UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Camila Maria Campos

Comitê de Classificadores em Bases de Dados Transacionais Desbalanceadas com Seleção de Características Baseada em Padrões Minerados

Juiz de Fora

2016

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Camila Maria Campos

Comitê de Classificadores em Bases de Dados Transacionais Desbalanceadas com Seleção de Características Baseada em Padrões Minerados

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Carlos Cristiano Hasenclever Borges
Coorientador: Victor Ströele de Andrade Menezes

Juiz de Fora
2016

Ficha catalográfica elaborada através do programa de geração
automática da Biblioteca Universitária da UFJF,
com os dados fornecidos pelo(a) autor(a)

Maria Campos, Camila.
Comitê de Classificadores em Bases de Dados Transacionais Desbalanceadas com Seleção de Características Baseada em Padrões Minerados / Camila Maria Campos. -- 2016.
95 f.

Orientador: Carlos Cristiano Hasenclever Borges
Coorientador: Victor Ströele de Andrade Menezes
Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação, 2016.

1. Mineração de Dados. 2. Regra de Associação. 3. Classificação. 4. Balanceamento em Bases de Dados. 5. Seleção de Características. I. Hasenclever Borges, Carlos Cristiano, orient. II. Ströele de Andrade Menezes, Victor, coorient. III. Título.

Camila Maria Campos

Comitê de Classificadores em Bases de Dados Transacionais Desbalanceadas com Seleção de Características Baseada em Padrões Minerados

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em 29 de Janeiro de 2016.

BANCA EXAMINADORA

Prof. D.Sc. Carlos Cristiano Hasenclever Borges - Orientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Victor Ströele de Andrade Menezes
Universidade Federal de Juiz de Fora

Prof. D.Sc. Heder Soares Bernardino
Universidade Federal de Juiz de Fora

Prof. D.Sc. Cristiano Grijó Pitangui
Universidade Federal dos Vales do Jequitinhonha e Mucuri

Dedico este trabalho a meus pais
Lourival e Maria da Consolação
e minha irmã Madalena.

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus por tudo que ele tem feito em minha vida.

Quem conviveu comigo durante esses dois anos sabe que não foi fácil, que a batalha foi dura, mas consegui vencer e hoje estou muito feliz, pois cada batalha vencida foi um aprendizado que me fez crescer e evoluir tanto pessoalmente quanto profissionalmente.

Agradeço aos meus pais Lourival e Maria da Consolação pelo imenso apoio e incentivo dado durante toda minha vida, e por tudo que vocês fizeram por mim, sei que além de estar vivendo o meu sonho estou vivendo o sonho de vocês também.

Agradeço a minha irmã Madalena, pelo acolhimento em sua casa, pela ajuda nas horas difíceis, por tudo que você tem feito por mim. Sou muito grata a você minha irmã.

Agradeço também ao meu namorado Guilherme, por ter me incentivado a participar do processo seletivo do mestrado, essa vitória também é um pouco sua. Obrigada pela compreensão e carinho durante todos esses anos.

Agradeço de forma especial ao meu orientador Cristiano que graças a ele, hoje sou uma mestranda, prestes a concluir meu curso, obrigada pelo acolhimento, por ter aceitado ser meu orientador, pelos conselhos e por tudo que você tem feito por mim durante esses dois anos.

Agradeço também meu coorientador Victor, também por ter aceito ser meu coorientador, e que juntamente com o Cristiano está me guiando para que o trabalho seja concluído e realizado da melhor maneira possível.

Foram inúmeras as vezes que pensei em desistir, mas Cristiano e Victor me levantavam dando conselhos, me fazendo refletir sobre tudo, e o que posso fazer para retribuir é agradecer tudo que vocês fizeram por mim.

Ganhei diversos amigos durante o mestrado, mas uma pessoa se destacou entre eles, ela me acolheu de braços abertos quando eu cheguei no PGCC e não conhecia ninguém, aos poucos fomos virando amigas e hoje eu agradeço a ela por tudo, pelos conselhos que as vezes são meio estranhos, mas esse é o jeito dela, pelas brigas, que foram poucas, por muitas vezes ter brigado por mim, ter aberto meus olhos, por virar uma leoa para me proteger quando alguma coisas estava acontecendo. Enfim, obrigada por tudo Alessandra, sei que nossa amizade irá continuar por muito tempo.

Agradeço também minha amiga Brígida, que apesar de já estar junto de Deus, as vezes sinto que ela me dá força para continuar, e vencer mais essa etapa em minha vida.

Agradeço também aos meu novos amigos Vinícius, Humberto, Tassio, Marcos e Rafael, pelas risadas nos intervalos das aulas, por me levantar nas horas de dificuldade e tristeza. Obrigada por tudo meus amigos.

Agradeço também minha amiga Ana Mara Figueiredo que vem me acompanhando nessa trajetória de estudos desde a graduação, e que por muitas vezes me ajudou a levantar a cabeça e seguir em frente.

Agradeço também a CAPES pelo apoio financeiro.

Enfim agradeço a todos que de alguma forma contribuíram para a realização deste trabalho e a todos que torceram e torcem por mim.

RESUMO

Os resultados dos problemas de classificação por regras de associação sofrem grande influência da estrutura dos dados que estão sendo utilizados. Uma dificuldade na área é a resolução de problemas de classificação quando se trata de bases de dados desbalanceadas. Assim, o presente trabalho apresenta um estudo sobre desbalanceamento em bases de dados transacionais, abordando os principais métodos utilizados na resolução do problema de desbalanceamento.

Além disso, no que tange ao desbalanceamento, este trabalho propõe um modelo para realizar o balanceamento entre classes, sendo realizados experimentos com diferentes métodos de balanceamento e métodos *ensemble*, baseados em comitê de classificadores. Tais experimentos foram realizados em bases transacionais e não transacionais com o intuito de validar o modelo proposto e melhorar a predição do algoritmo de classificação por regras de associação. bases de dados não transacionais também foram utilizadas nos experimentos, com o objetivo de verificar o comportamento do modelo proposto em tais bases.

Outro fator importante no processo de classificação é a dimensão da base de dados que, quando muito grande, pode comprometer o desempenho dos classificadores. Neste trabalho, também é proposto um modelo de seleção de características baseado na classificação por regras de associação. Para validar o modelo proposto, também foram realizados experimentos aplicando diferentes métodos de seleção nas bases de dados. Os resultados da classificação obtidos utilizando as bases contendo as características selecionadas pelos métodos, foram comparados para validar o modelo proposto, tais resultados apresentaram-se satisfatórios em relação aos demais métodos de seleção.

Palavras-chave: Mineração dados, Regra de Associação, Classificação, Balanceamento em Bases de Dados, Seleção de Características.

ABSTRACT

The results of Classification Based on Associations Rules (CBA) are greatly influenced by the used data structure. A difficulty in this area is solving classification problems when it comes to unbalanced databases. Thus, this paper presents a study of unbalance in transactional and non-transactional databases, addressing the main methods used to solve the unbalance problem.

In addition, with respect to the unbalance problem, this paper proposes a model to reach the balance between classes, conducting experiments with different methods of balancing and ensemble methods based on classifiers committee. These experiments were performed in transactional and non-transactional databases, in order to validate the proposed model and improve Classification Based on Associations Rules prediction.

Another important factor in the classification process is database dimensionality, because when too large, it can compromise the classifiers performance. In this work, it is also proposed a feature selection model based on the rules of CBA. Aiming to validate this model, experiments were also performed applying different features selection methods in the databases.The classification results obtained using the bases containing the features selected by the methods were compared to validate the proposed model, these results were satisfactory in comparison with other methods of selection.

Keywords: Data Mining, Association Rule, Classification, Balancing Database, Feature Selection.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AdaBoost Adaptative Boosting

ADT Alternating Decision Tree

ARN Association Rules Networks

Bagging Booststrap Aggregating

CAR Class Association Rules

CBA Classification Based on Association

CCDD Comitê de Classificadores para Dados Desbalanceados

DIC Dynamic Itemset Counting

EBS Ensemble Based Systems

SCBR Seleção de Características Baseada em Regras

SEAR Sequential Efficient Association Rules

SMOTE Synthetic Minority Over-sampling TEchnique

SNP Single Nucleotide Polimorphism

SPEAR Sequential Partitioning Efficient Association Rules

SUMÁRIO

1 INTRODUÇÃO

Com o passar dos anos e com o avanço da tecnologia alguns estabelecimentos comerciais, que antes armazenavam seus dados em cadernos, cadernetas ou de maneira impressa para controle e consultas futuras, sentiram a necessidade da criação de um sistema de armazenamento de informações de forma organizada e de fácil acesso, visto que a quantidade de informações aumentava de forma significativa (CHEN et al., 1996).

Para realizar tal tarefa foram utilizados bancos de dados, cujo propósito é armazenar informações de forma organizada. Fazendo uma analogia, antigamente um funcionário de uma empresa representava uma pasta de arquivo, hoje em dia ele representa um registro no banco de dados. Diversos bancos de dados foram construídos desde então, cada um de acordo com a necessidade da empresa.

Tem-se como exemplo real da utilização de banco de dados para armazenamento de informações, quando os clientes vão em supermercados, padarias ou lojas em geral para realizar suas compras, na maioria das vezes, é realizada a leitura do código do produto que está sendo adquirido. Esse código é lido e enviado para um banco de dados e, futuramente, pode ser utilizado para consultas ou mesmo para auxiliar na tomada de decisão da empresa.

Diariamente, dados e mais dados são inseridos nessas bases o que as tornam volumosas e difíceis de serem exploradas. Ao longo do tempo, diversas tecnologias foram criadas com o intuito de explorar grandes volumes de dados. A mineração de dados é a tecnologia mais utilizada atualmente. Diversas definições sobre mineração de dados podem ser encontradas na literatura (HAND et al., 2001) (CABENA et al., 1998)(FAYYAD et al., 1996) e ela pode ser entendida como: explorar grande quantidade de dados a procura de padrões consistentes e úteis. Tais padrões dificilmente seriam descobertos explorando os dados de forma manual, devido ao grande volume de dados e a complexidade das relações.

Existem diversas maneiras de realizar a tarefa de mineração de dados e também diversos algoritmos que realizam tais tarefas. Segundo (LAROSE, 2005), as tarefas mais comuns de Mineração de Dados são Classificação, Regressão, Estimativa, Previsão, Agrupamento e Associação. Tais modelos apresentam estratégias para o problema de descoberta de conhecimento de grandes bases de dados.

A mineração em bases de dados transacionais ainda pode ser uma tarefa com um alto custo computacional, principalmente devido ao número de itens envolvidos e ao crescimento contínuo da base. Diversos trabalhos foram realizados na tentativa de minimizar este custo e melhorar o desempenho. Em (HAN et al., 2000) apresenta-se três técnicas para que a mineração em bases transacionais seja mais eficiente.

A primeira técnica indica que a base de dados deve ser comprimida para um melhor desempenho, ou seja, a estrutura deve ser composta por uma quantidade menor de dados, com perda mínima de informação. A segunda técnica é a mineração baseada em árvore, que consiste em um método de crescimento padrão da árvore para evitar a geração de um grande número de candidatos, o que também pode fazer com que a mineração seja um processo custoso. Finalmente, a terceira é um método de divisão e conquista baseado no conceito de particionamento da base, utilizado para decompor a mineração em um conjunto de tarefas menores, a fim de reduzir o espaço de busca.

Além do custo computacional, outros fatores podem prejudicar o desempenho da tarefa de mineração de dados. Uma das características que podem comprometer a qualidade da mineração é o desbalanceamento da base manipulada em relação a possíveis rótulos. Bases de dados com classes desbalanceadas podem representar um problema quando se tem um procedimento de aprendizado supervisionado, visto que o classificador obtido tende a ser enviesado com mais intensidade pela classe com maior quantidade de dados, também conhecida como classe majoritária, e desprezar as classes menos representativas, conhecidas como classe minoritária (QAZI; RAZA, 2012).

Em contrapartida, existem técnicas utilizadas para melhorar a predição de classificação. Mas o que é predição de classificação? Predição de classificação significa que, após treinar um classificador utilizando uma determinada base com instâncias que representam uma classe, o classificador através de uma hipótese de indução gerada, deve inferir qual a classe de determinadas instâncias baseando-se no processo e aprendizagem realizado anteriormente. Essas técnicas, baseadas em bases de dados rotuladas, são conhecidas como aprendizagem supervisionada.

Um dos métodos utilizados para melhorar a predição de um classificador é a seleção de características, ou seja, algumas características ou atributos contidos na base de dados são mais representativos (relevantes) para a representação da base como um todo. Tais características são selecionadas utilizando métodos robustos de busca na base de da-

dos podendo ser combinados com algoritmos de classificação, como é o caso do modelo conhecido como encapsulado.

Este trabalho trata diretamente do desempenho de uma abordagem específica para a classificação em bases transacionais desbalanceadas e a utilização dos padrões minerados para um procedimento de seleção de características. A seguir, apresenta-se a sequência em que será desenvolvido o trabalho.

1.1 DEFINIÇÃO DO PROBLEMA

O trabalho trata do problema de classificação através de regras de associação utilizando bases transacionais desbalanceadas.

Uma outra questão a ser considerada para esta classe de dados transacionais, é a tendência de se ter muitos itens ou atributos disponíveis porém com alto desequilíbrio em relação à relevância na classificação. Desta forma, um procedimento de seleção de itens ou características é determinante para realizar predições eficientes. Modelos de seleção de características baseados em regras de associação previamente geradas são pouco comuns na literatura.

Problemas como não obter regras de associação após o processo de classificação, impossibilitam o procedimento de seleção de características. Outro comportamento comum quando se usa regras de associação em bases desbalanceadas é o desequilíbrio entre o número de regras obtidas para a classe majoritária em relação às minoritárias, podendo, inclusive, não obter regras para estas classes. Este fato influencia muito uma seleção de características ou atributos baseada em regras de associação, visto que as regras obtidas servirão de base para a seleção das características.

Com o intuito de minimizar o problema de desbalaceamento em bases de dados visando a geração de regras de associação relevantes para selecionar características, são propostos dois métodos que serão apresentados ao longo do desenvolvimento do trabalho.

1.2 MOTIVAÇÃO

Diversas são as áreas que tem as bases de dados transacionais como sendo a representação de melhor forma de armazenar os dados gerados, sendo, talvez, o caso mais representativo descrito por bases relacionadas a transações comerciais como, por exemplo, em farmácias,

supermercados, etc. Entre as principais características deste tipo de dado, motivado por seu padrão construtivo, está sua estreita relação com métodos baseados em regras de associação para viabilizar minerações de interesse.

Em algumas situações, é relevante que se avalie tais bases transacionais no contexto de aprendizagem supervisionado. A maneira mais simples de viabilizar esta adaptação é por meio da transformação de um item ou atributo (binário ou discreto) da base de dados em rótulo ou classe. Dependendo do item escolhido, é comum a obtenção de bases desbalanceadas após a transformação.

Assim, as principais questões relativas a este tipo de dado e as consequências de tal adaptação, são tratados neste trabalho visando que se tenha melhores ferramentas para a busca de conhecimentos relevantes nos dados. Inicialmente, cabe ressaltar, que todos os desenvolvimentos foram realizados tendo como base regras de associação, objetivando confirmar sua aplicabilidade na área.

1.3 OBJETIVOS

Considera-se que o objetivo principal do trabalho é desenvolver uma estratégia de aprendizagem supervisionada baseado em regras de associação para o trato de base de dados transacionais desbalanceadas, com identificação das características ou itens relevantes.

Para que este objetivo seja alcançado, é necessária a realização de algumas etapas que direcionam objetivos específicos a serem tratados, a saber:

- Avaliação do uso de regras de associação na construção de modelos de aprendizado supervisionado;
- Desenvolvimento de um modelo para classificação de bases transacionais desbalanceadas que apresente robustez na predição com qualidade nas regras de associação obtidas;
- Apresentação de uma abordagem de seleção de características baseadas em regras de associação.

1.4 ORGANIZAÇÃO DO TEXTO

No Capítulo 2 serão descritos os conceitos básicos relacionados às regras de associação utilizando bases de dados transacionais, sendo apresentado o algoritmo de referência *Apriori*. São descritos também alguns algoritmos que realizam mineração em regras de associação.

O Capítulo 3 trata do uso específico de regras de associação em aprendizagem supervisionada, bem como do caso específico em que se tem bases transacionais desbalanceada. São mencionados alguns métodos que realizam manipulações em bases de dados desbalanceadas.

No Capítulo 4 serão apresentados dois modelos clássicos baseados em comitê de classificadores utilizados para melhorar a predição na classificação. Apresenta-se, também, o modelo proposto neste trabalho para balanceamento de dados, baseado em comitê de classificadores, com o intuito de melhorar a predição de classificação, com a construção de regras de associação consistentes.

Segue-se, no Capítulo 5 apresentando os modelos usuais para seleção de características em aprendizagem supervisionada. Descreve-se, então, uma abordagem proposta para a seleção de características baseada em regras de associação.

O Capítulo 6 trata da realização de experimentos computacionais para avaliar os modelos propostos para bases de dados transacionais. Testes com o comitê de classificadores para dados desbalanceados e com a estratégia de seleção baseada em regras apresentadas são realizados de forma a possibilitar uma real avaliação dos modelos. Uma análise abordando os resultados obtidos também é disponibilizada.

Finalizando, no Capítulo 7, apresentam-se as conclusões do trabalho realizado, bem como indicam-se alguns possíveis trabalhos futuros a serem desenvolvidos.

2 REGRA DE ASSOCIAÇÃO

Regras de associação representam padrões onde a ocorrência de eventos simultâneos é alta, ou seja, a probabilidade de um determinado item B ocorrer quando um item A também ocorre, em uma mesma transação. O objetivo de minerar regras de associação é localizar todos os conjuntos de itens que ocorrem com frequência de forma simultânea na base de dados e gerar regras a partir dos conjuntos encontrados. As regras de associação para a descoberta de conhecimento são consideradas como sendo um método não supervisionado, e é o segundo método mais utilizado em aplicações, sendo o primeiro método a tarefa de classificação (HU et al., 1999),(MA, 1998).

O problema de mineração de regras de associação foi apresentado por (AGRAWAL et al., 1993), onde é tratado o problema de minerar grandes quantidades de dados de uma cesta de supermercado, a fim de localizar conjuntos de itens que atendessem a uma confiança mínima. A mineração de regra de associação pode ser aplicada em diversos problemas, como por exemplo, análise de uma cesta de supermercado, análise de compras utilizando cartão de crédito, informações sobre possíveis produtos comprados por determinados clientes, empréstimos bancários, detecção de fraudes, dentre outros, sendo a mais utilizada em análises de cestas de supermercado (BERRY; LINOFF, 2004).

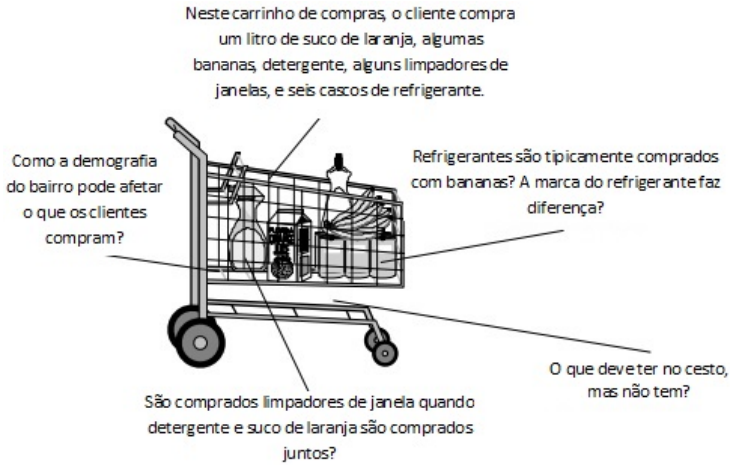

Figura 2.1: Carrinho de compras (BERRY; LINOFF, 2004)

A figura 2.1 representa compras realizadas por um cliente. Analisando previamente sem muitos detalhes, é possível perceber que o cliente comprou suco de laranja, bananas, detergentes, limpadores de vidro, dentre outros. A partir da análise das compras pode-se

conhecer um pouco mais da rotina dos clientes. Clientes podem ir ao supermercado muitas vezes para comprar poucos produtos ou para fazer uma compra enorme, por exemplo, uma lista contendo todos os produtos que serão utilizados durante o mês. Cada cliente pode comprar produtos diferentes uns dos outros, pois o que é interessante para um pode não ser interessante para outro. Além de produtos, outros fatores interessantes podem ser analisados, como por exemplo, a hora em que os produtos são mais vendidos, quais produtos são mais vendidos em conjunto e que são mais propícios à promoção. Tais informações podem impactar na disposição dos produtos no supermercado, fazendo com que clientes sejam atraídos por produtos que certamente não seriam comprados se estivessem dispostos de outra maneira.(BERRY; LINOFF, 2004).

