CLASSIFICAÇÃO DE LINFOMAS UTILIZANDO UMA ABORDAGEM BASEADA EM ÁRVORES DE DECISÃO

Laura de Oliveira Fernandes Moraes

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Carlos Eduardo Pedreira

Rio de Janeiro
Fevereiro de 2016

CLASSIFICAÇÃO DE LINFOMAS UTILIZANDO UMA ABORDAGEM BASEADA EM ÁRVORES DE DECISÃO

Laura de Oliveira Fernandes Moraes

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Valmir Carneiro Barbosa, Ph.D.

Prof. Geraldo Bonorino Xexéo, D.Sc.

Prof. José Alberto Orfao de Matos Correia e Vale, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
FEVEREIRO DE 2016

Moraes, Laura de Oliveira Fernandes

Classificação de linfomas utilizando uma abordagem baseada em árvores de decisão/Laura de Oliveira Fernandes Moraes. – Rio de Janeiro: UFRJ/COPPE, 2016.

IX, 49 p.: il.; 29, 7cm.

Orientador: Carlos Eduardo Pedreira

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2016.

Referências Bibliográficas: p. 46 – 48.

1. Árvore de Decisão. 2. Classificação Supervisionada. 3. Citometria de Fluxo. I. Pedreira, Carlos Eduardo. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CLASSIFICAÇÃO DE LINFOMAS UTILIZANDO UMA ABORDAGEM BASEADA EM ÁRVORES DE DECISÃO

Laura de Oliveira Fernandes Moraes

Fevereiro/2016

Orientador: Carlos Eduardo Pedreira

Programa: Engenharia de Sistemas e Computação

Esta dissertação apresenta uma metodologia de apoio ao diagnóstico de pacientes com linfomas. É proposta uma solução através de árvores de decisão para o diagnóstico diferencial de linfomas a partir de dados de citometria de fluxo. Por diagnóstico diferencial, entende-se escolher ou eliminar linfomas dentre alguns tipos pré-definidos. Os citômetros de fluxos são aparelhos usados, entre outras funções, no diagnóstico de linfomas. Permitem medir diversos parâmetros de milhares de células simultaneamente, gerando informações individualizadas de cada uma destas células a partir de uma amostra de sangue periférico ou de medula óssea colhida de um paciente. Com o método proposto se constrói uma árvore utilizando um modelo de regressão logística regularizada com o algoritmo do Lasso em cada nó. Analisa-se a combinação de múltiplas variáveis em cada passo a fim de escolher ou eliminar classes, implementando o diagnóstico diferencial. O modelo em árvore foi escolhido por sua interpretabilidade, deixando a decisão do diagnóstico a um especialista humano. Um requisito essencial desse projeto é evitar falsos negativos. Para atingir esse objetivo, dois métodos foram acoplados à árvore para verificar o resultado antes de informá-lo ao tomador de decisão. Esses métodos agregam ao resultado dado pela árvore, classes extras que devem ser consideradas para o diagnóstico. Os atributos do modelo são medidas feitas a partir de anticorpos monoclonais que tem custo elevado. Dessa forma, torna-se importante pela motivação de custo e também pela qualidade do resultado de classificação de padrões, que os atributos possam ser usados de forma parcial. Para analisar os resultados fora da amostra foram utilizadas técnicas de validação cruzada, *leave-one-out* e um conjunto de teste com observações não vistas no treinamento. Os experimentos indicaram que essa abordagem produz os resultados compatíveis com os esperados pelos especialistas.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

LYMPHOMA CLASSIFICATION USING A DECISION TREE APPROACH

Laura de Oliveira Fernandes Moraes

February/2016

Advisor: Carlos Eduardo Pedreira

Department: Systems Engineering and Computer Science

This work presents a methodology to support lymphoma diagnosis. A decision tree approach is proposed to perform a lymphoma differential diagnosis using flow cytometry data. Differential diagnosis is to select or eliminate lymphomas among some possible predefined categories. Flow cytometers are equipments used to diagnose lymphoma among other diseases. It is possible to monitor thousands of cells and their parameters simultaneously. It generates individualized information about each cell using patient's peripheral blood or bone marrow sample. The proposed method builds the tree using in each node a regularized logistic regression algorithm called Lasso. It analyzes multiple combinations of attributes in each decision step in order to choose or eliminate classes for the differential diagnosis. The tree approach was chosen because the final decision is made by a human specialist and its representation is easy to understand. A key project requirement is to avoid false negatives. To accomplish this goal, two procedures were added while constructing the tree. Their objective is to double check the outcome and add extra possible outcomes to the diagnosis given to the decision maker. The parameters monitored in the cytometer and used as attributes in the predictor model are actually monoclonal antibodies with an associated high cost. Therefore, motivated by the antibodies cost and also the result quality, another research in this dissertation is to analyze if these antibodies can be used in a partial way. Cross-validation and leave-one-out techniques and a test dataset containing observations not used in training were used to validate the model. The results given by this approach are compatible with those expected by the specialists.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A citometria de fluxo é o método usado rotineiramente para caracterização e diagnóstico de HIV e de doenças neoplásicas: leucemias, linfomas e tumores sólidos, influenciando na escolha do tratamento aplicado ao paciente (1) (2).

O citômetro de fluxo é um equipamento que tem a capacidade de medir e analisar simultaneamente diversas características individuais de milhares de células por segundo. Originalmente concebido como um instrumento de pesquisa, passou, com o aprimoramento da qualidade dos lasers, sistemas ópticos, eletrônicos e fluídicos, a apoiar diagnósticos clínicos rotineiramente. A resolução das visualizações foram melhoradas e o custo de fabricação, manutenção e operação dos citômetros e seus componentes diminuído, aumentando o leque de abrangência de suas aplicações (3). Na última década, ficou evidente a necessidade de se criar novas ferramentas e algoritmos para a representação, visualização e análise inteligente e automatizada dos dados, de maneira a conseguir extrair informação do montante processado. Esse movimento de aplicação de técnicas de informática à citometria, ganhou muita força nos últimos anos e segue com muitas questões em aberto (4) (5) (6).

Trazendo a questão para o ponto de vista de classificação de padrões, pode-se considerar cada uma das medidas do citômetros (20 a 30, tipicamente) como uma dimensão. Cada célula de um determinado paciente seria então um ponto em $\mathbb{R}^n$, onde n representa a quantidade de atributos medidos pelo citômetro. As células de uma determinada doença se encontram normalmente em uma mesma região do espaço, podendo-se fazer uma relação entre a caracterização de doenças e um problema de classificação de padrões em $\mathbb{R}^n$ (2) (6).

1.1 Citometria de fluxo

A tecnologia dos citômetros de fluxo evoluiu bastante ao longo dos anos, a seguir descrevemos muito brevemente alguns de seus fundamentos básicos. Primeiramente, é construída uma suspensão celular com anticorpos monoclonais conjugados a

substâncias fluorescentes (fluorocromos). Esta suspensão é submetida a uma pressão negativa, fazendo com que as células sejam enfileiradas em um tubo de espessura muito pequena (capilar) para entrar no citômetro. Dentro dele, as células são submetidas a feixes de laser. Há dois sensores físicos: o *Forward Scatter* (ou FSC), na linha do feixe de laser e o *Side Scatter* (ou SSC), perpendicular ao feixe. O FSC e SSC medem a dispersão da luz e estão relacionados a propriedades físicas da célula. O FSC está relacionado ao tamanho celular, enquanto o SSC com a granularidade da célula e sua complexidade interna como a forma do núcleo e rugosidade da membrana. Além disso, há lasers e detectores de diferentes comprimentos de onda para medir a fluorescência da célula frente a cada anticorpo. A intensidade de fluorescência emitida é medida para identificar suas características fenotípicas e é proporcional à excitação dos fluorocromos, por sua vez, proporcional à quantidade de locais de ligação entre os anticorpos e seus respectivos antígenos (3) (4). Todos os detectores geram sinais elétricos proporcionais às suas medidas. O esquemático do citômetro pode ser visto na Figura 1.1.

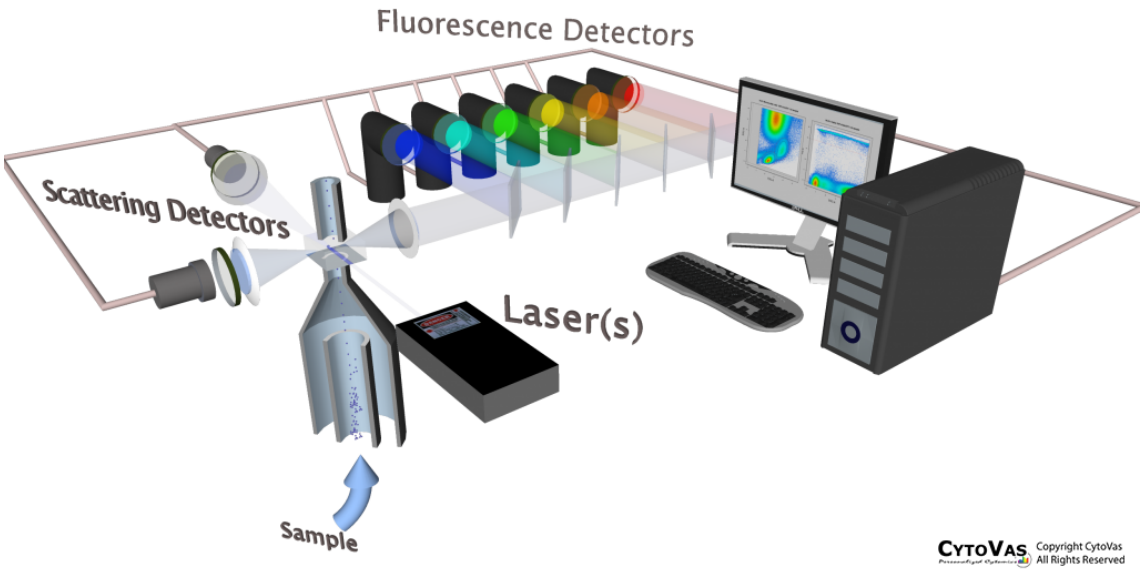


Figura 1.1: Esquemático de um citômetro. Um feixe de luz é direcionado às partículas em meio líquido e os detectores medem sua dispersão e fluorescência. Os sinais elétricos gerados são passados ao computador.

1.2 Objetivo

O objetivo deste trabalho é gerar e testar um esquema de apoio ao diagnóstico de linfomas através de um modelo de classificação supervisionada, utilizando dados disponibilizado pelo citômetro. Dentro desta proposta, além do resultado, as decisões ao longo do processo de classificação devem poder ser compreendidas por um ser

humano, para que o diagnóstico final possa incluir também a experiência e o olhar crítico do profissional que assina o laudo. Nesta proposta, realiza-se a classificação em etapas, escolhendo ou eliminando um conjunto de doenças a cada passo, e assim, oferecendo transparência na interpretabilidade do modelo.

Conforme apurado com especialistas em diagnóstico de linfomas, um requisito essencial que precisava ser introduzido neste projeto é a não eliminação precoce de uma doença do conjunto de linfomas possíveis. É sempre preferível abrir a possibilidade de diagnóstico com mais de uma doença como resultado final, do que correr o risco de eliminar equivocadamente o tipo de linfoma verdadeiro. Isto é, é fundamental procurar-se evitar falsos negativos.

