Aplicação de Algoritmos de Clusterização Baseados em Otimização por Enxame de Partículas em Bases de Dados Educacionais

Trabalho de Conclusão de Curso
Engenharia da Computação

Aluno: Pedro José Buarque Lins do Santos
Orientador: Prof. Dr. Carmelo José Albanez Bastos Filho

Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação

Pedro José Buarque Lins dos Santos

APLICAÇÃO DE ALGORITMOS DE CLUSTERIZAÇÃO BASEADOS EM OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS EM BASES DE DADOS EDUCACIONAIS

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco - Universidade de Pernambuco.

Recife, outubro de 2017

ESCOLA POLITÉCNICA
DE PERNAMBUCO

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 27 de novembro de 2017, às 8:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **PEDRO JOSE BUARQUE LINS DOS SANTOS**, orientado pelo professor **Carmelo José Albanez Bastos Filho**, sob título **Aplicação de Algoritmos de Clusterização Baseados em Otimização por Enxame de Partículas em Bases de Dados Educacionais**, a banca composta pelos professores:

Alexandre Magno Maciel
Carmelo José Albanez Bastos Filho

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

☒ Aprovada ☐ Aprovada com Restrições* ☐ Reprovada

e foi-lhe atribuída nota: 10 (DEZ)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá OITO dias para entrega da versão final da monografia a contar da data deste documento.

ALEXANDRE MAGNO MACIEL

CARMELO JOSÉ ALBANEZ BASTOS FILHO

* Este documento deverá ser encadernado juntamente com a monografia em versão final.

ESCOLA POLITÉCNICA
DE PERNAMBUCO

Autorização de publicação de PFC

Eu, **PEDRO JOSE BUARQUE LINS DOS SANTOS** autor do projeto de final de curso intitulado: **Aplicação de Algoritmos de Clusterização Baseados em Otimização por Enxame de Partículas em Bases de Dados Educacionais;** autorizo a publicação de seu conteúdo na internet nos portais da Escola Politécnica de Pernambuco e Universidade de Pernambuco.

O conteúdo do projeto de final de curso é de responsabilidade do autor.

Pedro José Buarque Lins dos Santos

PEDRO JOSE BUARQUE LINS DOS SANTOS

Carmelo José Albanez Bastos Filho

Orientador(a) Carmelo José Albanez Bastos Filho

coorientador(a)

Professor de TCC Sérgio Campello Oliveira

Data

Alguns homens vêem as coisas como são, e dizem 'Por quê?' Eu sonho com as coisas que nunca foram e digo 'Por que não?'. - **George Bernard Shaw**

Agradecimentos

Agradeço por todo o conhecimento adquirido ao longo desses árduos cinco anos em que estudei na Escola Politécnica de Pernambuco, que permitiram o meu desenvolvimento acadêmico e profissional, me capacitando a ter uma visão mais crítica do mundo e de utilizar meu conhecimento para melhorá-lo.

Agradeço a todo o corpo docente da Escola Politécnica de Pernambuco principalmente ao Prof. Carmelo José Albanez Bastos Filho pela inspiração e dedicação excepcional como professor orientador deste trabalho, e a Mariana G. da M. Macedo, Hugo Siqueira e Clodomir J. Santana Jr. pelos inesquecíveis momentos em nosso grupo de pesquisa e grande amizade, sempre dando o melhor de si para alcançar os melhores resultados.

Agradeço a todos os meus queridos amigos, que tive a oportunidade de conhecer ao longo desses anos na Universidade de Pernambuco, Bettina Cavalcanti, Gabriel Chamie, Jonas Cordeiro, Daniel de França Figueroa, Juliana Araújo, Pedro Queiroz, Raul Barreto, Artur Vasconcelos, Cristiano Silva, Fernando Baptisella, Caio Albuquerque, Larissa Moura, Matheus Henrique, meus queridos Batutinhas, por todos os bons momentos que passamos juntos, e por mais que venhamos a seguir caminhos distintos, possamos sempre encontrar um no outro o conforto de um amigo.

Agradeço a minha mãe, Maria de Fátima Buarque Lins dos Santos que foi sempre presente, e que me apoiou em todos os momentos difíceis, fazendo todos os sacrifícios necessários para que eu pudesse ter uma educaçao de qualidade para enfrentar os desafios dessa vida. Agradeço ao meu pai, Manoel Francisco dos Santos por ser uma grande inspiração e exemplo a ser seguido, e que embora tenha partido para o plano espiritual, sempre me guiou e me instruiu em vida, a ser um homem íntegro e honesto.

Concluo agradecendo mais uma vez a todas as pessoas que direta ou indiretamente auxiliariam na construção deste trabalho, e contribuiram para o meu crescimento pessoal e profissional, me ajudando a ser uma pessoa melhor.

Resumo

Um desafio das plataformas de educação online é acompanhar as dificuldades dos alunos. A tarefa de agrupamento de dados permite a detecção de padrões, encontrando assim, dificuldades comuns em grupos de estudantes. Este trabalho utilizou os algoritmos de agrupamento baseados em meta-heurísticas: *Particle Swarm Clustering* (*PSC*), *Particle Swarm Optimization for Clustering* (*PSOC*), e técnicas híbridas entre os algoritmos *K-Means* e *Particle Swarm Optimization, (KMPSOC, PSOKM)* para realizar o agrupamento em bases educacionais. A análise e determinação dos grupos foi realizada a partir das seguintes métricas: Distância *Intra-Cluster*, Distância *Inter-Cluster*, Erro Quantizado e a Estatística *Gap*. Dentre as técnicas mencionadas, o algoritmo *Particle Swarm Optimization for Clustering* (*PSOC*) se destacou por ser o mais leve, possuir maior velocidade de convergência e apresentar os menores valores para desvio padrão, o que mostra um confiança do algoritmo ao reportar resultados.

Palavras-Chave: Agrupamento de dados, Distância *Intra-Cluster*, Distância *Inter-Cluster*, Erro Quantizado, Estatística *Gap*, Padrões nos dados, Meta-heurísticas.

Abstract

A challenge of online educational platforms is the accompaniment of students difficulties. The clustering task allow us to detect patterns on data, thus bringing together common difficulties on groups of students. This work uses the following clustering algorithms based on metaheuristics: *Particle Swarm Clustering* (*PSC*), *Particle Swarm Optimization for Clustering* (*PSOC*) and hybrid techniques using *K-Means* and *Particle Swarm Optimization, (KMPSOC, PSOKM)* to perform the clustering task on educational databases. The analysis and subsequent definition of the number of groups, was made using the following metrics: *Intra-Cluster Distance*, *Inter-Cluster Distance*, Quantization Error, and the *Gap* Statistic. Among the mentioned techniques, the algorithm *Particle Swarm Optimization for Clustering* (*PSOC*) was highlighted as being the lightest, having the fastest convergence speed and presenting the smallest values for standard deviation, which shows a higher confidence of the algorithm when reporting results.

Keywords: Clustering, *Intra-Cluster Distance*, *Inter-Cluster Distance*, Quantization Error, *Gap* Statistic, Patterns on data, Metaheuristics.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Motivação e caracterização do problema

Com o constante avanço das tecnologias na era da informação, o grande volume de dados gerados a todo momento, dificulta a percepção de informação. O uso de técnicas sofisticadas se faz necessário para a extração do conhecimento. Com isso, o aprendizado é facilitado, evitando que informações relevantes, que muitas vezes poderiam ter um papel importante na tomada de decisão, não passem despercebidas [1].

Dentre as diversas tarefas utilizadas no processo de descoberta do conhecimento, o agrupamento de dados (do inglês *clustering*) é bastante utilizado para desvendar a organização de padrões existentes nos dados através de grupos (do inglês *clusters*) consistentes. Por se tratar de uma tarefa de aprendizado não-supervisionado, a busca por padrões é baseada nas similaridades internas e nas dissimilaridades externas dos grupos encontrados [2]. Devido à natureza complexa da tarefa de agrupamento, essa atividade, muitas vezes modelada como um problema de otimização, é considerada NP-Complexo (do inglês *NP-hard*) [3].

Um cenário interessante onde é possível aplicar essas técnicas, são em sistemas de educação a distância. Esse tipo de plataforma permite que usuários tenham acesso a diversos tipos de conteúdo, podendo o estudo ser feito no ritmo individual de cada aluno, e portanto permitindo uma dinâmica mais

interessante para pessoas com tempo comprometido e impossibilitadas de se deslocar à sala de aula [4].

Um dos desafios dessas plataformas é a dificuldade de realizar o acompanhamento de cada aluno, e posteriormente indicar conteúdos de reforço para aqueles com dificuldades específicas. Várias técnicas possuem a capacidade de resolver problemas de agrupamento. No entanto, as técnicas baseadas em meta-heurísticas que se destacam são as derivadas do algoritmo Otimização por Enxame de Partículas (do inglês *Particle Swarm Optimization*). Uma vez que já se mostraram capazes de alcançar soluções refinadas para esse problema, sendo portanto mais eficazes do que técnicas tradicionais [5].

O projeto se propõe a utilizar de técnicas de agrupamento de dados, baseadas em Otimização por Enxames de Partículas para detecção de grupos homogêneos em bases educacionais com o objetivo de identificar dificuldades de alunos, e assim recomendar conteúdos específicos para o crescimento individual de cada aluno num sistema de educação a distância [6].

1.2 Objetivo

Este trabalho tem como objetivo principal implementar e comparar técnicas de agrupamento de dados em bases de dados de uma plataforma de educação a distância. Serão implementadas as seguintes técnicas baseadas em meta-heurísticas: *Particle Swarm Clustering* (*PSC*), *Particle Swarm Optimization for Clustering* (*PSOC*), e técnicas híbridas entre os algoritmos *K-Means* e *Particle Swarm Optimization* [7] [8] [9]. Simultaneamente, as mesmas serão comparadas com o algoritmo tradicional *K-Means*. Ademais, será realizada uma análise sobre o desempenho dos algoritmos que destacará o algoritmo mais apropriado ao problema proposto. Posteriormente, com a análise do algoritmo vencedor, será feito o estudo dos padrões, das similaridades e das dissimilaridades encontradas em cada grupo para uma posterior recomendação de conteúdos apropriados.

1.2.1 Objetivo específico

Para que seja possível atingir os objetivos principais, as seguintes metas foram cumpridas :

- Implementar as técnicas de agrupamento de dados baseadas na meta-heurísticas *PSO* na linguagem de programação Python;
- Implementar a técnicas de agrupamento de dados tradicional *K-Means* para comparação, também na linguagem Python;
- Pesquisar, implementar, aplicar e analisar métricas apropriadas para a determinação da escolha do número de grupos que melhor divide os estudantes na base de dados educacional;
- Identificar as principais características (variáveis) que definem os estudantes pertencentes aos diferentes grupos encontrados.

1.3 Estrutura da monografia

Esta monografia está dividida em 6 capítulos. No Capítulo 2, será realizada uma revisão sobre o processo de descoberta de conhecimento (do inglês, *Knowledge Discovery in Databases, KDD*) e suas principais etapas. No Capítulo 3 uma breve revisão sobre o problema de agrupamento de dados, juntamente com uma análise do algoritmo *K-Means*. Posteriormente, no Capítulo 4, será feita uma revisão da literatura sobre Inteligência de Enxames, juntamente com a apresentação do algoritmo Otimização por Enxames de Partículas e os algoritmos de agrupamento de dados dele derivados. No Capítulo 5 são apresentados os experimentos realizados, bem como uma análise estatística dos resultados. Concluindo este trabalho no Capítulo 6, com as considerações finais e propostas para trabalhos futuros.

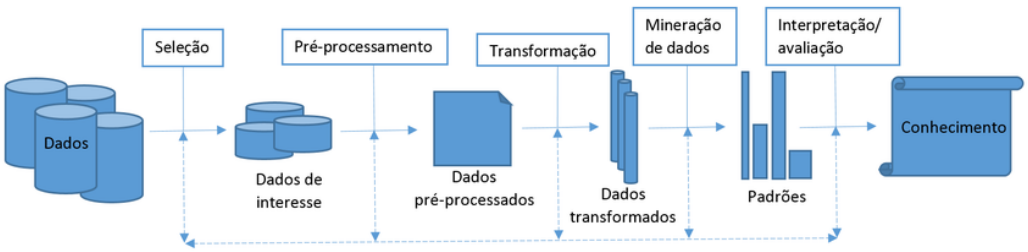
Capítulo 2

Análise de Dados

2.1 Processo de Descoberta do Conhecimento

Métodos tradicionais que visam a transformação de dados em conhecimento dependem de uma extensa análise e interpretação manual. Com o crescimento constante no volume de dados gerados diaramente na era da informação, essa metodologia se mostra lenta, cara e altamente subjetiva. O processo de descoberta do conhecimento (do inglês *Knowledge Discovery in Databases, KDD*) propõe uma análise sistemática de grandes volumes de dados, se mostrando eficaz na resolução do problema de sobrecarga de informação [10] [11] [12]. O processo de *KDD* se divide em 5 etapas que podem ser visualizadas na Figura 2.1. As próximas subseções se dedicam a explicar com mais detalhes cada uma dessas etapas.

Figura 2.1: Etapas pertinentes ao processo de descoberta do conhecimento (do inglês *Knowledge Discovery in Databases, KDD*).



2.1.1 Seleção de Dados

Na primeira etapa, chamada de seleção de dados, é escolhido o conjunto de dados pertinente ao domínio do problema. Esse conjunto deve ser escolhido de forma a conter todas a variáveis (comumente chamadas de atributos, do inglês *features*) inicialmente consideradas relevantes acerca da natureza do problema proposto. O processo de seleção normalmente é executado com o auxílio de um especialista da área.

2.1.2 Pré-Processamento de Dados

Posterior à etapa de seleção, o pré-processamento de dados busca remover valores atípicos (do inglês *outliers*) e tendências errôneas. Para isso o auxílio do especialista se mostra fundamental, pois sua perícia no domínio do problema, juntamente com a análise estátistica dos dados, ajudarão a indicar se um dado é um *outlier* ou não. Essa etapa é mostra importante por se preocupar na localização de inconsistências e tratamento de valores ausentes, removendo assim qualquer tipo de informação desnecessária ou redundante, que acabam por prejudicar os resultados das etapas posteriores.

2.1.3 Transformação de Dados

Na etapa de transformação de dados, preocupa-se encontrar uma melhor representação dos dados para que posteriormente, possam ser fornecidos como entrada para a etapa de mineração. Como os algoritmos de aprendizagem ficam inviabilizados de processar dados em formato não númerico, são aplicadas técnicas para conversão de atributos em valores númericos, normalização dos dados e redução de váriaveis. Além disso, condensa-se nessa etapa, a geração de atributos derivados, que são oriundos de operações realizadas entre uma ou mais variáveis da base de dados original.

2.1.4 Mineração de Dados

Na etapa de mineração de dados (do inglês *Data Mining*) contempla-se a exploração e análise da base de dados de forma automática, com o objetivo

de encontrar padrões ou regras nos dados [13]. É nessa etapa onde se aplicam as técnicas inteligentes para extração do conhecimento. Apesar de todas as etapas serem cruciais na extração do conhecimento, a etapa de mineração de dados é a que mais recebe destaque na literatura e normalmente envolve seis classes de tarefas [12]. Essas tarefas são:

- Classificação (*Classification*): É uma tarefa de aprendizado supervisionado, que consiste em aprender uma função (*target function*) capaz de mapear cada uma das instâncias de entrada à umas das classes de saída pré-definidas.

- Regressão (*Regression*): Similar à classificação, também consiste de um problema supervisionado, porém diferente de encontrar uma função que mapeia as intâncias em diferentes classes, o objetivo da regressão é mapeia as entradas em saídas pertencentes ao conjunto dos números reais.

- Agrupamento (*Clustering*): O problema de agrupamento de dados é de natureza não-supervisionada, sendo uma tarefa comumente descritiva cujo objetivo é identificar um número finito de grupos (comumente chamados de *clusters*), de forma que os elementos de cada grupo possuam características em comum.

- Sumarização (*Summarization*): Envolve um conjunto de técnicas para encontrar uma descrição compacta do dados. Normalmente inclui mecanismos para visualização e geração de relatórios.

- Modelagem Dependente (*Dependency Modeling*): O objetivo desta tarefa é detectar dependências ou extrair regras entre as variáveis. Existe em dois níveis: A nível estrutural, que especifica, frequentemente de forma gráfica, como duas ou mais variáveis se relacionam, e no aspecto quantitativo, onde a intensidade das relações é medida através de alguma escala numérica [12].