Uma regra de associação tem a seguinte forma: Se comprou $\boldsymbol{A}$ então compra $\boldsymbol{B}$, cuja representação é $\boldsymbol{A} \rightarrow \boldsymbol{B}$, onde $\boldsymbol{A}$ é o antecedente da regra e $\boldsymbol{B}$ é o consequente da regras. Algumas medidas são associadas às regras e utilizadas para informar o nível de importância e confiabilidade de uma regra. Essas medidas são suporte, confiança, *lift* e convicção, também chamadas de medidas de interesse, sendo o suporte e a confiança as mais utilizadas. O suporte representa a relevância da regra. Regras com valores de suporte considerado elevados tendem a ser mais importantes. A confiança representa a confiabilidade de uma regra; regras com valores de confiança elevados tendem ser mais confiáveis (MOTTA, 2010), (HAN et al., 2011).

Em regras de associação $\boldsymbol{R}$ a medida de suporte $\boldsymbol{S}$ em uma transação $\boldsymbol{A} \rightarrow \boldsymbol{B}$ representa as transações em que $\boldsymbol{A}$ e $\boldsymbol{B}$ aparecem juntos em um conjunto de dados $\mathbf{D}$ divido pelo número total de transações $\boldsymbol{T}$, definido por:

$$\boldsymbol{S} = \frac{(\boldsymbol{A} \cup \mathbf{B})}{T} \tag{2.1}$$

A confiança C em uma transação $\boldsymbol{A} \rightarrow \boldsymbol{B}$ representa as transação em que $\boldsymbol{A}$ e $\boldsymbol{B}$ ocorrem juntos em um conjunto de dados $\mathbf{D}$, divido pelo número de transações $\boldsymbol{T}$ que contém somente $\boldsymbol{A}$, é definido por:

$$C = \frac{(\mathbf{A} \cup \mathbf{B})}{T(A)} \tag{2.2}$$

Alguns algoritmo para gerar regras de associação permitem que os usuários estabeleçam previamente, tanto o valor mínimo de suporte *(MinSupport)* e confiança *(MinMetric)* quanto a métrica*(MetricType)* utilizada para cálculo de importância da regra. As regras

de associação podem ser consideradas fortes ou fracas dependendo dos valores de confiança e suporte. Regras de associação fortes são regras que possuem um valor de suporte e confiança maiores ou iguais aos valores preestabelecidos pelos usuários.

Para melhor entendimento do que vem a ser regras fortes e regras fracas, tem-se como exemplo uma regra de associação contendo os itens $Manteiga \rightarrow Leite$, cujo suporte e confiança preestabelecidos pelo usuário são respectivamente: **suporte**= 50% **confiança**= 80%

$$Manteiga \rightarrow Leite\ [0{,}7][0{,}9]$$

Após a realização da mineração por regra de associação a regra $Manteiga \rightarrow Leite$ apresentou um suporte igual a 70%, isso significa que em 70% de todas as transações os itens Manteiga e Leite foram comprados juntos, e a confiança de 90% significa que 90% dos clientes que compraram Manteiga também compraram Leite. Assim a regra $Manteiga \rightarrow Leite$ é considerada forte, pois apresentou um suporte e confiança superiores aos valores preestabelecidos.

2.1 ALGORITMO APRIORI

Além de resolver o problema de mineração de regra de associação (AGRAWAL et al., 1993), propôs um algoritmo chamado *Apriori* capaz de reduzir o espaço de busca para obter melhor eficiência na geração dos níveis de *itemsets* (i.e., conjunto de itens ordenados em ordem alfabética) frequentes. O algoritmo *Apriori* é um dos algoritmos mais conhecidos e utilizados em mineração de regra de associação e representa um avanço na tecnologia de mineração de dados (HASTIE et al., 2005)(SRIVASTAVA et al., 2000). As três fases do algoritmo são realizadas a cada iteração até que o critério de parada seja atendido, ou seja, até que o número de gerações seja igual à zero ou quando não for possível fazer a relação entre os elementos.

O algorimto *Apriori* é executado em três etapas:

- Fase de geração dos candidados(*itemsets*).
- Fase de Poda.
- Fase de validação.

É importante ressaltar que a primeira fase é a mais exaustiva, visto que o algoritmo faz consultas no banco de dados para gerar a lista de candidatos, isso faz com que a primeira etapa seja computacionalmente custosa. Após esse primeiro passo, todos as demais são de rápida execução.

A seguir é apresentado um exemplo do funcionando do algoritmo *Apriori*, bem como as fases de geração, poda e validação.

A tabela 2.1 representa transações em uma base de dados, com seus respectivos identificadores. Cada transação representa as compras de um clientes.

ID da Compra	Compra
1	Leite, Pão, Manteiga, Suco
2	Leite, Açucar, Biscoito
3	Leite, Ovo, Manteiga
4	Pão, Biscoito, Café

Tabela 2.1: Transações no banco de dados

Os itens Leite, Pão, Manteiga, Suco, Açúcar, Biscoito, Ovo e Café serão substituídos respectivamente, por 1, 2, 3, 4, 5, 6, 7 e 8, para facilitar o entendimento.

É importante destacar que(C_1) representa os *itemsets* candidatos na primeira fase, o (C_2) representa os itemsets candidatos da segunda fase e o (C_3) representa os *itemsets* candidatos da terceira fase. Assim também é definido para o suporte onde (F_1) representa o valor de suporte da primeira fase, (F_2) representa o valor de suporte da segunda fase e (F_3) representa o valor de suporte da terceira fase.

Na primeira fase, todos os *itemsets* são considerados candidatos, visto que é a primeira vez que o algoritmo faz varredura no banco de dados então é realizado somente a contagem do suporte, dando origem à tabela 2.2.

Itemsets Candidatos = C_1	Suporte =F_1
{1}	3
{2}	2
{3}	2
{4}	1
{5}	1
{6}	2
{7}	1
{8}	1

Tabela 2.2: Contagem de suporte

Após a geração dos candidatos com seus respectivos valores de suporte, é realizado a poda, considerando o valor de suporte mínimo igual a 50%, ou seja, suporte mínimo = 2; com isso os *itemsets* Suco {4}, Açúcar {5}, Biscoito {7} e Ovo{8} foram excluídos por não atingir um suporte mínimo igual a 50%.

A segunda iteração é gerada a partir do suporte (F_1) da primeira iteração, ou seja, para criar os *itemsets* candidatos (C_2) é necessário analisar o suporte (F_1). Também na segunda iteração será analisado o valor de suporte (F_2)para que os *itemsets* candidatos que não atingirem o valor de suporte mínimo sejam retirados da lista de candidatos.

Itemsets Candidados C_2	Suporte = F_2
{1}	3
{2}	2
{3}	2
{6}	2

Tabela 2.3: *Itemset* candidatos restantes após a poda

Como pode ser visto na tabela 2.3 os *itemsets* que atingiram o valor de suporte mínimo são selecionados para a próxima fase, onde a mesma consiste na combinação de todos os itens um a um, onde será feito uma nova varredura no banco de dados para calcular o suporte dos *itemsets* combinados. A tabela 2.4 representa essa combinação após a varredura do banco e os respectivos valores de suporte.

Na tabela 2.4 os *itemsets* candidatos {1,2}, {1,6}, {2,3},{2,6} e {3,6} não atenderam os valor de suporte mínimo e, sendo assim, os mesmos serão excluídos. Apenas o *itemset* {1,3} atingiu o valor de suporte mínimo estabelecido.

A propriedade *Apriori* só considera um *itemset* frequente se todos os seus subitens também forem frequentes, caso contrato o *itemset* será eliminado. Com isso pode não

Itemsets Candidados C_3	Suporte = F_3
{1,2}	1
{1,3}	2
{1,6}	1
{2,3}	1
{2,6}	1
{3,6}	0

Tabela 2.4: Combinação de *itemsets* candidatos

existir a possibilidade de criação de regras utilizando os demais *itemsets*, fazendo com que o critério de parada fosse alcançado.

Uma vez que o critério de parada foi atingido, será realizado a fase de geração das regras de associação a partir dos *itemsets* restantes, ou seja, ao valor de suporte mínimo é maior ou igual ao suporte mínimo preestabelecido.

As regras são geradas combinado os *itemsets* frequentes.

Como pode ver visto na tabela 2.4, apenas os *itemsets* frequentes {1,3} atenderam o valor de suporte mínimo, então as regras serão geradas utilizando esses dois *itemsets*, como mostra a tabela 2.5.

Itemsets Candidados C_4	Suporte = F_4
{1,3}	2
{3,1}	2

Tabela 2.5: Combinação de *itemsets* candidatos

Para geração das regras de associação, é preciso diferenciar todos os subconjuntos não vazios de cada *itemset* frequente, mas como no exemplo apenas dois *itemsets* foram selecionados, não serão gerados os conjuntos não vazios. Contudo são calculadas as confianças:

$$C(1 \rightarrow 3) = \frac{2}{3} = 0,66$$

$$C(3 \rightarrow 1) = \frac{2}{2} = 1$$

Calculando a porcentagem dos valores das confianças tem-se, $C(1 \rightarrow 3)$ igual a 66% e $C(3 \rightarrow 1)$ igual a 100%. Considerando uma confiança mínima preestabelecida igual a 75%, é identificado apenas uma regra que atende todos os requisitos.

Em virtude dos fatos mencionados pode-se afirmar que o algoritmo Apriori é suficientemente capaz de gerar regras de associação interessantes, fortes e confiáveis.

2.2 ALGORITMOS DE REGRAS DE ASSOCIAÇÃO

Com o objetivo de minimizar o tempo de processamento e tornar a busca por regras de associação menos custosa, diversos algoritmos foram criados a partir de estudos realizados após a criação do *Apriori*. Todos os algoritmos se baseiam na definição: dado um conjunto de dados ***D***, um suporte mínimo ***S*** e confiança mínima ***C***, as regras de associação existentes serão selecionadas se satisfazem aos valores mínimos de suporte e confiança, e exclusivas caso contrário. (HAN et al., 2000), (LIU, 2007).

O primeiro algoritmo criado para gerar itens frequentes conhecido como **AIS** é apresentado em (AGRAWAL et al., 1993). O algoritmo faz várias consultas no banco dados e cria os candidatos de acordo com a banco de dados de transações.

Outro algoritmo de regra de associação é o **SEAR** (*Sequential Efficient Association Rules*), apresentado em (MUELLER, 1998) é similar ao algoritmo Apriori; a única diferença é o armazenamento dos candidatos, uma vez que os candidatos são armazenados numa árvore de prefixo.

O algoritmo **SPEAR** (*Sequential Partitioning Efficient Association Rules*) é semelhante ao **SEAR** mas utiliza a ideia de particionamento da base porém difere da ideia do algoritmo ***Partition***.

Após o desenvolvimento do algoritmo Apriori surgiram algumas variações, como o **APRIORI-TID** (AGRAWAL et al., 1996) e o **APRIORI-HIBRIDO** (KOSTERS et al., 1999). O algoritmo **APRIORI-TID** possui características semelhantes ao Apriori; eles diferem apenas na parte de acesso ao banco de dados, uma vez que o **APRIORI-TID** faz apenas duas consultas no banco de dados. Por isso é considerado como uma versão otimizada do Apriori.

O **APRIORI-HIBRIDO** é a junção do Algoritmo Apriori e **APRIORI-TID**. As primeiras etapas utilizadas são do Apriori, a partir de um determinado momento ele utiliza as funções do **APRIORI-TID**.

(SAVASERE et al., 1995) apresentada o algoritmo ***Partition***, cuja proposta é ser mais eficiente e reduzir a sobrecarga de E/S. O algoritmo é executado em duas fases. Na primeira fase a base de dados é divida em pequenas partes. Na segunda fase, são gerados

todos os conjuntos de *itemsets*. Ao final da primeira fase os conjuntos de *itemsets* são misturados, para gerar os *itemsets* potencialmente frequentes. A segunda fase consiste no cálculo da frequência de todos os *itemsets* potencialmente frequentes da primeira fase, para formar um conjunto de *itemsets* frequentes.

Proposto por (BRIN et al., 1997) o **DIC** (*Dynamic itemsetCounting*) reduz o número de consultas realizadas no banco de dados os *itemsets* são contados dinamicamente e podem possuir tamanhos variados.

O algoritmo apresentado em (HAN et al., 2000) conhecido como ***FP-Growth*** (frequent-pattern growth), utiliza uma abordagem diferente dos demais algoritmos, uma vez que não é gerado a lista de candidatos. É realizado uma pesquisa na base para encontrar itens frequentes e colocá-los em uma lista. A lista é ordenada de forma decrescente de acordo com a frequência. Uma nova consulta no banco é realizada para a construção da árvore ***FP-tree***. Nesse algoritmo a mineração é realizada na árvore e não no banco de dados, ou seja, *itemsets* que não atingem o suporte mínimo são podados da árvore.

Para gerar as regras de associação que serão utilizadas neste trabalho, será utilizado o algoritmo Apriori.

3 CLASSIFICAÇÃO EM BASES TRANSACIONAIS

Bancos de dados transacionais são bases de dados que possuem diversas informações relacionadas a uma empresa, tais como: clientes, produtos, itens comprados, ID de compra, pagamento, dentre outros. Diversas manipulações também podem ser realizadas neste tipo de banco de dados como por exemplo, o cadastramento de um novo cliente, cadastramento de produtos, consultas de funcionários, etc. Em suma, base de dados transacionais representam a história de uma empresa (FAYYAD et al., 1996).

A mineração em bases de dados transacionais geralmente é feita através da obtenção de regras de associação. Quando se trata de bases transacionais rotuladas, ou seja, sujeitas a classificação, busca-se uma hipótese de indução que prediz adequadamente a classe de novas instâncias a serem avaliadas. Neste caso, não tão intenso, o uso de regras de associação para a geração de tal hipótese, aplicando-se, geralmente, técnicas padrões de aprendizagem supervisionada.

As bases de dados transacionais geralmente são utilizadas na mineração de regras de associação, ou seja, aplicações onde se deseja encontrar relações entre itens. Já a classificação consiste em analisar a base de dados a fim de encontrar padrões que descrevem o comportamento da base, ou dependências entre as instâncias para classificar de forma correta novas instâncias, ou seja, instâncias que são desconhecidas pelo classificador.

Porém, alguns estudos propõem métodos para aumentar a predição da classificação utilizando regras de associação. Esses métodos usualmente baseiam-se na mineração das regras de associação e a partir das regras, buscam a construção de classificadores eficientes (ZIMMERMANN; RAEDT, 2004).

A utilização da classificação baseada em regras de associação pode apresentar alguns benefícios. No que diz respeito à mineração das regras de associação, os benefícios estão relacionados à forma de avaliação da base, uma vez que, na mineração de regra de associação a base de dados é analisada por completo a procura de relação entre as instâncias, permitindo que todas as instâncias sejam consideradas de acordo com sua relevância para a construção das regas. No que diz respeito à classificação utilizando regras de associação, o benefício é a existência de algoritmos eficientes, para a geração das regras a serem

utilizadas na classificação das instâncias desconhecidas. Assim, tem-se a expectativa de serem desenvolvidos classificadores eficientes e precisos através de regras de associação precisas(ALVES, 2007).

3.1 BASES TRANSACIONAIS EM APRENDIZAGEM SUPERVISIONADA

A aprendizagem supervisionada implica em um tipo de aprendizado direcionado por dados relativos ao objeto, previamente conhecidos e caracterizados. Para melhor entendimento, tem-se como exemplo um médico em um determinado hospital que atende pacientes com os seguintes sintomas: dor de cabeça, febre, dor no corpo e diarréia, ou seja, sintomas que podem caracterizar casos de dengue. Já sendo de conhecimento do médico que estes sintomas são provenientes de dengue, o próximo paciente que chegar ao hospital com esses mesmos sintomas, terá um diagnóstico mais rápido e preciso. Logicamente, a qualidade do diagnóstico vai depender de quais variáveis (sintomas) o médico considerou e a quantidade de pacientes que foram avaliados pelo mesmo de forma correta(banco de dados).

Isso também acontece na aprendizagem supervisionada, onde as classes são previamente rotuladas, ou seja, é fornecido para o classificador uma base de dados onde os atributos e as classes são devidamente identificadas. Essa base serve de referência para uma fase conhecida como treinamento que possibilita a criação de um classificador que representa uma hipótese de indução com o maior nível de generalização. A partir daí, o classificador deve ser capaz de predizer com maior nível de precisão possível a classe de novas instâncias, baseando-se no conhecimento representado pela hipótese de indução obtida.

A base de dados de interesse deste trabalho é uma base de dados transacional real de compras de clientes com características de aprendizagem supervisionada, pois as classes das transações foram definidas previamente. Tais classes representam vendas realizadas nos turno da manhã e da tarde pelos clientes. Para esta base, a classificação através de regras de associação tem como objetivo avaliar os padrões de transações para cada turno.

Por se tratar de uma base de dados transacional foi utilizado o algoritmo CBA na obtenção das regras de classificação. A utilização deste algoritmo pode ser justificada por dois motivos. Primeiro por ser um algoritmo construído baseado em um algoritmo de

referência para obtenção de regras de associação Apriori e segundo por ser bem consolidado e bastante utilizado em classificação de bases de dados transacionais (YIN; HAN, 2003). O CBA (MA, 1998) é capaz de gerar as regras precisas de acordo com os valores de suporte e confiança previamente definidos, tais regras são utilizadas no processo de classificação.(ALVES, 2007).

Além do algoritmo CBA, pode-se citar outros algoritmos de classificação, também baseados em regras de associação, como, por exemplo, o ADT (FREUND; MASON, 1999), CMAR (LI et al., 2001), CPAR (YIN; HAN, 2003), GARC (CHEN et al., 2006) e CorClass (ZIMMERMANN; RAEDT, 2004).

O algoritmo CBA trabalha em duas etapas. A primeira etapa consiste na realização da extração das regras de associação, baseando-se no algoritmo Apriori (AGRAWAL et al., 1993), cujo lado direito da regra está restrito a classe, onde é criado um subconjunto de regras. Este subconjunto é chamado CAR*(Class Association Rules)*. O subconjunto deve satisfazer o suporte mínimo e confiança mínima estabelecidas previamente. A segunda etapa é baseada na construção do classificador a partir das regras geradas, onde esse classificador é uma lista de regras ordenadas seguindo uma ordem de importância (LIU et al., 2001).

O algoritmo utiliza o sistema de poda com o objetivo de reduzir o número de regras geradas e, com isso, evitar uma explosão de combinações. O sistema de poda utilizado no CBA é baseado no algoritmo C4.5, onde é utilizado a taxa de erro pessimista das regras de associação. Após a poda é realizada a classificação, onde o classificador percorre a lista de regras de associação gerada e verifica qual é a primeira regra que se encaixa com a instância que está sendo classificada, associando a ela a classe da regra selecionada (LIU et al., 2001).

Porém para que o CBA possa ser utilizado e apresente um resultado satisfatório na classificação, mostrou-se importante que se tenha um equilíbrio entre o número de dados de cada rótulo.

3.2 DESBALANCEAMENTO EM BASES DE DADOS

O rótulo de interesse da base utilizada, a saber, o turno em que as transações foram efetuadas, trouxe como característica da mesma um desbalanceamento inerente à realização das transações em cada período considerado. Testes iniciais indicaram que este desbalan-

ceamento, para este tipo de dado, é bastante influente na obtenção das regras relacionadas às transações. Basicamente, pode-se considerar que as regras não apresentam o mesmo padrão qualitativo nem quantitativo na representação de cada rótulo. O caminho natural para contornar tal comportamento passa por uma tentativa de balancear os parâmetros para a obtenção das regras ou balancear a própria base antes da obtenção das mesmas, que foi o caminho adotado.

O problema de desbalanceamento entre as classes em bases de dados tem sido considerado cada vez com mais interesse pela comunidade de mineração de dados, visando a resolução de problemas de classificação. Uma base é considerada desbalanceada quando uma classe apresenta um número muito maior de instâncias que as demais classes, influenciando negativamente na construção de classificadores que, geralmente, apresentam a tendência de uma melhor predição na classe majoritária em detrimento da predição das classes com menor número de representantes.

Se a diferença entre as classes majoritária e minoritária for crescendo, aumenta-se a dificuldade na predição da classe minoritária(CHAWLA et al., 2002), cujos dados podem ser enquadrados na categoria de casos raros que apresentam alto nível de dificuldade tanto na detecção quanto na identificação (WEISS, 2004).

Questões relativas à métrica utilizada para a geração de preditores em bases desbalanceadas também são bastante relevantes, visto que um erro na classe majoritária nem sempre tem a mesma relevância ou impacto de se predizer errado um dado pertencente a classe minoritária. A escolha adequada da medida de predição que será adotada no processo de construção da hipótese de indução, pode influenciar completamente na qualidade do resultado obtido.

Com o objetivo de reduzir o desbalanceamento entre as classes, diversos métodos tem sido propostos para realizar o balaceamento das mesmas. Tais métodos tem como principal função a redistribuição das instâncias relativas às classes que apresentam desbalaceamento significativo.

Um dos métodos utilizados para tratar o desbalaceamento entre as classes é conhecido como *Undersampling*. Este método utiliza apenas um subconjunto da classe majoritária para tratar de desbalanceamento, não utilizando as instâncias não selecionadas para compor o subconjunto. A escolha das instâncias para a criação do subconjunto geralmente é realizada de forma aleatória.

