UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Ruth Cândida Catharine Silva Reis

Aplicação de aprendizado ativo em um algoritmo de detecção de novidade em fluxos contínuos

Uberlândia, Brasil

2017

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Ruth Cândida Catharine Silva Reis

Aplicação de aprendizado ativo em um algoritmo de detecção de novidade em fluxos contínuos

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

Orientador: Elaine Ribeiro de Faria Paiva

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2017

Ruth Cândida Catharine Silva Reis

Aplicação de aprendizado ativo em um algoritmo de detecção de novidade em fluxos contínuos

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

Trabalho aprovado. Uberlândia, Brasil, 03 de Agosto de 2017:

Elaine Ribeiro de Faria Paiva
Orientador

Maurício Cunha Escarpinati

Rodrigo Sanches Miani

Uberlândia, Brasil
2017

Agradecimentos

Quero agradecer primeiramente à minha orientadora Elaine. Obrigada pela dedicação que teve com o nosso trabalho do começo ao fim. Obrigada pela paciência que demonstrou comigo em todos os momentos, sempre com muita calma e humildade, você tem o dom de ensinar!

Aos meus amigos que sempre me deram apoio nos momentos em que eu estava desanimada da faculdade. Nunca faltaram amigos que me incentivasse a continuar. Agradeço à Daniela, que esteve comigo desde o cursinho me ajudando e sempre sendo amiga.

À minha família que suportou todos os meus dias maus acreditando que eu conseguiria. Em específico a minha mãe, que independente das circunstâncias me ajudou a realizar esse sonho.

E por fim, quero agradecer à Deus, que fez da faculdade uma escola da vida e tudo contribuiu para o meu amadurecimento. Eu sei que independente das minhas frustrações, Deus estava comigo a cada passo que eu dava me demonstrando Seu grande amor. Tudo o que eu fiz foi primeiramente pra Ele.

Resumo

O crescimento exponencial da tecnologia proporcionou uma aquisição de dados em grande escala, com isso surgiu a necessidade de se manipular essas grandes quantidades de dados que aparecem de forma ilimitada e em alta velocidade. Trabalhar com fluxos contínuos de dados se tornou uma tarefa desafiadora. Algoritmos tradicionais de aprendizado de máquina se tornaram obsoletos para atuarem em cenários dinâmicos, esses algoritmos entendem que o fluxo dos dados é finito e a natureza dos dados é estacionária. Nos últimos anos pesquisadores tem desenvolvido trabalhos para tratar classificação e detecção de novidade em fluxos contínuos. Classificar dados em um ambiente dinâmico e uma tarefa custosa, visto que classificação em fluxos contínuos está diretamente relacionada a mudança e evolução de conceitos, o algoritmo precisa atualizar o seu modelo de decisão de forma incremental para melhor classificar os dados que vão surgindo ao longo do tempo. Este trabalho tem por objetivo melhorar a capacidade preditiva de um algoritmo de classificação existente na literatura, o MINAS. Para isso, foram propostas novas implementações de técnicas de aprendizado ativo comumente utilizadas na literatura, estas técnicas supõem invocar um especialista de domínio para rotular apenas uma amostra de instâncias. Foram realizados experimentos comparando as abordagens propostas com a versão original do MINAS e foi realizada uma análise do impacto disso na performance do algoritmo. Os resultados indicam que a metodologia empregada têm potencial para resolver problemas relacionados a classificação em cenários dinâmicos.

Palavras-chave: fluxos contínuos, detecção de novidade, aprendizado ativo, aprendizado de máquina

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AM	Aprendizado de Máquina
AA	Aprendizado Ativo
DN	Detecção de Novidade
FCDs	Fluxos Contínuos de Dados
MD	Mineração de Dados
SVM	Máquina de Vetor Suporte (Support Vector Machine)

Sumário

1 Introdução

Os Fluxos Contínuos de Dados (FCDs, do inglês *data streams*) podem ser definidos como uma grande quantidade de dados que são sequências infinitas geradas continuamente, normalmente em alta velocidade (AGGARWAL, 2007), (GAMA, 2010), (GAMA; GABER, 2007). Uma das características dos FCDs é que eles não são estacionários, ou seja, a distribuição que gera os dados pode mudar. Assim, os conceitos aprendidos podem evoluir ao longo do tempo. Novos conceitos podem surgir e conceitos antigos podem desaparecer (PAIVA, 2014).

Segundo Gama (2010), FCDs são uma sequência de objetos ilimitada que precisam ser acessados na ordem em que aparecem e lidos apenas uma vez (ou um pequeno número de vezes), possuindo também limitações de tempo, armazenamento e processamento. Os FCDs possuem características diferentes dos dados que atuam no cenário batch. Dentre elas pode-se citar:

- Geralmente os dados chegam em alta velocidade e de forma contínua.
- Possuem tamanho ilimitado.
- Os dados evoluem ao longo do tempo gerando uma distribuição não estacionária.
- Os dados, uma vez processados, são descartados ou armazenados em uma memória, sendo que eles não podem ser recuperados de forma rápida.

Várias aplicações do mundo real geram dados em fluxo contínuo. Nas últimas décadas vê-se um grande interesse em manipular esses fluxos de dados, a fim de que se possa retirar informações importantes, úteis e de grande valor. São exemplos de algumas aplicações que geram FCDs: sistemas de segurança, redes de computadores, mercado financeiro e medicina.

Segundo Berry e Linoff (1997), Mineração de Dados (MD) é a exploração e a análise de uma vasta quantidades de dados, com a finalidade de descobrir padrões e regras significativas. É possível encontrar na literatura vários algoritmos desenvolvidos para atender à MD em FCDs (AGGARWAL et al., 2003), (DOMINGOS; HULTEN, 2000), (CHANG; LEE, 2005). A classificação é uma das importantes tarefas da MD. São fornecidos exemplos para o treinamento que já são rotulados, o modelo de decisão deverá analisar esse conjunto rotulado e aprender a classificar novos exemplos (SPINOSA; CARVALHO; GAMA, 2008). Recentemente, muitos algoritmos foram desenvolvidos para lidar com a tarefa de classificação em FCDs, dentre eles pode-se citar: ECSMiner (MASUD et

al., 2011), CLAM (AL-KHATEEB et al., 2012a), WOCSVM adaptativo (KRAWCZYK; WOŹNIAK, 2013) e HS-Trees (TAN; TING; LIU, 2011).

Como os FCDs não são estacionários, o modelo de decisão induzido pelo algoritmo de classificação precisa evoluir ao longo do tempo. Em FCDs, pode-se definir dois tipos de evolução: i) mudança de conceito (*concept drift*), é uma mudança nas definições das classes (conceitos) à medida que o tempo passa, portanto há mudança na distribuição a partir da qual os dados são gerados (ELWELL; POLIKAR, 2011). ii) evolução de conceito (*concept evolution*) está relacionado a classes emergentes, ou seja, o número de classes aumenta com o passar do tempo (MASUD et al., 2011).

É importante que esses dois tipos de evolução sejam tratados, pois em FCDs o ambiente não é estacionário. Em cenários *batch* onde os algoritmos tradicionais de AM assumem que o número de classes é previamente definido, e devem pertencer a no mínimo uma das classes que estão no conjunto predefinido (PARK; SHIM, 2010). Sendo assim, em cenários de FCDs não é possível aplicar essas técnicas. Algoritmos para FCDs devem ser capazes de atualizar o modelo de decisão e se adequar às novas classes que vão surgindo ao longo do tempo e identificando ruídos e *outliers* para que eles sejam descartados.

Outro conceito bastante abordado em FCDs é a detecção de novidade (DN) - do inglês *novelty detection*. Uma novidade pode ser definida como uma tarefa de aprendizagem que consiste na identificação de novos conceitos que o sistema não tem conhecimento durante a fase de treino (GAMA, 2010). Esta ideia está diretamente relacionada com o conceito de classificação, pois quando novas classes do problema surgem, estas são identificadas inicialmente como novidades, e, futuramente, precisam ser incorporadas ao modelo de decisão.

Frente aos cenários de mudança de conceito, evolução de conceito e presença de *outliers*, uma das abordagens usadas pelos algoritmos da literatura para atualizar o modelo de decisão é usar exemplos rotulados (MASUD et al., 2011), (MASUD et al., 2010), (AL-KHATEEB et al., 2012a), (FARID; RAHMAN, 2012), (FARID et al., 2013). Em FCDs, a tarefa de rotular todos os exemplos do fluxo é muito custosa e, por vezes, inviável de ser realizada.

Uma das recentes abordagens usadas para lidar com tal problema é o uso de aprendizado ativo (do inglês *active learning*), que consiste em uma estratégia de aprendizado supervisionado no qual o algoritmo tenta selecionar amostras de exemplos que são mais coesas e representativas para serem rotuladas. Sendo assim o algoritmo solicita que um especialista de domínio faça o processo de rotulação dos exemplos das amostras que foram selecionadas pelo algoritmo (CHERMAN, 2013). Assim, os usuários poderão de forma ativa colaborar na correta identificação das novidades que podem surgir ao longo do FCDs, contribuindo assim para melhorar a acurácia do classificador.

O aprendizado ativo será abordado com mais afinco neste projeto, sua finalidade é a interação com um humano onde este participa do processo de rotulação dos dados de forma que estes dados possam ser usados para atualizar o modelo de decisão. Supor que todos os dados estão rotulado para atualizar um modelo de decisão é irrealístico para FCDs. Supor que nenhum dado é rotulado também não é uma ideia adequada para vários problemas. No entanto, escolher um conjunto de dados e solicitar que o usuário rotule somente este conjunto pré-selecionado parece ser uma ideia interessante para aumentar a qualidade do modelo de decisão. Assim, o modelo de decisão será atualizado usando esse conjunto de exemplos rotulados. Durante o processo de entrada de FCD's no sistema, quando este se deparar com uma novidade, poderá ser solicitado ao usuário que rotule estes exemplos que estão causando dúvida no sistema,e , assim o modelo usado para classificar novos dados será atualizado.

1.1 Justificativa

Vários algoritmos foram desenvolvidos para a tarefa de detecção de novidade em FCDs. Dentre eles, pode-se citar: OLINDDA (SPINOSA et al., 2009), DETECT-NOD (HAYAT; HASHEMI, 2010), MCM (MASUD et al., 2010), WOCSVM adaptive (KRAWCZYK; WOŹNIAK, 2013), SONDE (ALBERTINI; MELLO, 2007), Tree for ND (FARID; RAHMAN, 2012) e MINAS (PAIVA, 2014). Todos esses algoritmos exigem que o seu modelo de decisão seja atualizado constantemente a fim de refletir as mudanças nos dados e o aparecimento de novas classes.

Em geral, duas estratégias têm sido usadas para atualizar o modelo de decisão. A primeira estratégia é chamada supervisionada (MASUD et al., 2011), (MASUD et al., 2010), (AL-KHATEEB et al., 2012a), (FARID; RAHMAN, 2012), (FARID et al., 2013), na qual espera-se que os exemplos sejam rotulados após serem processados. Usando os exemplos rotulados o modelo de decisão pode ser atualizado. No entanto, rotular todos os exemplos é uma tarefa custosa e tediosa, especialmente em FCDs nos quais os dados chegam constantemente.

A segunda estratégia é chamada não-supervisionada (SPINOSA et al., 2009), (HAYAT; HASHEMI, 2010), (TAN; TING; LIU, 2011), (ALBERTINI; MELLO, 2007), (KRAWCZYK; WOŹNIAK, 2013), e consiste em atualizar o modelo de decisão sem usar o rótulo dos exemplos. No entanto, muitas vezes, quando o modelo é atualizado sem nenhum tipo de supervisão, ele acaba perdendo desempenho. Além disso, muitas vezes pode haver dados rotulados disponíveis para serem usados.

Uma terceira estratégia tem sido utilizada para atualizar o modelo de decisão usando aprendizado ativo (PAIVA, 2014), (ABDALLAH et al., 2016). Esta técnica consiste na amostragem de dados, na qual são selecionados exemplos que provêm maior ganho de

informação. Enquanto um aprendiz passivo obtém todos os dados rotulados de uma única vez, um aprendiz ativo seleciona quais exemplos ele gostaria de saber o rótulo (ZHU et al., 2007). Algoritmos de aprendizado ativo podem reduzir substancialmente a quantidade de exemplos rotulados necessários para se construir um bom classificador (SETTLES, 2010). As vantagens de usar essa técnica para atualizar o modelo de decisão em FCDS são:

- Aumentar o desempenho preditivo de um classificador;
- Garantir uma maior precisão na rotulação do exemplos através da intervenção de um especialista de domínio;
- Detectar com eficiência classes pouco representativas, que normalmente foram esquecidas pelo modelo, ou seja, mal classificadas.

Um dos recentes algoritmos da literatura para detecção de novidade em FCDs, chamado MINAS (*MultI-class learNing Algorithm for data Streams*) (PAIVA, 2014), usa uma técnica de aprendizado ativo para atualizar o modelo de decisão. No entanto, a técnica usada por ele é bastante simples e consiste em: i) agrupar os exemplos não rotulados, ii) identificar grupos de exemplos válidos, iii) identificar os grupos válidos como extensões das classes conhecidas do problema ou como novidades, iv) escolher o centroide de cada um dos grupos categorizados como novidade na última janela de dados, e solicitar o especialista de domínio para rotulá-los.

Este trabalho, baseia-se na ideia de que técnicas mais elaboradas de aprendizado ativo podem melhorar o desempenho do algoritmo MINAS na tarefa de classificação e DN em FCDs. Além disso, o estudo e análise de tais técnicas permitirão decidir a viabilidade de usá-las em outros algoritmos para DN em FCDs já desenvolvidos na literatura.

