



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA E ESTATÍSTICA

**OTIMIZAÇÃO HIERÁRQUICA DO PACOTE
TestFraud PARA DETECÇÃO DE FRAUDE EM
TESTES**

Paulo Germano Sousa

Orientação: Prof. Dr. Héilton Ribeiro Tavares
Coorientação: Profa. Dra. Maria Regina Madruga Tavares

Belém
2020

Paulo Germano Sousa

**OTIMIZAÇÃO HIERÁRQUICA DO PACOTE
TestFraud PARA DETECÇÃO DE FRAUDE EM
TESTES**

Dissertação apresentada ao Curso de Mestrado em Matemática e Estatística da Universidade Federal do Pará, como pré-requisito para a obtenção do título de Mestre em Estatística.

Orientação: **Prof. Dr. Héilton Ribeiro Tavares**

Coorientação: **Profa. Dra. Maria Regina Madruga Tavares**

Belém

2020

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

S725o Sousa, Paulo Germano
Otimização hierárquica do pacote TestFraud para detecção de fraude em testes / Paulo Germano Sousa. — 2020.
67 f.

Orientador(a): Prof. Dr. Héilton Ribeiro Tavares
Coorientação: Prof^a. Dra. Maria Regina Madruga Tavares
Dissertação (Mestrado) - Programa de Pós-Graduação em Matemática e Estatística, Instituto de Ciências Exatas e Naturais, Universidade Federal do Pará, Belém, 2020.

1. Métodos para detecção de fraude em testes. 2. Avaliação em larga escala. 3. Método hierárquico. 4. Taxa de falso positivo. I. Título.

CDD 310

Paulo Germano Sousa

OTIMIZAÇÃO HIERÁRQUICA DO PACOTE *TestFraud* PARA
DETECÇÃO DE FRAUDE EM TESTES

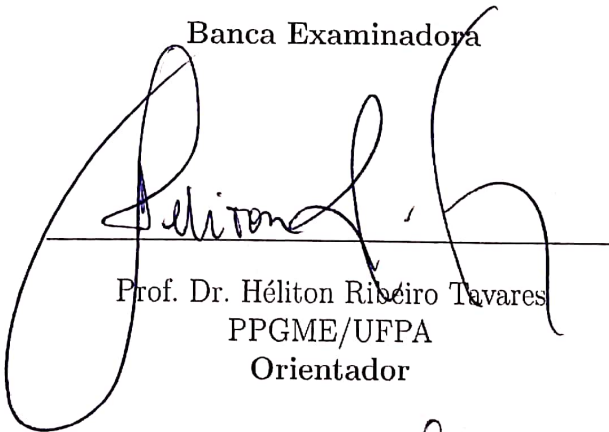
Esta Dissertação foi julgada e aprovada para a obtenção do grau de Mestre em Estatística, no Programa de Pós-Graduação em Matemática e Estatística da Universidade Federal do Pará.

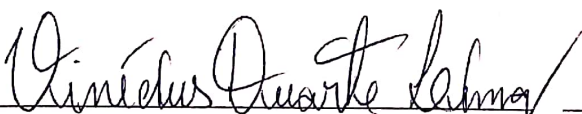
Belém, 14 de fevereiro de 2020

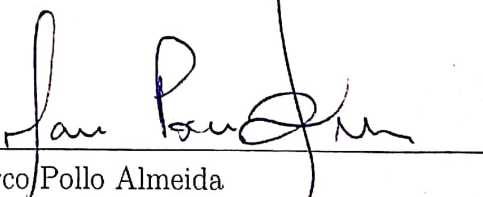
João Marcelo B Protázio

Prof. Dr. João Marcelo Brazão Protázio
(Coordenador do Programa de Pós-Graduação em Matemática e Estatística – UFPA)

Banca Examinadora


Prof. Dr. Héilton Ribeiro Tavares
PPGME/UFPA
Orientador


Prof. Dr. Vinícius Duarte Lima
FAEST/UFPA
Examinador Externo


Marco Pollo Almeida
EMATER-PA
Examinador Externo

Aos meus pais

Agradecimentos

Agradeço em primeiro lugar a Deus pela vida e pela oportunidade de sempre evoluir.

Aos meus pais, João Soares de Sousa e Jorgina Germano Sousa, pelo imenso amor e sacrifícios na minha formação acadêmica.

A minha irmã, Hilda Soares, pela parceria e apoio nos momentos difíceis. A minha sobrinha, Yasmin Rodrigues, pelo carinho de sempre.

Aos orientadores Prof. Dr. Héilton Tavares e Prof. Dr. Regina Tavares, que com toda sua paciência e dedicação orientaram-me nessa caminhada.

Ao Prof. Dr. Marcelo Protázio e os demais professores do PPGME, que tanto contribuíram para minha formação.

À UFPA, pelo incentivo, espaço, apoio e infraestrutura oferecida para o desenvolvimento deste projeto.

Aos meus amigos, Robinson Ortega, Aline Soares, Jessyca Soares, Jadiel Alves, Aline Klayse, Alexandre Lima, Marcondes Brito e Miguel Monteiro. Este último que tanto colaborou para desenvolvimento desse estudo.

*“Se temer que suspeitem ser sua narrativa inverídica,
lembre-se da probabilidade.”*

JOHN GAY

Resumo

Este estudo objetiva propor o método hierárquico no pacote *TestFraud* construído no ambiente R para identificar indícios de fraudes em testes. Esta área tem recebido grande importância teórica e em aplicações nos últimos anos, mas ainda carece de aprimoramentos. É comum nas avaliações em larga escala a presença de um grande número de examinados, o que dificulta a aplicação dos métodos de detecção em tais avaliações, pois eles se baseiam na comparação entre pares de respostas de indivíduos, acarretando em elevado tempo de processamento computacional na detecção de indivíduos que transgrediram o exame. Ainda, algumas avaliações envolvem etapas ou áreas diferentes, tal como o ENEM, que avalia quatro áreas do conhecimento. Na metodologia aqui proposta, os pares de indivíduos detectados na etapa k servirão de base de entrada na etapa $k + 1$. Nos estudos de simulação, o método hierárquico reduziu significativamente o tempo de execução dos índices. Além disso, foram realizadas inspeções dos métodos de detecção para o controle da taxa de falso positivo. Por fim, conclui-se com uma aplicação do método supracitado em dados reais do ENEM-2018 para a cidade de Teresina-PI.

PALAVRAS-CHAVE: Métodos para detecção de fraude em testes, Avaliação em larga escala, Método hierárquico, Taxa de falso positivo.

Abstract

This study aims to propose the hierarchical method in the *TestFraud* package built in the R environment to identify evidence of test fraud. This area has received great theoretical and application importance in recent years, but still needs improvement. The presence of a large number of evaluated is common in large-scale evaluations, which makes the detection methods difficult to apply in such evaluations, since they are based on the comparison between pairs of responses of individuals, resulting in high computational processing time to identify those who have committed fraud. Also, some assessments involve different steps or areas, such as ENEM, which assesses four areas of knowledge. In the methodology proposed here, the pairs of individuals detected in step k will serve as the input base in step $k + 1$. In simulation studies, the hierarchical method significantly reduced the execution time of the indices. Finally, it concludes with an application of the method mentioned above in real data from ENEM-2018 for the city of Teresina-PI.

KEYWORDS: Methods for detecting cheating on tests, Large scale assessment, Hierarchical Method, False Positive-Rate.

Sumário

Agradecimentos	vi
Resumo	viii
Abstract	ix
Lista de Tabelas	xii
Lista de Figuras	xiv
1 Introdução	1
1.1 Aspectos Gerais	1
1.2 Justificativa e importância da dissertação	2
1.3 Objetivos	3
1.3.1 Objetivo geral	3
1.3.2 Objetivos específicos	3
1.4 Organização da dissertação	3
2 Síntese dos principais métodos da área	5
2.1 Teoria da Resposta ao Item	5
2.1.1 Modelo Logístico de 3 parâmetros	5
2.1.2 Estimativa por Máxima Verossimilhança Marginal	7
2.1.3 Estimativa dos Parâmetros dos Itens	8
2.1.4 Estimativa das proficiências	10
2.1.5 Modelo de Resposta Nominal	10
2.2 Métodos de detecção	11
2.2.1 Índice ω	11
2.2.2 Teste da Binomial Generalizada (GBT)	12
2.2.3 Índice K	13
2.2.4 Índices K_1 e K_2	15
2.2.5 Índices S_1 e S_2	16
2.2.6 Pacote <i>TestFraud</i>	18
2.3 Testes de Hipóteses	21
2.3.1 Tipos de erros	21
2.3.2 Nível de confiança α	22
2.3.3 Taxa de falso positivo	22

3	Metodologia Proposta	24
3.1	Suporte computacional	24
3.2	Método Hierárquico	26
4	Resultados	29
4.1	Estudo de Simulação	30
4.1.1	Avaliação dos índices	30
4.1.2	Desempenho da Otimização Hierárquica	32
4.2	Aplicação em Dados Reais	35
4.2.1	Distribuição dos Escores	35
4.2.2	Distribuição das Proficiências	38
4.2.3	Detecção de Fraude	41
5	Considerações Finais	47
5.1	Trabalhos Futuros	48
	Referências Bibliográficas	49
	Apêndice A Algoritmo para análise da taxa de falso positivo	51

Lista de Tabelas

2.1	Medidas do tempo de execução em microssegundos da função <i>irtprob</i> usando 100 repetições	18
2.2	Medidas do tempo de execução em milissegundos da porção do código utilizada para computação dos índices K_1 , K_2 , S_1 e S_2 usando 1.000 repetições	19
2.3	Tipos de erros em um teste de hipóteses.	21
2.4	Probabilidade de não cometer erro Tipo I para T	23
4.1	Tempo de simulação computacional do processamento (em horas) dos índices no pacote <i>TestFraud</i> sem e com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=5\%$	33
4.2	Tempo de simulação computacional do processamento (em horas) dos índices no pacote <i>TestFraud</i> com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=5\%$	34
4.3	Tempo de simulação computacional do processamento (em horas) dos índices no pacote <i>TestFraud</i> com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=2\%$	34
4.4	Tempo de simulação computacional do processamento (em horas) dos índices no pacote <i>TestFraud</i> com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=1\%$	34
4.5	Tempo de simulação computacional do processamento (em horas) dos índices no pacote <i>TestFraud</i> com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=0,5\%$	35
4.6	Tempo de simulação computacional do processamento (em horas) dos índices no pacote <i>TestFraud</i> com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=0,1\%$	35
4.7	Tempo de processamento computacional (em horas) dos índices no pacote <i>TestFraud</i> sem e com o método hierárquico para 1.728.870 pares da prova do ENEM-2018 em Teresina-PI, $\alpha=5\%$	42

4.8	Distribuição dos 40 examinados, suspeitos de fraude por <i>cola</i> , com maior frequência nos pares finais do processo hierárquico. ENEM-2018 em Teresina-PI.	44
4.9	Descrição dos examinados, segundo a posição no banco de dados, suspeitos de fraude por <i>cola</i> que tiveram ligação com o indivíduo 8466 nos pares finais do processo hierárquico. ENEM-2018 em Teresina-PI.	45
4.10	Descrição dos examinados, segundo a posição no banco de dados, suspeitos de fraude por <i>cola</i> que tiveram ligação com o indivíduo 3301 nos pares finais do processo hierárquico. ENEM-2018 em Teresina-PI.	46

Lista de Figuras

2.1	Representação de uma Curva Característica do Item	6
2.2	Funções que calculam probabilidades baseado no MRN no pacote TestFraud e CopyDetect respectivamente	19
2.3	Porção do código que computa objetos para obtenção dos índices K_1 , K_2 , S_1 , S_2 no pacote Testfraud	20
2.4	Porção do código que computa objetos para obtenção dos índices K_1 , K_2 , S_1 , S_2 no pacote Copydetect	20
3.1	Ilustração de um processador com 4 núcleos	25
3.2	Fluxograma do método hierárquico.	27
3.3	Fluxograma do método hierárquico para o ENEM.	28
4.1	Taxas de falso positivo (erro tipo I) dos índices para resultados simulados de respostas nominais.	31
4.2	Valores de erro quadrático médio para os índices de resultados simulados de respostas nominais.	31
4.3	Taxas de falso positivo (erro tipo I) dos índices para resultados simulados de respostas nominais com escore mínimo de 30.	32
4.4	Histograma dos escores da prova de Linguagens, Códigos e suas Tecnologias, ENEM-2018, Teresina-PI.	36
4.5	Histograma dos escores da prova de Ciências Humanas e suas Tecnologias, ENEM-2018, Teresina-PI.	37
4.6	Histograma dos escores da prova de Ciências da Natureza e suas Tecnologias, ENEM-2018, Teresina-PI.	37
4.7	Histograma dos escores da prova de Matemática e suas Tecnologias, ENEM-2018, Teresina-PI.	38
4.8	Histograma das proficiências da prova de Linguagens, Códigos e suas Tecnologias, ENEM-2018, Teresina-PI.	39
4.9	Histograma das proficiências da prova de Ciências Humanas e suas Tecnologias, ENEM-2018, Teresina-PI.	40
4.10	Histograma das proficiências da prova de Ciências da Natureza e suas Tecnologias, ENEM-2018, Teresina-PI.	40
4.11	Histograma das proficiências da prova de Matemática e suas Tecnologias, ENEM-2018, Teresina-PI.	41
4.12	Fluxograma do método hierárquico para o ENEM-2018, Teresina-PI. . . .	43

Capítulo 1

Introdução

1.1 Aspectos Gerais

Em concursos de grande repercussão, seja para acesso as universidades ou a cargos públicos no Brasil, existe a necessidade de o certame ocorrer com lisura, assim como assegura as leis brasileiras, Código Penal, Art. 311-A [5]. Essa seriedade nos concursos pode ser ameaçada por tentativas de fraudes, uma das maneiras é através de *cola*. Esta consiste em obtenção de respostas de um outro candidato próximo ao examinado e por meio de comunicação eletrônica, sendo esta, altamente prejudicial ao exame devido ao grande número de examinados envolvidos, como copiadores das respostas e os indivíduos de alta proficiência, como fontes das respostas. Os métodos de detecção de transgressão são voltados para a fraude por *cola*, onde a análise consiste na comparação de respostas entre pares de examinados. Essa análise objetiva detectar similaridade incomum entre as respostas dos indivíduos, geralmente de alta proficiência. Logo, a aplicação desses métodos estatísticos é imprescindível em exames de larga escala para dar maior verossimilhança aos resultados obtidos.

Por outro lado, apesar da evolução desses métodos de detecção de fraude nos últimos anos [8], a aplicação em avaliações envolvendo um grande número de indivíduos é improvável devido ao demasiado tempo de processamento computacional. Isso ocorre devido a comparação de todas as combinações de respostas entre os examinados. Por exemplo, em um teste com j indivíduos participantes, todas as possíveis combinações de respostas será de $\frac{j(j-1)}{2}$ pares, que serão analisados. Assumindo $j = 1.000.000$ candidatos, ter-se-iam 499.999.500.000 pares a serem considerados para computação de similaridade. Esse quantitativo de pares de respostas não permite a utilização dos métodos estatísticos em tempo hábil. Com o objetivo de reduzir esse tempo de processamento, foi proposto por Souza (2019) o Pacote *TestFraud* em que as implementações de funções otimizadas e processamento em paralelo tornaram os cálculos de detecção menos lento. Assim, há a necessidade

de mais otimizações e implementações para poder torna os métodos estatísticos aplicáveis em grandes avaliações.

Nessas avaliações em larga escala, usam-se testes de proficiência e questionários sociodemográficos para identificar os fatores relacionados ao desempenho. Esses testes são elaborados com base em matrizes de referência, que indicam os conhecimentos avaliados para cada área de conhecimento. Cujas finalidades é descrever as competências e habilidades esperadas em cada nível de complexidade. Dessa forma, por meio de avaliações padronizadas [7] compara-se os resultados obtidos com os esperados. Com base nesses resultados, pode-se inferir sobre a qualidade do ensino de uma cidade, estado ou país, além de servir de subsídio para as políticas públicas relacionadas a educação. Nessa linha de pensamento, uma das principais avaliações em larga escala no Brasil é o Exame Nacional do Ensino Médio (ENEM), reformulado em 2009, destaca-se por ser utilizado como forma parcial ou integral de seleção de estudantes para as principais universidades públicas do país. Este exame, também, é utilizado como critério para seleção com objetivo de ingressar no ensino superior, tais como os programas: Financiamento Estudantil (FIES), Programa Universidade para Todos (Prouni) e Ciências Sem Fronteiras (CsF).

