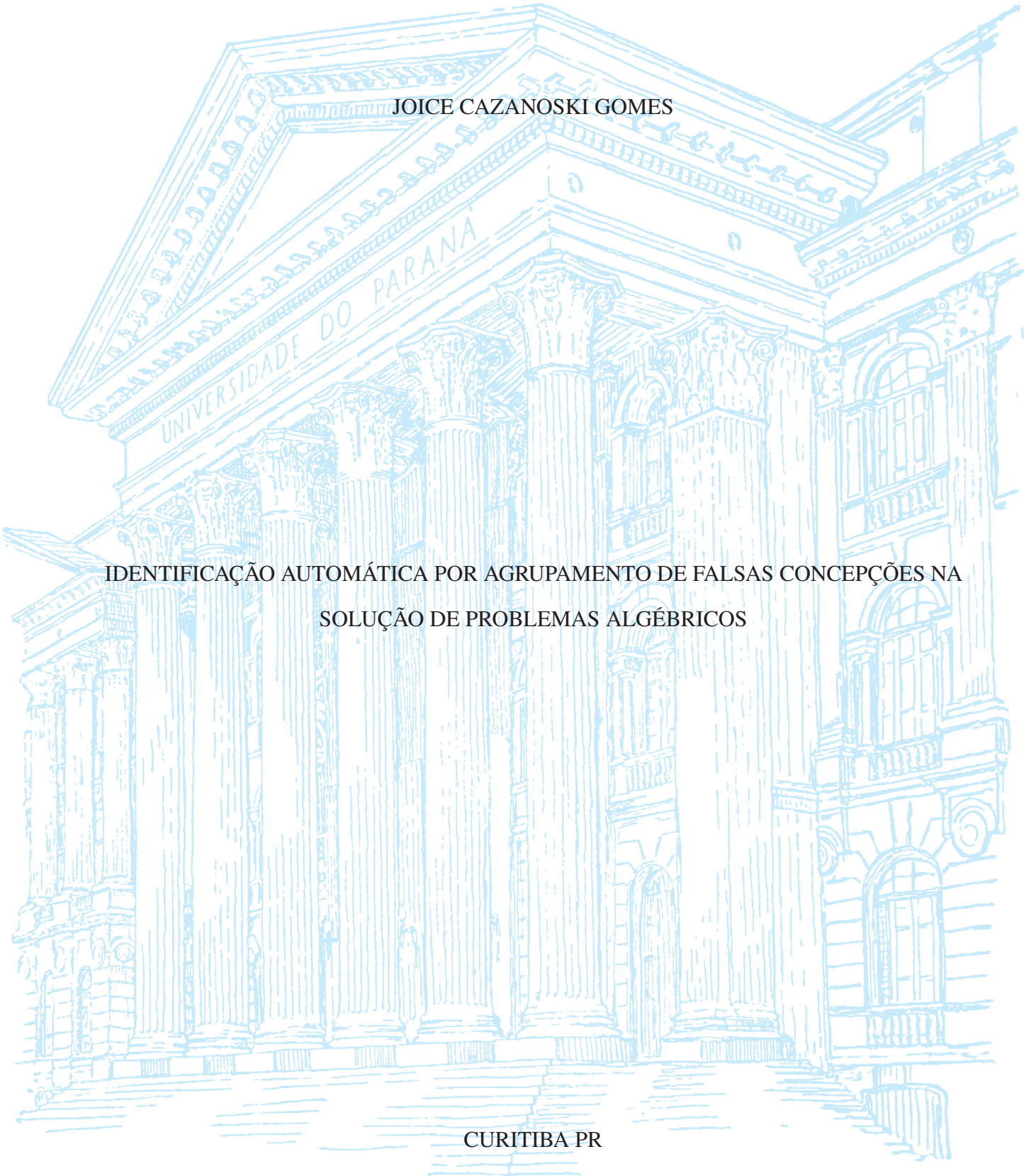
UNIVERSIDADE FEDERAL DO PARANÁ

JOICE CAZANOSKI GOMES




# IDENTIFICAÇÃO AUTOMÁTICA POR AGRUPAMENTO DE FALSAS CONCEPÇÕES NA SOLUÇÃO DE PROBLEMAS ALGÉBRICOS

CURITIBA PR

2024

JOICE CAZANOSKI GOMES

# IDENTIFICAÇÃO AUTOMÁTICA POR AGRUPAMENTO DE FALSAS CONCEPÇÕES NA SOLUÇÃO DE PROBLEMAS ALGÉBRICOS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Prof. Dr. Patricia A. Jaques.

CURITIBA PR

2024

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Gomes, Joice Cazanoski
Identificação automática por agrupamento de falsas concepções na solução de problemas algébricos / Joice Cazanoski Gomes. – Curitiba, 2024.
1 recurso on-line : PDF.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.

Orientador: Patricia Augustin Jaques Maillard

1. Aprendizado do computador. 2. Análise por agrupamento. 3. Sistemas tutoriais inteligentes. 4. Falsas concepções (Informática). I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Informática. III. Maillard, Patricia Augustin Jaques. IV. Título.

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

# TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da dissertação de Mestrado de **JOICE CAZANOSKI GOMES** intitulada: **IDENTIFICAÇÃO AUTOMÁTICA POR AGRUPAMENTO DE FALSAS CONCEPÇÕES NA SOLUÇÃO DE PROBLEMAS ALGÉBRICOS**, sob orientação da Profa. Dra. PATRICIA AUGUSTIN JAQUES MAILLARD, que após terem inquirido a aluna e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestra está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 13 de Dezembro de 2024.

Assinatura Eletrônica
16/12/2024 11:00:05.0
PATRICIA AUGUSTIN JAQUES MAILLARD
Presidente da Banca Examinadora

Assinatura Eletrônica
21/12/2024 20:01:43.0
CRISTIAN CECHINEL
Avaliador Externo (UNIVERSIDADE FEDERAL DE PELOTAS)

Assinatura Eletrônica
20/12/2024 12:15:28.0
ELAINE HARADA TEIXEIRA DE OLIVEIRA
Avaliador Externo (UNIVERSIDADE FEDERAL DO AMAZONAS)

Assinatura Eletrônica
16/12/2024 22:37:13.0
RACHEL CARLOS DUQUE REIS
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

*"Qualquer tecnologia suficientemente avançada é indistinguível da magia"*
*– Arthur C. Clarke.*

## AGRADECIMENTOS

Gostaria de expressar minha mais profunda gratidão àqueles que contribuíram para que este trabalho fosse realizado e para que eu chegasse a este momento tão especial em minha vida acadêmica e pessoal. À minha orientadora, Patricia, minha sincera gratidão pela paciência, sabedoria e por todo o conhecimento compartilhado. Mesmo diante de tantos desafios que passamos nesses anos de mestrado, sua presença constante e ajuda foram fundamentais. Sua orientação foi essencial para que este trabalho fosse concluído com qualidade. Agradeço também à UFPR, pela oportunidade de realizar este mestrado, e pelo acolhimento ao aceitar minha transferência de outra universidade. À minha família, em especial aos meus pais e minhas irmãs, que sempre acreditaram em mim, me apoiaram em todas as escolhas e me inspiraram a nunca desistir dos meus sonhos. Por fim, mas não menos importante, agradeço a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho. Cada contribuição foi fundamental para que este sonho se tornasse realidade.

A todos, o meu muito obrigada!

## RESUMO

Os erros são uma parte importante do processo de aprendizagem de matemática porque representam uma má compreensão de um conceito, uma falsa concepção ou *misconception*. Por esta razão, o diagnóstico de erros pode ajudar professores e ambientes inteligentes de aprendizagem a escolher o tipo de assistência mais apropriado para o aluno. Trabalhos anteriores utilizaram abordagens como sistemas especializados baseados em regras *(bug library)* e algoritmos de agrupamento (*clustering*) para identificar conceitos errados. A biblioteca de bugs exige um trabalho intensivo dos desenvolvedores para identificar previamente todos os possíveis equívocos e regras de codificação para cada um deles. Além disso, estas soluções não são capazes de detectar *misconceptions* para os quais as regras não foram explicitamente codificadas no sistema. As soluções baseadas em agrupamento superam estes dois inconvenientes aprendendo automaticamente a identificar *misconceptions* a partir dos erros mais comuns cometidos pelos estudantes. Para a correta e eficiente identificação de *misconceptions*, as soluções de agrupamento devem escolher uma representação adequada do problema e seus passos e utilizar algoritmos de agrupamento capazes de identificar padrões de acordo com a solução escolhida. Este trabalho propõe a comparação de diferentes algoritmos de clustering, incluindo Breathing K-Means (BKMeans) e Density-Based Spatial Clustering of Applications with Noise (DBSCAN) como novas abordagens, com o K-Modes utilizado na solução proposta em Gomes e Jaques (2020). Esta comparação visa avaliar se as novas soluções representam um avanço em relação ao método anterior. Utilizando também árvores de expressão para representar etapas da solução do problema algébrico, este estudo avalia a eficácia e eficiência dessa abordagem na identificação de *misconceptions* no processo e resolução de equações de primeiro grau. Os dados foram coletados de um sistema tutor inteligente para auxiliar os alunos a resolver equações de primeiro grau e compreendem 1.319 registros de passos incorretos de alunos nesse tutor. Para os testes, utilizamos árvores de expressão para representar as equações em formato padronizado e vetorizado, os algoritmos de agrupamento identificaram 51 clusters com DBSCAN, 93 com K-Modes e 94 com BKMeans. Embora o DBSCAN tenha gerado menos clusters, apresentando uma maior taxa de ruído, K-Modes e BKMeans foram mais eficazes na formação de grupos bem definidos. Em relação à avaliação da qualidade dos clusters, o Silhouette Score médio foi de 0,94 para o DBSCAN, e de 0,96 para os algoritmos K-Modes e BKMeans, indicando que, embora o DBSCAN consegue detectar clusters com boa coesão, os demais algoritmos apresentaram resultados ligeiramente superiores, com melhor separação e menor dispersão. Além da análise quantitativa, este trabalho também estabelece conexões entre os clusters identificados e as categorias teóricas de *misconceptions* descritas na literatura, demonstrando que os padrões descobertos pelos algoritmos correspondem a *misconceptions* matemáticas reconhecidas, como erros de vocabulário, erros computacionais e convicções errôneas.



Palavras-chave: Falsas concepções. Clusterização. Algoritmos de Agrupamento, Aprendizado de Máquina. Aprendizagem de Matemática. Sistema Tutor Inteligente.

## ABSTRACT

Errors play a crucial role in mathematics learning, often reflecting a poor understanding of a concept or a misconception. Therefore, diagnosing errors can help educators and intelligent learning environments choose the most appropriate assistance for students. Previous work has used rule-based expert systems (bug libraries) and clustering algorithms to identify incorrect concepts. Bug libraries require extensive developer effort to pre-identify all possible misconceptions and coding rules for each one. Furthermore, these solutions are unable to detect misconceptions for which rules have not been explicitly coded into the system. Clustering-based solutions overcome these two limitations by automatically learning to identify misconceptions from the most common errors made by students. To correctly and efficiently identify misconceptions, clustering solutions must choose an appropriate representation of the problem and its steps and use clustering algorithms capable of detecting patterns based on the chosen solution. This study proposes a comparison of different clustering algorithms, including Breathing K-Means (BKMeans) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) as novel approaches, alongside K-Modes, which was used in the solution proposed by Gomes and Jaques (2020). The objective of this comparison is to assess whether these new approaches represent an improvement over the previous method. Using expression trees to represent steps in solving algebraic problems, this study evaluates the effectiveness and efficiency of this approach in identifying misconceptions in the process of solving first-degree equations. The data was collected from an intelligent tutoring system designed to assist students in solving first-degree equations and consists of 1,319 records of incorrect student steps in this tutor. For the tests, expression trees were used to represent the equations in a standardized and vectorized format. The clustering algorithms identified 51 clusters with DBSCAN, 93 with K-Modes, and 94 with BKMeans. While DBSCAN generated fewer clusters with higher noise, K-Modes, and BKMeans were more effective in forming well-defined groups. In terms of cluster quality evaluation, the average Silhouette Score was 0.94 for DBSCAN, and 0.96 for both K-Modes and BKMeans, indicating that while DBSCAN was able to detect clusters with good cohesion, the other algorithms produced slightly superior results, with better separation and less dispersion. In addition to the quantitative analysis, this study also establishes connections between the identified clusters and the theoretical categories of misconceptions described in the literature. The findings demonstrate that the patterns detected by the algorithms correspond to well-documented mathematical misconceptions, such as vocabulary errors, computational errors, and erroneous beliefs.



Keywords: Misconception. Clustering. Machine Learning. Mathematic Learning. Intelligent Tutoring System.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE ACRÔNIMOS

| | |
|---|---|
| DINF | Departamento de Informática |
| PPGINF | Programa de Pós-Graduação em Informática |
| STI | Sistemas Tutores Inteligenetes |
| UFPR | Universidade Federal do Paraná |

**SUMÁRIO**

# 1 INTRODUÇÃO

No cenário educacional brasileiro, os dados do Sistema de Avaliação da Educação Básica (SAEB, 2017) revelam um quadro preocupante: mais de 63% dos alunos do 9º ano apresentam conhecimento insuficiente de matemática. No ensino médio, a situação é ainda mais alarmante, com quase 72% dos alunos demonstrando conhecimento considerado baixo, sendo que 22,5% encontram-se no nível mais baixo de proficiência. Estas estatísticas refletem desafios profundos no ensino e aprendizagem da matemática em nosso país, desafios estes que demandam novas abordagens e soluções inovadoras.

Estudos na área de educação matemática indicam que as falsas concepções (ou *misconceptions*, em inglês) têm um impacto significativo na aprendizagem (Schmidt, 1997), podendo explicar, em parte, os números preocupantes apresentados. Estas falsas concepções representam compreensões incorretas de conceitos fundamentais que, quando não identificadas e corrigidas a tempo, comprometem todo o desenvolvimento subsequente do aprendizado matemático. Dessa forma, ao identificar e corrigir precocemente, é possível proporcionar uma base sólida para o aprendizado subsequente, por meio de intervenções específicas de professores e tutores computacionais (Fuchs et al., 2021).

A identificação de falsas concepções matemáticas durante o processo de aprendizagem é um desafio significativo tanto para educadores quanto para sistemas computacionais de apoio ao ensino. Este desafio se intensifica na área de álgebra, onde a compreensão abstrata de conceitos é essencial para o progresso do estudante em matemática avançada.

Os Sistemas Tutores Inteligentes (STIs) representam uma abordagem inovadora no auxílio ao processo de ensino e aprendizagem, visando fornecer feedback personalizado e adaptativo aos estudantes. No entanto, a arquitetura atual desses sistemas frequentemente se depara com desafios significativos, particularmente no que diz respeito à identificação e tratamento de erros de aprendizagem. A maioria dos STIs ainda utiliza bibliotecas de erros pré-programadas (*bug libraries*). Esse método tradicional requer a codificação manual de regras para cada possível erro, demandando um esforço considerável de especialistas e introduz uma rigidez que compromete a capacidade do sistema de compreender e responder a novas falsas concepções.

Embora existam trabalhos envolvendo ambientes inteligentes de aprendizagem que buscam identificar falsas concepções na matemática, a identificação totalmente automática ainda é considerada um desafio (Yasin, 2017). Uma revisão sistemática da literatura revela que a maioria dos estudos nesta área está limitada a aplicação em domínios específicos da matemática (Fatio et al., 2020; Jabal e Rosjanuardi, 2019; Küçük et al., 2011; Untarti e Kusuma, 2019), sem abordar de forma abrangente o problema da identificação automática e contínua de *misconceptions* através de métodos de aprendizado de máquina. Além disso, há uma escassez de pesquisas que abordem a representação adequada de expressões matemáticas, particularmente em estruturas complexas como equações algébricas.

Nesse contexto, este trabalho se baseia nas contribuições iniciais de Gomes e Jaques (2020), que propuseram uma abordagem baseada em agrupamento para identificação automática de falsas concepções em álgebra. Aquela pesquisa utilizou normalização de equações (substituindo valores numéricos por variáveis padronizadas) e aplicou o algoritmo de *clustering* K-modes para agrupar padrões de erros semelhantes. Embora promissora, a abordagem anterior apresentou limitações importantes, como dificuldades no tratamento de estruturas algébricas complexas (especialmente no tratamento de parênteses e frações), presença de ruído nos clusters identificados

e dependência de um único algoritmo de agrupamento. O presente trabalho avança essa linha de pesquisa ao introduzir um método de representação mais robusto baseado em árvores de expressão e ao comparar o desempenho de múltiplos algoritmos de clustering (DBSCAN, K-modes e BKMeans) na identificação de *misconceptions* na resolução de equações algébricas.

Diante deste cenário, o problema central desta pesquisa é como identificar automaticamente, sem intervenção humana, padrões de falsas concepções algébricas a partir dos passos de resolução de problemas realizados por estudantes em sistemas tutores inteligentes?