- Detecção e Mudança de Desvios (*Change and Deviation Detection*): Envolve a busca de anomalias, valores normativos e instâncias incomuns

para uma posterior investigação.

2.1.5 Interpretação de Dados

A fase de interpretação conclui o processo de *KDD*, que normalmente é realizado com o auxílio de um especialista do domínio. Nesta etapa os resultados obtidos no processo de mineração são analisados e verifica-se se eles fazem sentido. Caso o resultado obtido não seja satisfatório, pode-se retornar à qualquer uma das etapas anteriores ou inclusive recomeçar todo o processo. Normalmente quando se tem um resultado insatisfatório, são feitas novas mudanças nos dados, troca-se o algoritmo de aprendizagem da etapa de mineração ou ajusta-se os seus hiperparâmetros.

Capítulo 3

Agrupamento de Dados

3.1 O Problema do Agrupamento de Dados

O Agrupamento de dados é um problema de natureza não supervisionada, onde o objetivo principal é encontrar padrões nos dados. O padrões encontrados durante o processo de agrupamento são codificados em grupos (do inglês *clusters*). Os algoritmos de agrupamento objetivam construir os grupos obedecendo a regra de que grupos distintos devem possuir uma alta dissimilaridade externa, e instâncias pertencentes a um mesmo grupo possuam um alta similaridade interna. Por se tratar de uma tarefa de aprendizagem não supervisionada, não existem classes conhecidas a *priori*. Portanto, avaliar o resultado de algoritmos de agrupamento é uma tarefa bastante complexa e requer um especialista no domínio no problema para validar se os resultados fazem sentido.

3.2 Componentes da Tarefa de Agrupamento

A busca de padrões na tarefa de agrupamento de dados envolve uma metodologia sistemática que pode ser dividida nas seguintes etapas [14]:

- **Representação dos Padrões** (*Pattern Representation*): Inclui as etapas de extração ou seleção de atributos, definição a *priori* do número de grupos, e normalização dos dados.

- **Definição de uma Medida de Similaridade** (*Pattern Similarity Measure*): Normalmente é utilizada uma medida de distância apropriada ao domínio do problema que caracteriza semelhança entre duas instâncias.

- **Agrupamento dos Dados** (*Clustering*): Aplicação do algoritmo de agrupamento. As estratégias para essa tarefa podem incluir algoritmos com diferentes abordagens, como agrupamento clássico, agrupamento difuso, agrupamento hierárquico, agrupamento por partição, entre outros.

- **Abstração dos Dados** (*Data Abstraction*): Consite no processo de se obter uma representaccão simplificada dos dados para análise, no caso da tarefa de agrupamento, um maneira bastante comum é obter uma descrição da natureza de cada grupo.

- **Análise dos Grupos** (*Assessment of Output*): Etapa de validação da saída do algoritmo de agrupamento. Nesta etapa se faz um estudo da validade dos grupos e verifica-se se o agrupamento foi capaz de extrair alguma informação dos dados. Esta etapa requer o auxílio de um especialista.

3.3 Técnicas de Agrupamento de Dados

Como mencionado na seção 3.2, somente na fase de agrupamento é feita a escolha e aplicação do algoritmo. O problema de agrupamento pode ser interpretado de diversas maneiras por que o significado de *cluster* não pode ser precisamente definido [15]. Cada algoritmo utiliza técnicas diferentes para realizar a extração dos grupos, que são divididos taxonomicamente de acordo com a literatura como segue [14]:

- **Aglomerativo ou Divisionista** (*Agglomerative or Divisive*): A estratégia aglomerativa considerada cada instância individual como um único grupo, posteriomente utiliza medidas para unir grupos considerados semelhantes até um determinado critério de parada ser atingido.

Diferentemente, a abordagem divisionista começa considerando a base como um único grupo, posteriormente utiliza técnincas para particionar em grupos menores até um determinado critério de parada.

- **Monotético ou Politético** (*Monothetic or Polythetic*): Está associado ao uso sequencial ou simultâneo dos atributos. Em algoritmos que utilizam a abordagem monotética os atributos são utilizados de forma sequencial para dividir os dados. Algoritmos politéticos utilizam atributos de forma simultânea, por exemplo, o algoritmos *K-Means*, que utiliza todos os atributos para computar distâncias euclidianas e dividir os dados.

- **Clássico ou *Fuzzy*** (*Hard or Fuzzy:*): Algoritmos clássicos particionam o conjunto de dados em grupos disjuntos, ou seja, uma dada instância pertence a um único grupo. Na abordagem *Fuzzy* cada instância possui um grau de pertinência a cada um dos grupos. A metodologia *fuzzy* pode ser convertida para a clássica atribuindo-se cada instância para o grupo com maior grau de pertinência.

- **Deteminístico ou Estocástico** (*Deterministic or Stochastic*): Os algoritmos determinista utilizam medidas analíticas para minimizar a função de erro utilizada para separar os dados. Em contrapartida, algoritmos estocátiscos utilizam valores aleatórios dentro de um espaço de buscas pré-definido.

- **Incremental ou Não-Incremental** (*Incremental or Non-Incremental:*): Algoritmos de agrupamento que utilizam a abordagem incremental utilizam uma instância por vez. Essas instâncias normalmente são processadas em pequenas quantidade constantes (*batch*). Diferentemente os modelos não-incrementais trabalham com o lote completo de dados de um só vez.

3.4 *K-Means*

O algortimo *K-Means* é frequentemente utilizado na literatura de agrupamento de dados, sempre sendo utilizado como base de comparação com outros algoritmos. Trata-se de um algoritmo de agrupamente genérico bastante simples, porém bastante eficiente e de baixo custo computacional, tornando-o bastante popular. Possui bom desempenho em grandes bases de dados e possui bons resultados em diversas aplicações práticas [16] [17] [18] [19].

Para solucionar o problema de agrupamento o algoritmo *K-Means* utiliza uma função de similiaridade para agrupar instâncias semelhantes. A similaridade entre duas instância é muitas vezes mensurada através de uma função de distância. A versão clássica do algoritmo *K-Means* utiliza como função de similaridade a função de distância euclidiana. Para mensurar a distância euclidiana entre duas instâncias x_i e x_j, ambas de dimensão d, utiliza-se a Equação (3.1)

$$d(x_i, x_j) = \sqrt{\sum_{p=1}^{d}(x_{i,p} - x_{j,p})^2}. \tag{3.1}$$

O algoritmo *K-Means* pode ser visto com mais detalhes no **Algoritmo 1**. Primeiramente o algoritmo inicializa cada um dos k centróides c_k de forma aleatória, em seguida cada instância x_i é adicionada ao grupo cuja distância ao centróide c_k é mínima. Uma das etapas principais do algoritmo é o recálculo dos centróides de acordo com a Equação (3.2)

$$c_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x_i^k. \tag{3.2}$$

A execução do algoritmo chega ao fim quando um critério de parada é atingido. O critério de parada normalmente é modelado como uma função de erro. Diferentes implementações do *K-Means* utilizam diferentes critérios de parada, como por exemplo estagnação de centróides ou quando nenhuma instância x_i mudar de grupo [20] [21]. A versão tradicional do algortimo *K-Means* utiliza como função a ser minimizada a Equação (3.3), que representa a soma dos quadrados das distâncias entre cada uma das instâncias x_i e seus

respectivos centróides.

$$\epsilon = \sum_{k=1}^{K} \sum_{i=1}^{n_k} |x_i^k - c_k|^2. \tag{3.3}$$

O algoritmo *K-Means* utilizado neste projeto foi implementado utilizando a linguagem de programação *Python*. Diferentemente da versão clássica, o algoritmo implementado utilizou a estratégia de medóides, que inicializa os centróides utilizando k instâncias aleatórias x_i dentro da base de dados. Essa escolha foi feita considerando que uma das desvantagens da inicialização clássica de centróides aleatórios é mais suscetível à geração de grupos vazios.

Algoritmo 1: Pseudo-Código do *K-means*.

Entrada: Base com i instâncias e d dimensões, número k de grupos;
Saída: Base particionada em k grupos;

```
início
    Inicialização aleatória de cada centroíde c_k;
    repita
        para cada x_i na base de dados faça
            Atribuir x_i ao grupo cuja distância ao centróide c_k é
             mínima;
        fim
        Atualização dos centróides c_k de acordo com a equação 3.2;
    até critério de parada ser atingido;
fim
```

3.5 Métricas

Tendo a abordagem divisionista, não-incremetal e particionista como escolha, uma das preocupaccões principais durante a aplicação de algoritmos de agupamento de dados é a escolha de métricas que permitem definir qual o valor mais apropriado para o número k de partições à serem feitas no conjunto inicial de dados. As métricas devem também avaliar a qualidade da composição dos grupos, bem como a capacidade do mesmo em identificar os

padrões nos dados. Para este trabalho foram utilizadas as metricas *Inter-Cluster Distance*, *Intra-Cluster Distance*, Erro Quantizado e a Estatística *Gap* que são explicadas com mais detalhes nas próximas seções [22] [23].

3.5.1 *Intra* e *Inter Cluster Distance*

As métricas *Intra* e *Inter Cluster Distance* são utilizadas para validar a qualidade do algoritmo no que concerne a capacidade de encontrar regiões de alta densidade, de forma que, tais regiões sejam distantes entre si. A métrica *Intra Cluster Distance* avalia a composição interna dos grupos, computando a distância euclidiana entre quaiquer duas instâncias x_i e x_j. Quando aplicada nos resultados de um algoritmo de agrupamento, é possível inferir a qualidade do resultado tendo em consideração que instâncias pertencentes a um mesmo grupo devem ser próximas entre si, resultando em baixo valor para esta métrica. A métrica *Inter Cluster Distance* avalia o comportamento externo dos grupos, de forma que as regiões distintas de alta densidade sejam distantes umas das outras. Essa avaliação se dá através da soma das distâncias entre os pares de centroídes, que deve ser maximizada. Seja $d(x_i, x_j)$ a distância euclidiana entre as instâncias x_i e x_j e seja $d(c_k, c_{k'})$ a distância euclidiana entre os centroídes c_k e $c_{k'}$, as Equações (3.4) e (3.5) denotam as métricas *Intra* e *Inter Cluster Distance* respectivamente.

$$D_{intra} = \sum_{k} \frac{1}{2N_k} \sum_{x_i, x_j \in C_k} d(x_i, x_j). \tag{3.4}$$

$$D_{inter} = \sum_{\forall k, k' | k \neq k'} d(c_k, c_{k'}). \tag{3.5}$$

3.5.2 Erro Quantizado

A métrica erro quantizado é uma métrica genérica que permite acompanhar o comportamento e medir a qualidade de algoritmos de agrupamento para uma dada base de dados [24]. Embora não seja capaz de informar sobre a qualidade individual dos grupos formados, a métrica de erro quantizado permite acompanhar a eficiência do algoritmo para valores crescentes de k,

o número de grupos. Essa métrica foi primeiramente introduzida com uma função objetivo para algoritmos de clusterização baseados em otimização por enxames de partículas [25] [1]. Seja k o número de grupos, $|C_k|$ o número de elementos do k-ésimo grupo e $d(x_i, c_k)$ a distância euclidiana entre a instância x_i pertencente ao grupo C_k ao centróide c_k, a Equação (3.6) denota a métrica erro quantizado.

$$J_e = \frac{\sum_k \sum_{\forall x_i \in C_k} d(x_i, c_k)/|C_k|}{N_k}. \tag{3.6}$$

Para valores crescentes de k, o comportamento esperado é a granularização dos dados. Portanto, a medida que k aumenta, espera-se que a composição dos grupos encontrados torne-se cada vez menor. Portanto, a medida que incrementamos o valor de k, espera-se obter menores valores para a métrica de erro quantizado.

3.5.3 Estatística *Gap*

A Estatística *Gap* é considerada bastante eficiente na decisão do melhor valor para o número de grupos [26] [22]. A aplicação desta métrica é indicada quando a abordagem de agrupamento é particionista, pois existe dificuldade por parte da Estatística *Gap* em avaliar bases onde há sobreposição entre dois ou mais grupos [22].

A idéia principal da Estatística *Gap* se baseia na constatação de que, mudanças na dispersão interna dos grupos à medida que incrementa-se o número k de grupos é esperada, tendo-se com referência uma distribuição de dados aleatória [27]. Primeiramente, na Equação (3.7), soma-se as distâncias euclidianas entre quaisquer pares de instâncias x_i e x_j pertencentes ao k-ésimo grupo.

$$D_k = \sum_{\forall x_i, x_j \in C_k} d(x_i, x_j). \tag{3.7}$$

Posteriormente, utiliza-se a Equação (3.7) para se calcular a dispersão interna dos grupos para valores cresentes de k através da Equação (3.8), onde

efetua-se a divisão por dois para desconsiderar as distâncias computadas duas vezes.

$$W_k = \sum_{i=1}^{K} \frac{1}{2n_r} D_i. \tag{3.8}$$

Uma vez computada a dispersão interna dos grupos, compara-se a versão amortizada $\log W_k$ com o valor esperado de uma distribuição de dados aleatória. A Equação (3.9) resume todo esse processo inicial, onde E_n^* representa o valor esperado de uma distribuição de dados aleatória.

$$Gap_n(k) = E_n^*(\log W_k) - \log W_k. \tag{3.9}$$

O melhor valor para o número de grupos, denotado de $\hat{k}$, é encontrado maximizando a equação (3.9). Porém na prática, computar E_n^* pode não refletir o comportamento esperado. Portanto, estima-se $E_n^*(\log W_k)$ utilizando-se B amostras de W_k^* , cada uma extraída de uma amostra de dados aleatória diferente. A Equação (3.10) mostra como a métrica *Gap* é calculada na prática.

$$Gap(k) = \frac{1}{B} \sum_{b} \log W_k^* - \log W_k. \tag{3.10}$$

Portanto, seja s_k o desvio padrão das B amostras de W_k^*, pode-se estimar o valor ótimo para o número de grupos ao se escolher o menor valor de k que obedeçe a desigualdade:

$$Gap(k) \geq Gap(k+1) - s_{k+1}. \tag{3.11}$$

Capítulo 4

Inteligência de Enxames

4.1 Introdução à Inteligência de Enxames

Inteligência de Enxames são modelos computacionais inspirados em sistemas de enxames naturais. Muitos desses modelos propostos na literatura já foram aplicados com sucesso em muitos problemas do mundo real [28]. As técnicas baseadas em Inteligência de Enxames tem maior destaque na resolução de problemas complexos onde os algoritmos tradicionais não demonstram ser capazes de trazer uma solução satisfatória.

Para a técnicas computacionais tradicionais os problemas precisam ser bem definidos, resultados devem ser previsíveis e uma solução para o problema deve ser encontrada num intervalo de tempo finito utilizando-se os computadores atuais. Pelo fato de muitos problemas do mundo real não serem capazes de atingir tais requisitos, as técnicas de Inteligência de Enxames propões mecanismos de busca genéricos para encontrar soluções bastante próximas da solução ótima em um intervalo de tempo bem menor do que os algoritmos tradicionais [28].

Embora os algoritmos baseados em Inteligência de Enxames compartilhem uma estrutura similar ao tradicionais, eles diferem no mecanismo de busca. É através deste mecanismo que os agentes constituintes do algoritmo se deslocarão pelo espaço de busca defindo pelo problema, para procurar por uma solução ótima. Dentre os vários algoritmos baseados em enxames,

destacam-se Otimização por Colônia de Formigas (do inglês *Ant Colony Optimization*) [29], Colônia Artificial de Abelhas (do inglês *Artificial Bee Colony*) [30], Sistemas Imunológicos Artificiais (do inglês *Artificial Immune System*) [31] e Otimização por Enxames de Partículas (do inglês *Particle Swarm Optimization*) [32], sendo o último o foco primário desta obra e será explicado com mais detalhes na próxima seção.

4.2 Otimização por Enxames de Partículas

Otimização por Enxames de Particulas, frequentemente chamado apenas de *PSO* pela literatura, é uma metaheurística baseada em Inteligência de Enxames e inspirada no comportamento de grupos de animais, como por exemplo, cardume de peixes e bando de aves. Foi proposto inicialmente em 1995 por Eberhart e Kennedy [32], sendo subsequentemente utilizado com sucesso em vários problemas de busca e otimização.