O exame por citometria de fluxo é rotineiramente feito em diversas etapas. A cada etapa um conjunto de anticorpos (marcadores) é utilizado. Esses anticorpos monoclonais são custosos. A utilização de menos marcadores no diagnóstico pode ajudar a baratear o custo do exame. Além disso, buscou-se emular a "estrutura por tubos" que é rotineiramente utilizada. Essa questão será melhor detalhada nos próximos capítulos.

Vale ressaltar uma dificuldade intrínseca ao problema. O banco de dados aqui utilizado é um banco cedido pelo consócio *EuroFlow* (7), agregando dados de diversos grupos europeus. Ainda assim, devido a raridade da doença, o banco é composto do que em geral, em computação, seria considerado "de poucas observações". São 24 medidas gerando uma relativa alta dimensionalidade para os aproximadamente 300 pacientes disponíveis. Portanto, investiga-se um método que seja capaz de criar um modelo de generalização mesmo quando se tem poucas observações.

1.3 Resumo das contribuições

As principais contribuições desta dissertação são:

- ✓ Proposta de um modelo de árvores com regressão logística para apoio a decisão em diagnóstico diferencial de linfomas (Seção 3.2).
- ✓ Implementação da seleção de atributos através da técnica conhecida por 'Lasso' (Seção 3.2).
- ✓ Adaptação de esquemas tradicionais de árvore para contemplar a necessidade de se evitar falsos negativos (Seção 3.3).
- ✓ Adaptação de esquemas tradicionais de regressão logística para contemplar a necessidade de se evitar falsos negativos (Seção 3.3).
- ✓ Emulação (com implementação) do esquema de tubos (Seção 3.4)

- ✓ Implementação do esquema proposto em um banco de dados real com esquemas de teste fora da amostra (Capítulo 4).
- ✓ Discussão de resultados obtidos (Capítulo 4).

1.4 Organização do documento

Esta dissertação está organizada da maneira a seguir. O capítulo 2 faz uma revisão da literatura sobre árvores de decisão, definindo-a em termos conceituais e explicando os métodos tradicionais de criação e poda da árvore. Ainda, são apresentadas técnicas para evitar o superajuste do modelo aos dados e uma introdução ao conceito de florestas.

O capítulo 3 descreve o método proposto. Neste capítulo, o conjunto de dados é apresentado, seguido da metodologia utilizada. Ainda, duas técnicas de ajuste fino com o objetivo de tornar o modelo mais robusto a falsos negativos são expostas. Finalmente, é introduzida a segunda metodologia utilizada, uma variação da primeira, porém visando a aproximar o modelo do procedimento real do exame de citometria.

Após a descrição da metodologia, uma série de experimentos são apresentados. Esses experimentos estão descritos no capítulo 4. Resultados incluem técnicas de validação cruzada e *leave-one-out*, além de dados não utilizados no treinamento.

Finalizando esta dissertação, o capítulo 5 apresenta um pequeno resumo deste trabalho, em conjunto com resultados e objetivos atingidos. Além disso, algumas considerações para investigação são indicadas como trabalhos futuros.

Capítulo 2

Árvores de Decisão

Uma árvore é um caso especial de grafo, onde o grafo é direcionado, conexo e acíclico, contendo $n-1$ arestas, onde n é o número de vértices. Portanto, em uma árvore direcionada deve existir no máximo um único caminho entre quaisquer par de vértices. A árvore é também um grafo planar, ou seja, possui caminhos que não se cruzam, e bipartido, quando o grafo pode ser particionado em dois conjuntos V_1 e V_2 e cada vértice em V_1 está conectado a pelo menos um vértice pertencente a V_2 (8).

As árvores de classificação são estruturas de dados em árvore usadas na tarefa de classificar observações em uma de algumas categorias pré-definidas. Uma árvore de classificação é uma estrutura hierárquica em que o usuário é guiado através de um caminho até um estágio final. Ela é feita de nós (os vértices), onde há somente um nó que não possui arestas entrantes, chamado nó raiz. Todos os outros nós possuem exatamente uma aresta entrante. Nós que possuem arestas que saem (grau do vértice é maior ou igual a 1) são chamados de nós internos. Os que não possuem são as folhas (nós terminais ou nós de decisão, possuem grau menor ou igual a 1) e contém a classificação final dada por aquele caminho ou um vetor de afinidade (probabilidade) daquela saída em relação aos dados de entrada (8) (9). Em uma árvore, cada nó interno divide o espaco em dois ou mais sub-espaços (galhos) de acordo com o resultado de um modelo de predição (10). Seguindo a definição de árvore, esses galhos nunca voltam a se encontrar, o que garante que cada observação de entrada será mapeada somente a uma folha (9).

2.1 Compreensibilidade

Um conceito importante a ser discutido é a compreensibilidade do método. Compreensibilidade pode ser definida como a habilidade de entender a resposta de determinado preditor (11). Dependendo da aplicação, é importante o tomador de decisão entender o motivo pelo qual determinado modelo chega a uma determinada conclusão. Na medicina, essa informação fornece subsídios ao tomador de

decisão e pode ajudar na escolha de um melhor tratamento (12). Estudos nessa área envolvem pedir aos usuários para realizar simples tarefas que se baseiam na interpretação dos modelos de classificação (11) (12). Critérios objetivos e subjetivos foram considerados. Algumas condições observadas nesses estudos que ajudam na compreensibilidade do modelo são:

1. Tamanho da representação. Representações menores possuem uma lógica mais simples de se acompanhar. Esse princípio é baseado na Navalha de Occam (*Occam's Razor*), que dá prioridade a explicações com menos suposições (11) (12).

2. Regras simples. Regras que incluem menos combinações de variáveis, tornam o processo mais intuitivo (11) (12).

3. Independência entre ramos de árvore. Sub-árvores replicadas aumentam a estrutura e a consequente complexidade do raciocínio (11).

4. Utilização de somente parte dos atributos (seleção de atributos). Atributos irrelevantes confundem o leitor na hora de extrair conhecimento da estrutura (11).

5. Utilização de cores para melhorar a leitura das diferentes classes em um nó (11).

6. *Layout* da árvore. Por exemplo, a representação em texto é mais complicada de se entender do que visualizações gráficas (como as representações gráficas que podem ser encontradas nos *softwares* Weka (13) e Orange (14)) (11).

7. Nomes significativos para os atributos (11).

8. Conhecimento do usuário sobre o domínio da aplicação (11).

Por possuir uma visualização de interpretação mais intuitiva, as árvores de decisão são preferíveis como métodos de apoio a decisão de profissionais no lugar de processos *black-box* ou não-lineares, como redes neurais e SVM.

2.2 Treinamento da árvore

Na fase de treinamento, como em outros métodos de classificação, normalmente, busca-se minimizar o erro de generalização ao construir um modelo. Porém, outros objetivos específicos dessa estrutura também podem ser considerados, como, por exemplo: minimizar o número de nós ou a altura da árvore.

Foi demonstrado que construir uma árvore binária ótima em relação ao número de testes esperados para classificar uma observação não vista é um problema NP-completo (15), enquanto, encontrar uma árvore ótima consistente com um conjunto finito de observações é um problema NP-difícil (16). Esses resultados foram importantes para guiar os algoritmos de treinamento a construir árvores quase ótimas, utilizando critérios heurísticos. Em geral, são os algoritmos gulosos, onde ótimos locais são escolhidos a cada nó. Essa abordagem, no entanto, não garante que o resultado final seja um ótimo global (17). Outros problemas, como encontrar a árvore ótima equivalente a uma árvore de decisão (18) e construir uma árvore de decisão ótima baseada em tabelas de decisão (19) também são demonstrados como problemas NP-difícil. Portanto, só se torna viável encontrar árvores ótimas em problemas de baixa complexidade. E, ainda assim, utilizando técnicas heurísticas para encontrar a solução.

2.2.1 Fase de crescimento

Existem dois tipos de algoritmos para a fase de crescimento da árvore: *top-down* e *bottom-up*, com clara preferência na literatura pelos *top-down* (9). Os algoritmos *top-down*, como o nome indica, começam a construir a árvore a partir do nó raiz e consistem em recursivamente escolher um modelo preditivo para o nó e reparti-lo em filhos até que um critério de parada seja atingido. Os algoritmos *bottom-up* funcionam de maneira inversa, as folhas para cada possível classe são criadas e as folhas mais próximas são unidas em nós pais até se chegar ao nó raiz. Este tipo de algoritmo apresenta diversas deficiências e são mais raros de serem encontrados (20). Os algoritmos mais comuns são: ID3 (21), C4.5 (22) e CART (23). Esses algoritmos diferenciam-se principalmente na escolha dos critérios de parada da árvore e divisão do nó. Mas assemelham-se por serem algoritmos gulosos, *top-down* e univariados (um nó interno é dividido de acordo com valor de somente um atributo). Uma versão genérica desses algoritmos pode ser vista no pseudo-código 1 (Retirado de ROKACH (9)).

Alguns dos critérios de divisão mais utilizados são (9):

1. **Funções da impureza dos conjuntos**: é o grau de heterogeneidade dos dados. Quando este índice se aproxima de zero, o nó é puro. Por outro lado, quanto maior o número de classes uniformemente distribuídas em um nó, maior a impureza da divisão. Existem algumas variações dessa medida, são elas:

   (a) **Ganho de informação**: utilizado no algoritmo ID3, utiliza a medida de entropia para cálculo da impureza. Para calcular o ganho de informação, compara-se o grau de entropia do nó pai (antes da divisão) com o grau

Procedure 1 Fase de Crescimento

Entrada: S: Conjunto de treinamento
Entrada: A: Conjunto de atributos de entrada
Entrada: y: Classe de cada observação
criterioParada: critérios para para o crescimento da árvore
criterioDivisao: método para avaliar a melhor divisão

```
1: procedure FaseDeCrescimento(S, A, y)
2:     T ← Árvore com somente nó raiz
3:     se criterioParada(S) então
4:         T ← folha com o valor y mais comum em S como label
5:     senão
6:         ∀a_i ∈ A encontre a_i que dê o melhor criterioDivisao(a_i, S)
7:         Rotule o nó como a_i
8:         v ← possíveis valores do atributo a_i
9:         para cada valor v_i ← v_0 até v faça
10:            subarvore_i ← FasedeCrescimento(S_{v_i}, A, y)
11:            T ← filho(subarvore_i)
12:        fim para
13:    fim se
14: fim procedure
```

de entropia dos nós filhos (após a divisão). A divisão que produzir o maior ganho é a divisão escolhida. O ganho de informação é dado pela fórmula 2.1:

$$GI = entropia(pai) - \sum_{i=1}^{n} \frac{N_i * entropia(filho_i)}{N} \tag{2.1}$$

onde n é o número de nós filhos, N é o número de observações existentes no nó e N_i o número de observações existentes no nó filho após a divisão. A entropia é dada pela fórmula 2.2:

$$entropia = -\sum_{i=1}^{c} p_i \log_2 p_i \tag{2.2}$$

onde p_i é a frequência relativa de cada classe no nó (fração dos registros pertencentes à classe i) e c o número total de classes.