A partir desse problema, este trabalho tem como objetivo desenvolver e avaliar métodos automatizados para a identificação de falsas concepções (*misconceptions*) e erros recorrentes cometidos por estudantes durante o processo de aprendizagem de álgebra, utilizando técnicas de aprendizado de máquina para identificar padrões em dados coletados de um sistema tutor inteligente.

Para alcançar o objetivo, este trabalho adota uma abordagem metodológica estruturada em seis etapas principais:

1. **Coleta de dados:** Utilização de registros de log do Sistema Tutor Inteligente PAT2Math (Personal Affective Tutor to Math), coletados de estudantes do sétimo ano de escolas da grande Porto Alegre. Foram recuperados especificamente os registros que contêm passos incorretos de resolução de equações, totalizando 1.319 entradas.

2. **Pré-processamento e representação:** As equações foram padronizadas substituindo valores numéricos por símbolos padronizados e, em seguida, convertidas para o formato de árvores de expressão, utilizando notação pós-fixa. Esta representação permite capturar a estrutura hierárquica das operações matemáticas de forma não ambígua.

3. **Extração e vetorização de características:** A partir das árvores de expressão, foram extraídas características relevantes como número total de nós, contagem de operadores, contagem de variáveis e profundidade da árvore. Estas características foram vetorizadas para criar uma representação numérica adequada para os algoritmos de *clustering*.

4. **Aplicação de algoritmos de clustering:** Foram implementados e comparados três algoritmos distintos: DBSCAN (Density-Based Spatial Clustering of Applications with Noise), K-Modes e BKMeans (Breathing K-Means). Cada algoritmo foi configurado com parâmetros específicos determinados empiricamente para otimizar o desempenho.

5. **Avaliação quantitativa:** A qualidade dos agrupamentos foi avaliada utilizando o *Silhouette Score*, que mede a coesão interna e a separação entre clusters. Esta métrica permite comparar objetivamente o desempenho dos diferentes algoritmos.

6. **Análise qualitativa:** Os clusters gerados foram analisados para identificar padrões significativos que representam falsas concepções. Estes padrões foram relacionados com as categorias teóricas estabelecidas na literatura, particularmente as classificações propostas por Holmes et al. (2013).

Com essa metodologia, esse trabalho visa contribuir para o avanço do estado da arte na identificação automática de falsas concepções matemáticas através de uma nova abordagem baseada em árvores de expressão para representar equações algébricas no contexto de análise automatizada, criando um método que não depende de bibliotecas de erros pré-programadas, permitindo a identificação de padrões não previstos previamente;

No contexto educacional, as contribuições se refletem em auxiliar professores na identificação precoce de falsas concepções, permitindo intervenções pedagógicas direcionadas a

partir da possibilidade de sistemas tutores mais adaptativos que podem personalizar o feedback com base nos padrões específicos de erro identificados.

Esta dissertação está organizada da seguinte forma: No Capítulo 2, apresentamos os fundamentos teóricos que embasam este trabalho, incluindo conceitos de Sistemas Tutores Inteligentes, *misconceptions* na matemática e árvores de expressão. O Capítulo 3 discute os trabalhos relacionados, contextualizando nossa pesquisa no panorama mais amplo dos estudos sobre detecção de *misconceptions*. No Capítulo 4, detalhamos a metodologia proposta, descrevendo os processos de coleta de dados, pré-processamento, representação por árvores de expressão e aplicação dos algoritmos de clustering. O Capítulo 5 apresenta os resultados obtidos com cada algoritmo, seguidos de uma análise comparativa e discussão da relação entre os clusters identificados e as categorias teóricas de *misconceptions*. Por fim, o Capítulo 6 traz as conclusões do trabalho e sugestões para trabalhos futuros.

# 2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo estabelece os fundamentos teóricos que alicerçam esta pesquisa, apresentando conceitos essenciais para a compreensão do trabalho desenvolvido. Inicialmente, abordamos os Sistemas Tutores Inteligentes na Seção 2.1, explorando sua definição, classificação e funcionamento, com ênfase especial no PAT2Math, sistema utilizado como base para este estudo. Em seguida, aprofundamos a discussão sobre *misconceptions* na matemática na Seção 2.2, distinguindo-as de erros simples e apresentando taxonomias relevantes, como a proposta por Holmes e colegas (2013), que categorizam as falsas concepções em *misconceptions* de vocabulário, erros computacionais e convicções errôneas. O capítulo também introduz as árvores de expressão na Seção 2.3 como estruturas de dados adequadas para representar equações matemáticas, destacando sua importância para a análise computacional dos padrões algébricos. Por fim, exploramos técnicas de clustering na Seção 2.4, incluindo Breathing KMeans, K-Modes e DBSCAN, fundamentais para a metodologia de identificação automática de *misconceptions* desenvolvida neste trabalho. Esta fundamentação teórica estabelece as bases conceituais necessárias para compreender a abordagem metodológica proposta nos capítulos subsequentes.

## 2.1 SISTEMAS TUTORES INTELIGENTES

Sistemas tutores inteligentes (STIs) são sistemas computacionais de ensino que utilizam técnicas de inteligência artificial (IA) para oferecer instrução personalizada e adaptativa aos estudantes (Woolf, 2009). Esses sistemas são projetados para simular a interação entre um aluno e um tutor humano, proporcionando suporte e orientação durante o processo de aprendizagem (Anderson et al., 1995). Os STIs conseguem identificar as necessidades individuais dos estudantes, adaptando-se ao seu ritmo e estilo de aprendizagem, a fim de melhorar a eficácia do ensino (Corbett et al., 2001). Além disso, esses sistemas fornecem *feedback* imediato e contextualizado, auxiliando os alunos a superar dificuldades e aprimorar suas habilidades (Aleven et al., 2011).

Sistemas tutores inteligentes foram desenvolvidos para diversos domínios, como matemática, ciências, linguagem e programação. Uma classificação possível é conforme o tipo de interação que eles desenvolvem com os alunos, que pode ser baseado em conversação, baseado em simulação, baseado em tarefas e baseado em aprendizado colaborativo (Duffy e Roehler, 1995).

Os sistemas baseados em conversação focam na interação por meio de diálogos com os alunos, utilizando recursos de processamento de linguagem natural para orientar e esclarecer dúvidas. Um exemplo é o AutoTutor, que auxilia os estudantes em diversos tópicos através de conversas em linguagem natural (Graesser et al., 2005; Nye et al., 2014).

Os sistemas baseados em simulação proporcionam um ambiente virtual no qual os alunos podem explorar e aprender por meio da experimentação. Esses sistemas são geralmente utilizados para ensinar habilidades práticas ou conceitos complexos. Um exemplo é o SimCityEDU, que ajuda os alunos a entenderem conceitos relacionados ao planejamento urbano e à sustentabilidade (Ventura et al., 2014).

Os sistemas baseados em aprendizado colaborativo promovem a interação entre os alunos e facilitam a troca de ideias e a construção do conhecimento em conjunto. Esses sistemas podem utilizar técnicas de inteligência artificial para mediar discussões, propor atividades e incentivar a colaboração. Betty's Brain é um sistema tutor inteligente que ajuda os estudantes a aprenderem conceitos de ciências através da construção e ensino de um agente de software

chamado Betty (Leelawong e Biswas, 2008). Os alunos trabalham juntos para ensinar Betty, desenvolvendo e refinando modelos mentais de conceitos complexos, enquanto recebem feedback do sistema sobre a precisão de suas representações e a eficácia de seu ensino.

Os sistemas baseados em tarefas focam em auxiliar os alunos na realização de tarefas específicas, fornecendo orientação passo a passo e feedback personalizado. Estes sistemas são comuns no ensino de matemática e programação. Um exemplo é o Cognitive Tutor, que oferece suporte individualizado aos estudantes em matemática (Anderson et al., 1995; Ritter et al., 2007). Existem dois tipos de STIs baseados em tarefas: baseados em passos ou em respostas. No STI baseado em passos, o estudante insere todos os passos necessários para a resolução de um problema e recebe feedback para cada etapa. Já no STI baseado em resposta, o feedback é dado apenas para a solução final (Vanlehn, 2011).

Para acompanhar e fornecer suporte ao aluno durante o processo de resolução de um problema de um determinado domínio, um STI baseado em passos é dividido em dois laços: o externo (*Outer Loop*) e o interno (*Inner Loop*). O *outer loop* determina, com base no conhecimento inferido pelo tutor, qual deve ser a próxima tarefa que o aluno deverá solucionar, enquanto o *inner loop* auxilia o aluno na resolução passo a passo em uma tarefa para o aluno aprender. Esse laço é executado para cada etapa, dando feedback mínimo (certo ou errado) ao final de cada uma delas, até a resolução completa (Vanlehn, 2016).

### 2.1.1 Biblioteca de erros

As bibliotecas de erros (*bug libraries*) são um componente importante de sistemas tutores inteligentes que aumentam sua eficácia ao catalogar os erros comuns dos alunos. As bibliotecas podem identificar e corrigir erros comparando as respostas dos alunos com erros conhecidos e oferecendo feedback direcionado. O desenvolvimento dessas bibliotecas é uma tarefa árdua que envolve a geração manual de uma lista de todos os erros possíveis. No entanto, os avanços no aprendizado de máquina resultaram na criação de bibliotecas de erros dinâmicas, que podem ser geradas e atualizadas automaticamente com base nas interações dos alunos. Essa abordagem não só reduz a carga de trabalho dos professores, mas também aumenta significativamente a adaptabilidade e escalabilidade dos STI's, resultando em uma melhor aprendizagem dos alunos e em um ensino mais individualizado (Baffes, 1994).

### 2.1.2 PAT2Math

O STI utilizado neste trabalho é o PAT2Math (*Personal Affective Tutor to Math*), o qual é um tutor baseado em passos e emprega Inteligência Artificial para auxiliar estudantes no aprendizado e resolução de equações algébricas de primeiro grau, dando feedback e orientação a cada etapa do processo de resolução (Jaques et al., 2013). Sua arquitetura segue o modelo clássico de STIs, onde o laço externo do PAT2Math possui uma sequência de tarefas fixas, enquanto o laço interno auxilia na resolução passo a passo.

O modelo de domínio do PAT2Math é caracterizado como um sistema especialista baseado em regras que possibilita a resolução dos exercícios e a avaliação das respostas dos estudantes (Jaques et al., 2008). Este modelo contém o conhecimento necessário sobre as regras e operações algébricas para resolver equações de primeiro grau, permitindo que o sistema verifique se os passos executados pelo aluno estão corretos. No PAT2Math, o modelo cognitivo, sendo uma extensão do modelo de domínio, é responsável por avaliar e corrigir, se necessário, cada etapa enviada pelo aluno (Seffrin et al., 2012).

As equações para os alunos resolverem são organizadas em níveis de dificuldade. Cada nível contém planos com equações isomórficas que funcionam com as mesmas operações

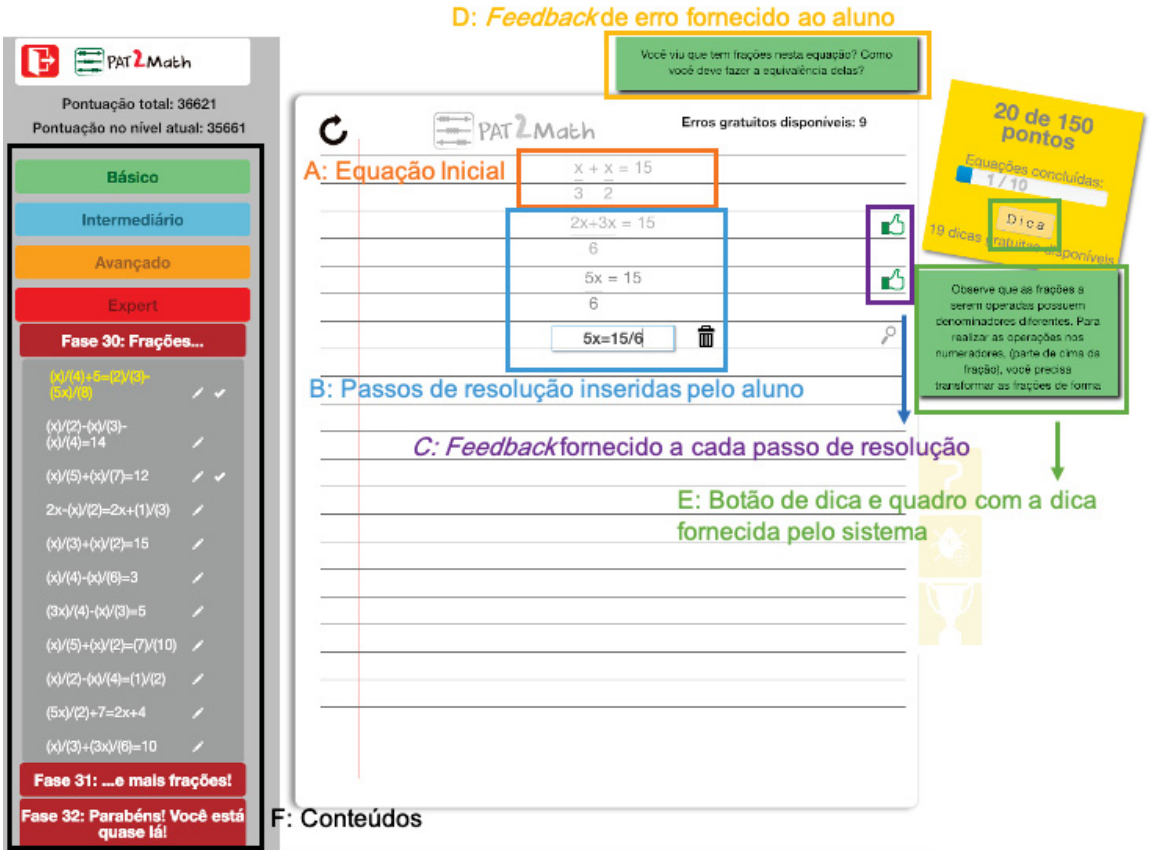



Figura 2.1: Interface do STI PAT2Math

algébricas. O aluno deve resolver um número mínimo de equações em um plano que mostre que ele ou ela domina as operações algébricas trabalhadas para acessar o próximo plano.

Na Figura 2.1, é apresentado o funcionamento do PAT2Math. A figura é dividida em seis quadros que ilustram as etapas de funcionamento do sistema.

O quadro A mostra a equação inicial fornecida pelo sistema ao aluno, que não pode ser alterada. A partir dessa equação, o aluno insere passos de resolução no quadro B, conforme julgar adequado para chegar à solução da equação. O sistema identifica os passos corretos e permite a inserção de um próximo passo, fornecendo um *feedback* mínimo após cada passo inserido pelo aluno.

No quadro C, é ilustrado um *feedback* positivo do sistema, indicando que o passo inserido pelo aluno está correto. Porém, caso o passo esteja incorreto, o sistema fornece um *error-feedback* juntamente com uma mensagem de ajuda, conforme ilustrado no quadro D.

Se o aluno precisar de ajuda para prosseguir na resolução de uma equação, pode solicitar uma dica ao sistema no botão de dicas. O botão de dicas e um exemplo de dica são exibidos no Quadro E.

Por fim, o quadro F mostra os conteúdos que já foram ou que ainda serão vistos pelo aluno. Nessa versão mais recente da interface *web* do PAT2Math, foram inseridos componentes de gamificação, como pontuação, *ranking* da turma, erros e dicas gratuitas, visando evitar comportamentos como *gaming the system* e *help refusal* (Azevedo et al., 2018).

#### *2.1.2.1 Integração do Modelo de Detecção Automática de Misconceptions*

O sistema atual do PAT2Math já possui um mecanismo de detecção e resposta a erros dos estudantes, porém baseado em uma biblioteca de erros (bug library) pré-programada. Esta abordagem apresenta limitações significativas, por requerer codificação manual de regras para cada possível erro e não consegue identificar *misconceptions* para as quais não foram explicitamente criadas regras.

O modelo de detecção automática de *misconceptions* desenvolvido neste trabalho pode ser integrado ao PAT2Math como um componente adicional que opera paralelamente ao modelo pedagógico. Esta integração ocorrerá em duas etapas principais:

A solução proposta consiste na integração de um novo componente de detecção automática de *misconceptions*, projetado para operar paralelamente ao modelo pedagógico existente. O sistema processaria em tempo real os registros de passos incorretos dos estudantes.

Uma próxima etapa consiste em transformar a detecção de *misconceptions* em ações pedagógicas. Para os estudantes, isso significaria receber um feedback personalizado. A proposta representa uma transformação significativa do PAT2Math, pretendendo convertê-lo de um sistema baseado em regras estáticas para um ambiente de aprendizagem adaptativo. O diferencial está na capacidade de identificar *misconceptions* em tempo real. Esta integração constitui uma proposta teórica e ainda não foi implementada, representando um potencial caminho de evolução para o sistema de tutoria matemática.

