

HEITOR MURILO GOMES

**TEORIA DE REDES SOCIAIS APLICADA AO
PROBLEMA DE CLASSIFICAÇÃO *ONLINE*
COM MUDANÇA DE CONCEITO**

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

CURITIBA

2012

HEITOR MURILO GOMES

**TEORIA DE REDES SOCIAIS APLICADA AO
PROBLEMA DE CLASSIFICAÇÃO *ONLINE*
COM MUDANÇA DE CONCEITO**

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Área de Concentração: *Metodologias e Técnicas de Computação*

Orientador: Prof. Dr. Fabrício Enembreck

CURITIBA

2012

Sumário

Capítulo 1

Introdução16

1.1 Motivação e Hipótese17

1.2 Objetivos.....19

1.3 Organização do trabalho19

Capítulo 2

Classificação *Online*.....20

2.1 Conceitos e desafios21

2.2 Configuração *batch versus* Configuração *online*.....22

2.3 Mudança de conceito24

2.3.1 *Forgetting*28

2.3.2 *Detectors*28

2.3.3 *Contextual*.....28

2.3.4 *Dynamic Ensemble Classifiers*29

2.4 Conjuntos de classificadores30

2.4.1 *Bagging*.....32

2.4.2 *Online Bagging*.....33

2.4.3 *Dynamic Weighted Majority*.....33

2.4.4 *ASHT Bagging*34

2.4.5 *ADWIN Bagging*.....35

2.5 Métodos de avaliação36

2.5.1 Configuração *batch*.....36

2.5.2 Configuração *online*37

2.6 Geradores de dados.....39

2.6.1 *Random Tree Generator*39

2.6.2 SEA.....39

2.6.3 AGRAWAL.....40

2.7 Considerações finais41

Capítulo 3

Análise de Redes Sociais	42
3.1 Conceitos fundamentais.....	43
3.2 Teoria dos Grafos	44
3.2.1 Conceitos	45
3.2.2 Propriedades	49
3.2.3 Caminhamentos	53
3.2.4 Conectividade	54
3.3 Medidas de proeminência.....	56
3.3.1 Centralidade de grau.....	57
3.3.2 Centralidade de proximidade.....	58
3.3.3 Centralidade de intermediação	59
3.3.4 Coeficiente de agrupamento	60
3.4 Topologias	61
3.4.1 Redes aleatórias	62
3.4.2 Redes de mundo pequeno	63
3.4.3 Redes livres de escala	64
3.5 Considerações finais	65

Capítulo 4

Metodologia.....	66
4.1 Algoritmo SAE.....	66
4.2 Combinação das decisões	71
4.3 Adição de Especialistas	77
4.4 Remoção de Especialistas.....	78
4.5 Atualização das Conexões	79
4.6 Considerações finais	80

Capítulo 5

Experimentos	82
5.1 Protocolo experimental.....	82
5.2 Experimentos com diferentes configurações do SAE	86
5.3 Comparativo entre o SAE e outros conjuntos de classificadores	93
5.5 Análise de Redes Sociais aplicada as redes obtidas pelo SAE.....	98

5.4 Considerações finais	103
Capítulo 6	
Conclusão	104
Referências Bibliográficas	107
Apêndice A	
Medidas de prominência coletadas	112
Apêndice B	
O <i>Framework</i> MOA	132
Apêndice C	
Resultados dos experimentos	136

Lista de Figuras

Figura 2.1 O ciclo de Classificação <i>Online</i> , adaptado de (Bifet <i>et al</i> , 2011)	24
Figura 2.2 Mudança de conceito abrupta.....	26
Figura 2.3 Mudança de conceito gradual.....	27
Figura 2.4 Três funções do gerador AGRAWAL, adaptado de (Bifet <i>et al</i> , 2012).....	41
Figura 3.1 Exemplo de Sociograma “Afinidade entre colegas de turma”	45
Figura 3.2 Grafo representando a relação “trabalhou com”	47
Figura 3.3 Subgrafos gerados por nodos (a) e por conexões (b)	48
Figura 3.4 Estados possíveis da tríade.....	48
Figura 3.5 Um grafo e suas propriedades	51
Figura 3.6 Um grafo completo e d-regular	52
Figura 3.7 Grafo com um nodo isolado	52
Figura 3.8 Caminhamentos, trilhas e caminhos.....	54
Figura 3.9 Grafo estrela, circular e linha	57
Figura 4.1 Diagrama de um período.....	68
Figura 4.2 Alterações na estrutura da rede ao final do período.....	69
Figura 4.3 Pseudocódigo – Treinamento do algoritmo SAE.....	70
Figura 4.4 Problema de classificação não linear	72
Figura 4.5 Superfícies de separação	72
Figura 4.6 Pseudocódigo – Combinação das decisões	76
Figura 4.7 Pseudocódigo – Adição de especialista.....	77
Figura 4.8 Pseudocódigo – Remoção de especialistas	79
Figura 4.9 Pseudocódigo – Atualização das conexões	80
Figura 5.1 Rede após quatro períodos ($Cs_{min} = 0.01$) para o gerador RTS.....	88
Figura 5.2 Redes após quatro períodos ($Cs_{min} = 1.00$) para o gerador RTS	88
Figura 5.3 SAE (experimento RTC).....	89
Figura 5.4 SAE (experimento SEA-2).....	90
Figura 5.5 SAE (experimento AGRAWAL-3).....	91
Figura 5.6 SAE e <i>Hoeffding Tree</i> (HF) (experimento SEA-2).....	92
Figura 5.7 SAE e <i>Naïve Bayes</i> (NB) (experimento SEA-2).....	92
Figura 5.8 SAE, ADWIN e ASHT (experimento RTS)	93

Figura 5.9 SAE, ADWIN e ASHT (experimento RTC).....	94
Figura 5.10 SAE e ADWIN (experimento AGRAWAL-1)	95
Figura 5.11 SAE e ASHT (experimento AGRAWAL-1)	95
Figura 5.12 SAE, ADWIN e ASHT (experimento SEA-2).....	96
Figura 5.13 SAE e DWM (experimento AGRAWAL-4).....	97
Figura 5.14 SAE e DWM (# Classificadores) (experimento AGRAWAL-4).....	97
Figura 5.15 Medidas de proeminência (AGRAWAL-3, SAE-1 e SAE-2)	99
Figura 5.16 Medidas de proeminência (AGRAWAL-1, SAE-1)	99
Figura 5.17 Medidas de proeminência (AGRAWAL-1, SAE-2)	100
Figura 5.18 Medidas de proeminência (SEA-4, SAE-1).....	100
Figura 5.19 Medidas de proeminência (SEA-4, SAE-2).....	101
Figura 5.20 SAE (SEA-4).....	101
Figura 5.21 Medidas de proeminência (AGRAWAL-2, SAE-1)	102
Figura 5.22 Medidas de proeminência (AGRAWAL-2, SAE-1)	102
Figura 5.23 Rede no período 19 (AGRAWAL-2, SAE-2).....	102
Figura B.1 Interface gráfica do MOA	133
Figura B.2 Parâmetros DWM: Visão da GUI	134

Lista de Tabelas

Tabela 2.1 Mudança de conceito abrupta	26
Tabela 2.2 Mudança de conceito gradual	27
Tabela 2.3 28 combinações de 3 classificadores, adaptado de (Kuncheva, 2004)	32
Tabela 2.4 Atributos do gerador AGRAWAL, adaptado de (Bifet <i>et al</i> , 2012)	40
Tabela 4.1 Combinação ótima de 3 classificadores	73
Tabela 4.2 Combinação de 4 classificadores	73
Tabela 4.3 Combinação de 4 classificadores agrupando decisões	74
Tabela 5.1 Configurações do método de avaliação	85
Tabela 5.2 Configurações dos geradores de dados	85
Tabela 5.3 Parâmetros do SAE	86
Tabela 5.4 Configurações do SAE utilizadas nos experimentos	87
Tabela 5.5 Medidas de proeminência	98
Tabela A.1 Medidas de proeminência (RTS, SAE-1)	112
Tabela A.2 Medidas de proeminência (RTS, SAE-2)	113
Tabela A.3 Medidas de proeminência (RTC, SAE-1)	114
Tabela A.4 Medidas de proeminência (RTC, SAE-2)	115
Tabela A.5 Medidas de proeminência (SEA-1, SAE-1)	116
Tabela A.6 Medidas de proeminência (SEA-1, SAE-2)	117
Tabela A.7 Medidas de proeminência (SEA-2, SAE-1)	118
Tabela A.8 Medidas de proeminência (SEA-2, SAE-2)	119
Tabela A.9 Medidas de proeminência (SEA-3, SAE-1)	120
Tabela A.10 Medidas de proeminência (SEA-3, SAE-2)	121
Tabela A.11 Medidas de proeminência (SEA-4, SAE-1)	122
Tabela A.12 Medidas de proeminência (SEA-4, SAE-2)	123
Tabela A.13 Medidas de proeminência (AGRAWAL-1, SAE-1)	124
Tabela A.14 Medidas de proeminência (AGRAWAL-1, SAE-2)	125
Tabela A.15 Medidas de proeminência (AGRAWAL-2, SAE-1)	126
Tabela A.16 Medidas de proeminência (AGRAWAL-2, SAE-2)	127
Tabela A.17 Medidas de proeminência (AGRAWAL-3, SAE-1)	128
Tabela A.18 Medidas de proeminência (AGRAWAL-3, SAE-2)	129

Tabela A.19 Medidas de proeminência (AGRAWAL-4, SAE-1).....	130
Tabela A.20 Medidas de proeminência (AGRAWAL-4, SAE-2).....	131
Tabela C.1 Média da taxa de acerto	137
Tabela C.2 Tempo de execução (CPU Time) mensurado em segundos	137
Tabela C.3 Quantidade média de classificadores	138

Lista de Símbolos

G	Grafo não direcionado de única relação dicotômica.
G_s	Subgrafo de um grafo G .
n_i	i -ésimo nodo de um grafo.
N	Conjunto de todos os nodos de um grafo.
c_i	i -ésima conexão em um grafo.
C	Conjunto de todas as conexões em um grafo.
G	Cardinalidade do conjunto N . Análogo a $ N $.
L	Cardinalidade do conjunto L . Análogo a $ C $.
$d(n_i)$	Grau de um nodo. Mensura a quantidade de conexões com o ator.
\bar{d}	Grau médio. Sumariza o grau de todos os nodos de um grafo.
S_D^2	Variância. Indica diferenças na quantidade de conexões para cada nodo.
C_2^g	Total de pares (díades) não ordenados em um grafo.
Δ	Densidade. Representa a proporção de conexões existentes relativa ao total de conexões possíveis.
W	Caminhamento. Sequência de nodos e conexões incidentes.
T	Trilha. Tipo especial de caminhamento onde as conexões não podem repetir.
P	Caminho. Tipo especial de trilha onde os nodos não podem repetir.
\cdot^{-1}	Caminhamento, trilha ou caminho em ordem reversa.
$d(n_i, d_j)$	Distância Geodésica. Menor caminho (geodésico) entre dois nodos.

\bar{D}	Distância Geodésica Média. Representa média de todas as distâncias geodésicas de um grafo.
$e(n_i)$	Excentricidade. Representa a maior $d(n_i, n_j)$ entre todos os j nodos alcançáveis pelo nodo i .
$\max_{i,j} d(n_i, n_j)$	Diâmetro. Representa a maior $e(n_i)$ dentre todos os atores da rede.
a_i	Ator de uma rede social.
$C'_D(a_i)$	Centralidade de grau padronizada. Análoga ao grau de um nodo, porém normalizada segundo o grau máximo ($g-1$).
$C_C(a_i)$	Centralidade de proximidade. Mensura o quão rapidamente um ator pode interagir com outros em uma rede social.
$C_B(a_i)$	Centralidade de intermediação. Mensura a proeminência de um ator relativa a quantidade de geodésicos em que este é incluído.
$C'_B(a_i)$	Centralidade de intermediação padronizada.
$C_i(a_i)$	Coefficiente de agrupamento. Mensura o quão próximo os atores adjacentes a um ator a_i estão de formar um <i>clique</i> .
\bar{C}	Coefficiente de agrupamento médio. Indica o quão coesos são os subgrafos que formam um grafo.
C_s	Coefficiente de similaridade (SAE).
c	Comprimento de um período (SAE).
Tx_{minR}	Taxa de acerto mínimo da rede (SAE).
Tx_{minE}	Taxa de acerto mínimo do especialista (SAE).
CS_{min}	Coefficiente de similaridade mínimo entre especialistas (SAE).
CS_{max}	Coefficiente de similaridade máximo entre especialistas (SAE).

Lista de Abreviaturas

ADWIN	<i>Adaptive Window</i>
ASHT	<i>Adaptive Size Hoeffding Tree</i>
ARS	Análise de Redes Sociais
DWM	<i>Dynamic Weight Majority</i>
K-NN	<i>K-Nearest Neighbors</i>
SAE	<i>Social Adaptive Ensemble</i>
SEA	<i>Streaming Ensemble Algorithm</i>
NB	<i>Naïve Bayes</i>
HF	<i>Hoeffding Tree</i>
RTG	<i>Random Tree Generator</i>
RTS	<i>Random Tree Simple</i>
RTC	<i>Random Tree Complex</i>

Resumo

Este trabalho compreende o desenvolvimento do método de conjunto de classificadores baseado em Redes Sociais SAE (*Social Adaptive Ensemble* – Conjunto de Classificadores Adaptativos Sociais) para classificação de *streams* de dados (Classificação *Online*). No âmbito da Classificação *Online*, o problema de mudança de conceito é tido como um dos mais importantes. Métodos baseados em conjunto de classificadores apresentam vantagens para Classificação *Online* em que mudanças de conceito ocorrem, uma vez que estes podem adaptar os classificadores que os compõe, removendo-os, adicionando novos ou atualizando os existentes na ocorrência de uma mudança de conceito. A abstração do conjunto de classificadores como uma Rede Social adiciona a noção de conexão entre os classificadores, a qual mensura o quão similar dois classificadores são com relação às predições feitas por eles, dessa forma, o conjunto deixa de ser dependente exclusivamente de métricas locais de performance para avaliação de cada classificador. Além disso, é possível identificar padrões de comportamento entre os classificadores e analisar questões como redundância e diversidade. A estratégia de evolução do conjunto foi desenvolvida a partir da noção de classificadores similares. O SAE dispõe os classificadores em uma estrutura de rede adaptada aos últimos dados observados. A adaptação do conjunto é feita de forma a reforçar a generalidade dos classificadores e a taxa de acerto. São apresentados testes com *streams* de dados estáveis, com mudança de conceito gradual e abrupta. Em comparação com outros algoritmos, que representam o estado da arte da literatura, o SAE obteve taxa de acerto similar e, para grande parte dos experimentos, um tempo de processamento significativamente menor.

Palavras-Chave: Conjunto de classificadores, Análise de Redes Sociais, Classificação *Online*, Mineração de *Streams* de dados.

Abstract

This work encompasses the development of a new ensemble classifier based on Social Networks named SAE (Social Adaptive Ensemble) for Stream Classification. Within Stream Classification, concept drift is considered one of the most important issues. Ensemble methods present advantages on Stream Classification where concept drift occurs, since they can adaptively update its component classifiers, removing, adding new or resetting them. Using a Social Network abstraction for the ensemble adds the notion of connection between classifiers, which measure how similar predictions from two classifiers are, thus, the ensemble is not exclusively dependent on local performance metrics for each classifier evaluation. Besides that, it is possible to identify patterns among classifiers, such that it is possible to analyze classifiers' redundancy and diversity. The ensemble evolution strategy was developed based on the notion of similar classifiers. SAE arranges classifiers in a network structure adapted to the last seen data. The adaption is made to enforce ensemble generalization and accuracy. Experiments are reported for streams with no drift, abrupt and gradual drift. In comparison to state of the art algorithms, SAE has achieved similar accuracy and, for most cases, demands significantly less processing time.

Keywords: Ensemble Classifiers, Social Network Analysis, Data Stream Classification, Data Stream Mining.

Capítulo 1

Introdução

Os recentes avanços tecnológicos relativos a dispositivos e softwares para aquisição e armazenamento de dados permitiram que organizações adquirissem e armazenassem um volume elevado de dados. Essas organizações costumam estar mais interessadas no conhecimento que pode ser extraído dos dados do que nos dados brutos em si. Por exemplo, uma montadora de carros pode rastrear os comentários sobre os seus modelos de carros em uma rede social. De posse destes comentários, a montadora pode prever em qual dos modelos ela deve investir para obter o maior retorno. Esse tipo de configuração dos dados é denominada uma fonte de dados de fluxo contínuo, ou *stream* de dados, e ela adiciona características únicas a tarefa de classificação, como: volume de dados elevado ou infinito (muitos comentários), instabilidade na distribuição (usuários podem mudar de opinião), período de validade dos dados (comentários de meses atrás podem não ser tão úteis quanto os atuais), rápida adição de novos dados (toda vez que um cliente fizer um comentário sobre um modelo de carro da montadora).

As características apresentadas podem ser sumarizadas em quatro restrições que devem ser atendidas para que um classificador de *streams* de dados seja eficiente. A primeira restrição está relacionada com a memória disponível. Os dados não podem ser armazenados para processamento posterior, pois isso não é viável (*stream* de dados potencialmente infinita) ou não é relevante (dados antigos não representam o conceito atual). A segunda restrição diz respeito ao tempo de processamento necessário para processar cada instância. Instâncias serão perdidas se o tempo entre a chegada de uma nova instância for menor do que o tempo que o classificador

precisa para atualizar o seu modelo. A terceira restrição assume que o conceito que representa a distribuição de classes entre as instâncias pode mudar ao longo do tempo, conseqüentemente, o classificador precisa adaptar o seu modelo para se adaptar ao novo conceito. A última restrição, a qual não é limitada a *streams* de dados, dita que o modelo gerado pelo classificador deve ser tão preciso quanto possível com relação a taxa de acerto.

Para lidar com estas quatro restrições (espaço, tempo, mudança de conceito e taxa de acerto) foram propostos vários algoritmos, como o DWM (Maloof e Kolter, 2003), o ASHT *Bagging* (Bifet et al, 2009), o *Hoeffding Tree* (Domingos e Hulten, 2000), dentre outros. Uma atenção especial foi dada para os algoritmos baseados em conjuntos de classificadores, uma vez que estes costumam obter uma taxa de acerto maior e em algumas situações facilitam a adaptação do modelo do classificador frente a uma mudança de conceito, apesar destes exigirem mais memória e tempo de processamento.

Como é apontado em (Polikar, 2006) existem dois componentes principais que devem ser considerados no desenvolvimento de um método baseado em conjunto de classificadores. O primeiro diz respeito a estratégia de construção do conjunto, a qual deve reforçar a diversidade entre os classificadores. A segunda é relacionada a forma como é realizada a combinação das predições, a qual deve ressaltar predições corretas e ofuscar as incorretas. Além destes dois componentes, quando os dados são representados por uma *stream* de dados é importante incluir uma estratégia de atualização em resposta a mudanças de conceito.

O algoritmo *Social Adaptive Ensemble* (SAE) apresentado neste trabalho, foca nestes três aspectos de um método baseado em conjunto de classificadores para *streams* de dados, isto é, diversidade, combinação e adaptação. No SAE os classificadores são dispostos em uma estrutura de rede social, na qual os classificadores representam os atores e as conexões representam a similaridade, relativa as predições passadas, entre eles.

1.1 Motivação e Hipótese

Medidas como a quantidade de erros cometidos, ou a taxa de acerto podem ser mensuradas para cada classificador de um conjunto. Uma estratégia simples é remover aqueles com baixa taxa de acerto e manter os demais, adicionando novos conforme antigos são removidos. Neste caso não ocorre a identificação de uma mudança de conceito e o modelo se

adapta a mudança naturalmente. Além da estratégia de atualização do conjunto, dois outros componentes devem ser levados em consideração no desenvolvimento do conjunto de classificadores, são eles: a diversidade do conjunto e o método de combinação das previsões individuais. Construir um conjunto diverso de classificadores, com relação as suas previsões, é importante para permitir que a combinação destas seja melhor do que um único classificador. A maneira como as previsões dos classificadores são combinadas deve ressaltar previsões corretas e evitar as incorretas.

A hipótese em que está baseado este trabalho é de que para se obter um conjunto de classificadores diverso e permitir uma combinação adequada destes é importante mensurar a similaridade existente entre as previsões dos classificadores que formam o conjunto. Mensurar a similaridade entre os classificadores é útil para identificar quais são dissimilares e apontar quais são redundantes. A combinação de vários classificadores redundantes não permite melhorar a taxa de acerto, além disso, a existência de muitos classificadores redundantes ofusca previsões corretas feitas por pequenos subconjuntos de classificadores dissimilares. A partir da definição da similaridade entre classificadores é possível combinar as previsões a nível de subconjuntos similares e depois combinar as decisões destes subconjuntos na decisão final.

Obter um conjunto de classificadores diversos não é trivial, especialmente na Classificação *Online*. Em (Oza e Rusell, 2001) foram propostos métodos que treinam os classificadores utilizando subconjuntos dos dados (extensões do *Bagging* (Breiman, 1996) e *Boosting* (Freund e Schapire, 1997) para a Classificação *Online*). Em (Maloof e Kolter, 2003), a diversidade do conjunto é obtida a partir da adição dos classificadores em momentos diferentes, porém uma vez adicionados eles serão sempre treinados nas mesmas instâncias. Em (Bifet *et al*, 2009), o conjunto é formado por árvores de decisão que possuem limitações quanto a altura máxima que cada uma pode obter (*ASHT Bagging*). Todos estes métodos criam os classificadores de modo a maximizar a diversidade, porém não é feita uma mensuração explícita para verificar a diversidade entre os classificadores. Alguns destes métodos têm o conjunto renovado com uma frequência maior, como o DWM (Maloof e Kolter, 2003), porém a renovação é feita com base na taxa de acerto individual dos classificadores e não considera a similaridade entre eles.

Para garantir a diversidade do conjunto no algoritmo SAE é utilizado um método de *Bagging online* (Oza e Russel, 2001), a adição de classificadores em diferentes momentos (DWM (Maloof e Kolter, 2003)), além do uso de mecanismos de identificação e remoção de classificadores redundantes. As taxas de acerto individual e do conjunto também são consideradas para remover classificadores que estão abaixo de um limiar mínimo e adicionar novos classificadores quando o conjunto possui uma taxa de acerto baixa. De modo geral, o algoritmo SAE combina propostas existentes na literatura com a identificação das relações entre classificadores para manter um conjunto diverso.

1.2 Objetivos

Este trabalho tem como objetivo global o desenvolvimento de um método baseado em conjunto de classificadores (*Social Adaptive Ensemble – SAE*) que abstrai o conjunto como uma rede social por meio da identificação das relações entre os classificadores. Os objetivos específicos incluem o estudo do relacionamento entre os classificadores, a implementação de um modelo evolutivo de rede baseado no modelo de rede livre de escala, a análise do comportamento do SAE frente a *streams* de dados com e sem mudanças de conceito e a comparação entre o SAE e os demais métodos baseados em conjunto de classificadores presentes na literatura.

1.3 Organização do trabalho

A organização deste trabalho é feita da seguinte maneira: No capítulo 2 são apresentados os conceitos relativos à Classificação *Online* e a mudança de conceito. No capítulo 3 é apresentada a Análise de Redes Sociais, incluindo conceitos da Teoria dos Grafos, medidas de proeminência e topologias de rede. No capítulo 4 é apresentado o algoritmo SAE. No capítulo 5 são apresentados os experimentos realizados e a discussão dos mesmos.

Capítulo 2

Classificação *Online*

Motivados pelas tecnologias de obtenção e persistência de dados, empresas, grupos de pesquisa e governos, passaram a armazenar um conjunto cada vez maior de dados. Com a intenção de melhor aproveitar o potencial das bases de dados mantidas por estas instituições foi estabelecida a área de aprendizagem de máquina. Inicialmente, os problemas a serem solucionados tratavam da extração de conhecimento de um grande volume de dados (mineração de dados) ou da obtenção de um modelo capaz de prever a classe de um novo exemplo (classificação). Devido ao avanço na geração e propagação de dados em alguns domínios, tornou-se inviável o armazenamento dos dados para a execução posterior de um algoritmo de mineração de dados. Para tratar deste tipo de problema tornou-se necessário alterar a forma com que os algoritmos operam sobre os dados. Esta nova forma de interpretação dos dados foi denominada *data streams* (fluxo contínuo de dados ou *streams* de dados).

Com o intuito de permitir o aprendizado a partir de *streams* de dados foi estabelecida a área de pesquisa de Classificação *Online*. Algoritmos de Classificação *batch*¹ não são diretamente aplicáveis em ambientes onde os dados são disponibilizados em um fluxo contínuo. Isso acontece devido aos requisitos que um problema de Classificação *Online* demanda, tais como: processar um volume elevado de dados, alterações incrementais no modelo e possíveis mudanças de conceito. Uma mudança de conceito ocorre quando o conceito que define uma classe muda ao longo do tempo. Algoritmos desenvolvidos para Classificação *Online* devem ser

¹ Algoritmos de aprendizagem de máquina tradicionais como C4.5 (Quinlan, 1993), K-NN (Aha *et al*, 1991), etc.

capazes de aprender o novo conceito descartando o antigo. Alguns pesquisadores reconhecem o problema de mudança de conceito como o principal desafio da Classificação *Online*.

Nas seções seguintes deste capítulo são apresentados os principais conceitos e dificuldades relativas à Classificação *Online*. Os objetivos e desafios pertinentes a área de Classificação *Online* são apresentados na seção 2.1. As diferenças entre Classificação *batch* e *Online* são elucidadas na seção 2.2. A seção 2.3 é dedicada a apresentar a mudança de conceito (*concept drift*) e as principais abordagens para este problema. A seção 2.4 apresenta a ideia por trás dos conjuntos de classificadores e alguns métodos baseados em conjunto de classificadores para a tarefa de Classificação *Online*. A seção 2.5 apresenta os métodos de avaliação disponíveis para *streams* de dados correlacionando-os com os métodos tradicionais utilizados na Classificação *batch*. A seção 2.6 apresenta geradores de dados, os quais são utilizados para simular *streams* de dados usadas para a avaliação de classificadores. A seção 2.7 compreende as considerações finais do capítulo.

Os conceitos apresentados neste capítulo servem de base para a compreensão do método de conjunto de classificadores proposto neste trabalho (algoritmo SAE). Também são apresentados os métodos de avaliação e outros algoritmos baseados em conjuntos de classificadores para Classificação *Online*.

2.1 Conceitos e desafios

Data streams (*streams* de dados) é o nome atribuído a uma fonte de dados contínua s , onde a cada t unidades de tempo uma nova instância i torna-se disponível para ser processada. Algoritmos de classificação desenvolvidos para operar com *streams* de dados devem estar prontos para lidar com uma massa de dados elevada, possivelmente infinita, de forma que cada instância só poderá ser inspecionada uma única vez sendo descartada logo em seguida, a fim de liberar memória para as instâncias subsequentes. Outras características pertinentes a algoritmos para Classificação *Online*:

- Não existe controle sobre a ordem em que as instâncias são apresentadas;
- O algoritmo deve ser capaz de atualizar seu modelo de forma incremental, aperfeiçoando-o a cada nova instância que lhe é apresentada; e

- O algoritmo deve estar pronto para classificar uma nova instância “a qualquer momento” (propriedade *anytime*) independente da quantidade de instâncias que já lhe foram apresentadas (Bifet *et al*, 2011).

Streams de dados podem ser classificadas como estacionárias (*stationary*) ou evolucionárias (*evolving*). No caso das estacionárias a definição do conceito não muda ao longo do tempo, já nas evolucionárias ocorrem mudanças de conceito, as quais devem ser detectadas e tratadas apropriadamente pelo classificador. O problema de mudança de conceito é descrito na seção 2.3.

Algoritmos para a Classificação *Online* necessitam de métodos de avaliação apropriados que permitam ao usuário não só verificar a taxa de acerto de um determinado algoritmo, mas também se ele é aplicável para o problema em questão, ou seja, se ele atende a requisitos mínimos de tempo de processamento e uso de memória. Portanto a performance de tais algoritmos pode ser interpretada como um problema tridimensional (Bifet *et al*, 2011) que compreende: Taxa de acerto, tempo de processamento e consumo de memória. Os métodos de avaliação da taxa de acerto para Classificação *Online* são apresentados na seção 2.5.2.

2.2 Configuração *batch* versus Configuração *online*

Os algoritmos convencionais de classificação, referenciados por configuração *batch*, operam assumindo que o conjunto de treinamento estará disponível como um todo e que qualquer exemplo pode ser recuperado quando necessário a um baixo custo computacional, em outras palavras, é possível reavaliar uma instância previamente disponibilizada se necessário. Algoritmos para a classificação *Online*, referenciados por configuração *online*, operam sob a prerrogativa de que os exemplos do conjunto de treinamento serão apresentados um de cada vez, em uma ordem que não pode ser controlada. Consequentemente, para que um algoritmo opere na configuração *online* é necessário que ele atenda a alguns requisitos que não existem na configuração *batch*. Segundo (Bifet *et al*, 2011), os requisitos são:

1. Processar um exemplo de cada vez e inspecioná-lo no máximo uma vez. Cada exemplo deve ser processado ou ignorado na ordem em que chega e descartado logo em seguida. Contudo, não existe nenhuma regra que impeça o algoritmo de armazenar um pequeno conjunto de exemplos para serem utilizados em curto prazo desde que o requisito 2 seja respeitado.

2. Utilizar uma quantidade limitada de memória. Uma das maiores motivações para o emprego de modelos baseados em *streams* de dados é que estes permitem o processamento de volumes de dados de ordens muito mais elevadas do que a memória disponível. Isto é possível desde que seja imposto um limite para o espaço em memória a ser utilizado pelo algoritmo e que este seja capaz de respeitá-lo na construção do modelo. Esta restrição pode ser amenizada utilizando arquivos temporários em memória secundária, porém esta abordagem deve levar em consideração o requisito 3.

3. Processar em tempo limitado. Para que um algoritmo seja considerado escalável para um número qualquer de exemplos é necessário que ele execute, ao menos, linearmente em relação ao tamanho da entrada. Para obter este comportamento em uma configuração *online* é necessário especificar um limite de tempo que deve ser respeitado pelo algoritmo ao processar um exemplo. Ainda mais crítica é a situação em que o algoritmo deve executar em tempo real, o que significa que ele deve processar cada exemplo na mesma velocidade, ou ainda mais rápido, do que os dados são disponibilizados. Caso isso não seja possível, inevitavelmente, dados serão perdidos.

4. Estar pronto para prever a qualquer momento (propriedade “*anytime*”). Um algoritmo ideal deve estar pronto para fornecer o melhor modelo possível a qualquer momento, independente da quantidade de exemplos que tenham sido apresentados a ele. Isto implica que o processo de atualização do modelo deve ser eficiente (requisito 3), logo é esperado que não seja preciso reconstruir todo o modelo para atualizá-lo.

O processo de Classificação *Online* pode ser dividido em três passos, ilustrados na Figura 2.1 e apresentados a seguir:

1. O algoritmo recebe o próximo exemplo disponível na *stream* (requisito 1).
2. Este exemplo é processado, atualizando o modelo sem exceder o limite de espaço (requisito 2) ou de tempo (requisito 3).
3. O algoritmo está pronto para receber um novo exemplo ou fornecer um modelo capaz de prever a classe de um exemplo desconhecido (requisito 4).

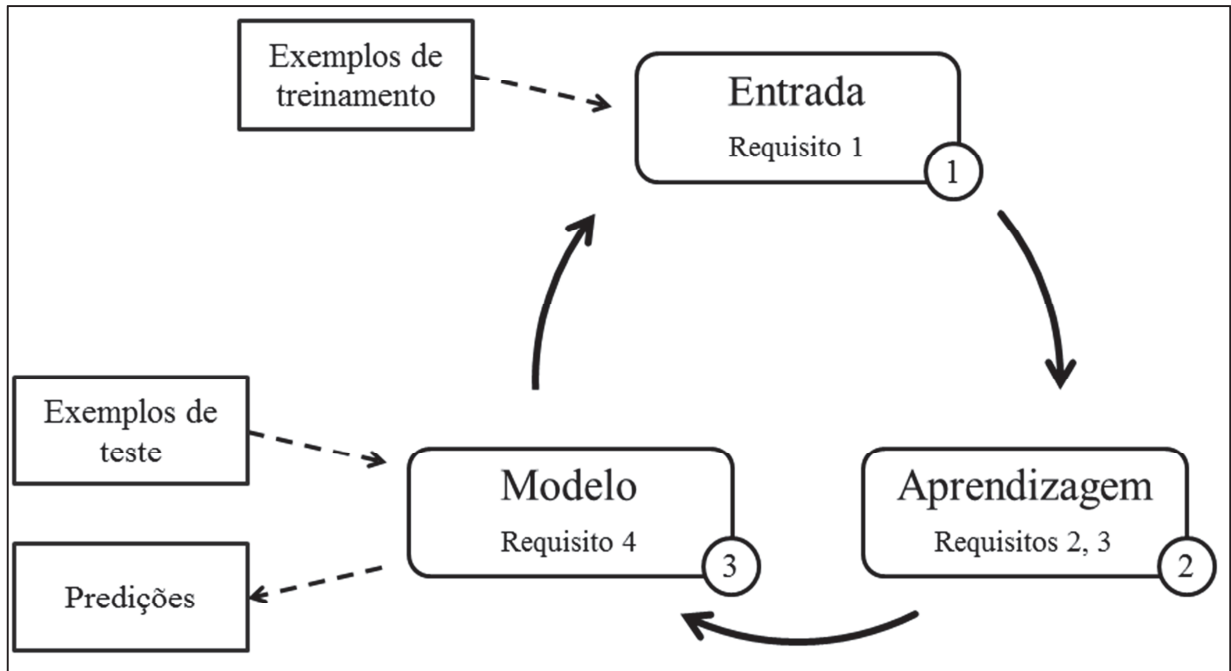


Figura 2.1: O ciclo de Classificação *Online*, adaptado de (Bifet *et al*, 2011)

2.3 Mudança de Conceito

Um conceito é representado por uma função lógica que mapeia valores de atributos (qualitativos ou quantitativos) para uma determinada classe (atributo meta). Esta função pode assumir a forma de uma disjunção de conjunções, como por exemplo:

Atributos: Preço, Qualidade e Compra (classe).

Domínios: Preço=Qualidade={muito baixo,baixo,normal,alto,muito alto}, Compra = {sim, não}.

Conceito A: $\text{Preço} \in \{\text{muito baixo, baixo}\} \wedge \text{Qualidade} \in \{\text{alto}\} \rightarrow \text{Compra} = \{\text{sim}\}$

O conceito A representa a hipótese de que se uma instância qualquer assumir os valores “muito baixo” ou “baixo” para Preço e “alto” para Qualidade, então Compra (classe) assumirá o valor “sim”.

