



**UNIVERSIDADE FEDERAL DO
MARANHÃO**
Programa de Pós-Graduação em Ciência da
Computação

Antonio Carlos Ribeiro Raposo

***Análise da Curva de Aprendizagem de Alunos e
Capacidade Preditiva de Modelos no Ensino da
Programação***

**São Luís
2020**

Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Análise da Curva de Aprendizagem de Alunos e Capacidade Preditiva de Modelos no Ensino da Programação

Antonio Carlos Ribeiro Raposo

**São Luís - MA
2020**

Antonio Carlos Ribeiro Raposo

Análise da Curva de Aprendizagem de Alunos e Capacidade Preditiva de Modelos no Ensino da Programação

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFMA, como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Maranhão - UFMA

Centro de Ciências Exatas e Tecnologia - CCET

Programa de Pós-Graduação em Ciência da Computação - PPGCC

Orientador: Prof. Dr. Carlos de Salles Soares Neto

São Luís - MA

2020

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

Ribeiro Raposo, Antonio Carlos.

Análise da Curva de Aprendizagem de Alunos e Capacidade Preditiva de Modelos no Ensino da Programação / Antonio Carlos Ribeiro Raposo. - 2020.

61 p.

Orientador(a): Carlos de Salles Soares Neto.

Dissertação (Mestrado) - Programa de Pós-graduação em Ciência da Computação/ccet, Universidade Federal do Maranhão, ccet, 2020.

1. Curva de Aprendizagem. 2. Ensino de Algoritmos. 3. Módulo de Estudante. 4. Predição. I. Soares Neto, Carlos de Salles. II. Título.

Antonio Carlos Ribeiro Raposo

Análise da Curva de Aprendizagem de Alunos e Capacidade Preditiva de Modelos no Ensino da Programação

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFMA, como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Prof. Dr. Carlos de Salles Soares Neto
Orientador

**Prof. Dr. Mário Antonio Meireles
Teixeira**
Universidade Federal do Maranhão

Prof. Dr. Celso Alberto Saibel Santos
Universidade Federal do Espírito Santo

São Luís - MA

2020

À minha mãe, minha família e meus amigos, e a todos que me ajudaram nesta jornada.

Agradecimentos

Primeiramente agradeço a minha mãe, Silva Stela, que sempre me deu liberdade para tomar minhas próprias decisões e me incentivou a ser a pessoa que sou hoje, me dando carinho, afeto e estabilidade.

Agradeço ao meu irmão, que durante muito tempo foi meu ponto de referência e guia na vida.

Agradeço à minha namorada Alanna, que tem me ensinado que existe um lado mais simples e divertido da vida.

Agradeço também ao meu amigo e orientador, Carlos de Salles, por ter me apoiado e moldado no pesquisador que sou hoje. Ainda tenho muito o que aprender com você.

A todos os meus amigos, que me ajudaram diretamente ou indiretamente, sem vocês nada disso teria dado certo. Especialmente para Allan Cruz, que eu considero como meu segundo irmão, você é e sempre vai ser minha fonte de inspiração. Queria agradecer também a Thais, que eu considero como a irmã que eu nunca tive. A Ana Isabel por todo apoio e encorajamento que sempre me deu. Queria agradecer à Jullyanna, Jordan e Danilo, meus companheiros do Canadá que participaram dessa jornada junto comigo. Aos meus amigos da Liga pokemon, Breno, Frank, Vinicius, Tiago, entre outros. Aos meus amigos do Florata. Matheus, JV, PP, Cassiano, Vicente, Glauber, Andrade Susu. Aos meus amigos da Ludorama e do magic, com quem eu sempre divido grandes risadas.

Queria agradecer também à equipe da Hangar 18, principalmente para o Vinicius Maciel, esse último ano aprendi muito com vocês e espero que nossa jornada continue.

Agradeço também ao meu amigo Fernando, que nos deixou ano passado tão cedo. Você nunca será esquecido por nós meu amigo.

A todos aqueles que eu não mencionei ou me falhou a memória, me perdoem e meu muito obrigado.

*“A felicidade pode ser encontrada mesmo nas horas mais difíceis, se você lembrar de
acender a luz”
(Alvo Dumbledore)*

Resumo

Aprender algoritmos requer o conhecimento de fundamentos como abstração, decomposição e identificação de padrões. O processo de aprendizagem é modular com etapas que são bem definidas e pré-requisitos uma das outras, o que faz com que seja difícil para o professor identificar as necessidades do aluno. Além disso, a correção manual das tarefas dos alunos acaba sendo proibitiva do ponto de vista do tempo para os professores. Estes problemas são maximizados ao se ter uma turma com muitos alunos. Uma alternativa para amenizar tais problemas é o uso de Sistemas Tutores Inteligentes. A modelagem e predição do conhecimento é parte fundamental de um tutor inteligente, visto que é através dessas informações que o tutor toma suas decisões e adapta o ensino dos alunos. Este trabalho propõe-se a apresentar uma análise qualitativa entre os modelos de conhecimento de alunos no domínio da programação visando identificar as melhores abordagens para a modelagem e as peculiaridades do domínio. Uma das contribuições deste trabalho foi a coleta de dados no domínio da programação para a criação da base de dados COSMODS. Além disso, foi feita uma análise da curva de aprendizagem e capacidade preditiva dos modelos gerados através da metodologia proposta. A análise mostrou que os erros de compilação influenciam diretamente na representação do conhecimento dos alunos e capacidade preditiva dos modelos. Por fim, foi feita uma discussão dos impactos que os erros de compilação têm sobre os modelos gerados.

Palavras-chaves: Módulo de Estudante , Curva de Aprendizagem, Predição, Ensino de Algoritmos.

Abstract

Learning algorithms requires knowledge of fundamentals such as abstraction, decomposition and pattern identification. The learning process is modular with steps that are well defined and prerequisites for each other, which makes it difficult for the teacher to identify the student's needs. In addition, a manual correction of students' tasks is prohibitive for the teacher because of time restrictions. These problems are maximized for the class with many students. An alternative to alleviate such problems is the use of Intelligent Tutoring Systems. The modeling and prediction of knowledge is a fundamental part of an intelligent tutor, since it is through this information that the tutor makes his decisions and adapts the students' learning process. This work proposes to present a qualitative analysis between the knowledge models of students in the domain of programming aiming to identify the best approaches for modeling and the peculiarities of the domain. One of the contributions of this work was the collection of data in the programming domain for the creation of the COSMODS database. In addition, an analysis of the learning curve and predictive capacity of the models generated through the proposed methodology was made. The analysis showed that the compilation errors directly influence the representation of students' knowledge and predictive capacity of the models. Finally, there was a discussion of the impacts that compilation errors have on the generated models.

Keywords: Student Model, Learning Curve, Prediction, Teaching of algorithms.

Lista de ilustrações

Figura 1 – Arquitetura do COSMO	17
Figura 2 – <i>Dashboard</i> do COSMO	18
Figura 3 – Área de resposta do COSMO	19
Figura 4 – Interface Histórico de Atividades	19
Figura 5 – Arquitetura do COSMO	20
Figura 6 – Estrutura de um plugin	20
Figura 7 – Estrutura tradicional de um STI	21
Figura 8 – Modelo BKT	23
Figura 9 – Metodologia do Trabalho	27
Figura 10 – Etapas de um MSL	28
Figura 11 – Comparação do RMSE dos modelos	35
Figura 12 – Curva de Aprendizagem Vol 1 - Variáveis	41
Figura 13 – Curva de Aprendizagem Vol 1 - Condicionais	41
Figura 14 – Curva de Aprendizagem Vol 2 - Variáveis	41
Figura 15 – Curva de Aprendizagem Vol 2 - Condicionais	41
Figura 16 – Curva de Aprendizagem Vol 3 - Variáveis	42
Figura 17 – Curva de Aprendizagem Vol 3 - condicionais	42
Figura 18 – Curva de Aprendizagem Vol 1 - Variáveis	43
Figura 19 – Curva de Aprendizagem Vol 1 - Condicionais	43
Figura 20 – Curva de Aprendizagem Vol 3 - Variáveis	44
Figura 21 – Curva de Aprendizagem Vol 3 - Condicionais	44
Figura 22 – Curva de Aprendizagem Vol 1 - Conhecimento 1	45
Figura 23 – Curva de Aprendizagem Vol 1 - Conhecimento 2	45
Figura 24 – Curva de Aprendizagem Vol 2 - Conhecimento 1	46
Figura 25 – Curva de Aprendizagem Vol 2 - Conhecimento 2	46
Figura 26 – Curva de Aprendizagem Vol 3 - Conhecimento 1	46
Figura 27 – Curva de Aprendizagem Vol 3 - Conhecimento 2	46
Figura 28 – Questões Respondidas vs Notas	51
Figura 29 – Comparativo da curva de aprendizagem entre dataset COSMODS e COSMO-F	53

Lista de tabelas

Tabela 1 – Comparativo entre os Trabalhos	35
Tabela 2 – Resultado dos Modelos BKT-EM	47
Tabela 3 – Resultado dos Modelos BKT - CGD	48
Tabela 4 – Resultado dos modelos PFA	49
Tabela 5 – Tabela geral de RMSE	54
Tabela 6 – Tabela de acurácia dos modelos	55

Lista de abreviaturas e siglas

AVA - Ambiente Virtual de Aprendizagem

STI - Sistemas Tutorial Inteligente

BKT - *Bayesian Knowledge Tracing*

PFA - *Performance Factor Analysis*

$p(L_o)$ - Probabilidade de Conhecimento a Priori

$p(T)$ - Probabilidade de Transição

$p(S)$ - Probabilidade de *Slip*

$p(G)$ - Probabilidade de *Guess*

RMSE - *Root Mean Square Error*

EM - *Expectation Maximization*

CGD - *Conjugate Gradient Descent*

UFMA - Universidade Federal do Maranhão

MSL - Mapeamento Sistemático de Literatura

Sumário

1	INTRODUÇÃO	14
1.1	Objetivo	15
1.2	Organização do Trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Ambientes Virtuais de Aprendizagem	17
2.1.1	COSMO	17
2.2	Sistemas Tutores Inteligentes	21
2.3	Técnicas de modelagem de conhecimento	22
2.3.1	<i>Bayesian Knowledge Tracing</i>	22
2.3.2	Abordagens para ajuste dos parâmetros do modelo	24
2.3.3	Performance Factor Analysis	25
2.4	Métricas Para avaliação de modelos	26
2.4.1	<i>Root Mean Square Error</i>	26
3	METODOLOGIA	27
3.1	Levantamento Bibliográfico	27
3.1.1	Questões de Pesquisa	28
3.1.2	<i>String</i> de Pesquisa	29
3.1.3	Fonte de Pesquisa	29
3.1.4	Critério de Inclusão e Exclusão de artigos	30
3.1.5	Processo de Seleção de Artigos	30
3.1.6	Estratégia de Extração de Informação	30
3.2	Seleção dos Datasets	31
3.3	Seleção das Técnicas de Modelagem de Conhecimento	32
3.4	Coleta de Dados	32
3.5	Análise dos Modelos e Discussão dos Resultados	32
3.6	Resultados da Metodologia	33
3.6.1	Resultado do Levantamento Bibliográfico	33
3.6.2	Resultado da Seleção dos Datasets	36
3.6.3	Resultado da Seleção das técnicas de modelagem de conhecimento	36
3.6.4	Resultado da Coleta de Dados	37
3.6.4.1	Erros de Compilação	38
4	RESULTADOS E ANÁLISE DOS MODELOS	39
4.1	Análise da representação da curva de aprendizagem	39

4.1.1	COSMODS : Análise da curva de aprendizagem do Voluntário 1	40
4.1.2	COSMODS : Análise da curva de aprendizagem do Voluntário 2	40
4.1.3	COSMODS: Análise da curva de aprendizagem do Voluntário 3	42
4.1.4	COSMO-F : Análise da curva de aprendizagem do Voluntário 1	42
4.1.5	COSMO-F : Análise da curva de aprendizagem do Voluntário 3	43
4.1.6	KDD : Análise da curva de aprendizagem do Voluntário 1	44
4.1.7	KDD : Análise da curva de aprendizagem do Voluntário 2	45
4.1.8	KDD : Análise da curva de aprendizagem do Voluntário 3	46
4.2	Análise da capacidade preditiva dos modelos	47
4.2.1	BKT - <i>Expectation-Maximization</i>	47
4.2.2	BKT - <i>Conjugate Gradient Descent</i>	48
4.2.3	<i>Performance Factor Analysis</i>	49
5	DISCUSSÃO DE RESULTADOS	51
5.1	Coleta de Dados	51
5.2	Análise dos Modelos	52
5.2.1	Análise da Curva de Aprendizagem	53
5.2.2	Análise da Capacidade Preditiva dos Modelos	54
5.2.2.1	Discussão sobre os resultados de RMSE	54
5.2.2.2	Discussão sobre os resultados da Acurácia	55
5.3	Erros de Compilação	56
6	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	58
	REFERÊNCIAS	60

1 Introdução

O aprendizado de algoritmos é um tema complexo, com inúmeras iniciativas didáticas que tentam aperfeiçoar a experiência de aprendizes e professores ¹ (RIBEIRO et al., 2018). Aprender algoritmos requer o conhecimento de fundamentos como abstração, decomposição e identificação de padrões, os quais nem sempre são bem tratados em uma disciplina introdutória de algoritmos.

Além disso, o processo de aprendizagem de algoritmos é modular com etapas bem definidas, que são pré-requisitos umas das outras. O progresso dessas etapas é fundamental para o avanço do aluno durante o ensino. Isso torna a avaliação da curva de aprendizagem dos alunos por parte do professor mais difícil de se analisar, uma vez que a dificuldade do aluno pode não estar em entender um conceito novo e sim em ter falhado em entender um conceito antigo. O problema da análise da curva de aprendizagem é ainda maior com uma grande quantidade de alunos por turma, visto que o professor precisa fazer tal avaliação individualmente para cada aluno.

Adicionalmente, a correção manual das tarefas apresentadas pelos alunos acaba se tornando proibitiva, pois requer muito tempo do professor. Entretanto, essas tarefas permitem que o professor faça uma análise da curva de aprendizagem dos alunos de maneira mais prática e empírica.

Em suma, o processo de aprendizagem de algoritmos requer fundamentos que são complexos de serem assimilados. Além disso, a coleta e análise de dados não é trivial, visto que se trata de conhecimento modular e com pré-requisitos bem definidos. Uma possível solução para tal problema é o uso de Sistemas Tutores Inteligentes.

Sistemas Tutores Inteligentes (STI) são sistemas com propósitos educacionais que incorporam técnicas de inteligência artificial para dar suporte ao processo de aprendizagem. São sistemas que têm como objetivo prover instruções e *feedback* imediato para os estudantes (PSOTKA; MASSEY; MUTTER, 1988). Um STI é capaz de prover conteúdo personalizado e se adaptar as necessidades do estudante para maximizar a estratégia educacional (Ramesh; Rao; Ramanathan, 2015).

A estrutura típica de um STI é composta por quatro módulos: módulo de estudante, que é a representação do estado de conhecimento do aluno em tal momento; módulo de domínio, que representa as regras de um determinado domínio, módulo de tutor, que regula a comunicação entre domínio e estudante e apresenta as tomadas de decisões; e interface de usuário, que torna possível a interação do aluno com o sistema.

¹ <https://code.org/>

A modelagem e predição do conhecimento dos alunos é parte fundamental de um tutor inteligente (Käser et al., 2017). Em um STI, a decisão de qual tarefa o aluno deverá fazer é baseado em uma estimativa do conhecimento e predição do módulo de estudante. A capacidade preditiva e o comportamento do módulo de estudante influencia diretamente as decisões do modelo (ROLLINSON; BRUNSKILL, 2015).

Há poucos estudos que indicam qual o efeito que um domínio tem sobre a representação do conhecimento do aluno. O domínio da programação e do ensino de algoritmos apresentam uma media de tentativa/erro maior que outros domínios devido à natureza do ensino. Isso faz com que representação da curva de aprendizagem dos alunos não esteja de acordo com o que é observado em sala de aula, parcialmente invalidando as decisões de um STI.

Existem diversas técnicas para a modelagem de conhecimento de alunos. *Bayesian Knowledge Tracing* se mostrou como a técnica de modelagem de conhecimento mais popular quando se trata de tutores inteligentes (YUDELSON, 2016). *Performance Factor Analysis* se mostrou como uma alternativa para a representação de conhecimento dos alunos na literatura (JR; CEN; KOEDINGER, 2009).

Neste trabalho, propõe-se uma análise qualitativa entre os modelos de conhecimento de alunos no domínio da programação a fim de identificar possíveis peculiaridades e impacto que o domínio tem sobre a capacidade preditiva dos modelos e representação da curva do conhecimento dos alunos.

1.1 Objetivo

O objetivo central desse trabalho é fazer uma análise qualitativa entre modelos de conhecimento de alunos no domínio de programação e suas peculiaridades. Como objetivos específicos temos:

- Coleta de dados relacionados ao domínio de programação;
- Seleção dos algoritmos de modelagem aplicáveis ao contexto baseado na literatura;
- Análise da representação da curva de aprendizagem pelos modelos;
- Análise da capacidade preditiva dos modelos;

1.2 Organização do Trabalho

O trabalho segue a seguinte organização: O [Capítulo 2](#) apresenta os conceitos e técnicas de modelagem de conhecimento utilizados na elaboração da proposta deste

trabalho.

