UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Jessiane Gomes Andrade

Agrupamento de Dados Online via Combinação de Partições

Uberlândia, Brasil

2021

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Jessiane Gomes Andrade

Agrupamento de Dados Online via Combinação de Partições

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Daniel Duarte Abdala

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2021

Jessiane Gomes Andrade

Agrupamento de Dados Online via Combinação de Partições

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 17 de Junho de 2021:

Daniel Duarte Abdala

Elaine Ribeiro de Faria Paiva

Mauricio Cunha Escarpinati

Uberlândia, Brasil
2021

Agradecimentos

Primeiro quero agradecer a Deus por mais esta oportunidade em minha vida de finalizar este curso, segundo a minha família que vem me apoiando durante todo esse percurso e serviram de base para mim, em especial a minha irmã gêmea Jéssica que esteve ao meu lado todo esse tempo. Ao meu companheiro Guilherme que me ajudou a continuar firme e até o momento vem sendo minha base juntamente com a minha família. A todos os meus amigos que foram essenciais nesta jornada, pois me incentivaram a seguir firme nesta estrada. Aos meus professores por me passar o conhecimento, me aconselhar, orientar e me mostrar o melhor caminho a seguir, em especial ao meu orientador, professor e amigo Daniel Abdala por toda paciência e dedicação ao longo desse trabalho.

"Uma experiência nunca é um fracasso, pois sempre demonstra algo."
(Thomas Edison)

Resumo

Este trabalho tem como objetivo avaliar o impacto da utilização de técnicas consensuais de agrupamento de dados no contexto de agrupamento de fluxos de dados. Três algoritmos de agrupamento de dados online foram utilizados neste estudo. Adicionalmente três funções consensuais foram utilizadas. A adaptação das funções consensuais e da métrica de distância utilizada foi necessária pois estas originalmente funcionavam para dados em *batch*. Um experimento foi desenvolvido para avaliar o impacto da sua utilização conjunta. Os resultados preliminares apontam que as funções consensuais podem entregar resultados confiáveis mesmo em situações de envelhecimento de dados.

Palavras-chave: *Clustering*, *Ensemble Clustering*, *Online Clustering*, Agrupamento de dados, Fluxo de Dados, *Data Stream*, Agrupamento de Dados Online, Combinação de Partições.

Abstract

Ensemble Clustering in Data Stream

This paper aims to evaluate the impact of using consensual data clustering techniques in the context of clustering data streams. Three online data clustering algorithms were used in this study. Additionally, three consensual functions were used. The adaptation of the consensus functions and the distance metric used was necessary as these originally worked for batch data. An experiment was developed to assess the impact of their joint use. Preliminary results indicate that consensual functions can deliver reliable results even in data aging situations.

Keywords: Clustering, Ensemble Clustering, Online Clustering, Data Stream.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ADO	Agrupamento de Dados Online
ADCP	Agrupamento de Dados via Combinação de Partições
AG	Algoritmos Genéticos
BoK	*Best of K*
BOEM	*The Best One Elements Move*
CF	*Cluster Feature*
DS	*Data Stream*
FP	Falsos Positivos
FN	Falsos Negativos
FMN	Fatoração de Matrizes não Negativas
PCPK	Partições Consensuais Ponderadas via Kernel
PPD	Agrupamento com base em Programação Pré Definida
SoD	Soma das Distâncias
RI	Índice Rand ou *Rand Index*
VP	Verdadeiros Positivos
VN	Verdadeiros Negativos

Sumário

1 Introdução

Nas últimas décadas acompanhamos um grande crescimento em armazenamento de dados ocasionados por pessoas e dispositivos sempre conectados. Com o avanço das tecnologias, principalmente de *hardware* e *software*, armazenar e manipular informações ficou cada vez mais fácil e acessível. Com esta grande quantidade de dados, torna-se necessário extrair o máximo possível de informações relevantes. A maioria dos dados acumulados sempre trazem padrões e informações que podem ser analisadas de perto e existem métodos que podem ajudar na extração destas informações. Isso é importante porque esses dados possuem informações que podem ajudar nas tomadas de decisões relacionadas a um tipo de negócio, como por exemplo: recomendações de compras para clientes, análises clínicas de pacientes, identificação de atividades suspeitas, entre outras.

Agrupamento de dados (*Clustering*) é uma técnica importante na análise exploratória de dados (análise de conjuntos de dados para resumir suas principais características, muitas vezes com métodos visuais). Essa técnica consiste em organizar um conjunto de dados por estruturas subjacentes através de um agrupamento individual ou hierarquia de grupos. Esse conceito é muito utilizado em vários campos de estudo, principalmente na estatística, onde é importante a análise de padrões em um conjunto de informações. É uma ferramenta para explorar estruturas de dados que não requer uma hipótese comum à maioria dos métodos estatísticos e costuma ser chamado de aprendizagem não supervisionada (onde não existe um professor, o sistema deve descobrir sozinho as relações e padrões) (JAIN; DUBES et al., 1988).

Clustering é justamente a técnica de mineração de dados para fazer agrupamentos automáticos de dados segundo seu grau de semelhança. O critério de semelhança faz parte da definição do problema e do algoritmo. Esta técnica está sendo utilizada em uma grande diversidade de aplicações como *data mining*, segmentação de imagens, classificação de padrões (doenças), entre outras. Existem muitos algoritmos de agrupamentos: *KNN*, *K-means*, *single link*, *fuzzy c-means*, entre outros; entretanto, não há um algoritmo de *clustering* capaz de agrupar corretamente qualquer conjunto de dados (VEGA-PONS; RUIZ-SHULCLOPER, 2011). Isso se dá porque cada algoritmo de agrupamento de dados possui uma característica que indica qual tipo de dado ele consegue ser mais assertivo no processamento, então é necessário ter informações suficientes, como a distribuição do dado, se possui muitos *outliers*, se é um dado hierárquico ou talvez a quantidade de grupos existentes para poder fazer a escolha do melhor algoritmo. (XU; TIAN, 2015).

Para conjuntos de dados muito grandes é comum realizar uma prática de *batch clustering* que consiste em dividir o conjunto de dados em conjunto menores de tamanho

fixo com o intuito de reduzir o custo computacional, pois não utiliza toda a base de dados em cada iteração, havendo uma combinação ao fim para se obter o resultado (ALONSO, 2013). *Online Clustering* é uma prática com o mesmo objetivo do *batch clustering*, porém o conjunto de dados está em constante crescimento, ou seja, contém um fluxo de dados. Formalmente, um fluxo de dados S é uma enorme sequência de dados x^1, x^2 , ..., x^N, isto é, $S = \{x^i\}_{i=1}^{N}$, que é potencialmente ilimitado ($N \to \infty$) . Cada objeto é descrito por um vetor de atributo n-dimensional $x^i = \left[x_j^i\right]_{j=1}^{n}$ que pertence a um espaço Ω que pode ser contínuo, categórico ou misto (SILVA et al., 2013). Em outras palavras, novos objetos são adicionados ao conjunto de dados regularmente.

Porém a grande maioria dos algoritmos de agrupamento de dados conhecidos atualmente não são adequados para um modelo online, já que eles trabalham com a suposição de que todos os dados estão disponíveis desde o início. A maioria dos algoritmos de *Online Clustering* também não mantém os objetos recebidos, eles executam o processamento acerca do lote, produzem um resultado e o descartam logo em seguida, por isso não conseguem manter continuamente um bom e consistente agrupamento da sequência observada. Além disso, conjuntos de dados estáticos sempre podem ser identificados com uma distribuição de dados específica. Consequentemente, pode se aplicar um algoritmo de agrupamento eficiente para este tipo de distribuição em particular, porém com a grande mudança que ocorre em *stream* de dados, nem sempre a distribuição é mantida, ela pode variar, e esses algoritmos não são projetados para se adaptarem a mudanças nas características dos dados, tornando-se assim ineficientes.

Por outro lado, *ensemble clustering* ou combinação de partições, consiste em gerar um conjunto de agrupamentos (*clusterings*) a partir de um mesmo conjunto de dados e depois combinar os resultados em um agrupamento final. Este novo método de executar *clustering* vem ganhando espaço pois permite a utilização de algoritmos diferentes (ou pontos de começo diferentes para o mesmo algoritmo). Como o processo integra informações sobre todas as partições na combinação o mesmo elimina possíveis erros individuais de cada agrupamento, demonstrando assim uma solução apropriada com maior confiabilidade (VEGA-PONS; RUIZ-SHULCLOPER, 2011). Este método de agrupamento é utilizado em conjuntos de dados conhecidos.

Este projeto visa explorar e estudar a possibilidade de utilizar o conceito de *Ensemble Clustering* no contexto de *Online Clustering* com o objetivo de melhorar a confiabilidade dos resultados e melhorá-los potencialmente.

1.1 Motivação

Como demonstrado anteriormente, a maioria dos algoritmos para Agrupamento de Dados online (ADO) não são precisos, pois eles não levam em consideração as mudanças

que podem ocorrer no conjunto e fica inviável o reprocessamento já que os dados não são mantidos para análise posterior. Logo, é possível que um resultado mais consistente seja obtido pela combinação dos resultados de vários algoritmos de agrupamento do que um em particular.

1.2 Hipótese

Em agrupamento de dados online, assim como no contexto onde os dados já são conhecidos, a confiabilidade dos resultados pode ser melhorada utilizando o método de combinação de partição ao invés dos resultados que são produzidos por algoritmos individualmente.

1.3 Justificativa

Nos últimos anos, a necessidade de analisar dados em tempo real vem se tornando grande e expressiva. Empresas investem pesado em criação de sistemas *big data*, como o *Netflix* que construiu um sistema de recomendação, que retorne respostas com base nos dados adquiridos de forma mais rápida e precisa.

Agrupamento de dados online é uma das principais ferramentas para exploração de dados no contexto de *Big Data* e possuem muitos problemas com sua confiabilidade e qualidade que são ignorados. Logo, explorar maneiras de melhorar a sua confiabilidade e qualidade pode maximizar a aplicabilidade destas ferramentas em conjunto de dados que estão em constante expansão.

Sendo assim, esse trabalho se justifica em utilizar métodos de combinação de partições em agrupamento de dados online procurando aumentar a precisão dos resultados obtidos.

1.4 Objetivos

O objetivo geral deste trabalho refere-se a investigar a possibilidade de introduzir ideias de *Ensemble Clustering* no contexto de *Online Clustering*, ou seja, utilizar a ideia de consenso de múltiplos classificadores na decisão da classe de padrões advindos de *streams* de dados.

Como objetivos específicos elencam-se:

- Revisão do estado da arte;
- Seleção de um conjunto de dados representativo para os testes;

- Implementação de diversos algoritmos de *Online Clustering*;
- Adaptação dos métodos de *Ensemble Clustering*, em especial suas funções consensuais para atuar no método online;
- Elaboração e execução do protocolo de testes.

O restante do texto se organiza como segue: no capítulo 2 será apresentada uma revisão bibliográfica dos trabalhos relacionados e relevantes para a discussão deste texto; o capítulo 3 apresenta um resumo sobre os principais tópicos relevantes ao desenvolvimento e entendimento desse trabalho; o capítulo 4 explicará toda metodologia desenvolvida para se verificar a utilização do método de combinação de partição no contexto de agrupamento online; o capítulo 5 falará um pouco sobre os conjuntos de dados e os experimentos utilizados e, consequentemente, terá um tópico para conclusão do trabalho.

2 Revisão Bibliográfica

Ao longo da pesquisa para embasamento deste projeto, nenhum trabalho que utiliza técnicas de agrupamento de dados via combinações de partições no contexto de agrupamento de dados online foi encontrado. Porém, neste trabalho serão abordados tópicos muito importantes da teoria de agrupamento de dados: *Online Clustering* (Agrupamento de dados online) e *Ensemble Clustering* (Agrupamento de dados via combinação de partições).

Em (XU; TIAN, 2015) temos uma definição sobre agrupamento de dados, segue:

- Os objetos no mesmo *cluster* devem ser semelhantes o máximo possível;
- Os objetos em diferentes *clusters* devem ser o mais diferente possível;
- Medidas de similaridade e dissimilaridade devem ser claras e ter um significado prático.

Além disso, os autores discutem sobre os elementos básicos envolvidos no processo de agrupamento como as medidas de distância ou similaridade e os indicadores de avaliação. Ele analisa os dois grupos de algoritmos considerados como tradicionais e os modernos, levando em conta os pontos fortes e fracos para cada um mencionado nesses grupos e apresenta uma tabela de comparação para compreensão entre os algoritmos citados.

Continuando em agrupamento de dados, temos o livro (JAIN; DUBES et al., 1988) onde se tem uma introdução, também, sobre o assunto. O texto comenta algoritmos informais e suas aplicações, além de introduzir algumas ferramentas para avaliar os resultados dos agrupamentos e a exploração do dado. Logo no início do livro, comenta sobre a importância da representação correta dos dados, enfatizando que os algoritmos de agrupamento devem combinar com o tipo de dado, se esse não for corretamente compreendido pode levar a interpretação errônea do resultado do agrupamento. Aplicando algoritmos de agrupamento em processamento de imagens e fazendo a interpretação dos resultados obtidos.

Temos uma abordagem mais ampla por (AGGARWAL; REDDY, 2014), o qual introduz os principais métodos de agrupamentos e reflete sobre diferentes domínios de problemas e cenários como dados multimídia, textos, dados biológicos, dados categóricos, dados online, entre outros. Também dá uma atenção especial a problemas recentes como análise de dados em redes sociais.

Em *Online Clustering* o artigo (SILVA et al., 2013) faz uma introdução sobre agrupamento de dados online, sumarizando os estudos e técnicas desenvolvidas, apresentando uma visão geral das metodologias experimentais usualmente utilizadas, focando em assuntos relevantes que não foram cuidadosamente considerados na literatura como: i) fornecer uma taxonomia que permite ao leitor identificar cada trabalho pesquisado com relação a aspectos importantes no agrupamento de fluxo de dados; ii) analisar a influência do tempo no agrupamento de dados online; iii) analisar as metodologias experimentais usualmente empregadas na literatura; e iv) fornecer uma série de referências que descrevem aplicações de *clustering* de *stream* de dados em diferentes domínios como detecção de ataques, análise do mercado de ações e monitoramento de redes.

O livro (GAMA, 2010) é totalmente dedicado a técnicas de aprendizado online, ele contém um capítulo inteiro sobre agrupamentos de dados online onde aborda suas principais estruturas. Discorre sobre os principais métodos utilizados nesse campo como: agrupamentos de partições, agrupamentos hierárquicos, *micro clustering* e *grid clustering*.

Em *Ensemble Clustering* também existem vários *surveys* que o sumarizam. (GHAEMI et al., 2009) apresenta os desafios e taxonomias, descreve sobre funções de consenso e combinação de partições incluindo *Hypergraph Partitioning*, *Voting Aproach*, Informação Mútua, Funções Baseadas na Co-associação e *Finite Mixture Model* explicando suas vantagens e desvantagens e a complexidade computacional e compara as características dos algoritmos de combinação de agrupamentos como complexidade computacional, robustez, simplicidade e precisão em diferentes conjuntos de dados em técnicas anteriores.

(VEGA-PONS; RUIZ-SHULCLOPER, 2011) apresenta uma visão sobre os métodos de combinação de agrupamentos, discute as características destes métodos para ajudar na seleção mais adequada para a solução. Apresenta uma taxonomia dessas técnicas e comenta a importância de levar em consideração as particularidades do problema na hora da escolha do método.

Além disso, foram levados em consideração trabalhos como a tese de doutorado (ABDALA, 2010), que explora os métodos fazendo uma referência simples e precisa dos desenvolvimentos e trabalhos anteriores, mostrando uma visão abrangente do estado da arte da combinação de agrupamentos, além de comentar os desenvolvimento futuros, tendo como objetivo discutir problemas onde a escolha do melhor método de agrupamento é desconhecido. Também apresenta uma avaliação do desempenho do conjunto de técnicas de combinação no contexto de segmentação de imagens. Introduz um método de agrupamento baseado em uma nova formulação para o problema da partição mediana, em que o desempenho deste método é avaliado em relação a outros métodos bem conhecidos de agrupamento. Um novo método de agrupamento é introduzido por mesclar *ensemble* e *constrained clustering*. Apresenta uma revisão do protocolo de imagem, conhecido como tensor de difusão, e uma nova metodologia de segmentação de fibra é proposta para ori-

entar o processo de segmentação semi-supervisionado.

3 Fundação Teórica

Neste capítulo serão revisados os principais tópicos necessários para o desenvolvimento e avaliação deste trabalho.