Assim, devido à grande importância das avaliações nacionais da educação, em particular o exame citado acima, é de suma relevância que os testes avaliativos sejam precisos e que o processo ocorra com extrema credibilidade na aplicação e nos resultados dos mesmos. Desta forma, terão estimativas confiáveis sobre as proficiências dos candidatos avaliados, além da evolução da qualidade do ensino. Esta estimativa pode ser viesada por transgressões nas provas aplicadas, como já dito anteriormente, a forma mais prejudicial é a fraude por *cola*. Dessa forma, os métodos estatísticos de detecção são imprescindíveis para identificação desses possíveis delitos. Em virtude disso, é fundamental que a verificação ocorra em tempo hábil, para que os indivíduos que infringiram sejam retirados da seleção sem comprometer o cronograma estabelecido. Portanto, as otimizações computacionais são imprescindíveis na computação da velocidade dos índices responsáveis por identificar os suspeitos de fraude.

1.2 Justificativa e importância da dissertação

Em avaliações educacionais em larga escala, por exemplo o ENEM, necessitam que o processo ocorra com integralidade devido ao seu grande impacto na sociedade, além

das inferências sobre a qualidade da educação básica brasileira. Dessa forma, os métodos estatísticos de detecção de fraude em testes são de grande importância, pois podem identificar ilícitos nos exames. Por outro lado, há a necessidade de otimizações computacionais que reduzam o tempo de cálculo desses métodos, afim de torná-los aplicáveis em tais avaliações.

1.3 Objetivos

1.3.1 Objetivo geral

Otimizar pelo método hierárquico o pacote *TestFraud* na linguagem R para a detecção de fraude em testes de larga escala.

1.3.2 Objetivos específicos

1. Descrever os métodos estatísticos de detecção de fraude por *cola* em testes que foram utilizados neste estudo;
2. Avaliar as taxas de falso positivo para cada índice aplicado;
3. Hierarquizar o teste de acordo com a ordem de aplicação de cada área de conhecimento para computação de similaridade;
4. Realizar estudos de simulação em relação ao tempo de processamento na computação dos índices segundo os níveis de significância estatística e tamanho de população;
5. Aplicar o pacote *TestFraud* otimizado na prova do ENEM do ano 2018 realizada em Teresina-PI.

1.4 Organização da dissertação

Este trabalho encontra-se dividido em 6 capítulos, a saber:

- Capítulo 1: realiza-se uma apresentação sobre a importância dos métodos estatísticos de detecção de fraude por *cola* em testes e sua relação com os recursos computacionais, além dos objetivos alcançados.

- Capítulo 2: tem-se uma breve descrição sobre a Teoria da Resposta ao Item (TRI), apresentação dos métodos estatísticos de detecção de fraude aplicados no presente trabalho e uma breve introdução a Teoria dos Testes de Hipóteses, que é necessária para utilização dos mesmos.
- Capítulo 3: explica-se a metodologia proposta neste estudo para otimização do tempo de computação dos métodos estatísticos de detecção de fraude;
- Capítulo 4: discute-se a aplicação do método hierárquico em dados simulados e reais;
- Capítulo 5: apresentam-se as considerações finais do estudo e proposta para trabalho futuro.

Capítulo 2

Síntese dos principais métodos da área

2.1 Teoria da Resposta ao Item

A proficiência de um examinado em determinada área de conhecimento poder ser medida por meio de duas abordagens, a da Teoria Clássica das Medidas (TCM) e a Teoria da Resposta ao Item (TRI). A característica principal da primeira é analisar e interpretar a prova com base no número de acertos (quantidade de itens considerados corretos). A segunda apresenta grandes vantagens sobre a TCM, duas delas é que essa permite a comparação entre populações que tenham alguns itens em comum e evolução dos resultados ao longo do tempo. Segundo Andrade, Tavares e Valler [1], uma das principais características da TRI é que ela tem como elementos centrais os itens.

A TRI baseia-se em um conjunto de modelos estatísticos que procuram representar a probabilidade de um indivíduo dar certa resposta a um item como função dos parâmetros deste e a da proficiência do examinado. Esta teoria possui a suposição de independência local, onde os itens são respondidos de forma independente por cada indivíduo de acordo com a sua habilidade [1].

2.1.1 Modelo Logístico de 3 parâmetros

Dentre os modelos propostos da TRI para análise de itens dicotomizados (considerados como certo ou errado), o mais utilizado na área de avaliações educacionais, em larga escala, é o modelo de 3 parâmetros (ML3), inclusive é o modelo utilizado no ENEM para estimar as proficiências dos examinados nas quatro áreas de conhecimentos. O ML3 é dado por:

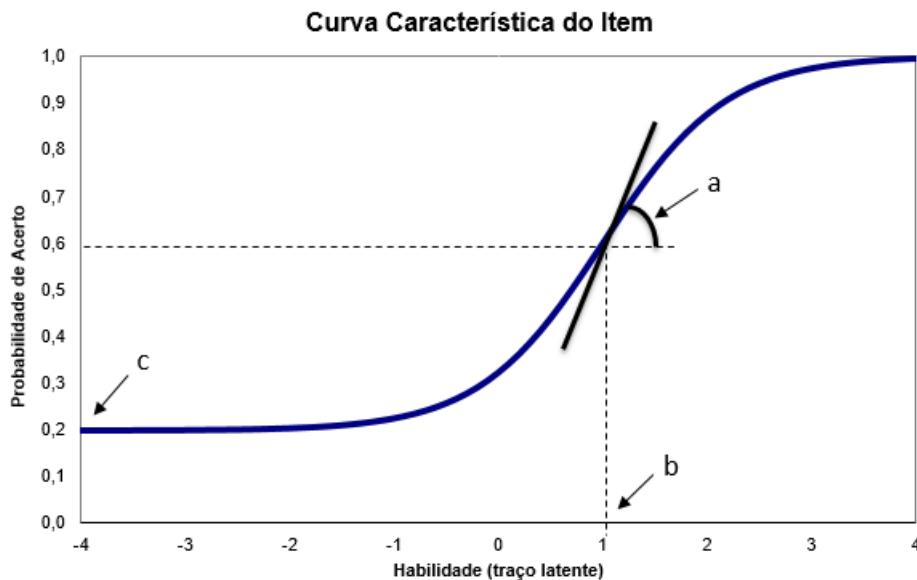
$$P(U_{ij} = 1|\theta_j) = c_i + (1 - c_i) \frac{1}{1 + e^{-Da_i(\theta_j - b_i)}}, \quad (2.1)$$

com $i = 1, 2, \dots, I$, e $j = 1, 2, \dots, n$, em que:

- $P(U_{ij} = 1|\theta_j)$ é a probabilidade do indivíduo j com traço latente θ_j acertar o item i ;
- b_i é o parâmetro de dificuldade (ou de posição) do item i , medido na mesma escala de θ_j ;
- a_i é o parâmetro de discriminação (ou inclinação) do item i , com valor proporcional à inclinação da Curva Característica do Item no ponto b_i ;
- c_i é o parâmetro de acerto casual do item i ;
- D é um fator de escala, constante e igual a 1. Utiliza-se o valor 1,702 quando desejar-se que a função logística forneça resultados semelhantes ao da função ogiva normal.

A representação gráfica (Figura 2.1) da associação existente entre os parâmetros do modelo (a_i , b_i e c_i) e a Função de Resposta do Item ($P(U_{ij} = 1|\theta_j)$) é denominada de Curva Característica do Item (CCI).

Figura 2.1 *Representação de uma Curva Característica do Item*



Fonte: Elaborado pelos Autores.

A Curva Característica do Item indica a probabilidade de resposta correta a um item em função de um nível de habilidade do respondente. A habilidade θ e o parâmetro de dificuldade b_i estão medidos na mesma escala, a inclinação na curva informa a capacidade de

discriminação do item (parâmetro a_i) e o parâmetro de acerto casual c_i informa a probabilidade de um indivíduo com baixa proficiência acertar o item, por ser uma probabilidade seus valores estão entre 0 e 1.

Os demais modelos dicotomizados são casos particulares do ML3. Para o modelo logístico de 1 parâmetro (modelo Rasch) faz-se $c_i = 0$ e $a_i = 1$ e para o modelo de 2 parâmetros, tem-se $c_i = 0$.

Nesses modelos, a estimativa dos parâmetros (a_i, b_i, c_i) dos itens e da habilidade (θ_j) é feita via Máxima Verossimilhança Marginal [1].

2.1.2 Estimação por Máxima Verossimilhança Marginal

A estimação das proficiências dos indivíduos e dos parâmetros dos itens são etapas fundamentais da Teoria da Resposta ao Item. Ao aplicar esta teoria pode-se encontrar três situações:

- (i) parâmetros dos itens conhecidos e habilidades desconhecidas;
- (ii) habilidades dos indivíduos conhecidas e os parâmetros dos itens desconhecidos;
- (iii) as habilidades desconhecidas e parâmetros dos itens também desconhecidos.

Das três situações citadas, a mais comum é a (iii), por isso esta seção irá abordar a metodologia para estimar (tornar conhecidos) simultaneamente as habilidades e os parâmetros dos itens. Dentre os métodos, destaca-se a estimação por Máxima Verossimilhança Marginal (MVM). Antes da introdução ao método da MVM, algumas notações e suposições são necessárias para o desenvolvimento do modelo [1]. Considera-se as seguintes notações: seja θ_j a habilidade e U_{ji} a variável aleatória que representa a resposta do indivíduo j ao item i , com

$$U_{ji} = \begin{cases} 1, & \text{resposta correta} \\ 0, & \text{resposta incorreta} \end{cases}$$

ainda,

- n : o número total de examinados na amostra;
- $\mathbf{U}_j = (U_{j1}, \dots, U_{jI})$: o vetor aleatório de respostas do examinado j ;

- $\mathbf{U}_{..} = (U_{1.}, U_{2.}, \dots, U_{n.})$: o conjunto integral das respostas;
- u_{ji} , \mathbf{u}_j e $\mathbf{u}_{..}$: as respostas observadas.
- $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)$: o vetor de habilidades dos n indivíduos;
- $\boldsymbol{\zeta} = (\zeta_1, \dots, \zeta_n)$: o conjunto de parâmetros dos itens.

Para a utilização da TRI, são necessárias duas principais suposições, são elas:

- as respostas oriundas de indivíduos diferentes são independentes;
- os itens são respondidos de forma independente por cada indivíduo (Independência Local), fixada sua habilidade.

Em relação ao método da Máxima Verossimilhança Marginal proposto por Bock e Aitkin [2], os autores indicam dois estágios presentes no método:

- Estágio 1: realização da estimação dos parâmetros dos itens;
- Estágio 2: realização da estimação dos traços latentes (habilidades).

O MVM necessita inicialmente de suposições adicionais, a princípio considera-se uma distribuição de probabilidade para o traço latente, geralmente associa-se as habilidades (θ_j) uma variável aleatória com distribuição contínua e função densidade de probabilidade $g(\theta | \boldsymbol{\eta})$. De modo geral, é usual supor que θ segue uma distribuição normal com média zero e desvio-padrão igual a um.

2.1.3 Estimação dos Parâmetros dos Itens

Com as definições descritas anteriormente, tem-se que a probabilidade marginal de \mathbf{U}_j é dada por

$$P(\mathbf{u}_j | \boldsymbol{\zeta}, \boldsymbol{\eta}) = \int_{\mathbb{R}} P(\mathbf{u}_j | \theta, \boldsymbol{\zeta}, \boldsymbol{\eta}) g(\theta | \boldsymbol{\eta}) d\theta = \int_{\mathbb{R}} P(\mathbf{u}_j | \theta, \boldsymbol{\zeta}) g(\theta | \boldsymbol{\eta}) d\theta,$$

Usando a independência entre as respostas de diferentes indivíduos (suposição da TRI), pode-se escrever a probabilidade associada ao vetor de respostas $\mathbf{U}_{..}$ como

$$P(\mathbf{u}_{..} | \boldsymbol{\zeta}, \boldsymbol{\eta}) = \prod_{j=1}^n P(\mathbf{u}_j | \boldsymbol{\zeta}, \boldsymbol{\eta}) \quad (2.2)$$

Embora a verossimilhança poder ser escrita conforme a expressão (2.2), a abordagem de *Padrões de Respostas* é frequentemente utilizada [1]. Dado que um teste possui I itens no total, com 2 possíveis respostas para cada item (0 ou 1), há portanto $S = 2^I$ padrões de respostas. Sendo assim, quando o número de examinados é grande em relação ao número de itens em um teste, pode haver vantagens computacionais em trabalhar com a frequência de ocorrências dos diferentes padrões de resposta. Neste sentido, será considerado este raciocínio. Agora, o índice j não representará um indivíduo, mas sim um padrão de resposta.

Seja r_j o número de ocorrências distintas do padrão de resposta j , e ainda $s \leq \min(n, S)$ o número de padrões de resposta com $r_j > 0$. Segue que

$$\sum_{j=1}^s r_j = n. \quad (2.3)$$

Pela suposição da independência entre as respostas de diferentes indivíduos, tem-se que os dados seguem uma distribuição *Multinomial*, conforme a expressão abaixo:

$$L(\boldsymbol{\zeta}, \boldsymbol{\eta}) = \frac{n!}{\prod_{j=1}^s r_j!} \prod_{j=1}^s P(\mathbf{u}_j | \boldsymbol{\zeta}, \boldsymbol{\eta})^{r_j}, \quad (2.4)$$

segue a log-verossimilhança como

$$L(\boldsymbol{\zeta}, \boldsymbol{\eta}) = \log \left\{ \frac{n!}{\prod_{j=1}^s r_j!} \right\} + \sum_{j=1}^s r_j \log P(\mathbf{u}_j | \boldsymbol{\zeta}, \boldsymbol{\eta}). \quad (2.5)$$

As equações de estimação para os parâmetros dos itens são obtidas por

$$\frac{\partial \log L(\boldsymbol{\zeta}, \boldsymbol{\eta})}{\partial \zeta_i} = 0, \quad i = 1, \dots, I. \quad (2.6)$$

Segundo os desenvolvimentos descritos em Andrade et al. [1], chega-se as seguintes equações de estimação:

$$a_i : D(1 - c_i) \sum_{j=1}^s r_j \int_{\mathfrak{R}} [(u_{ji} - P_i)(\theta - b_i)] W_i g_j^*(\theta) d\theta = 0, \quad (2.7)$$

$$b_i : -Da_i(1 - c_i) \sum_{j=1}^s r_j \int_{\mathfrak{R}} [(u_{ji} - P_i)] W_i g_j^*(\theta) d\theta = 0, \quad (2.8)$$

$$c_i : \sum_{j=1}^s r_j \int_{\mathfrak{R}} \left[(u_{ji} - P_i) \frac{W_i}{P_i^*} \right] g_j^*(\theta) d\theta = 0, \quad (2.9)$$

onde,

$$g_j^*(\theta) = g(\theta|\mathbf{u}_j, \boldsymbol{\zeta}, \boldsymbol{\eta}) = \frac{P(\mathbf{u}_j|\theta, \boldsymbol{\zeta}) g(\theta|\boldsymbol{\eta})}{P(\mathbf{u}_j|\boldsymbol{\zeta}, \boldsymbol{\eta})}. \quad (2.10)$$

A expressão (2.10) representa a função densidade de probabilidade condicional da habilidade da população. As equações de estimação (2.7), (2.8) e (2.9) não possuem solução explícita, sendo assim necessário a utilização de algum método numérico, por exemplo o algoritmo de *Newton-Raphson*. Também tem sido muito frequente na TRI aplicar o método *Hemite-Gauss*, conhecido como *método de quadratura gaussiana*.

2.1.4 Estimação das proficiências

Dentre os métodos de estimação das proficiências destaca-se a estimação de θ_j pela média da posteriori $g_j^*(\theta)$ (ou EAP: Expected a Posteriori), um método Bayesiano que consiste em obter a esperança da posteriori, sendo esta dada por:

$$\hat{\theta}_j \equiv E(\theta|\mathbf{u}_j, \boldsymbol{\zeta}, \boldsymbol{\eta}) = \frac{\int_{\mathbb{R}} \theta g(\theta|\boldsymbol{\eta}) P(\mathbf{u}_j|\theta, \boldsymbol{\zeta}) d\theta}{\int_{\mathbb{R}} g(\theta|\boldsymbol{\eta}) P(\mathbf{u}_j|\theta, \boldsymbol{\zeta}) d\theta}. \quad (2.11)$$

Este método de estimação da habilidade tem a vantagem de não precisar de nenhum método iterativo para a solução, pois pode ser calculada diretamente. Alguns autores (Mislevy e Stocking [12]) recomendam esta escolha para a estimação das proficiências.