Segundo (Tsymbol, 2006) um dos maiores problemas relativos à tarefa de classificação é que em muitos domínios do mundo real, o conceito depende de fatores que pertencem a um contexto desconhecido (*hidden context*). Um exemplo disso é o padrão de compra dos

consumidores de um supermercado, o qual pode variar de acordo com características pertencentes ao contexto desconhecido como a inflação, época do ano, campanhas publicitárias, dentre outras características não disponibilizadas para o algoritmo de aprendizagem na forma de atributos. Alterações provenientes do contexto desconhecido podem induzir mudanças radicais ou moderadas em um conceito. Estas mudanças são conhecidas como mudanças de conceito (*concept drift*)².

A causa de tais mudanças não pode ser determinada, tampouco prevista pelo algoritmo de aprendizagem, uma vez que ele não dispõe dos atributos para distingui-las. Portanto um algoritmo de aprendizagem eficiente deve ser capaz apenas de detectar que uma mudança ocorreu e se adaptar rapidamente a ela. Um dos maiores problemas inerentes à mudança de conceito é diferenciar entre mudanças reais e ruído. Segundo (Widmer e Kubate, 1996) um algoritmo de aprendizado ideal deve combinar robustez a ruídos e sensibilidade para mudanças reais de conceito.

De forma pragmática, uma mudança de conceito ocorre quando instâncias que outrora eram mapeadas por uma função lógica (Conceito A) para uma classe X qualquer deixam de mapear para X , dando lugar a uma nova função lógica (e.g. Conceito B) que mapeia, de forma estável, para classe X . Pequenas variações (mapeamentos instáveis) são esperadas e não denotam uma mudança de conceito, são apenas ruídos. Mudanças de conceito podem ocorrer de forma abrupta ou gradual. Para classificar uma mudança entre abrupta ou gradual é preciso analisar o tamanho da sua janela de mudança Δx , ou seja, quantas instâncias são necessárias para que o novo conceito se torne estável. Considerando que uma mudança começa a ocorrer a partir de uma instância i_x e que ela se torna estável a partir da instância $i_{x + \Delta x}$, se Δx for igual a 1 a mudança foi abrupta, se Δx for significativamente maior do que 1 então a mudança ocorreu de forma gradual. Por exemplo, considerando a sequência de instâncias $I = (i_1, i_2, i_3, i_4, i_5, i_6, i_7, i_8, i_9, i_{10})$, o conceito A ($Preço \in \{\text{muito baixo, baixo}\} \wedge Qualidade \in \{\text{alto}\} \rightarrow Classe = \{\text{sim}\}$) e o conceito B ($Preço \in \{\text{normal, alto}\} \wedge Qualidade \in \{\text{alto, muito alto}\} \rightarrow Classe = \{\text{sim}\}$), a

² O termo *concept drift* foi inicialmente cunhado por Schlimmer e Granger em 1986 no artigo intitulado *Incremental learning from noisy data* (Schlimmer e Granger, 1986).

Tabela 2.1 e a Figura 2.2 apresentam um exemplo de mudança abrupta. Na Tabela 2.1 são destacadas em negrito as instâncias em que a mudança de conceito ocorre.

Instância	Classe	Instância	Classe
$i_1 =$ Preço (muito baixo) Qualidade(alto)	Sim	$i_6 =$ Preço (alto) Qualidade(alto)	sim
$i_2 =$ Preço (baixo) Qualidade(alto)	Sim	$i_7 =$ Preço (normal) Qualidade(muito alto)	sim
$i_3 =$ Preço (normal) Qualidade(alto)	Não	$i_8 =$ Preço (normal) Qualidade(alto)	sim
$i_4 =$ Preço (baixo) Qualidade(alto)	Sim	$i_9 =$ Preço (alto) Qualidade(muito alto)	sim
$i_5 =$ Preço (baixo) Qualidade(alto)	Sim	$i_{10} =$ Preço (baixo) Qualidade(alto)	não

Tabela 2.1: Mudança de conceito abrupta

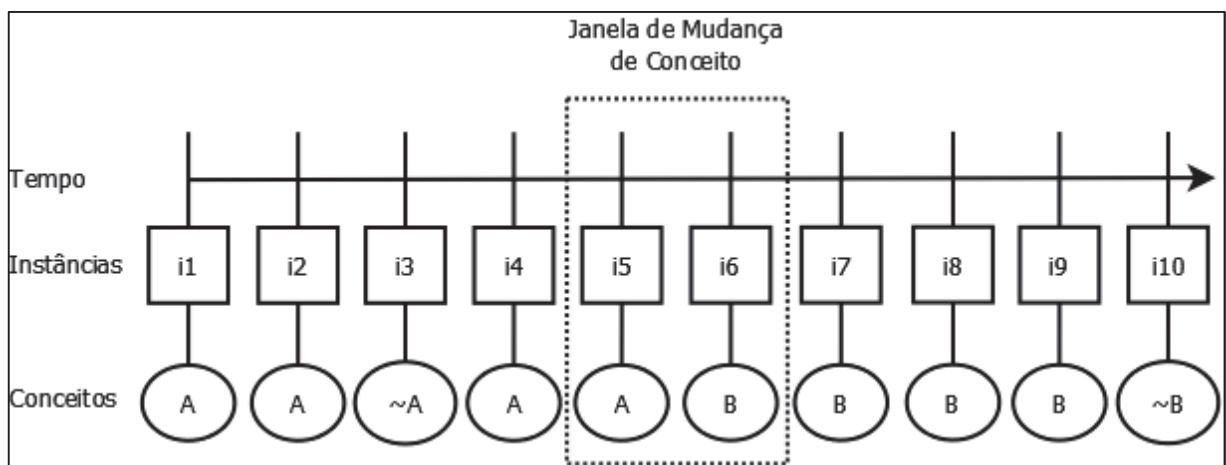


Figura 2.2: Mudança de conceito abrupta

Em uma mudança gradual a janela de mudança possui n instâncias, as quais podem pertencer ao conceito antigo (A no exemplo) ou ao novo (B no exemplo). Esta área de mudança pode ser modelada como uma função de probabilidade que representa a chance de uma instância atender ao conceito A ou ao conceito B . Esta função de probabilidade pode, por exemplo, ser uma função linear $b = (c - x) / \Delta x$, tal que c é o índice da instância atual, x o índice da primeira instância da janela de mudança e Δx a diferença entre x e o último índice da janela. A probabilidade de uma instância i_c qualquer pertencer ao conceito B é $P(i_c = B) = b$, e de atender ao conceito A é $P(i_c = A) = 1 - b$. Na Tabela 2.2 e na Figura 2.3 é apresentado um exemplo de mudança gradual de conceito usando a função linear b .

i	Situação	Atributo(valor) → classe	Conceito	$P(A) = 1 - b$	$P(B) = b$	$b = (c - x) / \Delta x$
1	A estável	Preço(muito baixo) Qualidade(alto) → sim	A	1	0	
2	A estável	Preço(baixo) Qualidade(alto) → sim	A	1	0	
3	A estável	Preço(alto) Qualidade(alto) → não	$\sim A$	1	0	
4	Início ($i_x = i_4$)	Preço(baixo) Qualidade(alto) → sim	A	1	0	$(4 - 4) / 4 = 0$
5	Zona de mudança	Preço(normal) Qualidade(alto) → sim	B	0.75	0.25	$(5 - 4) / 4 = 0.25$
6	Zona de mudança	Preço(baixo) Qualidade(alto) → sim	A	0.50	0.50	$(6 - 4) / 4 = 0.5$
7	Zona de mudança	Preço(alto) Qualidade(muito alto) → sim	B	0.25	0.75	$(7 - 4) / 4 = 0.75$
8	Fim ($i_{x+\Delta x} = i_8$)	Preço(normal) Qualidade(alto) → sim	B	0	1	$(8 - 4) / 4 = 1$
9	B estável	Preço(baixo) Qualidade(muito alto) → não	$\sim B$	0	1	
10	B estável	Preço(alto) Qualidade(muito alto) → sim	B	0	1	

Tabela 2.2. Mudança de conceito gradual

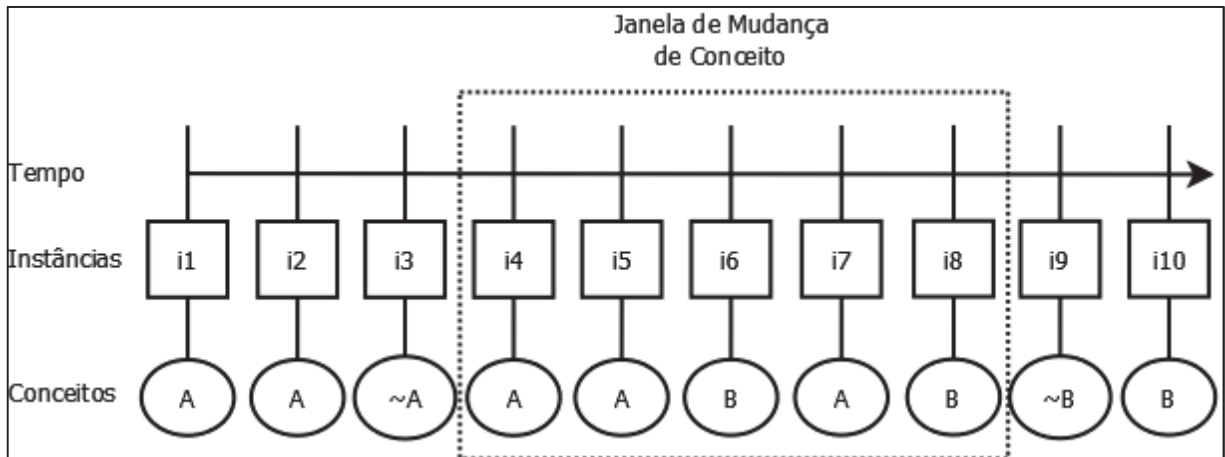


Figura 2.3: Mudança de conceito gradual

É de suma importância que um algoritmo de aprendizagem seja capaz de detectar uma mudança de conceito rapidamente para que sua taxa de acerto seja afetada apenas em alguns instantes, preferencialmente somente durante a janela de mudança.

Segundo (Žliobaite, 2010), as quatro estratégias mais utilizadas para aprendizado com mudança de conceito são *Forgetting* (esquecimento), *Detectors* (detectores), *Contextual* (contextual) e *Dynamic Ensemble Classifiers* (conjunto de classificadores dinâmico), apresentados nas subseções 2.3.1, 2.3.2, 2.3.3 e 2.3.4, respectivamente.

2.3.1 *Forgetting*

Uma das maneiras de reagir a uma mudança de conceito é “esquecendo” exemplos antigos e treinando o modelo apenas com exemplos recentes. Esta estratégia de “esquecimento” é baseada na hipótese de que os exemplos vão perdendo a sua relevância para a classificação com o passar do tempo. O maior problema está em definir qual o intervalo de exemplos a ser utilizado para novos ciclos de treinamento. Uma abordagem simples é definir janelas de tamanho fixo, as quais são atualizadas em períodos constantes, garantindo que exemplos antigos serão desconsiderados dando lugar a exemplos novos. O tamanho da janela influencia diretamente a qualidade do classificador. Ao utilizar uma janela pequena, o classificador se torna instável, isto significa que um modelo robusto dos dados não é obtido e conseqüentemente a taxa de acerto será baixa. Apesar disso, mudanças abruptas são rapidamente detectadas em janelas pequenas, tornando o modelo mais flexível. O uso de janelas com grandes quantidades de exemplos faz com que o classificador se beneficie durante períodos estáveis (sem mudanças de conceito) ao custo de permanecer com uma baixa taxa de acerto até que a janela com um novo conceito seja considerada, tornando o modelo estável, porém inflexível.

2.3.2 *Detectors*

A ideia por trás dos métodos de detecção de mudança de conceito é a de controlar a taxa de erro do algoritmo (Gama *et al*, 2004). É possível detectar se uma mudança ocorreu observando os dados (Bifet e Gavaldà, 2007), os parâmetros para o classificador (Su *et al*, 2008), ou ainda os erros de classificação do modelo (Baena-Garcia *et al*, 2006). Os métodos de detecção normalmente indicam a instância na qual a mudança começou e reiniciam o treinamento do modelo a partir dela, obtendo desta forma janelas dinâmicas de treinamento. Esta abordagem é implicitamente associada com mudanças abruptas, pois a identificação de mudanças graduais se torna difícil uma vez que a detecção da mudança gradual pode demorar.

2.3.3 *Contextual*

Os modelos baseados no contexto (*Contextual*) constroem vários modelos e alternam entre eles de acordo com os dados atuais. Esta estratégia está intrinsicamente ligada com mudanças de conceito recorrentes. Estas mudanças são uma terceira alternativa aos dois outros

tipos de mudanças previamente citadas (abrupta e gradual). Conceitos recorrentes alternam entre os estados de ativo e inativo com certo grau de regularidade. Os modelos adaptados a um determinado modelo podem ser salvos e reutilizados em situações futuras, fazendo com que o sistema não tenha que repetir o processo de aprendizado para um conceito previamente aprendido. A maior dificuldade está em identificar quando um conceito volta a se tornar ativo e quais modelos devem ser armazenados para utilização futura.

2.3.4 *Dynamic Ensemble Classifiers*

Nos métodos de conjuntos de classificadores dinâmicos (*Dynamic Ensemble Classifiers*) é utilizado um conjunto de classificadores de forma que a saída de cada um deles é combinada para formar a decisão final do conjunto. O classificador pelo qual o conjunto é formado é referenciado por classificador base. Algumas abordagens baseadas em conjuntos de classificadores não são dependentes de um classificador base em particular como o algoritmo DWM (Kolter e Maloof, 2003) e o SEA (Street e Kim, 2001), porém existem conjuntos que manipulam parâmetros do classificador base, como o algoritmo ASHT *Bagging* (Bifet *et al*, 2009).

Como ressaltado em (Kuncheva, 2004) não existe um consenso ou taxonomia bem definida quanto ao modo de operação de métodos baseados em conjunto de classificadores. Contudo, usualmente o conjunto é formado por n classificadores, sendo que tal cardinalidade pode variar conforme a estratégia de atualização do conjunto. A propriedade dinâmica é oriunda da característica de atualização do modelo (adicionando, removendo, substituindo ou atualizando classificadores) e está relacionada com a aplicação de conjuntos de classificadores para a configuração *online*. É possível que exista algum tipo de ponderação no voto de cada um dos classificadores do conjunto. Esta ponderação pode ser usada, por exemplo, para reforçar o peso do voto dos classificadores que possuem uma maior taxa de acerto (algoritmo DWM).

Normalmente a adaptação de um conjunto de classificadores não é rápida, logo estes métodos são usualmente utilizados em problemas que envolvem mudanças de conceito graduais (Žliobaite, 2010). A criação e exclusão de classificadores podem indicar períodos estáveis (pouca ou nenhuma adição ou exclusão) ou de mudança (várias exclusões e adições).

Alguns exemplos de métodos baseados em conjuntos de classificadores para Classificação *Online*, com mudança de conceito, são os algoritmos DWM (Kolter e Maloof, 2003), FLORA

(Widmer e Kubate, 1996), SEA (Street e Kim, 2001), ADWIN *Bagging* (Bifet *et al*, 2009) e o ASHT *Bagging* (Bifet *et al*, 2009). Os algoritmos DWM, ADWIN *Bagging* e ASHT *Bagging* são apresentados em maiores detalhes na seção 2.4, pois estes serviram de inspiração para o desenvolvimento do conjunto de classificadores proposto neste trabalho (SAE) e também foram utilizados nos experimentos.

2.4 Conjuntos de classificadores

Segundo (Polikar, 2006), dois componentes devem ser levados em consideração para a construção de um algoritmo baseado em conjunto de classificadores. O primeiro é a estratégia de construção do conjunto de forma que ele seja o mais diverso possível. O segundo diz respeito à combinação da saída dos classificadores de forma que decisões corretas sejam ressaltadas e as incorretas canceladas. Além destes, na configuração *online* é importante considerar algum método de adaptação do conjunto, uma vez que mudanças de conceito são esperadas.

O objetivo geral em conjuntos de classificadores é fazer com que cada classificador seja tão único quanto possível, particularmente com respeito aos erros de classificação (Polikar, 2006). Se um conjunto é formado por classificadores que predizem incorretamente instâncias diferentes, de forma a complementar-se, então este conjunto é dito diverso e possivelmente terá poder de classificação superior ao alcançável pelos classificadores que o formam quando considerados individualmente. Nem sempre a combinação dos classificadores resulta em uma taxa de acerto melhor. Por exemplo, considerando um problema contendo 10 instâncias e um conjunto formado por 3 classificadores, tal que cada um tem taxa de acerto igual a 60%, ou seja, é capaz de classificar corretamente 6 das 10 instâncias. Existem 28 variações possíveis de forma que cada um dos 3 classificadores possua 60% de taxa de acerto. A Tabela 2.3 apresenta estas 28 combinações (nas linhas) e o resultado da combinação da saída dos classificadores segundo a votação majoritária. Na Tabela 2.3 as colunas *a* até *h* indicam o somatório de classificações corretas feitas para as mesmas instâncias para os 3 classificadores simultaneamente (coluna *a*), para apenas o primeiro e último simultaneamente (coluna *b*), e assim sucessivamente. Na Tabela 2.3 a combinação ótima possui uma taxa de acerto de 90% (identificador 1). Na prática é improvável que a combinação ótima ocorra, porém é possível que uma boa combinação (linhas 2 a 12) ou equivalente à taxa de acerto individual (linhas 13 a 23) ocorram. Existe a possibilidade

da combinação dos classificadores alcançar uma taxa de acerto pior do que a taxa de acerto individual e isso ocorre com maior frequência para as situações em que os três classificadores classificam corretamente as mesmas instâncias (linhas 24 a 28).

Em (Kuncheva, 2004) são apresentadas 3 motivações para a aplicação de conjuntos de classificadores, sendo elas: Estatística, Computacional e Representacional.

Estatística. Considerando que a partir de D_K amostras de um conjunto de treinamento D sejam gerados K classificadores, tal que cada um deles possui 100% de taxa de acerto sobre o subconjunto D_K utilizado no seu treinamento. Possivelmente, a capacidade de generalização destes classificadores é diferente de forma que a aplicação dos mesmos sobre um conjunto disjunto de D (teste) incorreria em performances variadas. De um ponto de vista estatístico é mais seguro fazer uso da média das predições individuais destes K classificadores do que utilizar apenas um deles, desta forma o risco de selecionar o classificador com a pior generalização é eliminado. Existe a possibilidade de que a qualidade preditiva do conjunto não seja maior do que a do melhor classificador individual, contudo ela tende a ser melhor do que a do pior classificador individual.

Computacional. Alguns classificadores podem convergir para máximos locais. Supondo que os máximos locais de K classificadores se encontram próximos ao máximo global. Sendo assim, existe um modo de combiná-los que resulta em um classificador mais próximo do classificador ótimo do que qualquer um deles individualmente.

Representacional. O tipo do classificador utilizado pode não ser passível de representar a superfície de separação do problema. Por exemplo, um classificador baseado em árvores de decisão como o C4.5 (Quinlan, 1993) é capaz de realizar separações lineares, logo quando aplicado para um problema em que os dados estão distribuídos de forma não linear, uma única árvore de decisão não é capaz de alcançar o classificador ótimo, que para um problema não linear, também será não linear. Contudo, a combinação de um conjunto de classificadores lineares é capaz de aproximar uma superfície não linear de forma similar a um classificador não linear (Kuncheva, 2004).

ID	a	b	c	d	e	f	g	h	Acerto Geral	Geral - Individual
	111	101	011	001	110	100	010	000		
1	0	3	3	0	3	0	0	1	0.9	0.3
2	2	2	2	0	2	0	0	2	0.8	0.2
3	1	2	2	1	3	0	0	1	0.8	0.2
4	0	2	3	1	3	1	0	0	0.8	0.2
5	0	2	2	2	4	0	0	0	0.8	0.2
6	4	1	1	0	1	0	0	3	0.7	0.1
7	3	1	1	1	2	0	0	2	0.7	0.1
8	2	1	2	1	2	1	0	1	0.7	0.1
9	2	1	1	2	3	0	0	1	0.7	0.1
10	1	2	2	1	2	1	1	0	0.7	0.1
11	1	1	2	2	3	1	0	0	0.7	0.1
12	1	1	1	3	4	0	0	0	0.7	0.1
13	6	0	0	0	0	0	0	4	0.6	0
14	5	0	0	1	1	0	0	3	0.6	0
15	4	0	1	1	1	1	0	2	0.6	0
16	4	0	0	2	2	0	0	2	0.6	0
17	3	1	1	1	1	1	1	1	0.6	0
18	3	0	1	2	2	1	0	1	0.6	0
19	3	0	0	3	3	0	0	1	0.6	0
20	2	1	1	2	2	1	1	0	0.6	0
21	2	0	2	2	2	2	0	0	0.6	0
22	2	0	1	3	3	1	0	0	0.6	0
23	2	0	0	4	4	0	0	0	0.6	0
24	5	0	0	1	0	1	1	2	0.5	-0.1
25	4	0	0	2	1	1	1	1	0.5	-0.1
26	3	0	1	2	1	2	1	0	0.5	-0.1
27	3	0	0	3	2	1	1	0	0.5	-0.1
28	4	0	0	2	0	2	2	0	0.4	-0.2

Tabela 2.3: 28 combinações de 3 classificadores, adaptado de (Kuncheva, 2004)

Nas seções 2.4.1, 2.4.2, 2.4.3, 2.4.4 e 2.4.5 são apresentados, respectivamente, os algoritmos baseados em conjunto de classificadores *Bagging* (Breiman 1996), *Online Bagging* (Oza e Russel, 2003), *Dynamic Weighted Majority* (DWM) (Kolter e Maloof, 2003), *ADWIN Bagging* (Bifet et al, 2009) e *ASHT Bagging* (Bifet et al, 2009).

2.4.1 Bagging

O *Bootstrap Aggregating* (*Bagging*) forma um conjunto de classificadores de tamanho fixo (k), os quais são treinados a partir de subconjuntos de instâncias D_j . Os subconjuntos

(*bootstraps*) são formados por instâncias selecionadas de forma aleatória com reposição (reamostragem) a partir do conjunto de treinamento D . Todos os subconjuntos tem o mesmo número de instâncias N que corresponde ao número de instâncias no conjunto de treinamento.

A classificação de uma nova instância é obtida a partir da votação majoritária do conjunto. O *Bagging* obtém melhores resultados quando o classificador base utilizado é instável (e.g. árvore de decisão). Isso ocorre devido à diversidade obtida pelo conjunto, a qual permite uma representação mais abrangente da superfície de separação. Inicialmente em (Breiman, 1996) foram apresentados resultados empíricos e argumentos baseados em heurísticas para justificar o funcionamento do *Bagging*. Posteriormente, em (Bühlmann e Yu, 2002), a intuição por trás do *Bagging* que o indicava como uma técnica de redução de variância foi confirmada.

2.4.2 Online Bagging

O principal desafio para a aplicação do *Bagging* no contexto da Classificação *online* é a necessidade de possuir o conjunto de treinamento D completo antes de gerar os subconjuntos D_j para o treinamento de cada classificador. Em (Oza e Russel, 2001), os autores apresentam um processo para simular a geração de subconjuntos D_j para *streams* de dados. Dada observação de que a probabilidade individual de cada exemplo ser selecionado para um subconjunto é governada por uma distribuição Binomial, o que faz com que o processo de amostragem possa ser aproximado considerando cada instância de uma vez e aleatoriamente decidindo se esta instância será incluída no subconjunto sendo construído. O problema com essa solução é que ela depende que o tamanho do conjunto de treinamento N seja conhecido. Contudo, assumido que $N \rightarrow \infty$, uma assertiva válida para *streams* de dados, Oza e Russel concluíram que a distribuição Binominal tende a uma distribuição de *Poisson* com $\lambda = 1$. São apresentadas demonstrações e experimentos em (Oza e Russel, 2001), as quais comprovam que o *Bagging online* converge para o algoritmo *Bagging* (Breiman, 1996) original.

2.4.3 Dynamic Weighted Majority

O *Dynamic Weighted Majority* (DWM) não possui um tamanho fixo para o conjunto de classificadores, isto é, ele aumenta e diminui a quantidade de classificadores com base em heurísticas de atualização do conjunto. Assim como o algoritmo que inspirou a sua criação

(*Weighted Majority* (Littlestone e Warmuth, 1994)) o resultado da classificação é dado por meio da votação ponderada.

Para cada classificador, denominado *expert*, é atribuído um peso que representa a qualidade do mesmo. Este peso é reforçado quando o *expert* acerta e diminuído quando ele erra. A classificação do conjunto é dada pela decisão individual de cada *expert* ponderada sobre seu peso. Novos *experts* são adicionados sempre que a decisão do conjunto for incorreta. Esta heurística pode ser problemática para conjuntos de dados contendo um grau elevado de ruído, pois serão criados muitos classificadores. Como medida preventiva para este problema é definido um parâmetro³ que indica após quantas instâncias de treinamento (período) será feita uma nova tentativa de atualização do conjunto. *Experts* são removidos caso seu peso esteja abaixo de um limiar (parâmetro para o algoritmo).

No DWM a adaptação à mudança de conceito ocorre de forma natural. Quando uma mudança de conceito acontece os *experts* treinados no conceito anterior passam a decidir incorretamente sobre as novas instâncias afetando a decisão geral. Novos *experts* são criados a cada classificação incorreta do conjunto, ao passo que os *experts* antigos são removidos. Vale ressaltar que a heurística de atualização gradual do DWM resguarda o algoritmo contra ruídos nos dados, porém também evita uma rápida ação no caso de uma mudança abrupta de conceito.

No artigo original (Kolter e Maloof, 2003), os autores apresentam testes que evidenciam que os resultados obtidos pelo DWM não são influenciados apenas pela qualidade do classificador base, mas também pela estratégia de atualização. Nestes testes foram utilizados o algoritmo *Naïve Bayes* e o C4.5 como classificadores base.

2.4.4 ASHT *Bagging*

A ideia por trás do algoritmo *Adaptive-Size Hoeffding Tree (ASHT) Bagging* é a de que árvores de decisão menores (poucos níveis) se adaptam mais rapidamente a mudanças, ao passo que, árvores maiores (muitos níveis) obtêm performance melhor durante períodos com pequena ou nenhuma mudança. No algoritmo *ASHT Bagging* é almejado melhorar a performance do *Online Bagging* (seção 2.4.2) para *streams* de dados com mudança de conceito, aumentando a diversidade do conjunto a partir do emprego de árvores de decisão com alturas variadas.

³ Este parâmetro influencia indiretamente o tempo de execução do algoritmo, pois com menos *experts* as etapas de treinamento, teste e atualização ocorreram mais rapidamente.

Segundo (Bifet *et al*, 2011), foi observado que o método de *Boosting* (Freund e Schapire, 1997) tende a produzir um conjunto mais diverso do que o *Bagging*, e isso tem sido apontado como um fator de aumento da taxa de acerto do algoritmo.

O classificador base para o ASHT *Bagging* é o algoritmo *Adaptive-Size Hoeffding Tree* (ASHT). O ASHT é uma derivação do algoritmo *Hoeffding Tree* (Domingos e Hulten, 2000), em que é possível limitar o tamanho máximo da altura da árvore segundo um parâmetro. No ASHT, após ocorrida a divisão de um nodo, caso a altura máxima seja ultrapassada alguns nós devem ser removidos (operação também referenciada por *reset* da árvore), podendo isto ser feito de duas formas diferentes. A primeira opção é remover a raiz e todos os seus filhos mantendo apenas o nó que gerou a última divisão, tornando-o a nova raiz. A segunda opção é remover todos os nós, isto é, reiniciar completamente a árvore.

No ASHT *Bagging*, a altura máxima da árvore ASHT na posição n do conjunto é o dobro da altura máxima da árvore na posição $(n-1)$. Por exemplo, em um conjunto com 5 árvores, tal que a primeira tenha tamanho limite igual a 2, o tamanho limite das demais árvores será 4, 8, 16 e 32, respectivamente. Cada árvore tem um peso proporcional ao inverso do quadrado do seu erro, e tem seu erro monitorado a partir da média ponderada móvel exponencial (*exponential weighted moving average* ou EWMA) com $\alpha = 0.01$.

É importante observar que no ASHT *Bagging* as árvores são sempre reiniciadas mesmo em momentos em que não estejam ocorrendo mudanças de conceito. Contudo, esse comportamento não influencia negativamente a capacidade de classificação do conjunto devido ao peso atribuído a cada árvore. As árvores menores são reiniciadas com maior frequência e conseqüentemente se adaptam melhor a regiões de mudança. Em contrapartida, as árvores maiores demoram mais para serem reiniciadas e acumulam mais “conhecimento” sobre os dados, obtendo assim melhores resultados para regiões estáveis (sem mudança de conceito).

2.4.5 ADWIN *Bagging*

O ADWIN *Bagging* combina o método de *Online Bagging* (seção 2.4.2) com o algoritmo *Adaptive Window* (ADWIN) (Bifet e Gavaldà, 2007) para detecção de mudanças. O ADWIN pertence ao grupo de algoritmos detectores de mudança (seção 2.3.2). O seu emprego é adequado em problemas em que as instâncias possuem atributos contínuos. O ADWIN mantém uma janela

de tamanho variável com as últimas instâncias na forma de um histograma. Esta janela tem o tamanho máximo estatisticamente consistente com a hipótese: “não houve mudanças no valor médio dentro desta janela” (Bifet *et al*, 2011). Um fragmento da janela é descartado caso não existam evidências de que o valor médio dele seja diferente do valor médio do resto da janela. Mudanças são identificadas a partir da análise do tamanho da janela, isto é, quando a janela diminui de tamanho existe uma indicação de mudança de conceito.

2.5 Métodos de avaliação

O comportamento de um algoritmo de Classificação *Online* possui três dimensões de interesse: espaço, tempo e taxa de acerto. É relevante ressaltar que estas três medidas são interdependentes, isto é, ao aumentar o espaço em memória que um algoritmo pode utilizar, possivelmente, este terá uma maior taxa de acerto e executará mais rapidamente, o contrário tende a ser verdade também, ou seja, se o espaço for muito reduzido o algoritmo provavelmente deverá adaptar o modelo (descartar informações) para que este não ultrapasse os limites especificados pelo usuário, causando uma degradação da taxa de acerto e maior tempo de processamento.

É comum que a medida de maior interesse quando observada a performance de um classificador seja a sua taxa de acerto. A taxa de acerto é tipicamente medida como a porcentagem de classificações corretas, ou seja, o algoritmo com maior precisão é aquele que comete menos erros de classificação. Esta avaliação deve ser feita com um conjunto de dados separado daquele utilizado para treinar o algoritmo, normalmente denominado conjunto de teste. Nas seções 2.4.1 e 2.4.2 são apresentados os métodos usuais de avaliação de classificadores para a configuração *batch* e *online*, respectivamente, no que diz respeito à distribuição dos conjuntos de treinamento e teste.

2.5.1 Configuração *batch*

Em geral, os problemas envolvendo classificação na configuração *batch* tratam de tirar o maior proveito possível de uma quantidade limitada de dados. Isto ocorre, porque quando existe uma escassez de dados é interessante que todos sejam utilizados para a construção (treinamento) do modelo. Porém se todos os dados forem utilizados para treinamento não sobram dados para testar o modelo e se não for possível avaliar o modelo sobre um conjunto disjuncto dos dados de

treinamento, não é possível chegar a conclusões convincentes sobre a taxa de acerto do modelo, e mais importante ainda, o poder de generalização⁴ de tal modelo não pode ser mensurado.

A seguir são apresentadas estratégias de especificação (separação) dos conjuntos de treinamento e teste, as quais podem ser estudadas em maiores detalhes em (Kohavi, 1995).

Holdout. Divide os dados em dois conjuntos mutualmente exclusivos, o conjunto de treinamento e o de teste (ou *holdout*). Manter os dois conjuntos separados garante que a capacidade de generalização do algoritmo pode ser mensurada. A maior crítica a este método é que os dados não são utilizados eficientemente, pois alguns exemplos jamais serão utilizados para treinamento. Isto é um problema ainda maior para situações onde a quantidade de dados é muito baixa.

Cross-validation (validação cruzada). Maximiza o uso de exemplos tanto para treinamento quanto para teste. Na validação cruzada de k partes (*k-fold cross-validation*) os dados são aleatoriamente divididos em k partes independentes de tamanhos aproximadamente iguais, sucessivamente uma parte age como o conjunto de teste enquanto as demais são combinadas para treinamento, até que todas tenham sido utilizadas uma vez como conjunto de teste. A estimativa final sobre a taxa de acerto é obtida a partir da divisão do número total de classificações corretas pelo número total de exemplos.

Leave-one-out. É um caso especial de validação cruzada onde cada parte contém apenas um único exemplo. O processo ocorre da mesma forma que na validação cruzada com a diferença de que o conjunto de teste sempre é formado por apenas um exemplo. A característica mais atrativa deste método é que ele é determinístico, diferente da validação cruzada que depende de uma distribuição aleatória dos dados, porém para classificadores onde não é trivial que o modelo “esqueça” um exemplo esta abordagem torna-se custosa, pois exige que N modelos sejam criados, tal que N é o número total de exemplos.

2.5.2 Configuração *online*

Quando é realizada a avaliação de um classificador na configuração *online* uma das maiores preocupações é referente a construção de um método de avaliação capaz de ilustrar consistentemente a taxa de acerto do algoritmo ao longo do tempo. A preocupação relativa ao

⁴ Generalização é a capacidade de prever exemplos que não foram usados para treinamento, isto é, desconhecidos pelo classificador.

máximo aproveitamento dos exemplos disponíveis para a classificação existente na configuração *batch* não é tão importante na configuração *online*, uma vez que nesta a quantidade de exemplos é considerada abundante. A seguir são apresentados três métodos de avaliação para a configuração *online*, sendo que o primeiro trata-se de uma extensão natural do método *Holdout* e os outros dois exploram propriedades únicas da configuração *online*.

Periodic Holdout. Para rastrear a evolução (ou degradação) da taxa de acerto do modelo ao longo do tempo, este pode ser avaliado utilizando o método *Holdout* como na configuração *batch* periodicamente (e.g. a cada milhão de exemplos). Testar o algoritmo muito frequentemente pode influenciar negativamente o tempo (de processamento) para a avaliação do modelo. Os exemplos do conjunto *Holdout* (teste) ainda não devem ter sido considerados para o treinamento, ou seja, devem ser exemplos recém disponibilizados pela *stream*. Caso haja a necessidade de tirar o maior proveito possível dos dados, após o término do teste os exemplos do conjunto *Holdout* podem então ser utilizados para treinar o modelo.

Interleaved Test-Then-Train. Este método intercala treinamento e teste, tal que cada exemplo é utilizado primeiro para teste e logo em seguida para treinar o classificador. Desta forma é garantido o máximo aproveitamento dos exemplos disponíveis. O teste de forma incremental provê a taxa de acerto média do modelo. Esta característica pode ser interpretada como uma desvantagem do método, pois dificulta a análise do real potencial de classificação do classificador em um ponto específico no tempo, uma vez que a taxa de acerto neste momento é influenciada por erros nas classificações anteriores. Além disso, mensurar o tempo de processamento das etapas de treinamento e de teste torna-se difícil neste modelo.