No *PSO*, as entidades responsáveis pela busca são denominadas de partículas e constituem o conjunto $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ de soluções candidatas para o problema de otimização. Cada partícula p_i está associada a uma posição $\mathbf{x}_i^t$ e a uma velocidade $\mathbf{v}_i^t$ na iteração t. As partículas se deslocam no espaço de busca utilizando conhecimento prévio de suas posições, bem como o conhecimento global do enxame. A cada iteração cada uma das partículas é avaliada de acordo com uma função objetivo especificada pelo problema. A posição e velocidade de cada partícula se movendo num espaço D-dimensional obedecem as equações (4.1) e (4.2):

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}, \tag{4.1}$$

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_1\mathbf{r_1} \cdot (\mathbf{pbest}_i^t - \mathbf{x}_t^i) + c_2\mathbf{r_2} \cdot (\mathbf{gbest}^t - \mathbf{x}_i^t). \tag{4.2}$$

Onde t denota a iteração atual, w denota o peso inercial do algoritmo *PSO*, $\mathbf{r_1}$ e $\mathbf{r_2}$ são vetores de várias aleatórias exraídas de uma distribuição norma no intervalo $[0, 1]$. As constantes $\mathbf{c_1}$ e $\mathbf{c_2}$ são, respectivamente, os

coeficientes cognitivo e social, $\mathbf{pbest}_i^t$ representa a melhor posição avaliada pela partícula $\mathbf{p}_i$ até o momento da iteração t, e $\mathbf{gbest}^t$ representa a melhor posição encontrada até a iteração t. O peso inercial w é utilizado para evitar um fenômeno chamado de *particle explosion*, que pode acarretar em algumas partículas escapando do espaço de buscas. As constantes $\mathbf{c_1}$ e $\mathbf{c_2}$ podem ser melhor calibradas para cada problema, entretanto pesquisas indicam que bons resultados são obtidos ao se utilizar o valor padrão 2,05 para ambas [32]. Também é comum limitar a velocidade das partículas utilizando uma velocidade máxima v_{max}. O **Algoritmo 2** resume com mais detalhes o funcionamente interno do algoritmo PSO.

Algoritmo 2: Pseudo-Código do *PSO*.

```
1  início
2      Inicialize aleatoriamente as posições e velocidades de cada
         partícula p_i;
3      Avalie e atualize pbest_i de cada partícula p_i;
4      Atualize gbest;
5      repita
6          para cada partícula p_i no exame faça
7              Atualize a velocidade v_i da da partícula p_i de acordo com
                 a Equação (4.2);
8              Atualize a posição x_i da da partícula p_i de acordo com a
                 Equação (4.1);
9          fim
10         para cada partícula p_i no exame faça
11             Avalie e atualize pbest_i de cada partícula p_i;
12         fim
13         Atualize gbest;
14     até t ≤ t_max ou critétio de parada ser atingido;
15 fim
```

4.3 Abordando o Problema de Agrupamento

Devido a sua estrutura genérica os algoritmos de otimização baseados em Inteligência de Enxames podem ser adaptados para resolução de diversos problemas. Nas próximas seções mostra-se como é possível adaptar o algoritmo *PSO* para resolver o problema de agrupamento de dados.

4.3.1 *PSOC*

Como foi visto na Seção 4.2, cada partícula $\mathbf{p}_i$ do algoritmo *PSO* é codificada num espaço D-dimensional. Para solucionar o problema de agrupamento de dados, o algoritmo *Particle Swarm Optimization for Clustering* (*PSOC*) codifica cada partícula como uma solução do problema de agrupamento. Para a abordagem particionista do problema de agrupamento, uma solução é definida como um conjunto $\{c_1, c_2, ..., c_k\}$, formado pelos centroídes responsáveis por definir os k grupos. Portanto, para o algoritmo *PSOC* cada particula é codificada com um conjunto contendo k centroídes candidatos para solução do problema como mostra a Equação (4.3):

$$\mathbf{x}_i = \{\mathbf{c}_{i1}, \mathbf{c}_{i2}, ..., \mathbf{c}_{ik}\}. \tag{4.3}$$

Onde $\mathbf{x}_i$ é o vetor posição da partícula $\mathbf{p}_i$ e $\mathbf{c}_{ij}$ representa o j-ésimo centróide componente da partícula $\mathbf{p}_i$. Para resolver o problema de agrupamento, o algoritmo *PSOC* utiliza com função objetivo padrão o erro quantizado, descrito pela Equação (3.6) [33]. Entretanto, nos experimentos realizados com a base de dados proposta, o uso do erro quantizado como função objetivo não forneceu bons resultados. Portanto, foi utilizada como função objetivo o *Sum of Squared Erros (SSE)*, a mesma função minimizada pelo algoritmo *K-Means* e descrita pela Equação (3.3) [6]. Ademais, o algoritmo *PSOC* é bastante similar a versão original do algoritmo *PSO*.

4.3.2 *PSC*

O Algoritmo *Particle Swarm for Clustering* (*PSC*) apresenta uma codificação diferente das partículas em relação ao *PSOC* para o problema de

agrupamento. Nessa abordagem, cada partícula $\mathbf{p}_i$ representa um único centróide $\mathbf{c}_k$, que navega pelo espaço de buscas até encontrar a posição ótima correspondente aos centróide das regiões de alta densidade na base de dados [1]. Como cada partícula é um único centróide, a solução para o problema neste caso, é todo o enxame.

Em um processo iterativo, cada instância $\mathbf{z}_i$ da base é apresentada a cada uma das partículas, que compõem os centróides soluções, de forma a definir a partícula mais próxima. A partícula mais próxima da instância $\mathbf{z}_i$ é definida como $\mathbf{p}^i_{win}$, a partícula vencedora para a instância $\mathbf{z}_i$. Essa partícula vencedora se desloca em direção a instância $\mathbf{z}_i$ considerando a sua melhor posição até a iteração corrente, representado por $\mathbf{p}^i_{best}$, e a melhor posição global para a instância $\mathbf{z}_i$, representado por $\mathbf{g}_{best}$. Assim, a partícula vencedora terá as coordenados de seu centróide e sua velocidade atualizadas de acordo com as equações (4.4) e (4.5) respectivamente:

$$\mathbf{c}_i^{t+1} = \mathbf{c}_i^t + \mathbf{v}_i^{t+1}, \tag{4.4}$$

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + \boldsymbol{\varphi}_1 \cdot (\mathbf{pbest}_i^t - \mathbf{c}_i^t) + \boldsymbol{\varphi}_2 \cdot (\mathbf{gbest}_i^t - \mathbf{c}_i^t) + \boldsymbol{\varphi}_3 \cdot (\mathbf{z}_i - \mathbf{c}_i^t). \tag{4.5}$$

Onde w é o peso inercial e $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2$ e $\boldsymbol{\varphi}_3$ são vetores de várias aleatórias exraídas de uma distribuição norma no intervalo $[0, 1]$ que representam respectivamente, o termo cognitivo, social e o termo auto-organizável. Durante a iteração inicial é possível que uma dada partícula $\mathbf{p}_i$ nunca seja a vencedora para nenhuma instância $\mathbf{z}_i$. Nesse caso, é preciso fazer uma validação extra para que está partícula tenha suas componentes atualizadas de forma apropriada. Para esse cenário particular, a partícula tem sua componente de velocidade atualizada utilizando uma soma ponderada das componentes da partícula que venceu mais vezes, denotada por $\mathbf{c_{most-win}}$. A Equação (4.6) descreve como a atualização da velocidade é feita para este cenário.

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + \boldsymbol{\varphi}_4(\mathbf{c_{most-win}}^t - \mathbf{c}_i^t). \tag{4.6}$$

O **Algoritmo 3** descreve com mais detallhes o funcionamento do algoritmo *PSC*.

Algoritmo 3: Pseudo-Código do *PSC*

início
  Inicialize aleatoriamente cada partícula $\mathbf{p}_i$ como um centróide $\mathbf{c}_i$;
  Inicialize aleatoriamente a velocidade $\mathbf{v}_i$ de cada partícula $\mathbf{p}_i$;
  repita
    para *cada instância* $\mathbf{z}_i$ *na base* **faça**
      Calcule a distância entre cada partícula $\mathbf{p}_i$ à $\mathbf{z}_i$;
      Defina $\mathbf{p}^i_{win}$ como a partícula de menor distância à $\mathbf{z}_i$;
      se $d(\mathbf{p}^i_{win}, \mathbf{z}_i) \leq d(\mathbf{p}^i_{best}, \mathbf{z}_i)$ **então**
        Defina $\mathbf{p}^i_{best}$ como $\mathbf{p}_{win}$;
      fim
      se $d(\mathbf{p}^i_{win}, \mathbf{z}_i) \leq d(\mathbf{p}_{best}, \mathbf{z}_i)$ **então**
        Defina $\mathbf{g}_{best}$ como $\mathbf{p}_{win}$;
      fim
      Atualize a velocidade de $\mathbf{p}_{win}$ de acordo com (4.5) ;
      Atualize a posição de $\mathbf{p}_{win}$ de acordo com (4.4) ;
    fim
    para *cada partícula* $\mathbf{p}_i$ *no exame* **faça**
      se *partícula* $\mathbf{p}_i$ *nunca venceu* **então**
        Atualize a velocidade de $\mathbf{p}_i$ de acordo com (4.6) ;
        Atualize a posição de $\mathbf{p}_i$ de acordo com (4.4) ;
      fim
    fim
    Atualize w;
  até *critétio de parada ser atingido*;
fim

4.3.3 Técnicas híbridas entre *K-Means* e *PSO*

A hibridizacão entre as técnincas *K-Means* e *Particle Swarm Optimization* é bem simples e pode se dar de duas formas. A primeira delas, é denotada de *PSOKM* [34], executa o algoritmo *K-Means* até a sua convergência. Posteriormente, a solução é utilizada como uma das partículas para o algoritmo *PSO* modificado para clustering. As demais partículas são inicializada aleatoriamente. A segunda abordagem, denotada de *KMPSOC* [8], utilizada o processo inverso do *PSOKM*. Primeiramente, o algoritmo *PSOC* é executado até a sua convergência. Finalmente a solução encontrada pelo *PSOC* é utilizada como os centróides iniciais do algoritmos *K-Means*. Nesse processo, o algoritmo *K-Means* tem grandes ganhos em sua velocidade de convergência, bem como um maior refinamento na qualidade dos centróides finais.

Capítulo 5

Experimentos e Análise de Resultados

5.1 Descrição das Bases de Dados

Visando a descoberta de perfis referentes aos erros gramaticais cometidos em bases de dados educacionais, as técnicas de agrupamento baseadas em meta-heurísticas descritas nesta monografia juntamente com o algoritmo *K-Means* foram aplicadas numa base contendo 20 erros gramaticais comuns cometidos por um conjunto de 250 usuários, do quarto e quinto período, de uma plataforma de educação online de um curso de graduação em Pedagogia de uma universidade brasileira. A escolha desses dois grupos em particular, se deve ao fato de que, estudantes que se enquadram nessas categorias possuem um alto índice de desistência do curso. A base foi construída a partir de um motor que extrai os posts dos fóruns da plataforma online, coleta esses erros e posteriormente utiliza uma outra ferramente que conta os tipos e as quantidades de erros gramaticais cometidos por cada aluno [35]. Por questões de sigilo, tanto o nome da universidade quanto o nome dos estudantes foi omitido. Os 20 tipos de erros gramaticais são descritos com mais detalhes na Tabela 5.1, e representam os erros mais comuns cometidos pelos estudantes.

Para compreensão e comparação dos resultados, o problema de agrupamento foi dividido em dois cenários principais, cujo objetivo, é comparar com

os experimentos feitos em [36]. Os cenários **4T7A** e **5T6A** correspondem, à aplicação das técnicas de agrupamento baseadas em meta-heurísticas nas bases de dados do quarto e quinto período, respectivamente. Para a realização dos experimentos referentes a esses dois cenários, foram selecionadas 7 caracteristicas mais relevantes para a base do quarto período e 6 características mais relevantes para a base do quinto período. Essa seleção foi resultado do auxílio de uma especialista da área e da remoção de todas as variáveis que possuiam apenas valores nulos e aquelas apresentando uma média inferior a 0,05. Posteriormente a essa primeira etapa de agrupamentos, serão realizadas análises da composição interna dos grupos encontrados para ambos os experimentos. Nessa etapa, será feita uma análise de resultados apenas do algoritmo vencedor, para ambos os experimentos **4T7A** e **5T6A**. Essa etapa é necessária para validar a capacidade do algoritmo em encontrar padrões nos dados e para verficar se os grupos encontrados fazem sentido, considerando o problema proposto.

5.2 Cenário 4T7A

Nas próximas seções, será realizada uma análise das métricas aplicadas sobre os resultados obtidos pelos algoritmos de agrupamento na base de dados do quarto período. Para esse experimento, foram utilizadas 7 variáveis consideradas relevantes para os alunos do quarto período. Para essa decisão, foi necessária a ajuda de um especialista da área. As sete várias extraídas da tabela 5.1 foram: (1) Uso de advérbios, (7) Concordância Nominal, (8) Colocação de Pronomes, (10) Concordância Verbal, (12) Uso de Crase, (13) Uso de Gerúndio e (16) Uso de Pontuação.

5.2.1 Resultados do algoritmo *K-Means*

Os gráficos para o resultados das métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* podem ser vistos na Figura 5.1. O algoritmo *K-Means* apresentou um comportamento esperado para a métrica *Intra-Cluster Distance*, sendo capaz de minimizá-la para os crescentes valores de k, denotando

Tabela 5.1: Tipos de erros gramaticais encontrados na base de dados.

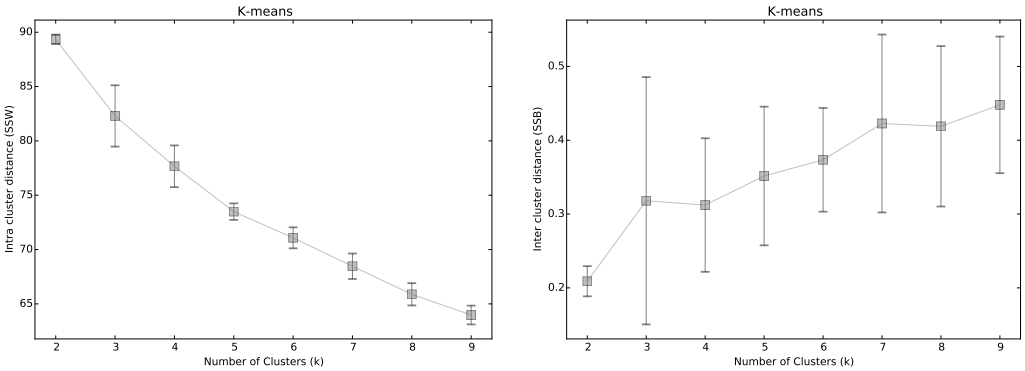
Índice	Categoria	Descrição
1	adv	Uso de advérbios
2	adv | con	Concordância entre Adjetivo e Substantivo
3	aha	Uso de há/a
4	ali	Outros
5	cjc	Uso de Conjunções
6	cmt	Concordância entre Tempo e Modos Verbais
7	con	Concordância Nominal
8	cop	Colocação de Pronomes
9	cop | pro	Uso de mim e ti
10	cov	Concordância Verbal
11	cov|reg	Verbo Fazer
12	cra	Uso de Crase
13	ger	Uso de Gerúndio
14	mal	Uso de mal/mau
15	pro	Uso de Pronomes
16	ptn	Uso de Pontuação
17	reg	Regência Verbal
18	ren	Regência Nominal
19	sem	Pleonasmo Severo
20	ver	Uso de Verbos

a capacidade do algoritmo de encontrar grupo homogêneos. Embora este seja o comportamento esperado, isso não nos permite ter uma resposta a respeito do número de grupos.

Posteriormente, pode-se observar a dificuldade do algoritmo em maximizar a métrica *Inter-Cluster Distance*, possuindo oscilações crescentes e decrescentes, denotando dificuldade de encontrar grupos afastados. Entretanto, isso nos permite inferir que o valor 3 para o número de grupos seja o mais apropriado, uma vez que é o menor valor para k que representa um máximo local seguindo por um decrescimento na curva, que representa a queda de performance do algoritmo. Porém, como o valor para k igual a 3 apresenta alto desvio padrão, essa não se mostra uma resposta segura. Portanto pelo gráfico da métrica *Inter-Cluster Distance*, é possível inferir que o valor 2 seria

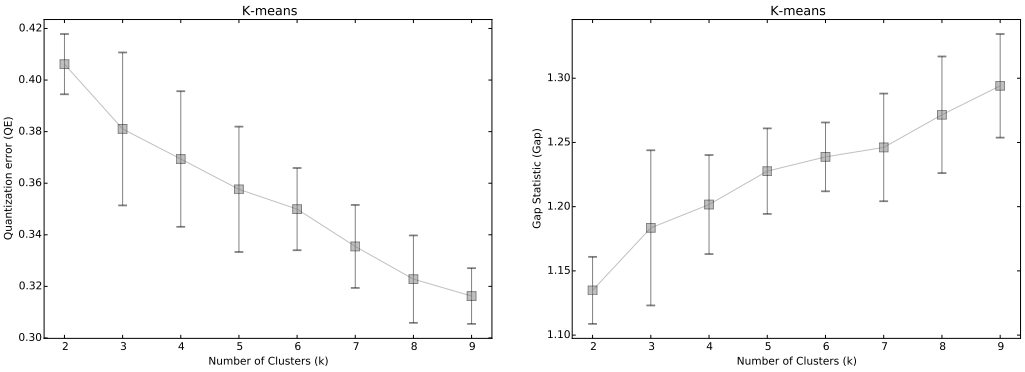
o mais apropriado para o número de grupos.

