CEETEPS – PROGRAMA DE PÓS-GRADUAÇÃO
MESTRADO EM TECNOLOGIA: GESTÃO, DESENVOLVIMENTO E FORMAÇÃO

JOSÉ CASSIANO GRASSI GUNJI

APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS NA AVALIAÇÃO DE RESULTADOS DE TESTE DE COMPETÊNCIA DE LEITURA DE PALAVRAS (TCLP)

SÃO PAULO
MARÇO DE 2009

José Cassiano Grassi Gunji	APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS NA AVALIAÇÃO DE RESULTADOS DE TCLP	Mestrado em Tecnologia

Gunji, José Cassiano Grassi

G975a Aplicação de técnicas de mineração de dados na avaliação de resultados de teste de competência de leitura de palavras (TCLP) / José Cassiano Grassi Gunji. - São Paulo: CEETEPS, 2009.

62 f.

Dissertação (Mestrado) - Centro Estadual de Educação Tecnológica Paula Souza, 2009.

1. Mineração de dados. 2. Alfabetização. 3. Inteligência artificial. Título.

CDU 372.4:519.6

JOSÉ CASSIANO GRASSI GUNJI

APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS NA AVALIAÇÃO DE RESULTADOS DE TESTE DE COMPETÊNCIA DE LEITURA DE PALAVRAS (TCLP)

Dissertação apresentada como exigência parcial para obtenção do Título de Mestre em Tecnologia no Centro Estadual de Educação Tecnológica Paula Souza, no Programa de Mestrado em Tecnologia: Gestão Desenvolvimento e Formação, sob orientação do Prof. Dr. Maurício Amaral de Almeida.

SÃO PAULO
MARÇO DE 2009

JOSÉ CASSIANO GRASSI GUNJI

APLICAÇÃO DE TÉCNICAS DE MINERAÇÃO DE DADOS NA AVALIAÇÃO DE RESULTADOS DE TESTE DE COMPETÊNCIA DE LEITURA DE PALAVRAS (TCLP)

PROF. DR. MAURÍCIO AMARAL DE ALMEIDA

PROF. DR. MARCELO DUDUCHI

PROF. DR. ELIZEU COUTINHO MACEDO

SÃO PAULO, _____ DE _____ DE _____

A meus pais, José Gunji e Iara Marina Grassi Gunji, cujo esforço de vida foi o maior responsável por me conduzir até aqui. Meu eterno agradecimento.

Agradecimentos

Agradeço ao meu orientador, Maurício Amaral de Almeida pelo inestimável apoio e orientação, sua disponibilidade irrestrita para dividir conhecimento, experiência e sabedoria.

Agradeço aos professores do programa de pós-graduação do Centro Paula Souza pelo belo trabalho desenvolvido nas atividades acadêmicas.

Agradeço a Avelino Luiz Rodrigues, cuja intervenção foi fundamental para que esta dissertação se tornasse possível.

Agradeço a Luciane Gonzalez Valle, que habilmente e decisoriamente ajudou-me a construir a pessoa que sou hoje, sem a qual esta dissertação, entre outras coisas, não existiriam.

Agradeço à minha irmã, Monise Grassi Gunji, cujo apoio, descontração, bom-humor e amizade tanto contribuem para que tudo se torne prazeroso.

Agradeço à Alessandra Nakhle, que surgiu como um presente do destino, trocou a ordem de tudo, mas fez com que tudo agora faça sentido.

A frase mais excitante a se ouvir na ciência, a que anuncia novas descobertas, não é "Eureka!", mas "Que engraçado..." (Isaac Asimov)

Resumo

GUNJI, J. C. G. **Aplicação de técnicas de mineração de dados na avaliação de resultados de teste de competência de leitura de palavras (TCLP)**. Dissertação (Mestrado em Tecnologia), Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2009. 61 p.

O objetivo desta dissertação é analisar os resultados obtidos com a aplicação do Teste de Competência de Leitura de Palavras (TCLP) com uma abordagem complementar, utilizando-se técnicas de Mineração de Dados (MD). O TCLP tem como finalidade avaliar o progresso do processo de alfabetização de alunos do Ensino Fundamental. O teste classifica um aluno como estando na série apropriada ao seu desenvolvimento, adiantado ou atrasado. Tal classificação é feita com ferramentas estatísticas. Com a aplicação de MD, este trabalho explora a possibilidade de que os resultados da aplicação do teste tragam ocultas informações não evidentes e as tornem úteis. Foram aplicadas três tarefas de MD aos dados: classificação, agrupamento (seguido de nova classificação) e extração de regras de associação. Os resultados obtidos são apresentados após a aplicação dos algoritmos e da interpretação de seus resultados. Foram observadas informações não evidentes quanto aos dados do teste, quanto aos alunos que responderam ao teste e quanto à formulação do teste em si.

Palavras-chave: Mineração de Dados, Inteligência Artificial, leitura, alfabetização, avaliação.

Abstract

GUNJI, J. C. G. **Applying data mining techniques to the evaluations of word reading competency test (WRCT) results**. Dissertation (Master degree in Technology), Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2009. 61 p.

The purpose of this dissertation is to analyze the results obtained with the use of the Word Reading Competency Test (WRCT) with a complimentary approach, using Data Mining (DM) techniques. The WRCT has as purpose to evaluate the progress of the literacy learning of fundamental school students. The test classifies the student as being at the correct grade to its development, ahead or behind. Such classification is done by means of statistical tools. By applying DM, this work explores the possibility that results of the application of the test hides information that is not evident and turns them into useful ones. Three DM tasks were applied to the data: classification, clustering (followed by a new classification) and association rules extraction. Obtained results are presented after the algorithms were applied and the results were interpreted. Information not evident was observed concerning the data of the test, the students who performed the test and as to the test formulation itself.

Keywords: Data Mining, Artificial Intelligence, reading, literacy, evaluation.

Lista de Figuras

Lista de Tabelas

Lista de Quadros

Sumário

1 Introdução

1.1 Linha de Pesquisa

Esta dissertação de mestrado enquadra-se na Linha de Pesquisa Gestão e Desenvolvimento de Tecnologias da Informação Aplicadas, desenvolvida pelo Centro Estadual de Ensino Tecnológico Paula Souza (CEETEPS).

1.2 Projeto de Pesquisa

A invenção e o desenvolvimento de sistemas de computação sempre foram motivados para que esses ofereçam ferramentas e recursos para facilitar, automatizar, agilizar e ampliar o alcance e o desempenho de muitas atividades humanas. Por esse motivo é natural que as atividades da Ciência da Computação assumam freqüentemente um papel multidisciplinar. Pode-se estudar computação de maneira pura, entretanto, é muito freqüente que se estude computação para se resolver um problema real. Este problema real, muitas vezes, é um problema de outra área de conhecimento. Com o avanço das técnicas, da tecnologia e da teoria da Ciência da Computação, uma segmentação dela tem ocorrido em diversas especialidades. Exemplos de especialidades são o Cálculo Numérico, a Computação Gráfica, Simulação, Elementos Finitos e Inteligência Artificial. A pesquisa de assuntos da computação com finalidade nela mesma tem sido cada vez mais freqüente. Essa não é uma crítica à pesquisa pura em Ciência da Computação. De fato, ela é essencial. Mas também é essencial a aplicação prática de suas conclusões e descobertas. Assim, essa dissertação propõe o seguinte Problema de Pesquisa:

1.3 Problema de Pesquisa

Como a aplicação de técnicas de Mineração de Dados (MD) pode oferecer mais informações da avaliação do resultado da aplicação do Teste de Competência em Leitura de Palavras (TCLP) além das informações já oferecidas pelo tratamento estatístico?

A seguir são comentadas brevemente as variáveis relacionadas ao problema de pesquisa.

1.3.1 Mineração de Dados

A Mineração de Dados é um conjunto de técnicas estudado na Inteligência Artificial (IA), uma especialidade da Ciência da Computação. A IA fornece um conjunto de técnicas e algoritmos úteis

para que sistemas computadorizados consigam resolver problemas que a computação tradicional não consegue resolver, seja por limitações teóricas (o problema não pode ser descrito de maneira prática) ou práticas (a memória ou o tempo de processamento são impraticáveis ou infinitos) (RUSSELL; NORVIG, 2004). Não se espera que um algoritmo de IA seja perfeito ou que sempre acerte em suas conclusões. Um nível de desempenho tão elevado ainda não é possível com o desenvolvimento atual. Uma exigência assim seria equivalente a não permitir que um produto de engenharia fosse utilizado a não ser que seu desempenho seja perfeito. Assim, não se permitiria que nenhum automóvel fosse operado a não ser que ele jamais sofresse uma pane mecânica. O que se espera desse algoritmo é que ele forneça alguma vantagem sobre a abordagem tradicional do problema. Essa vantagem pode ser o tempo para se chegar a uma resposta; pode ser o índice de sucesso; pode ser simplesmente encontrar uma solução qualquer quando outros métodos não conseguem, entre outras. Voltando à analogia do automóvel, vale a pena utilizar um automóvel, mesmo que falível, pois seu uso oferece mais vantagens do que desvantagens (FAUSETT, 1994; RUSSELL; NORVIG, 2004).

Dentre as técnicas abordadas pela IA encontram destaque as técnicas de Mineração de Dados. A MD avalia grandes quantidades de dados procurando por conhecimento oculto neles (REZENDE, 2005; 2003). Esse conhecimento pode ser a identificação de padrões de comportamento, tendências e relações de dependência entre as variáveis relacionadas no conjunto de dados.

1.3.2 Teste de Competência em Leitura de Palavras

O TCLP é um teste aplicado normalmente a crianças do Ensino Infantil e do Ensino Fundamental. Sua finalidade é aferir o estágio de desenvolvimento de suas habilidades de leitura e, assim, classificar o aluno como estando em sua série adequada, atrasado ou adiantado. Para tanto, a formulação do teste explora as características das etapas de aprendizado da leitura: **logográfica**, **alfabética** e **ortográfica** (NIKAEDO; KURIYAMA; MACEDO, 2007). No estágio logográfico, a criança tende a identificar a palavra pelo seu aspecto visual. Na etapa seguinte, a alfabética, a criança já consegue decodificar a palavra utilizando as regras da língua. Na etapa final, a ortográfica, a criança já é considerada um leitor bem-sucedido, sendo capaz de acessar o significado semântico da palavra diretamente.

A abordagem que separa o processo de aprendizado de leitura nas etapas citadas no parágrafo acima não é a única. Nikaedo, Kuriyama e Macedo (2007) explicam que do ponto de vista da evolução da escrita, podem ser descritos cinco estágios sucessivos de aprendizagem:

1. **Pré-silábico**: As crianças fazem traços no papel sem a intenção de realizar o registro sonoro do que foi proposto, pois ainda não compreendem a relação entre o registro gráfico e o aspecto sonoro da fala;
2. **Silábico sem valor sonoro**: A criança começa a tentar estabelecer relações entre o contexto sonoro e o contexto gráfico, mas sem a intenção de fazê-lo;
3. **Silábico com valor sonoro**: A criança começa a tentar estabelecer relações entre o contexto sonoro e o contexto gráfico, com a intenção de fazê-lo;
4. **Silábico-alfabético**: É uma fase de transição em que a criança não deixa de utilizar as estratégias aprendidas nos estágios anteriores, mas passa a compreender a escrita em termos dos fonemas;
5. **Alfabético**: Cada um dos caracteres da escrita corresponde a valores sonoros menores que as sílabas e, sistematicamente, a criança analisa a sonoridade dos fonemas antes de escrever.

1.4 Hipótese de Pesquisa

O uso de técnicas de MD complementa os resultados obtidos pela aplicação de análises estatísticas aos dados obtidos com o TCLP, tornando evidentes novas informações úteis.

O TCLP tem como principal propósito comparar o desenvolvimento de uma criança com o desempenho médio de crianças em seu mesmo nível de escolaridade. Entretanto, como notado por Capovilla, Varanda e Capovilla (2006) e por Macedo, Capovilla, Nikaedo, *et al* (2005), certas condições presentes nos indivíduos que respondem ao teste, como surdez e diversos tipos de dislexia, alteram significativamente os resultados que essas crianças fornecem. Uma análise cuidadosa desses resultados permite a observação de tendências notáveis no comportamento desses dados. Notando-se que a identificação de padrões e tendências em um conjunto de dados é um problema típico da IA, a aplicação dessas técnicas no tratamento dos resultados do TCLP oferece o potencial para se encontrar esse conhecimento no conjunto de dados de maneira automatizada.

1.5 Objetivos

Com o escopo da pesquisa assim delimitado, podemos definir seu objetivo geral:

1.5.1 Objetivo Geral

Aplicar técnicas de mineração de dados no estudo dos resultados obtidos da aplicação do TCLP.

1.5.2 Objetivos Específicos

As técnicas de mineração de dados podem ser divididas em algumas tarefas. Neste estudo, foram escolhidas três tarefas. Assim, os objetivos específicos são a aplicação de cada uma dessas tarefas. Elas são:

- Tarefa de classificação;
- Tarefa de agrupamento;
- Tarefa de extração de regras de associação.

O significado de cada uma dessas tarefas será abordado ao longo desta dissertação.

1.6 Justificativa

Ainda na graduação, o autor foi apresentado a uma nova abordagem para a solução de problemas normalmente difíceis de serem resolvidos, a Inteligência Artificial. A IA promove uma mudança de paradigmas de programação e mesmo de raciocínio lógico (RUSSELL; NORVIG, 2004; REZENDE, 2005; 2003; FAUSETT, 1994) que rapidamente seduziram o autor, o qual decidiu assumir essa especialização como sua principal linha de pesquisa acadêmica. Estudantes de IA muitas vezes se interessam também por Psicologia. Historicamente, a Psicologia e a Neurologia serviram de inspiração para as primeiras pesquisas em IA (RUSSELL; NORVIG, 2004). Por isso a IA empresta diversos termos dessas áreas de estudo, como aprendizado, reação, redes neurais, comportamento entre outros. O autor, além de compartilhar desse interesse natural por Psicologia, também experimentou um longo tratamento psicoterapêutico que lhe proporcionou uma extensa experiência pessoal e conhecimento empírico nessa área, aguçando ainda mais seu interesse por Psicologia. Existe no CEETEPS um projeto de pesquisa que envolve o estudo de diversos testes psicológicos com o uso de ferramentas da Ciência da Computação. Desse modo torna-se natural a elaboração dessa dissertação inserida no projeto de pesquisa desenvolvido no CEETEPS, aplicando técnicas de Mineração de Dados, uma parte da IA, que visa complementar a análise já realizada na aplicação do Teste de Competência em Leitura de Palavras, uma atividade relacionada à Psicologia e à Educação.

O aspecto interdisciplinar desse trabalho, que aplica técnicas de MD para complementar a avaliação tradicional dos resultados obtidos com a aplicação do TCLP traz uma nova perspectiva à

condução do estudo e da aplicação da Psicologia. O uso da Ciência da Computação ainda é recente na aplicação prática da Psicologia (MACEDO; CAPOVILLA; NIKAEDO; *et al*, 2005), expondo uma lacuna de conhecimento que essa dissertação pretende ocupar. Academicamente, esse trabalho propõe-se a integrar aspectos de duas ciências, oferecendo a ambas uma oportunidade de crescimento. Para a Ciência da Computação, é introduzida uma nova aplicação para a MD; enquanto que para a Psicologia é apresentada uma nova ferramenta para complementar a avaliação tradicional dos resultados obtidos com a aplicação do TCLP, sendo facilmente adaptável para o uso com outros testes.

O TCLP em sua interpretação tradicional já desempenha um papel importante no acompanhamento do aprendizado de crianças do Ensino Infantil e do Ensino Fundamental (MACEDO; CAPOVILLA; NIKAEDO; *et al*, 2005; CAPOVILLA; CAPOVILLA; VIGGIANO; *et al*, 2005). A natureza dos resultados obtidos com a aplicação desse teste é adequada à aplicação de técnicas de MD, a qual tem como maior mérito a identificação de propriedades não observáveis com a análise estatística tradicional (REZENDE, 2005; 2003). Essas propriedades, assim expostas com o uso da MD, podem ampliar o valor dos resultados da aplicação desse teste, ao incrementar a quantidade e a qualidade das informações obtidas por ele. Com isso, há a possibilidade da ampliação do escopo de aplicação do TCLP para além do universo para o qual fora desenvolvido ao se identificar informações que não foram originalmente procuradas no desenvolvimento do teste.

1.7 Organização

Este trabalho está organizado da seguinte maneira: No capítulo 2 o Teste de Competência de Leitura de Palavras é explicado em maiores detalhes. No capítulo 3 é discutida a Mineração de Dados e o processo de aplicação de suas técnicas. No capítulo 4 os resultados da aplicação das técnicas de MD aos resultados do TCLP são registrados e discutidos. As conclusões são discutidas no capítulo 5.

2 O Teste de Competência em Leitura de Palavras

O TCLP avalia o desenvolvimento da leitura ao longo das etapas de aprendizado. Trata-se de um teste que, em sua versão tradicional é aplicado por meio de papel e lápis (CAPOVILLA; VARANDA; CAPOVILLA, 2006), mas possui uma versão eletrônica aplicada via Internet (MACEDO; CAPOVILLA; NIKAEDO; *et al*, 2005). O teste é composto de 8 itens de treino e 70 itens de teste reunidos num caderno de aplicação. Cada item é composto de uma figura e um elemento escrito. Esse elemento escrito pode ser uma palavra correta ou uma pseudopalavra. Pseudopalavras são seqüências de caracteres que compõe um todo pronunciável, mas que não possui um significado. A tarefa do examinado é circular os itens corretos e marcar com um "X" os itens incorretos.

Os 70 itens do TCLP são formados por 7 tipos de pares figura-palavra distribuídos aleatoriamente ao longo do teste, com dez itens de teste para cada tipo. Dois dos 7 tipos são compostos por pares de figura-palavra em que a palavra escrita é correta. Os demais tipos são compostos de pares de figuras-pseudopalavras, cada um representando um tipo diferente de erro. Cada erro tem a finalidade de verificar diferentes estratégias de leitura e mostrar possíveis falhas no processo de aprendizagem.

Os 2 tipos com itens corretos são: 1) palavras corretas regulares (cr), como FADA sob a figura de uma fada; 2) palavras corretas irregulares (ci), como TÁXI, sob a figura de um táxi. Os cinco 5 tipos com itens incorretos são: 3) palavras semanticamente incorretas, que diferem das figuras às quais estão associadas, ou seja, vizinhas semânticas (ts), como RÁDIO, sob a figura de um telefone; 4) pseudopalavras estranhas (pe), como MELOCE sob a figura de um palhaço; 5) pseudopalavras homófonas (ph), como JÊNIU sob a figura de um gênio; 6) Pseudopalavras pseudo-homógrafas com trocas fonológicas, ou seja, vizinhas fonológicas (tf), como MÁCHICO sob a figura de um mágico; 7) Pseudopalavras pseudo-homógrafas com trocas visuais, ou seja, vizinhas visuais (tv), como TEIEUISÃO, sob a figura de uma televisão (CAPOVILLA *et al*, 2006; MACEDO *et al*, 2005). A Figura 1 Mostra alguns exemplos destes tipos de questões.