Prequential. Neste método o teste é intercalado com o treinamento de modo similar ao *Test-Then-Train*. A diferença reside na existência de um fator de desvanecimento (*fading factor*), o qual diminui o impacto que as predições incorretas iniciais têm sobre as predições futuras. Em (Gama, Sebastião e Rodrigues, 2009) é demonstrado que o *Prequential* converge para resultados equivalentes aos obtidos pelo *Periodic Holdout*.

Considerando as vantagens e desvantagens dos métodos de avaliação apresentados, no presente trabalho o *Prequential* foi selecionado para a realização dos experimentos. A justificativa para essa escolha é devido ao *Prequential* incluir características desejáveis pertencentes aos outros dois métodos, como o aproveitamento dos dados de teste para

treinamento (*Test-Then-Train*) e a apresentação da taxa de acerto para o período atual (*Periodic Holdout*).

2.6 Geradores de dados

Geradores de dados são frequentemente utilizados para a avaliação de métodos de Classificação *Online*. Essa preferência se deve à flexibilidade que geradores oferecem com relação às configurações dos experimentos, além de facilitarem a repetição destes por outros pesquisadores. Os geradores utilizados nos experimentos deste trabalho são o *Random Tree Generator* (Domingos e Hulten, 2000), SEA (Street e Kim, 2001) e AGRAWAL (Agrawal *et al*, 1993), apresentados nas seções 2.6.1, 2.6.2 e 2.6.3 respectivamente.

2.6.1 *Random Tree Generator*

O gerador *Random Tree Generator* (RTG) (Domingos e Hulten, 2000) constrói uma árvore de decisão selecionando cada atributo aleatoriamente para dividir os nodos e assina-la uma classe aleatória para cada folha. Após a construção da árvore, novos exemplos são gerados a partir da atribuição de valores aleatórios uniformemente distribuídos para cada atributo. A classe destes novos exemplos é obtida a partir do caminharmento na árvore segundo os valores de seus atributos. É possível alterar a quantidade de atributos numéricos, nominais e também a quantidade de classes para o RTG. O RTG não inclui mudanças de conceito, isto é, gera um fluxo de dados estável. Devido ao uso de uma estrutura de árvore de decisão para gerar os dados é assumido que o RTG tende a favorecer classificadores baseados em árvores de decisão (Bifet *et al*, 2012).

2.6.2 SEA

O gerador SEA (Street e Kim, 2001) gera *streams* de dados com três atributos contínuos. Todos os três atributos (f_1, f_2 e f_3) tem valores variando entre 0 e 10, contudo apenas os dois primeiros (f_1 e f_2) são relevantes para determinação da classe. São gerados pontos aleatórios em um espaço com três dimensões, cada uma correspondendo a um dos três atributos. Os pontos são divididos em 4 blocos que correspondem a 4 diferentes funções (conceitos). Em cada bloco um ponto pertence à classe 1 se $f_1 + f_2 \leq \theta$ e a classe 0 caso contrário. O limiar θ , utilizado para

separar as duas classes, assume os valores 8 (bloco 1), 9 (bloco 2), 7 (bloco 3) e 9.5 (bloco 4). É possível inserir ruído nos valores de classe, sendo o valor padrão 10%.

2.6.3 AGRAWAL

O gerador AGRAWAL (Agrawal *et al*, 1993) produz *streams* de dados contendo seis atributos contínuos e três nominais. Em (Agrawal *et al*, 1993) foram definidas dez funções diferentes para gerar valores de classes binárias a partir dos valores dos atributos. Na Tabela 2.4 são apresentados os atributos e seus respectivos intervalo de valores. Na Figura 2.4 são apresentadas três das dez funções definidas para determinação dos valores de classe⁵.

Para simular ruídos nos dados é incluído um fator de perturbação no gerador. Este fator altera o valor real de um atributo adicionando um valor de desvio, definido de acordo com uma distribuição aleatória uniforme. O intervalo de valores para o desvio é uma porcentagem do intervalo do atributo (fator de perturbação).

Atributo	Descrição	Tipo do atributo	Intervalo de valores
<i>Salary</i>	salário	Contínuo	uniformemente distribuído entre 20 mil e 150 mil
<i>Commission</i>	comissão	Contínuo	Se salary < 75 mil então 0 senão uniformemente distribuído entre 10 mil e 75 mil
<i>Age</i>	idade	Contínuo	uniformemente distribuído entre 20 e 80
<i>Elevel</i>	nível educacional	Nominal	uniformemente selecionado entre 0 e 4
<i>Car</i>	fabricante do carro	Nominal	uniformemente selecionado entre 1 e 20
<i>Zipcode</i>	zipcode da cidade	Nominal	uniformemente selecionado entre 9 zipcodes
<i>Hvalue</i>	valor da casa	Contínuo	uniformemente distribuído entre 0.5k100000 e 1.5k100000, tal que $k \in \{1..9\}$ variando de acordo com o valor de zipcode
<i>Hyears</i>	a quantos anos possui a casa	Contínuo	uniformemente distribuído entre 1 e 30
<i>Loan</i>	valor do empréstimo	Contínuo	uniformemente distribuído entre 0 e 500 mil

Tabela 2.4: Atributos do gerador AGRAWAL, adaptado de (Bifet *et al*, 2012)

⁵ As outras 7 funções podem ser encontradas em (Bifet *et al*, 2012) na página 35.

Função 1	Função 2
Se $age < 40$ OU $age \geq 60$ então $classe = A$ Senão $classe = B$	Se $((age < 40) \text{ E } (50000 \leq salary \leq 100000))$ OU $((40 \leq age < 60) \text{ E } (75000 \leq salary \leq 125000))$ OU $((age \geq 60) \text{ E } (25000 \leq salary \leq 75000))$ então $classe = A$ Senão $classe = B$
Função 3	
Se $((age < 40) \text{ E } (elevel \in [0...1]))$ OU $((40 \leq age < 60) \text{ E } (elevel \in [1...3]))$ OU $((age \geq 60) \text{ E } (elevel \in [2...4]))$ então $classe = A$ Senão $classe = B$	

Figura 2.4: Três funções do gerador AGRAWAL, adaptado de (Bifet *et al*, 2012)

2.7 Considerações finais

Neste capítulo foram elucidados os conceitos básicos de Classificação *Online* como o ciclo de classificação, o problema de mudança de conceito, os métodos de avaliação e como eles diferem daqueles existentes para a Classificação *batch*, as dimensões pelas quais um problema de Classificação *Online* pode ser mensurado (taxa de acerto, tempo e espaço) e finalmente foram apresentados algoritmos baseados em conjunto de classificadores. O conjunto de classificadores enunciado neste trabalho faz uso do método de *Bagging Online* para treinamento dos classificadores, além de empregar uma estratégia de atualização (adição/remoção) similar aquela utilizada pelo algoritmo DWM. Os experimentos apresentados neste trabalho foram realizados sobre os geradores de dados RTG, SEA e AGRAWAL. Os resultados dos experimentos foram reportados com base no método de avaliação *Prequential* devido as vantagens que este apresenta sobre os demais métodos para a avaliação de *streams* de dados com mudança de conceito. No próximo capítulo são apresentados os conceitos fundamentais sobre Redes Sociais, os quais são de suma importância para a compreensão do conjunto de classificadores proposto neste trabalho.

Capítulo 3

Análise de Redes Sociais

Redes sociais proveem uma maneira precisa de definir conceitos sociais importantes (Wasserman e Faust, 1994). Os modelos baseados em redes conceitualizam estruturas sociais (comportamentais, econômicas, políticas, etc.) como padrões de relações entre indivíduos. A interpretação de um problema sob a perspectiva de uma rede permite responder a perguntas comuns sobre ciências sociais de modo preciso e formal. A perspectiva da rede social engloba teorias, modelos e aplicações que são expressas por meio de conceitos relacionais e formalismos matemáticos embasados na teoria dos grafos e na estatística (Wasserman e Faust, 1994).

O método de conjunto de classificadores proposto neste trabalho se vale de diversos conceitos presentes na Análise de Redes Sociais (ARS). Por exemplo, a disposição dos classificadores formadores do conjunto é feita sob a perspectiva da ARS, ou seja, cada classificador é um ator. A partir desta abstração do conjunto de classificadores é possível avaliá-lo a partir de medidas de proeminência (seção 3.3) ou ainda detectar mudanças topológicas (seção 3.4) na rede em decorrência de uma mudança de conceito.

No restante deste capítulo são apresentados os conceitos fundamentais da análise de redes sociais (3.1), teoria de grafos (seção 3.2), medidas de proeminência (seção 3.3) e topologias de redes sociais (seção 3.4).

3.1 Conceitos fundamentais

Uma rede social é formada por um conjunto de atores, uma ou mais relações e um conjunto de conexões entre pares de atores. As relações definem quais conexões podem existir entre um par qualquer de atores. Os atores (e suas ações) são vistos como interdependentes ao invés de unidades autônomas independentes. As conexões são interpretadas como canais para transferência ou fluxo de recursos materiais ou abstratos (Wasserman e Faust, 1994).

A característica mais básica da Análise de Redes Sociais (ARS), que a difere de outros tipos de análises, é o uso de informação estrutural para estudar e testar teorias. A estrutura da rede é interpretada como um provedor de oportunidades ou de restrições para as ações do indivíduo (Wasserman e Faust, 1994). Apesar de ser possível coletar e analisar atributos inerentes apenas aos indivíduos, o foco da ARS são as medidas e inferências que podem ser realizadas a partir das conexões entre indivíduos. Conseqüentemente, as medidas realizadas em uma rede social não podem ser feitas utilizando métodos estatísticos tradicionais (teste t, correlação canônica, etc.). Algumas das medidas praticáveis em uma rede social, denominadas medidas de proeminência ou variáveis estruturais, são detalhadas na seção 3.3

A ARS provê métodos para estudar propriedades estruturais em diferentes níveis de agregação, como: atores individuais, pares de atores (díades), trios de atores (tríades), subgrupos ou ainda a rede completa. Os conceitos formais relativos aos diferentes níveis de agregação são explicados na seção 3.2. As teorias e preocupações substantivas ao redor do problema servem para determinar quais níveis devem ser estudados, além de determinar quais variáveis estruturais devem ser analisadas.

A Análise de Redes Sociais é inerentemente um esforço interdisciplinar (Wasserman e Faust, 1994). A vantagem da fundamentação matemática para representação de conceitos sociais é o desenvolvimento de modelos de processos testáveis e teorias explicativas. Segundo (Wasserman e Faust, 1994), as três principais fundações matemáticas para ARS são:

- Teoria dos Grafos: Provê propriedades de representação de uma rede social e um conjunto de conceitos que pode ser usado para o estudo formal de suas propriedades.
- Estatística: É importante para estudos que envolvem reciprocidade (*reciprocity*), mutualidade (*mutuality*), equilíbrio (*balance*) e transitividade (*transitivity*).

- Modelos algébricos: Amplamente utilizados para estudar redes multi-relacionais. Estes modelos utilizam operações algébricas para estudar combinações de relações, como por exemplo, “é amigo de” e “pede conselhos a”.

3.2 Teoria dos Grafos

A teoria dos grafos provê uma linguagem para denotar propriedades estruturais das redes sociais. Esta linguagem inclui um conjunto de conceitos fundamentais que podem ser usados para referenciar de forma precisa e inequívoca tais propriedades. Além disso, a formalização matemática, inerente à teoria dos grafos, permite que ideias, aparentemente subjetivas, sejam expressas de forma precisa possibilitando que estas sejam mensuradas, reproduzidas e testadas. Consequentemente, o uso de grafos para representar estruturas sociais permite a prova de teoremas e a dedução de afirmações que podem ser comprovadas por meio de testes determinísticos.

A representação visual de uma rede social por meio de um grafo é comumente denominada sociograma (sociogram⁶). Esta representação visual permite aos pesquisadores evidenciar padrões que de outra forma provavelmente passariam despercebidos. Por exemplo, na Figura 3.1 é apresentado um sociograma que demonstra a afinidade entre colegas de turma. Visualmente, fica evidente a preferência de todos os colegas por Jonas. Contudo, para muitos problemas torna-se impraticável a análise utilizando a representação visual do sociograma devido à grande quantidade de elementos que o compõe. Alternativamente, um sociograma pode ser representado completamente por meio de uma matriz, comumente denominada sociomatriz (*sociomatrix*). O uso de uma sociomatriz facilita a representação computacional, porém não torna mais simples a análise visual de uma rede com muitos elementos.

⁶ O termo *sociogram* foi empregado pela primeira vez por Moreno em (Moreno, 1934).

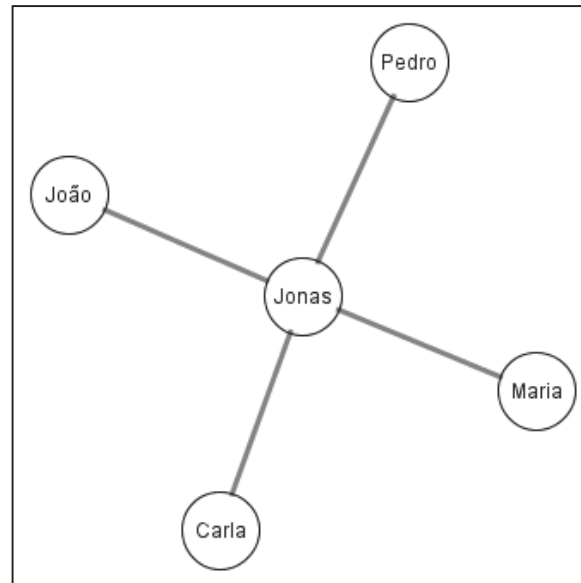


Figura 3.1: Exemplo de Sociograma “Afinidade entre colegas de turma”

As seções seguintes enunciam conceitos relativos à teoria de grafos diretamente ligados a este trabalho. Na seção 3.1.1 é apresentada a terminologia e os conceitos fundamentais como grafo, subgrafo, díades e tríades. Na seção 3.1.2 a atenção é voltada para propriedades de nodos individuais, como o grau do nodo, e propriedades pertinentes a um conjunto de nodos como o grau médio de um grafo (ou subgrafo) a sua variância e densidade. Na seção 3.1.3 são discutidos os diversos tipos de caminhamentos e conceitos deles derivados como distância geodésica, excentricidade e diâmetro. Na seção 3.1.4 é apresentado o conceito de conectividade e suas implicações como pontes e linhas de corte.

3.2.1 Conceitos

Em sua forma mais essencial um grafo pode ser dividido em dois tipos, grafos direcionados e não direcionados. Um grafo não direcionado representa uma relação onde a ordem dos componentes não é relevante, por exemplo: a relação “**é irmão de**”, “João **é irmão de** Pedro” é intercambiável por “Pedro **é irmão de** João”. Um grafo direcionado, ou dígrafo, representa uma relação onde a ordem dos componentes permite diferentes interpretações, por exemplo: na relação “**é pai de**”, “João **é pai de** Pedro” tem um sentido diferente de “Pedro **é pai de** João”.

Formalmente um grafo G é composto por dois conjuntos de informação: o conjunto de nodos, $N = \{n_1, n_2, \dots, n_g\}$ e o conjunto de conexões, $C = \{c_1, c_2, \dots, c_L\}$ entre pares de nodos. Uma conexão c qualquer existe entre dois nodos desde que estes satisfaçam a relação dicotômica⁷ R do grafo. Uma conexão c_x é formada por um par não ordenado de nodos, tal que $c_x = (n_i, n_j)$. A conexão c_x somente existirá se n_i e n_j atenderem a relação R inerente ao grafo G . Uma vez que $(n_i, n_j) = (n_j, n_i)$, não existe nenhuma vantagem em incluir ambas no conjunto C , portanto apenas uma das duas deve existir em C . G é dito um grafo simples, caso nele não existam conexões do tipo *loop* (laço reflexivo) e ele defina apenas uma relação⁸. Um *loop* ou laço reflexivo é uma conexão de um nodo com ele mesmo, $c_y = (n_k, n_k)$. A cardinalidade do conjunto de nodos N , $|N|$, e do conjunto de conexões C , $|C|$, são denotadas por g e L , respectivamente. Neste trabalho toda vez que for citado apenas o termo “grafo” assume-se que tratasse de um grafo simples.

Os nodos de um grafo representam os atores da rede social. As conexões definem conceitos subjetivos sobre os atores, como “Pedro **trabalhou com** Carlos”. Na Figura 3.2 é apresentado um sociograma que demonstra a relação “**trabalhou com**” entre colegas e ex-colegas de trabalho.

Um nodo n_i é dito adjacente a outro nodo n_j se existe uma conexão c_x , tal que $c_x \in C$. O nodo n_i é incidente com uma conexão c_x , e vice-versa, se e somente se $n_i \in c_x$. Grafos onde o conjunto C é vazio, $C = \emptyset$, são denominados grafos vazios. Grafos com apenas um nodo são denominados triviais, todos os demais grafos são denominados não triviais. Como ressaltado em (Wasserman e Faust, 1994), grafos triviais e vazios são de pouco (ou nenhum) interesse no contexto da ARS.

⁷ Uma relação dicotômica (*dichotomous tie*), em um grafo, representa a presença ou ausência da conexão entre dois nodos. Uma relação valorada (ou ponderada), em um grafo, associa um peso na conexão entre dois nodos.

⁸ Um grafo pode ser multi-relacional. Estes tipos de grafos permitem mais de uma relação, consequentemente dois nodos podem ter mais de uma conexão entre si representando diferentes relacionamentos.

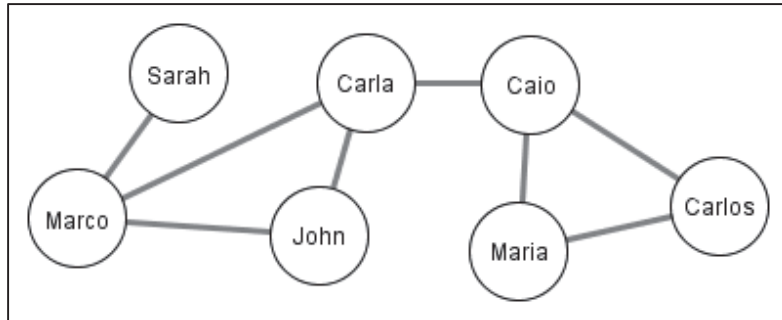


Figura 3.2: Grafo representando a relação “trabalhou com”.

Uma das características discriminantes das redes sociais é a possibilidade de realizar análises em diferentes níveis. Pode-se analisar o problema considerando toda a rede, somente uma parte dela ou a nível de indivíduos. Parte de uma rede social pode ser representada por um subgrafo. Formalmente G_s é um subgrafo de G desde $N_s \subset N$ e $C_s \subset C$, tal que N_s denota os nodos de G_s e C_s as suas conexões. Um subgrafo G_s qualquer pode não incluir todas as conexões incidentes sobre seus os nodos, isso depende da forma com que o subgrafo foi obtido. Existem ao menos duas maneiras de se extrair um subgrafo G_s a partir de um grafo G . A primeira delas é selecionando um subconjunto de nodos em G e incluindo apenas as conexões existentes entre nodos pertencentes a este subconjunto, este tipo de subgrafo é denominado gerado por nodos (*node-generated*). A segunda maneira é selecionando um subconjunto de conexões em G e incluindo apenas os nodos presentes no subconjunto, este tipo de subgrafo é denominado gerado por conexões (*line-generated*). Exemplos de subgrafos gerados a partir do grafo da Figura 3.2 podem ser visualizados na Figura 3.3. Vale ressaltar que no subgrafo (a) da Figura 3.3 a conexão “Carla-Caio” é omitida devido ao nodo “Caio” não estar dentre os nodos determinados para o subgrafo.

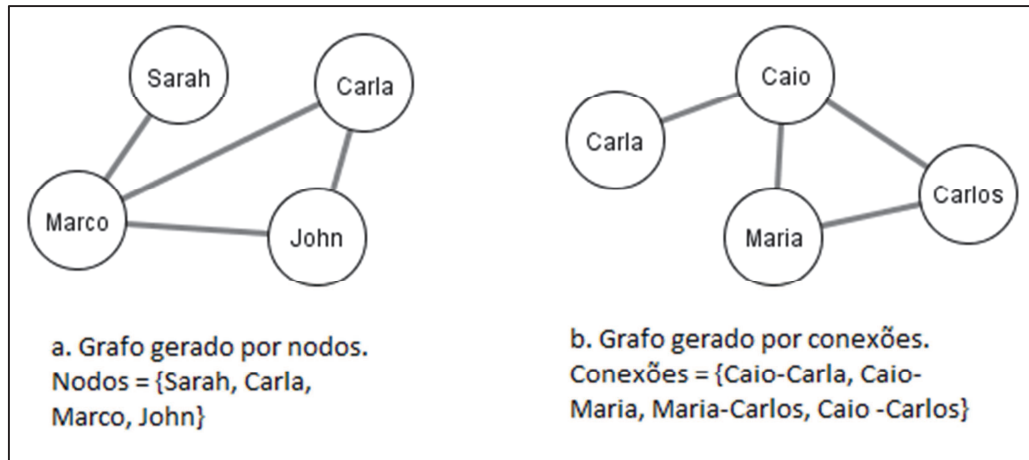


Figura 3.3: Subgrafos gerados por nodos (a) e por conexões (b)

Subgrafos gerados por nodos são amplamente utilizados na análise de subgrupos coesos (*cohesive subgroups*). Estes métodos focam em subconjuntos de atores dentre os quais as conexões são relativamente fortes, numerosas ou próximas (Wasserman e Faust, 1994).

Subgrafos formados por pares de atores (díades) ou trios (tríades) tem um papel fundamental na ARS. Uma díade (*dyad*) representa um par de atores e a possível conexão entre eles. Um conjunto não ordenado de nodos em um grafo qualquer pode estar em um de dois possíveis estados: ambos os nodos são adjacentes ou nenhum deles é. Consequentemente, existem apenas dois estados diádicos para uma relação não direcionada em um grafo, presença ou ausência. Um subgrafo onde todas as díades são adjacentes é denominado *clique* (subgrupo fechado).

Uma tríade (*triad*) é composta por três nodos e suas respectivas conexões. Um trio de atores pode assumir quatro estados possíveis, representados na Figura 3.4.

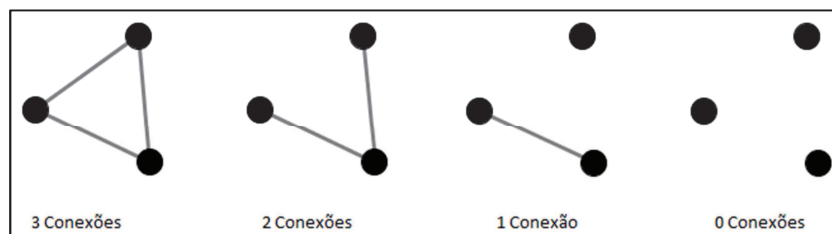


Figura 3.4: Estados possíveis da tríade

Em dígrafos é de grande interesse identificar se a tríade é transitiva, por exemplo, se “ i gosta de j ” e “ j gosta de k ” então “ i gosta de k ”, e se ela é equilibrada, por exemplo, se “ i gosta de j ” e “ j gosta de i ” então i e j devem ter uma opinião similar sobre um terceiro ator k (i.e. ambos gostam ou desgostam de k), e se “ i desgosta de j ” e “ j desgosta de i ”, então i e j devem ter uma opinião diferente sobre k (i.e. se um gosta o outro desgosta). Em grafos, um exemplo de medida analisada ao nível de tríades é a tríade proibida (*forbidden triad*) de Granovetter. Esta tríade proibida é aquela onde apenas duas conexões estão presentes. Granovetter argumenta em (Granovetter, 1973) que se um ator x tem uma conexão com o ator y e o ator x mantém uma conexão com o ator z , é improvável que a conexão entre o ator y e z não exista (tríade desequilibrada).

3.2.2 Propriedades

Muitas das propriedades discutidas nesta seção servem de base, ou são utilizadas diretamente, para o cálculo das medidas de proeminência em redes sociais (seção 3.2). Ao final desta seção é apresentado um exemplo que compreende todas as propriedades descritas.

Grau do nodo. O grau de um nodo, denotado por $d(n_i)$, representa o número de conexões incidentes sobre ele, ou (equivalentemente) a quantidade de nodos adjacentes a ele. O valor do grau pode assumir um valor no intervalo $[0, g-1]$. No contexto da ARS, o grau de um nodo mensura a sua “atividade”. O cálculo do grau do nodo é apresentado na equação 3.1.

$$d(n_i) = \sum_{c_x \in C \wedge n_i \in c_x} c_x \quad (3.1)$$

Nodo Isolado. Nodos com $d(n_i) = 0$ são denominados isolados.

Grau médio. O grau de médio, denotado por \bar{d} , sumariza o grau de todos os nodos do grafo em um valor médio. É possível calcular o valor médio somando todos os graus individuais e dividindo por g , equação 3.2, ou simplesmente dividindo $2L$ por g , equação 3.3.

$$\bar{d} = \frac{\sum_{i=1}^g d(n_i)}{g} \quad (3.2)$$

$$\bar{d} = \frac{2L}{g} \quad (3.3)$$

Variância. A variância, denotada por S_D^2 , indica diferenças na quantidade de conexões incidentes em cada nodo. Ela é calculada utilizando a equação 2.4.

$$S_D^2 = \frac{\sum_{i=1}^g (d(n_i) - \bar{d})^2}{g} \quad (3.4)$$

D-regular. Um grafo com é dito d -regular, tal que d assume o valor do grau comum a todos os nodos, quando todos os nodos tem o mesmo grau e consequentemente $S_D^2 = 0$. A caracterização de um grafo como d -regular indica uma regularidade na distribuição das conexões entre os nodos.

Densidade. A densidade de um grafo, denotada por Δ , representa a proporção de conexões existentes (L) relativas ao total de conexões possíveis. O total de conexões é equivalente ao total de pares não ordenados de nodos possíveis, isto é C_2^g , equação 3.5. A densidade varia de 0, se $L = 0$, até 1, se $L = g \cdot (g - 1)/2$. O cálculo da densidade é apresentado na equação 3.6.

$$C_2^g = \binom{g}{2} = \frac{g!}{2!(g-2)!} = \frac{g \cdot (g-1)}{2} \quad (3.5)$$

$$\Delta = \binom{g}{2} = \frac{L}{g \cdot (g-1)/2} = \frac{2L}{g \cdot (g-1)} \quad (3.6)$$

Grafo completo. Um grafo onde todos os nodos são adjacentes é denominado completo. Tal grafo possui $\Delta = 1$ e, conseqüentemente, todas as conexões possíveis estão presentes ($L = g(g - 1)/2$).

Todas as propriedades que foram discutidas nesta seção podem ser mensuradas em um grafo G qualquer ou em um de seus subgrafos. Na Figura 3.5 é apresentado um exemplo com as medidas para cada uma das propriedades do grafo. Na Figura 3.6 é apresentado um grafo completo e d -regular. Na Figura 3.7 é apresentado um grafo com um nodo isolado.

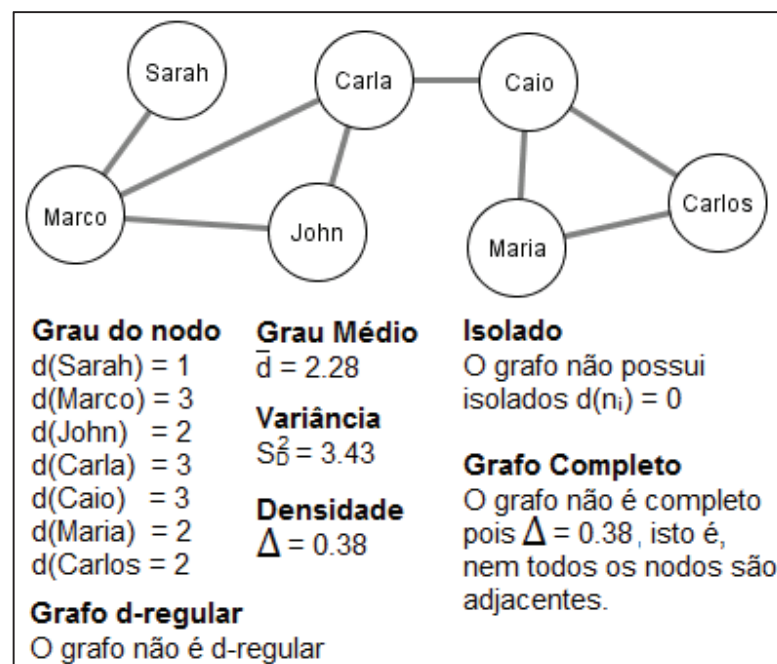


Figura 3.5: Um grafo e suas propriedades

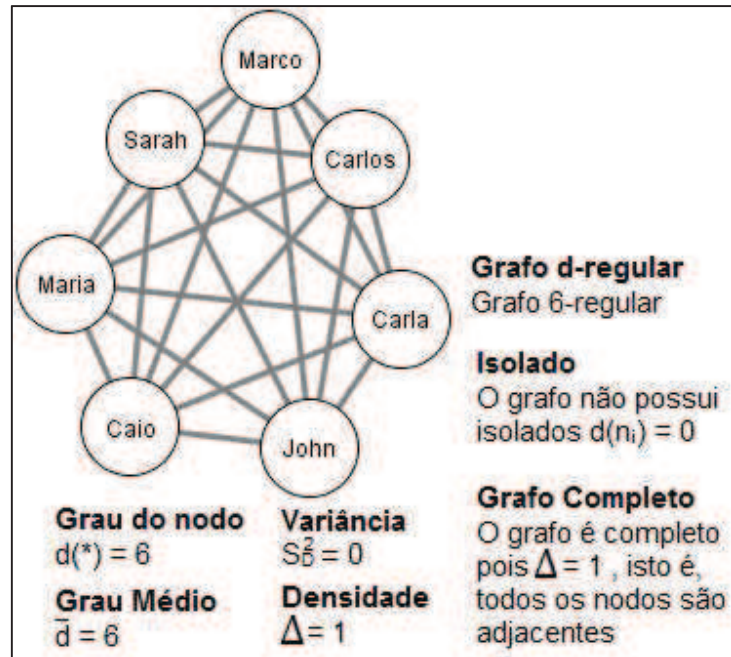


Figura 3.6: Um grafo completo e d-regular

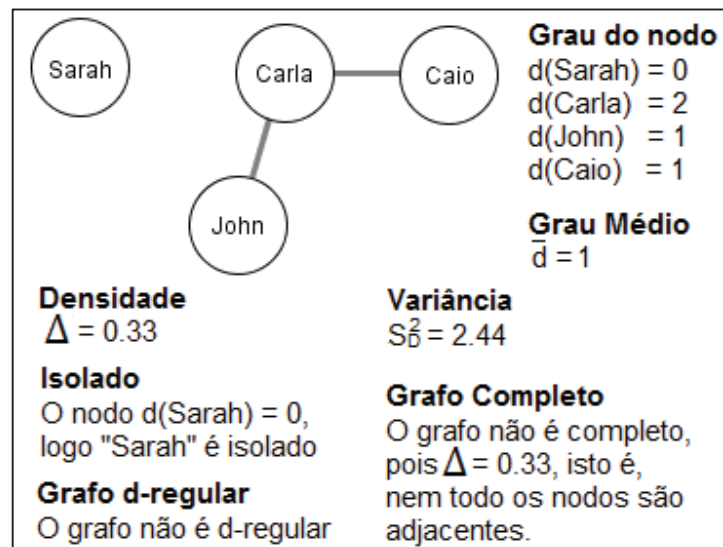


Figura 3.7: Grafo com um nodo isolado

3.2.3 Caminhamentos

Além de analisar as conexões diretas (ocorrem entre nodos adjacentes) pode ser interessante analisar se um nodo é “alcançável” a partir de outro nodo. Em um grafo quando existem conexões entre nodos adjacentes de forma a criar uma sequência partindo de um nodo n_i para outro n_j diz-se que n_i é indiretamente ligado a n_j e vice-versa, ou que existe uma conexão fraca (*weak tie*) entre n_i e n_j .

Na ARS é frequentemente importante saber se é possível alcançar um ator n_i a partir de outro ator n_j . Caso seja possível, é interessante também saber de quantas formas isso pode ser feito, e quais dessas são ótimas com relação à quantidade de atores entre n_i e n_j . A seguir são apresentados os conceitos de caminhada (*walk*), trilha (*trail*), caminho (*path*), caminhada fechado (*closed walk*), jornada (*tour*) e ciclos (*cycles*).

Caminhamento. Um caminhada, denotado por W , é uma sequência de nodos e conexões, iniciando e finalizando em nodos, onde cada nodo é incidente com as conexões que o sucedem e o precedem na sequência (Wasserman e Faust, 1994). O nodo inicial e o final não precisam ser o mesmo. Nodos e conexões podem ocorrer mais de uma vez no caminhada. O comprimento do caminhada é dado pelo número de conexões presentes nele. Caso uma conexão aconteça mais de uma vez no caminhada, ela é contada novamente. O primeiro e o último nodo de um caminhada são denominados origem (*origin*) e término (*terminus*), respectivamente. O caminhada inverso, denotado por W^{-1} , é o caminhada W listado exatamente na ordem oposta (Wasserman e Faust, 1994).

Trilha. Uma trilha, denotada por T , é um tipo mais restrito de caminhada onde todas as conexões são distintas, porém os nodos podem repetir.

Caminho. Um caminho, denotado por P , é um tipo mais restrito de trilha onde além de existirem apenas conexões distintas, todos os nodos também devem ser distintos. No contexto da ARS os caminhos são comumente mais estudados do que os caminhamentos. Uma possível justificativa para essa preferência é a inexistência de ambiguidades na representação de caminhos, isto é, um caminho com os mesmos nodos e as mesmas conexões sempre terá o mesmo comprimento, isso não é verdade para caminhamentos.

Na Figura 3.8 é possível identificar tanto o caminhamento W_1 , quanto a trilha T_1 visitam os mesmos nodos que o caminho P_1 , porém o fazem revisitando nodos (T_1) e conexões (W_1).

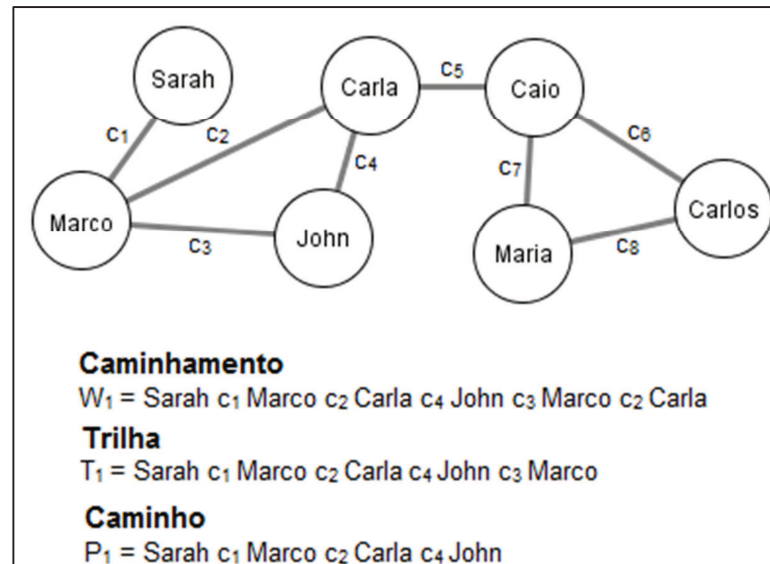


Figura 3.8: Caminhamentos, trilhas e caminhos.