3.1 Agrupamento de Dados Online

Os algoritmos de *online clustering* são úteis em uma ampla gama de aplicações. Com tais algoritmos podemos realizar a detecção de tópicos em mídia de transmissão e agrupamento de padrões climáticos para detectar eventos extremos como ciclones e o mercado de ações. Em sua maioria, são aplicados a problemas que envolvem um fluxo de dados, incluindo previsão, tomada de decisão em tempo real e aprendizados com custo restrito. Esses fluxos de dados podem assumir várias formas, até mesmo um conjunto de dados tão grande que os algoritmos só podem acessá-lo sequencialmente (CHOROMANSKA; MONTELEONI, 2012).

Extrair conhecimento de um fluxo de dados contínuo é um grande desafio. Grande parte das técnicas de mineração de dados supõe que existe uma quantidade finita de dados para ser trabalhada, a qual é armazenada fisicamente e analisada em várias etapas por um único algoritmo em modo *batch*. Porém, para realizar a mineração de dados em cima de um fluxo de dados (*Data Stream*) deve-se levar em consideração as seguintes restrições:

1. Os dados são recebidos de forma contínua;
2. Não há um controle sobre a ordem em que os os dados devem ser processados;
3. O fluxo de dados é, potencialmente, de um tamanho ilimitado;
4. Os objetos de dados são descartados após serem processados (embora ainda seja possível armazenar parte da informação por um período de tempo utilizando um mecanismo de esquecimento para descartar mais tarde);
5. A distribuição de probabilidade do processo de geração de dados desconhecidos pode mudar ao longo do tempo.

Tendo estas restrições em mente, precisamos que nossos algoritmos:

- Forneçam resultados através de um processamento rápido e incremental de objeto de dados;

- Adaptem-se rapidamente à mudança dos dados, detectando quando novos *clusters* podem aparecer ou outros desaparecer;
- Gerem uma representação do modelo que seja compacta e que não cresça com a quantidade de objetos processados (nem mesmo de forma linear);
- Detectar e corrigir rapidamente *outliers* (valores que estão muito afastados dos demais da série, aparentemente inconsistentes).

Podemos dividir o algoritmo em duas etapas (*online* e *offline*) onde na primeira etapa é feita uma abstração para resumir o fluxo de dados com a ajuda de algumas estruturas de dados, com o intuito de contornar problemas e restrições relacionados a memória e espaço das aplicações de fluxo contínuo. Essas estruturas de dados resumem o fluxo de dados afim de preservar o significado dos objetos originais sem ter a necessidade de armazená-los. Para as mesmas, podemos ter *Prototype Arrays* (vetores de objetos), *Corset Trees*, *Data Grids* e *Feature Vectors*, sendo o último o que daremos mais atenção, pois foi a estrutura escolhida para utilizarmos neste trabalho. Também comentaremos brevemente os algoritmos de agrupamento de dados online escolhidos que utilizam essa estrutura, sendo estes: *Single Pass K Means*, *Birch* e *CluStream*. A segunda etapa do algoritmo é o que chamamos de *Clustering Step* (SILVA et al., 2013).

3.1.1 Feature Vectors

(GAMA, 2010)Um *feature vector* ou um *cluster features* (CF), é uma representação compacta de um conjunto de objetos. É uma tripla (N, LS, SS), usada para guardar as estatísticas suficientes de um conjunto de dados:

- **N** é o número de objetos no conjunto de dados observado;
- **LS** é um vetor de mesma dimensão do conjunto de dados, que guarda a soma linear dos N objetos;
- **SS** é um vetor da mesma dimensão do conjunto de dados, que guarda a soma quadrática dos N objetos.

As propriedades dos *cluster features* são:

- **Incremental** - Se um ponto **x** é adicionado ao *cluster*, as estatísticas são atualizadas da seguinte forma:

$$LSa \leftarrow LSa + x$$

$$SSa \leftarrow SSa + x^2$$

$$Na \leftarrow Na + 1$$

- **Aditivo** - Se **A** e **B** são conjunto disjuntos, a união deles é igual a soma de suas partes. A propriedade aditiva nos permite mesclar *sub-clusters* incrementalmente.

$$LSc \leftarrow LSa + LSb$$

$$SSc \leftarrow SSa + SSb$$

$$Nc \leftarrow Na + Nb$$

Um *feature vector* tem informação suficiente para calcular as normas L_1 (distância de Manhattan) e L_2 (distância Euclidiana):

$$L_1 = \sum_{i=1}^{n} |x_{ai} + x_{bi}|$$

$$L_2 = \sqrt{\sum_{i=1}^{n} (x_{ai} - x_{bi})^2}$$

- **Centróide**, definido como centro de um *cluster*:

$$\vec{X}0 = \frac{LS}{N}$$

- **Raio**, define a distância média entre os objetos e o centróide:

$$R = \sqrt{\frac{\sum_{1}^{N} (\vec{X}_i - \vec{X})^2}{N}}$$

3.1.2 Single Pass K-Means

K-means é um dos algoritmos de agrupamentos mais utilizados. Ele constrói uma partição de um conjunto de objetos em k *clusters*, de modo a minimizar alguma função objetiva, usualmente uma função de erro quadrático, o que implica em *clusters* arredondados. Um parâmetro k é fixado limitando a aplicabilidade real à transmissão e à evolução dos dados (GAMA, 2010). Existem muitas variações e utilizações do k-means no cenário de *data stream* (O'CALLAGHAN et al., 2002).

(FARNSTROM; LEWIS; ELKAN, 2000) propõe o algoritmo de *Single Pass K-Means*. A ideia principal é usar um *buffer* onde os pontos do conjunto de dados são mantidos de forma comprimida. O fluxo de dados é processado em blocos. Todo o espaço disponível no buffer é preenchido com pontos do fluxo. Usando estes pontos, localiza-se k centros de tal forma que a Soma das Distâncias (SoD) dos pontos de dados para o centro mais próximo seja minimizada. Somente os k centróides (que representam os resultados de agrupamento) são mantidos, com os k *cluster features* correspondentes. Nas seguintes iterações, o *buffer* é inicializado com os k-centróides, encontrados na iteração anterior, ponderados pelos k *cluster features* e pontos de dados recebidos do fluxo (FARNSTROM; LEWIS; ELKAN, 2000). O *single-pass k-Means* é incremental, melhorando a solução dada dos dados adicionais e também utiliza um tamanho fixo (GAMA, 2010).

3.1.3 BIRCH

O sistema BIRCH (Redução Iterativa Equilibrada e *Clustering* usando hierarquias) (ZHANG; HEWITT, 1997) , comprime os dados, criando uma estrutura hierárquica, *CF-tree*, onde cada nó é uma tupla (Cluster Feature), que contém estatísticas suficientes para descrever um conjunto de dados, e comprimir todas as informações em uma árvore composta de CF. Foi projetado para grandes conjunto de dados, levando em consideração as restrições de tempo e memória (GAMA, 2010). BIRCH requer dois parâmetros definidos pelo usuário: B o fator de ramificação ou o número máximo de entradas em cada nó não folha, e T o diâmetro máximo (ou raio) de qualquer CF em um nó folha. BIRCH tenta achar os melhores grupos em relação a memória disponível, enquanto minimiza a quantidade de entrada e saída. A *CF-tree* cresce por agregação, obtendo apenas uma passagem sobre os dados como resultado da complexidade O(N) (GAMA, 2010).

3.1.4 Leader Clustering

(SILVA et al., 2013) O Leader Clustering é um algoritmo de um passo só e que não requer informações prévias para o numero de clusters. No entanto é um algoritmo instável, seu desempenho depende da ordem dos dados e também do conhecimento da distância correta.

3.2 Medidas de Validação

(WARRENS; HOEF, 2019) Em aprendizado de máquina não supervisionado existe uma grande necessidade das partições geradas pelos algoritmos de agrupamento serem comumente avaliadas com os chamados índices de validação externa. Pesquisadores tendem a usar e reportar índices que quantificam a concordância entre duas partições.

Distância	Métrica
Rand	Não
Rand Ajustado	Não
Jaccard	Não
Mirkin	Sim
Medida-F	Não
Folks&Mallows	Não
Dongen	Sim
Correspondência do gráfico bipartido	Não
Informação Mutua	Não
Variação da Informação	Sim
Taxa de Erro	Não

Tabela 1 – Lista de medidas de similaridade

Para tal, medidas de similaridade são utilizadas. O objetivo de uma função de similaridade é proporcionar uma medida de quão semelhante ou diferente duas partições são. Medidas de Validação é um dos conceitos fundamentais nesse trabalho.

Esta subseção revisa alguns detalhes dessas medidas. A Tabela 1 lista as medidas de similaridade e, além disso, indica se elas são métricas. (ABDALA, 2010) Medidas de similaridades são valiosas pois melhoram a compreensão humana ao contrário de uma função de distância arbitrária. Para que uma função de distância seja considerada uma métrica deve obedecer às quatro condições dadas:

1. $d(x, y) \geq 0$ (não pode conter números negativos);
2. d(x, y) = 0 se e somente se x = y;
3. d(x, y) = d(y, x) (simetria);
4. $d(x, z) \leq d(x, y) + d(y, z)$ (desigualdade dos triângulo).

Existem várias maneiras de medir a similaridade entre duas partições. Elas podem ser classificadas como:

- Contagem de Pares;
- Correspondência de conjuntos;
- Teoria da Informação.

3.2.1 Contagem de Pares

Dado um conjunto de objetos S=$\{O_1, ..., O_n\}$, supondo que partições foram geradas por classificadores arbitrários, logo U = $\{u_1, ..., u_R\}$ e V = $\{v_1, ..., v_C\}$ representam duas partições de S, logo U e V são subconjuntos de S; $U_{i=1}^{R} u_i = S = V_{j=1}^{C} v_i$; $u_i \bigcap u_{i'} = 0 = v_i \bigcap v_{i'}$ para $1 \leq i \neq i' \leq R$ e $1 \leq j \neq j' \leq C$. Com essas duas partições U e V podemos escrever em forma de tabela de contingência com $n_{i.}$ e $n_{.j}$ referindo-se respectivamente a quantidades de objetos na classe u_i e v_j assim como na Tabela 2 (HUBERT; ARABIE, 1985).

A tabela de contingência resumirá os resultados dos testes de classificação e também permite visualizar o desempenho de um algorítimo de classificação, onde relata o número de Falsos Positivos (FP), Falsos Negativos (FN), Verdadeiros Positivos (VN) e Verdadeiros Negativos (VN). Isso permite uma análise mais detalhada do que a mera proporção de classificações corretas.

(HUBERT; ARABIE, 1985) As medidas de correspondência entre U e V sobre os pares de objetos são classificados na Tabela 2 de contingência RxC. Especificamente,

existem 4 tipos diferentes entre os pares distintos $\binom{n}{2}$ (número de pares não ordenados em um conjunto de N elementos):

- N_{11} - Pares de objetos que são colocados na mesma classe em U e na mesma classe em V;
- N_{00} - Pares de objetos que são colocados em diferentes classes em U e em diferente classes em V;
- N_{10} - Pares de objetos que são colocados na mesma classe em U e em diferente classes em V;
- N_{01} - Pares de objetos que são colados em diferentes classes em U e na mesma classe em V.

Os 4 tipos de contagem sempre satisfazem a igualdade:

$$N_{11} + N_{00} + N_{10} + N_{01} = \frac{N(N-1)}{2}$$

		Partição V				
	Classes	v_1	v_2	...	v_C	Somas
Partição U	u_1	n_{11}	n_{12}	...	n_{1C}	$n_{1.}$
	u_2	n_{21}	n_{22}	...	n_{2C}	$n_{2.}$
	.	.	.		.	.
	.	.	.		.	.
	.	.	.		.	.
	u_R	n_{R1}	n_{R2}	...	n_{RC}	$n_{R.}$
	Somas	$n_{.1}$	$n_{.2}$	...	$n_{.C}$	$n_{..}$ = n

Tabela 2 – Notação para comparar duas partições (HUBERT; ARABIE, 1985)

N_{11} e N_{00} são considerados as concordâncias nas classificações dos pares de objetos enquanto N_{10} e N_{01} representa as divergências entre os mesmo. Obviamente se C representa o total de concordâncias e D o total de divergências, logo C + D = $\binom{n}{2}$, onde um coeficiente binomial é considerado igual a 0 quando m = 0 ou m = 1 em $\binom{m}{2}$. Intuitivamente duas partições são similares quando produz valores relativamente grande para C e pequenos valores para D.

As fórmulas explicitadas na tabela 3, a seguir, podem ser obtidas para expressar o número de pares de cada tipo como uma função de n, $n_{i.}$, $n_{.j}$ e n_{ij}.

Tipo	Formula
N_{11}	$\frac{1}{2}\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}(n_{ij}-1)$
N_{00}	$\frac{1}{2}(n^2+\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2-(\sum_{i=1}^{R} n_{i.}^2+\sum_{j=1}^{C} n_{.j}^2))$
N_{01}	$\frac{1}{2}(\sum_{j=1}^{C} n_{.j}^2-\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2)$
N_{10}	$\frac{1}{2}(\sum_{i=1}^{R} n_{i.}^2-\sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2)$

Tabela 3 – Fórmulas para os 4 tipos de pares de objetos (HUBERT; ARABIE, 1985)

Várias medidas de similaridades são baseadas nesse método de contagem de pares. A definição dos mais relevantes é apresentada nos parágrafos a seguir.

Índice Rand: O índice Rand (RAND, 1971) é uma medida de semelhança que permite a avaliação de algoritmos de agrupamento. Ele é feito comparando duas partições, sendo uma onde os valores verdadeiros são conhecidos. Isso dá uma medida de semelhança dentro de um intervalo [0,1]. O valor 0 é produzido nos casos em que as duas partições comparadas são completamente diferentes. É definido como:

$$Rand(P_1, P_2) = \frac{N_{11} + N_{00}}{(N(N-1))/2}$$

É importante lembrar que o índice de Rand não é normalizado pelo acaso, logo uma versão chamada índice Rand ajustado proposto por (HUBERT; ARABIE, 1985) é dado para sanar essa limitação. É definida a seguir:

$$ARand(P_1, P_2) = \frac{Rand(P_1, P_2) - E[Rand]}{1 - E[Rand]}$$

O índice Rand ajustado tem um intervalo maior [-1, 1] onde 1 é obtido quando duas partições são idênticas.

Índice Jaccard: O Índice de Jaccard (BEN-HUR; ELISSEEFF; GUYON, 2001) dá uma medida de similaridade entre o intervalo de [0, 1]. É definido a seguir:

$$J(P_1, P_2) = \frac{N_{11}}{N_{11} + N_{10} + N_{01}}$$

Distância Mirkin: A distância de Mirkin (MIRKIN, 1998) é uma versão ajustada do índice de Rand. É importante enfatizar que a distância de Mirkin é de fato uma métrica. Onde 0 é obtido se duas partições são iguais e valores positivos caso não sejam. É definida a seguir:

$$M(P_1, P_2) = 2(N_{10} + N_{01})$$

Medida-F: A medida-F (BASU; BANERJEE; MOONEY, 2004), baseada nas medidas de *precision* e *recall*, também pode ser calculada utilizando a comparação de pares. É definida a seguir:

$$Precision(P_1, P_2) = \frac{N_{11}}{N_{10}} \quad Recall(P_1, P_2) = \frac{N_{11}}{N_{01}}$$

$$Medida - F(P_1, P_2) = \frac{Precision \times Recall}{Precision + Recall}$$

Fowlkes & Mallows: Fowlkes & Mallows (FOWLKES; MALLOWS, 1983) introduz o índice a seguir:

$$F(P_1, P_2) = -1\sqrt{W_1(P_1, P_2) \times W_2(P_1, P_2)}$$

Onde W_1 e W_2 são valores calculados como mostrado abaixo. Este índice retorna um valor no intervalo de [0, 1].

$$W_1(P_1, P_2) = \frac{N_{11}}{\sum_{i=1}^{k} N_i \times (N_i - 1)/2} \qquad W_2(P_1, P_2) = \frac{N_{11}}{\sum_{j=1}^{l} N_j \times (N_j - 1)/2}$$

Este trabalho irá utilizar o método de comparação de pares como uma das medidas de similaridade, portanto não entrará em detalhes para correspondência de conjuntos e teoria da informação.

3.2.2 Correspondência de Conjuntos

Índices de correspondência de conjuntos são baseados nas correspondências de todo o *cluster*, usualmente utilizando as partes correspondentes do *cluster* enquanto ignora as partes que não correspondem. São eles:

Taxa de erro: É uma medida direta que dá a porcentagem de padrões erroneamente classificadas em comparação com um resultado verdadeiro conhecido. A taxa de erro é calculada comparando a informação do resultado verdadeiro disponível com a partição a ser comparada. A correspondência entre os dois conjuntos é estabelecida calculando todas as permutações possíveis dos *labels*. Um modo eficiente para calcular as permutações de rótulo é conseguida por meio do algoritmo de húngaro (FRANK, 2005).