2.1.5 Modelo de Resposta Nominal

O Modelo de Resposta Nominal (MRN) proposto por Bock [3] foi desenvolvido com o objetivo de dar maior precisão para as estimativas de proficiências (θ_j), pois, usa toda a informação contida nas respostas dos examinados. Dessa forma, leva-se em conta a probabilidade de um avaliado j selecionar uma particular alternativa v , dentre V_i opções possíveis, do item i . O MRN é definido por:

$$P_{iv}(\theta_j) = \frac{e^{(\zeta_{iv} + \lambda_{iv}\theta_j)}}{\sum_{v=1}^{V_i} e^{(\zeta_{iv} + \lambda_{iv}\theta_j)}}, \quad (2.12)$$

com $i = 1, 2, \dots, I$, $j = 1, 2, \dots, n$, e $v = 1, 2, \dots, V_i$. Em cada θ_j , a soma das probabilidades sobre as V_i opções, $\sum_{v=1}^{V_i} P_{iv}(\theta_j)$ é 1. As quantidades ζ_{iv} e λ_{iv} são parâmetros denominados, respectivamente, de intercepto e inclinação do item para alternativa v do item i . Além disso, a estimação dos parâmetros dos itens e as habilidades θ_j podem ser estimados pelos métodos de máxima verossimilhança.

2.2 Métodos de detecção

2.2.1 Índice ω

Com o intuito de detectar cópias em testes, o índice ω analisa todas as respostas idênticas, isso implica que verifica as similaridades entre respostas corretas e incorretas entre dois candidatos, chamados de fonte (s) e copiador (c). Assim, Wollack [19] considerou h_{cs} como o número de itens respondidos de forma igual entre os indivíduos c e s em um teste de múltipla escolha com opções $v = 1, \dots, V_i$. Portanto, condiciona-se às respostas de s , para se definir h_{cs} como

$$h_{cs} = \sum_{i=1}^I 1[u_{ic} = u_{is}], \quad (2.13)$$

para $i = 1, 2, \dots, I$, representando o i -ésimo item, u_{ic} e u_{is} são as opções do item i escolhidas pelos examinados c e s , respectivamente, e

$$1[u_{ic} = u_{is}] = \begin{cases} 1, & \text{se } c \text{ e } s \text{ selecionaram a mesma alternativa } v_i, \\ 0, & \text{c. c.} \end{cases} \quad (2.14)$$

A distribuição do número de itens respondidos de forma idêntica no item i entre os examinados c e s , ou seja, h_{cs} , é obtida calculando-se a probabilidade de c selecionar as respostas providas por s dado sua habilidade (θ_c), o vetor de respostas do examinado s (U_s) e a matriz de parâmetros dos itens (ξ). Assim, o valor esperado dessa distribuição é

$$\begin{aligned} E(h_{cs}|\theta_c, U_s, \xi) &= E \left[\sum_{i=1}^I 1(u_{ic} = u_{is}|\theta_c, U_s, \xi) \right] \\ &= \sum_{i=1}^I E [1(u_{ic} = u_{is}|\theta_c, U_s, \xi)] \\ &= \sum_{i=1}^I [P(u_{ic} = u_{is}|\theta_c, U_s, \xi)], \end{aligned} \quad (2.15)$$

considerando que as respostas dos indivíduos aos itens são localmente independentes e a partir das Equações (2.14) e (2.15), condicionando U_s e os parâmetros dos itens, h_{cs} é a soma de variáveis Bernoulli independentes cada uma com probabilidade, na respectiva, de sucesso, isto é, com média igual a

$$P(u_{ic} = u_{is}|\theta_c, U_s, \xi), \quad (2.16)$$

e portanto, para obter $P(u_{ic} = u_{is} | \theta_c, U_s, \xi)$ neste trabalho usa-se o MRN, descrito na Seção 2.1.5.

Em virtude do Teorema Central do Limite (TCL), ω tem distribuição assintoticamente normal padrão, assim expressa

$$\omega = \frac{h_{cs} - E(h_{cs} | \theta_c, U_s, \xi)}{\sigma_{h_{cs}}}, \quad (2.17)$$

onde o desvio-padrão de h_{cs} é dado por

$$\sigma_{h_{cs}} = \sqrt{\sum_{i=1}^I [P(u_{ic} = u_{is} | \theta_c, U_s, \xi)][1 - P(u_{ic} = u_{is} | \theta_c, U_s, \xi)]}. \quad (2.18)$$

É possível obter evidências que o indivíduo c cometeu fraude a partir da comparação do valor observado de ω com o valor crítico (tabelado) para o nível de significância (α) adotado. Segundo Sotaridona [15] e Wollack [19] quanto maior o valor de ω mais forte é a evidência de que c copiou de s .

2.2.2 Teste da Binomial Generalizada (GBT)

O índice *GBT* ou Teste da Binomial Generalizada (Van de Linden & Sotaridona [17]) analisa o número de respostas coincidentes entre dois indivíduos. Sendo P_{M_i} a probabilidade das respostas dos examinados de c e s ao item i coincidirem, essa probabilidade é expressa por

$$P_{M_i} = \sum_{v=1}^{V_i} P_{civ} \cdot P_{siv}, \quad (2.19)$$

onde P_{civ} e P_{siv} são, respectivamente, as probabilidades dos indivíduos c e s responderem a mesma alternativa do item i . Usa-se um modelo de resposta para calcular as probabilidades, em geral o MRN.

Com base em (P_{M_i}), tem-se que a probabilidade de ocorrência de exatamente n respostas iguais em I itens é igual a

$$f_I(n) = \sum \left(\prod_{i=1}^I P_{M_i}^{u_i} (1 - P_{M_i})^{1-u_i} \right), \quad (2.20)$$

sendo

$$u_i = \begin{cases} 1, & \text{se } c \text{ e } s \text{ respondem identicamente ao item } i, \\ 0, & \text{c.c.} \end{cases} \quad (2.21)$$

e

\sum : todas as possibilidades de combinações de n respostas coincidentes em I itens.

Portanto, a partir do número de respostas iguais, incorretas (w_{cs}) e corretas (R_{cs}), pode-se calcular o índice GBT como a cauda superior da distribuição binomial composta, assim definido

$$\sum_{n=w_{cs}+R_{cs}}^I f_I(n). \quad (2.22)$$

Por fim, é avaliado se o valor obtido em (2.22) é menor que o nível de significância α preestabelecido para detectar suspeita de fraude [21].

2.2.3 Índice K

Baseando-se apenas nas coincidências de respostas incorretas (entre um par de examinados) foi proposto o índice K , Holland (1996) [9]. Na construção desse índice seguiu-se a nomenclatura dos anteriores, definindo c e s como fonte e copião das respostas, respectivamente. Além, das seguintes notações pertinentes:

- j , com ($j = 1, \dots, J$), denotando os examinados;
- i , com ($i = 1, \dots, I$), denotando os itens;
- v , com ($v = 1, \dots, V_i$), denotando as alternativas de um item;
- w_j sendo o número de respostas “erradas” do examinado j ;
- r , com $r = 1, \dots, c', \dots, R$, denotando os subgrupos de examinados, sendo que cada subgrupo tem um número distinto de respostas incorretas, R é o número total de subgrupos ($R = I + 1$, salvo se houver algum subgrupo vazio), além disso, cada subgrupo possui no mínimo um examinado e que $\sum_{r=1}^R n_r = J - 1$, denota-se aqui c' como o subgrupo ao qual o examinado c pertence e n_r é o número total de examinados de cada subgrupo r ;
- j' , com $j' = 1, \dots, n_r$, denotando os examinados dentro de um subgrupo r específico.
- $\mathbf{M}_r = (M_{r1}, \dots, M_{rj'}, \dots, M_{rn_r})$ sendo um vetor dos números de respostas incorretas idênticas às da fonte em um particular subgrupo r ;

- $\mathbf{M}_{c'} = (M_{c'1}, \dots, M_{c'n_r})$ denotando o vetor do número de respostas incorretas idênticas às da fonte de $n_{c'}$ examinados do subgrupo c' , sendo este o subgrupo que possui o mesmo número de respostas incorretas do copiador.
- $m_{rj'}$ sendo o valor observado do número de respostas incorretas idênticas entre o examinado rj' e s ;
- $Q_r = \frac{w_r}{I}$ como a proporção de respostas incorretas de um subgrupo r , sendo w_r o número de respostas erradas do subgrupo r e I é o número total de itens do teste.

O índice K possui duas formulações para ser obtido, a primeira utilizando uma distribuição amostral empírica e a segunda através de uma distribuição teórica.

A construção do índice K de forma empírica utiliza os dados empíricos de J examinados respondendo a I itens. Para essa construção tem-se que:

- definir o grupo de examinados com o mesmo número de respostas incorretas de c (subgrupo c');
- definir para cada examinado do subgrupo c' , definir o número de itens incorretos idênticos ao examinado s , obtendo-se assim o vetor $\mathbf{M}_{c'}$.

Com base nessas definições, calcula-se o índice K como a proporção de examinados com o mesmo número de respostas incorretas do copiador e cujo número de respostas incorretas correspondentes com as da fonte ($m_{c'j'}$) é maior ou igual ao número de respostas erradas iguais entre c e s ($m_{c'c}$). Assim, esse índice é dado por

$$K = \frac{\sum_{j'=1}^{n_{c'}} I_{c'j'}}{n_{c'}}, \quad (2.23)$$

onde

$$I_{c'j'} = \begin{cases} 1, & \text{se } m_{c'j'} \geq m_{c'c}, \\ 0, & \text{c.c.} \end{cases}, \quad (2.24)$$

Dessa forma, quanto menor o valor de K maior será a evidência que examinado c copiou do indivíduo s . A qualidade dessa evidência é dependente do tamanho do subgrupo particular de c , pois para um número de pequeno de examinados nesse subgrupo o valor obtido de K não é preciso [14].

Entretanto, uma alternativa para contornar a imprecisão em subgrupos pequenos, proposta por Holland (1996) [9], é obter o índice a partir de uma distribuição teórica do

número de respostas incorretas iguais entre c' (indivíduo qualquer do subgrupo de c) e s , sendo esta variável aleatória denominada por M com distribuição binomial, assim denotada

$$M \stackrel{aprox.}{\sim} Bin(w_s, p), \quad (2.25)$$

onde w_s é o número de respostas incorretas de s e p é a probabilidade esperada de M .

Então, a probabilidade do número de respostas incorretas idênticas iguais as da fonte (s), pelo avaliador c' , ser maior que $m_{c'c}$ é dado por

$$K^* = P(M \geq m_{c'c}) = \sum_{w=m_{c'c}}^{w_s} \binom{w_s}{w} (p_{c'}^*)^w (1 - p_{c'}^*)^{w_s - w}. \quad (2.26)$$

Nessa forma de cálculo do índice K^* é necessário estimar o parâmetro p do modelo probabilístico. Segundo Holland (1996) [9], a estimativa é denotada por $p_{c'}^*$ e obtida por

$$p_{c'}^* = \frac{\bar{m}_{c'}}{w_s}, \quad (2.27)$$

sendo

$$\bar{m}_{c'} = \frac{\sum_{j'=1}^{n_{c'}} m_{c'j'}}{n_{c'}}. \quad (2.28)$$

Outra forma de estimar p , segundo Holland (1996) [9], é através do método de regressão linear, onde é utilizado a proporção de respostas incorretas (Q_r) de cada subgrupo com a variável explicativa. Demonstrou-se empiricamente que p_r^* é linearmente relacionado a Q_r , sendo p_r^* definido de modo análogo em 2.27. Seja \hat{p}_r a estimativa de p_r^* usando Q_r . A expressão para \hat{p}_r utilizando regressão linear é:

$$\hat{p}_r = \begin{cases} a + bQ_r, & \text{se } 0 < Q_r \leq 0.3; \\ [a + 0.3b] + 0.4b[Q_r - 0.3], & \text{se } 0.3 < Q_r \leq 1. \end{cases} \quad (2.29)$$

Para os autores Sotaridona & Meijer (20002) [14] os valores a e b devem ser definidos para o modelo de regressão de duas partes, sendo estas condicionadas ao valor Q_r . Holland (1996) usou $a = 0,085$ e diferentes valores para b baseado na configuração do teste específico utilizado.

2.2.4 Índices K_1 e K_2

Uma nova proposta foi apresentado por Sotaridona & Meijer (2002) [14] onde o objetivo é estimar p_r^* através de \hat{p}_1^* e \hat{p}_2^* , sendo estes baseados, respectivamente, a partir

de uma regressão linear e uma quadrática utilizando Q_r como variável explicativa. As estimativas de p_r^* , são duas versões do índice K , chamados de K_1 e K_2 , e são definidas conforme a seguir

$$K_1 = P(M \geq m_{c'c}) = \sum_{w=m_{c'c}}^{w_s} \binom{w_s}{w} (\hat{p}_1^*)^w (1 - \hat{p}_1^*)^{w_s-w} \quad (2.30)$$

e

$$K_2 = P(M \geq m_{c'c}) = \sum_{w=m_{c'c}}^{w_s} \binom{w_s}{w} (\hat{p}_2^*)^w (1 - \hat{p}_2^*)^{w_s-w}. \quad (2.31)$$

É de grande importância destacar que \hat{p}_1^* e \hat{p}_2^* utilizam os dados de todos os R subgrupos para estimar p , o que difere de $p_{c'}^*$ que usa apenas as informações do subgrupo c' para estimar p . Esses mesmos autores mostraram que \hat{p}_2^* gerou melhores estimativas para p do que \hat{p}_1^* e $p_{c'}^*$.

2.2.5 Índices S_1 e S_2

Sotaridona & Meijer (2003) [15] propuseram o índice S_1 , o qual é similar aos índices K_1 e K_2 , pois é baseado no número de respostas incorretas iguais entre os examinados c' e s , que neste estudo essa variável aleatória é denominada por M . A distinção de S_1 é que essa variável aleatória segue uma distribuição de Poisson, enquanto K_1 e K_2 atribuem uma distribuição binomial para M .

Por outro lado, situação semelhante ocorre para estimação do parâmetro desconhecido da distribuição. Neste índice, a esperança do modelo de probabilidade Poisson ou média de $M(\mu)$ é estimado a partir de um modelo log-linear, dado por

$$S_1 = P(M \geq m_{c'c}) = \sum_{w=m_{c'c}}^{w_s} \frac{e^{-\hat{\mu}_{c'}} \hat{\mu}_{c'}^w}{w!}, \quad (2.32)$$

onde $\hat{\mu}_{c'}$ é a estimativa para μ usando o modelo log-linear, sendo este dado por:

$$\log(\mu_r) = \beta_0 + \beta_1 w_r, \quad \forall r, \quad (2.33)$$

em que β_0 e β_1 são parâmetros do modelo, μ_r é o valor esperado da variável Poisson $M_{r,j'}$ e w_r é o número de respostas incorretas do subgrupo r . Em virtude desse modelo tem-se que $\hat{\mu}_{c'}$ é dado por

$$\hat{\mu}_{c'} = e^{\beta_0 + \beta_1 w_{c'}}. \quad (2.34)$$

No artigo supracitado, foi desenvolvido o índice S_2 . Em comparação aos índices K , K_1 ,

K_2 e S_1 , esse índice é mais informativo, pois considera tanto as respostas incorretas quanto corretas em seu cálculo. Assim, considera-se $M_{rj'}^*$ como a soma entre o número de respostas coincidentes incorretas e o número de respostas coincidentes corretas ponderadas, ambas entre os examinados s e rj' pertencente a um subgrupo r específico. A expressão $M_{rj'}^*$ é dada por

$$M_{rj'}^* = M_{rj'} + \sum_{i^*} \delta_{i^*rj'}, \quad (2.35)$$

sendo $\delta_{i^*rj'}$ a estimativa da informação de cópia do item i^* pelo examinado rj' , e i^* representado os itens respondidos corretamente pela fonte. O termo $\delta_{i^*rj'}$ é definida por:

$$\delta_{i^*rj'} = f(P_{i^*rj'}) = d_1 e^{d_2 P_{i^*rj'}}, \quad (2.36)$$

em que $0 \leq \delta_{i^*rj'} \leq 1$. Além, $P_{i^*rj'}$ a probabilidade do examinado rj' responder corretamente ao item i^* . Logo, pelo método da máxima verossimilhança $P_{i^*rj'}$ é estimado por

$$\hat{P}_{i^*rj'} = \frac{\sum_{j'=1}^{n_r} I_{(u_{i^*rj'}=u_{i^*s})}}{n_r}, \quad (2.37)$$

sendo

$$I_{(u_{i^*rj'}=u_{i^*s})} = \begin{cases} 1, & \text{se } j' \text{ responder corretamente ao item } i^*, \\ 0, & \text{c.c.} \end{cases} \quad (2.38)$$

Os valores d_2 e d_1 são dados por

$$d_2 = - \left(\frac{1+g}{g} \right), \quad (2.39)$$

$$d_1 = - \left(\frac{1+g}{1-g} \right)^{d_2 P_{i^*c}}, \quad (2.40)$$

sendo g a probabilidade de individuo que desconhece o item acertá-lo ao acaso, ou seja, se um item é composto por V alternativas então $g = 1/V$ [15].