3.2.4 Conectividade

A partir dos conceitos de caminhamento é possível derivar outras propriedades úteis inerentes a grafos, como por exemplo, a menor distância entre dois nodos (geodésico), a excentricidade de um nodo, diâmetro do grafo e sua conectividade (conexo ou desconexo). Um grafo conexo é aquele onde existe pelo menos um caminho entre cada par de nodos, ou seja, todos os nodos são alcançáveis a partir de qualquer nodo. Um grafo desconexo não possui caminhos entre todos os nodos, ele possui subconjuntos de nodos que se comunicam (subgrafos conexos), porém não tem comunicação com os outros subconjuntos. Os subgrafos conexos, que formam um grafo desconexo, são denominados componentes. Podem existir nodos e conexões que são críticos para a existência da conectividade em um grafo. Estes são denominados pontos de corte (*cutpoints*) e pontes (*bridges*), respectivamente.

Geodésico. O caminho de menor comprimento entre um par de nodos é denominado geodésico. Caso existam múltiplos caminhos com o mesmo comprimento entre o mesmo par, todos eles serão referenciados como geodésicos.

Distância geodésica. A distância geodésica, denotada por $d(i, j)$, é o comprimento do geodésico entre um par de nodos n_i e n_j qualquer. Se não existir um caminho entre os nodos n_i e n_j a distância geodésica entre eles é dita infinita, ou seja, $d(i, j) = \infty$. Em um grafo, $d(i, j) = d(j, i)$.

Distância geodésica média. Representa o número médio de nodos entre quaisquer pares de nodos no grafo. A equação 3.7 apresenta o cálculo da distância média, denotada por \bar{D} .

$$\bar{D} = \sum_{j>i}^N d(a_i, a_j) \quad (3.7)$$

Excentricidade. A excentricidade (Hage e Harary, 1995), denotada por $e(n_i)$, de um nodo n_i é o maior geodésico entre n_i e todos os demais $(g - 1)$ nodos, em outras palavras, ela mensura o quão longe um nodo esta do nodo mais distante dele no grafo. A equação 3.8 apresenta o cálculo da excentricidade de um nodo.

$$e(n_i) = \max d(a_i, a_j) \quad (3.8)$$

Diâmetro do grafo. O diâmetro de um grafo é a maior distância geodésica, $\max_{i,j} d(n_i, n_j)$, entre os nodos mais distantes entre si no grafo, isto é, a maior excentricidade dentre todos os nodos (Wasserman e Faust, 1994). Grafos desconexos tem diâmetro infinito. Grafos completos possuem diâmetro igual a 1.

Ponto de corte. Um nodo n_i é um ponto de corte se o número de componentes presentes no grafo que o contém é menor do que o número de componentes no grafo resultante da eliminação de n_i . O conceito de ponto de corte pode ser estendido de um único nodo para um conjunto de nodos necessários para manter o grafo conectado (Wasserman e Faust, 1994).

Pontes. Analogamente ao ponto de corte, uma ponte representa uma conexão necessária para manter o grafo conectado.

3.3 Medidas de proeminência

Um dos usos primários da teoria dos grafos na ARS é a identificação dos atores mais importantes da rede social (Wasserman e Faust, 1994). Diversas medidas de proeminência foram desenvolvidas com o intuito de determinar quais são os atores mais proeminentes (ou importantes) baseado na localização dos mesmos na rede. A identificação dos atores proeminentes depende da medida utilizada. A determinação de qual medida se aplica a um determinado problema é de natureza subjetiva, pois depende do problema em questão e da forma como ele foi modelado, por exemplo, uma rede representada por um dígrafo engloba um conjunto de medidas possíveis, maior do que uma rede representada por um grafo não direcionado.

Existem duas grandes classes de medidas de proeminência. A primeira delas é a centralidade, ela se aplica tanto a dígrafos quanto a grafos e tem como principal representante a centralidade de grau (análoga ao grau do nodo). A segunda classe diz respeito às medidas de prestígio, as quais só são aplicáveis para dígrafos, uma vez que levam em consideração a direção das conexões para extrair medidas como o grau de chegada (*indegree*) e de saída (*outdegree*) de um ator.

Uma forma alternativa de classificar as medidas de proeminência é analisar as hipóteses utilizadas em suas definições. Autores como David Knoke e Ronald Burt (Knoke e Burt, 1983) partem do princípio de que um ator é proeminente caso ele se faça visível para vários atores, isto é, mantenha conexões diretas (fortes) com muitos atores. Já Charles Hubbel (Hubbel, 1965) e Noah Friedkin (Friedkin, 1991) ressaltam que além das conexões diretas é importante avaliar também as conexões indiretas que um ator mantém para identificá-lo como proeminente. A *centralidade de grau* e o *prestígio* são exemplos de medidas que levam em consideração somente as conexões fortes do ator e a *centralidade de proximidade* e a de *intermediação* são exemplos de medidas que consideram tanto as conexões diretas quanto as indiretas.

É possível estender as medidas de proeminência de forma a obter índices relativos a grupos de atores. Tais índices auxiliam na comparação entre diferentes redes.

As medidas detalhadas nas seções seguintes são referentes somente à classe de proeminência de centralidade. A discussão é realizada no contexto de grafos não direcionados de única relação dicotômica. Para considerações sobre *prestígio*, centralidade no contexto de dígrafos, grafos com múltiplas relações e relações ponderadas (não dicotômicas) o leitor é

direcionado para (Wasserman e Faust, 1994). Na Figura 3.9 são apresentados três grafos que servem como referência para a introdução das medidas de proeminência de centralidade (seção 3.2.1).

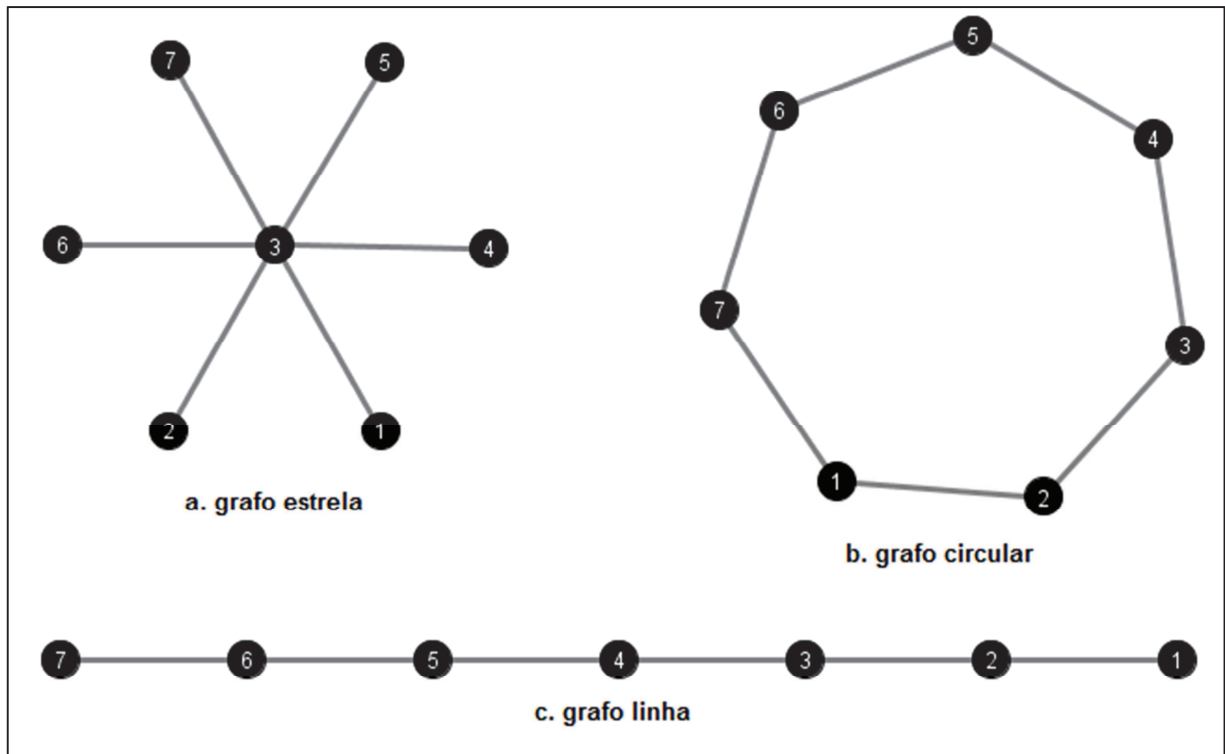


Figura 3.9: Grafo estrela, circular e linha.

3.3.1 Centralidade de grau

A centralidade de grau (*degree centrality*) trata de evidenciar os atores com maior atividade na rede, isto é, atores que possuem conexões diretas com uma grande quantidade de outros atores. A medida de centralidade de grau de um ator, denotada por $C_D(a_i)$, é análoga ao grau de um nodo, $d(n_i)$, detalhada na seção 3.1.2. É interessante normalizar tal medida para que esta deixe de estar sujeita às dimensões da rede. Uma padronização da centralidade de grau, $C'_D(a_i)$, proposta por (Wasserman e Faust, 1994) é apresentada na equação 3.9.

$$C'_D(a_i) = \frac{d(n_i)}{g - 1} \quad (3.9)$$

$C'_D(a_i)$ mensura a proporção de atores adjacentes a a_i e conseqüentemente não depende de g . A partir da versão padronizada torna-se possível a comparação entre atores membros de redes cujos tamanhos diferem. Atores com valor de $C'_D(a_i)$ próximo de 1 conectam-se diretamente com vários atores e são identificados como proeminentes, em contrapartida atores com o valor de $C'_D(a_i)$ próximo de 0 possuem poucas conexões e são identificados como pouco relevantes sob o aspecto da centralidade de grau.

Na Figura 3.9, o grafo estrela apresenta o ator 3 com a centralidade de grau máxima, $C'_D(3) = 1$, pois este conecta-se com todos os outros atores, enquanto que os demais atores possuem centralidade de grau baixa, pois todos conectam-se apenas ao ator 3. Já o grafo circular (Figura 3.9) demonstra uma situação onde todos os atores possuem a mesma centralidade de grau (grafo 2-regular) e conseqüentemente não apresenta um ator proeminente.

3.3.2 Centralidade de proximidade

A centralidade de proximidade (*closeness centrality*) considera não somente as conexões diretas, mas também as indiretas de um ator para classificá-lo como proeminente. A ideia é a de que um ator é central caso ele possa interagir rapidamente com todos os demais (Wasserman e Faust, 1994). Interagir rapidamente significa “percorrendo pequenas distâncias”. Outra forma de interpretar a centralidade de proximidade é partindo do princípio de que a centralidade de um ator é inversamente proporcional à distância existente entre ele e cada um dos demais atores.

Sabidussi (1996) propôs que a centralidade de proximidade do ator seja calculada em função da soma dos geodésicos entre o ator e todos os demais. A equação 3.10 demonstra o cálculo da centralidade de proximidade como o inverso da soma de todos os geodésicos.

$$C_C(a_i) = \left[\sum_{j=1}^g d(a_i, a_j) \right]^{-1} \quad (3.10)$$

O valor máximo da centralidade de proximidade, $C_C(a_i) = (g - 1)^{-1}$, ocorre quando o ator possui conexões diretas com todos os demais atores, por exemplo, este é o caso do ator 3 do grafo estrela da Figura 3.9. O valor mínimo da centralidade de proximidade, $C_C(a_i) = 0$, ocorre

quando um ou mais atores não são alcançáveis, por exemplo, na ocorrência de um ator isolado ou ainda em um grafo desconexo. Como ressaltado em (Wasserman e Faust, 1994), a centralidade de proximidade só é relevante em grafos conexos, pois caso contrário ela assumirá valor 0 para todos os atores.

Assim como na centralidade de grau foram feitas sugestões para padronizar a centralidade de proximidade de forma a obter o valor em uma escala independente da magnitude do grafo. Tal como mencionado em (Wasserman e Faust, 1994), Beauchamp (1965) propôs a padronização apresentada na equação 3.11 para tal padronização.

$$C'_c(a_i) = (g - 1) \cdot C_c(a_i) = \frac{(g - 1)}{[\sum_{j=1}^g d(a_i, a_j)]} \quad (3.11)$$

3.3.3 Centralidade de intermediação

Um ator com valores de centralidade de grau e de proximidade baixos pode mesmo assim ser de suma importância para os demais atores da rede. A proeminência deste ator é determinada pela sua posição estratégica com relação aos caminhos existentes entre os pares de atores. A sua posição o coloca na situação de intermediador das interações entre díades não adjacentes.

A ideia fundamental é a de que um ator é central caso ele esteja entre os geodésicos de vários outros atores (Wasserman e Faust, 1994). Diversos autores comentaram o uso de uma medida de centralidade que levasse em conta estas características, porém como mencionado em (Wasserman e Faust, 1994), foi Anthonisse (1971) e Freeman (1977) que primeiramente quantificaram a centralidade de intermediação (*betweenness centrality*). Para determinar a centralidade de intermediação de um ator a_i qualquer é preciso primeiro verificar em quantos geodésicos este ator está presente. Assume-se g_{jk} como a quantidade de geodésicos entre dois atores a_k e a_j e as ocorrências em que a_i está presente como $g_{jk}(a_i)$. É assumido que todos os geodésicos entre a_j e a_k possuem a mesma probabilidade de serem selecionados, ou seja, $1/g_{jk}$. A centralidade de intermediação, denotada por $C_B(a_i)$, é apresentada na equação 3.12.

$$C_B(a_i) = \sum_{j < k} g_{jk}(a_i) / g_{jk} \quad (3.12)$$

O valor máximo assumido por $C_B(a_i)$ é o total de díades excluindo o ator a_i , ou seja, $(g - 1) \cdot (g - 2) / 2$. O valor mínimo é 0 e indica que a_i não faz parte de nenhum geodésico. É possível padronizar $C_B(a_i)$ utilizando a equação 3.13.

$$C'_B(a_i) = \frac{C_B(a_i)}{[(g - 1) \cdot (g - 2) / 2]} \quad (3.13)$$

A centralidade de intermediação pode ser analisada ao nível da rede como um todo, a partir da centralidade de intermediação média (C_B). Como ressaltado por Wasserman e Faust (1993), a C_B permite comparar redes diferentes no que diz respeito à heterogeneidade da intermediação entre os membros destas. Na equação (3.14) é apresentada a centralidade de intermediação média, como definida em (Freeman 1979).

$$C_B = \frac{2 \sum_{i=1}^g [C_B(n^*) - C_B(n_i)]}{[(g - 1)^2 (g - 2)]} \quad (3.14)$$

Diferentemente da proximidade, a intermediação pode ser calculada mesmo para grafos desconexos. Vale ressaltar também que a noção de intermediação pode ser associada ao conceito previamente discutido de ponto de corte, contudo um ator não precisa necessariamente ser um ponto de corte para possuir um valor alto de centralidade de intermediação.

3.3.4 Coeficiente de agrupamento

O coeficiente de agrupamento local, denotado por C_i , mensura o quão próximo os atores adjacentes a um ator a_i estão de formar um *clique* (componente fortemente conexo). Esta medida foi introduzida por Duncan J. Watts e Steven Strogatz (1998) com a finalidade de determinar se uma dada rede possuía topologia de mundo pequeno. A topologia de redes de mundo pequeno é detalhada na seção 3.3.2. O cálculo de C_i é apresentado na equação 3.15.

$$C_i(a_i) = 2E_i/k_i(k_i - 1) \quad (3.15)$$

O valor de E_i representa a quantidade de conexões entre os vizinhos de a_i e k_i denota a quantidade de atores adjacentes a a_i . Caso os atores adjacentes a a_i componham um *clique*, o número de conexões entre eles será igual a $k_i(k_i - 1)/2$ e, conseqüentemente, $C_i = 1$.

Uma medida útil para a ARS é o coeficiente de agrupamento médio, denotado por \bar{C} , o qual assume o valor médio do coeficiente de agrupamento local para cada ator relativo a toda a rede. A equação 3.16 demonstra o cálculo para o coeficiente de agrupamento médio.

$$\bar{C} = \sum_{i=0}^g C_i(a_i) / g \quad (3.16)$$

3.4 Topologias

Nesta seção são apresentadas as topologias de redes sociais. O primeiro modelo compreende as redes aleatórias (Erdős e Rényi, 1960). A construção de tais redes é baseada na hipótese de que a existência ou não de uma conexão entre um par qualquer de atores é dada por uma probabilidade p . A sua criação data de 1958 e é obra do trabalho dos matemáticos húngaros Paul Erdős e Alfréd Rényi.

O segundo modelo, denominado redes de mundo pequeno (ou modelo Watts-Strogatz), incorpora características das redes aleatórias e das regulares⁹. Isto significa que, esta topologia possui um alto coeficiente de agrupamento (como as regulares) e uma distância média baixa (como as aleatórias). Duncan Watts e Steven Strogatz enunciaram o modelo de redes de mundo pequeno (Watts e Strogatz, 1998) baseados nos estudos de “mundo pequeno” (*small world*) conduzidos por Stanley Milgram em (Milgram, 1967).

O último modelo apresentado, denominado redes livres de escala (ou modelo Barabási-Albert), visa modelar as redes presentes no mundo real de forma mais precisa do que as redes aleatórias e as de mundo pequeno. Neste modelo, o objetivo principal é representar a dinâmica

⁹ O termo rede regular, neste contexto, é análogo ao termo grafo regular (definido na seção 3.2.2), onde todos os nodos possuem o mesmo grau.

das redes reais, para isso, os autores focam em modelar os elementos de construção (*assembly*) e de evolução (*growth*) da rede.

3.4.1 Redes aleatórias

Uma rede aleatória inicia com n atores e um parâmetro p_c . Tal que p_c é a probabilidade de haver uma conexão entre qualquer par de atores. O resultado é uma rede com aproximadamente $p_c * N(N - 2)/2$ conexões, com um baixo coeficiente de agrupamento e uma distância média baixa. A probabilidade p_c ¹⁰ é aplicável a todos os atores da rede, isto significa que todos têm chances iguais de se conectarem com quaisquer outros atores. Contudo, como ressaltado em (Réka e Barabasi, 2002), vários pesquisadores questionaram o modo com que as conexões são distribuídas em uma rede aleatória. Estes questionamentos foram embasados em modelos de redes existentes no mundo real, como por exemplo, a *World Wide Web* (WWW). A distribuição de conexões entre os atores que formam a WWW não segue uma distribuição de probabilidade global, isto é, nem todos os atores (sites) possuem chances iguais de se conectarem uns com os outros.

De qualquer modo, redes aleatórias são amplamente utilizadas em estudos empíricos devido a características que a um modelo obtido a partir de uma distribuição de probabilidade provê. Por exemplo, uma vez que uma rede aleatória é gerada seguindo uma distribuição de probabilidade, muitas das suas medidas de proeminência podem ser calculadas a partir de argumentos probabilísticos (Réka e Barabasi, 2002).

Na equação (3.17) é apresentado o cálculo do coeficiente de agrupamento médio para uma rede aleatória. Vale ressaltar que este cálculo depende somente do grau médio (seção 3.2.2., equação (3.2) e (3.3)) da rede e do número total de atores (g), já o cálculo genérico para o coeficiente de agrupamento médio (equação (3.16)) depende do cálculo dos coeficientes de agrupamento de cada ator.

$$C_{rand} = \frac{\bar{d}}{g} \quad (3.17)$$

¹⁰ Neste trabalho a probabilidade p de conexão nas redes aleatórias é representada por p_c (probabilidade de conexão) de modo a não confundi-la com as probabilidades envolvidas nos demais modelos.

Outra propriedade interessante das redes aleatórias é que características emergem de forma abrupta para determinados valores de p_c . Os autores em (Réka e Barabási, 2002) ressaltam que o objetivo principal do estudo de grafos aleatórios é determinar em qual probabilidade p_c uma determinada propriedade Q se tornará presente.

3.4.2 Redes de mundo pequeno

Este modelo tem suas origens em sistemas sociais reais onde a maioria das pessoas mantém conexões com as pessoas de seu convívio diário (colegas de trabalho, vizinhos, irmãos, etc.). Contudo estes indivíduos também se conectam a pessoas distantes (geograficamente e/ou socialmente) deles, como por exemplo, amigos que residem em outros países ou parentes que enriqueceram. As redes de mundo pequeno foram inicialmente estudadas por Stanley Milgram (1967).

Em (Watts e Strogatz, 1998) é proposto um algoritmo que recebe um único parâmetro que determina a probabilidade p_r de uma conexão c entre atores a_i e a_j quaisquer ser refeita, fazendo com que o ator a_i passe a se conectar a um outro ator a_k . Desta forma, uma conexão local ($a_i \Leftrightarrow a_j$) possivelmente torna-se uma conexão de longa distância ($a_i \Leftrightarrow a_k$), diminuindo assim a distância média da rede. Este algoritmo, como exposto em (Réka e Barabási, 2002), é apresentado a seguir.

(1) Inicie com ordem. Comece com um reticulado de forma anel com N nodos em que cada nodo é conectado com os seus K primeiros vizinhos ($K/2$ de cada lado). Para que a rede seja esparsa, porém conexa o tempo todo, considere $N \gg K \gg \ln(N) \gg 1$.

(2) *Randomize*. Reconecte aleatoriamente cada conexão do reticulado com probabilidade p_r tal que auto-conexões e conexões duplicadas sejam ignoradas.

A variação $p_c \in [0, 1]$ faz com que a rede varie entre um reticulado, $p_r = 0$, e uma rede aleatória, $p_r = 1$. Deste modo, redes de mundo pequeno, em que $0 < p_r < 1$, oscilam entre as redes regulares e as aleatórias.

3.4.3 Redes livres de escala

Segundo (Barabási e Réka, 1999), a emergência de uma rede livre de escala depende de dois elementos fundamentais que juntos são capazes de modelar a dinâmica de redes do mundo real. O primeiro deles diz respeito à evolução da rede. Nas topologias descritas previamente assume-se que o processo inicia com um número fixo de atores (redes aleatórias) ou com um número fixo de atores e algumas conexões (redes de mundo pequeno). Contudo, o processo de formação de uma rede no mundo real normalmente diverge destes, iniciando com um número pequeno de atores e expandindo ao longo do tempo. O segundo elemento se refere à forma com que as conexões são criadas. Nos outros modelos é assumida uma probabilidade (p_c ou p_r) que se aplica igualmente a todos os atores. Porém, em uma rede real o que é observado é que normalmente atores com mais conexões tendem a receber ainda mais conexões, ao passo que atores com menos conexões tendem a receber menos. Esta propriedade é identificada com Anexação Preferencial (*Preferential Attachment*). A fórmula da Anexação Preferencial pode ser observada na equação (3.18).

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j} \quad (3.18)$$

A probabilidade de um ator receber uma nova conexão, depende da quantidade de conexões que ele já possui. Em contrapartida, a probabilidade de conexão nas redes aleatórias e de mundo pequeno é independente do grau do ator. Mais precisamente, a distribuição de grau dos atores é dada segundo uma lei de potência (equação (3.19)), com γ variando entre 1 e 3 (Réka e Barabási, 2002). Vale ressaltar que o expoente de dimensionamento (γ) é independente da quantidade de atores na rede.

$$P(k) \sim k^\gamma \quad (3.19)$$

Em (Réka e Barabási, 2002) é apresentado o seguinte algoritmo para construção de redes livres de escala:

(1) Crescimento (*Growth*). Iniciando com um valor pequeno (m_0) de atores, a cada unidade de tempo, adiciona-se um novo ator com a_i com c conexões que o conectam a m diferentes atores já presentes na rede.

(2) Anexação Preferencial. Ao escolher os atores com os quais o novo ator a_i irá se conectar, é assumido que a probabilidade Π que a_i se conecte com um ator a_j qualquer depende da centralidade de grau do ator a_j (equação (3.9)).

É relevante mencionar que após t unidades de tempo este procedimento resulta em uma rede com $t + m_0$ atores e $m * t$ conexões.

3.5 Considerações finais

Este capítulo apresentou a Análise de Redes Sociais, incluindo seus conceitos fundamentais e teóricos (sob a perspectiva da Teoria dos Grafos). Também foram apresentadas medidas de proeminência e as três topologias de redes sociais (aleatórias, mundo pequeno e livre de escala). O método de conjunto de classificadores proposto neste trabalho dispõe os classificadores em uma estrutura em rede e utiliza vários dos princípios apresentados neste capítulo. O capítulo seguinte descreve o algoritmo SAE que implementa o método proposto.

Capítulo 4

Metodologia

No capítulo 2 foram apresentadas as dificuldades inerentes a Classificação *Online*. Dentre elas, destaca-se a mudança de conceito, reconhecida por vários pesquisadores como o maior desafio associado à Classificação *Online* (Bifet *et al*, 2011). Este trabalho apresenta o algoritmo baseado em conjunto de classificadores *Social Adaptive Ensemble* (SAE), o qual é aplicável para *streams* de dados com a presença (ou não) de mudanças de conceito.

Neste capítulo é apresentado em detalhes o algoritmo SAE. A seção 4.1 apresenta os conceitos gerais do SAE. Na seção 4.2 é apresentada a estratégia de combinação das predições individuais. Na seção 4.3 é explicado quando e como novos classificadores são adicionados. A seção 4.4 compreende a estratégia de remoção de especialistas com baixa taxa de acerto ou redundantes. A forma como as conexões são atualizadas é detalhada na seção 4.5. A última seção, 4.6, compreende as considerações finais do capítulo.

4.1 Algoritmo SAE

Como ressaltado na seção 2.4, um método baseado em conjunto de classificadores deve ser composto por classificadores diversos e possuir um método de combinação que ressalte classificações corretas e anule as incorretas. Além disso, no problema de Classificação *Online*, deve ser levada em consideração alguma estratégia para recuperação do conjunto frente a mudanças de conceito.

No SAE a diversidade dos especialistas (classificadores) é garantida por meio do treinamento de especialistas nas instâncias incorretamente classificadas pelo conjunto (rede) no

período anterior e também pelo treinamento individual dos especialistas em amostras diferentes destas instâncias (*Bagging online*). Contudo, como esta estratégia não garante que especialistas não venham a se tornar redundantes é importante especificar um método para identificação e remoção de especialistas redundantes (seção 4.4).

A combinação das decisões no SAE é feita em dois níveis. O primeiro nível reúne as decisões de especialistas similares¹¹ (subredes), enquanto que o segundo combina as decisões do primeiro na decisão da rede. Esta estratégia impede que conjuntos homogêneos de especialistas dominem a decisão da rede com base na sua cardinalidade. Além disso, caso os membros de uma subrede possuam decisões divergentes, isso é refletido no peso desta perante a decisão da rede. A ideia por trás disso é que se a decisão da subrede for dividida, isso indica que esta é menos relevante do que a decisão unânime de outra subrede.

Como o SAE utiliza uma abstração de rede social para a estruturação do conjunto é relevante especificar as características pertencentes ao ator (especialista), conexões, subredes e rede. Cada especialista (e) tem associado a ele um classificador (m_e), a quantidade de acertos de classificação no período atual (a_e) e o momento em que ele foi adicionado (t_e). Cada conexão (k) compreende um especialista de origem (o_k), um de destino (d_k), a quantidade de ações iguais entre o_k e d_k no período atual (s_k) e se ela está ativa¹² ou não (v_k). As subredes (R_S) são compostas por um conjunto de especialistas similares (E_S). A rede compreende o conjunto de todos os especialistas (E), de todas as conexões (C) e a quantidade de acertos da rede no período atual (a_R).

A execução do algoritmo é dividida em períodos de comprimento c , tal que c indica a quantidade de instâncias de treinamento consideradas no período. Cada período (p), exceto o primeiro, depende daquele imediatamente anterior no que diz respeito à estrutura da rede. Desta forma o que é observado no período p_n , influencia as decisões realizadas no período p_{n+1} . A quantidade de períodos para uma execução do algoritmo é indeterminada, uma vez que esta depende do comprimento, possivelmente infinito, da *stream* de dados. No diagrama da Figura 4.1 é apresentada a estrutura de um período.

¹¹ O conceito de similaridade entre especialistas é explicado na seção 4.5.

¹² Uma conexão é dita ativa quando os dois especialistas ligados por ela são identificados como similares.

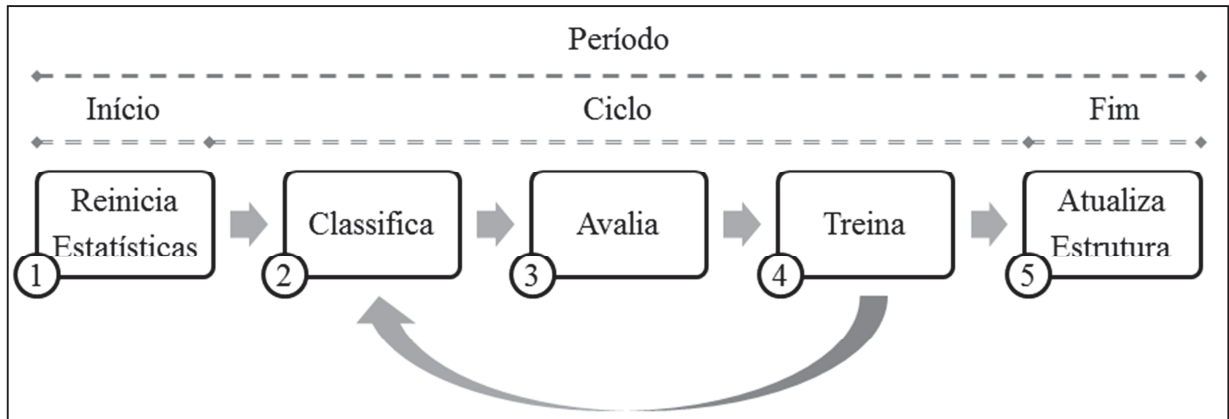


Figura 4.1: Diagrama de um período.

Na primeira etapa são reiniciadas as estatísticas relativas às classificações corretas da rede (a_R), de cada especialista e (a_e) e também aquelas relativas à similaridade entre pares de especialistas (s_k). As etapas dois a quatro são repetidas até que tenha sido alcançada a quantidade máxima de instâncias para o período (c). A combinação das decisões das subredes é feita na etapa dois para prever a classe da instância sendo processada. Na etapa três é verificado se a classe predita corresponde à classe esperada, se sim, é incrementado a_k , caso contrário, a instância é armazenada no conjunto de instâncias incorretamente classificadas (I_{er}). A etapa quatro compreende o treinamento individual de todos os especialistas (E) da rede (R) de forma similar ao método de *Bagging online* (Oza e Russel, 2001). Por fim, na etapa cinco é atualizada a estrutura da rede, sendo possível a inclusão de um especialista e /ou a remoção de vários especialistas, além da ativação e desativação de conexões com base na similaridade entre especialistas observada no último período. Caso um novo especialista seja adicionado ele será treinado no conjunto de instâncias incorretamente classificadas do período anterior (I_{er}). Na Figura 4.2 é apresentado um diagrama que representa as alterações na rede ao final de cada período.

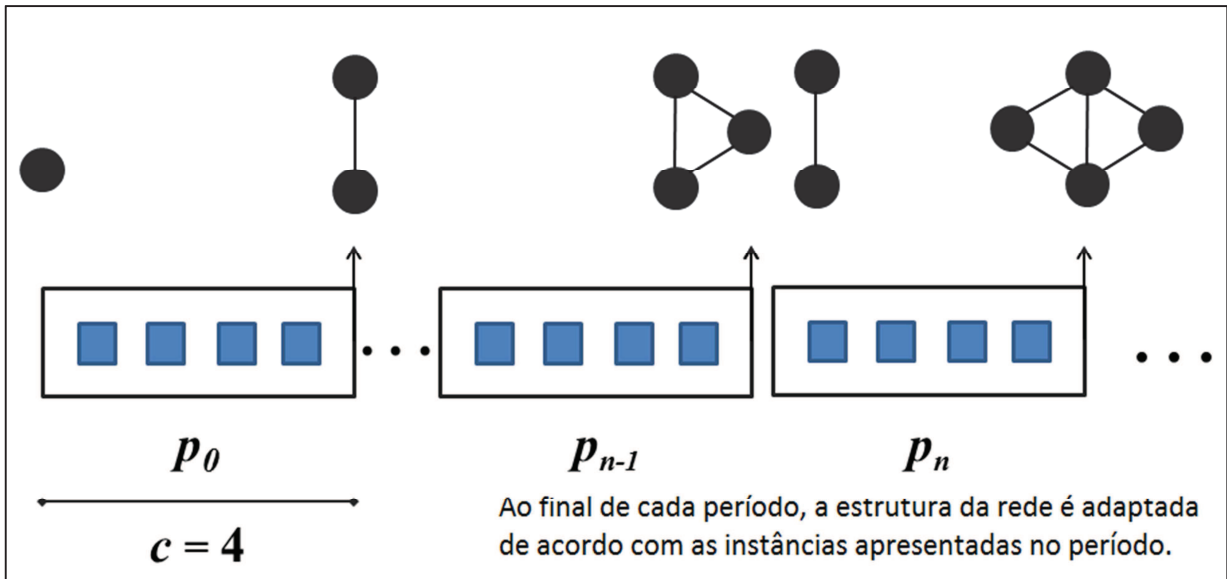


Figura 4.2: Alterações na estrutura da rede ao final do período.

É importante ressaltar que ao processar uma fonte de dados possivelmente infinita, alguns cuidados devem ser tomados. O primeiro é que o algoritmo não pode depender da existência de uma variável por toda sua execução, já que o valor desta tende ao infinito e não será possível armazená-lo. O segundo é referente ao armazenamento temporário de dados, que deve ser feito com cautela para não ultrapassar os limites da memória disponível. Para viabilizar a implementação do algoritmo assume-se que o comprimento dos períodos (c) é finito, e conseqüentemente, torna-se factível armazenar os valores das estatísticas dentro do período, contudo estas devem ser reiniciadas a cada novo período. O armazenamento temporário das instâncias incorretamente classificadas (I_{er}) é limitado ao tamanho do período (c).

Na Figura 4.3 é apresentado o pseudocódigo do treinamento para o algoritmo SAE. Ao início do algoritmo é adicionado o primeiro especialista da rede (linha 1, Figura 4.3), outros especialistas são adicionados, conforme necessário, pela subrotina *Adiciona.especialista* (linha 15, Figura 4.3). A subrotina *Extração.medidas* (linha 17, Figura 4.3) é responsável por extrair medidas de proeminência utilizadas para a análise do algoritmo como: Centralidade de grau média, Centralidade de agrupamento médio, Centralidade de intermediação média, dentre outras. No capítulo 5, especificamente na seção 5.4, são apresentadas e discutidas as inferências que podem ser feitas a partir das medidas de proeminência. O apêndice A são apresentadas os valores para as medidas de proeminência extraídas durante os experimentos realizados.