Índice Dongen: O índice Dongen (DONGEN, 2000) é uma distância com base no confronto de conjuntos. Ela converte o valor máximo apenas se as duas partições são exatamente as mesmas. Também foi provado que esta distância é uma métrica.

Bipartite Graph Matching: Esse índice calcula uma correlação um-para-um entre os agrupamentos de elementos de imagem, tentando maximizar seu relacionamento (JIANG et al., 2006).

3.2.3 Teoria da Informação

Índices da teoria da Informação são baseados no conceito de informação mútua. Esses índices avaliam a diferença de informação entre duas partições. Segue:

Informação mútua: A informação mútua (STREHL; GHOSH; MOONEY, 2000) é um índice amplamente utilizado. Ele mede o quanto a informação é compartilhada entre as duas variáveis analisadas.

Variação da informação: Mede quanta informação é ganha ou perdida ao mudar a variável aleatória de *P1* para *P2* (MEILĂ, 2003).

3.3 Agrupamento de Dados via Combinação de Partições

(ABDALA, 2010) Combinação de agrupamentos, mais conhecido como agrupamento de dados via combinações de partições, emergiu como uma opção válida no agrupamento de dados que lida de uma forma elegante com o problema de escolher o agrupamento de dados em casos que pouco ou nada é conhecido sobre o conjunto, esse método é representado na Figura 1. Quando aplicamos um algorítimo de agrupamento a um conjunto de dados, este impõe uma organização aos dados que segue um critério interno, portanto esta técnica funciona como uma forma de suavizar o resultado final quando diferentes partições pode potencialmente apresentar diferentes distribuições. É dividida em dois principais processos: **Geração de partições a partir do conjunto de dados original** e **Combinações das partições em um resultado final**. Logo combinações de partições podem ser utilizadas quando:

1. A distribuição de dados não é conhecida;
2. Para suavizar os resultados onde um algoritmo de agrupamento adequado não pode ser identificado;
3. Para melhorar o resultado final do agrupamento, reunindo informações entre diferentes partições.

Alguns trabalhos (FRED; JAIN, 2005),(TOPCHY et al., 2004) tentaram definir um conjunto de propriedades que aprovam o uso de métodos de agrupamento de conjuntos, as 4 propriedades a seguir são selecionadas conforme seu grau de relevância:

- **Robustez** - o Agrupamento de Dados via Combinação de Partições (ADCP) deve apresentar melhor desempenho geral do que qualquer um dos algoritmos de agrupamento individuais usados para gerar as partições no conjunto;

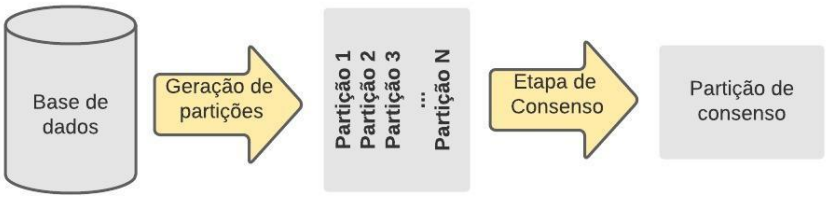

Figura 1 – Agrupamento de Dados via Combinação de Partições (ABDALA, 2010)

- **Consistência** - o resultado consensual deve ser de alguma forma muito semelhante a todas as partições únicas combinadas no conjunto;
- **Novidade** - os métodos de agrupamento de conjuntos devem ser capazes de alcançar resultados inatingíveis por qualquer algoritmo de agrupamento tradicional;
- **Estabilidade** - os resultados consensuais devem apresentar menor sensibilidade ao ruído e *outliers*.

Os passos para agrupamento de dados via combinações de partições são:

- **Geração de partições** é responsável por criar M partições usando um conjunto de dados fornecido como entrada. Este passo é crítico, uma vez que o resultado alcançado por qualquer função de consenso será condicionada à informação disponível no conjunto. Diferentes algoritmos de agrupamento de dados, parâmetros de inicialização ou visões dos dados são usados para criar um conjunto de partições;
- **Combinação de partições** por meio de uma função de consenso, alguns métodos requerem uma representação intermediária do *ensemble clustering* para a execução da função;
- **Avaliação da partição de consenso**, existem muitos métodos para realizar essa avaliação e métodos de avaliação especificamente projetados para aproveitar as informações fornecidas pelo conjunto.

Existem dois principais métodos de funções de consenso que são: baseados na co-ocorrências de padrões e na formulação da média de uma partição.

3.3.1 Métodos baseados na formulação da média da partição

A média da partição é definida como aquela que minimiza a soma das distâncias entre ela e todas as partições no conjunto. Pode ser formalmente declarado da seguinte maneira:

Dado M partições P1, . . . , PM e a distância simétrica d(.,.), encontrar o P* tal que:

$$P^* = argmin \sum_{i=1}^{M} d\left(P_i, P\right)$$

Algoritmos Genéticos (AG): Como o nome sugere, o AG ao longo da evolução de gerações consecutivas infere a partição consensual. A maioria dos métodos desse tipo usa a informação disponível no conjunto para criar potenciais populações de partição consensual inicial. Depois que cada população é criada, os cromossomos são evoluídos por uma função de adequação e os mais aptos são usados para alimentar a mutação e qualquer outros operadores genéticos (ABDALA, 2010).

Métodos baseados em Fatoração de Matrizes Não Negativas (FMN): métodos FMN referem-se a fatoração de uma matriz Z não negativa em outra duas matrizes. Z ≈ AB, onde A e B também são matrizes não negativas. Um exemplo desse método pode ser encontrado em (LI; DING; JORDAN, 2007).

Métodos de Kernel: (VEGA-PONS; CORREA-MORRIS; RUIZ-SHULCLOPER, 2010) introduziu um método Kernel conhecido como Algoritmo de Partições Consensuais Ponderadas via Kernel (PCPK). Neste método as partições de consenso são definidas da seguinte forma:

$$SoD(\mathbb{P}) = argmax \sum_{i=1}^{M} \omega_i \cdot \kappa(P, P_i)$$

Onde ω_i é um peso associado a partição P_i e κ é uma medida de similaridade entre as partições, que é uma função Kernel (SCHÖLKOPF et al., 2002). Mais detalhes desse método podem ser encontrados em (VEGA-PONS; JIANG; RUIZ-SHULCLOPER, 2011).

Métodos Baseados na Heurística: Entre os trabalhos mais relevantes propostos para resolver o problema de ADCP pela média da partição, (GODER; FILKOV, 2008) apresenta uma coleção de seis heurísticas. Sendo algumas delas:

1. **Best of K (BoK):** Uma das mais simples propostas é essencialmente um processo de seleção conhecido como *Best of K*, isto é, seleciona a participação considerada melhor ou mais representativa entre todos os conjuntos de partição. Isto se dá por selecionar iterativamente cada partição $\mathbb{P}$ e calcular a soma das distâncias (SoD)

entre a partição selecionada e as que restaram no conjunto. A partição com o menor valor para a SoD é selecionada como uma partição de consenso;

$$P^* = BoK(\mathbb{P}) = \min \sum_{i=1}^{M} d(P_i, \bar{P})$$

2. **The Best One Element Moves (BOEM):** BOEM começa com a escolha de uma partição de consenso entre o conjunto de partições. O algoritmo segue testando cada possível *label* iterativamente para cada partição restante do conjunto de partições, mantendo a *label* com o menor valor para a SoD.

Agrupamento com base em Programação Pré Definida (PPD): PPD é motivado pela observação dos valores de semelhança entre os pares que não fornecem informações suficientes para algoritmos de agrupamento. Com isso em mente, os autores propõem codificar as soluções obtidas pelos resultados de agrupamento individual por uma *string* multidimensional (SINGH et al., 2010). Mais detalhes desse método pode ser encontrado em (ABDALA, 2010).

3.3.2 Métodos baseados na Co-ocorrência de partições

Em (FRED; JAIN, 2005), os autores exploraram a ideia de acumulação de evidências, combinando as partições geradas pelas M tentativas de K-Means em uma matriz de co-associação. Esta matriz é usada mais tarde como uma nova medida de similaridade para um algoritmo de agrupamento hierárquico aglomerativo padrão. O método pode ser dividido em duas etapas. Alguns desses métodos são apresentados abaixo.

Rotulagem e Votação: Métodos baseados em nova rotulagem e votação dependem primeiro que o problema de correspondência de *label* seja resolvido. Uma vez que a correspondências dos *labels* para todas as partições no conjunto são resolvidas um processo de votação decide a partição consensual.

Acumulação de Evidencias, Matrizes de Co-associação: A suposição subjacente é baseada no fato de que os objetos pertencentes ao mesmo *cluster* natural serão provavelmente colocados no mesmo *cluster* entre diferentes partições. Uma matriz de co-associação é calculada com valores variando de 0 (sem associação) para 1 (máximo de associação). Depois de calcular a matriz de co-associação, os *clusters* gerais provavelmente podem ser encontrados em torno da diagonal principal. O trabalho da função de consenso com base neste método é resolver as pequenas divergências que ocorrem em outras regiões da matriz de co-associação. Para a etapa de combinação, este método usa ainda outro algoritmo de agrupamento, mais especificamente, um algoritmo de agrupamento hierárquico, a matriz de co-associação é considerada como um novo espaço de dados e usada como entrada para o algoritmo hierárquico. O resultado produzido é a partição de consenso.

Grafos e Hiper-Grafos: (STREHL; GHOSH, 2002) apresenta um dos primeiros trabalhos na área de agrupamento de conjuntos não supervisionados. Neste trabalho, três grafos baseados em heurísticas são propostas, nomeadamente AASP, HGPA e AAM. Essas heurísticas representam o conjunto de agrupamento como um hiper-grafo em que cada partição é codificada como uma hiper-aresta.

Algoritmos de Agrupamento Adaptados Localmente: Como o nome sugere, esta classe de função de consenso funciona sobre partições produzidas usando uma adaptação local de algoritmos de agrupamento propostos por (DOMENICONI et al., 2007).

4 Metodologia de Desenvolvimento e Pesquisa

Este capitulo irá explicar toda metodologia desenvolvida para se verificar a utilização de técnicas de ADCP, no contexto de ADO, para melhorar a confiabilidade do resultado. Para tal, um experimento foi projetado.

Como dito anteriormente, técnicas de ADCP são utilizadas quando pouco, ou nada é conhecido sobre o conjunto de dados, o que é uma grande característica de *stream* de dados onde nada pode ser concluído dado que não temos o dado completo.

O projeto foi desenvolvido em **Python 3**, uma linguagem de programação de alto nível e muito versátil por ser multiparadigma. Possui tipagem dinâmica e tem uma sintaxe concisa e clara. Além disso possui recursos poderosos em sua biblioteca e em módulos ou *frameworks* criados pela comunidade. Está entre uma das linguagens mais utilizadas para ciência de dados, inteligência artificial e aprendizado de máquina. Os módulos criados são apresentados na tabela 4 o código fonte está disponível no *github* (ANDRADE, 2021).

Módulos	Scripts
dados	daoarquivo.py
cluster	cluster.py clusterfeatures.py
algoritimosclustering	simplekmeans.py birch.py leader.py
medidassimilaridade	contagempares.py
funcoesconsenso	marjorityvoting.py bestofk.py bestoneelementsmove.py
principal	validador.py

Tabela 4 – Lista de módulos e seus respectivos scripts

Os algoritmos de agrupamento online serão utilizados para computarmos o *baseline* inicial, logo, o módulo de algoritmos de agrupamento não será discutido com detalhes nesse capítulo, assim como o de *cluster*, onde se tem alguns cálculos característicos para o conceito de ADO abordados na seção 3.1 desse trabalho.

As partições geradas a partir dos algoritmos de agrupamento online passam pela funções de consenso. As funções implementadas nesse trabalho foram a BoK e a BOEM, que utilizam o método de contagem de pares e terá o seu desenvolvimento para *data stream* explicado na seção 4.1 e 4.2 e também o *marjority voting* que será abordado na seção 4.3. Após isso, todos os resultados, tanto das funções de consenso quanto dos algoritmos de

agrupamento, irão gerar medidas como média, índice Rand e desvio padrão por meio de um avaliador que será explicado na seção 4.5.

4.1 Adequação do Método de Contagem de Pares para Stream de Dados

Assim como discutido na seção 3.2 deste trabalho, a contagem de pares é um dos métodos utilizados para se obter alguma das medidas de validação ou similaridade entre partições. O mais utilizado é o índice Rand. Esse método, no entanto, foi idealizado para trabalhar com o conjunto de dados completo e sua classificação inteira já conhecida, logo, para sua utilização em *stream* de dados, alguns ajustes se fazem necessários. Esta seção apresenta tais ajustes e um exemplo numérico para sua elucidação.

Algoritmo 1: Atualiza Valores

Entradas: label_part1 - classificação do dado para a partição considerada verdadeira
label_part2 - classificação do dado para a partição que será comparada.

```
tamanho_dataset ← +1
if numero_classe bigger then (label_part1 + 1) then
    diferença ← (label_part1 + 1) - numero_classe
    numero_classe ← numero_classe + diferença
end
if numero_cluster bigger then (label_part2 + 1) then
    diferença ← (label_part2 + 1) - numero_cluster
    numero_cluster ← numero_cluster + diferença
end
index_classe ← label_part1
index_cluster ← label_part2
```

Para se fazer possível o cálculo dos índices que utilizam esse método, algumas variáveis tiveram que ser mantidas em memória para serem atualizadas conforme o dado vai sendo recebido e processado. São elas:

- Tamanho do dado;
- Vetor de classes (grupos criados pela partição considerada verdade);
- Vetor de *clusters* (grupos criados pela partição que será comparada);
- Matriz de confusão;
- Dimensão da matriz de confusão (Número de classes x Número de *clusters*).

Quando a entrada do dado gera as *labels* para o dado naquele momento, essas classificações são utilizadas para atualizar as variáveis mencionadas acima. É importante

ressaltar que quando o algoritmo de agrupamento classifica o dado, a *label* designada para o mesmo não sofre alterações. É somado 1 ao tamanho do dado e verificado se as *labels* recebidas para Partição 1 existem no vetor de classes e se a da Partição 2 existe no vetor de *clusters*. Caso não existam, a *label* recebida pela Partição 1 é inserida como novo elemento ao vetor de classes e a *label* da Partição 2 é inserida como novo um elemento ao vetor de *clusters*. O número de quantidade de elementos dos mesmos (classes e *clusters*) é aumentado em mais um, conforme mostrado no algoritmo 1.

Com essas variáveis atualizadas, podemos assim criar ou atualizar a matriz de confusão (matriz de contingência). Criamos a matriz de confusão caso ela não exista. Se a dimensão da matriz existente for diferente do número de classes x número de *cluster*, teremos que redimensionar a mesma. Após isso, nós somamos 1 ao valor da matriz na posição do índex da classe pelo índex do *cluster*, assim como mostrado no algoritmo 2. Com base nessa matriz, iremos calcular os 4 tipos de números (N's) mencionados na seção 3.2 que chamamos aqui de matriz de confusão de pares.

Algoritmo 2: Atualiza Matriz de Confusão

```
if matriz_confusao is None then
    matriz_confusao[numero_classe][numero_cluster]
end
if dimensao(matriz_confusao) not equal (numero_classe, numero_cluster) then
    redimensiona(matriz_confusao, (numero_classe,numero_cluster))
end
matriz_confusao[index_classe][index_cluster] ← +1
```

Conforme mostra o algoritmo 3, a matriz de confusão de pares é calculada a partir da matriz de confusão atualizada. Pela soma das linhas, soma das colunas e a soma dos valores da tabela de confusão ao quadrado, conseguimos os valores para os N's.

T_i	T_1	T_2	T_3	T_4	...
DS	DS1	DS2	DS3	DS4	...
P1	1	1	2	3	...
P2	1	2	2	3	...
MC	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\\1\end{bmatrix}$	$\begin{bmatrix}1&1\\0&1\end{bmatrix}$	$\begin{bmatrix}1&1&0\\1&0&0\\0&0&1\end{bmatrix}$	...
N's	$\begin{bmatrix}0&0\\0&0\end{bmatrix}$	$\begin{bmatrix}0&2\\0&0\end{bmatrix}$	$\begin{bmatrix}2&2\\2&0\end{bmatrix}$	$\begin{bmatrix}8&2\\2&0\end{bmatrix}$	...
RI	1.0	0.0	0.33	0.66	...