1.2 Objetivos

1.2.1 Objetivo Geral

Pesquisar e implementar novas formas de aprendizado ativo em um recente algoritmo da literatura para detecção de novidade em fluxos contínuos de dados.

1.2.2 Objetivos Específicos

- Analisar e comparar os diferentes métodos de aprendizado ativo existentes na literatura para FCDs, a fim de identificar o que melhor se adequa ao problema. Estudar também os métodos desenvolvidos para cenários *batch* e verificar a viabilidade de usá-los em cenários de FCDs.

- Implementar um ou mais métodos de aprendizado ativo da literatura e adaptá-los para serem usados pelo algoritmo para detecção de novidade chamado MINAS.

- Comparar a nova versão do algoritmo MINAS com a versão original e analisar se houve uma melhora no desempenho do mesmo. Para a comparação, medidas de validação usadas para a classificação em FCDs e medidas de validação para aprendizado ativo serão estudadas e usadas.

1.3 Organização da Monografia

O trabalho está organizado da seguinte forma:

- **Capítulo 2:** faz a definição de FCDs, quais os cenárias em que são aplicados e quais os problemas encontrados devido as suas características específicas. Trata do processo de classificação e detecção de novidade em FCDs e cita alguns algoritmos da literatura que trabalham neste tipo de cenário.
- **Capítulo 3:** faz uma descrição detalhada do algoritmo MINAS, define o que é aprendizado ativo, apresenta as técnicas mais utilizadas e faz algumas propostas de implementação de aprendizado ativo no MINAS.
- **Capítulo 4:** este capítulo relata os resultados das recentes implementações de aprendizado ativo no MINAS, os resultados encontrados são comparados à versão original, que implementa técnicas simples de aprendizado ativo.
- **Capítulo 5:** este capítulo apresentará as conclusões do trabalho.

2 Revisão Bibliográfica

2.1 Introdução

Extrair informações úteis de grandes quantidades de dados tem sido uma tarefa desafiadora para os pesquisadores. Os algoritmos projetados para tratar FCDs atuam em um cenário diferente dos algoritmos tradicionais de Aprendizado de Máquina (AM), isso devido ao caráter dinâmico presente em FCDs. Os algoritmos de classificação que trabalham neste tipo de ambiente devem ser capazes de se adequar as constantes mudanças que aparecem ao longo do fluxo.

Dessa forma, a tarefa de Detecção de Novidade (DN) é imprescindível neste processo, pois é ela que identifica o surgimento de novos conceitos e a mudança nos conceitos existentes, que posteriormente irão atualizar o modelo de decisão. Visando melhorar o processo de classificação e DN em FCDs, estão sendo exploradas técnicas de aprendizado ativo que auxiliam no processo de escolha das instâncias a serem rotuladas para atualizar o modelo de decisão.

Este capítulo descreve as diferentes situações que estão presentes em FCDs. A seção 2.2 apresenta os FCDs, exemplos de aplicações do mundo real que geram dados em fluxo e quais os desafios encontrados para manipular grandes quantidades de dados. A seção 2.3 traz o conceito de classificação em FCDs e como os algoritmos da literatura foram desenvolvidos para esta finalidade. A seção 2.4 detalha em qual etapa do processo de classificação a DN é aplicada e como é realizada a atualização do modelo de decisão. A seção 2.5 relata como são implementadas as técnicas de aprendizado e seus desafios. A seção 2.6 faz uma breve descrição de algoritmos presentes na literatura que foram desenvolvidos para classificar dados e detectar novidades em FCDS.

2.2 Fluxos Contínuos de Dados

Para Babcock et al. (2002), Fluxos Contínuos de Dados (FCDs) são sequências de dados que chegam de forma contínua, em geral, em alta velocidade, com tamanho ilimitado e o sistema não tem controle sobre a ordem na qual os exemplos chegam. Muitas pesquisas relacionadas a FCDs têm sido feitas nos últimos anos e algoritmos estão sendo desenvolvidos para atuar nesse cenário, como por exemplo, ECSMiner (MASUD et al., 2011), OLINDDA (SPINOSA et al., 2009), CLAM (AL-KHATEEB et al., 2012a), MINAS (PAIVA, 2014). Desenvolver algoritmos para trabalhar com cenários FCDs é uma tarefa custosa por se tratar de uma sequência de exemplos gerados continuamente, em alta

velocidade e a distribuição que gera os dados pode mudar ao longo do tempo (SILVA et al., 2013).

Segundo Guha e Mishra (2016), os avanços recentes em software e hardware proporcionaram uma aquisição maior de dados em grande escala, o que caracteriza ambientes dinâmicos, enquanto bases de dados tradicionais supõe cenários estáticos. Com as inovações tecnológicas em alta, estão surgindo inúmeras aplicações do mundo real com essas características. Alguns dos exemplos típicos incluem:

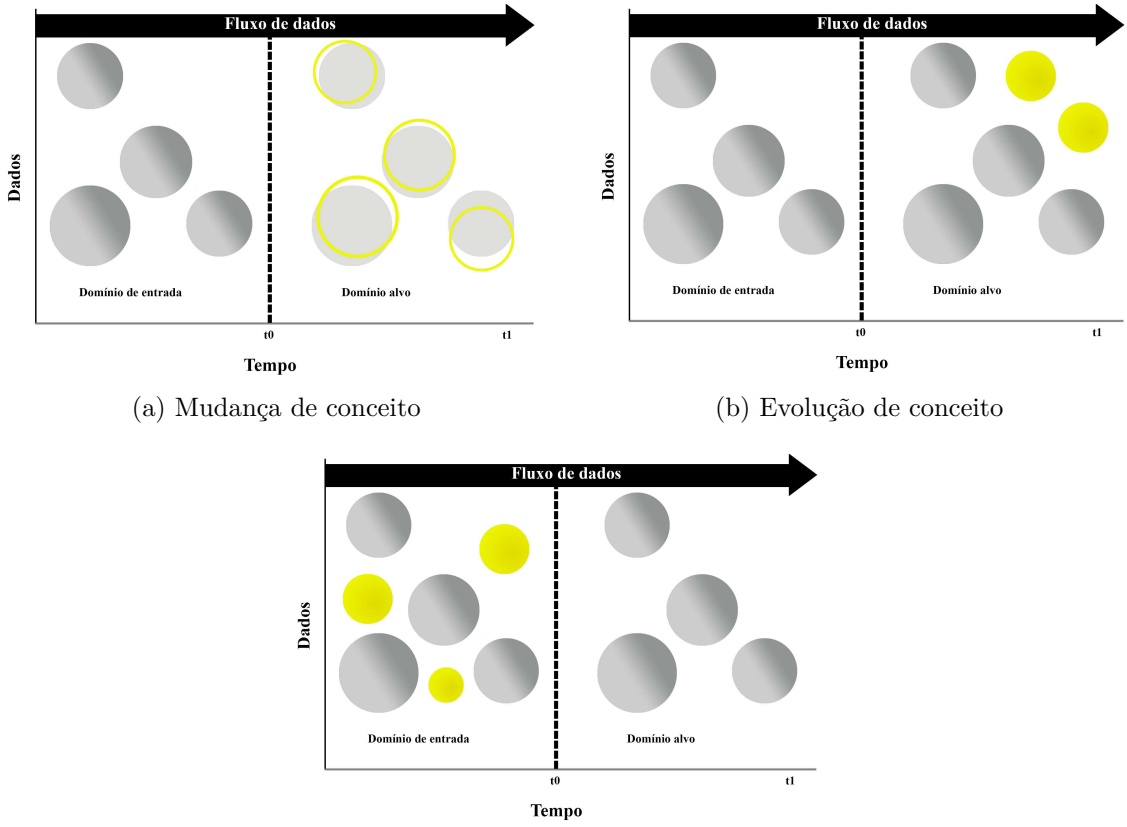
- Sistemas de segurança: realizam o monitoramento por meio de câmeras de segurança que capturam imagens a fim de identificar intrusos;
- Redes de computadores: análise da rede, monitoramento de pacotes para detecção de padrões não usuais existentes, ou ainda, detecção de invasores;
- Mercado Financeiro: dados relacionadas à bolsa de valores são analisadas e os resultados são gerados trazendo informações importantes para investidores. Fraudes em cartões de créditos é uma das importantes aplicações para essa área;
- Medicina: à medida que novos casos são estudados e resultados diferentes são obtidos, o modelo inicial conhecido de uma doença pode mudar à partir de novos padrões detectados.

Os FCDs possuem as seguintes características (PAIVA, 2014):

- O processo de geração dos dados não é estacionário, isto é, a distribuição de probabilidade pode mudar ao longo do tempo;
- O fluxo pode ser considerado infinito, e, portanto, os exemplos não podem ser armazenados em memória;
- Os elementos que compõem os dados são recebidos de maneira contínua.

Essas características não permitem que algoritmos tradicionais de mineração dados (MD) sejam aplicados facilmente em FCDs (MAHDIRAJI, 2009).

Para Elwell e Polikar (2011), mudança de conceito (*concept drift*) é uma mudança na distribuição a partir do qual os dados são gerados, ou seja, refere-se a uma mudança na definição das classes (conceitos) ao longo do tempo. Um exemplo disso é a mudança no padrão de compra de um novo cliente (ver Fig.1a). Uma evolução de conceito (*concept evolution*) é o aparecimento de uma nova classe, diferente das classes já conhecidas, representando assim uma evolução, por exemplo, o surgimento de um novo tipo de ataque em uma rede de computadores (ver Fig.1b). Além de lidar com o aparecimento de novos



(a) Mudança de conceito

(b) Evolução de conceito

(c) Esquecimento de conceito

Figura 1 – Mudanças em fluxos de dados - Adaptada de (Abdallah, 2016).

conceitos, FCDs também exigem o esquecimento de conceitos desatualizados e abandonados que não tem utilidade na atividade atual do fluxo e ocupam espaço (ABDALLAH et al., 2016)(ver Fig.1c).

2.3 Classificação em FCDs

A classificação em FCDs tem por objetivo predizer, com alta acurácia, a classe dos novos exemplos que chegam ao longo do fluxo (PAIVA, 2014). Na fase de aprendizado supervisionado, um conjunto de exemplos para os quais a classe (ou rótulo) é conhecida, é fornecido ao algoritmo de aprendizado supervisionado. O objetivo do algoritmo é construir um modelo que possa determinar automática e corretamente a classe de novos exemplos encontrados, considerando que o modelo de decisão aprendido evolui ao longo do fluxo.

Muitos algoritmos de classificação foram desenvolvidas para trabalhar em um tradicional cenário em lote (*batch*), estão entre eles: árvores de decisão (ROKACH; MAIMON, 2008), SVM (STEINWART; CHRISTMANN, 2008) e *Naive Bayes*, nos qual o ambiente é estático, o modelo de decisão não muda e o algoritmo assume que todos os dados estão na memória. Aprender novos conceitos a partir de fluxos contínuos de dados, de forma a atualizar o modelo de decisão constantemente, é significativamente diferente do

aprendizado tradicional, no qual uma vez criado o modelo, este não se altera ao longo do tempo. Vários algoritmos projetados para cenários estáticos tem sido adaptados e novos algoritmos estão sendo desenvolvidos para atuar em cenários dinâmicos.

Em geral, os algoritmos de FCDs trabalham com duas fases: *online* e *offline*. Na fase offline o algoritmo recebe exemplos rotulados, que são usados para induzir um modelo de decisão ou atualizar um modelo existente. Na fase online acontece a classificação de novos exemplos, à medida que novos dados são adquiridos. O modelo construído é atualizado constantemente de forma a refletir as características mais atuais do fluxo.

De acordo com Aggarwal (2007), o desafio mais complexo na classificação em cenários FCDs está relacionada a mudança de conceito, isto é, a mudança na distribuição dos dados ao longo do tempo. Além disso, outras questões precisam ser tratadas, tais como: evolução de conceitos, detecção e remoção de ruídos e *outliers*, e manutenção de uma boa acurácia à medida que os dados evoluem. O algoritmo deve ter a capacidade de atualizar de forma incremental o modelo de decisão, fazendo com que a aplicação tenha uma aprendizagem contínua.

A atualização do modelo de decisão é um importante problema a ser tratado na tarefa de classificação em FCDs. A atualização do modelo pode ser feita com ou sem *feedback* externo. Algoritmos que usam *feedback* externo assumem que o rótulo verdadeiro de todos os exemplos estará disponível mesmo que seja com um certo atraso. Assim, o algoritmo faz uma atualização do modelo de decisão de tempos em tempos à medida que se tem o rótulo dos exemplos. Os algoritmos que não utilizam *feedback* tem seu modelo atualizado sem a informação verdadeira do rótulo dos exemplos. Há também a opção com *feedback* parcial, nesse caso um pequeno conjunto de exemplos rotulados é usado para atualizar o modelo.

2.4 Detecção de Novidade em FCDs

Detecção de novidade é a tarefa de classificar dados de teste que diferem em algum aspecto dos dados usados para treinamento (PIMENTEL et al., 2014). Esta é uma tarefa importante em classificação de FCDs, uma vez que estes representam distribuições de dados não-estacionárias. Segundo Gama (2010), a DN tem por objetivo detectar padrões emergentes no fluxo de dados que podem identificar o aparecimento de um novo conceito, uma mudança nos conceitos conhecidos ou a presença de ruídos.