(b) **Razão do ganho**: variação do ganho de informação, também foi desenvolvida por Quinlan (autor do ID3) para penalizar atributos contínuos, que por possuírem mais possibilidades de divisão, acabam tendo vantagem no critério anterior. O objetivo é ponderar o ganho de informação

pelo valor da entropia:

$$RG = \frac{GI}{entropia(nó)} \tag{2.3}$$

Esse critério não está definido quando o denominador é igual a zero e tende a favorecer denominadores de valor baixo. Portanto, Quinlan ainda sugere a realização do ganho em duas etapas: primeiro se calcula o ganho de informação para todos os atributos e segundo, seleciona-se somente aqueles que tiveram um desempenho no mínimo igual à media do ganho de informação para se calcular a razão do ganho. É utilizado no algoritmo C4.5.

(c) **Índice Gini**: critério utilizado no algoritmo CART, é um índice de dispersão estatístico que mede a divergência entre as distribuições de probabilidade. Assim como o critério anterior, o ganho do Gini é dado pela fórmula 2.4:

$$GI = gini(pai) - \sum_{i=1}^{n} \frac{N_i * gini(filho_i)}{N} \tag{2.4}$$

onde n é o número de nós filhos, N é o número de observações existentes no nó e N_i o número de observações existentes no nó filho após a divisão. O índice Gini é dado pela fórmula 2.5:

$$gini = 1 - \sum_{i=1}^{c} p_i \tag{2.5}$$

onde p_i é a frequência relativa de cada classe no nó (fração dos registros pertencentes à classe i) e c o número total de classes. Quando, nas árvores de classificação com partições binárias, se utiliza o critério de Gini tende-se a isolar num ramo os registros que representam a classe mais frequente. Quando se utiliza a entropia, balanceia-se o número de registros em cada ramo.

(d) **DKM**: nomeado a partir de seus autores (*Dietterich, Kearns e Mansour*), é um critério para atributos binários. Seus autores provaram que este critério constrói árvores menores que outras medidas de impureza para um certo nível de acurácia. O ganho é calculado como nos casos anteriores e o DKM segue a fórmula 2.6:

$$dkm = 2\sqrt{p_0 * p_1} \tag{2.6}$$

onde p_0 é a frequência relativa da classe 0 e p_1 da classe 1.

2. **Critérios binários**: podem ser utilizados somente quando os valores dos atributos serão divididos em dois grupos, como acontece nas árvores binárias.

   (a) **Ortogonal**: procura medir o cosseno entre os vetores de probabilidade de dois grupos (24). Se todas as observações de determinado grupo estão em uma partição, o cosseno entre os ângulos resulta em zero e a medida chega ao valor máximo, um. É dada pela fórmula 2.7:

   $$ort = 1 - cos(p_0, p_1) \tag{2.7}$$

   onde p_0 é a frequência relativa do atributo de valor a_0 e p_1 do de valor a_1.

   (b) **Kolmogorov-Smirnov**: mede a distância entre duas distribuições de probabilidade (25). Pode ser calculada com a fórmula 2.8:

   $$KS = |F(0) - F(1)| \tag{2.8}$$

   onde $F(0)$ é a distribuição univariada cumulativa do atributo de valor a_0 e $F(1)$ do de valor a_1.

A fase de crescimento da árvore continua até que algum critério de parada seja atingido. São exemplos de critério (9):

1. Todas as observações pertencem a mesma classe.
2. A altura máxima da árvore foi atingida.
3. O número de observações em um nó é menor que o número mínimo estabelecido.
4. Se o nó for dividido, o número mínimo de observações no nó filho será menor que o número mínimo estabelecido.
5. A melhor divisão não produz resultados melhores que um *threshold* mínimo.

2.3 *Overfitting*, validação cruzada e *pruning*

Uma das potenciais desvantagens das árvores decisão é a sua facilidade em se especializar. A maneira recursiva com o qual os algoritmos em árvores são construídos, facilitam o *overfitting*, uma vez que é possível aprofundar-se tanto quanto se queira no ajuste da árvore.

Árvores com muitos nós em relação ao tamanho do conjunto de treinamento provavelmente possuem um erro dentro da amostra pequeno. No entanto, isto não

significa que a generalização para amostras fora do conjunto de treino esteja boa. É possível que a árvore esteja memorizando o resultado das observações ao invés de entender os padrões. O contrário também é possível acontecer. Erros dentro da amostra altos podem levar a baixa generalização por não aprender corretamente os padrões encontrados nos dados.

2.3.1 Validação cruzada e *leave-one-out*

Uma técnica possível para evitar o *overfitting* é dividir o conjunto de testes em treino e teste, onde as observações em teste não são utilizadas no treinamento. Elas são consideradas observações fora da amostra e servem para verificar o erro de generalização do classificador. No entanto, em conjuntos com poucas observações, essa separação pode levar a poucas amostras em cada conjunto e, consequentemente, a baixa generalização.

Para evitar um possível viés dado pela separação aleatória das observações em dois conjuntos fixos, a validação cruzada separa os dados em n conjuntos e utiliza combinações diferentes desses conjuntos para escolher os dados de treinamento e teste. Estendendo esse conceito para uma observação por conjunto, e utilizando como teste somente um conjunto e todos os outros para treinamento, define-se o método de *leave-one-out*. Repete-se esse procedimento para cada observação. Ao final, é obtida uma estimativa do erro de generalização do modelo. (26)

2.3.2 *Pruning*

A técnica anterior pode ser utilizada em qualquer método de construção de modelos de predição. Uma segunda técnica mais particular deste método para se evitar a especialização da árvore, é a poda da árvore (*pruning*). Existem dois tipos de poda: a pré-poda (a parada do crescimento da árvore mais cedo) e a pós-poda (criar uma árvore completa e depois podá-la). De acordo com QUINLAN (27), crescer e depois podar é mais custoso mas produz resultados mais confiáveis.

O método de poda foi proposto em BREIMAN (23), utilizando critérios de parada da árvore mais frouxos. Esse processo produz árvores mais especializadas, aumentando a probabilidade de *overfitting*. Após a parada, a árvore então é submetida à poda, tendo galhos que não são considerados úteis na generalização do modelo removidos. Além de melhorar a generalização da árvore, outra vantagem da poda é a simplificação de sua visualização, contribuindo com sua compreensibilidade, conforme descrito na seção 2.1.

Assim como os algoritmos de crescimento da árvore, os algoritmos de poda também podem ser *top-down* ou *bottom-up* e seguem o pseudo-código 2 (Retirado de ROKACH (9)).

Procedure 2 Fase de Poda
Entrada: S: Conjunto de treinamento
Entrada: y: Classe de cada observação
Entrada: T: Árvore a ser podada

1: **procedure** FASEDEPODA(S, y, T)
2: **repita**
3: Escolha um nó t em T onde removendo-o maximiza a melhora em algum critério
4: **se** $t \neq \phi$ **então**
5: $T \leftarrow poda(T, t)$
6: **fim se**
7: **enquanto** $t \neq \phi$
8: **fim procedure**

A seguir está a descrição de alguns dos algoritmos de poda:

1. **Cost Complexity Pruning**: algoritmo *bottom-up*, proposto no CART, utiliza uma combinação de estimativa de erro e complexidade da árvore. Na primeira etapa, é construída uma sequência de árvores T_0, T_1, ..., T_k, onde T_0 é árvore original e T_k a árvore formada pelo nó raiz. As árvores intermediárias são árvores podadas, onde T_{i+1} é uma sub-árvore de T_i, com um nó interno transformado em folha. Para cada árvore T_i, existem algumas possibilidades de sub-árvores. Para escolher a melhor sub-árvore para montar a sequência, utiliza-se a fórmula 2.9:

$$\alpha = \frac{erro(pruned(T,t),S) - erro(T,S)}{folhas(T) - folhas(pruned(T,t))} \tag{2.9}$$

onde $pruned(T,t)$ é uma sub-árvore de T, $error(T,S)$ é a quantidade de observações contidas na amostra S classificadas erroneamente e $folhas(T)$ é a número de folhas na árvore. Essa fórmula procura buscar a sub-árvore que produz o melhor equilíbrio entre o erro por remover-se um nó e a quantidade de folhas remanescentes. Após a sequência T_0, T_1, ..., T_k ser criada, testa-se essas árvores com dados de teste ou com validação-cruzada afim de encontrar o menor erro de generalização.

2. **Reduced Error Pruning**: proposto em QUINLAN (28), é um método *bottom-up* que calcula o erro a cada nó se este nó (e suas sub-árvores) for substituído por uma folha. Se o erro permanecer igual, o nó é podado e a verificação continua para o nó pai. O algoritmo para quando a acurácia da árvore diminui. Pode ser usado com um *threshold* para a diferença do erro e produz melhores resultados de generalização se o conjunto de teste for diferente do conjunto de treinamento.

3. **Pessimistic Pruning**: a grande vantagem desse método é que ele poda a árvore utilizando uma estatística baseada nos dados de treinamento, sem a necessidade de um conjunto de dados extra para validação ou validação cruzada. É um método *top-down* que assume uma distribuição binomial da estimativa de erro e aplica a correção contínua (*continuity correction*), um ajuste feito nos dados para aproximar uma distribuição contínua (distribuição normal) de uma discreta (distribuição binomial) (29). Usam as equações 2.10, 2.11 e 2.12 para obter o erro:

$$erro(nó) = misclassified(nó) + \frac{folhas(pruned(T,t))}{2} \tag{2.10}$$

$$erro(subárvore) = misclassified(subárvore) + \frac{folhas(T)}{2} \tag{2.11}$$

O termo $\frac{1}{2}$ é a penalidade estimada de uma folha na complexidade da árvore. A sub-árvore só é mantida se o erro para ela for menor que o erro da árvore sem o nó, acrescido de um desvio padrão:

$$erro(subárvore) \leq erro(nó) + \sqrt{erro(nó) * \frac{N - erro(nó)}{N}} \tag{2.12}$$

onde N é o número de elementos no nó.

4. **Error-Based Pruning (EBP)**: implementado no algoritmo C4.5, é considerado o sucessor do *Pessimistic Pruning*. Assim como seu antecessor, a distribuição da estimativa de erro é assumida como binomial. No entanto, este método calcula um intervalo de confiança das classificações errôneas utilizando como base a aproximação da distribuição binomial da distribuição normal para uma quantidade grande de amostras. Utiliza-se o limite superior deste intervalo de confiança para estimar o erro nas folhas, conforme a equação 2.13. No algoritmo C4.5, o intervalo utilizado é um intervalo padrão de 25%.

$$erro(nó) \leq misclass(nó) + Z_\alpha * \sqrt{misclass(nó) * \frac{N - misclass(nó)}{N}} \tag{2.13}$$

onde N é o número de elementos no nó, Z_α é o inverso da distribuição cumulativa normal e α o nível de significância desejado. São comparados três erros: o erro do nó com as sub-árvores, o erro do nó podado e o erro ao substituir o nó pelo seu filho com maior quantidade de elementos (técnica chamada de *subtree raising*).

2.4 Outros Algoritmos de Árvore

Os métodos de crescimento e poda aqui apresentados constroem uma árvore com base em dados rotulados, criando um preditor para uma classificação supervisionada. Existem ainda outros métodos de criação de árvore, baseados no conhecimento de especialistas ou que são a combinação de diferentes modelos.