Tabela 5 – Exemplo de uma comparação entre duas partições online

Algoritmo 3: Calcula Matriz de Confusão de Pares

```
soma_linhas[]
soma_colunas[]
matriz_confusao_transposta ← transposta(matriz_confusao)
soma_quadrados ← 0
for i ← 0 até numero_classe do
    soma ← 0
    for j ← 0 até numero_cluster do
        soma ← +matriz_confusao[i][j]
    end
    append(soma_linhas, soma)
end
for j ← 0 até numero_cluster do
    soma ← 0
    for i ← 0 até numero_classes do
        soma ← +matriz_confusao[i][j]
    end
    append(soma_colunas, soma)
end
for i ← 0 até numero_classe do
    for j ← 0 até numero_cluster do
        soma_quadrados ← +matriz_confusao[i][j]^2
    end
end
N[1][1] ← soma_quadrados - tamanho_dataset
N[0][1] ← soma(produto(matriz_confusao, soma_colunas)) - soma_quadrados
N[1][0] ← soma(produto(transposta_matriz_confusao, soma_linhas)) -
 soma_quadrados
N[0][0] ← tamanho_dataset^2 - N[0][1] - N[1][0] - soma_quadrados
```

A partir dessa matriz de confusão de pares onde cada posição representa os tipos dos N's, o índice Rand e o índice Rand ajustado são calculados.

Suponhamos que um conjunto de *data stream* (DS) seja processado por dois algoritmos gerando as partições P1 e P2 ao longo de um período de tempo T_i. A tabela 5 mostra a evolução da comparação entre essas duas partições geradas ao longo do tempo, a evolução das matrizes de confusão e dos *N's*, como a variação do índice Rand.

4.2 Soma de Distâncias

Nesse trabalho, vamos utilizar dois métodos baseados na formulação da média da partição que é definida como a minimização das somas das distâncias o *Best of K* (BoK) e o *The Best One Element Move* (BOEM). As implementações desses algoritmos podem ser encontradas nos scripts *bestofk.py* e *bestoneelementsmove.py*.

Como falado anteriormente na seção 3.3, o BoK é uma das mais simples maneiras de seleção conhecida para escolher a melhor partição representativa entre todas as partições. O BOEM começa com a escolha de uma partição, que nesse trabalho será feito

pelo BoK, e segue testando cada *label* possível para cada partição restante do conjunto de partição. Para esse experimento, algumas mudanças tiveram que ser aplicadas para a realidade de *stream* de dados.

Diferente do processo normal, no qual esses métodos de combinação estão acostumados a decidir a partição mais representativa, em dados online não possuímos todo o dados para processamento e comparação do todo, portanto o processamento do dado é feito a cada tempo T_i que o dado é disponibilizado. Para o cálculo da distância entre partições, utilizamos o método de contagem de pares online com o valor do *Rand Index* (RI) sendo considerada a distância (medida de similaridade). A cada chegada do dado, uma partição é selecionada como representativa daquele momento pelo BoK, ou seja, em um próximo tempo pode ser que ele decida por uma das outra partições pertencentes ao conjunto de partição. Consequentemente, esse resultado é refletido no que não recebe toda partição para o processamento BOEM. Em vez disso, recebe a *label* que foi selecionada naquele dado momento e processa em cima da mesma, iterando entre as outras possíveis resposta e comparando o restante das partições que não foram selecionadas.

A partir do momento que o método de contagem de pares é atualizado, também é atualizada uma matriz de distâncias que tem o tamanho da quantidade de classificadores para o dado, onde cada coluna e linha representa uma partição. Levando em consideração que *d(P1, P2) = d(P2, P1)* essa matriz é uma matriz triangular. Dessa forma, ficou mais fácil popular todas as distâncias e calcular a soma das mesmas para decidir qual das partições que será a melhor representativa. Na tabela 6 mostra um exemplo para classificação usando o BoK.

T_i	T_1	T_2	T_3	T_4	T_5	...
DS	DS1	DS2	DS3	DS4	DS5	...
P1	1	1	2	3	3	...
P2	1	2	2	3	3	...
P3	1	2	3	2	3	...
MD	$\begin{bmatrix} 0,00 & 1,00 & 1,00 \\ 1,00 & 0,00 & 1,00 \\ 1,00 & 1,00 & 0,00 \end{bmatrix}$	$\begin{bmatrix} 0,00 & 0,00 & 0,00 \\ 0,00 & 0,00 & 1,00 \\ 0,00 & 1,00 & 0,00 \end{bmatrix}$	$\begin{bmatrix} 0,0 & 0,33 & 0,66 \\ 0,33 & 0,0 & 0,66 \\ 0,66 & 0,66 & 0,0 \end{bmatrix}$	$\begin{bmatrix} 0,0 & 0,66 & 0,66 \\ 0,66 & 0,0 & 0,66 \\ 0,66 & 0,66 & 0,0 \end{bmatrix}$	$\begin{bmatrix} 0,0 & 0,80 & 0,60 \\ 0,80 & 0,0 & 0,60 \\ 0,60 & 0,60 & 0,0 \end{bmatrix}$	...
SoD	$\begin{bmatrix} 2,00 & 2,00 & 2,00 \end{bmatrix}$	$\begin{bmatrix} 0,00 & 1,00 & 1,00 \end{bmatrix}$	$\begin{bmatrix} 1,00 & 1,00 & 1,33 \end{bmatrix}$	$\begin{bmatrix} 1,33 & 1,33 & 1,33 \end{bmatrix}$	$\begin{bmatrix} 1,40 & 1,40 & 1,20 \end{bmatrix}$	...
BoK	P1	P1	P1	P1	P3	...

Tabela 6 – Exemplo dos resultados do BoK online

4.3 Marjority Voting

Como comentado na seção 3.3, o *Marjority Voting* (MV) é um método de co-ocorrência de partições feito por meio de uma votação entre as partições geradas pelos algoritmos formais de agrupamento online, ou seja, cada dado entra para o processo de votos onde a maior ocorrência da mesma classificação é selecionada pelo algoritmo, como mostra a tabela 7.

T_i	T_1	T_2	T_3	T_4	...
DS	DS1	DS2	DS3	DS4	...
P1	1	1	2	3	...
P2	1	2	2	3	...
P3	1	2	3	2	...
MV	1	2	2	3	...

Tabela 7 – Exemplo dos resultado do (*Marjority Voting*) (MV) online

4.4 Serializador - Simulador de *Data streams*

Embora fosse possível utilizar *data streams* reais, tais como *posts* de redes sociais, dados provenientes de sensores, entre outros, seria complicado avaliar, pois não possuímos, geralmente, *ground truths* para eles. Uma forma controlada de fazer isso seria utilizar um *dataset offline* e simular sua apresentação aos algoritmos como um *stream* de dados. Para tal, foi necessário criar um artefato de *software* simples com o objetivo de apresentar aos algoritmos de ADO um objeto de dado por unidade de tempo.

No nosso trabalho, o *script daoarquivo.py* funciona como esse serializador, onde, dado um arquivo csv com o separador |, ele consome esse arquivo em formato de *dataframe*, transformando na estrutura aceitável para processamento dos algoritmos de agrupamento, mantendo o controle de quais linhas ele já leu e qual será a próxima. Isso é útil também caso fosse utilizar mini *batches* de dados, onde você pode indicar o tamanho da partição do dado para ser processado ou também pode selecionar os dados de modo randômico.

4.5 Avaliador - Qualidade em série temporal

A avaliação de resultados de *clustering* é um assunto complexo sem um método definitivo para resolver o problema, tal como apresentamos na fundamentação teórica. No

contexto de online *clustering*, o assunto é ainda mais obscuro, com pouquíssimas referências ao tema. Para avaliar quantitativamente o resultado, idealizou-se uma avaliação (da informação disponível até o momento) sob a forma de uma série temporal. O módulo de avaliação toma o resultado cumulativo dos algoritmos de online *clustering* e produz uma série temporal do índice de qualidade desejado, no nosso caso o índice Rand. O *script* também fornece a média e desvio padrão cumulativo até o T_i desejado criando arquivos csv e gráficos com os resultados gerados para cada *dataset* analisado. O código pode ser encontrado no arquivo *validador.py*.

5 Datasets e Experimentos

Este capítulo discrimina os conjuntos de dados utilizados e descreve o experimento para validação da possibilidade de utilizar o conceito de Agrupamento de Dados via Combinação de Partições (ADCP) para Agrupamento de Dados Online (ADO).

5.1 Datasets

O repositório de aprendizado de máquina UCI (DUA; GRAFF, 2017) é uma coleção de banco de dados amplamente utilizado pela comunidade para a experimentação empírica em algoritmos desse nicho. Seu uso assegura a avaliação cruzada entre os algoritmos similares, este fato por si só garante a sua validade. Atualmente, esta coleção de *datasets* é composta por cerca de 588 conjuntos de dados organizados por diferentes critérios, tais como tipos de análise (categórica, regressão, agrupamento e outros), tipos de atributos, tipo de dados, entre outros. Para todos os conjuntos de dados, um único *ground truth* é fornecido. Neste trabalho, sete *datasets* são utilizados, dado o fato que muitos desses conjuntos de dados disponíveis podem não ser adequados para o método aqui proposto. A tabela a seguir resume os bancos de dados selecionados, bem como o número de padrões, o número de atributos, o número de agrupamentos, e um marcador que indica se o conjunto de dados teve que ser editado, a fim de explicar atributos não-numéricos ou em falta.

Datset	N. padrões	N. Atributos	N. Classes	Editado
constraceptive	1473	9	3	não
german	1000	24	2	não
optic	1000	64	10	não
pageblocks	5473	10	5	não
satellite	6435	36	7	não
magic	19020	10	2	não
yeast	1484	8	10	não

Tabela 8 – Datasets selecionados do UCI

Os *streams* de dados normalmente lidam com milhões de pontos de dados, no entanto esses *datasets* utilizados são pequenos. O objetivo deste trabalho não é apenas avaliar utilizando conjunto de dados reais, mas também garantir a reprodutibilidade dos resultados.

5.2 Experimentos

Os experimentos desse trabalho foram divididos em duas partes que serão explicadas nessa seção. O objetivo do experimento é explicar o resultado obtido pelos algoritmos de ADO individuais e quando é utilizado técnicas de ADCP.

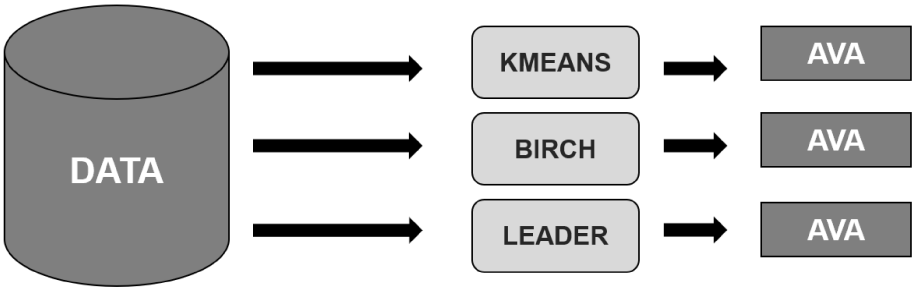

Figura 2 – Experimento parte 1.

Experimento parte 1: A partir do serializador, o *dataset* é simulado como uma *stream*. Cada dado no tempo T_i é classificado por três tipos de algoritmos ADO: *Simple-Pass k-Means*, *Leader* e o BIRCH. Esses algoritmos são definidos no capítulo 3 na seção 3.1. Após essa classificação, são calculadas métricas para esses resultados, logo, é calculado o índice Rand, a média e o desvio padrão para comparação dos resultados a cada tempo T_i. Para cada *dataset* escolhido para esse trabalho, temos como resultados as classificações feitas pelos algoritmos de ADO e os cálculos das métricas, aqui mencionado como série temporal, conforme demonstrado na Figura 2.

Experimento parte 2: A partir dos resultados da classificação obtida pelos algoritmos de ADO no tempo T_i, calculamos os consensos das partições até o momento. Para este, foram utilizados 3 algoritmos de ADCP, sendo eles: *Best of K* (BoK), *Best One Element Moves* (BOEM) e o *Majority Voting* (MV). Esses algoritmos são definidos na seção 3.3 e suas implementações no formato online são explicadas no capítulo 4. Após essa classificação, são geradas métricas para esses resultados, logo é calculado o índice Rand, a média e o desvio padrão para comparação dos resultados a cada tempo T_i. Então, para cada *dataset* escolhido para esse trabalho, com o resultados que obtivemos das classificações feitas pelos algoritmos de ADO, utilizamos o resultado para fazer as classificações dos algoritmos de ADCP. Finalmente, com os resultados dessas classificações também calculamos as métricas aqui mencionadas como série temporal, conforme demonstrado na Figura 3.

Destacamos que essas não são as funções de consenso mais sofisticadas, existem algoritmos mais elaborados, como métodos baseados em fatoração de matrizes não negativas ou métodos baseados em grafos. O objetivo deste trabalho é verificar se conseguimos aumentar a confiabilidade dos resultados obtidos em ADO utilizando ADCP.

O anexo B traz um exemplo de um resultado calculado do algoritmo BOEM para

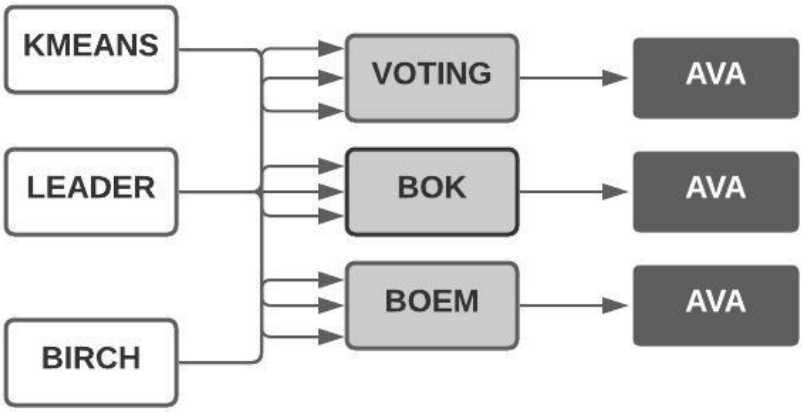

Figura 3 – Experimento parte 2.

o *dataset german*. Essa tabela contém os valores do índice Rand, média e desvio padrão para cada dado processado em um tempo T_i (quando este dado foi disponibilizado pelo serializador). Além disso, temos os gráficos mostrando esses resultados ao longo do tempo (série temporal), na Figura 5 para o Rand, Figura 11 para média e Figura 12 para o desvio padrão. A partir desses resultados, conseguimos comparar os algoritmos entre eles e analisar o comportamento toda vez que um dado novo é processado. Todos os gráficos para a comparação entre os algoritmos aqui estudados dos valores de Rand estão sendo comentados nos próximos parágrafos.

Para avaliar os resultados da aplicação da técnica de ADPC no contexto de ADO foi confeccionado um avaliador para verificar a qualidade das partições geradas em cada tempo T_i e ao final do processo. Para este fim, selecionou-se sete bases de dados publicamente disponíveis em (DUA; GRAFF, 2017), e largamente utilizadas pela comunidade de reconhecimento de padrões, para fins de comparação entre os resultados obtidos e o *ground truth* desses dados. A introdução a essa base de dados foi feita na seção 5.1 deste trabalho.

Os sete *datasets* foram processados pelos 3 algoritmos de ADO (*Simple-Pass k-Means*, *Leader* e BIRCH) e pelos algoritmos de ADCP (BoK, BOEM, MV) com os resultados gerados pelos algoritmos anteriores como é explicado acima nos experimentos. Para se avaliar esses resultados utilizou-se a métrica de similaridade RI (RAND, 1971) como medida de comparação. Valores próximos a um, significa que há concordâncias entre a partição analisada e o *ground truth*.

Devemos levar em consideração, na análise desses resultados, que o BoK algoritmo de ADPC, em tese, deveria selecionar a melhor partição para a classificação dos dados, porém no modelo online, onde o dado é classificado no tempo T_i (quando esse se encontra disponível), essa seleção da melhor partição pode mudar e isso é mostrado no exemplo

dado na Tabela 6 na seção 4.2. Além disso o BoK é utilizado, nesse caso, como selecionador da melhor partição para o algoritmo de agrupamento BOEM. Então essa troca de melhor partição que ocorre no BoK reflete nos resultados do BOEM. A medida de distância (similaridade) utilizada por ambos é o *Rand Index*.

Outra coisa que devemos levar em consideração quando olharmos para esses resultados é que não existe tratamento para quando duas ou mais soma das distâncias (SoD) são parecidas, logo, os algoritmos que utilizam esse método retornam a primeira partição que encontram com o menor valor.

A Figura 4 mostra os resultados das classificações para o *dataset contraceptive* e observamos que o *Leader* mostrou a melhor classificação. Conseguimos ver que o algoritmo MV consegue maior concordância que os outros algoritmos de agrupamento online aqui utilizados, exceto pelo *Leader*. O BoK acaba tendo uma classificação similar ao *Simple-Pass k-Means* e o BOEM é o pior tipo de agrupamento para esse caso.

