



Universidade Federal da Bahia
Instituto de Matemática

Programa de Pós-Graduação em Ciência da Computação

**APLICANDO REDES DE FUNÇÃO DE BASE
RADIAL E CADEIAS DE MARKOV PARA A
DETECÇÃO DE MUDANÇAS DE CONCEITO
EM FLUXOS DE DADOS**

Ruivaldo Azevedo Lobão Neto

DISSERTAÇÃO DE MESTRADO

Salvador
13 de Dezembro de 2019

RUIVALDO AZEVEDO LOBÃO NETO

**APLICANDO REDES DE FUNÇÃO DE BASE RADIAL E CADEIAS
DE MARKOV PARA A DETECÇÃO DE MUDANÇAS DE
CONCEITO EM FLUXOS DE DADOS**

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Ricardo Araújo Rios

Salvador
13 de Dezembro de 2019

RESUMO

A quantidade de informações produzidas por sistemas computacionais tem crescido de forma acentuada nas últimas décadas. Uma parcela expressiva dessas informações é produzida na forma de fluxos contínuos de dados, que são sequências constantes e potencialmente infinitas. Esses fluxos são, em sua maioria, não-estacionários, podendo sofrer variações na distribuição dos dados ou no contexto do processo gerador. Estas alterações são denominadas mudanças de conceito e podem impactar negativamente a performance de modelos de aprendizado aplicados. Para mitigar este problema, pesquisadores desenvolveram métodos especializados na detecção de mudanças de conceito. Entretanto, os métodos propostos apresentam limitações ao serem aplicados em cenários com fluxos contínuos, como a necessidade de rotulação por especialistas ou a incapacidade de atender às restrições de tempo de processamento e de uso dos recursos computacionais desses cenários. Este trabalho propõe um novo método de detecção baseado em Redes de Função de Base Radial (RBF) e Cadeias de Markov, denominado RBFChain. As redes RBF realizam, em sua camada intermediária, um processo de ativação que implicitamente produz grupos das observações recepcionadas ao longo do tempo. Simultaneamente, cadeias de Markov modelam as transições de centro ocorridas no agrupamento formado. Mudanças de conceito são detectadas quando o centro ativo do agrupamento é alterado e a probabilidade da transição, no modelo markoviano, excede um limite. O método proposto se diferencia dos trabalhos existentes por detectar mudanças em tempo de execução, de forma computacionalmente eficiente e independente de rótulos. Para avaliar o RBFChain como um detector de mudanças de conceito viável, uma análise de sensibilidade, precisão e tolerância ao ruído foi realizada usando conjuntos de dados sintéticos, e os resultados comparados com os principais algoritmos disponíveis na literatura. Além disso, a técnica foi aplicada ao problema de classificação de fixações e sacadas na atividade de monitoramento ocular: uma questão relevante para diferentes áreas do conhecimento, pois muitos experimentos comportamentais usam essas informações como um fator de análise. Os resultados com conjuntos de dados sintéticos sugerem que o RBFChain é estatisticamente melhor ou equivalente aos principais detectores presentes na literatura. Ademais, a técnica desenvolvida também apresentou bons resultados quando aplicada ao problema de monitoramento ocular, sendo capaz de classificar fixações e sacadas em tempo real e com precisão equivalente ao estado da arte.

Palavras-chave: Fluxo Contínuo de Dados. Mudança de Conceito. Rede de Função de Base Radial. Cadeia de Markov. Monitoramento Ocular.

ABSTRACT

The amount of information produced by computer systems has grown dramatically in recent decades. A significant share of this volume is produced as data streams, which are constant and potentially infinite sequences. Most of these streams are non-stationary and may vary over time. These changes are called concept drifts and can negatively impact the performance of applied learning models. To mitigate this problem, researchers developed specialized methods for detecting concept drifts. However, the proposed methods have limitations when applied in scenarios with data streams, such as the need for expert labeling or the inability to meet the processing time and computational resource constraints of these scenarios. This work proposes a new detection method based on Radial Basis Function Networks (RBF) and Markov Chains, called RBFChain. RBF networks perform, in their intermediate layer, an activation process that implicitly produces groups of observations received over time. Simultaneously, the Markov Chain models the transitions between centers that occur in the formed cluster. Concept drifts are detected when the active center of the cluster changes, and the probability of this transition in the Markov model exceeds a threshold. The proposed method differs from existing works in that it detects changes at run time, in a computationally efficient and independent of labels way. To assess RBFChain as a viable concept drift detector, an analysis of sensitivity, accuracy, and noise tolerance was performed using synthetic datasets, and results were compared to the main algorithms available in the literature. Also, the technique was applied to the fixation and saccade classification problem in eye-tracking: a relevant issue for different areas of knowledge, because many behavioral experiments use this information as an analysis factor. Results with synthetic datasets suggest that RBFChain is statistically better or equivalent to the main detectors present in the literature. Moreover, the developed method also presented good results when applied to the eye-tracking problem, being able to classify fixations and saccades in real-time and with accuracy equivalent to the state of the art.

Keywords: Data Stream. Concept Drift. Radial Basis Function Network. Markov Chain. Eye-tracking.

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Contexto e Motivação	1
1.2 Hipótese e Objetivo	2
1.3 Contribuições	3
1.4 Organização do trabalho	3
Capítulo 2—Fundamentação Teórica	5
2.1 Considerações Iniciais	5
2.2 Fluxos Contínuos de Dados e Aprendizado de Máquina	5
2.3 Mudança de Conceito	7
2.3.1 Terminologia	9
2.3.2 Algoritmos para Detecção de Mudança de Conceito	10
2.3.3 Ferramenta - MOA	12
2.4 Redes de Função de Base Radial	14
2.5 Cadeias de Markov	16
2.6 Monitoramento Ocular - Fixações e Sacadas	18
2.7 Trabalhos Relacionados	19
2.8 Considerações Finais	22
Capítulo 3—RBFChain	23
3.1 Considerações Iniciais	23
3.2 Visão Geral	23
3.3 RBFChain	25
3.4 Considerações Finais	26
Capítulo 4—Experimentos e Resultados	27
4.1 Considerações Iniciais	27
4.2 Configuração dos Experimentos	27
4.3 Critérios de avaliação	28
4.4 Pettitt	29
4.5 RBFChain	30
4.6 Considerações Finais	34

Capítulo 5—Conclusões e Trabalhos Futuros	35
5.1 Discussões	35
5.2 Trabalhos Futuros	35

LISTA DE FIGURAS

2.1	Mudança de Conceito Virtual vs. Mudança de Conceito Real	8
2.2	Padrões de ocorrência de Mudanças de Conceito	8
2.3	MOA - Tela Inicial	13
2.4	MOA - Configuração detector	14
2.5	Arquitetura RBF	15
2.6	Exemplo: Cadeia de Markov com três estados. Fonte: Wikipedia (2019).	17
2.7	Exemplo de identificação de fixações e sacadas. Fonte: König e Buffalo (2014).	18
3.1	Arquitetura RBFChain	24
3.2	Exemplo de funcionamento do algoritmo	26
4.1	Representação Gráfica - Sem mudanças de conceito	31
4.2	Representação Gráfica - Mudanças Abruptas	33
4.3	Representação Gráfica - Mudanças Graduais	34

LISTA DE TABELAS

2.1	Terminologia - Mudança de Conceito (ZLIOBAITE, 2010)	9
2.2	Resumo - Algoritmos de detecção (SETHI; KANTARDZIC, 2017)	12
4.1	Configuração da classe <i>BasicConceptDriftPerformanceEvaluator</i>	28
4.2	Indicadores analisados	29
4.3	Resultados - Método de Pettitt	29
4.4	Parâmetros utilizados para cada algoritmo	30
4.5	Experimento 1 - Fluxo sem mudanças de conceito	30
4.6	Experimento 2 - Fluxo com mudanças de conceito abruptas	32
4.7	Experimento 3 - Fluxo com mudanças de conceito graduais	33

LISTA DE ALGORITMOS

1 RBFCHAIN 25

INTRODUÇÃO

1.1 CONTEXTO E MOTIVAÇÃO

Nos últimos anos, o volume de dados produzidos por sistemas computacionais tem crescido de forma acentuada. Esse crescimento foi favorecido por avanços tecnológicos recentes, como a pervasividade dos dispositivos móveis, a popularização das redes sociais e a expansão da internet das coisas (COHEN et al., 2009). A dimensão desse aumento foi verificada por Zwolenski e Weatherill (2014), os quais estimaram que, entre os anos de 2014 e 2020, a quantidade de informações produzidas anualmente irá aumentar de 4,4 zettabytes (trilhões de gigabytes) para 44 zettabytes.

Parte significativa dessas informações é produzida na forma de sequências ininterruptas e potencialmente infinitas (AGGARWAL, 2006). Na literatura, sequências com essas características são denominadas Fluxos Contínuos de Dados (FCDs) e estão presentes em diversos domínios de aplicação como, por exemplo, monitoramento do mercado financeiro (ZHOU et al., 2015), acompanhamento de tráfego rodoviário (WANG et al., 2015), gerenciamento de redes de telecomunicação (DELATTRE; IMBERT, 2015), análise de sentimento em tempo real (KRANJC et al., 2015) e sistemas de prevenção e identificação de intrusos (KENKRE; PAI; COLACO, 2015).

Para extrair informações úteis dessa grande quantidade de dados, pesquisadores têm aplicado técnicas da área de Aprendizado de Máquina (AM), a qual estuda algoritmos que melhoram seu desempenho conforme ganham experiência (MITCHELL, 1997). Entretanto, as estratégias tradicionais de Aprendizado de Máquina têm aplicação limitada para contextos com fluxos contínuos de dados, pois nesses cenários os algoritmos devem atender às severas restrições de tempo de execução e de uso dos recursos computacionais (BIFET; KIRKBY, 2009).

Além dessas limitações, as técnicas de Aprendizado de Máquina, quando aplicadas em contextos com fluxos contínuos, também devem lidar com variações na distribuição dos dados ou no contexto do processo gerador. Essas alterações são denominadas Mudanças de Conceito (GAMA, 2010) e a sua ocorrência pode impactar a acurácia do modelo.

Inicialmente, a atualização periódica dos modelos obtidos a partir das técnicas de AM foi utilizada como estratégia para evitar a perda de acurácia causada por tais mudanças. Contudo, esta solução é pouco sofisticada e computacionalmente custosa. Diante disso, pesquisadores propuseram técnicas de detecção de mudanças de conceito baseadas em monitoramento da performance do modelo (GAMA et al., 2014). Estes métodos identificam a ocorrência de mudanças com maior precisão, permitindo que o modelo de decisão seja atualizado somente quando necessário. Exemplos de algoritmos baseados nesta abordagem, incluem: DDM (GAMA et al., 2004), EDDM (BAENA-GARCÍA et al., 2006), ADWIN (BIFET; GAVALDÀ, 2007), ECDD (ROSS et al., 2012), PL (Bach; Maloof, 2008), FCWM (SEBASTIÃO et al., 2010) e STEPDP (NISHIDA; YAMAUCHI, 2007).

Entretanto, as técnicas baseadas em monitoramento necessitam que o rótulo correto de cada exemplo esteja disponível. Em muitos cenários, o tempo ou o custo para obter esses rótulos é proibitivo (AGGARWAL, 2006). Consequentemente, foram desenvolvidos novos algoritmos independentes de rótulos. Nestes métodos, a detecção se baseia na identificação de exemplos que não se enquadram na estrutura conhecida dos dados (SPINOSA; CARVALHO; GAMA, 2007). Geralmente, essa análise é implementada com base em técnicas de agrupamento, detecção de *outliers* ou medidas de dissimilaridade (RYU et al., 2012). Os seguintes algoritmos são exemplos desta abordagem: OLINDDA (SPINOSA; CARVALHO; GAMA, 2007), MINAS (FARIA; GAMA; CARVALHO, 2013), ECSMiner (MASUD et al., 2011) e GC3 (SETHI; KANTARDZIC; HU, 2016).

Todavia, segundo Aggarwal (2006), as técnicas de detecção de mudanças de conceito presentes na literatura ainda apresentam limitações ao serem aplicadas em cenários com fluxos contínuos de dados. Os algoritmos dependentes de rótulo se tornam inviáveis, por causa do custo e do tempo necessário para obter os rótulos corretos. Enquanto as técnicas independentes têm dificuldade em atender às severas restrições de tempo de execução e de uso dos recursos computacionais desses cenários.

Diante disso, esta dissertação de mestrado propõe uma nova abordagem baseada em Redes de Função de Base Radial (RBF) e Cadeias de Markov para detecção de mudanças de conceito em fluxos contínuos de dados. O método desenvolvido se diferencia dos trabalhos existentes por detectar mudanças de conceito em tempo de execução, de forma computacionalmente eficiente, tolerante a ruídos e independente de rótulos.

1.2 HIPÓTESE E OBJETIVO

Com base nas observações citadas anteriormente, a seguinte hipótese foi formulada:

“A aplicação de Redes de Função de Base Radial e Cadeias de Markov em fluxos contínuos de dados permite a detecção de mudanças de conceito em tempo de execução, de forma computacionalmente eficiente, tolerante a ruídos e independente de rótulos.”

Assim, o objetivo principal deste trabalho foi a validação da hipótese elaborada. Para atingir este objetivo, um novo método de detecção de mudanças de conceito, baseado em Redes de Função de Base Radial e Cadeias de Markov, foi desenvolvido. A técnica implementada foi validada através de dois experimentos principais, um com dados sintéticos e outro baseado em dados de um problema do mundo real.

No primeiro experimento, a técnica foi aplicada a conjuntos de dados sintéticos e comparada aos principais métodos presentes na literatura. Este experimento permitiu validar a efetividade do método proposto e realizar uma análise detalhada da abordagem, uma vez que as características e os comportamentos dos fluxos são conhecidos.

No segundo experimento, o método foi adaptado para o problema de classificação de fixações e sacadas durante a atividade de monitoramento ocular. Este experimento permitiu demonstrar uma aplicação prática e relevante da solução proposta neste trabalho.

1.3 CONTRIBUIÇÕES

A principal contribuição desta dissertação consiste em um novo método para detecção de mudanças de conceito em fluxos contínuos de dados. O método desenvolvido é baseado em Redes de Função de Base Radial e Cadeias de Markov. As principais vantagens da técnica proposta em relação aos métodos existentes na literatura são:

- Detecção de mudanças de conceito em tempo de execução;
- Independência de rótulos;
- Tolerância a ruídos;
- Eficiência computacional.

