, àqueles que seriam obtidos por algoritmos de lote semelhantes;
- a versão incremental do HCS não adequou-se tão bem a uma grande variação na distribuição dos dados quanto ST;
- ST apresentou maus resultados quando a dependência entre atributos não foi clara;
- ST aprendeu, de forma mais lenta, modelos de qualidade idêntica ao HCS;
- características como ordem das instâncias inseridas no processo de aprendizagem, tamanho do passo de aprendizagem foram consideradas como influentes na qualidade final das redes produzidas;
- a restrição referente ao número máximo de pais possuiu efeito significativo, por vezes, negativos, nas métricas de qualidade analisadas.

Além disso, um novo pacote que pode ser utilizando como uma camada extra da biblioteca de mineração de dados, Weka, foi desenvolvido. Este pacote contém alguns das funcionalidades necessárias para manipular o aprendizado incremental utilizando esta biblioteca Java.

5.1 Trabalhos Futuros

Apesar dos resultados obtidos, acredita-se que há um enorme potencial de melhoria e mais trabalhos de pesquisa relacionados à aprendizagem incremental de estruturas de RBs podem ser desenvolvidos.

Durante os experimentos, foi percebido que o foco dos algoritmos incrementais é a velocidade do procedimento de aprendizagem. Ambos os algoritmos manusearam os dados, em média, mais rápido que o algoritmo em lote. O ponto mais notável dessa vantagem foi o fim

do procedimento, onde o algoritmo de aprendizagem em lote deveria lidar com a base completa dos dados, enquanto as versões incrementais apresentaram resultados semelhantes com apenas mais uma quantidade pequena de instâncias. No entanto, inúmeros contextos ainda necessitam de baixa complexidade computacional e alta qualidade de predição no tratamento dos dados, como contextos de Indústria 4.0.

Uma das principais dificuldades dos algoritmos foi a geração de modelos com arcos invertidos. Ou seja, eles foram capazes de descobrir a conexão entre os atributos, mas invertendo a relação causa-efeito. Este comportamento pode ser descoberto pelo especialista de contexto durante os passos do processo de aprendizagem. O conhecimento do especialista de domínio pode ser utilizado como mais uma fonte de conhecimento a posteriori no processo, já que ele é construído gradativamente. Isso significa que alterações no modelo, como adição de novos nós ou inversão de arcos para melhorar a compreensão humana, podem ser utilizadas no processo e uma provável conversão ao melhor modelo possível, em termos de acurácia e perda, pode acontecer mais rápido.

Além disso, diversos domínios reais de aplicação dos algoritmos de aprendizagem incremental ainda seguem um comportamento diferente das bases de dados reais utilizadas neste estudo. Em muito deles, há escassez de dados, alta divergência nas informações sobre as possíveis variáveis no modelo, adições de novos atributos durante o processo de aprendizagem. Portanto, também é necessário a aplicação e avaliação de algoritmos incrementais em ambientes ainda mais complexos.

Pretende-se realizar novos estudos considerando também a complexidade computacional dos algoritmos, além de abordar outras soluções presentes na literatura e que podem apresentar resultados promissores quando considera-se a complexidade computacional e a qualidade de predição.

Bibliografia

[1] Bruce Abramson, John Brown, Ward Edwards, Allan Murphy, and Robert L Winkler. Hailfinder: A bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12(1):57–71, 1996.

[2] Josep Roure Alcobé. Incremental hill-climbing search applied to bayesian network structure learning. In *Proceedings of the 15th European conference on machine learning, Pisa, Italy*, 2004.

[3] Shunichi Amari et al. *The handbook of brain theory and neural networks*. MIT press, 2003.

[4] Miguel Barao. Entropia, entropia relativa e informaçao mútua. *Universidade de Evora*, 2003.

[5] Ingo A Beinlich, Henri Jacques Suermondt, R Martin Chavez, and Gregory F Cooper. The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89*, pages 247–256. Springer, 1989.

[6] Irad Ben-Gal. Bayesian networks. *Encyclopedia of statistics in quality and reliability*, 1, 2008.

[7] David Budgen and Pearl Brereton. Performing systematic literature reviews in software engineering. In *Proceedings of the 28th international conference on Software engineering*, pages 1051–1052. ACM, 2006.

[8] Wray Buntine. Theory refinement on bayesian networks. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann Publishers Inc., 1991.