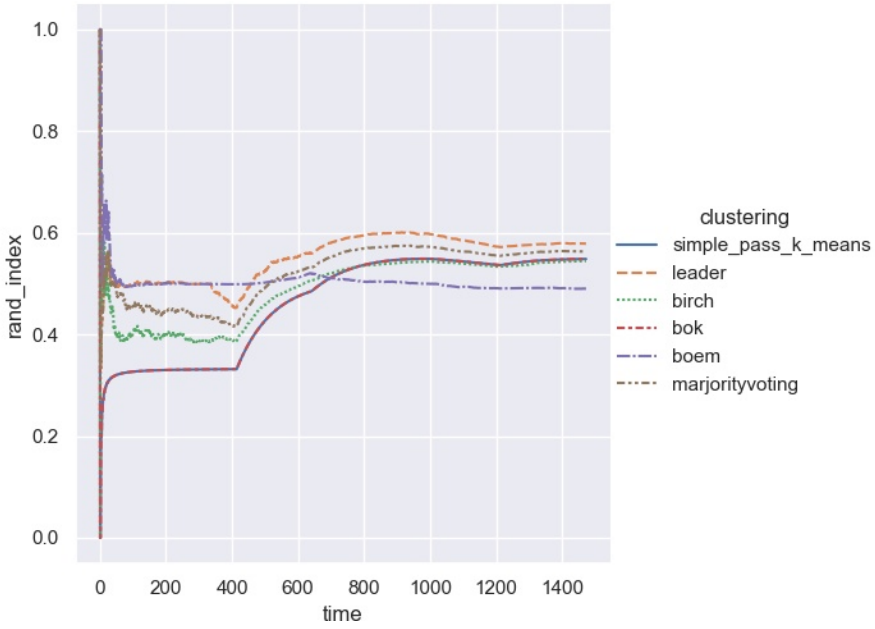

Figura 4 – Resultado comparativo para o *dataset constraceptive* utilizando métrica RI a cada tempo T_i.

A Figura 5 mostra os resultados das classificações para o *dataset german* e observamos que o BOEM e o MV, funções de consenso, conseguem mostrar maior concordância com o *ground truth*. Entre os algoritmos de agrupamento online temos somente o *Leader* com o melhor resultado.

A Figura 6 exibe os resultados das classificações para o *dataset magic* e observamos que quase todos os agrupamentos apresentam uma queda no tempo T_i=12500 nas concordâncias com o *ground truth*. O agrupamento feito pelo *Leader* continua sendo o melhor resultado, porém o MV possui um desempenho melhor que os outros dois algorit-

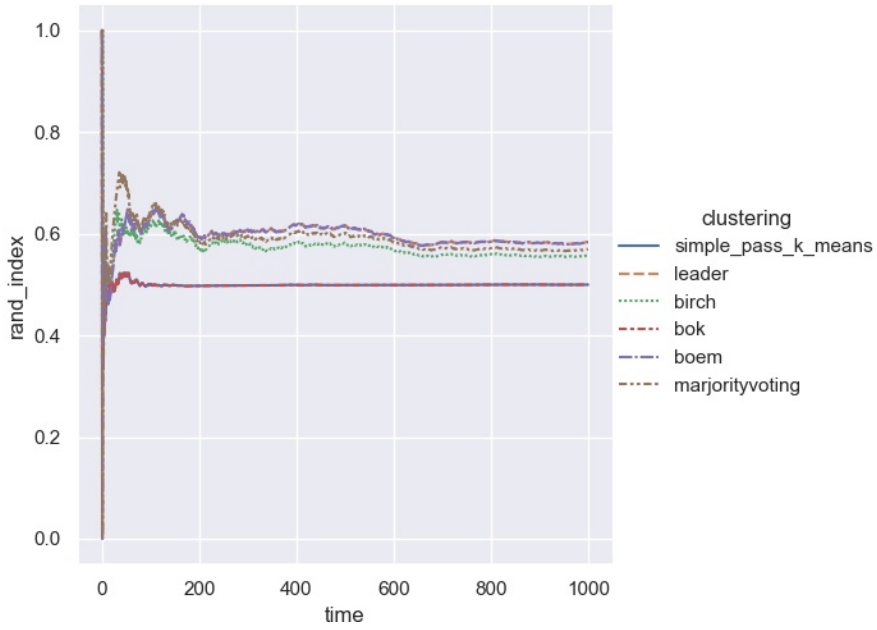

Figura 5 – Resultado comparativo para o *dataset german* utilizando métrica RI a cada tempo T_i.

mos de agrupamento online. As concordâncias para os agrupamentos feitos pelo BOEM e BIRCH, assim como as feitas pelo BoK e *Simple-Pass k-Means*, coincidem.

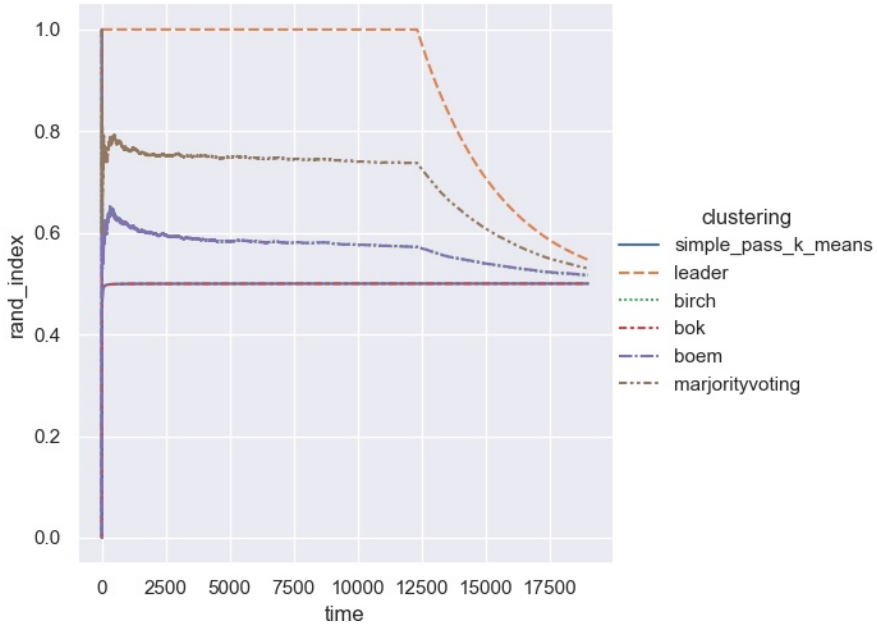

Figura 6 – Resultado comparativo para o *dataset magic* utilizando a métrica RI a cada tempo T_i.

A Figura 7 apresenta os resultados para o *dataset optic*, onde podemos observar que a maioria dos algoritmos conseguiram uma concordância maior que 0,8 ao final do tempo. O melhor resultado é o do *Leader*, porém o MV consegue alcançar resultados consistentes

para esse *dataset* em todo o tempo T_i e acima dos outros agrupamentos de *data stream* apresentados aqui. O BOEM apresentou a pior classificação, a partir do tempo T_i=200 ele só decresceu. O BoK apresentou um resultado ruim nos primeiros tempos, mas a partir do T_i=100 ele continuou crescendo até atingir uma concordância parecida ao BIRCH.

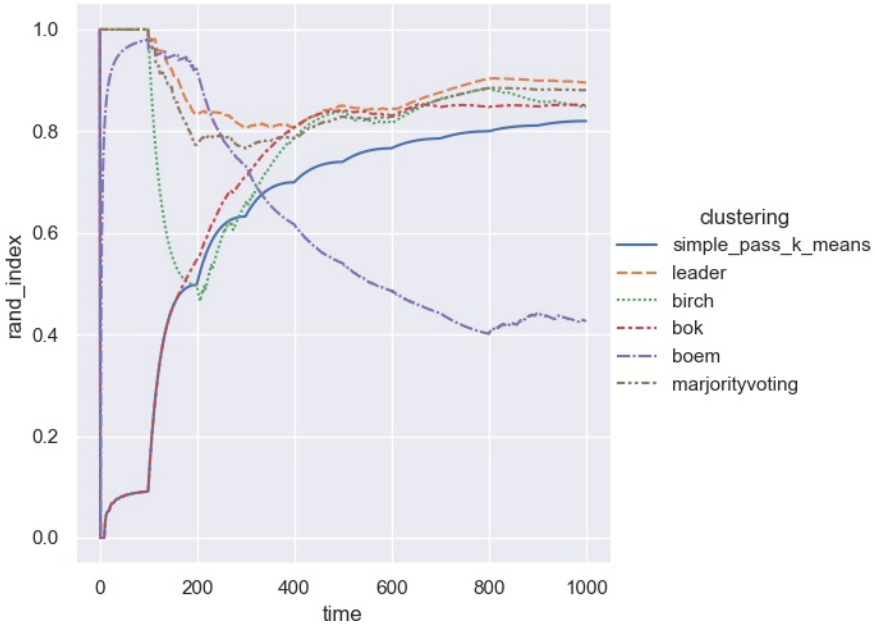

Figura 7 – Resultado comparativo para o *dataset optic* utilizando métrica RI a cada tempo T_i.

A Figura 8 mostra os resultados para o *dataset pageblocks* e conseguimos ver que o BoK, assim como *Simple-Pass K-Means*, apresentam uma concordância abaixo de 0.4 comparado ao resultado verdadeiro, sendo os piores algoritmos para utilizar na classificação desse conjunto de dados. Por outro lado, os demais algoritmos conseguem obter resultados acima de 0.8 a partir do tempo T_i=2000. O fato interessante, nesse caso, é que as funções de consenso, BOEM e MV, ficaram iguais ao melhor agrupamento feito pelos métodos online.

A Figura 9 exibe os resultados para o *dataset satellite*, onde observamos que o *Leader* mostra a maior concordância com o *ground truth* desse conjunto até o tempo T_i=4000, acompanhado do MV que mostra um comportamento parecido. Vemos também que o *Simple-pass K-Means* e o BoK, apesar de começarem com valores baixos para o RI, no tempo T_i=3000 mostram uma estabilidade quanto a essas classificações.

A Figura 10 apresenta os resultados para o *dataset yeast* e mostra que o BOEM na maioria do tempo mostrou concordâncias baixas em relação aos outros agrupamentos. O que mais se destaca nesse gráfico é que o BoK e o *Simple-Pass K-Means*, além de demonstrarem uma frequência para o resultado, também são os agrupamentos com a maior concordância para o *ground truth*.

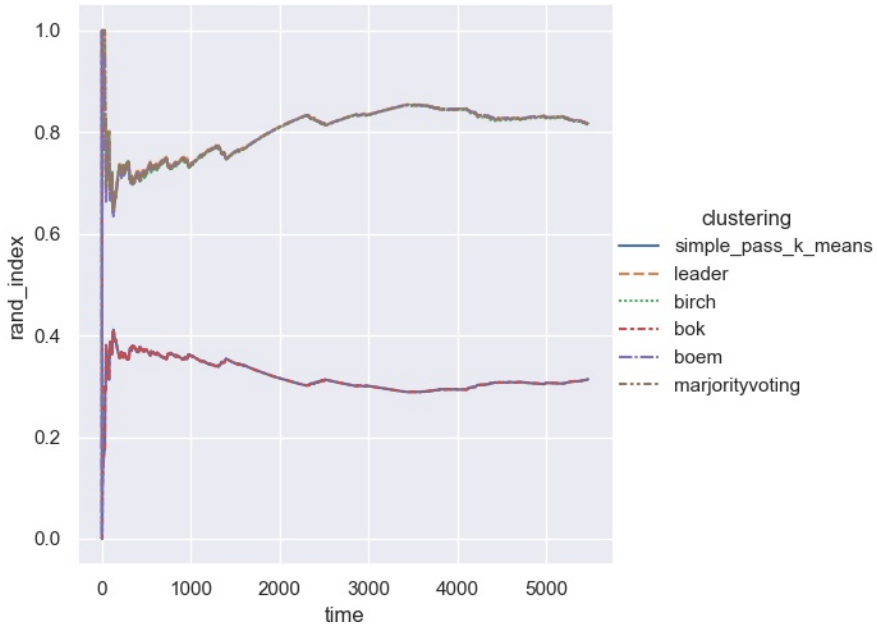

Figura 8 – Resultado comparativo para o *dataset pageblocks* utilizando métrica RI a cada tempo T_i.

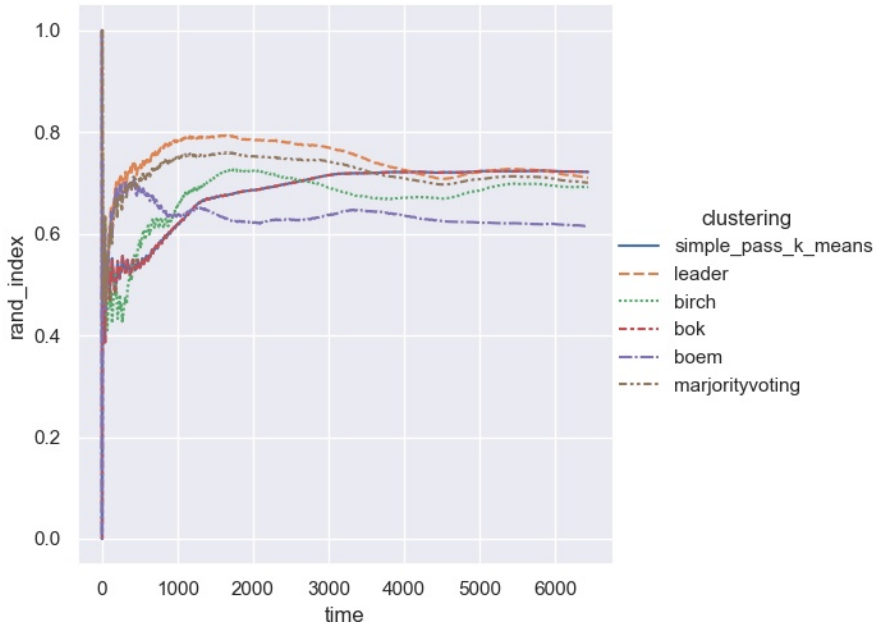

Figura 9 – Resultado comparativo para o *dataset satellite* utilizando métrica RI a cada tempo T_i.

A Tabela 9 traz os valores para RI, média e desvio padrão para todos os *datasets* aqui analisados e para cada algoritmo de agrupamento utilizado. Estes valores são os resultados finais do processamento de todo o dado (a média e o desvio padrão são calculados em relação ao valor de RI). Destacamos em azul os melhores valores para o RI final dos métodos de ADO; e em verde para os método ADCP para cada conjunto de dados. Os resultados nessa tabela só reforçam o que foi visto nos gráficos, pelo menos um dos

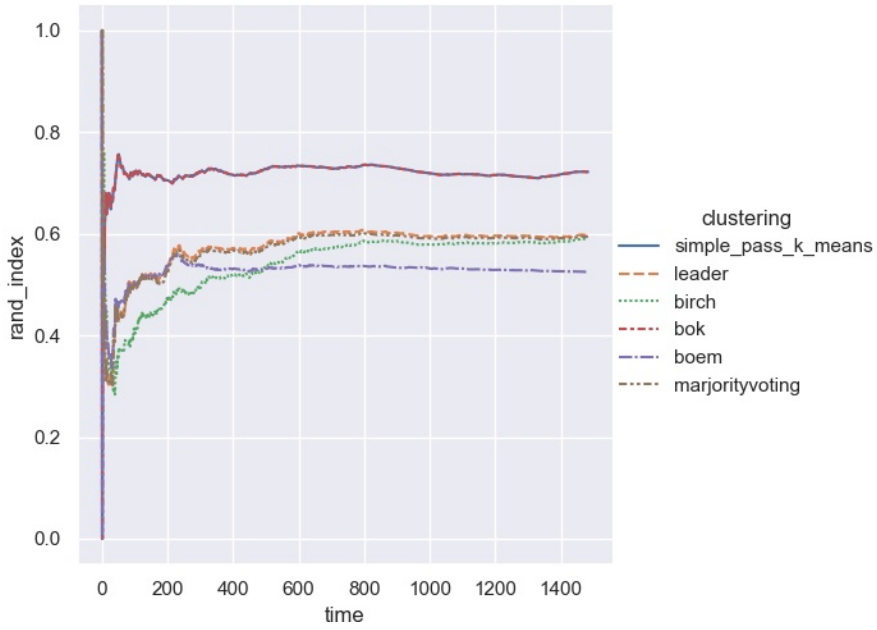

Figura 10 – Resultado comparativo para o *dataset yeast* utilizando métrica RI a cada tempo T_i.

agrupamentos feito pelas técnicas de ADCP mostrou resultados iguais ou melhores que alguns agrupamentos feitos pelas técnicas de ADO. Um dos destaques é para o *dataset pageblocks*, onde duas das funções de consenso desempenharam resultados melhores que os algoritmos *Simple-Pass K-Means* e o BIRCH ao final do tempo.