A contribuição secundária deste trabalho reside na adaptação do método desenvolvido para classificação de fixações e sacadas na atividade de monitoramento ocular. Um problema relevante para diferentes áreas do conhecimento, pois muitos experimentos comportamentais usam essas informações como fator de análise.

1.4 ORGANIZAÇÃO DO TRABALHO

O restante do texto está organizado da seguinte forma: o Capítulo 2 apresenta o referencial teórico que fundamenta esta pesquisa e os trabalhos relacionados. A solução proposta é apresentada em detalhes no Capítulo 3. A configuração dos experimentos, a avaliação da proposta e os resultados alcançados são apresentados no Capítulo 4. Por fim, as conclusões obtidas e os trabalhos futuros são apresentados no Capítulo 5.

FUNDAMENTAÇÃO TEÓRICA

2.1 CONSIDERAÇÕES INICIAIS

Este capítulo apresenta os principais conceitos utilizados nesta pesquisa e discute os trabalhos relacionados observados na literatura. Inicialmente, será abordada a relação entre Fluxos Contínuos de Dados e técnicas de Aprendizado de Máquina. Em seguida, o fenômeno Mudança de Conceito, os métodos de detecção e as ferramentas de análise serão discutidas. Logo após, as Redes de Função de Base Radial e as Cadeias de Markov são apresentadas. Posteriormente, o problema de classificação de fixações e sacadas em atividades de monitoramento ocular é detalhado. Por fim, os trabalhos relacionados observados na literatura são examinados.

2.2 FLUXOS CONTÍNUOS DE DADOS E APRENDIZADO DE MÁQUINA

Fluxos Contínuos de Dados (FCDs) podem ser definidos como sequências ininterruptas e potencialmente infinitas de eventos (AGGARWAL, 2006). Nestes fluxos, os eventos ocorrem em alta frequência, sendo necessário processá-los em tempo de execução. Além disso, por serem de tamanho potencialmente ilimitado, não é possível armazená-los de forma permanente em memória.

As características dos fluxos contínuos de dados implicam as seguintes restrições aos algoritmos que os processam (BIFET; KIRKBY, 2009):

1. É impossível armazenar todos os dados do fluxo. Somente uma pequena parcela pode ser processada e armazenada, enquanto o restante é descartado;
2. A velocidade de chegada dos eventos exige que sejam processados em tempo de execução;
3. A distribuição dos dados pode mudar com o tempo. Assim, os dados do passado podem se tornar irrelevantes ou mesmo prejudiciais para a descrição dos conceitos atuais.

A área de Aprendizado de Máquina (AM) estuda algoritmos que melhoram o seu desempenho conforme ganham experiência (MITCHELL, 1997). Esses algoritmos se dividem em duas categorias principais: não supervisionados (agrupamento ou *clustering*) e supervisionados (classificação ou regressão). Algoritmos de ambas as categorias foram adaptados para que pudessem ser aplicados em cenários com fluxos contínuos de dados. As principais características de cada categoria e as especializações propostas serão discutidas a seguir.

As técnicas não supervisionadas realizam o agrupamento automático de dados segundo o seu grau de semelhança. Essas técnicas têm como objetivo a formação de grupos com alta similaridade intragrupo e baixa similaridade intergrupo (JAIN; DUBES, 1988). Os seguintes algoritmos são exemplos de técnicas não supervisionadas para cenários em lote: K-Means (LLOYD, 2006), DBSCAN (ESTER et al., 1996), PAM (KAUFMAN; ROUSSEEUW, 1990) e OPTICS (ANKERST et al., 1999).

De acordo com Gama (2010), a principal dificuldade ao aplicar técnicas não supervisionadas em cenários com fluxos contínuos é a manutenção da qualidade e consistência dos grupos formados conforme novos dados são observados. Portanto, é necessário que os algoritmos atuem de forma incremental, evoluindo os grupos formados ao longo do tempo (BARBARÁ, 2002). Sendo assim, foram desenvolvidos métodos não supervisionados especializados para fluxos contínuos de dados. Os seguintes trabalhos são exemplos dessas especializações: CluStream (AGGARWAL et al., 2003), StreamKM++ (ACKERMANN et al., 2012), DenStream (CAO et al., 2006), D-Stream (LING; LING-JUN; LI, 2009) e ClusTree (KRANEN et al., 2011).

Os algoritmos supervisionados realizam previsões para novos exemplos utilizando um modelo criado a partir de uma base de treinamento (KOTSIANTIS, 2007). Se a predição é categórica, entende-se como um problema de classificação. Se a predição resulta em um valor numérico, trata-se de uma tarefa de regressão. Exemplos de algoritmos supervisionados para cenários em lote, incluem: árvores de decisão (BREIMAN et al., 1984), métodos baseados em regras, redes neurais e máquinas de vetores suporte (SVM) (VAPNIK, 1998).

Segundo Gama e Gaber (2010), as técnicas supervisionadas tradicionais não podem ser aplicadas a contextos com fluxos contínuos de dados, pois estes métodos não contemplam as severas restrições de uso de memória e de tempo de execução desses cenários. Dessa forma, novos algoritmos supervisionados foram propostos para esses contextos (DOMINGOS; HULTEN, 2000; BIFET et al., 2013; WANG et al., 2003; AGGARWAL et al., 2004; GAMA; ROCHA; MEDAS, 2003).

As especializações mencionadas buscam atender às restrições de uso de memória e de tempo de execução dos contextos com fluxos contínuos de dados. Contudo, não consideram que na maioria desses cenários as informações são geradas de forma não estacionária, e que a distribuição dos dados ou o contexto do processo gerador podem sofrer variações, alterando os resultados esperados. Na literatura, essas variações são denominadas mudanças de conceito e a sua ocorrência pode impactar a acurácia da técnica aplicada (GAMA, 2010).

Nesta dissertação, considera-se que os dados são obtidos a partir de fluxos contínuos de dados com ocorrência de mudanças de conceito.

2.3 MUDANÇA DE CONCEITO

Técnicas de Aprendizado de Máquina aplicadas a cenários com fluxos contínuos de dados devem ser capazes de lidar com alterações na distribuição dos dados ou no contexto do processo gerador. Essas alterações são denominadas Mudanças de Conceito (*Concept Drift* ou *Concept Shift*, em inglês) e podem alterar os resultados esperados (conceitos-alvo) dos algoritmos, prejudicando sua acurácia (WIDMER; KUBAT, 1996).

Na literatura, é comum utilizar a Teoria Bayesiana de Decisão (DUDA; HART; STORK, 2000) para descrever a tarefa de classificação. Esta descrição será utilizada como base para formalização do fenômeno de mudança de conceito.

Sendo $X \in \mathbb{R}^p$ uma instância em um espaço p -dimensional de atributos e $X \in c_i$ onde c_1, c_2, \dots, c_k é o conjunto de classes, o classificador ótimo para classificar $x \rightarrow c_i$ é determinado a partir das probabilidades a priori das classes $P(c_i)$ e pela função de densidade de probabilidade condicionada às classes $p(X|c_i), i = 1, \dots, k$. É possível definir um conceito como um conjunto de probabilidades a priori e condicionais das classes, como mostra a Equação 2.1:

$$S = \{(P(c_1), P(X|c_1)), (P(c_2), P(X|c_2)), \dots, (P(c_k), P(X|c_k))\} \quad (2.1)$$

Ainda segundo a Teoria Bayesiana, a classificação de uma instância X baseada na máxima probabilidade a posteriori pode ser obtida através da Equação 2.2:

$$p(c_i|X) = \frac{p(c_i) * p(X|c_i)}{p(X)} \quad (2.2)$$

Assim, é possível afirmar que há mudança de conceito entre os instantes t_0 e t_1 se:

$$\exists X : p_{t_0}(X, c) \neq p_{t_1}(X, c) \quad (2.3)$$

onde p_{t_0} e p_{t_1} denotam as distribuições de probabilidades conjuntas nos instantes t_0 e t_1 , respectivamente, para X e c (GAMA et al., 2014). Em outras palavras, um conjunto de dados possui resultados esperados legítimos em t_0 , mas este mesmo conjunto passa a ter resultados esperados diferentes, também legítimos, em t_1 (KOLTER; MALOOF, 2007).

De acordo com Gama et al. (2014), as mudanças de conceito podem ser categorizadas como virtuais ou reais. As mudanças virtuais são causadas por alterações na probabilidade a priori das classes, $P(c)$, e não alteram os conceitos-alvo. Enquanto que as mudanças de conceito reais surgem a partir de alterações na probabilidade a posteriori, $p(c|X)$, e modificam os resultados esperados. Os dois tipos de mudança de conceito estão representados na Figura 2.1.

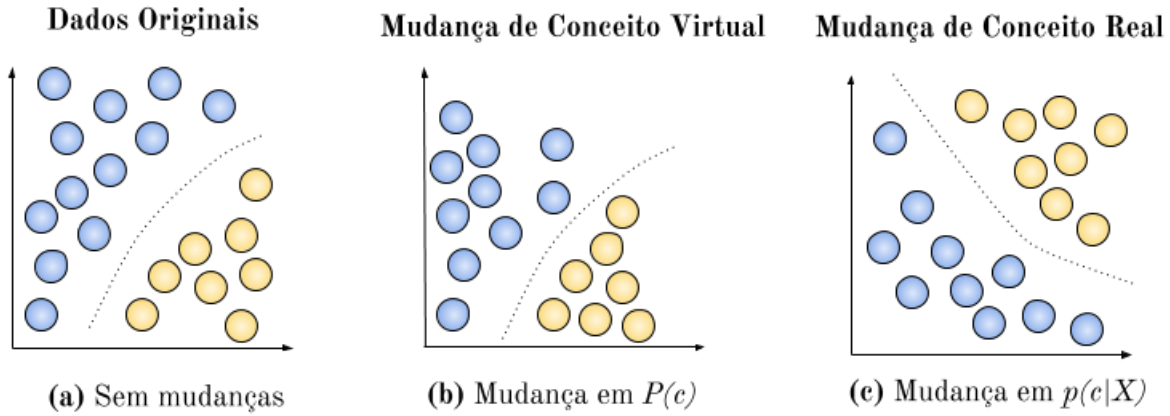


Figura 2.1 Mudança de Conceito Virtual vs. Mudança de Conceito Real

Conforme Zliobaite (2010), as mudanças de conceito podem ocorrer de forma abrupta, gradual, incremental ou recorrente. A Figura 2.2 ilustra estes padrões, utilizando círculos na cor azul para representar o conceito A e círculos na cor bege para o conceito B :



Figura 2.2 Padrões de ocorrência de Mudanças de Conceito

Na mudança abrupta, o conceito A é repentinamente substituído pelo conceito B (Figura 2.2 (a)).

Na mudança gradual, ocorre uma transição mais suave entre os conceitos A e B . Inicialmente, eventos pertencentes a ambos os conceitos coexistem. Com o passar do tempo, os eventos pertencentes ao conceito A diminuem de frequência, até pararem de ocorrer. Por fim, os eventos pertencentes a B se tornam predominantes (Figura 2.2 (b)).

A mudança incremental descreve a evolução de um único conceito ao longo do tempo. Essa evolução pode ser discretizada como uma sequência de conceitos consecutivos. Nesta

sequência, cada conceito intermediário difere pouco dos seus conceitos antecessor e sucessor. Portanto, as mudanças são notáveis apenas à longo prazo (Figura 2.2 (c)).

A mudança recorrente acontece quando um conceito anteriormente ativo reaparece após um determinado período de tempo. Contudo, não se trata de uma sazonalidade periódica, pois não é evidente o momento no qual o conceito voltará a ser ativo (Figura 2.2 (d)).

O método proposto nesta dissertação atua diretamente sobre o fluxo de dados, independentemente de rótulos. Logo, todas mudanças de conceito identificadas são tratadas como mudanças de conceito reais. Além disso, a técnica desenvolvida pode identificar mudanças em qualquer padrão de ocorrência.

2.3.1 Terminologia

O fenômeno mudança de conceito tem sido estudado em diferentes comunidades de pesquisa, incluindo Mineração de Dados, Aprendizado de Máquina, Estatística e Recuperação de Informação (ZLIOBAITE, 2010). Contudo, o tema apresenta diferentes nomeclaturas em cada comunidade. Na Tabela 2.1 são listados os termos correspondentes a mudança de conceito em cada área de pesquisa.

Tabela 2.1 Terminologia - Mudança de Conceito (ZLIOBAITE, 2010)

Área	Termos
Mineração de Dados	Mudança de Conceito
Aprendizado de Máquina	Mudança de Conceito, Mudança de Covariável
Computação Evolucionária	Ambiente Evolutivo, Ambiente em Mudança
IA e Robótica	Ambiente Dinâmico
Estatística, Séries Temporais	Não Estacionário
Recuperação de Informação	Evolução Temporal

Outra fonte comum de equívocos são os termos detecção de *outliers*, detecção de novidade, detecção de *change points* e detecção de mudança de conceito. Estes termos são muitas vezes utilizados de forma indistinta, mas, para o contexto deste trabalho, é importante distingui-los.

As técnicas para detecção de *outliers* têm como objetivo identificar padrões de dados em desacordo com o comportamento esperado. Estes padrões são geralmente classificados como anomalias ou ruídos (CHANDOLA; BANERJEE; KUMAR, 2009).

Os métodos para detecção de novidade identificam padrões ainda não observados, mas que se enquadram no comportamento esperado. Estes métodos se diferenciam das técnicas para detecção de *outliers* pois os novos padrões são incorporados ao modelo (CHANDOLA; BANERJEE; KUMAR, 2009).

As estratégias para detecção de *change points* identificam variações abruptas de valor, que podem representar transições entre estados, em séries temporais unidimensionais estacionárias (AMINIKHANGHAHI; COOK, 2017).

Por fim, cabe definir o que é detecção de mudança de conceito. Este termo abrange técnicas que monitoram a distribuição dos dados ou indicadores produzidos pelas técnicas de aprendizado aplicadas, com o objetivo de identificar variações que possam impactar a acurácia do modelo em uso (GAMA et al., 2014).

2.3.2 Algoritmos para Detecção de Mudança de Conceito

Os algoritmos para detecção de mudança de conceito caracterizam e quantificam as mudanças de conceito através da delimitação dos instantes ou intervalos de tempo em que as mudanças ocorrem (BASSEVILLE; NIKIFOROV, 1993).