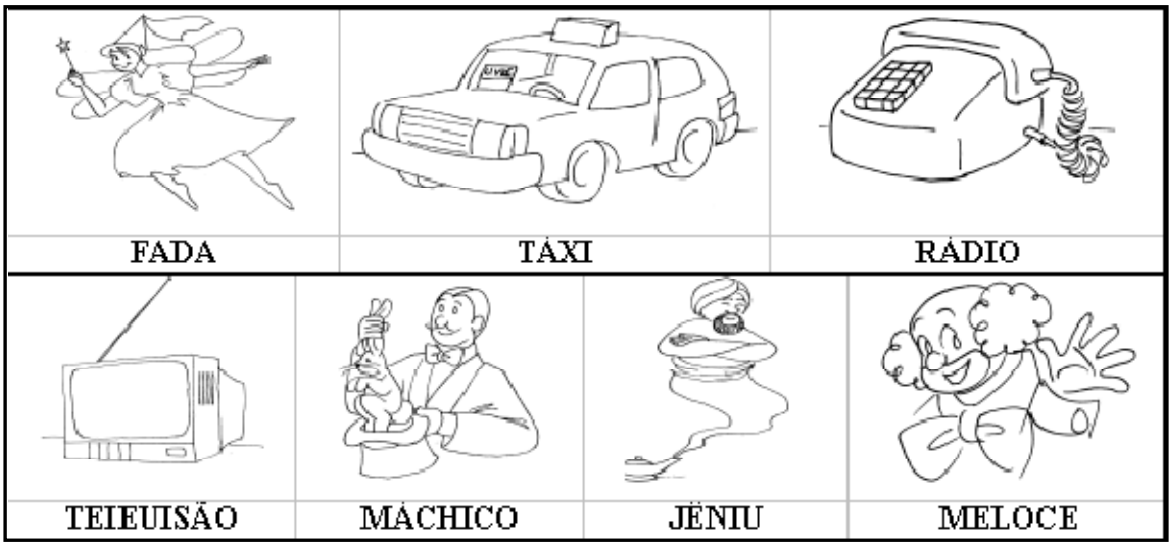

Figura 1: Exemplos dos diferentes tipos de questões que compõe o TCLP.

O TCLP é acompanhado de tabelas de normatização para avaliar o grau de desvio entre o padrão de leitura de um examinado e o padrão de leitura normal de seu grupo de referência de acordo com o nível de escolaridade e idade.

2.1 As Fases do Processo de Alfabetização

O processo de alfabetização se dá em três estágios (CAPOVILLA; CAPOVILLA; VIGGIANO; *et al*, 2005; CAPOVILLA; JOLY; FERRACINI; *et* al, 2004). O primeiro é o *logográfico*, em que o aluno trata a palavra escrita como se fosse uma representação pictoideográfica e visual; o segundo é o *alfabético*, em que, com o desenvolvimento da rota fonológica, o aluno aprende a fazer a decodificação grafo-fonêmica; e o *ortográfico*, em que, com o desenvolvimento da rota lexical, o aluno aprende a fazer a leitura visual direta de palavras de alta freqüência. Nota-se que, uma vez que o aluno passa de uma fase à seguinte, as fases anteriores não são abandonadas. Elas apenas ocorrem em menor freqüência e importância. Assim, as estratégias não são mutuamente excludentes, e podem coexistir simultaneamente no leitor e no escritor competente. Por exemplo, materiais como algarismos matemáticos e sinais de trânsito tendem a ser lidos pela estratégia logográfica. Já palavras novas precisam ser lidas pela estratégia fonológica. Finalmente, palavras conhecidas e familiares, ou de composição morfológica evidente, podem ser lidas mais rapidamente pela estratégia lexical de reconhecimento visual direto.

Capovilla, Varanda e Capovilla (2006) explicam que o TCLP foi desenvolvido com o propósito de identificar em qual estágio de alfabetização um aluno se encontra. Assim, determinados tipos de erro no teste sugerem dificuldades no processamento lexical, fonológico ou mesmo no logográfico.

Capovilla, Joly, Ferracini *et al* (2004) ressaltam que é fundamental conhecer as estratégias de leitura pois, nos distúrbios de leitura, pode haver alterações específicas em uma ou mais dessas estratégias com diferente impacto no diagnóstico da dislexia. A dislexia é um transtorno específico de aprendizagem, de origem neurobiológica. Ela é caracterizada pela dificuldade na correta e/ou fluente leitura de palavras, na escrita e nas habilidades de decodificação. Estas dificuldades são tipicamente decorrentes de um déficit no componente fonológico da linguagem que freqüentemente não é esperado em relação a outras habilidades cognitivas e à provisão de adequada instrução escolar. As conseqüências secundárias podem incluir problemas na compreensão de leitura sendo que a redução de experiência com leitura pode impedir a ampliação do vocabulário e do conhecimento geral. São identificam dois tipos clássicos de dislexia: dislexia fonológica e a dislexia morfêmica. Na dislexia fonológica há dificuldades na leitura pela transformação da letra em seus sons, porém, a leitura pelo reconhecimento visual da palavra está preservada. Logo, há dificuldades na leitura de pseudopalavras e palavras desconhecidas, mas a leitura de palavras familiares é adequada. Já na dislexia morfêmica há dificuldades na leitura pelo reconhecimento visual da palavra, sendo a leitura feita principalmente pela transformação das letras em seus sons. Logo, há dificuldades na leitura de palavras irregulares e longas, com regularizações (FRITH, 1985 *apud* GUNJI; ALMEIDA; DUDUCHI; *et al*, 2008; LYON, 2003 *apud* GUNJI; ALMEIDA; DUDUCHI; *et al*, 2008).

3 A Mineração de Dados

Rezende (2005; 2003) explica que MD é uma área multidisciplinar que incorpora técnicas utilizadas em diversas áreas da Inteligência Artificial (IA) e que sua finalidade é a "extração de conhecimento previamente desconhecido, implícito e potencialmente útil a partir de dados" (REZENDE, 2005, pg. 2). Em outras palavras: "O foco central de MD é o de como transformar dados armazenados em conhecimento, expresso em termos de formalismos de representação, tal como regras e relação entre dados" (REZENDE, 2005, pg. 2).

Chen, Han e Yu (1996) explicam que, para se conduzir uma mineração de dados efetiva, deve-se primeiro examinar que características um sistema de descoberta de conhecimento deve possuir e quais desafios são enfrentados ao se desenvolver técnicas de mineração de dados:

1. **Manuseio de diferentes tipos de dados**: Há muitos tipos de dados usados em diferentes aplicações. Deve-se esperar que um sistema de descoberta de conhecimento seja capaz de funcionar em um conjunto diverso de tipos de dados. Como a maioria dos bancos de dados é relacional, é essencial que um sistema de mineração de dados funcione eficientemente e eficazmente com dados relacionais. Além disso, muitos bancos de dados de interesse possuem tipos de dados complexos, como dados estruturados, hipertexto, dados em multimídia, dados temporais e espaciais. Um sistema poderoso deve ser capaz de lidar com esses tipos de dados também. Entretanto, a diversidade de tipos de dados e diferentes objetivos de mineração de dados torna irrealista a esperança de que um sistema de mineração de dados seja capaz de manipular todos estes tipos de dados. Assim, sistemas específicos devem ser construídos para mineração de conhecimento de tipos específicos de dados;
2. **Eficiência e escalabilidade de algoritmos de mineração de dados**: Para se extrair informação de quantidades elevadas de dados, os algoritmos de descoberta de conhecimento devem ser eficientes e escaláveis a grandes bancos de dados. Isso equivale a dizer que o tempo de processamento de um algoritmo deve ser previsível e aceitável em grandes bancos de dados. Algoritmos com complexidade exponencial, ou mesmo polinomial de média ordem, não serão de uso prático;
3. **Utilidade, confiabilidade e significância dos resultados da mineração de dados**: O conhecimento descoberto deve retratar com fidelidade o conteúdo do banco de dados e ser útil para certas aplicações. As imperfeições devem ser expressas por

meio de medidas de incerteza, na forma de regras aproximadas ou de regras qualitativas. Ruído e dados excepcionais devem ser tratados elegantemente por sistemas de mineração de dados;

4. **Expressão de vários tipos de resultados de mineração de dados**: Diferentes tipos de conhecimento podem ser descobertos de uma quantidade grande de dados. Além disso, pode-se desejar examinar o conhecimento descoberto de ângulos diferentes e apresentá-lo de maneiras diferentes. Isso requer que se expresse tanto as requisições de mineração de dados quanto o conhecimento descoberto em linguagens de alto nível ou interfaces gráficas, de modo que a tarefa de mineração de dados possa ser especificada por não especialistas e o conhecimento descoberto compreensível e diretamente utilizável pelos usuários;
5. **Mineração de conhecimento interativa em níveis diferentes de abstração**: Como é difícil prever o que exatamente pode ser descoberto de um banco de dados, uma consulta de mineração de dados de alto nível deve ser tratada como uma sondagem que pode revelar alguns aspectos para exames mais aprofundados. Descoberta interativa deve ser encorajada, de modo a permitir que o usuário possa refinar interativamente uma consulta de mineração de dados, mudar dinamicamente o foco dos dados, aprofundar progressivamente o processo de mineração de dados e apresentar os dados e os resultados da mineração de dados de maneira flexível, em múltiplos níveis de abstração e de diferentes ângulos;
6. **Minerar informação de fontes de dados diferentes**: As redes de computadores, tanto locais quanto distribuídas, incluindo a Internet, conectam muitas fontes de dados e compõe um banco de dados imenso e distribuído. Minerar conhecimento de fontes diferentes, de dados formatados ou não, com semânticas de dados diversas apresenta novos desafios à mineração de dados. Por outro lado, minerar dados pode ajudar a revelar regularidades de alto nível em bancos de dados heterogêneos, o que pode ser difícil de se conseguir por simples sistemas de consulta. E ainda, o grande tamanho dos bancos de dados, a ampla distribuição dos dados e a complexidade computacional de alguns métodos de mineração de dados motivam o desenvolvimento de algoritmos paralelos e distribuídos de mineração de dados;
7. **Proteção à privacidade dos dados e segurança**: Quando dados podem ser vistos de muitos ângulos diferentes e em níveis diferentes de abstração, isto ameaça o objetivo de se proteger a segurança dos dados e evitar a invasão de privacidade. É importante estudar quando a descoberta de conhecimento pode levar à invasão de

privacidade, e que medidas de segurança podem ser desenvolvidas para se evitar a disponibilização de informação privilegiada.

Chen, Han e Yu (1996) ressaltam que alguns desses requisitos podem possuir objetivos conflitantes. Por exemplo, o objetivo de se proteger a segurança dos dados pode entrar em conflito com o requisito de mineração interativa de níveis diferentes de conhecimento e em ângulos diversos.

O processo de MD pode ser dividido em três grandes etapas (REZENDE, 2005; 2003): Pré-processamento, extração de padrões e pós-processamento. Também se pode incluir nessa divisão uma fase anterior ao processo de MD, que se refere ao conhecimento do domínio e a identificação do problema, e uma fase posterior ao processo, que se refere à utilização do conhecimento obtido. A Figura 2 ilustra estas etapas.

Figura 2: Fases do processo de Mineração de Dados (REZENDE, 2005).

Inicia-se o processo de MD com o entendimento do domínio da aplicação, considerando aspectos como os objetivos dessa aplicação e as fontes de dados. A seguir, é feita uma seleção de dados a partir dessas fontes, de acordo com os objetivos do processo. Os conjuntos de dados resultantes dessa seleção são pré-processados, ou seja, recebem um tratamento para poderem ser submetidos aos métodos e ferramentas na etapa de extração de padrões.

A etapa de extração de padrões tem como produto usual a obtenção de preditores que, para um dado problema, fornecem uma decisão. Outro produto usual da MD é a obtenção de uma descrição de características intrínsecas nos dados. Essa descrição pode revelar propriedades não evidentes no conjunto de dados ou em subconjuntos dele.

Na etapa seguinte, a de pós-processamento, o conhecimento é avaliado quanto a sua qualidade e/ou utilidade para que, em caso positivo, seja utilizado para apoio a algum processo de tomada de decisão.

> *É importante notar que, por ser um processo eminentemente iterativo, as etapas da Mineração de Dados não são estanques, ou seja, a correlação entre as técnicas e métodos utilizados nas várias etapas é considerável, a ponto da ocorrência de pequenas mudanças em uma delas afetar substancialmente o sucesso de todo o processo. Portanto, os resultados de uma determinada etapa podem acarretar mudanças a quaisquer das etapas posteriores ou, ainda, o recomeço de todo o processo (REZENDE, 2005, p. 10).*

A escolha da tarefa de MD é feita de acordo com os objetivos desejáveis para a solução a ser encontrada. As tarefas possíveis de um algoritmo de extração de padrões podem ser agrupadas em atividades preditivas e descritivas. As atividades de predição consistem na generalização de exemplos ou experiências passadas com respostas conhecidas em um modelo capaz de reconhecer a classe de um novo exemplo. Os dois principais tipos de tarefas para predição são classificação e regressão. As atividades descritivas consistem na descoberta de comportamentos notáveis e recorrentes no conjunto de dados. Com este resultado, pode-se observar propriedades não evidentes nos dados brutos. Os dois tipos de tarefas para descrição são o agrupamento e as regras de associação.

A escolha do algoritmo é realizada de forma subordinada à linguagem de representação dos padrões a serem encontrados. Pode-se utilizar algoritmos indutores de árvores de decisão ou regras de produção, por exemplo, se o objetivo é realizar uma classificação.

A extração de padrões consiste da aplicação dos algoritmos de mineração escolhidos para a extração dos padrões embutidos nos dados.

O conhecimento extraído da aplicação do algoritmo de MD pode ser utilizado na resolução de problemas na vida real, seja por meio de um Sistema Inteligente ou de um ser humano como apoio a algum processo de decisão.

Rezende (2003; 2005) ainda observa que há essencialmente dois estilos para se fazer MD: *top-down* e *bottom-up*. No estilo *top-down*, o processo é iniciado com alguma hipótese a ser verificada. Nesse caso, em geral, é desenvolvido um modelo e este é então avaliado para se determinar se a hipótese é válida ou não. No estilo *bottom-up*, não é especificada uma hipótese para validação, apenas são extraídos padrões dos dados. Ainda neste estilo, a abordagem pode ser

supervisionada, que é quando se tem alguma idéia do que se está procurando, como também pode ser não-supervisionada, que é quando não se tem idéia do que se está procurando.

3.1 Tarefa de Classificação

Uma hipótese que pode ser feita sobre um conjunto de dados é a de que ele segue um padrão ou uma tendência. A tarefa de classificação da Mineração de Dados busca um descritor capaz de descrever este padrão ou tendência. Assim, de posse de um descritor obtido a partir de um conjunto de dados que represente um comportamento previsível, pode-se classificar um novo exemplo (QUINLAN, 1996). Para se construir um descritor, um banco de dados *E* é utilizado como um *conjunto de treinamento*, no qual cada linha de sua tabela consiste do mesmo conjunto de atributos múltiplos que as linhas em um grande banco de dados *W*. Adicionalmente, cada linha deve possuir uma identidade de *classe*, um atributo que especifica qual a classe a que este elemento de dado pertence. O objetivo da classificação é, primeiro, analisar o conjunto de treinamento e desenvolver um descritor ou modelo para cada classe usando os atributos disponíveis nos dados. Tal descritor é usado para classificar dados futuros no banco de dados *W* (CHEN; HAN; YU, 1996).

Um algoritmo classificador precisa ter seu desempenho aferido. Uma estratégia popular para isso é a v*alidação cruzada* (CHAWLA; CIESLAK; HALL; *et al*, 2008). Na validação cruzada, o conjunto de treinamento é dividido em *n* partes. Um classificador é obtido treinando-se contra *n*-1 dessas partes. Este classificador é então avaliado contra a parte restante e seu desempenho é registrado. Um novo classificador é treinado, mas dessa vez com um conjunto diferente de *n*-1 partes e então avaliado contra a parte restante. O processo é repetido *n* vezes, até que se tenha testado o desempenho do algoritmo contra o conjunto todo de treinamento.

3.1.1 Melhorando o Desempenho de um Classificador

A qualidade do classificador pode ser aumentada de algumas maneiras. Por exemplo, o algoritmo classificador pode ser aprimorado para casos específicos. Quinlan (1996) propõe um aprimoramento de seu popular algoritmo de classificação por árvores de decisão, o C4.5, para utilizar mais eficientemente atributos contínuos. Outras situações podem ser identificadas: Classificadores costumam funcionar bem quando classificam dados com um comportamento que segue o comportamento da maioria, ou seja, não são comportamentos raros. Mas dados que representam eventos raros podem ser de particular interesse, por exemplo, dados relacionados a fraudes, doenças, falhas mecânicas e regiões de interesse em simulações de grande escala. Chawla, Cieslak, Hall, *et al* (2008) exploram uma técnica para melhorar o desempenho de classificadores empregados

para identificar eventos raros deste tipo. Esta técnica altera os dados de treinamento do classificador executando duas atividades: A primeira é a sub-amostragem dos dados que representam comportamento majoritário; a segunda é a introdução de dados sintéticos representando comportamento minoritário. Deste modo, os comportamentos majoritários e minoritários passam a ter freqüências de ocorrência mais aproximadas, tornando o classificador mais equilibrado para os dois casos.

Sendo o algoritmo aprimorável ou não, ainda é possível utilizar técnicas de combinação de classificadores (*ensembles*). Combinações são uma coleção de classificadores que podem ser homogêneos (mesmo algoritmo classificador) ou heterogêneos (algoritmos diferentes) que, em conjunto, realizam a tarefa de classificação (GARCIA-PEDRAJAS; GARCIA-OSORIO; FYFE, 2007). A maioria das técnicas de combinação de classificadores procura encontrar uma combinação de classificadores que sejam tão exatos quanto possível, mas que também discordem entre si o quanto for possível. Combinações de classificadores podem ser criadas das seguintes maneiras:

- Usando-se esquemas de combinação diferentes;
- Usando-se modelos de classificação diferentes;
- Usando-se subconjuntos diferentes de atributos;
- Usando conjuntos de treinamento diferentes.

Provavelmente a última maneira de se criar combinações de classificadores seja a mais freqüente (GARCIA-PEDRAJAS; GARCIA-OSORIO; FYFE, 2007). Os métodos que usam conjuntos de treinamento diferentes podem ser divididos em dois grupos:

- Os que mudam a distribuição do conjunto de treinamento de forma adaptativa, baseando-se no desempenho dos classificadores anteriores. Métodos de Boosting são mais representativos deste grupo. Ex: AdaBoost e suas variantes e Arc-X4;
- Os que não adaptam a distribuição. Bagging é o algoritmo mais representativo deste grupo.

3.2 Tarefa de Agrupamento

Agrupamento de dados, também chamado de classificação não supervisionada, é uma tarefa de mineração onde os dados são agrupados de acordo com o seguinte princípio: Deve-se maximizar a similaridade dos elementos de uma mesma classe e minimizar a similaridade entre os elementos de classes diferentes (CHEN; HAN; YU, 1996). A tarefa de agrupamento é útil para a construção de partições significativas de um conjunto grande de objetos baseado em uma metodologia de “dividir

para conquistar", que decompõe um sistema de grande escala em componentes menores para simplificar seu entendimento.