2.4.1 Árvores a partir de extração de conhecimento

Suas regras são extraídas principalmente através de entrevista com especialistas. Esse tipo de levantamento e chamado de "*knowledge elicitation*" e é o mais heurístico dos métodos. A acurácia do método depende da complexidade do problema e, principalmente, da experiência do entrevistado em relação ao assunto. O gargalo deste método se encontra justamente na dificuldade de se encontrar um especialista que possa transmitir seu conhecimento ao responsável pela construção do modelo. Problemas complexos ficam altamente sub-treinados com esse tipo de abordagem (9).

2.4.2 Florestas

A principal contribuição das florestas é amenizar uma grande desvantagem dos métodos de construção de árvores: sua instabilidade. Nos algoritmos que comparam uma variável por nó, pequenas alterações nos dados podem modificar a variável escolhida para comparação. Consequentemente, árvores completamente diferentes podem ser criadas a cada modificação, fornecendo resultados distintos. Com florestas, múltiplas árvores são criadas a partir de subconjuntos dos conjuntos originais e seus modelos combinados para dar uma resposta mais global. Assim como as árvores simples buscam modelar a lógica feita ao se tomar uma decisão, pode ser feito um comparativo de florestas com uma enquete de várias opiniões antes de se tomar uma decisão final. Pesos até podem ser atribuídos para resultados considerados mais confiáveis e somados para se chegar a uma conclusão (30). O ponto fraco desta metodologia é que o resultado perde parte de sua compreensibilidade, se posicionando entre uma estrutura simples de árvore e regras de decisão (mais compreensíveis) e os modelos *black-box* (menos compreensíveis). Algumas das técnicas escolhidas para pesar os resultados são: votação, peso por performance, combinação *bayesiana*, somatório de distribuição, entre outras (9).

Capítulo 3

Métodos e Dados

Neste capítulo descreve-se a base de dados e a metodologia proposta como uma possível solução para o problema de diagnóstico de linfoma brevemente descrito no capítulo introdutório. Faz parte também deste capítulo, o tratamento feito na base de dados e uma rápida discussão do método proposto.

Por diagnóstico diferencial, objetivo desse trabalho, entende-se a distinção entre diversos tipos de linfoma. Busca-se assim qual o diagnóstico de linfoma do paciente, dentro de um conjunto pré-estabelecido. O método propõe a construção de uma árvore utilizando um modelo de regressão logística em cada nó e analisa a combinação de múltiplas variáveis em cada passo da decisão a fim de escolher ou eliminar classes, implementando o diagnóstico diferencial. O intuito é oferecer um modelo no qual a palavra final seja de um especialista humano, deixando transparente as etapas de decisão. Para atingir esse objetivo, exibe-se ao tomador de decisão, além da resposta dada pelo classificador, os atributos que foram dominantes para que o classificador chegasse a determinada conclusão.

3.1 Base de dados

A base de dados é constituída por dados provenientes de citometria de fluxo de pacientes diagnosticados com linfomas. Esses dados foram gerados por diversos grupos de pesquisa pertencentes ao consórcio *EuroFlow* (7).

Para cada célula de cada paciente tem-se disponível uma medida de fluorescência para cada um dos 24 anticorpos utilizados nos exames. Dessa forma, cada anticorpo (marcador) provê uma medida para cada célula. Do ponto de vista de classificação de padrões, tem-se 24 atributos medidos e, portanto, um espaço de 24 dimensões. Nesta dissertação trabalha-se com a mediana das células em cada um destes atributos para cada um dos pacientes. Assim, cada paciente é representado por um ponto em $\mathbb{R}^{24}$ constituído pelas medianas das intensidades de fluorescência de cada um dos marcadores. A utilização das medianas, ao invés da totalidade das medidas se

aplica porque que o objetivo do trabalho é analisar as diferenças entre os pacientes (pacientes com um mesmo diagnóstico estarão, através de suas medianas, aproximadamente na mesma região do espaço $\mathbb{R}^{24}$), e não as variâncias em um mesmo paciente. Cada linfoma é tratado como um rótulo de classe que se deve atribuir a cada paciente.

O banco de dados é composto por 283 pacientes diagnosticados (por pesquisadores do consórcio *EuroFlow*) com um entre oito possíveis tipos de linfoma, a saber: *Burkitt Lymphoma* (BL), *Diffuse Large B-cell Lymphoma* (diferenciando-se em 2 tipos, aqui referenciados por CD10- e CD10+), *Chronic Lymphocytic Leukemia* (CLL), *Follicular Lymphoma* (FL), *Hairy Cell Leukemia* (HCL), *Lymphoplasmacytic Lymphoma* (LPL) agrupado com *Marginal Zone Lymphoma* (MZL) e *Mantle Cell Lymphoma* (MCL), conforme resumido nas Tabelas 3.1 e 3.2.

Classes	Observações
BL	15
CD10-	24
CD10+	30
CLL	43
FL	36
HCL	33
LPL+MZL	66
MCL	36

Tabela 3.1: Quantidade de observações por classe

Atributo	Máx	Mín	Atributo	Máx	Mín	Atributo	Máx	Mín
CD10	8970	-446	CD27	18030	-393	CD62L	1879	-277
CD103	3501	-284	CD31	5461	-221	CD79b	41910	41
CD11c	35436	-203	CD38	15787	-65	CD81	29795	135
CD19	62732	804	CD39	14325	-65	CD95	11793	-195
CD20	157407	641	CD43	17174	-40	CXCR5	64559	21
CD200	56645	-55	CD45	11993	500	HLADR	66565	18
CD22	87137	-2	CD49d	4851	-66	IgM	18160	-188
CD23	5487	-338	CD5	7104	-110	LAIR1	23534	-9

Tabela 3.2: Valores máximo e mínimo medidos para cada atributo antes do pré-processamento (arredondados sem decimal).

Como anticorpos possuem diferentes afinidades com seus antígenos, é fundamental escalar os dados entre um valor máximo e mínimo comum a todos os atributos. Assim, é possível equilibrá-los e a análise se concentrará, principalmente, sobre a variação dos valores dentro de cada atributo. Os valores máximo e mínimo foram

escolhidos para 10 e 0, respectivamente. Valores de intensidade de fluorescência negativos são considerados ruídos e foram ajustados para 0 antes do escalamento.

3.2 Construção da árvore

A fase de treinamento possui duas macro etapas que se repetem nó a nó. A primeira delas consiste na parametrização de uma função logística baseada nas observações (pacientes) existentes naquele nó, conforme demonstrado no algoritmo 3. A segunda etapa é a classificação das observações utilizando o modelo criado na primeira etapa com o objetivo de dividi-las em dois grupos. Parte dos pacientes serão encaminhados para o filho esquerdo do nó, enquanto a outra parte para o filho direito. As etapas se repetem independentemente em cada filho até que o algoritmo atinja uma condição de parada. Este procedimento está detalhado no algoritmo 4. A solução final é obtida através de uma sucessão de regressões logísticas ao longo da árvore.

3.2.1 Etapa 1: parametrização da função logística

A regressão logística é um modelo de regressão (entrega como resposta um número real) que pode ser utilizado para classificação (pois sua resposta é delimitada entre 0 e 1), representando uma previsão da probabilidade da pertinência da observação testada às classes treinadas. Ela é essencialmente binária, ou seja, só pode ser utilizada para treinar dois grupos, que serão associados aos valores 0 e 1 (26).

Neste passo há observações de diferentes classes (categorias de linfoma) que precisam ser divididas em dois grandes grupos (os dois grupos definidos na regressão logística). No nó raiz da árvore, por exemplo, existem oito classes que serão divididas em dois grupos para o treinamento com a regressão logística. O desafio é escolher a combinação de classes em dois grupos que apresente o melhor resultado. Para isso foram definidos três critérios, utilizados em sequência, caso o critério anterior resulte em empate. Eles estão descritos a seguir na ordem em que são empregados:

1. Precisão do modelo: como busca-se o melhor modelo, uma métrica natural é a quantidade de pacientes corretos que esse modelo classifica. A precisão foi escolhida ao invés da acurácia total, pois, essa metodologia valoriza a pureza de pelo menos um dos grupos. Como divisões subsequentes virão para dar continuidade ao treinamento, e preferível ter um grupo puro com uma quantidade maior de elementos corretos e outro misturado do que ambos misturados com os elementos corretos mais dispersos em dois grupos.

2. Separabilidade arredondada: é calculado o nível de separabilidade entre os grupos, verificando a distância euclidiana do resultado da função logística (re-

sultado entre 0 e 1) de cada observação ao valor de 0,5 (metade da função logística, onde a observação possui 50% de probabilidade de pertencer a qualquer um dos grupos), balanceando pelo número de observações em cada grupo. É desejável grupos que estejam bem separados para diminuir a área de dúvida entre eles. No entanto, aqui foi utilizada a separabilidade arredondada em duas casas decimais, pois, apesar de desejar a máxima separabilidade entre grupos, é preciso manter o modelo simples e o menor ajustado possível, o que leva ao critério 3. O cálculo da separabilidade arredondada é dado pela equação 3.1:

$$sep(A,B) = \frac{\sum_{i=1}^{N_A} |logit(obs_{A_i}) - 0,5|}{N_A} + \frac{\sum_{i=1}^{N_B} |logit(obs_{B_i}) - 0,5|}{N_B} \tag{3.1}$$

onde N_A e N_B representam a quantidade de elementos presentes nos grupos A e B respectivamente, obs_A e obs_B são as observações pertencentes a cada um dos grupos e $logit()$ é o resultado da regressão logística, o indicador da probabilidade de pertinência da observação aos grupos.

3. Quantidade de atributos: esse critério parte do princípio que, dado que a precisão e a separabilidade dos modelos são equivalentes, quanto menor a quantidade de atributos selecionados, menor a complexidade do modelo e seus graus de liberdade. Esse critério contribui na diminuição da esparsidade dos dados (31), deixando mais evidente as similaridades entre as observações. Ainda, a menor quantidade de atributos diminui o custo computacional e o uso de memória. Como mais uma vantagem inclua a simplificação do modelo, que se torna mais compreensível por humanos.

Tomando como exemplo o nó raiz e utilizando as 8 categorias presentes no banco de dados, existem 127 combinações distintas. Essas combinações são calculadas pensando-se na separação em 2 grupos das diferentes doenças (um grupo com 1 e o outro com as 7 restantes; um grupo com 2 e outro com as 6 restantes, etc.). A equação 3.2 demonstra como se calcula as possibilidades dos grupos no nó raiz. Essa fórmula pode ser escrita em sua forma fechada, conforme a equação 3.3, e utilizada para calcular a quantidade de divisões existentes em qualquer nó, substituindo n pela quantidade de classes existentes no nó. Ressalta-se que a ordem das doenças no grupo não importa e se a combinação aparece no grupo A e depois no grupo B é considerada uma repetição, uma vez que a árvore cuja combinação aparece no grupo A é considerada um espelho da árvore que contém a combinação no grupo B, pois levam à mesma classificação final.

$$\frac{\binom{8}{1} + \binom{8}{2} + \binom{8}{3} + \binom{8}{4} + \binom{8}{5} + \binom{8}{6} + \binom{8}{7}}{2} = 127 \tag{3.2}$$

$$2^{n-1} - 1 \tag{3.3}$$

onde n é a quantidade de classes existentes no nó.