Com base nos experimentos, vemos que o BoK, um método para selecionar a melhor partição, na maioria dos casos, não desempenhou bem o seu papel. O BOEM que, apesar de ser um algoritmo custoso, costuma dar ótimos resultados quando a analise é feita em *batch*, mostrou-se bem fraco pro contexto online. Isso pode estar atrelado ao fato da mudança da escolha da partição feita pelo BoK, onde o BOEM deixa de selecionar a melhor *label*. O MV, uma função de consenso simples entre as partições, apesar de não ter reproduzido o melhor resultado absoluto na avaliação (uma das funções do agrupamento feito por combinação de partição é produzir um resultado confiável) conseguiu se manter com resultados acima de pelo menos dois agrupamentos de dados online na maioria dos casos.

Existem outros resultados, como gráficos da média e desvio padrão a cada tempo T_i para cada *dataset* produzidos por esse experimento, assim como tabelas com esses valores. Um exemplo desses resultados estão nos anexos A e B como amostra e os resultados completos para este trabalho podem ser encontrados na pasta de resultados no *github* (ANDRADE, 2021).

Métodos	Datasets		Constraceptive	German	Magic	Optic	Pageblocks	Satellite	Yeast
ADO	K-Means	*RI*	0,55	0,50	0,50	0,82	0,31	**0,72**	**0,72**
		μ	0,46	0,50	0,50	0,64	0,32	**0,68**	**0,72**
		σ	0,10	0,02	0,01	0,23	0,03	**0,07**	**0,03**
	Leader	*RI*	**0,58**	**0,58**	**0,55**	**0,90**	**0,82**	0,71	0,60
		μ	**0,55**	**0,60**	**0,89**	**0,88**	**0,81**	0,74	0,57
		σ	**0,05**	**0,03**	**0,16**	**0,06**	**0,05**	0,04	0,05
	BIRCH	*RI*	0,54	0,56	0,52	0,85	0,81	0,69	0,59
		μ	0,49	0,57	0,57	0,80	0,80	0,67	0,54
		σ	0,06	0,03	0,03	0,13	0,05	0,06	0,07
ADCP	MV	*RI*	**0,56**	0,57	**0,53**	**0,88**	**0,82**	0,70	0,59
		μ	**0,52**	0,59	**0,70**	**0,86**	**0,80**	0,72	0,57
		σ	**0,06**	0,04	**0,08**	**0,06**	**0,05**	0,04	0,05
	BoK	*RI*	0,55	0,50	0,50	0,85	0,31	**0,72**	**0,72**
		μ	0,46	0,50	0,50	0,70	0,32	**0,68**	**0,72**
		σ	0,10	0,02	0,01	0,25	0,03	**0,07**	**0,03**
	BOEM	*RI*	0,49	**0,58**	0,52	0,43	**0,82**	0,61	0,52
		μ	0,50	**0,60**	0,57	0,61	**0,80**	0,63	0,53
		σ	0,03	**0,03**	0,03	0,20	**0,05**	0,02	0,03

Tabela 9 – Valores finais para índice Rand - *RI*, média - μ e desvio padrão - σ para cada método e *dataset*

Conclusão

Este trabalho investigou a possibilidade de aplicar métodos de Agrupamento de Dados via Combinação de Partição (ADCP) no contexto de Agrupamentos de Dados Online (ADO) com o objetivo de aumentar a confiabilidade dos resultados. Três algoritmos de ADCP foram adaptados para o processamento do dado incremental, utilizados, avaliados e tiveram seus resultados comparados com os resultados obtidos pelos algoritmos de ADO. Com base nos experimentos, notamos que esses agrupamentos de consenso não produzem os melhores resultados, mas produzem resultados que estão entre os melhores para cada *dataset*. Além disso, pelo menos um obteve um resultado acima dos resultados de outros dois algoritmos de agrupamento online utilizados. Dos três algoritmos de ADCP estudados, o que apresentou melhores resultados na maioria dos experimentos aqui executados foi o *Marjority Voting*. Apesar de ser uma técnica simples, ela se mostrou consistente na maioria dos casos e obteve resultados muito próximos ao melhor agrupamento online feito. Com base nisso, concluímos que utilizar funções de consenso no contexto de agrupamento de dados online pode ser uma estratégia melhor para aumentar a confiabilidade desses resultados do que utilizar um único algoritmo de agrupamento online.

Até onde sabemos, este foi o único trabalho até o momento que considerou a possibilidade de se aplicar técnicas de combinação de partições no contexto de agrupamento de dados online.

Como trabalho futuro, pode ser aplicado o mesmo experimento desenvolvido aqui usando métricas internas de qualidade e *datasets* que são comumente utilizados no contexto de agrupamentos online. Além disso, pretendemos adaptar outros métodos de consenso mais elaborados para o contexto de fluxo de dados.

Referências

ABDALA, D. D. *Ensemble and Constrained Clustering with Applications.* Tese (Doutorado) — Universität Münster, 2010. Citado 7 vezes nas páginas 7, 17, 23, 27, 28, 29 e 30.

AGGARWAL, C. C.; REDDY, C. K. Data clustering. *Algorithms and applications. Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra*, Citeseer, 2014. Citado na página 16.

ALONSO, J. B. K-means vs mini batch k-means: A comparison. 2013. Citado na página 13.

ANDRADE, J. G. *Agrupamento de Dados Online via Combinação de Partição.* 2021. Disponível em: <https://github.com/SyaneAndrade/agrupamento_dados_combinacao_particao_online>. Citado 2 vezes nas páginas 32 e 46.

BASU, S.; BANERJEE, A.; MOONEY, R. J. Active semi-supervision for pairwise constrained clustering. In: SIAM. *Proceedings of the 2004 SIAM international conference on data mining*. [S.l.], 2004. p. 333–344. Citado na página 26.

BEN-HUR, A.; ELISSEEFF, A.; GUYON, I. A stability based method for discovering structure in clustered data. In: *Biocomputing 2002*. [S.l.]: World Scientific, 2001. p. 6–17. Citado na página 25.

CHOROMANSKA, A.; MONTELEONI, C. Online clustering with experts. In: *Artificial Intelligence and Statistics*. [S.l.: s.n.], 2012. p. 227–235. Citado na página 19.

DOMENICONI, C. et al. Locally adaptive metrics for clustering high dimensional data. *Data Mining and Knowledge Discovery*, Springer, v. 14, n. 1, p. 63–97, 2007. Citado na página 31.

DONGEN, S. V. Performance criteria for graph clustering and markov cluster experiments. In: CITESEER. *NATIONAL RESEARCH INSTITUTE FOR MATHEMATICS AND COMPUTER SCIENCE IN THE*. [S.l.], 2000. Citado na página 26.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. 2017. Disponível em: <http://archive.ics.uci.edu/ml>. Citado 2 vezes nas páginas 39 e 41.

FARNSTROM, F.; LEWIS, J.; ELKAN, C. Scalability for clustering algorithms revisited. *SIGKDD explorations*, Citeseer, v. 2, n. 1, p. 51–57, 2000. Citado na página 21.

FOWLKES, E. B.; MALLOWS, C. L. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, Taylor & Francis Group, v. 78, n. 383, p. 553–569, 1983. Citado na página 26.

FRANK, A. On kuhn's hungarian method—a tribute from hungary. *Naval Research Logistics (NRL)*, Wiley Online Library, v. 52, n. 1, p. 2–5, 2005. Citado na página 26.

FRED, A. L.; JAIN, A. K. Combining multiple clusterings using evidence accumulation. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 27, n. 6, p. 835–850, 2005. Citado 2 vezes nas páginas 27 e 30.

GAMA, J. *Knowledge discovery from data streams*. [S.l.]: Chapman and Hall/CRC, 2010. Citado 4 vezes nas páginas 17, 20, 21 e 22.

GHAEMI, R. et al. A survey: clustering ensembles techniques. *World Academy of Science, Engineering and Technology*, v. 50, p. 636–645, 2009. Citado na página 17.

GODER, A.; FILKOV, V. Consensus clustering algorithms: Comparison and refinement. In: SIAM. *2008 Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments (ALENEX)*. [S.l.], 2008. p. 109–117. Citado na página 29.

HUBERT, L.; ARABIE, P. Comparing partitions. *Journal of classification*, Springer, v. 2, n. 1, p. 193–218, 1985. Citado 4 vezes nas páginas 8, 23, 24 e 25.

JAIN, A. K.; DUBES, R. C. et al. *Algorithms for clustering data*. [S.l.]: Prentice hall Englewood Cliffs, 1988. v. 6. Citado 2 vezes nas páginas 12 e 16.

JIANG, X. et al. Distance measures for image segmentation evaluation. *EURASIP Journal on Advances in Signal Processing*, Springer, v. 2006, p. 1–10, 2006. Citado na página 26.

LI, T.; DING, C.; JORDAN, M. I. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In: IEEE. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. [S.l.], 2007. p. 577–582. Citado na página 29.

MEILĂ, M. Comparing clusterings by the variation of information. In: *Learning theory and kernel machines*. [S.l.]: Springer, 2003. p. 173–187. Citado na página 27.

MIRKIN, B. Mathematical classification and clustering: From how to what and why. In: *Classification, data analysis, and data highways*. [S.l.]: Springer, 1998. p. 172–181. Citado na página 25.

O'CALLAGHAN, L. et al. Streaming-data algorithms for high-quality clustering. In: IEEE. *Proceedings 18th International Conference on Data Engineering*. [S.l.], 2002. p. 685–694. Citado na página 21.

RAND, W. M. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, Taylor & Francis Group, v. 66, n. 336, p. 846–850, 1971. Citado 2 vezes nas páginas 25 e 41.

SCHÖLKOPF, B. et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. [S.l.]: MIT press, 2002. Citado na página 29.

SILVA, J. A. et al. Data stream clustering: A survey. *ACM Computing Surveys (CSUR)*, ACM, v. 46, n. 1, p. 13, 2013. Citado 4 vezes nas páginas 13, 17, 20 e 22.

SINGH, V. et al. Ensemble clustering using semidefinite programming with applications. *Machine learning*, Springer, v. 79, n. 1, p. 177–200, 2010. Citado na página 30.

STREHL, A.; GHOSH, J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, v. 3, n. Dec, p. 583–617, 2002. Citado na página 31.

STREHL, A.; GHOSH, J.; MOONEY, R. Impact of similarity measures on web-page clustering. In: *Workshop on artificial intelligence for web search (AAAI 2000)*. [S.l.: s.n.], 2000. v. 58, p. 64. Citado na página 27.

TOPCHY, A. et al. Adaptive clustering ensembles. In: IEEE. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.* [S.l.], 2004. v. 1, p. 272–275. Citado na página 27.

VEGA-PONS, S.; CORREA-MORRIS, J.; RUIZ-SHULCLOPER, J. Weighted partition consensus via kernels. *Pattern Recognition*, Elsevier, v. 43, n. 8, p. 2712–2724, 2010. Citado na página 29.

VEGA-PONS, S.; JIANG, X.; RUIZ-SHULCLOPER, J. Segmentation ensemble via kernels. In: IEEE. *The First Asian Conference on Pattern Recognition*. [S.l.], 2011. p. 686–690. Citado na página 29.

VEGA-PONS, S.; RUIZ-SHULCLOPER, J. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, World Scientific, v. 25, n. 03, p. 337–372, 2011. Citado 3 vezes nas páginas 12, 13 e 17.

WARRENS, M. J.; HOEF, H. van der. Understanding partition comparison indices based on counting object pairs. *arXiv preprint arXiv:1901.01777*, 2019. Citado na página 22.

XU, D.; TIAN, Y. A comprehensive survey of clustering algorithms. *Annals of Data Science*, Springer, v. 2, n. 2, p. 165–193, 2015. Citado 2 vezes nas páginas 12 e 16.

ZHANG, D.-X.; HEWITT, G. M. Insect mitochondrial control region: a review of its structure, evolution and usefulness in evolutionary studies. *Biochemical Systematics and Ecology*, Elsevier, v. 25, n. 2, p. 99–120, 1997. Citado na página 22.

Apêndices

APÊNDICE A – Gráficos da Média e do Desvio Padrão para o dataset German

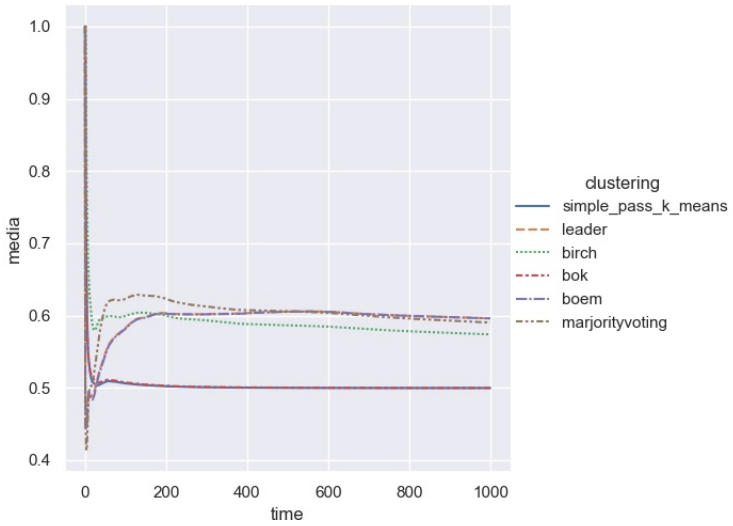

Figura 11 – Gráfico para o *dataset german* utilizando o calculo da média a cada tempo T.

Observando os gráficos aqui apresentados mostram que apesar de inicialmente ambas as medidas de desvio padrão e média mostrarem uma grande variação, eles estabilizam com o passar do tempo, convergindo para um único valor.

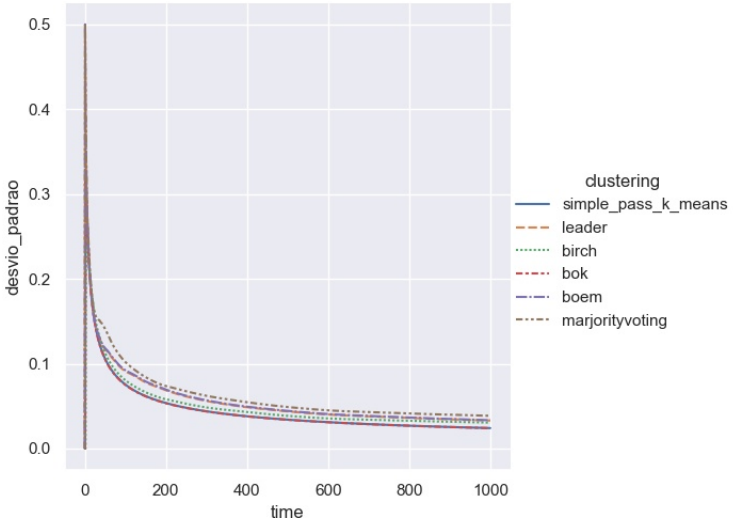

Figura 12 – Gráfico para o *dataset german* utilizando o calculo do desvio padrão a cada tempo T.

APÊNDICE B – Tabela de resultados do BOEM para o Dataset German

A tabela abaixo mostra os resultados para cada tempo do conjunto de dados *german*. Inicialmente percebemos que como não se tem muita informações sobre esse *dataset*, os valores ficam variando bastante, mas com passar do tempo eles tendem a se estabilizar.

Tabela 10 – Resultados para o *dataset german* com valores para Índice de Rand - *RI*, Média - μ e Desvio Padrão - σ para o método de Agrupamento de dados BOEM a cada tempo T_i .