A tarefa de agrupamento identifica regiões densamente populadas, de acordo com algum tipo de medida de distância, em um conjunto de dados multidimensional e grande. Dado um grande conjunto de pontos multidimensionais, o espaço de dados normalmente não é ocupado uniformemente pelos pontos. O agrupamento de dados identifica regiões ocupadas de maneira esparsa e densa e, assim, descobre os padrões de distribuição do conjunto de dados.

3.2.1 Seleção de Exceções

Uma variação útil da tarefa de agrupamento é a identificação de exceções. Exceções são dados que apresentam um comportamento atípico quando comparados ao comportamento do resto da população. Este tipo de dado pode indicar ruído, falhas na coleta de dados ou conteúdo malicioso, mas sua identificação vem sendo usada para tarefas como limpeza de dados, detecção de fraudes e invasões. Um exemplo é o uso desta tarefa que companhias de cartão de crédito vêm fazendo atualmente: Algoritmos de seleção de exceções vêm sendo usados para detectar usos de cartão de crédito fora do padrão de comportamento habitual de seu dono. Assim, é possível detectar quando um cartão foi roubado ou clonado. Ghothing, Parthasarathy e Otey (2008) apresentam uma técnica para identificar exceções em conjuntos de dados com elevado número de dimensões. O primeiro passo é a escolha de uma definição de exceção baseada em distância. Elas podem ser:

- Exceções são pontos para os quais há menos do que *p* outros pontos com distância menor que *d*;
- Exceções são os *n* pontos com a maior distância a seu *k*-ésimo vizinho;
- Exceções são os *n* pontos com a maior distância média a seus *k* vizinhos.

A seguir os dados são separados em diversos containeres (bins) de modo que pontos que estão próximos mutuamente tendam a se concentrar no mesmo container. São arbitrados *k* centros aleatórios. Cada ponto é atribuído a seu centro mais próximo, gerando *k* partições. Então, são encontrados *k* centros para cada *k* partição e prossegue-se iterativamente por um número fixo de iterações. Após isso, prossegue-se recursivamente caso o tamanho das partições seja maior que um valor definido pelo usuário. Tal estratégia garante que pontos próximos tendam a ser colocados no mesmo container. Cada container é vasculhado seqüencialmente para se encontrar os vizinhos aproximadamente mais próximos de cada ponto de dados. É usada uma extensão do algoritmo *Nested Loops* para se encontrar exceções no conjunto de dados que foi organizado em container. Para cada ponto, inicia-se procurando por vizinhos mais próximos dentro de seu próprio container. Se o container todo foi vasculhado e não se encontraram *k* vizinhos aproximadamente mais

próximos, prossegue-se vasculhando o próximo container. Esta busca continua iterativamente até que *k* vizinhos aproximadamente mais próximos tenham sido descobertos.

3.3 Tarefa de Mineração de Regras de Associação

Minerar regras de associação em um banco de dados é obter um conjunto de regras na forma "$A_1 \wedge \cdots \wedge A_m \Longrightarrow B_1 \wedge \cdots \wedge B_n$", onde A_i(para $i \in \{1, \ldots, m\}$) e B_j(para $j \in \{1, \ldots, n\}$) são pares de atributo e valor dos conjuntos de dados relevantes no banco de dados. Por exemplo, de um grande conjunto de dados transacional pode-se descobrir uma regra de associação que diz que se um consumidor compra leite, ele costuma comprar também pão na mesma transação (CHEN; HAN; YU, 1996).

A seguir é feita a definição formal do problema de mineração de regras de associação, como especificadas por Agrawal, Imielinshi e Swami (1993) e por Agrawal e Srikant (1994): Seja $\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$ um conjunto de literais, chamados itens. Seja $\mathcal{D}$ um conjunto de transações, onde cada transação T é um conjunto de itens tais que $T \subset \mathcal{I}$. Associado a cada transação há um identificador único, chamado TID. Diz-se que a transação T contém X, um conjunto de alguns itens em $\mathcal{I}$, se $X \subseteq T$. Uma regra de associação é uma implicação na forma $X \Longrightarrow Y$, onde $X \subset \mathcal{I}$, $Y \subset \mathcal{I}$ e $X \cap Y = \emptyset$. A regra $X \Longrightarrow Y$ é válida no conjunto de transações $\mathcal{D}$ com confiança c se em $c\%$ das transações em $\mathcal{D}$ que contém X também contêm Y. A regra $X \Longrightarrow Y$ tem suporte s se em $s\%$ das transações em $\mathcal{D}$ existe $X \cup Y$.

Dado um conjunto de transações $\mathcal{D}$, a tarefa de mineração de regras de associação é gerar todas as regras de associação que possuam um suporte e uma confiança maiores do que um suporte mínimo e uma confiança mínima especificadas pelo usuário.

3.3.1 Aprimorando a Tarefa de Mineração de Regras de Associação

Um problema encontrado no final do processo de mineração de regras de associação é que muitos algoritmos geram uma enorme quantidade de padrões, dificultando consideravelmente sua análise (DOMINGUES; REZENDE, 2004). Uma abordagem para se minimizar este problema é o uso de taxonomias. Taxonomias refletem uma caracterização coletiva ou individual de como os itens podem ser hierarquicamente classificados. A Figura 3 representa um exemplo de taxonomia de itens de dados transacionais.

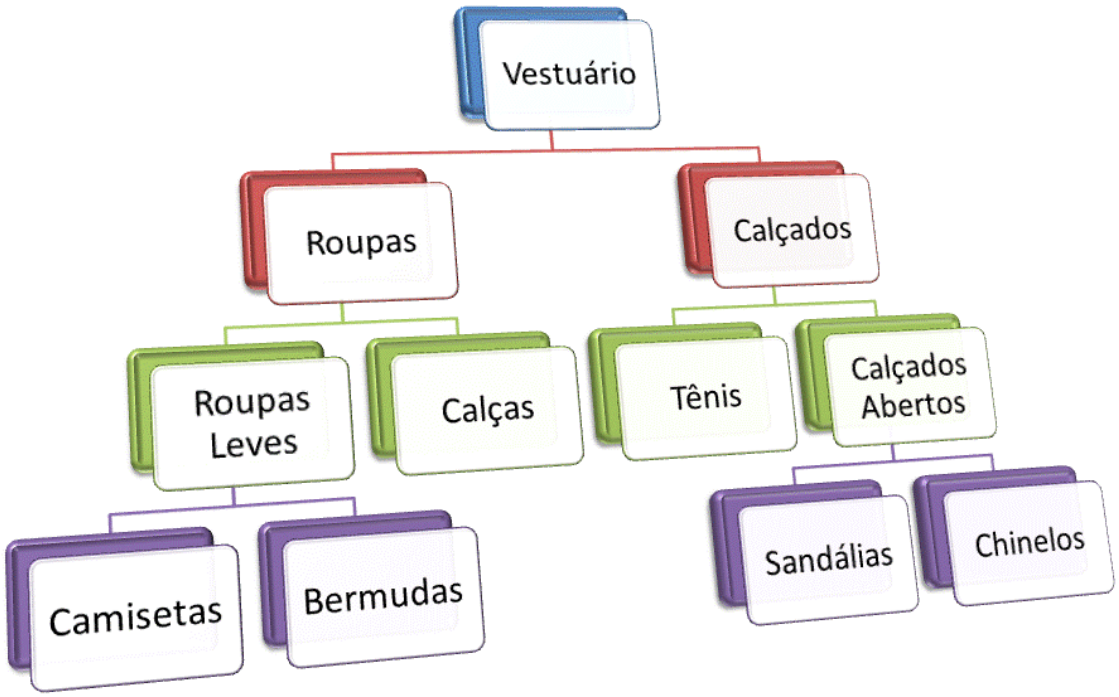

Figura 3: Exemplo de uma taxonomia de itens de dados transacionais. Fonte: O autor.

Regras simples (cujos elementos são compostos apenas por itens terminais na taxonomia) podem não ter suporte suficiente para serem incluídas na solução, mas podem representar conhecimento interessante ao serem agrupadas segundo uma taxonomia. Regras muito específicas podem ser generalizadas para melhorar sua compreensibilidade. Finalmente, regras interessantes podem ser identificadas com o uso de informações contidas nas taxonomias. Por exemplo, observe as regras de associação listadas no Quadro 1

camiseta & chinelo ⇒ boné
camiseta & sandália ⇒ boné
bermuda & sandália ⇒ boné
bermuda & chinelo ⇒ boné

Quadro 1: Exemplo de regras de associação. Fonte: O autor.

Em uma primeira etapa, pode-se aplicar a taxonomia que especifica que camisetas e bermudas são roupas leves (vide Figura 3). Assim, as quatro regras originais são simplificadas em apenas duas, como é mostrado na Figura 4.

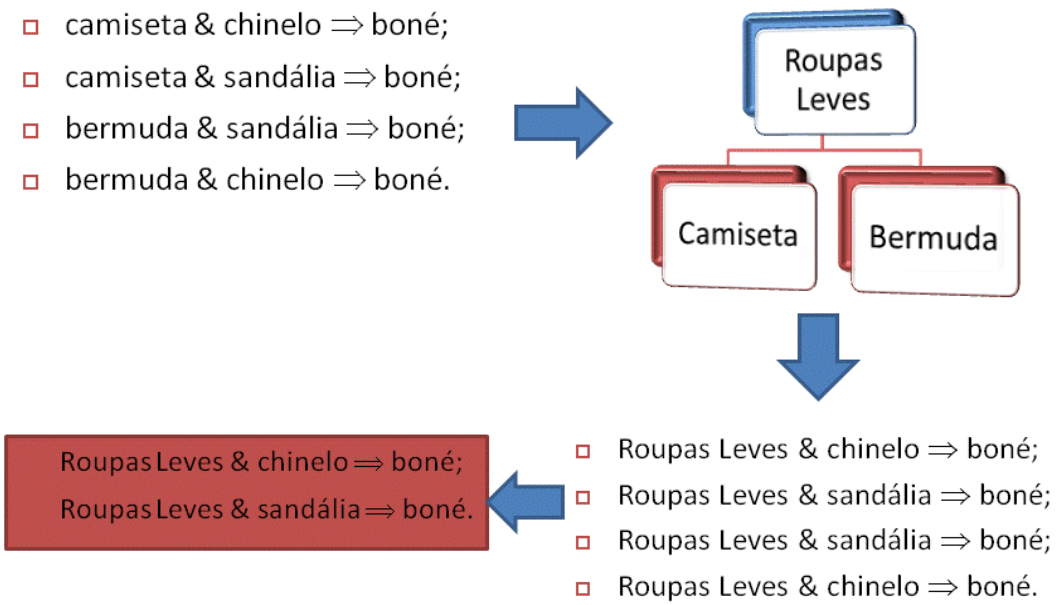

Figura 4: Primeira simplificação das regras de associação usando taxonomias. Fonte: O autor.

A seguir, as novas regras podem ser simplificadas novamente, notando que pela taxonomia, sandálias e chinelos são calçados abertos. Desta simplificação resulta uma nova regra de associação, como é mostrado na Figura 5.

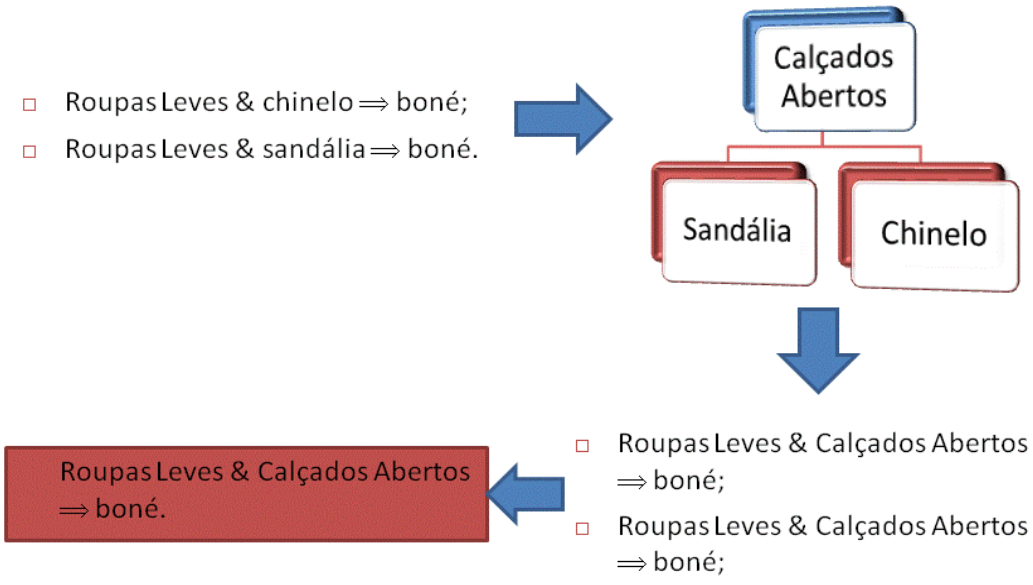

Figura 5: Segunda simplificação das regras de associação usando taxonomias. Fonte: O autor.

Os algoritmos de mineração de regras de associação procuram relações com valores elevados de suporte e confiança, visando a seleção de regras com grande força (elevada confiança) e grande significância estatística (elevado suporte). Romão, Niederauer, Martins *et al* (1999) propõe uma

abordagem diferente: Em seu artigo, eles introduzem o conceito de *suporte máximo*. Limitando o valor do suporte das regras de associação encontradas de modo que sejam **menores** que o valor especificado, seu algoritmo encontra regras que representam comportamentos de exceção.

4 Análise dos Dados

A seguir são registrados os resultados da análise por Mineração de Dados de um conjunto de resultados da aplicação do TCLP.

4.1 Método

Foram utilizados os dados obtidos da aplicação do TCLP a alunos de diversas escolas da região metropolitana de São Paulo. São estudantes de 1ª a 8ª série do Ensino Fundamental da rede pública e particular de ensino. A Tabela 1 ilustra a distribuição dos alunos que participaram do estudo quanto ao sexo e à série que cursavam.

Atributo	*Distribuição*			
aluno_sexo	***Valores***	***Contagem***	***Percentual***	***Histograma***
	F	777	46,55 %	
	M	892	53,45 %	
aluno_serie	***Valores***	***Contagem***	***Percentual***	***Histograma***
	1ª	766	45,90 %	
	2ª	430	25,76 %	
	3ª	212	12,70 %	
	4ª	177	10,61 %	
	8ª	84	5,03 %	

Tabela 1: Caracterização dos alunos participantes do TCLP. Fonte: O Autor.

Deste ponto em diante é utilizada a nomenclatura dos tipos de pares figura-palavra do TCLP em sua forma abreviada, a qual está registrada na Tabela 2.

Tipo	*Abreviação*
Correta Regular	cr
Correta Irregular	ci
Vizinha Semântica	ts
Vizinha Visual	tv
Vizinha Fonológica	tf
Pseudopalavra Homófona	ph
Pseudopalavra Estranha	pe

Tabela 2: Abreviações dos tipos de pares figura-palavra utilizadas nos resultados. Fonte: o autor.

Os dados dos resultados do TCLP estavam disponíveis em um arquivo usando uma notação transacional. Nesta notação, cada linha do arquivo representava o resultado a uma única questão do teste de um determinado aluno. Para que os dados fossem utilizados em mineração de dados, foi necessária a tradução desta notação transacional em uma mais apropriada. Nesta nova notação, cada linha do arquivo obtido representa todos os dados disponíveis de uma aplicação do TCLP a um aluno. Em outras palavras, cada linha deste novo arquivo apresenta os dados pessoais de um aluno, suas respostas às questões individuais do teste e a consolidação de seu resultado por tipo de questão. Ao final da tradução obteve-se um total de 1669 linhas, cada uma representando uma aplicação do TCLP a um aluno. Na Tabela 3 pode-se observar o resumo estatístico do desempenho dos alunos na amostra estudada. Nesta tabela nota-se que o desempenho dos alunos em cada tipo de teste está com média acima de 5. Também nota-se que a média de questões respondidas aproxima-se bastante do total de questões presentes no teste.

Atributo	*Mínimo*	*Máximo*	*Média*	*Desvio Padrão*	*Desv. Padr./Média*
total_cr	0	10	7,1049	3,2094	0,4517
total_ci	0	10	6,6016	2,9970	0,4540
total_ts	0	10	7,8999	2,9382	0,3719
total_tv	0	10	6,3535	3,2018	0,5039
total_tf	0	10	5,8766	3,1406	0,5344
total_ph	0	10	4,8910	3,1953	0,6533
total_pe	0	10	7,8850	2,9860	0,3787
total_respondidas	0	70	67,9341	9,4419	0,1390

Tabela 3: Caracterização estatística básica do desempenho dos alunos no TCLP. Fonte: O Autor.

Como nem todos os alunos responderam a todas as questões foi necessário o tratamento destes casos. As questões, então, passaram a ter três resultados possíveis: “Acerto”, “erro” e “não fez”. Os totais continuam sendo a soma dos acertos em cada categoria de teste. Também está disponível nos dados o tempo de resposta a cada questão. Entretanto, há ocorrências de testes em que o tempo não foi registrado. Durante a tradução, optou-se por marcar como tempo zero as questões em que a medição de tempo não foi feita, assim como os tempos de questões que não

foram respondidas. Deste modo, esta informação pode ser facilmente descartada durante a análise. Vale ressaltar que não foi feita uma análise dos dados usando as informações de tempo disponíveis nos dados. Tal análise fica sugerida para estudos futuros.

Para a atividade de Mineração de Dados, foi escolhido o sistema Tanagra. Este é um sistema integrado para análises estatísticas e de Mineração de Dados (RAKOTOMALALA, 2005). Nele, há uma coleção de algoritmos para diversas tarefas de mineração de dados. É um sistema de código aberto, o que significa que as implementações dos algoritmos presentes no sistema são de domínio público, permitindo seu estudo detalhado. Também por isso há a possibilidade de que se implemente novos algoritmos de MD utilizando a infra-estrutura disponibilizada pelo sistema. Seu uso é livre e gratuito para fins didáticos e de pesquisa, o que acaba sendo de grande valia para que os resultados apresentados nesta dissertação sejam facilmente repetidos pelo leitor. Os gráficos utilizados para a visualização de alguns dos dados obtidos foram feitos utilizando o sistema VisIt[1], o qual também é um sistema de código aberto de uso livre e gratuito. Os resultados fornecidos pelo sistema Tanagra precisaram sofrer uma tradução para que fossem corretamente interpretados pelo sistema VisIt. Todas as traduções de dados mencionadas nesta seção foram feitas pelo autor em linguagem de programação Java, a qual também é distribuída em modalidade de código aberto. As implementações dos programas de tradução, por serem de aplicação muito específica, não foram incluídas nesta dissertação, mas podem ser obtidas entrando-se em contato com o autor.

4.2 Resultados

Nas próximas seções são expostos e discutidos os resultados obtidos pela análise dos dados do Teste de Competência de Leitura de Palavras utilizando-se Mineração de Dados.

4.2.1 Tarefa de Classificação

A primeira tarefa executada no conjunto de dados foi a tarefa de classificação. Nela é feita a hipótese de que os dados disponíveis representam o comportamento típico da população de alunos que respondem ao TCLP. Esta tarefa obtém um classificador, um modo de se classificar um novo aluno de acordo com o exemplo oferecido pelo conjunto original de dados. O conjunto de dados utilizado pelo algoritmo para se obter um classificador é chamado de **conjunto de treinamento**.