Apesar de eficiente e simples, o método *Undersampling* pode obter resultados insatisfatórios quando uma parte da classe majoritária é ignorada (LIU et al., 2009). Excluir instâncias pode afetar o desempenho do classificador, visto que instâncias importantes podem não ser escolhidas.

Para que essas instâncias possam ser ignoradas com segurança, uma análise prévia deve ser realizada, a fim de verificar se a exclusão dessas instâncias pode afetar de forma significativa o desempenho do classificador. Essa análise não é realizada no método padrão de *Undersampling*.

O *Oversampling* também é um método desenvolvido para tratar o problema de desbalanceamento. Tal método funciona de forma contrária ao método *Undersampling*, reorganizando a classe minoritária. O *Oversampling* tem como principal função replicar os dados da classe minoritária através de escolha randômica com reposição para diminuir o desbalanceamento entre as classes. A replicação dos dados da classe minoritária também pode ser pouco efetiva devido a sobre-amostragem de dados, com a possibilidade de tornar o processo de construção do classificador custoso computacionalmente se a base de dados for relativamente grande (QAZI; RAZA, 2012).

Um dos métodos mais utilizados visando o balanceamento de bases de dados, é o SMOTE *(Synthetic Minority Over-sampling TEchnique)* apresentado em (CHAWLA et al., 2002). Como o próprio nome já diz, é uma técnica que cria dados sintéticos a partir dos dados existentes da classe minoritária e não adotando a sobre-amostragem por substituição de dados. Os dados sintéticos da classe minoritária são criados a partir dos vizinhos mais próximos com o objetivo de aumentar o espaço de decisão da classe minoritária. É preciso ter cuidado na criação dos dados sintéticos para não desconfigurar a distribuição original dos dados, o que geraria uma distorção na hipótese de classificação obtida com grande prejuízo na qualidade das predições.

As técnicas apresentadas, podem ser consideradas bem representativas de modelos para balanceamento. Apesar de serem bastantes utilizadas quando se trata de diminuir os efeitos de desbalanceamento entre classes, não apresentam garantia de bom funcionamento para todos os casos devido às estratégias adotadas em seus processos construtivos. Excluir, replicar ou até mesmo gerar dados sintéticos podem resolver o problema de desequilíbrio entre as classes, mas podem gerar variações e distorções prejudicais na distribuição original das instâncias de cada classe. Desta forma, é difícil identificar até onde é benéfico a

utilização destes modelos.

No caso do uso de regras de associação para viabilizar um processo de classificação, problemas adicionais podem se apresentar quando tais modelos de balanceamento são adotados. Desequilíbrio no cálculo das medidas de suporte e confiança geralmente utilizadas, podem comprometer a qualidade das regras geradas. Além disso, a representatividade das características ou itens em bases transacionais não apresenta a mesma uniformidade das bases de dados padrão, onde toda instância apresenta valores para todos os atributos. Assim, os procedimentos descritos para balanceamento podem distorcer bastante os padrões de itens ativos da base original.

A seguir, apresenta-se um modelo para o trato do desbalanceamento quando se tem bases transacionais sujeitas a um processo de classificação construído através de regras de classificação. Pretende-se que a estratégia desenvolvida apresente uma menor variação na representação dos dados originais devido ao balanceamento obtendo, assim, resultados mais robustos e confiáveis.

4 UM MODELO PARA A CLASSIFICAÇÃO EM BASES DE DADOS TRANSACIONAIS

Este capítulo apresenta um modelo para a classificação em bases de dados transacionais baseado em comitê de classificadores para tratar o desbalanceamento da base.

A técnica a ser descrita apresenta uma estratégia de balanceamento utilizando de forma bem específica os dados tanto da classe minoritária quanto da classe majoritária. Baseia-se na construção de classificadores base, balanceados, por meio do uso mais completo possível dos dados da classe minoritária e, evitando a exclusão de dados da classe majoritária, construindo os classificadores base por meio de um procedimento randômico. Estes classificadores base serão a referência para a construção do procedimento de predição a ser adotado. A técnica a ser apresentada evita que alguns parâmetros sejam ajustados, tais como, suporte e confiança, uma vez que só serão feitas manipulações na base de dados original.

Como a estratégia para balanceamento a ser apresentada baseia-se em um modelo de predição conhecido como comitê de classificadores, apresenta-se a seguir as principais referências da área. Ressalta-se que a ideia de comitê não é adotada usualmente para tratar desbalanceamento de bases, mas sim para viabilizar o aumento do nível de predição nas instâncias da classe minoritária pelo uso adequado de classificadores base que compõem o comitê. Sendo portanto, independente das características e distribuição das instâncias nas bases em que serão aplicados.

4.1 COMITÊ DE CLASSIFICADORES

Para tornar clara a explicação do que vem a ser um comitê de classificadores, tem-se como exemplo um diagnóstico médico onde um especialista, por si só não consegue determinar as causas de uma determinada doença por algum motivo, necessitando do auxilio de mais especialistas, com opiniões diferentes para ajudar na avaliação do paciente, ou seja, o especialista prefere ter várias opiniões sobre a situação do paciente, apresentando um diagnóstico final através de algum tipo de combinação das avaliações individuais.

No comitê de classificadores, são gerados vários classificadores conhecidos como classi-

ficadores base, que irão participar da construção da hipótese de indução através de alguma combinação predeterminada de seus resultados.

Existem vários métodos utilizados para combinar os resultados dos classificadores base, ou seja, que compõem um comitê de classificadores, dentre eles estão os métodos mais adequados a procedimentos de regressão como: soma, soma ponderada, média, média ponderada, mediana, produto, dentre outros. Entre os modelos de combinação mais adotados em problemas de classificação destacam-se os métodos baseados em votação: votação majoritária, votação majoritária ponderada, dentre outros. Neste trabalho, apenas a votação majoritária foi utilizada para combinar os resultados dos classificadores base.

A votação majoritária consiste na escolha da classe para a instância avaliada como sendo a classe que obtiver maior quantidade de votos em relação aos classificadores base. Assim, quando uma instância desconhecida precisa ser classificada, os resultados dos classificadores base são determinados para esta instância e a classe que obtiver maior quantidade de votos será a escolhida como a classe prevista da classificação.

Dentre as principais técnicas de criação de comitê de classificadores, estão o *boosting* e o *bagging*, que serão explicadas na próxima seção.

4.2 *BOOSTING*

O *Boosting* é uma técnica de aprendizagem de máquina apresentada em (SCHAPIRE, 1990) utilizando uma estratégia construtiva que visa transformar um classificador considerado fraco em termos de nível de predição em um classificador que possa ser definido como forte, ou seja, classificadores com menor poder de classificação transformados em um classificador com maior poder de classificação. Seu funcionamento consiste em atribuir pesos para as instâncias do conjunto de treinamento e, ao final de cada classificador base gerado, verificar o erro na predição para que na etapa seguinte as instâncias com maior erro recebam mais peso (OHNO, 2011). Por fim, um modelo de votação combina os classificadores para gerar o resultado final. Pode-se dizer que o *Boosting* é basicamente um método que tem sua estratégia de aprendizagem direcionada pelos erros na classificação das instâncias.

A partir do desenvolvimento do *Boosting*, diversos algoritmos foram criados, sendo o mais conhecido. O algoritmo de *Boosting* apresentado por (FREUND; SCHAPIRE, 1997) chamado de *AdaBoost (Adaptative Boosting)*. O funcionamento do *AdaBoost (Adaptative*

Boosting) consiste na escolha de um classificador de referência, ou seja, o classificador que será utilizado para realizar a classificação das instâncias. A distribuição de pesos é atualizada a cada iteração do *Adaboost* e indica o nível de dificuldade na classificação do conjunto de treinamento. O peso é aumentando para as instâncias que foram classificados incorretamente, e diminuído para exemplos que foram classificados corretamente. Por fim é realizado a combinação com os resultados dos classificadores base gerados. Essa combinação pode ser realizada de diversas formas, mas geralmente é utilizada a votação majoritária.

4.3 *BAGGING*

O *Bagging* foi o primeiro algoritmo para a construção de EBS *(Ensemble Based Systems)*, foi criado por (BREIMAN, 1996) e é um dos métodos mais simples de comitê de classificadores.

No *Bagging* o conjunto de dados de treinamento é referência para a construção de um número predefinido de amostragens com a mesma quantidade de elementos do conjunto de treinamento, selecionadas de forma aleatória com reposição. Desta forma, sendo baseadas em escolha aleatória com reposição, algumas instâncias podem se repetir em uma mesma amostra, enquanto outras são selecionadas para algumas amostras. Cada amostra ou subconjunto de treinamento criado gera um classificador base. Cada subconjunto, como dito, tem o mesmo número de instâncias da base original.

Assim como no *Boosting*, o resultado da classificação é a combinação das saídas dos classificadores base gerados por votação majoritária, ou seja, para uma determinada instância, a classe que receber o maior número de votos dos classificadores base, será a classe adotada.

O *Bagging* tem como objetivo melhorar a estabilidade e precisão na classificação (OHNO, 2011).

4.4 UM MODELO DE COMITÊ DE CLASSIFICADORES PARA BASES TRANSACIONAIS

A bases de dados de interesse para a avaliação de predição e utilizada para o desenvolvimento deste trabalho é uma base transacional que possui registros de compras de cliente

em um supermercado. As classes correspondem ao turno em que os produtos foram adquiridos, ou seja, turno da manhã e da tarde. As instâncias das classes referentes ao período da tarde superam numericamente as instâncias da classe referente ao turno da manhã, o que gera um desbalanceamento entre classes.

Como visto na seção 3.2, o desbalanceamento de uma base é caracterizado quando a quantidade de instâncias de uma classe é relativamente maior que da outra classe. Com o intuito de tratar esse desbalanceamento em uma base de dados transacional, o modelo proposto tem por objetivo buscar um balanceamento a fim de melhorar a predição da classificação, uma vez que a utilização da base desbalanceada pode causar distorções nas predições obtidas para as classes consideradas.

Nos problemas com essas particularidades as classes minoritárias tendem a ter sua classificação dificultada pelo aumento de falsos positivos da classe majoritária.

A motivação para o desenvolvimento deste modelo se dá pelo fato de que os métodos mais conhecidos para tratar desbalanceamentos apresentador anteriormente foram aplicados neste tipo de base de dados, mas não apresentaram resultados satisfatórios, conforme pode ser visto nos experimentos numéricos.

A estratégia de balanceamento apresentada neste trabalho consiste na utilização do maior número possível das instâncias da classe minoritária e na divisão da classe majoritária em parcelas que utilizem todas as instâncias pelo menos uma vez para a composição dos subconjuntos de dados que irão gerar os classificadores base de um comitê.

Esta estratégia, além de balancear os subconjuntos de treinamento dos classificadores base, utiliza as instâncias da classe minoritária com uma maior intensidade, visto que participam geralmente em sua totalidade de todos os subconjuntos que compõem o comitê. A expectativa é um maior foco na predição desta classe, visto que as instâncias escolhidas para os subconjuntos da classe majoritária apresentam maior variabilidade devido ao seu maior número na base de dados original e a opção por utilizar todas as instâncias pelo menos em um subconjunto base.

Utilizando a classe minoritária, geralmente formada pelas mesmas instâncias em toda as bases e uma parcela da classe majoritária na construção de cada base, garante-se uma diferenciação entre os subconjuntos base. Isso possibilita a formação de um conjunto de classificadores que definirão o comitê de classificadores para bases desbalanceadas, podendo obter resultados mais eficientes para estas bases, devido à combinação de classi-

ficadores individuais e a forma específica de construção dos subconjuntos base.

A seguir é apresentado, em detalhe, o método de balanceamento de bases transacionais, desenvolvido que foi adaptado de um trabalho onde é aplicado em problemas de previsão de insolvência (HORTA, 2010).

Dado um conjunto de treinamento desbalanceado B_{td}, considerando que seja formado por duas classes, sem perda de generalidade. Os dados de B_{td} podem ser separados em dois conjuntos distintos, a saber:

- $B_{t_m} \rightarrow$ contendo as instâncias da classe minoritária;
- $B_{t_M} \rightarrow$ que recebe as instâncias da classe majoritária.

onde o número de instâncias da classe majoritária é maior ou bem maior do que as instâncias da classe minoritária, ou seja:

$$\#(B_{t_M}) > \#(B_{t_m}) \quad \text{ou} \quad \#(B_{t_M}) \gg \#(B_{t_m}) \tag{4.1}$$

com $\#(*)$ sendo a cardinalidade do conjunto em questão. O método desenvolvido para a geração do comitê de classificadores define, inicialmente, a estratégia para gerar cada um dos conjuntos de treinamento base (B_{tb}) que gerarão os classificadores que irão compor o comitê. Parte-se da premissa que esses conjuntos de treinamento base devem ser balanceados. Desta forma, deve-se definir o número de instâncias de cada classe que irão compor cada base (n_{ic}). Este valor deve ter como referência para sua determinação três parâmetros:

- $\#(B_{t_M})$;
- $\#(B_{t_m})$;
- número de classificadores base do comitê (n_{cb}).

A variável n_{cb} é um parâmetro do algoritmo que deve ser previamente definido. A estratégia adotada para o cálculo do n_{ic} dos classificadores base é definida na forma (HORTA, 2010):

$$n_{ic} = \max(\#(B_{t_m}), \#(B_{t_M})/n_{cb}) \tag{4.2}$$

com $\max(*,*)$ sendo o operador que assume o máximo valor entre os argumentos. Fica claro, por esta definição, que existe uma correlação intrínseca entre os valores de n_{ic} e n_{cb}, ou seja, o parâmetro n_{cb} deve ser adotado de forma que, pelas dimensões da base de dados em relação as classes majoritária e minoritária, o número de instâncias por classe (n_{ic}) apresente um tamanho considerado razoável. Deve-se ressaltar, também, que todos os conjuntos base do comitê (B_{tb}) tem em sua composição todas as instâncias da classe minoritária.

Definido o valor de n_{ic}, a construção dos n_{cb} conjuntos de treinamento base é feita de acordo com:

- Classe minoritária:
  - copia-se todas as instâncias da classe minoritária para o conjunto base;
  - se necessário, completa-se as instâncias faltantes com a escolha randômica das instâncias desta classe, permitindo a reposição.
- Classe majoritária:
  - copia-se todos os dados da classe majoritária da base original de treinamento (B_{td}), em seqüencia, para gerar todos os n_{cb} conjuntos base B_{tb};
  - se necessário, completa-se as instâncias faltantes do último conjunto base com a escolha rândômica das instâncias desta classe, permitindo a reposição.

Caso sobrem instâncias na classe majoritária, as mesmas podem ser distribuídas pelos conjunto de dados base ou acopladas ao último conjunto base gerado.

Para melhor entendimento apresenta-se como exemplo uma base de dados com duas classes A e B, com respectivamente 650 e 100 instâncias. Adotando-se n_{cb} com o valor de 7, deve-se gerar os 7 conjuntos base que formarão o comitê. A seguir, calcula-se o valor de n_{ic} baseando-se na equação (4.2), obtendo o valor de 100. Assim, cada conjunto base será formado por 200 instâncias, número considerado razoável. Desta forma, os conjuntos base terão todas as instâncias da classe minoritária em sua composição, como destacado. No caso das instâncias da classe majoritária, os 6 primeiros conjuntos base de treinamento serão compostos por instâncias distintas. O último conjunto terá em sua composição as 50 instâncias distintas ainda não utilizadas da classe majoritária, sendo complementado com outras 50 instâncias escolhidas de forma aleatória com reposição do total das 650

instâncias da classe. Este procedimento é facilmente adaptável para problemas multi-classes. A seguir, apresenta-se na figura 4.1 o pseudo-código do comitê de classificadores para dados desbalanceados (CCDD).

Algoritmo CCDD
Início
Obtenha a base de treinamento desbalanceada B_{td}
Defina n_{cb}
Separe B_{td} em B_{t_m} e B_{t_M}
Calcule $n_{ic} = \max(\#(B_{t_m}), \#(B_{t_M})/n_{cb})$
% construção dos n_{cb} classificadores base B_{tb}
Para $i = 1, n_{cb}$ **faça**
% dados da classe minoritária do $i-$ésimo B_{tb}
$B^i_{tb} \leftarrow B_{t_m}$
% se necessário completar com escolha aleatória com reposição
Para $j = \#(B_{t_m} + 1), n_{ic}$ **faça**
$B^i_{tb} \leftarrow B^i_{tb} \cup B_{t_m}(rand(1, \#(B_{t_m}))$
Fim para
% dados da classe majoritária do $i-$ésimo B_{tb}
Para $j = 1, n_{ic}$ **faça**
$B^i_{tb} \leftarrow B^i_{tb} \cup B_{t_M}((i-1)n_{ic} + j)$
Fim para
Fim para
% se necessário completar dados do último B_{tb} do comitê
Se $\#(B_{t_M}) < n_{cb}\, n_{ci}$ **então**
Para $j = \#(B_{t_M}) + 1, n_{cb}\, n_{ic}$ **faça**
$B^{n_{cb}}_{tb} \leftarrow B^{n_{cb}}_{tb} \cup B_{t_M}(rand(1, \#(B_{t_M})))$
Fim para
Fim se
Treine os n_{cb} classificadores base
%classificação de novas instâncias
Aplique votação majoritária para classificar os dados da base de teste
Fim

Figura 4.1: Pseudo-código do comitê de classificadores para dados desbalanceados

4.5 CONSIDERAÇÕES

A ideia utilizada na construção de bases distintas para a criação do conjunto de classificadores base que compõem o comitê só é viável quando a base de dados apresenta um desbalanceamento entre as instâncias da classe.

Quanto maior a quantidade de classificadores base para a geração de um comitê, melhor será o resultado da classificação final, uma vez que o resultado de diversos classificadores em conjunto tende a ser melhor que um classificador individual se o processo construtivo for adequado.

Um outro ponto importante que deve ser destacado quando se trata do uso de classificadores baseados em regras de associação, é a falta de necessidade da adoção de parâmetros de suporte e confiança específicos para cada classe, uma vez que serão feitas apenas manipulações na base de base dados, sem perda de dados ou informações importantes. Ressalta-se que algumas técnicas específicas para bases transacionais desbalanceadas são totalmente construídas baseadas nestes ajustes.

Outro ponto que deve ser destacado da base é a relação da razão do desbalanceamento e o número de classificadores base (n_{cb}) utilizados. Quanto maior o desbalanceamento, tende-se a usar mais classificadores base na construção do comitê. Desta forma, tem-se um comitê composto por mais classificadores, o que diminui o efeito do desbalanceamento por meio do aumento da robustez gerada pelo incremento na dimensão do comitê.

A estratégia baseada em comitê de classificadores para fazer o balanceamento da base de dados transacionais, traz a expectativa de obtenção de regras mais relevantes e discriminativas para a classe majoritária, mas principalmente para as possíveis classes minoritárias.

Procedimentos de seleção de características padrões poderiam ser utilizados para tais bases transacionais. Porém, regras de associação mais efetivas e confiáveis trazem a perspectiva de serem utilizadas como referência para a construção de um método de seleção de características baseado em regras obtidas previamente. Modelos de seleção de características tendo regras de associação como base construtiva são raramente encontrados na literatura.

No capítulo seguinte, esta ideia será desenvolvida, visando apresentar um modelo de seleção baseado em regras de associação representativas que apresente resultados relevantes em relação a métodos transacionais de seleção de característica e que tenha as vantagens inerentes para bases transacionais.

5 UMA ESTRATÉGIA DE SELEÇÃO DE CARACTERÍSTICAS BASEADA EM PADRÕES

Neste capítulo é apresentada uma estratégia de seleção de características baseada em padrões minerados, a saber, regras de associação obtidas em um processo de aprendizagem supervisionada, ou seja, as características serão selecionadas de acordo com as regras. A expectativa é avaliar a técnica a ser apresentada neste capítulo em relação técnicas mais tradicionais de seleção, visto não ser muito usual um modelo baseado em regras de associação.

A motivação para selecionar características através da classificação de padrões minerados dá-se pelo fato de que, como são apresentadas regras após a classificação, é viável que se encontre características relevantes nas associações obtidas. Por exemplo, regras com altos valores de suporte e confiança são mais representativas do que regras com valores baixos de suporte e confiança, então as características que compõem tais regras podem ser consideradas como tendo níveis adequados de relevância.

Para se determinar uma estratégia de como as características seriam selecionadas, diversos experimentos foram realizados utilizando suporte e confiança como indicadores para avaliar as características por meio das regras obtidas. Porém ao final, não apresentaram resultados satisfatórios com a seleção adotada. Experimentos utilizando múltiplos valores de suporte para a obtenção das regras em bases desbalanceadas foram realizados, porém sem sucesso.

Análises comparativas com métodos tradicionais de seleção de características para avaliar a eficiência e a competitividade de um método baseado em regras são cruciais. Deseja-se que o método tenha um bom desempenho quando comparado com os métodos tradicionais de seleção de características.

A seguir serão apresentados alguns métodos mais comuns utilizados para realizar a tarefa de seleção de características.