Observa-se que $M_{rj'}^*$ é um caso particular de $M_{rj'}$ quando não há respostas corretas coincidentes entre rj' e s , pois o segundo termo da Equação (2.35) zera. Por outro lado, quando não há respostas incorretas coincidentes entre rj' e s o primeiro termo da Equação (2.35) zera e $M_{rj'}^* = \sum_{i^*} \delta_{i^*rj'}$, tornando-se uma variável sensível para todo conjunto de respostas. Em aplicações o valor de $M_{rj'}^*$ é tratado como um número inteiro [15]. Então, S_2 é determinado a partir de

$$S_2 = P(M^* \geq m_{c'c}^*) = \sum_{w=m_{c'c}^*}^I \frac{e^{-\hat{\mu}_{c'}} \hat{\mu}_{c'}^w}{w!}, \quad (2.41)$$

sendo $m_{c'c}^*$ o número observado de coincidências incorretas e corretas ponderada entre os indivíduos c e s e M^* a variável aleatória sobre a distribuição de Poisson. Assim como é feito para o índice S_1 , usa-se o modelo log-linear para estimar média de M^* . Logo, pequenos valores de S_2 indicam que a cópia ocorreu [15].

2.2.6 Pacote *TestFraud*

Na implementação do pacote *TestFraud* os autores [16] procuraram corrigir os códigos fonte de maior tempo de processamento no pacote *CopyDetect*. As principais mudanças em relação a este pacote foram:

1. Diminuição de laços de repetições (*for*);
2. Diminuição de condições (*if... else...*);
3. Otimização e predefinição na computação de objetos;
4. Agrupamento nos cálculos dos índices variantes (K, K_1, K_2, S_1, S_2) e dos índices ω e *GBT*;
5. Processamento em paralelo.

Essa diminuição de laço de repetição pode ser visualizada na Figura 2.2, onde a implementação da função que calcula as probabilidades do MRN estão nas linhas de 1 a 6 (*TestFraud*) e nas linhas de 9 a 20 (*CopyDetect*). Comparando a função nos dois pacotes para 100 repetições, Tabela 2.1, a média do tempo de computação é menor no *TestFraud*. Em relação ao agrupamento nos cálculos dos índices (K, K_1, K_2, S_1, S_2), Figura 2.3, no

Tabela 2.1 *Medidas do tempo de execução em microssegundos da função irtprob usando 100 repetições*

Pacote	Mín	Q_1	Média	Mediana	Q_3	Máx
<i>TestFraud</i>	36,1	38,6	50,3	40,1	41,8	7.423,9
<i>CopyDetect</i>	1.010,7	1.027,4	1.258,0	1.041,9	1.067,4	148.372,0

Fonte: Souza (2019) [16].

Figura 2.2 Funções que calculam probabilidades baseado no MRN no pacote *TestFraud* e *CopyDetect* respectivamente

```

1 irtprob_TestFraud <- function(ability, item.param) {
2   r <- ncol(item.param)/2
3   ps <- exp((item.param[ ,1:r]*ability)+item.param[ ,1:r+r])
4   prob <- ps/rowSums(ps)
5   prob
6 }
7
8
9 irtprob_CopyDetect <- function(ability, item.param) {
10  prob <- matrix(nrow = nrow(item.param), ncol = ncol(item.param)/2)
11  for (i in 1:nrow(prob)) {
12    ps <- c()
13    for (j in 1:ncol(prob)) {
14      ps[j] = exp((item.param[i, j] * ability) +
15                item.param[i, j + ncol(prob)])
16    }
17    prob[i, ] = ps/sum(ps)
18  }
19  prob
20 }

```

Fonte: Souza (2019) [16].

pacote *TestFraud* em comparação com o *CopyDetect*, Figura 2.4, obteve-se menor média do tempo de processamento nesse pacote, conforme Tabela 2.2 para 1.000 repetições. Segundo Souza (2019) [16], a melhoria no desempenho se deve muito a retirada de transformações nos objetos *smatrix1* (Figura 2.4, linha 12) e *smatrix2* (Figura 2.4, linha 15) utilizando o comando *as.data.frame*, sendo estas transformações não necessárias para a computação dos índices. Este autor ainda cita como outro fator importante, a retirada de condições (Figura 2.4, linhas 10 e 27), sendo estas substituídas no *TestFraud* por um objeto denominado *pos* (Figura 2.3, linha 3) que identifica as posições que devem ser utilizadas no laço *for*, além da predefinição dos objetos *pr* e *pj* como um vetor de *NA*'s (Figura 2.3, linha 4).

Tabela 2.2 Medidas do tempo de execução em milissegundos da porção do código utilizada para computação dos índices K_1 , K_2 , S_1 e S_2 usando 1.000 repetições

Pacote	Mín	Q_1	Média	Mediana	Q_3	Máx
<i>TestFraud</i>	158,1	161,1	187,4	165,4	174,0	1.107,7
<i>CopyDetect</i>	360,8	374,5	437,1	387,0	529,1	1.323,6

Fonte: Souza (2019) [16].

Figura 2.3 Porção do código que computa objetos para obtenção dos índices K_1 , K_2 , S_1 , S_2 no pacote *Testfraud*

```

1 ws <- sum(form2[pa[2], ] == 0, na.rm = TRUE)
2 incorrect.items <- which(form2[pa[2], ] == 0)
3 pos <- which(lengths!=0)
4 pr <- pj <- rep(NA, (I+1))
5 prob <- weight <- matrix(nrow=(I+1), ncol=I)
6 g <- 1/length(options)
7 d2 <- -(1+g)/g
8 p1 <- ((1+g)/(1-g))*exp(1)
9 for(j in pos){
10 smatrix1 <- matrix(rep(as.matrix(form[pa[2], incorrect.items]),
11                       lengths[j]), nrow=lengths[j], byrow=TRUE)
12 smatrix2 <- matrix(rep(as.matrix(form2[pa[2], ]), lengths[j]),
13                   nrow=lengths[j], byrow=TRUE)
14 pr[j] <- mean(rowSums(form[subgroups[[j]], incorrect.items]==smatrix1))/ws
15 compare <- (form2[subgroups[[j]], ]==1)&(smatrix2==1)
16 prob[j,] <- colMeans(compare)
17 weight[j,] <- p1^(prob[j, ]*d2)
18 pj[j] <- mean(((compare)*1)%%as.matrix(weight[j, ]))
19 }

```

Fonte: Souza (2019) [16].

Figura 2.4 Porção do código que computa objetos para obtenção dos índices K_1 , K_2 , S_1 , S_2 no pacote *Copydetect*

```

1 ws <- sum(form2[pa[2], ] == 0, na.rm = TRUE)
2 incorrect.items <- which(form2[pa[2], ] == 0)
3 pr <- c()
4 prob <- matrix(nrow = (ncol(form) + 1), ncol = ncol(form))
5 weight <- matrix(nrow = (ncol(form) + 1), ncol = ncol(form))
6 pj <- c()
7 g = 1/length(resp.options)
8 d2 = -(1 + g)/g
9 for (j in 1:(ncol(form) + 1)) {
10 if (length(subgroups[[j]]) != 0) {
11 incorrect.items <- which(form2[pa[2], ] == 0)
12 smatrix1 <- as.data.frame(matrix(rep(as.matrix(form[pa[2],
13 incorrect.items]), length(subgroups[[j]])),
14 nrow = length(subgroups[[j]]), byrow = TRUE))
15 smatrix2 <- as.data.frame(matrix(rep(as.matrix(form2[pa[2], ],
16 length(subgroups[[j]])),
17 nrow = length(subgroups[[j]]), byrow = TRUE))
18 emp.agg <- rowSums(form[subgroups[[j]], incorrect.items] == smatrix1,
19 na.rm = TRUE)
20 pr[j] = mean(emp.agg, na.rm = TRUE)/ws
21 prob[j, ] <- colMeans((form2[subgroups[[j]], ] == 1) & (smatrix2 == 1),
22 na.rm = TRUE)
23 weight[j, ] <- (((1 + g)/(1 - g)) * exp(1))^(prob[j, ] * d2)
24 pj[j] <- mean(((form2[subgroups[[j]], ] == 1 &
25 smatrix2 == 1) * 1) %%% t(t(weight[j, ])), na.rm = TRUE)
26 }
27 else if (length(subgroups[[j]]) == 0) {
28 pr[j] = NA
29 pj[j] = NA
30 }
31 }

```

Fonte: Souza (2019) [16].

Portanto, a utilização do processamento em paralelo e as modificações feitas nas funções que computam os índices de similaridade em respostas de múltipla escolha tornaram o pacote *TestFraud* mais rápido na computação dos cálculos em comparação com o pacote

CopyDetect. Dessa forma, a partir das melhorias desse pacote é possível implementar o método hierárquico apresentado na Seção 3.2.

2.3 Testes de Hipóteses

Nesta Seção apresenta-se a teoria dos testes de hipóteses necessárias para aplicações dos métodos estatísticos de detecção de fraude descritos anteriormente. Onde são apresentados os possíveis erros ao assumir determinada hipótese.

O interesse principal reside no nível de significância adotado para o erro do tipo I. Este erro tem relação direta com taxa de falso positiva, que é considerar um par de indivíduos como suspeito de *cola* quando na realidade não houve fraude.

2.3.1 Tipos de erros

Nas aplicações há interesse em tomar a decisão de aceitar ou rejeitar um par de examinados como suspeito de fraude, por *cola*, com base na similaridade entre as respostas. Então, pode-se concluir por uma das duas hipóteses: “ H_0 : o par de indivíduos não é suspeito de *cola*” e a alternativa “ H_1 : o par de indivíduos é suspeito de *cola*”. A decisão de aceitar H_1 (ou rejeitar H_0) como verdadeira, pode-se estar cometendo um erro, pois, apesar da alta similaridade, o par de examinados pode não ter colado.

Por outro lado, situação semelhante pode acontecer com relação à aceitação de H_0 como verdadeira, e nesse caso se estaria considerando um par de examinados não suspeito quando na realidade ele é. Esses dois tipos de equívocos são denominados, respectivamente, erros dos tipos I e II. A situação está descrita na Tabela 2.3.

Tabela 2.3 *Tipos de erros em um teste de hipóteses.*

Decisão	H_0 é verdadeira	H_0 é falsa
Aceitar H_0	correto	erro tipo II
Rejeitar H_0	erro tipo I	correto

Fonte: Elaborada pelos autores.

As probabilidades de cometer os erros tipos I e II são conhecidas na literatura [4] por α e β , respectivamente. O erro tipo I também é denominado de *falso positivo*, enquanto o erro tipo II é conhecido como *falso negativo*.

2.3.2 Nível de confiança α

A construção de um teste de hipóteses parte da fixação no nível de significância α . Dessa forma, esse procedimento pode levar à rejeição da hipótese nula para um valor α e à não rejeição para um valor menor, conforme comparação do valor da estatística de teste com o valor tabelado (região crítica).

Uma forma alternativa de preceder é apresentar a probabilidade de significância ou nível descritivo ou p -valor [6]. Nesta maneira, o que se faz é indicar a probabilidade de se obter uma estatística de teste mais extrema que a estatística observada, sob as condições de H_0 ser verdadeira.

Neste estudo foi adotado o procedimento do p -valor, pois nos índices avaliados o pacote *TestFraud* já apresenta cada p -valor individualmente.

2.3.3 Taxa de falso positivo

As conclusões sobre rejeitar H_0 pode trazer grandes consequências. Por exemplo, na medicina, um paciente ao realizar um exame físico em que o resultado indica a presença de uma doença quando na realidade ela não existe.

Nos métodos de detecção de fraude em testes ocorre semelhante situação, considerar um par de examinados suspeitos de *cola* no teste quando na realidade não existe esse tipo de fraude. A proporção de pares classificados erroneamente como suspeitos é denominado, segundo Zopluoglu et al. [21], taxa de falso positivo (FP).

Dessa forma, grande são os esforços para que os índices apresentados nesta dissertação retornem estimativas próximas dos valores de α adotados nos testes. Uma alternativa de obtenção de um nível do erro tipo I mais preciso foi proposto por Souza [16], onde a criação da estatística T é soma das indicadoras de detecção de suspeita de fraude para cada um dos 7 índices. Na tabela 2.4 tem-se o controle do erro tipo I segundo os níveis de significância α .

Tabela 2.4 *Probabilidade de não cometer erro Tipo I para T.*

α	T						
	1	2	3	4	5	6	7
0,001	0,99841	0,99958	0,99987	0,99994	0,99996	0,99998	0,99999
0,005	0,99200	0,99714	0,99895	0,99932	0,99961	0,99981	0,99992
0,010	0,98413	0,99347	0,99732	0,99815	0,99883	0,99942	0,99977
0,020	0,96841	0,98501	0,99312	0,99498	0,99659	0,99822	0,99920
0,050	0,92146	0,95489	0,97646	0,98162	0,98596	0,99218	0,99585

Fonte: Souza [16].

Nesse estudo, para $T = 2$ (pelo menos 2 dos 7 índices detectar fraude) tem-se o valor mais próximo do α adotado. Nesta dissertação utilizou-se a estatística $T = 1$ (pelo menos um dos 7 índices detectar fraude) para determinar os pares de indivíduos suspeito de fraude que irão para o próximo nível hierárquico, com base na significância nominal. O objetivo é ser menos restritivo no início do processo e ir aumentando o rigor no decorrer do mesmo.

Capítulo 3

Metodologia Proposta

O armazenamento de grande base dados (ou Big Data) estão cada vez mais frequente na estatística, como por exemplos, operadores de telefonia, bancos, testes educacionais em larga escala entre outros. Essas bases demandam elevado tempo de computação para suas análises. Nesse sentido, existe a necessidade de técnicas computacionais que reduzam o tempo das tarefas. Uma das opções é o processamento distribuído, que consiste em executar de forma paralela as tarefas e assim dividindo o tempo de execução.

Através dessa execução em paralelo no R e da proposta do pacote *TestFraud* [16] começou a ser possível a utilização dos métodos estatísticos de detecção de fraude em avaliações com grande número de examinados. Antes, a computação dos métodos de similaridade de respostas em um teste era feita pelo pacote *CopyDetect*, desenvolvido por Zopluoglu [20], porém, o tempo de processamento era inviável, considerando uma prova com muitos indivíduos.

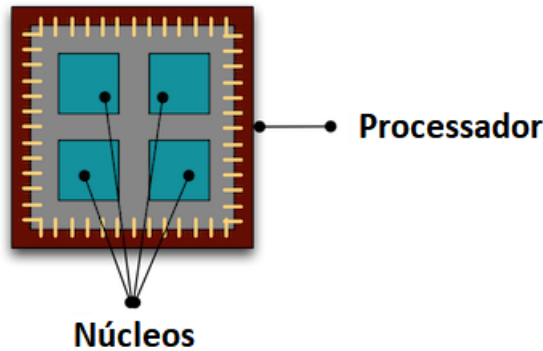
Por outro lado, considerando o cenário do ENEM, onde tem-se um mês para detectar suspeita de fraude sem comprometer os prazos do certame, o tempo de processamento do pacote *TestFraud* ainda carece de otimização. Neste sentido, apresenta-se a otimização hierárquica do supracitado pacote, no qual os pares de indivíduos detectados na etapa k servirão de base de entrada na etapa $k + 1$. Estas etapas são as diferentes áreas de avaliação do exame.

3.1 Suporte computacional

O CPU (Central Processing Unit) ou processador é um chip de silício que processa todas as informações enviadas pelo hardware (memória, HD, placa-mãe e outros dispositivos) e as operações solicitadas pelo software. Os computadores atuais possuem vários processadores e estes também possuem diversos núcleos (componente central do sistema

operativo), por exemplos dual-core (2 núcleos) e quad-core (4 núcleos). Tem-se na Figura 3.1 a representação do quad-core.