Após explicitar todas as combinações possíveis no nó, escolhe-se a melhor combinação seguindo os critérios definidos anteriormente.

Para estimar os pesos da função logística em cada nó, foi utilizada uma versão regularizada do algoritmo da regressão logística, que minimiza a equação 3.4 (32). É utilizada a penalidade $\lambda \sum |\beta|$, onde λ é o parâmetro de ajuste da regularização e β é a matriz de pesos resultante da regressão logística. A vantagem da regularização sobre a regressão logística tradicional é que ela diminui os coeficientes, reduzindo a variância dos valores estimados e diminuindo a possibilidade de *overfitting*. Ainda, comparando a regularização *ridge* com o Lasso, o Lasso é capaz de reduzir alguns coeficientes a zero, agindo também como um seletor de atributos.

$$(\hat{\alpha}, \hat{\beta}) = \min(\sum_{i=1}^{n}(y_i - \alpha - x_i^T\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|) \tag{3.4}$$

onde y_i é a classe original da observação i, α é o viés do modelo, x_i é a observação i, λ é uma parâmetro não negativo de regularização e β é a matriz de pesos resultante da regressão logística.

3.2.2 Etapa 2: classificação das observações

Nesta fase cada paciente é classificado utilizando o modelo criado na etapa 1 e direcionado ao filho esquerdo ou direito do nó. Cada filho funciona de maneira independente e repetirá o procedimento a partir da etapa 1; a não ser que atinja um dos critérios de parada. São eles:

1. Inexistência de um grupo puro. Um grupo é considerado puro se contém somente observações classificadas corretamente.

2. Ao menos uma das categorias de linfomas (classes) presentes no nó deve possuir no mínimo 40% das observações de seus pacientes presentes.

3. Presença de observações de somente uma classe no nó.

Essas condições foram escolhidas para evitar a criação de modelos (nós) especializados em características de poucos pacientes. A última condição de parada é quando um nó possui somente observações de uma única classe, cessando a necessidade de qualquer separação.

A Figura 3.1 mostra um exemplo simplificado com 4 classes (BL, CLL, HCL e MCL). Todos os hiperplanos desenhados nesses exemplos são somente uma indicação

Procedure 3 Construção da árvore - parametrização da função logística

Entrada: *classes*: Todas as categorias dos linfomas existentes no nó
Entrada: *observacoes*: Todas as observações pertencentes a *classes*
Saída: *pesos*: Pesos do modelo existente no nó

```
1: procedure ETAPA1(classes, observacoes)
2:     /* Explicita todas as combinações */
3:     combinacoes ← [ ⁸C₁, ⁸C₂, ⁸C₃, ⁸C₄]
4:     grupos ← separarEmGrupos(combinacoes)
5:     precisao ← [ ]
6:     separabilidade ← [ ]
7:     quantidade ← [ ]
8:     para comb_i ← grupos[0] até grupos[end] faça
9:         /* Calcula os pesos e em seguida classifica as observações */
10:        pesosCombinacao ← lasso(comb_i[grupoA], comb_i[grupoB])
11:        comb_i[pesos] ← pesosCombinacao
12:        resultadoLogistica ← prever(pesosCombinacao, observacoes)
13:        precisaoCombinacao ← matrizConfusao(round(resultadoLogistica), observacoes)
14:        /* Recupera precisão para cada combinação */
15:        precisao[comb_i] ← precisaoComb
16:        /* Calcula a separabilidade dos grupos */
17:        separabilidade[comb_i] ← round(distanciaEuclid(resultadoLogistica, observacoes), 2)
18:        /* Conta a quantidade de pesos não-zerados para cada combinação */
19:        quantidade[comb_i] ← size(pesosComb>0)
20:    fim para
21:    /* Critério 1: maior precisão */
22:    indexMelhorCombinacao ← index(max(precisao))
23:    /* Caso critério 1 empate */
24:    se size(indexMelhorCombinacao) > 1 então
25:        /* Critério 2: maior separabilidade entre grupos */
26:        indexMelhorCombinacao ← index(max(separabilidade))
```

Procedure 3 Continuação (parametrização da função logística)

```
27:        /* Caso critério 2 empate */
28:        se size(indexMelhorCombinacao) > 1 então
29:            /* Critério 3: menor quantidade de pesos */
30:            indexMelhorCombinacao ← index(min(quantidade))
31:        senão
32:            /* Caso critério 3 empate, aleatório entre os melhores */
33:            indexMelhorCombinacao ← random(indexMelhorCombinacao)
34:        fim se
35:    fim se
36:    melhorCombinacao ← combinacoes[indexMelhorCombinacao]
37:    grupoA ← melhorCombinacao[grupoA]
38:    grupoB ← melhorCombinacao[grupoB]
39:    pesos ← melhorCombinacao[pesos]
40: fim procedure
```

Procedure 4 Classificação das observações

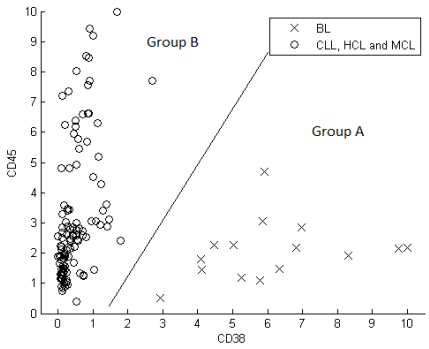
Entrada: $grupoA$: Classes pertencentes ao grupo A
Entrada: $grupoB$: Classes pertencentes ao grupo B
Entrada: $pesos$: Pesos do modelo existente no nó
Entrada: $total$: Número total de elementos por classe
Entrada: $observacoes$: Todas as observações pertencentes as classes existentes nos grupos A e B

```
1:  procedure ETAPA2(grupoA, grupoB, pesos)
2:      filhoEsquerdo ← [ ]
3:      filhoDireito ← [ ]
4:      /* Classifica cada observação e direciona-a ao filho esquerdo ou direito */
5:      para obs ← observacoes[0] até observacoes[end] faça
6:          filho ← classificar(pesos, obs)
7:          se filho == esquerdo então
8:              filhoEsquerdo.append(filho)
9:          senão
10:             filhoDireito.append(filho)
11:         fim se
12:     fim para
13:     /* Verifica se critério de parada 1 se aplica */
14:     grupoPuro ← False
15:     se (filhoEsquerdopertencegrupoA)||(filhoDireitopertencegrupoB) então
16:         grupoPuro ← True
17:     senão
18:         retorna
19:     fim se
```

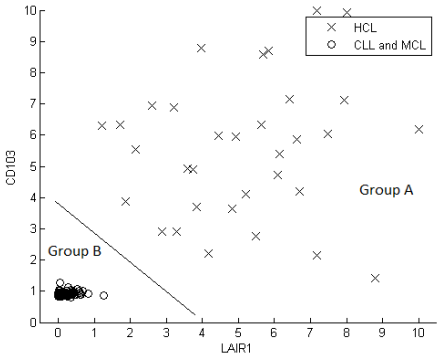
Procedure 4 Continuação (classificação das observações)

```
20:     /* Verifica se critério de parada 2 se aplica */
21:     filhoEsquerdoCrit2 ← False
22:     totalFilhoEsquerdo ← contaElementosPorClasse(filhoEsquerdo)
23:     para classe ← grupoA[0] até grupoA[end] faça
24:         se totalFilhoEsquerdo[classe]/total[classe] > 0.4 então
25:             filhoEsquerdoCrit2 ← True
26:         fim se
27:     fim para
28:     filhoDireitoCrit2 ← False
29:     totalFilhoDireito ← contaElementosPorClasse(filhoDireito)
30:     para classe ← grupoB[0] até grupoB[end] faça
31:         se totalFilhoDireito[classe]/total[classe] > 0.4 então
32:             filhoDireitoCrit2 ← True
33:         fim se
34:     fim para
35:     /* Verifica se critério de parada 3 se aplica */
36:     filhoEsquerdoCrit3 ← False
37:     se size(totalFilhoEsquerdo > 0) > 1 então
38:         filhoEsquerdoCrit3 ← True
39:     fim se
40:     filhoDireitoCrit3 ← False
41:     se size(totalFilhoDireito > 0) > 1 então
42:         filhoDireitoCrit3 ← True
43:     fim se
44:     se (filhoEsquerdoCrit2 == True)&&(filhoEsquerdoCrit3 == True)
  então
45:         recomecaEtapa1(filhoEsquerdo)
46:     fim se
47:     se (filhoDireitoCrit2 == True)&&(filhoDireitoCrit3 == True) então
48:         recomecaEtapa1(filhoDireito)
49:     fim se
50: fim procedure
```

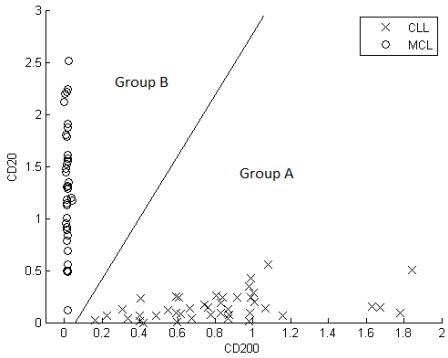
da divisão entre as classes. Os modelos foram criados com dimensões maiores que $\mathbb{R}^2$ e para representação foram escolhidas as 2 dimensões que possuem mais peso na classificação.



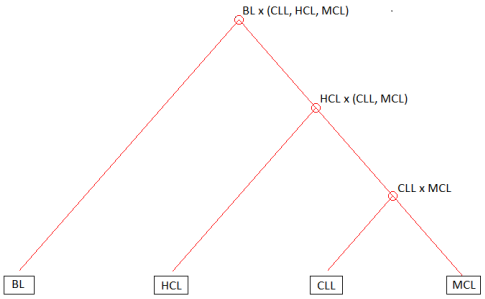
(a) Separe as categorias em dois grupos. Grupo A contém BL e grupo B é composto por CLL, HCL e MCL. Crie um modelo linear que separe esses dois grupos. Verifique se as condições de parada se aplicam. Grupo A possui somente pacientes de uma classe, então a árvore para de crescer a partir do filho esquerdo. Grupo B ainda possui 3 classes, a árvore continuará a crescer a partir do filho direito.



(b) Grupo A agora é formado por pacientes com HCL e grupo B por CLL e MCL. O modelo linear do nó é criado. As mesmas condições de parada da letra (a) se aplicam aqui. A árvore não crescerá mais no filho esquerdo deste nó e continuará crescendo no filho direito.



(c) Grupo A possui pacientes com CLL e grupo B com MCL. Depois que o modelo linear é criado, cada grupo possui somente uma doença por nó. A árvore para de crescer em ambos os filhos.



(d) Estrutura final da árvore. Nas folhas as classes estimadas pelo modelo. Nos nós que não são folhas, um modelo linear (os criados nas letras (a), (b) e (c)).

Figura 3.1: Passo a passo da construção da árvore.