5.1 SELEÇÃO DE CARACTERÍSTICA EM APRENDIZAGEM SUPERVISIONADA

Modelos de seleção de características vem sendo desenvolvidos com bastante interesse por pesquisadores da área de aprendizagem de máquina, principalmente, pela crescente demanda gerada por bases de dados com número bastante alto de atributos e número de instâncias limitado. Estas bases tem se apresentado em áreas como Biologia, Economia, entre outras.

Tal técnica, tem como principais funções reduzir a dimensionalidade da base, eliminando características redundantes, irrelevantes, inúteis ou ruidosas, que podem prejudicar o processo de classificação (RAJESWARI, 2015), (LING; HUI, 2014).

Em algumas aplicações que demandam manipulação em bases de dados, como por exemplo, em mineração de dados, a seleção de características é considerada de grande importância, visto que tais técnicas podem reduzir a dimensionalidade da base de dados sem comprometer a qualidade da informação.

Dois fatores podem ser considerados motivadores para que a seleção de características desperte a atenção de pesquisadores da área de aprendizagem supervisionada, sendo ambos relacionados ao desempenho dos algoritmos. Os fatores são: alguns algoritmos de aprendizagem tendem a ter seu desempenho afetado devido ao fato de existirem características irrelevantes e redundantes na base de dados; e a redução da dimensionalidade da base de dados, diminui a dimensão do espaço de versões, deixando o problema teoricamente mais simples e, consequentemente, computacionalmente mais viável em relação ao desempenho de alguns algoritmos de aprendizagem de máquina (XIE et al., 2009). Logicamente, estas vantagens estão intrinsecamente relacionadas à qualidade de seleção realizada. Uma seleção mal feita compromete o processo, inclusive podendo piorar a qualidade da predição.

A todo momento nós, seres humanos, realizamos seleção de características. Tem-se, como exemplo, um processo seletivo de uma determinada empresa visando contratar funcionários para determinados cargos. Para tanto, os candidatos a uma vaga devem possuir graduação completa, um histórico escolar relativamente bom, entre outros atributos.

Todavia, nem todos os candidatos possuem tais características para concorrer à vaga da empresa e, neste caso, alguns candidatos irão se destacar em relação aos outros. O

mesmo acontece com as bases de dados, onde algumas características são fundamentais para discriminar determinados rótulos associados aos dados.

Para verificar quais características são mais importantes em um processo de classificação, modelos avaliam a base de dados à procura dessas características. Alguns modelos são agregados a métodos específicos de aprendizagem de máquina e são conhecidos como métodos encapsulados ou *wrappers*, outros são associados aos algoritmos de busca de padrões intrínsecos das características e são conhecidos como métodos de filtro.

(DASH; LIU, 1997) consideram quatro etapas básicas e essenciais para realizar o processo de seleção de características. As etapas consistem em um procedimento para geração de candidatos, para que o próximo candidato do subconjunto possa ser avaliado; a função de avaliação, cujo objetivo é avaliar o conjunto de características gerado; uma condição de parada da execução, para que o processo não entre em *loop* infinito; e um processo de validação, para verificar se o subconjunto de características gerado é um subconjunto válido.

A figura 5.1 a seguir, ilustra os quatro passos para executar o processo de seleção de características.

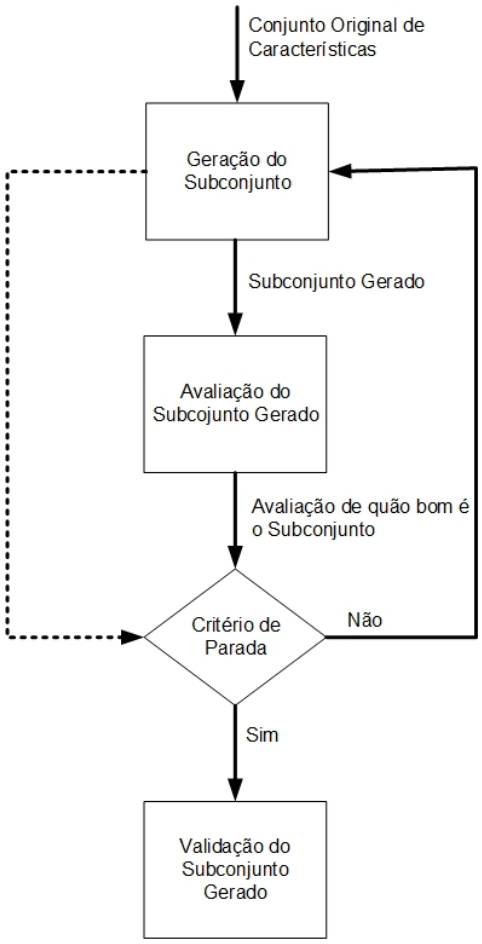

Figura 5.1: Processo de seleção de características, adaptado de (DASH; LIU, 1997)

.

A figura 5.1 representa os quatro passos mencionados anteriormente. Primeiramente, é realizada uma busca na base de dados utilizando parâmetros pré-definidos para gerar o subconjunto de características.

A fase de geração de características pode iniciar de três formas: com todas as características iniciais selecionadas, conhecida como busca para trás ou *backward selection*; sem nenhuma característica selecionada, conhecida também como busca para frente ou *forward selection*; ou com um subconjunto aleatório de características.

Uma função de avaliação é definida para avaliar o subconjunto gerado e faz comparações entre os valores dos subconjuntos gerados posteriormente. Se o valor do subconjunto for melhor que o valor do subconjunto anterior, ele será substituído pelo subconjunto que apresentar melhor valor. Isso acontece até que a critério de parada seja satisfeito e a execução findada.

Como informado anteriormente, alguns métodos são utilizados para realizar o pro-

cesso de busca, sendo encapsulados ou em filtro. A principal diferença entre os modelos de seleção de características é a utilização de um algoritmo de classificação no modelo encapsulado para direcionar a seleção.

A figura 5.2 a seguir, ilustra o modelo de seleção de características por filtro.

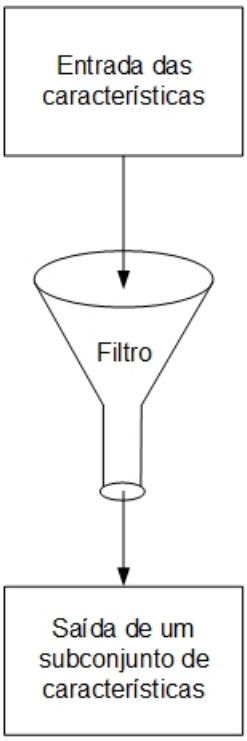

Figura 5.2: Modelo de seleção por filtro (FREITAS, 2013), p. 67

.

A seleção por filtro, como o próprio nome diz, tem por objetivo filtrar as características consideradas relevantes, independente da utilização de um algoritmo de aprendizado supervisionado para avaliar as possíveis seleções. A seleção por filtro visa manter, o maior número possível de informações relevantes detectadas ao longo de todo o conjunto de dados, uma vez que é realizada a separação das características importantes das características irrelevantes, seguindo critérios predefinidos. Este critérios envolvem, usualmente, medidas intrínsecas entre as características que compõem a base, sendo realizado antes do processo de aprendizagem. Assim, a seleção por filtro é considerada um pré-processamento visando tornar a base de dados mais adequada para o processo de aprendizagem (JOHN et al., 1994)(FREITAS, 2013).

Em contrapartida, o método de encapsulamento ou *wrapper* é executado levando-se em consideração um algoritmo de classificação que avalia as características selecionadas e, através de um processo de otimização, busca o subconjunto ótimo de atributos para

o classificador considerado. Nesta abordagem o algoritmo de classificação é executado várias vezes com diferentes subconjuntos de dados e seu desempenho é utilizado para avaliar quão bom é o subconjunto de características selecionadas. A figura 5.3 ilustra esse processo.

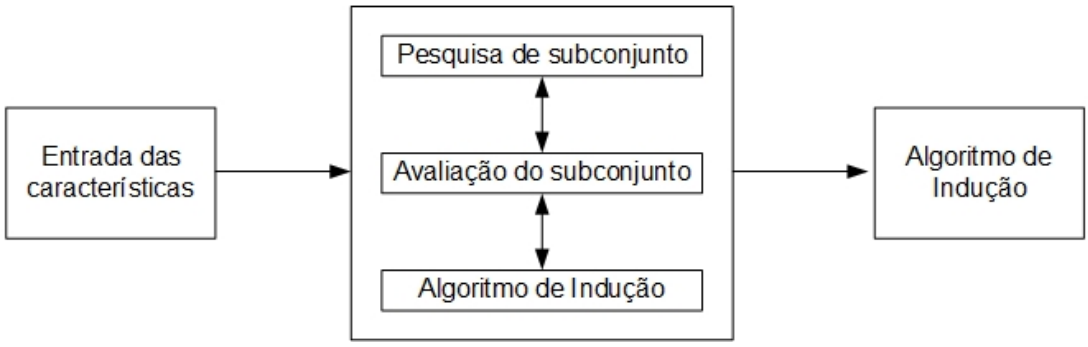

Figura 5.3: Modelo de seleção pelo método de encapsulamento ou *wrapper* (JOHN et al., 1994) p. 124

.

É importante destacar as vantagens e desvantagens dos métodos de encapsulamento e o método de filtro para selecionar o subconjunto de características. Enquanto o método encapsulado seleciona um subconjunto de características para otimizar determinado algoritmo de classificação, o método de filtro realiza a seleção sem a utilização de um classificador, ou seja, a seleção de características é independente do classificador, podendo ser aplicado a qualquer tipo de classificador utilizando as características selecionadas pelo método. Utilizando o método encapsulado, para um determinado classificador, o nível de precisão desse classificador tende a aumentar, uma vez que a seleção foi otimizada para este classificador (FREITAS, 2013). Em relação ao custo computacional, modelos encapsulados são mais custosos computacionalmente visto que necessitam treinar e avaliar classificadores para os diversos subconjuntos a serem avaliados no decorrer da otimização.

A seguir será apresentado um método de seleção de característica construído com base em regras de associação geradas por um procedimento de classificação aplicado na base original.

Alguns métodos de seleção de características, também baseado em regras, propostos na literatura também serão descritos, visando se ter uma referência dos modelos existentes nesta linha.

5.2 UM MODELO DE SELEÇÃO DE CARACTERÍSTICAS

O modelo de seleção de características proposto tem como objetivo selecionar as características a partir da avaliação de regras de associação obtidas por um procedimento de classificação que, logicamente, deve ser capaz de gerar tais regras.

Antes de propor um modelo de seleção de características baseados em regra de associação realizou-se uma prospecção visando detectar e avaliar métodos de seleção que seguem esta estratégia construtiva.

Alguns trabalhos que se embasam nesta ideia são encontrados na literatura. Entende-se que o reduzido número é um indicativo de que esta estratégia não está bem estabelecida como um método se seleção de características.

Em (LING; HUI, 2014) a seleção de características é baseada em análises de agrupamento e regras de associação. Neste trabalho, é proposto uma metodologia para seleção de funcionalidades em um sistema de sensores. No método apresentado, todos os atributos são agrupados em algum atributo classe, obtido utilizando um método de análise de agrupamento. Cada classe é examinada de acordo com o resultado da mineração de regra de associação, com o intuito de encontrar os atributos mais relevantes em um determinado conjunto de dados. São analisadas apenas regras que atingem um suporte igual a 0,8 e confiança igual a 1. As regras de associação são classificadas de acordo com o nível de importância definido como:

$$importância(X \Rightarrow Y) = \left(\frac{f_{confiança}(Y \Rightarrow X)}{f_{suporte}(X)}\right) = \log\left(\frac{p(X/Y)}{P(X)}\right) \tag{5.1}$$

As regras com nível de importância inferior a 0,8 serão excluídas, e os atributos contidos em tais regras, não serão selecionados.

Em (XIE et al., 2009) é apresentado um algoritmo para realizar a seleção de características baseado na avaliação de regras de associação que estão estritamente ligadas a uma determinada classe. O algoritmo é composto por três etapas. A primeira corresponde à geração das regras de associação, sendo geradas utilizando o algoritmo *Apriori* (AGRAWAL et al., 1993). A segunda etapa consiste na seleção das características de acordo com as classes em que as regras pertencem. Finalizando, a terceira etapa determina a criação de um conjunto de teste para validar a seleção de característica proposta pelo modelo.

Em (CHAWLA, 2010) também é realizado a seleção de características através de regras

de associação. Para gerar tais regras são utilizadas ARN(*Association Rules Networks*). Primeiramente, são obtidas as regras de associação. Um parâmetro é utilizando neste trabalho para selecionar as regras mais representativas. Este parâmetro é chamado *top-k*, o mesmo é equivalente a confiança, e não precisa ser fixado. As vantagens apontadas no trabalho é a não especificação de valores de suporte e confiança, e a utilização do parâmetro *top-k* para encontrar regras em níveis elevados. Após a geração das regras, é aplicado um algoritmo de agrupamento nessas regras que permite selecionar as características mais representativas contidas nas mesmas.

Todos estes trabalhos são avaliados e apresentam resultados razoáveis em relação à seleção realizada. Apresenta-se, a seguir, as diretrizes construtivas para o método de seleção baseado em regras de associação desenvolvido. Inicialmente destaca-se que o modelo de seleção de características proposto funciona basicamente em quatro etapas, a saber:

- Em uma etapa inicial, regras de associação para a base original são obtidas utilizando um algoritmo que gere regras como, por exemplo, o CBA juntamente com o *Apriori*. Deve-se ressaltar que, se a base de dados apresentar-se desbalanceada, primeiramente deve balanceá-la para que as regras obtidas sejam relevantes em relação as classes envolvidas.

- A seguir as regras obtidas são ordenadas de acordo com sua importância para cada uma das classes. Se não ocorrer a geração de regras, ou for geradas uma quantidade considerada pequena para uma das classes envolvidas, deve-se refazer a etapa anterior alterando valores de suporte e confiança do algoritmo *Apriori*. Visto que as regras são avaliadas por classe, é viável que se adote valores de suporte e confiança diferentes para cada classe, visando aprimorar as regras obtidas. As regras que possuem elevados valores de suporte e confiança são mais representativas, por esse motivo são alocadas primeiramente. É importante que se tenha um número razoável de regras para cada classe.

- Após a ordenação das regras por classe, é selecionada a mesma quantidade de regras para as classes. Este número de regras por classe n_{rc} é um parâmetro do algoritmo.

- Finalizando, após a seleção das características, é realizada através da determinação dos atributos que compõem as primeiras n_{rc} regras das classes, é aplicado qualquer

algoritmo de aprendizado supervisionado.

Para melhor entendimento tem-se, como exemplo, duas classes A e B, onde após a realização da classificação, obteve-se como resultado 12 regras para a classe A e 50 para a classe B. Considerando o número do n_{rc} igual a 10, serão selecionadas as características contidas nas 10 primeiras regras da classe A e as características contidas nas 10 primeiras regras da classe B. Cada característica é selecionada apenas uma vez em todas as classes.

O que difere o desenvolvido neste trabalho para os métodos propostos em (LING; HUI, 2014), (XIE et al., 2009) e (CHAWLA, 2010), é a utilização da informação relativa às classes às quais as regras de associação pertencem. A seleção realizada pelo método apresentado, leva em consideração, de uma forma considerada efetiva, as classes de cada regra de associação obtida. Desta forma, é possível determinar as características selecionadas de acordo com sua relevância para cada classe especificamente.

A seguir, apresenta-se o pseudo-código do algoritmo de seleção de características baseado em regras (SCBR):

Algoritmo SCBR
Início
Defina n_{rc}, n_c % número de regras consideradas e número de classes
Declare C_{rc}^i, $i = 1, n_c$ % conjuntos para receber regras por classe
Declare C_{cs} % conjunto para receber as características selecionadas
Repita
Defina parâmetros do classificador CBA (suporte e confiança)
Gere os conjuntos de regras de associação via CBA ou comitê de CBA
% Separe os conjuntos de regras das n_c classe
Para $i = 1, n_c$ **faça**
$C_{rc}^i \leftarrow$ regras da classe i
Fim para
Até $\#(C_{rc}^i) \geq n_{rc}$, $\forall i$
Para $i = 1, n_c$ **faça**
Ordene C_{rc}^i por importância das regras
Para $j = 1, n_{rc}$ **faça**
$C_{cs} \leftarrow C_{cs} \cup$ características das regras em $C_{rc}^i(j)$
Fim para
Fim para
Utilize os atributos em C_{cs} para classificação
Fim

Figura 5.4: Pseudo-código do método de seleção baseado em regras

É interessante destacar que o processo de seleção de característica proposto tem estratégia construtiva que combina elementos dos modelos de seleção encapsulado e em filtro.

Primeiramente é utilizado um algoritmo de classificação em regras de associação para gerar as regras, porém sem uma estratégia de otimização como se faz em modelos encapsulados. A seguir, é realizada uma filtragem nos atributos que compõem essas regras, porém sem utilizar as métricas usuais dos modelos em filtro.

5.3 CONSIDERAÇÕES

Para chegar em um método de seleção de características baseado em regras de associação obtidas de um procedimento de classificação, diversos experimentos foram realizados com diferentes algoritmos de classificação. A maior dificuldade encontrada durante o desenvolvimento deste modelo foi a definição de valores de suporte e confiança, uma vez que se o valor de suporte fosse muito alto, algumas regras poderiam ser eliminadas, e se o valor de suporte fosse muito baixo, regras muito específicas poderiam ser apresentadas.

O método de seleção de características proposto traz como benefício em relação a classificação a possibilidade de utilização de qualquer algoritmo de aprendizado após finalizada a etapa de seleção, uma vez que as características não foram obtidas por meio da otimização em relação a algum algoritmo específico.

Apesar das dificuldades encontradas durante o processo de desenvolvimento do modelo, tem-se a expectativa que o mesmo seja competitivo e eficiente nos experimentos que serão realizados em relação a técnicas já estabelecidas de seleção de características.

Antes de sua aplicação em bases transacionais, optou-se por testar o algoritmo SCBR em um problema de grande interesse atual e que permite uma avaliação mais adequada do nível da seleção efetuada, visto que se conhece a solução dos problemas sintéticos utilizados. Trata-se da identificação de polimorfismos de base única (SNP's) em marcadores genéticos que sejam associados a um determinado fenótipo (OLIVEIRA, 2015).

Serão utilizadas duas bases de dados geradas por um pacote escrito na linguagem R, construídos com graus distintos de dificuldade no que tange a obtenção dos SNP's ou atributos responsáveis pela variação do fenótipo de interesse.

Como dito, a grande vantagem destes experimentos, com uso de dados sintéticos, está no conhecimento prévio do subconjunto de característica ótimas que deve ser encontrado,

permitindo, assim, uma real avaliação do modelo testado. Deve-se ressaltar que trata-se de um problema de difícil solução onde, dependendo da complexidade das iterações entre os chamados SNP's causais que compõem a base de dados, dificilmente consegue-se identificar o subconjunto ótimo de atributos (OLIVEIRA, 2015). A seguir, apresenta-se uma descrição das bases utilizadas, complementada com alguns conceitos relativos à identificação de SNP's, finalizando com a aplicação do SCBR nestes dados para avaliar seu potencial.

5.3.1 DESCRIÇÃO DAS BASES DE MARCADORES DO TIPO SNP

Os dois conjuntos de dados gerados por simulação mimetizam a ação do genótipo de indivíduos (seres humanos, animais ou plantas) sobre determinado fenótipo ou característica. As informações do genótipo de cada indivíduo da população advêm de mutações não-deletérias que se estabeleceram na população com uma frequência mínima de 1% (BROOKES, 1999). Essas mutações são denominadas de polimorfismos de base única (do inglês *Single Nucleotide Polymorphism* - SNP) e o pressuposto básico de estudos de associação em escala genômica (do inglês - *Genome-wide Association Studies* - GWAS) é que os mesmos podem ser usados para explicar a característica considerada, pois considera-se que existirão SNPs em alto desequilíbrio de ligação[1] com o(s) verdadeiro(s) locus(ci) da(s) mutação(ões) causal(is). Em GWAS, são capturados diversos SNPs ao longo de cada cromossomo para aumentar a chance de encontrar regiões (genes[2]) no genoma que tenham associação com o fenótipo em questão. Entretanto, muitos dos SNPs considerados não tem qualquer tipo de relação causal com a característica, logo, é de suma importância selecionar adequadamente os SNPs informativos e eliminar os não informativos.

Em geral, o número de SNPs é superior ao número de indivíduos na população, logo, é necessário que seja aplicado um ou mais algoritmos serialmente para seleção de atributos (SNPs) para descobrir os SNPs causais. A descoberta de SNPs relacionados às doenças complexas, denominada características quantitativas, resultam de complexas interações entre numerosos fatores ambientais e alelos[3] de muitos genes (WANG et al., 2005).

[1] Associação não-aleatória entre SNPs (PIERCE, 2010). No contexto de seleção de atributos, SNPs com alto desequilíbrio de ligação gerarão variáveis codificadas numericamente altamente correlacionadas.