[9] C Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

[10] Guo Chunsheng and Sun Qiquan. Incremental structure optimize of bayesian network based on the lossless decomposition. In *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*, volume 2, pages 155–159. IEEE, 2010.

[11] Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

[12] Sami Demiroluk and Kaan Ozbay. Adaptive learning in bayesian networks for incident duration prediction. *Transportation Research Record: Journal of the Transportation Research Board*, 2460(1):77–85, 2014.

[13] Pedro M Domingos and Geoff Hulten. Catching up with the data: Research issues in mining data streams. In *DMKD*, 2001.

[14] Benjamin Durakovic. Design of experiments application, concepts, examples: state of the art. *Periodicals of Engineering and Natural Sciences (PEN)*, 5(3), 2017.

[15] Chin-Feng Fan and Yuan-Chang Yu. Bbn-based software project risk management. *Journal of Systems and Software*, 73(2):193–203, 2004.

[16] Norman Fenton and Martin Neil. *Risk assessment and decision analysis with Bayesian networks*. Crc Press, 2012.

[17] Douglas Fisher, Ling Xu, and Nazih Zard. Ordering effects in clustering. In *Machine Learning Proceedings 1992*, pages 163–168. Elsevier, 1992.

[18] Douglas H Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine learning*, 2(2):139–172, 1987.

[19] M Julia Flores, Ann E Nicholson, Andrew Brunskill, Kevin B Korb, and Steven Mascaro. Incorporating expert knowledge when learning bayesian network structure: a medical case study. *Artificial intelligence in medicine*, 53(3):181–204, 2011.

[20] Jeff Forbes, Timothy Huang, Keiji Kanazawa, and Stuart Russell. The batmobile: Towards a bayesian automated taxi. In *IJCAI*, volume 95, pages 1878–1885, 1995.

[21] Nir Friedman and Moises Goldszmidt. Sequential update of bayesian network structure. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 165–174. Morgan Kaufmann Publishers Inc., 1997.

[22] Maxime Gasse, Alex Aussem, and Haytham Elghazel. A hybrid algorithm for bayesian network structure learning with application to multi-label learning. *Expert Systems with Applications*, 41(15):6755–6772, 2014.

[23] John H Gennari, Pat Langley, and Doug Fisher. Models of incremental concept formation. *Artificial intelligence*, 40(1-3):11–61, 1989.

[24] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. "O'Reilly Media, Inc.", 2017.

[25] David Heckerman. A tutorial on learning with bayesian networks. In *Learning in graphical models*, pages 301–354. Springer, 1998.

[26] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[27] Hao Huang, Hantao Song, Fengzhan Tian, Yuchang Lu, and Quande Wang. A comparatively research in incremental learning of bayesian networks. In *Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, volume 5, pages 4260–4264. IEEE, 2004.

[28] Dirk Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. *Bioinformatics*, 19(17):2271–2282, 2003.

[29] Kawal Jeet, Nitin Bhatia, and Rajinder Singh Minhas. A bayesian network based approach for software defects prediction. *ACM SIGSOFT Software Engineering Notes*, 36(4):1–5, 2011.

[30] Tonáš Kočka and Robert Castelo. Improved learning of bayesian networks. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 269–276. Morgan Kaufmann Publishers Inc., 2001.

[31] Wai Lam. Bayesian network refinement via machine learning approach. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 20(3):240–251, 1998.

[32] Wai Lam and Fahiem Bacchus. Learning bayesian belief networks: An approach based on the mdl principle. *Computational intelligence*, 10(3):269–293, 1994.

[33] Wai Lam and Fahiem Bacchus. Using new data to refine a bayesian network. In *Uncertainty Proceedings 1994*, pages 383–390. Elsevier, 1994.

[34] Pat Langley. Order effects in incremental learning. *Learning in humans and machines: Towards an interdisciplinary learning science. Pergamon*, 136:137, 1995.

[35] Pedro Larrañaga, Roberto Murga, Mikel Poza, and Cindy Kuijpers. Structure learning of bayesian networks by hybrid genetic algorithms. In *Learning from Data*, pages 165–174. Springer, 1996.

[36] Steffen L Lauritzen. The em algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis*, 19(2):191–201, 1995.

[37] Eunchang Lee, Yongtae Park, and Jong Gye Shin. Large engineering project risk management using a bayesian belief network. *Expert Systems with Applications*, 36(3):5880–5887, 2009.