SAE(S)	
S	<i>Stream</i> de dados que disponibiliza uma instância i a cada t instantes
R	Rede de especialistas
e	Um único especialista
$d_i^{(R)}$	Classe predita pela Rede R para a instância i
i	Instância formada por um vetor de atributos (\vec{x}_i) e uma classe (y_i)
a_R	Quantidade de classificações corretas feitas pela Rede no período
I_{er}	Instâncias incorretamente classificadas no período
c	Tamanho do período, ou seja, quantidade de instâncias processadas antes de uma nova atualização da Rede
E	Conjunto que contém todos os especialistas
1.	$R = R \cup \{e\}$
2.	Enquanto <i>existe.próxima(S)</i>
3.	<i>Reinicia.estatísticas(R)</i> //Etapa 1
4.	Para $j = 0$ até c //Ciclo do período
5.	$i = próxima(S)$
6.	$d_i^{(R)} = Combina.decisões(R, i)$ //Etapa 2
7.	Se $d_i^{(R)} = y_i$ //Etapa 3
8.	$a_R = a_R + 1$
9.	Senão
10.	$I_{er} = I_{er} \cup \{i\}$
11.	Para $e \in E$ //Etapa 4
12.	<i>Treina(e, i)</i>
13.	<i>Remove.especialistas(R)</i>
14.	<i>Atualiza.conexões(R)</i>
15.	<i>Adiciona.especialista(R, I_{er}, a_R)</i> //Etapa 5
16.	<i>Gera.subredes(R)</i>
17.	<i>Extração.medidas(R)</i>

Figura 4.3: Pseudocódigo - Treinamento do algoritmo SAE

A subrotina *Gera.subredes* (linha 16, Figura 4.3) encontra os componentes fracamente conectados para construir as subredes do próximo período. O treinamento individual dos especialistas (linha 12, Figura 4.3) presentes na rede é feito por meio do método *Online Bagging*

(seção 2.4.2). O treinamento é aplicado de forma diferente quando apenas um classificador compõe a rede, neste caso todas as instâncias de treinamento são repassadas para ele.

4.2 Combinação das decisões

A segunda etapa do algoritmo corresponde à combinação das decisões individuais dos especialistas na decisão da rede. Como mencionado na seção 2.4, o objetivo combinação das decisões de um conjunto de classificadores é ressaltar classificações corretas e cancelar as incorretas. Assumindo que todos os especialistas são diversos a votação majoritária simples incorreria em bons resultados. Porém, apesar das duas abordagens para induzir um conjunto diverso, que incluem o treinamento de novos classificadores em instâncias incorretamente classificadas no período anterior (seção 4.3) e a amostragem dos dados de treinamento (*Online Bagging*), é possível que as instâncias corretamente classificadas por alguns especialistas sejam, aproximadamente, as mesmas que as classificadas corretamente por outros. O problema inerente à existência de uma grande quantidade de especialistas capazes de classificar apenas as mesmas instâncias é que estes ofuscam a decisão de outros especialistas dotados da capacidade de classificar corretamente outras instâncias devido a sua cardinalidade. Na Figura 4.4 é apresentado um exemplo de problema de decisão binário (\bullet e $+$) não linear e na Figura 4.5 são delimitadas as instâncias corretamente classificadas por 5 especialistas, tal que 3 deles, são similares em suas decisões, conjunto A , ao passo que os 2 restantes, conjunto B , divergem dos membros de A e concordam entre si.

Considerando que uma instância i a ser classificada se encontra no domínio dos membros do conjunto B , e que fosse utilizada a votação majoritária, a decisão dos membros de A ofuscaria a decisão dos membros de B , simplesmente porque eles estão em maior número. Isto é um problema, pois os membros de B são mais aptos a classificar instâncias nesta região. Vale lembrar que as instâncias presentes na superfície coberta pelos membros de B pode ser resultado de uma mudança de conceito.

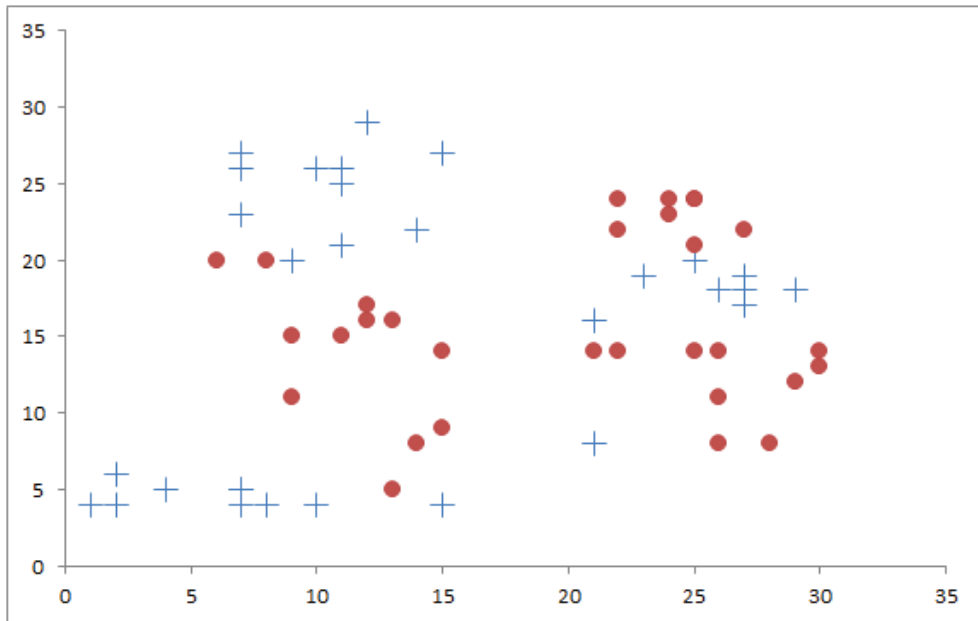


Figura 4.4: Problema de classificação não linear

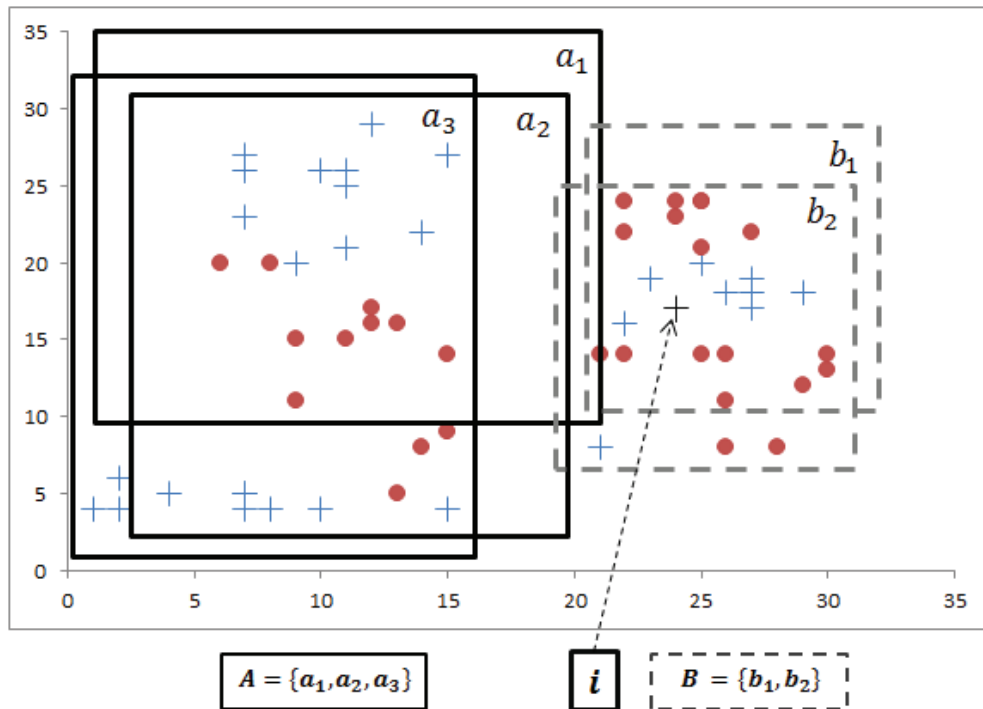


Figura 4.5: Superfícies de separação.

Outro exemplo seria uma situação em que os especialistas existentes na rede são diversos e a sua combinação, por meio da votação majoritária, resulta em uma melhoria da taxa de acerto individual. Na Tabela 4.1 é apresentado um problema de classificação binário contendo 10 instâncias e 3 especialistas. Cada especialista é capaz de classificar corretamente 60% das instâncias. A combinação por meio da votação majoritária destes 3 especialistas resulta em uma taxa de acerto de 90% (combinação ótima¹³).

Combinação (ótima) de 3 classificadores					
#instância	classe real	e_1	e_2	e_3	Rede
1	X	X	O	X	X
2	X	X	O	X	X
3	X	X	O	X	X
4	X	X	X	O	X
5	X	X	X	O	X
6	O	O	O	X	O
7	O	X	X	X	X
8	O	X	O	O	O
9	O	X	O	O	O
10	O	X	O	O	O

Tabela 4.1: Combinação ótima de 3 classificadores.

Combinação de 4 classificadores						
#instância	classe real	e_1	e_2	e_3	e_4	Rede
1	X	X	O	X	X	X
2	X	X	O	X	X	X
3	X	X	O	X	X	X
4	X	X	X	O	X	X
5	X	X	X	O	O	?
6	O	O	O	X	X	?
7	O	X	X	X	X	X
8	O	X	O	O	O	O
9	O	X	O	O	O	O
10	O	X	O	O	O	O

Tabela 4.2: Combinação de 4 classificadores.

Intuitivamente a adição de um novo especialista melhoraria a taxa de acerto ou não iria alterá-la. Contudo, em alguns casos, a adição de um especialista pode ressaltar as decisões

¹³ Existem 28 combinações dos resultados para 3 classificadores e 10 instâncias para um problema de classificação binário. O exemplo da Tabela 4.1 representa uma das configurações que resulta na combinação ótima. Na seção 2.4 são apresentadas as outras 27 combinações.

incorretas de outro especialista presente na rede. Na Tabela 4.2 é apresentado o resultado da inclusão do especialista e_4 , o qual classifica de forma muito similar ao especialista e_3 .

A partir da inclusão do especialista e_4 duas decisões corretas da rede (instâncias 5 e 6) tornaram-se empates e dada a quebra aleatória dos empates a rede pode não obter a taxa de acerto de 90%. A realização de uma votação em dois níveis faz com que o peso do voto individual dos especialistas e_3 e e_4 não seja tão influente na decisão da rede quanto o voto individual dos especialistas e_1 e e_2 . Na Tabela 4.3 é apresentado o resultado da votação da rede quando os votos dos especialistas e_3 e e_4 são combinados.

Combinação com 4 classificadores – agrupando e_3 e e_4							
#instância	classe real	e_1	e_2	e_3	e_4	e_3-e_4	Rede
1	X	X	O	X	X	X	X
2	X	X	O	X	X	X	X
3	X	X	O	X	X	X	X
4	X	X	X	O	X	?	X
5	X	X	X	O	O	O	X
6	O	O	O	X	X	X	X
7	O	X	X	X	X	X	X
8	O	X	O	O	O	O	O
9	O	X	O	O	O	O	O
10	O	X	O	O	O	O	O

Tabela 4.3: Combinação de 4 classificadores agrupando decisões.

É possível alterar o exemplo da Tabela 4.3 para que e_4 classifique incorretamente uma instância classificada corretamente por e_3 gerando um empate na votação e_3-e_4 , e consequentemente, induzindo um erro de classificação na rede. Por exemplo, caso e_4 classificasse a instância 3 como O , o voto e_3-e_4 resultaria em um empate e a votação da rede também (e_1 e e_2 votam X e O para instância 3). Contudo, o conjunto recebe um novo classificador somente quando a rede não está sendo capaz de classificar novas instâncias satisfatoriamente (seção 4.3), possivelmente em decorrência de uma mudança de conceito. O modelo de combinação foi enunciado com a intenção de reforçar a diversidade na combinação para que não haja um domínio da votação por parte de especialistas similares e não é garantido que sempre irá incorrer na solução ótima.

Na Figura 4.6 é apresentado o pseudocódigo da combinação das decisões. O primeiro passo da combinação é obter a classe predita para cada uma das subredes (linha 4 até 10, Figura

4.6). A classe predita por cada uma das subredes é ponderada (linha 9, Figura 4.6) de acordo com a ocorrência desta na subrede, por exemplo, se uma subrede s é formada por 5 especialistas e 3 deles predizem a classe X para a instância i , então $p_i^{(s)} = 3/5 \cong 0.66$. Na ocorrência de um empate, a decisão da subrede não é levada em consideração. Após identificar a classe predita para todas as subredes é feita a combinação destas para obtenção da predição da rede. A classe predita pela rede é a que obtiver o maior valor do somatório das ponderações (linha 11, Figura 4.6). Uma questão importante com relação à combinação das decisões é a ocorrência de empates. Por exemplo, caso existam apenas 2 especialistas e suas classificações sejam divergentes (e.g. o primeiro prediz X e o segundo O) haverá um empate. Na implementação do SAE os empates são quebrados de forma aleatória. A similaridade entre especialistas é atualizada logo após a combinação (linha 12 até 15, Figura 4.6).

Combina. decisões(R, i)

- $d_i^{(e)}$ Classe predita pelo especialista e para a instância i
 $D_i^{(e)}$ Conjunto contendo as predições dos especialistas para a instância i
 $d_i^{(s)}$ Classe predita pelas subredes
 $D_i^{(s)}$ Conjunto contendo as predições das subredes para a instância i
 $d_i^{(R)}$ Classe predita pela rede R para a instância i
 $D_i^{(R)}$ Conjunto formado por tuplas (d, p) representando a predição para a instância i obtida por cada uma das subredes da rede R , tal que d denota a classe predita e p a ponderação desta
 $p_i^{(s)}$ Ponderação da classe predita pela subrede s para instância i , tal que $0.5 < p_i^{(s)} \leq 1$
1. $D_i^{(R)} = D_i^{(e)} = \emptyset$
 2. Para $R_S \subset R$
 3. $D_i^{(s)} = \emptyset$
 4. Para $e \in R_S$
 5. $d_i^{(e)} = \text{classifica}(e, i)$
 6. $D_i^{(e)} = D_i^{(e)} \cup \{d_i^{(e)}\}$
 7. $D_i^{(s)} = D_i^{(s)} \cup \{d_i^{(e)}\}$
 8. $d_i^{(s)} = \max D_i^{(s)}$
 9. $p_i^{(s)} = |\{d, d = d_i^{(s)} \wedge d \in D_i^{(s)}\}| / |D_i^{(s)}|$
 10. $D_i^{(R)} = D_i^{(R)} \cup \{(d_i^{(s)}, p_i^{(s)})\}$
 11. $d_i^{(R)} = \arg \max_{d \in D_i^{(R)}} \sum_{(d,p)} p$
 12. Para $j = 0$ até $|D_i^{(e)}|$ //Laço para atualizar similaridades
 13. Para $w = 0 \wedge w \neq j$ até $|D_e|$
 14. Se $D_i^{(e)}[j] = D_e[w]$
 15. $\text{incrementa.similaridade}(j, w)$
 16. Retorna $d_i^{(R)}$

Figura 4.6: Pseudocódigo – Combinação de decisões

4.3 Adição de Especialistas

A adição de especialistas ocorre sempre que a taxa de acerto da rede (Tx_R) estiver abaixo de um limiar mínimo (Tx_{minR}). De forma a reforçar a diversidade da rede, os novos especialistas são treinados a partir dos erros (I_{er}) do último período. É importante ressaltar que é criado no máximo um especialista por período, isto é feito para evitar a adição de vários especialistas que sendo incluídos no mesmo momento e treinados nas mesmas instâncias (I_{er}) se tornariam redundantes. Uma consequência desejável desta estratégia de inclusão é que se caso esteja ocorrendo uma mudança de conceito, treinar nos erros recentes irá acelerar o processo de recuperação da rede, pois o novo especialista será adaptado aos erros dos antigos, ou seja, será adaptado ao novo conceito.

O pseudocódigo para a adição de especialistas é apresentado na Figura 4.7. Após ser adicionado um novo especialista, é adicionada uma conexão inativa entre ele e todos os demais especialistas (linha 7, Figura 4.7).

<i>Adiciona.especialista(R, I_{er}, Tx_R, a_R)</i>	
Tx_R	Taxa de acerto da Rede no período
I_{er}	Instâncias classificadas incorretamente no período
a_R	Quantidade total de acertos da Rede no período
Tx_{minR}	Parâmetro que delimita a taxa de acerto mínima da Rede
c	Comprimento do período
C	Conjunto contendo todas as conexões entre especialistas
E	Conjunto contendo todos os especialistas da Rede
1.	$Tx_R = a_R/c$
2.	Se $Tx_R < Tx_{minR}$
3.	$e_n = novo.especialista()$
4.	$treina(e_n, I_{er})$
5.	$E = E \cup \{e_n\}$
6.	Para $e \in R$
7.	$C = C \cup \{(e, e_n, 0, desativada)\}$

Figura 4.7: Pseudocódigo – Adição de especialista

4.4 Remoção de Especialistas

A remoção de especialistas, assim como a adição, ocorre ao final de cada período, porém esta pode resultar na remoção de mais de um especialista. Para que especialistas se mantenham na rede eles precisam atender a dois requisitos.

O primeiro requisito é que o especialista deve possuir uma taxa de acerto mínima (Tx_{minE}). A justificativa para isso é que mesmo que um especialista agregue valor do ponto de vista da diversidade da rede, caso este tenha uma taxa de acerto muito baixa dificilmente ele contribuirá de forma adequada para a decisão final.

O segundo requisito é que o especialista não pode ser redundante. Na prática pode não ser possível obter um conjunto de especialistas com uma grande diversidade. Logo, existe a possibilidade de que todos os especialistas sejam considerados similares e a votação em dois níveis passa a ser uma votação majoritária simples. Neste caso, é importante remover aqueles que tomam as mesmas decisões ou que possuem uma similaridade muito alta, pois eles distorcem a votação da rede. Outra motivação, não menos importante, diz respeito ao desperdício de recursos de memória e processamento. É assumido que se dois especialistas sempre tomam as mesmas decisões, um deles é dispensável.

Na Figura 4.8 é apresentado o pseudocódigo da remoção de especialistas. Quando dois especialistas são reconhecidos como redundantes, é removido aquele que está há mais tempo na rede (linha 11, Figura 4.8). Esta decisão foi feita sob o pretexto de que na ocorrência de uma mudança de conceito, um especialista que teve menos instâncias apresentadas ao seu modelo irá se adaptar mais facilmente.

Remove.especialistas(R)	
Tx_e	Taxa de acerto do especialista e
Tx_{minE}	Parâmetro que delimita a taxa de acerto mínima do especialista
a_e	Quantidade de acertos do especialista e
Cs	Coefficiente de similaridade
Cs_{max}	Parâmetro que delimita o limiar máximo de similaridade entre dois especialistas para que estes não sejam considerados redundantes
1.	Para $e \in E$ //Remoção por taxa de acerto insatisfatória
2.	$Tx_e = a_e / c$
3.	Se $Tx_e < Tx_{minE}$
4.	$E = E - \{e\}$
5.	Para $k \in C$
6.	Se $e \in k$
7.	$C = C - \{k\}$
8.	Para $k \in C$ //Remoção dos redundantes
9.	$Cs = s_k / c$
10.	Se $Cs \geq Cs_{max}$
11.	$e_v = max_idade(o_k, d_k)$
12.	$E = E - \{e_v\}$
13.	Para $k_x \in C$
14.	Se $e_v \in k_x$
15.	$C = C - \{k_x\}$

Figura 4.8: Pseudocódigo – Remoção de especialistas.

4.5 Atualização das Conexões

As conexões são de suma importância para a determinação dos redundantes e para construção das subredes. Um componente fundamental para a determinação de uma conexão é identificar o quão semelhante é um par de especialistas. Para isto é especificado o Coeficiente de Similaridade (Cs), apresentado na equação (4.1). Na Equação (4.1) $S_k(i, j)$ representa a quantidade de ações similares entre os especialistas i e j para o último período, incluindo classificações corretas e incorretas. O Cs é normalizado segundo o comprimento do período (c).

$$Cs(i, j) = \frac{S_k(i, j)}{c} \quad (4.1)$$

A todo o momento na rede existem, aproximadamente, $n(n - 1)/2$ conexões. Porém, nem todas estas conexões podem estar ativas. A ativação de uma conexão depende do coeficiente de similaridade mínimo (Cs_{min}), que especifica quando uma conexão entre um par de especialistas deixa de ser coincidência e passa a ser uma relação. Portanto, as conexões servem para dois propósitos. O primeiro é agrupar especialistas similares e o segundo é identificar redundantes. Desta forma a combinação das decisões dos especialistas é obtida pela decisão das subredes, a qual independe da quantidade de especialistas e, também, os especialistas redundantes podem ser identificados e removidos.

Na Figura 4.9 é apresentado o pseudocódigo da atualização das conexões.

<i>Atualiza.conexões(R)</i>	
Cs	Coeficiente de similaridade
Cs_{min}	Parâmetro que delimita a similaridade mínima entre especialistas para a ativação da conexão
1.	Para $k \in C$
2.	$Cs = s_k/c$
3.	Se $Cs < Cs_{min}$
4.	$a_c = desativada$
5.	Senão
6.	$a_c = ativada$

Figura 4.9: Pseudocódigo – Atualização das conexões

4.6 Considerações finais

Neste capítulo foi apresentado o método de conjunto de classificadores instanciado no algoritmo SAE. No SAE, a combinação dos votos é feita em dois níveis (subredes e rede). A influência do voto de um especialista na rede é dada pela quantidade de especialistas similares a ele existentes na rede. Quanto maior a quantidade de especialistas similares a ele, menor será seu peso na decisão da rede. Na ocasião de uma mudança de conceito essa forma de combinação, aliada a estratégia de inclusão, ressalta o voto dos especialistas melhor adaptados ao novo conceito. Quando não estiver ocorrendo uma mudança de conceito, caso um novo especialista

seja adicionado, ele simplesmente será incorporado por uma subrede devido a sua similaridade com os membros desta, fazendo com que o voto dele não seja, incorretamente, ressaltado.

Idealmente, as estratégias de inclusão, remoção e combinação são adequadas, porém elas dependem dos parâmetros estipulados para o algoritmo. A definição dos valores para os parâmetros do algoritmo não é uma tarefa trivial, pois os valores ótimos dos parâmetros variam dependendo da *stream* de dados. Nos experimentos (capítulo 5) são analisadas diferentes combinações de valores para os parâmetros com a finalidade de determinar o impacto destas sobre a taxa de acerto e o tempo de processamento.

Capítulo 5

Experimentos

Neste capítulo são apresentados e discutidos os resultados obtidos a partir da aplicação do SAE para diversos geradores de dados *stream* com e sem mudança de conceito. Todos os experimentos foram feitos utilizando o framework MOA (Bifet *et al.*, 2012). O MOA possui implementados diversos geradores de dados, métodos de avaliação e classificadores para dados *stream*. A implementação dos algoritmos SAE e DWM¹⁴ foram feitas no framework MOA para realização dos experimentos. Maiores detalhes sobre o uso e extensão (adição de classificadores) do MOA são apresentados no apêndice B.

Este capítulo se encontra dividido de forma que a primeira seção (5.1) compreende uma explicação detalhada do protocolo experimental utilizado nos experimentos. A segunda seção (5.2) apresenta a avaliação dos experimentos variando parâmetros de configuração do SAE e análise das redes obtidas. A terceira seção (5.3) apresenta a comparação do SAE frente a outros algoritmos baseados em conjunto de classificadores. A quarta e última seção (5.4) inclui as considerações finais sobre os experimentos.

5.1 Protocolo experimental

As bases de dados usadas nos experimentos são oriundas de geradores de dados. Os geradores utilizados neste trabalho são: *Random Tree Generator* (Domingos e Hulten, 2000),

¹⁴ O MOA inclui diversos classificadores, porém o algoritmo DWM não faz parte deste conjunto, e teve de ser implementado para a realização dos experimentos.

SEA (Street e Kim, 2001) e AGRAWAL (Agrawal *et al*, 1993). As implementações utilizadas destes geradores são aquelas disponíveis no MOA (Bifet *et al*, 2012).

As dimensões a serem consideradas na avaliação de um método de Classificação *Online*, são: Taxa de acerto, Tempo e Memória. Para mensurar a taxa de acerto, neste trabalho, optou-se por utilizar o método de avaliação *Prequential* (Gama, Sebastião e Rodrigues, 2009) apresentado na seção 2.5.2. A comparação do tempo de execução entre algoritmos serve como um indicador de performance, porém esta não é capaz de expressar de forma inequívoca a real complexidade de um algoritmo. Para mensurar de forma mais precisa possível o tempo de processamento, no MOA é contabilizado somente o tempo em que o processo se encontra em execução (*CPU time*).

Quantificar o uso de memória além de complexo, afeta diretamente a medida de tempo de execução, pois exige processamento extra para cada instância de treinamento ou teste. Quando métodos baseados em conjuntos de classificadores são utilizados é possível, em determinados casos, simplificar a quantificação do uso de memória a partir da quantidade de classificadores mantidos pelo algoritmo. Para as situações em que o classificador base é estável com relação ao uso de memória, como o algoritmo *Naïve Bayes*, esta métrica pode ser considerada com certa confiança. Contudo, em métodos baseados em árvores de decisão ou, de modo mais acentuado, no aprendizado baseado em instâncias, esta medida pode levar a conclusões errôneas sobre a quantidade real de memória utilizada. Devido à impossibilidade de mensurar fielmente a memória, o protocolo experimental proposto limita-se a ilustrar a quantidade de classificadores existentes, não com a intenção de representar o consumo de memória, mas para permitir à análise sobre a estratégia de inclusão e remoção de classificadores para os algoritmos SAE e DWM.

Para permitir uma comparação quantitativa do SAE frente a outros algoritmos classificadores foram selecionados três representantes de conjuntos de classificadores diferentes. O primeiro é o algoritmo DWM, representando um conjunto de classificadores em que a quantidade de classificadores existentes varia ao longo do tempo. O segundo algoritmo, ASHT *Bagging*, representa a classe de conjunto de classificadores em que a generalização depende de classificadores com parâmetros diferentes. No caso do ASHT *Bagging*, o parâmetro é a altura das árvores de decisão (*Adaptive Size Hoeffding Trees*). O terceiro algoritmo, ADWIN *Bagging*, representa um método híbrido de *ensemble* e *detector*, unindo o algoritmo *Bagging* na sua versão online (Oza e Rusell, 2001) ao algoritmo de detecção de mudanças ADWIN (Bifet e Gavaldà,

2007). Os algoritmos DWM, *ADWIN Bagging* e *ASHT Bagging* são descritos nas seções 2.4.3, 2.4.4 e 2.4.5, respectivamente. Os parâmetros para os algoritmos seguem os valores padrão definidos nas publicações originais, sendo eles:

- DWM: $\beta = 0.5$, $\gamma = 0.01$.
- *ADWIN Bagging*: quantidade de classificadores = 10.
- *ASHT Bagging*: quantidade de classificadores = 10, altura da primeira árvore = 2.

Em (Maloof e Kolter, 2003) não é apresentado um valor padrão para o tamanho do período de atualização do DWM. Testes realizados com o DWM mostram que um valor muito baixo para o período de atualização faz com que o conjunto se torne instável devido às remoções e adições frequentes de classificadores. Por outro lado, caso o tamanho do período seja muito elevado o conjunto tende a receber poucas atualizações, geralmente convergindo para um conjunto de um único classificador. Nos experimentos apresentados na subseção 5.3, foram consideradas duas configurações diferentes do DWM, sendo uma com período igual a 5000 e outra com período 500. Períodos com valores de ordem elevada, como 50000, convergiam para um único classificador, enquanto que valores menores que 500 para conjuntos extremamente instáveis com uma taxa de acerto muito baixa.

Os experimentos englobam variações nos geradores RTG, SEA e AGRAWAL e nas configurações do método de avaliação *Prequential*. As configurações do método de avaliação *Prequential* são apresentadas na Tabela 5.1, enquanto que as características específicas de cada um dos geradores são apresentadas na Tabela 5.2. Com a intenção de prover resultados estatisticamente significativos todos os experimentos foram repetidos dez vezes variando a semente aleatória de cada gerador. Os resultados referentes à taxa de acerto e o tempo de execução são obtidos a partir da média dos resultados de cada uma das dez repetições.

Dados			Método de avaliação		
Gerador	#Instâncias	#Mudanças	Tipo da mudança	Frequência de amostragem	#Períodos
SEA AGRAWAL	1.000.000	1 2	Abrupta Gradual	100.000	10
RTG	1.000.000	Não se aplica	Não se aplica	100.000	10

Tabela 5.1: Configurações do método de avaliação

ID	Gerador	#Atributos nominais	#Atributos contínuos	#Mudanças	Tipo da mudança
RTS	RTG	5	5	0	Não se aplica
RTC	RTG	50	50	0	Não se aplica
SEA-1	SEA	0	3	1	Abrupta
SEA-2	SEA	0	3	2	Abrupta
SEA-3	SEA	0	3	1	Gradual
SEA-4	SEA	0	3	2	Gradual
AGRAWAL-1	AGRAWAL	3	6	1	Abrupta
AGRAWAL-2	AGRAWAL	3	6	2	Abrupta
AGRAWAL-3	AGRAWAL	3	6	1	Gradual
AGRAWAL-4	AGRAWAL	3	6	2	Gradual

Tabela 5.2: Configurações dos geradores de dados

Os geradores foram configurados de forma a abranger diferentes características observadas em *streams* de dados. As configurações RTS (*Random Tree Simple*) e RTC (*Random Tree Complex*) são oriundas do gerador *Random Tree Generator* (RTG) e representam dados *stream* em que o conceito se mantém estável. Estas duas configurações foram definidas em (Bifet *et al.*, 2012) para representar um problema simples de classificação online sem mudança de conceito (RTS) e outro mais complexo (RTC). As configurações para os geradores SEA e AGRAWAL variam de acordo com a quantidade e do tipo de mudança de conceito. Para os experimentos com 1 mudança de conceito, esta ocorre na instância 500.000 e para aqueles com 2 mudanças, estas ocorrem nas instâncias 333.333 e 666.666. Detalhes sobre como os geradores RTG, SEA e AGRAWAL operam são apresentados nas seções 2.6.1, 2.6.2 e 2.6.3, respectivamente.

Com a intenção de avaliar se a estratégia de evolução presente no SAE funciona com diferentes classificadores base foram selecionados dois classificadores diferentes para realização dos experimentos. Seguindo a ideia de que classificadores instáveis propiciam maior diversidade (seção 2.4) no conjunto, o primeiro classificador base selecionado para avaliação do SAE é um

representante dos métodos baseados em Árvores de Decisão. O algoritmo escolhido foi o *Hoeffding Tree*, devido a sua comprovada eficiência na configuração *online* (Bifet *et al*, 2011). O segundo classificador base é um representante dos métodos bayesianos, denominado *Naïve Bayes*. Este algoritmo não foi desenvolvido para operar diretamente na configuração *online* como o *Hoeffding Tree*, contudo seu treinamento pode ser adaptado para configuração *online* de forma direta (Bifet *et al*, 2011). As implementações dos algoritmos *Hoeffding Tree* e *Naïve Bayes* utilizadas são aquelas disponíveis no MOA (Bifet *et al*, 2012).

5.2 Experimentos com diferentes configurações do SAE

O objetivo desta seção é apresentar e discutir experimentos envolvendo o SAE com diferentes configurações de parâmetros. Os parâmetros do SAE são descritos na Tabela 5.3. Ao longo do restante deste capítulo são citadas 4 configurações diferentes para o algoritmo SAE, sendo elas SAE-1, SAE-2, SAE-3 e SAE-4. Os valores para os parâmetros destas configurações podem ser encontrados na Tabela 5.4.

Parâmetro	acrônimo	descrição
Taxa de acerto mínima da rede	Tx_{minR}	Taxa de acerto mínima que a rede deve obter para que não seja adicionado um novo especialista.
Taxa de acerto mínima do especialista	Tx_{minE}	Taxa de acerto mínima para que um especialista não seja removido.
Coeficiente de similaridade mínimo	Cs_{min}	Coeficiente de similaridade mínimo para que dois especialistas sejam identificados como similares.
Coeficiente de similaridade máximo	Cs_{max}	Coeficiente de similaridade máximo acima do qual dois especialistas passam a ser identificados como redundantes.
Tamanho do período	c	Quantidade de instâncias a serem consideradas antes de uma nova atualização da rede.

Tabela 5.3: Parâmetros para o SAE

identificador	$Tx_{\min R}$	$Tx_{\min E}$	Cs_{\min}	Cs_{\max}	c	classificador base
SAE-1	0.9	0.5	0.85	0.99	50000	Hoeffding Tree
SAE-2	0.9	0.5	0.85	0.9	50000	Hoeffding Tree
SAE-3	0.9	0.5	0.85	0.99	50000	Naïve Bayes
SAE-4	0.9	0.5	0.85	0.9	50000	Naïve Bayes

Tabela 5.4: Configurações do SAE utilizadas nos experimentos

A taxa de acerto, o tempo de execução e a topologia da rede dependem diretamente das estratégias de atualização do conjunto (capítulo 4) e dos valores dos parâmetros (Tabela 5.3). A taxa de acerto está intrinsecamente relacionada com a topologia da rede, pois a classificação é feita com base na disposição dos especialistas na rede (seção 4.1.1). O tempo de execução varia de acordo com o tamanho da rede, já que quanto maior a quantidade de especialistas, maior o tempo necessário para treiná-los e combinar suas predições.

O parâmetro Cs_{\min} determina quando as conexões entre especialistas se tornam ativas e serve para controlar a densidade da rede. Caso seja atribuído um valor baixo para Cs_{\min} , como 0.50, a rede se tornará fortemente conexa fazendo com que especialistas que possuem poucas predições similares sejam reconhecidos como similares. Nesta situação a votação em dois níveis se transforma em uma votação majoritária simples. O mesmo ocorre quando Cs_{\min} recebe um valor alto, como 1.0, pois apenas especialistas iguais terão suas conexões ativadas fazendo com que a rede seja totalmente desconexa. Para ilustrar esta situação nas Figuras 5.1 e 5.2 são apresentadas as redes obtidas após os primeiros quatro períodos para uma configuração com $Cs_{\min} = 0.01$ e $Cs_{\min} = 1.00$. A taxa de acerto em cada período é idêntica em ambas às configurações. A configuração do gerador utilizada foi a RTS.

Para *streams* de dados em que mudanças de conceito não ocorrem é observada uma convergência para uma rede fortemente conectada simplesmente porque os especialistas não são removidos e se tornam similares ao longo do tempo. Um indicativo de que não estão ocorrendo mudanças de conceito é a rede se tornar estável, ou seja, quando não são observadas muitas adições e remoções e a quantidade de especialistas não varia muito.

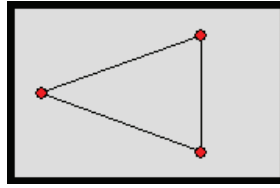


Figura 5.1: Rede após quatro períodos ($Cs_{min} = 0.01$) para o gerador RTS.



