Aplicação dos algoritmos K-Means e C4.5 aos resultados do teste de competência em leitura de palavras

José Cassiano Grassi Gunji
Centro Estadual de Educação Tecnológica Paula Souza (CEETEPS) – São Paulo, SP – Brasil
cassiano.gunji@gmail.com

Maurício Amaral de Almeida
Centro Estadual de Educação Tecnológica Paula Souza (CEETEPS) – São Paulo, SP – Brasil
madealmeida@gmail.com

Resumo: O Teste de Competência em Leitura de Palavras é uma ferramenta útil para se medir o nível de desenvolvimento das habilidades de leitura de estudantes em fase de alfabetização. Neste artigo, os resultados obtidos de uma aplicação do TCLP são submetidos a técnicas de Mineração de Dados. A aplicação do algoritmo de agrupamento K-Means identificou alunos que demonstraram comportamentos distintos. A seguir, a aplicação do algoritmo de classificação C4.5 forneceu um classificador capaz de identificar rapidamente tais comportamentos.

Palavras chave: Mineração de Dados, K-Means, C4.5, Alfabetização, Educação.

1 Introdução

O Teste de Competência em Leitura de Palavras (TCLP) é uma ferramenta útil para se medir o nível de desenvolvimento das habilidades de leitura em estudantes em alfabetização. Ele apresenta resultados significativos quando aplicado a alunos variando do terceiro ano do Ensino Infantil até o oitavo ano do Ensino Fundamental [1]. Apesar disso, o TCLP é mais usualmente empregado na avaliação de alunos do primeiro ao quarto ano do Ensino Fundamental [2]. Com a aplicação deste teste, é possível a medição do nível e do estágio da alfabetização do aluno e como ele se compara com a média dos alunos em seu nível escolar. Este artigo investiga a possibilidade de se obter mais informações da aplicação do TCLP com o auxílio de técnicas de Mineração de Dados (MD), uma especialização da Inteligência Artificial (IA). Para tanto é apropriada uma breve discussão sobre estes dois assuntos de natureza interdisciplinar, o TCLP e a MD.

1.1 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 de papel e lápis [1], mas possui uma versão eletrônica aplicada via Internet [2]. 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 cruzar (assinalar com um “X”) os itens incorretos. Há 7 tipos de itens distribuídos aleatoriamente ao longo do teste, com dez itens de teste para cada tipo. Dois dos tipos são compostos por palavras corretas. Os demais tipos são compostos de pseudopalavras, cada um representando um tipo diferente de erro. Cada erro tem a finalidade de diagnosticar um tipo diferente de falha no processo de aprendizado. A Figura 1 mostra alguns exemplos de itens de teste que compõe o TCLP.

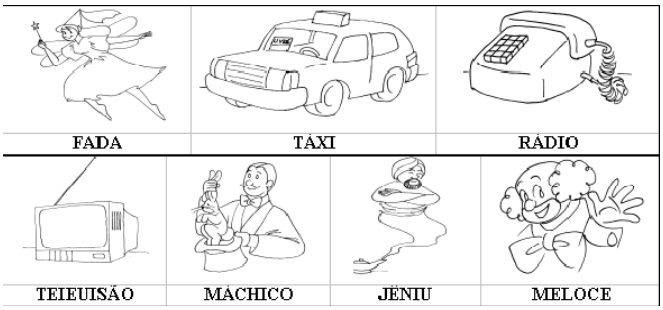

Figura 1: Exemplos de pares figura-palavra que compõe os itens de teste do TCLP [1].

Os 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 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, F.; CAPOVILLA, A.; VIGGIANO *et al*, 2005; MACEDO; CAPOVILLA; NIKAEDO *et al*, 2005).

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.

1.1.1 As Fases do Processo de Alfabetização

O processo de alfabetização se dá em três estágios [3], [4]. 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 competentes.

Capovilla, Joly, Ferracini, *et al* [4] 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” [7] *apud* [5], pg. 19. [5], [7] *apud* [5], pg. 19 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.