Figura 3.1 *Ilustração de um processador com 4 núcleos*



Fonte: Souza (2019)

Quanto mais núcleos, menores serão os tempos de execução dos cálculos. Nesse sentido, para a computação de cálculos em avaliações em larga escala, por exemplo o ENEM, é necessário, além de mais núcleos, um software adequado. Dentro os livres (concede liberdade ao usuário para executar, acessar e modificar o código fonte, e redistribuir cópias com ou sem modificações), o R (ou linguagem R) é o mais utilizado atualmente. Essa linguagem é compatível com os sistemas operacionais Windows, Linux, Unix e MacOS. Além disso, o R permite o processamento em paralelo ou distribuído (um sistema que interliga vários nós de processamento simultâneo). Por isso, o R foi o software utilizado nas análises estatísticas desta dissertação.

Em relação ao processamento em paralelo, o R oferece vários pacotes voltados para melhorar o desempenho, conforme página: *CRAN Task View: High - Performance and Parallel Computing with R*. Dentre esses pacotes disponíveis, foram utilizados nesse estudo *doParallel*, *parallel* e *foreach*. Esse funciona como interface entre estes dois últimos. O pacote *doParallel* é responsável pelos mecanismos necessários e gerenciamento do processamento em paralelo. Neste pacote, é necessário um tipo de registro, no qual utiliza a função *registerDoParallel* para especificar o número de processos a ser utilizado na paralelização, o que depende do uso ou não de parâmetro. Para o Windows (sistema utilizado na máquina de teste desse estudo) são criados três processadores (mais detalhes sobre *doParallel* em Weston & Calaway, 2019) [18].

Máquina de teste

Em todos os resultados obtidos nesta dissertação utilizou-se o computador com processador *AMD Ryzen 7 2700*, que possui 8 núcleos físicos com capacidade de executar 16 *threads*, ou seja, possui capacidade de emular 16 núcleos (físicos e lógicos), e opera à frequência de 3.2 Ghz (Max Turbo 4.1 GHz), com 32 GB de memória RAM, Cache L3: 16MB, Cache L2: 4MB, Potência: 65 W. Utilizou-se o sistema operacional Windows 10 Pro 64 bits.

3.2 Método Hierárquico

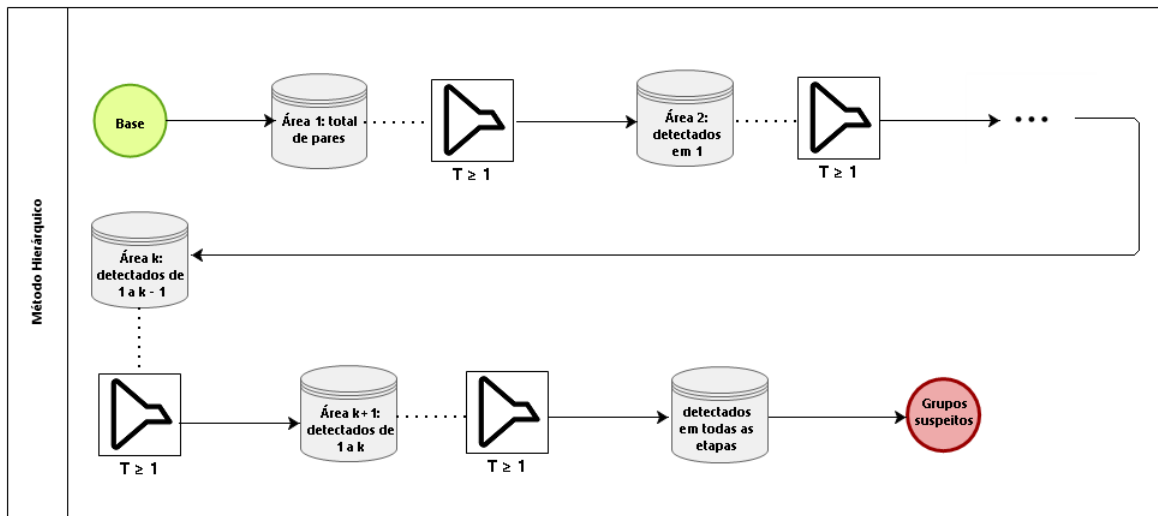
Algumas avaliações educacionais envolvem etapas ou áreas diferentes no mesmo exame. É o exemplo do ENEM, onde esse exame é dividido em quatro áreas, a saber:

1. Linguagens, Códigos e suas Tecnologias;
2. Ciências Humanas e suas Tecnologias;
3. Ciências da Natureza e suas Tecnologias;
4. Matemática e suas Tecnologias.

Em avaliações como essa, em larga escala, há a necessidade de a detecção de fraude ocorrer em tempo hábil. A partir disso, é proposto a otimização hierárquica do pacote *TestFraud*, cujo o objetivo é reduzir o tempo de computação dos índices.

Conforme a Figura 3.2, é ilustrado a hierarquização do exame segundo a ordem de aplicação das áreas. Os pares suspeitos de fraude ($T \geq 1$: pelo menos um dos 7 índices detectar suspeita de *cola*) na área 1 servirão de base na área 2 e assim por diante, até a última área. De maneira geral, os pares de indivíduos detectados na etapa k servirão de base de entrada na etapa $k + 1$. Segundo discutido na Seção 2.3, a quantidade de pares de examinados suspeito de transgressão na etapa k vai depender do nível de significância α adotado na etapa $k - 1$. Conseqüentemente o tempo de computação dos métodos de identificação nos níveis posteriores vai depender do valor nominal adotado do erro tipo I nos níveis anteriores.

Figura 3.2 Fluxograma do método hierárquico.



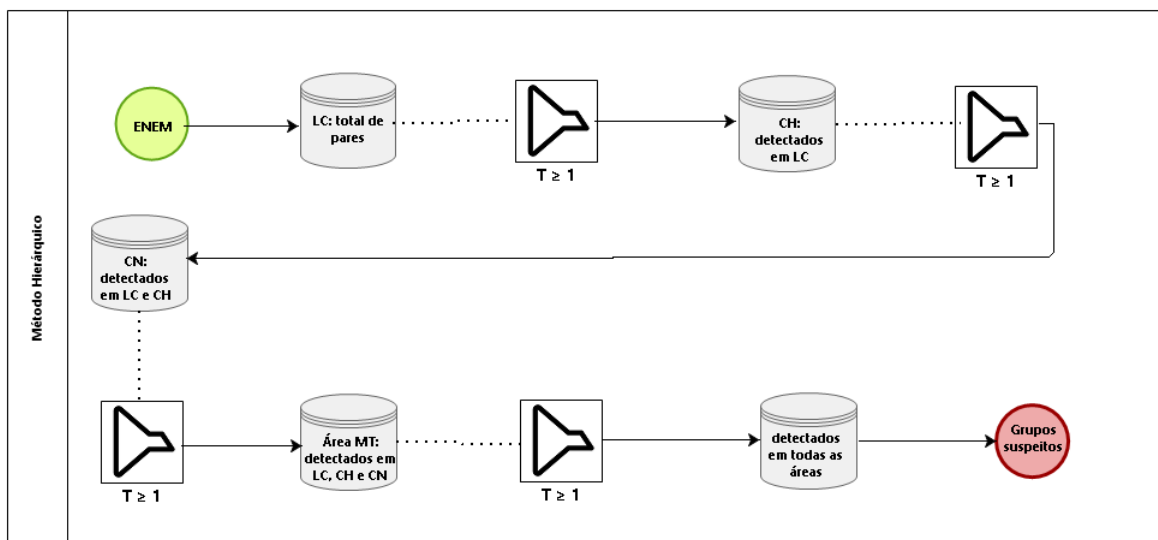
Powered by
bizagi
Modeler

Fonte: Elaborado pelos Autores.

Em relação ao ENEM, a análise da identificação de fraude por *cola* conforme método hierárquico será descrito pelo fluxograma da Figura 3.3, de acordo com a ordem de aplicação das provas. Na prova de Linguagens, Códigos e suas Tecnologias (LC) tem-se a formação de todos os pares. Os suspeitos de fraude nessa área servirão de base para área de Ciências Humanas e suas Tecnologias (CH). Assim também, como os detectados em CH servirão de filtro para de prova de Ciências da Natureza e suas Tecnologias (CN). Logo, o total de pares analisados em Matemática e suas Tecnologias (MT) será os suspeito em CN, pelo menos um índice detectar fraude, e tem-se por fim os pares de indivíduos detectados como fraude nas quatro áreas do exame.

Logo, o método hierárquico utiliza toda a informação contida nos 7 índices e considera como suspeitos de fraude os examinados identificados em todas as áreas da avaliação. Desse modo, esta metodologia é conservadora em aceitar um determinado par de examinados como coladores. Esse fator, contribui para diminuição da quantidade de indivíduos a serem investigados pela autoridade policial competente. Outro aspecto, é que o tempo de computação dos métodos estatísticos se torna viável nos prazos do certame.

Figura 3.3 Fluxograma do método hierárquico para o ENEM.



Fonte: Elaborado pelos Autores.

Capítulo 4

Resultados

Em primeiro, realizou-se a avaliação dos 7 índices aplicados nesse estudo com base na taxa de falso positivo (FP). Essa avaliação foi realizada para uma população simulada, sem fraude, de $J = 5.000$, gerando assim um total de 12.497.500 pares analisados. Essa quantidade suficientemente grande fornece convergência das estimativas. Assim, foi possível identificar os índices com melhores taxas de FP, mesmo em populações com alta similaridade. Ainda em dados simulados, objetivando otimizar o tempo de processamento computacional dos índices descritos na Seção 2.2 aplicou-se o método hierárquico onde houve significativa redução do tempo de cálculo para identificação de fraude. Os resultados também sugerem adotar níveis de significância maiores nas etapas iniciais do processo. De forma geral, a proposta de hierarquização foi eficiente quanto a meta inicial propostas, tornar a utilização dos métodos estatísticos de detecção de fraude menos lenta.

Quanto a aplicação em dados reais, foi utilizado o método hierárquico para identificar possíveis transgressões na prova do ENEM de 2018 para os candidatos que realizaram a prova na capital do Piauí, Teresina. A motivação de escolha dessa cidade é devido aos inúmeros casos de tentativas de fraudes em teste divulgados pela empresa, além da baixa quantidade de examinados. De início, realizou-se a análise descritiva das proficiências e escores dos examinados, cuja análise é de extrema importância para aplicação dos testes estatísticos de detecção de fraude. Os escores são definidos pela soma dos itens (1:correto; 0:incorreto) de cada examinado j , com base na TCM, enquanto as proficiências são estimadas pela TRI, conforme respostas dicotomizadas ou nominais. Para essas duas medidas foram construídos os histogramas e calculadas as medidas de posição e dispersão. Em relação a detecção de fraude por *cola*, a metodologia proposta foi eficiente em listar os suspeitos de transgressões ao exame.

4.1 Estudo de Simulação

4.1.1 Avaliação dos índices

Os sete índices apresentados na Seção 2.2 foram avaliados com o objetivo de verificar a taxa de falso positivo (FP), calculado pelo algoritmo do Apêndice A, em dois cenários diferentes (ambos sem presença de fraude). No primeiro cenário, foi simulado um exame com $I = 45$ itens, $V = 5$ alternativas e aplicados a uma população de $J = 5.000$ (ou 12.497.500 pares), cuja a ideia é verificar os índices que retornam a FP mais próxima do α adotado. Na Figura 4.1 tem-se a descrição das estimativas de erro tipo I segundo os níveis de significância nominais (0,1%; 0,5%; 1%; 2%; 5%). Para todos os métodos estatísticos de detecção de fraude, as taxas de FP foram abaixo do valor esperado para cada nível nominal. Os índices mais precisos foram K_1 e ω , enquanto K e S_2 mais conservadores (baixa taxa de erro).

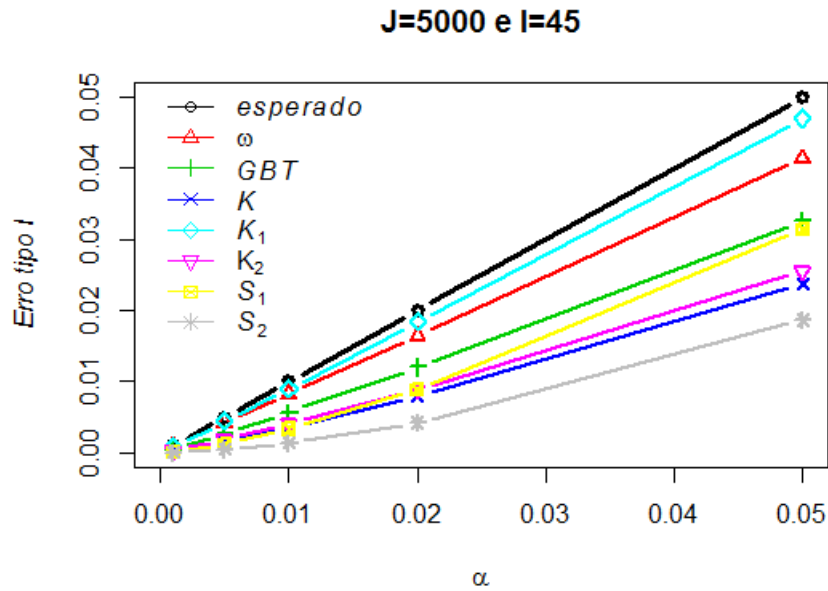
Todavia, os resultados obtidos diferem da literatura [21] em relação a ordem de eficiência dos índices. Em respostas nominais, Zopluglu et al. (2017) obteve ω como melhor índice e K_1 apenas como terceiro. Para S_2 , GBT e demais variantes de K não houve divergência com a literatura, sendo-os classificados como conservadores.

Além disso, para mesma população simulada foi obtido a probabilidade do erro tipo I para 50 níveis de significância estabelecidos, variado de 0,001 até 0,05. Para as taxas de retorno ou FP, conforme cada índice, foram cálculos o Erro Quadrático Médio (EQM), onde os resultados são apresentados na Figura 4.2. Os resultados dos métodos K_1 e ω tiveram menores valores de EQM. Por outro lado, S_2 e K os maiores valores.

Já para o segundo cenário, a ideia é demonstrar que os índices sofrem alterações à medida que a similaridade entre os indivíduos aumenta. Considerando os mesmos parâmetros da simulação anterior, com a diferença que nesse cenário apenas comparou-se os pares com escore mínimo de 30, ou seja, adotando um critério de escore mínimo como proposto por Souza (2019). Assim, a quantidade de pares analisados reduziu de 12.497.500 para 1.999.000. Nos resultados obtidos, Figura 4.3, os métodos ω e GBT tiveram taxas maiores que o valor esperado, os demais métodos foram menores que os níveis nominais. As derivações dos índices K apresentaram valores mais precisos.

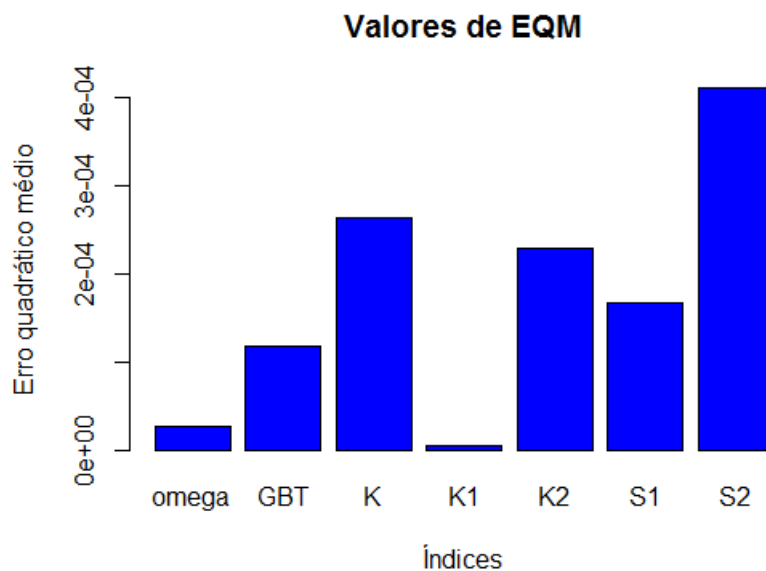
Portanto, os índices aplicados neste estudo são sensíveis as diversas mudanças nos parâmetros estabelecidos. Primeiro, deve-se considerar os modelos de respostas da TRI,

Figura 4.1 Taxas de falso positivo (erro tipo I) dos índices para resultados simulados de respostas nominais.