Esses algoritmos se dividem em duas categorias, conforme a necessidade de rotulação dos dados (ZLIOBAITE, 2010):

Algoritmos Explícitos/Supervisionados Dependem da rotulação dos dados por um especialista. Estes rótulos são utilizados no cálculo de medidas de performance como taxa de erro e acurácia, que são monitoradas ao longo do tempo. Mudanças de conceito são sinalizadas quando essas medidas atingem um limite previamente definido.

Algoritmos Implícitos/Não Supervisionados Independem da rotulação por especialistas. As mudanças de conceito são detectadas a partir do monitoramento de características dos próprios dados ou de indicadores produzidos pelas técnicas de aprendizado aplicadas. Apesar de serem mais propensos a alarmes falsos, são uma alternativa para cenários onde a obtenção de rótulos é dispendiosa, demorada ou inviável.

Segundo Gama et al. (2014), os algoritmos *explícitos / supervisionados* podem ser segmentados em três subcategorias:

Métodos Baseados em Análise Sequencial Avaliam continuamente os indicadores de performance (por exemplo: taxa de erro) da técnica de aprendizado aplicada. A mudança de conceito é detectada quando esses indicadores atingem um limite pré-definido. Os algoritmos *Cumulative Sum (CUSUM)*, *PageHinkley (PH)* (PAGE, 1954) e *Geometric Moving Average (GMA)* (ROBERTS, 2000) são representantes desta subcategoria.

Abordagens baseadas em Estatística Identificam mudanças de conceito através da análise de parâmetros estatísticos como média e desvio padrão associados aos resultados das predições. Os métodos *Drift Detection Method (DDM)* (GAMA et al., 2004), *Early Drift Detection Method (EDDM)* (BAENA-GARCÍA et al., 2006), *Exponentially Weighted Moving Average (EWMA)* (ROSS et al., 2012) e *Reactive Drift Detection Method (RDDM)* (BARROS et al., 2017) são exemplos desta subcategoria.

Métodos baseados em Janelas Utilizam uma janela de tamanho fixo para sumarizar informações passadas e uma janela deslizante para sumarizar os dados recentes. Uma diferença significativa entre as distribuições dessas janelas implica na ocorrência de mudança de conceito. Esta diferença é verificada a partir de testes estatísticos ou desigualdades matemáticas, considerando como hipótese nula a igualdade das distribuições. Os algoritmos *Adaptive Windowing (ADWIN)* (BIFET; GAVALDÀ, 2007), *SeqDrift* (PEARS; SAKTHITHASAN; KOH, 2014), *HDDMA* e *HDDMW* (BLANCO et al., 2015) pertencem a esta subcategoria.

De forma similar, os algoritmos *implícitos / não supervisionados* também foram divididos em três subcategorias (GONÇALVES et al., 2014):

Deteção de Novidade / Métodos de Agrupamento Utilizam técnicas derivadas dos métodos de agrupamento e de detecção de *outliers* para identificar padrões ainda não observados. A partir dessa identificação, são realizados cálculos de distância e/ou densidade para confirmar a ocorrência de mudança de conceito (RYU et al., 2012). Os métodos *OLINDDA* (SPINOSA; CARVALHO; GAMA, 2007), *MINAS* (FARIA; GAMA; CARVALHO, 2013), *Woo* (RYU et al., 2012), *DETECTNOD* (Hayat; Hashemi, 2010), *ECSMiner* (MASUD et al., 2011) e *GC3* (SETHI; KANTARDZIC; HU, 2016) fazem parte desta subcategoria.

Monitoramento de distribuição multivariada Monitoram diretamente a distribuição dos dados para cada atributo. A distribuição de um conjunto de treinamento é sumarizada e utilizada como referência. Esta referência é, então, comparada à distribuição dos dados do conjunto atual. Diferenças significativas entre esses conjuntos indicam a ocorrência de mudança de conceito. Os algoritmos *CoC* (Lee; Magoulès, 2012), *HDDDM* (Ditzler; Polikar, 2011), *PCA-detect* (KUNCHEVA, 2008) são representantes desta subcategoria.

Monitoramento dependente de modelo Requerem a aplicação de um algoritmo de classificação probabilístico, pois as mudanças de conceito são detectadas a partir do monitoramento da probabilidade a posteriori calculada (ZLIOBAITE, 2010). Esta restrição permite reduzir a ocorrência de falsos positivos e torna o processo computacionalmente eficiente, pois apenas um único fluxo univariado de valores é observado. Os métodos *A-distance* (DREDZE; OATES; PIATKO, 2010), *CDBD* (LINDSTROM; NAMEE; DELANY, 2013) e *Margin* (DRIES; RÜCKERT, 2009) integram esta subcategoria.

Um resumo das categorias, subcategorias e técnicas elencadas é apresentado na Tabela 2.2.

Tabela 2.2 Resumo - Algoritmos de detecção (SETHI; KANTARDZIC, 2017)

Categoria	Subcategoria	Algoritmos
Algoritmos Explícitos/Supervisionados	Métodos Baseados em Análise Sequencial	Cumulative Sum (CUSUM)
		PageHinkley (PH) (PAGE, 1954)
		Geometric Moving Average (GMA) (ROBERTS, 2000)
	Abordagens baseadas em Estatística	Drift Detection Method (DDM) (GAMA et al., 2004)
		Early Drift Detection Method (EDDM) (BAENA-GARCÍA et al., 2006)
		Exponentially Weighted Moving Average (EWMA) (ROSS et al., 2012)
		Reactive Drift Detection Method (RDDM) (BARROS et al., 2017)
	Métodos baseados em Janelas	Adaptive Windowing (ADWIN) (BIFET; GAVALDÀ, 2007)
		SeqDrift (PEARS; SAKTHITHASAN; KOH, 2014)
		HDDMA/HDDMW (BLANCO et al., 2015)
Algoritmos Implícitos/Não Supervisionados	Detecção de Novidade / Métodos de Agrupamento	OLINDDA (SPINOSA; CARVALHO; GAMA, 2007)
		MINAS (FARIA; GAMA; CARVALHO, 2013)
		Woo (RYU et al., 2012)
		DETECTNOD (Hayat; Hashemi, 2010)
		ECSMiner (MASUD et al., 2011)
		GC3 (SETHI; KANTARDZIC; HU, 2016)
	Monitoramento de distribuição multivariada	CoC (Lee; Magoulès, 2012)
		HDDDM (Ditzler; Polikar, 2011)
		PCA-detect (KUNCHEVA, 2008)
	Monitoramento dependente de modelo	A-distance (DREDZE; OATES; PIATKO, 2010)
		CDBD (LINDSTROM; NAMEE; DELANY, 2013)
		Margin (DRIES; RÜCKERT, 2009)

O método de detecção proposto neste trabalho se enquadra na categoria de algoritmos *implícitos / não supervisionados*, mais especificamente na subcategoria *detecção de novidades / métodos de agrupamento*.

2.3.3 Ferramenta - MOA

Atualmente, o *MOA – Massive Online Analysis*¹ é o principal framework para mineração de dados em fluxos contínuos (BIFET et al., 2010). O framework é de código-aberto² e apresenta uma comunidade bastante ativa e crescente. A aplicação é composta por uma ampla coleção de algoritmos da área de Aprendizado de Máquina, contemplando técnicas de classificação, regressão, agrupamento, busca por padrões, detecção de *outliers*, detecção de mudanças de conceito e sistemas de recomendação. Além das implementações, também estão disponíveis rotinas para avaliação dessas técnicas. A aplicação é desenvolvida em Java, o que permite a sua execução nos principais sistemas operacionais e a integração com o projeto WEKA (HALL et al., 2009).

O MOA divide as suas funcionalidades em tarefas (*tasks*). Estas tarefas podem ser executadas a partir da interface gráfica (GUI) ou por linha de comando. A interface gráfica permite executar múltiplas tarefas de forma concorrente, controlar suas execuções e visualizar os resultados parciais. A tela principal da aplicação é demonstrada na Figura 2.3.

¹<https://moa.cms.waikato.ac.nz/>

²<https://github.com/Waikato/moa>

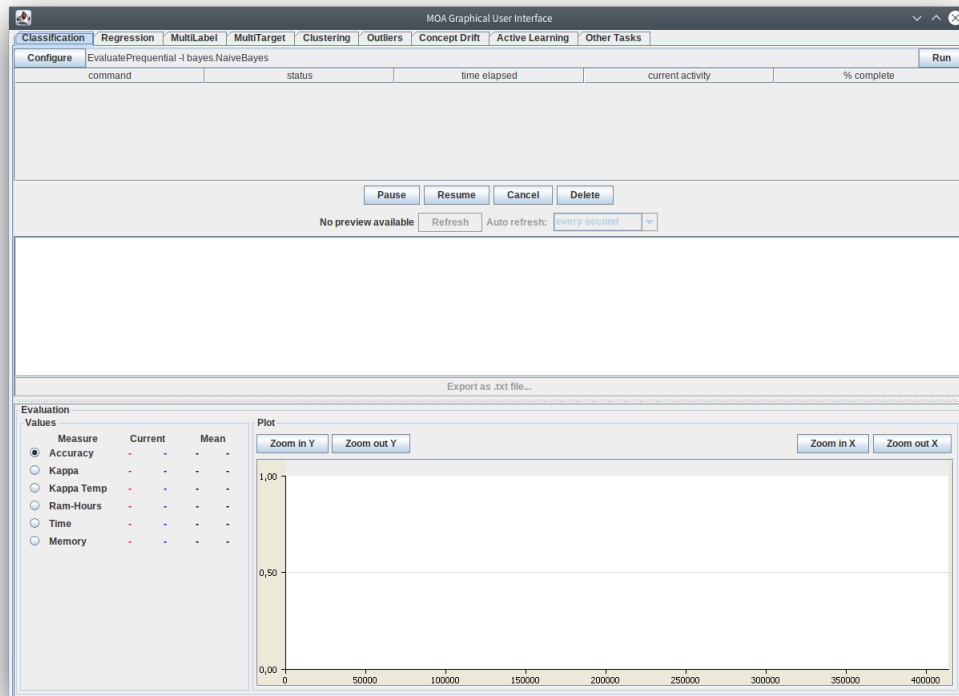


Figura 2.3 MOA - Tela Inicial

A aplicação é capaz de ler arquivos em formato ARFF³, além de permitir a produção de fluxos de dados dinamicamente, através de geradores. Alguns dos geradores de fluxo disponíveis no MOA são: *Random Trees* (DOMINGOS; HULTEN, 2000) *SEA* (STREET; KIM, 2001), *STAGGER* (SCHLIMMER; GRANGER, 1986), *Rotating Hyperplane* (WANG et al., 2003), *Random RBF*, *LED* (GAMA; ROCHA; MEDAS, 2003), *Waveform* (GAMA; ROCHA; MEDAS, 2003), e *Function* (JIN; AGRAWAL, 2003).

Outra funcionalidade importante do framework é a possibilidade de adicionar mudanças de conceito a fluxos estacionários existentes. Esse processo é realizado através de uma função sigmóide, que modela o evento de mudança de conceito como uma combinação balanceada de duas distribuições homogêneas, que caracterizam os conceitos-alvo antes e depois da mudança. Além destes conceitos, o usuário também pode definir o momento da mudança e a sua duração (BIFET et al., 2010).

A arquitetura do framework é modular, o que permite a implementação de novos algoritmos de forma trivial. Por exemplo, para incluir um novo algoritmo de detecção de mudanças de conceito, basta estender a classe abstrata *AbstractChangeDetector* e implementar o algoritmo desejado. A janela de configuração deste detector, similar a Figura 2.4, será criada dinamicamente, a partir dos atributos definidos na classe.

³Attribute-Relation File Format

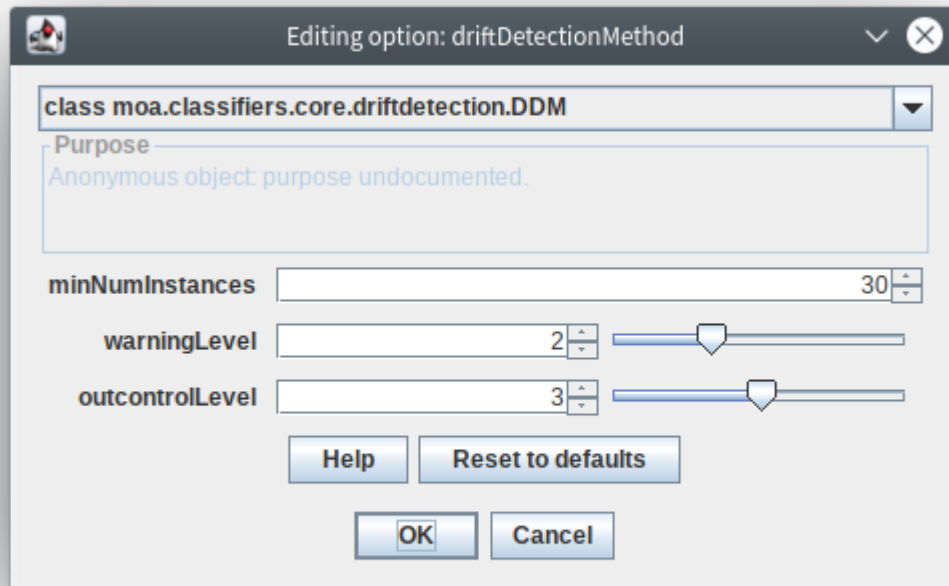


Figura 2.4 MOA - Configuração detector

Os principais algoritmos para detecção de mudanças de conceito possuem implementação disponível no MOA. Além dessas implementações, a ferramenta também disponibiliza rotinas de avaliação, que analisam a acurácia e o desempenho desses métodos.

Nesta dissertação, o MOA foi utilizado para realização do experimento com dados sintéticos, gerados a partir da própria ferramenta. A técnica proposta foi implementada no MOA e comparada aos principais algoritmos disponíveis. Por fim, os testes realizados foram avaliados através da classe *BasicConceptDriftPerformanceEvaluator*, disponibilizada pela ferramenta.

2.4 REDES DE FUNÇÃO DE BASE RADIAL

Uma Rede de Função de Base Radial (redes RBF, em português, ou *radial basis function network*, *RBF network*, em inglês) pode ser definida como um modelo de múltiplas camadas alimentadas adiante (*feedforward*), capaz de analisar padrões complexos e resolver problemas não-linearmente separáveis, utilizando uma abordagem de aproximação de funções. Estas redes têm como principal diferencial a sua forma de ativação, realizada

através do cálculo da distância entre o dado e um centro definido (BRAGA; CARVALHO; LUDERMIR, 2007).