## 2.2 MISCONCEPTIONS NA MATEMÁTICA

Para fundamentar adequadamente este estudo, é essencial estabelecer uma distinção clara entre “erros” e “falsas concepções” (misconceptions) no contexto da aprendizagem matemática. Estas distinções são fundamentais não apenas para a compreensão teórica do problema, mas também para o desenvolvimento de abordagens eficazes de identificação e intervenção.

**Erros** são desvios pontuais do procedimento correto, que podem ocorrer por diversas razões circunstanciais como desatenção, cansaço, interpretação equivocada de uma notação específica, ou falhas na execução de um algoritmo que o estudante compreende conceitualmente. Os erros são geralmente isolados, inconsistentes e podem ser facilmente corrigidos quando apontados ao estudante.

Por exemplo, quando um estudante calcula erroneamente $3 \times 4 = 13$, mas ao ser questionado percebe imediatamente o equívoco, isso caracteriza um erro simples, não uma *misconception*. Como destacado por Abimbola (1988), os erros podem ser identificados e corrigidos pelo próprio aluno e não necessariamente indicam uma falta de conhecimento conceitual.

**Misconceptions** (falsas concepções), por outro lado, são compreensões estruturalmente incorretas de um conceito matemático fundamental. Elas representam modelos mentais equivocados que o estudante desenvolve e utiliza consistentemente. Schmidt (1997) define *misconception* como um erro decorrente de uma compreensão insuficiente ou incorreta de um conceito essencial para a resolução de uma tarefa.

As *misconceptions* são caracterizadas pela sua persistência e sistematicidade. Quando um estudante comete repetidamente o mesmo tipo de erro em diferentes contextos, aplicando consistentemente um raciocínio incorreto, provavelmente estamos diante de uma *misconception*.

Por exemplo, se um estudante resolve equações do tipo $x + a = b$ como $x = b + a$ (sem inverter a operação), isso indica uma *misconception* sobre o conceito fundamental de equações algébricas e operações inversas. Mesmo quando alertado sobre o erro, o estudante tenderá a repeti-lo, pois o problema está na sua compreensão conceitual.

Sison e Shimura (1998) acrescentam que uma *misconception* também pode ser originada por traços comportamentais, como tédio ou distração, que impedem uma compreensão completa do conteúdo apresentado. Na matemática, a compreensão adequada dos conceitos é fundamental para a resolução de exercícios, tornando as *misconceptions* muito comuns e, às vezes, inevitáveis no processo de aprendizagem (MEC, 1997).

Além disso, com base nas revisões sistemáticas citadas em The Learning Agency Lab (2023), apresentamos algumas das *misconceptions* mais comuns na matemática. Um exemplo é a crença de que a multiplicação sempre resulta em um número maior, onde um aluno acredita que ao multiplicar 0,5 por 0,5, o resultado será maior, quando, na verdade $0,5 \times 0,5 = 0,25$. Outro exemplo é a ideia de que a divisão sempre resulta em um número menor, por exemplo $1 \div 0,5 = 2$. Outro erro frequente é a interpretação errônea de frações como números inteiros, onde um aluno vê a fração $\frac{3}{4}$ e acredita que 3 e 4 são números separados em vez de uma única quantidade representando três partes de um total de quatro, o que pode levar a somar ou subtrair as frações incorretamente, como pensar que $\frac{3}{4} + \frac{1}{4}$ é igual a $\frac{4}{4}$, o que é correto, mas pode ser confundido como 4 em vez de 1. Finalmente, há a crença de que a adição de zeros à direita de um número sempre aumenta seu valor, onde um aluno acredita que adicionar zeros após a vírgula em 3,2 resulta em um número maior, como 3,200, quando, na verdade, 3,2 e 3,200 são equivalentes. Estas *misconceptions* destacadas são baseadas em revisões sistemáticas de estudos anteriores, que ajudam a entender os desafios que os alunos enfrentam na aprendizagem matemática e fornecem percepções sobre como abordar esses problemas de maneira eficaz. A Tabela 2.1 lista as *misconceptions* elencadas por The Learning Agency Lab (2023) que estão especificamente relacionadas à resolução de equações de primeiro grau.

É possível observar que, na matemática, os *misconceptions* estão associados à estrutura das equações e não aos valores numéricos utilizados (Elmadani et al., 2012). Por exemplo, as equações $(x + 3 = 6)$ e $(x + 2 = 4)$ têm a mesma estrutura e podem ser representadas por $(x + a = b)$. Dessa forma, se aplicarmos essa estrutura padronizada no exemplo acima, ainda é possível observar e entender qual foi o erro cometido $(x + a = b \mid x = b + a)$.

A correção de *misconceptions*, em contraste com a simples correção de erros isolados, é importante porque atinge mais alunos e melhora a compreensão do assunto. Holmes et al. (2013) propõem que tal abordagem não apenas facilita uma assimilação mais robusta dos conceitos matemáticos, mas também fomenta o desenvolvimento do pensamento crítico e estimula uma reflexão ponderada sobre a essência da disciplina matemática. Além disso, o artigo aponta que, ao identificar os principais *misconceptions* ao invés de focar em erros individuais do aluno, os professores conseguem tomar decisões mais informadas sobre o que ensinar e quando intervir individualmente.

### 2.2.1 Categorias de *misconceptions* segundo Holmes

Holmes et al. (2013) propõem uma taxonomia abrangente que classifica as *misconceptions* matemáticas em três categorias principais. Esta classificação fornece um framework teórico contra o qual podemos avaliar os padrões identificados algoritmicamente.

#### *2.2.1.1 Misconceptions de vocabulário*

Esta categoria refere-se a interpretações incorretas da terminologia matemática e à compreensão deficiente do significado preciso de termos ou conceitos específicos. Estas *misconceptions* ocorrem quando o estudante atribui significados imprecisos ou incorretos a expressões, símbolos ou notações matemáticas.

Exemplos comuns incluem:

| Tópico | Concepção errônea | Exemplo |
|---|---|---|
| Equações e desigualdades | Dificuldade com a sintaxe da notação algébrica | Escrever $2x+3 = 5$ como $2x = 5 + 3$ |
| | Acreditar que o sinal de igual significa "a resposta é"em vez de expressar uma relação | Interpretar $3x+4 = 7$ como "a resposta é 7", sem realizar a operação correta |
| | Erro na ordem de inversão (erro de reversão) | Escrever $x - 5 = 10$ como $5 - x = 10$ |
| | Não verificar a solução ou cometer erros ao verificar a solução | Resolver $2x = 8$ para $x = 4$ e não substituir $x$ na equação original para verificar a solução |
| | Dificuldade em entender o conceito de combinar ou não combinar termos semelhantes | Combinar $3x + 2x$ corretamente como $5x$, mas combinar $3x+4$ incorretamente como $7x$ |
| | Acreditar que as propriedades comutativa e associativa são verdadeiras para subtração e divisão | Pensar que $a - b = b - a$ ou $a/b = b/a$ |
| Expressões e operações algébricas | Com expressões que envolvem subtração, os estudantes escrevem incorretamente a expressão, como por exemplo, $4 - n$ em vez de $n - 4$ | Representar "4 subtraído de um número"como $4 - n$ em vez de $n - 4$ |
| | Visualizar variáveis como rótulos ou unidades, ou acreditar que o valor de uma variável está relacionado à sua posição no alfabeto | Acreditar que $x$ tem sempre um valor maior que $y$ porque "x"aparece antes de "y"no alfabeto |
| | Não entender que as variáveis podem representar quantidades variáveis | Pensar que, se $x = 2$ em uma equação, então $x$ sempre será 2 em todas as equações |
| | Aplicação incorreta da operação inversa (adicionado pela autora) | Resolver $4x = 8$ como $x = 8 - 4$ |

Tabela 2.1: Concepções errôneas na resolução passo a passo de equações de primeiro grau, com exemplos.
Fonte: The Learning Agency Lab (2023)

- Interpretar o sinal de igualdade (=) como “o resultado é” em vez de uma relação de equivalência entre expressões
- Confundir o significado de operadores matemáticos, como interpretar a subtração como simples remoção sem considerar a ordem
- Mal-entendidos sobre o significado de variáveis, confundindo-as com unidades ou valores fixos

#### *2.2.1.2 Erros computacionais*

Esta categoria refere-se a falhas sistemáticas na aplicação de procedimentos ou algoritmos matemáticos. Diferentemente de simples enganos aritméticos (que seriam categorizados como erros), os erros computacionais representam padrões consistentes de aplicação incorreta de procedimentos matemáticos devido a uma compreensão inadequada de sua fundamentação.

Exemplos incluem:

- Aplicação consistentemente incorreta das regras de sinais na álgebra (como sempre converter um sinal negativo para positivo ao mudar de lado em uma equação)
- Padrões recorrentes de omissão de etapas essenciais em algoritmos matemáticos
- Aplicação incorreta da ordem das operações

É importante notar que a categoria "Erros Computacionais" na taxonomia de Holmes não se refere a simples enganos de cálculo (como $7 + 8 = 14$), mas sim a padrões sistemáticos que refletem uma incompreensão procedural.

#### *2.2.1.3 Convicções errôneas*

Representando falhas mais profundas na compreensão matemática, as convicções errôneas são crenças conceituais falsas que os estudantes desenvolvem sobre a natureza dos objetos e operações matemáticas. Estas *misconceptions* são frequentemente as mais difíceis de corrigir por estarem enraizadas de forma fundamentalmente incorreta.

Exemplos significativos incluem:

- A crença de que a multiplicação sempre resulta em um número maior (falha ao compreender a multiplicação por frações ou números negativos)
- A noção de que a divisão sempre produz um resultado menor que o dividendo (sem considerar casos como divisão por frações menores que 1)
- A ideia de que equações algébricas só podem ter uma solução

## 2.3 ÁRVORES DE EXPRESSÃO

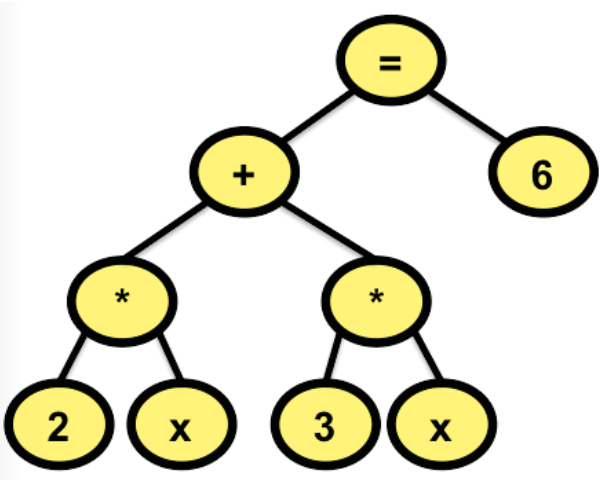
Uma árvore de expressão é uma estrutura de dados em forma de árvore binária, utilizada para representar expressões matemáticas. Uma árvore binária é uma árvore em que cada nó pode ter no máximo dois filhos, chamados de filho esquerdo e filho direito (Knuth, 1998). Na árvore de expressão, cada nó representa um operador, enquanto as folhas representam os operandos, ou seja, as constantes e variáveis da expressão. Na Figura 2.2, podemos observar um exemplo da representação da equação $2x + 3x = 6$ em uma árvore de expressão.

Usar uma árvore de expressão para representar uma expressão matemática apresenta diversas vantagens. Primeiramente, permite uma avaliação fácil da expressão ao percorrer a árvore e realizar as operações em uma ordem natural. Além disso, permite uma modificação fácil da equação, como mudar a ordem das operações ou adicionar, ou remover termos, e possibilita a simplificação da expressão aplicando identidades e regras matemáticas.

Adicionalmente, usar árvores de expressão para representar expressões matemáticas esclarece a ordem das operações (Lample e Charton, 2019). Consequentemente, considera a precedência e associatividade e elimina a necessidade de parênteses. Pesquisas anteriores indicaram que a representação de expressões matemáticas na resolução de problemas pode influenciar os resultados de algoritmos de aprendizagem de máquina (Gomes e Jaques, 2020).

Como mencionado na Seção 2.2, em matemática, um *misconception* está associado à estrutura da equação e não aos valores numéricos. Dessa forma, a utilização da árvore de expressão nos permite representar essas equações de maneira estruturada e sem ambiguidades em relação às operações (Lample e Charton, 2019).

Figura 2.2: Representação da equação $2x + 3x = 6$ em uma árvore de expressão




## 2.4 CLUSTERING

A clusterização, também conhecida como análise de agrupamento, é uma técnica fundamental na área de aprendizado de máquina e mineração de dados. Trata-se de um processo de agrupamento de objetos ou dados similares em clusters, com base em suas características ou propriedades comuns. Essa técnica visa principalmente encontrar estruturas ou padrões intrínsecos nos dados, sem a necessidade de rótulos ou categorias predefinidas. Ela permite identificar grupos naturalmente formados nos dados, o que pode levar a uma melhor compreensão dos fenômenos subjacentes e auxiliar na tomada de decisões (Aggarwal e Reddy, 2014).

Existem várias abordagens para realizar a clusterização, e a escolha do método depende das características dos dados e dos objetivos do problema em questão. Nesta seção, discutiremos brevemente alguns dos princípios e abordagens utilizados nesse trabalho.

### 2.4.1 Breathing KMeans

O algoritmo *Breathing KMeans* (BKMeans) é uma versão aprimorada do tradicional K-Means, projetado para resolver as limitações relacionadas à inicialização dos centroides e à convergência em dados complexos. O BKMeans introduz uma estratégia de "respiração", onde novos centroides são adicionados em áreas de alto erro (etapa de inspiração) e centroides de baixa utilidade são removidos (etapa de expiração), permitindo uma adaptação dinâmica e não local dos centroides. Diferentemente do algoritmo tradicional, que depende fortemente da inicialização dos centroides e pode exigir múltiplas execuções para alcançar resultados precisos, o BKMeans executa essas etapas iterativamente para refinar e melhorar a qualidade dos clusters de maneira mais eficiente. Além disso, mantém a simplicidade do K-Means, utilizando a inicialização e ajustando a profundidade de respiração para equilibrar a qualidade da solução e o tempo de computação. Como resultado, não só produz melhores resultados de clusterização, mas também reduz significativamente o tempo de execução, tornando-se uma alternativa melhor para tarefas de clusterização complexas (Fritzke, 2020).

### 2.4.2 K-Modes

O K-Modes é um algoritmo de clusterização especificamente projetado para agrupar dados categóricos, sendo uma extensão do algoritmo K-Means tradicional. No K-Modes, a distância entre os pontos é calculada com base na dissimilaridade dos atributos categóricos,

utilizando uma medida de distância baseada em modos, ao invés da distância euclidiana usada no K-Means. Esse algoritmo visa particionar os dados em K clusters, onde cada cluster é representado por um modo, sendo o valor mais frequente para cada atributo no cluster. O processo de clusterização envolve a seleção inicial de centroides, a atribuição de pontos de dados ao cluster mais similar e a atualização iterativa dos modos até a convergência. Essa abordagem é particularmente útil para lidar com conjuntos de dados que contêm variáveis categóricas. De acordo com Huang, o K-Modes mantém a eficiência do K-Means enquanto adapta o algoritmo para lidar com a natureza discreta e não ordenada dos dados categóricos, garantindo resultados localmente ótimos e uma aplicação mais ampla em diversos contextos práticos (Huang, 1997; Wang, 2009).

### 2.4.3 Clusterização Baseada em Densidade

O DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) é um algoritmo de clusterização baseado em densidade que consegue descobrir clusters de forma arbitrária em um conjunto de dados. Essa abordagem é especialmente útil quando os clusters têm formas irregulares ou quando a quantidade de clusters é desconhecida. A principal vantagem do DBSCAN é que ele não requer que o número de clusters seja especificado previamente.

O funcionamento do DBSCAN envolve dois parâmetros principais: o raio (epsilon), que define a vizinhança de um ponto de dados, e o número mínimo de pontos (MinPts) que um ponto deve ter em sua vizinhança para ser considerado um ponto central de um cluster. O algoritmo DBSCAN opera da seguinte maneira: um ponto é considerado um ponto central se possui pelo menos MinPts em sua vizinhança. Para cada ponto central, o DBSCAN tenta expandir o cluster agregando pontos densamente conectados. Se um ponto está no raio de um ponto central, ele é adicionado ao cluster. Pontos que não pertencem a nenhum cluster são considerados ruído ou outliers. No entanto, a eficácia do DBSCAN depende da escolha adequada dos parâmetros, que podem ser difíceis de determinar sem conhecimento prévio do conjunto de dados (Ester et al., 1996; Khan et al., 2014).