Figura 5.2: Redes após quatro períodos ($Cs_{min} = 1.00$) para o gerador RTS.

Existem situações em que a rede permanece em um constante estado de instabilidade. Uma destas situações é evidenciada quando o parâmetro Tx_{minR} nunca consegue ser ultrapassado pelo conjunto, por exemplo, se $Tx_{minR} = 0.95$ para um problema difícil de classificação (como o RTC), a cada período um novo especialista será inserido. Se $Tx_{minE} = 0.5$ nenhum especialista será removido, pois o conceito é estável e todos os especialistas existentes conseguem obter uma taxa de acerto acima de 0.5. Duas características indesejáveis são obtidas quando esta situação ocorre. A primeira é que o conjunto cresce linearmente ao longo do tempo degradando o tempo de processamento sem melhorias na taxa de acerto. A segunda característica diz respeito à taxa de acerto que é afetada devido ao conjunto não ser renovado, o que limita a diversidade entre os classificadores. O impacto desta situação pode ser reduzido fazendo com que o parâmetro Cs_{max} assumira um valor não muito alto, como $Cs_{max} = 0.9$. Desta forma, o conjunto é renovado frequentemente, pois os especialistas possuem uma maior chance de se tornarem redundantes e, conseqüentemente, de serem removidos. No gráfico da Figura 5.3 é apresentada uma comparação entre duas execuções do SAE para o experimento RTC, sendo uma com $Cs_{max} = 0.99$ (SAE-1) e a outra com $Cs_{max} = 0.90$ (SAE-2).

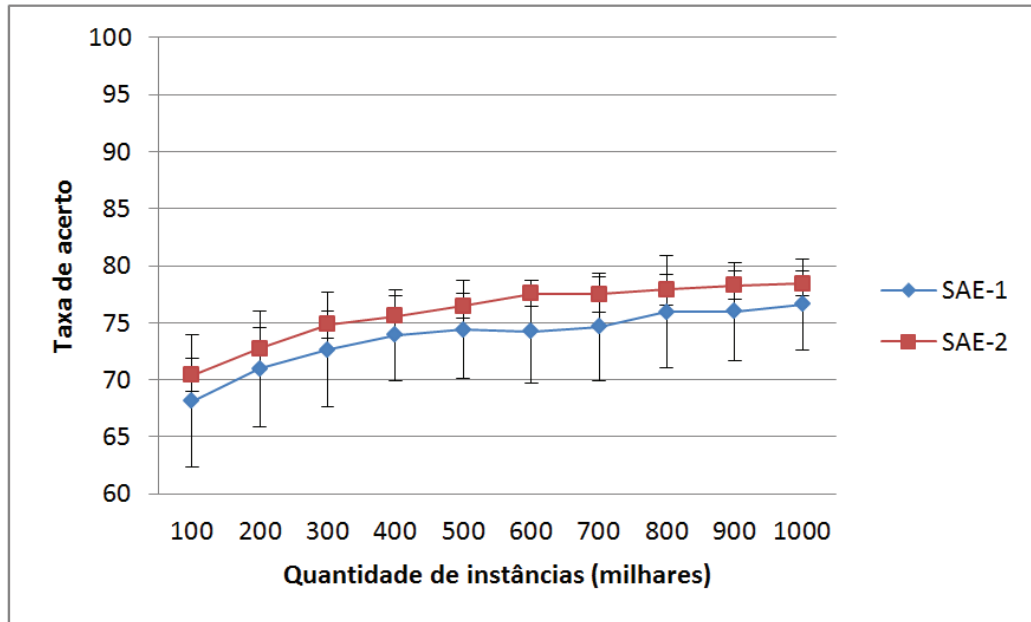


Figura 5.3: SAE (RTC).

Uma outra situação de instabilidade evidenciada nos experimentos diz respeito às alterações ocorridas na rede devido a mudanças de conceito. É importante ressaltar que em ocasiões de mudança de conceito abruptas os especialistas adaptados ao conceito antigo tendem a ser removidos da rede dando lugar a novos especialistas. Durante a mudança de conceito a taxa de acerto da rede tende a ser recuperada rapidamente devido ao novo especialista ter uma maior influência na votação do que os especialistas existentes (seção 4.1.1). Porém, no período que precede àquele em que houve a mudança de conceito é provável que o novo especialista seja incorporado a uma subrede. Nesse caso, se houverem muitos especialistas presentes nessa subrede o recém-adicionado terá sua votação subjugada pelos especialistas presentes na subrede. Para estas situações é obtida uma melhor taxa de acerto quando o tamanho da rede é pequeno e consequentemente a subrede também, pois o especialista recém-adicionado mesmo que associado a subrede não terá o peso de seu voto drasticamente ofuscado. Para manter uma quantidade baixa de especialistas é possível aumentar o $Tx_{\min E}$, porém caso o valor seja muito elevado e a *stream* de dados represente um problema de classificação complexo a rede pode ser formada por um único especialista, o qual é substituído por um novo a cada período. Outra forma de manter a quantidade de especialistas baixa na rede é diminuindo o valor do Cs_{\max} para que menos especialistas similares sejam mantidos na rede. O gráfico da Figura 5.4 apresenta um comparativo entre duas configurações do SAE

(SAE-1 e SAE-2) para o experimento SEA-2. No gráfico da Figura 5.4 as linhas ortogonais ao eixo X indicam as localizações das mudanças de conceito.

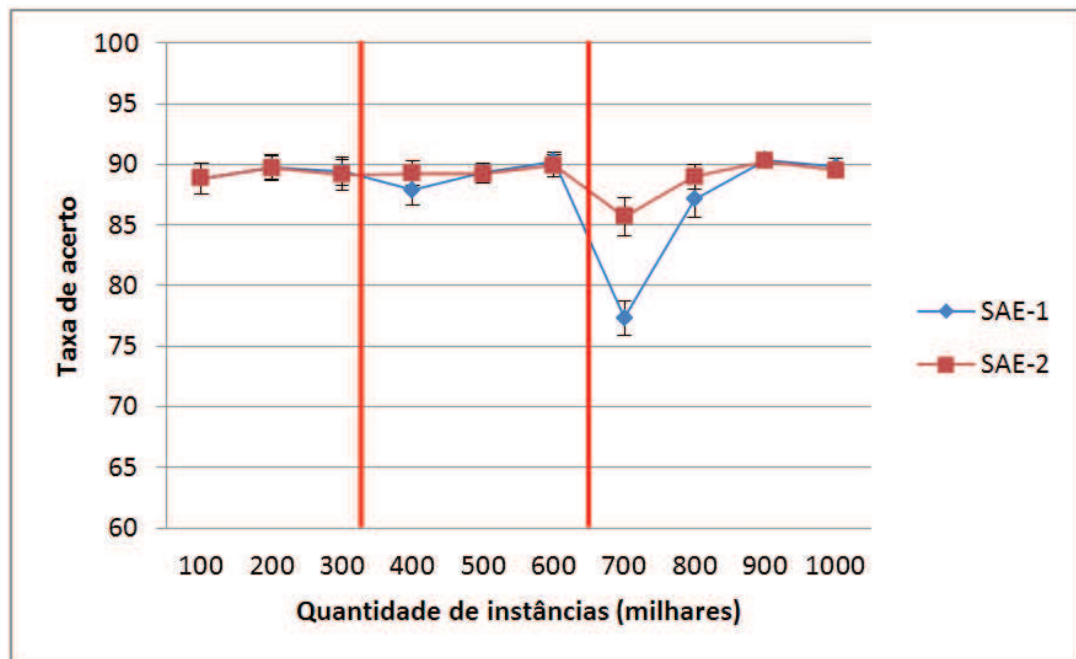


Figura 5.4: SAE (SEA-2).

Na ocorrência de uma mudança de conceito gradual, o algoritmo tem um comportamento similar ao apresentado durante as mudanças de conceito abruptas. A configuração SAE-1 não sofre um impacto tão grande durante uma mudança de conceito gradual, como pode ser observado no gráfico da Figura 5.5, o qual apresenta a comparação entre o SAE-1 e SAE-2 para o experimento AGRAWAL-3.

Ao menos duas questões importantes ao apresentar um método baseado em conjunto de classificadores devem ser respondidas, sendo elas:

- Seria um conjunto formado pelo classificador X melhor do que um único classificador X?
- A estratégia de adaptação do conjunto depende diretamente do classificador base empregado?

Para responder a ambas as perguntas são utilizados os gráficos presentes nas Figuras 5.6 e 5.7, que apresentam a taxa de acerto para o experimento SEA-2 ao ser empregado o SAE com o classificador base *Hoeffding Tree* (HF) e o SAE com o classificador base *Naïve Bayes* (NB), respectivamente.

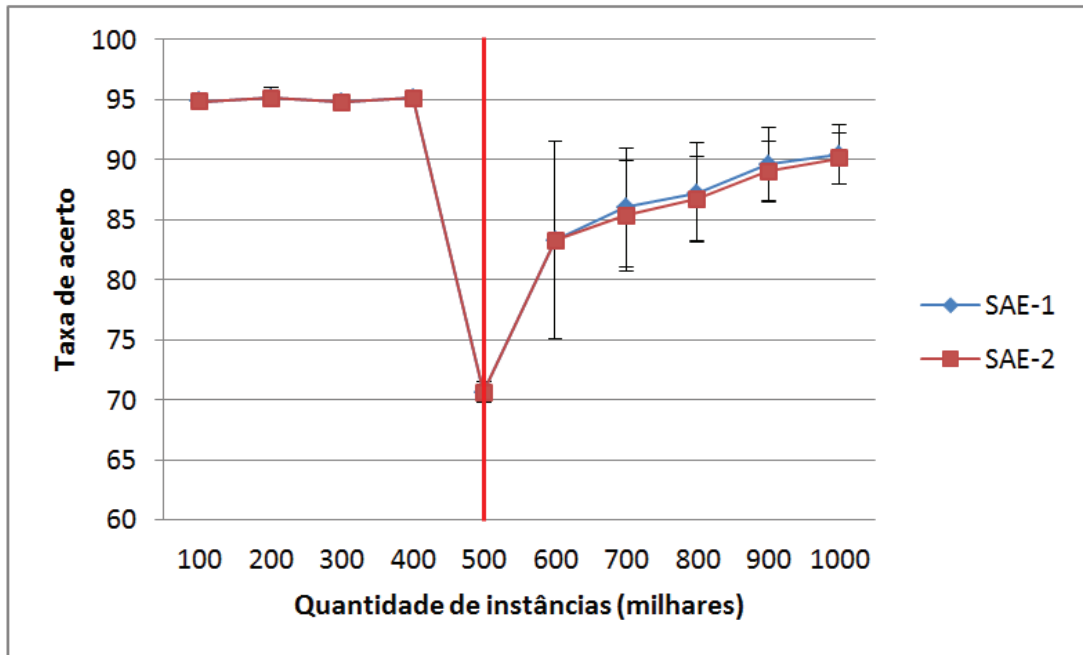


Figura 5.5: SAE (AGRAWAL-3).

No gráfico da Figura 5.6 é possível observar que o classificador *Hoeffding Tree* se adapta de forma mais lenta a mudança de conceito. No gráfico da Figura 5.7 o classificador *Naïve Bayes* não consegue se recuperar após a primeira mudança e reestabelecer a taxa de acerto obtida no início da *stream*. É possível notar que a taxa de acerto obtida pelo SAE-1 e SAE-2 são similares à taxa de acerto obtida pelo SAE-3 e SAE-4 para o experimento SEA-2. Contudo, isso não ocorre para todos os experimentos. Em especial naqueles onde não ocorre mudança de conceito a taxa de acerto do SAE-3 ou SAE-4 é muito inferior à obtida pelo SAE-1 ou SAE-2 (ver apêndice C).

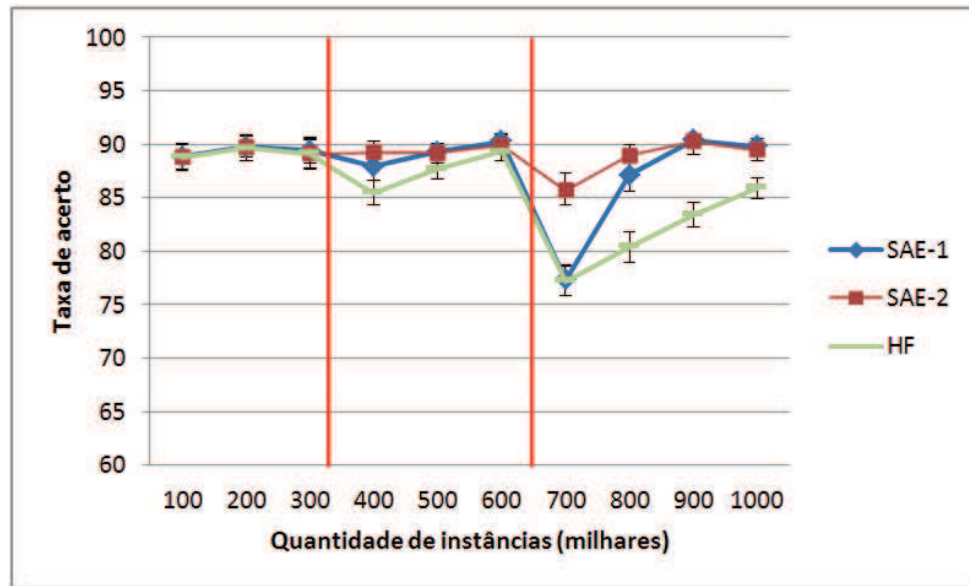


Figura 5.6: SAE e *Hoeffding Tree* (HF) (SEA-2).

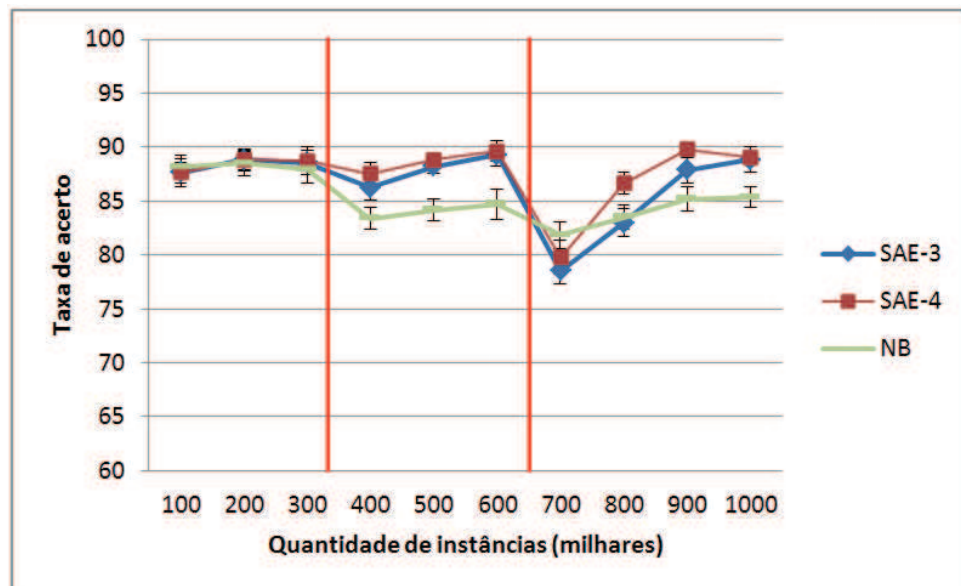


Figura 5.7: SAE e *Naïve Bayes* (NB) (SEA-2).

Levando em consideração a melhor performance obtida para todos os experimentos foram escolhidas as configurações SAE-1 e SAE-2 para comparação do SAE frente a outros algoritmos baseados em conjuntos de classificadores nas seções 5.3 e 5.4.

5.3 Comparativo entre o SAE e outros conjuntos de classificadores

O problema de Classificação *Online* possui três dimensões importantes: Taxa de acerto, Tempo de processamento e Memória. Idealmente, um classificador para a configuração *online* deve obter a melhor taxa de acerto possível, sem ultrapassar os limites de espaço (memória) e tempo (processamento) disponíveis, além de adaptar-se adequadamente a mudanças de conceito abruptas ou graduais. Os experimentos apresentados nesta seção visam comparar como o algoritmo SAE se apresenta em comparação com outros algoritmos baseados em conjunto de classificadores desenvolvidos para operar com dados *stream* com mudança de conceito. No apêndice C são apresentados os resultados de todos os experimentos, nesta seção são apresentados somente os resultados mais significativos para discussão.

Os gráficos das Figuras 5.8 e 5.9 apresentam a taxa de acerto dos algoritmos SAE (SAE-1 e SAE-2), ADWIN *Bagging* e ASHT *Bagging* para os experimentos RTS e RTC.

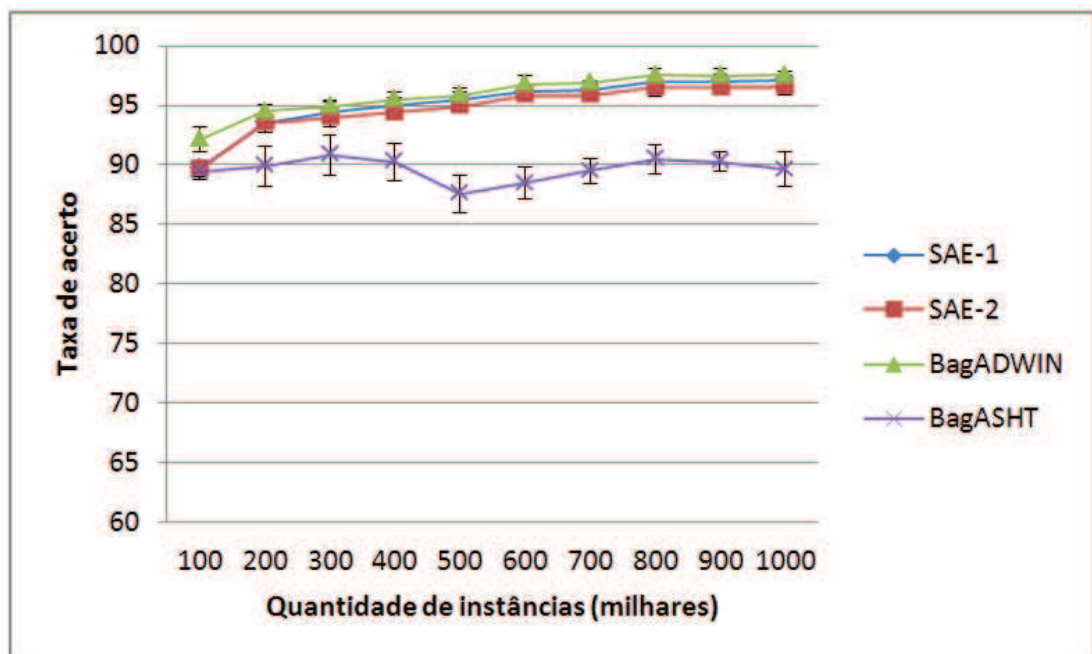


Figura 5.8: SAE, ADWIN e ASHT (RTS).

É visível a partir do gráfico da Figura 5.8 que o SAE e o ADWIN *Bagging* possuem uma taxa de acerto muito próxima. O tempo médio de processamento exigido pelo SAE-1 e SAE-2 é 35 e 17 segundos, respectivamente, enquanto que o ADWIN *Bagging* demanda 147 segundos. No

gráfico da Figura 5.9 o SAE-1 e SAE-2 possuem valores médios distantes dos resultados obtidos pelo *ADWIN Bagging*, além de demandar um tempo de processamento maior (889 e 860 segundos, SAE-1 e SAE-2 respectivamente, e 774 segundos *ADWIN Bagging*).

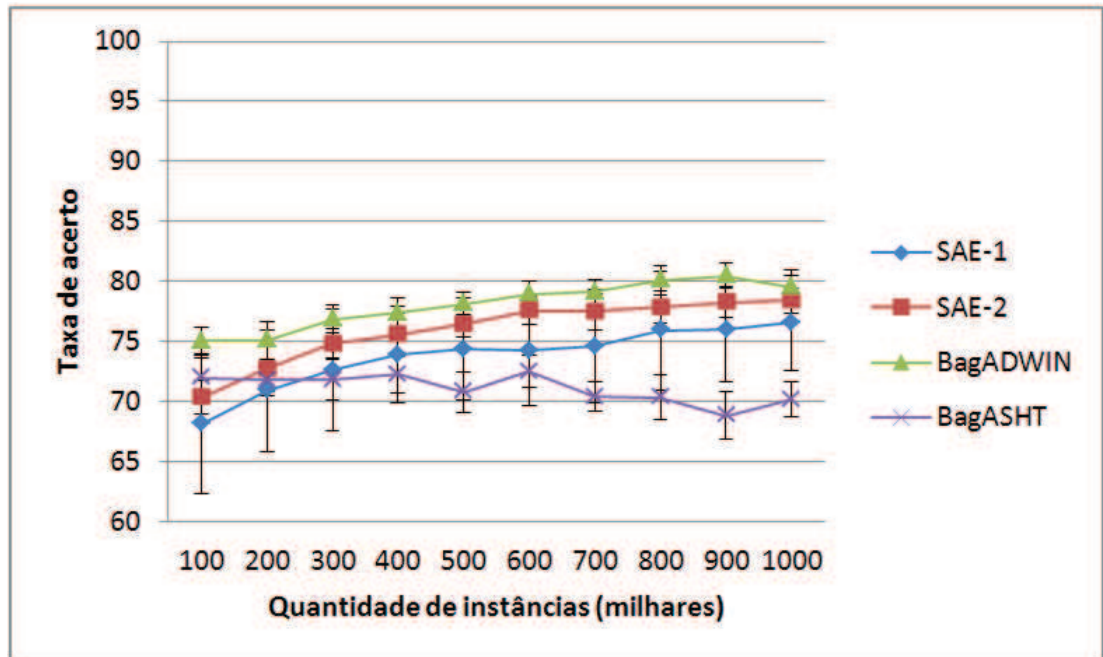


Figura 5.9: SAE, ADWIN e ASHT (RTC).

Nas Figuras 5.10 e 5.11 são apresentados os gráficos comparando o SAE-1 e SAE-2 ao *ADWIN Bagging* e ao *ASHT Bagging*, respectivamente, para o experimento AGRAWAL-1 (uma mudança de conceito abrupta). No gráfico da Figura 5.10 é importante observar a diferença significativa entre a taxa de acerto do *ADWIN Bagging* e o SAE-1 e SAE-2 após a mudança de conceito. Tanto o SAE-1, quanto o SAE-2 não se recuperam da mudança rapidamente. Neste experimento o tempo de processamento obtido pelo SAE (SAE-1 = 37s e SAE-2 = 37s) foi significativamente menor que o obtido pelo *ADWIN Bagging* (98s).

O gráfico da Figura 5.11 expõe uma degradação do *ASHT Bagging* após ter aparentemente começado a se recuperar da mudança de conceito ocorrida, ao passo que, o SAE-1 e SAE-2 iniciam a recuperação, mesmo que de forma lenta. A diferença entre o tempo de processamento do SAE (SAE-1 = 36s e SAE-2 = 37s) e do *ASHT Bagging* (67s) é menos significativa do que aquela observada entre o SAE e o *ADWIN Bagging*, contudo ainda é relevante.

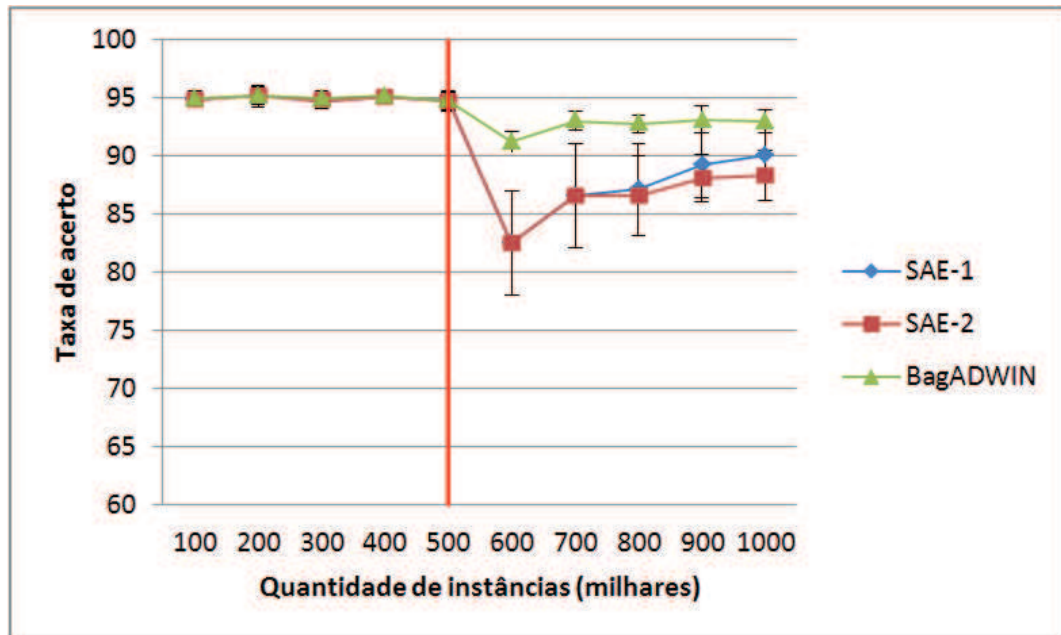


Figura 5.10: SAE e ADWIN (AGRAWAL-1).

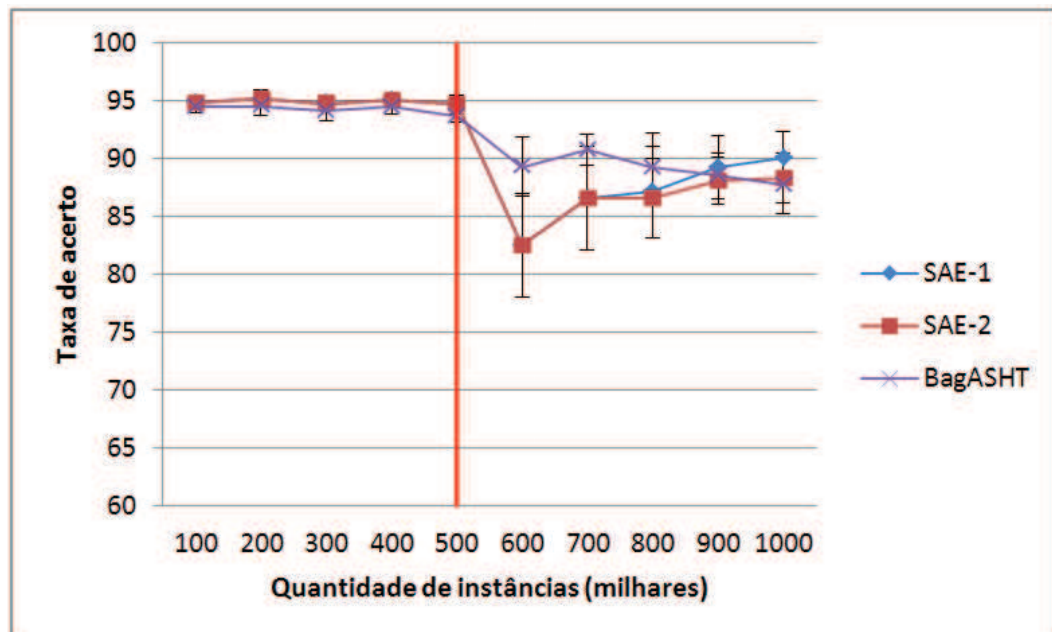


Figura 5.11: SAE e ASHT (AGRAWAL-1).

Na Figura 5.12 é apresentado o gráfico comparando o SAE-1 e SAE-2 ao ADWIN *Bagging* e ao ASHT *Bagging* para o experimento SEA-2 (duas mudanças de conceito abruptas). Neste experimento é válido ressaltar a pequena queda na taxa de acerto na ocorrência da segunda mudança de conceito para o SAE-2 e para o ASHT *Bagging*. Também é relevante mencionar o

tempo de processamento do SAE-2 (19s), o qual foi significativamente menor que todos demais (SAE-1 = 58s, ADWIN *Bagging* = 80s, ASHT *Bagging* = 34s).

A Figura 5.13 apresenta a comparação entre o SAE-1 e SAE-2 e o DWM-1 e DWM-2 para o experimento AGRAWAL-4 (duas mudanças de conceito graduais). Uma vez que tanto o SAE quanto o DWM mantêm conjuntos de classificadores dinâmicos é relevante analisar a quantidade de classificadores mantidos por ambos (Figura 5.14).

A partir da análise do gráfico da Figura 5.13 é possível perceber uma queda drástica, em decorrência da primeira mudança de conceito, na taxa de acerto do DWM-1 e, principalmente, para o DWM-2. O gráfico da Figura 5.14 (retângulos situados no eixo X indicam as zonas de mudança de conceito), apresenta uma característica desejável que é a remoção de classificadores redundantes da rede mantida pelo SAE-2 sem um grande impacto na taxa de acerto.

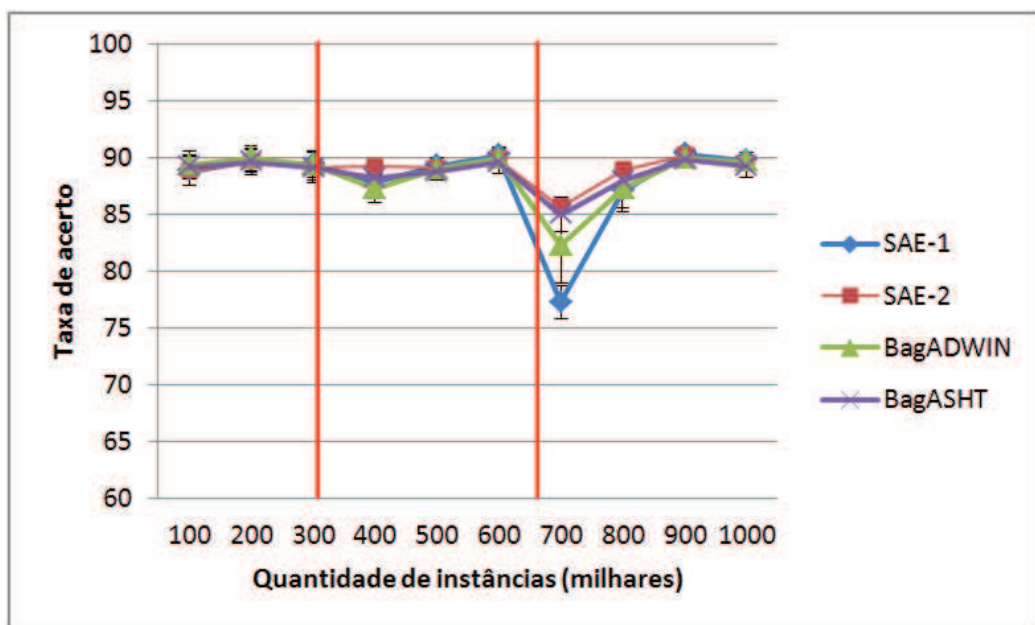


Figura 5.12: SAE, ADWIN e ASHT (SEA-2).

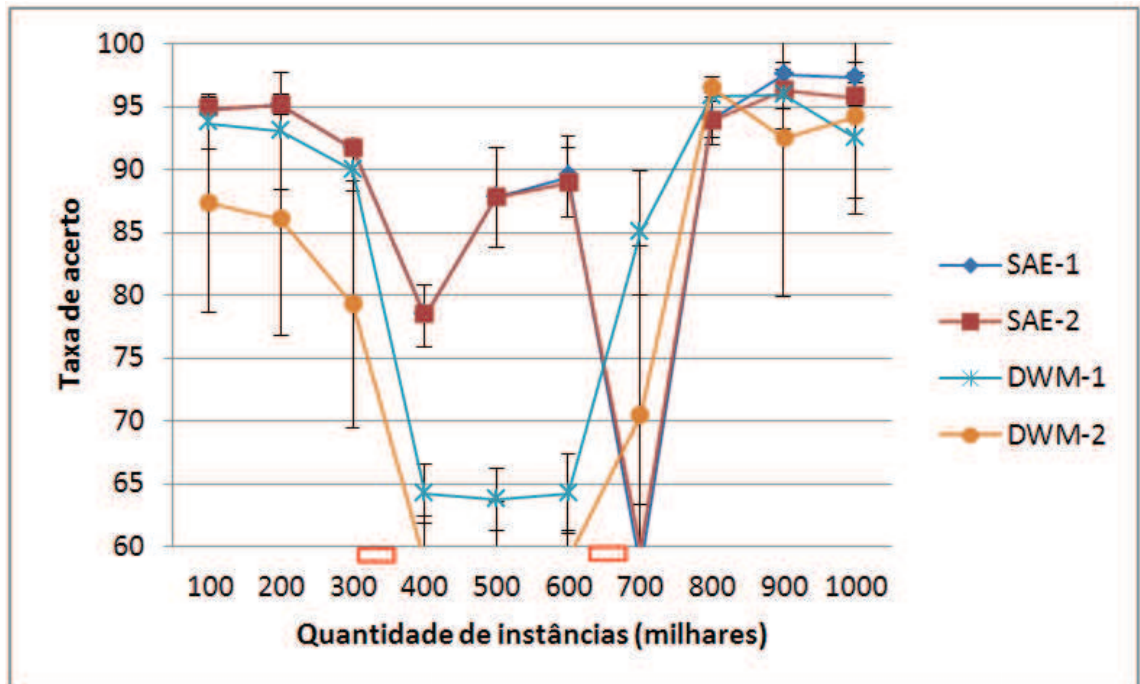


Figura 5.13: SAE e DWM (AGRAWAL-4).

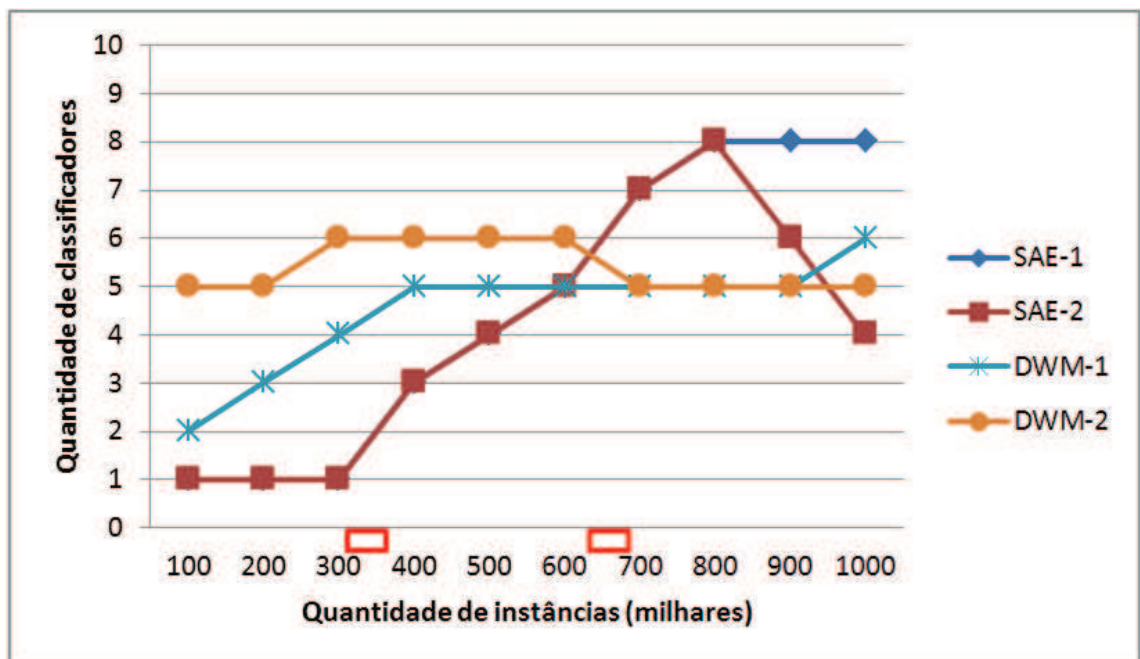


Figura 5.14: SAE e DWM (# Classificadores) (AGRAWAL-4).