A arquitetura de uma rede de função de base radial, em sua forma mais básica, envolve três camadas. A camada de entrada representa os atributos do problema. As unidades dessa camada não realizam processamento e simplesmente distribuem as entradas para os neurônios na camada intermediária. A camada intermediária, única camada oculta da rede, é constituída por funções de ativação de base radial, que atuam como os neurônios da rede. Por fim, a camada de saída pondera, através de pesos, os resultados da camada intermediária, agregando-os linearmente para compor a resposta final da rede (ROJAS, 1996). A Figura 2.5 demonstra essa arquitetura.

Na literatura, as funções Gaussianas (Função 2.4) são as funções de ativação mais usuais em redes RBF, sendo definidas por seus centros e larguras:

$$\varphi(v_i) = e^{-(\sigma r)^2} \quad (2.4)$$

onde, r é a distância euclidiana ($\|\mathbf{v}_i - \mathbf{c}_i\|$), v é o valor de entrada, c_i representa o centro e σ é o parâmetro limitador do raio.

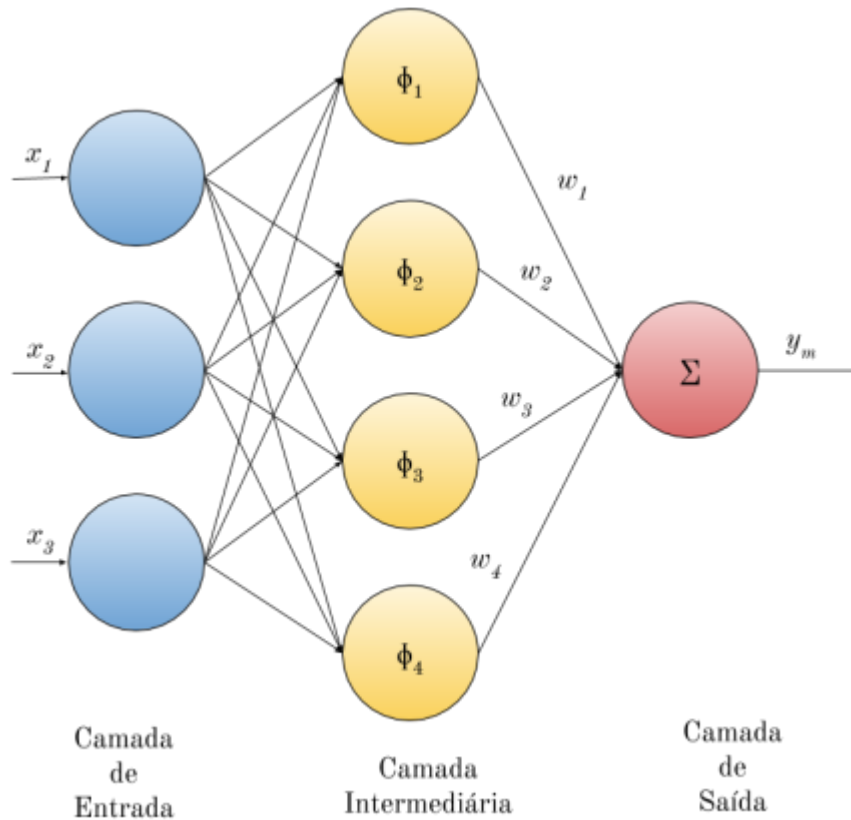


Figura 2.5 Arquitetura RBF

A modelagem das redes RBF define a entrada como um vetor de números reais $x \in R^n$ e o resultado da rede como uma função escalar desse vetor $\varphi : R^n \rightarrow R$. Esta função

é obtida a partir da combinação linear dos resultados das funções de ativação (Equação 2.5).

$$\varphi(x) = \sum_{i=1}^N a_i \rho(\|x - c_i\|) \quad (2.5)$$

onde N é o número de neurônios na camada intermediária, c_i é o centro do neurônio i , $\|\dots\|$ é a norma euclidiana e a_i é o peso (bias) atribuído ao neurônio para realização da combinação linear.

A técnica proposta nesta dissertação utiliza uma rede RBF adaptada. Na versão adaptada, apenas as camadas inicial e intermediária são aplicadas, e a Gaussiana é utilizada como função de ativação. Esta adaptação é possível, pois a camada intermediária cria, implicitamente, durante o processo de ativação, grupos com os dados recebidos ao longo do tempo. Dessa forma, quando o centro ativo do cluster formado é alterado, uma possível mudança de conceito é sinalizada.

Além da rede RBF adaptada, o método proposto também utiliza uma Cadeia de Markov para modelar as transições entre centros e obter tolerância a ruídos. As Cadeias de Markov serão apresentadas na próxima seção.

2.5 CADEIAS DE MARKOV

Equações determinísticas podem ser utilizadas para descrever sistemas que permitem traçar toda a sua evolução a partir da condição inicial. Contudo, em alguns sistemas isso não é possível, pois estes apresentam diferentes possibilidades evolutivas. Nestes casos, processos estocásticos são utilizados (TAYLOR; KARLIN; TAYLOR, 1998).

Um processo estocástico $\{X_t : t \in T\}$ é uma coleção de variáveis aleatórias indexadas no tempo. O conjunto de possíveis valores de X_t , $S = \{s_1, s_2, \dots\}$, é denominado espaço dos estados e T denota o espaço temporal. A partir desta definição e considerando que o processo estocástico esteja no estado s_i e no tempo $t - 1$, a probabilidade do processo estar no estado s_j no tempo t é dada pela Equação 2.6:

$$\mathbb{P}(X_t = s_j | X_{t-1} = s_i, \dots, X_0 = s_0) \quad (2.6)$$

A probabilidade resultante desta equação é denominada probabilidade de transição (de um passo) do estado s_i para o estado s_j .

Uma Cadeia de Markov, ou Processo de Markov, é um processo estocástico no qual a probabilidade de um sistema estar em determinado estado em um dado período de tempo depende apenas do estado no período imediatamente anterior. Esse tipo de processo também é conhecido como *memoryless process* (processo sem memória), pois os estados passados são ignorados. Esta característica é demonstrada através da Equação 2.7:

$$\mathbb{P}(X_t = s_j | X_{t-1} = s_i, \dots, X_0 = s_0) = \mathbb{P}(X_t = s_j | X_{t-1} = s_i) = p_{ij} \quad (2.7)$$

Isto é, um processo de Markov pode assumir os estados a_1, a_2, \dots, a_r , de tal modo que a probabilidade de transição de um estado a_i para um estado a_j seja P_{ij} (um valor dependente apenas de i e j). Dessa forma, é possível constituir uma matriz com todas

probabilidades de transição entre os estados. Esta matriz é denominada como matriz estocástica e um exemplo é apresentado na Equação 2.8:

$$P = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1r} \\ P_{21} & P_{22} & \dots & P_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ P_{r1} & P_{r2} & \dots & P_{rr} \end{bmatrix} \quad (2.8)$$

A fim de ilustrar os conceitos apresentados, considere o diagrama de estados representado na Figura 2.6. Este diagrama detalha um processo Markoviano com três estados possíveis, em um determinado instante de tempo:

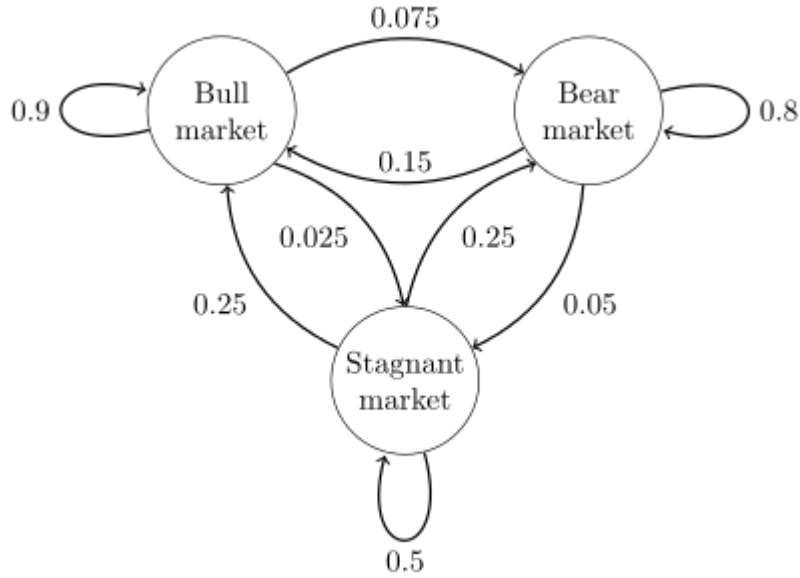


Figura 2.6 Exemplo: Cadeia de Markov com três estados. Fonte: Wikipedia (2019).

Este diagrama pode ser representado como uma matriz estocástica, demonstrada na Equação 2.9.

$$P = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}. \quad (2.9)$$

Neste trabalho, os estados da Cadeia de Markov são formados pelos centros ativados na rede RBF adaptada. As transições entre centros (ativação) são refletidas no modelo markoviano, aumentando a probabilidade da transição vigente por um valor parametrizável (α) e diminuindo proporcionalmente a probabilidade das outras transições. Esta operação é realizada respeitando as seguintes condições:

- A probabilidade de uma transição deve ser sempre maior ou igual a zero ($P_{ij} \geq 0$);
- A soma das probabilidades de todas transições a partir de um determinado estado deve ser sempre igual a um ($\sum_1^j P_{ij} = 1$).

Além de modelar as mudanças de conceito ocorridas, a Cadeia de Markov também é responsável pela tolerância a ruídos. Isto ocorre, pois apenas transições cuja probabilidade $P_{ij} \geq \delta$, onde δ é um valor parametrizável, podem sinalizar mudanças de conceito.

2.6 MONITORAMENTO OCULAR - FIXAÇÕES E SACADAS

A análise rigorosa dos movimentos oculares não é uma técnica experimental recente. Entretanto, o desenvolvimento de tecnologias não invasivas de monitoramento favoreceu o aumento da sua utilização. Atualmente, a técnica é empregada em diferentes tarefas experimentais (DUCHOWSKI, 2002), como, por exemplo: localização da atenção visual, detecção de deficiências cognitivas, avaliação de estratégias de pesquisa visual, entre outras. Estas tarefas produzem informações que são utilizadas como fator de análise por um vasta gama de pesquisas, em diferentes áreas do conhecimento.

Antes de descrever a atividade de monitoramento ocular, é necessário apresentar alguns conceitos básicos sobre o funcionamento da visão. Os olhos precisam estar em constante movimento para que a retina possa formar uma imagem nítida. Este movimento frequente é conhecido como movimento sacádico. Ao analisar uma imagem ou cena, o olho tende a permanecer fixo durante alguns milissegundos nas áreas mais significativas (fixações); depois disso, o olho se move em direção a uma nova zona de interesse (sacadas). Este processo é demonstrado na Figura 2.7, utilizando pontos mais densos para representar as fixações e linhas para as sacadas.



Figura 2.7 Exemplo de identificação de fixações e sacadas. Fonte: König e Buffalo (2014).

Durante as sacadas, o cérebro não tem tempo hábil para processar as imagens transmitidas pelo sistema visual, causando uma deterioração da imagem formada na retina.

Por esse motivo, o cérebro só atribui semântica às imagens formadas durante os períodos de fixação. Isso implica que o estudo das fixações é essencial para a compreensão e interpretação dos movimentos oculares.

Portanto, é fundamental utilizar técnicas automatizadas para segmentar fixações e sacadas durante a análise de trajetórias oculares. A literatura dispõe de diversos algoritmos para realização desta tarefa. As técnicas mais utilizadas realizam a segmentação com base em limites de velocidade e/ou aceleração, pois estes valores são significativamente maiores durante as sacadas (OTERO-MILLAN et al., 2008). Enquanto outros algoritmos realizam a segmentação através de características como densidade, dispersão ou áreas de interesse.

Contudo, a aplicação de diferentes algoritmos em uma mesma trajetória pode produzir resultados divergentes. Isto ocorre, pois a maioria dos métodos disponíveis na literatura é afetada pela escolha dos parâmetros e pelas características do experimento em análise (frequência, amplitude, etc.). Consequentemente, a atividade de identificação de fixações e sacadas permanece altamente subjetiva.

Visando mitigar este problema, König e Buffalo (2014) propõem um novo método, denominado ClusterFix. Este método se diferencia por ser completamente automatizado e independente de parâmetros. Essas características permitem que a técnica seja utilizada como um padrão referencial, possibilitando a comparação da segmentação de trajetórias oculares entre experimentos, laboratórios e algoritmos.

Nesta dissertação, o método desenvolvido para identificação de mudanças de conceito em fluxos de dados é adaptado para segmentação de fixações e sacadas. A técnica adaptada foi empregada em dois conjuntos de dados, oriundos do monitoramento ocular de dois símios. Para validar a aplicabilidade da técnica, os mesmos conjuntos de dados foram submetidos ao algoritmo ClusterFix e comparados. A configuração e os resultados obtidos neste experimento são detalhados no Capítulo 4.

2.7 TRABALHOS RELACIONADOS

Durante a elaboração desta dissertação, foi realizada uma pesquisa na literatura em busca de trabalhos que propusessem métodos para identificação de mudanças de conceito, de forma online e independente de rótulos, em fluxos contínuos de dados. Também foram estudadas técnicas que pudessem subsidiar o desenvolvimento de novos algoritmos que atendam a esses requisitos.

Inicialmente, foram analisados os algoritmos *Implícitos / Não Supervisionados* pertencentes a subcategoria *Deteção de Novidade / Métodos de Agrupamento*. Estes algoritmos são descritos a seguir.

O algoritmo *OnLine Novelty and Drift Detection Algorithm* (OLINDDA) utiliza a técnica de agrupamento *K-means* para monitorar e se adaptar ao surgimento de novos padrões (SPINOSA; CARVALHO; GAMA, 2007). Os exemplos recepcionados são incluídos em uma fila e são periodicamente agrupados. Os grupos resultantes desse processamento definem se os exemplos devem ser incorporados a um grupo existente ou formar um novo grupo.