[38] Shuohao Li, Jun Zhang, Boliang Sun, and Jun Lei. An incremental structure learning approach for bayesian network. In *Control and Decision Conference (2014 CCDC), The 26th Chinese*, pages 4817–4822. IEEE, 2014.

[39] Weiyi Liu, Kun Yue, Mingliang Yue, Zidu Yin, and Binbin Zhang. A bayesian network-based approach for incremental learning of uncertain knowledge. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 26(01):87–108, 2018.

[40] Brandon Malone, Matti Järvisalo, and Petri Myllymäki. Impact of learning strategies on the quality of bayesian networks: An empirical evaluation. In *UAI*, pages 562–571, 2015.

[41] Søren Holbech Nielsen and Thomas D Nielsen. Adapting bayes network structures to non-stationary domains. *International Journal of Approximate Reasoning*, 49(2):379–397, 2008.

[42] Miloslav Nosal and E Nosal. Statistical distributions in java and internet. In *Proceedings of the ISSAC*, pages 7–10. Citeseer, 2002.

[43] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[44] Parag C Pendharkar, Girish H Subramanian, and James A Rodger. A probabilistic model for predicting software development effort. *IEEE Transactions on software engineering*, 31(7):615–624, 2005.

[45] Robert W Robinson. Counting unlabeled acyclic digraphs. In *Combinatorial mathematics V*, pages 28–43. Springer, 1977.

[46] Josep Roure Alcobe. Incremental methods for bayesian network structure learning. *AI Communications*, 18(1):61–62, 2005.

[47] Saeed Samet, Ali Miri, and Eric Granger. Incremental learning of privacy-preserving bayesian networks. *Applied Soft Computing*, 13(8):3657–3667, 2013.

[48] Ramon Sangüesa and Ulises Cortés. Learning causal networks from data: a survey and a new algorithm for recovering possibilistic causal networks. *AI Communications*, 10(1):31–61, 1997.

[49] Da Shi and Shaohua Tan. Incremental learning bayesian networks for financial data modeling. In *Intelligent Control, 2007. ISIC 2007. IEEE 22nd International Symposium on*, pages 41–46. IEEE, 2007.

[50] Da Shi and Shaohua Tan. Incremental learning bayesian network structures efficiently. In *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pages 1719–1724. IEEE, 2010.

[51] Michael Shwe and Gregory Cooper. An empirical analysis of likelihood-weighting simulation on a large, multiply connected medical belief network. *Computers and Biomedical Research*, 24(5):453–475, 1991.

[52] Luiz Silva, João Nunes, Mirko Perkusich, Kyller Gorgônio, Hyggo Almeida, and Angelo Perkusich. Continuous learning of the structure of bayesian networks: a mapping study. In *Bayesian Networks*. IntechOpen, 2018.

[53] Fengzhan Tian, Hongwei Zhang, Yuchang Lu, and Chunyi Shi. Incremental learning of bayesian networks with hidden variables. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 651–652. IEEE, 2001.

[54] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[55] Kim Verbert, R Babuška, and Bart De Schutter. Bayesian and dempster–shafer reasoning for knowledge-based fault diagnosis–a comparative study. *Engineering Applications of Artificial Intelligence*, 60:136–150, 2017.

[56] Yongheng Wang, Hui Gao, and Guidan Chen. Predictive complex event processing based on evolving bayesian networks. *Pattern Recognition Letters*, 105:207–216, 2018.

[57] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[58] Amanullah Yasin and Philippe Leray. Incremental bayesian network structure learning in high dimensional domains. In *Modeling, Simulation and Applied Optimization (ICMSAO), 2013 5th International Conference on*, pages 1–6. IEEE, 2013.

[59] Barbaros Yet, Zane B Perkins, Todd E Rasmussen, Nigel RM Tai, and D William R Marsh. Combining data and meta-analysis to build bayesian networks for clinical decision support. *Journal of biomedical informatics*, 52:373–385, 2014.

[60] Kun Yue, Qiyu Fang, Xiaoling Wang, Jin Li, and Weiyi Liu. A parallel and incremental approach for data-intensive learning of bayesian networks. *IEEE transactions on cybernetics*, 45(12):2890–2904, 2015.

[61] Yifeng Zeng, Yanping Xiang, and Saulius Pacekajus. Refinement of bayesian network structures upon new data. *International Journal of Granular Computing, Rough Sets and Intelligent Systems*, 1(2):203–220, 2009.