3.3 Robustez contra falsos negativos

Como ressaltado no capítulo 1, um dos fatores fundamentais neste desenvolvimento é evitar falsos negativos mesmo que o preço a pagar sejam diagnósticos duplos ou triplos. Em uma comparação com uma classificação binária, a sensibilidade seria mais valorizada ao calcular o *score* final do modelo. Para este problema, é essencial que sejam evitados falsos negativos para não induzir o médico a um diagnóstico errado ao eliminar alguma categoria de linfoma (classe) prematuramente. Os algoritmos de árvore tradicionais classificam as observações de acordo com a classe mais provável, oferecendo somente uma opção como resposta. Para aumentar a robustez contra falsos negativos, ainda seguindo o princípio da árvore, essa dissertação propõe uma região de incerteza, onde uma observação pode ser classificada em mais de uma classe. Foram criadas duas abordagens, explicadas em detalhes abaixo.

3.3.1 Comparações a partir da folha

Durante a construção do modelo, após atingir uma folha com uma classe (*singleton*), esta classe é treinada individualmente contra as outras classes, como descrito no pseudo-algoritmo 5.

Procedure 5 Treinamento da região de incerteza

Entrada: *classeSingleton*: Classe da folha com uma doença
Entrada: *todasClasses*: Todas as classes
Entrada: *noPai*: Nó pai (nó do singleton)

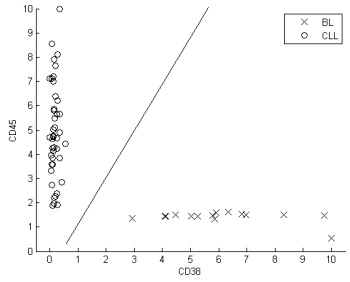
```
1: procedure ETAPA3(classeSingleton, todasClasses, noPai)
2:     /* Esteja na condição de parada onda há somente observações de uma classe
   */
3:     para classe ← todasClasses[0] até todasClasses[end] faça
4:         se classe! = classeSingleton então
5:             pesos ← lasso(classeSingleton, classe)
6:             /* Adiciona resultado do Lasso como um nó filho do nó singleton */
7:             adicionaNo(pesos, noPai, classeSingleton, classe)
8:         fim se
9:     fim para
10: fim procedure
```

Isso significa que se o modelo começou tentando separar 8 doenças, após atingir um *singleton* em uma folha, este *singleton* será treinado contra as outras 7 classes individualmente, criando mais 7 nós com 7 modelos distintos. Quando uma observação está sendo testada, ela será testada também nessas 7 comparações individuais. Para ser classificada com um resultado único, a observação deve ser classificada como pertencente à classe dominante em todas as comparações indivi-

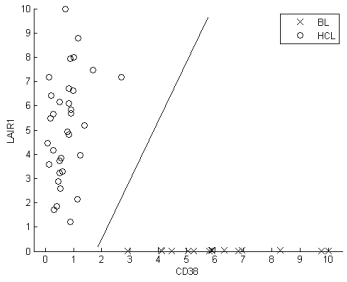
duais. Se a observação é classificada como pertencente a outra classe em quaisquer das comparações, essa classe extra será incluída como possibilidade de resposta, oferecendo assim um resultado múltiplo (*multilabel*). A Figura 3.2 continua o exemplo da Figura 3.1. Novamente, os hiperplanos desenhados são somente uma indicação da separação entre as classes.

3.3.2 Resultado da função logística

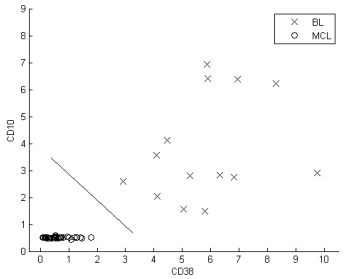
Uma abordagem complementar para criar essa região é utilizando o resultado da função logística. A função logística entrega como resultado a probabilidade entre 0 e 1 da observação pertencer ao grupo B. Se o número está mais próximo de 1, a probabilidade é maior. Se está mais próximo de 0, isso significa que é mais provável que a observação não pertença ao grupo B. Neste caso, as classes estão sempre divididas em dois grandes grupos (A e B). Logo, se a observação não pertence ao grupo B, ela pertenceria ao grupo A. Mas onde se encontra o limite para definir que uma observação pertence a um grupo e não a outro? O diferencial utilizado nesta parte é a utilização de dois limites ao invés de somente um (tipicamente escolhido para 0,5). Durante o treinamento, o algoritmo foi construído de modo a estabelecer um limite superior e um inferior. As observações cujo resultado seja maior que o limite superior serão classificadas como pertencentes ao grupo B. As observações cujo resultado seja menor que o limite inferior serão associadas ao grupo A. Se o resultado cair entre os limites, a observação será classificada com mais de um rótulo, pertencendo às classes presentes nos grupos A e B. A Figura 3.3 utiliza dados simulados para ilustrar a função logística com os limites superiores e inferiores estabelecidos. Os limites para a região de incerteza são definidos durante a criação do nó da árvore, em seu treinamento. O limite superior deve ficar logo acima da primeira observação do grupo oposto, no caso o grupo A (a observação que possuir o maior valor da função logística). Portanto, o limite superior será a metade da distância euclidiana (considerando somente o resultado da função logística) entre a primeira observação do grupo A e observação imediatamente acima desta pertencente ao grupo B. O mesmo raciocínio pode ser aplicado para definir o limite inferior. Será fixada na metade da distância euclidiana entre a última observação do grupo B (de valor mais próximo de 0) e a observação imediatamente abaixo do grupo A.



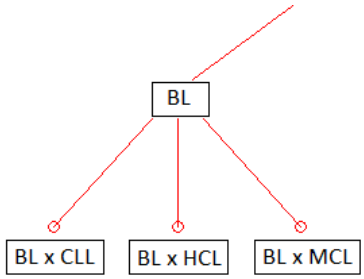
(a) Após atingir uma folha classificada como BL, a observação será testada em modelos lineares individualmente contra as outras 3 doenças. Essa Figura apresenta o modelo linear de BL x CLL.



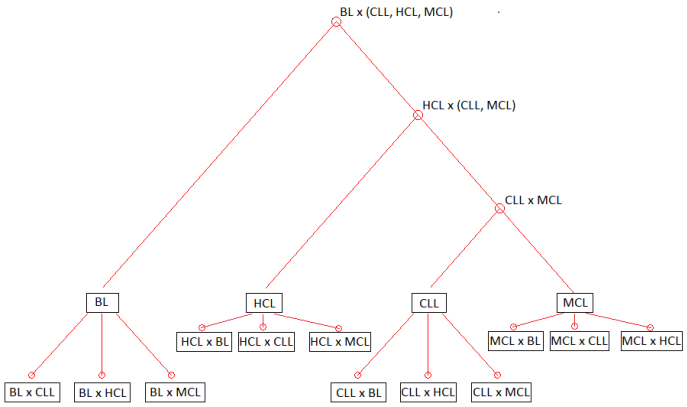
(b) Para ser classificada unicamente como BL, a observação deve ser classificada como BL nas três comparações. Esta Figura mostra o modelo linear de BL x HCL.



(c) Do contrário, a observação será classificada como BL, mas também como as outras classes nas quais ela foi classificada nessas comparações individuais. Esta Figura apresenta o modelo linear de BL x MCL.



(d) Representação da folha com as três comparações finais adicionadas.



(e) Representação completa da árvore. Cada *singleton* possui os nós com as comparações individuais adicionadas.

Figura 3.2: Método 1 para aumentar a robustez contra falsos negativos: comparações individuais.

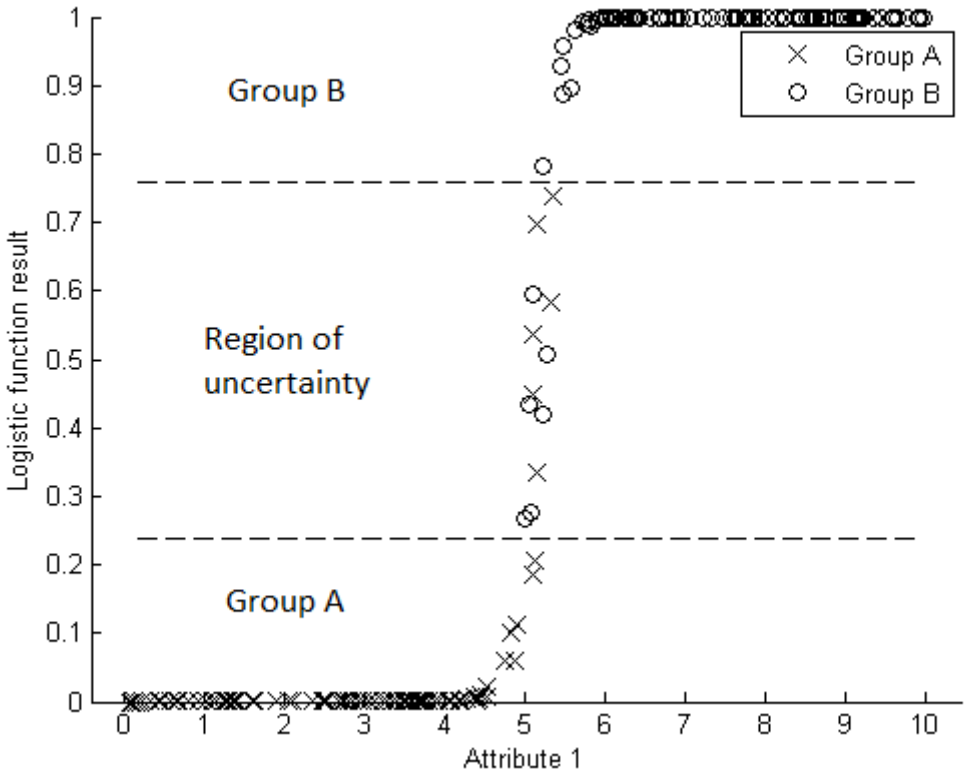


Figura 3.3: Representação ilustrativa (com dados simulados) da função logística. Todas as observações cujo resultado é maior que o limite superior serão classificadas como pertencentes ao grupo B; todas as observações cujo resultado é inferior ao limite inferior, serão classificadas como pertencentes ao grupo A; e todas as observações entre os limites são classificadas como pertencentes aos grupos A e B.

3.4 Emulando o exame de citometria de fluxo

Nesta seção, propõe-se um esquema de árvore que emule a forma como são usualmente realizados os exames nos laboratórios. Esta abordagem consiste em separar os atributos em conjuntos, reproduzindo a maneira na qual o procedimento do exame de citometria é realizado. Assim, executa-se uma parte do exame (utilizando um subconjunto dos anticorpos disponíveis) e dependendo do resultado se decide (ou não) executar uma outra fase com um subconjunto adicional de anticorpos.

Um máximo de oito marcadores (atributos) por vez são permitidos no exame devido a limitações físicas do citômetro. Portanto, os 24 marcadores foram divididos em 5 grupos, chamados tubos. Esta divisão está exposta na Tabela 3.3. Os marcadores em cada tubo foram definidos em um estudo do consórcio *EuroFlow* anterior de modo a minimizar utilização de múltiplos tubos no diagnóstico (33).