Capítulo 3 apresenta a metodologia utilizada para atingir o objetivo proposto por este trabalho. O capítulo descreve um mapeamento sistemático de literatura que foi feito para atingir a etapa de levantamento bibliográfico, além de descrever os critérios para seleção de técnicas de modelagem de conhecimento, datasets e análise para este trabalho.

Capítulo 4 descreve os resultados obtidos através da metodologia descrita no capítulo anterior. O capítulo apresenta o resultado do mapeamento sistemático de literatura, além de descrever os resultados da análise da curva de aprendizagem e capacidade preditiva dos modelos.

Capítulo 5 apresenta uma discussão dos resultados obtidos durante trabalho. O capítulo descreve uma discussão sobre a etapa da coleta de dados, análise da curva de aprendizagem e capacidade preditiva dos modelos e o impacto que os erros de compilação tem sobre tais modelos. Por fim, o **Capítulo 6** apresenta as conclusões do trabalho e o direcionamento para trabalhos futuros

2 Fundamentação Teórica

As seções a seguir apresentam conceitos chaves para este trabalho, como Ambientes virtuais de aprendizagem (AVA), Tutores Inteligentes e as técnicas de modelagem de conhecimento como *Bayesian Knowledge Tracing* e *Performance Factor Analysis*. Além disso, o ambiente virtual de aprendizagem COSMO é mais detalhado.

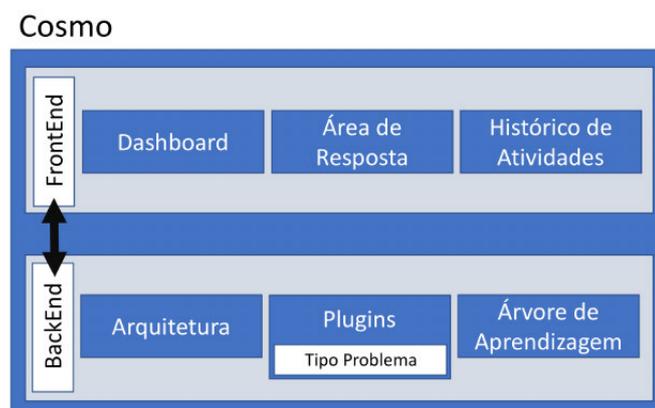
2.1 Ambientes Virtuais de Aprendizagem

Ambientes Virtuais de Aprendizagem são sistemas online, destinados a dar suporte a atividades através da tecnologia da educação e comunicação (ALMEIDA, 2003). Os AVAs ganharam espaço no ensino devido a sua capacidade de se adaptar às necessidades do aluno. Os AVAs permitem que o processo de aprendizado e também de ensino sejam mais personalizáveis (MEHLECKE; TAROUÇO, 2003).

2.1.1 COSMO

Cosmo é um AVA extensível a *plugins* focado no ensino e prática de algoritmos (RABÊLO-JÚNIOR et al., 2018). Ele foi desenvolvido pelo laboratório Telemidia-MA na Universidade Federal do Maranhão. O ambiente é composto por uma arquitetura tradicional com *Front-End* e *Back-end* interligada que possui 6 funcionalidades destacadas como mostra a Figura 1.

Figura 1 – Arquitetura do COSMO

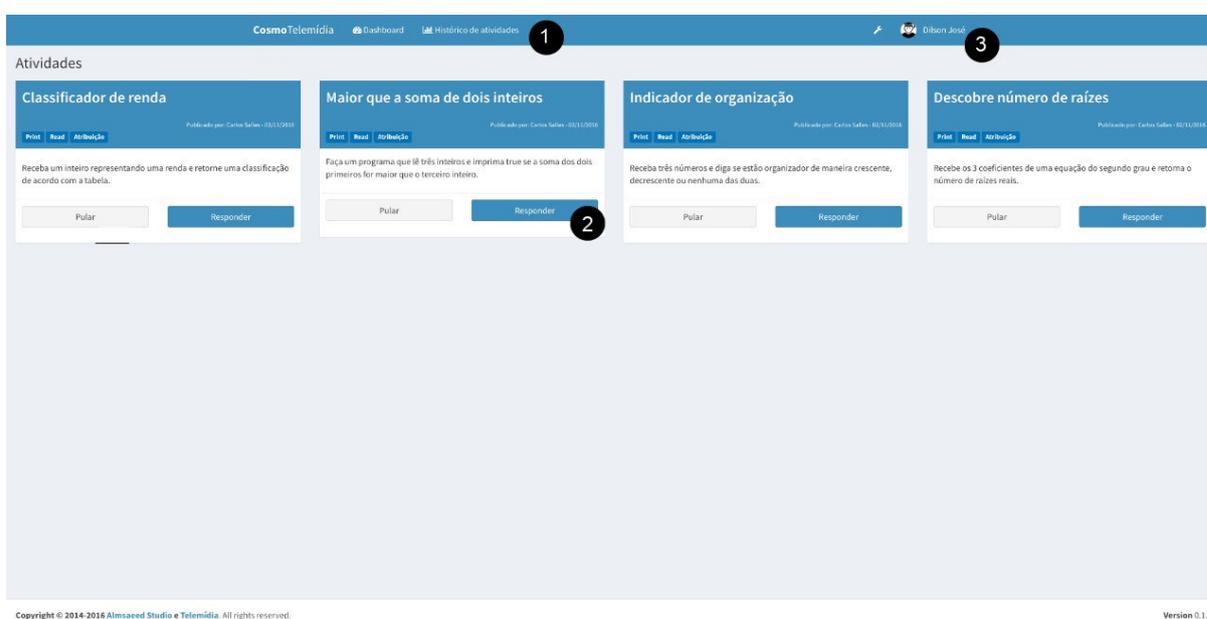


Fonte: (RABÊLO-JÚNIOR et al., 2018)

De acordo com (RABÊLO-JÚNIOR et al., 2018), as funcionalidades do COSMO são descritas da seguinte forma:

- Front-End

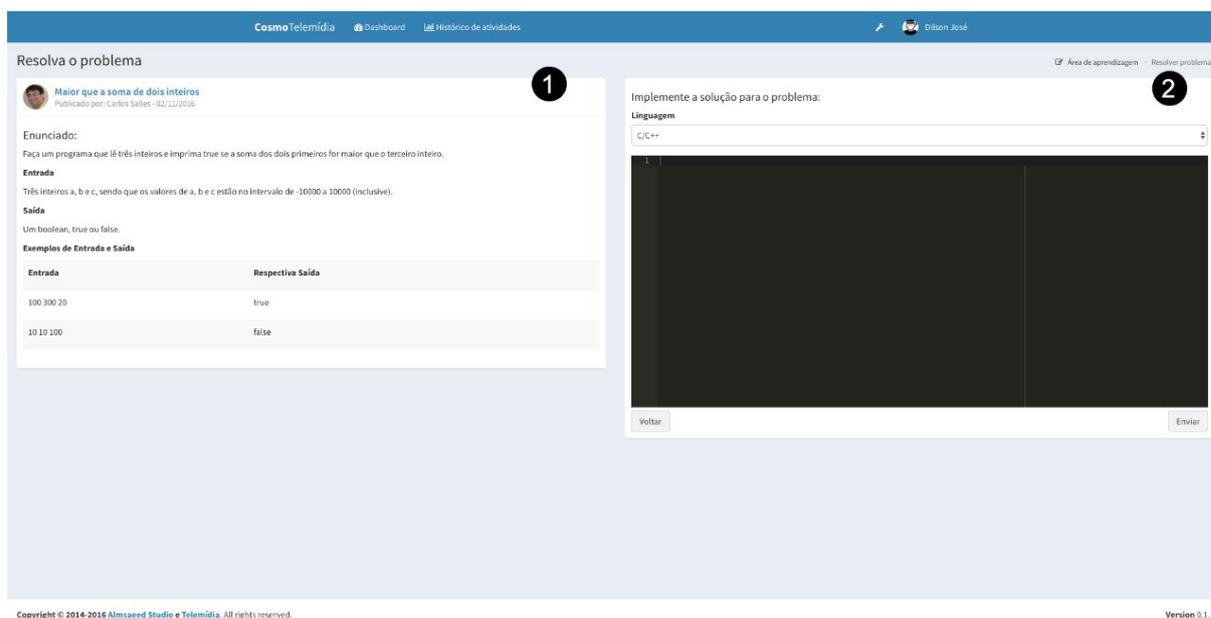
- *Dashboard* : É a funcionalidade responsável por apresentar as atividades para os alunos. No *Dashboard* são listadas quatro atividades aleatórias sobre assuntos definidos pela árvore de aprendizagem. A Figura 2 apresenta um exemplo de *dashboard* no COSMO organizada em 5 pontos. (1) é referente as caixas de atividades. (2) apresenta o botão de pular atividade, quando o aluno pressiona tal botão o sistema seleciona outra atividade para o aluno aleatoriamente respeitando a árvore de aprendizagem. (3) é o botão de perfil do usuário com suas informações. (4) a ação de responder, botão que leva a Área de Resposta. Por fim (5) apresenta o menu do usuário, com links para o *Dashboard* e Histórico de atividade.

Figura 2 – *Dashboard* do COSMO

Fonte: (RABÊLO-JÚNIOR et al., 2018)

- Área de Resposta: Área responsável por apresentar o problema a ser solucionado para o aluno. Ao clicar responder em uma das atividades no *dashboard*, o aluno é então levado para uma tela como na Figura 3. (1) corresponde ao enunciado da questão a ser resolvida, com possível entrada e saída para o problema. (2) apresenta onde o aluno apresenta a solução para o problema, podendo escolher qual linguagem a ser usada para o mesmo. O tipo de problema apresentado pela área de resposta é definido pelos *plugins* inclusos no ambiente que definem qual o tipo de problema.
- Histórico de Atividades: É a área reservada para apresentar o histórico de atividades já resolvidas pelo aluno no sistema. Além disso, essa área também

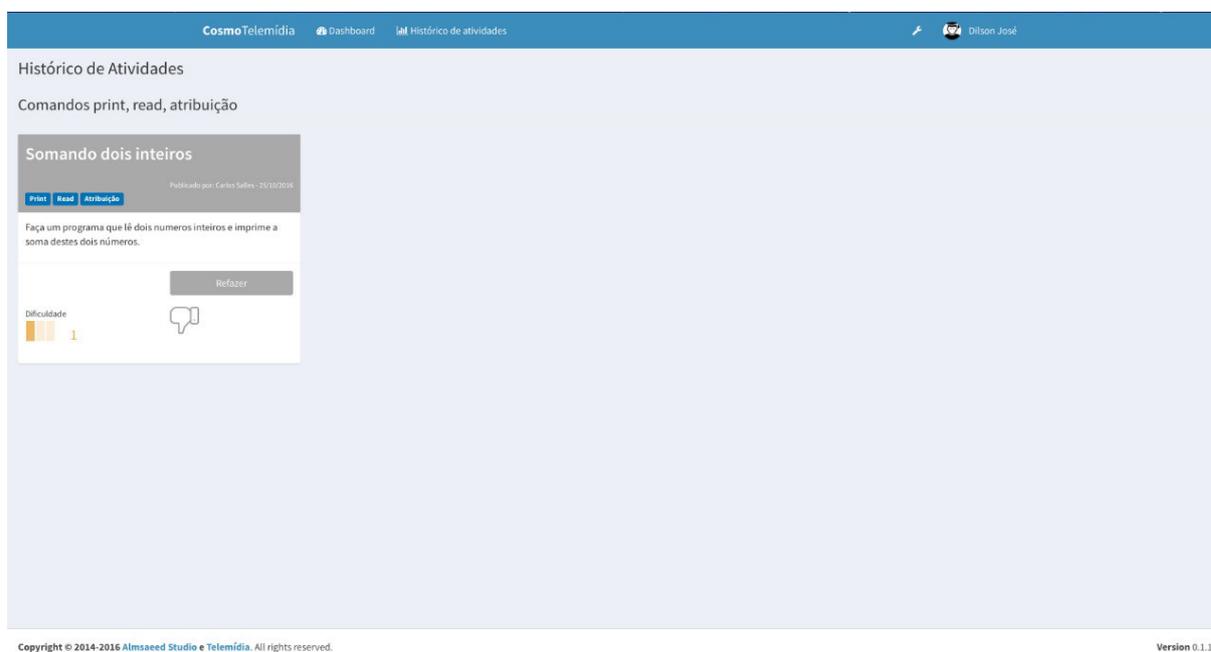
Figura 3 – Área de resposta do COSMO



Fonte: (RABÊLO-JÚNIOR et al., 2018)

permite que o aluno consiga responder novamente uma atividade. A Figura 4 apresenta a interface da funcionalidade de histórico de atividades no cosmo.

Figura 4 – Interface Histórico de Atividades



Fonte: (RABÊLO-JÚNIOR et al., 2018)

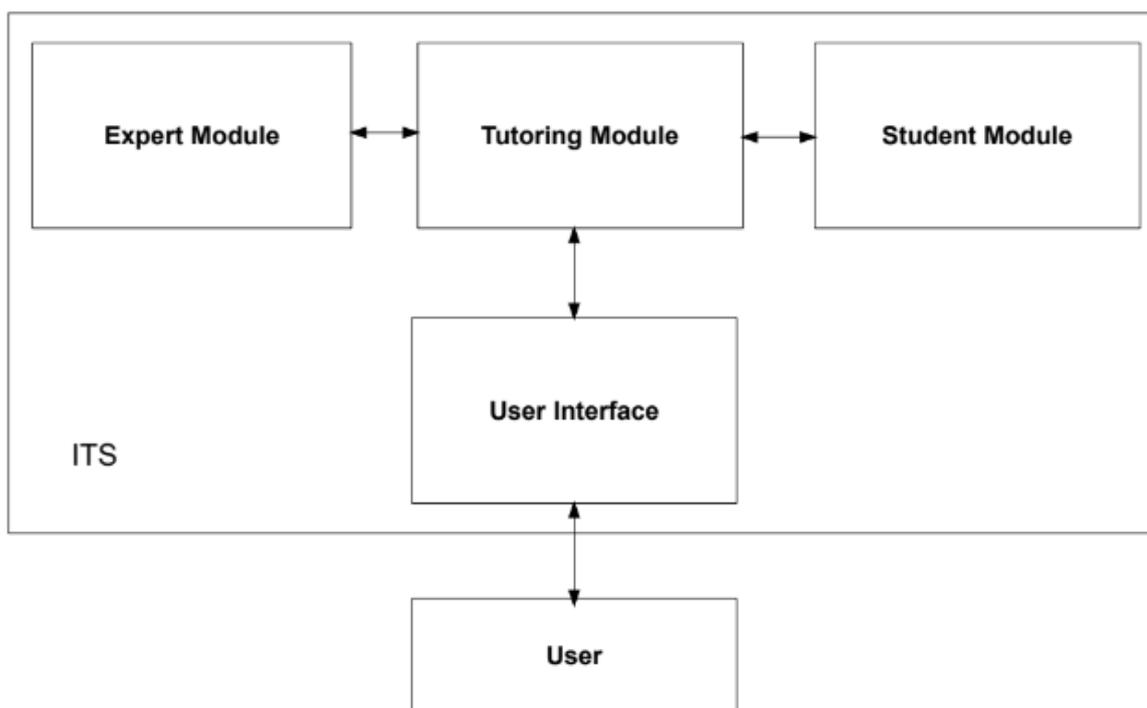
- *Back-End*

- Árvore de Aprendizagem: É responsável por armazenar e gerenciar o conteúdo de determinado conhecimento, aplicando as regras que foram modeladas pelo professor ao apresentar tal conteúdo para o aluno.

2.2 Sistemas Tutores Inteligentes

Sistemas tutores inteligentes (STI) são sistemas com propósitos educacionais que incorporam técnicas de inteligência artificial para dar suporte ao processo de aprendizagem. São sistemas que têm como objetivo prover instruções e *feedback* imediato para os estudantes (PSOTKA; MASSEY; MUTTER, 1988). Essas aplicações são capazes de prover conteúdo personalizado e adaptado às dificuldades e ao perfil dos estudantes, a fim de maximizar o sucesso da estratégia educacional (Ramesh; Rao; Ramanathan, 2015).

Figura 7 – Estrutura tradicional de um STI



Fonte: (Ramesh; Rao; Ramanathan, 2015)

A estrutura de um STI é tipicamente descrita na literatura como uma composição de quatro módulos: módulo de aluno (*Student Module*), que representa o estado de conhecimento do aluno em dado momento; módulo de domínio (*Expert Module*), que representa as regras de um determinado domínio; módulo de tutoria (*Tutoring Module*), que regula a comunicação entre o domínio e o estudante; módulo de usuário (*User Interface*), que torna possível a interação do aluno com o sistema (Ramesh; Rao; Ramanathan, 2015)(NWANA, 1990)(FREEDMAN; ALI; MCROY, 2000)(NKAMBOU; MIZOGUCHI;

BOURDEAU, 2010)(BRUSILOVSKIY, 1994). A Figura 7 apresenta uma visualização dos modelos e suas relações entre si.