Figura 5.1: Gráfico das Métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* aplicada nos resultados do algoritmo de agrupamento *K-Means*.



A Figura 5.2 mostra os gráficos das métricas Erro Quantizado e Estatística *Gap* para o algoritmo *K-Means*. Como a métrica Erro Quantizado também faz uma avaliação interna dos grupos, percebe-se que o algoritmo teve sucesso na minimização deste métrica, apresentado mais uma vez o comportamento esperado. O algoritmo também obteve sucesso na maximização da Estatística *Gap*. Utilizando a Equação (3.11), o valor 2 se mostra o mais apropriado para o algoritmo *K-Means*.

Figura 5.2: Gráfico das Métricas Erro Quantizado e Estatística *Gap* aplicada nos resultados do algoritmo de agrupamento *K-Means*.

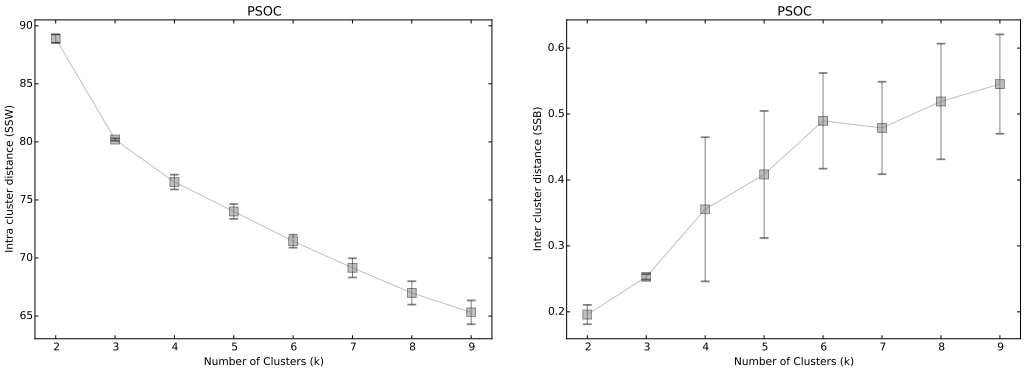


5.2.2 Resultados do algoritmo *PSOC*

Os gráficos para o resultados das métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* podem ser vistos na Figura 5.3. O algoritmo também obteve o comportamento esperando ao minimizar a métrica *Intra-Cluster Distance*, não permitindo inferir uma resposta a respeito do número de grupos.

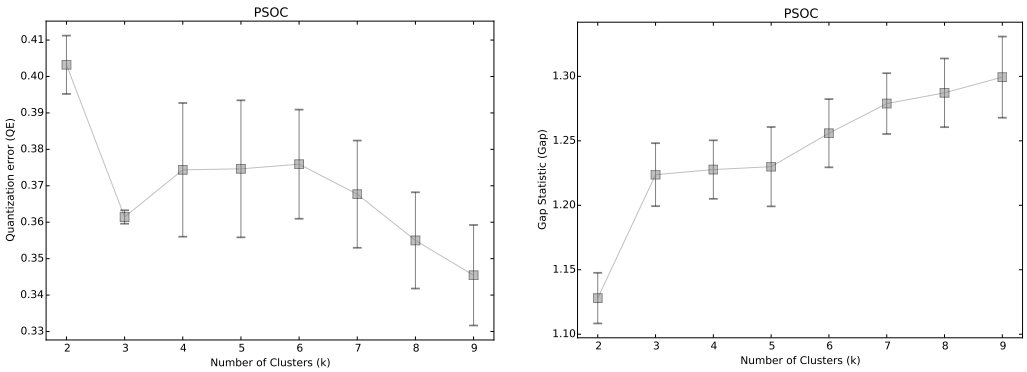
Posteriormente, pode-se observar que o algoritmo teve algumas diminuições em sua performance ao tentar maximizar a métrica *Inter-Cluster Distance*. O algoritmo se comporta bem para valores crescentes de k até atingir o valor 7, onde sofre uma queda abrupta. Essa queda na curva indica uma possível estagnação do algoritmo, que mostra que k igual a 6 é um bom inidicador para o número de grupos.

Figura 5.3: Gráfico das Métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* aplicada nos resultados do algoritmo de agrupamento *PSOC*.



A Figura 5.4 mosta os resultados das métricas Erro Quantizado e Estatística *Gap* para o algoritmo *PSOC*. Analisando a curva do Erro Quantizado, observa-se um queda abrupta para k igual a 3, seguida da incapacidade do algoritmo em funcionar bem para valores de k subsequêntes. Isso é um bom indicativo de que para o Erro Quantizado o valor 3 é ideal para o número de grupos. Para a métirca Estatística *Gap*, vê-se pelo gráfico que o valor de k igual a 3 é o menor valor de k a satisfazer a Equação (3.11).

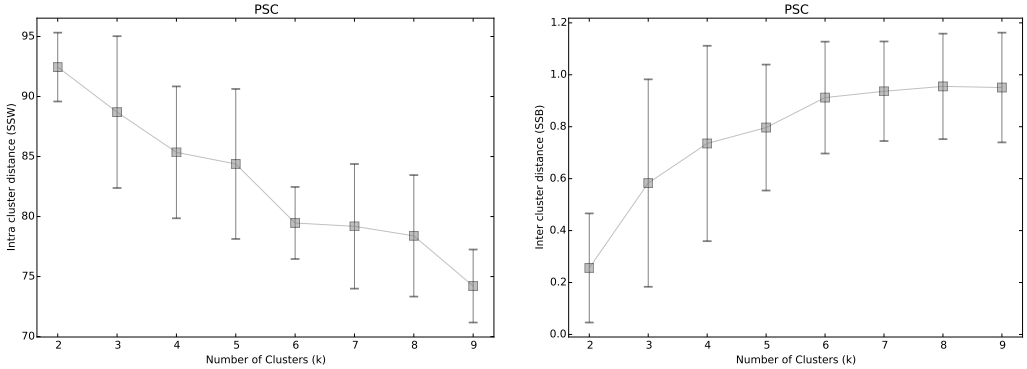
Figura 5.4: Gráfico das Métricas Erro Quantizado e Estatística *Gap* aplicada nos resultados do algoritmo de agrupamento *PSOC*.



5.2.3 Resultados do algoritmo *PSC*

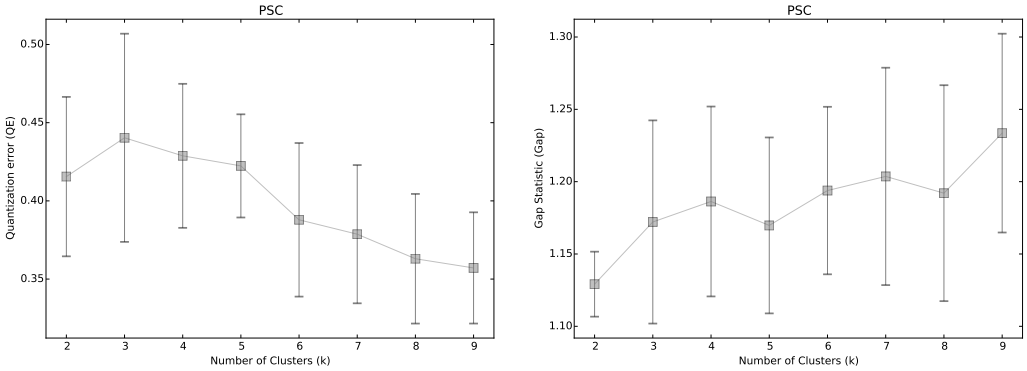
A Figura 5.5 mostra as métricas *Intra* e *Inter Cluster Distance* para o algoritmo *PSC*. Para a métrica *Intra Cluster Distance*, o valor 6 se mostra um bom indicativo para o número de grupos, pois observa-se que a taxa de decrescimento da curva cai consideravelmente para os valores subsequente de k, mostrando que o algoritmo não consegue resultados mais interessantes a partir desse ponto. O algoritmo obteve sucesso em maximizar a métrica *Inter Cluster Distance*, obtendo assim o resultado esperado que não nos permite inferir um valor apropriado para o número de grupos.

Figura 5.5: Gráfico das Métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* aplicada nos resultados do algoritmo de agrupamento *PSC*.



Para a métrica Erro Quantizado, pode-se observar na Figura 5.6 que o valor 3 é um bom candidato para o número de grupos, pois é a partir desse ponto que o algoritmo se mostra ineficiente em minimizar o erro quantizado. Para a Estatística *Gap* no lado direito da Figura 5.6, observa-se que o menor valor de k que satisfaz a Equação (3.11) é para k igual a 3, sendo portanto o melhor valor para o número de grupos de acordo com esta métrica.

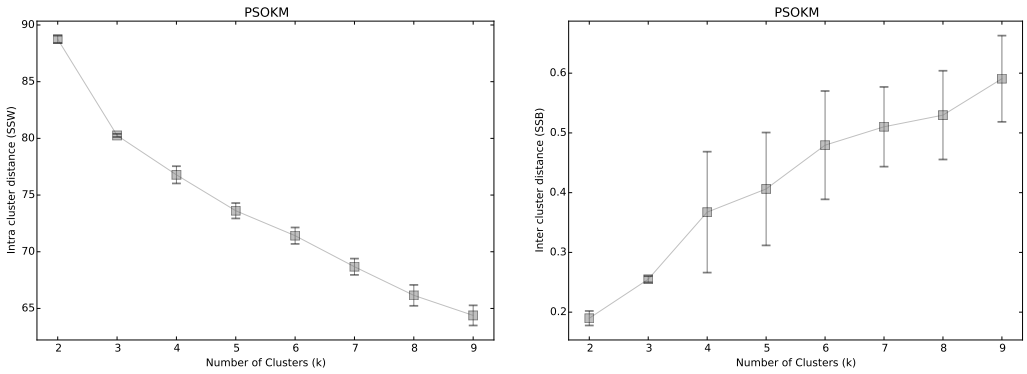
Figura 5.6: Gráfico das Métricas Erro Quantizado e Estatística *Gap* aplicada nos resultados do algoritmo de agrupamento *PSC*.



5.2.4 Resultados do algoritmo *PSOKM*

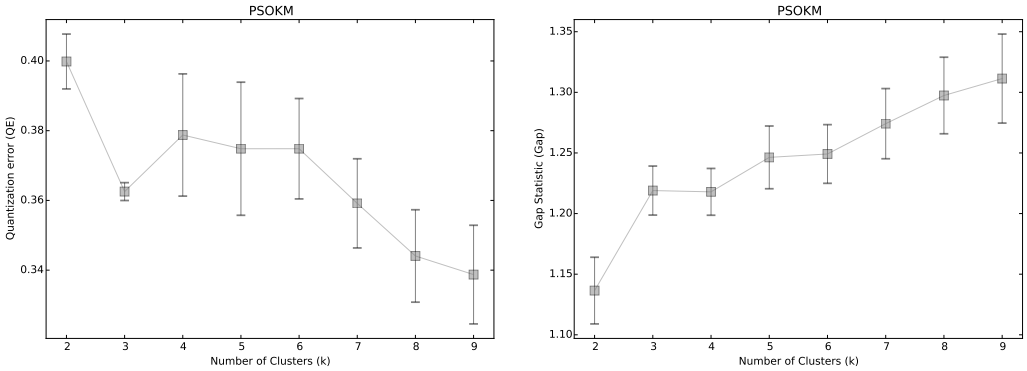
Para ambas as métricas *Intra* e *Inter Cluster Distance*, o algoritmo aprensetou o comportamento esperado, como pode ser visto na Figura 5.7, não sendo possível portanto inferir nenhum valor para o número de grupos a partir destas métricas.

Figura 5.7: Gráfico das Métricas Intra-Cluster Distance e Inter-Cluster Distance aplicada nos resultados do algoritmo de agrupamento *PSOKM*.



Para a métrica Erro Quantizado, ao lado esquedo da Figura 5.8 , obsevamos que o algoritmo tem uma queda abrupta para k igual a 3, sendo seguida de um comportamento crescente para esta métrica. Os valores crescentes para valores subsequentes de k na métrica de Erro Quantizado indicam que o valor 3 é o mais apropriado para o número de grupos. Ao lado direito da Figura 5.8, observa-se que para a métrica *Gap* o valor 3 também se mostra ideal para o número de grupos.

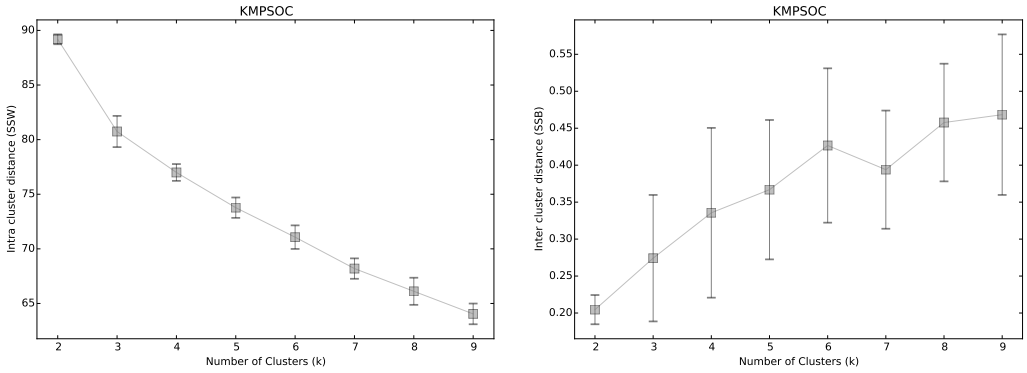
Figura 5.8: Gráfico das Métricas Erro Quantizado e Estatística *Gap* aplicada nos resultados do algoritmo de agrupamento *PSOKM*.



5.2.5 Resultados do algoritmo *KMPSOC*

No lado esquerdo da Figura 5.9 vê-se que o algoritmo *KMPSOC* minimiza de forma esperada a métrica *Intra-Cluster Distance*, portanto não se pode inferir um valor para o número de grupos para esta métrica. No lado direito da figura 5.9, observa-se que para valores menores que 7, o algoritmo se comporta bem ao maximizar a métrica *Inter-Cluster Distance*. Percebe-se que para k igual a 7 há uma queda abrupta na performance do algoritmo, seguido por um crescimento com uma taxa bastante reduzida, o que pode indicar uma estagnação do algoritmo para esta métrica. Portanto, para a métrica *Inter Cluster Distance* o valor 6 se mostra o mais indicado para o número de grupos.

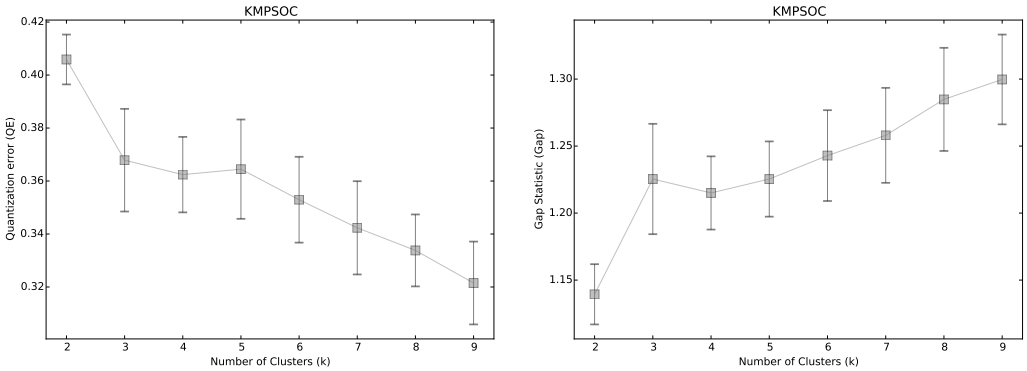
Figura 5.9: Gráfico das Métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* aplicada nos resultados do algoritmo de agrupamento *KMPSOC*.