### 2.4.4 Índice de silhueta

O Índice de silhueta (*Silhouette Score*) é uma métrica amplamente utilizada para avaliar a qualidade de clusters formados por algoritmos de agrupamento, medindo o quão similar um ponto é ao seu próprio cluster (coesão) em comparação com outros clusters (separação). Essa métrica fornece uma forma intuitiva de entender a proximidade dos pontos dentro do mesmo cluster e a separação entre clusters diferentes. O índice varia de -1 a 1, no qual valores altos (próximos a 1) indicam que os pontos estão bem agrupados e separados de outros clusters, sugerindo uma boa estrutura. Valores próximos a 0 indicam clusters sobrepostos, e valores negativos sugerem que muitos pontos podem estar no cluster errado. Esta métrica é especialmente útil para comparar a desempenho de diferentes algoritmos, como BKMeans, DBSCAN e K-Modes, e também para determinar o número ideal de clusters. Além disso, o *Silhouette Score* pode ser visualizado graficamente, facilitando a interpretação e validação dos resultados (Rousseeuw, 1987).

## 2.5 CONSIDERAÇÕES FINAIS

Neste capítulo, apresentamos os conceitos fundamentais que embasam o presente trabalho. Iniciamos com uma discussão sobre Sistemas Tutores Inteligentes, com foco especial no sistema PAT2Math, que serviu como fonte dos dados analisados nesta pesquisa. Em seguida,

exploramos o conceito de *misconceptions* na matemática, diferenciando-os de simples erros e apresentando as principais categorias teóricas propostas na literatura.

Também introduzimos as árvores de expressão como uma forma estruturada de representar equações algébricas, destacando suas vantagens para a análise computacional de padrões matemáticos. A combinação destes três elementos teóricos fornece o alicerce para a metodologia desenvolvida neste trabalho.

A compreensão dos STIs e seu funcionamento é crucial para contextualizar a origem e natureza dos dados utilizados. O entendimento das *misconceptions* matemáticas define o que estamos buscando identificar, enquanto as árvores de expressão oferecem o meio através do qual podemos representar e analisar as equações algébricas de forma estruturada.

Por fim, apresentamos os fundamentos teóricos e técnicos que sustentam a metodologia de nossa pesquisa, com uma exploração aprofundada das técnicas de clustering, detalhando métodos essenciais como o Breathing KMeans, K-Modes, DBSCAN e a métrica de Índice de Silhueta. Cada um desses algoritmos oferece abordagens distintas para agrupamento de dados, revelando diferentes estratégias para identificar padrões e estruturas em conjuntos de dados complexos. A compreensão dessas técnicas de clustering é crucial para o desenvolvimento de nossa abordagem de identificação de *misconceptions* matemáticas.

No próximo capítulo, abordaremos os trabalhos relacionados, contextualizando nossa pesquisa no panorama mais amplo dos estudos sobre detecção de *misconceptions* em sistemas de tutoria inteligente. Essa revisão permitirá situar nossa contribuição metodológica em relação aos avanços existentes na área de educação matemática assistida por tecnologia.

# 3 TRABALHOS RELACIONADOS

Esta seção descreve os trabalhos relacionados ao presente estudo, que descrevem métodos para identificação de *misconceptions* em diferentes áreas de estudo. Para a seleção dos trabalhos descritos foram utilizadas as palavras-chave *misconceptions diagnosis*, *mathematical misconceptions*, *algebra misconceptions* e *machine learning* na busca de artigos no Google Acadêmico e nas bibliotecas digitais científicas SOL, ACM, IEEE e Springer. Os trabalhos foram pré-selecionados com base no seu título e após uma nova seleção foi realizada a partir da leitura do resumo.

## 3.1 FELDMAN ET AL. (2018): SÍNTESE DE PROGRAMAS

O trabalho de Feldman et al. (2018) visou identificar *misconceptions* desconhecidas na literatura. Foram realizados dois experimentos principais. O primeiro experimento tentou gerar programas de reconstrução para 40 erros catalogados por Ashlock (Ashlock, 1990), um pesquisador reconhecido por seu trabalho extensivo na análise e classificação de erros sistemáticos no aprendizado de matemática. Ashlock desenvolveu uma taxonomia abrangente de erros matemáticos comuns, com foco especial nos padrões recorrentes observados em estudantes do ensino fundamental. O primeiro experimento tentou gerar programas de reconstrução para 40 erros descritos por Ashlock, enquanto o segundo experimento usou dados de estudantes coletados pela MetaMetrics. O trabalho propôs uma forma de identificação de misconceptions, explorando as respostas dos estudantes e analisando todos os possíveis caminhos para obter tais respostas. A identificação de *misconceptions* foi realizada por meio de uma abordagem de síntese de programas, que reconstrói procedimentos lógicos potencialmente empregados pelos alunos, baseando-se exclusivamente nas respostas finais fornecidas. Esta metodologia permite a análise de caminhos conceituais possíveis para elucidar as concepções equivocadas na resolução de problemas de matemática. O primeiro experimento visou medir quantas *misconceptions* conhecidas foram identificadas. Eles conseguiram identificar 28 das 40 *misconceptions* conhecidas (70%). Para o segundo experimento, utilizaram-se dados de estudantes coletados do sistema de aprendizagem MetaMetrics. Os dados compreenderam as respostas de 296 alunos para 32 problemas de adição e subtração. Grupos de soluções com três ou mais soluções do mesmo aluno foram definidos, resultando em 868 grupos, dos quais 111 apresentaram pelo menos uma *misconception*. Os resultados indicaram que o algoritmo conseguiu reconstruir a solução para 86% dos grupos, dos quais 77 foram classificados como precisos (o algoritmo reconstruiu a *misconception* corretamente) e parcialmente precisos (a reconstrução não representou o processo de solução real). Os autores observaram que o MetaMetrics permite apenas a entrada da solução final do problema é uma limitação, uma vez que os passos no processo de resolução dos alunos, que poderiam ajudar na identificação de misconceptions, são omitidos. A solução proposta também pressupõe que todos os problemas dentro do mesmo grupo são resolvidos de maneira semelhante, o que nem sempre é verdadeiro. Assim, o ruído algorítmico pode ser produzido por erros aleatórios que não são resultados de *misconceptions*. Esse estudo inova na identificação de *misconceptions* desconhecidos na literatura, empregando uma abordagem analítica das respostas. A relevância deste trabalho para o trabalho proposto está na sua abordagem sistemática e na utilização de dados reais de estudantes.

## 3.2 ANDERSSON E JOHANSSON (2015): CLUSTERIZAÇÃO

O trabalho de Andersson e Johansson (2015) objetivou identificar *misconceptions* em um ambiente inteligente de aprendizagem projetado para ensinar matemática a crianças de 6 a 10 anos, por meio de uma versão digitalizada do jogo bancário Montessori, onde os alunos executam operações de adição e subtração com material concreto simulado. Para identificar misconceptions, o estudo analisou características como a configuração atual dos elementos no tabuleiro interativo, os itens nas áreas de troca, a resposta dada ao problema matemático, e a quantidade de elementos nas áreas específicas para trocas. Os autores propuseram o uso de técnicas de agrupamento para agrupar erros e identificar *misconceptions* relacionadas a esses erros. Uma primeira solução foi desenvolvida utilizando o algoritmo k-means, que exigia que o número de clusters de dados existentes fosse conhecido antecipadamente. A segunda implementação utilizou o algoritmo DBSCAN, que determinou automaticamente o número de clusters, mas exigia o conhecimento de certos valores de parâmetros. Os autores fizeram uma modificação adicional empregando uma variante fuzzy de DBSCAN (FN-DBSCAN), que atribui um nível de confiança a cada erro associado a um cluster. No entanto, em um teste com usuários reais, o algoritmo não foi capaz de identificar misconceptions, possivelmente devido ao pequeno número de pontos de dados de diferentes alunos. Para uma avaliação mais precisa da solução, os autores desejavam testá-la com *misconceptions* comuns. Portanto, eles criaram usuários simulados com *misconceptions* conhecidas. Após vinte iterações, o sistema identificou 107 erros e uma possível *misconception*. Após quarenta rodadas, foram identificados 206 erros e três possíveis *misconceptions*. Os autores concluíram que o algoritmo se comporta melhor analisando os dados do usuário a longo prazo. Focando em um ambiente de aprendizagem inteligente para crianças, esse trabalho utiliza técnicas de agrupamento para identificar *misconceptions*. As implementações dos algoritmos k-means e DBSCAN são diretamente aplicáveis ao trabalho proposto, porém aplicados em um diferente contexto.

## 3.3 ELMADANI ET AL. (2012): MINERAÇÃO DE DADOS

Elmadani et al. (2012) emprega mineração de dados para identificação de *misconceptions* no EER-Tutor, utilizado para ensinar criação de modelos conceituais de banco de dados (Entidade-Relacionamento - ER) e caracterizado como um STI baseado em restrições. A interface de entrada do sistema permite que os usuários criem esquemas EER que satisfazem um conjunto de requisitos e submetam suas soluções, que são verificadas quanto a violações de restrições. Essas restrições avaliam a solução do estudante em termos de correção semântica e sintática. O sistema registra informações detalhadas de cada sessão, incluindo cada tentativa do estudante em cada problema e seu resultado, bem como o histórico de cada restrição violada. Embora ele forneça feedback para erros de temporização, ele não pode determinar se esses erros são devidos a *misconceptions*. Os autores usaram o algoritmo FP-Growth de mineração de regras para identificar possíveis *misconceptions* com base nas relações mais comuns entre violações de restrições nos modelos ER construídos pelos alunos. O banco de dados consiste em registros detalhados de 1135 modelos de estudantes e as violações de restrições associadas. Um especialista analisou os dados para identificar possíveis *misconceptions* sobre os erros coletados. O próximo passo foi gerar regras de associação com um nível mínimo de confiança de 0,9. O processamento de dados realizado pelo algoritmo gerou 912 conjuntos. Em média, uma restrição apareceu em 44 conjuntos. O maior número de ocorrências foi de 238; eles apareceram no trabalho de pelo menos 6700 alunos sendo considerados *misconceptions*. Os autores começaram a desenvolver uma hierarquia baseada em resultados, indicando que uma violação de restrição no topo de

uma hierarquia pode indicar que o resto do ramo está errado. Os especialistas analisaram os resultados e os consideraram satisfatórios. Nesse estudo é destacada a aplicabilidade de técnicas de mineração de dados na identificação de padrões de erro. Essa abordagem é particularmente relevante devido ao foco na análise de padrões de respostas erradas, sendo um dos aspectos centrais da proposta atual.

## 3.4 GOMES E JAQUES (2020): CLUSTERIZAÇÃO COM DADOS NORMALIZADOS

O trabalho de Gomes e Jaques (2020) é o mais semelhante e também usa algoritmos de clustering para identificar *misconceptions* algébricas em um sistema de tutoria baseado em etapas que ajuda os alunos a resolver equações passo a passo. Neste trabalho, as equações foram normalizadas e agrupadas com k-modes. Para os testes, recuperaram apenas os registros nos quais o aluno digitou a resposta errada. Os dados consistem em etapas de resolução de equações de primeiro grau e *feedback* mínimo do tutor (se um passo estava correto ou incorreto). Os resultados indicaram ser possível identificar *misconceptions* automaticamente com clustering. No entanto, a estrutura interna usada para representar a equação e o grande número de termos especiais em uma equação (por exemplo, parênteses) levaram a alguns problemas e resultados incorretos. Embora este seja o trabalho mais semelhante, nosso trabalho difere em dois aspectos. Primeiro, usa uma árvore de expressão para representar as etapas de resolução de problemas dos alunos em vez da normalização das equações. O trabalho de Gomes e Jaques (2020) usa equações normalizadas, mas sem tratamento para caracteres especiais. Essa decisão levou a saídas com estrutura inconsistente. Além disso, esses termos adicionais em uma equação tornam a representação maior para cada entrada, tornando-a mais difícil de agrupar. Segundo, o trabalho proposto usa o algoritmo DBSCAN em vez do k-modes. O uso de clustering categórico (k-modes) resulta em clusters que têm alguns resultados ruidosos que, em alguns deles, não representam um cenário real ou uma *misconception*. Já com o uso do DBSCAN, uma das suas vantagens é lidar melhor com dados ruidosos. Isso ocorre porque o DBSCAN não exige que todos os pontos de dados em um cluster tenham a mesma forma. Como resultado, o DBSCAN pode identificar clusters que não são perfeitamente esféricos, gerando menos ruído.

## 3.5 CONSIDERAÇÕES FINAIS

Este trabalho realizou uma revisão de estudos que empregam algoritmos para identificação de *misconceptions* em problemas matemáticos. As metodologias abordadas incluem o uso de árvores de expressão, técnicas de agrupamento e algoritmos de mineração de dados. Os trabalhos citados são relevantes para o atual projeto, pois cada um desses estudos aborda a identificação de *misconceptions* na área da matemática utilizando diferentes metodologias e ferramentas, fornecendo informações e abordagens que podem ser comparadas ou até mesmo integradas na metodologia proposta.

Por exemplo, o trabalho de Feldman et al. (2018) propôs uma abordagem para identificar misconceptions, explorando as respostas dos alunos e analisando todas as possíveis soluções para um problema. Já Andersson e Johansson (2015) utilizaram técnicas de agrupamento para agrupar erros e identificar *misconceptions*. Elmadani et al. (2012) utilizou o algoritmo FP-Growth (de mineração de regras) para identificar possíveis *misconceptions* com base nas relações mais comuns entre violações de restrições nos modelos ER construídos pelos alunos.

O trabalho proposto difere das abordagens anteriores em dois aspectos. Primeiramente, adota o use de árvores de expressão para representar as etapas de resolução de problemas dos alunos, oferecendo uma alternativa à normalização de equações, conforme explorado em Gomes

e Jaques. Adicionalmente, utiliza o algoritmo DBSCAN para o agrupamento das soluções, em contraste ao k-modes, que apresenta alguns resultados ruidosos e inconsistentes, que pode ser uma alternativa viável e eficaz para identificar de forma automática, sem intervenção humana, *misconceptions* em problemas matemáticos.

Tabela 3.1: Comparação trabalhos relacionados

| Trabalho | Método de Aprendizado de Máquina | Área de Estudo | Algoritmos |
|---|---|---|---|
| Feldman et al. (2018) | Análise de respostas | Diversas áreas | Exploração de caminhos |
| Andersson e Johansson (2015) | Agrupamento | Matemática básica | K-means, DBSCAN, FN-DBSCAN |
| Elmadani et al. (2012) | Mineração de regras | Criação de modelos conceituais de Banco de Dados | FP-Growth, Regras de Associação |
| Gomes e Jaques (2020) | Agrupamento | Resolução de equações | K-Modes |
| Trabalho Proposto | Agrupamento de árvores de expressão | Resolução de equações | DBSCAN, K-Modes, BKMeans |

Analisando a Tabela 3.1, podemos observar alguns padrões significativos. A maioria dos trabalhos recentes nesta área tem adotado técnicas de agrupamento (clustering) como abordagem principal para identificação de misconceptions, embora com variações nos algoritmos específicos empregados. Esta tendência reflete a adequação das técnicas não supervisionadas para descobrir padrões naturais em dados educacionais sem depender de categorizações prévias.

Em termos de áreas de aplicação, poucos trabalhos focam especificamente na resolução de equações algébricas, ressaltando a contribuição potencial do presente estudo.

Quanto aos algoritmos, nota-se uma evolução da aplicação de técnicas mais simples para abordagens mais sofisticadas e especializadas. O trabalho de Andersson e Johansson (2015) já explorava a comparação entre diferentes algoritmos de clustering (K-means, DBSCAN e FN-DBSCAN), estabelecendo um precedente importante para nossa abordagem comparativa.

Ao analisar as limitações destes trabalhos, identificamos alguns padrões recorrentes:

- **Dependência de dados completos:** O trabalho de Feldman et al. (2018) demonstra limitações por basear-se apenas nas respostas finais dos estudantes, sem acesso aos passos intermediários. Esta dependência de dados completos restringe a capacidade de identificar *misconceptions* que se manifestam durante o processo de resolução.

- **Necessidade de grandes volumes de dados:** Andersson e Johansson (2015) demonstraram que seus algoritmos necessitavam de grandes volumes de dados coletados ao longo do tempo para identificar padrões significativos, o que pode limitar a aplicabilidade em cenários com dados escassos.

- **Intervenção humana na interpretação:** Elmadani et al. (2012) ainda dependia significativamente de especialistas humanos para interpretar os resultados e confirmar a presença de misconceptions, limitando a automatização completa do processo.

- **Representação inadequada das estruturas:** O trabalho de Gomes e Jaques (2020) enfrentou desafios na representação adequada de estruturas matemáticas complexas, particularmente com operadores especiais e parênteses, resultando em clusters com inconsistências estruturais.

Nossa proposta atual busca superar estas limitações através de uma representação mais robusta baseada em árvores de expressão e da comparação sistemática de múltiplos algoritmos de clustering. Esta abordagem permite capturar com fidelidade a estrutura hierárquica das expressões matemáticas, reduzir a necessidade de intervenção humana na interpretação dos resultados, e identificar *misconceptions* mesmo com volumes moderados de dados.

# 4 MÉTODO

Este trabalho propõe a comparação de diferentes algoritmos de clustering para identificar *misconceptions* algébricas no log de respostas dos estudantes em um sistema de tutoria baseado em etapas. Também utilizamos a estrutura de árvore de expressão para representar as etapas de resolução de problemas dos estudantes.