O módulo de aluno é considerado componente principal de um STI. Ele é responsável pela representação do conhecimento do estudante. O STI cria essa representação utilizando técnicas de modelagem de conhecimento como *Bayesian Knowledge Tracing* (BKT) ou *Performance Factor Analysis* (PFA). Quando o módulo do aluno desvia do padrão estabelecido pela estratégia de ensino, é então ativada uma *flag* que indica para o módulo de tutor para que alguma ação seja tomada para que o aluno volte ou continue no caminho estabelecido pela estratégia de ensino. O módulo de aluno também pode ser baseado em restrições. Tutores baseados em *constraints* observam as respostas dos alunos e as comparam com as regras estabelecidas pelo modelo de domínio, se algo violar as regras ou *constraints* a resposta é considerada errada e o sistema dá um *feedback* mesmo estando certo ou errado (MITROVIC; MARTIN; SURAWEEERA, 2007)(MITROVIC, 2012).

O módulo de domínio é responsável pelas regras de negócio do conhecimento. Ele contém conceitos, regras e estratégias de ensino sobre o domínio a ser ensinado. O módulo de domínio pode ser construído baseado em *constraint-based modelling* (OHLSSON, 1994). Um módulo baseado em *constraint-based modelling* apresenta regras para a possível solução, se algo for contra as regras a solução é considerada errada (MITROVIC; OHLSSON, 2006).

Já o módulo de tutoria recebe informação tanto do módulo de domínio como do módulo de aluno. Baseado nas informações colhidas o tutor faz uma escolha para estratégia de ensino do aluno. Quando o aluno se desvia do previsto, o tutor então toma ações imediatas para melhorar a experiência do aluno.

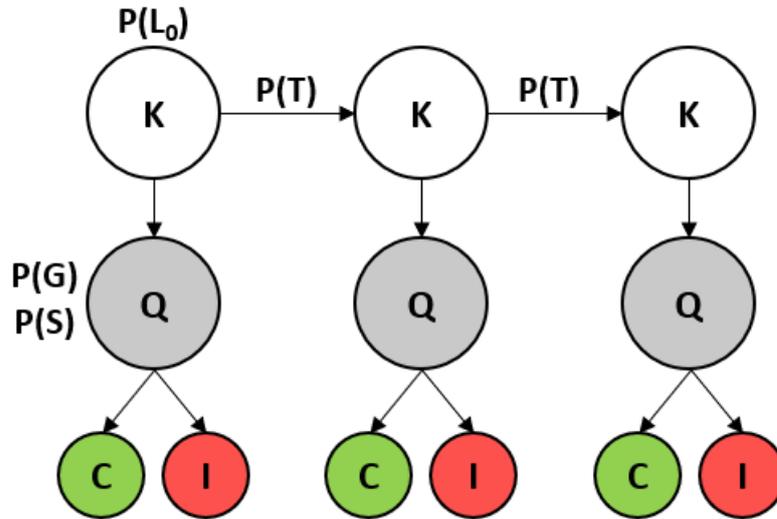
2.3 Técnicas de modelagem de conhecimento

Esta seção apresenta um aprofundamento das técnicas de modelagem de conhecimento utilizadas neste trabalho.

2.3.1 *Bayesian Knowledge Tracing*

Bayesian Knowledge Tracing é uma das técnicas de modelagem de conhecimento mais populares na literatura relacionada a tutores inteligentes (YUDELSON, 2016). Os modelos BKT são uma derivação de *Hidden Markov Models* e construído por variáveis latentes e observáveis (Käser et al., 2017). Variáveis latentes representam o conhecimento acerca de um determinado conteúdo, determinando se o aluno possui determinado conhecimento ou não. Já variáveis observáveis representam as respostas atribuídas pelos alunos a determinado conteúdo, que podem ser classificadas como corretas ou incorretas.

Figura 8 – Modelo BKT



O modelo básico do BKT é apresentado na Figura 8. Segundo (YUDELSON; KOEDINGER; GORDON, 2013), esse modelo é definido por quatro parâmetros seguintes: a probabilidade inicial de se possuir um determinado conhecimento K a priori, definida $p(L_0)$; a probabilidade de que a habilidade será aprendida em cada oportunidade de praticar o conhecimento, definida $p(T)$; a probabilidade do estudante cometer um erro quando já detém um determinado conhecimento, definida $p(S)$; e a probabilidade do estudante acertar uma questão ao aplicar uma habilidade ainda não conhecida, definida $p(G)$.

A variável Q corresponde à resposta atribuída por um aprendiz a uma determinada questão. Essas variáveis possuem valores binários, isto é, podem ser corretas ou incorretas. Com base nesses parâmetros, pode-se inferir quanto ao conhecimento de um estudante na n ésima oportunidade de aplicar um conhecimento, que é definido por $P(L_n)$.

De acordo com (DAVID; SEGAL; GAL, 2016), as hipóteses básicas desse modelo são: 1) o aprendiz é avaliado de forma dicotômica em cada oportunidade de aprendizado; e 2) o aprendiz pode estar no estado conhecido ou desconhecido para cada componente de conhecimento, e, uma vez aprendida, a habilidade jamais é esquecida.

Usando-se os parâmetros definidos acima, prever quando a resposta de um aprendiz está correta na oportunidade n pode ser calculada por meio da Lei das Probabilidades Totais, da seguinte forma:

$$P(\text{correto}_n) = P(L_n) \times (1 - P(S)) + (1 - P(L_n)) \times P(G) \quad (2.1)$$

De forma similar, a probabilidade de que a resposta seja incorreta ao tempo n pode ser calculada pela fórmula a seguir:

$$P(\text{incorreto}_n) = P(L_n) \times P(S) + (1 - P(L_n)) \times (1 - P(G)) \quad (2.2)$$

Segundo (DAVID; SEGAL; GAL, 2016), há dois estágios distintos ao se utilizar o modelo de conhecimento. O primeiro estágio consiste no ajuste dos parâmetros do modelo a partir dos dados, denominado de *fitting*. O segundo estágio consiste em usar o modelo para inferir o conhecimento do estudante ao longo do tempo com base em suas respostas. Dada a observação de uma resposta de estudante no tempo n (correta ou incorreta), a probabilidade $p(L_n)$ de que o estudante conheça a habilidade pode ser calculada pelo Teorema de Bayes. Assim, quando uma resposta correta é observada, a probabilidade pode ser obtida da seguinte forma:

$$P(L_n | \text{correto}_n) = \frac{P(L_n) \times (1 - P(S))}{P(\text{correto}_n)} \quad (2.3)$$

em que $P(\text{correto}_n)$ foi definida na Equação 1. Quando uma resposta incorreta é observada, a probabilidade é obtida da seguinte maneira:

$$P(L_n | \text{incorreto}_n) = \frac{P(L_n) \times P(S)}{P(\text{incorreto}_n)} \quad (2.4)$$

em que $P(\text{incorreto}_n)$ foi definida na Equação 2.

Por último, mostra-se como o conhecimento do estudante acerca do conteúdo pode ser atualizado dadas suas interações com o sistema. O cálculo é a soma de duas probabilidades: a probabilidade posterior de que o estudante já conheça a habilidade, e a probabilidade de que o estudante não detenha a habilidade, mas foi capaz de aprendê-la.

$$P(L_n) = P(L_{n-1} | \text{obs}_{n-1}) + (1 - P(L_{n-1} | \text{obs}_{n-1})) \times P(T) \quad (2.5)$$

2.3.2 Abordagens para ajuste dos parâmetros do modelo

O modelo tradicional do BKT é vulnerável ao problema estatístico da identificabilidade, isto é, uma mesma estimativa de desempenho pode ser resultado de várias combinações distintas de parâmetros do modelo (BECK; CHANG, 2007). Dessa forma, múltiplas combinações de $P(L_0)$, $P(T)$, $P(G)$ e $P(S)$ podem resultar em uma mesma taxa de erro para $P(\text{correto}_n)$.

Para resolver esse problema, O Trabalho de (BECK; CHANG, 2007) propôs avaliar as diferentes soluções para encontrar as mais plausíveis, restringindo os parâmetros do modelo por meio de uma probabilidade prévia obtida de todos os componentes de conhecimento. Contudo, segundo (BAKER; CORBETT; ALEVEN, 2008), essa solução pode levar a um novo problema, a degeneração do modelo, que é a obtenção de parâmetros que refletem um comportamento paradoxal.

Isso acontece quando $P(S)$ e $P(G)$ são maiores que 0.5. Quando $P(S) > 0.5$, o estudante que sabe um conteúdo tem maior chance de errar do que de acertar a resposta.

Por outro lado, quando $P(G) > 0.5$, o estudante que não sabe um conteúdo tem maior chance de acertar do que de errar a resposta.

Há diversas maneiras de se fazer o ajuste dos parâmetros para tentar resolver os problemas de degeneração e identificabilidade. A mais simples consiste na atribuição de valores fixos aos parâmetros $P(L_0)$, $P(T)$, $P(G)$ e $P(S)$ para cada componente de conhecimento, conforme (CORBETT; ANDERSON, 1994). Nesse caso, $P(S)$ e $P(G)$ devem receber valores menores que 0.3, como forma de evitar a degeneração do modelo.

Outras abordagens para ajuste dos parâmetros são: *Expectation-Maximization* (BECK; CHANG, 2007), *Conjugate Gradient Search* (CORBETT; ANDERSON, 1994), *Discretized Brute-Force Search* (YUDELSON; KOEDINGER; GORDON, 2013), *Contextual Guess-and-Slip* (BECK; CHANG, 2007) e *Individual Difference Weights* (CORBETT; ANDERSON, 1994).

2.3.3 Performance Factor Analysis

Performance Factor Analysis (PFA) é um modelo de regressão logística que consegue prever a acurácia baseada no número de fracassos e acertos do aluno anteriormente (Kurup; Joshi; Shekhokar, 2016)(JR; CEN; KOEDINGER, 2009). PFA é baseado em uma reconfiguração de *Learning Factor Analysis*(CEN; KOEDINGER; JUNKER, 2006).

$$m(i, j \in KCs, n) = \alpha_i + \sum_{j \in KCs} (\beta_j + \gamma_j, n_i, j) \quad (2.6)$$

$$p(m) = \frac{1}{1 + e^{-m}} \quad (2.7)$$

A forma padrão de *Learning Factor Analysis* (LFA) é mostrada na equação 2.6, onde m é um valor que representa o aprendizado acumulado do estudante i usando um ou mais KC(*knowledge components*) j . A dificuldade dos KCs são capturados pelo parâmetro β para cada KC. E a frequência de práticas anteriores para cada KC é a função de n observações para cada estudante i com KC j (adicionando γ para cada observação). A equação 2.7 é uma função logística usada para converter m em valores de predição probabilística. Esse modelo é equivalente à equação 1 com $\gamma = 0$ e apenas um valor para β (JR; CEN; KOEDINGER, 2009).

$$m(i, j \in KCs, f) = \sum_{j \in KCs} (\beta_j + \gamma_j S_{i,j} + \rho_j f_{i,j}) \quad (2.8)$$

Já PFA é uma variação de *Learning Factor Analysis* que utiliza a forma padrão do modelo de regressão logística com adição da performance do estudante como variável

dependente. A fórmula para PFA é bastante similar e é apresentada na equação 2.8. PFA reconfigura LFA nas variáveis independentes, removendo a variável do estudante α e mudando as variáveis KCs com a questão a ser respondida. EM PFA β não mais captura a dificuldade do KCs e sim da questão a ser resolvida. Esse modelo também adiciona dois parâmetros γ e ρ para cada habilidade refletindo os efeitos dos sucessos e falhas anteriores (GONG; BECK; HEFFERNAN, 2011).

Entretanto PFA não leva em consideração a ordem de acertos e erros do aluno ao fazer seu cálculo, o que pode vir a ser um problema. Um exemplo bem simples de se entender é se o aluno responder 4 questões, 2 corretas primeiramente e duas incorretas, e a ordem for invertida, o resultado da curva de aprendizado do aluno deverá ser possivelmente diferente, entretanto isso não acontece. Trabalhos como (GONG; BECK; HEFFERNAN, 2011) fazem uma proposta para adicionar um fator de envelhecimento para os dados, ou seja, quanto mais velho for o dado, menos impacto no cálculo ele deverá ter.

2.4 Métricas Para avaliação de modelos

Essa seção apresenta as métricas usadas para a avaliação de modelos nesse trabalho.

2.4.1 Root Mean Square Error

Root Mean Squared Error (RMSE) é frequentemente usado para medir as diferenças entre valores previstos por modelos e os valores observados. Segundo (CHAI; DRAXLER, 2014), RSME é bastante usado para medir a performance do modelo nas áreas de meteorologia, qualidade do ar e pesquisa de clima entre outras áreas.

RMSE também é bastante usado para a modelagem do conhecimento de estudantes. Diversos estudos como (YUDELSON; KOEDINGER; GORDON, 2013) e (GONG; BECK; HEFFERNAN, 2010) fazem uso de RMSE para avaliar os modelos gerados. Alguns estudos, como (PELÁNEK, 2015), questionam a capacidade de se analisar um modelo com um único valor numérico.

a formula de RMSE é dada a seguir :

$$RMSE = \sqrt{\sum_{i=1}^n \left(\frac{P - O}{n}\right)^2} \quad (2.9)$$

Onde P é o valor previsto, O é o valor observado e n é o numero de observações.

3 Metodologia

Este capítulo tem por objetivo apresentar a metodologia utilizada para alcançar os objetivos deste trabalho. A Figura 9 apresenta um detalhamento das etapas metodológicas.

Figura 9 – Metodologia do Trabalho



3.1 Levantamento Bibliográfico

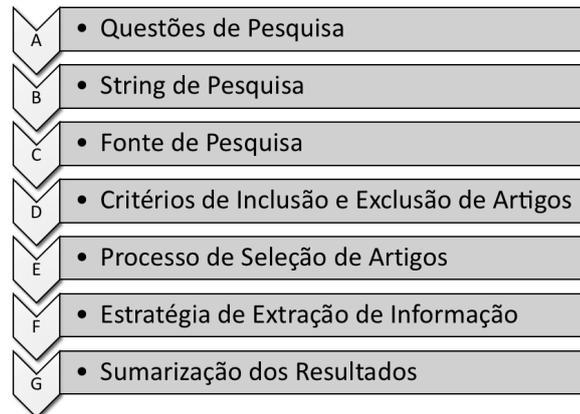
A primeira etapa do trabalho consiste de um levantamento bibliográfico do estado da arte. O método de mapeamento sistemático de literatura (MSL) foi escolhido para realizar tal tarefa.

Um MSL é um método de busca projetado para fornecer uma ampla visão geral de uma área de pesquisa (KEELE et al., 2007). As etapas de uma MSL são apresentadas na Figura 10.

Cada item das etapas da MSL possui um objetivo específico definido a seguir :

- **Questões de Pesquisa:** Nesta etapa são definidas as questões que direcionam a pesquisa. O resultado do mapeamento é definido por estas questões;

Figura 10 – Etapas de um MSL



- **String de Pesquisa:** A *string* de pesquisa é um conjunto de palavras-chave que compõem o tema de pesquisa, ela é utilizada como entrada no processo de mapeamento;
- **Fonte de Pesquisa:** Local onde se encontram artigos, periódicos e outros elementos de base acadêmica;
- **Critérios de Inclusão e Exclusão de Artigos:** A consulta de elementos nas fontes de pesquisa podem se tornar excessivas dependendo da *string* de pesquisa. Devido a isso é necessário que exista um critério de inclusão e exclusão de elementos com base nas questões de pesquisa;
- **Processo de Seleção de Artigos:** O processo de seleção pode ser executado de forma manual ou automática, baseado nos critérios de inclusão e exclusão;
- **Estratégia de Extração de Informação:** Esta etapa apresenta a estratégia para a extração de informação dos elementos obtidos através das etapas anteriores, além de classificar e sumarizar as características existentes nos elementos;
- **Sumarização dos Resultados:** É a consolidação do MSL, apresentando o resultado de todos os elementos levantados em resposta às questões de pesquisa;

3.1.1 Questões de Pesquisa

O desenvolvimento do MSL começa na etapa das Questões de Pesquisa. Através dela, é possível encontrar os artigos que deverão responder às questões ao final do MSL.

A concepção das questões foi definida de forma empírica, a partir do entendimento até então obtido sobre o tema. As questões de pesquisa podem ser observadas a seguir:

- Quais são as mais populares técnicas de modelagem de conhecimento da literatura;

- Quais são os trabalhos de modelagem de conhecimento com foco no ensino de algoritmos;
- Quais as métricas de análise de modelos de conhecimento mais populares;

3.1.2 *String* de Pesquisa

Com as questões de pesquisa definidas, foram selecionadas palavras-chave que serão utilizadas nas fontes de pesquisa para buscar os elementos.

Um levantamento de possíveis palavras-chave foi realizado para ajudar a definir a *string* de pesquisa. Foram priorizadas palavras em inglês para englobar um maior número de elementos.