Neste trabalho, a tarefa de classificação foi feita utilizando-se o algoritmo C4.5 (QUINLAN, 1996). Trata-se de um algoritmo clássico para esta tarefa e é freqüentemente utilizado como

[1] Disponível na Internet no endereço: https://wci.llnl.gov/codes/visit/home.html.

parâmetro de comparação para se aferir o desempenho de novos algoritmos. Aplica-se o algoritmo especificando-se um conjunto de parâmetros e um atributo de classificação. O resultado que se obtém da aplicação deste algoritmo é uma árvore de decisão. A árvore de decisão pode ser entendida como uma série de perguntas feitas sobre os parâmetros. De acordo com a resposta dada a cada pergunta obtém-se a classificação do novo elemento em uma classe definida pelo atributo de classificação.

Esta tarefa de MD é feita no estilo *bottom up*, pois não se está procurando averiguar a validade de nenhuma hipótese sobre os dados. O algoritmo é aplicado apenas para se obter um classificador (REZENDE, 2005; 2003).

O desempenho do algoritmo precisa ser aferido. Uma maneira de se fazer isso é utilizando-se a técnica de **validação cruzada**. Nesta técnica, o conjunto de treinamento é dividido em n partes, também chamadas de vias. O algoritmo é então aplicado a n-1 partes e o classificador obtido é testado contra a n-ésima parte. O processo é repetido n vezes. Além disso, a validação toda pode ser repetida outras vezes, dividindo o conjunto de treinamento em subconjuntos diferentes a cada repetição. Assim, o desempenho do algoritmo é aferido utilizando-se todo o conjunto de treinamento contra todo o conjunto de treinamento, mas de maneira disjunta. Os dados utilizados na fase de treinamento são diferentes dos dados utilizados na fase de teste. Isto se faz necessário porque, caso o desempenho do algoritmo seja testado contra o próprio conjunto de treinamento, o resultado desta aferição torna-se artificialmente mais elevado do que ele seria em um caso real (GARCIA-PEDRAJAS; GARCIA-OSORIO; FYFE, 2007; RAKOTOMALALA, 2005; REZENDE, 2003). Nesta dissertação foi feita a validação cruzada com 10 vias e 5 repetições.

A primeira aplicação do algoritmo de classificação foi feita com os parâmetros sexo, escola, total de respostas e total de acertos por categoria de questão. O atributo de classificação foi a série de cada aluno. O algoritmo foi aferido utilizando-se validação cruzada como descrita no parágrafo anterior e seu resultado é apresentado na Tabela 4.

Razão de erro	*0,4341*

Predição de valores

Valor	***Razão de acerto***
1a	0,7441
2a	0,4130
3a	0,2818
4a	0,3923
8a	0,8019

Matriz de confusão

	1a	***2a***	***3a***	***4a***	***8a***	Soma
1a	2838	682	169	57	68	3814
2a	815	880	241	175	20	2131
3a	245	254	297	235	23	1054
4a	91	209	228	346	8	882
8a	59	9	13	2	336	419
Soma	4048	2034	948	815	455	8300

Tabela 4: Matriz de confusão da primeira aplicação do algoritmo C4.5. Fonte: O autor.

A Tabela 4 é interpretada da seguinte maneira: A razão de erro indica o número de vezes em que o classificador erra dividido pelo número de elementos de teste. Na seção predição de valores, a razão de acerto é o número de vezes em que o classificador acerta a classificação dividido pelo número de elementos que realmente pertencem à esta classificação. Na matriz de confusão, as colunas representam o resultado do classificador e as linhas representam o valor real dos dados. Assim, na diagonal principal são listados os totais em que o classificador acerta a classificação. As demais células representam os erros. Tomando-se como exemplo a célula na coluna 2a e linha 4a, a matriz indica que o algoritmo classificou 209 testes como sendo de 2ª série, mas que na verdade representavam alunos de 4ª série. A razão de acerto indica que o algoritmo comporta-se bem apenas classificando alunos que estejam na 1ª e na 8ª série. Nas demais séries, o classificador erra mais do que acerta. Além disso, observando-se a árvore de decisão obtida quando o algoritmo é aplicado ao conjunto de treinamento inteiro, nota-se que a escola aparece logo na sua raiz. A árvore toda pode ser conferida no Apêndice 1. Como a razão de erro desta aplicação do algoritmo é elevada, optou-se por ressaltar no Quadro 2 apenas um trecho da árvore de decisão obtida. Ter a escola como primeiro parâmetro da árvore indica que a escola é o parâmetro mais importante na seleção da série dos alunos. Isto faz sentido quando nota-se que cada escola participou do teste com uma seleção de alunos de determinadas séries. Por exemplo, há escolas em que apenas participaram alunos da 1ª série, enquanto outras participaram com alunos das 4 primeiras séries. Assim conclui-se que esta árvore de decisão não é muito útil por dois motivos: Sua razão de erro é razoavelmente elevada e ela faz a classificação de alunos segundo um critério trivial, que acrescenta pouco conhecimento ao conjunto de testes.

- escola_id em [18.]
  - total_pe < 7,5000 então aluno_serie = **1a** (77,31 % de 119 exemplos)
  - total_pe >= 7,5000
    - total_ci < 4,5000 então aluno_serie = **1a** (75,00 % de 52 exemplos)
    - total_ci >= 4,5000
      - total_ci < 7,5000
        - total_ci < 5,5000 então aluno_serie = **1a** (53,85 % de 13 exemplos)
        - total_ci >= 5,5000
          - total_ph < 3,5000 então aluno_serie = **2a** (64,29 % de 14 exemplos)
          - total_ph >= 3,5000 então aluno_serie = **1a** (60,00 % de 15 exemplos)
      - total_ci >= 7,5000 então aluno_serie = **2a** (73,13 % de 67 exemplos)

Quadro 2: Trecho da árvore de decisão gerada pela primeira aplicação do algoritmo C4.5. Fonte: O autor.

A árvore de decisão, como o trecho representado pelo Quadro 2, é interpretada da seguinte maneira: O primeiro nó da árvore, a raiz, pergunta se escola_id é igual a 18. Caso a resposta seja "sim", segue-se para o nó seguinte. O nó seguinte pergunta se total_pe é menor do que 7,5. Caso a resposta seja "sim", então o aluno em teste pertence à 1ª série. Esta regra é verdadeira em 119 exemplos do conjunto de treinamento e destes, 77,31% estavam realmente na 1ª série. Caso a

resposta à pergunta anterior seja “não”, segue-se para o próximo nó, que pergunta se total_pe é maior ou igual a 7,5, e assim por diante, até que o aluno em teste tenha sido classificado.

A segunda aplicação do algoritmo foi feita deixando-se de fora a escola. Assim, o atributo de classificação continua sendo a série, mas os parâmetros foram apenas o sexo, o número de respostas e o número de acertos em cada categoria de testes. A matriz de confusão desta aplicação é representada pela Tabela 5.

Razão de erro		*0,5187*						
Predição de valores		Matriz de confusão						
Valor	*Razão de acerto*		*1a*	*2a*	*3a*	*4a*	*8a*	**Soma**
1a	0,7488	1a	2856	679	169	103	7	3814
2a	0,2759	2a	1138	588	201	199	5	2131
3a	0,2135	3a	322	280	225	227	0	1054
4a	0,3685	4a	179	180	195	325	3	882
8a	0,0024	8a	330	76	10	2	1	419
		Soma	4825	1803	800	856	16	8300

Tabela 5: Matriz de confusão da segunda aplicação do algoritmo C4.5. Fonte: O autor.

Observando-se a tabela acima, nota-se pela razão de acerto que o algoritmo apenas consegue classificar alunos da 1ª série. Conclui-se, então, que o algoritmo não é capaz de classificar os alunos segundo este conjunto de parâmetros e atributo de classificação. Por isso e pelo fato desta árvore ser apreciavelmente grande, optou-se por não reproduzi-la no texto desta dissertação. Entretanto, isso não significa que esta árvore de decisão seja inútil. Nela, o parâmetro sexo aparece pouco, e quando o faz, aparece apenas em três folhas da árvore, indicando que é um parâmetro de pouca importância. Como o parâmetro sexo também não aparece na primeira árvore, há indícios fortes de que o sexo da criança não influencia seu desempenho no TCLP.

Seguindo esta abordagem, o algoritmo foi aplicado uma terceira vez, agora retirando-se o sexo do conjunto de atributos. A matriz de confusão do algoritmo nestas condições é representado pela Tabela 6. Novamente nota-se que o algoritmo apenas consegue classificar alunos da 1ª série. Assim, esta árvore de decisão também não foi reproduzida neste texto, dada sua baixa utilidade.

Razão de erro		*0,5161*						
Predição de valores		Matriz de confusão						
Valor	*Razão de acerto*		*1a*	*2a*	*3a*	*4ª*	*8a*	**Soma**
1a	0,7583	1a	2892	656	160	99	7	3814
2a	0,2717	2a	1148	579	191	206	7	2131
3a	0,1983	3a	322	274	209	249	0	1054
4a	0,3810	4a	178	181	185	336	2	882
8a	0,0000	8a	331	76	10	2	0	419
		Soma	4871	1766	755	892	16	8300

Tabela 6: Matriz de confusão da terceira aplicação do algoritmo C4.5. Fonte: O autor.

4.2.2 Tarefa de Agrupamento

Os algoritmos de agrupamento são aplicados a um conjunto de dados e o resultado esperado é que o algoritmo consiga separar os dados em grupos de forma que elementos pertencentes a um mesmo grupo sejam o mais semelhantes possíveis entre si e que elementos de grupos diferentes sejam o mais diferentes possível entre si (CHEN; HAN; YU, 1996). Para esta tarefa foram escolhidos dois algoritmos, K-Means (JIANG; TSENG; SU, 2001) e Mapas Auto-Organizados de Kohonen (Kohonen self-organizing maps) (KOHONEN, 1998). Ambos algoritmos clássicos freqüentemente utilizados como parâmetro de comparação para novos algoritmos. A tarefa de agrupamento é feita no estilo *bottom up*, pois os algoritmos não são restringidos quanto aos agrupamentos a encontrar.

4.2.2.1 Agrupamento Utilizando K-Means

O algoritmo K-Means não é capaz de escolher o número de agrupamentos a utilizar para separar os dados. Esta escolha fica a cargo do usuário. Escolher o número de agrupamentos, então, costuma ser uma atividade interativa. Inicia-se a aplicação do algoritmo com um número pequeno de agrupamentos, tipicamente 3, então observa-se os resultados obtidos. Habitualmente espera-se que o algoritmo consiga separar os dados entre os agrupamentos de forma que todos contenham uma quantidade não muito pequena de dados. Um agrupamento com poucos dados pode sugerir que tal agrupamento é supérfluo, que não apresenta dados com importância estatística. Entretanto, quando surgem agrupamentos com poucos membros, eles não devem ser descartados imediatamente. Tais agrupamentos podem identificar dados com comportamento atípico e raro. No caso do TCLP, há a possibilidade de que um algoritmo de agrupamento identifique alunos que apresentam dificuldades de aprendizado acima do normal. Feita a análise dos resultados do algoritmo com um certo número de agrupamentos, deve-se repetir o processo para outros números de agrupamentos.

A aplicação do algoritmo K-Means foi feita ao conjunto de dados do TCLP utilizando-se como parâmetros de comparação os totais de acertos por tipo de questão e o número de questões respondidas ao todo. Os melhores resultados foram observados quando o algoritmo separou os dados em 4 agrupamentos. Os agrupamentos, seus nomes (descrição) e a quantidade de membros que cada um contém estão representados na Tabela 7.

Agrupamento	*Descrição*	*Tamanho*
agrupamento n°1	c_kmeans_1	57
agrupamento n°2	c_kmeans_2	869
agrupamento n°3	c_kmeans_3	435
agrupamento n°4	c_kmeans_4	308

Tabela 7: Tamanho dos agrupamentos encontrados pelo algoritmo K-Means utilizando 4 agrupamentos. Fonte: O autor.

A premissa de um algoritmo de agrupamento é reunir dados semelhantes em um mesmo grupo, mas o algoritmo não expõe como esses dados são semelhantes entre si. Essa é uma tarefa do usuário na fase de pós-processamento. Uma maneira de se obter esta informação é visualizar os dados graficamente.

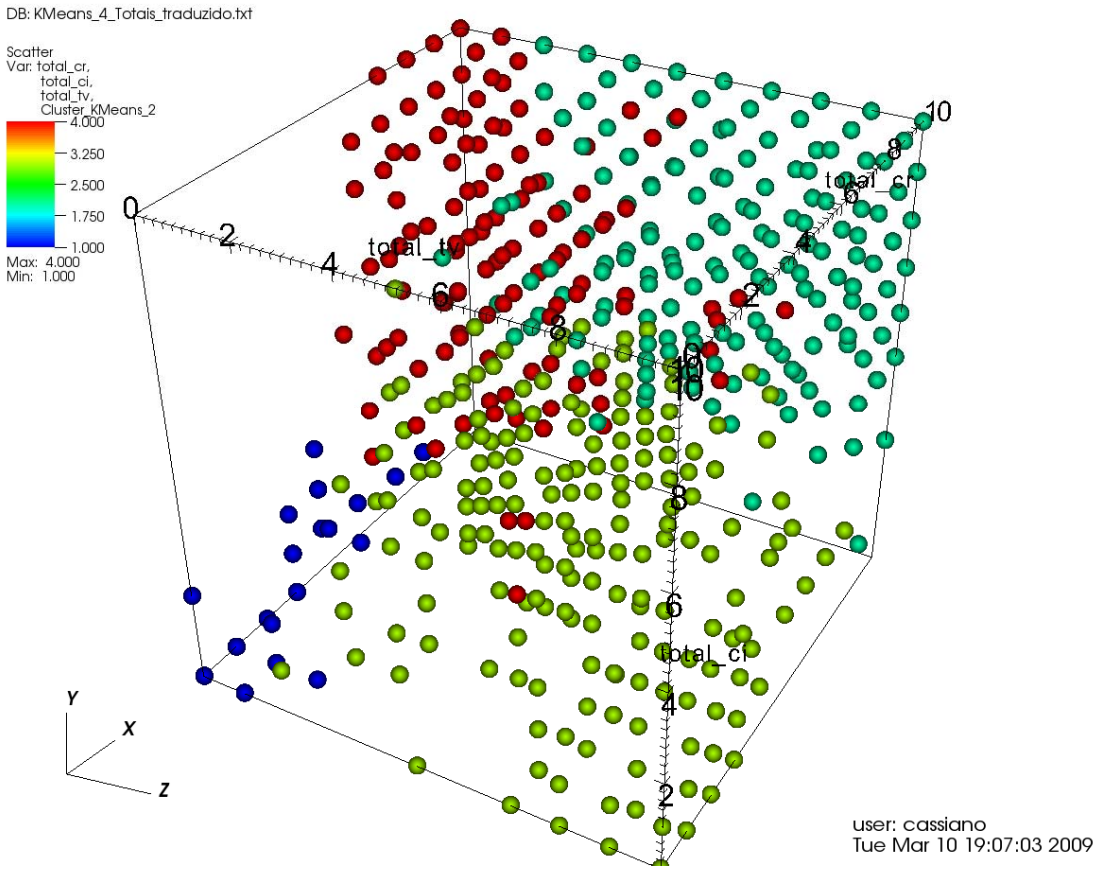

Figura 6: Agrupamentos obtidos por K-Means em um gráfico de dispersão: X – total_cr, Y – total_ci, Z – total_tv e cor – agrupamentos. Agrupamentos assumem as cores: Azul – c_kmeans_1, verde – c_kmeans_2, amarelo – c_kmeans_3 e vermelho – c_kmeans_4. Fonte: O autor.

Na Figura 6 pode-se observar como os agrupamentos, representados pelas diferentes cores, se distribuem de acordo com três parâmetros de agrupamento. Pode-se observar que o agrupamento c_kmeans_1 (azul) concentra elementos com baixa pontuação em questões do tipo cr, ci e tv. O agrupamento c_kmeans_2 (verde) apresenta pontuação elevada nos três parâmetros. O agrupamento c_kmeans_3 (amarelo) apresenta pontuação baixa em cr e ci e pontuação alta em tv. Por fim o agrupamento c_kmeans_4 (vermelho) apresenta pontuação elevada em cr e ci mas reduzida em tv. Os gráficos tridimensionais apresentados neste capítulo estão reproduzidos no Apêndice 2 em versão estereográfica. São gráficos que podem ser vistos com o auxílio de óculos tridimensionais.

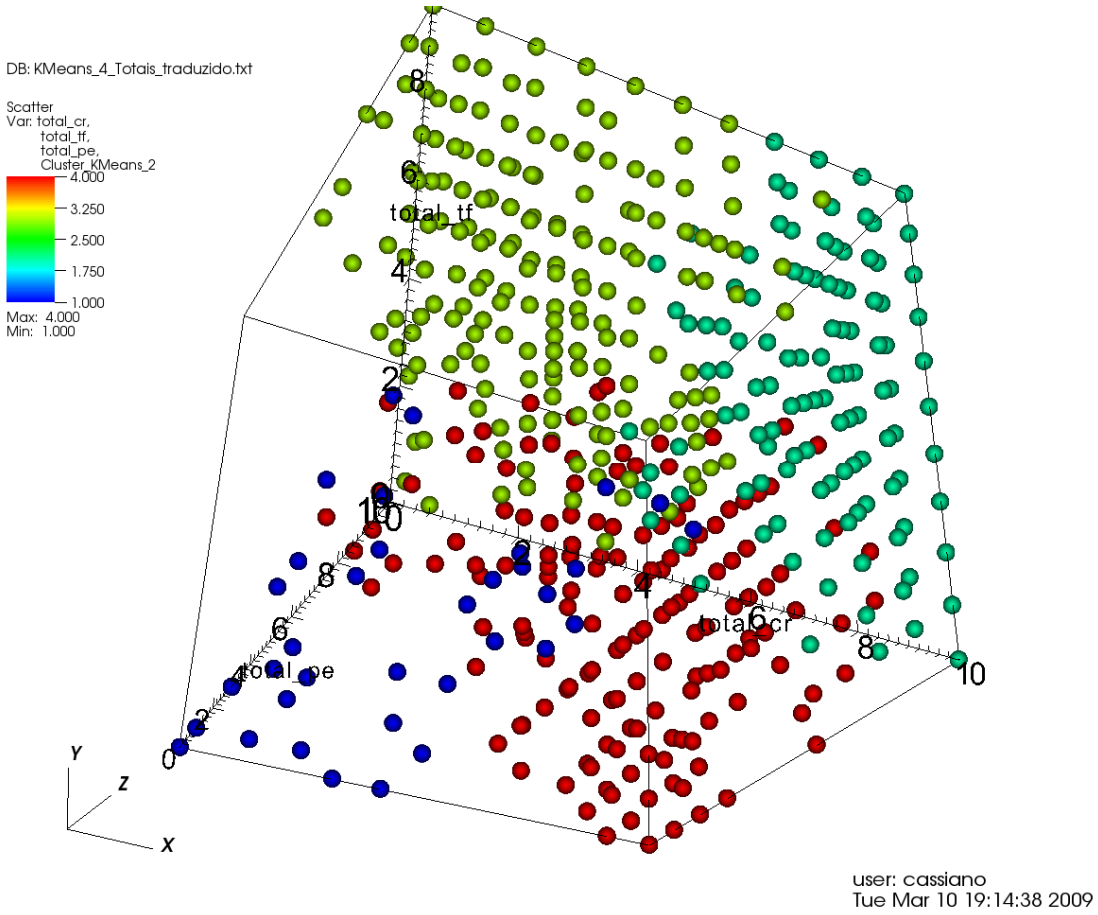

Figura 7: Agrupamentos obtidos por K-Means em um gráfico de dispersão: X – total_cr, Y – total_tf, Z – total_pe e cor – agrupamentos. Agrupamentos assumem as cores: Azul – c_kmeans_1, verde – c_kmeans_2, amarelo – c_kmeans_3 e vermelho – c_kmeans_4. Fonte: O autor.