As etapas para o desenvolvimento do trabalho consistem em: (1) coleta de dados, (2) pré-processamento e representação das equações utilizando árvores de expressão. A partir dessa representação, (3) fazemos a extração e vetorização das características dessas expressões, (4) aplicamos as transformações necessárias para cada algoritmo. Com os dados representados corretamente para cada teste, (5) aplicamos os algoritmos de clusterização para identificar os padrões das equações representando *misconceptions*. E, por fim, (6) aplicamos uma métrica de avaliação da qualidade dos agrupamentos (Silhouette Score), e (7) analisamos qualitativamente os agrupamentos gerados, investigando se os padrões identificados pelos algoritmos correspondem aos *misconceptions* esperados e verificando a consistência dos grupos formados. Essas etapas são explicadas mais detalhadamente nas próximas subseções.

## 4.1 COLETA DE DADOS

Como explicado na Seção 2.1, o PAT2Math é um STI que auxilia os estudantes na resolução de equações de primeiro grau, ou seja, fornece assistência para cada etapa da resolução do problema pelo aluno na forma de um *feedback* mínimo (certo ou errado), *feedback* de erro e dicas quando o aluno fica bloqueado e não sabe como proceder.

A coleta de dados para este estudo foi realizada entre fevereiro de 2017 e maio de 2019, em escolas da rede pública da região metropolitana de Porto Alegre. Durante o período de coleta, os estudantes utilizaram o sistema PAT2Math como parte regular de suas atividades curriculares de matemática, sob a supervisão de seus professores. No total, os estudantes resolveram 330 equações diferentes, distribuídas em diversos níveis de dificuldade conforme a progressão do conteúdo programático. Estas equações variavam desde formas simples como $x + a = b$ até estruturas mais complexas envolvendo frações, múltiplas operações e parênteses. A interação dos estudantes com o sistema gerou um conjunto de dados abrangente, contendo 1.319 registros de passos incorretos, utilizados para a análise neste trabalho.

Para cada etapa do aluno, o banco de dados PAT2Math armazena um registro. O registro de cada etapa tem o ID do aluno, o nível de dificuldade da equação, a etapa atual fornecida pelo aluno, sua etapa anterior e o feedback mínimo fornecido pelo PAT2Math para a etapa (se está certo ou errado). Devemos primeiro detectar e avaliar o erro que o aluno cometeu para identificar uma concepção errada. Consequentemente, só precisamos recuperar os registros onde a etapa atual está errada. Então, neste estudo, usamos as etapas erradas e anteriores.

A Tabela 4.1 apresenta uma amostra representativa dos dados coletados do sistema PAT2Math para análise neste estudo. Cada linha da tabela contém um par de passos sequenciais na resolução de uma equação algébrica por um estudante, onde a coluna “Passo Anterior” representa a equação em um determinado estágio da resolução. Este é o passo que o estudante havia completado corretamente ou que estava tentando modificar. E a coluna “Passo Errado” representa a etapa de resolução identificada pelo sistema como incorreta. Este passo contém o erro ou *misconception* que procuramos identificar.

Tabela 4.1: Exemplos dos dados coletados

| **Passo Anterior** | **Passo Errado** |
|---|---|
| x + 7 = 12 | x = 12 + 7 |
| x + 3 = 15 | x = 15 + 3 |
| x = 15 - 8 | x = 2 |
| x + 5 = -4 | x = -4 + 5 |
| x = -4 - 5 | x = -1 |
| x = -4 - 5 | x = 1 |
| x = -4 - 5 | x = 5 |
| x = -11 - 7 | x = 18 |
| x = -11 - 7 | x = 18 |
| x + 6 = -12 | x = 12 - 6 |
| x + 6 = -12 | x = -12 + 6 |

A análise destes pares permite identificar padrões sistemáticos de erro que revelam *misconceptions* específicas. Por exemplo, nas duas primeiras linhas da tabela, observamos o mesmo padrão de erro: ao tentar isolar a variável $x$, o estudante move o termo constante para o outro lado da equação sem inverter a operação, resultando em uma adição quando deveria ser uma subtração. Este é um exemplo clássico de *misconception* na aplicação das regras de manipulação algébrica.

Também podemos observar outros tipos de erros, como cálculos aritméticos incorretos (por exemplo, $x = 15 - 8$ resultando em $x = 2$ em vez de $x = 7$), e erros de sinal nas operações com números negativos (como $x = -4 - 5$ calculado como $x = -1$, $x = 1$ ou $x = 5$, quando o resultado correto seria $x = -9$).

## 4.2 PRÉ-PROCESSAMENTO DOS DADOS

Uma das principais etapas desse trabalho é o pré-processamento dos dados de entrada, dividido em três etapas: 1) padronização das equações, 2) criação de árvores de expressão e 3) extração e 4) vetorização das características das equações.

Na primeira etapa (**Etapa 1**), a equação foi padronizada substituindo valores numéricos por letras. Uma falsa concepção algébrica não depende dos valores numéricos dos termos de uma equação. Portanto, $x + 3 = 6$ e $x + 2 = 4$ representam equações isomórficas, que podem ser denotadas por $x + a = b$, o que ainda permite identificar a falsa concepção. Isso é possível porque expressões matemáticas, como equações, possuem uma forma estruturada diferente de respostas textuais.

Na segunda etapa (**Etapa 2**), a equação foi representada como uma árvore de expressão usando a notação pós-fixa, também conhecida como "notação polonesa reversa" (NPR). Uma árvore de expressão é uma maneira de representar expressões matemáticas em uma estrutura hierárquica semelhante a uma árvore. Cada nó interno da árvore representa um operador e cada nó folha representa um operando. Usar uma árvore de expressão para representar uma expressão matemática tem várias vantagens. O uso de árvores de expressão para representar expressões matemáticas esclarece a ordem das operações (Lample e Charton, 2019). Consequentemente, cuida da precedência e associatividade e elimina a necessidade de parênteses. Pesquisas anteriores indicaram que a representação da expressão matemática na resolução de problemas pode ter um efeito nos resultados dos algoritmos de aprendizado de máquina (Gomes e Jaques, 2020). A NPR é uma notação matemática na qual os operadores seguem seus operandos. Ela não

precisa de parênteses como as notações infixas, tornando seu uso mais simples para algoritmos de aprendizado de máquina.

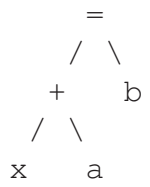
Na terceira etapa (**Etapa 3**), características são extraídas das árvores de expressão resultantes. As características incluem o número total de nós, a contagem de operadores, a contagem de variáveis e a profundidade da árvore. Em seguida, (**Etapa 4**) essas características são vetorizadas, transformando as características extraídas em uma matriz numérica, facilitando o processamento com algoritmos de *clustering*.

Abaixo está um exemplo ilustrando cada uma das etapas realizadas, utilizando a equação `"x + 5 = 7"` como referência.

1. **Padronização das equações**: A equação `"x + 5 = 7"` é padronizada para `"x + a = b"`.

2. **Criação de árvores de expressão**:

   Exemplo de árvore de expressão para a equação `x + a = b`:




3. **Extração e vetorização das características**:

```
{ 'total_nodes': 5,
  'operator_count': {'=': 1, '+': 1},
  'variable_count': 2,
  'depth': 2 }
```

4. **As características são transformadas em vetores numéricos:** `[5, 1, 1, 2, 2]`

## 4.3 CLUSTERING

Para o agrupamento, foram utilizados três algoritmos distintos: *DBSCAN*, *k-modes* e *BKMeans*. Cada um desses algoritmos possui características e parâmetros específicos, os quais são detalhados a seguir.

### 4.3.1 DBSCAN

O *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) é um método de agrupamento baseado em densidade que identifica clusters em um conjunto de dados ao detectar regiões de alta densidade separadas por regiões de baixa densidade. O algoritmo é particularmente eficaz em identificar clusters de formas arbitrárias e é robusto a ruídos e outliers.

O processo do DBSCAN inicia selecionando um ponto aleatório e, em seguida, identifica todos os pontos que estão em uma distância $\epsilon$ deste ponto. Se o número de pontos dentro desse raio for igual ou superior a um número mínimo de pontos (*minPts*), um novo cluster é formado. Este processo é repetido para cada ponto no cluster recém-formado, expandindo-o até que não haja mais pontos que satisfaçam os critérios de densidade. Pontos que não pertencem a nenhum cluster são considerados ruído.

Uma característica distintiva do DBSCAN, em comparação com algoritmos como o *k-modes*, é que não é necessário definir o número de clusters a priori. No entanto, é fundamental definir os seguintes parâmetros:

- $\epsilon$ **(eps)**: distância máxima para que um ponto seja considerado vizinho de outro ponto no cluster. Neste trabalho, utilizamos $\epsilon = 0{,}3$.
- ***minPts***: número mínimo de pontos necessários para formar um cluster denso. Utilizamos $minPts = 2$.

Esses valores foram escolhidos com base em experimentações preliminares para equilibrar a sensibilidade do algoritmo na detecção de clusters significativos nos dados.

### 4.3.2 K-MODES

O algoritmo *k-modes* é uma extensão do *k-means* projetada para lidar com dados categóricos. Em vez de utilizar a distância euclidiana, como no *k-means*, o *k-modes* utiliza uma medida de dissimilaridade adequada para variáveis categóricas, como a distância de Hamming. O algoritmo agrupa os dados atribuindo cada ponto ao cluster cujo modo (valor mais frequente) é mais próximo, atualizando os modos iterativamente para minimizar a dissimilaridade total dentro dos clusters.

Os principais parâmetros do *k-modes* são:

- **k**: número de clusters. Neste estudo, definimos $k = 100$.
- **init**: método de inicialização dos modos. Utilizamos o método Huang', adequado para conjuntos de dados categóricos.
- **n_init**: número de vezes que o algoritmo será executado com diferentes modos iniciais. Definimos $n\_init = 5$ para aumentar a probabilidade de encontrar uma solução ótima.

### 4.3.3 BREATHING KMEANS (BKMEANS)

O *Breathing KMeans* (BKMeans) é uma variante do *k-means* que incorpora um mecanismo de "respiração" para ajustar dinamicamente os tamanhos dos clusters durante o processo de agrupamento. Isso permite que o algoritmo lide melhor com dados que possuem clusters de diferentes tamanhos e densidades.

Os parâmetros principais para o *BKMeans* são:

- **k**: número inicial de clusters. Utilizamos $k = 100$.
- **breathing_rate**: taxa de respiração utilizada para ajustar dinamicamente o tamanho dos clusters. Usamos $breathing\_rate = 5$, um valor que pondera entre qualidade e tempo de execução.
- **tolerance**: critério de convergência para parar as iterações. Usamos o valor padrão de 0,0001.

A escolha de $k = 100$ para ambos os algoritmos foi fundamentada na busca por um equilíbrio entre a granularidade dos clusters e a interpretabilidade dos resultados, com base em análises anteriores. Ao iniciar com um número maior de clusters, os algoritmos têm a

flexibilidade de identificar padrões mais específicos e sutis nos dados, o que é especialmente relevante em conjuntos de dados complexos ou heterogêneos. Esse valor elevado de $k$ permite que os algoritmos explorem uma variedade maior de possíveis agrupamentos, refinando iterativamente os clusters para melhor representar a estrutura intrínseca dos dados. Além disso, utilizar o mesmo valor de $k$ em ambos os algoritmos, facilita a comparação direta dos resultados, proporcionando uma avaliação consistente do desempenho e capacidade de identificação de padrões nos dados.

## 4.4 MÉTRICA DE AVALIAÇÃO: SILHOUETTE SCORE

Para avaliar a qualidade dos agrupamentos obtidos pelos algoritmos de clusterização, utilizamos o Silhouette Score (Rousseeuw, 1987), uma métrica que fornece uma interpretação e validação da consistência dos dados nos clusters identificados. Essa métrica é baseada em uma análise da coesão (distância média do ponto aos demais pontos do mesmo cluster) e da separação (menor distância média do ponto aos pontos de outros clusters). O Silhouette Score varia de -1 a 1 e fornece uma medida da qualidade do agrupamento para cada ponto.

Valores próximos de 1 indicam que o ponto está bem alocado dentro de seu cluster e bem separado dos outros clusters, sugerindo uma boa definição dos agrupamentos. Valores próximos de 0 indicam que o ponto está na fronteira entre clusters, sem uma clara distinção. Valores negativos sugerem que o ponto pode ter sido atribuído ao cluster incorreto, indicando um possível erro de agrupamento.

O Silhouette Score médio de todos os pontos fornece uma medida geral da qualidade do agrupamento. Um valor médio elevado indica que os clusters estão bem definidos e separados entre si.

No contexto deste trabalho, calculamos o Silhouette Score médio após aplicar cada algoritmo de clusterização aos dados, o que nos permitiu:

- Comparar a eficácia dos diferentes algoritmos, observando qual algoritmo produz clusters com maior coesão interna e melhor separação externa.
- Avaliar a adequação dos parâmetros escolhidos, como o valor de $k$ nos algoritmos *k-modes* e *BKMeans*, verificando se esses valores resultam em uma estrutura de clusters significativa.

Como o Silhouette Score depende das distâncias entre os pontos, a escolha da medida de distância apropriada e a padronização dos dados são cruciais para obter uma avaliação confiável. Nos algoritmos que utilizam distâncias métricas (como *BKMeans* e *DBSCAN*), asseguramos a padronização dos dados para evitar que características com magnitudes maiores influenciassem indevidamente o cálculo das distâncias.

## 4.5 PSEUDOCÓDIGO

O pseudocódigo apresentado no Código 1 ilustra as etapas necessárias para realizar a clusterização de equações utilizando os algoritmos KModes, BKMeans ou DBSCAN. A seguir, apresentamos uma explicação detalhada de cada parte do pseudocódigo:

**1) Função `carregar_dados`:** Esta função é responsável por carregar as equações que servirão como entrada para o processo de clusterização. As equações podem ser provenientes de uma base de dados, de um arquivo ou de qualquer outra fonte de dados estruturados.

**2) Função `extrair_caracteristicas`:** Esta função realiza a extração das características das equações. Para cada equação, uma árvore de expressão é gerada utilizando a função `gerar_arvore_expressao`, e as características relevantes são extraídas com a função `extract_features_from_tree`. As características incluem:

- Número total de nós na árvore;
- Contagem de operadores presentes;
- Número de variáveis utilizadas;
- Profundidade máxima da árvore.

**3) Função `vetorizar_caracteristicas`:** Após a extração, as características das equações estão em formato de dicionário ou estrutura similar. Esta função converte essas características em uma representação vetorial numérica, utilizando técnicas como a vetorização com um *DictVectorizer*. Isso facilita a análise pelos algoritmos de clusterização, que requerem dados numéricos.

**4) Função `padronizar_dados`:** Alguns algoritmos de clusterização são sensíveis à escala das características. Esta função aplica uma padronização aos dados vetoriais, normalmente utilizando um *StandardScaler*, para que todas as características tenham média zero e variância um. A padronização é especialmente necessária para os algoritmos BKMeans e DBSCAN.

**5) Função `aplicar_algoritmo_clusterizacao`:** Esta função aplica o algoritmo de clusterização escolhido aos dados. Os passos incluem:

- Inicializar o algoritmo com os parâmetros necessários (`parametros`), que podem variar dependendo do algoritmo (por exemplo, o número de clusters para KModes e BKMeans, ou `eps` e `min_samples` para DBSCAN);
- Executar o método `fit_predict` para realizar a clusterização;
- Calcular a pontuação média de Silhouette usando `silhouette_score`, que avalia a qualidade dos agrupamentos formados.

**6) Função `main`:** Esta é a função principal que coordena todo o processo:

- Carrega as equações utilizando `carregar_dados`;
- Extrai as características das equações com `extrair_caracteristicas`;
- Vetoriza as características através de `vetorizar_caracteristicas`;
- Verifica se o algoritmo escolhido é BKMeans ou DBSCAN. Se for, aplica a padronização dos dados usando `padronizar_dados`, pois esses algoritmos são sensíveis à escala das características, ou seja, a magnitude dos valores pode influenciar significativamente os resultados da clusterização. Ao padronizar as características, garantimos que nenhuma delas terá um impacto desproporcional no processo de formação dos clusters.
- Aplica o algoritmo de clusterização com `aplicar_algoritmo_clusterizacao`;
- Recebe os clusters resultantes e a pontuação de Silhouette para análise.

Este pseudocódigo demonstra a aplicação prática de diferentes algoritmos de clusterização para agrupar equações matemáticas, utilizando árvores de expressão para representar as características estruturais de cada equação. Ao converter as equações em representações vetoriais e aplicar técnicas de clusterização, é possível identificar padrões e agrupamentos nos dados, o que pode ser útil para análises educacionais e para a identificação de erros comuns entre os alunos.