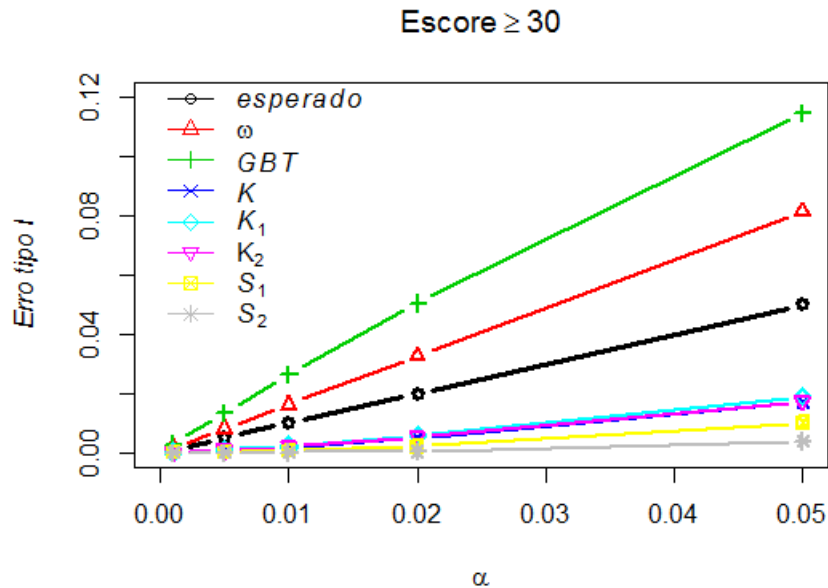
Fonte: Elaborado pelos Autores.

Figura 4.2 Valores de erro quadrático médio para os índices de resultados simulados de respostas nominais.



Fonte: Elaborado pelos Autores.

Figura 4.3 Taxas de falso positivo (erro tipo I) dos índices para resultados simulados de respostas nominais com escore mínimo de 30.



Fonte: Elaborado pelos Autores.

dicotomizados ou nominais, como descrito na literatura [21] afetam as estimativas do erro tipo I. Em contrapartida, os pares com alta similaridade nas respostas, como por exemplo, na adoção de um escore mínimo, ou um quantil à direita ou até mesmo em uma população com alto nível de acerto em um exame, podem afetar significativamente as taxas de falso positivo dos índices. Dessa forma, os métodos estatísticos de detecção de fraude conservadores (K e suas derivações) são bastantes importantes em populações com alta similaridade de respostas.

4.1.2 Desempenho da Otimização Hierárquica

Nos estudos de simulação, gerou-se populações de tamanhos diferentes (variando de 1.000 à 5.000) e valores nominais de α distintos (0,1%; 0,5%; 1%; 2%; 5%), ambos para uma prova de 180 itens dividido em quatro áreas. A ideia é verificar o impacto das combinações de quantidades de pares e níveis de significância no tempo de processamento dos métodos estatísticos de detecção de fraude. Esse tempo de execução foi medido pelo pacote *microbenchmark* [10].

Conforme Tabela 4.1, o método hierárquico no pacote *TestFraud* reduziu em torno de

73% o tempo de cálculo dos 7 índices utilizados nesse estudo, em comparação ao mesmo pacote sem hierarquia. Para uma população de 1.000 indivíduos (ou 499.500 pares) o tempo de computação do pacote *TestFraud* sem hierarquia foi de 11.25043 horas, enquanto o otimizado em apenas 3,064339 horas, o que resulta numa redução relativa de 72,76%. O tamanho máximo de pares simulados foi de 12.497.500 (população de 5.000), resultado em 281,48594 e 76,65702 horas, respectivamente, sem e com hierarquia. Neste método a média por par foi de 0,02208 segundos e para esse a média foi quase quatro vezes maior, 0,08108 segundos.

Tabela 4.1 *Tempo de simulação computacional do processamento (em horas) dos índices no pacote TestFraud sem e com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=5\%$.*

População (J)	Sem hierarquia	Hierárquico	Variação
1.000	11,25043	3,06439	-72,76%
2.000	45,02424	12,27349	-72,74%
3.000	101,32142	27,57809	-72,78%
4.000	180,14199	49,09107	-72,75%
5.000	281,48594	76,65702	-72,77%

Fonte: Elaborado pelos autores.

Outro ponto importante é que na Tabela 4.1 o valor adotado para o erro tipo I foi de 5%, o que contribui para um maior número de pares nas etapas seguintes de detecção e conseqüentemente maior tempo de computação, ou seja, um nível mais conservador diminuiria ainda mais o tempo de execução. Essa situação é descrita nas Tabelas de 4.2 à 4.6.

Nessas Tabelas, o período de processamento é menor em cada nível inferior, como era de se esperar, pois tem-se menos pares nessas etapas. Em relação a uma população de $J = 5.000$, o tempo de cálculo reduz para 72,66696 horas, considerando $\alpha = 2\%$. Para valores nominais menores, a tendência é minimizar ainda mais esse tempo. Considerando esse mesmo tamanho de universo, tem-se os seguintes tempos de computação, em horas: 71,50629; 70,93900; 70,48355, respectivamente, para os erros nominais 1%, 0,5%, 0,1%. Para os demais tamanho de J ocorre situação semelhante.

Tabela 4.2 *Tempo de simulação computacional do processamento (em horas) dos índices no pacote TestFraud com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=5\%$.*

População (J)	Níveis de hierárquicos				Total
	1	2	3	4	
1.000	2,81261	0,23121	0,01901	0,00156	3,06439
2.000	11,25606	0,93358	0,07743	0,00642	12,27349
3.000	25,33036	2,06556	0,16844	0,01374	27,57809
4.000	45,03550	3,72245	0,30768	0,02543	49,09107
5.000	70,37149	5,77308	0,47361	0,03885	76,65702

Fonte: Elaborado pelos autores.

Tabela 4.3 *Tempo de simulação computacional do processamento (em horas) dos índices no pacote TestFraud com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=2\%$.*

População (J)	Níveis de hierárquicos				Total
	1	2	3	4	
1.000	2,81261	0,08885	0,00281	0,00009	2,90435
2.000	11,25606	0,35558	0,01123	0,00035	11,62323
3.000	25,33036	0,80019	0,02528	0,00080	26,15662
4.000	45,03550	1,42267	0,04494	0,00142	46,50453
5.000	70,37149	2,22304	0,07023	0,00222	72,66696

Fonte: Elaborado pelos autores.

Tabela 4.4 *Tempo de simulação computacional do processamento (em horas) dos índices no pacote TestFraud com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=1\%$.*

População (J)	Níveis de hierárquicos				Total
	1	2	3	4	
1.000	2,81261	0,04464	0,00071	0,00001	2,85796
2.000	11,25606	0,17863	0,00283	0,00004	11,43757
3.000	25,33036	0,40199	0,00638	0,00010	25,73883
4.000	45,03550	0,71471	0,01134	0,00018	45,76173
5.000	70,37149	1,11680	0,01772	0,00028	71,50629

Fonte: Elaborado pelos autores.

Tabela 4.5 *Tempo de simulação computacional do processamento (em horas) dos índices no pacote TestFraud com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=0,5\%$.*

População (J)	Níveis de hierárquicos				Total
	1	2	3	4	
1.000	2,81261	0,02250	0,00018	0,00000	2,83529
2.000	11,25606	0,09005	0,00072	0,00001	11,34683
3.000	25,33036	0,20264	0,00162	0,00001	25,53463
4.000	45,03550	0,36028	0,00288	0,00002	45,39869
5.000	70,37149	0,56297	0,00450	0,00004	70,93900

Fonte: Elaborado pelos autores.

Tabela 4.6 *Tempo de simulação computacional do processamento (em horas) dos índices no pacote TestFraud com o método hierárquico para uma avaliação dividido em quatro áreas, cada uma com $I=45$, segundo o tamanho da população e $\alpha=0,1\%$.*

População (J)	Níveis de hierárquicos				Total
	1	2	3	4	
1.000	2,81261	0,00447	0,00001	0,00000	2,81709
2.000	11,25606	0,01790	0,00003	0,00000	11,27398
3.000	25,33036	0,04028	0,00006	0,00000	25,37070
4.000	45,03550	0,07161	0,00011	0,00000	45,10722
5.000	70,37149	0,11189	0,00018	0,00000	70,48355

Fonte: Elaborado pelos autores.

4.2 Aplicação em Dados Reais

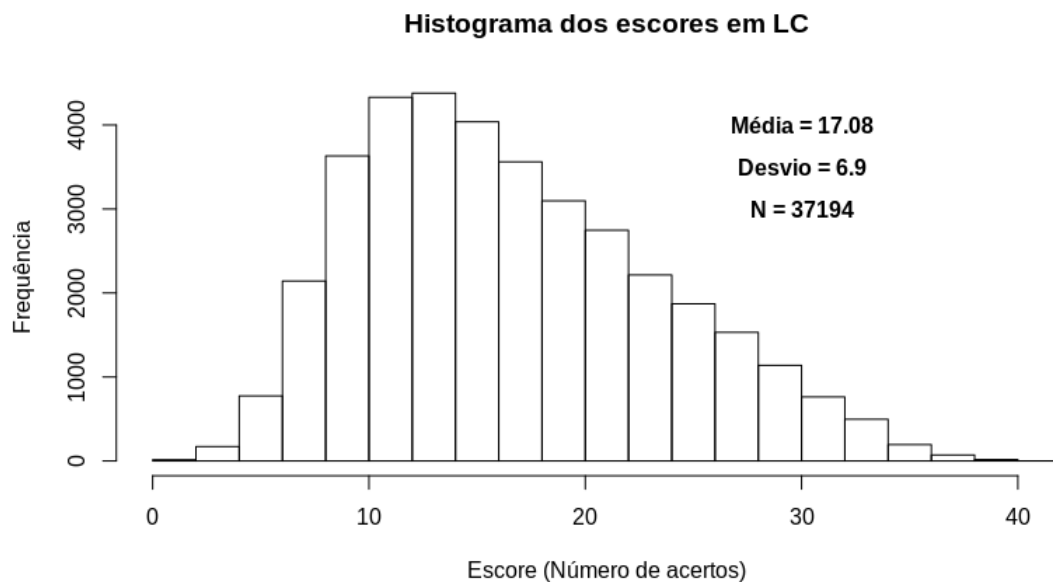
4.2.1 Distribuição dos Escores

A TCM analisa os itens com o objetivo de selecionar os melhores, geralmente de um banco de itens, considerando a dificuldade, a discriminação e a correlação bisserial das respostas. Para cada item considera-se 0 em caso de erro e 1 para acerto. Assim, denominados como escore a soma dicotomizada das respostas aos de uma prova.

Dessa forma, para o ENEM-2018 em Teresina-PI, obteve-se o total de 37.194 candidatos que tiveram presença nas quatro áreas do exame. Nesta população de estudo, foram construídos os histogramas dos escores para cada área de conhecimento, com 45 itens por área. Na Figura 4.4, tem-se a distribuição da prova de Linguagens, Códigos e Suas Tecnologias

(LC). Nessa prova, obteve-se a maior média de acertos (17,08 itens) com desvio padrão de 6,90 itens. O Coeficiente de Variação (CV), razão entre o desvio padrão e a média, foi de 40,42%. Nota-se em LC leve assimetria a direita, conforme Coeficiente de Assimetria de Pearson (ASP), igual a 0,513. Em relação ao achatamento da distribuição, teve-se um Coeficiente Percentílico de Curtose (CP) igual a 2,620, indicando uma distribuição aproximadamente platicúrtica ($CP < 3,000$).

Figura 4.4 *Histograma dos escores da prova de Linguagens, Códigos e suas Tecnologias, ENEM-2018, Teresina-PI.*



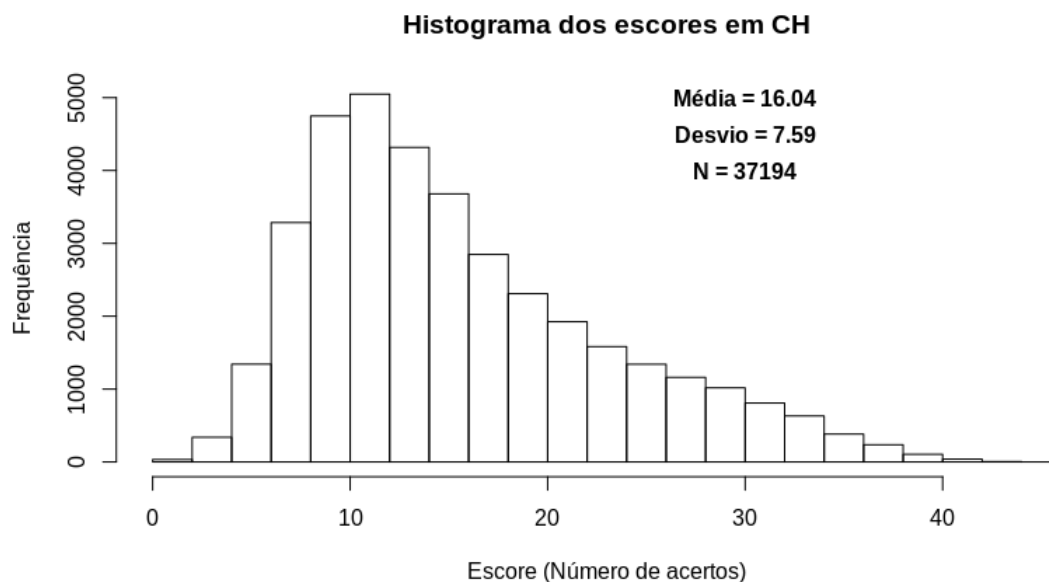
Fonte: Elaborado pelos autores.

A prova de Ciências Humanas e Suas Tecnologias (CH) apresentou a maior variabilidade ($CV = 47,30\%$). Nessa prova, a média de acertos foi de 16,04 itens com desvio padrão de 7,59 itens. Quanto a assimetria, a distribuição dos escores de CH, Figura 4.5, é assimétrica positiva ($ASP = 0,852$). Além disso, pode-se classificar essa distribuição como leptocúrtica ($CP = 3,130$).

Em relação a prova de Ciências da Natureza e suas Tecnologias (CN), conforme descrito na Figura 4.6, a prova apresenta a menor média de acertos (11,91 itens) e desvio padrão de 5,24. Nas distribuições dos escores, essa prova apresenta assimétrica positiva ($ASP = 1,645$), o que indica baixa frequência de candidatos com escores maiores. O CV dessa área foi de 43,94% e CP de 6,947 (leptocúrtica).

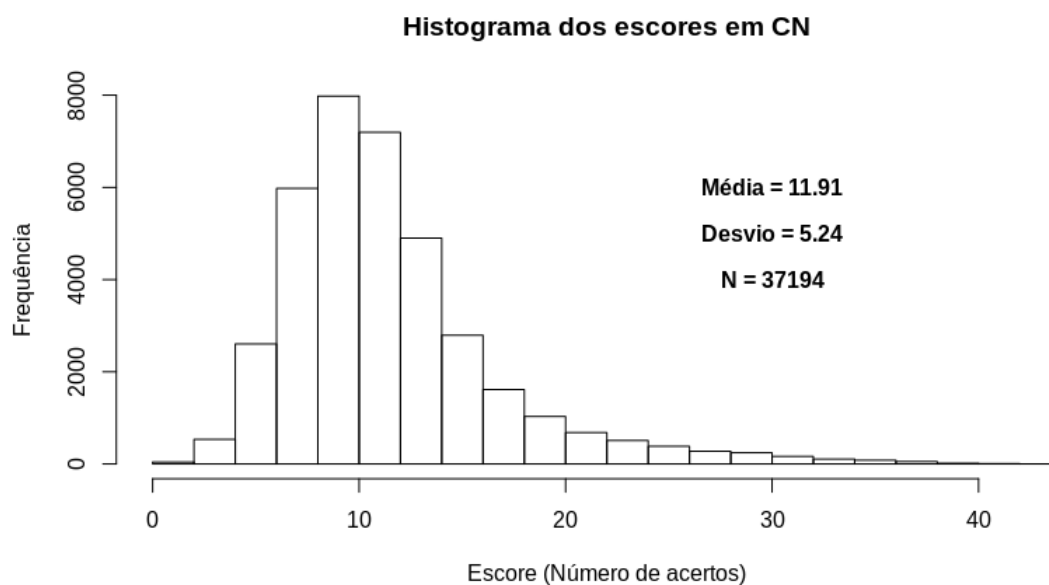
A última prova do ENEM 2018 é a prova Matemática e suas Tecnologias (MT), cuja

Figura 4.5 *Histograma dos escores da prova de Ciências Humanas e suas Tecnologias, ENEM-2018, Teresina-PI.*



Fonte: Elaborado pelos autores.