Prosseguindo com a interpretação dos agrupamentos utilizando-se agora a Figura 7, nota-se que o agrupamento c_kmeans_1 (azul) apresenta desempenho ruim também em tf e pe. O agrupamento c_kmeans_2 (verde) apresenta bom desempenho também em tf e pe. O agrupamento c_kmeans_3 (amarelo) apresenta desempenho bom em tf e pe e o agrupamento c_kmeans_4 (vermelho) apresenta desempenho ruim em tf e pe.

Procedendo com a análise desta mesma forma para os demais parâmetros, podemos caracterizar os agrupamentos segundo os parâmetros como mostrado na Tabela 8.

Agrupamento	*cr*	*ci*	*ts*	*tv*	*tf*	*ph*	*pe*	*Total de respostas*
c_kmeans_1	↓	↓	–	↓	↓	↓	↓	↓
c_kmeans_2	↑	↑	↑	↑	–	–	↑	↑
c_kmeans_3	↓	↓	↑	↑	↑	↑	↑	↑
c_kmeans_4	↑	↑	↓	↓	↓	↓	↓	↑

Tabela 8: Comportamento dos elementos de cada agrupamento de acordo com os parâmetros. Legenda: ↑ - valor elevado, ↓ - valor reduzido, – - valor médio ou sem tendência. Fonte: O autor.

Na Tabela 8 pode-se observar que o agrupamento c_kmeans_1 apresenta um desempenho baixo em todos os parâmetros utilizados, agrupando os alunos com os piores resultados. O agrupamento c_kmeans_2 reúne os alunos que obtiveram desempenho elevado em todos os parâmetros. O agrupamento c_kmeans_3 reúne alunos que obtiveram desempenho ruim em cr e ci, mas bom desempenho em todos os outros testes. Lembrando-se que a tarefa do aluno é circular questões de tipo cr e ci e marcar com um "X" as demais questões, percebe-se que o algoritmo reuniu neste agrupamento os alunos que tenderam a marcar com um "X" todas as questões do teste. Por fim, o agrupamento c_kmeans_4 reúne alunos com desempenho bom em cr, ci e total de questões respondidas, mas baixo desempenho nos demais tipos de teste. Isso indica que este agrupamento reúne alunos que tenderam a circular todas as questões do teste.

4.2.2.1.1 Análise em Estilo *Top Down*

Uma outra forma de se obter conhecimento dos agrupamentos obtidos na seção anterior é executar uma tarefa de MD em estilo *top down*. Neste caso, assume-se a hipótese de que os agrupamentos obtidos são representativos de alguma característica importante dos dados. Esta hipótese pode ser testada com um algoritmo de classificação. Neste estudo, foi empregado novamente o algoritmo C4.5, obtendo-se a matriz de confusão mostrada na Tabela 9. Vale lembrar que a avaliação de desempenho do algoritmo continua sendo feita com validação cruzada de 10 vias e 5 repetições.

Razão de erro		*0,0816*				
Predição de valores		Matriz de confusão				
Valor	*Razão de acerto*		*c_kmeans_1*	*c_kmeans_2*	*c_kmeans_3*	*c_kmeans_4*	*Soma*
c_kmeans_1	0,9315	c_kmeans_1	1958	49	60	35	2102
c_kmeans_2	0,9345	c_kmeans_2	53	2254	64	41	2412
c_kmeans_3	0,8973	c_kmeans_3	67	87	1878	61	2093
c_kmeans_4	0,9055	c_kmeans_4	50	59	51	1533	1693
		Soma	2128	2449	2053	1670	8300

Tabela 9: Matriz de confusão da aplicação do algoritmo C4.5 na classificação de agrupamentos do algoritmo K-Means. Fonte: O autor.

Na Tabela 9 pode-se notar que as razões de acerto são bastante elevadas, indicando que o algoritmo C4.5 consegue classificar muito bem os alunos nos quatro agrupamentos construídos pelo algoritmo K-Means. Razões de acerto assim tão elevadas seriam motivo de destaque em uma situação comum, entretanto, os agrupamentos utilizados como atributos de classificação foram definidos segundo um critério matemático e previsível por um algoritmo. É natural que a técnica de classificação consiga recuperar alguns aspectos deste critério, o que acaba se refletindo em razões de

acerto elevadas. Em outras palavras, já era esperado que o algoritmo C4.5 obtivesse um desempenho elevado nesta tarefa.

A árvore de decisão obtida é reproduzida no Quadro 3. Observando-se esta árvore, nota-se pela linha 1.1 que alunos que responderam menos que 42 questões pertencem ao agrupamento c_kmeans_1. Pela análise da seção anterior concluiu-se que este agrupamento reúne alunos que obtiveram baixo desempenho no teste. A árvore de decisão, então, ressalta um motivo para tal: Estes alunos desistiram de responder o teste por completo. O agrupamento c_kmeans_2 reúne alunos que tiveram bom desempenho. A árvore de decisão classifica a maioria dos alunos desse agrupamento com a regra 2.2.2.2, onde diz que alunos que responderam mais que 47 questões (regra 2), que acertaram mais que 6 questões tipo cr (regra 2.2), que acertaram mais que 7 questões do tipo pe (regra 2.2.2) e que acertaram mais que 5 questões do tipo ci (regra 2.2.2.2) pertencem ao agrupamento c_kmeans_2. O agrupamento c_kmeans_3 reúne alunos que marcaram a maioria das questões como incorretas indistintamente. A maioria dos membros desse grupo é classificada pela regra 2.1.2.2.2, que diz que um aluno que responda mais que 47 questões (regra 2), que acerte menos que 7 questões do tipo cr (regra 2.1), que acerte mais que 6 questões do tipo pe (regra 2.1.2), que acerte mais que 4 questões do tipo tf (regra 2.1.2.2) e que acerte mais que 4 questões do tipo ph (2.1.2.2.2) apresentou a tendência de marcar como erradas todas as questões. De maneira semelhante, o agrupamento c_kmeans_4 que reúne alunos que tenderam a marcar como corretas todas as questões, é classificado principalmente pela regra 2.2.1.1. Esta regra especifica que alunos que responderam mais que 47 questões (regra 2), que acertaram mais que 6 questões tipo cr (regra 2.2), que acertaram menos que 8 questões tipo pe (regra 2.2.1) e que acertaram menos que 5 questões do tipo tf (regra 2.2.1.1) tenderam a marcar como corretas todas as questões. Isso sugere que é possível identificar se o aluno tende a marcar todas as questões do teste como corretas ou incorretas de maneira indiscriminada observando-se seu desempenho em apenas alguns tipos de questão.

1. total_respondidas < 48,0000
   - 1.1. total_respondidas < 41,5000 então Cluster_KMeans_2 = **c_kmeans_1** (100,00 % de 54 exemplos)
   - 1.2. total_respondidas >= 41,5000 então Cluster_KMeans_2 = **c_kmeans_3** (70,00 % de 10 exemplos)
2. total_respondidas >= 48,0000
   - 2.1. total_cr < 6,5000
     - 2.1.1. total_pe < 6,5000
       - 2.1.1.1. total_ph < 5,5000
         - 2.1.1.1.1. total_tf < 4,5000 então Cluster_KMeans_2 = **c_kmeans_4** (100,00 % de 40 exemplos)
         - 2.1.1.1.2. total_tf >= 4,5000
           - 2.1.1.1.2.1. total_ph < 4,5000 então Cluster_KMeans_2 = **c_kmeans_4** (72,00 % de 25 exemplos)
           - 2.1.1.1.2.2. total_ph >= 4,5000 então Cluster_KMeans_2 = **c_kmeans_3** (70,00 % de 20 exemplos)
       - 2.1.1.2. total_ph >= 5,5000 então Cluster_KMeans_2 = **c_kmeans_3** (85,11 % de 47 exemplos)
     - 2.1.2. total_pe >= 6,5000
       - 2.1.2.1. total_tf < 4,5000
         - 2.1.2.1.1. total_ph < 5,5000 então Cluster_KMeans_2 = **c_kmeans_2** (50,00 % de 20 exemplos)
         - 2.1.2.1.2. total_ph >= 5,5000 então Cluster_KMeans_2 = **c_kmeans_3** (81,25 % de 16 exemplos)
       - 2.1.2.2. total_tf >= 4,5000
         - 2.1.2.2.1. total_ph < 4,5000
           - 2.1.2.2.1.1. total_cr < 4,5000 então Cluster_KMeans_2 = **c_kmeans_3** (100,00 % de 13 exemplos)
           - 2.1.2.2.1.2. total_cr >= 4,5000 então Cluster_KMeans_2 = **c_kmeans_2** (58,82 % de 17 exemplos)
         - 2.1.2.2.2. total_ph >= 4,5000 então Cluster_KMeans_2 = **c_kmeans_3** (98,38 % de 308 exemplos)

2.2. total_cr >= 6,5000
  2.2.1. total_pe < 7,5000
    2.2.1.1. total_tf < 4,5000 então Cluster_KMeans_2 = **c_kmeans_4** (97,01 % de 201 exemplos)
    2.2.1.2. total_tf >= 4,5000
      2.2.1.2.1. total_ts < 7,5000 então Cluster_KMeans_2 = **c_kmeans_4** (80,77 % de 26 exemplos)
      2.2.1.2.2. total_ts >= 7,5000 então Cluster_KMeans_2 = **c_kmeans_2** (84,62 % de 13 exemplos)
  2.2.2. total_pe >= 7,5000
    2.2.2.1. total_ci < 5,5000
      2.2.2.1.1. total_ci < 3,5000 então Cluster_KMeans_2 = **c_kmeans_3** (68,75 % de 16 exemplos)
      2.2.2.1.2. total_ci >= 3,5000
        2.2.2.1.2.1. total_ph < 5,5000 então Cluster_KMeans_2 = **c_kmeans_2** (97,62 % de 42 exemplos)
        2.2.2.1.2.2. total_ph >= 5,5000
          2.2.2.1.2.2.1. total_tv < 6,5000 então Cluster_KMeans_2 = **c_kmeans_3** (61,54 % de 13 exemplos)
          2.2.2.1.2.2.2. total_tv >= 6,5000 então Cluster_KMeans_2 = **c_kmeans_2** (100,00 % de 11 exemplos)
    2.2.2.2. total_ci >= 5,5000 então Cluster_KMeans_2 = **c_kmeans_2** (98,58 % de 777 exemplos)

Quadro 3: Árvore de decisão que classifica os dados nos agrupamentos obtidos por K-Means. Fonte: O autor.

4.2.2.2 Agrupamento Utilizando Mapas Auto-Organizados de Kohonen

Assim como ocorre com o algoritmo K-Means, o algoritmo de Mapas Auto-Organizados de Kohonen (Kohonen-SOM, sigla em inglês) não escolhe a quantidade de agrupamentos, esta escolha fica a cargo do usuário. Neste algoritmo, os agrupamentos são organizados em uma matriz m x n. Inicialmente escolhe-se um número pequeno para m e n, tipicamente 2. Avalia-se o resultado obtido e, então, decide-se se é necessário modificar os valores de m e n. Na análise realizada neste estudo, encontrou-se resultados satisfatórios com a classificação em uma matriz de agrupamentos de dimensão 2 x 2. Assim como na análise feita com a classificação pelo algoritmo K-Means, é desejável que a quantidade de membros em cada agrupamento obtido por SOM seja alta. Novamente, agrupamentos com poucos membros podem ser encontrados. Caso isto ocorra, tais agrupamentos não devem ser descartados imediatamente, pelos mesmos motivos que agrupamentos com poucos membros obtidos por K-Means não devem ser descartados.

O algoritmo foi aplicado utilizando-se como parâmetros de agrupamento os totais de acertos em cada categoria de questão e o número de questões respondidas. Os agrupamentos encontrados estão sumarizados na Tabela 10. Nela é mostrado o número de elementos encontrados para cada agrupamento. A nomenclatura dos agrupamentos segue o padrão c_som_l_c, onde l é a linha e c é a coluna da matriz de agrupamentos. Assim, o número de elementos dentro do agrupamento c_som_1_2 é 302.

	1	*2*
1	471	302
2	493	403

Tabela 10: Topografia dos agrupamentos obtidos com o algoritmo de agrupamento por Mapas Auto-Organizados de Kohonen. Fonte: O autor.

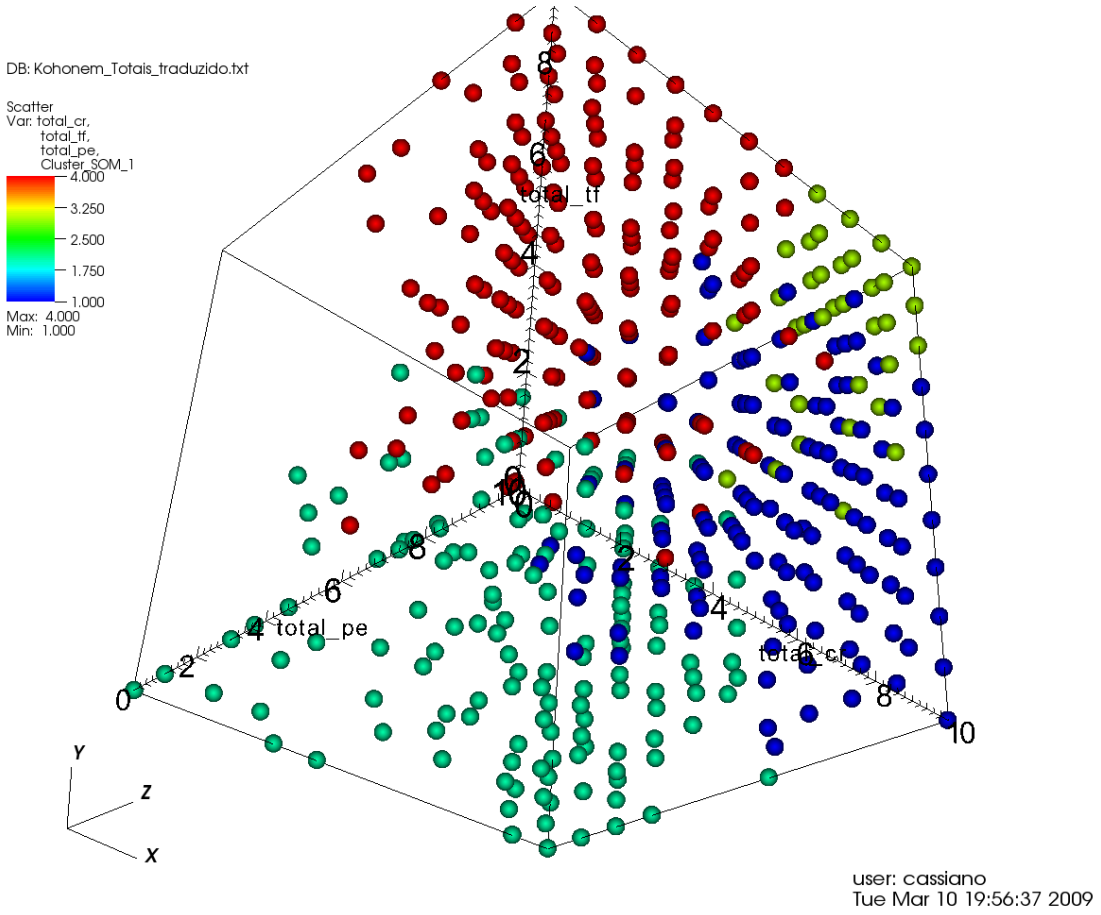

Figura 8: Agrupamentos obtidos por Mapas Auto-Organizados de Kohonen em um gráfico de dispersão: X – total_cr, Y – total_tf, Z – total_pe e cor – agrupamentos. Agrupamentos assumem as cores: Azul – c_som_1_1, verde – c_som_1_2, amarelo – c_som_2_1 e vermelho – c_som_2_2. Fonte: O autor.

Pela observação da Figura 8 pode-se notar que o agrupamento c_som_1_1 (azul) reúne indivíduos que obtiveram bom resultado em cr e pe, mas baixo desempenho em tf. O agrupamento c_som_1_2 (verde) obteve baixo desempenho em tf e pe, enquanto que seu desempenho em cr não apresenta tendências aparentes. O agrupamento c_som_2_1 (amarelo) apresenta bom desempenho em cr, tf e pe. O agrupamento c_som_2_2 (vermelho) apresenta bom desempenho em pe e tf, mas apresenta desempenho ruim em cr.

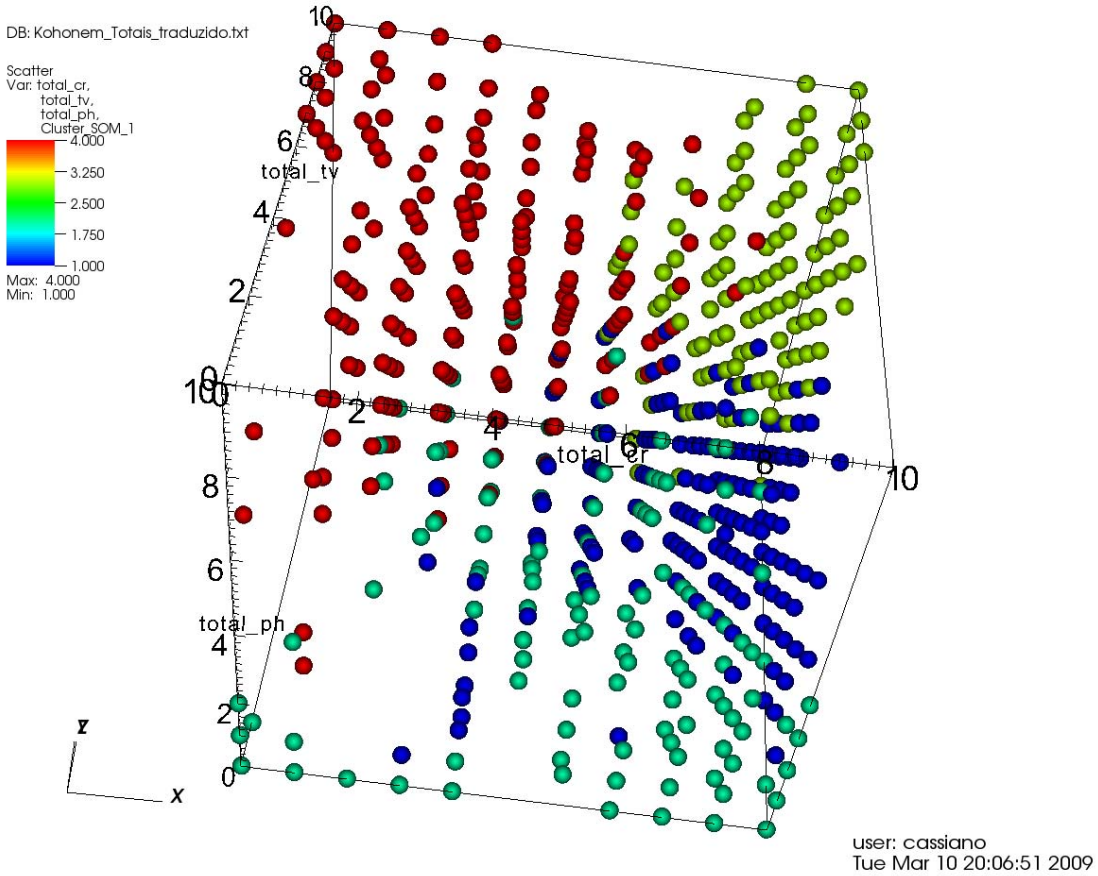

Figura 9: Agrupamentos obtidos por Mapas Auto-Organizados de Kohonen em um gráfico de dispersão: X – total_cr, Y – total_tv, Z – total_ph e cor – agrupamentos. Agrupamentos assumem as cores: Azul – c_som_1_1, verde – c_som_1_2, amarelo – c_som_2_1 e vermelho – c_som_2_2. Fonte: O autor.