Nos últimos anos, vários pesquisadores desenvolveram algoritmos para classificação em FCDs que também trabalham com a tarefa de DN. Entre eles estão: Masud et al. (2011), Spinosa et al. (2009), Hayat e Hashemi (2010), Farid e Rahman (2012), Al-Khateeb et al. (2012a), Al-Khateeb et al. (2012b) e Paiva (2014). Desenvolver algoritmos para DN em FCDs apresenta inúmeros desafios que incluem:

- Mudança de conceito: que dificulta a diferenciação de novos conceitos e conceitos conhecidos;
- Contextos recorrentes: conceitos que já foram esquecidos surgem novamente e podem ser confundidos com o aparecimento de novos conceitos;
- Evolução de conceitos: ocorre quando o número de classes aumenta com o passar do tempo;
- Ruído ou *outlier*: estes podem ser confundidos com o surgimento de novos conceitos.

Em geral, os sistemas de classificação que tratam a tarefa de DN, na fase online, tentam identificar exemplos que não podem ser classificados pelo modelo decisão atual. Algumas propostas simplesmente identificam estes exemplos como anômalos. Propostas mais refinadas, armazenam estes exemplos em uma memória temporária e fazem uma análise futura para tentar identificar o surgimento de uma novidade.

Embora a maioria dos trabalhos para DN em FCDs considere que apenas uma classe novidade pode surgir a cada janela de tempo, alguns trabalhos como MINAS (PAIVA, 2014) e MCM (MASUD et al., 2010), assumem que diferentes padrões novidade podem surgir ao mesmo tempo, sendo importante diferenciá-los.

A atualização do modelo de decisão a partir das novidades encontradas pode ser feita usando *feedback* externo ou sem *feedback* (como explicado na seção 2.2). Uma interessante abordagem, usada por algoritmos como MINAS (PAIVA, 2014) e AnyNovel (ABDALLAH et al., 2016), é usar técnicas de aprendizado ativo para escolher, dentre os exemplos não classificados pelo modelo de decisão atual e usado para criar os padrões novidade, quais deles devem ser rotulados a fim de se atualizar o modelo de decisão. A próxima seção discute o aprendizado ativo neste tipo de cenário.

2.5 Aprendizado Ativo em FCDs

A tarefa de classificação possui uma abordagem chamada aprendizagem ativa (do inglês *active learning*), que permite selecionar um conjunto de instâncias e solicitar um especialista de domínio para rotulá-las (TONG, 2001). As instâncias rotuladas são então usadas para atualizar o modelo de decisão. As técnicas de aprendizado ativo tentam superar o problema de atualização do modelo de decisão, que supõe que todos os dados são rotulados por uma abordagem que escolhe os exemplos a serem rotulados por um especialista (SETTLES, 2010). Dessa forma, espera-se minimizar o custo para rotular dados. Esta é uma interessante abordagem a ser usada em FCDs, uma vez que neste cenário torna-se impraticável rotular todos os dados. No entanto, rotular uma porção dos

dados, em especial aqueles que representam mudanças nos conceitos conhecidos, pode ser uma abordagem factível em vários cenários envolvendo FCDs.

O principal desafio da aprendizagem ativa é identificar quais instâncias que deveriam ser rotulados para se alcançar uma maior previsão, tendo em vista que não se pode rotular todas as instâncias (ZHU et al., 2007).

Segundo Ienco, Žliobaitė e Pfahringer (2014), as estratégias de aprendizado ativo em FCDs, além de serem capazes de induzir o aprendizado de um classificador, devem ser capazes de:

- Ao longo do tempo equilibrar o custo ao se realizar o processo de rotular instâncias;
- Preservar a distribuição dos dados de entrada para detectar alguma alteração;
- Detectar onde as mudanças estão acontecendo ao longo do fluxo de dados.

Trabalhos estão sendo desenvolvidos para abordar o problema de escassez de rótulo em fluxos contínuos (KLINKENBERG, 2001). Grande parte destes trabalhos tem como metodologia fazer um particionamento dos FCDs e dividi-los em lotes. Então, o processo de aprendizado ativo é aplicado dentro de cada lote. Assume-se que os dados presentes no lote estão em um ambiente estacionário. O objetivo desta técnica é treinar um classificador mais preciso utilizando uma pequena porção de dados para minimizar os custos de rotulagem.

Considerando os sistemas que trabalham com DN em FCDs usando aprendizado ativo, uma abordagem comum é, agrupar os exemplos não classificados pelo modelo de decisão, que são considerados potenciais novidades, e escolher um representante do grupo para ser rotulado. Essa estratégia é usada, por exemplo, pelos algoritmos MINAS (PAIVA, 2014) e AnyNovel (ABDALLAH et al., 2016).

2.6 Trabalhos para DN em FCDs

O objetivo deste seção é descrever alguns dos recentes algoritmos de DN presentes na literatura para FCDs. Serão abordados os seguintes algoritmos: MINAS (PAIVA, 2014), OLINDDA (SPINOSA et al., 2009), ECSMiner (MASUD et al., 2011), CLAM (ALKHATEEB et al., 2012a) e ANYNOVEL (ABDALLAH et al., 2016). Cada algoritmo é composto por duas fases: a primeira é chamada *offline*, nesta fase ocorre a indução do modelo de decisão, que tem por base um conjunto de dados rotulados. A segunda fase é a *online*, nesta etapa o algoritmo recebe os dados de forma contínua e os classifica. Nesta fase também ocorre a detecção de novidade e a atualização do modelo de decisão, sendo que esta é realizada de forma *offline* por alguns algoritmos.

ECSMiner (MASUD et al., 2011) e CLAM (AL-KHATEEB et al., 2012a) são algoritmos criados para trabalharem com DN, que consideram a tarefa de classificação multiclasse. O ECSMiner pega um *chunk* de dados e constrói um comitê de classificadores de árvores de decisão. Após a construção da árvore é realizado um agrupamento em cada nó folha. Os microgrupos gerados são utilizados para explicar os exemplos do fluxo. Os exemplos não explicados pelo modelo de decisão, ou seja, não estão dentro de nenhum microgrupo, são marcados como *outliers* e armazenados para futura análise. Quando há um número suficiente de elementos marcados como *outliers* eles são agrupados e os grupos válidos que estão suficientemente distante dos grupos conhecidos são marcados como novidade. Um das limitações do ECSMiner é que ele supõe que somente uma classe novidade aparece por vez a cada *chunk* de dados. Se mais classes aparecem são simplesmente rotulados como novidade, não diferenciando os padrões-novidade que fazem parte da classe emergente.

O algoritmo CLAM usa uma estratégia parecida com o algoritmo ECSMiner, no entanto, ao invés de usar um comitê de classificadores, ele usa um conjunto de C comitês, um para cada classe conhecida do problema. Cada classificador de cada comitê é formado por um conjunto de microgrupos. Classificar um exemplo é identificar qual microgrupo cujo centro está mais próximo ao elemento. Ao receber um novo exemplo, cada comitê existente faz a verificação se este exemplo pode ser classificado usando o comitê de classificadores atual, caso contrário ele é marcado como desconhecido. Para se classificar um elemento como desconhecido é necessário analisar o limite de decisão de cada classificador representado pelos micro-grupos. Se na maioria dos classificadores o elemento está fora do limite de decisão, então este elemento é classificado como desconhecido. O CLAM usa uma técnica de DN parecida com a do ECSMiner, não faz diferenciação de padrão novidade no mesmo bloco de dados.

O OLLINDDA (SPINOSA; CARVALHO; GAMA, 2008) é um algoritmo para DN que considera que na fase *offline*, somente exemplos da classe normal estão disponíveis para treinar o modelo de decisão inicial. O sub-modelo normal é criado na fase *offline* os sub-modelos extensão e novidade são criados na fase online. O OLINDDA utiliza um algoritmo de agrupamento para produzir k grupos que são representados por um centro e um raio. O modelo normal é representado por esse conjunto de k grupos. Além disso, uma macro-esfera é criada de forma a manter um macrogrupo de todos os k grupos criados. A macro-esfera é usada para separar o conceito normal do conceito extensão e novidade. Quando novos grupos criados a partir de exemplos não classificados pelo modelo, estão dentro da macro-esfera eles são considerados extensões do conceito conhecido. Quando estão de fora da macro-esfera, são considerados novidade. Para se classificar um novo exemplo calcula-se a distância entre o centroide do grupo encontrado e o centroide do grupo mais próximo. Se a distância for menor que o raio do grupo, o exemplo é explicado pelo modelo atual, caso contrário é marcado como desconhecido. Grupos de exemplos

desconhecidos são usados para criar os submodelos extensão e novidade.

O MINAS (PAIVA, 2014) também é um algoritmo desenvolvido para cenários de FCDs. Na fase offline o algoritmo constrói o seu modelo de decisão utilizando a proposta supervisionada. A fase de aprendizado é iniciada a partir da entrada de um conjunto de dados rotulados com diferentes classes de exemplo. Os exemplos de cada classe são submetidos a um algoritmo de agrupamento, como K-Means ou Clustream. Após a execução do algoritmo de agrupamento cada microgrupo gerado possui um marcador de tempo t que representa o último exemplo que foi inserido no grupo e o rótulo que indica qual a classe ele pertence.

Caso os exemplos que chegam ao longo do fluxo, não sejam explicados pelo atual modelo, estes são marcados com o perfil desconhecido. Exemplos desconhecidos são guardados em uma memória temporária para posteriormente serem analisados. Quando um número mínimo de exemplos é atingido na memória temporária, o MINAS executa um agrupamento para identificar se existe algum padrão formado por esses exemplos não rotulados. Caso os novos microgrupos encontrados sejam válidos é medido a distância destes com os demais microgrupos já existentes. Assim, o algoritmo identifica se o novo grupo é uma novidade ou uma extensão dos conceitos conhecidos, baseando-se num limiar pré-definido, e incorpora este novo grupo ao modelo de decisão.

O algoritmo MINAS apresenta duas forma de atualização do modelo de decisão. A primeira delas é sem *feedback*, ou seja, a medida que novos microgrupos são criados, a partir de exemplos não rotulados, estes são incorporados ao modelo. A segunda é usando *feedback* parcial. Para isso, o MINAS utiliza uma técnica de aprendizado ativo, que consiste em solicitar ao especialista de domínio o rótulo dos micro-grupos marcados como novidade ou extensão.

AnyNovel (ABDALLAH et al., 2016) é um algoritmo para tratar DN relacionada a mudança e evolução de conceitos em FCDs em um cenário específico, reconhecimento de atividades a partir de dados gerados por sensores. AnyNovel apresenta uma proposta para trabalhar com mudanças na distribuição dos dados utilizando técnicas de aprendizado ativo. Os dados que chegam ao longo do fluxo são armazenados em uma memória temporária, esta memória divide a corrente de dados em n blocos que são liberados para análise somente quando o buffer está cheio de instâncias. Quando o *buffer* atinge seu limite máximo o algoritmo tenta fazer uma decisão sobre os dados e classificá-los em: existente, quando os dados são identificados como parte de um bloco que já existe, novidade, quando satisfaz todas as condições de um novo conceito e desconhecido, quando existe um alto grau de incerteza entre as decisões de existente e novidade. Um conceito encontrado é considerado uma possível novidade quando este determinado bloco de dados está muito distante dos demais *clusters*.

A aprendizagem ativa no AnyNovel acontece de duas maneiras: o especialista de

domínio é chamado quando são identificados blocos de dados desconhecidos, ou quando se detecta uma mudança de conceito nos dados.

2.7 Considerações finais

Neste capítulo foram apresentados alguns algoritmos de detecção de novidade presentes na literatura para FCDs. Embora existam alguns algoritmos para tratar FCDs, em especial detecção de novidade, a maioria dos algoritmos tenta atualizar o modelo de decisão de tempos em tempos usando estratégias supervisionadas ou não supervisionadas. Poucos trabalhos utilizam a estratégia de aprendizado ativo, que tem como vantagem a intervenção de um especialista de domínio para rotular os exemplos, isso resulta em maior eficácia, diminui os erros na classificação e aumenta a capacidade preditiva do algoritmo.

Este trabalho tem o objetivo de melhorar o desempenho da tarefa de classificação do algoritmo MINAS. Para isso serão utilizados diferentes abordagens de aprendizado ativo e analisados os impactos dessas abordagens no desempenho do MINAS.

3 Proposta

3.1 Introdução

Este capítulo irá tratar a detecção de novidade no algoritmo MINAS de forma mais detalhada, apresentando como o algoritmo trabalha para encontrar uma novidade em um cenário multiclasse e como o aprendizado ativo pode atuar neste contexto. Será descrita a implementação do aprendizado ativo na sua versão original, chamada MINAS-AA, e as novas abordagens propostas serão apresentadas.

A seção 3.2 descreve a versão original do MINAS detalhando suas duas fases e como é feito o processo de detecção de novidade. A seção 3.3 indica como foi implementado o aprendizado ativo na versão original do MINAS. A seção 3.4 traz novas propostas de aprendizado ativo para melhorar o desempenho do MINAS. A seção 3.5 detalha as medidas de avalição presentes no MINAS, as quais serão usadas neste trabalho.

3.2 O algoritmo MINAS

O MINAS é um algoritmo desenvolvido para tratar o problema de classificação, com multiclasse e DN, em um ambiente de FCDs. Dentro deste cenário, o algoritmo possui propriedades que o permitem trabalhar de forma dinâmica, visto que mudança e evolução de conceitos são fatores que estão relacionados a FCDs. O MINAS, assim como a maioria dos algoritmos de classificação para FCDs é composto por duas fases: *offline* e *online*.