5.4 Análise de Redes Sociais aplicada as redes obtidas pelo SAE

As medidas de proeminência consideradas para a análise são apresentadas na tabela 5.5. Estas medidas são obtidas ao final de cada período. Para a configuração experimental apresentada, em que todas as *streams* de dados contém 1 milhão de instâncias e o tamanho do período é sempre igual a 50 mil, isso significa que a cada 50 mil instâncias as medidas são extraídas novamente. Detalhes sobre as medidas de proeminência utilizadas são apresentados no capítulo 3. As tabelas contendo as medidas de proeminência para todos os experimentos estão inclusas no Apêndice A.

Medida	Descrição
\bar{d}	Centralidade de grau média. Sumariza o grau de todos os atores da rede.
S_D^2	Variância. Indica diferenças na quantidade de conexões para cada ator.
Δ	Densidade. Representa a proporção de conexões existentes relativa ao total de conexões possíveis.
\bar{C}	Coefficiente de agrupamento médio. Indica o quão coesas são as subredes que formam o grafo.
C_B	Centralidade de intermediação média. Sumariza a centralidade de intermediação de todos os atores da rede.

Tabela 5.5: Medidas de proeminência

Algumas medidas de proeminência, as quais dependem da distância entre atores, como a centralidade de proximidade, diâmetro, excentricidade, para citar algumas, não foram consideradas devido à característica da rede obtida pelo algoritmo ser desconexa a maior parte do tempo. As medidas de Densidade, Coeficiente de agrupamento médio e Centralidade de intermediação média variam no intervalo [0,1] e, portanto, são reportadas nos mesmos gráficos. Já as medidas de Centralidade de grau média e Variância, dependem da quantidade de atores na rede. Estas duas últimas são apresentadas em conjunto com a quantidade de atores da rede.

Tanto o SAE-1 quanto o SAE-2 geraram redes totalmente desconexas em todos os períodos para o experimento AGRAWAL-3. Isso significa que em todos os períodos deste experimento as estratégias de inclusão e treinamento dos especialistas fizeram com que eles fossem criados e mantidos diversos. A taxa de acerto elevada e a rápida recuperação a mudança de conceito são indicadores de que manter um conjunto diverso permite melhorias no que diz respeito à taxa de acerto e a recuperação frente a mudanças de conceito. Na Figura 5.15 são

apresentadas as medidas de proeminência para o experimento AGRAWAL-3. É válido ressaltar que apesar do SAE-1 e SAE-2 terem a mesma topologia de rede, os resultados relativos à taxa de acerto são diferentes. Isto acontece devido à forma como os especialistas são treinados (*Bagging online*).

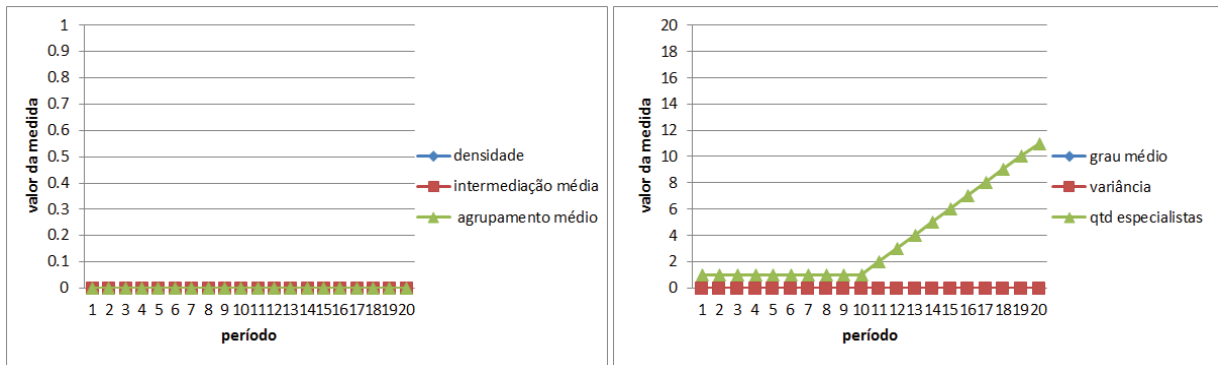


Figura 5.15: Medidas de proeminência (AGRAWAL-3, SAE-1 e SAE-2)

Nas Figuras 5.16 e 5.17 são apresentados os gráficos relativos ao experimento AGRAWAL-1 para o SAE-1 e SAE-2, respectivamente. Neste experimento é observado um padrão similar entre o SAE-1 e SAE-2 até o período 18, quando o SAE-2 remove um especialista redundante. No período 20 um novo especialista é adicionado para o SAE-2, contudo a rede se mantém totalmente desconexa (densidade = 0). A remoção do especialista no período 18 (SAE-2) impacta a taxa de acerto (ver Figura 5.11) e como um novo especialista é adicionado no período 20, é possível que esta remoção não tenha ocorrido no melhor momento.

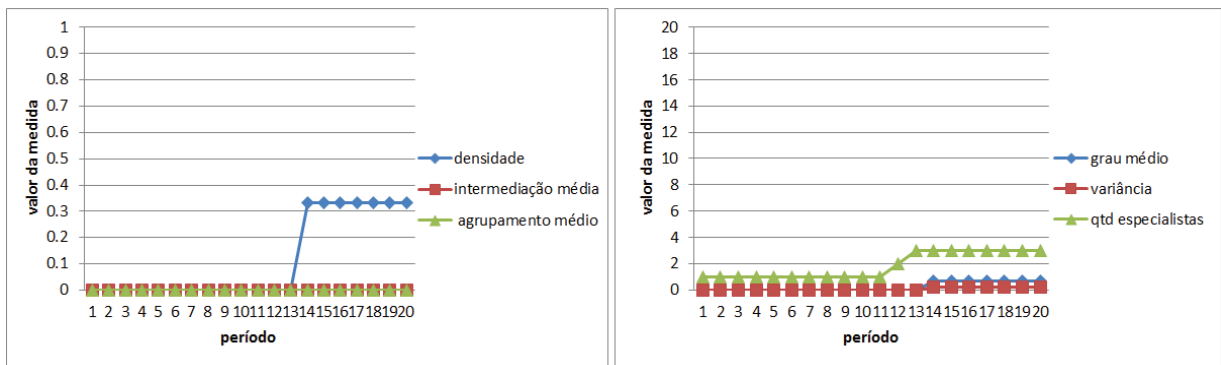


Figura 5.16: Medidas de proeminência (AGRAWAL-1, SAE-1)

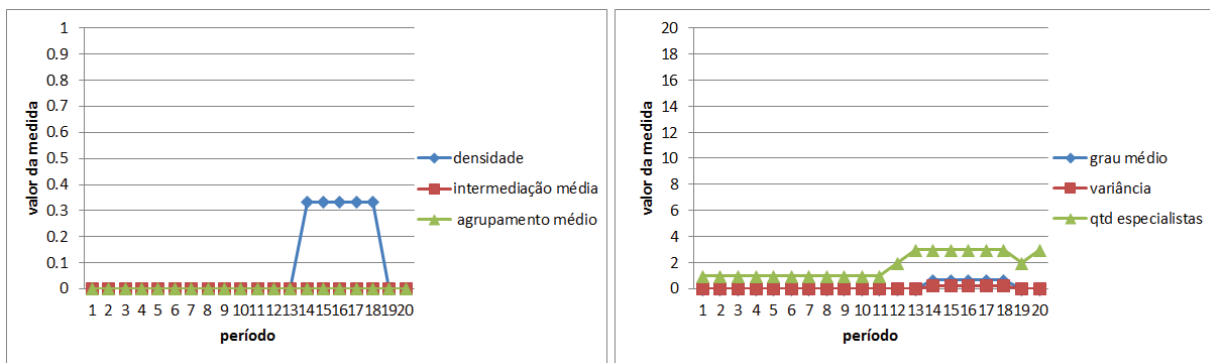


Figura 5.17: Medidas de proeminência (AGRAWAL-1, SAE-2)

Nas Figuras 5.18 e 5.19 são apresentados os gráficos relativos ao experimento SEA-4 para o SAE-1 e SAE-2, respectivamente. O experimento apresenta uma configuração em que o SAE-1 adiciona um novo especialista em praticamente todos os períodos. Enquanto que o SAE-2 mantém um conjunto estável com 3 especialistas. Apesar da quantidade de especialistas ser estável no SAE-2, após o período 3 sempre é removido um especialista redundante e adicionado um novo. Esse padrão cíclico é caracterizado pela remoção do especialista mais antigo por se tornar redundante com o segundo mais antigo. No período seguinte o especialista que era o segundo mais antigo no período anterior é removido e assim sucessivamente. Mesmo mantendo um conjunto maior o SAE-1 não obtém uma taxa de acerto melhor do que o SAE-2, especialmente nos períodos de mudança de conceito. Na Figura 5.20 é apresentado o gráfico relativo a taxa de acerto do SAE-1 e SAE-2 para o experimento SEA-4.

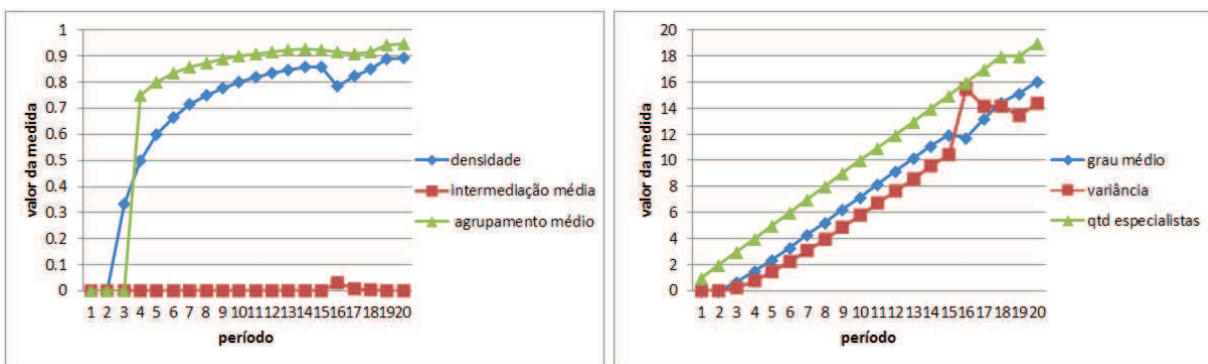


Figura 5.18: Medidas de proeminência (SEA-4, SAE-1)

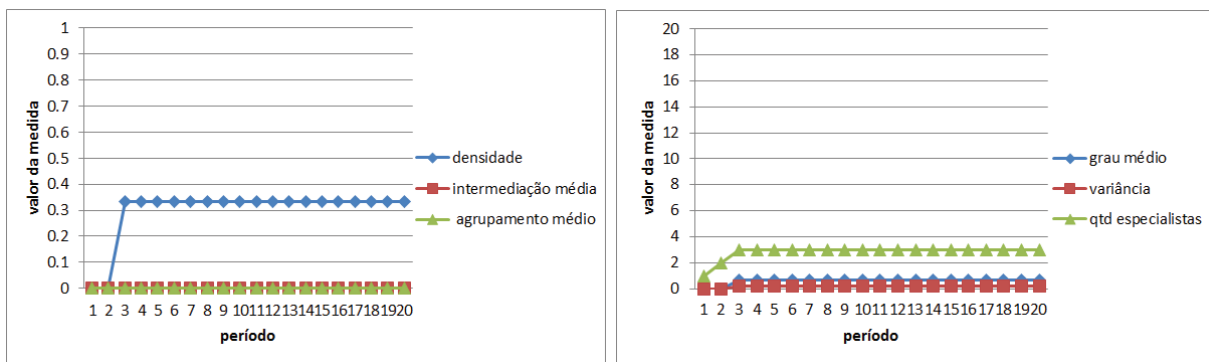


Figura 5.19: Medidas de proeminência (SEA-4, SAE-2)

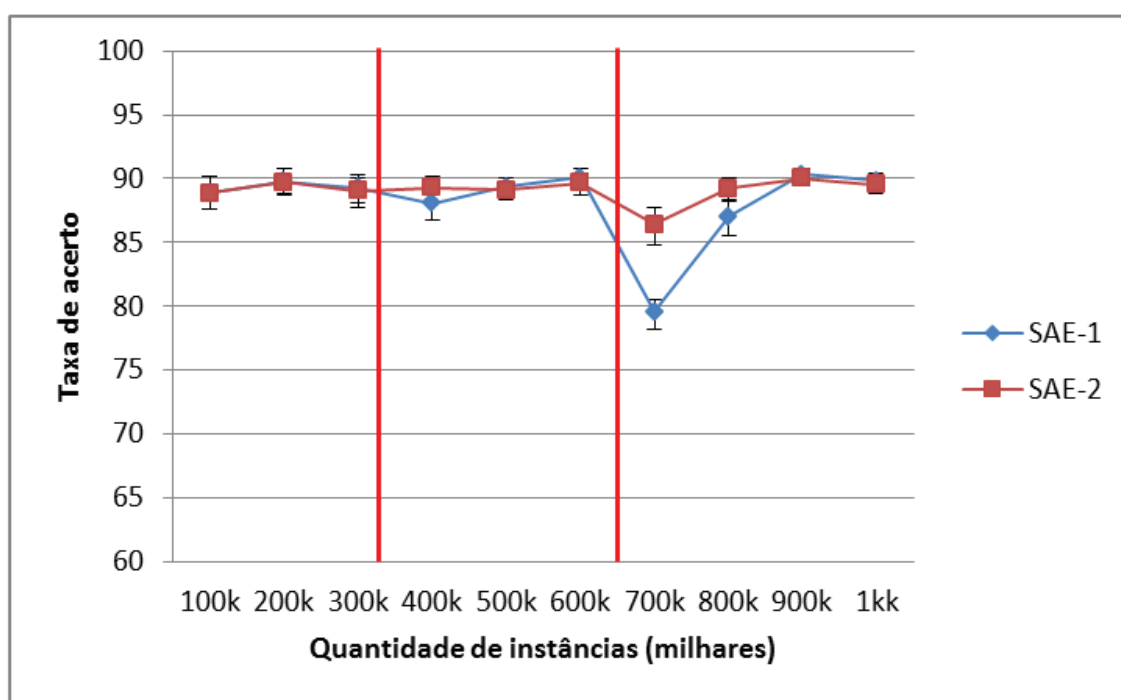


Figura 5.20: SAE (SEA-4)

Nas Figuras 5.21 e 5.22 são apresentados os gráficos relativos ao experimento AGRAWAL-2 para o SAE-1 e SAE-2, respectivamente. A análise das medidas de proeminência possibilita a identificação de padrões satisfatórios e insatisfatórios no SAE. Por exemplo, um aumento repentino na Centralidade de Intermediação média indica a existência de especialistas que intermediam conexões entre especialistas que não estão conectados entre si. Esta situação é indesejável, pois significa que especialistas que não são similares entre si fazem parte da mesma subrede. Quando especialistas que não são similares compõe a mesma subrede, a votação em dois

níveis é prejudicada. Na Figura 5.23 é apresentada a rede no período 19 do SAE-2, quando ocorre um aumento na centralidade de intermediação.

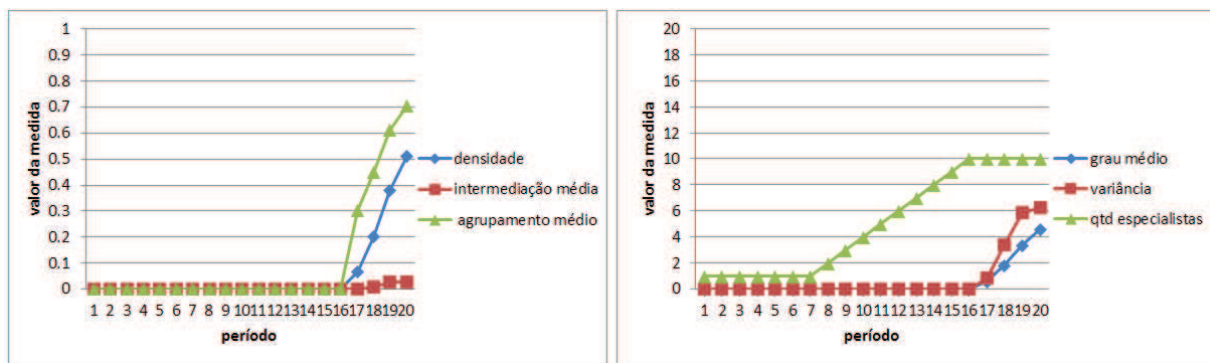


Figura 5.21: Medidas de proeminência (AGRAWAL-2, SAE-1)

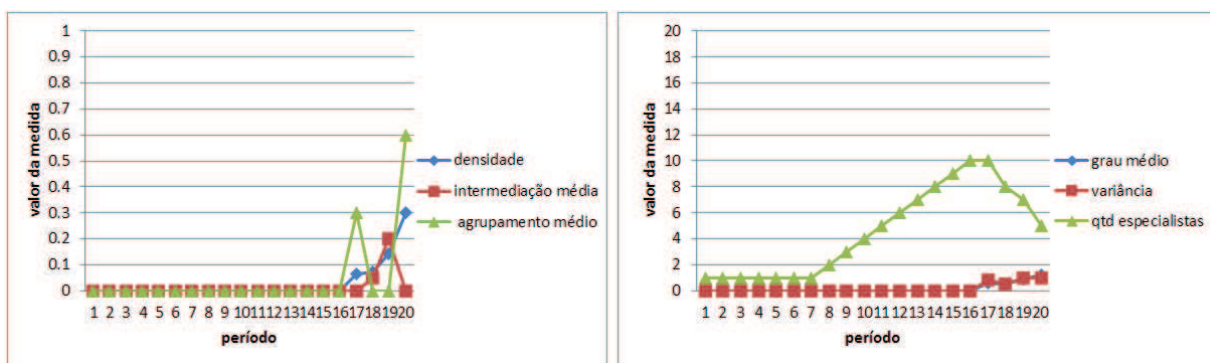


Figura 5.22: Medidas de proeminência (AGRAWAL-2, SAE-2)

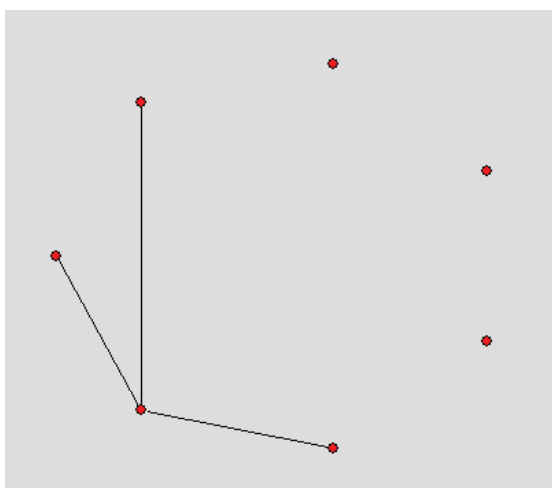


Figura 5.23: Rede no período 19 (AGRAWAL-2, SAE-2)

5.5 Considerações finais

Neste capítulo foram apresentados e discutidos experimentos realizados com o SAE (seção 5.2), comparações entre o SAE e outros algoritmos baseados em conjuntos de classificadores (seção 5.3) e uma análise sob a perspectiva da rede para o SAE (seção 5.4). Com base nos experimentos realizados é possível concluir que o SAE é um algoritmo adequado para Classificação *online*, não devido a alguns resultados isolados, mas sim pela possibilidade de aplicá-lo em *streams* de dados com características variadas (sem mudança e com mudança abrupta ou gradual) utilizando a mesma configuração de parâmetros e obtendo uma taxa de acerto equivalente a dos outros algoritmos para a Classificação *Online* que representam o estado da arte, mas, em geral, utilizando consideravelmente menos recursos computacionais. A abstração da Rede Social permite uma análise em diferentes níveis sobre o SAE. As medidas de proeminência podem ser usadas posteriormente como fontes de informação para alteração dos valores dos parâmetros em tempo de execução, por exemplo, aumentando o coeficiente de similaridade mínimo ($C_{s_{min}}$) quando for detectado um aumento na Centralidade de intermediação média.

Capítulo 6

Conclusão

As abordagens existentes para a geração de um conjunto diverso de classificadores em problemas de Classificação *Online* são, em geral, baseadas em alguma heurística de reamostragem (*Online Bagging* e suas variações), ou na assertiva de que o conjunto será diverso devido à inclusão dos classificadores ao longo do tempo (DWM). No algoritmo SAE, estas abordagens são exploradas com a adição da noção de similaridade entre especialistas. Dispor o conjunto como uma rede social inclui diversos benefícios e torna possível inferências a partir da estrutura da rede. As estratégias de remoção de classificadores e de combinação são possíveis graças à estrutura de rede formada. Agrupar especialistas, com base na similaridade, evita que um subconjunto com vários especialistas similares domine a classificação devido a sua cardinalidade. Além disso, é possível identificar especialistas redundantes para que estes sejam removidos, melhorando o desempenho no que diz respeito ao processamento e, em algumas situações, a taxa de acerto.

Os experimentos realizados indicam que a recuperação a mudanças de conceito graduais ou abruptas ocorre naturalmente devido a estratégia de combinação dos votos em dois níveis. A partir dos experimentos é possível concluir que o SAE seria o classificador indicado para *streams* de dados com mudança de conceito (abrupta ou gradual) em que o tempo de processamento deve ser o mais rápido possível sem perder a característica de rápida adaptação à mudança de conceito. O SAE tende a ter uma taxa de acerto não muito melhor do que um único classificador quando a

stream de dados não compreende mudanças de conceito. Essa característica é compartilhada por outros conjuntos de classificadores como o ASHT *Bagging* e o DWM.

A existência de vários parâmetros para o SAE torna a tarefa de determiná-los complexa. Possivelmente, seria obtida uma melhor taxa de acerto e um tempo de processamento menor caso esses parâmetros fossem diferentes para cada experimento. Uma evidência disso é o experimento RTC, em que o SAE obteve a pior combinação taxa de acerto e tempo de processamento. Ambas as configurações do SAE apresentadas (SAE-1 e SAE-2) especificam a taxa de acerto mínima da rede (Tx_{minR}) como 90%. Para o experimento RTC, em nenhum momento a rede consegue obter taxa de acerto superior a 80%, portanto, ao final de cada período um novo especialista sempre é adicionado. Além disso, especialistas dificilmente obtêm taxa de acerto menor que 50% (Tx_{minE}), tampouco se tornam redundantes devido à alta complexidade da *stream* de dados, e conseqüentemente, não são removidos. O resultado disso é uma rede composta por muitos especialistas, que não consegue melhorar a taxa de acerto significativamente em relação a um único classificador, pois não ocorrem mudanças de conceito (situação em que o SAE se sobressai).

Considerando classificadores base capazes de gerar o modelo de forma incremental (*Hoeffding Tree* ou *Naïve Bayes*), foi observado nos experimentos que os especialistas só são diversos por um determinado período de tempo após a sua inclusão na rede. Após vários períodos, especialistas que inicialmente tomavam decisões diferentes tendem a se tornarem cada vez mais similares devido à adaptação natural do classificador base. Essa observação reforça a necessidade de remover especialistas redundantes e também de combinar as decisões em dois níveis.

Para trabalhos futuros seria interessante explorar as medidas de proeminência obtidas para a rede. A aplicação do conhecimento obtido a partir das medidas de proeminência pode ser feito com o intuito de determinar se o modelo está ajustado para o fluxo de dados ou não, isto é, se ele se encontra estável (poucas alterações na topologia da rede) ou instável (remoções e adições frequentes). Outra possibilidade é utilizar as variações observadas nos valores das medidas de proeminência para alterar os valores dos parâmetros durante a execução do algoritmo, por exemplo, aumentar e diminuir o tamanho do período (c), ou ainda alterar os coeficientes de similaridade mínimo (CS_{min}) e máximo (CS_{max}). Outros trabalhos futuros incluem:

- Determinar um método de quebra de empate não aleatório.
- Alterar o método de combinação para que este contemple *streams* de dados com mais de duas classes.
- Desenvolvimento de outras estratégias de combinação baseadas em *cliques*, tríades balanceadas ou díades.
- Comparação entre outras estratégias de adição de especialistas baseadas não somente na taxa de acerto da rede e sem o limite de um classificador por período.
- Adição de um conjunto de validação para ser utilizado ao final do período com o intuito de avaliar a similaridade entre os especialistas. Desta forma, a similaridade não seria mensurada para cada instância de treinamento, mas somente ao final do período.

Referências Bibliográficas

AGRAWAL, R.; IMILIELINSKI, T; SWANI, A. *Database Mining: A Performance Perspective*, IEEE Transactions on Knowledge and Data Engineering, 5(6), 1993.

AHA, D. W.; KIBLER, D; ALBERT, M. K. *Instance-based learning algorithms*, Machine Learning, v. 6, p. 37-66, 1991.

ANTHONISSE, J. M. *The Rush in a Graph*, Amsterdam: Mathematische Centrum, 1971.

BAENA-GARCIA, M.; DEL CAMPO-AVILA, J.; FIDALGO, R.; BIFFET, A.; GAVALDÀ, R.; MORALES-BUENO, R. *Early drift detection method*, In Proc. of the 4th ECML PKDD int. workshop on Knowledge Discovery From Data Streams (IWKDDSD'06), p. 77–86, 2006.

BEAUCHAMP, M. A. *An improved index of centrality*, Behavioral Science, v. 10, p. 161-163, 1965.

BIFET, A.; FRANK, E.; HOLMES, G.; PFAHRINGER, B. *Accurate ensembles for data streams: Combining restricted hoeffding trees using stacking*, Journal of Machine Learning Research - Proceedings Track, v. 13, p. 225–240, 2010.

BIFET, A.; GAVALDÀ R. *Learning from time-changing data with adaptive Windowing*, In Proc. of SIAM int. conf. on Data Mining (SDM'07), p. 443–448, 2007.

BIFET, A.; HOLMES, G.; PFAHRINGER, B.; KIRKBY, R.; GAVALDÀ R. *New Ensemble Methods For Evolving Data Streams*, In 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009.

BIFET, A.; HOLMES, G.; KIRKBY, R.; PFAHRINGER, B. *MOA Data Stream Mining – A Practical Approach*, COSI (Centre for Open Software Innovation), 2011.

BIFET, A.; HOLMES, G.; KIRKBY, R.; PFAHRINGER, B. *MOA Massive On-line Analysis*, Disponível em: <http://moa.cs.waikato.ac.nz>, acesso em: 30 de abril de 2012.

BREIMAN L. *Bagging predictors*, Machine Learning, v. 24(2), p. 123–140, 1996.

BÜHLMANN, P.; YU, B. Analyzing bagging. *Annals of Statistics* 30, p. 927-961, 2002.

DOMINGOS, P.; HULTEN, G. *Mining high-speed data streams*, In International Conference on Knowledge Discovery and Data Mining, p. 71–80, 2000.

ERDÖS, P.; RÉNYI, A. *On the evolution of random graphs*, Publications of the Mathematical Institute of the Hungarian Academy of Sciences, v. 5, p. 17-61, 1960.

FREEMAN, L. C. *A set of measures of centrality based on betweenness*, *Sociometry*, v. 40, p. 35-41, 1977.

FREUND, Y.; SCHAPIRE, R. E. *A decision-theoretic generalization of on-line learning and an application to boosting*, *Journal of Computer and System Sciences*, v. 55(1), p. 119-139, 1997.

FRIEDKIN, N. E. *Theoretical foundations for centrality measures*, *American Journal of Sociology*, v. 96, p. 1478-1504, 1991.

GAMA, J.; MEDAS, P.; CASTILLO, G.; RODRIGUES, P. *Learning with drift detection*, SBIA Brazilian Symposium on Artificial Intelligence, p. 286-295, 2004.

GAMA, J.; SEBASTIÃO, R.; RODRIGUES, P. P. *Issues in evaluation of stream learning algorithms*, In 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 329-338, 2009.

GRANOVETTER, M. *The strength of weak ties*, American Journal of Sociology, v. 78, p. 1360-1380, 1973.

HAGER, P.; HARARY, F. *Eccentricity and centrality in networks*, Social Networks, v. 17, p. 57-63, 1969.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. *The WEKA Data Mining Software: An Update*, SIGKDD Explorations, V. 11, 2009.

HUBBEL, C. H. *An input-output approach to clique detection*, Sociometry, v. 28, p. 277-299, 1965.

KLINKENBERG, R.; JOACHIMS, T. *Detecting concept drift with support vector machines*, In ICML '00: Proc. of the 17th Int. Conf. on Machine Learning, p. 487-494, 2000.

KNOKE D.; BURT, R. S. *Prominence*, Applied Network Analysis, p. 195-222, 1983.

KOHAVI, Ron. *A study of cross-validation and bootstrap for accuracy estimation and model selection*, International Joint Conference on Artificial Intelligence, p. 1137-1145, 1995.

KOLTER, J. Z.; MALOOF, M. A. *Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift*, Proceedings of 3rd International IEEE Conference on Data Mining, IEEE Press, p. 123-130, 2003.

KUNCHEVA, L. I. *Combining Pattern Classifiers: Methods and Algorithms*, John Wiley & Sons, 2004.

LITTLESTONE, N.; WARMUTH, M. K. *The weighted majority algorithm*, Information and Computation 108(2), p. 212-261, 1994.

MILGRAM, S. *The small world problem*, Psychology today, v. 2, p. 60-67, 1967.

MORENO, J. L. *Who shall survive? : Foundations of Sociometry, Group Psychotherapy, and Sociodrama*, Nervous and Mental Disease Publishing Co., 1934.

OZA, N. C.; RUSSELL, S. *Online bagging and boosting*, Artificial Intelligence and Statistics, p. 105–112, 2001.

POLIKAR, R. *Ensemble based systems in decision making*, IEEE Circuits and Systems Magazine, v. 6(3), p. 21-45, 2006.

QUINLAN, R. *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo CA, 1993.

RÉKA, A.; BARABÁSI, A-L. *Statistical Mechanics of complex networks*. Reviews of modern physics, v. 74, 2002.

SABIDUSSI, G. *The centrality index of a graph*, Psychometrika, v. 31, p. 581-603, 1966.

SCHLIMMER, J.; GRANGER, R. *Incremental learning from noisy data*, Machine Learning, v. 1(3), p. 317–354, 1986.

STREET, N. W.; KIM, Y. *A streaming ensemble algorithm (SEA) for large-scale classification*, In International Conference on Knowledge Discovery and Data Mining, p. 377–382, 2001.

SU, B.; SHEN, Y.; XU, W. *Modeling concept drift from the perspective of classifiers*, In Prof. of the conference on Cybernetics and Intelligent Systems, p. 1055–1060, 2008.

TSYMBAL, A. *The problem of concept drift: Definitions and related Work*, Technical Report TCD-CS-2004-15, Department of Computer Science (University of Dublin, Trinity College), 2004.

WASSERMAN, S.; FAUST, K. *Social Network Analysis – Methods and Applications*, Cambridge Press, 1994.

WATTS, D. J.; STROGATZ, S.H. *Collective dynamics of small-world networks*, Nature, v. 393, n.6684, p. 440-442, 1998.

WIDMER, G., KUBAT M. *Learning in the presence of concept drift and hidden contexts*, Machine Learning, v. 23 (1), p. 69-101, 1996.

ŽLIOBAITE, I. *Adaptive Training Set Formation*, PhD Thesis, Vilnius University (Lithuania), 2010.