A interpretação da Figura 9 indica que o agrupamento c_som_1_1 (azul) reúne indivíduos com baixo desempenho em tv e ph. O agrupamento c_som_1_2 (verde) reúne alunos que também apresentam desempenho baixo em tv e ph, mas diferentemente do agrupamento c_som_1_1, apresentam desempenho homogêneo em cr. O agrupamento c_som_2_1 (amarelo) reúne alunos que obtiveram bons resultados em tv e ph. Finalmente, o agrupamento c_som_2_2 (vermelho) reúne os alunos que obtiveram bons resultados em tv e ph.

Agrupamento	*cr*	*ci*	*ts*	*tv*	*tf*	*ph*	*pe*	*Total de respostas*
c_som_1_1	↑	↑	↑	↓	↓	↓	↑	↑
c_som_1_2	–	–	↓	↓	↓	↓	↓	↑
c_som_2_1	↑	↑	↑	↑	↑	↑	↑	↑
c_som_2_2	↓	↓	↑	↑	↑	↑	↑	↑

Tabela 11: Comportamento dos elementos de cada agrupamento de acordo com os parâmetros. Legenda: ↑ - valor elevado, ↓ - valor reduzido, – - valor médio ou sem tendência. Fonte: O autor.

Realizando-se a avaliação visual dos dados como mostrado nos parágrafos acima, pode-se registrar as tendências de comportamento dos agrupamentos como foi feito na Tabela 11. Assim, nota-se que o agrupamento c_som_1_1 apresenta desempenho bom em cr, ci, ts e pe, mas ruim nos demais testes. Esta combinação de resultados indica que este agrupamento reúne alunos que ainda não realizaram a transição entre os estágios alfabético e ortográfico de leitura (CAPOVILLA; VARANDA; CAPOVILLA, 2006). O agrupamento c_som_1_2 reúne alunos que tenderam a marcar como corretas todas as questões do teste, como ocorreu com o agrupamento c_kmeans_4. O agrupamento c_som_2_1 reúne os alunos que apresentaram melhor desempenho, indicando que já fizeram a transição para o estágio ortográfico de leitura. Por fim, o agrupamento c_som_2_2 reúne os alunos que tenderam a marcar como incorretas todas as questões do teste, como ocorreu com o agrupamento c_kmeans_3.

4.2.2.2.1 Análise em Estilo *Top Down*

Assim como foi feito na seção 4.2.2.1.1, foi feita a aplicação do algoritmo de classificação C4.5 aos agrupamentos obtidos com o algoritmo Kohonen-SOM.

Razão de erro	*0,0837*
Predição de valores	Matriz de confusão

Valor	*Razão de acerto*
c_som_1_1	0,9180
c_som_1_2	0,9247
c_som_2_1	0,8964
c_som_2_2	0,9228

	c_som_1_1	*c_som_1_2*	*c_som_2_1*	*c_som_2_2*	*Soma*
c_som_1_1	1747	45	58	53	1903
c_som_1_2	52	2209	60	68	2389
c_som_2_1	50	59	1678	85	1872
c_som_2_2	51	61	53	1971	2136
Soma	1900	2374	1849	2177	8300

Tabela 12: Matriz de confusão da aplicação do algoritmo C4.5 na classificação de agrupamentos do algoritmo de Mapas Auto-Organizados de Kohonen. Fonte: O autor.

A matriz de confusão, obtida por validação cruzada de 10 vias e 5 repetições está reproduzida na Tabela 12, nota-se que a razão de acerto do classificador é alta. Neste parágrafo vale o mesmo comentário feito na seção 4.2.2.1.1, ou seja, como os agrupamentos obtidos pelo algoritmo Kohonen-SOM seguem um critério matemático bem definido, é natural que se espere que o algoritmo C4.5 consiga reproduzir um pouco dessa lógica em sua árvore de decisão, a qual é reproduzida no Quadro 4.

1. total_pe < 6,5000
    1.1. total_ph < 4,5000
        1.1.1. total_ts < 6,5000 então Cluster_SOM_1 = **c_som_1_2** (96,62 % de 237 exemplos)
        1.1.2. total_ts >= 6,5000
            1.1.2.1. total_tf < 2,5000 então Cluster_SOM_1 = **c_som_1_2** (73,68 % de 19 exemplos)
            1.1.2.2. total_tf >= 2,5000 então Cluster_SOM_1 = **c_som_1_1** (62,50 % de 16 exemplos)
    1.2. total_ph >= 4,5000
        1.2.1. total_tf < 5,5000
            1.2.1.1. total_ts < 6,5000
                1.2.1.1.1. total_tv < 5,5000 então Cluster_SOM_1 = **c_som_1_2** (96,88 % de 32 exemplos)
                1.2.1.1.2. total_tv >= 5,5000 então Cluster_SOM_1 = **c_som_2_2** (45,45 % de 11 exemplos)
            1.2.1.2. total_ts >= 6,5000 então Cluster_SOM_1 = **c_som_2_2** (60,00 % de 10 exemplos)
        1.2.2. total_tf >= 5,5000 então Cluster_SOM_1 = **c_som_2_2** (85,11 % de 47 exemplos)

2. total_pe >= 6,5000
  2.1. total_cr < 6,5000
    2.1.1. total_respondidas < 38,0000 então Cluster_SOM_1 = **c_som_1_1** (78,57 % de 14 exemplos)
    2.1.2. total_respondidas >= 38,0000
      2.1.2.1. total_cr < 4,5000 então Cluster_SOM_1 = **c_som_2_2** (97,59 % de 291 exemplos)
      2.1.2.2. total_cr >= 4,5000
        2.1.2.2.1. total_ci < 4,5000
          2.1.2.2.1.1. total_ph < 5,5000 então Cluster_SOM_1 = **c_som_1_1** (61,54 % de 13 exemplos)
          2.1.2.2.1.2. total_ph >= 5,5000 então Cluster_SOM_1 = **c_som_2_2** (93,75 % de 32 exemplos)
        2.1.2.2.2. total_ci >= 4,5000 então Cluster_SOM_1 = **c_som_1_1** (66,04 % de 53 exemplos)
  2.2. total_cr >= 6,5000
    2.2.1. total_tf < 6,5000
      2.2.1.1. total_ci < 4,5000 então Cluster_SOM_1 = **c_som_1_1** (50,00 % de 16 exemplos)
      2.2.1.2. total_ci >= 4,5000
        2.2.1.2.1. total_pe < 7,5000 então Cluster_SOM_1 = **c_som_1_1** (74,07 % de 27 exemplos)
        2.2.1.2.2. total_pe >= 7,5000
          2.2.1.2.2.1. total_ph < 2,5000 então Cluster_SOM_1 = **c_som_1_1** (99,49 % de 195 exemplos)
          2.2.1.2.2.2. total_ph >= 2,5000
            2.2.1.2.2.2.1. total_tv < 8,5000
              2.2.1.2.2.2.1.1. total_ph < 5,5000 então Cluster_SOM_1 = **c_som_1_1** (97,14 % de 105 exemplos)
              2.2.1.2.2.2.1.2. total_ph >= 5,5000
                2.2.1.2.2.2.1.2.1. total_tf < 4,5000 então Cluster_SOM_1 = **c_som_1_1** (90,91 % de 11 exemplos)
                2.2.1.2.2.2.1.2.2. total_tf >= 4,5000 então Cluster_SOM_1 = **c_som_2_1** (71,43 % de 14 exemplos)
            2.2.1.2.2.2.2. total_tv >= 8,5000
              2.2.1.2.2.2.2.1. total_tf < 5,5000
                2.2.1.2.2.2.2.1.1. total_cr < 9,5000 então Cluster_SOM_1 = **c_som_1_1** (63,64 % de 11 exemplos)
                2.2.1.2.2.2.2.1.2. total_cr >= 9,5000 então Cluster_SOM_1 = **c_som_2_1** (70,00 % de 10 exemplos)
              2.2.1.2.2.2.2.2. total_tf >= 5,5000 então Cluster_SOM_1 = **c_som_2_1** (100,00 % de 20 exemplos)
    2.2.2. total_tf >= 6,5000
      2.2.2.1. total_ci < 3,5000 então Cluster_SOM_1 = **c_som_2_2** (58,33 % de 12 exemplos)
      2.2.2.2. total_ci >= 3,5000
        2.2.2.2.1. total_tv < 8,5000
          2.2.2.2.1.1. total_ph < 3,5000
            2.2.2.2.1.1.1. total_tf < 8,5000
              2.2.2.2.1.1.1.1. total_tv < 7,5000 então Cluster_SOM_1 = **c_som_1_1** (95,45 % de 22 exemplos)
              2.2.2.2.1.1.1.2. total_tv >= 7,5000 então Cluster_SOM_1 = **c_som_2_1** (53,85 % de 13 exemplos)
            2.2.2.2.1.1.2. total_tf >= 8,5000 então Cluster_SOM_1 = **c_som_2_1** (73,33 % de 15 exemplos)
          2.2.2.2.1.2. total_ph >= 3,5000 então Cluster_SOM_1 = **c_som_2_1** (92,31 % de 65 exemplos)
        2.2.2.2.2. total_tv >= 8,5000 então Cluster_SOM_1 = **c_som_2_1** (99,16 % de 358 exemplos)

Quadro 4: Árvore de decisão que classifica os dados nos agrupamentos obtidos por Kohonen-SOM. Fonte: O autor.

Observando-se esta árvore de decisão, nota-se que a maioria dos membros do agrupamento c_som_1_1 é classificado pelas regras 2.2.1.2.2.1 e 2.2.1.2.2.2.1.1. Este é o agrupamento dos alunos ainda em estágio alfabético de leitura. Os membros do agrupamento c_som_1_2 podem ser classificados principalmente pela regra 1.1.1. Este é o agrupamento dos alunos que tenderam a marcar todas as questões como corretas. A regra 2.2.2.2.2 classifica a maior parte dos membros do agrupamento c_som_2_1, que é o agrupamento que reúne os alunos já em estágio ortográfico de leitura. Por fim, a regra 2.1.2.1 classifica a maior parte dos membros do agrupamento c_som_2_2, que reúne os alunos que tenderam a marcar como erradas todas as questões do teste. Fica a cargo do leitor verificar quais são os atributos que o algoritmo C4.5 destacou como importantes para a classificação de cada uma dessas regras. Esta verificação é feita de maneira análoga à feita na seção 4.2.2.1.1.

4.2.3 Tarefa de Obtenção de Regras de Associação

Os dados dos resultados do TCLP são disponíveis em formato transacional, ou seja, estão disponíveis os resultados de alunos específicos. Cada aluno, representado por uma linha do arquivo de dados, é chamado de **transação**. Uma transação é composta de respostas a cada questão do teste, chamadas de **itens.** Quando um aluno responde corretamente à uma questão, diz-se que o item que representa a questão está presente na transação que representa o aluno. A tarefa de obtenção de Regras de Associação tem como finalidade obter relações entre os itens das transações. O significado de uma Regra de Associação pode ser entendido como: "Ocorrendo os itens I_1, I_2 e I_3, ocorre na mesma transação o iten I_4, com suporte s, confiança c e *lift* l". Vale revisar estes conceitos nesta seção (AGRAWAL; IMIELINSKI; SWAMI, 1993; AGRAWAL; SRIKANT, 1994):

- **Suporte**: Dada uma regra da forma "$I_1 \wedge I_2 \wedge I_3 \rightarrow I_4$", suporte indica o percentual de transações no conjunto de dados que apresentam os itens I_1, I_2, I_3 *e* I_4 simultaneamente. Representa a abrangência da regra;
- **Confiança**: Dada uma regra da forma "$I_1 \wedge I_2 \wedge I_3 \rightarrow I_4$", a confiança da regra indica o percentual de transações em que a regra é verdadeira dentro do conjunto de transações onde ocorre o antecedente (lado esquerdo da regra, "$I_1 \wedge I_2 \wedge I_3$"). Representa a força da regra;
- ***Lift***: É expresso pela equação:

$$\frac{\boldsymbol{Suporte(regra)}}{\boldsymbol{Suporte(antecedente)} * \boldsymbol{Suporte(Conseqüente)}}$$

  Indica a importância da regra.

O algoritmo A Priori foi escolhido para esta tarefa. É um algoritmo clássico e freqüentemente utilizado como parâmetro de comparação para outros algoritmos para a mesma tarefa. Ele é configurado especificando-se valores mínimos para suporte e confiança. O significado dessas grandezas fica claro nos parágrafos anteriores. Já o significado de *lift* merece um comentário mais extenso:

Imagine que em uma loja de conveniência observou-se que clientes que compram suco de laranja também compram leite com uma confiança de 75% e que a combinação de suco de laranja e leite apresenta um suporte de 30%. À primeira vista esta parece ser uma regra excelente, e na maioria dos casos realmente seria, pois ela apresenta valores elevados de confiança e suporte. Entretanto, o que aconteceria se os clientes dessa loja comprassem leite em 90% das transações? Neste caso, clientes que compram suco de laranja, na verdade, estão **menos** propensos a comprar leite do que clientes em geral. Para resolver este problema utiliza-se *lift*. Assim, assumindo-se que 40% dos clientes da loja compram suco de laranja, o valor de *lift* pode ser calculado:

$$\frac{30\%}{40\% * 90\%} = 0{,}83$$

Observa-se que o valor de *lift* aferido é menor do que 1, o que indica que a regra não oferece ganho de conhecimento (ORACLE..., 2008).

A aplicação do algoritmo A Priori foi feita utilizando-se os alunos como transações, e seus acertos à questões individuais como itens. O algoritmo foi limitado a gerar regras envolvendo até 4 itens, com suporte mínimo de 33% e confiança mínima de 75%. O resultado foi um conjunto com 1.152.761 regras de associação. As regras foram classificadas em ordem decrescente de *lift*, confiança e suporte, respectivamente, e as 20 primeiras regras estão registradas na Tabela 13

Número da Regra	*Antecedente*	*Conseqüente*	*Suporte*	*Confiança*	*Lift*
101434	questao_47_acerto_tf=Acerto/\questao_55_acerto_tv=Acerto/\ questao_20_acerto_pe=Acerto	questao_57_acerto _tf=Acerto	33.4	85.7	176.4
7029	questao_47_acerto_tf=Acerto/\questao_65_acerto_tv=Acerto	questao_57_acerto _tf=Acerto	33.4	84.0	172.9
101506	questao_47_acerto_tf=Acerto/\questao_53_acerto_tv=Acerto/\ questao_45_acerto_tf=Acerto	questao_57_acerto _tf=Acerto	33.5	83.9	172.7
101510	questao_47_acerto_tf=Acerto/\questao_53_acerto_tv=Acerto/\ questao_22_acerto_pe=Acerto	questao_57_acerto _tf=Acerto	33.1	83.4	171.6
101514	questao_47_acerto_tf=Acerto/\questao_41_acerto_tv=Acerto/\ questao_50_acerto_tv=Acerto	questao_57_acerto _tf=Acerto	33.7	83.1	171.1
101690	questao_47_acerto_tf=Acerto/\questao_50_acerto_tv=Acerto/\ questao_32_acerto_pe=Acerto	questao_57_acerto _tf=Acerto	33.9	83.1	171.0
101446	questao_47_acerto_tf=Acerto/\questao_55_acerto_tv=Acerto/\ questao_25_acerto_pe=Acerto	questao_57_acerto _tf=Acerto	33.6	83.0	170.7
101698	questao_47_acerto_tf=Acerto/\questao_50_acerto_tv=Acerto/\ questao_20_acerto_pe=Acerto	questao_57_acerto _tf=Acerto	34.3	82.7	170.2
101710	questao_47_acerto_tf=Acerto/\questao_50_acerto_tv=Acerto/\ questao_2_acerto_ts=Acerto	questao_57_acerto _tf=Acerto	33.9	82.6	170.0
101866	questao_47_acerto_tf=Acerto/\questao_48_acerto_tv=Acerto/\ questao_32_acerto_pe=Acerto	questao_57_acerto _tf=Acerto	33.6	82.6	170.0
101522	questao_47_acerto_tf=Acerto/\questao_41_acerto_tv=Acerto/\ questao_48_acerto_tv=Acerto	questao_57_acerto _tf=Acerto	33.4	82.5	169.8
101413	questao_57_acerto_tf=Acerto/\questao_55_acerto_tv=Acerto/\ questao_49_acerto_tf=Acerto	questao_47_acerto _tf=Acerto	33.1	90.9	169.4
7038	questao_47_acerto_tf=Acerto/\questao_53_acerto_tv=Acerto	questao_57_acerto _tf=Acerto	34.5	82.2	169.1
101886	questao_47_acerto_tf=Acerto/\questao_48_acerto_tv=Acerto/\ questao_2_acerto_ts=Acerto	questao_57_acerto _tf=Acerto	33.7	82.2	169.1
101538	questao_47_acerto_tf=Acerto/\questao_41_acerto_tv=Acerto/\ questao_30_acerto_ts=Acerto	questao_57_acerto _tf=Acerto	34.2	82.1	169.0
101874	questao_47_acerto_tf=Acerto/\questao_48_acerto_tv=Acerto/\ questao_20_acerto_pe=Acerto	questao_57_acerto _tf=Acerto	33.3	82.1	169.0
102034	questao_47_acerto_tf=Acerto/\questao_32_acerto_pe=Acerto/ \questao_45_acerto_tf=Acerto	questao_57_acerto _tf=Acerto	34.5	82.1	168.9
101622	questao_47_acerto_tf=Acerto/\questao_49_acerto_tf=Acerto/\ questao_20_acerto_pe=Acerto	questao_57_acerto _tf=Acerto	33.1	82.0	168.8
101461	questao_57_acerto_tf=Acerto/\questao_55_acerto_tv=Acerto/\ questao_22_acerto_pe=Acerto	questao_47_acerto _tf=Acerto	34.0	90.6	168.7
101518	questao_47_acerto_tf=Acerto/\questao_41_acerto_tv=Acerto/\ questao_37_acerto_tv=Acerto	questao_57_acerto _tf=Acerto	34.1	82.0	168.7

Tabela 13: Primeiras 20 regras obtidas com algoritmo A Priori ordenadas por *lift*. Fonte: O autor.