**Algoritmo 1:** Algoritmo de Agrupamento (KModes, BKMeans ou DBSCAN)

```
Data: Equações
Result: Clusters e Pontuação Silhouette
Função extrair_caracteristicas(equations):
    features = [];
    foreach equation em equations do
        tree = gerar_arvore_expressao(equation);
        features.append(extract_features_from_tree(tree));
    end
    return features;
Função vetorizar_caracteristicas(features):
    flattened_features = flatten_features(features);
    X = dict_vectorizer.fit_transform(flattened_features);
    return X;
Função padronizar_dados(X):
    X_scaled = standard_scaler.fit_transform(X);
    return X_scaled;
Função aplicar_algoritmo_clusterizacao(X, parametros):
    clusters = algoritmo.fit_predict(X);
    silhouette_avg = silhouette_score(X, clusters);
    return silhouette_avg, clusters;
Função main():
    equations = carregar_dados ();
    features = extrair_caracteristicas(equations);
    X = vetorizar_caracteristicas(features);
    if algoritmo é BKMeans ou algoritmo é DBSCAN then
        X = padronizar_dados(X);
    end
    silhouette_avg, clusters = aplicar_algoritmo_clusterizacao(X,
      parametros);
Início do Programa;
    main();
```

Este pseudocódigo é genérico e pode ser adaptado para qualquer um dos três algoritmos mencionados. Ao especificar o algoritmo e seus parâmetros na função `aplicar_algoritmo_clusterizacao`, o processo se ajusta conforme as necessidades específicas de cada método de clusterização. A padronização condicional dos dados garante que apenas os algoritmos que requerem essa etapa (BKMeans e DBSCAN) a realizem, mantendo a integridade dos dados para o KModes.

O uso de árvores de expressão na extração de características permite capturar a estrutura sintática das equações, fornecendo uma representação rica para a análise. A combinação dessas técnicas facilita a identificação de padrões, erros comuns e possíveis áreas de dificuldade entre os estudantes, contribuindo para aprimorar estratégias educacionais e personalizar o ensino.

A implementação completa do código e os resultados obtidos podem ser encontrados no repositório GitHub: `https://github.com/joicecg/masters-identify-misconceptions`.

## 4.6 CONSIDERAÇÕES FINAIS

Aqui apresentamos uma metodologia para a identificação de *misconceptions* em equações algébricas em registros de estudantes baseada em agrupamento (*clustering*). Inicialmente, descrevemos o processo de coleta de dados a partir do sistema de tutoria PAT2Math, que registrou as etapas de resolução de equações dos alunos. Em seguida, apresentamos as etapas de pré-processamento dos dados, incluindo a padronização das equações, a criação de árvores de expressão, e a extração e vetorização das características das equações. Esses passos foram fundamentais para preparar os dados para os algoritmos de *clustering*.

Discutimos a aplicação de três algoritmos diferentes: *DBSCAN*, *k-modes* e *BKMeans*. Cada algoritmo foi escolhido por suas características específicas e parâmetros adequados à natureza dos dados e ao objetivo do estudo. Por fim, abordamos a métrica de avaliação utilizada, o Silhouette Score, para avaliar a qualidade dos agrupamentos formados.

# 5 RESULTADOS

Após aplicadas as etapas de extração, pré-processamento e clusterização dos dados, descritas no Capítulo 4, aqui apresentamos os resultados da dissertação. Não foi aplicado nenhum tipo de tratamento ou etapa de pós-processamento adicional.

Para avaliar a qualidade dos agrupamentos formados, utilizamos a métrica de avaliação Silhouette Score. Essa métrica é essencial para validar a eficácia dos algoritmos aplicados e fornecer uma análise quantitativa da qualidade dos grupos formados.

Os resultados apresentados a seguir refletem diretamente a aplicação dos algoritmos *DBSCAN*, *k-modes* e *BKMeans* nos dados, juntamente com as observações baseadas na análise dos valores de Silhouette Score.

Em todos os testes realizados neste estudo, empregamos um conjunto de dados consistente e unificado, o qual também foi utilizado em Gomes e Jaques (2020). A utilização do mesmo conjunto de dados empregado nesse estudo anterior permite a análise de consistência e a possibilidade de avaliar as melhorias ou diferenças alcançadas pela implementação das novas técnicas propostas neste trabalho em relação aos métodos anteriormente aplicados.

## 5.1 KMODES

Para o primeiro teste, foi utilizado o mesmo algoritmo e parâmetros utilizados por Gomes e Jaques (2020) (um agrupamento categórico (k-modes)), com exceção do número de *clusters*. Para definir o número de *clusters* utilizados, usamos o *Elbow Method*, que retornou um valor de 100 *clusters*. Além disso, a abordagem difere apenas no formato de entrada. Nesse novo teste, as equações são representadas como árvores de expressão com notação pós-fixada, como explicado na Seção 4.2.

O resultado gerou um total de 93 clusters distintos, os quais são avaliados a seguir.

A Tabela 5.1 mostra uma parte dos resultados obtidos com a utilização do KModes. Cada linha representa um cluster identificado, com o respectivo *misconception* e exemplos das equações associadas.

Tabela 5.1: Clusters gerados pelo KModes

| ID | Misconception | Passo Anterior | Passo Errado |
|---|---|---|---|
| 0 | Aplicação incorreta de adição/subtração | $x + b = aa$ | $x = aa + b$ |
| 0 | Aplicação incorreta de adição/subtração | $x - b = -aa$ | $x = aa + b$ |
| 57 | Aplicação incorreta das regras de divisão | $(ax + bb)/c = (aaax)/b + c$ | $-(ax)/bb = (aaax)/b - c/d$ |
| 57 | Aplicação incorreta das regras de divisão | $(-ax + bb)/c = (aaax)/b + c$ | $-(ax)/bb = (aaax)/b - c/d$ |
| 69 | Manipulação incorreta de frações complexas | $(aax - bb)/c = aaax/b + c$ | $(aax - bb)/c = aaax/b + c$ |
| 69 | Manipulação incorreta de frações complexas | $(-ax + bbb)/cc - dddx = -aaaa$ | $-ax + bbb - cc \cdot dddx = -aaaa + b \cdot (cccx + dddx)$ |

Continua na próxima página

**Continuação da Tabela 5.1**

| ID | Misconception | Passo Anterior | Passo Errado |
|---|---|---|---|
| 75 | Má compreensão do isolamento da variável | $x = -a - b$ | $x = a$ |
| 75 | Má compreensão do isolamento da variável | $x = -aa - b$ | $x = aa$ |
| 77 | Combinação incorreta de termos semelhantes | $aax - bb - cx = ax - b - cc$ | $aax - bx = ax - b - cc - e$ |
| 77 | Combinação incorreta de termos semelhantes | $aax - b - cx = ax - b - cc$ | $aax - bx = ax - b - cc - d$ |

Com base nos clusters gerados pelo algoritmo KMODES para identificar concepções, a Tabela 5.2 apresenta uma visão geral dos clusters gerados pelo algoritmo conforme o número de instâncias, os IDs dos clusters correspondentes, uma descrição genérica para cada agrupamento e exemplos representativos de cada categoria:

Tabela 5.2: Características dos agrupamentos gerados pelo KMODES

| Número de Instâncias no Cluster | IDs dos Clusters | Característica dos Agrupamentos |
|---|---|---|
| **1** | 1, 2, 9, 10, 14, 20, 22, 23, 25, 27, 28, 32, 33, 35, 36, 40, 42, 45, 49, 51, 52, 55, 56, 58, 59, 63, 66, 67, 69, 71, 74, 77, 80, 81, 82, 83, 85, 86, 87, 88, 89, 90, 92 | **Padrões Únicos ou Muito Raros** Equações ou concepções que aparecem apenas uma vez, possivelmente representando casos excepcionais ou ruído nos dados. |
| **2-5** | 4, 6, 7, 11, 12, 15, 18, 19, 21, 24, 29, 30, 31, 34, 37, 38, 41, 43, 44, 46, 47, 50, 57, 60, 62, 64, 70, 76, 78, 91 | **Pequenos Agrupamentos** Padrões que ocorrem poucas vezes, indicando concepções menos comuns, mas ainda significativas. |
| **6-20** | 3, 5, 13, 16, 26, 39, 48, 61, 65, 68, 73, 84 | **Padrões de Frequência Moderada** Concepções que aparecem com uma frequência intermediária, representando conceitos ou equações relativamente comuns. |
| **21-50** | 0, 8, 17, 54, 79 | **Padrões de Alta Frequência** Concepções que ocorrem com bastante frequência, indicando conceitos amplamente reconhecidos ou utilizados. |
| **51-150** | 53 | **Padrões Muito Comuns** Concepções predominantes que aparecem muitas vezes, refletindo os conceitos mais comuns identificados pelo algoritmo. |

Continua na próxima página

**Continuação da Tabela 5.2**

| Número de Instâncias | IDs dos Clusters | Descrição |
|---|---|---|
| **151+** | 72, 75 | **Padrões Extremamente Comuns** Concepções dominantes que representam a maioria das instâncias, indicando os conceitos mais recorrentes e significativos. |

- **Padrões Únicos ou Muito Raros**:
  - **Cluster 87:** $x + (bx + c) = aa,\ ax^b + cx = aa$
  - **Cluster 59:** $\frac{x}{b} + c = a,\ x = a - b \cdot c$
- **Pequenos Agrupamentos**:
  - **Cluster 21:** $x = a - b,\ x = a \cdot -b$
  - **Cluster 62:** $\frac{x}{b} = aa,\ x = \frac{aa}{b}$
- **Padrões de Frequência Moderada**:
  - **Cluster 16:** $x + b = aa,\ x + b - c = aa$
  - **Cluster 68:** $x = \frac{aaa}{bb},\ x = \frac{aaa}{-bb}$
- **Padrões de Alta Frequência**:
  - **Cluster 4:** $x + b = aa,\ x = aa + b$
  - **Cluster 79:** $x + b = aa,\ a - bb = x$
- **Padrões Muito Comuns**:
  - **Cluster 53:** $ax = -a + b,\ x = \frac{aa}{b}$
- **Padrões Extremamente Comuns**:
  - **Cluster 72:** $x + b = -aa,\ x = -aa$
  - **Cluster 75:** $x + b = -a,\ x = a$

## 5.2 BREATHING KMEANS

O BKMeans incorpora um mecanismo de respiração que ajusta os tamanhos dos clusters durante o processo de clustering. Essa característica permite que o algoritmo lide melhor com dados que apresentam clusters de diferentes tamanhos e densidades, promovendo uma identificação mais precisa de padrões complexos.

A aplicação desse método resultou um total de 94 clusters distintos, os quais são avaliados a seguir.

A Tabela 5.3 mostra uma parte dos resultados obtidos com a utilização do BKMeans. Cada linha representa um cluster identificado, com o respectivo *misconception* e exemplos das equações associadas.

Tabela 5.3: Clusters gerados pelo BKMeans

| ID | Misconception | Passo Anterior | Passo Errado |
|---|---|---|---|
| 6 | Aplicação incorreta de adição/subtração | $x + b = aa$ | $x = aa + b$ |
| 6 | Aplicação incorreta de adição/subtração | $x - bb = -a$ | $x = a + bb$ |
| 11 | Aplicação incorreta das regras de divisão | $\frac{-ax+bb}{c} = \frac{aaax}{b} + c$ | $-\frac{ax}{bb} = \frac{aaax}{b} - \frac{c}{d}$ |
| 11 | Aplicação incorreta das regras de divisão | $\frac{ax+bb}{c} = \frac{aaax}{b} + c$ | $-\frac{ax}{bb} = \frac{aaax}{b} - \frac{c}{d}$ |
| 20 | Manipulação incorreta de frações complexas | $\frac{-ax+bbb}{cc} - dddx = -aaaa$ | $-ax + bbb - cc \cdot dddx = -aaaa + b \cdot (cccx + dddx)$ |
| 20 | Manipulação incorreta de frações complexas | $\frac{aax-bb}{c} = \frac{aaax}{b} + c$ | $\frac{aax-bb}{c} = \frac{aaax}{b} + c$ |
| 72 | Combinação incorreta de termos semelhantes | $aax - b - cx = ax - b - cc$ | $aax - bx = ax - b - cc - d$ |
| 72 | Combinação incorreta de termos semelhantes | $aax - bb - cx = ax - b - cc$ | $aax - bx = ax - b - cc - e$ |
| 73 | Interpretação errônea das operações com frações | $\frac{aax-bx}{cx} = \frac{ax+bbb}{cc}$ | $a - \frac{b}{c} = \frac{ax+bbb}{cc}$ |
| 73 | Interpretação errônea das operações com frações | $\frac{aax-bb}{c} = \frac{aaax}{b} + c$ | $a - \frac{b}{c} = \frac{ax+bbb}{cc}$ |
| 92 | Má compreensão do isolamento da variável | $x = aa - b$ | $x = a$ |
| 92 | Má compreensão do isolamento da variável | $x = -a - b$ | $x = a$ |

Com base nos clusters gerados pelo algoritmo BKMEANS para identificar concepções, a Tabela 5.4 apresenta uma visão geral dos clusters gerados pelo algoritmo conforme o número de instâncias, os IDs dos clusters correspondentes, uma descrição genérica para cada agrupamento e exemplos representativos de cada categoria:

Tabela 5.4: Características dos agrupamentos gerados pelo BKMEANS

| Número de Instâncias no Cluster | IDs dos Clusters | Característica dos Agrupamentos |
|---|---|---|
| **1** | 4, 7, 10, 20, 21, 22, 27, 29, 35, 36, 37, 40, 42, 45, 46, 47, 49, 55, 57, 59, 60, 62, 64, 68, 69, 70, 71, 72, 73, 74, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90 | **Padrões Únicos ou Muito Raros** Equações ou concepções que aparecem apenas uma vez, possivelmente representando casos excepcionais ou ruído nos dados. |

Continua na próxima página

**Continuação da Tabela 5.4**

| Número de Instâncias | IDs dos Clusters | Descrição |
|---|---|---|
| **2-5** | 2, 6, 8, 9, 12, 15, 17, 18, 23, 26, 30, 39, 48, 43, 44, 66, 50, 65, 54, 78, 51, 76, 52, 53, 67, 75, 56, 58, 61, 63, 77 | **Pequenos Agrupamentos** Padrões que ocorrem poucas vezes, indicando concepções menos comuns, mas ainda significativas. |
| **6-20** | 1, 11, 13, 14, 19, 28, 31, 32, 33, 34, 38, 41 | **Padrões de Frequência Moderada** Concepções que aparecem com uma frequência intermediária, representando conceitos ou equações relativamente comuns. |
| **21-50** | 0, 5, 16, 24, 25 | **Padrões de Alta Frequência** Concepções que ocorrem com bastante frequência, indicando conceitos amplamente reconhecidos ou utilizados. |
| **51-150** | 91 | **Padrões Muito Comuns** Concepções predominantes que aparecem muitas vezes, refletindo os conceitos mais comuns identificados pelo algoritmo. |
| **151+** | 3, 93 | **Padrões Extremamente Comuns** Concepções dominantes que representam a maioria das instâncias, indicando os conceitos mais recorrentes e significativos. |

- **Padrões Únicos ou Muito Raros**:
  - **Cluster 84:** $x + b = a,\ x + b = a + b$
  - **Cluster 81:** $-x - bbb = -aaa,\ x = aaa$
- **Pequenos Agrupamentos**:
  - **Cluster 8:** $\frac{-ax+bb}{c} = \frac{aaax}{b} + \frac{cx}{dd} - \frac{e}{ff},\ -\frac{ax}{bb} = \frac{aaax}{b} - \frac{c}{dd} + \frac{ex+ff}{g}$
  - **Cluster 56:** $-x = aa,\ -x = aa \times -b$
- **Padrões de Frequência Moderada**:
  - **Cluster 16:** $x + bb = aa,\ x + bb = aa - bb$
  - **Cluster 25:** $x + b = aa,\ x = aa - b$
- **Padrões de Alta Frequência**:
  - **Cluster 5:** $x + b = aa,\ x = aa + b$ (Múltiplas instâncias)
- **Padrões Muito Comuns**:
  - **Cluster 91:** $\frac{x}{bb} = -aa,\ x = \frac{aa}{b}$ (Múltiplas instâncias)
- **Padrões Extremamente Comuns**:

- **Cluster 3:** $x = -a - b,\ x + bb = -aa$ (Numerosas instâncias)
- **Cluster 93:** $x = aa - b,\ x = a + bb$ (Mais de 500 instâncias)

## 5.3 DBSCAN

O DBSCAN é utilizado para identificar clusters em regiões de alta densidade nas respostas dos estudantes, sem a necessidade de definir o número de clusters. O algoritmo é eficaz em lidar com dados ruidosos e detectar clusters de formas arbitrárias.

O resultado gerou um total de 51 clusters distintos, os quais são avaliados a seguir.

A Tabela 5.5 mostra uma pequena parte dos resultados obtidos por essa implementação. A tabela representa o conteúdo de três *clusters* gerados.