[2] Gene é normalmente definido como um segmento de DNA que contém as instruções para produzir uma determinada proteína, embora essa definição sirva para a maioria dos genes, vários deles produzem moléculas de RNA ao invés de proteínas como produto final (ALBERTS et al., 2009)

[3] Alelos ou genes alelos são as várias formas de um gene(PIERCE, 2010)

A mostra o processo de codificação numérica usado tanto para o genótipo, capturado pelos SNPs, quanto para o fenótipo mapeado por valores discretos binários. Por exemplo, os valores da variável SNP1 em relação aos quatro indivíduos na base de dados inicial original, indica que o alelo A (representado pela base nitrogenada A que está ligada à base nitrogenada T em um cromossomo) é o possui maior frequência alélica na população, enquanto que o alelo a (representado pela base nitrogenada G que está ligada à base nitrogenada C no cromossomo homólogo), a menor frequência alélica. Desta forma, o número 1 representa a ausência do alelo a ou o homozigoto de referência AA (não necessariamente dominante), o número 2 indica o heterozigoto Aa e o número 3 caracteriza o homozigoto variante aa (não necessariamente recessivo). No caso do SNP2, a base nitrogenada G é o alelo com menor frequência simbolizado por a, enquanto que a T é o alelo com maior frequência, representado por A. Para os SNPs 3 e 4, a codificação é análoga usada para os SNPs 1 e 2. Os valores fenotípicos são dados por dois estados possíveis: sadio (codificado como 1) e doente (codificado como 0). A matriz de dados de entrada para os dois conjunto de dados (1 e 2) processado pelo algoritmo de seleção SCBR, possui o formato da terceira e última matriz da 5.3.1.

Base de dados inicial original

	SNP1	SNP2	SNP3	SNP4	SNP5	Fenótipo
Indivíduo 1	AA	GG	GG	CT	TT	Doente
Indivíduo 2	AG	GG	GG	CT	TT	Sadio
Indivíduo 3	AG	GT	AG	TT	TT	Sadio
Indivíduo 4	GG	TT	AA	TT	CC	Doente

Base de dados inicial com codificação alélica

	SNP1	SNP2	SNP3	SNP4	SNP5	Fenótipo
Indivíduo 1	*AA*	*aa*	*AA*	*Aa*	*AA*	Doente
Indivíduo 2	*Aa*	*aa*	*AA*	*Aa*	*AA*	Sadio
Indivíduo 3	*Aa*	*Aa*	*Aa*	*aa*	*AA*	Sadio
Indivíduo 4	*aa*	*AA*	*aa*	*aa*	*aa*	Doente

Base de dados inicial com codificação numérica

	SNP1	SNP2	SNP3	SNP4	SNP5	Fenótipo
Indivíduo 1	1	3	1	2	1	0
Indivíduo 2	2	3	1	2	1	1
Indivíduo 3	2	2	2	3	1	1
Indivíduo 4	3	1	3	3	3	0

Figura 5.5: Codificação do conjunto de dados gerados pelo pacote SCRIME. Adaptado de (OLIVEIRA, 2015)

.

Essa simulação foi construída para mostrar que existe a possibilidade de utilizar o

SCBR em seleção de marcadores SNPs para problemas do tipo caso-controle, isto é, problemas de classificação. Esses problemas advêm de GWAS realizados em humanos, animais e plantas com relação a detecção de loci responsáveis pelo aumento do risco de desenvolvimento de doenças.

5.3.1.1 Conjunto de dados 1 - Base SNP com efeitos aditivos e não-aditivos

O número de controles (codificados como 0) e de casos (codificados como 1) foram iguais a, respectivamente, 138 e 862, o que mostra um cenário de classes desbalanceadas. Assim, a função que gera as classes é baseada em um modelo de regressão logística implementada a partir da função simulateSNPglm do pacote SCRIME do R e dada pela Expressão 1. A Expressão 2 mostra os valores adotados para os coeficientes betas.

As variáveis explicativas são descritas como: $(X1 = 1)$,$(X2 = 2)$, $(X3 = 3)$, $(X4 \neq 1) \wedge (X5 = 3)$ e $(X6 = 1) \wedge (X7 = 2) \wedge (X8 = 3)$, sendo o operador $\wedge$ o operador lógico de conjunção. As Expressões $(X1 = 1)$, $(X2 = 2)$ e $(X3 = 3)$ simbolizam somente efeitos aditivos. A Expressão $(X4 = 1) \wedge (X5 = 2)$ designa a interação entre os SNPs 4 e 5, ou seja, quando o SNP4 for diferente de 1 (genótipo homozigoto AA) e, simultaneamente, o SNP5 for igual a 3 (genótipo homozigoto aa), o efeito da interação será destacado pelo coeficiente β_4. Interpretação análoga pode ser dada para a expressão $L_5 = (SNP6 = 1) \wedge (SNP7 = 2) \wedge (SNP8 = 3)$ que traduz a interação entre o trio de marcadores formado pelos SNPs 6, 7 e 8, sendo a mesma potencializada pelo coeficiente $\beta 5$.

Simulador: SCRIME do software R
Função utilizada: simulateSNPglm
Parâmentros da função:
n.obs = 1000
n.snp = 100
list.ia = list(1,2,3,c(-1,3),c(1,2,3))
list.snp = list(1,2,3,c(4,5),c(6,7,8))
p.cutoff = 0.7
beta0=0
beta=c(2,1.3,0.9,2,3)
err.fun = NULL
maf=c(0.1,0.4)
rand=123)

$$Y = \beta_0 + \beta_1(X_1 = 1) + \beta_2(X_2 = 2) + \beta_3(X_3 = 3) + \beta_4(X_4 \neq 1) \wedge (X_5 = 3) + \beta_5(X_6 = 1) \wedge (X_7 = 2) \wedge (X_8 = 3)$$

$$Y = 0 + 2(X_1 = 1) + 1,3(X_2 = 2) + 0,9(X_3 = 3) + 2(X_4 \neq 1) \wedge (X_5 = 3) + (X_6 = 1) \wedge (X_7 = 2) \wedge (X_8 = 3)$$

A tabela 5.1 indica a distribuição do número de instâncias por classe.

Tabela 5.1: Total de instâncias por classe

Total de sadios(s)	**Total de doentes (n)**
127	862

5.3.1.2 Conjunto de dados 2 - Base SNP somente com efeitos aditivos (linear)

Esse conjunto de dados foi gerado de forma a considerar somente efeitos aditivos de cinco SNPs sobre o fenótipo. Todos os efeitos tem a mesma magnitude, ou seja, todos os cinco $\beta's$ são iguais a 5. A ideia foi construir um cenário com menor dificuldade do que o conjunto de dados 1 para que o algoritmo SCBR selecione o maior número de SNPs causais e elimine a maior quantidade de SNPs não causais. Espera-se que o desempenho do SCBR seja superior nesses dados, pois não existem interações entre as variáveis.

Simulador: SCRIME do software R
Função utilizada: simulateSNPglm
Parâmentros da função:
n.obs = 1000
n.snp = 100
list.ia = list(1,2,3,-1,-2)
list.snp = list(10,20,30,40,50)
p.cutoff = 0.7
beta0=0
beta=c(5,5,5,5,5)
err.fun = NULL
maf=c(0.1,0.4)
rand=123)

$$Y = \beta_0 + \beta_1(X_10 = 1) + \beta_2(X_20 = 2) + \beta_3(X_30 = 3) + \beta_4(X_40 \neq 1) \wedge \beta_50(X_50 \neq 2) \wedge (X_7 = 2) \wedge (X_8 = 3)$$

$$Y = 0 + 5(X_10 = 1) + 5(X_20 = 2) + 5(X_30 = 3) + 5(X_40 \neq 1) + 5(X_50 \neq 2)$$

A tabela 5.2 indica a distribuição do número de instâncias por classe.

Tabela 5.2: Total de instâncias por classe

Total de sadios (s)	**Total de doentes (n)**
62	936

5.3.2 VALIDAÇÃO DO MÉTODO SCBR

A seguir é apresentado o resultado da classificação utilizando as características selecionadas pelos métodos *Wrapper + Best First*, *Wrapper + Evolucionary* e o método proposto neste trabalho, o SCBR, aplicado as bases SNP com efeitos aditivos e a Base SNP somente com efeitos aditivos (linear), utilizando os algoritmos IBK, Decision Stump e CBA para realizarem a classificação.

5.3.3 BASE COM EFEITOS ADITIVOS E NÃO-ADITIVOS

Nesta subseção serão apresentados os resultados utilizando a base de dados SNP, juntamente com diferentes tipos de algoritmos de classificação. Serão utilizadas as base de dados cujas características foram selecionadas pelos métodos de seleção de características informados anteriormente.

A tabela a seguir apresenta o resultado da classificação utilizando a base de treinamento, juntamente com o algoritmo IBK como classificador.

Tabela 5.3: Base Treinamento - Algoritmo IBK

Nº de atributos	**Métodos de seleção**	**Classificador**	**Base Treinamento**	**Base Teste**
100	Sem seleção	IBK	83,84%	78,39%
30	*Wrapper + Evolucionary*	IBK	82,69%	78,39%
2	*Wrapper+ Best First*	IBK	99,23%	95,97%
10	SCBR	IBK	96,15%	94,47%

Analisando a tabela 5.3, pode-se perceber que o melhor resultado de classificação em termos de porcentagem é utilizando a base de treinamento e teste, contendo as características selecionadas pelo método *Wrapper + Best First*. Mas analisando a tabela 5.4

é possível perceber que esse valor elevado de porcetagem está relacionado com a classificação de instâncias da classe majoritária (***n***), explicando assim os 95,97% de acerto na classificação das instâncias. Porém o método SCBR apresentou melhor resultado de classificação das instâncias da classe minoritária (***s***), isso pode ser confirmado verificando a tabela 5.4.

Tabela 5.4: Base Teste - Algoritmo IBK

Algoritmo IBK	
Sem Seleção	***Wrapper + Evolucionary***
Matriz de Confusão	Matriz de Confusão
a b 13 14 |a = s 29 143 |b = n	a b 19 8 |a = s 35 137 |b = n
Wrapper + Best First	**SCBR**
Matriz de confusão	Matriz de Confusão
a b 25 2 |a = s 6 166 |b = n	a b 27 0 |a = s 11 161 |b = n

A tabela a seguir, apresenta os resultados da classificação utilizando o algoritmo *Decision Stump* como classificar aplicado à base sem seleção de características e às bases contendo as características selecionadas pelos métodos de seleção.

Tabela 5.5: Base Treinamento - Algoritmo *Decision Stump*

Nº de atributos	**Métodos de seleção**	**Classificador**	**Base Treinamento**	**Base Teste**
100	Sem seleção	DS	87,30%	83,41%
30	*Wrapper + Evolucionary*	DS	87,30%	83,41%
2	*Wrapper+ Best First*	DS	87,30%	83,41%
10	SCBR	DS	87,30%	83,41%

Tabela 5.6: Base Teste - Algoritmo *Decision Stump*

Algoritmo *Decision Stump*	
Sem Seleção	***Wrapper + Evolucionary***
Matriz de Confusão	Matriz de Confusão
a b 27 0 |a = s 33 139 |b = n	a b 27 0 |a = s 33 139 |b = n
Wrapper + Best First	**SCBR**
Matriz de confusão	Matriz de Confusão
a b 27 0 |a = s 33 139 |b = n	a b 27 0 |a = s 33 139 |b = n

Analisando os resultados apresentados nas tabelas 5.5 e 5.6 pode-se perceber que não houve ganho na classificação de instâncias utilizando a base contendo as características selecionadas pelos métodos de seleção visto que os resultados dos métodos são iguais ao resultado da base sem seleção de características.

A seguir é apresentada a tabela contendo os resultados da classificação utilizando o algoritmo CBA como classificador aplicado às bases de treinamento e teste.

Tabela 5.7: Base Treinamento - Algoritmo CBA

Nº de atributos	**Métodos de seleção**	**Classificador**	**Base Treinamento**	**Base Teste**
100	Sem seleção	CBA	99,23%	95,47%
30	*Wrapper + Evolucionary*	CBA	92,30%	84,92%
2	*Wrapper+ Best First*	CBA	99,23%	95,47%
10	SCBR	CBA	99,61%	96,48%

Tabela 5.8: Base Teste - Algoritmo CBA

Algoritmo CBA	
Sem Seleção	***Wrapper + Evolucionary***
Matriz de Confusão	Matriz de Confusão
a b 25 2 |a = s 7 165 |b = n	a b 21 6 |a = s 7 165 |b = n
Wrapper + Best First	**SCBR**
Matriz de confusão	Matriz de Confusão
a b 21 6 |a = s 3 169 |b = n	a b 27 0 |a = s 7 165 |b = n

Analisando a tabela 5.7 pode-se perceber que o algoritmo SCBR apresenta o melhor resultado em termos de porcentagem de classificação de instâncias, tanto para a base de treinamento quanto para a base de teste. Esse percentual está relacionado com o número de instâncias da classe minoritária (***s***) classificadas corretamente, ou seja, o algoritmo SCBR é o único algoritmo que conseguiu classificar de forma correta todas as instâncias da classe minoritária (***s***).

A base SNP com efeitos aditivos e interação também foi utilizada para realizar os experimentos de seleção de características apresentados em (OLIVEIRA, 2015), onde foram selecionadas 8 características consideradas como sendo as mais relevantes. Das 8 características, 3 foram selecionadas pelo SCBR. O método *Wrapper + Best First* selecionou apenas 2 características iguais as selecionadas em (OLIVEIRA, 2015). O método *Wrapper*

Tabela 5.9: Base Treinamento - Algoritmo IBK

Nº de atributos	**Métodos de seleção**	**Classificador**	**Base Treinamento**	**Base Teste**
100	Sem seleção	IBK	87,33%	86,43%
35	*Wrapper + Evolucionary*	IBK	74%	78,89%
4	*Wrapper+ Best First*	IBK	92,66%	91,45%
9	SCBR	IBK	94%	87,43%

+ Evolucionary, não selecionou nenhuma das características selecionadas em (OLIVEIRA, 2015).

Apesar o método de seleção SCBR não selecionar todas as características apresentadas em (OLIVEIRA, 2015), ele foi o único método que selecionou a maior quantidade de características sendo igual a 3.

A seguir é apresentado os resultados da seleção de características realizada na base SNP somente com efeitos aditivos (linear).

5.3.4 BASE SNP SOMENTE COM EFEITOS ADITIVOS (LINEAR)

A seguir serão apresentados os experimentos realizados com a Base SNP somente com efeitos aditivos (linear) utilizando a base de dados original, ou seja, contendo todas as características da base e a base de dados contendo as características selecionadas pelo Modelo Proposto(SCBR), *Wrapper + Best First* e *Wrapper + Evolucionary*

Vale ressaltar que o objetivo não é comparar o desempenho de cada algoritmo de classificação, e sim verificar se o algoritmo SCBR é eficiente e competitivo com as demais técnicas de seleção de característica.

A tabela 5.9 a seguir, apresenta o resultado da classificação utilizando o algoritmo IBK como classificador, aplicado às bases sem seleção de características e às bases contendo as caraterísticas selecionadas pelos métodos informados anteriormente.

Tabela 5.10: Base Teste - Algoritmo IBK

Algoritmo IBK	
Sem Seleção	***Wrapper + Evolucionary***
Matriz de Confusão	Matriz de Confusão
a b 6 6 |a = s 21 166 |b = n	a b 2 10 |a = s 32 155 |b = n
Wrapper + Best First	**SCBR**
Matriz de confusão	Matriz de Confusão
a b 12 0 |a = s 17 170 |b = n	a b 12 0 |a = s 25 162 |b = n

Analisando a tabelas 5.10 onde a mesma representa o resultado da classificação aplicado as bases de teste, pode-se observar que as bases contendo as características selecionadas pelos métodos *Wrapper + Best First* e SCBR apresentam melhor resultado na classificação das instâncias da classe minoritária. O método *Wrapper + Best First* apresentou melhor resultado em termos de porcentagem, devido a classificação correta de maior quantidade de instâncias da classe majoritária, explicando assim os 91,45% de predição na classificação para a base de teste.

A tabela 5.11 a seguir, apresenta os resultados da classificação utilizando o algoritmo *Decision Stump* (DS) como classificador, aplicado à base sem seleção de características e, aplicado às bases contendo as características selecionadas pelos métodos de seleção.

Tabela 5.11: Base Treinamento - Algoritmo *Decision Stump*

Nº de atributos	**Métodos de seleção**	**Classificador**	**Base Treinamento**	**Base Teste**
100	Sem seleção	DS	80%	70,85%
35	*Wrapper + Evolucionary*	DS	66,66%	93,96%
4	*Wrapper+ Best First*	DS	80%	70,85%
9	SCBR	DS	80%	70,85%

Analisando a tabela 5.11 é possível perceber a obtenção de um padrão na classificação utilizando as bases sem seleção de características, a base contendo as características pelo método *Wrapper + Best First* e a base contendo as características selecionadas utilizando o método SCBR. Apenas o método *Wrapper + Evolucionary* apresenta resultado de classificação diferente e relativamente ruim se comparado com os demais resultados apresentados para esse algoritmo. A tabela a seguir apresenta o resultado da classificação para a base de teste.

Tabela 5.12: Base Teste - Algoritmo *Decision Stump*

Algoritmo *Decision Stump*	
Sem Seleção	***Wrapper + Evolucionary***
Matriz de Confusão	Matriz de Confusão
a b 12 0 |a = s 58 129 |b = n	a b 0 12 |a = s 0 187 |b = n
Wrapper + Best First	**SCBR**
Matriz de confusão	Matriz de Confusão
a b 12 0 |a = s 58 129 |b = n	a b 12 0 |a = s 58 129 |b = n

Analisando a tabela 5.12, pode-se notar que a classificação em termos de número de instâncias classificadas de forma correta, apresenta resultado igual para dois dos três métodos apresentados, sendo eles *Wrapper + Best First* e o SCBR. Apesar do método *Wrapper + Evolucionary* apresentar uma classificação de 93% em termos de porcentagem de instâncias classificadas corretamente, pode-se perceber na tabela 5.12 que esse número é elevado visto que todas as instâncias da classe majoritária (***n***) foram classificadas de forma correta, fazendo com a porcentagem fosse elevada, em contrapartida, todas as instâncias da classe minoritária (***s***) foram classificadas de forma incorreta.

A tabela 5.13 a seguir, apresenta o resultado da classificação utilizando o algoritmo CBA como classificador, aplicado às bases sem seleção de características e às bases contendo as caraterísticas selecionadas pelos métodos informados anteriormente.

Tabela 5.13: Base Treinamento - Algoritmo CBA

Nº de atributos	**Métodos de seleção**	**Classificador**	**Base Treinamento**	**Base Teste**
100	Sem seleção	CBA	100%	98,49%
35	*Wrapper + Evolucionary*	CBA	96,66%	83,41%
4	*Wrapper+ Best First*	CBA	100%	96,98%
9	SCBR	CBA	100%	97,98%

Analisando a tabela 5.13 é possível perceber uma estabilidade no percentual acerto na classificação das instâncias em dois dos três métodos de seleção de características, sendo eles *Wrapper + Best First* e o SCBR. O método *Wrapper + Evolucionary* é o único método que apresentou uma classificação diferente em relação aos demais métodos. O comportamento da classificação utilizando o algoritmo CBA é igual na classificação utilizando o algoritmo *Decision Stump* como classificador, ou seja, a classificação permanece estável para dois métodos de seleção de distorcida para um terceiro método.

A tabela a seguir apresenta o resultado da classificação utilizando o algoritmo CBA como classificador aplicado à base de teste.

Tabela 5.14: Base Teste - Algoritmo CBA

Algoritmo CBA	
Sem Seleção	***Wrapper + Evolucionary***
Matriz de Confusão	Matriz de Confusão
a b 12 0 |a = s 3 184 |b = n	a b 2 10 |a = s 23 164 |b = n
Wrapper + Best First	**SCBR**
Matriz de confusão	Matriz de Confusão
a b 12 0 |a = s 6 181 |b = n	a b 12 0 |a = s 4 183 |b = n

Analisando as tabelas 5.13 e 5.14 pode-se perceber que os métodos de seleção de características não apresentaram resultados melhores do que a classificação utilizando todos as características, tanto em termos de porcentagem quanto em termos de instâncias classificadas corretamente.

Analisando os resultados da tabela 5.14 e métodos de seleção de características, é possível perceber que o algoritmo SCBR apresentou melhor classificação em relação aos demais métodos, tanto para o número de instâncias classificadas corretamente, quanto para a porcentagem de classificação, mantendo um equilíbrio na classificação das instâncias.

É importante ressaltar que a Base SNP somente com efeitos aditivos (linear) foi utilizada nos experimentos realizados em (OLIVEIRA, 2015) para a realização de seleção as características. Os experimentos retornaram como resposta, cinco características consideradas como sendo as mais representativas.

O algoritmo SCBR selecionou 9 características como sendo as características mais representativas. Das 9 características, 5 características são iguais as selecionadas em (OLIVEIRA, 2015).

O método *Wrapper + Best First* selecionou apenas 4 características, as quais também estão contidas na seleção realizada em (OLIVEIRA, 2015).