A fase *offline* consiste na fase inicial do algoritmo. Nesta etapa o modelo de decisão inicial é construído de forma supervisionada a partir de um conjunto de dados rotulados, contendo exemplos de uma ou mais classes, este processo é executado somente uma vez. Os microgrupos são criados pela execução de um algoritmo de agrupamento nos exemplos de cada uma das classes do problema. O MINAS propõe o uso dos algoritmos K-Means (LLOYD, 1982) (MACQUEEN et al., 1967) e CluStream (AGGARWAL et al., 2003) para esta tarefa, cujo o objetivo é criar um classificador composto de um conjunto de microgrupos. Sendo que, um microgrupo é uma estrutura comumente usada em FCDs (PAIVA, 2014), (SPINOSA et al., 2009), e representa um sumário estático dos dados. Ele é composto de 4 informações: número de instâncias do grupo (N), soma das N instâncias do grupo (LS), soma quadrada das N instâncias do grupo (SS) e um marcador de tempo associado ao último exemplo classificado pelo microgrupo (T), que carrega a informação do último exemplo que chegou no microgrupo.

Os microgrupos possuem duas importantes características: incrementabilidade,

este recurso permite que outros exemplos possam ser adicionados ao microgrupo com o passar do tempo, e a aditividade, recurso que permite a união de dois microgrupos. Um modelo criado usando esta estrutura traz como resultado um baixo custo computacional, já que o modelo não precisa ser reconstruído, apenas atualizado.

A fase *online*, também conhecida como fase de aplicação, recebe um fluxo contínuo de dados e classifica-os usando o modelo de decisão construído na fase *offline*. Inicialmente, os novos exemplos que forem explicadas pelo modelo de decisão, serão imediatamente classificados. Classificar um exemplo usando os microgrupos envolve calcular a distância entre o exemplo e o centroide do microgrupo mais próximo, se a distância encontrada for menor que o raio do microgrupo, então o exemplo é explicado pelo microgrupo e recebe o seu rótulo. Quando um exemplo é explicado por algum dos microgrupos presentes no modelo, o indicador de tempo t presente no microgrupo é atualizado para o valor do marcador de tempo do último exemplo que foi inserido. Caso o modelo de decisão não consiga classificar o exemplo, este é rotulado como desconhecido e vai para uma memória temporária para uma futura análise.

De tempos em tempos a memória temporária é analisada e caso ela possua um número mínimo de exemplos, um algoritmo de agrupamento é executado a fim de criar novos microgrupos a serem inseridos no modelo. Esses novos microgrupos criados passam por um processo de validação, que consiste em verificar se a silhueta do microgrupo possui valor maior que zero, usando a equação presente na Equação 3.1. Cada microgrupo precisa ser coeso e representativo. A coesão é avaliada pela silhueta, a representatividade é avaliada pela quantidade de elementos que o microgrupo possui. Se a quantidade de elementos é maior que o número mínimo, então o microgrupo é inserido no modelo de decisão. No entanto, antes de inserir, é necessário decidir se o microgrupo representa uma novidade ou uma extensão.

$$Silhueta = \frac{b - a}{max(b, a)} \tag{3.1}$$

A silhueta é utilizada para calcular a largura de um microgrupo recentemente criado. Nesta equação, a representa o desvio padrão das distâncias entre os exemplos do novo microgrupo e o centroide do mesmo, b representa a distância do centroide do novo microgrupo e o centroide do microgrupo mais próximo.

Após encontrar um novo microgrupo válido é necessário verificar se ele é uma novidade ou uma extensão de um conceito existente. Para isso, a próxima etapa é encontrar a distância entre o centroide do novo microgrupo e o centroide do microgrupo mais próximo. Se a distância encontrada for menor que um limiar T, então o novo microgrupo é uma extensão, o seu rótulo será o mesmo do microgrupo mais próximo. Caso contrário o microgrupo é uma novidade.

O modelo de decisão é atualizado na fase online de forma não-supervisionada, ou seja, não é levado em consideração o rótulo verdadeiro dos exemplos. O modelo pode ser atualizado somente de duas formas: no primeiro caso se o exemplo encontrado for explicado pelo modelo, o microgrupo (ou um conjunto de microgrupos) que o classificou pode ter o seu sumário estatístico atualizado e o exemplo recebe o rótulo deste microgrupo. No segundo caso, se um microgrupo válido for encontrado na memória temporária, este é rotulado como novidade ou extensão e inserido ao modelo.

Existe um processo no algoritmo MINAS que analisa a memória temporária a fim de eliminar os microgrupos mais antigos. Cada microgrupo possui um marcador associado ao último exemplo que foi classificado como pertencente a este microgrupo. Aqueles microgrupos que não recebem novos elementos por um certo período de tempo P e que não estão mais contribuindo para classificar novos elementos, são inseridos em uma memória *sleep*, assim o modelo não utiliza esses microgrupos para classificar exemplos. Esta memória *sleep* futuramente será utilizado no processo de tratamento de recorrência, ou seja, no tratamento de conceitos que aparecem, desaparecem e voltam a aparecer novamente.

Além disso, o MINAS também possui uma funcionalidade que elimina da memória temporária os elementos que não estão sendo mais usados para formar novos microgrupos. Cada elemento possui um marcador de tempo associado à sua inserção na memória temporária. Assim, elementos cujo marcador de tempo é muito antigo, em relação ao marcador atual, são removidos. Sendo assim, toda vez que um elemento é adicionada a memória temporária, esta é verificada com a finalidade de eliminar elementos cuja diferença entre marcador atual e marcador do elemento é menor que um limiar *ts*.

3.3 Aprendizado ativo no algoritmo MINAS

Aprendizado ativo é uma técnica pouco utilizada em FCDs, pois a maioria dos algoritmos para classificação existentes na literatura supõe que o rótulo dos exemplos estará sempre disponível. Essa é uma suposição que não é verificada em muitas aplicações do mundo real, uma vez que para se obter o rótulo verdadeiro dos exemplos exige muito tempo e esforço. A ideia apresentada pelo aprendizado ativo é permitir a intervenção de uma especialista de domínio para rotular alguns exemplos, nos quais há incerteza sobre sua classificação, a fim de melhorar o desempenho do classificador.

O MINAS-AA é a versão do MINAS que utiliza aprendizado ativo. A versão original do algoritmo MINAS entende que o rótulo verdadeiro dos exemplos não estão disponíveis na fase *online*, ou seja, o modelo de decisão é atualizado sem *feedback* do usuário. Já a versão MINAS-AA usa técnicas de aprendizado ativo a fim de melhorar o desempenho do algoritmo, selecionando alguns exemplos a serem rotulados pelo especialista. A técnica de aprendizado ativo do MINAS-AA consiste em: de tempos em tempos executar

uma rotina que solicita um especialista de domínio para rotular os novos microgrupos que foram criados, a partir dos exemplos marcados como desconhecidos e enviados à memória temporária. O centroide de cada microgrupo é selecionado como elemento a ser rotulado pelo especialista.

3.4 Novas propostas para melhorar o desempenho do algoritmo MINAS

Esta seção detalha as propostas de aprendizado ativo que foram estudadas e algumas propostas que visam melhorar o desempenho do algoritmo MINAS, alterando a forma como ele classifica novos exemplos. Consequentemente essas ideias foram implementadas no algoritmo MINAS.

3.5 Novas propostas para a execução do aprendizado ativo

Proposta 1: consiste em, na fase *online*, ao se identificar que um novo grupo válido é criado, invocar imediatamente uma rotina que chama um usuário especialista de domínio que irá rotular o centroide deste grupo. Essa proposta, embora execute várias requisições ao especialista, tem como objetivo rotular o novo microgrupo o mais rápido possível, de forma a usá-lo na classificação de outros exemplos.

Objetivo: analisar se o atraso ao rotular um novo microgrupo traz algum impacto no desempenho do algoritmo e analisar o custo/benefício entre o aumento do número de requisições do especialista para rotular novos exemplos e o desempenho do algoritmo.

Proposta 2: consiste em, de tempos em tempos verificar quais novos microgrupos foram criados e marcados como novidade e solicitar o usuário especialista para rotular esses microgrupos. Os grupos marcados como extensão das classes conhecidas do problema não serão rotulados pelo especialista.

Objetivo: diminuir o número de requisições feitas ao especialista e verificar se o desempenho continua o mesmo, ou seja, verificar se o algoritmo possui um bom método para identificar as extensões dos conceitos conhecidos do problema e se precisa ser melhorado somente com relação às novidades.

3.6 Propostas visando diminuir o número de exemplos das classes novidade classificadas como classes conhecidas

Um estudo detalhado sobre os tipos de erros cometidos pelo algoritmo MINAS foi feito. Para isso foi analisado a taxa de exemplos das classes novidade incorretamente

classificados nas classes conhecidas, e a taxa de exemplos das classes conhecidas incorretamente classificados como novidade. Após esse estudo constatou-se que a maior parte dos erros do algoritmo acontecem porque ele classifica exemplos das classes novidade incorretamente em uma das classes conhecidas.

A partir de tais resultados, foram estudadas formas de tentar melhorar o desempenho do algoritmo. Para isso duas propostas foram analisadas.

Proposta 3: esta ideia propõe diminuir o tamanho do raio dos microgrupos pertencentes ao modelo de decisão. Essa proposta é executada ao tentar classificar novos elementos na fase online do algoritmo. Uma vez que o raio do microgrupo é menor, o classificador fica mais restritivo, ou seja, ele só classifica aqueles exemplos que de fato ele tem certeza, ou seja, estão bem próximo ao centroide do microgrupo. Exemplos que estão na região de fronteira entre microgrupos são adicionados à memória temporária para análise futura.

Proposta 4: o objetivo desta proposta é aumentar a janela de esquecimento usada para eliminar elementos que foram colocados na memória temporária. A ideia é dar mais tempo para que estes elementos possam formar grupos válidos e de fato serem inseridos no modelo de decisão atual. Essa janela de esquecimento também é usada para mover microgrupos que não recebem novos elementos por um longo período de tempo.

3.7 Avaliação do aprendizado ativo

Utilizar medidas de avaliação para algoritmos de classificação multiclasse ainda é uma tarefa pouco explorada na literatura. A seguir, serão apresentadas as medidas de avaliação presentes no algoritmo MINAS, as quais estão voltadas para a avaliação de desempenho, e as medidas que também foram usadas relativas à análise do número de chamadas do especialista de domínio para rotular novos exemplos e o número de exemplos rotulados.

O algoritmo MINAS foi avaliado por seus autores usando apenas medidas de desempenho, as quais são descritas a seguir.

Matriz de confusão: é uma matriz que apresenta os resultados de um processo de classificação. Nas linhas estão as classes reais do problema e nas colunas, as classes preditas pelo algoritmo, sendo que o número de linhas é o mesmo que o número de colunas. A matriz de confusão a ser utilizada no contexto de DN em FCDs deve ser incremental, pois o número de linhas e colunas não são fixos, ou seja, à medida que um padrão novidade é detectado essa matriz precisa ser reajustada, sendo que, o número de padrões novidade que são encontrados não são necessariamente iguais ao número de classes novidade. Os criadores do MINAS observaram que a matriz de confusão convencional não era adequada ao problema, então propuseram o uso de uma matriz de confusão que não é quadrada e

que cresce ao longo do tempo.

A matriz gerada permite identificar onde o algoritmo está errando mais, se é classificar classe conhecida como novidade ou novidade como classe conhecida. Cada linha da matriz de confusão é representada da seguinte forma: ACC + Err + Unk = 1, onde ACC representa a taxa de exemplos que foram corretamente classificados, ERR representa a taxa de exemplos incorretamente classificados e Unk representa a taxa de exemplos marcados como desconhecidos. Na Tabela 1, C1 e C2 representam as classes conhecidas do problema que foram aprendidas na fase *offline* do algoritmo, N1 e N2 representam as novidade encontradas na fase *online* de classificação de exemplos e CM+1 e CM+2 são as novidades observadas ao longo do fluxo que foram incorporadas ao modelo. TN é o número de exemplos corretamente classificados dentro das classes conhecidas. FE representa o erro de classificação nas classes conhecidas do problema, FP representa os elementos do conceito normal incorretamente classificados como novidade ou desconhecido, FN é o número de exemplos de uma classe Ci qualquer incorretamente classificados em uma classe Cj e em uma classe Ci qualquer e TP é o número de exemplos da classe Ci corretamente classificados.

Tabela 1 – Matriz de confusão do MINAS.