Observando-se a Tabela 13 nota-se que o conseqüente da maioria das regras diz respeito à questão 57, que é uma questão do tipo tf, a palavra “Máchico”. Além disso, a maioria das regras envolve a questão 47 como parte do antecedente, que é uma questão do tipo tf, a palavra “Relóchio”. Observando-se mais regras do que as registradas aqui, nota-se que este comportamento se repete por pelo menos as primeiras 1000 regras ordenadas por *lift*, ou seja, as regras de maior importância. O significado desta constatação não pôde ser interpretado pelo autor, apenas foi notado que ocorrem diversas Regras de Associação de valor elevado (*lift*) envolvendo estas duas questões. A conclusão óbvia destas regras é a de que quem acerta a questão 47 costuma acertar também a questão 57, o que pode sugerir que ambas são equivalentes. Fica sugerido, assim, um estudo mais aprofundado destas regras de associação, preferencialmente por uma equipe multidisciplinar, capaz de, ao mesmo tempo, interpretar o valor destas regras tanto em termos de Mineração de Dados quanto em termos de Aprendizado de Leitura.

5 Conclusão

O Teste de Competência de Leitura de Palavras tem como finalidade classificar os alunos de acordo com sua série e seu estágio no processo de aprendizado de leitura. Para este fim acaba acumulando dados úteis para análises utilizando-se técnicas alternativas, como a Mineração de Dados. Nesta dissertação, os dados acumulados da aplicação do teste a 1669 alunos foram submetidos ao tratamento de três tarefas de Mineração de Dados: Classificação, agrupamento (seguido de classificação) e extração de regras de associação.

O resultado da tarefa de classificação foi classificadores de baixo desempenho, ou seja, não eram capazes de classificar os alunos em suas séries de maneira confiável. Apesar disso, a aplicação do algoritmo de classificação não foi uma tarefa sem proveito. A árvore de decisão resultante da primeira aplicação apresenta como primeiro critério de classificação a escola do aluno. Isso indica que o parâmetro escola é o mais importante na classificação da série de um aluno aleatório. Entretanto, esta informação não é útil, já que é sabido que cada escola participou com uma seleção distinta de séries. Algumas escolas participaram com alunos apenas da 1ª série, enquanto que outras participaram com alunos das quatro primeiras séries. Deste modo, fica claro que o parâmetro escola não deve ser utilizado para as demais tarefas de MD. Outro aspecto que ficou evidente nesta análise é que o parâmetro sexo não apresenta muito valor na classificação dos alunos quanto à série. Este parâmetro não apareceu na primeira árvore de decisão, tendo sido representado apenas na segunda, ocupando três ramos terminais da árvore (folhas), o que indica que são os parâmetros de menor peso. Além disso, nessas folhas o erro de classificação fica em torno de 50%, ou seja, o sexo não é um parâmetro útil na classificação dos alunos. Assim, para as demais tarefas de MD, este parâmetro não foi utilizado.

A tarefa de agrupamento foi executada com dois algoritmos: K-Means e Kohonen-SOM. Ambos foram configurados para dividir os alunos em quatro agrupamentos. Em ambos os casos, os algoritmos encontraram três agrupamentos com características semelhantes: Um primeiro agrupamento com alunos que tiveram bom desempenho, um segundo agrupamento com alunos que tenderam a marcar todas as questões como erradas e um terceiro agrupamento com alunos que tenderam a marcar todas as questões como corretas. Em outras palavras, dois algoritmos conseguiram, independentemente, identificar estes três comportamentos nos alunos que responderam ao teste. A diferença surge no quarto agrupamento. No caso do algoritmo K-Means, este agrupamento reúne alunos que tiveram baixo desempenho por terem desistido de responder todo o teste; e no caso do algoritmo Kohonem-SOM, o quarto agrupamento reúne alunos que ainda

não haviam realizado a transição do estágio alfabético para o estágio ortográfico de leitura. A seguir, os agrupamentos obtidos nesta etapa foram utilizados pelo algoritmo de classificação C4.5. As árvores de decisão resultantes registram quais são os resultados do teste mais significativos para se classificar um aluno aleatório segundo os agrupamentos obtidos na etapa anterior. Por exemplo, observando-se o Quadro 4, pela regra 1.1.1, nota-se que basta verificar baixo desempenho em questões do tipo pe, ph e ts para se concluir que trata-se de um aluno que tendeu a marcar todas as questões como corretas. Isso permite a automatização da classificação de um aluno aleatório quanto a estes agrupamentos. Basta para tanto implementar a árvore de decisão em um programa de computador.

A terceira tarefa de MD empregada foi a obtenção de Regras de Associação. Utilizando-se os dados disponíveis, foi aplicado o algoritmo A Priori, obtendo-se 1.152.761 regras de associação. Organizou-se estas regras por ordem decrescente de *lift*, confiança e suporte, respectivamente. Com isso, observou-se que há uma grande incidência de regras de grande importância envolvendo as questões 57 e 47, tanto como antecedentes quanto como conseqüentes. O significado desta constatação pode ser averiguado mais apropriadamente por uma equipe multidisciplinar, envolvendo especialistas em Mineração de Dados, em Leitura e no Teste de Competência de Leitura de Palavras. Assim, fica sugerida a continuação deste estudo por uma equipe composta desta maneira. A conclusão óbvia que pode ser tirada desta constatação é a de que o aluno que responde corretamente à questão 47 costuma também responder corretamente à questão 57 e vice-versa. Isto sugere que estas questões podem ser equivalentes e redundantes.

A hipótese de pesquisa desta dissertação de mestrado especifica que o uso de técnicas de MD é capaz de fornecer informações além das oferecidas pelo tratamento estatístico habitual dos resultados do TCLP. Isso foi comprovado no momento em que os algoritmos de agrupamento detectaram dois grupos de alunos que tenderam a responder ao teste de maneira errada, seja marcando todas as questões como corretas, seja marcando-as como incorretas. Além disso, com a aplicação do algoritmo A Priori de extração de Regras de Associação, observou-se que existe uma forte relação entre duas questões do teste. Esta informação pode ser útil no aprimoramento do próprio TCLP, caso seja observado em trabalhos futuros que tais questões são mesmo equivalentes.

Os objetivos específicos foram alcançados, ou seja, foi estudado um conjunto representante de técnicas de MD, com suas características e propriedades e estes foram aplicados ao problema em questão, o estudo dos resultados do TCLP. Uma metodologia de aplicação foi estabelecida e os resultados foram estudados, encontrando-se propriedades e características não evidentes no conjunto de dados. Isso permitiu que o objetivo geral fosse atingido, ou seja, foram identificadas novas informações presentes nos dados obtidos com uma aplicação do TCLP, complementando o

tratamento tradicional destes dados e fornecendo subsídios para a automatização da classificação de um aluno aleatório segundo os agrupamentos descobertos.

6 Referências

AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. **Mining association rules between sets of items in large databases.** In: Proceedings of the 1993 ACM SIGMOD international conference on management of data. Washington, D. C., p. 207 – 216, 1993. ISBN: 0-89791-592-5. Disponível na Internet: < http://portal.acm.org/citation.cfm?id=170072>. Acesso em 21/08/2008.

AGRAWAL, R.; SRIKANT, R. **Fast algorithms for mining association rules in large databases**. In: Proceedings of the 20th International Conference on Very Large Data Bases, p. 487-499, 1994. ISBN: 1-55860-153-8. Disponível na Internet: < http://www.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/vldb94_rj.pdf>. Acesso em 20/08/2008.

CAPOVILLA, A. G. S.; JOLY, M. C. R. A.; FERRACINI, F., *et al.,* **Estratégias de leitura e desempenho em escrita no início da alfabetização**: estratégias de leitura e alfabetização. In: *Psicologia Escolar e Educacional*. dez. 2004, vol. 8, n. 2, p.189-197. Disponível na Internet: <http://pepsic.bvs-psi.org.br/scielo.php?script=sci_arttext&pid=S1413-85572004000200007&lng=pt&nrm=iso>. ISSN 1413-8557. Acesso em 15/06/2009.

CAPOVILLA, F. C., VARANDA, C. e CAPOVILLA, A. G. S. **Teste de competência de leitura de palavras e pseudopalavras**: normatização e validação. *Psic*. dez. 2006, vol.7, no.2, p.47-59. Disponível na Internet: <http://pepsic.bvs-psi.org.br/scielo.php?script=sci_arttext&pid=S1676-73142006000200007&lng=pt&nrm=iso>. ISSN 1676-7314. Acesso em 15/06/2009.

CAPOVILLA, F.; CAPOVILLA, A. G. S.; VIGGIANO, K. *et al*. **Processos logográficos, alfabéticos e lexicais na leitura silenciosa por surdos e ouvintes**. Estud. psicol. (Natal). [online]. Jan./Apr. 2005, vol.10, no.1 [citado 03 Junho 2006], p.15-23. Disponível na World Wide Web: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-294X2005000100003&lng=en&nrm=iso> . ISSN 1413-294X. Acesso em: 03/06/2007.

CHAWLA, N. V.; CIESLAK, D. A.; HALL, L. O. *et al*. **Automatically countering imbalance and its empirical relationship to cost.** In Journal f Data Mining and Knowledge Discovery, Feb. 2008, ISSN 1384-5810 (Impresso), 1573-756X (online), DOI 10.1007/s10618-008-0087-0. Acesso em 15/06/2009.

CHEN, M.S.;HAN, J.; YU, P.S. **Data Mining**: An Overview from a Database Perspective, IEEE Transactions on Knowledge and Data Engineering, 8(6): 866-883, 1996.

DOMINGUES, M. A.; REZENDE, S. O. **Generalização de regras de associação usando taxonomias**. in Workshop de Computação da Região Sul, 1., Florianópolis - SC, 2004. Disponível na Internet: <http://inf.unisul.br/~ines/workcomp/cd/pdfs/2375.pdf>. Acesso em 16/03/2008.

FAUSETT, L. **Fundamentals of neural networks**: architectures, algorithms and applications. New Jersey: Prentice-Hall, 1994. ISBN 0-13-334186-0.

FRITH, U. **Beneath the surface of developmental dyslexia**. *In* Patterson, K., Marshall, J. Coltheart, M. eds. Surface dyslexia: Neuropsychological and cognitive studies of phonological reading: London, UK, Lawrence Erlbaum, 1985 *apud* GUNJI, J. C. G.; ALMEIDA, M. A. de; DUDUCHI, M. *et al*. **Uma análise de resultados do teste de competência em leitura de palavras usando mineração de dados**. In: XI Congresso Brasileiro de Informática em Saúde, Campos do Jordão, 2008.

GARCIA-PEDRAJAS, N.; GARCIA-OSORIO, C.; FYFE, C. **Nonlinear boosting projections for ensemble contruction.** In The Journal of Machine Learning Research, Vol. 8, Oct. 2007, pg. 1-33, ISSN: 1533-7928.

GHOTHING, A.; PARTHASARATHY, S.; OTEY, M. E. **Fast mining of distance-based outliers in high-dimensional datasets**. In: Journal Data Mining and Knowledge Discovery, V. 16, No. 3, p. 249 – 364, June, 2008. DOI: 10.1007/s10618-008-0093-2.

GUNJI, J. C. G.; ALMEIDA, M. A. de; DUDUCHI, M. *et al*. **Uma análise de resultados do teste de competência em leitura de palavras usando mineração de dados**. In: XI Congresso Brasileiro de Informática em Saúde, Campos do Jordão, 2008.

JIANG, M. F.; TSENG, S. S.; SU, C. M. **Two-ohase clustering process for outliers detection**. In: Pattern Recognition Letters, Vol. 22, p. 691 – 700, 2001. DOI: 10.1016/S0167-8655(00)00131-8

KOHONEN, T. **The self-organizing map**. In: Neurocomputing, Vol. 21, p. 1 – 6, 1998. DOI: 10.1016/S0925-2312(98)00030-7.

LYON, G.R. Defining **dyslexia, comorbidity, teachers' knowledge of language and reading**. In: Annals of Dyslexia. v. 53, p. 1-14, 2003 *apud* GUNJI, J. C. G.; ALMEIDA, M. A. de; DUDUCHI, M. *et al*. **Uma análise de resultados do teste de competência em leitura de palavras usando mineração de dados**. In: XI Congresso Brasileiro de Informática em Saúde, Campos do Jordão, 2008.

MACEDO, E. C. de; CAPOVILLA, F. C.; NIKAEDO, C. C. *et al*. **Teleavaliação da habilidade de leitura no ensino infantil fundamental**. Psicol. esc. educ. [online]. jun. 2005, vol.9, no.1 [citado 03 Junho 2006], p.37-46. Disponível na World Wide Web: <http://scielo.bvs-psi.org.br/scielo.php?script=sci_arttext&pid=S1413-85572005000100012&lng=pt&nrm=iso>. Acesso em: 03/06/2007. ISSN 1413-8557. Acesso em 15/06/2009.

NIKAEDO, C. C., KURIYAMA, C. T. e MACEDO, E. C. de. **Avaliação longitudinal de leitura e escrita com testes de diferentes pressupostos teóricos**. Psic. [online]. dez. 2007, vol.8, no.2, p.185-193. Disponível Internet: <http://pepsic.bvs-psi.org.br/scielo.php?script=sci_arttext&pid=S1676-73142007000200009&lng=pt&nrm=iso>. ISSN 1676-7314. Acesso em 15/06/2009.

ORACLE® data mining concepts 11g release 1 (11.1). Copyright © 2005, 2008. Disponível na Internet: <http://download.oracle.com/docs/cd/B28359_01/datamine.111/b28129/algo_apriori.htm>. Último acesso em 14/03/2009.

QUINLAN, J. R. **Improved Use of Continuous Attributes in C4.5** in Journal of Artificial Intelligence Research, Vol 4, 1996, pg. 77-90.

RAKOTOMALALA, R. **Tanagra** : un logiciel gratuit pour l'enseignement et la recherche, in *Actes de EGC* '2005, RNTI-E-3, vol. 2, p.697-702, 2005.

REZENDE, S. O. **Mineração de dados**. In: Encontro Nacional de Inteligência Artificial, 5., São Leopoldo – RS, 2005.

REZENDE, S. O. **Sistemas Inteligentes**: fundamentos e aplicações. Barueri, SP: Manole, 2003. ISBN 85-204-1683-7.

ROMÃO, W. ; NIEDERAUER, C. A. P. ; MARTINS, A. *et al*. **Extração de Regras de Associação em C&T**: O Algoritmo Apriori. In: XIX Encontro Nacional em Engenharia de Produção, 1999, Rio de Janeiro. XIX Encontro Nacional em Engenharia de Produção, 1999.

RUSSELL, S., NORVIG, P. **Inteligência Artificial**: trad. da 2ª ed. Rio de Janeiro: Elsevier, 2004. ISBN 85-352-1177-2.

TIBSHIRANI, R.; HASTIE, T. **Margin trees for high-dimensional classification**. In The Journal of Machine Learning Research, Vol. 8, Oct. 2007, pg. 637-652, ISSN: 1533-7928.

7 Apêndice 1

Árvore de Decisão Resultante da Primeira Aplicação do Algoritmo C4.5

- escola_id em [4.] então aluno_serie = **1a** (91,25 % de 80 exemplos)
- escola_id em [7.]
    - total_respondidas < 69,5000 então aluno_serie = **1a** (62,50 % de 16 exemplos)
    - total_respondidas >= 69,5000
        - total_tv < 7,5000 então aluno_serie = **8a** (73,33 % de 15 exemplos)
        - total_tv >= 7,5000
            - total_ph < 4,5000 então aluno_serie = **8a** (42,11 % de 19 exemplos)
            - total_ph >= 4,5000
                - total_ph < 7,5000 então aluno_serie = **2a** (48,00 % de 25 exemplos)
                - total_ph >= 7,5000
                    - total_ph < 9,5000
                        - total_tf < 9,5000 então aluno_serie = **4a** (53,85 % de 13 exemplos)
                        - total_tf >= 9,5000 então aluno_serie = **3a** (44,44 % de 18 exemplos)
                    - total_ph >= 9,5000 então aluno_serie = **4a** (57,89 % de 38 exemplos)
- escola_id em [10.]
    - total_ci < 1,5000 então aluno_serie = **1a** (100,00 % de 12 exemplos)
    - total_ci >= 1,5000
        - total_ts < 7,5000
            - total_ci < 5,5000 então aluno_serie = **1a** (85,71 % de 14 exemplos)
            - total_ci >= 5,5000
                - total_ph < 5,5000 então aluno_serie = **1a** (50,00 % de 12 exemplos)
                - total_ph >= 5,5000 então aluno_serie = **2a** (34,62 % de 26 exemplos)
        - total_ts >= 7,5000
            - total_cr < 4,5000
                - total_tv < 4,5000 então aluno_serie = **4a** (40,00 % de 10 exemplos)
                - total_tv >= 4,5000
                    - total_ci < 5,5000 então aluno_serie = **2a** (40,91 % de 22 exemplos)
                    - total_ci >= 5,5000 então aluno_serie = **4a** (48,84 % de 43 exemplos)
            - total_cr >= 4,5000
                - total_ci < 8,5000 então aluno_serie = **2a** (70,59 % de 17 exemplos)
                - total_ci >= 8,5000 então aluno_serie = **4a** (83,33 % de 12 exemplos)
- escola_id em [11.] então aluno_serie = **1a** (100,00 % de 1 exemplos)
- escola_id em [12.]
    - total_ts < 8,5000 então aluno_serie = **8a** (87,76 % de 49 exemplos)
    - total_ts >= 8,5000
        - total_cr < 6,5000 então aluno_serie = **8a** (87,50 % de 16 exemplos)
        - total_cr >= 6,5000 então aluno_serie = **1a** (90,14 % de 71 exemplos)
- escola_id em [13.]
    - total_cr < 3,5000 então aluno_serie = **1a** (92,98 % de 114 exemplos)
    - total_cr >= 3,5000
        - total_ts < 0,5000 então aluno_serie = **1a** (94,44 % de 36 exemplos)
        - total_ts >= 0,5000
            - total_cr < 7,5000
                - total_ci < 7,5000
                    - total_ci < 6,5000
                        - total_pe < 7,5000
                            - total_ts < 6,5000 então aluno_serie = **1a** (67,65 % de 34 exemplos)
                            - total_ts >= 6,5000 então aluno_serie = **2a** (57,14 % de 14 exemplos)
                        - total_pe >= 7,5000 então aluno_serie = **1a** (76,92 % de 39 exemplos)
                    - total_ci >= 6,5000 então aluno_serie = **2a** (54,55 % de 11 exemplos)
                - total_ci >= 7,5000 então aluno_serie = **1a** (83,33 % de 12 exemplos)
            - total_cr >= 7,5000
                - total_ts < 8,5000

- total_pe < 6,5000
  - total_tv < 1,5000 então aluno_serie = **1a** (56,52 % de 23 exemplos)
  - total_tv >= 1,5000 então aluno_serie = **2a** (76,92 % de 13 exemplos)
- total_pe >= 6,5000
  - total_pe < 8,5000 então aluno_serie = **1a** (80,00 % de 10 exemplos)
  - total_pe >= 8,5000 então aluno_serie = **2a** (60,00 % de 10 exemplos)

- total_ts >= 8,5000 então aluno_serie = **2a** (75,38 % de 65 exemplos)