1.2 A Mineração de Dados

Rezende [8], [9] 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” [10] *apud* [8], 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" [8], pg. 2.

O processo de MD pode ser dividido em três grandes etapas: 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 [8].

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 [8], pg. 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 [9], [8] 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.

2 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.

O primeiro passo na preparação dos dados foi a exclusão das respostas aos 8 itens de treino. Também foi calculada a somatória de acertos em cada categoria de pergunta. Por fim, a tabela foi formatada no padrão esperado pelo sistema Tanagra [11]. Ao final da tradução obteve-se um total de 1669 linhas, cada uma representando uma aplicação do TCLP a um aluno.

O sistema Tanagra foi escolhido, pois seu uso é livre e gratuito para fins de ensino e pesquisa. Além disso, ele reúne em uma única plataforma um acervo respeitável de recursos de estatística e algoritmos de MD. Por ser um *software* de código aberto, permite que alterações sejam feitas nos algoritmos ou que a plataforma seja utilizada na implementação de novos algoritmos. Outra característica interessante é que os algoritmos implementados neste sistema são bem conhecidos e bastante difundidos nos meios acadêmicos. 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.

O processo de extração de padrões foi feito interativamente. Vários algoritmos foram experimentados para a tarefa de MD de descrição. A abordagem foi sempre *bottom-up*, tanto na modalidade supervisionada quanto na não-supervisionada. Os resultados mais significativos estão expostos no tópico a seguir.

3 Resultados

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

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

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.

3.1 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 [12]. Para esta tarefa foi escolhido o algoritmo, K-Means [13], um algoritmo clássicos freqüentemente utilizado como parâmetro de comparação para novos algoritmos.

3.1.1 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 3.

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 3: 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.

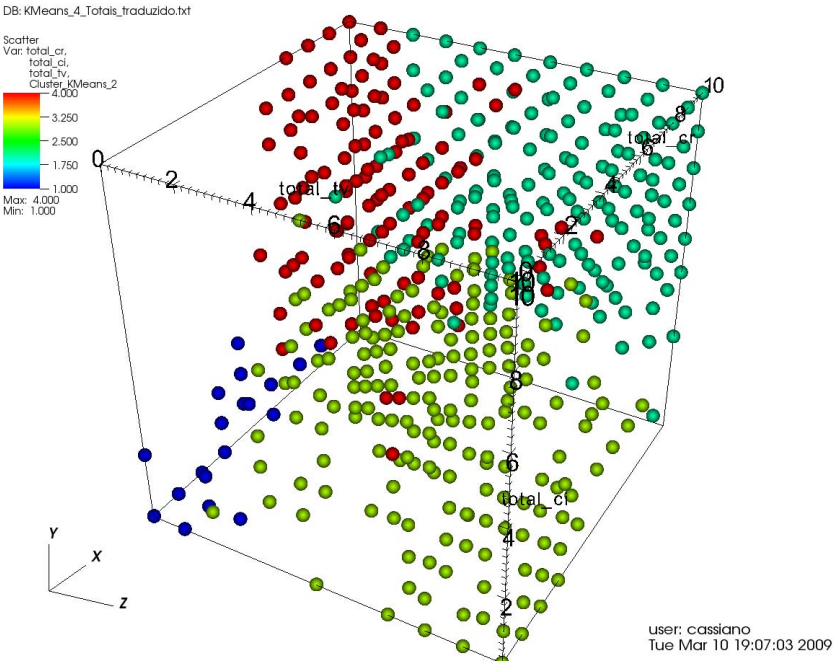

Figura 3: Agrupamentos obtidos por K-Means em um gráfico de dispersão: X – total_cr, Y – total_ci, Z – total_tv e cor – agrupamentos. Agrupamentos assumem as cores: Azul – c_kmeans_1, verde – c_kmeans_2, amarelo – c_kmeans_3 e vermelho – c_kmeans_4. Fonte: O autor.

Na Figura 3 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.