O método *Wrapper + Evolucionary* selecionou 35 características, porém apenas uma característica contida em (OLIVEIRA, 2015) foi selecionada pelo método.

Enfim, o algoritmo SCBR foi o único que selecionou todas as características também

selecionadas em (OLIVEIRA, 2015).

6 EXPERIMENTOS NUMÉRICOS

Este capítulo tem por objetivo apresentar os experimentos realizados durante todo o desenvolvimento do trabalho. É apresentado também as informações das bases de dados utilizadas nos experimentos.

Durante a realização dos experimentos diversas bases de dados foram utilizadas em diferentes etapas. Algumas bases foram retiradas do repositório *UCI*[1], onde estão dispostas diversas bases de dados utilizadas em Aprendizagem de Máquina. O *software* utilizado para realizar a classificação por regras de associação, seleção de características e comitê de classificadores, é o *Weka*[2], uma ferramenta *Open Source* desenvolvida pela Universidade de Waikato, na Nova Zelândia, e muito utilizada para realizar tarefas de mineração de dados.

A tabela a seguir apresenta as bases de dados, com suas respectivas quantidade de instâncias e atributos, bem como o tipo de base de dados.

Tabela 6.1: Bases de dados utilizadas nos experimentos

Bases de dados	**Instância**	**Atributos**	**Tipos de Bases**
Supermercado X	2659	104	Transacional
Base Supermarket	4627	217	Transacional
Base SNP com efeitos aditivos e interação	989	100	Não Transacional
Base SNP somente com efeito aditivo	998	100	Não Transacional
Base CH	3196	37	Não Transacional
Base Murshroom	8124	23	Não Transacional

Os atributos contidos nas bases de dados utilizadas nos experimentos, são do tipo nominal. Todas as bases de dados mencionadas possuem apenas duas classes.

A base de dados Supermercado X é uma base de dados transacional que contém informações de compras de cliente de um supermercado real, o nome Supermercado X foi dado para manter no anonimato o nome real do supermercado. Tal base é composta por duas classes, sendo elas manhã e tarde, referentes a produtos que foram comprados no turno da manhã e no turno da tarde. A classe manhã, representa a classe minoritária da base de dados. Isso pode é um problema quando a classificação é realizada sem nenhum tratamento nos dados, pois, como dito anteriormente pode ocorrer um viés de acerto da

[1] http://archive.ics.uci.edu/ml/
[2] http://www.cs.waikato.ac.nz/ml/weka/

classe majoritária.

A Base *Supermarket* também é uma base de dados transacional e muito utilizada na mineração de regra de associação e pode ser encontrada no pacote do *Weka*.

A justificativa da utilização da Base CH[3] pode ser definida por três motivos: primeiro por não ser uma base de dados do tipo transacional, mas possuir características de bases de dados transacionais, segundo é uma base que apresenta uma quantidade relevante de instâncias e terceiro por ser uma base de dados com atributos do tipo nominal.

Já Base *Mushromm* é uma base onde são apresentados 23 tipos de cogumelos. A base foi utilizada também por não ser uma base de dados transacional, porém também apresenta características de bases de dados transacionais e, por obter um valor relativamente alto de número de instâncias.

Em síntese, foi aplicado o método de balanceamento (CCDD) nas bases Supermercado X, Base somente com efeitos aditivos (linear) e Base SNP com efeitos aditivos e interação, por apresentarem grande desbalanceamento entre as classes. A seleção de características foi aplicada a todas as bases apresentadas neste trabalho.

O objetivo da utilização das bases de dados do tipo não transacional é analisar se a técnica de balanceamento proposto, e a técnica de seleção de características, apresentam resultados satisfatórios em diferentes tipos de bases de dados.

O objetivo desta etapa do trabalho é diminuir a tendência de maior classificação da classe majoritária e aumentar o número de instâncias classificadas como sendo da classe minoritária, mantendo assim o equilíbrio na classificação das instâncias

Para melhor entendimento do resultado da classificação utilizando uma base desbalanceada, tem-se, como exemplo, a classificação da base Supermercado X em sua forma original, ou seja, sem nenhum tratamento de desbalanceamento.

Tabela 6.2: Classificação sem tratamento de desbalanceamento.

Base Supermercado X		
Algoritmo CBA		
Classificadas corretamente: 71,42% Classificadas incorretamente: 28,57%		
Matriz de Confusão		
a	b	
0	600	|a= Manhã
0	1500	|b=Tarde

[3]http://www.hakank.org/weka/

Analisando a tabela 6.2, pode-se perceber que a porcentagem de instâncias classificadas corretamente foi de 71.42%, o que leva a crer que o modelo construído é um modelo "ótimo" em termos de porcentagem de instâncias classificadas corretamente.

Partindo para a analise da Matriz de Confusão, pode-se perceber que as instâncias da manhã foram classificadas como sendo instâncias da tarde, isso fez que com que porcentagem de instâncias classificadas corretamente fosse elevada. Explicando assim os 71.42% de classes classificadas corretamente.

Para a realização de todos os experimentos foi utilizado o algoritmo de classificação por regras de associação conhecido como CBA. A justificativa da utilização desse algoritmo dá-se por ser um algoritmo já consolidado, estudado e utilizado em diversos trabalhos quando se trata de classificação em regras de associação, utilizando base de dados transacionais.

Os primeiros experimentos realizados para tratar o desbalanceamento, foram utilizando os métodos *Oversampling*, *Undersamplig* e *SMOTE* com o objetivo de tentar minimizar o desbalanceamento entre as classes e melhorar a classificação da classe minoritária, já que esses métodos são conhecidos para minimizar o desbalanceamento entre as instâncias de uma base de dados.

Para tais experimentos alguns parâmetros foram definidos, como por exemplo, para utilizar o *Oversampling* tem-se a replicação de 100% das instâncias da classe minoritária. Para o *Undersamplig* foram utilizados 100% dos dados da classe majoritária. A remoção das instâncias é feita de forma aleatória. Para o *SMOTE*, tem-se a utilização de 100% dos dados da classe minoritária para a criação de dados sintéticos.

Como foi utilizado o algoritmo CBA para realizar a classificação à partir das regras de associação, e como o mesmo utiliza o algoritmo Apriori para gerar tais regras, fez-se necessário definir valores de suporte mínimo, confiança mínima, número de regras a serem geradas. Para verificar quais valores seriam mais adequados, foram realizados diversos experimentos com variações desses valores.

Para definir o valor de suporte mínimo, foram realizados experimentos variando o valor de suporte mínimo entre 0,1 à 0,001, sendo fixado em 0,01 por apresentar melhor resultado na classificação das instâncias. Para a confiança mínima foram realizados experimentos com variações nos valores de confiança mínima em 0,01 à 1, sendo fixado em 0,5 por também apresentar melhor resultado.

O algoritmo CBA, antes de realizar a classificação das instâncias por regras de as-

sociação, realiza uma poda nas instâncias menos representativas, ou seja, instâncias que não influenciam na classificação ou não atendem ao valor de suporte e confiança mínimos são cortadas antes da classificação. Por esse motivo foi escolhido que o número de regras geradas seria igual a 10000, visto que muitas dessas regras são excluídas no processo de poda.

Sendo assim, tem-se os valores de suporte mínimo igual a 0,01, confiança mínima igual a 0,5 e número de regras a serem geradas igual a 10000. Esses valores também foram utilizados para realizar todos os experimentos com o qual o CBA está envolvido. A seguir serão apresentados os resultados utilizando os métodos de balanceamento de classes desbalanceadas.

6.1 CLASSIFICAÇÃO EM BASES DE DADOS DESBALANCEADAS

É apresentado nesta seção os resultados da classificação utilizando métodos de balanceamento. Os métodos serão aplicados nas bases Supermercado X, Base SNP somente com efeitos aditivos (linear) e Base SNP com efeitos aditivos e interação. Será realizado uma análise detalhada nos resultados obtidos.

Para realizar os experimentos com essas bases de dados, os dados foram divididos em conjuntos de treinamento e teste. O conjunto de teste representa 20% do número de instâncias de cada classe. O restante foi utilizado no conjunto de treinamento. A divisão entre

Esse valor de 20% de instâncias de cada classe para a base de teste, foi escolhido após a realização de diversos experimentos, onde concluiu-se que se a quantidade de instâncias da classe de teste fossem maior que 20%, as instâncias da classe minoritária seriam prejudicadas na base de treinamento, visto que restaria um número muito pequeno de instâncias para serem utilizadas na base de treinamento. Portanto, esse valor de 20% de cada classe tornou-se um valor padrão neste trabalho para criação das bases de teste.

É importante ressaltar que o total de instâncias da base de treinamento pode se alterar ao longo do processo de balanceamento, visto que, cada método, faz manipulações na base de treinamento de formas distintas.

6.1.1 BASE SUPERMERCADO X

Para realizar os experimentos utilizando a Base Supermercado X, foram separados dois conjuntos, sendo eles de treinamento e teste. A base de dados contém 818 instâncias da classe manhã e 1985 da classe tarde, sendo selecionadas para a base de teste, 163 instâncias da classe manhã e 396 instâncias da classe da tarde, totalizando 559. A base de treinamento é composta por 600 instâncias da classe manhã e 1500 instâncias da classe tarde, totalizando 2100 instâncias da base de treinamento, como pode ser visto na tabela 6.3.

Tabela 6.3: Configuração da base Supermercado X

Classes	**Nº de instâncias por classe**	**Base treinamento**	**Base teste**
Manhã	763	600	163
Tarde	1896	1500	396

A tabela 6.4 apresenta o resultado da classificação após a aplicação de diferentes técnicas de balanceamento. Após o balaceamento, foi utilizado o algoritmo CBA para realizar a classificação das instâncias.

Tabela 6.4: Métodos de balanceamento

Método	**Base treinamento**	**Base Teste**
Sem balanceamento	71,42%	70,84%
Oversampling	60,71%	65,47%
Undersampling	61,83%	44,00%
CCDD	57,80%	56,88%

Analisando a tabela 6.4, pode-se perceber que apesar da porcetagem de classificação girar em torno de 70% de instâncias classificadas corretamente, tanto na base de treinamento quanto na base de teste, a classificação apresenta falhas na predição para a base sem aplicação de técnicas de balanceamento. Este fato, pode ser constatado na tabela 6.5 onde é apresentada a matriz de confusão obtida após a classificação das instâncias. Nela pode-se perceber que as instâncias da manhã foram completamente classificadas como tarde, tanto na base de treinamento quanto na base de teste, comprovando a análise realizada. Tal comportamento indica que a utilização da base de dados em seu formato original, irá criar um classificador fraco, ou seja, com baixo poder de predição.

Tabela 6.5: Matriz de Confusão - Base Supermercado X

Matriz de Confusão					
Base Treinamento			**Base teste**		
Sem balanceamento					
a	b		a	b	
0	600	|a = manhã	0	163	|a = manhã
0	1500	|b = tarde	0	396	|b = tarde
Oversampling					
a	b		a	b	
352	698	|a = manhã	33	130	|a = manhã
127	923	|b = tarde	63	333	|b = tarde
Undersampling					
a	b		a	b	
507	93	|a = manhã	124	39	|a = manhã
365	235	|b = tarde	274	122	|b = tarde
CCDD					
a	b		a	b	
445	155	|a = manhã	125	38	|a = manhã
731	769	|b = tarde	204	193	|b = tarde

Através dos resultados apresentados na tabela 6.4, pode-se perceber que o algoritmo CCDD em termos de porcentagem de instâncias classificadas corretamente, não apresentou resultado satisfatório para a base de treinamento.

Em termos de números de instâncias da classe minoritária classificadas de forma correta, pode-se observar na tabela 6.5 que o algoritmo CCDD apresentou melhor resultado, ou seja, 125 instâncias classificadas corretamente para a classe minoritária.

Apesar da classificação ser igual a 65% de instâncias classificadas corretamente para a base de treinamento, o método *Oversampling*, não apresentou resultado de acordo com o objetivo esperado, visto que o método não conseguiu classificar as instâncias da classe minoritária cujas quais são mais difíceis de serem classificadas.

O método *Undersampling*, apesar de ser o método mais simples de balanceamento de classe, também apresentou resultados satisfatórios, pois conseguiu classificar corretamente 124 instâncias da classe minoritária na base de teste. O *SMOTE* não apresentou nenhum resultado, por esse motivo, os mesmos não serão apresentados.

O método CCDD apresentou melhor resultados na classificação das instâncias da classe minoritária.

A seguir é apresentado os resultados de classificação da base SNP somente com efeitos aditivos (linear) utilizando os métodos de balanceamento *Oversampling*, *Undersampling*

e CCDD.

6.1.2 BASE SNP COM EFEITOS ADITIVOS E INTERAÇÃO

A seguir são apresentados os resultados de classificação sem a utilização de métodos de balanceamento de e com a utilização dos métodos de balanceamento. Vale ressaltar que a base SNP com efeitos aditivos e interação também não é uma base de dados do tipo transacional. Nesta base, também será utilizado o padrão adotado neste trabalho de 20% de cada classe para a criação da base de teste.

A base SNP com efeitos aditivos e interação é composta por 138 instâncias da classe ***s*** e 862 instâncias da classe ***n***. A base de teste é composta por 27 instâncias da classe ***s*** e 172 instâncias da classe ***n***. A base de treinamento é composta por 690 instâncias da classe ***n*** e 100 instâncias da classe ***s***, como pode ser observado na tabela 6.6.

Tabela 6.6: Configuração da base SNP com efeitos aditivos e interação

Classes	**Nº de instâncias por classe**	**Base treinamento**	**Base teste**
s	127	100	27
n	862	690	172

A tabela 6.7 apresenta o resultado da classificação após a aplicação de diferentes técnicas de balanceamento. Após o balaceamento, foi utilizado o algoritmo CBA para realizar a classificação das instâncias.

Tabela 6.7: Métodos de balanceamento

Método	**Base treinamento**	**Base Teste**
Sem balanceamento	87,35%	86,43%
Oversampling	98,86%	95,97%
Undersampling	97,5%	96,48%
CCDD	97,84%	97,48%

Analisando a tabela 6.7 pode-se observar o método *Oversampling* apresentou o melhor resultado em termos de porcentagem de classificação das instâncias. O método CCDD apresentou o melhor resultado para a base de teste.

Analisando a tabela 6.8, é possível observar que a base SNP com efeitos aditivos e interação apresenta o mesmo comportamento das bases Supermercado X e SNP Linear em relação ao número de instâncias classificadas corretamente, ou seja, as instâncias da classe minoritária são classificadas de forma incorreta quanto utiliza-se a base desbalanceada para realizar a classificação.

Tabela 6.8: Matriz de Confusão - Base SNP com efeitos aditivos e interação

Matriz de Confusão					
Base Treinamento			**Base teste**		
Sem balanceamento					
a	b		a	b	
0	100	|a = ***s***	0	27	|a = ***s***
0	690	|b = ***n***	0	172	|b = ***n***
Oversampling					
a	b		a	b	
395	0	|a = ***s***	26	1	|a = ***s***
9	386	|b = ***n***	7	165	|b = ***n***
Undersampling					
a	b		a	b	
100	0	|a = ***s***	27	0	|a = ***s***
5	95	|b = ***n***	7	165	|b = ***n***
CCDD					
a	b		a	b	
100	0	|a = ***s***	27	0	|a = ***s***
17	673	|b = ***n***	5	167	|b = ***n***

Analisando a tabela 6.8, pode-se perceber que o comportamento das bases anteriores se repete com esta base de dados quando é utilizado o CCDD na classificação, ou seja, apresenta resultado satisfatório aplicado na base de treinamento, porém é o método que apresenta melhor resultado de classificação na base de teste, tanto para a classificação das instâncias da classe minoritária, quanto para a classificação das instâncias da classe majoritária, mantendo assim um equilíbrio na classificação das instâncias.

Os demais métodos apresentaram uma classificação satisfatória da classe minoritária quando a classificação é realizada na base de teste.

6.1.3 BASE SNP SOMENTE COM EFEITOS ADITIVOS (LINEAR)

A Base SNP somente com efeitos aditivos (linear) é composta por 64 instâncias da classe ***s*** e 936 instâncias da classe ***n***. Foram retiradas 12 instâncias da classe ***s*** e 187 instâncias da classe ***n*** para compor a base de teste. A base de treinamento é composta por 749 instâncias da classe ***n*** e 50 da classe ***s***, totalizando 799 instâncias, como pode ser visto na tabela 6.9. Vale ressaltar que a Base SNP somente com efeitos aditivos (linear), não é uma base de dados transacional.

A tabela 6.4 apresenta o resultado da classificação após a aplicação de diferentes técnicas de balanceamento. Após o balaceamento, foi utilizado o algoritmo CBA para realizar

Tabela 6.9: Configuração da base SNP somente com efeitos aditivos (linear)

Classes	**Nº de instâncias por classe**	**Base treinamento**	**Base teste**
s	62	50	12
n	936	749	187

a classificação das instâncias.

Tabela 6.10: Métodos de balanceamento

Método	**Base treinamento**	**Base Teste**
Sem balanceamento	93,74%	91,45%
Oversampling	100%	99,4%
Undersampling	100%	94,47%
CCDD	98,7%	97,98%

Analisando a tabela 6.10, pode-se perceber que apesar de não ser uma base de dados transacional, a base SNP somente com efeitos aditivos (linear) apresentou o mesmo comportamento observado na base Supermercado X, após a realização da classificação sem nenhum tratamento da base. Neste caso tem-se, uma maior porcentagem de instâncias classificadas corretamente, estando em torno de 93% na classificação das instâncias, porém pode ser visto na tabela 6.11 que a predição das instâncias não foi realizada de forma correta, uma vez que todas as instâncias da classe ***s*** foram classificadas como sendo da classe ***n***.

Tabela 6.11: Matriz de Confusão - Base SNP somente com efeitos aditivos (linear)

Matriz de Confusão					
Base Treinamento			**Base teste**		
Sem balanceamento					
a	b		a	b	
0	50	|a = ***s***	0	12	|a = ***s***
0	749	|b = ***n***	0	187	|b = ***n***
Oversampling					
a	b		a	b	
399	0	|a = ***s***	12	0	|a = ***s***
0	399	|b = ***n***	1	186	|b = ***n***
Undersampling					
a	b		a	b	
50	0	|a = ***s***	12	0	|a = ***s***
0	50	|b = ***n***	9	178	|b = ***n***
CCDD					
a	b		a	b	
50	0	|a = ***s***	12	0	|a = ***s***
10	739	|b = ***n***	4	183	|b = ***n***

Pode ser observado na tabela 6.11 que classificação da base de dados original, culminou na criação de um classificador com baixo nível de predição, o que não é interessante.

Analisando a tabela 6.10 pode-se perceber que o CCDD apresentou resultado satisfatório para a base de treinamento, o mesmo acontece na base Supermercado X.

Ainda analisando a tabela 6.10, pode-se perceber que o método *Undersampling* apresentou 100% das instâncias classificadas corretamente, porém o número de instâncias utilizadas para treinar o classificador é consideravelmente baixo em realização ao número de instâncias apresentadas na base original treinamento, esse comportamento pode ser observado na tabela 6.11. Com isso fica difícil verificar se as 50 instâncias selecionadas representam a classe como um todo.

Analisando a base de teste da tabela 6.10, pode-ser perceber que apesar do método *Undersampling* apresentar 100% de instâncias classificadas corretamente para a base de treinamento, ele apresenta o pior resultado de classificação para a base de teste, o que indica que as instâncias da base de treinamento não representam a base como um todo, visto que o resultado com a base de teste obteve o pior desempenho.

Em suma, após as análises realizadas e os resultados apresentados, pode-se entender que o algoritmo CCDD também mostrou-se competitivo se comparado com métodos tradicionais de balanceamento, para essa base de dados.

Os demais métodos apresentaram uma classificação satisfatória da classe minoritária para a base de teste.

6.1.4 MÉTODOS *ENSEMBLE* DE CLASSIFICAÇÃO

Para validar o método de CCDD proposto neste trabalho, foram realizados experimentos utilizando diferentes métodos para aumentar predição de classificação das instâncias, mas agora baseados em um comitê de classificadores. Vale ressaltar que os comitês de classificadores não foram criados para tratar desbalanceamento entre classes, eles foram criados com o intuito de aumentar o nível de predição na classificação.

Os resultados apresentados utilizam os métodos *Boosting* e *Bagging* na tentativa de aumentar a predição da classificação. Além disso, são apresentados os experimentos realizados com o método de balanceamento proposto baseado em comitê de classificadores, cujo objetivo é comparar os resultados dos métodos tradicionais com os resultados do algoritmo CCDD.

Para a utilização dos métodos *Boosting* e *Bagging* foram necessários a fixação de alguns parâmetros.

Os experimentos foram realizados na ferramenta *Weka*, pois a mesma já possui pacotes com os métodos *Boosting* e *Bagging*, utilizando o CBA como classificador base.