A partir dos critérios definidos, foram decididas as palavras chaves a seguir:

- *Intelligent Tutoring Systems*
- *Student Model*
- *Domain Model*
- *Smart Tutor*
- *E-Learning*
- *Virtual Learning Environment*
- *Learning Curve*
- *Bayesian Knowledge Tracing*
- *Adaptive*
- *Knowledge Tracing*

Com estas palavras-chave foi elaborada uma *string* de busca com auxílio do operador lógico de busca *AND* e *OR*, obtendo o seguinte resultado:

- ('Adaptive' **OR** 'Domain Model' **OR** 'Model' **OR** 'Student Modelling') **AND** ('BKT' **OR** 'Knowledge Tracing') **AND** ('Comparison' **OR** 'Learning Curve' **OR** 'Prediction')

3.1.3 Fonte de Pesquisa

A fonte de pesquisa escolhida para busca dos artigos científicos foi a base de dados *IEEE* e *ACM*, pois estes atendem às expectativas necessárias para a pesquisa.

3.1.4 Critério de Inclusão e Exclusão de artigos

Para a inclusão e exclusão dos artigos, são adotados alguns critérios que classificam os artigos como participantes ou não do mapeamento.

Para os critérios de inclusão foram adotadas as seguintes métricas:

- **CI1 - Artigo completo ou curto:** Estando este disponível eletronicamente na base de dados selecionada na etapa de fonte de pesquisa;
- **CI2 - Título:** De acordo com as palavras-chave utilizadas na *string* de busca, as mesmas devem estar contidas ou possuir afinidades com o tema;
- **CI3 - Resumo:** De acordo com as questões de pesquisa, o resumo deverá possuir informações relevantes sobre o tema;
- **CI4 - Leitura Completa:** Deverá contemplar o que foi discutido no resumo e estar de acordo com as questões de pesquisa;

Para os critérios de exclusão foram adotados os seguintes critérios:

- **CE1 - Título**
- **CE2 - Resumo**
- **CE3 - Estudos que não tenham relação com objetivo do MSL.**
- **CE4 - Estudos repetidos de mesmo(s) autor(es);**

3.1.5 Processo de Seleção de Artigos

O processo de seleção de artigos é contemplado em etapas, sendo cada etapa um dos critérios de inclusão e exclusão definidos na etapa [3.1.4](#).

3.1.6 Estratégia de Extração de Informação

Os artigos selecionados devem ser submetidos ao processo de extração de dados. Foi definido um formulário de extração para coletar informações relevantes de cada artigo de maneira sistemática. A seguir tem-se cada campo do formulário e o que deve ser preenchido.

- **Nome:** Nome do Artigo.
- **Autor:** Nome dos Autores dos Artigos.
- **Ano:** Ano de publicação do Artigo.

- **Local da Publicação:** Local da publicação do artigo.
- **Modelagem de conhecimento:** Quais são as técnicas de modelagem de conhecimento o artigo utiliza?
- **Datasets:** Quais são os datasets utilizado no artigo, são datasets abertos?
- **Contribuição:** Qual a contribuição que o artigo? O que o artigo faz diferente de outros trabalhos?
- **Métricas:** O artigo apresenta métricas de análise? Se sim, quais são as métricas.

3.2 Seleção dos Datasets

Com o resultado do levantamento bibliográfico apresentado na etapa anterior, diversos datasets são identificados para serem utilizados neste trabalho. O objetivo dessa etapa é estabelecer critérios para a seleção de tais datasets.

Para este trabalho, a seleção de datasets se dá através de três critérios: tamanho; domínio e popularidade.

O critério tamanho é definido pela quantidade de entradas em determinado dataset, divididos em três categorias: pequenos, médios e grandes. Datasets pequenos são os possuem entre até 6 mil entradas. Já datasets médios são datasets que possuem entre 6 mil a 100 mil entradas. Datasets grandes são os que possuem mais de 100 mil entradas. Para este trabalho deve ser selecionado pelo menos um dataset de cada categoria.

Para a seleção dos datasets, o critério de domínio é definido pela proximidade do conteúdo do dataset com o domínio da programação. É dada preferência para datasets que possuem domínio voltado para ciência da computação e suas variações. Apenas em caso ausência de datasets com domínio na programação, são selecionados datasets de outros domínios.

Além disso, é selecionado um dataset que não possua afinidade com o domínio da programação a fim de identificar possíveis peculiaridades nos modelos gerados no domínio da programação.

O critério de popularidade é definido pela quantidade de trabalhos na etapa de levantamento bibliográfico (Seção 3.1) que fazem uso de determinado dataset.

Os resultados da etapa de seleção de datasets são apresentados na seção 3.6.2 deste trabalho.

3.3 Seleção das Técnicas de Modelagem de Conhecimento

A etapa de seleção das técnicas de modelagem de conhecimento utiliza as informações obtidas através da etapa de levantamento bibliográfico e é feita em paralelo a etapa de seleção de datasets.

A seleção das técnicas de modelagem de conhecimento que são utilizadas nesse trabalho se dá através da popularidade e importância das mesmas dentro dos trabalhos selecionados na etapa de levantamento bibliográfico.

Como critério de seleção, são consideradas as técnicas que aparecem com mais frequência e que tiveram um maior impacto dentro dos trabalhos selecionados. Tais técnicas foram obtidas através do levantamento bibliográfico realizado na etapa anterior.

Os resultados da etapa de seleção das técnicas de modelagem de conhecimento são apresentadas na Seção 3.6.3 deste trabalho

3.4 Coleta de Dados

Durante a aplicação da metodologia utilizada neste trabalho, a etapa de seleção de dataset falhou em apresentar uma base de dados de alunos no domínio de Algoritmos e Programação que faça uso de problemas programáveis para serem utilizadas neste trabalho. Para atingir o objetivo deste trabalho é então necessário que seja feita uma coleta de dados para a criação de um dataset com domínio focado no ensino de Algoritmos e Programação.

Para a coleta de dados, foi selecionado o ambiente virtual de aprendizagem COSMO como plataforma de ensino e fonte de dados. Para viabilizar os dados coletados, o COSMO foi utilizado em sala de aula nas disciplinas de Algoritmos I no curso de Ciência da Computação na Universidade Federal do Maranhão (UFMA) por um período de dois anos.

Os dados coletados em sala de aula são então compilados criando, assim, um dataset que é chamado de COSMODS e usado na modelagem de conhecimento dos alunos.

Os resultados da coleta de dados são apresentados na Seção 3.6.4 deste trabalho.

3.5 Análise dos Modelos e Discussão dos Resultados

Para analisar a qualidade dos modelos gerados, será feita uma análise da curva de aprendizagem dos alunos e da capacidade preditiva de tais modelos.

A análise da curva de aprendizagem é feita individualmente para cada aluno com a técnica de modelagem que melhor se adequar para a representação. É feito, então, um comparativo da representação da curva de aprendizagem de cada aluno para cada dataset.

A análise da capacidade preditiva é feita através do comparativo de métricas de predição de modelo como *root mean square error* (RMSE) e a acurácia literal para todos as técnicas apresentadas neste trabalho.

O resultado da análise dos modelos são apresentados no Capítulo 4 deste trabalho.

Já o Capítulo 5 apresenta a discussão dos resultados e as contribuições deste trabalho, além dos achados de pesquisa.

3.6 Resultados da Metodologia

Esta seção apresenta os resultados obtidos com a aplicação da metodologia proposta.

3.6.1 Resultado do Levantamento Bibliográfico

Esta subseção apresenta os resultados do levantamento bibliográfico obtidos através de um mapeamento sistemático de literatura.

O trabalho (JR; CEN; KOEDINGER, 2009) apresenta uma nova alternativa para *Knowledge Tracing* (KT). O trabalho argumenta que KT já vem sendo utilizado em várias formas em sistemas adaptativos. Entretanto é difícil utilizar KT para buscas em componentes de domínio e onde múltiplas habilidades estão envolvidas em um problema. O autor então propõem um novo modelo variante de *Learning Factor Analysis* (CEN; KOEDINGER; JUNKER, 2006) onde o ângulo de sua fórmula é baseado no desempenho do estudante, chamado de *Performance Factor Analysis*.

O trabalho faz um comparativo inicial entre os modelos PFA e BKT, com a conclusão que o PFA se mostra um método melhor que o BKT de maneira geral. Por outro lado, aponta que PFA não faz uso da ordem em que as atividades são feitas, o que pode ser um problema para o modelo.

O trabalho (GONG; BECK; HEFFERNAN, 2011) apresenta mais um comparativo entre PFA e BKT, onde se refaz o experimento de (JR; CEN; KOEDINGER, 2009) e vai além, indicando os problemas que cada modelo tem e apresenta possíveis soluções para tais problemas.

Segundo (GONG; BECK; HEFFERNAN, 2011), o BKT apresenta um problema relacionado a questões com múltiplas habilidades, uma vez que o algoritmo observa o histórico por habilidade. Uma maneira simples de solucionar o problema é dividir uma questão de múltiplas habilidades em diversas entradas de uma única habilidade.

Já para PFA, o modelo apresenta um problema relacionado a como ele observa os dados. O PFA leva em consideração os sucessos e fracassos passados, porém não considera

a ordem que os mesmos acontecem.

Para solucionar tal problema, o autor faz uma proposta de adicionar um fator de envelhecimento de dados ao longo do tempo, ou seja, quanto mais velhos forem os dados, menos impacto eles têm sobre o cálculo do algoritmo. As fórmulas são as seguintes:

$$sucessCount_t = \sum_{1 \leq k \leq t-1} P_k \delta^{t-1-k} \quad (3.1)$$

$$failureCount_t = \sum_{1 \leq k \leq t-1} |P_k - 1| \delta^{t-1-k} \quad (3.2)$$

Essas duas fórmulas substituem s e f nas equações originais do PFA onde as fórmulas são respectivamente representadas como:

$$s_{i,j} = \sum_{1 \leq k \leq t-1} P_k \quad (3.3)$$

$$f_{i,j} = \sum_{1 \leq k \leq t-1} |P_k - 1| \quad (3.4)$$

Ainda em (GONG; BECK; HEFFERNAN, 2011), se faz um comparativo entre várias métricas do BKT e PFA, comparando valores do PFA. Isso é feito com ou sem o fator de envelhecimento, e os valores de BKT com ou sem tratamento de múltiplas habilidades, além de comparar os resultados individuais de ambos os modelos.

O trabalho chega à conclusão de que PFA apresenta, para os dados utilizados no experimento, valores superiores quanto à capacidade preditiva dos modelos. Além disso, muitas decisões não foram cruciais para o desenvolvimento do trabalho, como separar questões de múltiplas habilidades no BKT.

Em (BAKER et al., 2011), por outro lado, se apresenta um comparativo entre as técnicas de modelagem tradicionais, como BKT e PFA. Há, ainda, o que o autor chama de conjunto de métodos, onde se é usado um conjunto de técnicas e métodos com pesos relativos à sua relevância para produzir a predição dos modelos. Entretanto, os resultados em geral não foram favoráveis para o conjunto de modelos. Os autores concluem que mais experimentos devem ser feitos com técnicas diferentes de modelagem para que se chegue em uma conclusão definitiva.

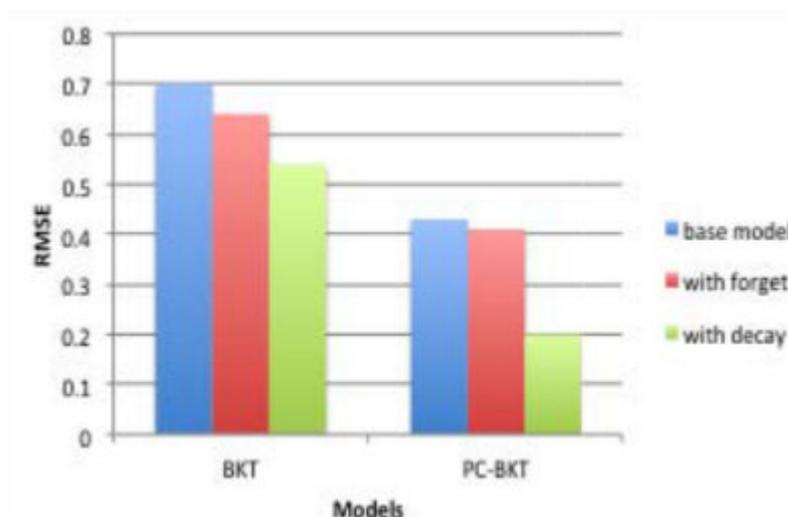
O trabalho (YUDELSON; KOEDINGER; GORDON, 2013) apresenta uma proposta de individualização dos parâmetros iniciais do BKT. O trabalho cria quatro variações individualização destes parâmetros iniciais do BKT e faz uma comparação entre o modelo tradicional e as quatro variações.

Ainda no mesmo artigo, os autores argumentam que a individualização dos parâmetros iniciais do BKT tem um resultado positivo sobre a sua acurácia. Os autores também concluem que adicionar individualização do parâmetro $p(Learn)$ apresenta resultados melhores do que a individualização do parâmetro $p(Init)$.

Outros trabalhos apresentam variações para o BKT e PFA. O trabalho (NEDUNGADI; REMYA, 2014) apresenta uma variação para o BKT chamada de PC-BKT (*Personalized Clustered Bayesian Knowledge Tracing*). PC-BKT utiliza conhecimentos anteriores sobre os alunos e as atividades para fazer uma clusterização dos alunos com base na mudança da capacidade de aprendizado para lidar com o problema de *frozen start*.

Além disso, uma continuação do trabalho (NEDUNGADI; REMYA, 2015) apresenta uma implementação da capacidade do aluno de esquecer um conteúdo. Os modelos BKT assumem que um aluno nunca consegue esquecer o conteúdo depois que é obtido. A Figura 11 apresenta um comparativo do RMSE entre o modelo BKT regular e PC-BKT com *forget* e *decay*.

Figura 11 – Comparação do RMSE dos modelos



Fonte: (NEDUNGADI; REMYA, 2015)

O trabalho conclui que o uso de *forget* e *decay* melhora consideravelmente a capacidade preditiva do modelo. Entretanto, é necessário um estudo futuro para determinar com quanto tempo o efeito do *decay* deve começar a fazer efeito na fórmula aplicada.

A Tabela 1 apresenta um resumo comparativo entre os trabalhos apresentados nesta seção.

Como visto na Tabela 1, não foi encontrado nenhum trabalho que faça uma análise das possíveis peculiaridades dos modelos gerados no domínio da programação.

Tabela 1 – Comparativo entre os Trabalhos

	BKT	PFA	RMSE	KDD CUP	<i>Programming</i>
(CEN; KOEDINGER; JUNKER, 2006)	X				
(JR; CEN; KOEDINGER, 2009)	X	X			
(GONG; BECK; HEFFERNAN, 2011)	X	X			
(BAKER et al., 2011)	X	X		X	
(NEDUNGADI; REMYA, 2014)	X		X	X	
(NEDUNGADI; REMYA, 2015)	X		X	X	
(Käser et al., 2017)	X	X	X	X	
Proposta deste Trabalho	X	X	X	X	X

3.6.2 Resultado da Seleção dos Datasets

Existem inúmeros datasets abertos de ambientes virtuais de aprendizagem em diversos domínios disponíveis para uso nesse trabalho. (Käser et al., 2017) realiza um experimento similar utilizando datasets como *Calcularis*, *Andes2*, *Cognitive Tutor*. A prioridade para seleção dos datasets para esse trabalho são três: tamanho; domínio e popularidade.

O dataset de álgebra KDD Cup 2010 (CUP, 2010) foi selecionado para este trabalho devido a seu tamanho. O Dataset contém cerca de 3300 alunos e aproximadamente 9 milhões de tentativas de resposta o que torna ele ideal para se fazer uma análise entre os modelos devido a sua diferença considerável de tamanho. Além disso, KDD Cup 2010 se mostrou o dataset mais popular aparecendo em diversas pesquisas encontradas na etapa de levantamento bibliográfico.

Além do dataset KDD CUP 2010, foi selecionado o dataset *Spanish Vocabulary Spring 2006* (KOEDINGER et al., 2010). O dataset *Spanish Vocabulary* é um dataset com domínio na língua espanhola e possui um total de 17000 tentativas de resposta com um total de 24 alunos. O dataset *Spanish Vocabulary Spring 2006* foi selecionado devido ao seu domínio ser em um espectro diferente ao do ensino de programação.

Devido a falha em encontrar um dataset com domínio no ensino de algoritmos e programação foi necessário a adição de mais uma etapa na metodologia para a coleta de dados em tal domínio a fim de criar um novo dataset a ser utilizado neste trabalho.

3.6.3 Resultado da Seleção das técnicas de modelagem de conhecimento

Durante o levantamento bibliográfico necessário para executar este trabalho, foram identificadas diversas técnicas de modelagem de conhecimento, cada uma com suas

vantagens e desvantagens. Entretanto, BKT E PFA (JR; CEN; KOEDINGER, 2009) se mostraram as mais populares para a modelagem de conhecimento de alunos. Diversos estudos como (GONG; BECK; HEFFERNAN, 2011)(Kurup; Joshi; Shekhokar, 2016)(JR; CEN; KOEDINGER, 2009) apontam essas técnicas como as mais aceitas pela comunidade acadêmica no âmbito de pesquisa de modelos de estudantes.