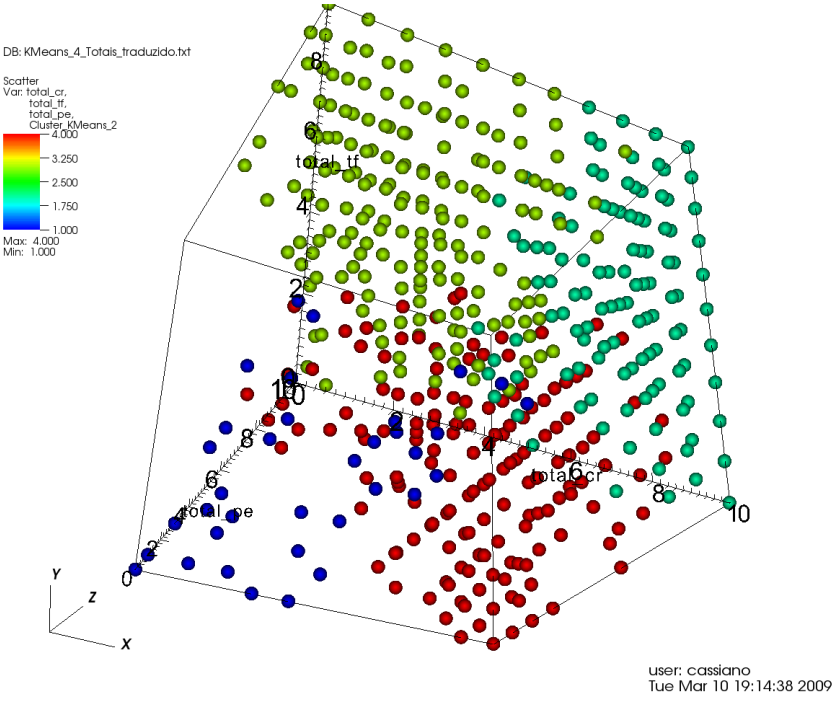

Figura 4: Agrupamentos obtidos por K-Means em um gráfico de dispersão: X – total_cr, Y – total_tf, Z – total_pe e cor – agrupamentos. Agrupamentos assumem as cores: Azul – c_kmeans_1, verde – c_kmeans_2, amarelo – c_kmeans_3 e vermelho – c_kmeans_4. Fonte: O autor.

Prosseguindo com a interpretação dos agrupamentos utilizando-se agora a Figura 4, 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 4.

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

Tabela 4: 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 4 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.

3.1.1.1 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 o algoritmo C4.5 [14], obtendo-se a matriz de confusão mostrada na Tabela 5.

Razão de erro	0,0816

Predição de valores

Valor	*Razão de acerto*
c_kmeans_1	0,9315
c_kmeans_2	0,9345
c_kmeans_3	0,8973
c_kmeans_4	0,9055

Matriz de confusão

	c_kmeans_1	*c_kmeans_2*	*c_kmeans_3*	*c_kmeans_4*	*Soma*
c_kmeans_1	1958	49	60	35	2102
c_kmeans_2	53	2254	64	41	2412
c_kmeans_3	67	87	1878	61	2093
c_kmeans_4	50	59	51	1533	1693
Soma	2128	2449	2053	1670	8300

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

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 [15], [11], [9]. Neste artigo foi feita a validação cruzada com 10 vias e 5 repetições.

Na Tabela 5 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 1. 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 1: Árvore de decisão que classifica os dados nos agrupamentos obtidos por K-Means. Fonte: O autor.

4 Conclusões

O TCLP é aplicado com o intuito de se verificar o desempenho de um aluno frente ao desempenho de outros alunos em seu mesmo nível de escolaridade. Esta comparação é feita com o uso de tabelas normatizadas, mediante um tratamento estatístico. A aplicação do algoritmo de agrupamento ao conjunto de dados evidenciou que os alunos que responderam ao teste se comportaram de maneiras diferentes, mas identificáveis. Foram identificados quatro grupos de alunos, os que obtiveram um bom desempenho, os que obtiveram desempenho baixo e os grupos que tenderam a responder todas as questões de uma mesma maneira, seja marcando todas as questões com um "X", seja marcando todas com um círculo. Não há muito o que se descobrir do estudo dos resultados do teste obtidos por alunos destes dois últimos grupos, pois eles assumiram um comportamento fixo, não responderam corretamente ao teste. Deste modo, é recomendável que se retire da amostra tais alunos. A aplicação do algoritmo de classificação fornece uma árvore de decisão útil para que se identifiquem rapidamente alunos que pertençam a estes grupos de pouco interesse. Os resultados destes alunos podem ser retirados da amostra com o uso desta árvore de decisão, e a amostra assim filtrada ser utilizada em estudos posteriores. Deste modo, os esforços podem se concentrar nos resultados de alunos que responderam corretamente ao teste.