Tabela 5.5: Clusters gerados pelo DBSCAN

| ID | Misconception | Passo Anterior | Passo Errado |
|---|---|---|---|
| 0 | Aplicação incorreta de adição/subtração | $x + b = aa$ | $x = aa + b$ |
| 0 | Aplicação incorreta de adição/subtração | $x - bb = -a$ | $x = a + bb$ |
| 0 | Aplicação incorreta de adição/subtração | $x - b = -aa$ | $x = aa + b$ |
| 1 | Má compreensão do isolamento da variável | $x = aa - b$ | $x = a$ |
| 1 | Má compreensão do isolamento da variável | $x = -a - b$ | $x = a$ |
| 1 | Má compreensão do isolamento da variável | $x = -aa - b$ | $x = aa$ |
| 38 | Manipulação incorreta de frações complexas | $x = aa \cdot \frac{bx+c}{d}$ | $x = \frac{ax+b}{cc}$ |
| 38 | Manipulação incorreta de frações complexas | $\frac{ax}{bb} = \frac{aax+bb}{ccc}$ | $ax = aax + \frac{bb}{cc}$ |
| 38 | Manipulação incorreta de frações complexas | $\frac{x-bb}{cc} = -x + \frac{bb}{cc}$ | $\frac{x}{bb} - \frac{ccc}{ddd} = \frac{-x+bb}{cc}$ |
| 39 | Combinação incorreta de termos semelhantes | $\frac{x-bb}{cc} = -x + \frac{bb}{cc}$ | $aax - bx = ax - b - cc - d$ |
| 39 | Combinação incorreta de termos semelhantes | $\frac{ax+bb}{c} = \frac{aaax}{b} + c$ | $ax = aax + \frac{bb}{cc}$ |
| 39 | Combinação incorreta de termos semelhantes | $\frac{x}{bb} - \frac{ccc}{ddd} = \frac{-x+bb}{cc}$ | $a - \frac{b}{c} = \frac{ax+bbb}{cc}$ |

Tabela 5.6: Características dos agrupamentos gerados pelo DBSCAN

| Número de Instâncias no Cluster | IDs dos Clusters | Característica dos Agrupamentos |
|---|---|---|
| **1** | - | **Padrões Únicos ou Muito Raros** Equações ou concepções que aparecem apenas uma vez, possivelmente representando casos excepcionais ou ruído nos dados. |
| **2-5** | 6, 9, 10, 16, 17, 18, 19, 20, 21, 24, 25, 27, 28, 29, 30, 32, 33, 35, 36, 37, 38, 39, 40, 43, 44, 45, 46, 47, 48, 49 | **Pequenos Agrupamentos** Padrões que ocorrem poucas vezes, indicando concepções menos comuns, mas ainda significativas. |
| **6-20** | 2, 5, 12, 14, 15, 22, 23, 26, 31, 34, 41, 42 | **Padrões de Frequência Moderada** Concepções que aparecem com uma frequência intermediária, representando conceitos ou equações relativamente comuns. |
| **21-50** | 4, 7, 8, 13 | **Padrões de Alta Frequência** Concepções que ocorrem com bastante frequência, indicando conceitos amplamente reconhecidos ou utilizados. |
| **51-150** | 11 | **Padrões Muito Comuns** Concepções predominantes que aparecem muitas vezes, refletindo os conceitos mais comuns identificados pelo algoritmo. |
| **151+** | 1, 3 | **Padrões Extremamente Comuns** Concepções dominantes que representam a maioria das instâncias, indicando os conceitos mais recorrentes e significativos. |
| * | 0 | **Ruído e entradas únicas** Todas as instâncias que não se encaixam em nenhum outro cluster |

- **Pequenos Agrupamentos**:
  - **Cluster 17:** $x = a - b,\ x^b = aa$
  - **Cluster 44:** $ax - b = -x + bb,\ ax + x = -a + bb$
- **Padrões de Frequência Moderada**:
  - **Cluster 15:** $x + b = aa,\ x + b - c = aa + b$
  - **Cluster 26:** $-ax = aa,\ x = \frac{aa}{-bx}$
- **Padrões de Alta Frequência**:
  - **Cluster 4:** $x - b = aa,\ x = aa - b$

  - **Cluster 13:** $x + b = -a,\ x + b - c = -a + b$

- **Padrões Muito Comuns**: Exemplos de Clusters:
  - **Cluster 11:** $aax = a,\ x = \frac{aa}{b}$
- **Padrões Extremamente Comuns**: Em geral, representam erros de cálculo
  - **Cluster 1:** $x = aa - b,\ x = aa$
  - **Cluster 3:** $x - bb = aa,\ x = -aa$

O fato de o DBSCAN não ter identificado clusters únicos pode ser atribuído as características do algoritmo e dos dados. Além disso, o DBSCAN cria um cluster especial identificado como -1, que agrupa todos os dados considerados "ruídos", ou seja, aqueles que não pertencem a nenhum cluster denso. Quando os dados são escassos ou distantes, o algoritmo pode identificar muitos pontos como ruído, sem alocar esses pontos em clusters específicos.

## 5.4 ANÁLISE DO SILHOUETE SCORE

Como métrica de avaliação os resultados obtidos foram:

- KModes: 0.9632164242942686
- BKMeans: 0.9632164132655501
- DBSCAN: 0.9392371231653901

Os Silhouette Scores dos três algoritmos indicam a qualidade do agrupamento de maneira geral. O KModes e o BKMeans apresentam pontuações quase iguais, o que sugere que ambos conseguem criar agrupamentos bem definidos, com boa coesão interna e separação entre os clusters. Esse desempenho é um indicativo de que esses algoritmos mantêm a qualidade do agrupamento consistentemente. Já o DBSCAN apresenta um desempenho ligeiramente inferior, indicando que, embora ainda produza agrupamentos bons, a qualidade é um pouco mais afetada em comparação aos outros dois. Esse desempenho inferior do DBSCAN pode ter sido causado por uma má escolha de parâmetros, como os valores de densidade e distância, que são sensíveis a ajustes e podem impactar a definição dos clusters, especialmente em dados com distribuição irregular ou complexa que podem representar clusters com diferentes densidades. Em resumo, KModes e BKMeans parecem ser robustos e estáveis, enquanto o DBSCAN pode exigir ajustes mais refinadas de seus parâmetros para manter a qualidade do agrupamento.

## 5.5 DISCUSSÃO

Os resultados demonstram que a utilização de árvores de expressão para representar as equações e a aplicação dos algoritmos DBSCAN, K-Modes e BKMeans permitiram a identificação de *misconceptions* de forma mais precisa e detalhada do que em trabalhos anteriores. A análise dos clusters gerados revelou padrões específicos de erros que podem ser utilizados para aprimorar o ensino de álgebra e o desenvolvimento de STIs mais eficazes.

É importante enfatizar que as limitações citadas por Gomes e Jaques (2020) foram na maioria resolvidas na abordagem atual, sendo elas:

1. **Representação inadequada de estruturas especiais**: Gomes e Jaques (2020) relataram dificuldades no tratamento de caracteres especiais como parênteses nas equações, resultando em estruturas inconsistentes nos clusters gerados. Esta limitação foi superada no presente trabalho através da adoção de árvores de expressão, que representam naturalmente a estrutura hierárquica das equações, incluindo parênteses e expressões complexas, sem ambiguidades. A notação pós-fixa utilizada na representação das árvores elimina a necessidade de tratamento especial para parênteses, preservando a estrutura lógica da equação.

2. **Ruído nos clusters**: O trabalho anterior também identificou a presença de resultados ruidosos em alguns clusters, que não representavam cenários reais ou *misconceptions* genuínas. A abordagem atual mitigou este problema com a representação através de árvores de expressão, que reduziu a variabilidade não significativa nos dados e com a implementação de algoritmos mais robustos.

3. **Limitação a um único algoritmo de clustering**: O presente trabalho expandiu significativamente esta abordagem, implementando e comparando três algoritmos distintos (DBSCAN, K-modes e BKMeans), cada um com características específicas que se complementam para uma identificação mais abrangente e robusta de *misconceptions*.

Estas melhorias metodológicas resultaram em clusters mais homogêneos e interpretáveis, como evidenciado pela qualidade dos agrupamentos (mensurada através do Silhouette Score) e pela maior correspondência entre os padrões identificados e as categorias teóricas de *misconceptions* estabelecidas na literatura.

A Figura 5.1 apresenta exemplos de clusters identificados no trabalho anterior (Gomes e Jaques, 2020). Ao comparar estes resultados com os clusters obtidos na abordagem atual (Tabelas 5.2, 5.4 e 5.6), podemos observar melhorias significativas em diversos aspectos. Os clusters anteriores (Figura 5.1) apresentavam frequentemente estruturas inconsistentes, com parênteses desbalanceados ou expressões mal formadas, como exemplificado pela equação "$x + bb - ccx) = aaa + bbb$"que contém um parêntese de fechamento sem o correspondente de abertura. Em contraste, os clusters obtidos com a representação por árvores de expressão na abordagem atual são estruturalmente consistentes e bem formados.

Enquanto os clusters anteriores frequentemente agrupavam equações que representavam diferentes tipos de misconceptions, os clusters atuais demonstram maior homogeneidade semântica, agrupando consistentemente equações que refletem o mesmo padrão conceitual de erro. Por exemplo, o cluster KMODES 0 (Tabela 5.2) contém exclusivamente exemplos da mesma misconception: a aplicação incorreta de adição/subtração ao isolar variáveis.

Por fim, a abordagem anterior apresentava dificuldades particulares com frações e expressões algébricas complexas. A metodologia atual, especialmente através da representação por árvores de expressão, permite identificar *misconceptions* em construções algébricas mais sofisticadas, como demonstrado nos clusters relacionados à manipulação incorreta de frações complexas (por exemplo, KMODES 69 e DBSCAN 38).

Estas melhorias qualitativas nos clusters identificados são corroboradas pelos resultados quantitativos do Silhouette Score, que indicam maior coesão interna e melhor separação entre clusters na abordagem atual. A combinação de uma representação estruturalmente rica (árvores de expressão) com algoritmos de clustering apropriados resultou em uma identificação mais precisa e interpretável de *misconceptions* algébricas.

É importante enfatizar que as limitações citadas por Gomes e Jaques (2020) foram na maioria resolvidas. Com base nesse resultado, pode-se observar que o uso da árvore de expressão, além de resolver alguns problemas citados acima, não afeta o desempenho do algoritmo.

Figura 5.1: Resultados utilizados como base

| Result | | Evaluation |
|---|---|---|
| **Previous Step** | **Wrong Step** | |
| $x - b = aa$ | $x = -aa$ | Calculation Error |
| | $x = aa$ | |
| $x + b = -aa$ | $x = -a$ | |
| $x + b = aa$ | $x = a$ | |
| $x = aa - b$ | $x = aa$ | |
| $(aax + bb) - (cc - ddx)$ $= (-aa + bbx) - (-ccx + dd)$ | $aax + bb - cc + ddx$ $= -aa + bbx + ccx + dd$ | Sign Rules misconception |
| $a * (x - c) - \frac{dx}{e} = a * \frac{x - c}{d}$ | $ax - b - \frac{cx}{d} = \frac{ax}{b}$ | Polynomial Multiplication misconception |
| $-x - bb = \frac{aa}{b}$ | $-\frac{x}{b} = aa + bb$ | Inverse Operations misconception |
| $x + bb = -a$ | $x = -a + bb$ | |
| $ax + bb = ax + bb$ | $ax + bx = aa - bb$ | |
| $x + b = aa$ | $x + b - c = aa - b$ | Unknown Isolation misconception |
| $-x - bbb = -aaa$ | $-x - bbb + ccc = -aaa + bbb$ | |
| $ax - bb = aa$ | $ax + bb = aa + bb$ | |
| $aax + bb = aax - bb$ | $aax - bbx = -aa - bb$ | No misconception or error |
| $(aax + bb) - (cc - ddx)$ $= (-aa + bbx) - (-ccx + dd)$ | $x = \frac{a}{b}$ | Noisy |
| | $aax = -aa$ | |
| $\frac{x}{bb} - \frac{ccc}{ddd} = \frac{-x + bb}{cc}$ | $\frac{x}{bb} = -x$ | |

(Gomes e Jaques, 2020)

A distribuição dos clusters sugere que existem poucos conceitos extremamente comuns, um número moderado de conceitos muito comuns e uma variedade de conceitos menos frequentes e únicos. Essa estrutura pode indicar uma hierarquia de concepções, onde alguns conceitos são fundamentais e amplamente utilizados, enquanto outros são mais especializados ou específicos. É interessante notar que, embora os conceitos mais comuns representem a base do conhecimento, as concepções menos frequentes podem fornecer percepções válidas sobre abordagens alternativas ou erros comuns.

Para uma análise mais detalhada, recomenda-se examinar as características específicas de cada cluster, como a complexidade das equações, a diversidade das variáveis envolvidas e a similaridade entre as concepções agrupadas. Isso permitirá uma compreensão mais profunda das tendências e padrões nas concepções identificadas. Além disso, uma análise da dispersão dentro de cada cluster ajudaria a identificar potenciais outliers ou padrões não tão evidentes em uma primeira observação.

Em relação ao estado da arte da área, esses resultados representam um avanço ao apresentar uma nova abordagem para a identificação de misconceptions, que se baseia na

análise da estrutura das equações e não apenas dos valores numéricos. A utilização de árvores de expressão permite uma representação mais completa e precisa das equações, facilitando a identificação de padrões e a diferenciação entre erros e *misconceptions*. A aplicação de diferentes algoritmos de clustering permitiu uma análise mais robusta e abrangente dos dados, a comparação entre os resultados dos algoritmos contribuiu para a validação da metodologia.

Além disso, este trabalho contribui para o desenvolvimento de STIs mais eficazes, ao possibilitar a coleta de informações detalhadas sobre as dificuldades dos estudantes e permitir a personalização do ensino. A identificação das *misconceptions* mais comuns e a análise dos padrões de erros podem ser utilizadas para aprimorar o feedback oferecido pelos STIs, tornando-o mais direcionado e eficaz.

### 5.5.1 Relação com as categorias teóricas

(Holmes et al., 2013) propuseram uma classificação de *misconceptions* matemáticas em três categorias principais: *Misconceptions* de Vocabulário, Erros Computacionais e Convicções Errôneas. Ao analisar os clusters identificados pelos algoritmos de agrupamento em nosso estudo, é possível estabelecer conexões diretas com estas categorias teóricas, conforme demonstrado na Tabela 5.7.

Tabela 5.7: Mapeamento entre clusters identificados e as categorias teóricas de Holmes et al. (2013)

| **Holmes** | **Clusters** | **Misconception** | **Exemplo** |
|---|---|---|---|
| Misconceptions de Vocabulário | K-modes: 0, 77<br>BKMeans: 6, 72<br>DBSCAN: 0, 39 | Aplicação incorreta de adição/subtração | $x + b = aa$<br>$x = aa + b$ |
| | | Combinação incorreta de termos semelhantes | $aax - bb - cx = ax - b - cc$<br>$aax - bx = ax - b - cc - e$ |
| Erros Computacionais | K-modes: 75, 79<br>BKMeans: 92, 93<br>DBSCAN: 1, 11 | Má compreensão do isolamento da variável | $x = -a - b$<br>$x = a$ |
| | | Erro de cálculo simples | $x = -aa - b$<br>$x = aa$ |
| Convicções Errôneas | K-modes: 57, 69<br>BKMeans: 11, 73<br>DBSCAN: 38 | Manipulação incorreta de frações complexas | $(aax - bb)/c = aaax/b + c$<br>$(aax - bb)/c = aaax/b + c$ |

Holmes et al. (2013) definem *misconceptions* de vocabulário como aqueles relacionados ao uso inadequado de terminologia e uma compreensão deficiente do significado de termos ou conceitos específicos em matemática. Em nosso estudo, identificamos diversos clusters que se alinham perfeitamente com esta categoria. Por exemplo, nos clusters **K-modes 0** e **BKMeans 6**, observamos a aplicação incorreta de operações de adição e subtração, onde os estudantes falham em compreender o significado correto do sinal de igualdade e das operações inversas, mantendo o termo na mesma posição ao transferi-lo para o outro lado da equação ($x + b = aa \Rightarrow x = aa + b$).

Erros computacionais são definidos como falhas predominantemente de cálculo, como a omissão de um sinal negativo durante o processo de resolução. Esta categoria foi abundantemente representada em nossos clusters. Nos nossos resultados, observamos erros relacionados ao isolamento da variável, onde estudantes incorretamente calculam $x = -a - b$ como $x = a$, omitindo ou invertendo incorretamente os sinais. Estes erros não representam necessariamente

uma incompreensão profunda do conceito, mas sim falhas no processamento algorítmico das operações.