Com isso em mente foram selecionadas ambas as técnicas. Entretanto, BKT possui diversas variações relacionadas a maneira como o modelo faz o *fitting* de seus dados, existem diversas abordagens como *Expectation-Maximization* (BECK; CHANG, 2007), *Conjugate Gradient Search* (CORBETT; ANDERSON, 1994), *Discretized Brute-Force Search* (YUDELSON; KOEDINGER; GORDON, 2013), *Contextual Guess-and-Slip* (BECK; CHANG, 2007) e *Individual Difference Weights* (CORBETT; ANDERSON, 1994). Para este trabalho foram selecionados *expectation-maximization* (BECK; CHANG, 2007) e *Conjugate Gradient Search* (CORBETT; ANDERSON, 1994).

O detalhamento dessas técnicas são apresentadas no Capítulo 2 de fundamentação teórica deste trabalho.

3.6.4 Resultado da Coleta de Dados

Inicialmente, para este trabalho, foi feita uma coleta de dados em sala de aula utilizando a plataforma de ensino COSMO desenvolvida pelo laboratório TeleMidia-MA. O COSMO é um ambiente virtual de aprendizagem para o ensino de algoritmos e tarefas programadas.

Durante 4 semestres, o COSMO foi utilizado dentro de sala de aula com turmas do primeiro semestre do curso de Ciência da Computação na Universidade Federal do Maranhão(UFMA). Ele foi adotado pelo professor como prática e exercícios durante as aulas.

Os alunos tiveram acesso ao sistema em uma primeira aula. Durante esse período foi feito um acompanhamento e um tutorial sobre como utilizar o sistema. Os alunos então utilizam o sistema dentro de sala de aula, sobre supervisão, para programar tarefas que foram produzidas pelo próprio professor da disciplina. Após algumas semanas de uso do sistema, os alunos então ganharam acesso ao mesmo e podem usar eles de casa para praticar e responder novas questões.

Com isso, foram coletadas todas as possíveis interações que os alunos têm com o sistema, todas as tentativas de responder uma atividade, todos os logins, onde os alunos clicaram e onde deixaram de clicar. Com isso foi compilada uma quantidade significativa de dados e foi feita a criação do dataset COSMODS.

COSMODS contem dados coletados entre 2017 a 2019 durante quatro semestres na

Universidade Federal do Maranhão(UFMA). Ele conta com os dados de aproximadamente 90 alunos e um total de aproximadamente 7000 tentativas de respostas aos problemas disponíveis dentro do sistema. O dataset possui um total de 24 questões de programação relacionadas a 2 tipos de conhecimento: Variáveis e Condicionais.

O dataset possui um total de 8 tabelas sendo elas as seguintes : *Activities*, *AttemptActivities*, *CategoryActivities*, *Classes*, *GroupActivities*, *HistoryActivities*, *Log*, *User*. A tabela *Activities* apresenta as listas de questões dentro do sistema. *AttemptActivities* é a tabela que possui as tentativas de resposta dos alunos para determinada questão, essa tabela apresenta dados como qual aluno está respondendo qual questão e qual o resultado da tentativa, além de outros dados como o código do aluno e qual matéria pertence aquela questão. *Classes* apresenta a tabela com todas as turmas do COSMO. *GroupActivities* mostra quais os componentes de conhecimentos existem no sistema, como variáveis e condicionais. *HistoryActivities* é um log das tentativas de resposta dos alunos, como qual linguagem ele escolheu utilizar para responder um problema, quanto tempo o sistema demorou executando o código do aluno, etc. *Log* é um log completo de toda a interação que os alunos tiveram com o sistema. *User* é a tabela que contém informações dos alunos, como login e senha.

3.6.4.1 Erros de Compilação

O dataset COSMODS considera as respostas dos alunos apenas como certo ou errado. Uma parte importante do aprendizado em uma disciplina introdutória em algoritmos é aprender a lidar com o compilador, erros de compilação de código e boas práticas de programação, etc. No dataset COSMODS quando um aluno responde uma questão não há separação entre erro de compilação e erro de lógica.

Essa falta de separação faz com que o modelo se com porte de maneira inesperada e acurácia do modelo gerado caia drasticamente, uma vez que a resposta do aluno pode estar correta do ponto de vista algorítmico, porém ele cometeu um deslize e não verificou seu código com cuidado. Com isso em mente foi então feita a remoção de todos os erros de compilação do dataset COSMODS e criação de um segundo dataset que vamos chamar de COSMO-F para uma análise mais profunda do impacto que o erro de compilação tem sobre os modelos gerados em relação ao dataset COSMODS.

4 Resultados e Análise dos Modelos

Este Capítulo apresenta uma análise da representação da curva de aprendizagem e capacidade preditiva dos modelos BKT e PFA.

O objetivo deste trabalho é fazer uma análise qualitativa e comparativa dos modelos de conhecimento de estudante. Para isso foram criados modelos para cada dataset selecionado em 3.6.2 utilizando as técnicas de modelagens selecionadas em 3.6.3.

Para os modelos BKT foi utilizado a ferramenta HMM-scalable (YUDELSON, 2020) para a criação do modelo que utiliza *Expectation-Maximization* e *Conjugate Gradient Search*. Para o modelo PFA, foi recriado a regressão logística utilizada pelo trabalho (JR; CEN; KOEDINGER, 2009).

Foi feita então uma análise das curvas de aprendizagem individuais dos alunos de cada dataset para cada conhecimento utilizando a variável de transição $P(T)$ produzida pelo BKT. Entretanto, devido a grande quantidade de alunos e para evitar redundância na análise foram selecionados somente 3 alunos e 2 conhecimentos por dataset para este trabalho.

Para a análise da acurácia dos modelos gerados pelos métodos PFA e BKT foram utilizadas as métricas de *Root Mean Square Error* (RMSE) e acurácia real (valor previsto e valor real) do modelo.

4.1 Análise da representação da curva de aprendizagem

Esta seção apresenta os resultados principais dos modelos gerados, por meio dos dados obtidos e da análise individual dos modelos de conhecimento e curva de aprendizagem considerada mais relevante. A curva de aprendizagem dos alunos foi gerada utilizando a variável $P(T)$ dos modelos BKT.

Para esta análise foram escolhidos três voluntários de cada dataset para ilustrar essa coerência, situações de contorno, riscos e limitações que o emprego do BKT pode gerar. Para o dataset COSMO-F foram usados os mesmos voluntários utilizados no dataset COSMODS a fim de fazer uma comparação da curva de aprendizagem e apresentar os efeitos que os erros de compilação tem nas mesmas. Tais informações são detalhadas nas subseções seguintes. Os voluntários do dataset *Spanish Vocabulary* não foram utilizados para evitar redundância nos resultados, uma vez que apresenta resultados similares ao do dataset KDD CUP.

Além disso, este trabalho conta com acesso as notas dos alunos da disciplina de algoritmos e a utiliza para fazer um comparativo com a curva de aprendizagem apresentada.

4.1.1 COSMODS : Análise da curva de aprendizagem do Voluntário 1

O Voluntário 1 atingiu um total de 24 questões corretas, com um total de 139 tentativas. A Figura 12 apresenta a sua curva de aprendizagem para o conhecimento Variáveis, elaborada com base nas probabilidades de transição $P(T)$ calculadas pelo BKT para a componente de conhecimento (condicionais e variáveis). A probabilidade de conhecimento inicial $P(L_0)$ para o presente voluntário foi estimada em 0.00. Dessa forma, o modelo apresentou o comportamento esperado, pois a curva mostra que o voluntário atinge o ápice muito rapidamente, e há pequenas reduções quando responde às questões incorretamente.

Para a outra componente de conhecimento Condicionais, o modelo teve um comportamento diferente, porém dentro do esperado. Como se pode observar na Figura 13, o Voluntário 1, em seu início, respondeu várias questões de forma precisa, fazendo com que sua probabilidade de transição $P(T)$ se aproximasse do máximo. Entretanto, em determinado momento, o Voluntário 1 chegou a uma questão e não conseguiu resolvê-la, mesmo realizando diversas tentativas (especificamente 68 tentativas erradas apenas em uma questão, que ele nunca veio a acertar). Isso fez com que sua probabilidade de transição tivesse uma redução, estabilizando-se em 0.4.

Isso mostra que os modelos gerados pelo BKT se comportaram de maneira adequada. Quando o Voluntário 1 acerta a questão, a probabilidade de transição sobe de maneira exponencial. Porém, quando erra, a descida é gradual. É interessante perceber que a probabilidade de transição estabiliza em 0.4, pois representa que uma parcela do conhecimento adquirido anteriormente não é perdido, mesmo após uma série de respostas incorretas. É provável que, quando o voluntário voltar a acertar questões desse componente de conhecimento, sua probabilidade de transição volte a subir novamente.

Por fim, quando a curva de aprendizagem tiver uma queda substancial como a relatada, o tutor inteligente deve agir para melhorar a experiência de aprendizado e o desempenho do aluno naquele conteúdo, oferecendo dicas para a resolução da atividade ou sugestões de outras atividades.

4.1.2 COSMODS : Análise da curva de aprendizagem do Voluntário 2

O Voluntário 2 respondeu a um total de 16 questões em 28 tentativas, obtendo um dos melhores desempenhos em termos de questões por tentativa. A Figura 14 apresenta o gráfico de sua curva de aprendizagem utilizando as probabilidades de transição para as questões de variáveis. Devido ao seu comportamento incomum ocasionado pelo grande

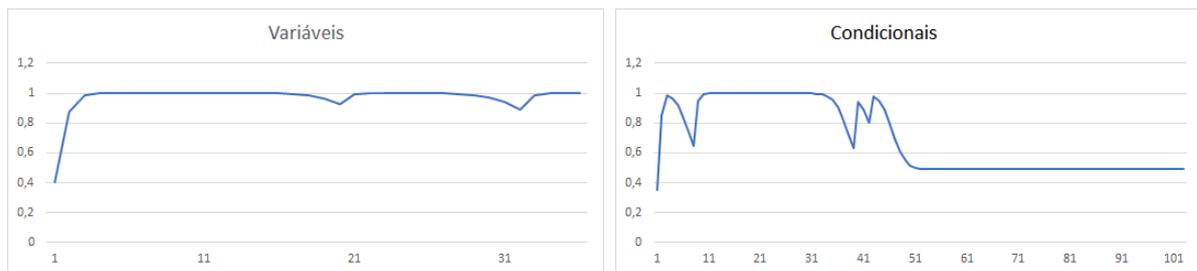


Figura 12 – Curva de Aprendizagem Vol 1 - Variáveis Figura 13 – Curva de Aprendizagem Vol 1 - Condicionais

numero de acertos em sequência, a sua curva de aprendizagem se aproxima do limite muito rapidamente, considerando que: 1) o aluno possuía algum conhecimento anterior $P(L_0) = 0.0$; e 2) o aluno assimilou o conhecimento necessário.

Para o conteúdo de condicionais, o Voluntário 2 realizou um total de 9 (nove) tentativas, das quais acertou 6 (seis) e errou 3 (três). A curva de aprendizagem estimada pelo modelo corresponde com os dados de uso do sistema, contudo, não reflete o comportamento do aluno em sala de aula, como se explicará adiante. Assim, seu nível de aprendizado teórico variou de 0.35, ao responder a primeira questão corretamente, a 1.00, a partir da quinta questão correta, como mostra a Figura 15.

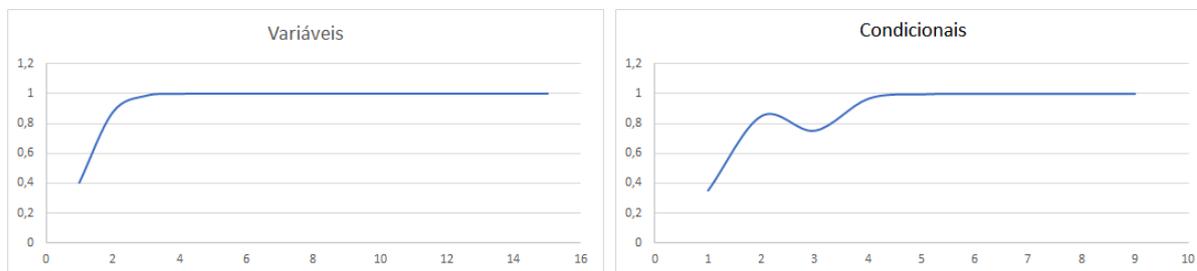


Figura 14 – Curva de Aprendizagem Vol 2 - Variáveis Figura 15 – Curva de Aprendizagem Vol 2 - Condicionais

Finalmente, deve-se alertar para o fato de que o Voluntário 2 apresentou uma das menores notas na avaliação da disciplina, tendo obtido 3.5, quando era esperado em torno de 7.0. Assim, com base nas estatísticas retiradas do modelo e no relatório de notas, pode-se concluir que o Voluntário 2 provavelmente teve acesso às respostas das questões (todas as suas submissões ocorreram apenas em horários das aulas de laboratório, onde estava próximo de outros alunos) e por isso não desenvolveu seu conhecimento como deveria. Informações como essas são importantes para que o professor possa atuar junto aos alunos, corrigindo os rumos do aprendizado.

4.1.3 COSMODS: Análise da curva de aprendizagem do Voluntário 3

O Voluntário 3 respondeu um total de 23 questões em 121 tentativas, obtendo um rendimento médio em termos de questões por tentativa. Para o tópico de comandos de variáveis, esse voluntário realizou 35 tentativas, obtendo acertos em 12 questões. A Figura 16 apresenta a curva de aprendizagem do estudante para esse conceito, mostrando que o voluntário teve dificuldades em duas questões desse assunto. De fato, ao analisar os registros de uso da plataforma, verifica-se que em duas questões o usuário precisou de aproximadamente 8 (oito) tentativas para chegar ao acerto.

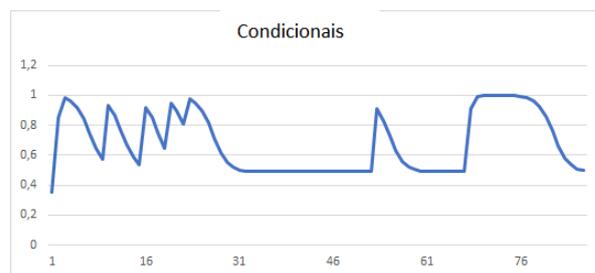


Figura 16 – Curva de Aprendizagem Vol 3 - Variáveis

Figura 17 – Curva de Aprendizagem Vol 3 - condicionais

Para o conteúdo de condicionais, o Voluntário 3 realizou um total de 86 (oitenta e seis) tentativas, tendo conseguido acertar 11 (onze) questões. Ao todo, o estudante realizou 75 (setenta e cinco) tentativas incorretas e não conseguiu acertar apenas 2 (duas) de 13 (treze) questões cadastradas para esse conceito. A sua curva de aprendizagem, mostrada na Figura 17, representa bem o comportamento desse estudante, com as oscilações típicas de quem está tendo contato com um assunto pela primeira vez. Assim, a curva mostra o declínio do nível de domínio daquele tópico quando ocorrem falhas sucessivas e a retomada quando os acertos voltam a ocorrer.

Por fim, verifica-se que, embora a probabilidade de transição para o conteúdo de condicionais tenha terminado em aproximadamente 0,5, o voluntário apresentou uma das melhores notas na primeira prova da disciplina, tirando 9,5 de um total de 10. Isso mostra a importância das tentativas e dos erros durante a etapa de consolidação de um conhecimento.

4.1.4 COSMO-F : Análise da curva de aprendizagem do Voluntário 1

O voluntário 1 atingiu um total de 24 questões corretas, com um total de 30 tentativas após a retirada dos erros de compilação cometidos por ele. A Figura 18 apresenta a sua nova curva de aprendizagem, que foi elaborada com base nas probabilidades de transição $P(T)$ calculadas pelo BKT para a componente de conhecimento (atribuições e variáveis). A probabilidade de conhecimento inicial $P(L_0)$ para o voluntário foi novamente

estimada em 0.00 mesmo após a remoção dos erros de compilação. Dessa forma, o modelo continua apresentando um comportamento esperado. A curva de aprendizagem do voluntário sobre rapidamente e os pequenos deslizes na sua curva causada pelos erros de compilação foram removidos. Tal representação gráfica reflete a realidade uma vez que o voluntário 1 apresentou um dos melhores resultados para tal conhecimento.

Para o outro componente de conhecimento condicionais, o modelo teve um resultado completamente diferente, porem ainda dentro do esperado. Como observado na Figura 19, o voluntário 1 em seu início respondeu várias questões de forma precisa, fazendo com que sua probabilidade de transição $P(T)$ se aproximasse do máximo rapidamente. Em comparação ao resultado obtido no dataset COSMODS apresentado na Figura 13, a nova curva de aprendizagem reflete mais naturalmente a curva de aprendizagem do voluntário 1. As 68 tentativas anteriores pelo aluno em uma única questão que ele nunca veio acertar acabou sendo somente erro de compilação que foi provavelmente causado por testes executados pela curiosidade dos alunos ou uma falta de entendimento sobre determinado aspecto do código.

Com isso em mente, as novas curvas de aprendizagem do voluntário 1 mostra que o modelo BKT gerado continua se comportando de maneira adequada para problemas computacionais. Quando o aluno acerta, a probabilidade de transição sobe de maneira exponencial. Porém, quando erra, a descida é gradual. É interessante perceber o efeito que os erros de compilação tem sobre a representação da curva de aprendizagem do aluno.