[62] Xuejiao Zhou and Mika Mäntylä. Defect bash-literature review. In *ENASE*, pages 125–131, 2013.

[63] Yun Zhou, Norman Fenton, and Martin Neil. Bayesian network approach to multinomial parameter learning using data and expert judgments. *International Journal of Approximate Reasoning*, 55(5):1252–1268, 2014.

Apêndice A

Mapa Conceitual de Soluções

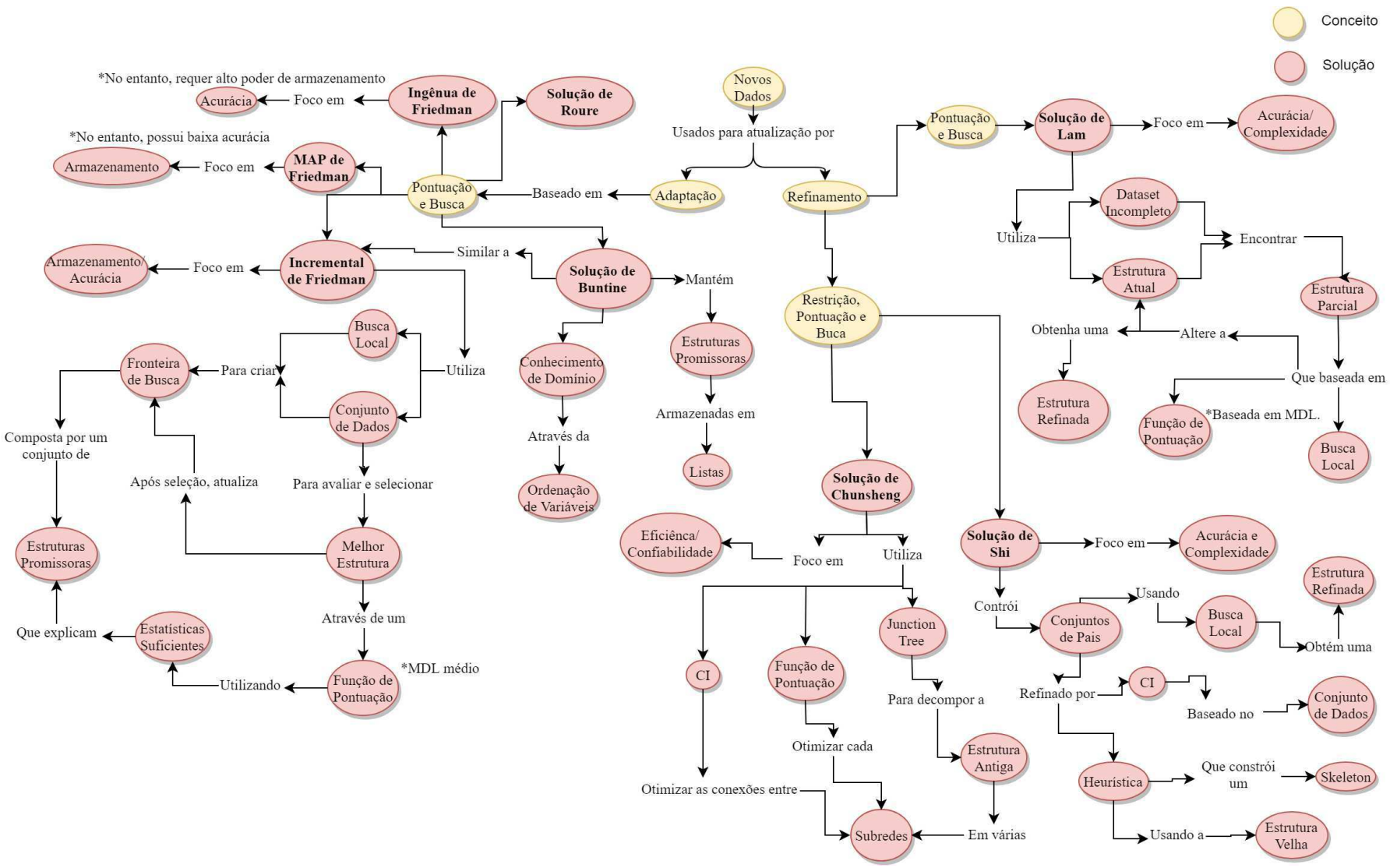

Figura A.1: Mapa conceitual sobre principais soluções

Apêndice B

Ensaios Experimentais

Os ensaios para o experimento que foram realizados para responder a questão de pesquisa **QP1** são descritos na Figura B.1. Este é um experimento fatorial multinível de 3 fatores com 36 ensaios bases e uma réplica.