	C1	C2	...	N1	N2
C1	TN	FE	FE	FP	FP
C2	FE	TN	FE	FP	FP
...	...	...	...	...	...
CM+1	FN	FN	FN	FN	TP
CM+2	FN	FN	FN	FN	TP

CER (*Combined Error Rate*): é uma medida de avaliação multiclasse que permite expressar os erros de classificação de um algoritmo. Para calcular o CER, somente os exemplos classificados são levados em consideração, os exemplos que foram marcados como desconhecido são desconsiderados. O CER é calculado por meio da média ponderada da taxa de falsos positivos e falsos negativos de cada classe. Na Equação 3.2 o #ExCi representa o número de exemplos pertencentes a classe Ci, #Ex representa o número total de elementos.

$$CER = \frac{\sum \frac{\#ExCi}{\#Ex} FPRi + \sum \frac{\#ExCi}{\#Ex} FNRi}{2} \tag{3.2}$$

***UnkR*:** pode ser descrita como a taxa de exemplos que o sistema de aprendizagem classificou com o perfil desconhecido. Exemplos desconhecidos estão presentes na matriz de confusão, porém o MINAS não os considera acertos, nem erros, eles são computados separadamente. UnkR é a taxa que calcula a média de exemplos marcados como desconhecidos que varia com o passar do tempo. Para efetuar este cálculo, para cada classe presente no

modelo, é calculada a taxa de exemplos desconhecidos e também as médias, ver Equação 3.3. Nesta equação #ExCi representa o número de exemplos pertencentes a classe Ci e #Unki representa o número de exemplos da classe i que foi marcado como desconhecido.

$$UnkR = \frac{\sum(\frac{\#Unki}{\#ExCi})}{M} \tag{3.3}$$

Neste trabalho também serão usadas medidas de avaliação que visem não somente avaliar o desempenho do classificador, mas também avaliar quantas vezes o processo de rotular instâncias foi chamado e quantas instâncias foram rotuladas pelo especialista. Assim, é possível comparar o desempenho do classificador versus o número de instâncias rotuladas. A seguintes medidas serão usadas:

- Quantidade de exemplos que vão para o buffer (nroExMovidosBuffer);
- Quantidade de grupos válidos (QtdGruposValidos);
- Quantas vezes o processo de aprendizado ativo foi disparado (QtdDisparoAA);
- Ao disparar o aprendizado ativo, qual a quantidade de grupos para rotular (QtdGruposRotulados);
- Em média, quantos grupos foram rotulados (mediaGruposRotulados);

3.8 Considerações finais

Neste capítulo foi descrito em detalhes as duas fases presentes no algoritmo MINAS, também foi apresentada a ideia de DN em FCDs e em qual fase esta é realizada. Foi explicada a proposta inicial de aprendizagem ativa e mais propostas utilizando este método foram feitas para melhorar a acurácia do algoritmo. Foram descritas as medidas de avaliação utilizadas pelo algoritmo MINAS, as quais foram desenvolvidas para DN em FCDs multiclasse.

O capítulo 4 apresentará os resultadas dos experimentos obtidos usando as propostas que foram apresentadas no capítulo 3, bem como a descrição de cada uma das bases usadas.

4 Experimentos

4.1 Introdução

No capítulo anterior foi descrito como é feito o processo de DN no MINAS utilizando AA. Também foram apresentados as novas propostas de aprendizagem ativa implementadas neste trabalho, bem como as medidas de avaliação a serem usadas nos experimentos. Este capítulo descreve as características de cada base de dados utilizadas, os experimentos e a análise dos resultados referentes as técnicas de AA anteriormente descritas e implementadas no MINAS.

4.2 Bases de dados utilizadas

Os experimentos realizados para as propostas feitas neste trabalho, foram realizados utilizando 7 bases de dados diferentes, entre elas estão bases reais e artificiais. A Tabela 2 mostra um resumo das principais características que compõem cada uma destas bases.

Tabela 2 – Resumo das bases de dados utilizadas nos experimentos.

Base de dados	Atributos	Instâncias	Classes	Classes no treino	Tipo de base
MOA	4	100.000	4	2	artificial
SynD	10	250.000	2	2	artificial
SynEDC	40	400.000	20	7	artificial
KDD	34	490.000	5	2	real
CoverType	54	581.000	7	3	real

É importante salientar que, as classes de treino que são descritas na Tabela 2, representam a quantidade de classes usadas para construir o modelo inicial. Após a execução do algoritmo, com o passar do tempo, novas classes são encontradas e incorporadas ao modelo de decisão.

4.2.1 Bases de dados artificiais

Dentre as bases de dados artificias utilizadas neste trabalho, estão: MOA (PAIVA, 2014), SynD (MASUD et al., 2011), (AL-KHATEEB et al., 2012a) e SynEDC (MASUD et al., 2011), (AL-KHATEEB et al., 2012a).

MOA: A base de dados MOA foi criada para simular as caracteristicas de FCDs, como mudança de conceito, aparecimento e desaparecimento de classes. Esta base foi gerada utilizando uma ferramenta chamada MOA – *Massive Online Analysis* (BIFET et al., 2010).

Tal ferramenta permite definir quais os parâmetros serão utilizados na criação de uma base. Para a criação da base de dados MOA, as seguintes configurações foram definidas: um número fixo de centroides, a posição e o rótulo do centroide (gerado aleatoriamente), também o centro de cada microgrupo possui o mesmo desvio padrão (raio). O gerador foi configurado para aparecer no máximo 2 classes ao longo do fluxo e a cada 30.000 exemplos ocorre um evento de aparecimento e/ou desaparecimento de classes. Esta base é composta por 100.000 exemplos.

SynD - *Synthetic data with only concept-drift*: Esta é uma base que possui 250.000 instâncias, composta somente por duas classes e simula apenas uma mudança de conceito, ou seja, não tem o aparecimento de novas classes. Cada exemplo que compõe a base possui um vetor de 10 características, sendo que cada uma delas está no intervalo entre 0 e 1. Os ruídos são introduzidos de forma aleatória e mudam o rótulo em cerca de 5% das instâncias.

SynEDC - *Synthetic data with concept-drift and novel-class*: Base que é composta por 400.000 instâncias, 20 classes com 40 atributos, os valores de cada atributo estão no intervalo [0,1]. Esta base faz a simulação de uma mudança de conceito e o aparecimento de novos classes ao longo do fluxo. Sua distribuição de probabilidade varia para que algumas classes apareçam e desapareçam com o passar do tempo. Para simular uma mudança de conceito, os valores médios de uma certa quantidade de atributos são modificados constantemente.

4.2.2 Bases de dados reais

Dentre as bases de dados reais que este trabalho utiliza, estão: KDD (STOLFO, 1999), e Covertype (DEAN, 1999).

KDD – Detecção de intrusão: Esta base é utilizada para detectar intrusos em uma rede de computadores, contendo 490.000 instâncias, uma versão resumida da base original. A KDD (KDDTe5Classes) simula um problema real de ciberataques, onde cada exemplo representa uma conexão na rede que pode ser classificado como normal ou como um dos 4 tipos diferentes de ataque. Cada exemplo contém 42 atributos, porém neste trabalho somente 34 foram usados e cada atributo tem seu valor variando de 0 a 1. Uma outra versão da base KDD (KDDTe5ClassesSoNormal) é utilizada, para esta base somente exemplos da classe normal compõem a fase de treinamento.

Covertype – Cobertura de floresta: Esta base prediz o tipo de cobertura de floresta a partir de informações como elevação, declive, tipo de solo, etc. A Covertype contém registros de diferentes tipos de cobertura. Os exemplos são compostos por 54 atributos numéricos de valores entre 0 e 1. A base contém 581.000 instâncias. Uma outra base utilizada é a fcTe, esta é uma versão modificada da base Covertype. Neste caso os dados

foram rearranjados para que em cada janela do fluxo ocorra no máximo 3 e no mínimo 2 classes.

Uma técnica de amostragem de dados é aplicada em cada uma das bases de dados. Cerca de 10% dos dados presentes na base são utilizados na fase de treinamento (*offline*), os outros 90% são utilizados para teste, fase *online* do algoritmo.

4.3 Experimentos

Esta seção irá apresentar os resultados dos experimentos que foram propostos por este trabalho.

Experimento 1: este experimento analisa os resultados da matriz de confusão,descrita na seção 3.7, na execução do MINAS-AA original. Seu objetivo é comparar a versão original do MINAS com as novas versões propostas e analisar a matriz de confusão gerada para cada base, identificando os pontos fracos do algoritmo.

Experimento 2: este experimento se refere a proposta 1 já descrita na seção 3.5. Ao se identificar um microgrupo válido aplica-se o aprendizado ativo, ou seja, o especialista de domínio é solicitado a rotular o centroide do novo microgrupo assim que ele é criado. O objetivo deste experimento é verificar se a espera para rotular um novo microgrupo criado tem impacto no desempenho do algoritmo.

Experimento 3: este experimento se refere a proposta 2 já descrita na seção 3.5. A proposta segue a ideia original de aprendizado ativo do MINAS, o que muda é a quantidade de elementos que serão rotulados pelo especialista. Para este experimento, somente os microgrupos, identificados pelo MINAS como novidade, serão rotulados.

Experimento 4: este experimento se refere a proposta 3 já descrita na seção 3.6 . Esta proposta foi implementada utilizando o MINAS original. O raio usado para identificar a qual microgrupo o exemplo pertence é modificado. Dois experimentos foram realizados reduzindo o tamanho do raio. No primeiro experimento, o tamanho do raio é reduzido em 20% com relação ao seu tamanho original. No segundo experimento, o raio é reduzido em 40% do seu tamanho.

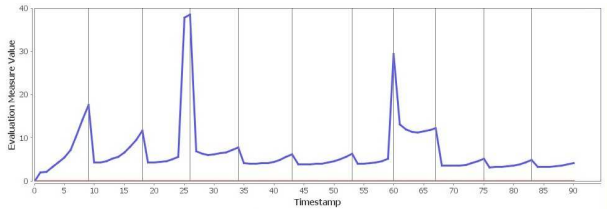
Experimento 5: este experimento se refere a proposta 4 já descrita na seção 3.6. Utilizando o MINAS original neste experimento, o tamanho da janela de esquecimento de elementos foi incrementada. No MINAS original a janela começa com 4000. Para analisar como o algoritmo se comporta, no primeiro experimento a janela é setada para o valor de 10.000 e no segundo para 20.000.

4.4 Resultados

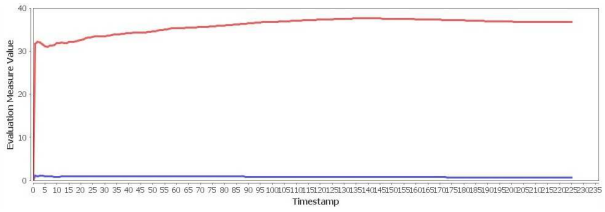
Esta seção apresenta as comparações dos resultados do MINAS Original e os resultados das abordagens propostas. Os objetos de análise são os gráficos e as matrizes de confusão geradas ao final da execução do algoritmo.

4.4.1 Minas Original

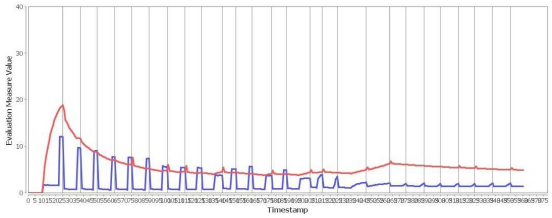
A Figura 2 representa o resultado do MINAS original para cada base. Os gráficos ilustrados e a Tabela 3 serão adotados como referência na comparação das abordagens propostas em relação à versão original do MINAS-AA. Para este trabalho a medida FMacro foi desconsiderada.



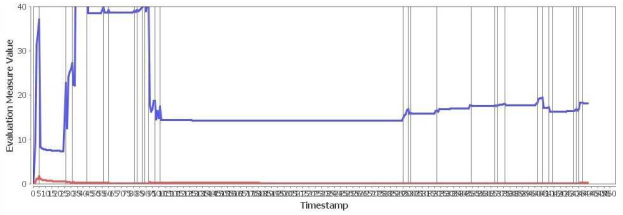
(a) MOA



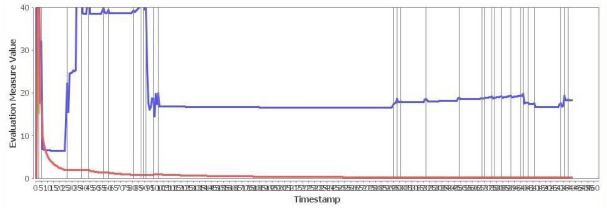
(b) SynD



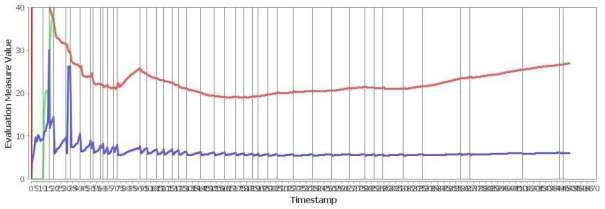
(c) SynEDC20D40Norm



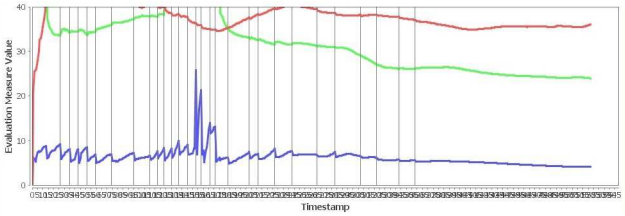
(d) KDDTe5Classes



(e) KDDTe5ClassesSoNormal



(f) fcTe



(g) Covertype

Figura 2 – Gráficos gerados a partir do MINAS Original para cada base de dados.

Tabela 3 – Resultado das medidas de avalição Minas original.

Base de dados	Qtd de exemplos que vão para o buffer	Qtd de grupos válidos	Nº de chamadas AA	Quantas vezes o AA foi executado	Média de grupos rotulados
MOA3	20.471	171	90	10	17,1
SynD	1.489	0	225	0	0
SynEDC20D40Norm	54.355	28	360	26	1,07
KDDTe5Classes	59.470	273	442	29	9,41
KDDTe5ClassesSoNormal	83.318	272	442	31	8,77
fcTe	99.053	976	450	43	22,69
covtypeOrigNorm	77.870	656	522	28	23,42

4.4.2 Experimento 1

Este experimento foi executado na versão original do MINAS-AA com a intenção de identificar onde estão os maiores erros do algoritmo. A análise verifica se os erros aconteciam porque o MINAS classifica incorretamente exemplos das classes já conhecidas como novidade ou se o erro está em classificar novidade como classes conhecidas do problema. Esse experimento foi importante para direcionar quais modificações no MINAS seriam importantes e como o aprendizado ativo poderia auxiliar.