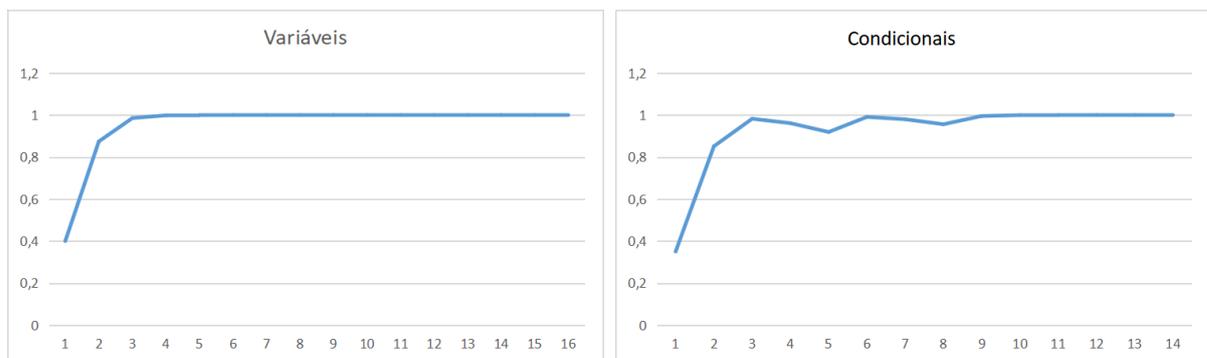


Figura 18 – Curva de Aprendizagem Vol 1 - Variáveis Figura 19 – Curva de Aprendizagem Vol 1 - Condicionais

4.1.5 COSMO-F : Análise da curva de aprendizagem do Voluntário 3

O voluntário 3 respondeu a um total de 23 questões em aproximadamente 41 tentativas após a retirada dos erros de compilação cometidos. Entretanto, ele respondeu a diversas questões várias vezes, o que fez com que a sua curva de aprendizagem não refletisse muito bem a realidade. Para conhecimento de variáveis, esse voluntario realizou 16 tentativas, obtendo acerto em 15 questões. A Figura 20 apresenta a nova curva de

aprendizagem do estudante para esse conceito, mostrando um resultado completamente diferente da Figura 16 após a remoção dos erros de compilação. Anteriormente foi identificado que ele precisou de 8 (oito) tentativas em duas questões para chegar ao acerto. Entretanto, as 8 tentativas foram identificadas como erro de compilação e não erro de lógica, o que faz com que a curva de aprendizagem não reflita a realidade, uma vez que o aluno entendeu o conceito necessário para resolução das questões.

Para o conteúdo de condicionais, o voluntário 3 acertou um total de 11 questões com 28 tentativas após a remoção dos erros de compilação. Entretanto, ele respondeu múltiplas questões mais de uma vez. A Figura 21 apresenta a nova curva de aprendizagem do voluntário, que é muito diferente da anterior mostrada na Figura 17. Entretanto, a nova curva de aprendizagem não reflete bem a realidade. Futuramente tem que se ter cuidado com as questões repetidas ao se calcular a curva de aprendizagem do aluno.

Vale lembrar que o voluntário 3 apresentou uma das melhores notas da turma de Algoritmos 1 na Universidade Federal do Maranhão, tirando 9,5 em um total de 10. A curva de aprendizagem gerada após a remoção dos erros de compilação reflete melhor o resultado do aluno em sala de aula, na prática.

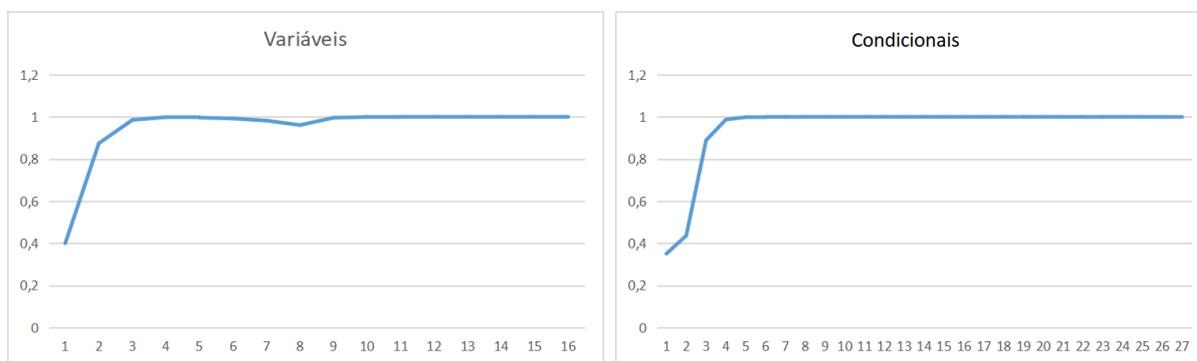


Figura 20 – Curva de Aprendizagem Vol 3 – Variáveis Figura 21 – Curva de Aprendizagem Vol 3 – Condicionais

4.1.6 KDD : Análise da curva de aprendizagem do Voluntário 1

O Voluntário 1 selecionado do dataset KDD para a análise atingiu um total de 16 questões corretas para o conhecimento 1 em um total de 23 tentativas. A Figura 22 apresenta a curva de aprendizagem, sendo elaborada com base nas probabilidades de transição $P(T)$ calculadas pelo BKT para este conhecimento. a $P(T)$ inicialmente tem uma elevação moderada devido ao fato das primeiras tentativas do voluntário terem sido incorretas, entretanto o $P(T)$ do voluntário continua a subir levemente representando o conhecimento adquirido a cada tentativa. Na sexta tentativa para o conhecimento o $P(T)$ tem uma subida exponencial chegando muito perto do 1 (valor absoluto). O Voluntário comete mais dois erros perto do final da curva porém não afeta a sua curva de aprendizagem,

assumindo que ele já adquiriu tal conhecimento. Com isso em mente, O comportamento da curva de aprendizagem do voluntário 1 não é diferente das outras curvas apresentadas até o momento neste, mostrando que é possível representar o conhecimento mesmo com uma quantidade de dados reduzidas, visto que a quantidade de dados nos outros datasets é consideravelmente menor.

Para o conhecimento 2, o voluntário acertou um total de 22 questões em 29 tentativas. A Figura 23 apresenta a curva de aprendizagem para o conhecimento 2. Inicialmente o $P(T)$ teve uma elevação moderada devido ao fato de ele errar 3 questões em sequência, após acertar uma única questão a sua curva de aprendizagem sobe drasticamente e conseqüentemente volta a descer drasticamente após ele errar mais uma sequência de 4 questões. Logo em seguida o $P(T)$ volta a subir e beira o valor máximo 1 (um).

A curva de aprendizagem do voluntário 1 para a base de dados KDD apresenta comportamento similar a curva de aprendizagem dos outros datasets, mostrando que não é necessário um número de dados muito grandes para representar o conhecimento de alunos dentro de um STI.

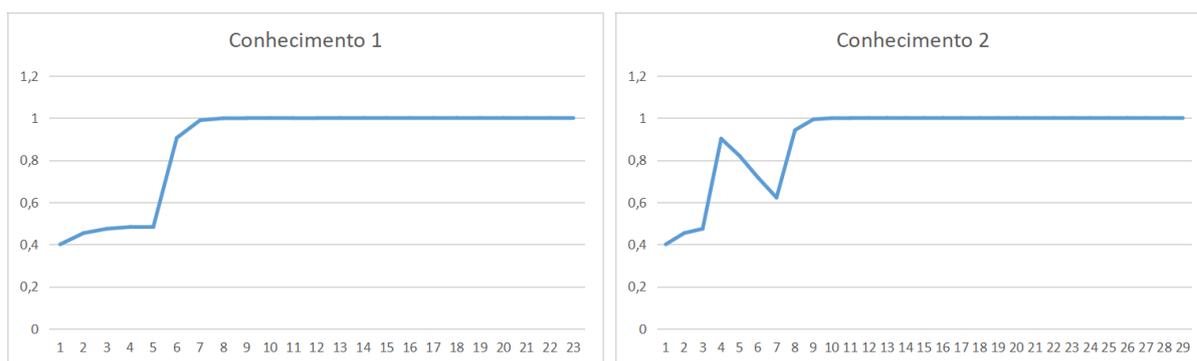


Figura 22 – Curva de Aprendizagem Vol 1 - Conhecimento 1 Figura 23 – Curva de Aprendizagem Vol 1 - Conhecimento 2

4.1.7 KDD : Análise da curva de aprendizagem do Voluntário 2

O voluntário 2 atingiu um total de 10 questões corretas em 14 tentativas para o conhecimento 1. A figura 24 apresenta a curva de aprendizagem elaborada com base na probabilidade de transação $P(T)$. O gráfico representa bem a curva de aprendizagem para o conhecimento 1 uma vez que seu desempenho após a subida considerável da sua curva na quarta tentativa não voltou a cair e suas respostas passaram todas a serem corretas. Ao analisar os dados das tentativas foi verificado que o voluntário errou por diversas vezes uma única questão e após ele acertar tal questão ele não errou mais.

Para o conhecimento 2 a curva de aprendizagem do voluntário 2 foi similar ao conhecimento 1, porém apresentando uma curva de aprendizagem completamente diferente do voluntário 1 para este conhecimento. Ele acertou um total de 8 questões em 14

tentativas. Uma análise mais profunda nos dados mostram que o voluntário 2 não teve muitas dificuldades no conhecimento 2, errando questões somente uma única vez.

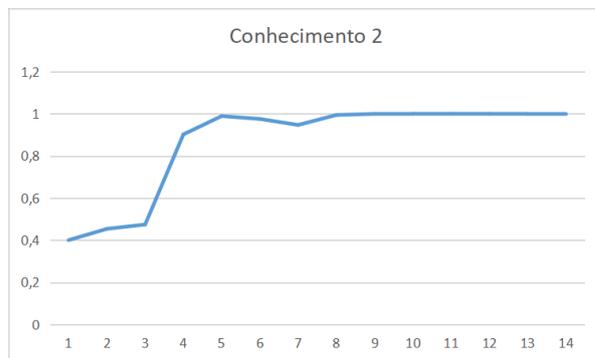
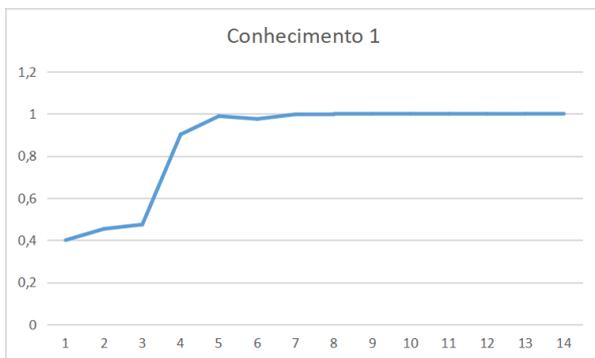


Figura 24 – Curva de Aprendizagem Vol 2 - Conhecimento 1 | Figura 25 – Curva de Aprendizagem Vol 2 - Conhecimento 2

4.1.8 KDD : Análise da curva de aprendizagem do Voluntário 3

O Voluntário 3 apresentou uma curva de aprendizagem para o conhecimento 1 semelhante a dos outros dois voluntários, tendo acertado 15 questões em 23 tentativas. A Figura 26 apresenta a curva de aprendizagem do voluntário. Ele errou 7 questões em sequência o que fez com que sua curva estabilizasse em torno de 0,5 e depois voltou a acertar fazendo com que sua curva subisse consideravelmente.

A curva de aprendizagem do voluntário 2 para o conhecimento 2 foi uma das mais interessantes. Ele apresentou um rendimento de 9 acertos em 18 tentativas, entretanto seus erros e acertos não foram em sequência como os outros voluntários. A Figura 27 apresenta a curva de aprendizagem do voluntário 2 para o conhecimento 2. Nela é possível observar que o aluno teve muita dificuldade inicialmente para o conhecimento, com sua curva variando por diversas vezes e tendo uma queda considerável durante um período inicial.

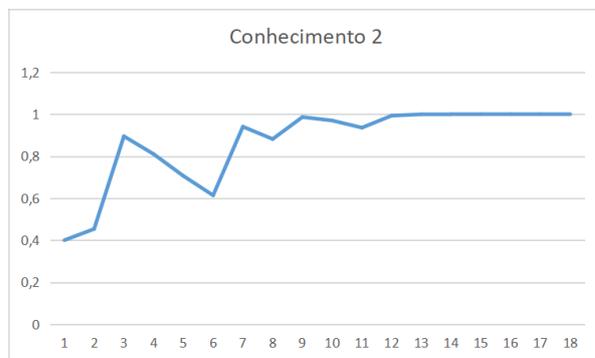
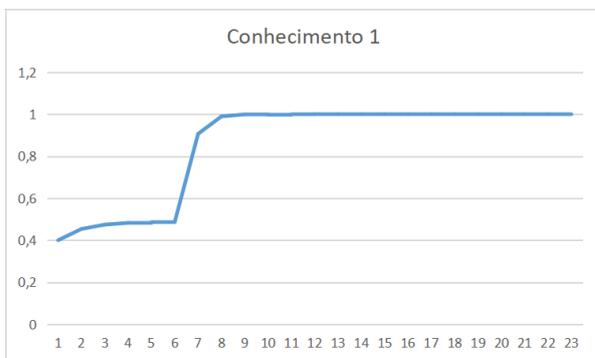


Figura 26 – Curva de Aprendizagem Vol 3 - Conhecimento 1 | Figura 27 – Curva de Aprendizagem Vol 3 - Conhecimento 2

4.2 Análise da capacidade preditiva dos modelos

Esta seção apresenta uma análise sobre a capacidade preditiva dos modelos. Para cada dataset selecionado neste trabalho foi feito um modelo utilizando BKT - *expectation Maximization*, BKT - *Conjugate Gradient Descent* e PFA.

Como métrica é utilizado RMSE e a acurácia real do modelo (porcentagem de acertos) para a análise.

4.2.1 BKT - *Expectation-Maximization*

A Tabela 2 apresenta os resultados obtidos pelos modelos criados utilizando a *Bayesian Knowledge Tracing - expectation-maximization*. Para o dataset COSMODS o modelo teve um RSME de 0.4080 e uma acurácia de 0.7698, resultado completamente normal dentro dos padrões de modelos criados. Entretanto RSME é uma métrica que diz por quanto o modelo está errado, ou seja, um valor de 0.4 para cima não é visto como um dos melhores valores. Vale ressaltar que os valores de RMSE não são valores absolutos e variam do que está tentando ser previsto. Por exemplo, se o modelo esta tentando prever a quantidade de cimento em um prédio e erra por alguns centímetros quadrados não é um problema, porem se a métrica observada for em metros ou kms o mesmo valor não é mais tão aceitável. Para esta tabela o valor observado é 0 e 1, sendo assim o valor de 0.4 pode ser considerável aceitável, porém no limite.

Já a acurácia do modelo COSMODS é de 0.7698, ou seja, apresenta uma acurácia relativamente boa para uma base de dados pequena. Em comparação, o modelo gerado para o dataset COSMO-F, sem erros de compilação, apresenta RMSE de 0.4248 e acurácia 0.7191. É interessante perceber que apesar de os erros de compilação terem sido removidos, o modelo apresenta um RMSE maior que a sua contraparte e uma acurácia menor. Acredita-se que esse resultado seja devido ao que está se avaliando em si, a acurácia do modelo. Como o que foi removido do modelo foram os erros de compilação, grande parte da capacidade de predição de erros do modelo foi retirada junto, o que faz com que a acurácia diminua.

Tabela 2 – Resultado dos Modelos BKT-EM

	RMSE	Accuracy
COSMODS	0.4080	0.7698
COSMO-F	0.4248	0.7191
KDD	0.3390	0.8571
Spanish Vocabulary	0.3823	0.8044

Já para o modelo utilizando o dataset KDD, o RMSE foi de 3390 e a acurácia de 0.8571. Em contraste com os dados obtidos pelos últimos dois datasets, os resultados são consideravelmente melhores. Acredita-se que o motivo dos resultados do modelo terem sido superiores se deva a quantidade de dados consideravelmente maior para o dataset KDD, criando assim um modelo melhor. Entretanto vale ressaltar que os valores para os datasets COSMODS e COSMO-F não são ruins e são aceitáveis para serem usados em um STI.

O dataset *Spanish Vocabulary* apresentou um resultado de RMSE 0.3823 e acurácia de 0.8044, tendo assim um resultado também superior aos outros dois modelos anteriores, porem inferior aos valores gerados pelo modelo KDD. Para os modelos BKT utilizando *expectation-maximization* como método de *fitting* ficou claro que quanto mais dados para a geração do modelo, melhor.