Na Figura 5.10, pode-se observar os gráficos das métricas Erro Quantizado e Estatística *Gap* para o algoritmo *KMPSOC*. Observe que para a métrica Erro Quantizado o algoritmo funciona bem, minimizando a métrica, para valores de k menores que 5. Para k igual a 5, a métrica é maximizada denotando um desgaste do algoritmo, que fica perceptível pela sua dificuldade em minimizar essa métrica. Portanto, para a métrica de Erro Quantizado, obtem-se um valor de k igual a 4 para o número de grupos. No lado direito da figura 5.10 pode-se observar que o valor de k igual a 3, é o menor valor de k que satisfaz a desigualdade (3.11), sendo o melhor valor para número

de grupos.

Figura 5.10: Gráfico das Métricas Erro Quantizado e Estatística *Gap* aplicada nos resultados do algoritmo de agrupamento *KMPSOC*.



5.2.6 Definição do número de grupos para o cenário 4T7A

Durante as simulações realizadas observa-se que as métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* não permitiram resultados conclusivos para a maioria dos algoritmos de agrupamento. Dentre as métricas utilizadas, a Estatística *Gap* se mostrou mais adequada para definir o número de grupos [22], tendo majoritariamento o valor 3 para o número de grupos sendo alcaçado pela maioria do algoritmos, resultado diferente do trabalho proposto em [36]. Analisando a confiança dos algoritmos ao reportar os resultados do problema de agrupamento, nota-se que o algoritmo *PSOC* foi o que apresentou os menores valores para o desvio padrão, sendo portanto o algoritmo considerado vencedor para o cenário 4T7A.

5.3 Cenário 5T6A

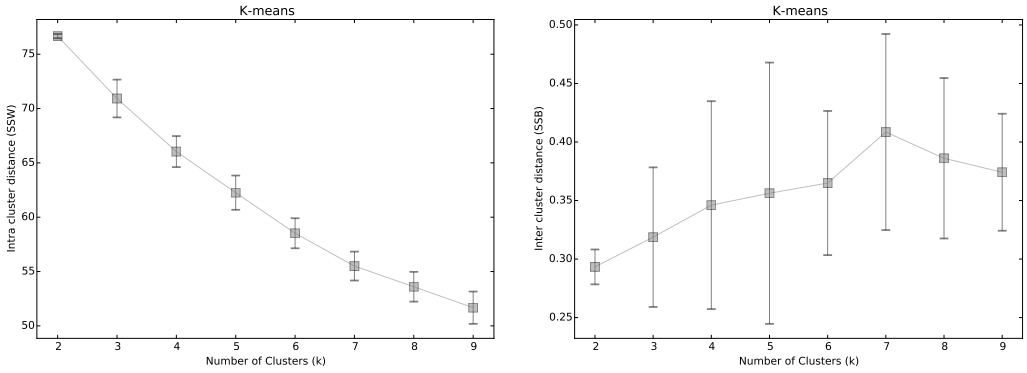
As próximas seções, se referem a análise das métricas aplicadas sobre os resultados obtidos pelos algoritmos de agrupamento na base de dados do quinto período. Para esse experimento foram utilizadas 6 variáveis conside-

radas relevantes para os alunos do quinto período. As seis várias extraídas da tabela 5.1 foram: (1) Uso de advérbios, (7) Concordância Nominal, (8) Colocação de Pronomes, (10) Concordância Verbal, (12) Uso de Crase e (13) Uso de Gerúndio.

5.3.1 Resultados do algoritmo *K-Means*

Para a métrica *Intra Cluster Distance* o algoritmo *K-Means* teve o comportamento esperado, minimizando a métrica, gerando portanto resultados inconclusivos. Em contrapartida, para a métrica *Inter Cluster Distance*, o algoritmo apresentou um comportamento de maximização até atinigir o máximo para k igual a 7. A partir desse momento o algoritmo começa a sofrer uma deficiência no processo de maximização, apresentando um comportamento não desejado de minimização para está métrica. Portanto, para a métrica *Inter Cluster Distance* pode-se concluir que o valor k igual a 7 é o mais apropriado para o número de grupos.

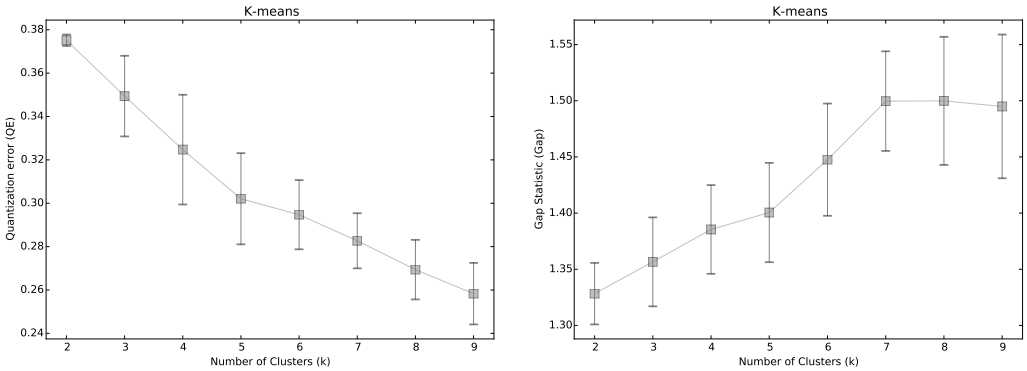
Figura 5.11: Gráfico das Métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* aplicada nos resultados do algoritmo de agrupamento *K-Means*.



Observe ao lado direito da Figura 5.12 que o algoritmo *K-Means* apresentou o comportamento esperado para a métrica Erro Quantizado, portanto não podemos chegar a nenhuma conclusão a respeito do número de grupo. Ademais, ao lado direito da Figura 5.12 pode-se observar que o valor de k igual a 2, é o menor valor de k que satisfaz a desigualdade (3.11), sendo o

melhor valor para número de grupos.

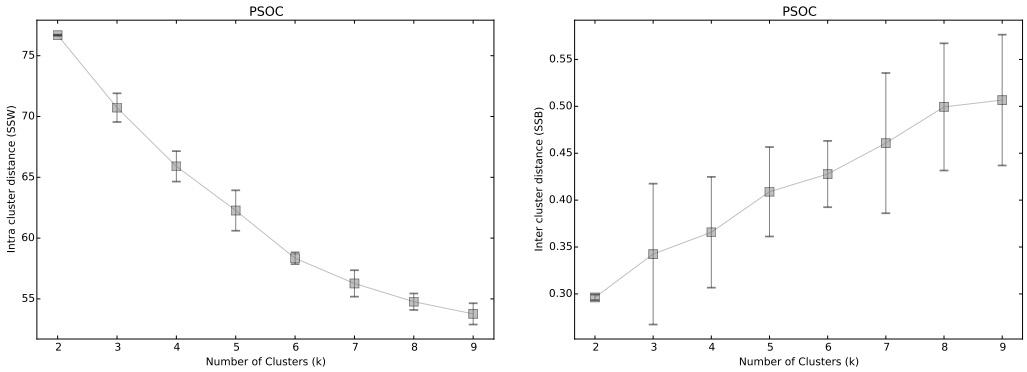
Figura 5.12: Gráfico das Métricas Erro Quantizado e Estatística *Gap* aplicada nos resultados do algoritmo de agrupamento *K-Means*



5.3.2 Resultados do algoritmo *PSOC*

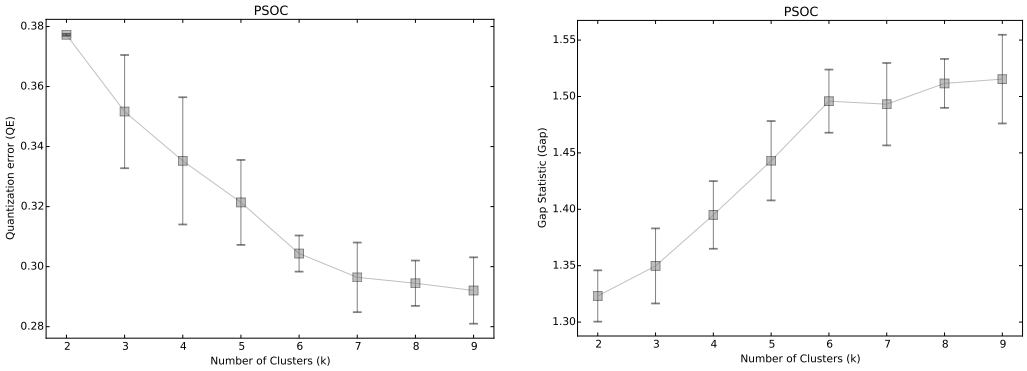
Para as métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* na Figura 5.13, e para métrica Erro Quantizado ao lado esquerdo da Figura 5.14, o algoritmo *PSOC* obteve o comportamento esperado. Portanto, a análise dessas três métricas não permite chegar a um resultado conclusivo a respeito do número de grupos.

Figura 5.13: Gráfico das Métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* aplicada nos resultados do algoritmo de agrupamento *PSOC*.



Para a métrica Estatística *Gap*, observa-se que o algoritmo consegue maximizar bem para valores de k menores que 6. Entretanto, ao analisar a curva, pode-se observar que o valor de k igual a 2, é o menor valor de k que satisfaz a desigualdade (3.11), sendo o melhor valor para número de grupos.

Figura 5.14: Gráfico das Métricas Erro Quantizado e Estatística *Gap* aplicada nos resultados do algoritmo de agrupamento *PSOC*.



5.3.3 Resultados do algoritmo *PSC*

Para o algoritmo *PSC*, o algoritmo obteve o comportamento esperado para a métrica *Intra-Cluster Distance*, que pode ser observado ao lado esquerdo da Figura 5.15. Para a métrica *Inter-Cluster Distance* o algortimo apresentou o comportamento de maximização esperado para valores de k menores que 6. Para k igual a 7, o algoritmo sofreu um minimização local seguida por uma maximização, porém com uma taxa de crescimento bastante reduzida. Portanto, para a métrica *Inter-Cluster Distance*, o valor 6 é o mais apropriado para o número de grupos.

Para a métrica Erro Quantizado, ao lado esquerdo da Figura 5.16, o algoritmo apresentou o comportamento esperando de minimização, o que não permite inferir um valor apropriado para o número de grupos. Para a métrica Estatística, *Gap* pode-se observar, ao lado direito da Figura 5.16, que o algoritmo enfrentou dificuldades no processo de maximização dessa métrica. Ao analisar a Estatística *Gap*, vê-se que o menor valor de k que

satisfaz a desigualdade (3.11) é para k igual a 2, sendo esse o melhor valor para número de grupos.

Figura 5.15: Gráfico das Métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* aplicada nos resultados do algoritmo de agrupamento *PSC*.

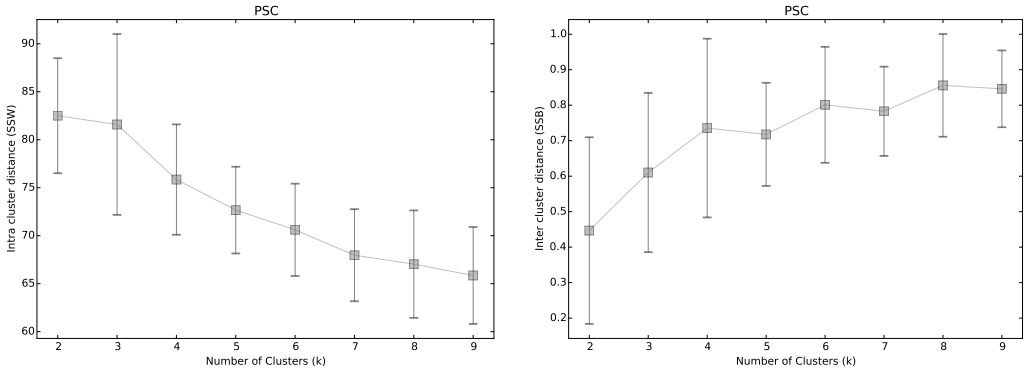
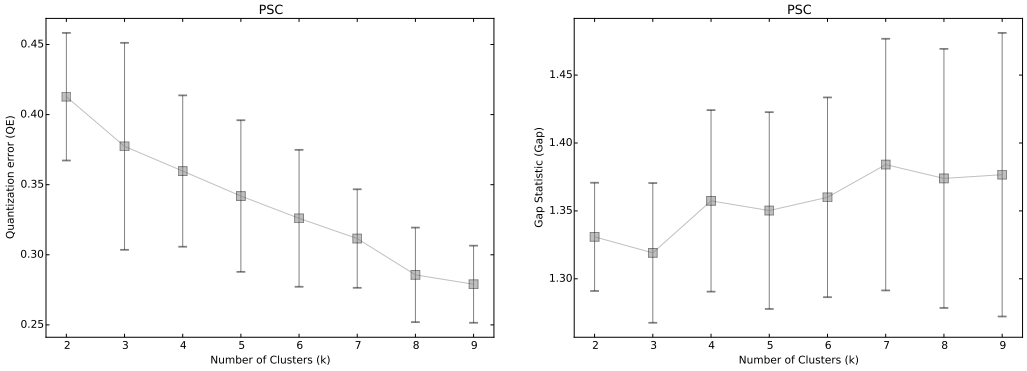


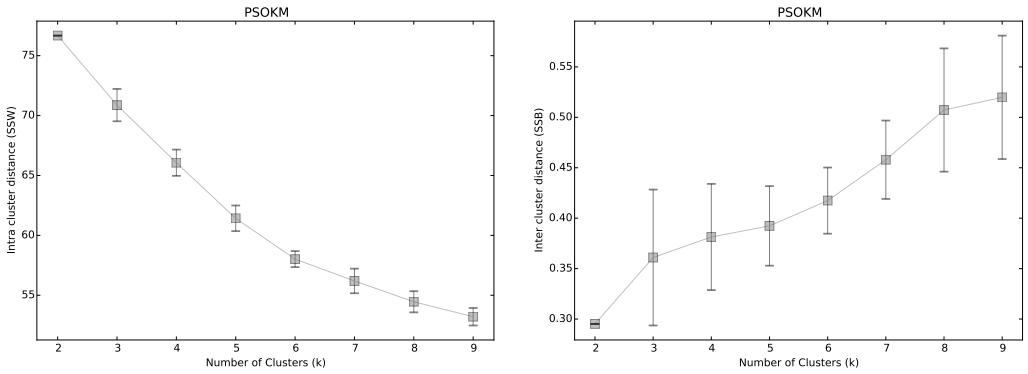
Figura 5.16: Gráfico das Métricas Erro Quantizado e Estatística *Gap* aplicada nos resultados do algoritmo de agrupamento *PSC*



5.3.4 Resultados do algoritmo *PSOKM*

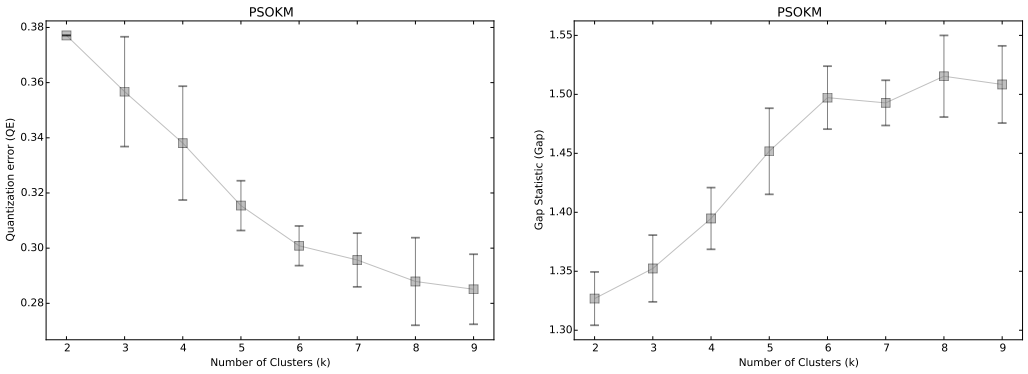
As métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* apresentaram os resultado esperados, minimização estritamente decrescente e maximização estritamente crescente, respectivamente, e que podem ser observados na figura 5.17. Portanto, nenhuma dessas duas métricas nos permite inferir alguma informação para o número de grupos.