Primeiramente é preciso definir quantos classificadores serão criados. Após analisar o tamanho da base de dados e realizar diversos experimentos, chegou-se a conclusão que o número ideal de classificadores para formar o comitê é igual a 5. Para utilizar o método CCDD, o número de classificadores a serem criados dependerá do tamanho da base de dados. Na base Supermercado X, o número de classificadores gerados é igual a 3. Logo após a escolha da quantidade de classificadores que irão compor o comitê, é necessário definir a porcentagem de reamostragem da base de dados. Para os métodos *Boosting* e *Bagging* foi realizado a reamostragem de 100% da base de dados.

A classificação das instâncias é realizada utilizando o algoritmo CBA com os mesmos parâmetros informados anteriormente.

A seguir, serão apresentados os resultados da classificação utilizando os métodos baseados em comitê de classificadores, aplicado somente na base Supermercado X.

Os métodos *Boosting* e *Bagging*, não apresentaram resultados para as bases SNP Linear e SNP, por esse motivo não serão apresentados resultados para estas bases. A tabela 6.12 apresenta os resultados da classificação utilizando o métodos *Boosting*, *Bagging* e o algoritmo CCDD.

Tabela 6.12: Métodos de balanceamento

Método	**Base treinamento**	**Base Teste**
Sem balanceamento	71,42%	70,84%
Boosting	72,28%	70,66%
Bagging	71,42%	70,84%
CCDD	57,80%	56,88%

Analisando o resultado do *Boosting* na tabela 6.12 na base de treinamento, obtem-se uma classificação de 72,28 % de instâncias classificadas corretamente, mas quando é analisado a matriz de confusão, nota-se na tabela 6.13 que a classe majoritária obteve maior número de instâncias classificadas, o que explica os 72,28%.

Tabela 6.13: Matriz de Confusão - Base Supermercado X

Matriz de Confusão					
Base Treinamento			**Base teste**		
Sem balanceamento					
a	b		a	b	
0	600	|a = manhã	0	163	|a = manhã
0	1500	|b = tarde	0	396	|b = tarde
Boosting					
a	b		a	b	
67	533	|a = manhã	12	151	|a = manhã
49	1451	|b = tarde	13	383	|b = tarde
Bagging					
a	b		a	b	
0	600	|a = manhã	0	163	|a = manhã
0	1500	|b = tarde	0	396	|b = tarde
CCDD					
a	b		a	b	
445	155	|a = manhã	125	38	|a = manhã
731	769	|b = tarde	204	193	|b = tarde

O *Booting* para a base de treinamento, não conseguiu apresentar resultados satisfatórios para a classe minoritária, em termos de número de instâncias classificadas corretamente como pode ser observado na tabela 6.13. Ainda analisando o *Boosting* na base de teste, tem-se como resultado 12 instâncias classificadas corretamente como manhã, mas apresenta um alto índice de instâncias da manhã sendo classificadas como tarde.

Analisando o resultado do método *Bagging*, pode-se perceber que o mesmo não foi capaz de classificar as instâncias da manhã, classificando todas as instâncias como tarde. O mesmo acontece com a base de teste, onde foram classificadas todas as instâncias da manhã como sendo tarde.

Analisando o algoritmo CCDD para a base de treinamento, obtêm-se 57,8% de instâncias classificadas corretamente, sendo 445 instâncias da manhã e 769 instâncias da tarde. É possível verificar um equilíbrio na classificação das instâncias, o que não ocorro nos demais métodos.

Analisando a base de teste pode-ser perceber que o algoritmo CCDD foi o único método que obteve maior número de instâncias classificadas como manhã.

Com a análise realizada e com os resultados apresentados, pode-se concluir que para essa base de dados, o algoritmo CCDD mostrou-se competitivo e eficiente se comparado com métodos tradicionais baseados em comitê de classificadores.

6.1.5 CONSIDERAÇÕES

As bases de dados Supermercado X, Base SNP somente com efeitos aditivos (linear) e Base SNP com efetivos aditivos e interação apresentavam um desbalanceamento considerável entre as classes. Os métodos apresentaram resultados satisfatórios. É importante destacar que cada método possui uma forma individual de tratamento da base de dados.

O número de instâncias utilizadas para treinar um classificador deve ser levado em consideração, visto que se uma base de dados obtiver um número de instâncias muito pequeno, será construído um bom classificador com a base de treinamento, e um classificador fraco para a base de teste, visto que o número de instâncias da base de treinamento, pode não representar a base de dados como um todo, alterando assim seu resultado.

As tabelas 6.5, 6.11 e 6.8, mostraram que a utilização de uma base de dados desbalanceada pode resultar em uma classificação ruim, cujo nível de predição em termos de instâncias é baixo, uma vez que a classe minoritária é classificada de forma errada.

Em geral o método de balanceamento proposto neste trabalho apresentou-se eficiente, visto que ele conseguiu classificar as instâncias da classe minoritária de forma correta, e competitivo, pois em alguns casos obteve resultados melhores ou próximos aos resultados apresentados pelo demais métodos de balanceamento utilizados neste trabalho.

6.2 SELEÇÃO DE CARACTERÍSTICAS

Esta seção apresenta os experimentos realizados com o modelo de seleção de características proposto e apresentado na seção 4.2, bem como os experimentos realizados com outros métodos de seleção de características. Para essa fase dos experimentos, diversos algoritmos de aprendizagem de máquina foram utilizados.

O objetivo nesta etapa do trabalho é validar o modelo proposto, e verificar se o mesmo é eficaz se comparado com métodos mais tradicionais. Para tanto, três métodos de busca foram utilizados para a realização dos experimentos, tais métodos são: busca evolucionária, busca exaustiva, e o algoritmo *Best First*, combinado com um método de seleção de características *wrapper*.

O classificador base utilizado nesta etapa, é o algoritmo CBA, cujos parâmetros já foram informados anteriormente.

Para a realização dos experimentos foram utilizados três algoritmos do pacote *Weka*:

o *IBK*(k-NN), *Desicion Stump* (árvore de decisão) e o CBA que é o algoritmo utilizado para a classificação em regras de associação.

A justificativa para a utilização do algoritmo CBA nesta etapa dá-se pelo fato de ser um algoritmo de classificação em regras de associação consolidado e bastante estudado para realização de tal tarefa.

As tabelas apresentadas nos próximos experimentos são referentes aos resultados da classificação sem a utilização de métodos de seleção de características e com a utilização de métodos de seleção de características.

A seguir é apresentado a descrição das tabelas:

A coluna Nº de atributos representa a quantidade de atributos contidos na base de dados original, e a quantidade de atributos selecionados após aplicação dos métodos de seleção de características.

A coluna Métodos de seleção representa os algoritmos de seleção de características utilizados para selecionar as características mais relevantes.

A coluna Classificador, representa o algoritmo utilizado para a classificação, tanto para a base de dados sem seleção de características, quanto para a base de dados com seleção de características.

As colunas Base Treinamento e Base Teste indicam os percentuais de acerto obtidos durante a classificação utilizando a base de treinamento e a base de teste.

6.2.1 BASE SUPERMERCADO X

Para a utilizar a Base Supermercado X, as características foram selecionadas na base de treinamento utilizando o SCBR e os demais algoritmos de seleção de características.

Vale ressaltar que o objetivo não é comparar o desempenho de cada algoritmo de classificação, mas sim verificar se as características selecionadas pelo SCBR, apresentam resultados válidos e competitivos se comparado aos métodos tradicionais de seleção.

A tabela 6.14 apresenta o resultado da classificação utilizando o algoritmo IBK como classificador aplicado às bases de treinamento e teste e a tabela 6.15 apresenta a matriz de confusão obtida na classificação utilizando a base de teste.

Tabela 6.14: Base Treinamento - Algoritmo IBK

Nº de atributos	**Métodos de seleção**	**Classificador**	**Base Treinamento**	**Base Teste**
104	Sem seleção	IBK	60%	45,43%
62	*Wrapper + Evolucionary*	IBK	62,69%	45,08%
10	*Wrapper+ Best First*	IBK	60,77%	38,10%
30	SCBR	IBK	59,26%	46,51%

Tabela 6.15: Base Teste - Algoritmo IBK

Algoritmo IBK					
Sem Seleção			***Wrapper + Evolucionary***		
Matriz de Confusão			Matriz de Confusão		
a	b		a	b	
140	23	|a = manhã	130	33	|a = manhã
282	114	|b = tarde	274	122	|b = tarde
Wrapper + Best First			**SCBR**		
Matriz de confusão			Matriz de Confusão		
a	b		a	b	
154	6	|a = manhã	126	37	|a = manhã
337	59	|b = tarde	262	134	|b = tarde

Analisando os resultados do algoritmo *IBK* na tabela 6.14 pode-se perceber que o resultado com maior percentual de classificação é utilizando a base de treinamento contendo as características selecionadas pelo método *Wrapper + Evolucionary*. O mesmo não acontece para a base de teste, visto que apesar de conseguir classificar corretamente o maior número de instâncias da classe minoritária (manhã), apresenta um elevado número de instâncias da classe majoritária (tarde) classificadas de forma incorreta como pode ser observado na tabela 6.15.

A tabela **??** a seguir, apresenta o resultado da classificação utilizando o algoritmo *Decision Stump* (DS) como classificador aplicado às bases de treinamento e teste.

Tabela 6.16: Base Treinamento - Algoritmo *Decision Stump*

Nº de atributos	**Métodos de seleção**	**Classificador**	**Base Treinamento**	**Base Teste**
104	Sem seleção	DS	52,75%	69,23%
62	*Wrapper + Evolucionary*	DS	51,85%	69,23%
10	*Wrapper+ Best First*	DS	57,32%	30,76%
30	SCBR	DS	53,02%	69,23%

Tabela 6.17: Base Teste - Algoritmo *Decision Stump*

Algoritmo *Decision Stump*					
Sem Seleção			***Wrapper + Evolucionary***		
Matriz de Confusão			Matriz de Confusão		
a	b		a	b	
18	145	|a = manhã	18	145	|a = manhã
27	369	|b = tarde	27	369	|b = tarde
Wrapper + Best First			**SCBR**		
Matriz de confusão			Matriz de Confusão		
a	b		a	b	
162	1	|a = manhã	18	145	|a = manhã
386	10	|b = tarde	27	369	|b = tarde

Analisando os resultados da tabela 6.16 tem-se como melhor percentual de classificação utilizando a base contendo as características selecionadas pelo método *Wrapper + Best First*, mas verificando a matriz de confusão da tabela 6.17 é possível perceber um elevado número de instâncias da classe majoritária (manhã) classificadas de forma incorreta na base de teste, apresentando a tendência de classificar instâncias da tarde como sendo instâncias da manhã. O comportamento oposto é observado no resultado da classificação dos demais métodos, onde existe a tendência de classificar instâncias da manhã como sendo instâncias da tarde.

É apresentado a seguir, o resultado da classificação utilizando o algoritmo CBA como classificador aplicado às bases de treinamento e teste.

Tabela 6.18: Base Treinamento - Algoritmo CBA

Nº de atributos	**Métodos de seleção**	**Classificador**	**Base Treinamento**	**Base Teste**
104	Sem seleção	CBA	57,80%	50,62%
62	*Wrapper + Evolucionary*	CBA	61,15%	44,36%
10	*Wrapper+ Best First*	CBA	61,63%	38,28%
30	SCBR	CBA	61,45%	51,16%

Tabela 6.19: Base Teste - Algoritmo CBA

Algoritmo CBA					
Sem Seleção			***Wrapper + Evolucionary***		
Matriz de Confusão			Matriz de Confusão		
a	b		a	b	
117	46	|a = manhã	126	37	|a = manhã
230	166	|b = tarde	274	122	|b = tarde
Wrapper + Best First			**SCBR**		
Matriz de confusão			Matriz de Confusão		
a	b		a	b	
150	13	|a = manhã	118	45	|a = manhã
332	64	|b = tarde	228	168	|b = tarde

Analisando a tabela 6.18 é possível notar que a base contendo as características selecionadas pelo *Wrapper + Best First* apresentou melhor resultado em termos de percentual de acerto na classificação, em relação os demais métodos. Mas analisando a matriz de confusão apresentada na tabela 6.19 pode-se perceber que apesar de apresentar uma maior quantidade de instâncias da classe minoritária (manhã) classificadas de forma correta, apresenta um viés na classificação das instâncias da classe majoritária (tarde) para a base de teste. Esse comportamento, também pode ser observado no resultado da classificação utilizando as bases contendo as características selecionadas pelos demais métodos de seleção.

Para validar o método de seleção de características baseados em regras (SCBR) proposto neste trabalho, foram realizados diversos experimentos utilizando diferentes tipos de reorganização da base de dados. Para tal, foram utilizados os métodos: **Resubstituição**, ou seja, o classificador é avaliado em quão bem ele prediz a classe das instâncias que ele foi treinado, o método **Validação Cruzada**, onde o classificador é avaliado por validação cruzada, dividindo a base de dados em subconjuntos, e o método **Percentual de Divisão** onde o classificador é avaliado o quão bem ele prevê uma determinada porcentagem dos dados que foram utilizados para o teste. (KIRKBY et al., 2007).

A tabela 6.20 representa a realização dos experimentos utilizando a base sem seleção de características e com as bases contendo as características selecionadas pelos métodos de seleção, utilizando o algoritmo IBK como classificador.

Tabela 6.20: Classificação com o IBK

Nº de Atributos	Métodos de seleção	Classificador	Resubstituição	Validação Cruzada	Percentual de Divisão
104	Sem seleção	IBK	60%	52,75%	50,49%
62	*Wrapper + Evolucionary*	IBK	62,69%	57,72%	58,24%
30	*Wrapper + Best First*	IBK	60,77%	59,19%	64,97%
10	SCBR	IBK	59,26%	55,48%	51,16%

Analisando o resultado da tabela 6.20 utilizando o método Resubstituição, é possível perceber um aumento de 2.692% na classificação, em comparação com a base original, utilizando o método de seleção de características *Wrapper + Evolucionary*, é notado também, um aumento de 0,77% na classificação em comparação com a base original utilizando o método *Wrapper + Best First*. Para o método de seleção SCBR não foi observado melhora significativa.

Analisando o método de Validação Cruzada é possível perceber um aumento de 4,97% na classificação utilizando a base contendo as características selecionadas pelo método *Wrapper + Evolucionary* em comparação com o resultado da classificação obtido utilizando a base original. Obteve-se um aumento de 6,4454% na classificação utilizando a base contendo as características selecionadas pelo método *Wrapper + Best First*. Por fim é observado um aumento de 2,73% na classificação utilizando a base contendo as características selecionadas pelo método SCBR.

Partindo para a analise do método *Percentual de Divisão*, é observado um aumento na classificação igual a 7,75% utilizando a base contendo as características selecionadas pelo métodos *Wrapper + Evolucionary*. Para o método *Wrapper + Best First* o aumento na classificação é igual a 14,48%. E por último o método SCBR, obteve um aumento de 0,67% na classificação em comparação com a base original.

Os melhores resultados de classificação apresentados por cada método está destacado de vermelho na tabela 6.20.

A tabela 6.21 a seguir apresenta os resultados da classificação utilizando o algoritmo *Decision Stump* (DS) como classificador.

Tabela 6.21: Classificação com o *Decision Stump*

Nº de Atributos	Métodos de seleção	Classificador	Resubstituição	Validação Cruzada	Percentual de Divisão
104	Sem seleção	DS	52,75%	51,25%	51,47%
62	*Wrapper + Evolucionary*	DS	51,85%	49,32%	48,93%
30	*Wrapper + Best First*	DS	57,32%	55,45%	59,91%
10	SCBR	DS	53,02%	50,48%	50,38%

Analisando o resultado do método de Resubstituição na tabela 6.21 é possível perceber um aumento de 4,57% na classificação utilizando a base de dados contendo as caracterís-

ticas selecionadas pelo método *Wrapper + Best First*. Para a base com as características selecionadas pelo método SCBR obteve-se um aumento de 0,27% na classificação. A base contendo as características selecionadas pelo método *Wrapper + Evolucionary* não apresentou resultado satisfatório na classificação em comparação com a base original.

Analisando o método de *Validação Cruzada* é possível observar um aumento de 4,2% na classificação utilizando a base contendo as características selecionadas pelo método *Wrapper + Best First*. Os demais métodos não apresentaram melhora significativa no resultado da classificação.

Para o método Percentual de Divisão, obteve-se um aumento de 8,445% utilizando a base com as características selecionadas pelo método *Wrapper + Best First*. Os demais métodos não apresentaram resultados satisfatórios de classificação em comparação com a base original.

Os melhores resultados de classificação apresentados por cada método está destacado de vermelho na tabela 6.21.

A tabela a seguir é apresenta os resultados da classificação utilizando o algoritmo CBA como classificador.

Tabela 6.22: Classificação com o CBA

Nº de Atributos	**Métodos de seleção**	**Classificador**	**Resubstituição**	**Validação Cruzada**	**Percentual de Divisão**
104	Sem seleção	CBA	57,80%	57,16%	54,16%
62	*Wrapper + Evolucionary*	CBA	61,15%	57,81%	49,73%
30	*Wrapper + Best First*	CBA	61,63%	60,48%	64,97%
10	SCBR	CBA	61,45%	58,56%	54,52%

Analisando o método de Resubstituição na tabela 6.22, percebe-se um aumento de 3,35% na classificação utilizando a base de dados contendo as características selecionadas pelo método *Wrapper + Evolucionary*. Para a classificação contendo as características selecionadas pelo método *Wrapper + Best First* o aumento é de 3,83%. Finalizando a análise do método de Resubstituição, o método SCBR apresenta um aumento de 3,65% na classificação em comparação com a base de dados original, ou seja, sem seleção de características.

Partindo para a analise do método de Validação Cruzada a classificação é apresentado um aumento de 0,65 % na classificação utilizando a base contendo as características selecionadas pelo método *Wrapper + Evolucionary*. Para a base contendo as características selecionadas pelo método *Wrapper + Best First* o aumento é de 3,32% na classificação. E por último a utilização da base contendo as características selecionadas pelo método

SCBR apresenta um aumento de 1,40% na classificação.

Analisando o método de Percentual de Divisão juntamente com a base contendo as características selecionadas pelo método *Wrapper + Evolucionary* é possível perceber que não houve melhora significativa na classificação. Porém, obtêm-se um aumento de 10,81% no aumento da classificação utilizando a base contendo as características selecionadas pelo método *Wrapper + Best First*. Finalizando as análises da tabela 6.22 é notado um aumento de 0,36% na classificação utilizando a base contendo as características selecionadas pelo método SCBR.

A seguir são apresentados os resultados da classificação da base Supermarket.

6.2.2 BASE *SUPERMARKET*

Os experimentos realizados em (Tiwari e Singh, TIWARI; SINGH) foram utilizados como referência por utilizar a base de dados Supermarket juntamente com o algoritmo IBK em seus experimentos. O diferencial deste trabalho é a utilização de três algoritmos de classificação sendo o *Decision Stump* e o CBA além do IBK, e a realização da seleção de características utilizando os métodos *Wrapper + Best First* e o SCBR. O método *Wrapper + Evolucionary* não selecionou nenhuma características para esse base, por esse motivo não foi possível realizar a classificação.

A base original contém 217 características. O método de seleção de características *Wrapper + Best First* selecionou apenas 8 características como sendo as mais representativas, enquanto o método SCBR selecionou 22 características. Após a realização de vários experimentos variando os valores de suporte, confiança e número de regras, chegou-se aos parâmetros que apresentaram melhor desempenho. Nesta fase tem-se como parâmetro do algoritmo CBA com suporte igual a 0,01, confiança igual 0,8 e o número de regra sendo igual a 250.

São apresentados os experimentos utilizando os métodos de Resubstituição, Validação Cruzada e Percentual de Divisão aplicados a base sem seleção de características e as bases contendo as características selecionadas pelos métodos de seleção informados anteriormente.

A tabela a seguir apresenta os resultados da classificação utilizando como classificador o algoritmo IBK.

Tabela 6.23: Classificação com o IBK

Nº de Atributos	Métodos de seleção	Classificador	Resubstituição	Validação Cruzada	Percentual de Divisão
217	Sem seleção	IBK	37,43%	36,80%	37,57%
8	*Wrapper + Best First*	IBK	68,40%	67,97%	67,45%
22	SCBR	IBK	38,10%	28,12%	38,46%

Analisando a tabela 6.23 e o método de Resubstituição, pode-se perceber um aumento considerável de 30,97% na classificação utilizando a base contendo as características selecionadas pelo método *Wrapper + Best First*. Para a base contendo as características selecionadas pelo método SCBR o aumento é de 0,66% em comparação com o resultado da base sem seleção seleção de características.

Realizando a análise do método de Validação Cruzada na tabela 6.23, obtêm-se um aumento de 31,17% na classificação em comparação com a base original, utilizando a base contendo as características selecionadas pelo método *Wrapper + Best First*. Não é observado aumento na classificação utilizando a base contendo as características selecionadas pelo SBCR.

Por fim, analisando os resultados apresentados pelo método de Percentual de Divisão na tabela 6.23, tem-se um aumento de 29,88% na classificação utilizando a base contendo as características selecionadas pelo método *Wrapper + Best First*. Já para a base contendo as características selecionadas pelo método SBCR o aumento é de 0,89% na classificação em comparação com a base sem seleção de características.