Figura 4.6 *Histograma dos escores da prova de Ciências da Natureza e suas Tecnologias, ENEM-2018, Teresina-PI.*

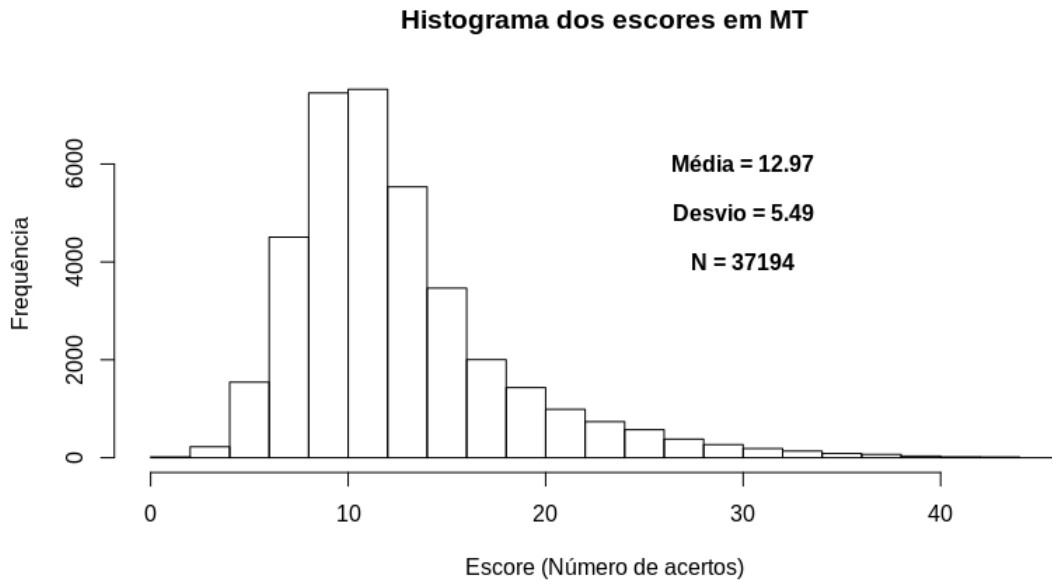


Fonte: Elaborado pelos autores.

a distribuição dos escores está na Figura 4.7. A média de acertos foi de 12,97 itens e um desvio padrão de 5,49 itens. A variabilidade relativa (CV) encontrada foi de 42,36%.

Quanto a forma da distribuição, essa área apresenta assimetria a direita ($ASP = 1,544$). Quanto a curtose, o CP (6,301) indica uma distribuição leptocúrtica.

Figura 4.7 *Histograma dos escores da prova de Matemática e suas Tecnologias, ENEM-2018, Teresina-PI.*



Fonte: Elaborado pelos autores.

Portando, a distribuição dos escores são de suma importância para os cálculos de detecção de fraude apresentados na Seção 2.2, pois em examinados de alta pontuação a similaridade entre as respostas é maior e conseqüentemente maior taxa de falso positivo. Além, dos estudos que visam de reduzir a quantidade de pares analisados, como por exemplo, o estudo de escores mínimos introduzido por Souza [16].

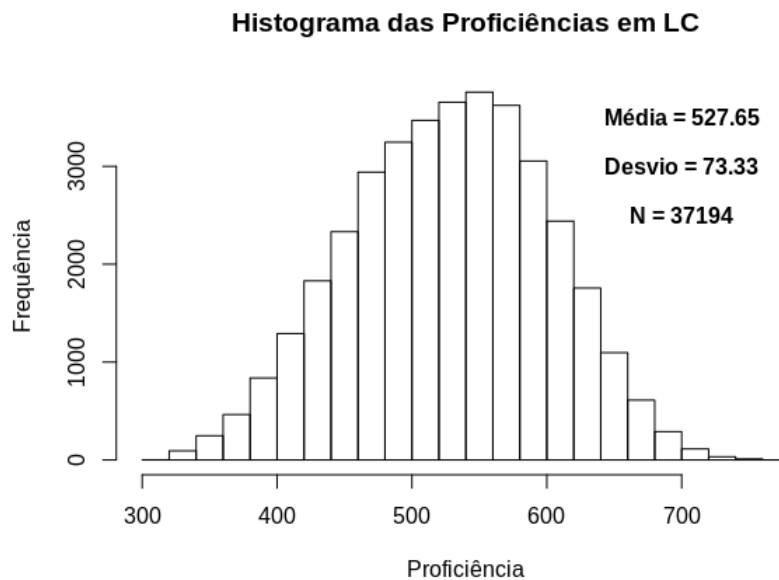
4.2.2 Distribuição das Proficiências

Como já discutido na Seção 2.1, a TRI permite estimar a habilidade (θ_j) de um examinado pelos modelos estatísticos, tendo como base os parâmetros dos itens e o tipo de respostas (dicotomizadas ou nominais). Então, denomina-se θ_j a proficiência estimada de um avaliado através da TRI. Nos histogramas seguintes, considerou-se os mesmos filtros da Seção anterior: examinados que fizeram a prova do ENEM-2018 em Teresina-PI e presença nas quatro áreas do exame.

Tem-se nas Figuras 4.8 e 4.9 as distribuições das proficiências das provas de Linguagens, Códigos e suas Tecnologias (LC) e Ciências Humanas e suas Tecnologias (CH), respecti-

vamente. A área de LC apresenta média de 527,65 e desvio padrão de 73,33. Enquanto na área de CH tem-se uma média (569,12) maior com desvio padrão de 79,69. A distribuição da prova de LC tem uma forma próxima de simetria ou uma leve assimetria a esquerda ($ASP = -0,092$), enquanto a CH tem assimetria a esquerda ($ASP = -0,241$). Quanto ao coeficiente de variação (CV), os valores foram 13,90% e 14,00%, respectivamente, as provas LC e CH. Em relação a curtose, ambas apresentam distribuições platicúrticas, LC ($CP = 2,561$) e CH ($CP = 2,216$).

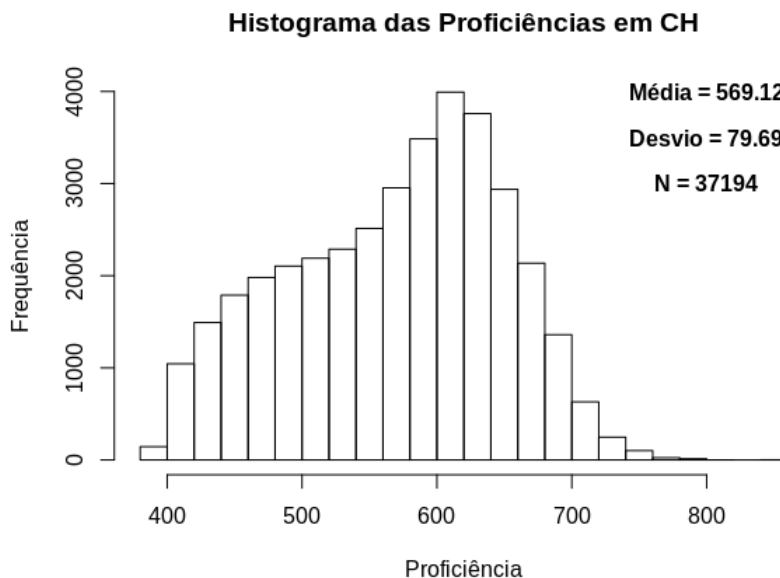
Figura 4.8 *Histograma das proficiências da prova de Linguagens, Códigos e suas Tecnologias, ENEM-2018, Teresina-PI.*



Fonte: Elaborado pelos autores.

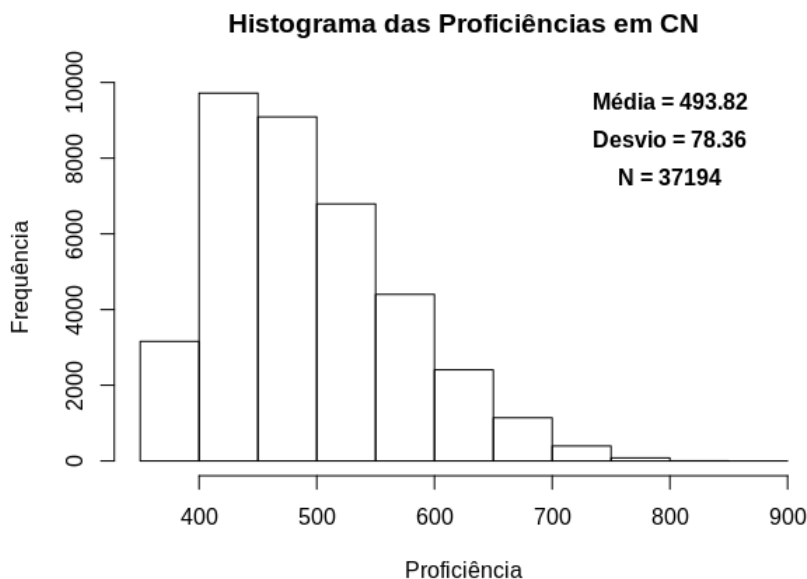
Ainda no ENEM de 2018, no segundo dia de avaliação foram realizadas as provas de Ciências da Natureza e suas Tecnologias (CN) e Matemática e suas Tecnologias (MT). Na distribuição da prova de CN, Figura 4.10, tem-se uma assimetria positiva ($ASP = 0,754$) e média de 493,82 (com desvio padrão de 78,36). A variabilidade relativa (CV) foi de 15,87% e sua distribuição é leptocúrtica ($CP = 3,131$). Na prova de MT, Figura 4.11, ocorre situação semelhante a distribuição do escore para essa mesma área, abordado na Seção anterior, onde há acentuada assimetria a direita ($ASP = 0,786$). Sua distribuição é aproximadamente mesocúrtica ($CP = 3,050$). Nessa assimetria, tem-se por consequência baixa frequência de notas (θ_j) maiores. Nesta prova, a média foi de 538,13 e desvio padrão de 110,35. Em relação ao CV, tem-se maior variabilidade (20,51%) entre todas as provas.

Figura 4.9 *Histograma das proficiências da prova de Ciências Humanas e suas Tecnologias, ENEM-2018, Teresina-PI.*



Fonte: Elaborado pelos autores.

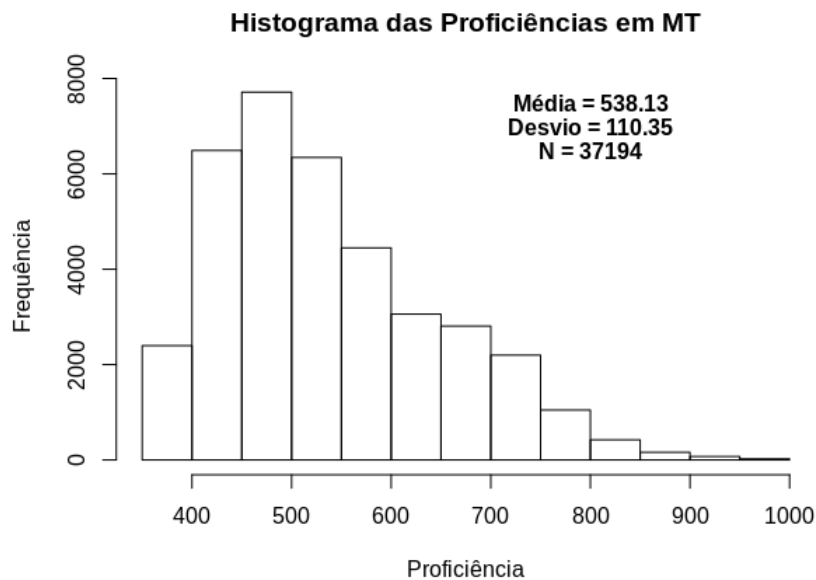
Figura 4.10 *Histograma das proficiências da prova de Ciências da Natureza e suas Tecnologias, ENEM-2018, Teresina-PI.*



Fonte: Elaborado pelos autores.

Por fim, as quatro áreas do ENEM de 2018 apresentam características diferentes, o que é esperado, pois as provas são calibradas de forma independente. As provas de LC e

Figura 4.11 *Histograma das proficiências da prova de Matemática e suas Tecnologias, ENEM-2018, Teresina-PI.*



Fonte: Elaborado pelos autores.

CH apresentam assimetria a esquerda, baixa frequência para notas inferiores, enquanto as provas de CN e MT assimetria a direita, baixa frequência para notas superiores.

Na próxima Seção é realizado as avaliações dos índices, onde é observado alteração das taxas de falso positivo para distribuições diferentes.

4.2.3 Detecção de Fraude

Em populações simuladas, a otimização hierárquica apresentou redução considerável do tempo de processamento computacional dos índices. Agora, o objetivo é aplicar esse método em dados reais. Essa aplicação foi realizada na base de dados do ENEM de 2018 para cidade de Teresina-PI. A supracitada base é disponibilizada pelo INEP (Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira), onde tem-se informações sobre as provas, gabaritos e respostas dos examinados. Nessa base, tem-se 37.194 candidatos que tiveram presença na quatro áreas do exame, conforme descrita na Seção 3.2. Do total de examinados, selecionou-se 5% dos indivíduos de maiores proficiências na prova de Linguagens, Códigos e suas Tecnologias (LC), primeiro nível de hierárquico. Esse critério de seleção foi adotado para redução do tempo de computação dos métodos estatísticos.

Dessa forma, aplicou-se os sete índices de detecção de fraude a 1.860 candidatos, os quais formaram o total de 1.728.870 pares de respostas.

Em relação ao tempo de cálculos dos índices, Tabela 4.7, o pacote *TestFraud* sem hierarquia executou os 1.728.870 pares em 58,45848 horas, considerando as quatro áreas do ENEM com 45 itens cada. Já o supracitado pacote com a opção hierárquica, o tempo de computação dos métodos estatísticos reduziu para 19,49037 horas ou uma redução relativa de 66,66%. A descrição da aplicação do método hierárquico é apresentada na Figura 4.12, onde nas áreas de LC, CH e CN adotou-se $\alpha = 0,05$ para os testes de significância dos sete índices. Nesses testes de significância, considerou-se um par como suspeita de fraude quando pelo menos um índice detectar fraude ($T \geq 1$). Na prova de LC (ou primeiro nível) foram detectados suspeitos de fraudes em 424.451 pares, o que representa uma taxa de detecção de 24,55% (divisão do total de pares suspeitos na área k pelo total de pares suspeitos da área $k - 1$). O segundo nível (CH) tem por bases os pares identificados como suspeitos em LC, dos quais permaneceu um total de 115.040 ou uma taxa de 27,10%. Na prova de CN (segundo nível hierárquico) teve-se uma taxa de 32,42% em relação a CH, o que resultou em 37.297 pares identificados como possíveis transgressões. No último nível tem-se a prova de MT, onde o valor nominal de α escolhido foi de 0,001. Nesta última adotou-se um caráter mais conservador devido à grande evidência sobre os pares finais. Dessa forma, 4.989 pares, taxa de 27,10% em relação a CN, apresentarem suspeitas de fraude nas quatro áreas do exame.