Grosseiramente, observa-se na Tabela 3 que metade dos alunos respondeu ao TCLP de maneira correta, evidenciando que já se encontram alfabetizados. Mas também se observa que metade dos alunos não respondeu corretamente ao teste. Eles marcaram a maioria das respostas da mesma maneira. É necessário averiguar qual o motivo para tal comportamento. Há a possibilidade deste comportamento evidenciar que os alunos não estejam entendendo o que eles devem fazer.

Por fim, a maior contribuição deste artigo é ilustrar como algoritmos de Mineração de Dados podem auxiliar na interpretação de grandes quantidades de dados. Algoritmos de agrupamento podem ajudar a identificar tendências e comportamentos comuns e algoritmos de classificação fornecem ferramentas que facilitam a rápida identificação de tais tendências e comportamentos.

5 Referências

[1] CAPOVILLA, F.; CAPOVILLA, A. G. S.; VIGGIANO, K. *et al*., **Silent reading by deaf and hearing readers:** logographic, alphabetical and lexical processes. *Estud. psicol. (Natal)*, Jan./Apr. 2005, vol.10, no.1, p.15-23. Disponível na Internet: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-294X2005000100003&lng=en&nrm=iso>. Acesso em: 21 abr. 2008. ISSN 1413-294X.

[2] MACEDO, E. C. de; CAPOVILLA, F. C.; NIKAEDO, C. C. *et al*. **Teleavaliação da habilidade de leitura no ensino infantil fundamental**. *Psicol. esc. educ.* Jun. 2005, vol.9, no.1, p.37-46. Disponível na Internet: <http://pepsic.bvs-psi.org.br/scielo.php?script=sci_arttext&pid=S1413-85572005000100012&lng=pt&nrm=iso>. ISSN 1413-8557.

[3] 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>. Acesso em: 21 abr. 2008. ISSN 1676-7314.

[4] 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, no.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>. Acesso em: 21 abr. 2008. ISSN 1413-8557.

[5] GUNJI, J. C. G. **Aplicação de técnicas de mineração de dados na avaliação de resultados de teste de competência de leitura de palavras (TCLP)**, 2009. Dissertação (Mestrado em Tecnologia: Gestão, Desenvolvimento e Formação), Centro Estadual de Educação Tecnológica Paula Souza, São Paulo.

[6] 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* [5].

[7] LYON, G.R. **Defining dyslexia**, comorbidity, teachers' knowledge of language and reading. *Annals of Dyslexia*. v. 53, p. 1-14, 2003 *apud* [5].

[8] REZENDE, S. O., **Mineração de dados**. In: *Encontro Nacional de Inteligência Artificial*, 5., São Leopoldo – RS, 2005. Disponível na Internet: <http://www.addlabs.uff.br/enia_site/dw/mineracaodedados.pdf>. Acesso em: 21 abr. 2008.

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

[10] WITTEN, I. H.; FRANK E. **Data mining**: practical machine learning tools and techniques with Java implementations. Morgan Kauffmann Publishers: 1999 *apud* REZENDE, S. O., **Mineração de dados**. In: *Encontro Nacional de Inteligência Artificial*, 5., São Leopoldo – RS, 2005. Disponível na Internet: <http://www.addlabs.uff.br/enia_site/dw/mineracaodedados.pdf>. Acesso em: 21 abr. 2008.

[11] RAKOTOMALA, 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.

[12] 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.

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

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

[15] 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.