4.2.2 BKT - *Conjugate Gradient Descent*

A tabela 3 apresenta os resultados obtidos pelos modelos criados utilizando *Bayesian Knowledge Tracing - Conjugate Gradient Descent*. O dataset COSMODS teve um resultado de RMSE 0.5494 e acurácia de 0.3280. Esses Resultados apresentando mostram um claro contraste entre os resultados obtidos entre EM e *Conjugate Gradient Descent*. Para RMSE obteve-se uma diferença de quase 0.14, mostrando assim que o *fitting* do modelo BKT-EM obteve uma taxa de erro médio bem inferior ao modelo BKT-CGD. Já a acurácia do modelo foi ainda pior com um resultado de 0.3280 com uma diferença de aproximadamente 0.46. Entretanto, para o modelos do dataset KDD onde a quantidade de dados é alta os modelos apresentaram valores similares, o que indica que o problema do *fitting* que utiliza *conjugate gradient descent* seja muito dependente da quantidade de dados presentes no modelo.

Para o dataset COSMO-F o modelo apresentou RMSE de 0.4065 e acurácia de 0.5016, mostrando uma melhora considerável em relação aos dados do modelo utilizando o COSMODS, com uma diferença de 0.02 para RMSE e aproximadamente 0.18 para acurácia. Apesar da melhora em acurácia, o RMSE se manteve praticamente igual, fazendo com que os resultados obtidos não sejam interessantes para serem usados em um possível STI.

Tabela 3 – Resultado dos Modelos BKT - CGD

	RMSE	Accuracy
COSMODS	0.5494	0.3280
COSMO-F	0.5104	0.5061
KDD	0.3566	0.8547
Spanish Vocabulary	0.7246	0.1916

Já o dataset *Spanish Vocabulary* apresentou um comportamento muito diferente dos outros modelos. Ele apresentou um RMSE de 0.7246 com uma acurácia de 0.1916. Os resultados obtidos para o dataset *Spanish Vocabulary* foram os piores resultados de todo o experimento. Entretanto, não foi possível identificar a razão desse resultado, acredita-se que seja devido a alguma peculiaridade do domínio.

Os resultados obtidos para os modelos que utilizam BKT - *Conjugate Gradient Descent* em geral se mostraram inferiores aos modelos EM. Com isso é possível concluir que *Conjugate Gradient Descent* não se adequa a um tipo de dataset que possuem poucos dados, como o COSMODS ou COSMO-F. Entretanto, com o passar do tempo a medida que mais dados sejam coletados isso possa vir a mudar.

4.2.3 Performance Factor Analysis

Os resultados obtidos dos modelos gerados utilizando PFA podem ser observados na Tabela 4. Para o dataset COSMODS foi obtido um RMSE de 0.4235 e uma acurácia de 0.7161. Os resultados obtidos se mostraram similares em comparação com os modelos obtidos pelo *Bayesian Knowledge Tracing*.

Já para o dataset COSMO-F o modelo apresentou um RMSE de 0.4065 e acurácia de 0.5016, tendo assim um RMSE melhor que o dataset COSMODS porém uma acurácia muito inferior. Os resultados obtidos utilizando PFA para o dataset COSMO-F também apresentaram uma queda de qualidade considerável em comparação ao BKT-EM. Em compensação os resultados foram melhores em todos os aspectos em comparação com o BKT - *Conjugate Gradient Descent*.

Tabela 4 – Resultado dos modelos PFA

	RMSE	Accuracy
COSMODS	0.4235	0.7161
COSMO-F	0.4065	0.5016
KDD	0.3192	0.8665
Spanish Vocabulary	0.3618	0.8176

O modelo utilizando o dataset KDD teve o seu melhor resultado com RMSE de 0.3192 e acurácia de 0.8665. Os resultados obtidos pelo modelo KDD se comportaram de maneira similar para todos os modelos gerados. Com isso é possível concluir que todos os algoritmos utilizados para a modelagem de conhecimento de aluno nesse experimento são viáveis, variando de situação para situação.

O dataset *Spanish Vocabulary* também obteve um dos seus melhores resultados, apresentando RMSE de 0.3618 e uma accuracy de 0.8176, sendo um pouco superior ao BKT-EM.

Performance Factor Analysis se mostrou, em geral como um dos melhores algoritmos de modelagem de conhecimento desse experimento, superando BKT-EM em alguns aspectos, porém sendo pior em outros.

5 Discussão de Resultados

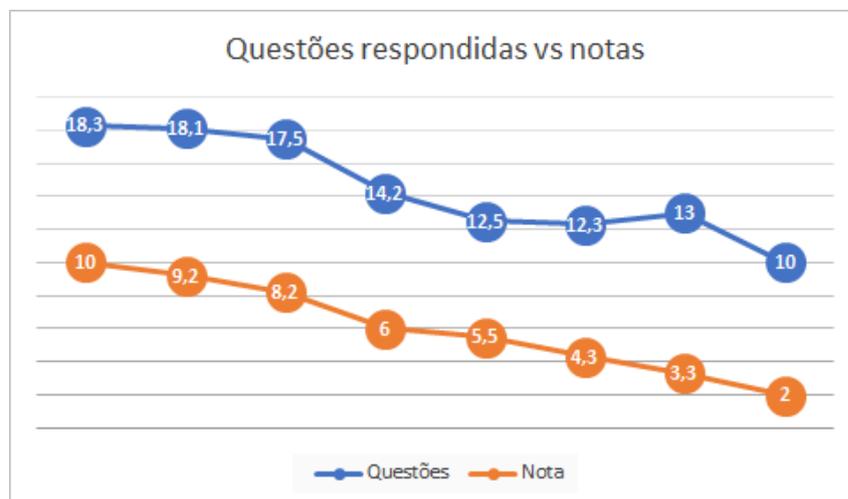
Este Capítulo apresenta uma discussão sobre os resultados obtidos neste trabalho.

5.1 Coleta de Dados

A coleta de dados para a criação do dataset COSMODS foi feita em sala de aula durante quatro semestres em uma turma de Algoritmos I do curso de ciência da computação na Universidade Federal do Maranhão. Durante esse período foram realizadas aulas práticas com os alunos onde foi utilizado o COSMO para responder tarefas programáveis dentro do sistema.

Em média, os alunos que participaram do dataset COSMODS obtiveram 7.17 na primeira avaliação da disciplina, enquanto os alunos que não usaram o sistema atingiram uma média de 5.8, de um total de 10. Além disso, notou-se que os alunos que praticaram mais, isto é, responderam à um número maior de questões, se saíram melhor na avaliação da disciplina, o que pode ser visto na Figura 28.

Figura 28 – Questões Respondidas vs Notas



Foram inspecionados e avaliados individualmente os dados de todos os voluntários. Há uma clara coerência entre a curva de aprendizagem dos alunos e as notas obtidas durante a primeira avaliação da disciplina de Algoritmos. Com esse resultado é possível inferir que as atividades presentes no dataset COSMODS representam de maneira coesa o ensino de algoritmos e práticas de programação para o grupo de alunos.

Durante a coleta de dados foi identificado que os alunos de Algoritmos I têm uma tendência a preparar as suas respostas fora do ambiente virtual de aprendizagem COSMO.

Isso é um problema para a coleta de dados devido ao fato de que o sistema deixa de coletar informações como possíveis erros que o aluno cometa, além de parecer que o aluno pouco faz uso do sistema. Durante o processo os alunos foram devidamente incentivados pelo professor da disciplina a utilizar o COSMO como sua principal plataforma para a produção de código fonte.

Um provável motivo para o uso indevido do sistema por parte dos alunos é o fator novidade que um compilador apresenta. Algoritmos I é o primeiro contato que os alunos têm com o ensino de programação e o fator novidade é parte importante do processo de aprendizagem. Algoritmos I apresenta o primeiro contato do aluno com um compilador, ambiente utilizado para a compilação de códigos fontes, o que faz com que eles queiram utilizar a nova tecnologia. Outro possível motivo é o fato de que o ambiente não apresenta a saída do código do aluno, somente uma mensagem de certo ou errado. A falta de saída do código fonte do aluno pode fazer com que o aluno migre para o compilador para entender melhor o que acontece durante a produção do código fonte.

Apesar dos problemas encontrados durante a coleta de dados, o dataset COSMODS apresenta uma gama de dados no domínio de Algoritmos e Programação e os problemas encontrados não foram suficientemente grandes para invalidar os dados contidos dentro do dataset.

Futuramente é interessante que o COSMO continue a ser utilizado dentro de sala de aula e se possível utilizado em outras turmas a fim de aumentar o tamanho do dataset.

5.2 Análise dos Modelos

Esta seção apresenta uma discussão sobre a análise dos modelos feita para atingir o objetivo proposto neste trabalho.

Para um tutor inteligente a curva de aprendizagem é a representação do conhecimento do aluno. Através da curva o tutor consegue identificar e inferir informações sobre qual a melhor estratégia a ser tomada para aprimorar a experiência e ensino do aluno. O objetivo do tutor vai ser tentar manter a curva de aprendizagem do aluno estável dentro do previsto pelo ensino, recomendando tarefas fáceis ou difíceis de acordo com a necessidade do aluno.

Já a capacidade preditiva do modelo é um outro pilar para a recomendação e tomada de decisão do tutor inteligente. A capacidade preditiva auxilia o tutor a prever se um aluno irá acertar ou não uma tarefa que o tutor recomende para o aluno. Normalmente o tutor tenta não recomendar tarefas que o aluno vá errar a fim de melhorar a experiência de ensino.

Sendo assim, a curva de aprendizagem e capacidade preditiva do modelo são fatores indispensáveis para a criação de um tutor inteligente.

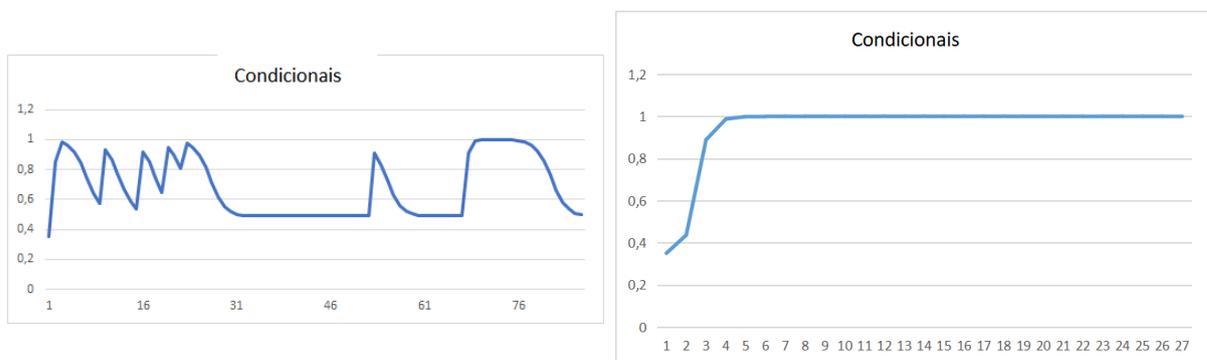
5.2.1 Análise da Curva de Aprendizagem

Este trabalho fez uma análise qualitativa sobre a curva de aprendizagem utilizando a variável $P(T)$ produzida por modelos BKT. Foi feita uma análise sobre a curva de aprendizagem dos modelos utilizando os datasets COSMODS, a sua variação COSMO-F, *Spanish Vocabulary* e o dataset KDD 2010. Entretanto, para evitar redundância nos resultados, foram escolhidos somente três alunos dos datasets COSMODS, COSMO-F e KDD CUP 2010 para ilustrar os pontos neste trabalho.

A análise mostra que é possível produzir uma representação de curva de aprendizagem usando a variável de transição $P(T)$ com uma quantidade reduzida de dados, o que faz com que o tamanho do dataset seja irrelevante para a produção da curva de aprendizagem nos tutores inteligentes. Entretanto, durante análise foi identificado que o domínio do dataset pode influenciar na sua curva de aprendizagem. COSMODS apresenta uma quantidade substancialmente maior de erros por tentativa/aluno do que os outros domínios. Isso se dá devido a natureza do domínio de programação onde grande parte do processo de produção de códigos fontes se dá por tentativa e erro, diferente de outros domínios.

Durante o experimento foi identificado que a curva de aprendizagem do domínio da programação apresentava formatos diferentes dos outros domínios. O motivo da curva de aprendizagem apresentar características diferentes do normal é a presença dos erros de compilação. Devido a isso, foi necessário a criação de um segundo dataset sem os erros de compilação a fim de se fazer um comparativo entre as curvas de aprendizagem de ambos os datasets considerados. A Figura 29 apresenta a curva de aprendizagem de um aluno com e sem os erros de compilação sendo levados em conta.

Figura 29 – Comparativo da curva de aprendizagem entre dataset COSMODS e COSMO-F



É visível que os erros de compilação têm um grande impacto na curva de aprendizagem dos alunos. Então para representar a curva de aprendizagem de alunos no domínio

da programação o dataset deve desconsiderar erros os erros de compilação ou então representar tais erros de maneira diferente a fim de identificar e tratá-los durante o processo de produção das curvas.

5.2.2 Análise da Capacidade Preditiva dos Modelos

Este trabalho apresentou um comparativo entre a capacidade preditiva dos modelos gerados utilizando técnicas de modelagem para diferentes datasets. O trabalho exibiu resultados de *Root Mean Square Error*(RMSE) e sua acurácia real para todos os algoritmos de modelagem.

5.2.2.1 Discussão sobre os resultados de RMSE

RMSE é uma métrica que diz por quanto o modelo está errado em relação ao valor de referência. Quando menor o valor apresentado, melhor o resultado para o modelo. Entretanto, tal valor não é absoluto e depende da variável que está sendo observada, um valor de 1 pode ser bom quanto se está olhando uma variável em centímetros e ruim em quilômetros. A tabela Geral dos valores do RMSE pode ser observada na Tabela 5.

Para o caso de *Expectation-Maximization* foi identificado que quanto maior a quantidade de dados melhor os resultados apresentados se tratando de RMSE. Isso provavelmente se dá devido a forma como *expectation-maximization* faz o *fitting* dos dados. KDD CUP 2010 apresentou o melhor resultado de RMSE, enquanto o dataset COSMO-F apresentou o pior resultado, visto que tem o menor número de dados.

Já *Conjugate Gradient Descent* apresentou resultados diferentes, KDD continua apresentando os melhores resultados se tratando de RMSE provavelmente devido a sua quantidade de dados, entretanto o dataset COSMODS apresentou resultados piores que a sua contraparte COSMO-F. A mudança dos valores entre o COSMODS e COSMO-F provavelmente se deve pelo fato de que grande parte das respostas consideradas erradas dentro do dataset COSMODS foram removidas ao remover os erros de compilação.

PFA apresentou resultados similares a *Expectation-Maximization* quando se observa os valores de RMSE. Novamente KDD CUP apresenta o melhor resultado dentre os valores observados. Entretanto PFA apresenta um valor melhor de RMSE para o dataset COSMO-F.

A análise dos valores do RMSE mostra que a variação dos resultados apresentados é relativamente pequena, o que não invalida os modelos apresentados. A pequena variação entre os valores de RMSE de um modelo para o outro mostra que a quantidade de dados, apesar de ser importante, não é um fator essencial para a produção de modelos, e que a técnica de modelagem apresenta um fator de importância maior quando se fala de RMSE.

Tabela 5 – Tabela geral de RMSE

	BKT-EM	BKT-CGD	PFA
COSMODS	0.4080	0.5494	0.4235
COSMO-F	0.4248	0.5104	0.4065
KDD CUP	0.3390	0.3566	0.3192
Spanish Vocabulary	0.3823	0.7246	0.3618

5.2.2.2 Discussão sobre os resultados da Acurácia

A análise da acurácia real dos modelos apresentou resultados interessantes. Novamente a quantidade de dados apresentados é um fator importante, porém não essencial, para o resultado dos modelos. Quanto maior a quantidade de dados, melhor o modelo consegue prever e acertar valores futuros. Entretanto há uma mudança drástica de valores quando se compara uma técnica de modelagem com outra. A Tabela 6 apresenta os valores de acurácia dos modelos deste trabalho.

Expectation-Maximization apresentou ótimos resultados para todos os datasets. KDD CUP apresentou o melhor resultado seguido do *Spanish Vocabulary*. Já COSMODS apresentou um resultado relativamente pior seguido pelo COSMO-F. O motivo para a mudança no valor da acurácia provavelmente se deve pela quantidade de dados. Apesar da mudança entre os valores de acurácia, BKT-EM se mostrou capaz de prever com certa acurácia mesmo com a quantidade de dados reduzida.

Conjugate Gradient Descent apresentou o pior resultado em comparação com as outras técnicas para todos os datasets. Para o dataset KDD o resultado foi similar ao apresentado em BKT-EM, entretanto o mesmo não pode ser dito para os outros datasets. A quantidade de dados é visivelmente um fator importante para *Conjugate Gradient Descent*, entretanto o dataset COSMO-F mesmo com sua quantidade de dados reduzida apresentou resultados melhor do que sua contraparte COSMODS. Em geral a técnica BKT-CGD se mostrou muito dependente da quantidade de dados e apresentando resultado voláteis demais para ser usada consistentemente em um tutor inteligente.

Performance Factor Analysis apresentou resultados similares novamente a *Expectation-Maximization*. Entretanto PFA se mostrou mais dependente da quantidade de dados presentes dentro do dataset. Os resultados apresentados para o dataset COSMO-F se mostraram muito inferiores aos resultados do COSMODS, dataset com os erros de compilação. Normalmente um aluno comete mais erros que acerto, então ao remover uma parte dos erros do dataset a técnica perde parte da sua capacidade preditiva.