Tubos	Marcadores presentes
Tubo 1	CD19*, CD20*, CD45*, CD5, CD38
Tubo 2	CD10, CD19*, CD20*, CD200, CD23, CD43, CD45*, CD79b
Tubo 3	CD11c, CD19*, CD20*, CD31, CD45*, CD81, IgM, LAIR1
Tubo 4	CD103, CD19*, CD20*, CD22, CD45*, CD49d, CD95, CXCR5
Tubo 5	CD19*, CD20*, CD27, CD39, CD45*, CD62L, HLADR,

Tabela 3.3: Marcadores por tubo. Os atributos marcados com "*" representam os *backbones*.

Os marcadores que se repetem nos tubos são chamados de *backbones* e são utilizados para identificar as células que mais se assemelham em cada rodada de análise. Com essa informação, é formado o painel completo (conjunto de todos os marcadores) de cada paciente (34).

O fluxo de trabalho funciona na seguinte ordem: se o resultado for múltiplo e entre as categorias de linfomas (classes) se encontra CLL ou MCL, o algoritmo adiciona os atributos do tubo 2 e cria uma nova árvore considerando somente as observações correspondentes às classes nas quais se ficou em dúvida (resultado da primeira árvore). O mesmo acontece se entre as classes do primeiro resultado está HCL. Mas, neste caso, serão adicionados os atributos presentes no tubo 3. É possível que seja necessário inserir os marcadores dos dois tubos simultaneamente, se entre os possíveis resultados se encontre HCL e CLL ou MCL. Se, após a nova árvore ainda não houver uma resposta única ou se o resultado múltiplo não incluir nenhuma das três doenças, então todos os marcadores são utilizados. Seguindo este princípio, a ideia é somente utilizar os marcadores necessários para identificação da categoria de linfoma e somente adicionar mais marcadores se os anteriores não forem suficientes para atribuir ao paciente uma única categoria. A Figura 3.4 diagrama o fluxo de

trabalho descrito.

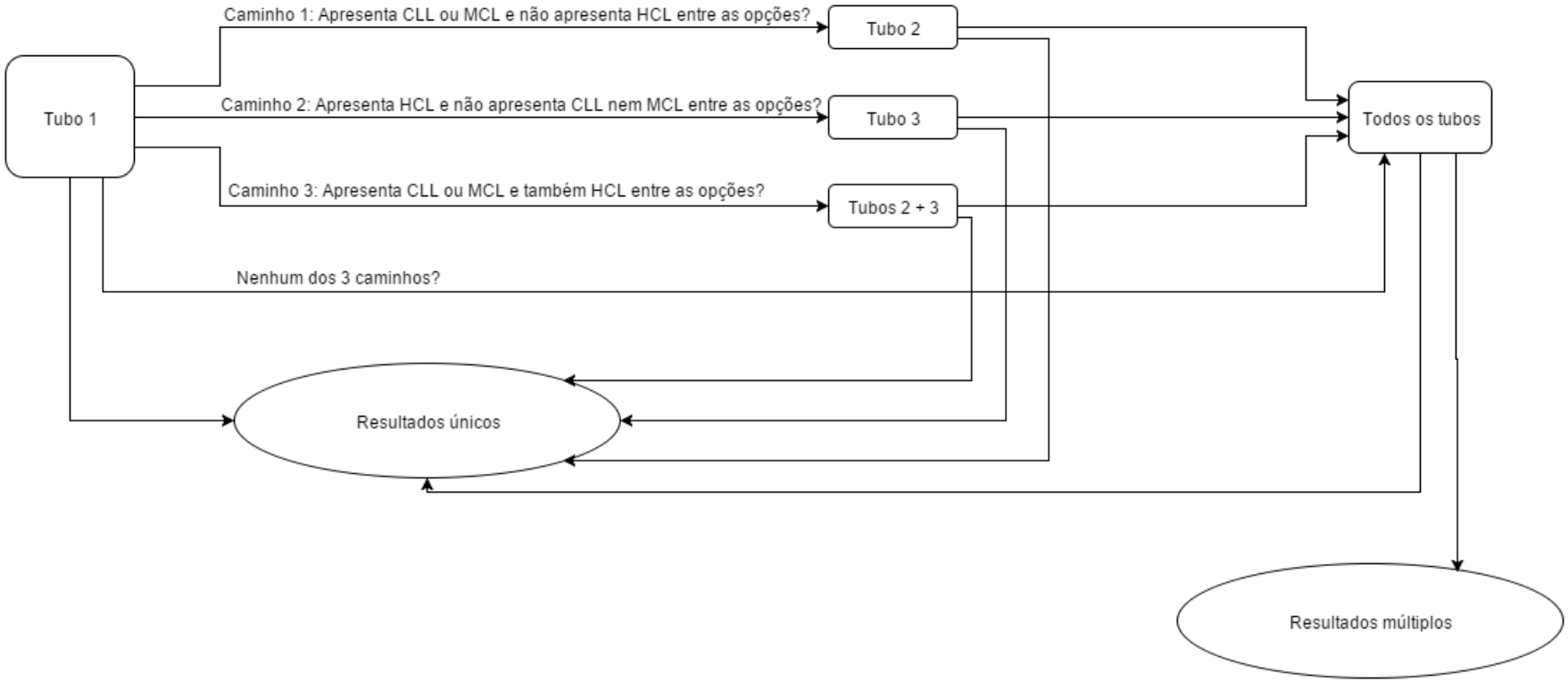


Figura 3.4: Fluxograma emulando o exame de citometria de fluxo.

Capítulo 4

Resultados e Discussões

Neste capítulo são apresentados os resultados obtidos ao implementar os esquemas detalhados no capítulo anterior. Foram projetados dois classificadores: o primeiro utilizou todos os marcadores enquanto o segundo implementa o esquema descrito na seção 3.4.

4.1 Metodologia utilizando todos os marcadores

Nesse capitulo não só se apresenta e se discute os resultados obtidos mas como se descreve brevemente as técnicas utilizas para fazer os testes fora da amostra. Utilizando a metodologia descrita nas Seções 3.2 e 3.3, foi construída uma árvore utilizando todas as observações existentes (Figura 4.1). Brevemente descrevendo para que o capítulo fique autocontido, em cada nó uma classe ou grupo de classes é separado das outras, até que se obtenha uma classe por nó. Durante o teste, a observação a ser testada começa a ser classificada pela raiz da árvore, e segue pela ramificação à esquerda ou à direita, de acordo com o resultado do modelo linear encontrado no nó. A classificação da observação estará terminada quando se chegar a uma folha.

O resultado geral dentro da amostra foi de 78.4% de acerto com classes únicas e 21.6% com respostas múltiplas, acertando a classificação de todas as observações. Foram escolhidos em seguida três métodos de validação fora da amostra. Os três marcadores de maior influência em cada divisão da árvore podem ser vistos na Tabela 4.1, juntamente com o total de atributos utilizados em cada separação. Nota-se que conforme a árvore vai se especializando, são necessários mais marcadores para separar as classes, ou seja, eles começam a se dispersar muito, utilizando muitos marcadores com pesos baixos. Na divisão CD10+ x FL todos os marcadores são utilizados. Esse comportamento será discutido mais a frente.

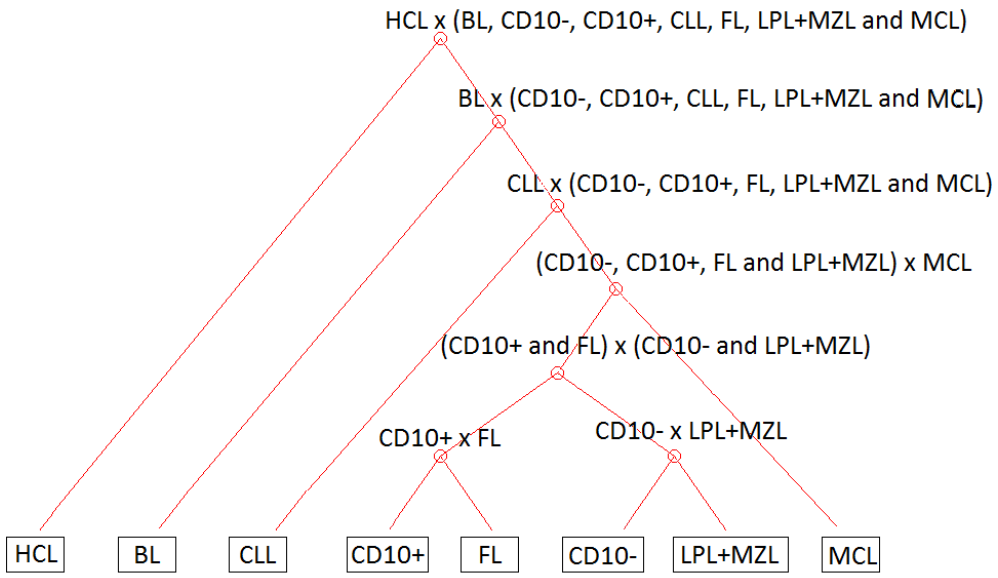


Figura 4.1: Árvore completa.

4.1.1 Validação cruzada e *leave-one-out*

Os dois primeiros métodos constroem a árvore utilizando o algoritmo de validação cruzada *leave-one-out*. A diferença está na maneira em como as comparações finais 2 a 2 são construídas (como descrito no pseudo-código 5). A primeira maneira utiliza validação cruzada em *30-folds*. Deste modo, um único grupo de comparações é construído para todas as observações a serem testadas. O segundo método utiliza *leave-one-out* também nessa parte, ou seja, para cada uma das 283 observações será construído um conjunto de comparações. Os resultados podem ser comparados nas Tabelas 4.2 e 4.3. A principal diferença entre os dois resultados é a quantidade de resultados múltiplos. Apesar do segundo método produzir um número maior de classificações únicas e corretas, também apresenta um número maior de respostas erradas. Como esta abordagem não utiliza o banco de dados completo para criar as comparações finais (Seção 3.3), ela trabalha com menos informações e a região de incerteza fica menor, pois há menos dados conflituosos. Proporcionalmente, a generalização foi positiva, uma vez que há mais classificações corretas e únicas do que resultados errados. No entanto, o primeiro método é mais apropriado se o objetivo é evitar falsos negativos, requisito chave neste projeto. Estando em posse do banco de dados completo, é possível aprender mais características comuns a diferentes classes, aumentando a região de incerteza. A matriz de confusão da Tabela 4.4 é uma matriz de confusão um pouco diferente da tradicional. Nela, cada linha representa 100% das observações da classe e a sua distribuição em resultados únicos, múltiplos e errados entre todas as classes. Os quadrados de fundo branco representam a porcentagem de acertos únicos da classe, os quadrados de cor cinza claro representam os acertos múltiplos e os quadrados de fundo escuro representam as indicações erradas.

Separação	Top 3 marcadores	Total de marcadores
HCL x (BL, CD10- , CD10+, CLL, FL, LPL+MZL e MCL)	LAIR1 (37.6%) CD103 (33.2%) CD22 (13%)	5
BL x (CLL, CD10- , CD10+, FL, LPL+MZL e MCL)	CD38 (32.5%) CD45 (27%) CD20 (11.2%)	8
CLL x (CD10- , CD10+, FL, LPL+MZL e MCL)	CD200 (25.6%) CD20 (24.9%) CD43 (12%)	13
(CD10- , CD10+, FL e LPL+MZL) x MCL	CD5 (22%) CD62 (14.3%) CD95 (13.1%)	17
(CD10+ e FL) x (CD10- e LPL+MZL)	CD43 (23.6%) CD10 (21.1%) CD27 (11.4%)	22
CD10+ x FL	LAIR1 (29.7%) CD43 (15.2%) CD200 (14.4%)	24
CD10- x LPL+MZL	CD43 (21.4%) CD200 (20%) CD11 (11.7%)	20

Tabela 4.1: Marcadores de maior peso em cada divisão.