Figura 5.17: Gráfico das Métricas *Intra Cluster Distance* e *Inter Cluster Distance* aplicada nos resultados do algoritmo de agrupamento *PSOKM*



O comportamento esperado também ocorre para a métrica Erro Quantizado, que pode ser observado ao lado esquerdo da Figura 5.18. Para a métrica gap, ao analisar a curva obtida, pode-se visualizar que o valor de k igual a 2 satisfaz a desiguldade (3.11), sendo portanto, o melhor valor para o número de grupos.

Figura 5.18: Gráfico das Métricas Erro Quantizado e Estatística *Gap* aplicada nos resultados do algoritmo de agrupamento *PSOKM*

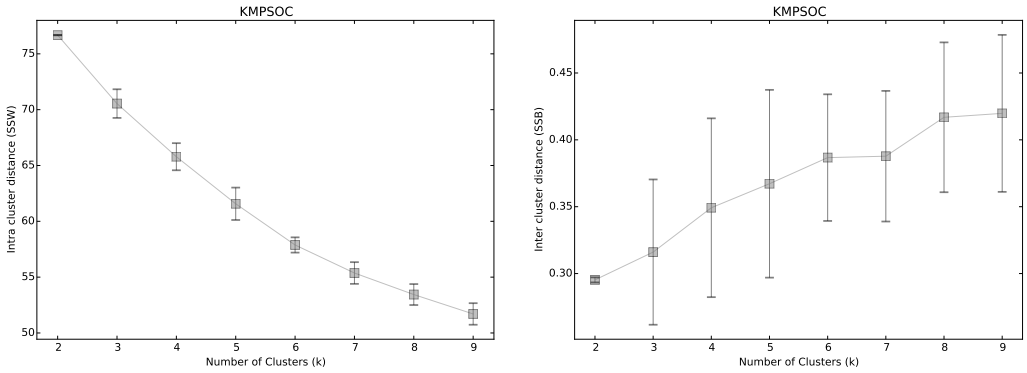


5.3.5 Resultados do algoritmo *KMPSOC*

O algoritmo *KMPSOC* apresenta o comportamento estritamente decrescente esperado para a métrica *Intra-Cluster Distance*, que pode ser observado

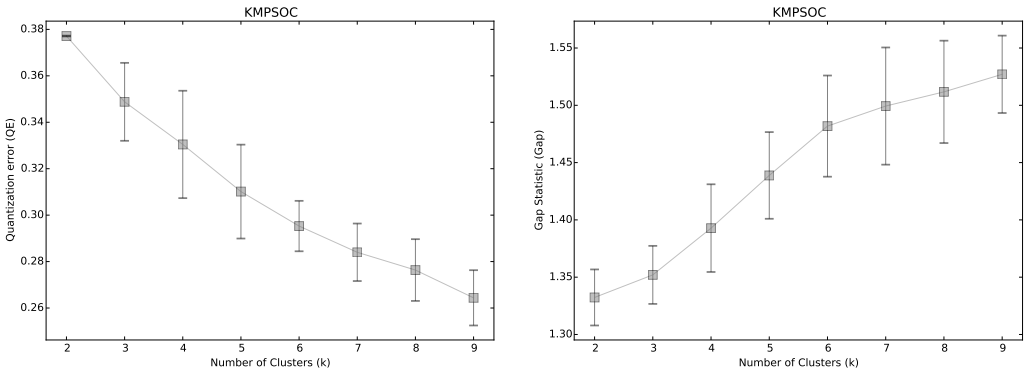
ao lado esquerdo da Figura 5.19, portanto não se pode concluir o número de grupos com o auxílio desta métrica. Para a métrica *Inter-Cluster Distance*, mais uma vez obtem-se o comportamento esperado. Portanto, nenhuma das métricas *Intra* ou *Inter-Cluster Distance* nos auxilia no processo de descoberta do número de grupos.

Figura 5.19: Gráfico das Métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* aplicada nos resultados do algoritmo de agrupamento *KMPSOC*.



Para a métrica Erro Quantizado, ao lado esquerdo da Figura 5.20, o algoritmo também apresentou o comportamento esperado de minimização, não permitindo a concluisão de um valor apropriado para o número de grupos. Entretanto, para a métrica Estatística *Gap*, ao lado direito da Figura 5.20, pode-se observar que o valor de k igual 2 é o menor valor que satisfaz a desigualdade (3.11), sendo o melhor valor para o número de grupos para essa métrica.

Figura 5.20: Gráfico das Métricas Erro Quantizado e Estatística *Gap* aplicada nos resultados do algoritmo de agrupamento *KMPSOC*



5.3.6 Definição do número de grupos para o cenário 5T6A

Assim como visto na Subseção 5.2.6, nota-se no experimento 5T6A a incapacidade das métricas *Intra-Cluster Distance* e *Inter-Cluster Distance* em gerar resultados assertivos a respeito do número de grupos quando aplicada ao algorimos de agrupamento apresentados nessa obra. Na análise da Estatística *Gap*, pode-se observar que a maioria do algoritmos chegou a conclusão de que o valor 2 seria o mais apropriado para o número de grupos. Analisando a confiança dos algoritmos ao reportar resultados, nota-se também que o algoritmo *PSOC* foi o que apresentou os menores valores de incerteza, também sendo o algoritmo considerado vencedor para o cenário 5T6A.

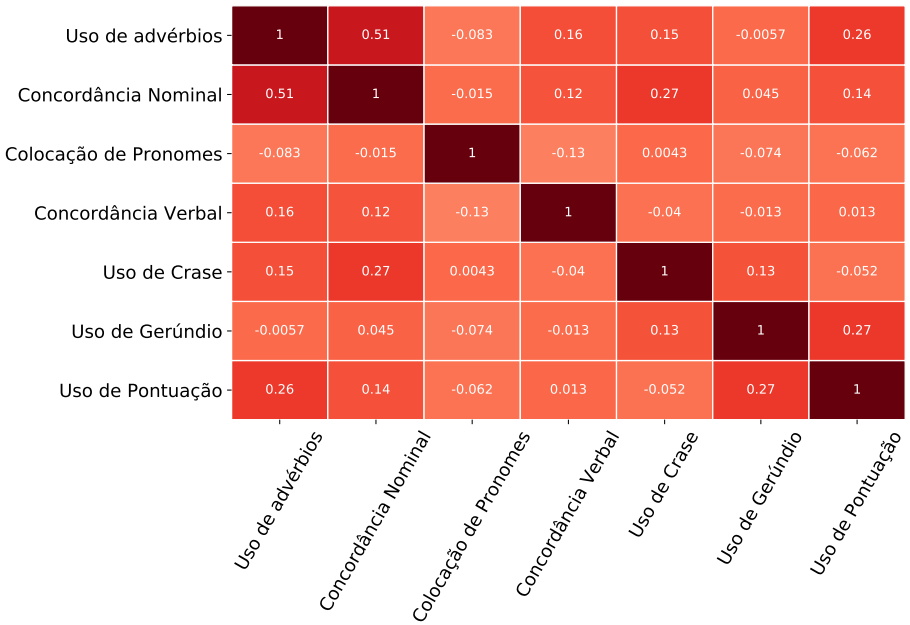
5.4 Análise dos grupos para o cenário 4T7A

Nesta seção, será feita uma validação dos resultados encontrados pelo algoritmo *PSOC* na base de dados do quarto período. O objetivo desta etapa é validar a capacidade do algoritmo em encontrar características discriminantes para cada um dos grupos encontrados, permitindo assim, identificar a pertinência de futuras instâncias.

5.4.1 Cenário 4T7A - Análise do Primeiro Grupo

A matriz de correlação entre a variáveis do primeiro grupo encontrado pelo algoritmo *PSOC* pode ser visualizada na Figura 5.21. Ao analisar a matriz de correlação, observamos valores baixos para as correlações, sendo todos abaixo de 0,6. A correlação entre duas variáveis é considerada forte quando possui seu valor entre 0,9 e 1. Entretanto, é possível realizar essa análise olhando para os valores máximos como referência, para inferir um padrão comportamental entre as variáveis. Excluindo os valores da diagonal principal, que sempre possuem correlação máxima igual a 1, pode-se concluir que para esse grupo, valores elevados de correlação entre as variáveis (1) Uso de Advérbios e (7) Concordância Nominal, sendo essa uma possível característica discriminante deste grupo.

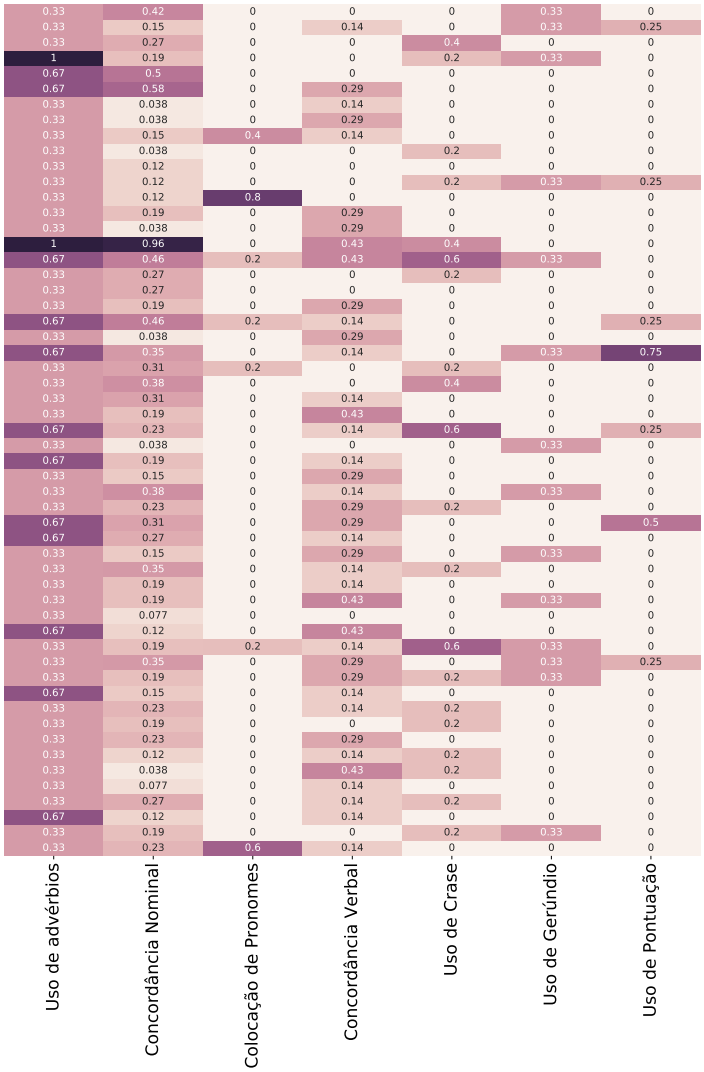
Figura 5.21: Matriz de correlação das variáreis do primeiro grupo encontrado pelo algoritmo *PSOC* para a base do quarto período.



A alta correlação entre as variáveis (1) Uso de Advérbios e (7) Concordância Nominal como características discriminantes para as instâncias per-

tencentes a esse grupo fica mais visível ao se analisar a distribuição das variáveis, que pode ser vista com mais detalhes na Figura 5.22. Perceba que as instâncias pertencentes a esse grupo apresentam valores elevados para essas duas variáveis, o que reforça a análise feita no estudo da matriz de correlação. Nesse cenário a plataforma de educação a distância poderia oferecer conteúdos relacionados a esses dois tópicos para auxílio dos alunos.

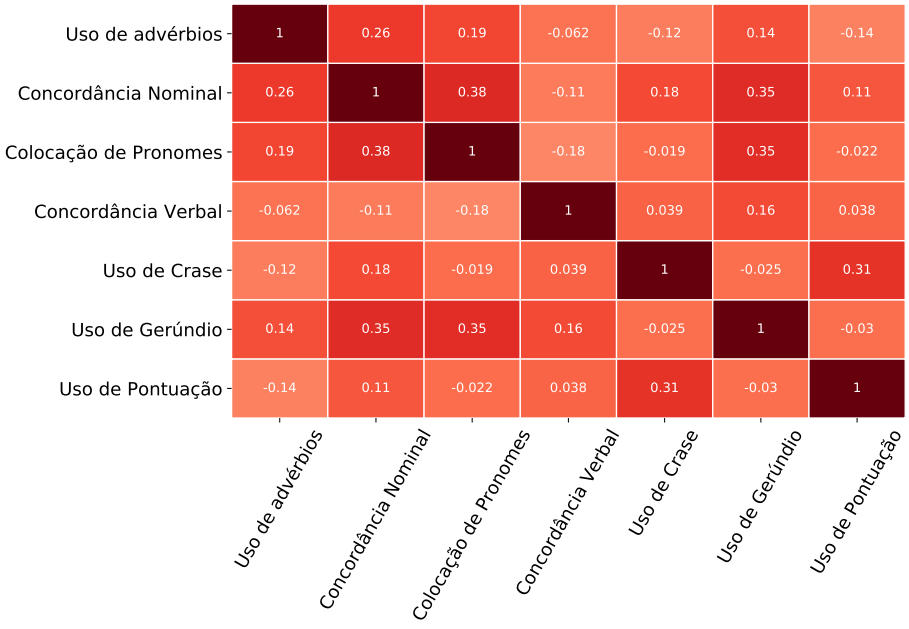
Figura 5.22: Distribuição das variáreis do primeiro grupo encontrado pelo algoritmo *PSOC* para a base do quarto período.



5.4.2 Cenário 4T7A - Análise do Segundo Grupo

O segundo grupo possui valores menores de correlação do que os encontrados no primeiro grupo, ou seja, o alunos desse grupo tendem a confundir menos os tópicos a serem estudados. Pode-se verificar na matriz, que pode ser vista com detalhes na Figura 5.23, que os maiores valores de correlação ocorrem para os pares de variáveis (7) Concordância Nominal e (8) Colocação de Pronomes, (7) Concordância Nominal e (13) Uso de Gerúndio. Embora esses sejam os pares de variáveis com maior valor para coeficiente de correlação, é necessária uma maior análise para definir quais atributos caracterizam as instâncias pertencentes a esse grupo.

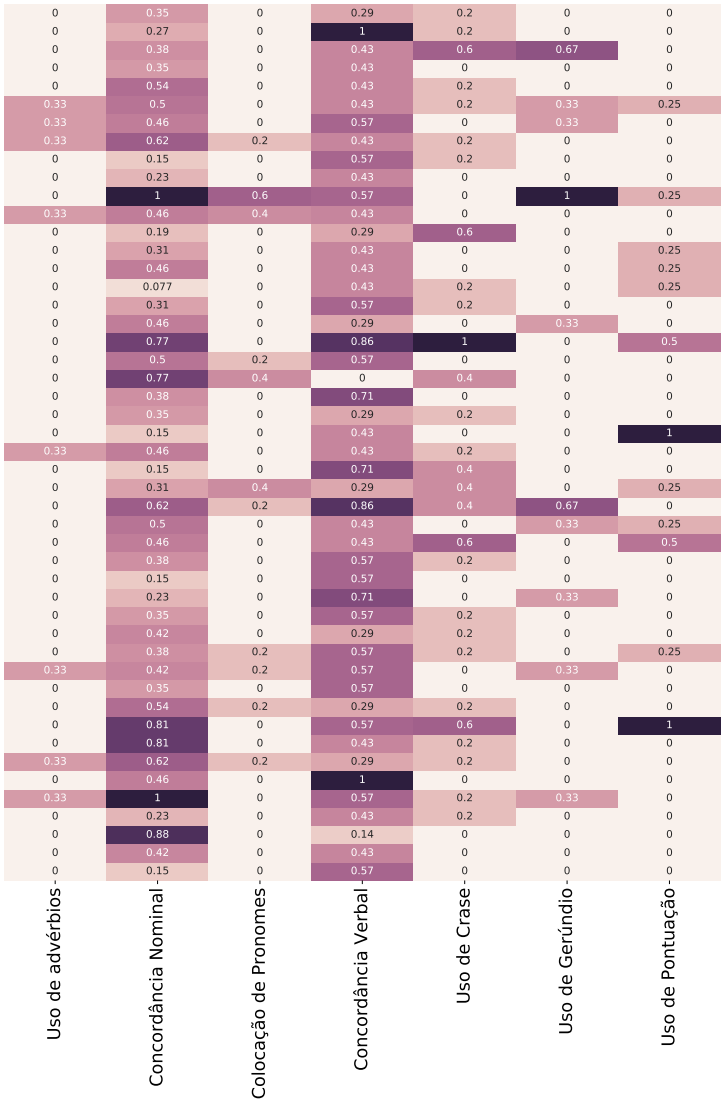
Figura 5.23: Matriz de correlação das variáreis do segundo grupo encontrado pelo algoritmo *PSOC* para a base do quarto período.