O algoritmo MINAS, proposto por Faria, Gama e Carvalho (2013), atua de forma si-

milar ao algoritmo OLLINDA, contudo utiliza *micro clusters* obtidos a partir da aplicação de uma técnica de agrupamento específica para fluxos contínuos de dados (CluStream) e estende a abordagem para permitir a aplicação em problemas multiclasse.

O algoritmo DETECTNOD (Hayat; Hashemi, 2010) utiliza um modelo de clusteração para delimitar os exemplos observados. Uma representação compacta desses grupos é produzida através da Transformada Discreta de Cosseno. Esta representação é utilizada, em conjunto com o algoritmo de vizinho mais próximo, para aproximar novos exemplos. Estes exemplos são categorizados como desvios ou novidades, conforme o grau de similaridade calculado.

Os algoritmos Woo (Lee; Wang; Ryu, 2007) e ECSMiner (MASUD et al., 2011) também se baseiam no conceito de *micro clusters*. A técnica Woo atribui um classificador para cada cluster formado. Estes classificadores são utilizados para verificar a similaridade de novos exemplos em relação ao cluster. Exemplos que não se enquadram em nenhum grupo são marcados como suspeitos e passam a ter sua densidade monitorada. Um número crescente de vizinhos no raio dos exemplos monitorados indica o surgimento de um novo conceito, induzindo o retreino do modelo e ajuste do centróide do agrupamento. O ECSMiner (MASUD et al., 2011) utiliza o conceito de Outliers Filtrados, que se refere a amostras que estão fora do limite de todos os clusters existentes, em uma abordagem similar ao algoritmo Woo.

O algoritmo GC3 (SETHI; KANTARDZIC; HU, 2016) estende a ideia de *micro clusters* ao aplicar um algoritmo de agrupamento baseado em grade e densidade. Dessa forma, novos padrões são identificados a partir do surgimento de novas áreas de alta densidade.

Os algoritmos baseados em técnicas de detecção de novidade utilizam métodos de agrupamento para identificar padrões emergentes. Portanto, também apresentam dificuldade para lidar com alta dimensionalidade e/ou dados binários, por dependerem do cálculo de distância, e são computacionalmente custosos.

A segunda etapa da pesquisa analisou técnicas para detecção de mudanças de conceito em séries temporais que atuam de forma online. Estas técnicas também são comumente denominadas como técnicas para detecção de *change points*. A maioria dos trabalhos identificados se baseia no monitoramento dos valores residuais da aplicação de um modelo. Nesta abordagem, um modelo estatístico ou de Aprendizado de Máquina é aplicado a um conjunto de observações extraídas da série temporal, representando um conceito. Assume-se que se não houver mudança de conceito, os valores residuais dessa aplicação devem ser estacionários. Os trabalhos identificados são apresentados a seguir.

Yamanishi e Takeuchi (2002) propõem um método de detecção online capaz de identificar outliers e mudanças de conceito em séries temporais. A abordagem aplica um modelo autoregressivo aos dados e atualiza os parâmetros de forma incremental, para que o peso de exemplos passados diminua ao longo do tempo. Atribui-se uma pontuação para cada observação, calculada com base na função de perda. Esta pontuação indica o grau de desvio do dado em relação ao modelo aplicado. Pontuações elevadas indicam alta possibilidade de que o dado seja um outlier. A detecção de mudança de conceito é feita através do monitoramento da média das funções de perda ao longo da série temporal. Todavia, esta abordagem só é aplicável a séries temporais com comportamento autoregressivo, o que limita o seu escopo de uso.

Gombay e Serban (2009) propõem um método diferente, também online, para detecção de mudanças em séries temporais autoregressivas, baseado no monitoramento dos parâmetros do modelo. Uma adaptação do método CUSUM (PAGE, 1954) é aplicada para identificar alterações nos parâmetros. A hipótese nula do algoritmo assume que estes parâmetros devem permanecer estacionários ao longo do tempo. Esta hipótese é testada para cada nova observação recebida e a mudança de conceito é sinalizada quando a diferença ultrapassa um limite estabelecido. Esta abordagem também apresenta limitação de escopo, pois é aplicável apenas a séries temporais autoregressivas.

A utilização de valores residuais para detecção de mudanças de conceito apresenta desvantagens. Por serem baseados na acurácia do modelo aplicado, estes métodos sofrem influência de problemas ocorridos durante a parametrização ou fase de treinamento. Em teoria, o monitoramento direto de características dos dados pode permitir uma identificação de mudanças de conceito mais robusta.

Poucos estudos utilizam as características das séries temporais para identificar mudanças de conceito. Neste contexto, o trabalho proposto por Boracchi e Roveri (2014) se destaca por apresentar um método de detecção de mudanças de conceito em séries temporais que atua de forma online, baseando-se no monitoramento da característica de auto-similaridade da série. Entretanto, o método proposto tem como principal limitação o fato de ser especializado para séries que apresentam auto-similaridade e periodicidade. Além disso, a abordagem utiliza uma considerável quantidade de memória, pois armazena uma sequência de dados que representa o comportamento estável ou regular da série temporal.

Para cenários nos quais os fluxos de dados não apresentam as propriedades esperadas pelas técnicas discutidas anteriormente, Costa e Mello (2014) propõem um novo método para detecção de mudanças de conceito que atua de forma online. Este trabalho se diferencia por utilizar um algoritmo de agrupamento hierárquico estável, conforme a propriedade proposta por Carlsson e Mémoli (2010). A abordagem cria janelas consecutivas de dados. A técnica de agrupamento é aplicada em cada janela e os dendogramas resultantes comparados, a fim de identificar mudanças de conceito na distribuição dos dados.

As técnicas para detecção de mudanças de conceito são, geralmente, aplicadas para minimizar o intervalo entre a ocorrência da mudança e a atualização do modelo aplicado. Diante disso, um método de detecção ideal deve:

- ser transparente para o usuário, detectando e informando explicitamente sobre a ocorrência de mudanças;
- ser computacionalmente eficiente;
- atuar de forma online, buscando minimizar o atraso de atualização do modelo; e
- basear-se em características dos dados que reflitam os conceitos presentes.

Por fim, a pesquisa realizada evidenciou a inexistência de técnicas que atendam a todos esses requisitos e que sejam computacionalmente eficientes, independentes de rótulos e

aplicáveis a fluxos de dados de qualquer natureza. Esta dissertação de mestrado tem como objetivo preencher esta lacuna, através da proposição de um novo método, baseado em Redes de Função de Base Radial e Cadeias de Markov, para identificação de mudanças de conceito em fluxos contínuos de dados.

2.8 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os principais conceitos utilizados nesta dissertação. A relação entre Fluxos Contínuos de Dados e técnicas de Aprendizado de Máquina foi abordada, o fenômeno Mudança de Conceito e os métodos de detecção foram detalhados, as Redes de Função de Base Radial e as Cadeias de Markov foram retratadas, e, por fim, o problema de classificação em sacadas e fixações durante atividades de monitoramento ocular foi discutido.

Também foram apresentados trabalhos que se relacionam com a proposta desta dissertação, permitindo notar que até o presente momento inexistem métodos para detecção de mudanças de conceito que possam ser aplicados a fluxos contínuos de dados de forma satisfatória. Diante disso, o presente trabalho demonstra relevância, ao propor um novo método capaz de identificar mudanças de conceito em fluxos de dados em tempo de execução, de forma computacionalmente eficiente, com tolerância a ruídos e sem depender de rótulos.

RBFChain**3.1 CONSIDERAÇÕES INICIAIS**

Este capítulo apresenta o RBFChain, que é o método de detecção de mudanças de conceito proposto nesta dissertação. A proposta é apresentada da seguinte forma: a Seção 3.2 inicia o capítulo com uma visão geral, identificando a arquitetura e a interação dos componentes. Na Seção 3.3, o algoritmo desenvolvido é demonstrado em pseudocódigo e o seu funcionamento é detalhado. Por fim, na Seção 3.4, são discutidas as considerações finais do capítulo.

3.2 VISÃO GERAL

A arquitetura do RBFChain é composta por dois componentes principais: uma Rede de Função de Base Radial e uma Cadeia de Markov. A rede RBF é responsável por agrupar os eventos recebidos, formando um cluster, cujo centro ativo é alterado conforme o evento em análise. As mudanças no centro ativo do cluster são modeladas através da Cadeia de Markov. No modelo Markoviano, os centros ativados formam os estados e as mudanças de centro, as transições. Estes componentes são representados graficamente na Figura 3.1. O restante desta seção discute o papel desempenhado por cada componente.

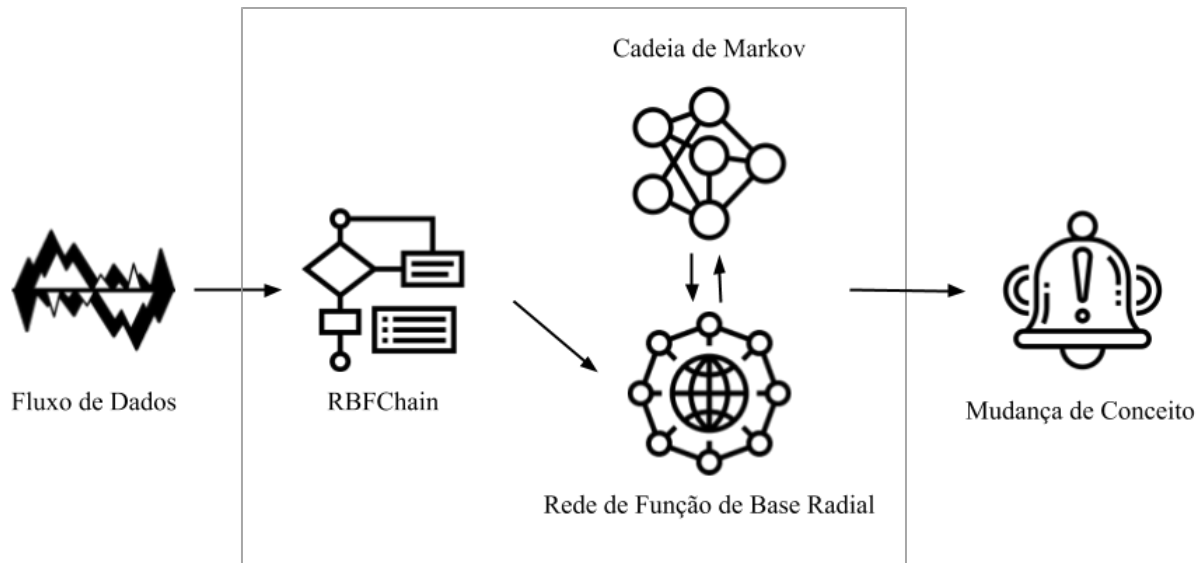


Figura 3.1 Arquitetura RBFChain

Antes de iniciar a descrição da arquitetura e do papel desempenhado por cada componente, é importante destacar que o método proposto requer a definição de quatro parâmetros para o seu funcionamento. Sendo dois relacionados a Rede de Função de Base Radial e dois a Cadeia de Markov:

- **Parâmetros relacionados a Rede de Função de Base Radial:**

- σ : limita o raio da função Gaussiana utilizada durante o processo de ativação;
- λ : estabelece um valor mínimo para ativação de um centro;

- **Parâmetros relacionados a Cadeia de Markov:**

- α : fator de incremento da probabilidade de transições ocorridas;
- δ : probabilidade para considerar uma transição uma mudança de conceito.

O RBFChain atua diretamente sobre o fluxo de dados. Os eventos observados são imediatamente submetidos ao processo de ativação da rede RBF. Neste processo, o valor de ativação do evento é calculado para cada um dos centros da rede, através da função Gaussiana. Dessa forma, o evento é vinculado ao centro com maior valor de ativação, desde que esse valor seja maior que λ . Se o evento não formar vínculo com nenhum dos centros anteriores, o próprio evento passa a ser o novo centro ativo da rede.

A rede RBF mantém em memória a informação do último centro ativado, permitindo identificar quando há alteração do centro ativo. Essas mudanças são modeladas na Cadeia de Markov, que aumenta a probabilidade da transição ocorrida pelo fator α e diminui

proporcionalmente o valor das outras transições possíveis para o estado originário. Após esta operação, o algoritmo verifica se o centro ativo anterior é diferente do atual. Se esta condição for verdadeira e a probabilidade da transição for maior que δ , uma mudança de conceito é sinalizada. Contudo, se a condição for verdadeira mas o valor da probabilidade é inferior a δ , o algoritmo apenas sinaliza uma *zona de alerta* (*warning zone*).

3.3 RBFCHAIN

O método proposto é demonstrado na forma de pseudocódigo em Algoritmo 1. O restante desta seção detalha o funcionamento do algoritmo através de uma execução de exemplo.

Entrada: $evento, \sigma, \lambda, \alpha, \delta$

Saída: *nulo, alerta* ou *mudanca*

início

```

centros  $\leftarrow$  (); centroAtual  $\leftarrow$  null; centroAtivo  $\leftarrow$  null;
mudanca  $\leftarrow$  falso;

para todo centro faça
    ativacao  $\leftarrow$  gaussiana(evento, centro,  $\sigma$ )
    se ativacao  $\geq \lambda$  então
        centroAtivo  $\leftarrow$  centro;
         $\lambda \leftarrow$  ativacao;

se centroAtivo == nulo então
    centros  $\leftarrow$  evento;
    centroAtivo  $\leftarrow$  evento;

se centroAtual  $\neq$  centroAtivo então
    mudanca  $\leftarrow$  verdadeiro;

probabilidade  $\leftarrow$  markov(centroAtual, centroAtivo,  $\alpha$ );
centroAtual  $\leftarrow$  centroAtivo;

se mudanca == verdadeiro e probabilidade  $\geq \delta$  então
    retorna mudanca;
senão se mudanca == verdadeiro e probabilidade  $< \delta$  então
    retorna alerta;
senão
    retorna nulo;

```

Algoritmo 1: RBFCHAIN

Para exemplificar a execução do algoritmo proposto neste projeto, considere o conjunto $S = \{0.11, 0.12, 0.13, 0.34, 0.45, 0.47, 0.33, 0.25, 0.14, 0.10\}$ como fonte de dados. Para este exemplo, os parâmetros foram definidos de forma empírica. O parâmetro σ foi definido com o valor 0.2 e o parâmetro λ foi fixado em 0.6. A seguir, será descrito o comportamento do algoritmo para cada dado recebido a partir de S .