Para analisar as matrizes de confusão é necessário identificar em cada tabela o que são classes conhecidas e o que são novidades. Para a matriz de confusão CoverType as classes conhecidas são *C1* e *C2*. As novidades são *C3, C4, C5 e C6*.

Para a matriz de consfusão KDDTe5Classes, as classes conhecidas são *C dos* e *C Normal*. As demais classes *C r2l*, *C probe* e *C u2r* são novidades.

Tabela 4 – Matriz de confusão CoverType MINAS original.

	C 1	C 2		C 3	C 4	C 5	C 6	C 7
C 1	69,18%	21,66%		0%	0%	0,01%	0,02%	0,21%
C 2	34,86%	58,73%		0%	0%	0,23%	0,24%	0,03%
C 3	1,13%	98,70%		0%	0%	0%	0%	0%
C 4	0%	100%		0%	0%	0%	0%	0%
C 5	15,20%	65,76%		0%	0%	13,48%	0,57%	0%
C 6	7,40%	90,76%		0%	0%	0%	1,33%	0%
C 7	78,31%	5,64%		0%	0%	0%	0%	7,87%

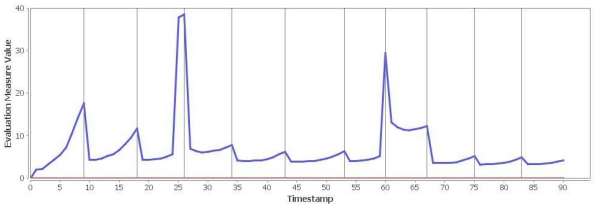
Tabela 5 – Matriz de confusão KDDTe5Classes, MINAS original.

	C dos	C Normal		C r2l	C probe	C u2r
C dos	98,13%	0,02%		0,001%	0,02%	0%
C normal	0,31%	89,08%		0,11%	0,12%	0%
C r2l	0%	10,80%		83%	0,94%	0%
C probe	0%	1,83%		0,08%	88,74%	0%
C u2r	0%	25,53%		10,64%	0%	0%

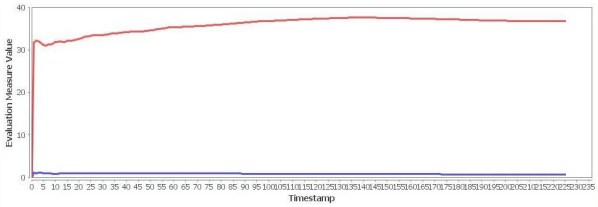
Ao analisar as matrizes de confusão, percebe-se que o algoritmo está errando mais em classificar novidade como classe conhecida do problema. Esses resultados foram identificados para todas as bases de dados testadas.

4.4.3 Experimento 2

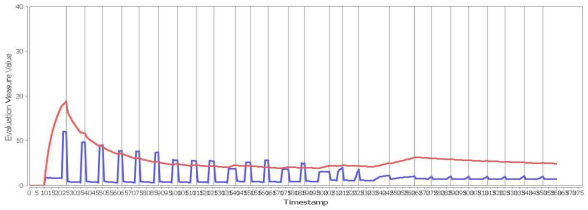
A Figura 3 apresenta o resultado dos gráficos do experimento 2 para cada base.



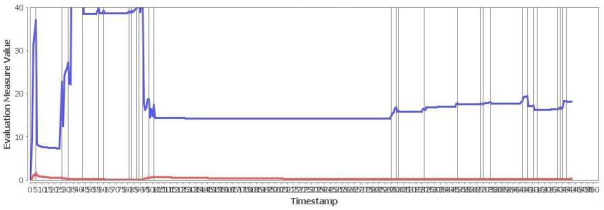
(a) MOA



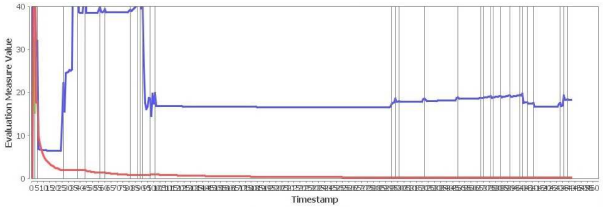
(b) SynD



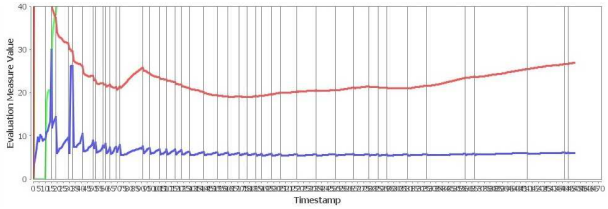
(c) SynEDC



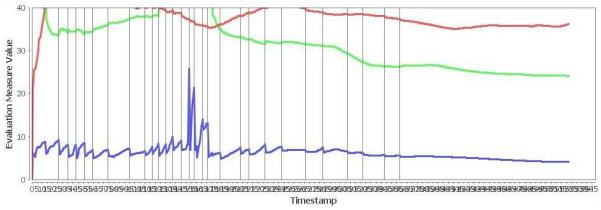
(d) KDDTe5



(e) KDDTe5Normal



(f) fcTe



(a) Covertype

Figura 3 – Gráficos gerados a partir do experimento 2 para cada base de dados.

Tabela 6 – Resultado das medidas de avalição experimento 2.

Base de dados	Qtd de exemplos que vão para o buffer	Qtd de grupos válidos	N° de chamadas AA	Quantas vezes o AA foi executado	Média de grupos rotulados
MOA3	20.471	171	171	171	1
SynD	1.489	0	0	0	0
SynEDC20D40Norm	54.355	28	28	28	1
KDDTe5Classes	59.470	273	273	273	1
KDDTe5ClassesSoNormal	83.118	272	272	272	1
fcTe	99.053	976	976	976	1
covtypeOrigNorm	77.870	656	656	656	1

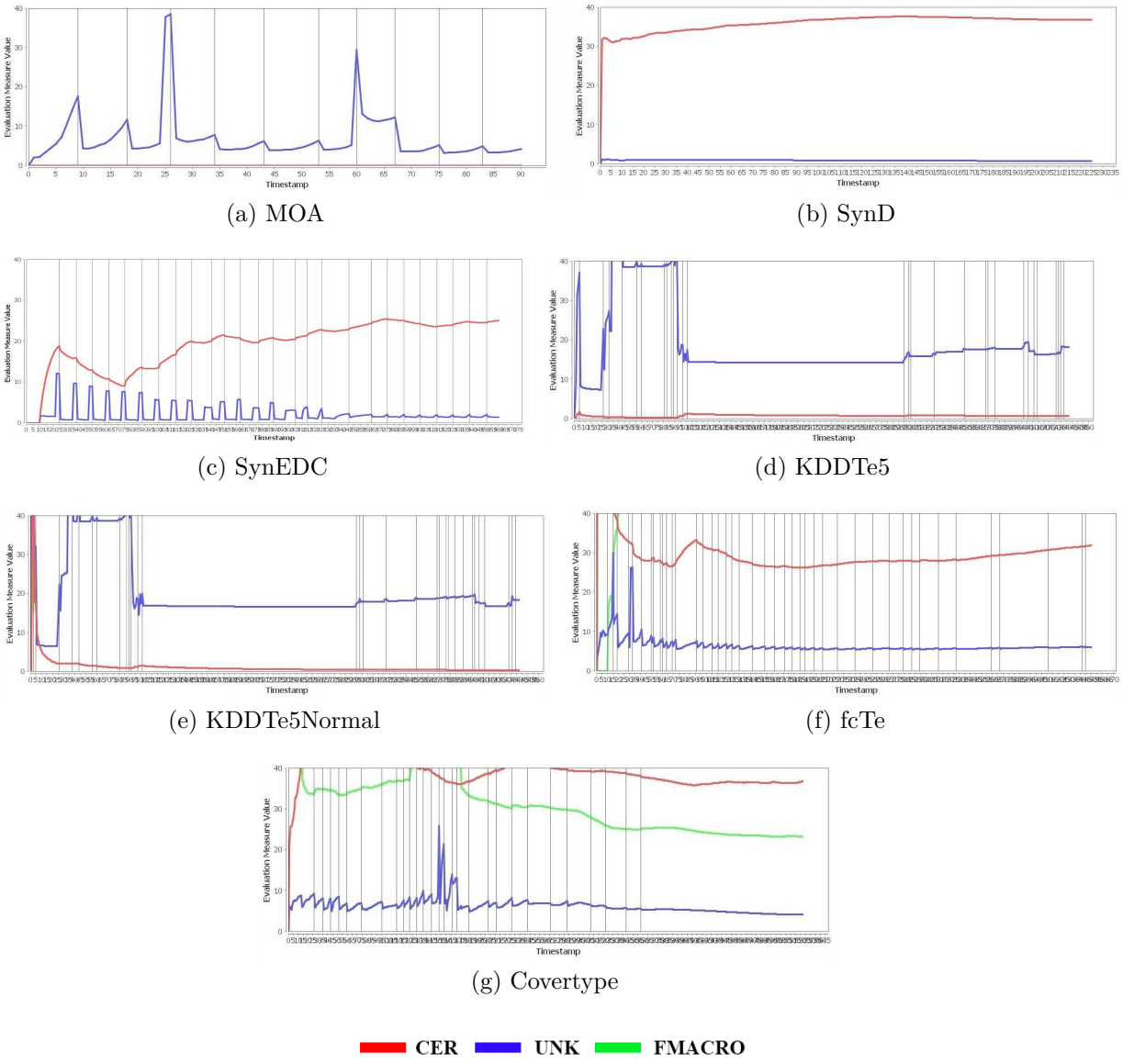
Este experimento tem como objetivo analisar o impacto na mudança da forma como o aprendizado ativo é chamado, ou seja, ao invés de ser executado de tempos em tempos, é executado sempre que um novo micro-grupo é validado. A partir da análise da Figura 3, pode-se perceber que em termos de desempenho não houve mudanças significativas. A segunda análise considerou a Tabela 6 em comparação com a Tabela 3, a fim de identificar os impactos que as chamadas de AA proporcionam ao algoritmo para cada base de dados.

Ao analisar as tabelas, observa-se que o número de chamadas de AA diminuiu consideravelmente para as bases de dados SynD, SynEDC20D40Norm, KDDTe5Classes e KDDTe5ClassesSoNormal com relação ao MINAS original. Para as bases MOA, fcTe e covtypeOrigNorm houve um aumento no número de chamadas de AA.

Observando os resultados do experimento 1, é possível concluir que, para as bases em que as chamadas ao especilista de domínio diminuiu, houve resultados relevantes, visto que anteriormente o especialista era invocado e não tinha grupos a serem rotulados.

4.4.4 Experimento 3

A Figura 4 apresenta o resultado dos gráficos do experimento 3 para cada base.



(a) MOA (b) SynD

(c) SynEDC (d) KDDTe5

(e) KDDTe5Normal (f) fcTe

(g) Covertype

Figura 4 – Gráficos gerados a partir do experimento 3 para cada base de dados.

Tabela 7 – Resultado das medidas de avalição experimento 3.

Base de dados	Qtd de exemplos que vão para o buffer	Qtd de grupos válidos	N° de chamadas AA	Quantas vezes o AA foi executado	Média de grupos rotulados
MOA3	20.471	171	90	10	4,5
SynD	1.489	0	225	0	0
SynEDC20D40Norm	54.355	28	360	8	1
KDDTe5Classes	59.470	273	442	24	4,08
KDDTe5ClassesSoNormal	83.118	272	442	22	4,68
fcTe	99.053	976	450	43	12,65
covtypeOrigNorm	77.870	656	522	28	15,17

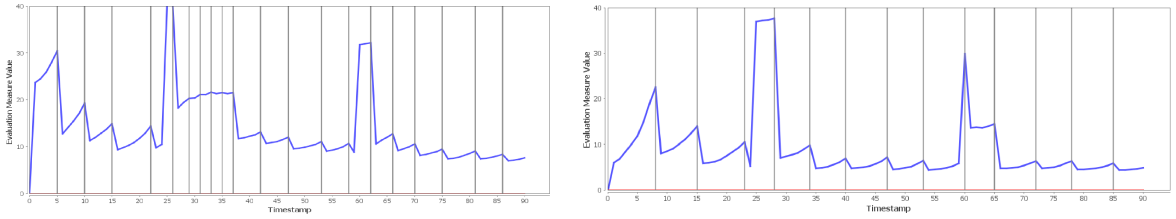
O objetivo deste experimento é analisar o impacto nos resultados do algoritmo caso apenas grupos que foram marcados como novidade fossem rotulados. A partir da análise da Figura 4, percebe-se que houve uma piora no desempenho do algoritmo. O gráfico que representa a base SynEDC nos mostra que a medida CER aumentou consideravelmente, ou seja, o algoritmo está errando mais ao classificar os elementos. Para as demais bases os resultados também foram negativos, porém em proporções menores. Comparando as tabelas 7 e 3, observa-se que a quantidade de chamadas do AA é a mesma, porém para este experimento houve uma queda na quantidade de grupos rotulados para as bases SynEDC, KDDTe5Classes e KDDTe5ClassesSoNormal.

É possível concluir que rotular somente os exemplos novidade e não rotular as extensões, impactou negativamente o algoritmo em termos de desempenho.