Analisando a distribuição das variáveis no segundo grupo do cenário **4T6A**, que pode ser visto na figura 5.24, percebe-se que alunos desse grupo possuem alto índice de erro para as variáveis (7) Concordância Nominal e (10) Concordância Verbal. A presença da variável (7) Concordância Nominal re-

força a análise anterior, pois ela aparece nos dois pares de maior correlação. Prosseguindo a análise da distribuição desse grupo, nota-se mais um fator discriminante: A variável (10) Concordância Verbal, não presente em nossa análise inicial da matriz de correlação. Portanto, pode-se inferir que valores altos para as variáveis (7) Concordância Nominal e (10) Concordância Verbal caracterizam o segundo grupo do cenário **4T6A**.

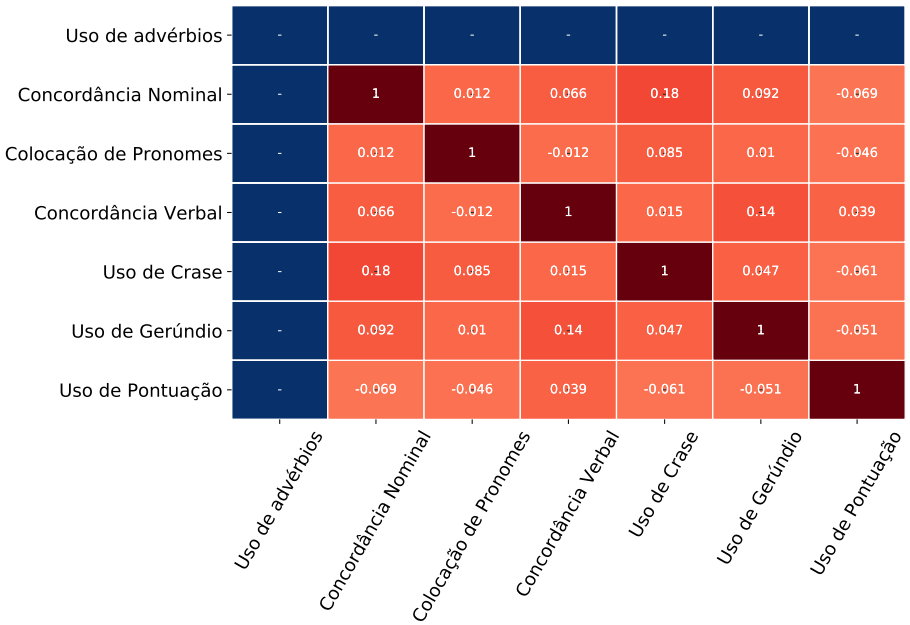
Figura 5.24: Distribuição das variáreis do segundo grupo encontrado pelo algoritmo *PSOC* para a base do quarto período.



5.4.3 Cenário 4T7A - Análise do Terceiro Grupo

O terceiro grupo encontrado possui os menores valores de correlação quando comparados com os do primeiro e segundo grupos. A priori pode-se inferir que esse grupo reune alunos com os melhores desempenhos. A Figura 5.25 mostra a matriz de correlação entre as variáveis do terceiro grupo. Uma característica peculiar desse grupo, é que alunos desse grupo possuem um desempenho ótimo na variável (1) Uso de advérbio. Uma consequência desse comportamento é um valor nulo para o desvio padrão, e portanto, a primeira linha e coluna do matriz de correlação são inexistentes.

Figura 5.25: Matriz de correlação das variáreis do terceiro grupo encontrado pelo algoritmo *PSOC* para a base do quarto período.



Reforçando a análise da matriz correlação, pode-se observar a distribuição das variáveis para o terceiro grupo na Figura 5.26. A coluna que representa a variável (1) Uso de advérbio é formada apenas por zeros. No geral tem-se valores muito baixos para as demais variáveis, o que mostra que alunos pertencentes a esse grupo possuem bom desempenho nesses tópicos.

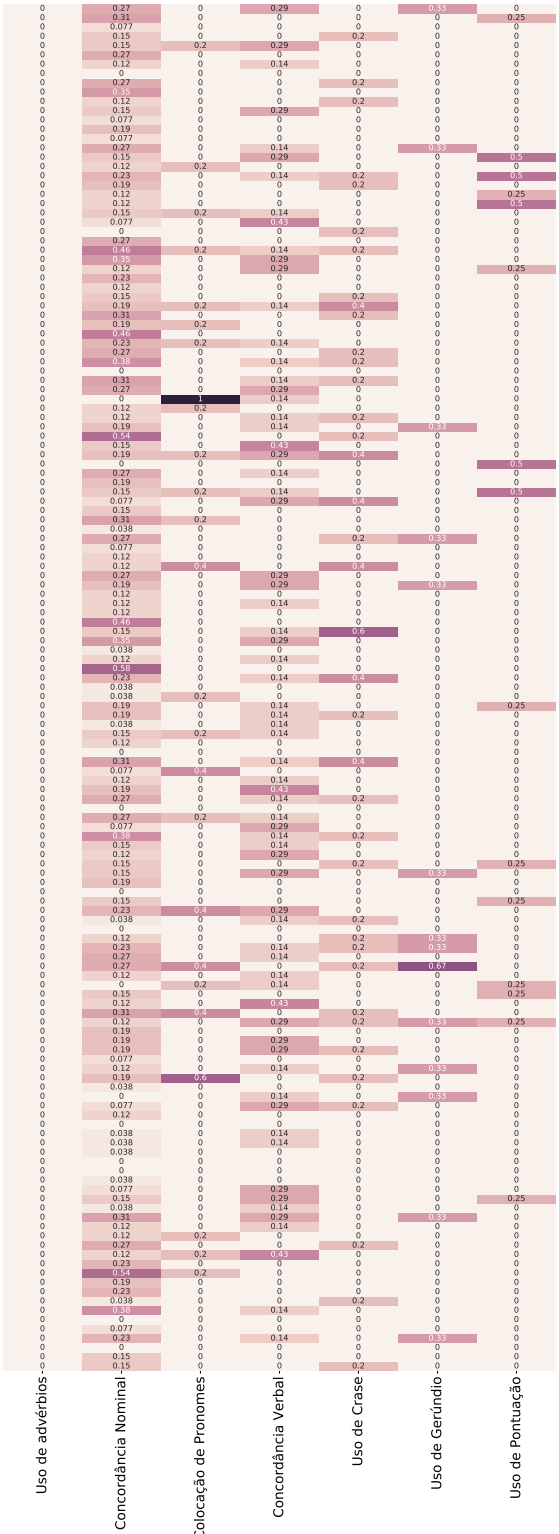


Figura 5.26: Distribuição das variáreis do terceiro grupo encontrado pelo algoritmo *PSOC* para a base do quarto período.

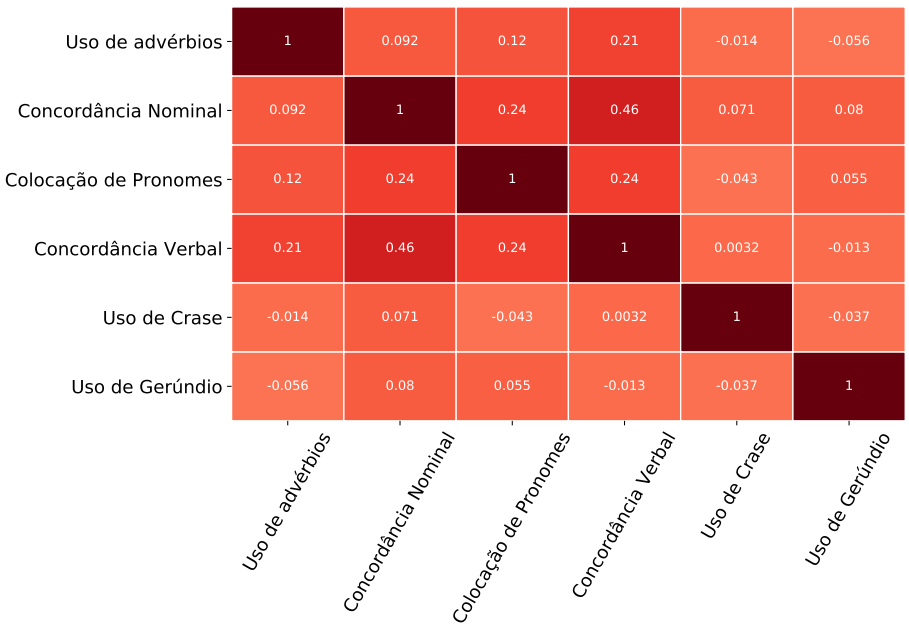
5.5 Análise dos grupos para o cenário 5T6A

Nesta seção, será feita uma validação dos resultados encontrados pelo algoritmo *PSOC* na base de dados do quinto período. Assim como na seção anterior, o objetivo aqui é validar características discriminativas para cada um dos grupos para o cenário **5T6A**.

5.5.1 Cenário 5T6A - Análise do Primeiro Grupo

No primeiro grupo encontrado do cenário **5T6A**, observa-se baixos valores de correlação na Figura 5.27. O maior valor ocorre entre as variáveis (7) Concordância Nominal e (10) Concordância Verbal, indicando que estudantes desse grupo tendem a confundir levemente esses dois assuntos. Com exceção desses tópicos, as baixas correlações entre as demais variáveis indicam que os estudantes tem um bom rendimento nos demais tópicos, fato que pode ser observado na distribuição das variáveis desse grupo nas Figuras 5.28 e 5.29.

Figura 5.27: Matriz de correlação das variáreis do primeiro grupo encontrado pelo algoritmo *PSOC* para a base do quinto período.



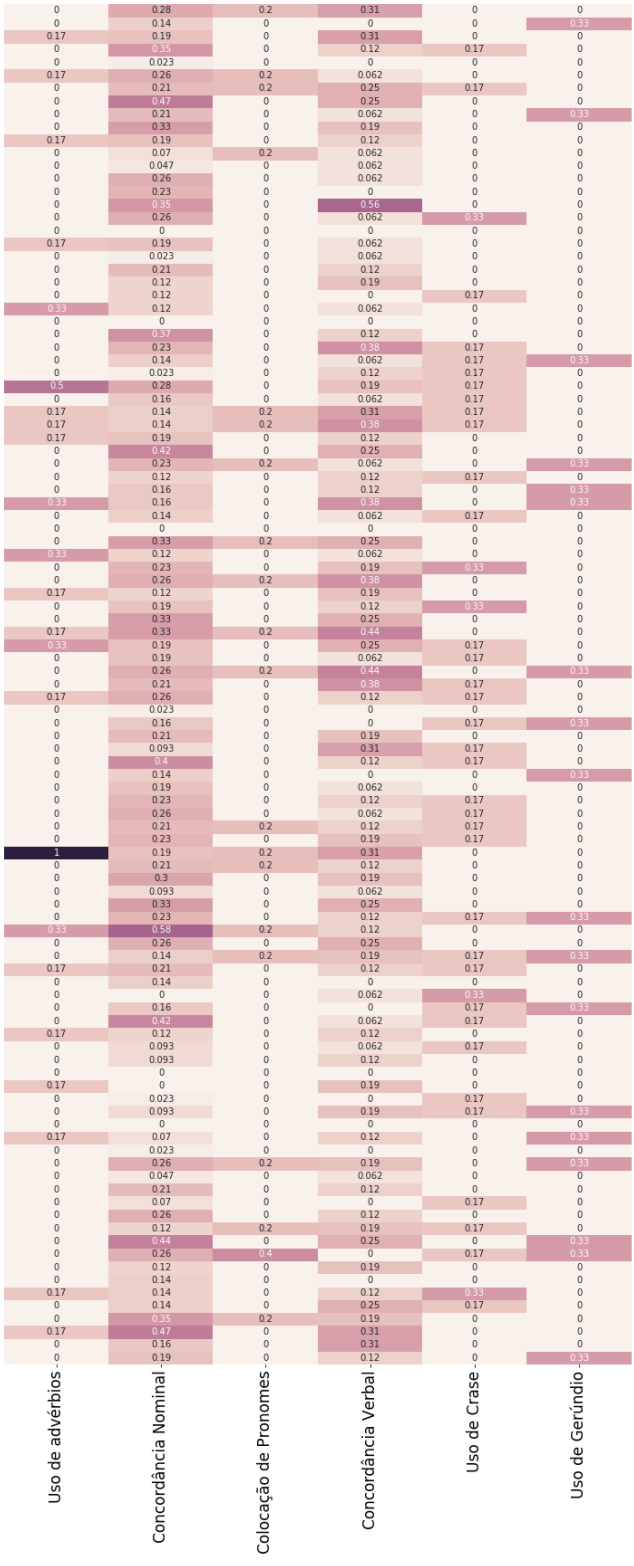


Figura 5.28: Primeira parte da distribuição das variáreis do primeiro grupo encontrado pelo algoritmo *PSOC* para a base do quinto período.

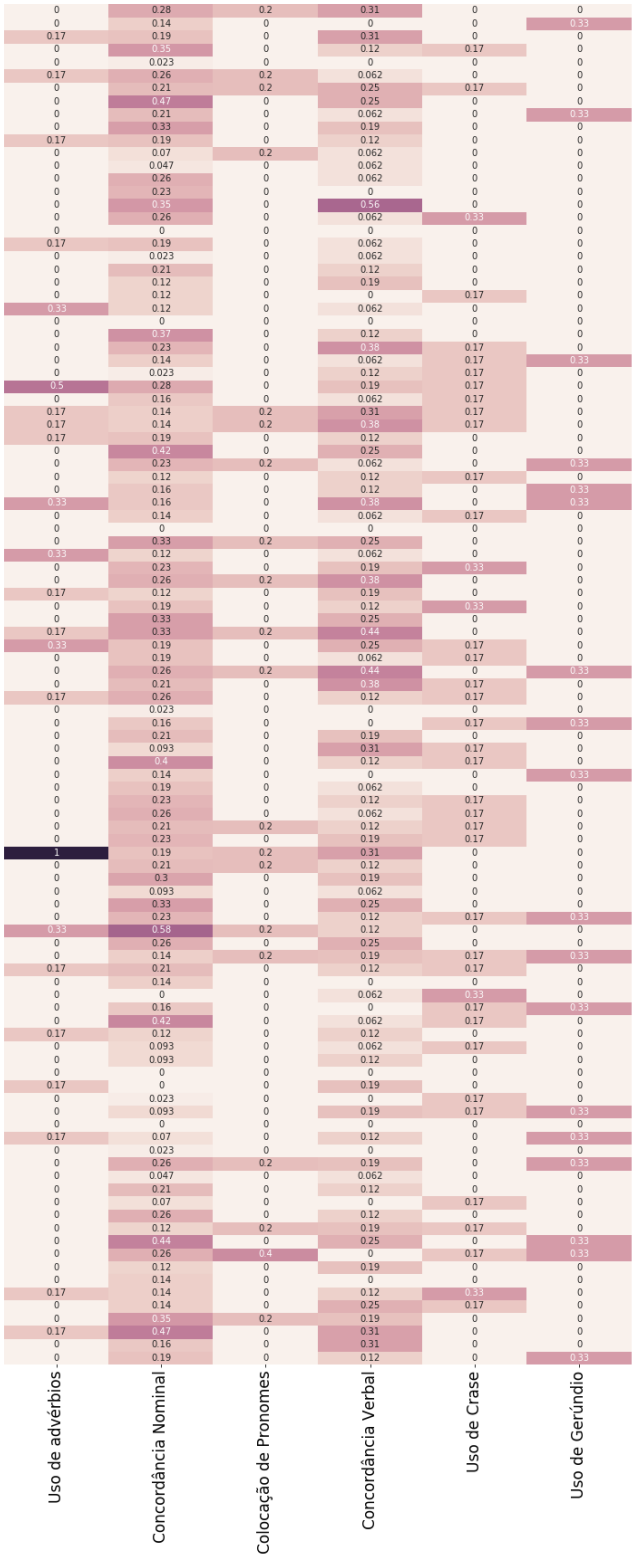


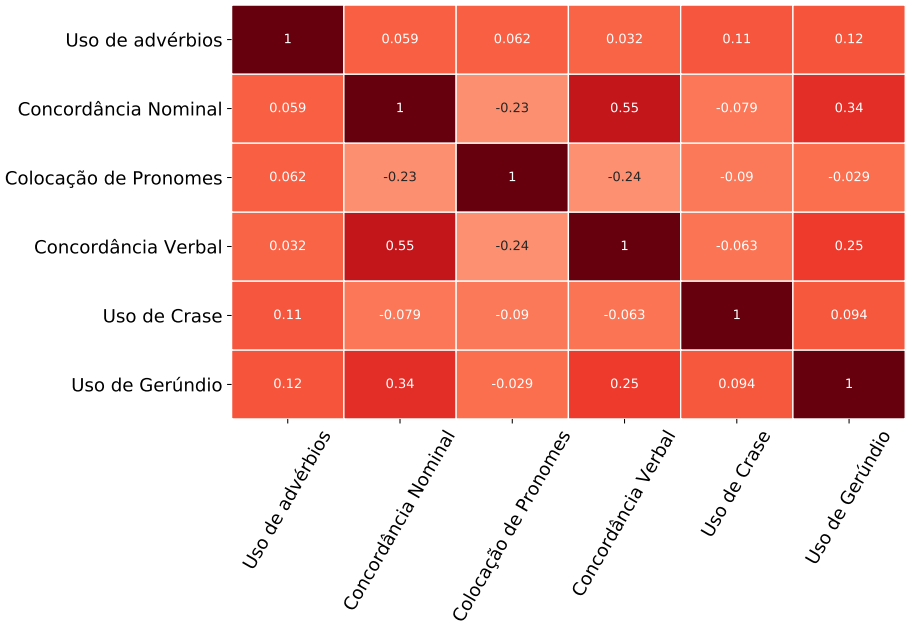
Figura 5.29: Segunda parte da distribuição das variáreis do primeiro grupo encontrado pelo algoritmo *PSOC* para a base do quinto período.