No instante $T1$, o valor 0.11 é recepcionado. Como não existem centros estabelecidos, o valor é definido como centro e ativado. Em $T2$ e $T3$ são recebidos, respectivamente, os valores 0.12 e 0.13 que são imediatamente vinculados ao centro atualmente ativo (0.11), pois seus valores de ativação foram maiores que 0.6 (λ).

No instante $T4$, o valor 0.34 não atinge o valor mínimo de ativação (λ) para ser vinculado ao centro ativo (0.11). Dessa forma, o valor é definido como um novo centro e ativado. Devido a alteração do centro ativo, o algoritmo sinaliza a ocorrência de mudança de conceito.

Os valores dos instantes $T5$, $T6$, $T7$ e $T8$ são vinculados ao último centro ativo (0.34). Contudo, em $T9$, o valor 0.14 é recebido e apresenta maior valor de ativação para o centro 0.11, que é reativado. Assim, uma nova mudança de conceito é sinalizada. Finalmente, o valor do instante $T10$, 0.10, é vinculado ao centro ativo (0.11).

A Figura 3.2 apresenta esse comportamento graficamente. Os círculos na cor amarela representam valores vinculados ao primeiro centro estabelecido (0.11 no momento $T1$), enquanto os de cor lilás foram ativados pelo segundo centro definido (0.34 em $T4$). As linhas verticais tracejadas, de cor vermelha, indicam a ocorrência de mudança de conceito.

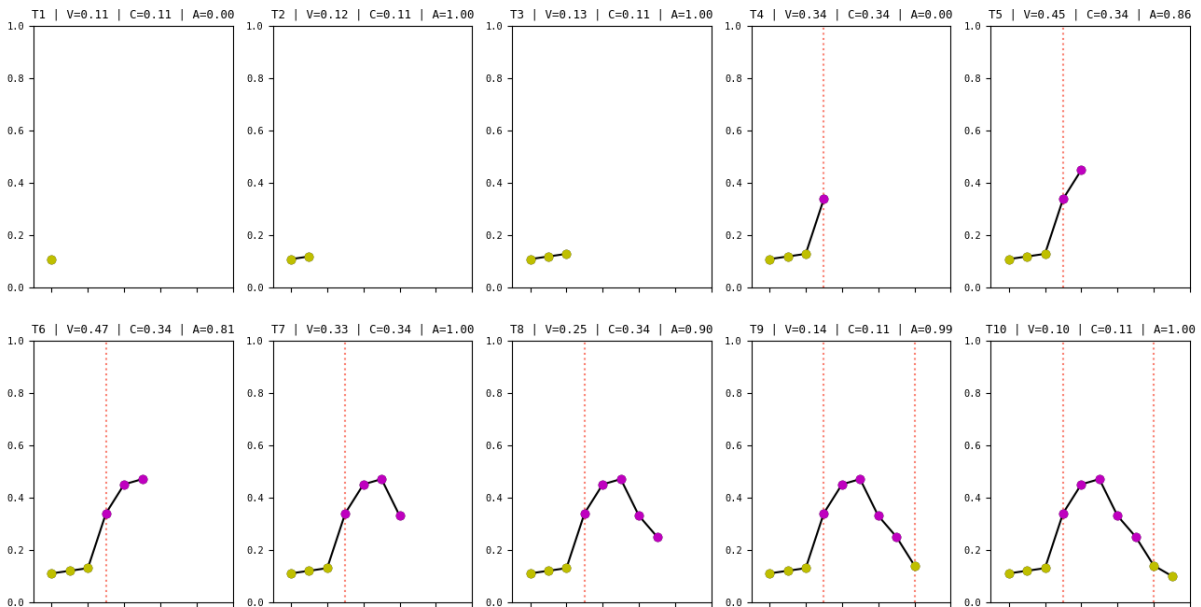


Figura 3.2 Exemplo de funcionamento do algoritmo

3.4 CONSIDERAÇÕES FINAIS

TODO

EXPERIMENTOS E RESULTADOS

4.1 CONSIDERAÇÕES INICIAIS

Este capítulo apresenta um conjunto de experimentos preliminares realizados com dados sintéticos, cujo objetivo foi demonstrar a viabilidade da proposta de mestrado. Os resultados obtidos se mostraram promissores, indicando que o tema de pesquisa deve continuar a ser investigado. A próxima seção apresenta como os dados sintéticos foram produzidos para condução dos experimentos.

4.2 CONFIGURAÇÃO DOS EXPERIMENTOS

Os fluxos de dados sintéticos foram produzidos através de classes geradoras do framework MOA e salvos em arquivos no formato ARFF. Cada fluxo produzido representa até dois conceitos e é composto por 2.500 observações, com valores entre 0 e 1. O restante desta seção apresenta as classes geradoras utilizadas, as modificações realizadas e os parâmetros aplicados.

A classe geradora *AbruptChangeGenerator* produz fluxos de dados sintéticos com mudanças de conceito abruptas. Os fluxos gerados simulam as mudanças através da intercalação de sequências com o valor 0.2 e sequências com o valor 0.8, conforme o tamanho definido para os conceitos. Com o objetivo de tornar os dados produzidos mais próximos da realidade, essa classe foi alterada¹ para permitir a adição de um ruído randômico, com amplitude configurável, para cada exemplo produzido. A versão com suporte a ruído foi utilizada nos experimentos para produção do fluxo com mudanças abruptas. O gerador foi parametrizado para produzir exemplos não binários, com conceitos formados com 400 instâncias e ruído limitado ao intervalo $[-0.1, 0.2]$ seguindo uma distribuição uniforme.

A classe *GradualChangeGenerator* produz fluxos sintéticos com mudanças de conceito graduais. Durante sua configuração, percebeu-se uma limitação, pois a classe só gera uma mudança gradual, independente do tamanho dos conceitos. Para superar esta limitação, a

¹<https://git.io/fjGEj>

classe foi modificada² para gerar uma quantidade de mudanças coerente com o tamanho definido. A classe modificada foi utilizada para gerar o fluxo sintético com mudanças graduais utilizado nos experimentos, sendo parametrizada para produzir exemplos não binários e conceitos compostos por 400 instâncias.

Por fim, a classe *NoChangeGenerator*, que produz fluxos sem ocorrência de mudanças de conceito, também foi modificada³ para permitir a incidência de ruído nos resultados gerados. Esta versão foi utilizada para produzir um fluxo sintético sem mudanças, com ruído entre $[-0.1, 0.1]$.

4.3 CRITÉRIOS DE AVALIAÇÃO

A classe de avaliação *BasicConceptDriftPerformanceEvaluator*, pertencente ao framework MOA, foi utilizada para avaliar o método proposto neste trabalho. Esta classe permite mensurar a acurácia e o desempenho de algoritmos para detecção de mudanças de conceito. Para utilizá-la, é necessário construir uma tarefa do tipo *EvaluateConceptDrift* através da aba *Concept Drift*, presente na tela inicial da aplicação. Para condução dos experimentos deste trabalho, a tarefa *EvaluateConceptDrift* foi configurada conforme descrito na Tabela 4.1.

Tabela 4.1 Configuração da classe *BasicConceptDriftPerformanceEvaluator*

Parâmetro	Valor	Observação
learner	ChangeDetectorLearner	O algoritmo de detecção de mudanças de conceito a ser testado é definido no atributo <i>driftDetectionMethod</i> da classe <i>ChangeDetectorLearner</i> .
stream	ARFFFileStream	Caminho para um dos arquivos <i>ARFF</i> descrito na seção anterior. O atributo <i>classIndex</i> deve ser definido como 0, pois não existem rótulos nestes conjuntos de dados.
instanceLimit	-1	Desabilita o limite de instâncias a serem processadas.
timeLimit	-1	Desabilita o limite de tempo de execução.
sampleFrequency	1	Uma linha de resultado do avaliador deve ser gerada para cada instância processada.

²<https://git.io/fjGue>

³<https://git.io/fjGu3>

A classe de avaliação utilizada produz diversos indicadores referentes a acurácia e performance do algoritmo executado. A Tabela 4.2 apresenta os indicadores analisados neste trabalho.

Tabela 4.2 Indicadores analisados

Indicador	Observação
Tempo de Processamento	Tempo médio (seg.) de processamento por instância.
Mudanças Existentes	Quantidade de mudanças existentes.
Mudanças Detectadas	Quantidade de mudanças detectadas corretamente.
Falso-positivos	Quantidade de mudanças detectadas erroneamente.
Atraso de Detecção	Quantidade média de instâncias até a detecção.

4.4 Pettitt

Durante a revisão da literatura, investigou-se a aplicabilidade das técnicas para detecção de *changing points* em problemas de detecção de mudança de conceito. Para tanto, foi realizada a implementação, no framework MOA⁴, de um novo algoritmo baseado no método estatístico proposto por Pettitt (1979).

O método de Pettitt é um teste não paramétrico, no qual se verifica se duas amostras Y_1, \dots, Y_t e Y_{t+1}, \dots, Y_T são da mesma população. A estatística $U_{t,T}$ faz uma contagem do número de vezes que um membro da primeira amostra é maior que um membro da segunda. A estatística $U_{t,T}$ é obtida através da Equação 4.1:

$$U_{t,T} = U_{t-1,T} + \sum_{j=1}^T \text{sgn}(Y_t - Y_j) \quad (4.1)$$

Onde $t = 2, \dots, T$; $\text{sgn}(x) = 1$ para $x > 0$; $\text{sgn}(x) = 0$ para $x = 0$ e $\text{sgn}(x) = -1$ para $x < 0$. A estatística $U_{t,T}$ é então calculada para os valores de $1 \leq t \leq T$ e a estatística $k(t)$ do teste de Pettitt é o máximo valor absoluto de $U_{t,T}$.

Os testes foram executados com os conjuntos de dados sintéticos produzidos na Seção 4.2. Os resultados obtidos são apresentados a seguir, na Tabela 4.3.

Tabela 4.3 Resultados - Método de Pettitt

Conjunto de Dados	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
Sem mudanças	3.19	0	0	4	—
Mudanças Abruptas	2.09	6	3	2	132
Mudanças Incrementais	1.48	6	4	2	133

⁴<https://git.io/fj8xu>

A análise dos resultados demonstra que o algoritmo se mostrou propenso à produção de falsos-positivo e, em comparação aos outros algoritmos abordados neste trabalho, computacionalmente ineficiente.

Por não apresentar resultados competitivos em relação ao estado da arte, esta linha de pesquisa foi suspensa, permitindo que os esforços se voltassem para a análise da aplicação de Redes de Função de Base Radial para detecção de mudanças de conceito.

4.5 RBFChain

Esta seção apresenta um conjunto de experimentos realizados com o objetivo de validar o algoritmo desenvolvido nesta pesquisa, denominado RBFChain. Embora os experimentos ainda não sejam suficientes para comprovar a hipótese proposta, a análise empírica aqui apresentada evidencia a importância de analisar a aplicação de Redes de Função de Base Radial para detecção de mudanças de conceito em fluxos contínuos de dados.

Para execução dos experimentos, foram utilizados três conjuntos de dados sintéticos, conforme descrito na seção 4.2. Além disso, os seguintes algoritmos foram utilizados para comparação: CUSUM, PageHinkley e ADWIN. A Tabela 4.4 lista os parâmetros utilizados para cada algoritmo.

Tabela 4.4 Parâmetros utilizados para cada algoritmo

Algoritmo	Parâmetros
RBFChain	$\sigma = 2; \lambda = 0.5$
CUSUM	$MinNumInstances = 30; \delta = 0.005; \lambda = 50$
PageHinkley	$MinNumInstances = 30; \delta = 0.005; \lambda = 50; \alpha = 1$
ADWIN	$\delta = 0.002$

O primeiro experimento utilizou o fluxo de dados sintético sem mudanças de conceito, a fim de avaliar a tendência de produção de falsos-positivo. O resultado desta análise pode ser visto na Tabela 4.5.

Tabela 4.5 Experimento 1 - Fluxo sem mudanças de conceito

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFChain	0.22	0	0	0	—
CUSUM	0.31	0	0	0	—
PageHinkley	0.24	0	0	0	—
ADWIN	0.21	0	0	0	—

Como pode ser observado, todos algoritmos testados demonstraram tolerância a ruídos e não indicaram nenhum falso positivo. Quanto a performance, o algoritmo proposto obteve a segunda melhor média em tempo de processamento. Superado pelo algoritmo ADWIN, por uma pequena margem. O comportamento dos algoritmos e do conjunto de dados utilizado é apresentado graficamente na Figura 4.1.

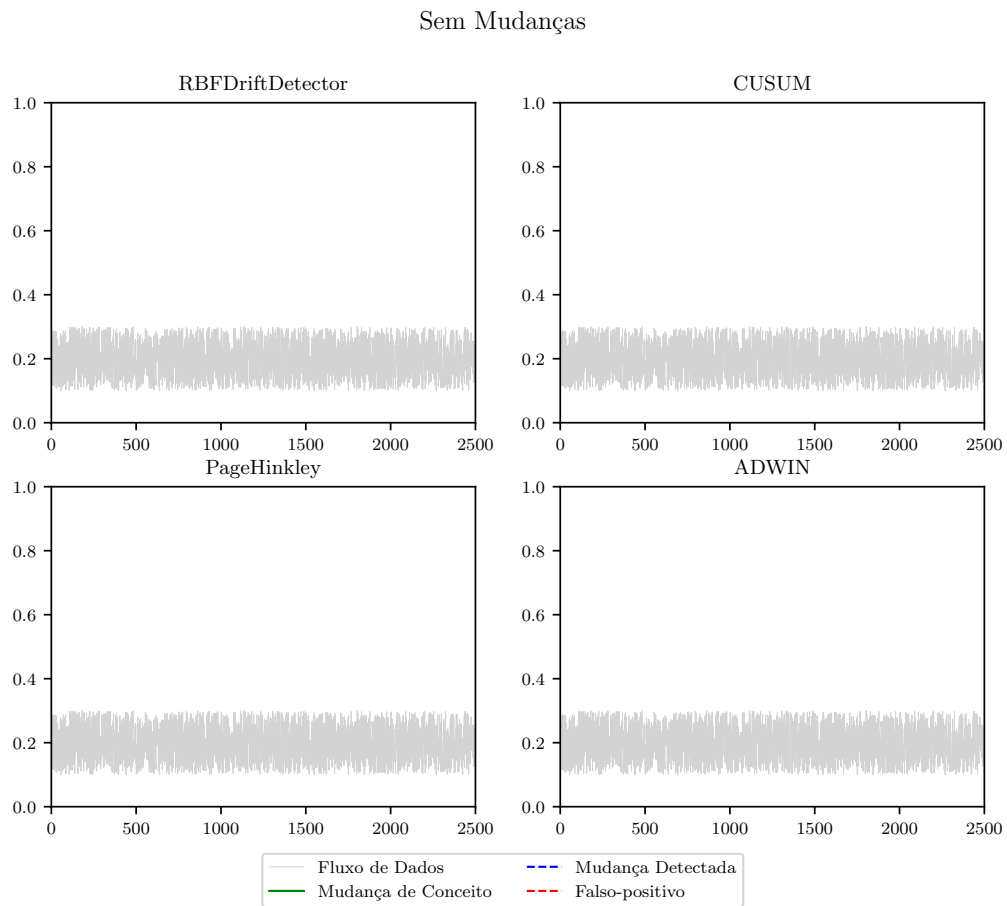


Figura 4.1 Representação Gráfica - Sem mudanças de conceito

Em seguida, utilizou-se a base de dados com mudanças abruptas como base de testes. Os resultados obtidos são apresentados na Tabela 4.6.