T_i	RI	μ	σ
1	1,00	1,00	0,00
2	0,00	0,50	0,50
3	0,33	0,44	0,42
4	0,50	0,46	0,36
5	0,40	0,45	0,32
6	0,47	0,45	0,30
7	0,52	0,46	0,27
8	0,57	0,47	0,26
9	0,61	0,49	0,25
10	0,53	0,49	0,24
11	0,49	0,49	0,23
12	0,47	0,49	0,22
13	0,46	0,49	0,21
14	0,46	0,49	0,20
15	0,47	0,49	0,19
16	0,47	0,48	0,19
17	0,47	0,48	0,18
18	0,48	0,48	0,18
19	0,49	0,48	0,17
20	0,49	0,48	0,17
21	0,50	0,49	0,16
22	0,52	0,49	0,16

T_i	RI	μ	σ
334	0,61	0,60	0,05
335	0,61	0,60	0,05
336	0,60	0,60	0,05
337	0,60	0,60	0,05
338	0,60	0,60	0,05
339	0,60	0,60	0,05
340	0,60	0,60	0,05
341	0,60	0,60	0,05
342	0,61	0,60	0,05
343	0,61	0,60	0,05
344	0,61	0,60	0,05
345	0,61	0,60	0,05
346	0,61	0,60	0,05
347	0,61	0,60	0,05
348	0,61	0,60	0,05
349	0,61	0,60	0,05
350	0,61	0,60	0,05
351	0,61	0,60	0,05
352	0,61	0,60	0,05
353	0,61	0,60	0,05
354	0,61	0,60	0,05
355	0,61	0,60	0,05

T_i	RI	μ	σ
667	0,58	0,60	0,04
668	0,58	0,60	0,04
669	0,58	0,60	0,04
670	0,58	0,60	0,04
671	0,58	0,60	0,04
672	0,58	0,60	0,04
673	0,58	0,60	0,04
674	0,58	0,60	0,04
675	0,58	0,60	0,04
676	0,58	0,60	0,04
677	0,58	0,60	0,04
678	0,58	0,60	0,04
679	0,58	0,60	0,04
680	0,58	0,60	0,04
681	0,58	0,60	0,04
682	0,58	0,60	0,04
683	0,58	0,60	0,04
684	0,58	0,60	0,04
685	0,58	0,60	0,04
686	0,58	0,60	0,04
687	0,58	0,60	0,04
688	0,58	0,60	0,04

23	0,53	0,49	0,16	356	0,61	0,60	0,05	689	0,58	0,60	0,04
24	0,54	0,49	0,15	357	0,61	0,60	0,05	690	0,58	0,60	0,04
25	0,55	0,49	0,15	358	0,60	0,60	0,05	691	0,58	0,60	0,04
26	0,56	0,49	0,15	359	0,60	0,60	0,05	692	0,58	0,60	0,04
27	0,57	0,50	0,15	360	0,60	0,60	0,05	693	0,58	0,60	0,04
28	0,58	0,50	0,14	361	0,60	0,60	0,05	694	0,58	0,60	0,04
29	0,59	0,50	0,14	362	0,60	0,60	0,05	695	0,58	0,60	0,04
30	0,57	0,51	0,14	363	0,60	0,60	0,05	696	0,58	0,60	0,04
31	0,57	0,51	0,14	364	0,61	0,60	0,05	697	0,58	0,60	0,04
32	0,58	0,51	0,14	365	0,60	0,60	0,05	698	0,58	0,60	0,04
33	0,59	0,51	0,14	366	0,60	0,60	0,05	699	0,58	0,60	0,04
34	0,60	0,52	0,13	367	0,60	0,60	0,05	700	0,58	0,60	0,04
35	0,61	0,52	0,13	368	0,61	0,60	0,05	701	0,58	0,60	0,04
36	0,59	0,52	0,13	369	0,60	0,60	0,05	702	0,58	0,60	0,04
37	0,59	0,52	0,13	370	0,60	0,60	0,05	703	0,58	0,60	0,04
38	0,58	0,52	0,13	371	0,61	0,60	0,05	704	0,58	0,60	0,04
39	0,58	0,52	0,13	372	0,61	0,60	0,05	705	0,58	0,60	0,04
40	0,59	0,53	0,13	373	0,61	0,60	0,05	706	0,58	0,60	0,04
41	0,60	0,53	0,13	374	0,60	0,60	0,05	707	0,58	0,60	0,04
42	0,60	0,53	0,13	375	0,61	0,60	0,05	708	0,58	0,60	0,04
43	0,61	0,53	0,12	376	0,61	0,60	0,05	709	0,58	0,60	0,04
44	0,62	0,53	0,12	377	0,61	0,60	0,05	710	0,58	0,60	0,04
45	0,60	0,54	0,12	378	0,61	0,60	0,05	711	0,58	0,60	0,04
46	0,61	0,54	0,12	379	0,61	0,60	0,05	712	0,58	0,60	0,04
47	0,61	0,54	0,12	380	0,61	0,60	0,05	713	0,58	0,60	0,04
48	0,62	0,54	0,12	381	0,61	0,60	0,05	714	0,58	0,60	0,04
49	0,62	0,54	0,12	382	0,61	0,60	0,05	715	0,58	0,60	0,04
50	0,63	0,54	0,12	383	0,61	0,60	0,05	716	0,58	0,60	0,04
51	0,63	0,55	0,12	384	0,61	0,60	0,05	717	0,58	0,60	0,04
52	0,64	0,55	0,12	385	0,61	0,60	0,05	718	0,58	0,60	0,04
53	0,64	0,55	0,12	386	0,61	0,60	0,05	719	0,58	0,60	0,04
54	0,65	0,55	0,12	387	0,61	0,60	0,05	720	0,59	0,60	0,04
55	0,63	0,55	0,12	388	0,61	0,60	0,05	721	0,58	0,60	0,04
56	0,64	0,55	0,12	389	0,61	0,60	0,05	722	0,58	0,60	0,04
57	0,62	0,55	0,12	390	0,62	0,60	0,05	723	0,58	0,60	0,04
58	0,63	0,56	0,11	391	0,62	0,60	0,05	724	0,58	0,60	0,04
59	0,63	0,56	0,11	392	0,62	0,60	0,05	725	0,58	0,60	0,04
60	0,62	0,56	0,11	393	0,62	0,60	0,05	726	0,58	0,60	0,04
61	0,62	0,56	0,11	394	0,62	0,60	0,05	727	0,58	0,60	0,04

62	0,63	0,56	0,11
63	0,61	0,56	0,11
64	0,60	0,56	0,11
65	0,61	0,56	0,11
66	0,61	0,56	0,11
67	0,62	0,56	0,11
68	0,62	0,57	0,11
69	0,61	0,57	0,11
70	0,61	0,57	0,11
71	0,62	0,57	0,11
72	0,62	0,57	0,11
73	0,62	0,57	0,11
74	0,63	0,57	0,10
75	0,62	0,57	0,10
76	0,62	0,57	0,10
77	0,61	0,57	0,10
78	0,61	0,57	0,10
79	0,60	0,57	0,10
80	0,61	0,57	0,10
81	0,60	0,57	0,10
82	0,60	0,57	0,10
83	0,61	0,57	0,10
84	0,61	0,57	0,10
85	0,61	0,57	0,10
86	0,61	0,57	0,10
87	0,62	0,58	0,10
88	0,62	0,58	0,10
89	0,62	0,58	0,10
90	0,62	0,58	0,10
91	0,62	0,58	0,10
92	0,62	0,58	0,10
93	0,61	0,58	0,10
94	0,62	0,58	0,09
95	0,62	0,58	0,09
96	0,62	0,58	0,09
97	0,62	0,58	0,09
98	0,63	0,58	0,09
99	0,63	0,58	0,09

395	0,62	0,60	0,05
396	0,62	0,60	0,05
397	0,62	0,60	0,05
398	0,62	0,60	0,05
399	0,62	0,60	0,05
400	0,62	0,60	0,05
401	0,62	0,60	0,05
402	0,62	0,60	0,05
403	0,62	0,60	0,05
404	0,62	0,60	0,05
405	0,62	0,60	0,05
406	0,62	0,60	0,05
407	0,62	0,60	0,05
408	0,62	0,60	0,05
409	0,62	0,60	0,05
410	0,62	0,60	0,05
411	0,62	0,60	0,05
412	0,62	0,60	0,05
413	0,62	0,60	0,05
414	0,62	0,60	0,05
415	0,62	0,60	0,05
416	0,62	0,60	0,05
417	0,62	0,60	0,05
418	0,61	0,60	0,05
419	0,61	0,60	0,05
420	0,61	0,60	0,05
421	0,61	0,60	0,05
422	0,61	0,60	0,05
423	0,61	0,60	0,05
424	0,61	0,60	0,05
425	0,61	0,60	0,05
426	0,61	0,60	0,05
427	0,61	0,60	0,05
428	0,61	0,60	0,05
429	0,62	0,60	0,05
430	0,61	0,60	0,05
431	0,61	0,60	0,05
432	0,62	0,60	0,05

728	0,58	0,60	0,04
729	0,58	0,60	0,04
730	0,58	0,60	0,04
731	0,58	0,60	0,04
732	0,58	0,60	0,04
733	0,58	0,60	0,04
734	0,58	0,60	0,04
735	0,58	0,60	0,04
736	0,58	0,60	0,04
737	0,58	0,60	0,04
738	0,58	0,60	0,04
739	0,58	0,60	0,04
740	0,58	0,60	0,04
741	0,58	0,60	0,04
742	0,58	0,60	0,04
743	0,58	0,60	0,04
744	0,58	0,60	0,04
745	0,58	0,60	0,04
746	0,58	0,60	0,04
747	0,58	0,60	0,04
748	0,58	0,60	0,04
749	0,58	0,60	0,04
750	0,58	0,60	0,04
751	0,58	0,60	0,04
752	0,58	0,60	0,04
753	0,58	0,60	0,04
754	0,58	0,60	0,04
755	0,58	0,60	0,04
756	0,58	0,60	0,04
757	0,58	0,60	0,04
758	0,58	0,60	0,04
759	0,58	0,60	0,04
760	0,58	0,60	0,04
761	0,58	0,60	0,04
762	0,58	0,60	0,04
763	0,58	0,60	0,04
764	0,58	0,60	0,04
765	0,58	0,60	0,04

100	0,63	0,58	0,09	433	0,62	0,60	0,05	766	0,58	0,60	0,04
101	0,63	0,58	0,09	434	0,62	0,60	0,05	767	0,58	0,60	0,04
102	0,64	0,58	0,09	435	0,62	0,60	0,05	768	0,58	0,60	0,04
103	0,64	0,58	0,09	436	0,62	0,60	0,05	769	0,58	0,60	0,04
104	0,64	0,58	0,09	437	0,62	0,60	0,05	770	0,58	0,60	0,04
105	0,64	0,58	0,09	438	0,62	0,60	0,05	771	0,58	0,60	0,04
106	0,65	0,58	0,09	439	0,62	0,60	0,05	772	0,58	0,60	0,04
107	0,64	0,59	0,09	440	0,62	0,60	0,05	773	0,58	0,60	0,04
108	0,64	0,59	0,09	441	0,62	0,60	0,05	774	0,58	0,60	0,04
109	0,64	0,59	0,09	442	0,62	0,60	0,05	775	0,58	0,60	0,04
110	0,65	0,59	0,09	443	0,62	0,60	0,05	776	0,58	0,60	0,04
111	0,65	0,59	0,09	444	0,62	0,60	0,05	777	0,58	0,60	0,04
112	0,65	0,59	0,09	445	0,61	0,60	0,05	778	0,58	0,60	0,04
113	0,65	0,59	0,09	446	0,61	0,60	0,05	779	0,58	0,60	0,04
114	0,64	0,59	0,09	447	0,61	0,60	0,05	780	0,58	0,60	0,04
115	0,65	0,59	0,09	448	0,61	0,60	0,05	781	0,58	0,60	0,04
116	0,65	0,59	0,09	449	0,61	0,60	0,05	782	0,58	0,60	0,04
117	0,64	0,59	0,09	450	0,61	0,60	0,05	783	0,58	0,60	0,04
118	0,64	0,59	0,09	451	0,61	0,60	0,05	784	0,58	0,60	0,04
119	0,64	0,59	0,09	452	0,61	0,60	0,05	785	0,58	0,60	0,04
120	0,64	0,59	0,09	453	0,61	0,60	0,05	786	0,58	0,60	0,04
121	0,63	0,59	0,09	454	0,61	0,60	0,05	787	0,58	0,60	0,04
122	0,63	0,59	0,09	455	0,61	0,60	0,05	788	0,59	0,60	0,04
123	0,64	0,59	0,09	456	0,61	0,60	0,05	789	0,58	0,60	0,04
124	0,64	0,59	0,09	457	0,61	0,60	0,05	790	0,58	0,60	0,04
125	0,63	0,59	0,09	458	0,61	0,60	0,05	791	0,58	0,60	0,04
126	0,63	0,59	0,09	459	0,61	0,60	0,05	792	0,58	0,60	0,04
127	0,64	0,59	0,09	460	0,61	0,60	0,05	793	0,58	0,60	0,04
128	0,63	0,59	0,09	461	0,61	0,60	0,05	794	0,58	0,60	0,04
129	0,63	0,59	0,09	462	0,61	0,60	0,05	795	0,58	0,60	0,04
130	0,63	0,59	0,08	463	0,61	0,60	0,05	796	0,58	0,60	0,04
131	0,62	0,60	0,08	464	0,61	0,60	0,05	797	0,58	0,60	0,04
132	0,61	0,60	0,08	465	0,61	0,60	0,05	798	0,58	0,60	0,04
133	0,62	0,60	0,08	466	0,61	0,60	0,05	799	0,58	0,60	0,04
134	0,62	0,60	0,08	467	0,61	0,60	0,05	800	0,58	0,60	0,04
135	0,61	0,60	0,08	468	0,61	0,60	0,05	801	0,58	0,60	0,04
136	0,61	0,60	0,08	469	0,61	0,60	0,05	802	0,59	0,60	0,04
137	0,61	0,60	0,08	470	0,61	0,60	0,05	803	0,59	0,60	0,04
138	0,60	0,60	0,08	471	0,61	0,60	0,05	804	0,59	0,60	0,04

139	0,61	0,60	0,08
140	0,61	0,60	0,08
141	0,61	0,60	0,08
142	0,61	0,60	0,08
143	0,61	0,60	0,08
144	0,61	0,60	0,08
145	0,61	0,60	0,08
146	0,61	0,60	0,08
147	0,61	0,60	0,08
148	0,62	0,60	0,08
149	0,62	0,60	0,08
150	0,62	0,60	0,08
151	0,62	0,60	0,08
152	0,62	0,60	0,08
153	0,62	0,60	0,08
154	0,63	0,60	0,08
155	0,63	0,60	0,08
156	0,62	0,60	0,08
157	0,62	0,60	0,08
158	0,63	0,60	0,08
159	0,63	0,60	0,08
160	0,63	0,60	0,08
161	0,63	0,60	0,08
162	0,63	0,60	0,08
163	0,63	0,60	0,08
164	0,64	0,60	0,08
165	0,64	0,60	0,08
166	0,64	0,60	0,08
167	0,63	0,60	0,08
168	0,64	0,60	0,08
169	0,64	0,60	0,08
170	0,63	0,60	0,08
171	0,63	0,60	0,08
172	0,63	0,60	0,07
173	0,62	0,60	0,07
174	0,63	0,60	0,07
175	0,62	0,60	0,07
176	0,62	0,60	0,07

472	0,61	0,60	0,05
473	0,61	0,60	0,05
474	0,61	0,60	0,05
475	0,61	0,60	0,05
476	0,61	0,60	0,05
477	0,61	0,60	0,05
478	0,61	0,60	0,05
479	0,61	0,61	0,05
480	0,61	0,61	0,05
481	0,61	0,61	0,05
482	0,61	0,61	0,05
483	0,61	0,61	0,05
484	0,61	0,61	0,05
485	0,61	0,61	0,05
486	0,61	0,61	0,04
487	0,61	0,61	0,04
488	0,61	0,61	0,04
489	0,61	0,61	0,04
490	0,61	0,61	0,04
491	0,61	0,61	0,04
492	0,61	0,61	0,04
493	0,61	0,61	0,04
494	0,62	0,61	0,04
495	0,62	0,61	0,04
496	0,61	0,61	0,04
497	0,61	0,61	0,04
498	0,62	0,61	0,04
499	0,62	0,61	0,04
500	0,62	0,61	0,04
501	0,61	0,61	0,04
502	0,62	0,61	0,04
503	0,62	0,61	0,04
504	0,61	0,61	0,04
505	0,61	0,61	0,04
506	0,61	0,61	0,04
507	0,61	0,61	0,04
508	0,61	0,61	0,04
509	0,61	0,61	0,04

805	0,59	0,60	0,04
806	0,59	0,60	0,04
807	0,59	0,60	0,04
808	0,59	0,60	0,04
809	0,59	0,60	0,04
810	0,59	0,60	0,04
811	0,59	0,60	0,04
812	0,59	0,60	0,04
813	0,59	0,60	0,04
814	0,59	0,60	0,04
815	0,58	0,60	0,04
816	0,59	0,60	0,04
817	0,59	0,60	0,04
818	0,59	0,60	0,04
819	0,59	0,60	0,04
820	0,58	0,60	0,04
821	0,58	0,60	0,04
822	0,59	0,60	0,04
823	0,58	0,60	0,04
824	0,58	0,60	0,04
825	0,58	0,60	0,04
826	0,59	0,60	0,04
827	0,58	0,60	0,04
828	0,58	0,60	0,04
829	0,58	0,60	0,04
830	0,58	0,60	0,04
831	0,58	0,60	0,04
832	0,58	0,60	0,04
833	0,58	0,60	0,04
834	0,58	0,60	0,04
835	0,58	0,60	0,04
836	0,58	0,60	0,04
837	0,58	0,60	0,04
838	0,58	0,60	0,04
839	0,58	0,60	0,04
840	0,58	0,60	0,04
841	0,58	0,60	0,04
842	0,58	0,60	0,04