Já as convicções errôneas são falhas mais profundas na compreensão matemática, manifestam-se em nosso estudo principalmente nos clusters relacionados a operações complexas, como divisão e manipulação de frações. Por exemplo, a aplicação incorreta das regras de divisão observada no cluster **K-modes 57** demonstra uma falsa concepção sobre como a divisão interage com outros operadores na equação. Similarmente, a manipulação incorreta de frações complexas sugere uma incompreensão mais profunda das propriedades fundamentais das operações com frações.

Também é interessante notar que alguns clusters identificados em nosso estudo apresentam características que se sobrepõem a múltiplas categorias de Holmes, sugerindo que as *misconceptions* reais dos estudantes frequentemente combinam aspectos de diferentes categorias teóricas. Por exemplo, a má interpretação das regras de inversão ao isolar variáveis pode ser interpretada tanto como um erro computacional quanto como uma convicção errônea sobre as propriedades fundamentais das operações algébricas.

### 5.5.2 Análise com o estado da arte

Ao realizar uma análise comparativa entre os resultados obtidos neste trabalho e aqueles relatados na literatura atual, é possível identificar avanços significativos tanto na metodologia quanto na qualidade de identificação de *misconceptions* em álgebra. A principal contribuição metodológica deste estudo reside na combinação inovadora de árvores de expressão com múltiplos algoritmos de clusterização, uma abordagem que permitiu superar várias limitações presentes em trabalhos anteriores.

O trabalho de Feldman et al. (2018) apresentou uma metodologia para identificação de *misconceptions* baseada em síntese de programas, alcançando 70% de identificação das *misconceptions* conhecidas. No entanto, sua abordagem dependia exclusivamente de respostas finais e apresentava limitações ao assumir homogeneidade no padrão de resolução dentro de um mesmo grupo. Nossa metodologia, em contraste, analisa todos os passos intermediários da resolução, identificando *misconceptions* com maior precisão (com um Silhouette Score médio superior a 0,93 para todos os algoritmos testados) e sem depender de pressupostos sobre a homogeneidade das soluções dentro de um grupo.

Comparando com a abordagem de Andersson e Johansson (2015), que também utilizou técnicas de clusterização para identificar misconceptions, nosso trabalho apresenta duas vantagens significativas. Primeiro, enquanto eles utilizaram dados simulados para validar seu método, nossa abordagem empregou dados reais de estudantes interagindo com um STI em ambiente escolar. Segundo, seus resultados demonstraram uma capacidade limitada de identificação (três possíveis *misconceptions* após quarenta rodadas), enquanto nossa abordagem identificou dezenas de padrões distintos de *misconceptions* a partir de um único conjunto de dados.

Em relação ao trabalho de Elmadani et al. (2012), que utilizou mineração de regras de associação para identificar misconceptions, nossa metodologia oferece uma vantagem crucial: a completa automatização do processo. Enquanto ele ainda depende de análise humana para interpretar os resultados e confirmar a presença de misconceptions, nossa abordagem baseada em clusterização e árvores de expressão permite a identificação automática de padrões que caracterizam *misconceptions* sem intervenção humana, representando um avanço significativo para aplicações em larga escala.

Quando comparamos especificamente com o trabalho anterior Gomes e Jaques (2020), que serviu como base inicial para este estudo, as melhorias são particularmente notáveis. A principal limitação no trabalho anterior estava na representação das equações, que não tratava

adequadamente caracteres especiais como parênteses, resultando em estruturas inconsistentes nos clusters. Nossa abordagem baseada em árvores de expressão resolveu completamente essa limitação, resultando em clusters mais homogêneos e interpretáveis, como evidenciado pelo Silhouette Score. Mas este trabalho difere significativamente do anterior não apenas pela representação em árvores de expressão, mas também pela implementação de algoritmos mais robustos e adequados ao problema (DBSCAN e BKMeans), pela aplicação de técnicas avançadas de processamento de dados, pela comparação sistemática entre diferentes métodos de clustering, e pela análise mais aprofundada da relação entre os clusters identificados e as categorias teóricas de *misconceptions*. Esta abordagem multifacetada representa uma evolução substancial em relação ao trabalho anterior, permitindo identificar padrões mais precisos e significativos nas *misconceptions* algébricas.

Portanto, este trabalho avança o estado da arte na identificação automática de *misconceptions* matemáticos em vários aspectos fundamentais. Por exemplo, com a representação estrutural preservando a semântica e hierarquia das operações e facilitando a extração de características relevantes.

Esse trabalho também possibilita a diminuição da dependência de bibliotecas de erros pré-definidas, já que nossa metodologia permite capacidade de descoberta não supervisionada de falsas concepções.

### 5.5.3 Limitações e trabalhos futuros

Apesar dos avanços, é importante reconhecer as limitações remanescentes em nossa abordagem. Os clusters gerados, embora homogêneos, ainda requerem interpretação humana para a classificação final do tipo específico de *misconception*. Esta limitação aponta para a necessidade de desenvolvimento de métodos para a caracterização semântica dos clusters identificados.

Outro trabalho futuro é a integração de técnicas de processamento de linguagem natural para fornecer *feedback* personalizado baseado nas *misconceptions* identificadas. Essa integração completaria o ciclo de adaptação, permitindo que o sistema não apenas identifique padrões de erro, mas também comunique efetivamente ao estudante a sua dificuldade.

## 5.6 ACESSO AOS RESULTADOS COMPLETOS

Somente uma amostra representativa dos clusters identificados foi apresentada neste capítulo. No entanto, reconhecendo a importância da transparência científica e o valor potencial dos resultados completos para pesquisadores da área, todos os clusters gerados pelos três algoritmos, juntamente com seus exemplos e análises detalhadas, estão disponíveis para consulta.

Todo o código-fonte utilizado neste trabalho, juntamente com os conjuntos de dados (anonimizados) e os resultados completos em formato digital estão disponíveis no repositório GitHub: `https://github.com/joicecg/masters-identify-misconceptions`. Este repositório inclui os resultados completos de todos os clusters gerados pelos algoritmos K-Modes, BKMeans e DBSCAN, a implementação dos algoritmos de clusterização com as configurações utilizadas e o conjunto de dados processado e anonimizado.

# 6 CONCLUSÃO

As falsas concepções são comuns no ensino de matemática, onde os estudantes podem desenvolver uma compreensão incorreta de conceitos e princípios matemáticos por várias razões e representam um desafio significativo para educadores e alunos, dificultando a construção de um entendimento sólido dos conceitos matemáticos. Este estudo propõe e compara abordagens para identificar falsas concepções em matemática. As técnicas apresentadas visam fornecer uma maneira mais precisa e eficaz de identificar essas falsas concepções para auxiliar os alunos a superá-las.

Nesse trabalho foram exploradas três abordagens para o agrupamento de dados: o algoritmo *DBSCAN*, o *k-modes* e o *Breathing KMeans* (BKMeans), cada um com suas características específicas. A aplicação do *DBSCAN*, que se distingue por sua capacidade de lidar com ruídos; A técnica de *k-modes* demonstra ser eficaz na identificação de agrupamentos homogêneos, mas com algumas limitações ao lidar com dados únicos, o que pode comprometer a interpretação do resultado. Já o *BKMeans* é útil para ajustar dinamicamente o tamanho dos clusters, permitindo uma flexibilidade maior na adaptação ao formato dos dados.

Para a entrada dos algoritmos de clustering, as respostas dos estudantes são inicialmente padronizadas para garantir que todas as variáveis contribuam igualmente para o processo de agrupamento. Depois, são representadas por meio de árvores de expressão. A partir dessa representação, são extraídas características relevantes das equações, como variáveis, operadores e a complexidade estrutural, sendo então vetorizadas para criar um formato numérico adequado para os algoritmos de clustering.

A avaliação dos agrupamentos foi realizada com base no Silhouette Score, o que possibilitou medir a coesão e a separação dos clusters gerados, comparando as diferentes abordagens propostas.

Quanto à análise qualitativa dos agrupamentos, podemos observar que, embora os conceitos mais comuns representem a base do conhecimento dos estudantes, as concepções menos frequentes oferecem uma perspectiva valiosa sobre abordagens alternativas ou equívocos recorrentes.

Os algoritmos testados têm capacidade de interpretar qualquer conjunto de dados com equações de primeiro grau. Todos os trabalhos relacionados apresentados são limitados a uma quantidade ou tipo específico de entrada, limitando o uso da solução. A metodologia proposta não só superou as limitações de estudos anteriores, mas também proporcionou uma ferramenta útil para melhorar a compreensão das dificuldades algébricas dos estudantes, contribuindo com um avanço para o aprimoramento de sistemas tutores e a personalização do aprendizado.

Os resultados apresentados continuam em um formato bruto, sem nenhum tipo de análise feita pelo algoritmo, criando uma dependência da interpretação de um agente humano. Como próximas etapas, podemos implementar uma etapa de pós-processamento que consiga interpretar e analisar os clusters e seus *misconceptions*. A saída dessa etapa seria uma forma fácil para o usuário final, seja ele o aluno ou professor, conseguir entender o problema identificado. Com essa informação podemos apresentar dicas e materiais extras para o aluno compreender o conceito ou ainda mostrar a forma correta de resolver.

Além disso, como possível pesquisa futura, podemos avaliar a aplicação de técnicas de aprendizagem supervisionada para refinar a classificação de misconceptions, organizando por níveis de complexidade. Essa abordagem permitiria também avaliar a gravidades desses erros,

podendo levar ao desenvolvimento de estratégias de ensino mais personalizadas, adaptadas às necessidades de cada estudante.

# REFERÊNCIAS


Abimbola, I. (1988). The problem of terminology in the study of student conceptions in science. *Science education*, 72(2):175–84.

Aggarwal, C. C. e Reddy, C. K. (2014). *Data clustering*. Chapman & Hall/CRC, London.

Aleven, V., Roll, I., Mclaren, B. M. e Koedinger, K. R. (2011). Help seeking and intelligent tutoring systems: Theoretical perspectives and a step towards theoretical integration. *Help seeking in academic settings: Goals, groups, and contexts*, páginas 55–94.

Anderson, J. R., Corbett, A. T., Koedinger, K. R. e Pelletier, R. (1995). Cognitive tutors: Lessons learned. Em *The Journal of the Learning Sciences*, volume 4, páginas 167–207. Taylor & Francis.

Andersson, J. e Johansson, H. (2015). Using clustering in a cognitive tutor to identify mathematical misconceptions. LU-CS-EX 2015-45, Lund University, Lund, Sweden. `https://lup.lub.lu.se/student-papers/record/5467510`. Acessado em: 2025-02-27.

Ashlock, R. B. (1990). *Error patterns in computation: A semi-programmed approach*. Merrill Publishing Company, Columbus.

Azevedo, O., Morais, F. e Jaques, P. A. (2018). Exploring gamification to prevent gaming the system and help refusal in tutoring systems. Em *European Conference on Technology Enhanced Learning (ECTEL)*, Leeds. Springer.

Baffes, P. T. (1994). *Automatic student modeling and bug library construction using theory refinement*. Tese de doutorado, The University of Texas at Austin, Austin.

Corbett, A. T., Koedinger, K. R. e Anderson, J. R. (2001). Cognitive tutors: Technology bringing learning science to the classroom. *The psychology of human learning*, 2:61–78.

Duffy, G. G. e Roehler, L. R. (1995). Strategic teaching frameworks: An approach to instruction in mixed-ability classrooms. *The Reading Teacher*, 48(6):534–543.

Elmadani, M., Mathews, M. e Mitrovic, A. (2012). Data-driven misconception discovery in constraint-based intelligent tutoring systems. Em *Proceedings of the 20th International Conference on Computers in Education*, Singapore. Asia-Pacific Society for Computers in Education.

Ester, M., Kriegel, H.-P., Sander, J. e Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Em *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, páginas 226–231, Portland. AAAI Press.

Fatio, N. A., Fatimah, S. e Rosjanuardi, R. (2020). The analysis of students' learning difficulties on system of linear equation in two variables topic. Em *Journal of Physics: Conference Series*, volume 1521. IOP Publishing.

Feldman, M. Q., Cho, J. Y., Ong, M., Gulwani, S., Popović, Z. e Andersen, E. (2018). Automatic diagnosis of students' misconceptions in k-8 mathematics. Em *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, página 264, Montréal. ACM.

Fritzke, B. (2020). Breathing k-means: Superior k-means solutions through dynamic k-values. *arXiv preprint arXiv:2006.15666*. https://arxiv.org/abs/2006.15666.

Fuchs, L. S., Newman-Gonchar, R., Schumacher, R., Dougherty, B., Bucka, N., Karp, K. S., Woodward, J., Clarke, B., Jordan, N. C., Gersten, R., Jayanthi, M., Keating, B. e Morgan, S. (2021). Assisting students struggling with mathematics: Intervention in the elementary grades. Relatório Técnico WWC 2021006, National Center for Education Evaluation and Regional Assistance (NCEE), Washington, DC.

Gomes, J. C. e Jaques, P. A. (2020). A data-driven approach for the identification of misconceptions in step-based tutoring systems. Em *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, páginas 1122–1131, Porto Alegre. SBC.

Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A. e Louwerse, M. M. (2005). Autotutor: A tutor with dialogue in natural language. *Behavioral Research Methods*, 37(2):180–193.

Holmes, V.-L., Miedema, C., Nieuwkoop, L. e Haugen, N. (2013). Data-driven intervention: correcting mathematics students' misconceptions, not mistakes. *The Mathematics Educator*, 23(1):24–44.

Huang, Z. (1997). A fast clustering algorithm to cluster very large categorical data sets in data mining. Em *Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, páginas 1–8, Tucson, Arizona. ACM Press.

Jabal, R. F. e Rosjanuardi, R. (2019). Identifying the secondary school students' misconceptions about number. Em *Journal of Physics: Conference Series*, volume 1157, página 042052. IOP Publishing.

Jaques, P. A., Lehmann, M. e Jaques, K. S. F. (2008). Avaliando a efetividade de um agente pedagógico animado emocional. Em *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 1, páginas 145–154.

Jaques, P. A., Seffrin, H., Rubi, G., De Morais, F., Ghilardi, C., Bittencourt, I. I. e Isotani, S. (2013). Rule-based expert systems to support step-by-step guidance in algebraic problem solving: The case of the tutor pat2math. *Expert Systems with Applications*, 40(14):5456–5465.

Khan, K., Rehman, S. U., Aziz, K., Fong, S. e Sarasvady, S. (2014). Dbscan: Past, present and future. Em *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, páginas 232–238. IEEE.

Knuth, D. E. (1998). *The Art of Computer Programming*. Addison-Wesley Professional, Boston.

Küçük, B. et al. (2011). Identifying the secondary school students' misconceptions about functions. *Procedia-Social and Behavioral Sciences*, 15:3837–3842.

Lample, G. e Charton, F. (2019). Deep learning for symbolic mathematics. arXiv preprint, arXiv:1912.01412, https://doi.org/10.48550/arXiv.1912.01412.

Leelawong, K. e Biswas, G. (2008). Designing learning by teaching agents: The betty's brain system. *International Journal of Artificial Intelligence in Education*, 18(3):181–208.

MEC (1997). Parâmetros curriculares nacionais: 5ª a 8ª séries. `http://portal.mec.gov.br/pnaes/195-secretarias-112877938/seb-educacao-basica-2007048997/12657-parametros-curriculares-nacionais-5o-a-8o-series`. Acessado em: 2019-12-04.

Nye, B., Graesser, A. e Hu, X. (2014). Autotutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, 24(4):427–469.

Ritter, S., Anderson, J. R., Koedinger, K. R. e Corbett, A. (2007). Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review*, 14(2):249–255.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Schmidt, H.-J. (1997). Students' misconceptions—looking for a pattern. *Science education*, 81(2):123–135.

Seffrin, H., Rubi, G. e Jaques, P. (2012). O modelo cognitivo do sistema tutor inteligente pat2math. Em *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 1, páginas 10–19.

Sison, R. e Shimura, M. (1998). Student modeling and machine learning. *International Journal of Artificial Intelligence in Education*, 9:128–158.

The Learning Agency Lab (2023). Math misconceptions: Mapping major math misunderstandings. `https://the-learning-agency-lab.com/the-learning-curve/math-misconceptions-mapping-major-math-misunderstandings/`. Acessado em: 2023-11-15.

Untarti, R. e Kusuma, A. B. (2019). Error identification in problem solving on multivariable calculus. Em *Journal of Physics: Conference Series*, volume 1188, página 012064. IOP Publishing.

Vanlehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221.

Vanlehn, K. (2016). Regulative loops, step loops and task loops. *International Journal of Artificial Intelligence in Education*, 2:107–112.

Ventura, M., Shute, V. J. e Kim, Y. J. (2014). Educational games for the classroom: The case of simcityedu. *International Journal of Gaming and Computer-Mediated Simulations*, 6(4):39–53.

Wang, J. (2009). *Encyclopedia of Data Warehousing and Mining*. IGI Global, Hershey, 2 edition.

Woolf, B. P. (2009). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann, Burlington.

Yasin, A. (2017). A review of research on the misconceptions in mathematics education. *Education Research Highlights in Mathematics, Science and Technology*, 2017:21–31.