Tabela 6 – Tabela de acurácia dos modelos

	BKT-EM	BKT-CGD	PFA
COSMODS	0.7698	0.3280	0.7161
COSMO-F	0.7191	0.5061	0.5016
KDD	0.8571	0.8547	0.8665
Spanish Vocabulary	0.8044	0.1916	0.8176

Com esses resultados, *Expectation-Maximization* se mostrou como a técnica de modelagem mais estável para ser usada em um tutor inteligente, apesar de que ambos PFA como BKT podem ser utilizadas ao mesmo tempo para um sistema de recomendação. É interessante ressaltar a importância que a quantidade de dados tem sobre os resultados dos modelos. Uma quantidade de dados elevada irá produzir resultados melhores, entretanto a diferença de predição entre um dataset com uma baixa quantidade de dados e uma alta quantidade é relativamente pequena, o que mostra que não é necessário para um tutor inteligente começar a funcionar com muitos dados.

5.3 Erros de Compilação

Um dos achados de pesquisa deste trabalho foi o impacto que os erros de compilação dos alunos tem sobre os resultados dos modelos criados. Durante este trabalho foi identificado que os erros de compilação tem uma grande influencia sobre os resultados da acurácia dos modelos e principalmente sobre a representação da curva de aprendizagem dos alunos. Isso é um problema porque a representação da curva de aprendizagem é o que norteia o processo de adaptabilidade do tutor inteligente.

Para a curva de aprendizagem, os erros de compilação apresentam um grande desafio uma vez que também fazem parte do processo de aprendizagem do aluno. Parte do processo de aprendizagem inicial do domínio da programação é aprender as regras de sintaxe de programação relativa a linguagem a ser aprendida. Entretanto os dados relacionados ao erro de aprendizagem não indicam o quanto o aluno aprendeu sobre lógica de programação ou algum conteúdo específico, como variáveis ou condicionais. Para a representação da curva de aprendizagem é interessante que os dados relacionados ao erro de compilação sejam removidos e a curva do aluno seja considerada somente os erros de lógica. É então necessário que a base de dados COSMODS represente as respostas dos alunos que apresentarem erros de compilação como uma observação diferente de certo ou errado.

Já para a acurácia do modelos, os erros de compilação apresentaram resultados mistos, dependente do tipo de técnica de modelagem utilizada. Para técnicas como BKT-EM os erros de compilação não influenciaram muito os resultados apresentados, a

variação dos resultados em comparação com o dataset com os erros de compilação pode ser atribuída a perda de dados relativa a remoção de tais erros. Para BKT-CGD a remoção dos erros de compilação melhoraram os resultados da acurácia do modelo, entretanto é valido apontar que o resultado da acurácia para os modelos BKT-CGD foi o pior das técnicas de modelagem apresentadas. Já para PFA a remoção dos erros de compilação faz com que a acurácia do modelo caia drasticamente , isso mostra que PFA é extremamente dependente da quantidade de dados que um dataset possui. De maneira geral, os erros de compilação influenciam os modelos para melhor ou pior de acordo com a técnica utilizada, levando isso em consideração a técnica de modelagem de conhecimento deve ser escolhida com cuidado. É possível argumentar tanto para a remoção dos erros de compilação quanto para admissão dos mesmos de acordo com a técnica que está sendo utilizada.

De maneira geral, é recomendado que os erros de compilação sejam tratados de maneira diferente de erros de logica a fim de facilitar a adaptabilidade da recomendação do tutor inteligente. Para a representação da curva de aprendizagem é imperativo que os erros sejam removidos e para a predição é dependente das técnicas utilizadas.

6 Considerações Finais e Trabalhos Futuros

Este trabalho apresentou uma análise qualitativa entre os modelos de conhecimento de alunos no domínio da programação e suas peculiaridades. Para atingir o objetivo deste trabalho inicialmente foi feito um levantamento bibliográfico através de um mapeamento sistemático de literatura. O mapeamento apontou BKT e PFA como as técnicas de modelagem mais utilizadas na comunidade científica. Além disso, o mapeamento apontou diversos datasets em vários domínios para serem utilizados neste trabalho. Entretanto o mapeamento falhou em apontar um dataset com o domínio voltado para programação e atendendo todos os requisitos necessários para Sistemas Tutores Inteligentes (STI). Devido a falha em encontrar um dataset no domínio da programação, foi necessária uma coleta de dados para a criação de um novo dataset em tal domínio.

Uma das contribuições deste trabalho foi a criação do dataset COSMODS. A coleta de dados para o dataset foi feita durante 4 semestres em turmas na disciplina de Algoritmos I do curso de Ciência da Computação na Universidade Federal do Maranhão (UFMA). O dataset conta com o registro de aproximadamente 7000 tentativas de respostas para as tarefas presentes no dataset por cerca de 90 alunos ao todo. O COSMODS conta com 24 questões relacionadas a 2 tipos de conhecimento: variáveis e condicionais. Além disso, a base de dados apresenta um log completo das ações dos alunos, podendo ser criada uma trilha do caminho de tarefas feito pelo aluno durante a disciplina.

Além disso, foi feita uma análise qualitativa da curva de aprendizagem e da capacidade preditiva dos modelos para o domínio da programação. A análise mostra que a representação da curva de aprendizagem não é dependente de uma quantidade significativa de dados. Além disso, o trabalho também aponta que os erros de compilação influenciam diretamente na representação de tal curva. Para a capacidade preditiva do modelo, a análise mostrou que a seleção da técnica de modelagem é o aspecto mais importante a ser levado em consideração. A quantidade de dados apresenta um fator significativo, porém a variação devida a oscilação na quantidade de dados não é grande o suficiente para invalidar a maioria dos modelos.

A curva de aprendizagem e a capacidade preditiva dos modelos são fatores indispensáveis para a recomendação de um Sistema Tutor Inteligente. É através da curva de aprendizagem que o tutor inteligente consegue inferir informações sobre o conhecimento do aluno e definir qual estratégia de ensino a ser utilizada. Já a capacidade preditiva é o principal fator que determina qual a tarefa a ser recomendada para o aluno. Durante este trabalho foi identificado que os erros de compilação apresentados em datasets no domínio da programação causam um grande impacto tanto na representação da curva de

aprendizagem quanto na capacidade preditiva do modelo. É interessante perceber que o domínio e a forma como as tarefas são elaboradas influenciam nos resultados gerados pelos modelos.

Retomando os objetivos específicos apresentados na Seção 1.1, temos:

Na Seção 3.6.4 tem-se o detalhamento de como a coleta foi feita para atingir o objetivo proposto neste trabalho. A coleta de dados foi feita durante quatro semestres na Universidade Federal do Maranhão em uma disciplina de Algoritmos 1 do curso de Ciências da Computação. Por fim os dados coletados foram compilados para a criação do dataset COSMODS.

A Seção 3.6.1 apresenta os resultados do levantamento bibliográfico feitos para a seleção das técnicas de modelagem que foram utilizadas neste trabalho. A Seção 3.6.3 apresenta as técnicas selecionadas e a motivação para a seleção das mesmas.

Na seção ?? tem-se o detalhamento da análise dos modelos. A Subseção 4.1 apresenta a análise feita na curva de aprendizagem produzida utilizando a variação de transição do modelo BKT. Já a Subseção 4.2 apresenta uma análise da capacidade preditiva dos modelos para cada técnica selecionada durante este trabalho.

Por fim, este trabalho gerou os artigos (RAPOSO; MARANHÃO; NETO, 2019) e (MARANHÃO et al., 2019) publicados no SBIE 2019 e SEMP 2019. O trabalho (RAPOSO; MARANHÃO; NETO, 2019) apresenta uma análise qualitativa da curva de aprendizagem dos alunos da disciplina de algoritmos no curso de ciência da computação da UFMA. Já o trabalho (MARANHÃO et al., 2019) apresenta uma nova abordagem para a criação de trilhas multimídias utilizando *Hyperknowledge* e o dataset COSMODS.

Como trabalhos futuros, uma das informações que o dataset COSMODS apresenta e que não é utilizada para a recomendação é o código fonte produzido pelos alunos. Futuramente é interessante que seja feita uma análise do código apresentados pelos alunos e que o resultado dessa análise seja incorporado pelo Sistema Tutor Inteligente na tomada de decisões. O código fonte apresentado pelos alunos mostra diversos padrões que podem ser usados para inferir informações sobre a compreensão dos alunos para tal conteúdo, além de prover o sistema com um possível caminho a se seguir para aprimorar a experiência do usuário. Além disso, deve ser feita uma coleta maior de dados, utilizando mais turmas de Algoritmos 1 a fim de diversificar os dados dentro do dataset.

Referências

- ALMEIDA, M. E. B. de. Educação a distância na internet: abordagens e contribuições dos ambientes digitais de aprendizagem. *Educação e pesquisa*, SciELO Brasil, v. 29, n. 2, p. 327–340, 2003. Citado na página 17.
- BAKER, R. S. d; CORBETT, A. T.; ALEVEN, V. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In: SPRINGER. *International conference on intelligent tutoring systems*. [S.l.], 2008. p. 406–415. Citado na página 24.
- BAKER, R. S. d et al. Ensembling predictions of student knowledge within intelligent tutoring systems. In: SPRINGER. *International Conference on User Modeling, Adaptation, and Personalization*. [S.l.], 2011. p. 13–24. Citado 2 vezes nas páginas 34 e 36.
- BECK, J. E.; CHANG, K.-m. Identifiability: A fundamental problem of student modeling. In: SPRINGER. *International Conference on User Modeling*. [S.l.], 2007. p. 137–146. Citado 3 vezes nas páginas 24, 25 e 37.
- BRUSILOVSKIY, P. L. The construction and application of student models in intelligent tutoring systems. *Journal of computer and systems sciences international*, New York: Scripta Technica, 1995-, v. 32, n. 1, p. 70–89, 1994. Citado na página 22.
- CEN, H.; KOEDINGER, K.; JUNKER, B. Learning factors analysis—a general method for cognitive model evaluation and improvement. In: SPRINGER. *International Conference on Intelligent Tutoring Systems*. [S.l.], 2006. p. 164–175. Citado 3 vezes nas páginas 25, 33 e 36.
- CHAI, T.; DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)?—arguments against avoiding rmse in the literature. *Geoscientific model development*, Copernicus GmbH, v. 7, n. 3, p. 1247–1250, 2014. Citado na página 26.
- CORBETT, A. T.; ANDERSON, J. R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, Springer, v. 4, n. 4, p. 253–278, 1994. Citado 2 vezes nas páginas 25 e 37.
- CUP, K. *KDD cup 2010*. 2010. Disponível em: <<https://pslclatashop.web.cmu.edu/KDDCup/>>. Citado na página 36.
- DAVID, Y. B.; SEGAL, A.; GAL, Y. K. Sequencing educational content in classrooms using bayesian knowledge tracing. In: ACM. *Proceedings of the sixth international conference on Learning Analytics & Knowledge*. [S.l.], 2016. p. 354–363. Citado 2 vezes nas páginas 23 e 24.
- FREEDMAN, R.; ALI, S.; MCROY, S. What is an intelligent tutoring system. *intelligence*, v. 11, n. 3, p. 15–16, 2000. Citado na página 21.
- GONG, Y.; BECK, J. E.; HEFFERNAN, N. T. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In: SPRINGER. *International conference on intelligent tutoring systems*. [S.l.], 2010. p. 35–44. Citado na página 26.

- GONG, Y.; BECK, J. E.; HEFFERNAN, N. T. How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *International Journal of Artificial Intelligence in Education*, IOS Press, v. 21, n. 1-2, p. 27–46, 2011. Citado 5 vezes nas páginas 26, 33, 34, 36 e 37.
- JR, P. I. P.; CEN, H.; KOEDINGER, K. R. Performance factors analysis—a new alternative to knowledge tracing. *Online Submission*, ERIC, 2009. Citado 6 vezes nas páginas 15, 25, 33, 36, 37 e 39.
- KEELE, S. et al. *Guidelines for performing systematic literature reviews in software engineering*. [S.l.], 2007. Citado na página 27.
- KOEDINGER, K. R. et al. A data repository for the edm community: The pslc datashop. *Handbook of educational data mining*, CRC Press, Boca Raton, FL, v. 43, p. 43–56, 2010. Citado na página 36.
- Kurup, L. D.; Joshi, A.; Shekhokar, N. A review on student modeling approaches in its. In: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. [S.l.: s.n.], 2016. p. 2513–2517. Citado 2 vezes nas páginas 25 e 37.
- Käser, T. et al. Dynamic bayesian networks for student modeling. *IEEE Transactions on Learning Technologies*, v. 10, n. 4, p. 450–462, Oct 2017. ISSN 1939-1382. Citado 3 vezes nas páginas 15, 22 e 36.
- MARANHÃO, D. S. et al. Towards the use of hyperknowledge in educational domain through multimedia trails. In: IEEE. *2019 IEEE International Symposium on Multimedia (ISM)*. [S.l.], 2019. p. 323–3233. Citado na página 59.
- MEHLECKE, Q. T. C.; TAROUÇO, L. M. R. Ambientes de suporte para a educação a distância: a mediação para aprendizagem cooperativa. *RENOTE: revista novas tecnologias na educação [recurso eletrônico]*. Porto Alegre, RS, 2003. Citado na página 17.
- MITROVIC, A. Fifteen years of constraint-based tutors: what we have achieved and where we are going. *User modeling and user-adapted interaction*, Springer, v. 22, n. 1-2, p. 39–72, 2012. Citado na página 22.
- MITROVIC, A.; MARTIN, B.; SURAWEERA, P. Intelligent tutors for all: The constraint-based approach. *IEEE Intelligent Systems*, IEEE, n. 4, p. 38–45, 2007. Citado na página 22.
- MITROVIC, A.; OHLSSON, S. Constraint-based knowledge representation for individualized instruction. University of Canterbury. Computer Science and Software Engineering., 2006. Citado na página 22.
- NEDUNGADI, P.; REMYA, M. Predicting students' performance on intelligent tutoring system—personalized clustered bkt (pc-bkt) model. In: IEEE. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*. [S.l.], 2014. p. 1–6. Citado 2 vezes nas páginas 35 e 36.
- NEDUNGADI, P.; REMYA, M. Incorporating forgetting in the personalized, clustered, bayesian knowledge tracing (pc-bkt) model. In: IEEE. *2015 International Conference on cognitive computing and information processing (CCIP)*. [S.l.], 2015. p. 1–5. Citado 2 vezes nas páginas 35 e 36.

- NKAMBOU, R.; MIZOGUCHI, R.; BOURDEAU, J. *Advances in intelligent tutoring systems*. [S.l.]: Springer Science & Business Media, 2010. v. 308. Citado na página 22.
- NWANA, H. S. Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, Springer, v. 4, n. 4, p. 251–277, 1990. Citado na página 21.
- OHLSSON, S. Constraint-based student modeling. In: *Student modelling: the key to individualized knowledge-based instruction*. [S.l.]: Springer, 1994. p. 167–189. Citado na página 22.
- PELÁNEK, R. Metrics for evaluation of student models. *Journal of Educational Data Mining*, Citeseer, v. 7, n. 2, p. 1–19, 2015. Citado na página 26.
- PSOTKA, J.; MASSEY, L. D.; MUTTER, S. A. *Intelligent tutoring systems: Lessons learned*. [S.l.]: Psychology Press, 1988. Citado 2 vezes nas páginas 14 e 21.
- RABÊLO-JÚNIOR, D. J. L. et al. Cosmo: Um ambiente virtual de aprendizado com foco na introdução de algoritmos. Universidade Federal do Maranhão, 2018. Citado 4 vezes nas páginas 17, 18, 19 e 20.
- Ramesh, V. M.; Rao, N. J.; Ramanathan, C. Implementation of an intelligent tutoring system using moodle. In: *2015 IEEE Frontiers in Education Conference (FIE)*. [S.l.: s.n.], 2015. p. 1–9. Citado 2 vezes nas páginas 14 e 21.
- RAPOSO, A. C.; MARANHÃO, D.; NETO, C. S. Análise do modelo bkt na avaliação da curva de aprendizagem de alunos de algoritmos. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2019. v. 30, n. 1, p. 479. Citado na página 59.
- RIBEIRO, R. B. et al. Gamificação de um sistema de juiz online para motivar alunos em disciplina de programação introdutória. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2018. v. 29, n. 1, p. 805. Citado na página 14.
- ROLLINSON, J.; BRUNSKILL, E. From predictive models to instructional policies. *International Educational Data Mining Society*, ERIC, 2015. Citado na página 15.
- YUDELSON, M. M. *Hmm-scalable*. 2020. Disponível em: <<https://github.com/myudelson/hmm-scalable>>. Citado na página 39.
- YUDELSON, M. V. Individualizing bayesian knowledge tracing. are skill parameters more important than student parameters?. *International Educational Data Mining Society*, ERIC, 2016. Citado 2 vezes nas páginas 15 e 22.
- YUDELSON, M. V.; KOEDINGER, K. R.; GORDON, G. J. Individualized bayesian knowledge tracing models. In: SPRINGER. *International conference on artificial intelligence in education*. [S.l.], 2013. p. 171–180. Citado 5 vezes nas páginas 23, 25, 26, 34 e 37.