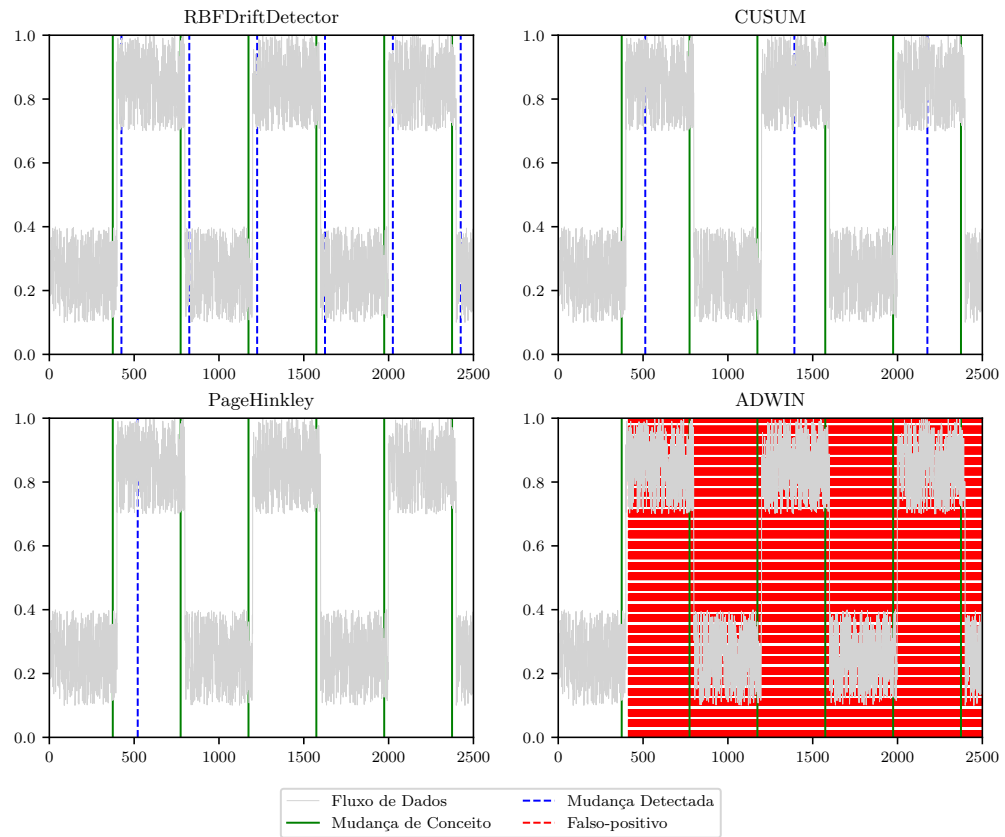
Tabela 4.6 Experimento 2 - Fluxo com mudanças de conceito abruptas

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFCChain	0.23	6	6	0	1
CUSUM	0.29	6	3	0	68
PageHinkley	0.22	6	1	0	17
ADWIN	0.21	6	2052	2046	9

Ao analisar este resultado, verifica-se que o algoritmo proposto neste trabalho identificou todas as mudanças de conceito existentes no fluxo sem indicar nenhum falso-positivo, sendo também o algoritmo com menor atraso de detecção. No quesito performance, a técnica desenvolvida obteve o terceiro melhor resultado. Paralelamente, os outros algoritmos analisados apresentaram baixa acurácia. O algoritmo CUSUM detectou apenas metade das mudanças e apresentou a maior taxa de atraso. Enquanto o método PageHinkley detectou apenas uma mudança. Por fim, o algoritmo ADWIN apresentou maior eficiência computacional e apesar de detectar as 6 mudanças existentes, mostrou-se hipersensível, pois foram detectados 2046 falsos-positivo.

Os resultados do segundo experimento são demonstrados graficamente na Figura 4.2. Nesta imagem, as linhas sólidas na cor verde representam as mudanças de conceito existentes no conjunto de dados analisado. As linhas tracejadas azuis representam mudanças de conceito identificadas corretamente, enquanto que as linhas tracejadas de cor vermelha representam falsos-positivo.

Mudanças Abruptas

**Figura 4.2** Representação Gráfica - Mudanças Abruptas

O último experimento realizado utilizou o fluxo sintético com mudanças de conceito graduais. Seu resultado é demonstrado na Tabela 4.7.

Tabela 4.7 Experimento 3 - Fluxo com mudanças de conceito graduais

Algoritmo	Tempo de processamento	Mudanças Existentes	Mudanças Detectadas	Falso-positivos	Atraso de Detecção
RBFChain	0.24	6	5	1	171
CUSUM	0.65	6	3	0	32
PageHinkley	0.26	6	1	0	4
ADWIN	0.27	6	2244	2238	1

Neste experimento, o algoritmo RBFChain identificou 5 das 6 mudanças existentes e sinalizou um falso-positivo. Ainda assim, obteve a melhor acurácia, apesar de apresentar

a maior taxa de atraso. Os outros algoritmos analisados apresentaram comportamentos similares aos do experimento anterior. O algoritmo CUSUM identificou metade das mudanças. O método PageHinkley identificou apenas uma alteração de conceito. E, finalmente, o algoritmo ADWIN voltou a se mostrar muito sensível, ao identificar 2238 falsos-positivo.

Por fim, o comportamento do conjunto de dados e dos algoritmos neste último experimento é apresentado na Figura 4.3. Na imagem, as linhas sólidas na cor verde representam as mudanças de conceito existentes no conjunto de dados analisado. As linhas tracejadas azuis representam mudanças de conceito identificadas corretamente, enquanto que as linhas tracejadas de cor vermelha representam falsos-positivo.

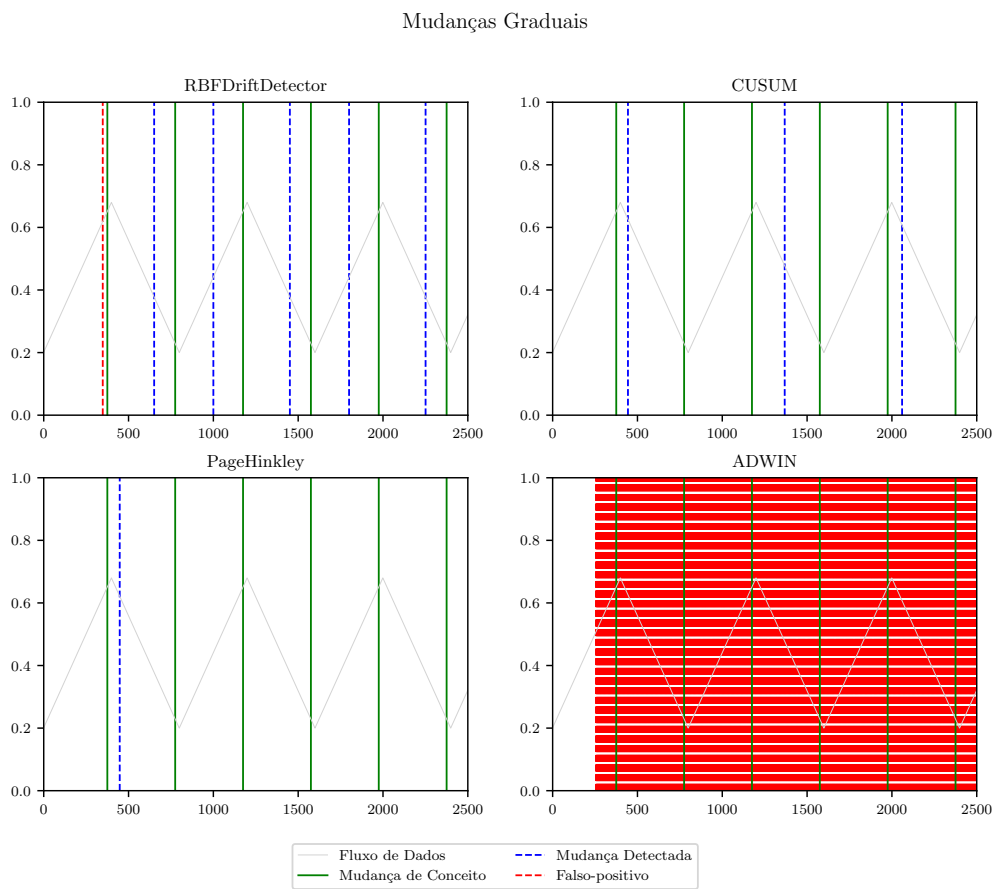


Figura 4.3 Representação Gráfica - Mudanças Graduais

4.6 CONSIDERAÇÕES FINAIS

...

CONCLUSÕES E TRABALHOS FUTUROS

5.1 DISCUSSÕES

...

Nesta seção foram apresentados os resultados dos experimentos iniciais deste trabalho de pesquisa, os quais visavam confirmar a aplicabilidade das Redes de Função de Base Radial na detecção de mudanças de conceito em fluxos contínuos de dados.

As seguintes etapas estão previstas para serem executadas: i) realizar experimentos com fluxos recorrentes e incrementais; ii) integrar uma cadeia de Markov ao método proposto, permitindo a análise do comportamento das mudanças e o aprimoramento da acurácia das detecções; iii) implementar os algoritmos OLINDDA, MINAS e DETECNOD para serem comparados; iv) implementar e validar o algoritmo no framework Tornado; e v) alterar o algoritmo para permitir que o centro se desloque dentro do grupo formado. Espera-se ainda realizar uma aplicação prática da abordagem proposta, utilizando-a em um conjunto de dados oriundo de um sistema do mundo real.

5.2 TRABALHOS FUTUROS

...

REFERÊNCIAS BIBLIOGRÁFICAS

- ACKERMANN, M. R. et al. Streamkm++: A clustering algorithm for data streams. *J. Exp. Algorithmics*, ACM, New York, NY, USA, v. 17, p. 2.4:2.1–2.4:2.30, maio 2012. ISSN 1084-6654. Disponível em: <http://doi.acm.org/10.1145/2133803.2184450>.
- AGGARWAL, C. C. *Data Streams: Models and Algorithms (Advances in Database Systems)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387287590.
- AGGARWAL, C. C. et al. A framework for clustering evolving data streams. In: *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29*. VLDB Endowment, 2003. (VLDB '03), p. 81–92. ISBN 0-12-722442-4. Disponível em: <http://dl.acm.org/citation.cfm?id=1315451.1315460>.
- AGGARWAL, C. C. et al. On demand classification of data streams. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2004. (KDD '04), p. 503–508. ISBN 1-58113-888-1. Disponível em: <http://doi.acm.org/10.1145/1014052.1014110>.
- AMINIKHANGHAHI, S.; COOK, D. J. A survey of methods for time series change point detection. *Knowl. Inf. Syst.*, Springer-Verlag New York, Inc., New York, NY, USA, v. 51, n. 2, p. 339–367, maio 2017. ISSN 0219-1377. Disponível em: <https://doi.org/10.1007/s10115-016-0987-z>.
- ANKERST, M. et al. Optics: Ordering points to identify the clustering structure. *SIGMOD Rec.*, ACM, New York, NY, USA, v. 28, n. 2, p. 49–60, jun. 1999. ISSN 0163-5808. Disponível em: <http://doi.acm.org/10.1145/304181.304187>.
- Bach, S. H.; Maloof, M. A. Paired learners for concept drift. In: *2008 Eighth IEEE International Conference on Data Mining*. [S.l.: s.n.], 2008. p. 23–32. ISSN 1550-4786.
- BAENA-GARCÍA, M. et al. Early drift detection method. In: *In Fourth International Workshop on Knowledge Discovery from Data Streams*. [S.l.: s.n.], 2006.
- BARBARÁ, D. Requirements for clustering data streams. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 3, n. 2, p. 23–27, jan. 2002. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/507515.507519>.
- BARROS, R. S. M. de et al. RDDM: reactive drift detection method. *Expert Syst. Appl.*, v. 90, p. 344–355, 2017.
- BASSEVILLE, M.; NIKIFOROV, I. V. *Detection of Abrupt Changes: Theory and Application*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993. ISBN 0-13-126780-9.

BIFET, A.; GAVALDÀ, R. Learning from time-changing data with adaptive windowing. In: *SDM*. SIAM, 2007. p. 443–448. ISBN 978-1-61197-277-1. Disponível em: <http://dblp.uni-trier.de/db/conf/sdm/sdm2007.html#BifetG07>.

BIFET, A. et al. Moa: Massive online analysis. *J. Mach. Learn. Res.*, JMLR.org, v. 11, p. 1601–1604, ago. 2010. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1756006.1859903>.

BIFET, A.; KIRKBY, R. Data stream mining a practical approach. Citeseer, 2009.

BIFET, A. et al. Efficient data stream classification via probabilistic adaptive windows. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2013. (SAC '13), p. 801–806. ISBN 978-1-4503-1656-9. Disponível em: <http://doi.acm.org/10.1145/2480362.2480516>.

BLANCO, I. I. F. et al. Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Trans. Knowl. Data Eng.*, v. 27, n. 3, p. 810–823, 2015.

BORACCHI, G.; ROVERI, M. Exploiting self-similarity for change detection. *2014 International Joint Conference on Neural Networks (IJCNN)*, p. 3339–3346, 2014.

BRAGA, A.; CARVALHO, A. C.; LUDERMIR, T. B. *Redes Neurais Artificiais: Teoria e aplicações*. LTC Editora, 2007. ISBN 9788521615644. Disponível em: <http://www.worldcat.org/isbn/9788521615644>.