Os melhores resultados de cada método são apresentados em vermelhos na tabela 6.23.

A seguir são apresentados os resultados da classificação utilizando o algoritmo *Decision Stump*(DS) como classificador.

Tabela 6.24: Classificação com o *Decision Stump*

Nº de Atributos	Métodos de seleção	Classificador	Resubstituição	Validação Cruzada	Percentual de Divisão
217	Sem seleção	DS	65,11%	64,40%	65,28%
8	*Wrapper + Best First*	DS	63,71%	63,71%	66,62%
22	SCBR	DS	65,11%	64,40%	65,28%

Analisando os resultados dos métodos de Resubstituição e Validação Cruzada na tabela 6.24 é possível perceber que não houve aumento nos percentuais de classificação utilizando as bases contendo as características selecionadas pelos métodos de seleção.

Apenas o método de Percentual de Divisão aplicado a base contendo as características selecionadas pelo método *Wrapper + Best First* apresentou aumento de 1,35% na classificação em comparação com a base sem seleção de características.

A seguir são apresentados os resultados da classificação utilizando o algoritmo CBA

como classificador.

Tabela 6.25: Classificação com o CBA

Nº de Atributos	Métodos de seleção	Classificador	Resubstituição	Validação Cruzada	Percentual de Divisão
217	Sem seleção	CBA	78,64%	———	74,50%
8	*Wrapper + Best First*	CBA	68,66%	67,99%	67,57%
22	SCBR	CBA	78,15%	75,23%	73,55%

Primeiramente é importante destacar que não foi possível obter os resultados da classificação usando o algoritmo CBA juntamente com o método de Validação Cruzada aplicados a base de dados original por questões de custo computacional, uma vez que a base possui um número considerável de instâncias. Isso fez com que o processamento fosse computacionalmente inviável, impossibilitando assim a geração dos resultados.

Analisando os resultados dos métodos de Resubstituição e de Percentual de Divisão na tabela 6.25, pode-se perceber que não houve aumento no percentual de classificação utilizando as bases contendo as características selecionadas pelos métodos de seleção, em comparação com o resultado obtido utilizando a base sem seleção para realizar a classificação.

Finalizando as analises dos resultado apresentados na tabela 6.22, a base contendo as características selecionadas pelo método SCBR apresentou melhor resultado em comparação com o método *Wrapper + Best First*.

A seguir, serão apresentados os resultados dos experimentos realizados na Base CH.

6.2.3 BASE CH

A Base CH contém 36 atributos, dos 36 atributos que compõem a base de dados, 12 atributos foram selecionados pelo método SBCR e 5 atributos foram selecionados pelo método *Wrapper + Best First*.

As tabelas a seguir apresentam os resultados dos experimentos realizados com os algoritmos de classificação CBA, *IBK* e *Decision Stump*, aplicados às bases sem seleção de características e com seleção de características.

É importante informar que o método *Wrapper+ Evolucionary* não selecionou nenhuma características para esta base, por esse motivo, não foram apresentados os resultados utilizando tal método de seleção de características.

Tabela 6.26: Classificação com o IBK

Nº de Atributos	Métodos de seleção	Classificador	Resubstituição	Validação Cruzada	Percentual de Divisão
36	Sem seleção	IBK	98,77%	96,49%	95,03%
5	*Wrapper + Best First*	IBK	94,33%	94,33%	94,83%
12	SCBR	IBK	82,75%	82,50%	82,15%

Analisando os resultados apresentados pelo método de Resubstituição na tabela 6.26, pode-se perceber que nenhum dos métodos de seleção de características apresentou resultado satisfatório se comparado com a base de dados original. O mesmo comportamento pode ser observado nos métodos de Validação Cruzada e Percentual de Divisão.

A seguir são apresentados os resultados utilizando o algoritmo *Decision Stump* (DS) como classificador.

Tabela 6.27: Classifcação com o *Decision Stump*

Nº de Atributos	Métodos de seleção	Classificador	Resubstituição	Validação Cruzada	Percentual de Divisão
36	Sem seleção	DS	66,05%	66,05%	66,69%
5	*Wrapper + Best First*	DS	66,05%	63,05%	66,69%
12	SCBR	DS	66,05%	66,05%	66,69%

Analisando os resultados de classificação utilizando o algoritmo *Decision Stump* na tabela 6.27, pode-se perceber que o comportamento se manteve estável tanto na classificação utilizando as características selecionadas pelos métodos de seleção, quanto para a classificação utilizando a base original, ou seja, não obteve alteração nos resultados.

A seguir são apresentados os resultados da classificação utilizando a algoritmo CBA como classificador.

Tabela 6.28: Classificação com o CBA

Nº de Atributos	Métodos de seleção	Classificador	Resubstituição	Validação Cruzada	Percentual de Divisão
36	Sem seleção	CBA	60,04%	60,07%	59,24%
5	*Wrapper + Best First*	CBA	94,33%	94,33%	93,83%
12	SCBR	CBA	79,34%	79,72%	77,73%

Analisando os resultados apresentados na tabela 6.28 para o método de Resubstituição, é possível perceber um aumento de 34,29% no percentual de classificação utilizando a base contando as características selecionadas pelo método *Wrapper + Best First*. Para a classificação da base contendo as características selecionadas pelo método SCBR o aumento é de 19,30% em relação com a base de dados sem seleção de características.

Analisando os resultados do método de Validação Cruzada na tabela 6.28, tem-se um aumento de 33,63% na classificação utilizando a base de dados contendo as características selecionadas pelo método *Wrapper + Best First* em comparação com a base original. A

classificação contendo as características selecionadas pelo método SCBR apresentou um aumento de 19,65% em comparação com a base original.

Finalizando as análises da tabela 6.28 e o método de Percentual de Divisão, é possível perceber que a base contendo as características selecionadas pelo método Wrapper + Best First apresenta um aumento de 34,59% no percentual de classificação em comparação com o resultado da base sem seleção de características, e para a classificação da base contendo as características selecionadas pelo método SCBR, o aumento é de 18,49% em relação com a base de dados sem seleção.

Para a base de dados CH o SBCR, apesar de não ser o método que apresentou os melhores resultados, os resultados de tal método mostraram-se competitivos em comparação com os resultados de classificação da base original.

A seguir, serão apresentados os resultados realizados com a base de dados *Mushroom*, aplicando os métodos de seleção de características, e utilizando os três algoritmos para realização a classificação.

6.2.4 BASE *MUSHROOM*

A *Base Mushroom* contém 23 atributos, dos 23 atributos, 10 foram selecionados pelo SCBR, 5 selecionados pelo método *Wrapper + Best First* e 4 foram selecionados pelo método *Wrapper + Evolucionary*.

As tabelas a seguir apresentam os resultados de classificação, utilizando a base original juntamente com os três algoritmos de classificação, e as bases contendo as características selecionadas pelos métodos de seleção de características.

Tabela 6.29: Classificação com o IBK

Nº de Atributos	**Métodos de seleção**	**Classificador**	**Resubstituição**	**Validação Cruzada**	**Percentual de Divisão**
23	Sem seleção	IBK	100%	100%	100%
4	*Wrapper + Evolucionary*	IBK	99,21%	99,21%	99,23%
5	*Wrapper + Best First*	IBK	99,06%	98,95%	99,03%
10	SCBR	IBK	99,80%	99,80%	99,85%

Analisando os resultados da tabela 6.29 e o método de Resubstituição, é possível perceber que não houve nenhuma melhora nos resultados de classificação utilizando as bases contendo as características selecionadas pelos métodos de seleção, em comparação com a base de dados sem seleção, porém vale ressaltar que a base contendo as características selecionadas pelo método SCBR apresentou melhor resultado de classificação se comparado com os resultados apresentados pelos demais métodos de seleção.

A tabela a seguir apresenta o resultado da classificação utilizando o algoritmo *Decision Stump* (DS) como classificador.

Tabela 6.30: Classificação com o *Decision Stump*

Nº de Atributos	**Métodos de seleção**	**Classificador**	**Resubstituição**	**Validação Cruzada**	**Percentual de Divisão**
23	Sem seleção	DS	88,67%	88,67%	89,21%
4	*Wrapper + Evolucionary*	DS	88,67%	88,67%	89,21%
5	*Wrapper + Best First*	DS	88,67%	88,67%	89,21%
10	SCBR	DS	88,67%	88,67%	89,21%

Os resultados de classificação apresentados na tabela 6.30 tanto para os métodos de reorganização da base quanto para os métodos de seleção de características apresentaram-se os mesmos em comparação com o resultado da classificação da base sem seleção de características, sem apresentar nem perda e nem aumento na classificação.

A seguir são apresentados os resultados utilizando o algoritmo CBA para realizar a classificação.

Tabela 6.31: Classificação com o CBA

Nº de Atributos	**Métodos de seleção**	**Classificador**	**Resubstituição**	**Validação Cruzada**	**Percentual de Divisão**
23	Sem seleção	CBA	97,83%	92,51%	93,37%
4	*Wrapper + Evolucionary*	CBA	98,81%	99,21%	99,23%
5	*Wrapper + Best First*	CBA	99,01%	98,97%	98,95%
10	SCBR	CBA	99,01%	99,01%	98,95%

Analisando os resultados da tabela 6.31 e o método de Resubstituição é possível notar um aumento de 0,98% no percentual de classificação utilizando a base contendo as características selecionadas pelo método *Wrapper + Evolucionary* e, um aumento de 1,18% na classificação utilizando a base contendo as características selecionadas pelas métodos *Wrapper + Best First* e SCBR.

Analisando os resultados do método de Validação Cruzada na tabela 6.31, obtêm-se um aumento de 6,70% na classificação utilizando a base contendo as características selecionadas pelo método *Wrapper + Evolucionary* e um aumento na classificação 6,46% utilizando a base com as características selecionas pelo método *Wrapper + Best First*. Já para a classificação utilizando a base com as características selecionadas pelo método SCBR, o aumento é de 6,50% na classificação.

Por fim, analisando os resultados apresentados pelo método de Percentual de Divisão, é possível observar um aumento de 5,86% do percentual de classificação utilizando a base com as características selecionadas pelo método *Wrapper + Evolucionary*. Para a classificação da base com as características selecionadas pelo método *Wrapper + Besti First* e pelo método SCBR o aumento é de 5,58%.

Os melhores resultados estão apresentados de vermelhos na tabela 6.31.

Finalizadas as análises é apresentado no próximo capítulo a conclusão deste trabalho.

7 CONCLUSÕES

Foca-se, neste trabalho, em estudar, avaliar e desenvolver ferramentas de aprendizagem supervisionada para um tipo de dado específico conhecido como bases de dados transacionais.

Envolveu-se, em uma primeira etapa nas avaliações a respeito do uso de tais regras no sentido de determinar seu desempenho quando associada a problemas de classificação. Destaca-se, como conclusão desta etapa, que o potencial relacionado à predição obtida está diretamente relacionado à qualidade das regras envolvidas.

Na expectativa de se gerar regras de associação de qualidade, que entende-se está diretamente relacionada ao nível de predição a ser obtido, desenvolveu-se um modelo visando diminuir os efeitos negativos provenientes de bases desbalanceadas. Assim, um modelo baseado em comitê de classificadores, devidamente balanceados, foi apresentado. Destaca-se, como vantagem do método um uso menos aleatório dos dados durante o processo de balanceamento, tanto da classe minoritária quanto da classe majoritária.

Experimentos em diversas bases de dados foram realizados, ressaltando-se que o classificador CBA sempre foi a referência adotada para geração das regras. Os resultados obtidos em alguns casos, indicaram um acréscimo de desempenho nas predições quando o balanceamento em comitê foi adotado.

Métodos de seleção de itens ou atributos baseados em regras de associação são pouco comuns na literatura, tanto para modelos em filtro quanto encapsulados. Novamente regras não confiáveis ou pouco representativas tendem a distorcer as variáveis selecionadas. Optou-se, neste trabalho, por apresentar uma abordagem de seleção de características baseadas em regras de associação, numa tentativa de aprofundar o entendimento da manipulação das regras para esse fim, de forma a obter um nível de seleção considerado eficiente.

Novamente, testes computacionais apresentam o indicativo da viabilidade do modelo, obtendo resultados bastante competitivos, indicando que pode-se considerar como promissora tal estratégia.

Em uma análise final, é importante ressaltar que os desenvolvimentos apresentados devem ser vistos como processo acoplado, onde o balaceamento influencia na qualidade

das regras de associação geradas. As regras utilizadas são relevantes para a qualidade dos itens selecionados que irão ser responsáveis pelo nível de predição de uma posterior classificação.

Diversas são as questões que ainda necessitam de pesquisas adicionais para melhor entendimento do uso de regras de associação em aprendizagem supervisionada. Algumas são mais diretas e de fácil avaliação, como verificar a importância do algoritmo CBA no processo de geração de regras por meio de comparações com outros modelos, avaliar a influência do nível de desbalanceamento das bases na qualidade dos resultados, verificar possíveis relações entre atributos selecionados e a frequência dos mesmos na base, entre outras.

Desafios maiores, também estão previsto de serem tratados em trabalhos futuros. O desenvolvimento de um novo modelo para geração de regras, que utilize métricas mais efetivas em relação às medidas padrões de suporte e confiança para este tipo de aplicação, é de grande interesse. Não se pode afirmar que a geração de instâncias artificiais para bases transacionais é eficiente. Dificuldades relacionadas ao padrão destas bases tornam árdua esta tarefa. Estudos e avaliações nesta linha podem abrir novos campos para desenvolvimentos dos modelos de balanceamento.

REFERÊNCIAS

AGRAWAL, R.; IMIELIŃSKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. In: ACM. **ACM SIGMOD Record**, 1993. v. 22, n. 2, p. 207–216.

AGRAWAL, R.; MANNILA, H.; SRIKANT, R.; TOIVONEN, H.; VERKAMO, A. I. et al. Fast discovery of association rules. **Advances in knowledge discovery and data mining**, AAAI/MIT Press Menlo Park, CA, v. 12, n. 1, p. 307–328, 1996.

ALBERTS, B.; JOHNSON, A.; LEWIS, J.; RAFF, M.; ROBERTS, K.; WALTER, P. **Biologia molecular da célula**, 2009.

ALVES, A. S. **Regras de Associação e Classificação em Ambiente de Computação Paralela Aplicadas a Sistemas Militares**. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2007.

BERRY, M. J.; LINOFF, G. S. **Data mining techniques: for marketing, sales, and customer relationship management**, 2004.

BREIMAN, L. Bagging predictors. **Machine learning**, Springer, v. 24, n. 2, p. 123–140, 1996.

BRIN, S.; MOTWANI, R.; ULLMAN, J. D.; TSUR, S. Dynamic itemset counting and implication rules for market basket data. In: ACM. **ACM SIGMOD Record**, 1997. v. 26, n. 2, p. 255–264.

BROOKES, A. J. The essence of snps. **Gene**, Elsevier, v. 234, n. 2, p. 177–186, 1999.

CABENA, P.; HADJINIAN, P.; STADLER, R.; VERHEES, J.; ZANASI, A. **Discovering data mining: from concept to implementation**, 1998.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. Smote: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, p. 321–357, 2002.

CHAWLA, S. Feature selection, association rules network and theory building. In: **FSDM**, 2010. p. 14–21.

CHEN, G.; LIU, H.; YU, L.; WEI, Q.; ZHANG, X. A new approach to classification based on association rule mining. **Decision Support Systems**, Elsevier, v. 42, n. 2, p. 674–689, 2006.

CHEN, M.-S.; HAN, J.; YU, P. S. Data mining: an overview from a database perspective. **Knowledge and data Engineering, IEEE Transactions on**, IEEE, v. 8, n. 6, p. 866–883, 1996.

DASH, M.; LIU, H. Feature selection for classification. **Intelligent data analysis**, Elsevier, v. 1, n. 1, p. 131–156, 1997.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37, 1996.

FREITAS, A. A. **Data mining and knowledge discovery with evolutionary algorithms**, 2013.

FREUND, Y.; MASON, L. The alternating decision tree learning algorithm. In: **icml**, 1999. v. 99, p. 124–133.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, Elsevier, v. 55, n. 1, p. 119–139, 1997.

HAN, J.; KAMBER, M.; PEI, J. **Data mining: concepts and techniques: concepts and techniques**, 2011.

HAN, J.; PEI, J.; YIN, Y. Mining frequent patterns without candidate generation. In: ACM. **ACM SIGMOD Record**, 2000. v. 29, n. 2, p. 1–12.

HAND, D. J.; MANNILA, H.; SMYTH, P. **Principles of data mining**, 2001.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J.; FRANKLIN, J. The elements of statistical learning: data mining, inference and prediction. **The Mathematical Intelligencer**, Springer, v. 27, n. 2, p. 83–85, 2005.

HORTA, R. A. M. **Uma metodologia de Mineraçao de Dados para a previsao de insolvência de empresas brasileiras de capital aberto**. Tese (Doutorado) — Dissertaçao (Doutorado em Engenharia Civil)-Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010.

HU, K.; LU, Y.; ZHOU, L.; SHI, C. Integrating classification and association rule mining: A concept lattice framework. In: **New Directions in Rough Sets, Data Mining, and Granular-Soft Computing**, 1999. p. 443–447.

JOHN, G. H.; KOHAVI, R.; PFLEGER, K. et al. Irrelevant features and the subset selection problem. In: **Machine Learning: Proceedings of the Eleventh International Conference**, 1994. p. 121–129.

KIRKBY, R.; FRANK, E.; REUTEMANN, P. Weka explorer user guide for version 3-5-6. 2007.

KOSTERS, W. A.; MARCHIORI, E.; OERLEMANS, A. A. Mining clusters with association rules. In: **Advances in Intelligent Data Analysis**, 1999. p. 39–50.

LAROSE, D. T. **Discovering Knowledge in Data: An Introduction to Data Mining**, 2005.

LI, W.; HAN, J.; PEI, J. Cmar: Accurate and efficient classification based on multiple class-association rules. In: IEEE. **Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on**, 2001. p. 369–376.

LING, W.; HUI, G. Feature selection based on fuzzy clustering analysis and association rule mining for soft-sensor. In: IEEE. **Control Conference (CCC), 2014 33rd Chinese**, 2014. p. 5162–5166.

LIU, B. **Web data mining: exploring hyperlinks, contents, and usage data**, 2007.

LIU, B.; MA, Y.; WONG, C.-K. Classification using association rules: weaknesses and enhancements. In: **Data mining for scientific and engineering applications**, 2001. p. 591–605.

LIU, X.-Y.; WU, J.; ZHOU, Z.-H. Exploratory undersampling for class-imbalance learning. **Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on**, IEEE, v. 39, n. 2, p. 539–550, 2009.

MA, B. L. W. H. Y. Integrating classification and association rule mining. In: **Proceedings of the 4th**, 1998.

MOTTA, C. G. L. da. **Metodologia para Mineração de Regras de Associação Multiníveis incluindo Pré e Pós-Processamento**. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2010.

MUELLER, A. Fast sequential and parallel algorithms for association rule mining: A comparison. 1998.

OHNO, A. **Detecção de Mudanças em Problemas de Classificação a partir de Classificadores Sociais**. Tese (Doutorado) — Pontifícia Universidade Católica do Paraná, 2011.

OLIVEIRA, F. C. d. **Um método para seleção de atributos em dados genômicos**. Tese (Doutorado) — Universidade Federal de Juiz de Fora, 2015.

PIERCE, B. A. **Genetics: a conceptual approachGenética: un enfoque conceptual.**, 2010.

QAZI, N.; RAZA, K. Effect of feature selection, smote and under sampling on class imbalance classification. In: **Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on**, 2012. p. 145–150.

RAJESWARI, K. Feature selection by mining optimized association rules based on apriori algorithm. **International Journal of Computer Applications**, Foundation of Computer Science, v. 119, n. 20, 2015.

SAVASERE, A.; OMIECINSKI, E. R.; NAVATHE, S. B. An efficient algorithm for mining association rules in large databases. Georgia Institute of Technology, 1995.

SCHAPIRE, R. E. The strength of weak learnability. **Machine learning**, Springer, v. 5, n. 2, p. 197–227, 1990.

SRIVASTAVA, J.; COOLEY, R.; DESHPANDE, M.; TAN, P.-N. Web usage mining: Discovery and applications of usage patterns from web data. **ACM SIGKDD Explorations Newsletter**, ACM, v. 1, n. 2, p. 12–23, 2000.

TIWARI, M.; SINGH, R. A benchmark to select classification algorithms for decision support systems. Citeseer.

WEISS, G. M. Mining with rarity: a unifying framework. **ACM SIGKDD Explorations Newsletter**, ACM, v. 6, n. 1, p. 7–19, 2004.

XIE, J.; WU, J.; QIAN, Q. Feature selection algorithm based on association rules mining method. In: IEEE. **Computer and Information Science, 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on**, 2009. p. 357–362.

YIN, X.; HAN, J. Cpar: Classification based on predictive association rules. In: SIAM. **SDM**, 2003. v. 3, p. 369–376.

ZIMMERMANN, A.; RAEDT, L. D. Corclass: Correlated association rule mining for classification. In: SPRINGER. **Discovery Science**, 2004. p. 60–72.