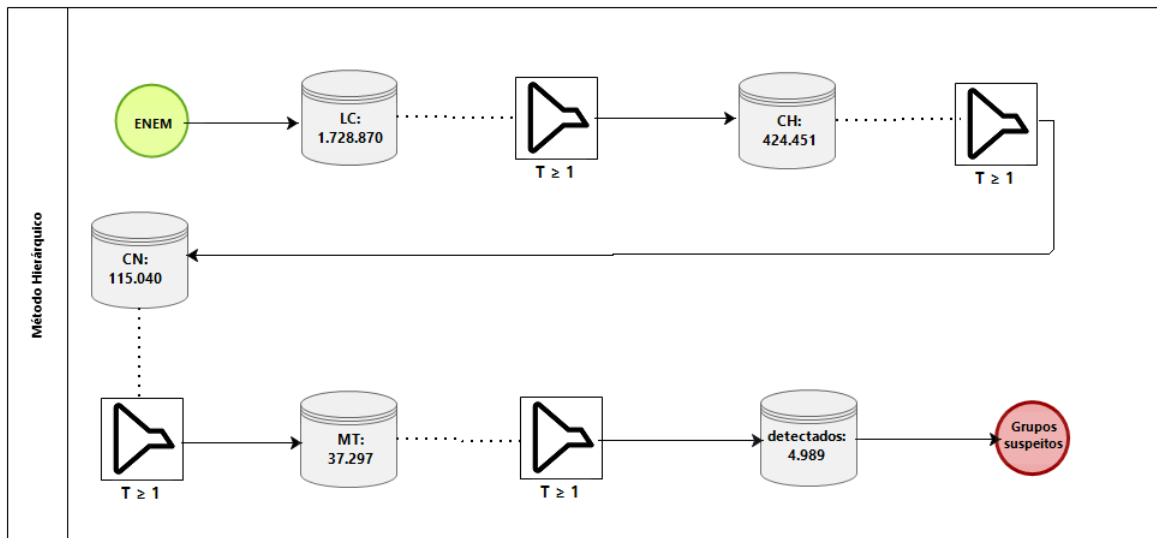
Tabela 4.7 *Tempo de processamento computacional (em horas) dos índices no pacote Test-Fraud sem e com o método hierárquico para 1.728.870 pares da prova do ENEM-2018 em Teresina-PI, $\alpha=5\%$.*

Métodos	Tempo (h)	Variação
Sem hierarquia	58,45848	-66,66%
Hierárquico	19,49037	-

Fonte: Elaborado pelos Autores.

A combinação desses pares finais (1.728.870) resultou em 639 candidatos suspeitos de fraude por *cola* em todas as etapas do exame. Tem-se na Tabela 4.8 a descrição dos 40 examinados mais frequentes na formação dos pares finais do processo. O indivíduo de posição 8466 no banco de dados teve pareado em 133 vezes, ou seja, este apresentou indícios de fraude com outros 133 examinados. O examinado de posição 3301 teve frequência de

Figura 4.12 Fluxograma do método hierárquico para o ENEM-2018, Teresina-PI.



Powered by
bizagi
Modeler

Fonte: Elaborado pelos autores.

formação de pares com outros 125 candidatos. Os avaliados nas ordens 7513 e 8683 tiveram repetições de, respectivamente, 118 e 106. Dentre os 40 com maior incidência de pares, as posições 8169 e 22741 tiveram a menor frequência, 58 incidências.

Tem-se nas Tabelas 4.9 e 4.10 a descrição da posição dos examinados no banco de dados que tiveram suspeita de fraude por *cola* com os indivíduos de ordens 8466 e 3301 na mesma base. Estes dois candidatos apresentaram os maiores número de interações nos pares finais do processo hierárquico, dessa forma tem-se grande evidência sobre a possível existência de transgressão ao exame nessas detecções.

Tabela 4.8 *Distribuição dos 40 examinados, suspeitos de fraude por cola, com maior frequência nos pares finais do processo hierárquico. ENEM-2018 em Teresina-PI.*

Posição do Examinado	Frequência	Posição do Examinado	Frequência
8466	133	29270	75
3301	125	19998	74
7513	118	25486	74
8683	106	3762	73
34344	105	8374	72
16223	101	7289	70
429	97	8837	70
1829	95	18274	69
10441	94	18982	69
22092	90	14400	68
23792	89	21220	68
15153	88	28572	68
25235	86	31396	68
6649	85	7633	65
7726	85	7845	64
7623	84	11986	64
25717	84	15963	61
3257	82	33671	61
8169	82	274	58
22741	78	12263	58
Total de suspeitos	639	-	-

Fonte: Elaborado pelos autores.

Tabela 4.9 *Descrição dos examinados, segundo a posição no banco de dados, suspeitos de fraude por cola que tiveram ligação com o indivíduo 8466 nos pares finais do processo hierárquico. ENEM-2018 em Teresina-PI.*

Posição	Posição	Posição	Posição	Posição
7633	3146	5670	7513	16305
28572	3207	5846	7623	16386
429	3257	6115	7675	23859
592	3277	6301	7678	25235
682	3301	6649	7726	25458
1267	3685	6879	8169	25486
1497	3762	7021	8187	25588
1829	5050	7024	8278	25712
1917	5456	7160	8374	25717
3025	5588	7354	8683	26439
8824	11868	13432	15200	26760
8837	11986	14400	15300	27323
8876	12182	14416	15523	15153
9078	12233	14556	15818	15159
9826	12258	14671	15846	22363
10441	12263	14824	15879	22556
10764	12649	14825	15963	22673
11314	12661	15085	16223	22729
11607	12698	22741	29270	18045
11775	12832	22949	31128	18274
16647	18575	23018	31134	28198
16688	18982	23060	31396	28237
16865	19610	23726	31800	28280
16901	20182	23792	33671	28706
17173	20460	21878	37033	17908
17328	20768	22092	21431	-
17401	21220	34344	36199	-

Fonte: Elaborado pelos autores.

Tabela 4.10 *Descrição dos examinados, segundo a posição no banco de dados, suspeitos de fraude por cola que tiveram ligação com o indivíduo 3301 nos pares finais do processo hierárquico. ENEM-2018 em Teresina-PI.*

Posição	Posição	Posição	Posição	Posição
7633	7137	12649	18392	23427
8466	7263	13281	18575	23516
28572	7289	14400	18588	23596
274	7513	14556	19610	23792
429	7532	14825	19646	23859
592	7623	15140	19742	24610
1151	7726	15153	19744	25235
1185	7845	15159	19998	25458
1267	8065	15300	20066	25486
1414	8169	15523	20098	25588
1652	8374	15818	20123	25591
1829	8683	15963	20815	25712
3257	9351	16035	21378	25717
3424	9709	16223	21497	27323
5219	9826	16305	22092	28130
5333	10278	16393	22363	28237
5456	10441	16732	22627	29270
5458	10477	16930	22673	30722
5670	10567	17328	22718	31167
6115	10764	17401	22729	31396
6649	11118	17802	22741	32529
6669	11314	17908	22949	33329
6694	11962	18028	23018	33671
7024	11975	18274	23155	34344
7136	11986	18353	23395	37033

Fonte: Elaborado pelos autores.

Capítulo 5

Considerações Finais

Primeiramente, foram apresentados nesse estudo os principais métodos estatísticos para detecção fraudes em testes (por *cola*), ressaltando as dificuldades da aplicação em exames envolvendo muitos indivíduos, devido ao demasiado tempo de processamento computacional. Neste sentido, umas das soluções apresentadas na literatura para contornar esse problema foi a construção do pacote *TestFraud*, onde o processamento em paralelo reduziu o tempo de execução das tarefas (ver [16], [13]).

Em segunda análise, avaliou-se as taxas de falso positivo dos índices estatísticos de detecção de fraude em dados simulados sem fraude. Conclui-se que nesses tipos de simulações alguns índices tiveram estimativas de erro do tipo I próximas do valor nominal, enquanto em outros as estimativas foram bastante conservadoras. Por outro lado, em situação de subgrupo de populações de alta proficiência, como por exemplo a adoção de quantil a direita, esses índices conservadores são bastantes úteis, pois tendem a acertar mais nessas situações.

Quanto ao tempo de processamento computacional dos métodos, a otimização hierárquica do pacote *TestFraud* reduziu em mais de 70% esse tempo para dados simulados. Dessa forma, a proposta desse método é fundamental para aplicação dos índices de similaridade em grandes populações de examinados. Outro ponto forte dessa proposta é que usa toda informação contida nos 7 métodos de identificação de fraude em várias etapas de detecção, aumentando ainda mais a evidência de transgressão nos pares finais do processo.

Por fim, a aplicação do método hierárquico em dados reais, ENEM de 2018 para Teresina-PI, demonstrou a eficiência e eficácia em descobrir possíveis fraudes no exame, indicando que os pares finais tiveram evidência de cola nas quatro áreas do exame, tendo a última etapa ou área (Matemática e suas Tecnologias) um nível de significância do teste bastante baixo ($\alpha = 0,001$), aumentando ainda mais a suspeita de transgressão ao exame

de tais pares. Em virtude disso, esse método servirá de base para diversos estudos que possam tornar possível a identificação de transgressões em avaliações em larga escala.

5.1 Trabalhos Futuros

Recomenda-se para trabalho futuro a otimização do pacote *TestFraud*, que consiste em fundir o método hierárquico com a seleção quantílica [11], pois ter-se-ia menos pares a serem analisados devido aos dois processos de eliminação, este por nível de proficiência e aquele por etapas de filtragens. Nesse sentido, será possível aplicar os métodos estatísticos de detecção de fraude em testes para o ENEM de todo o Brasil.

Referências Bibliográficas

- [1] ANDRADE, D. F.; TAVARES, H. R.; VALLE, R. C . *Teoria da Resposta ao Item: conceitos e aplicações*. ABE, São Paulo, 2000.
- [2] AITKIN M. BOCK, R. D. Marginal maximum likelihood estimation of item parameters: An application of a em algorithm. 46:433–459, 1981.
- [3] BOCK, R. D. *Estimating item parameters and latent ability when responses are scored in two or more nominal categories*. *Psychometrika*, 37(1):29–51, 1972.
- [4] BOLFARINE, H. E SANDOVAL, M. C. *Introdução à Inferência Estatística*. 2ª edição. Rio de Janeiro: Sociedade Brasileira de Matemática., 2010.
- [5] BRASIL. *Decreto-Lei 2.848, de 07 de dezembro de 1940. Código Penal*. Diário Oficial da União, Rio de Janeiro. 31 dez. 1940.
- [6] BUSSAB, W. O. MORETTIN, G de A. *Estatística Básica*. Ed Saraiva. 8ª Edição. Ed Saraiva., 2016.
- [7] CAED - Centro de Políticas Públicas e Avaliação da Educação, 2008. *O que é avaliação educacional?*. Disponível em: <http://www.portalavaliacao.caedufjf.net/pagina-exemplo/o-que-e-avaliacao-educacional/>. Acesso em: 20 dez. 2018.
- [8] CIZEK, G. J.; WOLLACK, J. A. *Handbook of quantitative methods for detecting cheating on tests*. Routledge New York, NY, 2017.
- [9] HOLLAND, P. W. *Assessing unusual agreement between the incorrect answers of two examinees using the K-index: Statistical theory and empirical support*. *ETS Research Report Series*, 1996(1):i–41, 1996.
- [10] MERSMANN, O. *microbenchmark: Accurate Timing Functions*, 2018. R package version 1.4-6.

-
- [11] MEZA, R. O. *Seleção quantílica no pacote TestFraud para detecção de fraudes em testes*. 2020. 42 f. Dissertação (Mestrado em Estatística) – Instituto de Ciências Exatas e Naturais, Universidade Federal de Pará, Belém.
- [12] ROBERT J MISLEVY and MARTHA L STOCKING. A consumer’s guide to logist and bilog. *Applied psychological measurement*, 13(1):57–75, 1989.
- [13] MORAES, A. N. *O estado da arte dos métodos estatísticos para detecção de fraudes em testes e aplicações*. 2019. 42 f. Dissertação (Mestrado em Estatística) – Instituto de Ciências Exatas e Naturais, Universidade Federal de Pará, Belém.
- [14] SOTARIDONA, L. S.; MEIJER, R. R. *Statistical properties of the K-index for detecting answer copying*. *Journal of Educational Measurement*, 39(2):115–132, 2002.
- [15] SOTARIDONA, L. S.; MEIJER, R. R. *Two new statistics to detect answer copying*. *Journal of Educational Measurement*, 40(1):53–69, 2003.
- [16] SOUZA, M. M. *Implementação e otimização do pacote TestFraud para detecção de fraude em testes*. 2019. 42 f. Dissertação (Mestrado em Estatística) – Instituto de Ciências Exatas e Naturais, Universidade Federal de Pará, Belém.
- [17] VAN DER LINDEN; WIM J.; SOTARIDONA, L. *Detecting answer copying when the regular response process follows a known response model*. *Journal of Educational and Behavioral Statistics*, 31(3):283–304, 2006.
- [18] STEVE WESTON and RICH CALAWAY. Getting started with doparallel and foreach. 2019.
- [19] WOLLACK, J. A. *A nominal response model approach for detecting answer copying*. *Applied Psychological Measurement*, 21(4):307–320, 1997.
- [20] ZOPLUOGLU, C. *CopyDetect: An R package for computing statistical indices to detect answer copying on multiple-choice examinations*. *Applied psychological measurement*, 37(1):93–95, 2013.
- [21] ZOPLUOGLU, C.; CIZEK, G. J.; WOLLACK, J. A. *Similarity, answer copying, and aberrance: Understanding the status quo*. CIZEK, G. J.; WOLLACK, J. A., “*Handbook of quantitative methods for detecting cheating on tests*,” New York, NY: Routledge, pages 25–46, 2017.

Apêndice A

Algoritmo para análise da taxa de falso positivo

```
#####  
##### Taxa de Falso Positivo  
#####  
#####  
  
#####  
#####  
##### C lculo taxa da falso positivo por ndice #####  
#####  
#####  
pares=read.csv("pares.csv", header = TRUE,dec = ".") ### base de pares  
pares_indices=pares[,4:10] #### colunas de p-valores para cada ndice  
alpha=c(0.001,0.005,0.01,0.02,0.05) #### alpha adotados  
tfp=matrix(0,length(alpha),ncol(pares_indices)) ### matriz de Falso  
positivo  
rownames(tfp)=alpha ### nome das linhas  
colnames(tfp)=c("omega","GBT","K","K1","K2","S1","S2") ### nome das  
colunas  
  
for (i in 1:length(alpha)){  
  
matrix_ind=matrix(0,nrow(pares_indices),7) ### matriz de indicadores  
"0" ou "1"  
for (z in 1:nrow(pares_indices)) {  
for (w in 1:ncol(pares_indices)) {  
if (pares_indices[z,w] < alpha[i]) {  
matrix_ind[z,w] = 1} else {  
matrix_ind[z,w] = 0}  
}  
}  
  
vetor=matrix(0,1,ncol(matrix_ind)) ### soma das colunas  
for (v in 1:ncol(matrix_ind)){  
vetor[v]=sum(matrix_ind[,v])  
}  
  
tfp[i,]=vetor/nrow(pares_indices)
```

```

}

#####
##### Gráfico taxa de falso positivo por ndice #####
#####
omega= tfp[,1]
GBT= tfp[,2]
K= tfp[,3]
K1= tfp[,4]
K2= tfp[,5]
S1= tfp[,6]
S2= tfp[,7]

plot(c(0,0.06),c(0,0.05),type="n",xlab=NA,ylab=NA,xlim=c(0,0.05),ylim=c
(0,0.05))
lines(alpha,alpha,type="b",col=1,lwd=3,pch=1) #esperado
lines(alpha,omega,type="b",col=2,lwd=2,pch=2)
lines(alpha,GBT,type="b",col=3,lwd=2,pch=3)
lines(alpha,K,type="b",col=4,lwd=2,pch=4)
lines(alpha,K1,type="b",col=5,lwd=2,pch=5)
lines(alpha,K2,type="b",col=6,lwd=2,pch=6)
lines(alpha,S1,type="b",col=7,lwd=2,pch=7)
lines(alpha,S2,type="b",col=8,lwd=2,pch=8)
title("J=5000_eI=45",xlab=expression(alpha),ylab=expression(italic("
Erro_tipo_I")))
legend(0,0.053,c(expression(italic(esperado)),expression(omega),
expression(italic(GBT)),
expression(italic(K)),expression(italic(K[1])),
expression(K[2]),expression(italic(S[1])),expression(
italic(S[2]))),
col=c(1,2,3,4,5,6,7,8),pch=c(1,2,3,4,5,6,7,8),lwd=1,bty="n")
#####
#####
##### Cálculo do EQM#####
#####
mdiff=matrix(0,length(alpha),ncol(tfp)) ### matriz de diferen a ao
quadrado
EQM=matrix(0,1,ncol(tfp)) #### verto com os EQMs de cada ndice
rownames(EQM)=c("estimativa") ### nome da linha
colnames(EQM)=c(expression(omega),"GBT","K","K1","K2","S1","S2") ###
nome das colunas

for (z in 1:ncol(tfp)){
for (i in 1:length(alpha)){
for (j in 1:ncol(tfp)){
mdiff[i,j]=(alpha[i]-tfp[i,j])^2
}
}
}

```

```
EQM[z]=sum(mdiff[,z])/length(alpha)
}
#####
#####
##### Gráfico do EQM#####
#####
barplot(EQM,xlab="Índices",ylab="Erro quadrático",
        main="Valores de EQM",ylim=c(0,max(EQM)),col="blue")
```