- escola_id em [15.] então aluno_serie = **1a** (100,00 % de 5 exemplos)
- escola_id em [17.]
  - total_ci < 6,5000
    - total_tf < 2,5000 então aluno_serie = **4a** (38,46 % de 13 exemplos)
    - total_tf >= 2,5000
      - total_tv < 7,5000 então aluno_serie = **1a** (75,00 % de 44 exemplos)
      - total_tv >= 7,5000
        - total_cr < 5,5000 então aluno_serie = **1a** (69,57 % de 23 exemplos)
        - total_cr >= 5,5000 então aluno_serie = **3a** (47,37 % de 19 exemplos)
  - total_ci >= 6,5000
    - total_ph < 4,5000
      - total_tv < 2,5000 então aluno_serie = **1a** (54,29 % de 35 exemplos)
      - total_tv >= 2,5000
        - total_cr < 9,5000
          - total_ph < 3,5000
            - total_tf < 7,5000
              - aluno_sexo em [F] então aluno_serie = **1a** (42,11 % de 19 exemplos)
              - aluno_sexo em [M] então aluno_serie = **3a** (44,44 % de 27 exemplos)
            - total_tf >= 7,5000 então aluno_serie = **2a** (63,16 % de 19 exemplos)
          - total_ph >= 3,5000 então aluno_serie = **3a** (63,64 % de 11 exemplos)
        - total_cr >= 9,5000
          - total_tf < 4,5000
            - aluno_sexo em [F] então aluno_serie = **2a** (38,89 % de 18 exemplos)
            - aluno_sexo em [M] então aluno_serie = **1a** (57,14 % de 14 exemplos)
          - total_tf >= 4,5000
            - aluno_sexo em [F] então aluno_serie = **2a** (40,00 % de 35 exemplos)
            - aluno_sexo em [M]
              - total_ph < 2,5000 então aluno_serie = **2a** (33,33 % de 21 exemplos)
              - total_ph >= 2,5000 então aluno_serie = **3a** (61,11 % de 18 exemplos)
    - total_ph >= 4,5000
      - total_ph < 9,5000
        - total_tv < 6,5000 então aluno_serie = **1a** (33,33 % de 15 exemplos)
        - total_tv >= 6,5000
          - total_pe < 9,5000 então aluno_serie = **3a** (54,55 % de 11 exemplos)
          - total_pe >= 9,5000
            - total_tf < 8,5000 então aluno_serie = **3a** (55,56 % de 36 exemplos)
            - total_tf >= 8,5000
              - total_ci < 8,5000 então aluno_serie = **3a** (69,23 % de 13 exemplos)
              - total_ci >= 8,5000 então aluno_serie = **4a** (54,35 % de 46 exemplos)
      - total_ph >= 9,5000 então aluno_serie = **2a** (45,95 % de 37 exemplos)
- escola_id em [18.]
  - total_pe < 7,5000 então aluno_serie = **1a** (77,31 % de 119 exemplos)
  - total_pe >= 7,5000
    - total_ci < 4,5000 então aluno_serie = **1a** (75,00 % de 52 exemplos)
    - total_ci >= 4,5000

- total_ci < 7,5000
  - total_ci < 5,5000 então aluno_serie = **1a** (53,85 % de 13 exemplos)
  - total_ci >= 5,5000
    - total_ph < 3,5000 então aluno_serie = **2a** (64,29 % de 14 exemplos)
    - total_ph >= 3,5000 então aluno_serie = **1a** (60,00 % de 15 exemplos)
- total_ci >= 7,5000 então aluno_serie = **2a** (73,13 % de 67 exemplos)

8 Apêndice 2

Gráficos tridimensionais em separação estereográfica para visualização com auxílio de óculos 3D, do tipo com a lente esquerda vermelha e a lente direita azul.

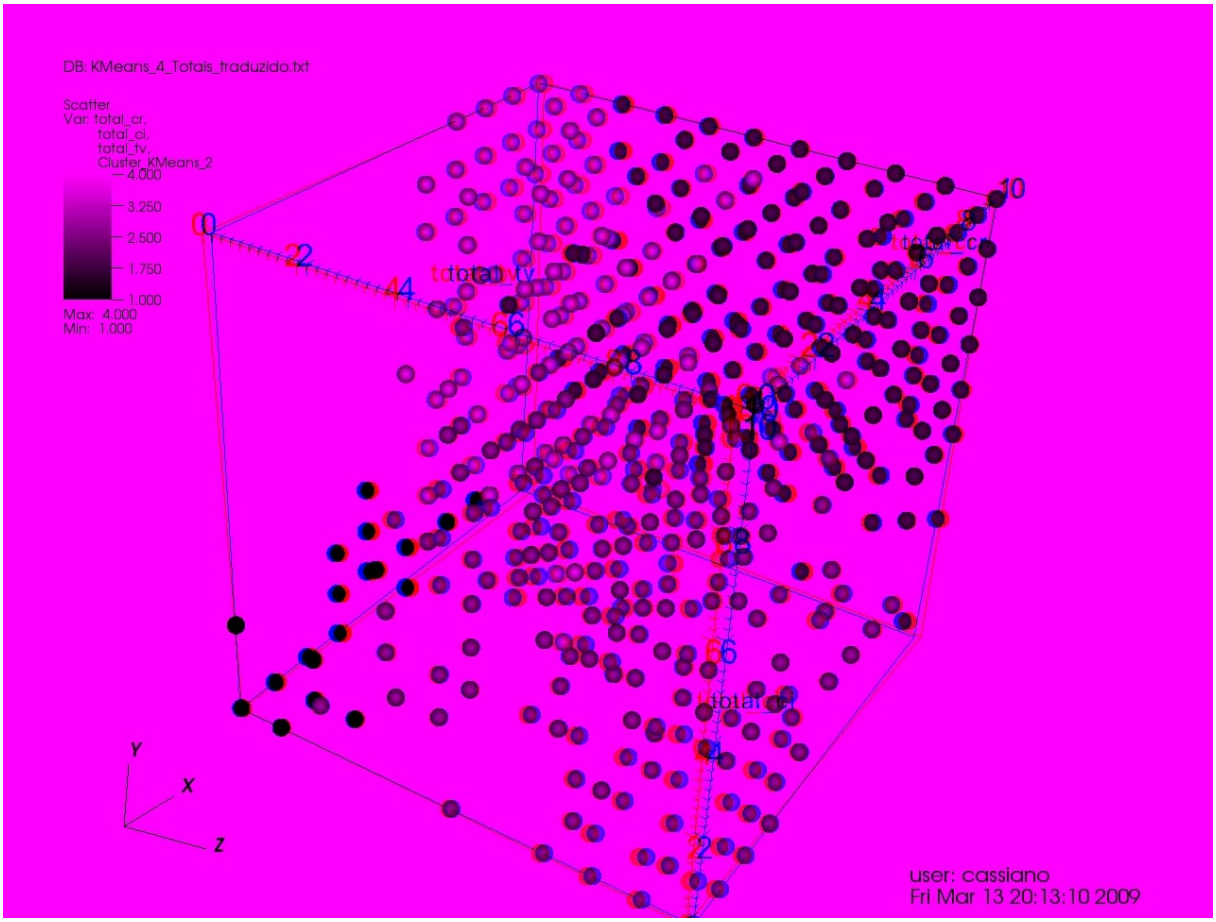

Figura 10: Visualização estereográfica mostrando agrupamentos obtidos por K-Means em um gráfico de dispersão: X – total_cr, Y – total_ci, Z – total_tv e cor – agrupamentos. Agrupamentos assumem as cores: Azul – c_kmeans_1, verde – c_kmeans_2, amarelo – c_kmeans_3 e vermelho – c_kmeans_4. Fonte: O autor.

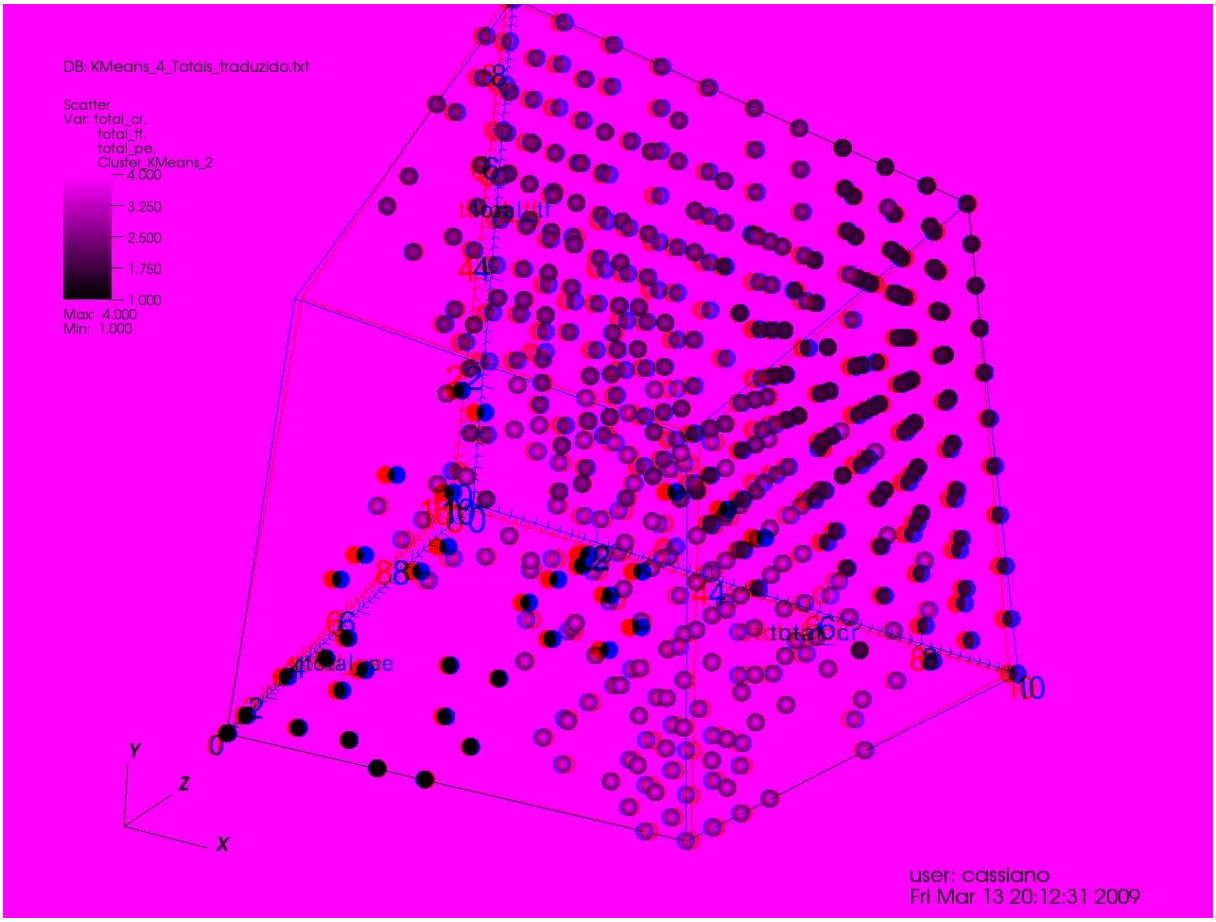

Figura 11: Visualização estereográfica mostrando agrupamentos obtidos por K-Means em um gráfico de dispersão: X – total_cr, Y – total_tf, Z – total_pe e cor – agrupamentos. Agrupamentos assumem as cores: Azul – c_kmeans_1, verde – c_kmeans_2, amarelo – c_kmeans_3 e vermelho – c_kmeans_4. Fonte: O autor.

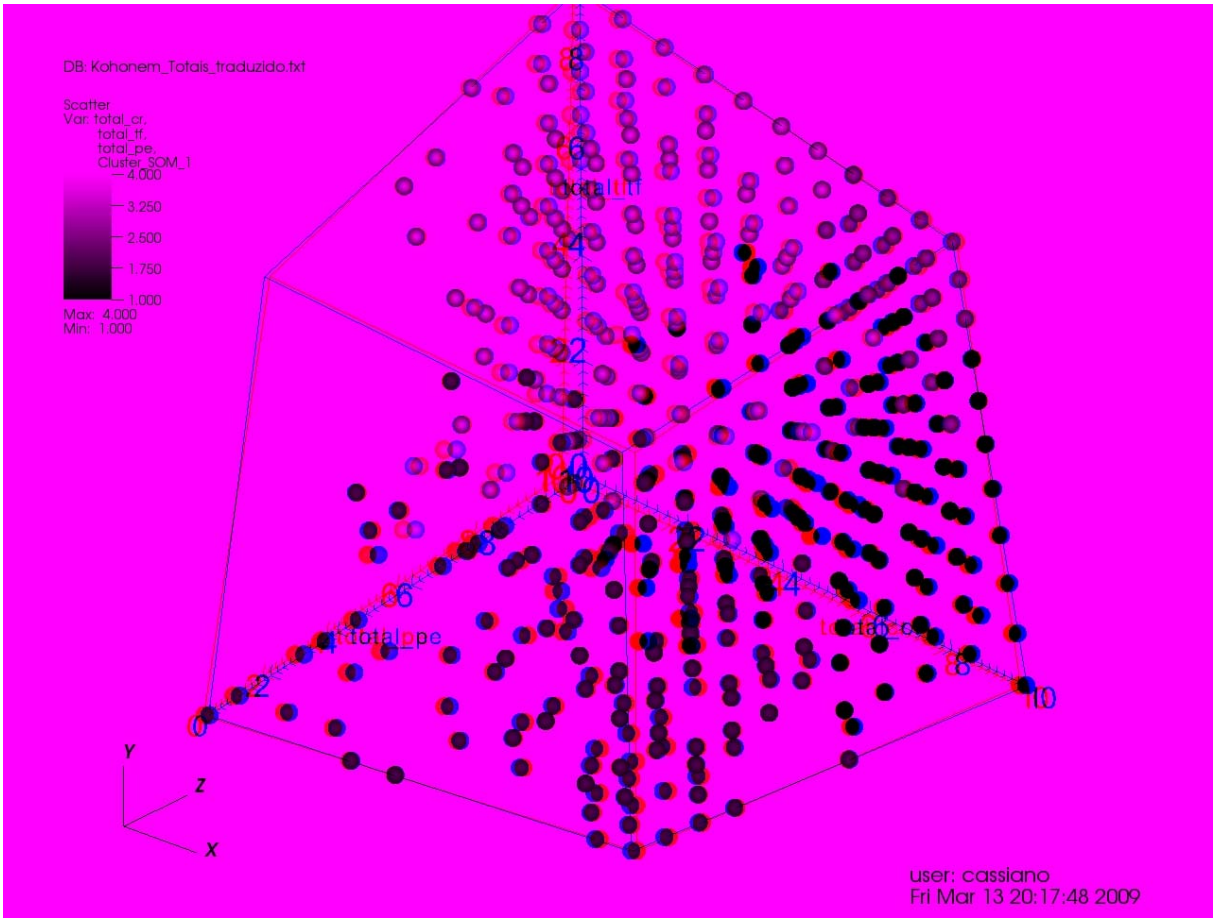

Figura 12: Visualização estereográfica mostrando agrupamentos obtidos por Mapas Auto-Organizados de Kohonen em um gráfico de dispersão: X – total_cr, Y – total_tf, Z – total_pe e cor – agrupamentos. Agrupamentos assumem as cores: Azul – c_som_1_1, verde – c_som_1_2, amarelo – c_som_2_1 e vermelho – c_som_2_2. Fonte: O autor.

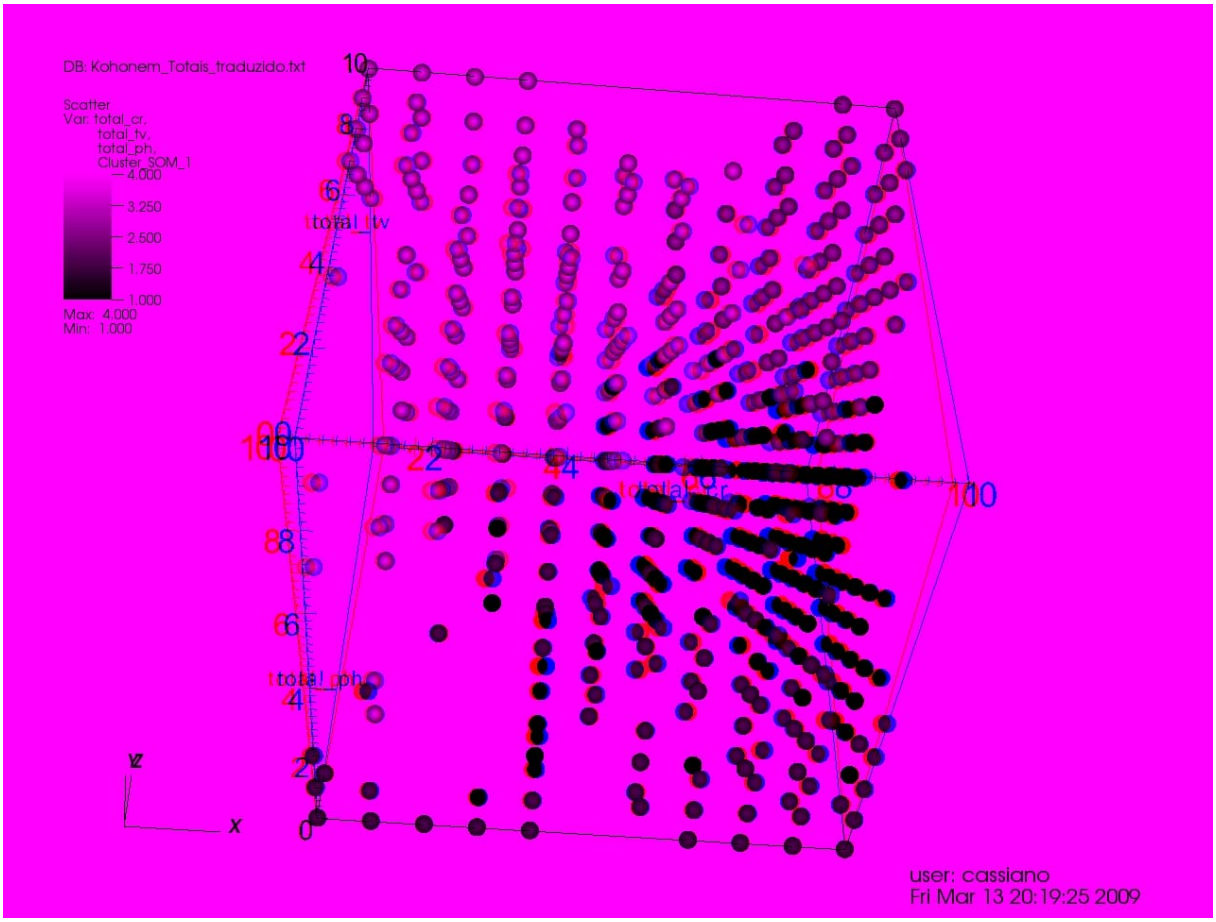

Figura 13: Visualização estereográfica mostrando agrupamentos obtidos por Mapas Auto-Organizados de Kohonen em um gráfico de dispersão: X – total_cr, Y – total_tv, Z – total_ph e cor – agrupamentos. Agrupamentos assumem as cores: Azul – c_som_1_1, verde – c_som_1_2, amarelo – c_som_2_1 e vermelho – c_som_2_2. Fonte: O autor.