4.4.5 Experimento 4: Raio

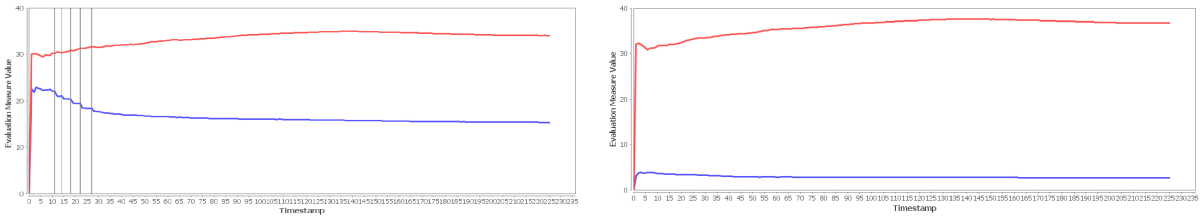
Uma vez que o experimento 1, identificou que os maiores erros do algoritmo MINAS-AA estão em classificar novidade como classe conhecida do problema, este experimento visa tornar o processo de classificação mais restritivo, marcando como desconhecido os exemplos que estão na região de fronteira a fim de que eles possam ser rotulados pelo especialista.

A Figura 11 apresenta os resultados dos gráficos para as bases analisadas.



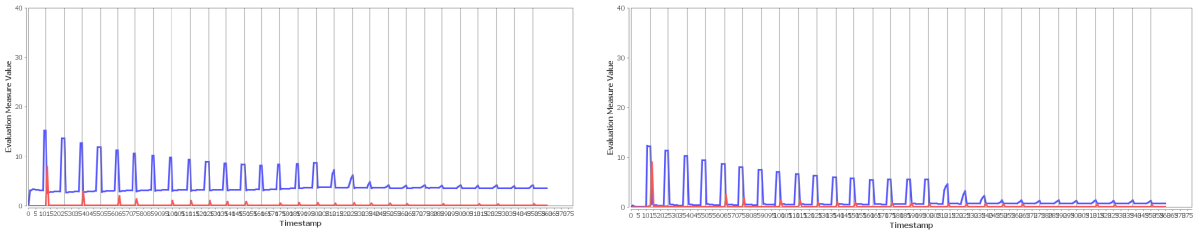
(a) MOA 60% do raio (b) MOA 80% do raio

Figura 5 – Gráficos modificando o tamanho do raio, base MOA.



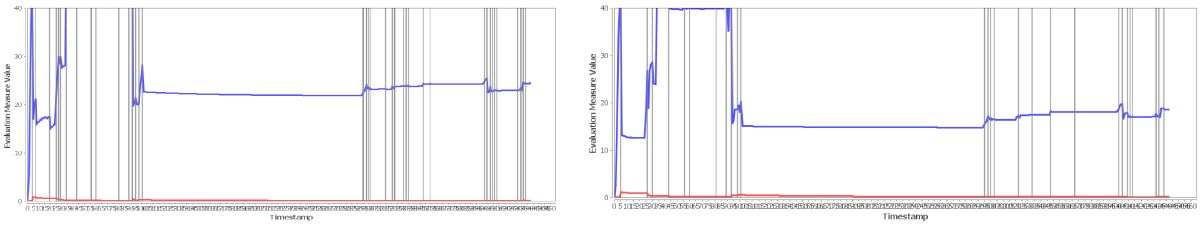
(a) SynD 60% do raio (b) SynD 80% do raio

Figura 6 – Gráficos modificando o tamanho do raio, base SynD.



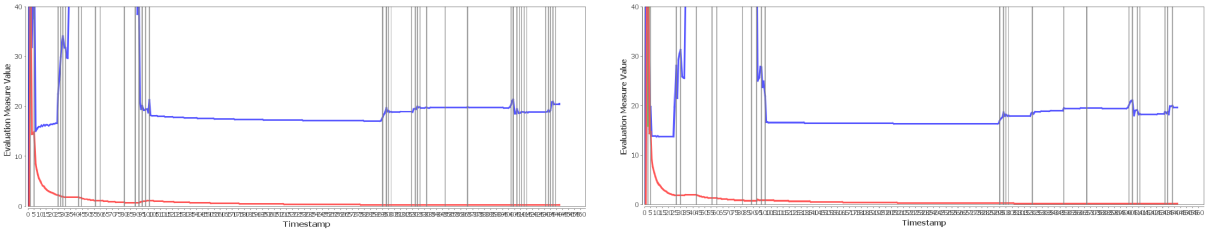
(a) SynEDC20D40Norm 60% do raio (b) SynEDC20D40Norm 80% do raio

Figura 7 – Gráficos modificando o tamanho do raio, base SynEDC.



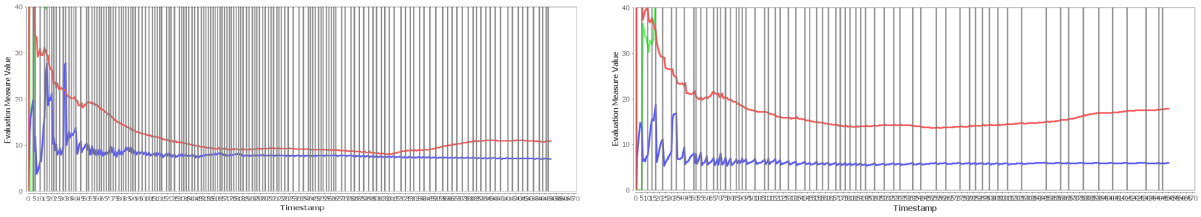
(a) KDDTe5Classes 60% do raio (b) KDDTe5Classes 80% do raio

Figura 8 – Gráficos modificando o tamanho do raio, base KDDTe5Classes.



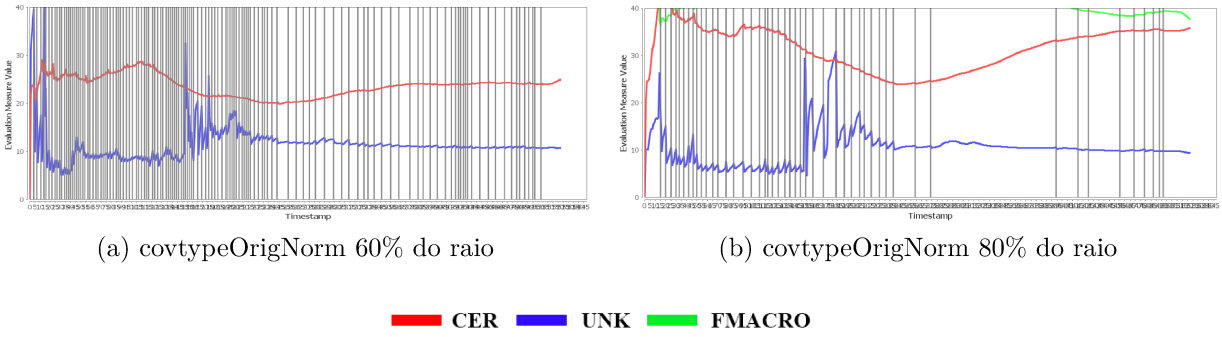
(a) KDDTe5ClassesSoNormal 60% do raio (b) KDDTe5ClassesSoNormal 80% do raio

Figura 9 – Gráficos modificando o tamanho do raio, base KDDTe5ClassesSoNormal.



(a) fcTe 60% do raio (b) fcTe 80% do raio

Figura 10 – Gráficos modificando o tamanho do raio, base fcTe.



(a) covtypeOrigNorm 60% do raio (b) covtypeOrigNorm 80% do raio

Figura 11 – Gráficos modificando o tamanho do raio, base CoverType.

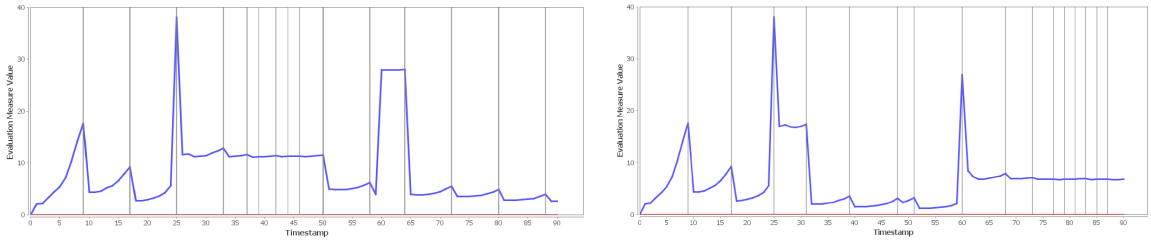
Para este experimento observou-se que ao diminuir o tamanho do raio, a taxa de elementos desconhecidos aumentou. A taxa de elementos identificados como desconhecidos pelo algoritmo é inversamente proporcional ao tamanho do raio. Percebe-se, ainda, uma queda na taxa de erros do algoritmo (CER). Esses resultados são identificados para todas as bases testadas.

Conclui-se que ao ser mais seletivo o algoritmo erra menos na classificação. Ao enviar os elementos para a memória a temporária, depois de um certo período de tempo, mais elementos com características semelhantes estarão presentes. Isso aumenta a possibilidade de formar mais grupos válidos que serão incorporados ao modelo de decisão.

4.4.6 Experimento 5: Janela

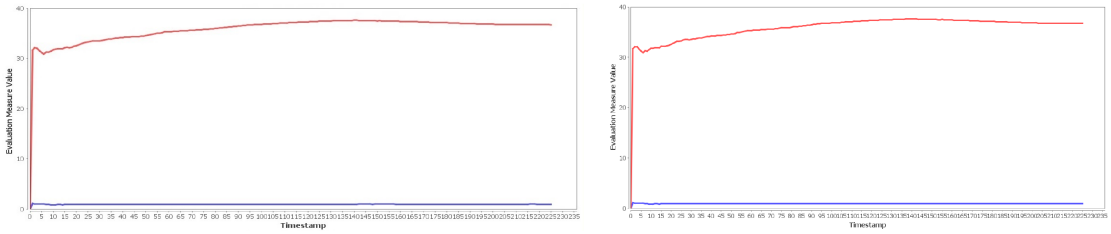
Outro ponto a ser analisado é o impacto que o tamanho da janela usada no MINAS provoca no seu desempenho. Esse experimento teve como objetivo dar mais tempo para que os elementos da memória temporária formassem grupos válidos.

O tamanho da janela é usado pelo MINAS para aumentar o período de tempo que os elementos permanecem na memória temporária e tratar a detecção de contextos recorrentes. Esta última abordagem não será discutida neste trabalho. A Figura 18 apresenta os resultados destes experimentos.



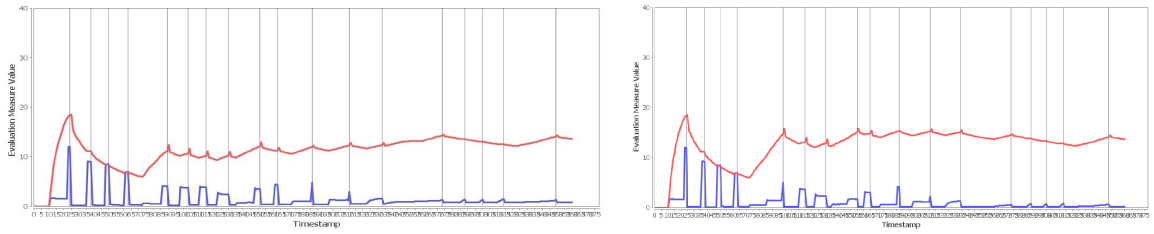
(a) MOA janela 10000 (b) MOA janela 20000

Figura 12 – Gráficos modificando o tamanho da janela, base MOA.



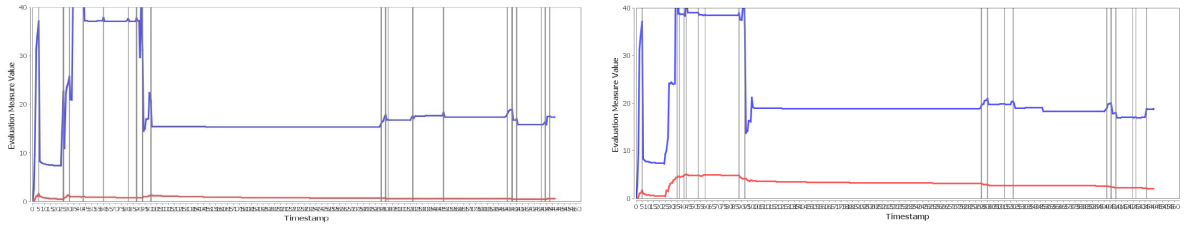
(a) SynD janela 10000 (b) SynD janela 20000

Figura 13 – Gráficos modificando o tamanho da janela, base SynD.



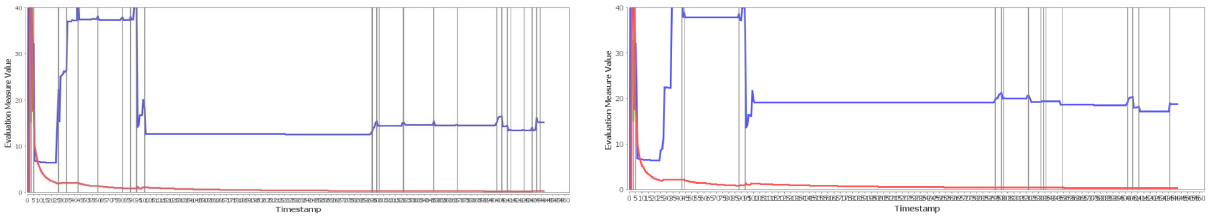
(a) SynEDC20D40Norm janela 10000 (b) SynEDC20D40Norm janela 20000

Figura 14 – Gráficos modificando o tamanho da janela, base SynEDC.