Os baixos valores de erro presentes na distribuição das variáveis reforça a análise da matriz de correlação, e permite inferir que o primeiro grupo encontrado pelo algoritmo *PSOC* para o cenário **5T6A** é composto por alunos que tem em média, um bom rendimento em todos os tópicos.

5.5.2 Cenário 5T6A - Análise do Segundo Grupo

O segundo grupo do cenário **5T6A** apresentou valores mais altos de correlação quanto comparados ao primeiro grupo, o que pode indica que estudantes pertencentes a esse grupo tendem a cometer muito mais erros e a confundir muito mais os tópicos apresentados. As correlações mais altas ocorrem entre as variáveis (7) Concordância Nominal e (10) Concordância Verbal, porém com valores muito maiores do que no primeiro grupo.

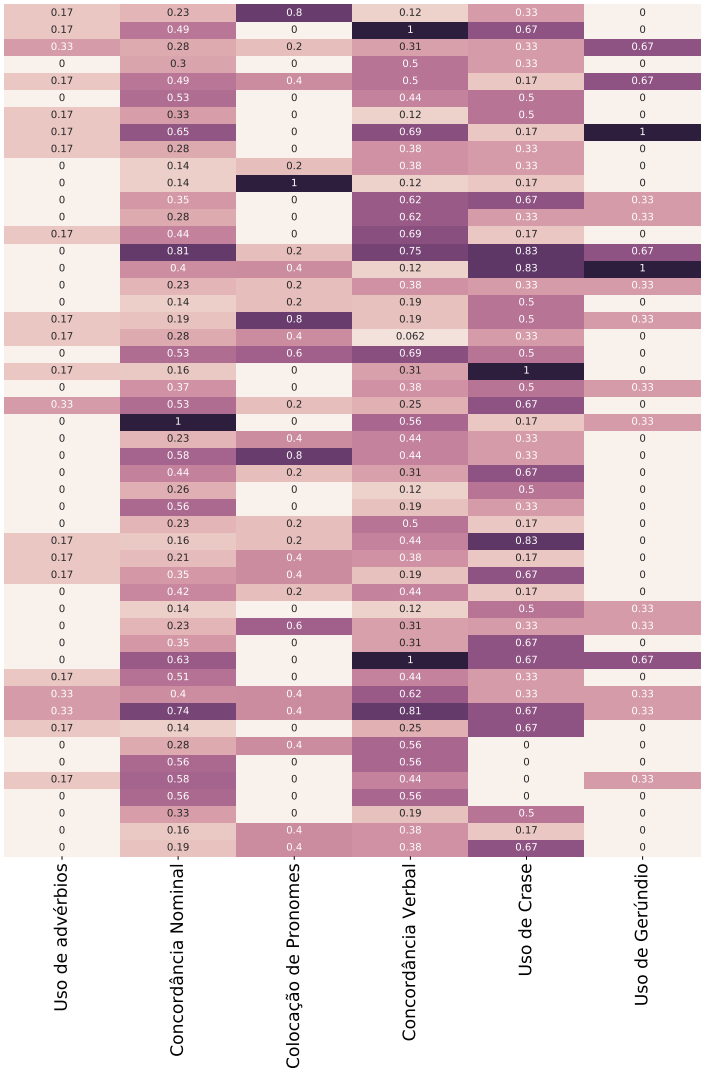
Figura 5.30: Matriz de correlação das variáreis do segundo grupo encontrado pelo algoritmo *PSOC* para a base do quinto período.



Também existe uma alta correlação entre as variáveis (7) Concordância Nominal e (12) Uso de Gerúndio. Observe que a variável (7) Concordância

Nominal aparece nos dois pares de máxima correlação, o que é um indicativo de que os estudantes desse grupo tendem a ter uma dificuldade maior nesse tópico.

Figura 5.31: Distribuição das variáreis do segundo grupo encontrado pelo algoritmo *PSOC* para a base do quinto período.



Ao se analisar a distribuição das variáveis para os estudantes desse grupo pode-se observar que esse grupo possui estudantes com bastante dificuldade nos tópicos referentes as variáveis (7) Concordância Nominal e (10) Concor-

dância Verbal, o que confirma a nossa primeira análise da matriz de correlação. Posteriormente, percebe-se que os alunos desse grupo possuem bastante dificuldade no tópico representado pela variável (12) Uso de Crase. De modo geral, percebe-se que os alunos desse grupo também possuem dificuldade moderada nos demais tópicos.

Os dois grupos encontrados no cenário **5T6A** se mostram complementares entre si. Enquanto o primeiro grupo tem alunos com bom desempenho na maioria dos tópicos, o segundo grupo apresenta estudantes com alta dificuldade na maioria dos tópicos. Pode-se perceber que ambos os grupos apresentam baixos valores de erro para o tópico representado pela variável (1) Uso de Advérbios, o que mostra que os alunos de modo geral tem facilidade nesse conteúdo.

Capítulo 6

Considerações Finais

Este projeto propôs utilizar algoritmos de agrupamento de dados baseados em técnicas de inteligência de enxames, em especial, as derivadas do algoritmo de Otimização por Enxame de Partículas em uma base contendo 20 erros gramaticais comuns cometidos por um conjunto de 250 usuários, do quarto e quinto período, de uma plataforma de educação online de um curso de graduação em Pedagogia de uma universidade brasileira. O agrupamento de dados nesse caso, visa identificar dificuldades em grupos de usuários dessas plataformas com o objetivo de aprimorar e auxiliar as experiências dos usuários no processo de construçãoo do conhecimento.

Embora o uso de técnicas semelhantes na resolução de problemas do mundo real seja uma realidade em vários lugares ao redor do mundo, no Brasil ainda existe pouco investimento em soluções que envolvam o conhecimento dessas áreas, especialmente as plataformas de educação a distância, que ainda possuem dificuldade no acompanhamento dos usuários.

A análise de dados se mostra muito importante nesse processo, permitindo a identificação dos pontos de deficiência dos usuários dessas plataformas, e viabiliza não apenas a atuação dos sistemas de recomendação, mas permite questionamentos sobre complexidades das disciplinas e metodologias de ensino. Este projeto visa mostrar como as técnicas de inteligência de enxames podem ser eficientes em aplicações do mundo real, juntamente com técnicas de mineração de dados, mostrando sua relevância cada vez mais notável na

resolução de problemas.

O algoritmo Otimização por Enxame de Partículas foi escolhido como base de derivação para algoritmos de agrupamento por se mostrar um dos mais usados e mais eficiente algoritmos de inteligência de enxames na literatura. Foram derivados e apresentados nesse projeto, três classes de algoritmos de agrupamento baseados em inteligência de enxames: *Particle Swarm Clustering* (*PSC*), *Particle Swarm Optimization for Clustering* (*PSOC*), e duas técnicas híbridas entre os algoritmos *K-Means* e *Particle Swarm Optimization, (KMPSOC, PSOKM).*

Para avaliar o desempenho dos algoritmos de agrupamento e definir o número apropriado de grupos para a base proposta, foram utilizadas métricas extensamente usadas na literatura: Distância *Intra-Cluster*, Distância *Inter-Cluster*, Erro Quantizado e a Estatística *Gap*. Na análise de desempenho dos algoritmos, o algoritmo *PSOC* se mostrou o mais eficiente por apresentar os melhores resultados nas métricas e os menores valores para desvio padrão, o que mostra uma maior confiança do algoritmo em reportar resultados.

Para otimizar o processo de análise de resultados, o problema de agrupamento foi dividido em dois cenários principais. Os cenários **4T7A** e **5T6A** que correspondem, à aplicação das técnicas de agrupamento baseadas em meta-heurísticas nas bases de dados do quarto e quinto período, respectivamente.

Para o quarto período, os experimentos mostraram que a base de dados proposta deveria ser particionada em três grupos distintos. Dos três grupos encontrados para o cenário **4T7A**, foi possível identificar dois grupos com dificuldades relacionadas, e um terceiro grupo encontrado pelo algoritmo, representando alunos com bom desempenho. Para o cenário **5T6A**, os experimentos mostraram que a base de dados proposta deveria ser particionada em dois grupos distintos. O primeiro grupo encontrado no cenário **5T6A** representa os alunos com bom desempenho nos tópicos apresentados, em contrapartida o segundo grupo encontrado representa os alunos com elevada dificuldade na maioria dos tópicos. Embora tenha sido possível extrair algumas informações no cenário **5T6A**, não foi possível definir um perfil mais detalhados sobre os alunos.

Como trabalhos futuros, pode ser realizada uma análise mais profunda sobre os grupos encontrados, aplicando-se novamente técnicas de agrupamento para refinar os padrões encontrados. Essa investigação pode ser extendida e comparada com o resuldados de algoritmos de agrupamento baseados em outras técnicas de inteligência de enxames, como por exemplo *Artificial Bee Colony (ABC)* e *Fish School Search (FSS)*. Além disso, seria interessante a aplicação de outras abordagens para o problema de agrupamento de dados, como por exemplo, algoritmos de agrupamento hierárquico e *Fuzzy*.

Referências Bibliográficas

[1] Sandra C. M. Cohen and Leandro N. de Castro. Data clustering with particle swarms. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1792–1798. IEEE, 2006.

[2] Shafiq Alam, Gillian Dobbie, Yun Sing Koh, Patricia Riddle, and Saeed Ur Rehman. Research on particle swarm optimization based clustering: a systematic review of literature and techniques. *Swarm and Evolutionary Computation*, 17:1–13, 2014.

[3] Eduardo Raul Hruschka, Ricardo JGB Campello, Alex A Freitas, and André C Ponce Leon F de Carvalho. A survey of evolutionary algorithms for clustering. *IEEE transactions on systems, man and cybernetics. Part C, Applications and reviews*, 39(2):133–155, 2009.

[4] Ashish Dutt, Saeed Aghabozrgi, Maizatul Akmal Binti Ismail, and Hamidreza Mahroeian. Clustering algorithms applied in educational data mining. *International Journal of Information and Electronics Engineering*, 5(2):112, 2015.

[5] Mariana G. M. Macedo and Carmelo J. A. Bastos-Filho. Clustering users based on the capacity to solve questions in an educational platform. *XIII Encontro Nacional de Inteligência Artificial e Computacional*, 2016.

[6] Pedro Santos, Mariana G. M. Macedo, Elliackin Figueiredo, Clodomir J. Santana Jr., Fabiana Soares, Hugo Siqueira, Alexandre Maciel, Anuradha Gokhale, and Carmelo J. A. Bastos-filho. Application of

PSO-Based Clustering Algorithms on Educational Databases. *Latin-American Conference on Computational Intelligence*, 2017.

[7] Agrupamento DEADE, Minera NA, and Educacionais DED. *Use of Clustering Algorithms in the Educational Data Mining*. 2006.

[8] A. Jaya Mabel Rani and Latha Parthiban. Improved particle swarm optimization and k-means clustering algorithm for news article. *Sustainable Energy and Intelligent Systems (SEISCON 2013), IET Chennai Fourth International Conference on*, 2013.

[9] Ching-Yi Chen and Fun Ye. Particle swarm optimization algorithm and its application to clustering analysis. In *Electrical Power Distribution Networks (EPDC), 2012 Proceedings of 17th Conference on*, pages 789–794. IEEE, 2012.

[10] Usama Fayyad, David Haussler, and Paul Stolorz. Kdd for science data analysis: Issues and examples. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 50–56, 1996.

[11] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.

[12] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.

[13] Michael J Berry and Gordon Linoff. *Data mining techniques: for marketing, sales, and customer support*. John Wiley & Sons, Inc., 1997.

[14] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[15] Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD explorations newsletter*, 4(1):65–75, 2002.

[16] Sachin Shinde and Bharat Tidke. Improved k-means algorithm for searching research papers. *International Journal of Computer Science & Communication Networks*, 4(6):197–202.

[17] LV Bijuraj. Clustering and its applications. In *Proceedings of National Conference on New Horizons in IT-NCNHIT*, page 169, 2013.

[18] Barkha Narang, Poonam Verma, and Priya Kochar. Application based , advantageous K-means Clustering Algorithm in Data Mining - A Review. 7(2):121–126, 2016.

[19] Chinedu Pascal Ezenkwu, Simeon Ozuomba, and Constance Kalu. Application of K-Means Algorithm for Efficient Customer Segmentation : A Strategy for Targeted Customer Services. *International Journal of Advanced Research in Artificial Intelligence*, 4(10):40–44, 2015.

[20] Leon Bottou and Yoshua Bengio. Convergence properties of the k-means algorithms. In *Advances in neural information processing systems*, pages 585–592, 1995.

[21] Laurence Morissette and Sylvain Chartier. The k-means clustering technique: General considerations and implementation in mathematica. *Tutorials in Quantitative Methods for Psychology*, 9(1):15–24, 2013.

[22] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[23] G Bel Mufti, P Bertrand, and EL Moubarki. Determining the number of groups from measures of cluster stability. In *Proceedings of international symposium on applied stochastic models and data analysis*, pages 17–20, 2005.

[24] MGH Omran. A pso-based clustering algorithm with application to unsupervised image classification. *University of Pretoria etd*, 2005.

[25] Mahamed GH Omran. *Particle swarm optimization methods for pattern recognition and image processing*. PhD thesis, 2006.

[26] Boris Mirkin. Choosing the number of clusters. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):252–260, 2011.

[27] Chinatsu Arima, Kazumi Hakamada, Masahiro Okamoto, and Taizo Hanai. Validity index for fuzzy k-means clustering using the gap statistic method. In *Sixteenth International Conference on Genome Informatics*, 2005.

[28] Hazem Ahmed and Janice Glasgow. Swarm intelligence: concepts, models and applications. *School Of Computing, Queens University Technical Report*, 2012.

[29] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

[30] Dervis Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

[31] Mohamed Bakhouya and Jaafar Gaber. An immune inspired-based optimization algorithm: Application to the traveling salesman problem. *Advanced Modeling and Optimization*, 9(1):105–116, 2007.

[32] James Kennedy. Particle swarm optimization. In *Encyclopedia of machine learning*, pages 760–766. Springer, 2011.

[33] DW Van der Merwe and Andries Petrus Engelbrecht. Data clustering using particle swarm optimization. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 1, pages 215–220. IEEE, 2003.

[34] Hamed Khoshdel and Barat Saman. A new hybrid learning-based algorithm for data clustering. In *Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on*, pages 095–100. IEEE, 2012.

[35] Fabiana M. B. Soares and Alexandre M. A. Maciel. Desenvolvimento de uma arquitetura utilizando mde para apoiar alunos de ead com dificuldades na gramática portuguesa. *Anais da Mostra de Extensão, Inovação e Pesquisa*, 3, 2016.

[36] Mariana G. M. Macedo, Elliackin M. N. Figueiredo, Fabiana M. B. Soares, Alexandre M. A. Maciel, Anuradha Gokhale, and Carmelo J. A. Bastos-filho. Clustering Students Based on Grammatical Errors for Online Education. *Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais*, 2009.

[37] Satyasai Jagannath Nanda and Ganapati Panda. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary computation*, 16:1–18, 2014.

[38] Cristóbal Romero and Sebastián Ventura. Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6):601–618, 2010.

[39] Catherine A Sugar and Gareth M James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):750–763, 2003.