177	0,62	0,60	0,07	510	0,61	0,61	0,04	843	0,58	0,60	0,04
178	0,62	0,60	0,07	511	0,61	0,61	0,04	844	0,58	0,60	0,04
179	0,62	0,60	0,07	512	0,61	0,61	0,04	845	0,58	0,60	0,04
180	0,62	0,60	0,07	513	0,61	0,61	0,04	846	0,58	0,60	0,04
181	0,62	0,60	0,07	514	0,61	0,61	0,04	847	0,58	0,60	0,04
182	0,62	0,60	0,07	515	0,61	0,61	0,04	848	0,58	0,60	0,04
183	0,62	0,60	0,07	516	0,61	0,61	0,04	849	0,58	0,60	0,04
184	0,62	0,60	0,07	517	0,61	0,61	0,04	850	0,58	0,60	0,04
185	0,61	0,60	0,07	518	0,61	0,61	0,04	851	0,58	0,60	0,04
186	0,61	0,60	0,07	519	0,61	0,61	0,04	852	0,58	0,60	0,04
187	0,61	0,60	0,07	520	0,61	0,61	0,04	853	0,58	0,60	0,04
188	0,61	0,60	0,07	521	0,61	0,61	0,04	854	0,58	0,60	0,04
189	0,61	0,60	0,07	522	0,61	0,61	0,04	855	0,58	0,60	0,04
190	0,61	0,60	0,07	523	0,61	0,61	0,04	856	0,58	0,60	0,04
191	0,61	0,60	0,07	524	0,61	0,61	0,04	857	0,58	0,60	0,04
192	0,60	0,60	0,07	525	0,61	0,61	0,04	858	0,58	0,60	0,04
193	0,60	0,60	0,07	526	0,61	0,61	0,04	859	0,58	0,60	0,04
194	0,60	0,60	0,07	527	0,61	0,61	0,04	860	0,58	0,60	0,04
195	0,60	0,60	0,07	528	0,61	0,61	0,04	861	0,58	0,60	0,04
196	0,59	0,60	0,07	529	0,61	0,61	0,04	862	0,58	0,60	0,04
197	0,60	0,60	0,07	530	0,61	0,61	0,04	863	0,58	0,60	0,04
198	0,59	0,60	0,07	531	0,61	0,61	0,04	864	0,58	0,60	0,04
199	0,59	0,60	0,07	532	0,61	0,61	0,04	865	0,58	0,60	0,04
200	0,59	0,60	0,07	533	0,61	0,61	0,04	866	0,58	0,60	0,04
201	0,59	0,60	0,07	534	0,61	0,61	0,04	867	0,58	0,60	0,04
202	0,59	0,60	0,07	535	0,61	0,61	0,04	868	0,58	0,60	0,04
203	0,59	0,60	0,07	536	0,61	0,61	0,04	869	0,58	0,60	0,04
204	0,59	0,60	0,07	537	0,61	0,61	0,04	870	0,58	0,60	0,04
205	0,59	0,60	0,07	538	0,61	0,61	0,04	871	0,58	0,60	0,04
206	0,59	0,60	0,07	539	0,61	0,61	0,04	872	0,58	0,60	0,04
207	0,59	0,60	0,07	540	0,61	0,61	0,04	873	0,58	0,60	0,04
208	0,59	0,60	0,07	541	0,61	0,61	0,04	874	0,58	0,60	0,04
209	0,59	0,60	0,07	542	0,61	0,61	0,04	875	0,58	0,60	0,04
210	0,59	0,60	0,07	543	0,61	0,61	0,04	876	0,58	0,60	0,04
211	0,60	0,60	0,07	544	0,61	0,61	0,04	877	0,58	0,60	0,04
212	0,60	0,60	0,07	545	0,61	0,61	0,04	878	0,58	0,60	0,04
213	0,59	0,60	0,07	546	0,61	0,61	0,04	879	0,58	0,60	0,04
214	0,59	0,60	0,07	547	0,61	0,61	0,04	880	0,58	0,60	0,04
215	0,59	0,60	0,07	548	0,61	0,61	0,04	881	0,58	0,60	0,04

216	0,59	0,60	0,07
217	0,59	0,60	0,07
218	0,60	0,60	0,07
219	0,60	0,60	0,07
220	0,60	0,60	0,07
221	0,60	0,60	0,07
222	0,60	0,60	0,07
223	0,60	0,60	0,07
224	0,60	0,60	0,07
225	0,60	0,60	0,07
226	0,60	0,60	0,07
227	0,61	0,60	0,07
228	0,60	0,60	0,07
229	0,60	0,60	0,07
230	0,60	0,60	0,06
231	0,60	0,60	0,06
232	0,60	0,60	0,06
233	0,60	0,60	0,06
234	0,60	0,60	0,06
235	0,60	0,60	0,06
236	0,60	0,60	0,06
237	0,60	0,60	0,06
238	0,60	0,60	0,06
239	0,60	0,60	0,06
240	0,60	0,60	0,06
241	0,60	0,60	0,06
242	0,60	0,60	0,06
243	0,60	0,60	0,06
244	0,60	0,60	0,06
245	0,60	0,60	0,06
246	0,60	0,60	0,06
247	0,60	0,60	0,06
248	0,60	0,60	0,06
249	0,60	0,60	0,06
250	0,60	0,60	0,06
251	0,60	0,60	0,06
252	0,60	0,60	0,06
253	0,60	0,60	0,06

549	0,61	0,61	0,04
550	0,60	0,61	0,04
551	0,60	0,61	0,04
552	0,61	0,61	0,04
553	0,60	0,61	0,04
554	0,60	0,61	0,04
555	0,60	0,61	0,04
556	0,60	0,61	0,04
557	0,60	0,61	0,04
558	0,60	0,61	0,04
559	0,60	0,61	0,04
560	0,60	0,61	0,04
561	0,60	0,61	0,04
562	0,60	0,61	0,04
563	0,60	0,61	0,04
564	0,60	0,61	0,04
565	0,60	0,61	0,04
566	0,60	0,61	0,04
567	0,60	0,61	0,04
568	0,60	0,61	0,04
569	0,60	0,61	0,04
570	0,60	0,61	0,04
571	0,60	0,61	0,04
572	0,60	0,61	0,04
573	0,60	0,61	0,04
574	0,60	0,61	0,04
575	0,60	0,61	0,04
576	0,60	0,61	0,04
577	0,60	0,61	0,04
578	0,60	0,61	0,04
579	0,60	0,61	0,04
580	0,60	0,61	0,04
581	0,60	0,61	0,04
582	0,60	0,61	0,04
583	0,60	0,61	0,04
584	0,60	0,61	0,04
585	0,60	0,61	0,04
586	0,60	0,61	0,04

882	0,58	0,60	0,04
883	0,58	0,60	0,04
884	0,58	0,60	0,04
885	0,58	0,60	0,04
886	0,58	0,60	0,04
887	0,58	0,60	0,04
888	0,58	0,60	0,03
889	0,58	0,60	0,03
890	0,58	0,60	0,03
891	0,58	0,60	0,03
892	0,58	0,60	0,03
893	0,58	0,60	0,03
894	0,58	0,60	0,03
895	0,58	0,60	0,03
896	0,58	0,60	0,03
897	0,58	0,60	0,03
898	0,58	0,60	0,03
899	0,58	0,60	0,03
900	0,58	0,60	0,03
901	0,58	0,60	0,03
902	0,58	0,60	0,03
903	0,58	0,60	0,03
904	0,58	0,60	0,03
905	0,58	0,60	0,03
906	0,58	0,60	0,03
907	0,58	0,60	0,03
908	0,58	0,60	0,03
909	0,58	0,60	0,03
910	0,58	0,60	0,03
911	0,58	0,60	0,03
912	0,58	0,60	0,03
913	0,58	0,60	0,03
914	0,58	0,60	0,03
915	0,58	0,60	0,03
916	0,58	0,60	0,03
917	0,58	0,60	0,03
918	0,58	0,60	0,03
919	0,58	0,60	0,03

254	0,60	0,60	0,06	587	0,60	0,61	0,04	920	0,58	0,60	0,03
255	0,60	0,60	0,06	588	0,60	0,61	0,04	921	0,58	0,60	0,03
256	0,60	0,60	0,06	589	0,60	0,61	0,04	922	0,58	0,60	0,03
257	0,60	0,60	0,06	590	0,60	0,61	0,04	923	0,58	0,60	0,03
258	0,60	0,60	0,06	591	0,60	0,61	0,04	924	0,58	0,60	0,03
259	0,60	0,60	0,06	592	0,60	0,61	0,04	925	0,58	0,60	0,03
260	0,60	0,60	0,06	593	0,60	0,61	0,04	926	0,58	0,60	0,03
261	0,60	0,60	0,06	594	0,60	0,61	0,04	927	0,58	0,60	0,03
262	0,60	0,60	0,06	595	0,60	0,61	0,04	928	0,58	0,60	0,03
263	0,60	0,60	0,06	596	0,59	0,61	0,04	929	0,58	0,60	0,03
264	0,60	0,60	0,06	597	0,59	0,61	0,04	930	0,58	0,60	0,03
265	0,60	0,60	0,06	598	0,59	0,61	0,04	931	0,58	0,60	0,03
266	0,60	0,60	0,06	599	0,59	0,61	0,04	932	0,58	0,60	0,03
267	0,60	0,60	0,06	600	0,59	0,61	0,04	933	0,58	0,60	0,03
268	0,60	0,60	0,06	601	0,59	0,61	0,04	934	0,58	0,60	0,03
269	0,60	0,60	0,06	602	0,59	0,61	0,04	935	0,58	0,60	0,03
270	0,60	0,60	0,06	603	0,59	0,61	0,04	936	0,58	0,60	0,03
271	0,60	0,60	0,06	604	0,59	0,61	0,04	937	0,58	0,60	0,03
272	0,61	0,60	0,06	605	0,59	0,61	0,04	938	0,58	0,60	0,03
273	0,60	0,60	0,06	606	0,59	0,61	0,04	939	0,58	0,60	0,03
274	0,60	0,60	0,06	607	0,59	0,60	0,04	940	0,58	0,60	0,03
275	0,60	0,60	0,06	608	0,59	0,60	0,04	941	0,58	0,60	0,03
276	0,60	0,60	0,06	609	0,59	0,60	0,04	942	0,58	0,60	0,03
277	0,60	0,60	0,06	610	0,59	0,60	0,04	943	0,58	0,60	0,03
278	0,60	0,60	0,06	611	0,59	0,60	0,04	944	0,58	0,60	0,03
279	0,60	0,60	0,06	612	0,59	0,60	0,04	945	0,58	0,60	0,03
280	0,60	0,60	0,06	613	0,59	0,60	0,04	946	0,58	0,60	0,03
281	0,60	0,60	0,06	614	0,59	0,60	0,04	947	0,58	0,60	0,03
282	0,60	0,60	0,06	615	0,59	0,60	0,04	948	0,58	0,60	0,03
283	0,61	0,60	0,06	616	0,59	0,60	0,04	949	0,58	0,60	0,03
284	0,61	0,60	0,06	617	0,59	0,60	0,04	950	0,58	0,60	0,03
285	0,61	0,60	0,06	618	0,59	0,60	0,04	951	0,58	0,60	0,03
286	0,61	0,60	0,06	619	0,59	0,60	0,04	952	0,58	0,60	0,03
287	0,61	0,60	0,06	620	0,59	0,60	0,04	953	0,58	0,60	0,03
288	0,60	0,60	0,06	621	0,59	0,60	0,04	954	0,58	0,60	0,03
289	0,60	0,60	0,06	622	0,59	0,60	0,04	955	0,58	0,60	0,03
290	0,60	0,60	0,06	623	0,58	0,60	0,04	956	0,58	0,60	0,03
291	0,60	0,60	0,06	624	0,58	0,60	0,04	957	0,58	0,60	0,03
292	0,60	0,60	0,06	625	0,58	0,60	0,04	958	0,58	0,60	0,03

293	0,60	0,60	0,06
294	0,61	0,60	0,06
295	0,60	0,60	0,06
296	0,60	0,60	0,06
297	0,61	0,60	0,06
298	0,61	0,60	0,06
299	0,61	0,60	0,06
300	0,61	0,60	0,06
301	0,61	0,60	0,06
302	0,61	0,60	0,06
303	0,60	0,60	0,06
304	0,60	0,60	0,06
305	0,61	0,60	0,06
306	0,61	0,60	0,06
307	0,61	0,60	0,06
308	0,60	0,60	0,06
309	0,60	0,60	0,06
310	0,60	0,60	0,06
311	0,60	0,60	0,06
312	0,61	0,60	0,06
313	0,61	0,60	0,06
314	0,60	0,60	0,06
315	0,60	0,60	0,06
316	0,60	0,60	0,06
317	0,60	0,60	0,06
318	0,60	0,60	0,06
319	0,60	0,60	0,06
320	0,61	0,60	0,06
321	0,60	0,60	0,06
322	0,60	0,60	0,05
323	0,60	0,60	0,05
324	0,60	0,60	0,05
325	0,60	0,60	0,05
326	0,60	0,60	0,05
327	0,61	0,60	0,05
328	0,61	0,60	0,05
329	0,61	0,60	0,05
330	0,61	0,60	0,05

626	0,58	0,60	0,04
627	0,58	0,60	0,04
628	0,58	0,60	0,04
629	0,58	0,60	0,04
630	0,58	0,60	0,04
631	0,58	0,60	0,04
632	0,58	0,60	0,04
633	0,58	0,60	0,04
634	0,58	0,60	0,04
635	0,58	0,60	0,04
636	0,58	0,60	0,04
637	0,58	0,60	0,04
638	0,58	0,60	0,04
639	0,58	0,60	0,04
640	0,58	0,60	0,04
641	0,58	0,60	0,04
642	0,58	0,60	0,04
643	0,58	0,60	0,04
644	0,58	0,60	0,04
645	0,58	0,60	0,04
646	0,58	0,60	0,04
647	0,58	0,60	0,04
648	0,58	0,60	0,04
649	0,58	0,60	0,04
650	0,58	0,60	0,04
651	0,58	0,60	0,04
652	0,58	0,60	0,04
653	0,58	0,60	0,04
654	0,58	0,60	0,04
655	0,58	0,60	0,04
656	0,58	0,60	0,04
657	0,58	0,60	0,04
658	0,58	0,60	0,04
659	0,58	0,60	0,04
660	0,58	0,60	0,04
661	0,58	0,60	0,04
662	0,58	0,60	0,04
663	0,58	0,60	0,04

959	0,58	0,60	0,03
960	0,58	0,60	0,03
961	0,58	0,60	0,03
962	0,58	0,60	0,03
963	0,58	0,60	0,03
964	0,58	0,60	0,03
965	0,58	0,60	0,03
966	0,58	0,60	0,03
967	0,58	0,60	0,03
968	0,58	0,60	0,03
969	0,58	0,60	0,03
970	0,58	0,60	0,03
971	0,58	0,60	0,03
972	0,58	0,60	0,03
973	0,58	0,60	0,03
974	0,58	0,60	0,03
975	0,58	0,60	0,03
976	0,58	0,60	0,03
977	0,58	0,60	0,03
978	0,58	0,60	0,03
979	0,58	0,60	0,03
980	0,58	0,60	0,03
981	0,58	0,60	0,03
982	0,58	0,60	0,03
983	0,58	0,60	0,03
984	0,58	0,60	0,03
985	0,58	0,60	0,03
986	0,58	0,60	0,03
987	0,58	0,60	0,03
988	0,58	0,60	0,03
989	0,58	0,60	0,03
990	0,58	0,60	0,03
991	0,58	0,60	0,03
992	0,58	0,60	0,03
993	0,58	0,60	0,03
994	0,58	0,60	0,03
995	0,58	0,60	0,03
996	0,58	0,60	0,03

331	0,61	0,60	0,05
332	0,61	0,60	0,05
333	0,61	0,60	0,05

664	0,58	0,60	0,04
665	0,58	0,60	0,04
666	0,58	0,60	0,04

997	0,58	0,60	0,03
998	0,58	0,60	0,03
999	0,58	0,60	0,03
1.000	0,58	0,60	0,03