Os ensaios para responder a questão de pesquisa **QP2** são descritos na Figura B.2. Este é um experimento fatorial multinível de 5 fatores com 72 ensaios bases e uma réplica.

Os ensaios para responder a questão de pesquisa **QP3** são descritos nas Figuras B.3, para IHCS, e B.4, para ST. Os experimentos adotados são experimentos fatoriais fracionados de 6 fatores com 16 ensaios bases e uma réplica.

A resolução desde experimento é IV, onde os gerados são E = ABC e F = BCD. A identidade é definida por I = ABCE = BCDF = ADEF.

Algoritmo	Ordem de Instâncias	Conjunto de Dados
IHCS	Randômica	Alarm
IHCS	Randômica	Asia
IHCS	Randômica	Car
IHCS	Randômica	Nursery
IHCS	Similar	Alarm
IHCS	Similar	Asia
IHCS	Similar	Car
IHCS	Similar	Nursery
IHCS	Dissimilar	Alarm
IHCS	Dissimilar	Asia
IHCS	Dissimilar	Car
IHCS	Dissimilar	Nursery
ST	Randômica	Alarm
ST	Randômica	Asia
ST	Randômica	Car
ST	Randômica	Nursery
ST	Similar	Alarm
ST	Similar	Asia
ST	Similar	Car
ST	Similar	Nursery
ST	Dissimilar	Alarm
ST	Dissimilar	Asia
ST	Dissimilar	Car
ST	Dissimilar	Nursery
HCS	Randômica	Alarm
HCS	Randômica	Asia
HCS	Randômica	Car
HCS	Randômica	Nursery
HCS	Similar	Alarm
HCS	Similar	Asia
HCS	Similar	Car
HCS	Similar	Nursery
HCS	Dissimilar	Alarm
HCS	Dissimilar	Asia
HCS	Dissimilar	Car
HCS	Dissimilar	Nursery

Figura B.1: Conjunto de ensaios do experimento para a **QP1**

Algoritmo	Base de Dados	Passo	Ordem de Instâncias	Rede Inicial
IHCS	Alarm	100	Randômica	Vazia
IHCS	Alarm	100	Randômica	Parcial
IHCS	Alarm	100	Similar	Vazia
IHCS	Alarm	100	Similar	Parcial
IHCS	Alarm	100	Dissimilar	Vazia
IHCS	Alarm	100	Dissimilar	Parcial
IHCS	Alarm	1000	Randômica	Vazia
IHCS	Alarm	1000	Randômica	Parcial
IHCS	Alarm	1000	Similar	Vazia
IHCS	Alarm	1000	Similar	Parcial
IHCS	Alarm	1000	Dissimilar	Vazia
IHCS	Alarm	1000	Dissimilar	Parcial
IHCS	Alarm	4000	Randômica	Vazia
IHCS	Alarm	4000	Randômica	Parcial
IHCS	Alarm	4000	Similar	Vazia
IHCS	Alarm	4000	Similar	Parcial
IHCS	Alarm	4000	Dissimilar	Vazia
IHCS	Alarm	4000	Dissimilar	Parcial
IHCS	Nursery	100	Randômica	Vazia
IHCS	Nursery	100	Randômica	Parcial
IHCS	Nursery	100	Similar	Vazia
IHCS	Nursery	100	Similar	Parcial
IHCS	Nursery	100	Dissimilar	Vazia
IHCS	Nursery	100	Dissimilar	Parcial
IHCS	Nursery	1000	Randômica	Vazia
IHCS	Nursery	1000	Randômica	Parcial
IHCS	Nursery	1000	Similar	Vazia
IHCS	Nursery	1000	Similar	Parcial
IHCS	Nursery	1000	Dissimilar	Vazia
IHCS	Nursery	1000	Dissimilar	Parcial
IHCS	Nursery	4000	Randômica	Vazia
IHCS	Nursery	4000	Randômica	Parcial
IHCS	Nursery	4000	Similar	Vazia
IHCS	Nursery	4000	Similar	Parcial
IHCS	Nursery	4000	Dissimilar	Vazia
IHCS	Nursery	4000	Dissimilar	Parcial
ST	Alarm	100	Randômica	Vazia
ST	Alarm	100	Randômica	Parcial
ST	Alarm	100	Similar	Vazia
ST	Alarm	100	Similar	Parcial
ST	Alarm	100	Dissimilar	Vazia
ST	Alarm	100	Dissimilar	Parcial
ST	Alarm	1000	Randômica	Vazia
ST	Alarm	1000	Randômica	Parcial
ST	Alarm	1000	Similar	Vazia
ST	Alarm	1000	Similar	Parcial
ST	Alarm	1000	Dissimilar	Vazia
ST	Alarm	1000	Dissimilar	Parcial
ST	Alarm	4000	Randômica	Vazia
ST	Alarm	4000	Randômica	Parcial
ST	Alarm	4000	Similar	Vazia
ST	Alarm	4000	Similar	Parcial
ST	Alarm	4000	Dissimilar	Vazia
ST	Alarm	4000	Dissimilar	Parcial
ST	Nursery	100	Randômica	Vazia
ST	Nursery	100	Randômica	Parcial
ST	Nursery	100	Similar	Vazia
ST	Nursery	100	Similar	Parcial
ST	Nursery	100	Dissimilar	Vazia
ST	Nursery	100	Dissimilar	Parcial
ST	Nursery	1000	Randômica	Vazia
ST	Nursery	1000	Randômica	Parcial
ST	Nursery	1000	Similar	Vazia
ST	Nursery	1000	Similar	Parcial
ST	Nursery	1000	Dissimilar	Vazia
ST	Nursery	1000	Dissimilar	Parcial
ST	Nursery	4000	Randômica	Vazia
ST	Nursery	4000	Randômica	Parcial
ST	Nursery	4000	Similar	Vazia
ST	Nursery	4000	Similar	Parcial
ST	Nursery	4000	Dissimilar	Vazia
ST	Nursery	4000	Dissimilar	Parcial