BREIMAN, L. et al. *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

CAO, F. et al. Density-based clustering over an evolving data stream with noise. In: GHOSH, J. et al. (Ed.). *SDM*. SIAM, 2006. p. 328–339. ISBN 978-1-61197-276-4. Disponível em: <http://dblp.uni-trier.de/db/conf/sdm/sdm2006.html#CaoEQZ06>.

CARLSSON, G.; MÉMOLI, F. Characterization, stability and convergence of hierarchical clustering methods. *J. Mach. Learn. Res.*, JMLR.org, v. 11, p. 1425–1470, ago. 2010. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1756006.1859898>.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 41, n. 3, p. 15:1–15:58, jul. 2009. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/1541880.1541882>.

COHEN, J. et al. Mad skills: New analysis practices for big data. *Proc. VLDB Endow.*, VLDB Endowment, v. 2, n. 2, p. 1481–1492, ago. 2009. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/1687553.1687576>.

Costa, F. G. d.; Mello, R. F. d. A stable and online approach to detect concept drift in data streams. In: *2014 Brazilian Conference on Intelligent Systems*. [S.l.: s.n.], 2014. p. 330–335.

DELATTRE, M.; IMBERT, B. *Method for management of data stream exchanges in an autonomic telecommunications network*. [S.l.]: Google Patents, 2015. US Patent 8,949,412.

Ditzler, G.; Polikar, R. Hellinger distance based drift detection for nonstationary environments. In: *2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE)*. [S.l.: s.n.], 2011. p. 41–48.

DOMINGOS, P.; HULTEN, G. Mining high-speed data streams. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2000. (KDD '00), p. 71–80. ISBN 1-58113-233-6. Disponível em: <http://doi.acm.org/10.1145/347090.347107>.

DREDZE, M.; OATES, T.; PIATKO, C. We're not in kansas anymore: Detecting domain changes in streams. In: . [s.n.], 2010. p. 585–595. Cited By 13. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-80053285630&partnerID=40&md5=2eea89f635e2cbc0920069028e9f7746>.

DRIES, A.; RÜCKERT, U. Adaptive concept drift detection. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, v. 2, n. 5-6, p. 311–327, 2009. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sam.10054>.

DUCHOWSKI, A. A breadth-first survey of eye tracking applications. *behavior res meth instrum comput (brmic)*. *Behavior research methods, instruments, & computers : a journal of the Psychonomic Society, Inc*, v. 34, p. 455–70, 12 2002.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification (2Nd Edition)*. New York, NY, USA: Wiley-Interscience, 2000. ISBN 0471056693.

ESTER, M. et al. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996. (KDD'96), p. 226–231. Disponível em: <http://dl.acm.org/citation.cfm?id=3001460.3001507>.

FARIA, E. R.; GAMA, J. a.; CARVALHO, A. C. P. L. F. Novelty detection algorithm for data streams multi-class problems. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2013. (SAC '13), p. 795–800. ISBN 978-1-4503-1656-9. Disponível em: <http://doi.acm.org/10.1145/2480362.2480515>.

GAMA, J. *Knowledge Discovery from Data Streams*. 1st. ed. [S.l.]: Chapman & Hall/-CRC, 2010. ISBN 1439826110, 9781439826119.

GAMA, J.; GABER, M. M. *Learning from Data Streams: Processing Techniques in Sensor Networks*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 3642092853, 9783642092855.

GAMA, J. et al. Learning with drift detection. In: BAZZAN, A. L. C.; LABIDI, S. (Ed.). *SBIA*. Springer, 2004. (Lecture Notes in Computer Science, v. 3171), p. 286–295. ISBN 3-540-23237-0. Disponível em: <http://dblp.uni-trier.de/db/conf/sbia/sbia2004.html#GamaMCR04>.

GAMA, J. a.; ROCHA, R.; MEDAS, P. Accurate decision trees for mining high-speed data streams. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2003. (KDD '03), p. 523–528. ISBN 1-58113-737-0. Disponível em: <http://doi.acm.org/10.1145/956750.956813>.

GAMA, J. a. et al. A survey on concept drift adaptation. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 46, n. 4, p. 44:1–44:37, mar. 2014. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/2523813>.

GOMBAY, E.; SERBAN, D. Monitoring parameter change in ar(p) time series models. *Journal of Multivariate Analysis*, v. 100, n. 4, p. 715 – 725, 2009. ISSN 0047-259X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0047259X08001826>.

GONÇALVES, P. M. et al. A comparative study on concept drift detectors. *Expert Systems with Applications*, v. 41, n. 18, p. 8144 – 8156, 2014. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417414004175>.

HALL, M. et al. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1656274.1656278>.

Hayat, M. Z.; Hashemi, M. R. A dct based approach for detecting novelty and concept drift in data streams. In: *2010 International Conference of Soft Computing and Pattern Recognition*. [S.l.: s.n.], 2010. p. 373–378.

JAIN, A. K.; DUBES, R. C. *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN 0-13-022278-X.

JIN, R.; AGRAWAL, G. Efficient decision tree construction on streaming data. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2003. (KDD '03), p. 571–576. ISBN 1-58113-737-0. Disponível em: <http://doi.acm.org/10.1145/956750.956821>.

KAUFMAN, L.; ROUSSEEUW, P. *Finding Groups in Data: an introduction to cluster analysis*. [S.l.]: Wiley, 1990.

KENKRE, P. S.; PAI, A.; COLACO, L. Real time intrusion detection and prevention system. In: SATAPATHY, S. C. et al. (Ed.). *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*. Cham: Springer International Publishing, 2015. p. 405–411. ISBN 978-3-319-11933-5.

KOLTER, J. Z.; MALOOF, M. A. Dynamic weighted majority: An ensemble method for drifting concepts. *J. Mach. Learn. Res.*, JMLR.org, v. 8, p. 2755–2790, dez. 2007. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1314498.1390333>.

KOTSIANTIS, S. B. Supervised machine learning: A review of classification techniques. In: *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2007. p. 3–24. ISBN 978-1-58603-780-2. Disponível em: <http://dl.acm.org/citation.cfm?id=1566770.1566773>.

KRANEN, P. et al. The clustree: Indexing micro-clusters for anytime stream mining. *Knowl. Inf. Syst.*, Springer-Verlag New York, Inc., New York, NY, USA, v. 29, n. 2, p. 249–272, nov. 2011. ISSN 0219-1377. Disponível em: <http://dx.doi.org/10.1007/s10115-010-0342-8>.

KRANJC, J. et al. Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the clowdflows platform. *Information Processing & Management*, v. 51, n. 2, p. 187 – 203, 2015. ISSN 0306-4573. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0306457314000296>.

KUNCHEVA, L. Classifier ensembles for detecting concept change in streaming data: Overview and perspectives. *Proc. Eur. Conf. Artif. Intell.*, p. 5–10, 2008. Cited By 70.

KÖNIG, S. D.; BUFFALO, E. A. A nonparametric method for detecting fixations and saccades using cluster analysis: Removing the need for arbitrary thresholds. *Journal of Neuroscience Methods*, v. 227, p. 121 – 131, 2014. ISSN 0165-0270. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0165027014000491>.

Lee, J.; Magoulès, F. Detection of concept drift for learning from stream data. In: *2012 IEEE 14th International Conference on High Performance Computing and Communication 2012 IEEE 9th International Conference on Embedded Software and Systems*. [S.l.: s.n.], 2012. p. 241–245.

Lee, Y. K.; Wang, L.; Ryu, K. H. A system architecture for monitoring sensor data stream. In: *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*. [S.l.: s.n.], 2007. p. 1026–1031.

LINDSTROM, P.; NAMEE, B. M.; DELANY, S. J. Drift detection using uncertainty distribution divergence. *Evolving Systems*, v. 4, n. 1, p. 13–25, Mar 2013. ISSN 1868-6486. Disponível em: <https://doi.org/10.1007/s12530-012-9061-6>.

LING, C.; LING-JUN, Z.; LI, T. Stream data classification using improved fisher discriminate analysis. *Journal of Computers*, 01 2009.

LLOYD, S. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, IEEE Press, Piscataway, NJ, USA, v. 28, n. 2, p. 129–137, set. 2006. ISSN 0018-9448. Disponível em: <http://dx.doi.org/10.1109/TIT.1982.1056489>.

MASUD, M. et al. Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Trans. on Knowl. and Data Eng.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 23, n. 6, p. 859–874, jun. 2011. ISSN 1041-4347. Disponível em: <http://dx.doi.org/10.1109/TKDE.2010.61>.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.

NISHIDA, K.; YAMAUCHI, K. Detecting concept drift using statistical testing. In: CORRUBLE, V.; TAKEDA, M.; SUZUKI, E. (Ed.). *Discovery Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 264–269. ISBN 978-3-540-75488-6.

OTERO-MILLAN, J. et al. Saccades and microsaccades during visual fixation, exploration, and search: Foundations for a common saccadic generator. *Journal of Vision*, v. 8, n. 14, p. 21–21, 12 2008. ISSN 1534-7362. Disponível em: <https://doi.org/10.1167/8.14.21>.

PAGE, E. S. Continuous Inspection Schemes. *Biometrika*, Biometrika Trust, v. 41, n. 1/2, p. 100–115, 1954. ISSN 00063444. Disponível em: <http://dx.doi.org/10.2307/2333009>.

PEARS, R.; SAKTHITHASAN, S.; KOH, Y. S. Detecting concept change in dynamic data streams - A sequential approach based on reservoir sampling. *Machine Learning*, v. 97, n. 3, p. 259–293, 2014.

PETTITT, A. A non-parametric approach to the change-point problem. *Journal of the Royal Statistical Society. Series C. Applied Statistics*, v. 28, 01 1979.

ROBERTS, S. W. Control chart tests based on geometric moving averages. *Technometrics*, American Society for Quality Control and American Statistical Association, Alexandria, Va, USA, v. 42, n. 1, p. 97–101, fev. 2000. ISSN 0040-1706. Disponível em: <http://dx.doi.org/10.2307/1271439>.

ROJAS, R. *Neural Networks: A Systematic Introduction*. Berlin, Heidelberg: Springer-Verlag, 1996. ISBN 3-540-60505-3.

ROSS, G. J. et al. Exponentially weighted moving average charts for detecting concept drift. *Pattern Recogn. Lett.*, Elsevier Science Inc., New York, NY, USA, v. 33, n. 2, p. 191–198, jan. 2012. ISSN 0167-8655. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2011.08.019>.

RYU, J. W. et al. An efficient method of building an ensemble of classifiers in streaming data. In: SRINIVASA, S.; BHATNAGAR, V. (Ed.). *Big Data Analytics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 122–133. ISBN 978-3-642-35542-4.

SCHLIMMER, J. C.; GRANGER, R. H. Incremental learning from noisy data. *Machine Learning*, v. 1, n. 3, p. 317–354, Sep 1986. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00116895>.

SEBASTIÃO, R. et al. Monitoring incremental histogram distribution for change detection in data streams. In: GABER, M. M. et al. (Ed.). *Knowledge Discovery from Sensor Data*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 25–42. ISBN 978-3-642-12519-5.

SETHI, T. S.; KANTARDZIC, M. On the reliable detection of concept drift from streaming unlabeled data. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 82, n. C, p. 77–99, out. 2017. ISSN 0957-4174. Disponível em: <https://doi.org/10.1016/j.eswa.2017.04.008>.

SETHI, T. S.; KANTARDZIC, M.; HU, H. A grid density based framework for classifying streaming data in the presence of concept drift. *Journal of Intelligent Information Systems*, v. 46, n. 1, p. 179–211, Feb 2016.

SPINOSA, E. J.; CARVALHO, A. P. de Leon F. de; GAMA, J. a. Olinda: A cluster-based approach for detecting novelty and concept drift in data streams. In: *Proceedings of the 2007 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2007. (SAC '07), p. 448–452. ISBN 1-59593-480-4. Disponível em: <http://doi.acm.org/10.1145/1244002.1244107>.

STREET, W. N.; KIM, Y. A streaming ensemble algorithm (sea) for large-scale classification. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2001. (KDD '01), p. 377–382. ISBN 1-58113-391-X. Disponível em: <http://doi.acm.org/10.1145/502512.502568>.

TAYLOR, H.; KARLIN, S.; TAYLOR, H. *An Introduction to Stochastic Modeling*. Elsevier Science, 1998. ISBN 9780126848878. Disponível em: <https://books.google.com.br/books?id=UtPgVrVthF8C>.

VAPNIK, V. N. *Statistical Learning Theory*. [S.l.]: Wiley-Interscience, 1998.

WANG, F. et al. Estimating online vacancies in real-time road traffic monitoring with traffic sensor data stream. *Ad Hoc Netw.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 35, n. C, p. 3–13, dez. 2015. ISSN 1570-8705. Disponível em: <https://doi.org/10.1016/j.adhoc.2015.07.003>.

WANG, H. et al. Mining concept-drifting data streams using ensemble classifiers. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2003. (KDD '03), p. 226–235. ISBN 1-58113-737-0. Disponível em: <http://doi.acm.org/10.1145/956750.956778>.

WIDMER, G.; KUBAT, M. Learning in the presence of concept drift and hidden contexts. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 23, n. 1, p. 69–101, abr. 1996. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A:1018046501280>.

WIKIPEDIA. *Markov chain* — *Wikipedia, The Free Encyclopedia*. 2019. <http://en.wikipedia.org/w/index.php?title=Markov%20chain&oldid=924413981>. [Online; accessed 03-November-2019].

YAMANISHI, K.; TAKEUCHI, J.-i. A unifying framework for detecting outliers and change points from non-stationary time series data. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2002. (KDD '02), p. 676–681. ISBN 1-58113-567-X. Disponível em: <http://doi.acm.org/10.1145/775047.775148>.

ZHOU, L. et al. Fpga based low-latency market data feed handler. In: XU, W. et al. (Ed.). *Computer Engineering and Technology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. p. 69–77. ISBN 978-3-662-45815-0.

ZLIOBAITE, I. Learning under concept drift: an overview. *CoRR*, abs/1010.4784, 2010. Disponível em: <http://arxiv.org/abs/1010.4784>.

ZWOLENSKI, M.; WEATHERILL, L. The digital universe rich data and the increasing value of the internet of things. *Australian Journal of Telecommunications and the Digital Economy*, v. 2, 10 2014.