Apêndice A

Medidas de proeminência coletadas

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0	0	0	0	0	3	0
4	200000	1	0	1	2	0	3	3
5	250000	1	0	1	2	0	3	3
6	300000	1	0	1	2	0	3	3
7	350000	1	0	1	2	0	3	3
8	400000	1	0	1	2	0	3	3
9	450000	1	0	1	2	0	3	3
10	500000	1	0	1	2	0	3	3
11	550000	1	0	1	2	0	3	3
12	600000	1	0	1	2	0	3	3
13	650000	1	0	1	2	0	3	3
14	700000	1	0	1	2	0	3	3
15	750000	1	0	1	2	0	3	3
16	800000	1	0	1	2	0	3	3
17	850000	1	0	1	2	0	3	3
18	900000	1	0	1	2	0	3	3
19	950000	1	0	1	2	0	3	3
20	1000000	1	0	1	2	0	3	3

Tabela A.1: Medidas de proeminência (RTS, SAE-1)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0	0	0	0	0	3	0
4	200000	1	0	1	2	0	3	3
5	250000	0	0	0	0	0	1	0
6	300000	0	0	0	0	0	1	0
7	350000	0	0	0	0	0	1	0
8	400000	0	0	0	0	0	1	0
9	450000	0	0	0	0	0	1	0
10	500000	0	0	0	0	0	1	0
11	550000	0	0	0	0	0	1	0
12	600000	0	0	0	0	0	1	0
13	650000	0	0	0	0	0	1	0
14	700000	0	0	0	0	0	1	0
15	750000	0	0	0	0	0	1	0
16	800000	0	0	0	0	0	1	0
17	850000	0	0	0	0	0	1	0
18	900000	0	0	0	0	0	1	0
19	950000	0	0	0	0	0	1	0
20	1000000	0	0	0	0	0	1	0

Tabela A.2: Medidas de proeminência (RTS, SAE-2)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0	0	0	0	0	3	0
4	200000	0	0	0	0	0	4	0
5	250000	0	0	0	0	0	5	0
6	300000	0.066667	0	0	0.3333	0.222222	6	1
7	350000	0	0	0	0	0	7	0
8	400000	0	0	0	0	0	8	0
9	450000	0.138889	0.09375	0.185185185	1.1111	1.432099	9	5
10	500000	0.111111	0.197530864	0	1	1	10	5
11	550000	0.036364	0.022222222	0	0.3636	0.413223	11	2
12	600000	0	0	0	0	0	12	0
13	650000	0.166667	0.039141414	0.435897436	2	1.692308	13	13
14	700000	0.175824	0.333990796	0.328571429	2.2857	2.489796	14	16
15	750000	0.152381	0.045133438	0.522222222	2.1333	2.382222	15	16
16	800000	0.166667	0.030899471	0.520833333	2.5	2.75	16	20
17	850000	0.132353	0.063020833	0.543137255	2.1176	2.692042	17	18
18	900000	0.111111	0.062283737	0.355555556	1.8889	3.876543	18	17
19	950000	0.163743	0.030803679	0.400250627	2.9474	6.786704	19	28
20	1000000	0.163158	0.091515338	0.551904762	3.1	6.79	20	31

Tabela A.3: Medidas de proeminência (RTC, SAE-1)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0	0	0	0	0	3	0
4	200000	0	0	0	0	0	4	0
5	250000	0	0	0	0	0	5	0
6	300000	0.066667	0	0	0.3333	0.222222	6	1
7	350000	0	0	0	0	0	7	0
8	400000	0	0	0	0	0	8	0
9	450000	0.138889	0.09375	0.185185185	1.1111	1.432099	9	5
10	500000	0.111111	0.197530864	0	1	1	10	5
11	550000	0.036364	0.022222222	0	0.3636	0.413223	11	2
12	600000	0	0	0	0	0	12	0
13	650000	0.166667	0.039141414	0.435897436	2	1.692308	13	13
14	700000	0.175824	0.333990796	0.328571429	2.2857	2.489796	14	16
15	750000	0.152381	0.045133438	0.522222222	2.1333	2.382222	15	16
16	800000	0.166667	0.030899471	0.520833333	2.5	2.75	16	20
17	850000	0.132353	0.063020833	0.543137255	2.1176	2.692042	17	18
18	900000	0.111111	0.062283737	0.355555556	1.8889	3.876543	18	17
19	950000	0.163743	0.030803679	0.400250627	2.9474	6.786704	19	28
20	1000000	0.163158	0.091515338	0.551904762	3.1	6.79	20	31

Tabela A.4: Medidas de proeminência (RTC, SAE-2)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0.333333	0	0	0.6667	0.222222	3	1
4	200000	0.5	0	0.75	1.5	0.75	4	3
5	250000	0.6	0	0.8	2.4	1.44	5	6
6	300000	0.666667	0	0.833333333	3.3333	2.222222	6	10
7	350000	0.714286	0	0.857142857	4.2857	3.061224	7	15
8	400000	0.714286	0	0.857142857	4.2857	3.061224	7	15
9	450000	0.75	0	0.875	5.25	3.9375	8	21
10	500000	0.75	0	0.875	5.25	3.9375	8	21
11	550000	0.777778	0	0.888888889	6.2222	4.839506	9	28
12	600000	0.8	0	0.9	7.2	5.76	10	36
13	650000	0.8	0.000833333	0.888888889	8	6.545455	11	44
14	700000	0.833333	0	0.916666667	9.1667	7.638889	12	55
15	750000	0.846154	0	0.923076923	10.154	8.591716	13	66
16	800000	0.857143	0	0.928571429	11.143	9.55102	14	78
17	850000	0.866667	0	0.933333333	12.133	10.51556	15	91
18	900000	0.875	0	0.9375	13.125	11.48438	16	105
19	950000	0.882353	0	0.941176471	14.118	12.45675	17	120
20	1000000	1	0	1	11	0	12	66

Tabela A.5: Medidas de proeminência (SEA-1, SAE-1)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0.333333	0	0	0.6667	0.222222	3	1
4	200000	0.333333	0	0	0.6667	0.222222	3	1
5	250000	0.333333	0	0	0.6667	0.222222	3	1
6	300000	0.333333	0	0	0.6667	0.222222	3	1
7	350000	0.333333	0	0	0.6667	0.222222	3	1
8	400000	0.333333	0	0	0.6667	0.222222	3	1
9	450000	0.333333	0	0	0.6667	0.222222	3	1
10	500000	0.333333	0	0	0.6667	0.222222	3	1
11	550000	0.333333	0	0	0.6667	0.222222	3	1
12	600000	0.333333	0	0	0.6667	0.222222	3	1
13	650000	0.333333	0	0	0.6667	0.222222	3	1
14	700000	0.333333	0	0	0.6667	0.222222	3	1
15	750000	0.333333	0	0	0.6667	0.222222	3	1
16	800000	0.333333	0	0	0.6667	0.222222	3	1
17	850000	0.333333	0	0	0.6667	0.222222	3	1
18	900000	0.333333	0	0	0.6667	0.222222	3	1
19	950000	0.333333	0	0	0.6667	0.222222	3	1
20	1000000	0.333333	0	0	0.6667	0.222222	3	1

Tabela A.6: Medidas de proeminência (SEA-1, SAE-2)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0.333333	0	0	0.6667	0.222222	3	1
4	200000	0.5	0	0.75	1.5	0.75	4	3
5	250000	0.6	0	0.8	2.4	1.44	5	6
6	300000	0.666667	0	0.833333333	3.3333	2.222222	6	10
7	350000	0.714286	0	0.857142857	4.2857	3.061224	7	15
8	400000	0.75	0	0.875	5.25	3.9375	8	21
9	450000	0.777778	0	0.888888889	6.2222	4.839506	9	28
10	500000	0.8	0	0.9	7.2	5.76	10	36
11	550000	0.818182	0	0.909090909	8.1818	6.694215	11	45
12	600000	0.833333	0	0.916666667	9.1667	7.638889	12	55
13	650000	0.846154	0	0.923076923	10.154	8.591716	13	66
14	700000	0.857143	0	0.928571429	11.143	9.55102	14	78
15	750000	0.847619	0.000570858	0.914529915	11.867	10.38222	15	89
16	800000	0.783333	0.03026455	0.914835165	11.75	15.4375	16	94
17	850000	0.823529	0.005952381	0.909803922	13.176	14.14533	17	112
18	900000	0.838235	0.002777778	0.91092437	13.412	13.18339	17	114
19	950000	0.888889	0	0.944444444	15.111	13.4321	18	136
20	1000000	0.866667	0.00014652	0.928571429	13	11.375	16	104

Tabela A.7: Medidas de proeminência (SEA-2, SAE-1)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0.333333	0	0	0.6667	0.222222	3	1
4	200000	0.333333	0	0	0.6667	0.222222	3	1
5	250000	0.333333	0	0	0.6667	0.222222	3	1
6	300000	0.333333	0	0	0.6667	0.222222	3	1
7	350000	0.333333	0	0	0.6667	0.222222	3	1
8	400000	0.333333	0	0	0.6667	0.222222	3	1
9	450000	0.333333	0	0	0.6667	0.222222	3	1
10	500000	0.333333	0	0	0.6667	0.222222	3	1
11	550000	0.333333	0	0	0.6667	0.222222	3	1
12	600000	0.333333	0	0	0.6667	0.222222	3	1
13	650000	0.333333	0	0	0.6667	0.222222	3	1
14	700000	0.333333	0	0	0.6667	0.222222	3	1
15	750000	0.333333	0	0	0.6667	0.222222	3	1
16	800000	0.333333	0	0	0.6667	0.222222	3	1
17	850000	0.333333	0	0	0.6667	0.222222	3	1
18	900000	0.333333	0	0	0.6667	0.222222	3	1
19	950000	0.333333	0	0	0.6667	0.222222	3	1
20	1000000	0.333333	0	0	0.6667	0.222222	3	1

Tabela A.8: Medidas de proeminência (SEA-2, SAE-2)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0.333333	0	0	0.6667	0.222222	3	1
4	200000	0.5	0	0.75	1.5	0.75	4	3
5	250000	0.6	0	0.8	2.4	1.44	5	6
6	300000	0.666667	0	0.833333333	3.3333	2.222222	6	10
7	350000	0.714286	0	0.857142857	4.2857	3.061224	7	15
8	400000	0.714286	0	0.857142857	4.2857	3.061224	7	15
9	450000	0.75	0	0.875	5.25	3.9375	8	21
10	500000	0.75	0	0.875	5.25	3.9375	8	21
11	550000	0.777778	0	0.888888889	6.2222	4.839506	9	28
12	600000	0.8	0	0.9	7.2	5.76	10	36
13	650000	0.818182	0	0.909090909	8.1818	6.694215	11	45
14	700000	0.833333	0	0.916666667	9.1667	7.638889	12	55
15	750000	0.846154	0	0.923076923	10.154	8.591716	13	66
16	800000	0.857143	0	0.928571429	11.143	9.55102	14	78
17	850000	0.866667	0	0.933333333	12.133	10.51556	15	91
18	900000	0.875	0	0.9375	13.125	11.48438	16	105
19	950000	0.875	0	0.9375	13.125	11.48438	16	105
20	1000000	1	0	1	12	0	13	78

Tabela A.9: Medidas de proeminência (SEA-3, SAE-1)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0.333333	0	0	0.6667	0.222222	3	1
4	200000	0.333333	0	0	0.6667	0.222222	3	1
5	250000	0.333333	0	0	0.6667	0.222222	3	1
6	300000	0.333333	0	0	0.6667	0.222222	3	1
7	350000	0.333333	0	0	0.6667	0.222222	3	1
8	400000	0.333333	0	0	0.6667	0.222222	3	1
9	450000	0.333333	0	0	0.6667	0.222222	3	1
10	500000	0.333333	0	0	0.6667	0.222222	3	1
11	550000	0.333333	0	0	0.6667	0.222222	3	1
12	600000	0.333333	0	0	0.6667	0.222222	3	1
13	650000	0.333333	0	0	0.6667	0.222222	3	1
14	700000	0.333333	0	0	0.6667	0.222222	3	1
15	750000	0.333333	0	0	0.6667	0.222222	3	1
16	800000	0.333333	0	0	0.6667	0.222222	3	1
17	850000	0.333333	0	0	0.6667	0.222222	3	1
18	900000	0.333333	0	0	0.6667	0.222222	3	1
19	950000	0.333333	0	0	0.6667	0.222222	3	1
20	1000000	0.333333	0	0	0.6667	0.222222	3	1

Tabela A.10: Medidas de proeminência (SEA-3, SAE-2)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0.333333	0	0	0.6667	0.222222	3	1
4	200000	0.5	0	0.75	1.5	0.75	4	3
5	250000	0.6	0	0.8	2.4	1.44	5	6
6	300000	0.666667	0	0.833333333	3.3333	2.222222	6	10
7	350000	0.714286	0	0.857142857	4.2857	3.061224	7	15
8	400000	0.75	0	0.875	5.25	3.9375	8	21
9	450000	0.777778	0	0.888888889	6.2222	4.839506	9	28
10	500000	0.8	0	0.9	7.2	5.76	10	36
11	550000	0.818182	0	0.909090909	8.1818	6.694215	11	45
12	600000	0.833333	0	0.916666667	9.1667	7.638889	12	55
13	650000	0.846154	0	0.923076923	10.154	8.591716	13	66
14	700000	0.857143	0	0.928571429	11.143	9.55102	14	78
15	750000	0.857143	0.000196232	0.923076923	12	10.4	15	90
16	800000	0.783333	0.03026455	0.914835165	11.75	15.4375	16	94
17	850000	0.823529	0.005952381	0.909803922	13.176	14.14533	17	112
18	900000	0.849673	0.002076125	0.916666667	14.444	14.1358	18	130
19	950000	0.888889	0	0.944444444	15.111	13.4321	18	136
20	1000000	0.894737	0	0.947368421	16.105	14.40997	19	153

Tabela A.11: Medidas de proeminência (SEA-4, SAE-1)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	2	0
3	150000	0.333333	0	0	0.6667	0.222222	3	1
4	200000	0.333333	0	0	0.6667	0.222222	3	1
5	250000	0.333333	0	0	0.6667	0.222222	3	1
6	300000	0.333333	0	0	0.6667	0.222222	3	1
7	350000	0.333333	0	0	0.6667	0.222222	3	1
8	400000	0.333333	0	0	0.6667	0.222222	3	1
9	450000	0.333333	0	0	0.6667	0.222222	3	1
10	500000	0.333333	0	0	0.6667	0.222222	3	1
11	550000	0.333333	0	0	0.6667	0.222222	3	1
12	600000	0.333333	0	0	0.6667	0.222222	3	1
13	650000	0.333333	0	0	0.6667	0.222222	3	1
14	700000	0.333333	0	0	0.6667	0.222222	3	1
15	750000	0.333333	0	0	0.6667	0.222222	3	1
16	800000	0.333333	0	0	0.6667	0.222222	3	1
17	850000	0.333333	0	0	0.6667	0.222222	3	1
18	900000	0.333333	0	0	0.6667	0.222222	3	1
19	950000	0.333333	0	0	0.6667	0.222222	3	1
20	1000000	0.333333	0	0	0.6667	0.222222	3	1

Tabela A.12: Medidas de proeminência (SEA-4, SAE-2)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	1	0
3	150000	0	0	0	0	0	1	0
4	200000	0	0	0	0	0	1	0
5	250000	0	0	0	0	0	1	0
6	300000	0	0	0	0	0	1	0
7	350000	0	0	0	0	0	1	0
8	400000	0	0	0	0	0	1	0
9	450000	0	0	0	0	0	1	0
10	500000	0	0	0	0	0	1	0
11	550000	0	0	0	0	0	1	0
12	600000	0	0	0	0	0	2	0
13	650000	0	0	0	0	0	3	0
14	700000	0.333333	0	0	0.6667	0.222222	3	1
15	750000	0.333333	0	0	0.6667	0.222222	3	1
16	800000	0.333333	0	0	0.6667	0.222222	3	1
17	850000	0.333333	0	0	0.6667	0.222222	3	1
18	900000	0.333333	0	0	0.6667	0.222222	3	1
19	950000	0.333333	0	0	0.6667	0.222222	3	1
20	1000000	0.333333	0	0	0.6667	0.222222	3	1

Tabela A.13: Medidas de proeminência (AGRAWAL-1, SAE-1)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	1	0
3	150000	0	0	0	0	0	1	0
4	200000	0	0	0	0	0	1	0
5	250000	0	0	0	0	0	1	0
6	300000	0	0	0	0	0	1	0
7	350000	0	0	0	0	0	1	0
8	400000	0	0	0	0	0	1	0
9	450000	0	0	0	0	0	1	0
10	500000	0	0	0	0	0	1	0
11	550000	0	0	0	0	0	1	0
12	600000	0	0	0	0	0	2	0
13	650000	0	0	0	0	0	3	0
14	700000	0.333333	0	0	0.6667	0.222222	3	1
15	750000	0.333333	0	0	0.6667	0.222222	3	1
16	800000	0.333333	0	0	0.6667	0.222222	3	1
17	850000	0.333333	0	0	0.6667	0.222222	3	1
18	900000	0.333333	0	0	0.6667	0.222222	3	1
19	950000	0	0	0	0	0	2	0
20	1000000	0	0	0	0	0	3	0

Tabela A.14: Medidas de proeminência (AGRAWAL-1, SAE-2)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	1	0
3	150000	0	0	0	0	0	1	0
4	200000	0	0	0	0	0	1	0
5	250000	0	0	0	0	0	1	0
6	300000	0	0	0	0	0	1	0
7	350000	0	0	0	0	0	1	0
8	400000	0	0	0	0	0	2	0
9	450000	0	0	0	0	0	3	0
10	500000	0	0	0	0	0	4	0
11	550000	0	0	0	0	0	5	0
12	600000	0	0	0	0	0	6	0
13	650000	0	0	0	0	0	7	0
14	700000	0	0	0	0	0	8	0
15	750000	0	0	0	0	0	9	0
16	800000	0	0	0	0	0	10	0
17	850000	0.066667	0	0.3	0.6	0.84	10	3
18	900000	0.2	0.007201646	0.45	1.8	3.36	10	9
19	950000	0.377778	0.026234568	0.61	3.4	5.84	10	17
20	1000000	0.511111	0.028292181	0.705238095	4.6	6.24	10	23

Tabela A.15: Medidas de proeminência (AGRAWAL-2, SAE-1)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	1	0
3	150000	0	0	0	0	0	1	0
4	200000	0	0	0	0	0	1	0
5	250000	0	0	0	0	0	1	0
6	300000	0	0	0	0	0	1	0
7	350000	0	0	0	0	0	1	0
8	400000	0	0	0	0	0	2	0
9	450000	0	0	0	0	0	3	0
10	500000	0	0	0	0	0	4	0
11	550000	0	0	0	0	0	5	0
12	600000	0	0	0	0	0	6	0
13	650000	0	0	0	0	0	7	0
14	700000	0	0	0	0	0	8	0
15	750000	0	0	0	0	0	9	0
16	800000	0	0	0	0	0	10	0
17	850000	0.066667	0	0.3	0.6	0.84	10	3
18	900000	0.071429	0.047619048	0	0.5	0.5	8	2
19	950000	0.142857	0.2	0	0.8571	0.979592	7	3
20	1000000	0.3	0	0.6	1.2	0.96	5	3

Tabela A.16: Medidas de proeminência (AGRAWAL-2, SAE-2)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	1	0
3	150000	0	0	0	0	0	1	0
4	200000	0	0	0	0	0	1	0
5	250000	0	0	0	0	0	1	0
6	300000	0	0	0	0	0	1	0
7	350000	0	0	0	0	0	1	0
8	400000	0	0	0	0	0	1	0
9	450000	0	0	0	0	0	1	0
10	500000	0	0	0	0	0	1	0
11	550000	0	0	0	0	0	2	0
12	600000	0	0	0	0	0	3	0
13	650000	0	0	0	0	0	4	0
14	700000	0	0	0	0	0	5	0
15	750000	0	0	0	0	0	6	0
16	800000	0	0	0	0	0	7	0
17	850000	0	0	0	0	0	8	0
18	900000	0	0	0	0	0	9	0
19	950000	0	0	0	0	0	10	0
20	1000000	0	0	0	0	0	11	0

Tabela A.17: Medidas de proeminência (AGRAWAL-3, SAE-1)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	1	0
3	150000	0	0	0	0	0	1	0
4	200000	0	0	0	0	0	1	0
5	250000	0	0	0	0	0	1	0
6	300000	0	0	0	0	0	1	0
7	350000	0	0	0	0	0	1	0
8	400000	0	0	0	0	0	1	0
9	450000	0	0	0	0	0	1	0
10	500000	0	0	0	0	0	1	0
11	550000	0	0	0	0	0	2	0
12	600000	0	0	0	0	0	3	0
13	650000	0	0	0	0	0	4	0
14	700000	0	0	0	0	0	5	0
15	750000	0	0	0	0	0	6	0
16	800000	0	0	0	0	0	7	0
17	850000	0	0	0	0	0	8	0
18	900000	0	0	0	0	0	9	0
19	950000	0	0	0	0	0	10	0
20	1000000	0	0	0	0	0	11	0

Tabela A.18: Medidas de proeminência (AGRAWAL-3, SAE-2)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	1	0
3	150000	0	0	0	0	0	1	0
4	200000	0	0	0	0	0	1	0
5	250000	0	0	0	0	0	1	0
6	300000	0	0	0	0	0	1	0
7	350000	0	0	0	0	0	1	0
8	400000	0	0	0	0	0	2	0
9	450000	0	0	0	0	0	3	0
10	500000	0	0	0	0	0	3	0
11	550000	0	0	0	0	0	3	0
12	600000	0	0	0	0	0	3	0
13	650000	0.333333	0	0	0.6667	0.222222	3	1
14	700000	0.166667	0	0	0.5	0.25	4	1
15	750000	0.1	0	0	0.4	0.24	5	1
16	800000	0	0	0	0	0	6	0
17	850000	0	0	0	0	0	7	0
18	900000	0.285714	0	0.571428571	1.7143	2.204082	7	6
19	950000	0.47619	0	0.714285714	2.8571	3.265306	7	10
20	1000000	0.666667	0.008333333	0.8	4	2.857143	7	14

Tabela A.19: Medidas de proeminência (AGRAWAL-4, SAE-1)

período	instâncias	densidade	intermediação média	agrupamento médio	grau médio	variância	qtd especialistas	qtd conexões
1	50000	0	0	0	0	0	1	0
2	100000	0	0	0	0	0	1	0
3	150000	0	0	0	0	0	1	0
4	200000	0	0	0	0	0	1	0
5	250000	0	0	0	0	0	1	0
6	300000	0	0	0	0	0	1	0
7	350000	0	0	0	0	0	1	0
8	400000	0	0	0	0	0	2	0
9	450000	0	0	0	0	0	3	0
10	500000	0	0	0	0	0	3	0
11	550000	0	0	0	0	0	3	0
12	600000	0	0	0	0	0	3	0
13	650000	0.333333	0	0	0.6667	0.222222	3	1
14	700000	0.166667	0	0	0.5	0.25	4	1
15	750000	0.1	0	0	0.4	0.24	5	1
16	800000	0	0	0	0	0	6	0
17	850000	0	0	0	0	0	7	0
18	900000	0.285714	0	0.571428571	1.7143	2.204082	7	6
19	950000	0.166667	0	0	0.5	0.25	4	1
20	1000000	0.333333	0	0	0.6667	0.222222	3	1

Tabela A.20: Medidas de proeminência (AGRAWAL-4, SAE-2)

Apêndice B

O *Framework* MOA

O *framework* MOA (Bifet *et al*, 2012) foi desenvolvido com a finalidade de prover ferramentas para o problema de análise de fontes de dados de fluxo contínuo (*data streams*). Além de algoritmos para a Classificação *Online*, o MOA compreende ainda:

- **Geradores de dados.** *Random Tree Generator*, SEA, STAGGER, etc.
- **Métodos de avaliação.** *Periodic Holdout*, *Test-Then-Train*, etc.
- **Agrupamento *Online* (*clustering*).**
- **Estatísticas.** Taxa de acerto média, Tempo de processamento¹⁵, etc.

O MOA possui uma interface gráfica (GUI) e também permite a execução de testes em modo texto (linha de comando), o que facilita a configuração e execução de testes em lote. Na Figura A.1 é apresentada a interface gráfica do MOA.

O MOA foi implementado em Java e segue o mesmo padrão do *framework* para mineração de dados WEKA (Hall *et al*, 2009), permitindo a sua extensão a partir da inclusão de novos algoritmos. É possível adicionar ao MOA algoritmos para classificação, agrupamento, geração de dados e métodos de avaliação. Além disso, pode-se utilizar algoritmos implementados no WEKA diretamente a partir do MOA e vice-versa.

¹⁵ As estimativas de tempo no MOA são feitas mensurando o tempo de execução da *thread* na CPU.

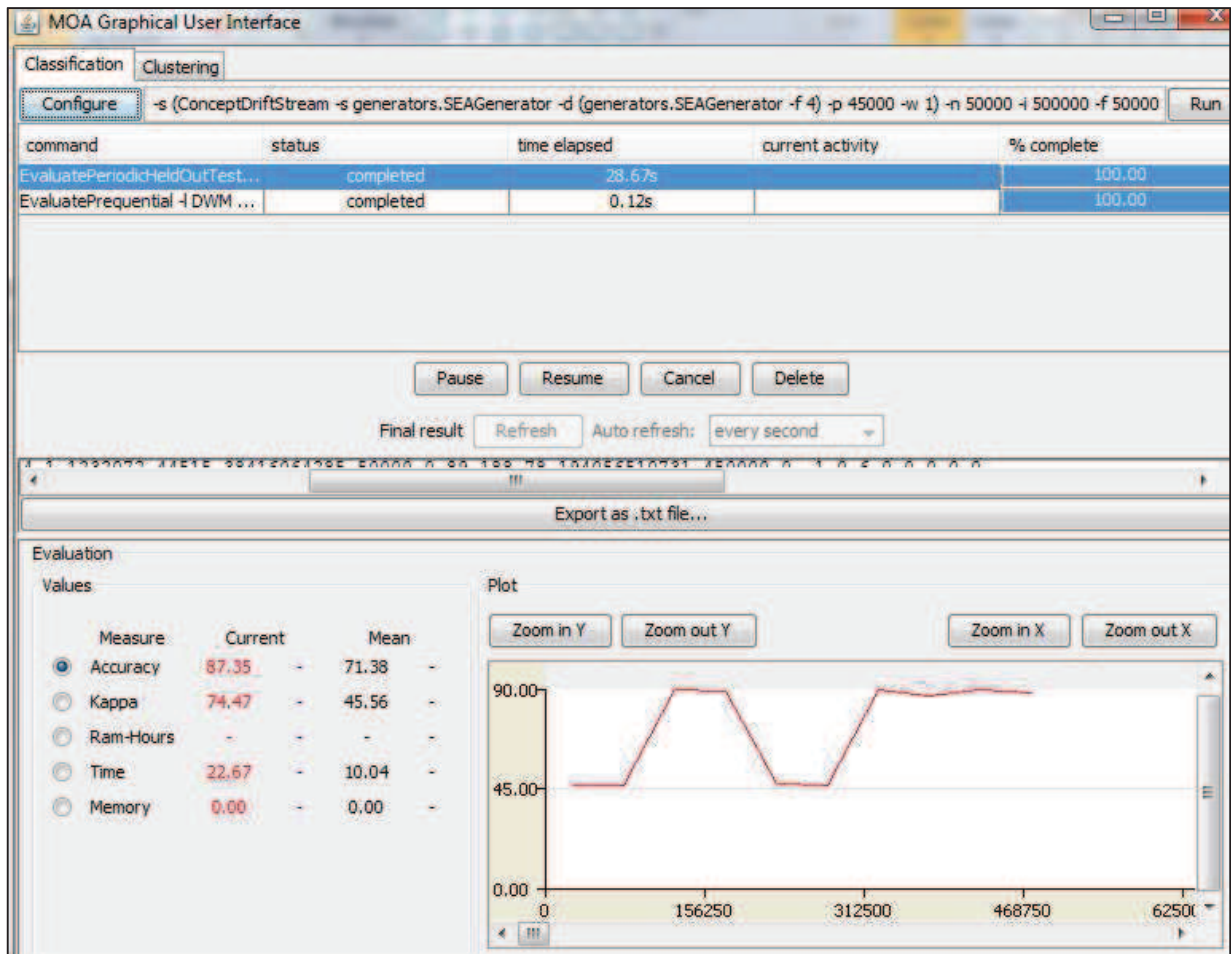


Figura B.1: Interface gráfica do MOA

No MOA são disponibilizados vários algoritmos para Classificação *Online*, porém o algoritmo DWM não faz parte deste grupo. Uma vez que o algoritmo DWM é utilizado neste para comparação com o algoritmo SAE, este foi implementado no MOA.

Para a adicionar de um novo classificador no MOA é preciso herdar a classe abstrata *AbstractClassifier*. Esta classe define o comportamento padrão de todo classificador para que este seja integrado ao *framework* e a GUI sem a necessidade de adaptações no código fonte do MOA. Os principais métodos da classe *AbstractClassifier* e que devem ser sobrescritos nas classes que dela herdam são:

1. **void resetLearningImpl()**. Prepara o classificador para uma nova execução.

2. **void trainOnInstanceImpl(Instance instance).** Treina o classificador com uma instância.
3. **double[] getVotesForInstance(Instance instance).** Retorna a classe predita para a instância.

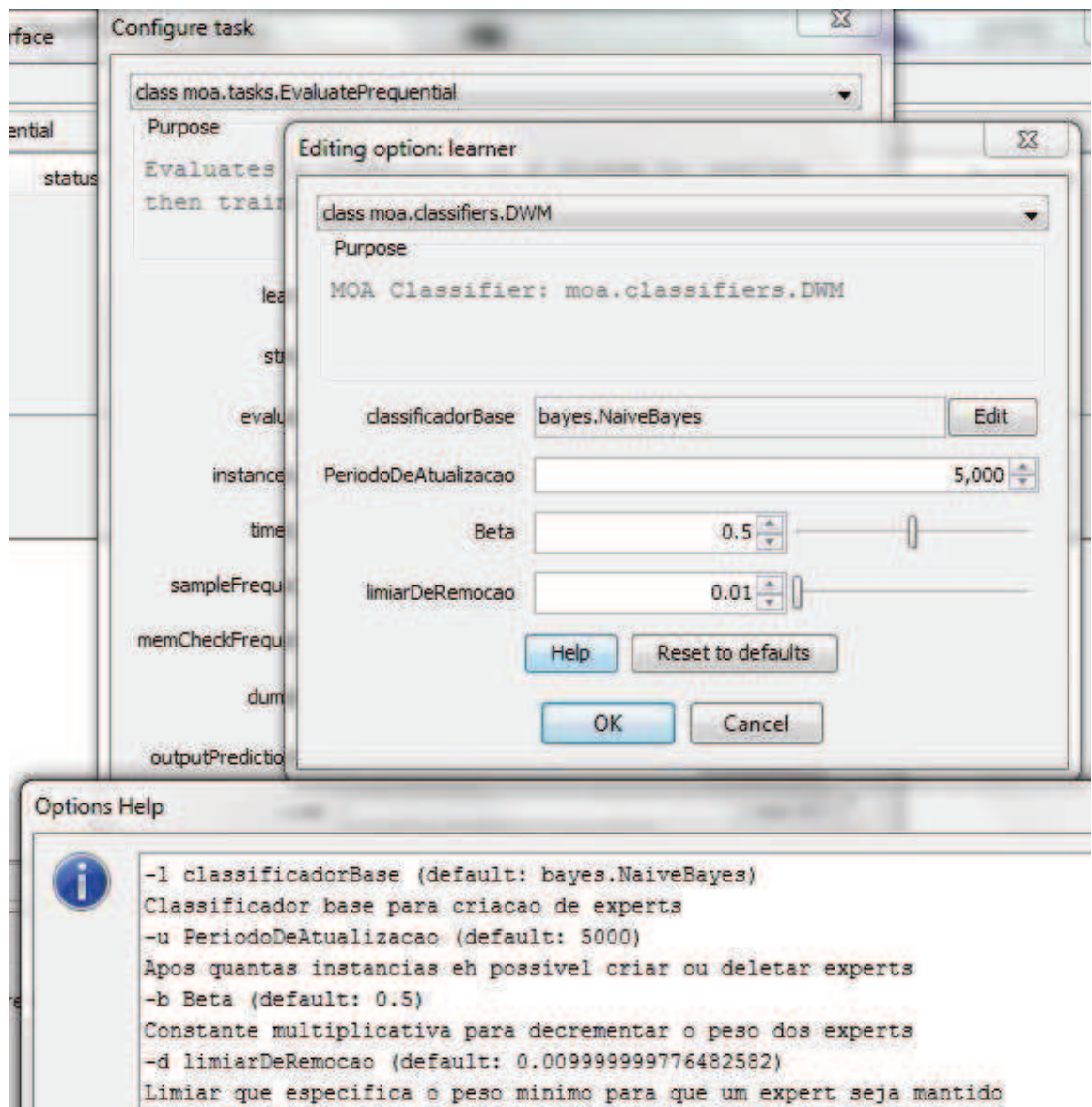


Figura B.2: Parâmetros DWM: Visão da GUI.

O valor retornado pelo método 3 é um vetor onde cada posição representa uma classe possível para o problema. A posição que tiver o maior valor é assumida como a classe predita.

Por exemplo: Considerando 3 classes possíveis, e o retorno de *getVotesForInstance* como (0.0, 9.0, 4.0), isso significa que a classe no índice 1 do vetor é a classe predita. Alguns classificadores, incluindo o DWM, possuem parâmetros definidos pelo usuário. Estes devem ser declarados como atributos com nível de acesso público na classe que implementa o classificador para que possam ser alterados na GUI. Na Figura A.2 são apresentados os parâmetros para o DWM na interface gráfica do MOA.

Apêndice C

Resultados dos experimentos

Nas Tabelas B.1, B.2 e B.3 são apresentados os resultados dos experimentos para média da taxa de acerto, tempo de processamento e quantidade média de classificadores, respectivamente. Os valores em negrito e grifados em verde representam os melhores resultados para o experimento, enquanto que os valores grifados em azul indicam o segundo lugar.

As médias apresentadas são relativas à média de dez períodos de amostras (*sample frequency*) do método de avaliação *Prequential*. Cada experimento foi repetido 10 vezes variando a semente aleatória do gerador de dados, logo cada uma das amostras do *Prequential* usadas para tirar a média apresentada na Tabela B.1 é a média daquela amostra em cada uma das 10 repetições. Nas Tabelas B.1, B.2 e B.3 para cada experimento (linha) é destacado o melhor resultado em negrito com grifo em verde e o segundo melhor resultado com grifo azul.

Média da taxa de acerto								
gerador	SAE-1	SAE-2	SAE-3	SAE-4	DWM-1	DWM-2	BagADWIN	BagASHT
RTS	95.17	94.77	73.65	73.63	93.34	84.64	95.92	89.64
RTC	73.74	75.97	61.86	65.97	70.43	63.61	78.10	71.11
SEA-1	89.46	89.22	88.46	88.74	88.42	86.39	89.26	89.05
SEA-2	88.02	89.05	86.69	87.64	88.39	86.39	88.43	88.68
SEA-3	89.15	89.03	88.29	88.46	88.05	85.81	89.25	88.77
SEA-4	88.22	89.08	86.91	87.89	88.46	86.43	89.48	88.80
AGRAWAL-1	91.02	90.67	76.93	76.96	78.73	73.24	93.82	91.69
AGRAWAL-2	89.02	88.50	68.37	70.86	85.56	80.78	94.73	90.95
AGRAWAL-3	88.70	88.51	72.84	75.27	75.65	69.33	88.45	88.48
AGRAWAL-4	88.42	88.25	67.81	70.61	83.80	78.39	91.60	88.78

Tabela C.1: Média da taxa de acerto

Tempo de Execução (CPU TIME - segundos)								
gerador	SAE-1	SAE-2	SAE-3	SAE-4	DWM-1	DWM-2	BagADWIN	BagASHT
RTS	35.03	18.21	65.23	32.11	41.92	63.44	147.68	86.28
RTC	889.53	860.50	249.70	309.43	361.25	432.07	774.34	659.30
SEA-1	49.46	19.63	22.84	17.39	23.52	31.89	82.45	34.85
SEA-2	58.83	19.48	27.15	17.66	23.73	32.13	80.18	34.94
SEA-3	45.75	17.55	22.16	17.45	23.24	32.32	78.68	44.53
SEA-4	54.30	17.85	25.96	17.62	24.03	32.23	73.54	40.74
AGRAWAL-1	36.24	37.66	54.60	34.23	38.23	52.65	98.04	67.06
AGRAWAL-2	68.72	65.42	56.88	45.61	35.02	44.24	128.81	95.07
AGRAWAL-3	42.50	43.46	57.55	37.88	38.29	65.50	113.24	81.40
AGRAWAL-4	50.93	57.13	58.04	44.00	37.68	59.79	100.98	89.19

Tabela C.2: Tempo de execução (CPU Time) mensurado em segundos

Quantidade de classificadores (média)								
gerador	SAE-1	SAE-2	SAE-3	SAE-4	DWM-1	DWM-2	BagADWIN	BagASHT
RTS	3	1	9	4	4	5	10	10
RTC	12	12	4	6	4	4	10	10
SEA-1	9	3	6	4	6	6	10	10
SEA-2	11	3	7	4	6	6	10	10
SEA-3	9	3	5	4	6	6	10	10
SEA-4	11	3	7	4	6	6	10	10
AGRAWAL-1	3	4	7	4	5	5	10	10
AGRAWAL-2	5	5	8	6	5	5	10	10
AGRAWAL-3	4	4	8	5	5	5	10	10
AGRAWAL-4	5	4	8	6	5	5	10	10

Tabela C.3: Quantidade média de classificadores