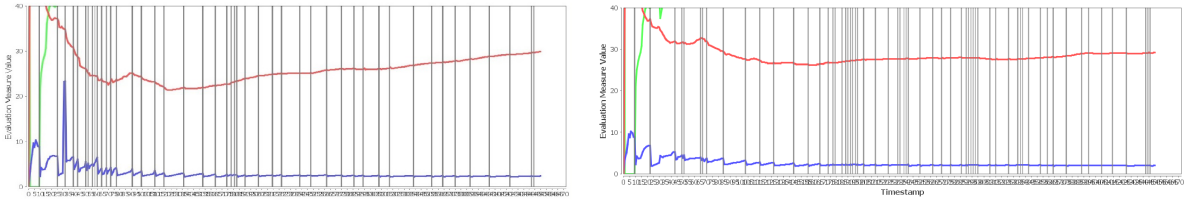
(a) KDDTe5Classes janela 10000 (b) KDDTe5Classes janela 20000

Figura 15 – Gráficos modificando o tamanho da janela, base KDDTe5Classes.



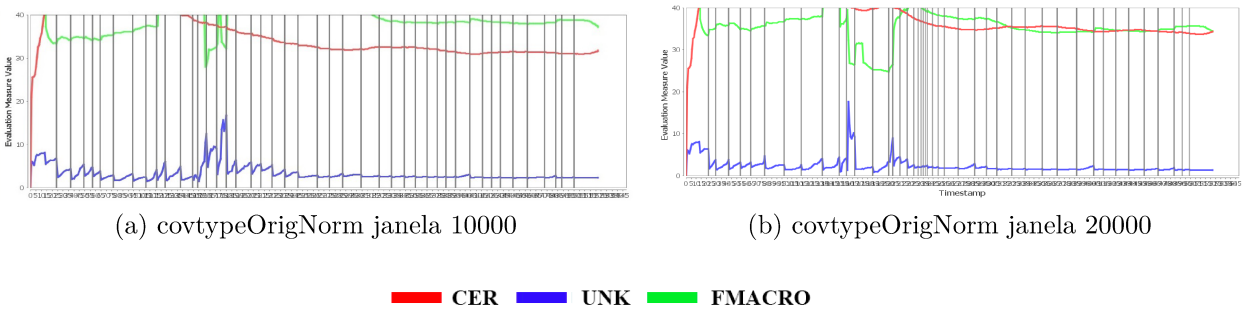
(a) KDDTe5ClassesSoNormal janela 10000 (b) KDDTe5ClassesSoNormal janela 20000

Figura 16 – Gráficos modificando o tamanho da janela, base KDDTe5ClassesSoNormal.



(a) fcTe janela 10000 (b) fcTe janela 20000

Figura 17 – Gráficos modificando o tamanho da janela, base fcTe.



(a) covtypeOrigNorm janela 10000 (b) covtypeOrigNorm janela 20000

Figura 18 – Gráficos modificando o tamanho da janela, base CoverType.

Os resultados observados neste experimento são contrários ao anterior. Com o aumento da janela, a taxa de erros de classificação (CER) também aumenta, diminuindo o desempenho do algoritmo.

Pode-se concluir que, aumentar o tempo em que os elementos permanecem na memória temporária afeta negativamente o desempenho do MINAS.

4.4.7 Considerações finais

Este capítulo apresentou um conjunto de experimentos elaborados para serem executados no algoritmo MINAS. Os experimentos foram realizados utilizando a metodologia de aprendizagem ativa aplicada a fluxos contínuos de dados e testados em 7 bases de dados diferentes, sendo elas reais e artificias.

As análises dos resultados mostram que a metodologia empregada têm potencial para resolver problemas relacionados a classificação em cenários dinâmicos. O próximo capítulo relata as principais contribuições deste trabalho e ideias para possíveis trabalhos futuros.

5 Conclusão

Este trabalho apresentou novas ideias para melhorar o processo de classificação em um algoritmo específico da literatura, o MINAS. Foi realizado um estudo dos principais algoritmos que trabalham com FCDs e, ainda, como tais algoritmos tratam os problemas de classificação nesse tipo de cenário. Estudou-se também, a abordagem de aprendizagem ativa como uma metodologia para se melhorar o desempenho do algoritmo MINAS ao se classificar novas instâncias. Novas propostas de aprendizagem ativa foram desenvolvidas e implementadas.

5.1 Contribuições

Este trabalho proporcionou as seguintes contribuições para a literatura:

- Avaliar o comportamento do MINAS ao variar as suas configurações;
- Encontrar as maiores deficiências do algoritmo e o que poderia ser feito para melhorar;
- Novas propostas de aprendizado ativo para o MINAS;
- Novas medidas que avaliam o aprendizado ativo.

5.2 Trabalhos futuros

Diante dos resultados obtidos, novas ideias surgiram para melhorar o desempenho do MINAS e podem ser consideradas na elaboração de trabalhos futuros.

- Testar mais bases utilizando as implementações de aprendizado ativo realizadas;
- Realizar novos experimentos;
- Propor novas implementações de AA;
- Aplicar as abordagens desenvolvidas em problemas presentes no mundo real.

Referências

ABDALLAH, Z. S. et al. Anynovel: detection of novel concepts in evolving data streams. *Evolving Systems*, Springer, v. 7, n. 2, p. 73–93, 2016. Citado 5 vezes nas páginas 12, 17, 19, 20 e 22.

AGGARWAL, C. C. *Data streams: models and algorithms*. [S.l.]: Springer Science & Business Media, 2007. v. 31. Citado 2 vezes nas páginas 10 e 18.

AGGARWAL, C. C. et al. A framework for clustering evolving data streams. In: VLDB ENDOWMENT. *Proceedings of the 29th international conference on Very large data bases-Volume 29*. [S.l.], 2003. p. 81–92. Citado 2 vezes nas páginas 10 e 24.

AL-KHATEEB, T. et al. Stream classification with recurring and novel class detection using class-based ensemble. In: IEEE. *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. [S.l.], 2012. p. 31–40. Citado 7 vezes nas páginas 11, 12, 15, 18, 20, 21 e 31.

AL-KHATEEB, T. M. et al. Cloud guided stream classification using class-based ensemble. In: IEEE. *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. [S.l.], 2012. p. 694–701. Citado na página 18.

ALBERTINI, M. K.; MELLO, R. F. de. A self-organizing neural network for detecting novelties. In: ACM. *Proceedings of the 2007 ACM symposium on Applied computing*. [S.l.], 2007. p. 462–466. Citado na página 12.

BABCOCK, B. et al. Models and issues in data stream systems. In: ACM. *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. [S.l.], 2002. p. 1–16. Citado na página 15.

BERRY, M. J.; LINOFF, G. *Data mining techniques: for marketing, sales, and customer support*. [S.l.]: John Wiley & Sons, Inc., 1997. Citado na página 10.

BIFET, A. et al. Moa: Massive online analysis. *Journal of Machine Learning Research*, v. 11, n. May, p. 1601–1604, 2010. Citado na página 31.

CHANG, J. H.; LEE, W. S. estwin: Online data stream mining of recent frequent itemsets by sliding window method. *Journal of Information Science*, Sage Publications Sage CA: Thousand Oaks, CA, v. 31, n. 2, p. 76–90, 2005. Citado na página 10.

CHERMAN, E. A. *Aprendizado de máquina multirrótulo: explorando a dependência de rótulos e o aprendizado ativo*. Tese (Doutorado) — Universidade de São Paulo, 2013. Citado na página 11.

DEAN, D. *CoverType: Forest CoverType*. 1999. Disponível em: <https://kdd.ics.uci.edu/databases/covertype/covertype.data.html>. Citado na página 32.

DOMINGOS, P.; HULTEN, G. Mining high-speed data streams. In: ACM. *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2000. p. 71–80. Citado na página 10.

ELWELL, R.; POLIKAR, R. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, IEEE, v. 22, n. 10, p. 1517–1531, 2011. Citado 2 vezes nas páginas 11 e 16.

FARID, D. M.; RAHMAN, C. M. Novel class detection in concept-drifting data stream mining employing decision tree. In: IEEE. *Electrical & Computer Engineering (ICECE), 2012 7th International Conference on*. [S.l.], 2012. p. 630–633. Citado 3 vezes nas páginas 11, 12 e 18.

FARID, D. M. et al. An adaptive ensemble classifier for mining concept drifting data streams. *Expert Systems with Applications*, Elsevier, v. 40, n. 15, p. 5895–5906, 2013. Citado 2 vezes nas páginas 11 e 12.

GAMA, J. *Knowledge discovery from data streams*. [S.l.]: CRC Press, 2010. Citado 3 vezes nas páginas 10, 11 e 18.

GAMA, J.; GABER, M. M. *Learning from data streams: processing techniques in sensor networks*. [S.l.]: Springer, 2007. Citado na página 10.

GUHA, S.; MISHRA, N. Clustering data streams. In: *Data Stream Management*. [S.l.]: Springer, 2016. p. 169–187. Citado na página 16.

HAYAT, M. Z.; HASHEMI, M. R. A dct based approach for detecting novelty and concept drift in data streams. In: IEEE. *Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of*. [S.l.], 2010. p. 373–378. Citado 2 vezes nas páginas 12 e 18.

IENCO, D.; ŽLIOBAITĖ, I.; PFAHRINGER, B. High density-focused uncertainty sampling for active learning over evolving stream data. In: *Proceedings of the 3rd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. [S.l.: s.n.], 2014. p. 133–148. Citado na página 20.

KLINKENBERG, R. Using labeled and unlabeled data to learn drifting concepts. In: *Workshop notes of the IJCAI-01 Workshop on Learning from Temporal and Spatial Data*. [S.l.: s.n.], 2001. p. 16–24. Citado na página 20.

KRAWCZYK, B.; WOŹNIAK, M. Incremental learning and forgetting in one-class classifiers for data streams. In: SPRINGER. *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*. [S.l.], 2013. p. 319–328. Citado 2 vezes nas páginas 11 e 12.

LLOYD, S. Least squares quantization in pcm. *IEEE transactions on information theory*, IEEE, v. 28, n. 2, p. 129–137, 1982. Citado na página 24.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, n. 14, p. 281–297. Citado na página 24.

MAHDIRAJI, A. R. Clustering data stream: A survey of algorithms. *International Journal of Knowledge-based and Intelligent Engineering Systems*, IOS Press, v. 13, n. 2, p. 39–44, 2009. Citado na página 16.

MASUD, M. et al. Classification and novel class detection in concept-drifting data streams under time constraints. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 23, n. 6, p. 859–874, 2011. Citado 7 vezes nas páginas 11, 12, 15, 18, 20, 21 e 31.

MASUD, M. M. et al. Addressing concept-evolution in concept-drifting data streams. In: IEEE. *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. [S.l.], 2010. p. 929–934. Citado 3 vezes nas páginas 11, 12 e 19.

PAIVA, E. R. d. F. *Detecção de novidade em fluxos contínuos de dados multiclasse*. Tese (Doutorado) — Universidade de São Paulo, 2014. Citado 12 vezes nas páginas 10, 12, 13, 15, 16, 17, 18, 19, 20, 22, 24 e 31.

PARK, C. H.; SHIM, H. Detection of an emerging new class using statistical hypothesis testing and density estimation. *International Journal of Pattern Recognition and Artificial Intelligence*, World Scientific, v. 24, n. 01, p. 1–14, 2010. Citado na página 11.

PIMENTEL, M. A. et al. A review of novelty detection. *Signal Processing*, Elsevier, v. 99, p. 215–249, 2014. Citado na página 18.

ROKACH, L.; MAIMON, O. Z. *Data mining with decision trees: theory and applications. Volume 69 of Series in machine perception and artificial intelligence*. [S.l.]: World Scientific Press, 2008. Citado na página 17.

SETTLES, B. Active learning literature survey. *University of Wisconsin, Madison*, v. 52, n. 55-66, p. 11, 2010. Citado 2 vezes nas páginas 13 e 19.

SILVA, J. A. et al. Data stream clustering: A survey. *ACM Computing Surveys (CSUR)*, ACM, v. 46, n. 1, p. 13, 2013. Citado na página 16.

SPINOSA, E. J.; CARVALHO, A. P. de Leon F de; GAMA, J. Cluster-based novel concept detection in data streams applied to intrusion detection in computer networks. In: ACM. *Proceedings of the 2008 ACM Symposium on Applied computing*. [S.l.], 2008. p. 976–980. Citado 2 vezes nas páginas 10 e 21.

SPINOSA, E. J. et al. Novelty detection with application to data streams. *Intelligent Data Analysis*, IOS Press, v. 13, n. 3, p. 405–422, 2009. Citado 5 vezes nas páginas 12, 15, 18, 20 e 24.

STEINWART, I.; CHRISTMANN, A. *Support vector machines*. [S.l.]: Springer Science & Business Media, 2008. Citado na página 17.

STOLFO, C. *KDD: Intrusion detector learning*. 1999. Disponível em: <https://kdd.ics.uci.edu/databases/kddcup99/>. Citado na página 32.

TAN, S. C.; TING, K. M.; LIU, T. F. Fast anomaly detection for streaming data. In: *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2011. v. 22, n. 1, p. 1511. Citado 2 vezes nas páginas 11 e 12.

TONG, S. *Active learning: theory and applications*. [S.l.]: Stanford University, 2001. Citado na página 19.

ZHU, X. et al. Active learning from data streams. In: IEEE. *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. [S.l.], 2007. p. 757–762. Citado 2 vezes nas páginas 13 e 20.