Figura B.2: Conjunto de ensaios do experimento para a **QP2**

Conjunto de Dados	Tamanho de Passo	Ordem de Instâncias	Rede Inicial	Número Máximo de Pais	nRSS
Alarm	100	Similar	Vazia	1	2
Nursery	100	Similar	Vazia	-	2
Alarm	2000 ou 4000	Similar	Vazia	-	-
Nursery	2000 ou 4000	Similar	Vazia	1	-
Alarm	100	Dissimilar	Vazia	-	-
Nursery	100	Dissimilar	Vazia	1	-
Alarm	2000 ou 4000	Dissimilar	Vazia	1	2
Nursery	2000 ou 4000	Dissimilar	Vazia	-	2
Alarm	100	Similar	Parcial	1	-
Nursery	100	Similar	Parcial	-	-
Alarm	2000 ou 4000	Similar	Parcial	-	2
Nursery	2000 ou 4000	Similar	Parcial	1	2
Alarm	100	Dissimilar	Parcial	-	2
Nursery	100	Dissimilar	Parcial	1	2
Alarm	2000 ou 4000	Dissimilar	Parcial	1	-
Nursery	2000 ou 4000	Dissimilar	Parcial	-	-

Figura B.3: Conjunto de ensaios do primeiro experimento para a **QP3**

Conjunto de Dados	Tamanho de Passo	Ordem de Instâncias	Rede Inicial	Número Máximo de Pais	Alpha
Alarm	100	Similar	Vazia	1	.9
Nursery	100	Similar	Vazia	-	.9
Alarm	2000 ou 4000	Similar	Vazia	-	.99
Nursery	2000 ou 4000	Similar	Vazia	1	.99
Alarm	100	Dissimilar	Vazia	-	.99
Nursery	100	Dissimilar	Vazia	1	.99
Alarm	2000 ou 4000	Dissimilar	Vazia	1	.9
Nursery	2000 ou 4000	Dissimilar	Vazia	-	.9
Alarm	100	Similar	Parcial	1	.99
Nursery	100	Similar	Parcial	-	.99
Alarm	2000 ou 4000	Similar	Parcial	-	.9
Nursery	2000 ou 4000	Similar	Parcial	1	.9
Alarm	100	Dissimilar	Parcial	-	.9
Nursery	100	Dissimilar	Parcial	1	.9
Alarm	2000 ou 4000	Dissimilar	Parcial	1	.99
Nursery	2000 ou 4000	Dissimilar	Parcial	-	.99

Figura B.4: Conjunto de ensaios do segundo experimento para a **QP3**