Fora da amostra usando validação cruzada				
	Classificação correta	Classificação múltipla	Classificação errada	Total por classe
BL	12 (80%)	1 (6.7%)	2 (13.3%)	15 (100%)
CD10-	4 (16.7%)	16 (66.6%)	4 (16.7%)	24 (100%)
CD10+	7 (23.3%)	20 (66.7%)	3 (10%)	30 (100%)
CLL	43 (100%)	0 (0%)	0 (0%)	43 (100%)
FL	7 (19.5%)	26 (72.2%)	3 (8.3%)	36 (100%)
HCL	33 (100%)	0 (0%)	0 (0%)	33 (100%)
LPL+MZL	35 (53%)	30 (45.5%)	1 (1.5%)	66 (100%)
MCL	31 (86.1%)	4 (11.1%)	1 (2.8%)	36 (100%)
Total por tipo de classificação	172 (60.8%)	97 (34.3%)	14 (4.9%)	283 (100%)

Tabela 4.2: Resultado fora da amostra utilizando validação cruzada na construção das comparações finais

Fora da amostra usando *leave-one-out*				
	Classificação correta	Classificação múltipla	Classificação errada	Total por classe
BL	13 (86.6%)	1 (6.7%)	1 (6.7%)	15 (100%)
CD10-	7 (29.2%)	13 (54.1%)	4 (16.7%)	24 (100%)
CD10+	14 (46.6%)	11 (36.7%)	5 (16.7%)	30 (100%)
CLL	42 (97.7%)	0 (0%)	1 (2.3%)	43 (100%)
FL	11 (30.5%)	20 (55.6%)	5 (13.9%)	36 (100%)
HCL	33 (100%)	0 (0%)	0 (0%)	33 (100%)
LPL+MZL	36 (54.6%)	28 (42.4%)	2 (3%)	66 (100%)
MCL	31 (86.1%)	3 (8.3%)	2 (5.6%)	36 (100%)
Total por tipo de classificação	187 (66.1%)	76 (26.8%)	20 (7.1%)	283 (100%)

Tabela 4.3: Resultado fora da amostra utilizando *leave-one-out* na construção das comparações finais

Valor verdadeiro \ Valor previsto	BL		CD10-		CD10+		CLL		FL		HCL		LPL+MZL		MCL	
BL	80	3.3	0	0	13.4	3.3	0	0	0	0	0	0	0	0	0	0
CD10-	4.2	2.1	16.7	28.8	1.4	4.5	0	1	1.4	0	0	0	12.5	23.2	1.4	2.8
CD10+	3.3	7.3	1.7	6.1	23.3	26.1	0	0.7	3.3	19.4	0	0	1.7	5.2	0	1.9
CLL	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
FL	0	0	0.9	0.7	2.8	35	0	0.9	19.4	35	0	0	3.7	0.7	0.9	0
HCL	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
LPL+MZL	0	0	0	19.8	0	0.9	0	0	0	1.6	0	0	53	22.1	1.5	1.1
MCL	0	0	2.8	0	0	0	0	1.4	0	1.9	0	0	0	3.2	86.1	4.6

Tabela 4.4: Matriz de confusão modificada: a porcentagem de participação da classe em resultados múltiplos (fundo cinza claro) e errados (fundo escuro) é indicada. Os quadrados brancos indicam a porcentagem de acertos únicos. Cada linha soma 100%.

Uma característica importante desta árvore é que a separação em dois grupos nos nós deram preferência a separar uma categoria das demais. Analisando os critérios expostos na Seção 3.2.1 para essa escolha, em 4 dos 5 nós, a escolha é determinada pelo critério 3 (menor quantidade de atributos). Isso porque, devido à alta dimensionalidade dos atributos em relação à quantidade de observações, é possível encontrar modelos cuja a separabilidade é bem alta, chegando a quase 100% na maioria dos casos. Como esta separabilidade está sendo arredondada em duas casas decimais, as separabilidades empatam, sendo decidido o modelo utilizando o critério 3, pois são necessários menos atributos para identificar as características de uma patologia do que uma mistura de diferentes patologias com características próprias. O único caso em que isso não acontece é na divisão (CD10+ e FL) x (CD10- e LPL+MZL). Neste nó, essa separabilidade é maior do que outras opções de divisão do nó, mesmo com o arredondamento. As categorias FL e CD10+ permanecem juntas até a última folha da árvore e possuem uma extensa região de incerteza, conforme apontado na Figura 4.2, o que indica uma semelhança forte entre essas categorias. De fato, há estudos que indicam a transformação do *Follicular Lymphoma* em *Diffuse Large B-cell Lymphoma* (35). Como a metodologia proposta permite classificar uma observação como pertencente a mais de uma classe, é possível utilizar o nível de interseção entre as classes e calcular o grau de transformação entre elas em que a observação se encontra. Esta dissertação não possui este objetivo e seriam necessários dados rotulados para confirmar tal proposição. Mas é uma hipótese levantada a partir da análise dos resultados desse trabalho que pode ser explorada em trabalhos futuros.

4.1.2 Validação com nova base de dados independente

O terceiro método para validação fora da amostra consiste em testar o modelo construído com todas as observações (árvore da Figura 4.1) com uma base de dados completamente independente, não utilizada em nenhuma fase no processo de treinamento. Esta base consiste em um conjunto de 96 novos pacientes, novamente cedidos pelo EuroFlow, distribuídos de acordo com a Tabela 4.5.

Analisando esta tabela nota-se algumas classes com nenhuma observação. O teste foi feito considerando somente as observações existentes, não sendo adicionadas observações extras ou simuladas nas classes que não possuem nenhuma observação. Os resultados desse experimento podem ser vistos na Tabela 4.6.

Recalculando os resultados fora da amostra anteriores considerando somente o universo de observações pertencentes às classes existentes nos novos pacientes foi obtido um índice de performance geral semelhante, isto é, excluindo de todos os testes as classes BL, CD10+ e CD10-. A comparação está exposta na Tabela 4.7.

O fato do resultado geral ser semelhante indica que a estimativa de erro fora

Classes	Observações
BL	0
CD10-	0
CD10+	0
CLL	28
FL	17
HCL	11
LPL+MZL	6
MCL	34

Tabela 4.5: Quantidade de observações por classe (novos pacientes)

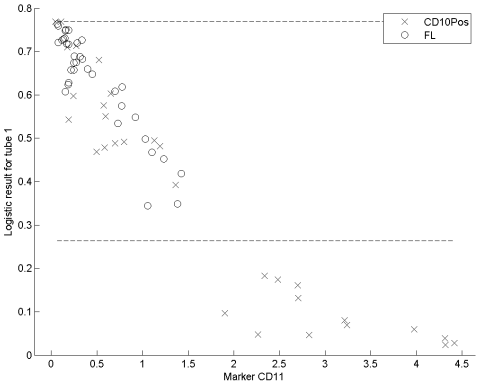
Fora da amostra para novos pacientes				
	Classificação correta	Classificação múltipla	Classificação errada	Total por classe
CLL	22 (78.6%)	3 (10.7%)	3 (10.7%)	28 (100%)
FL	5 (29.4%)	11 (64.7%)	1 (5.9%)	17 (100%)
HCL	11 (100%)	0 (0%)	0 (0%)	11 (100%)
LPL+MZL	2 (33.3%)	3 (50%)	1 (16.7%)	6 (100%)
MCL	31 (91.2%)	3 (8.8%)	0 (0%)	34 (100%)
Total por tipo de classificação	71 (74%)	20 (20.8%)	5 (5.2%)	96 (100%)

Tabela 4.6: Resultado fora da amostra para novos pacientes.

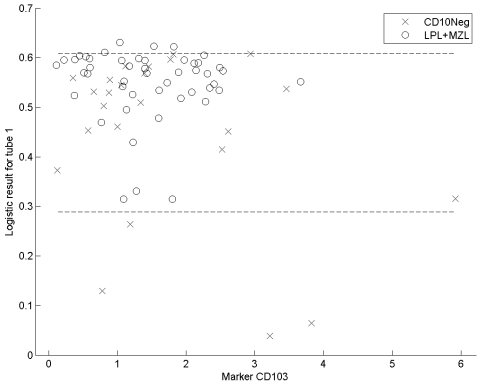
da amostra dada pela validação cruzada está próxima do observado na prática e pode ser estendida para as classes que não puderam ser avaliadas com dados novos. Quando analisamos as classes individualmente, é possível notar uma perda de performance no linfoma CLL. As categorias LPL+MZL também tiveram a acurácia diminuída. Contudo, a amostra dessa classe é muito pequena para que os valores em porcentagens possam ser comparados. As outras classes se mantiveram equilibradas com os resultados anteriores.

Comparação dos métodos fora da amostra			
	Classificação correta	Classificação múltipla	Classificação errada
Validação Cruzada	71.5%	23.8%	4.7%
Leave-one-out	69.7%	28%	2.3%
Novos casos	74%	20.8%	5.2%

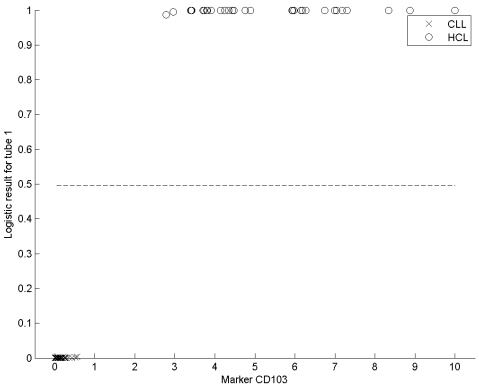
Tabela 4.7: Comparação dos resultados de cada método de validação considerando somente as classes CLL, FL, HCL, LPL+MZL e MCL.



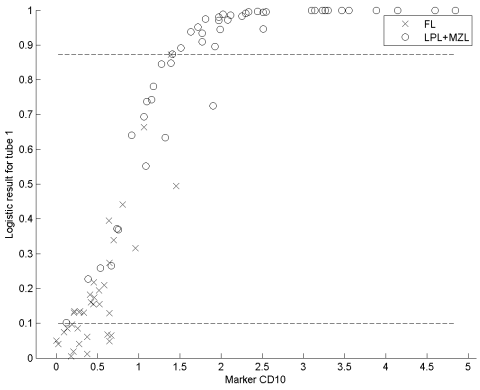
(a) Resultado da função logística para a comparação de classes CD10+ e FL. Repare que existe uma grande área em que a patologia CD10+ é possível de ser identificada, mas que há uma extensa região de incerteza entre as categorias. A área em que existe somente observações FL é pequena.



(b) Resultado da função logística para a comparação de classes CD10- e LPL+MZL. Há uma região clara para a classificação única de observações em CD10- ou em LPL+MZL, mas a região de incerteza é bem densa, apesar de não ser extensa.



(c) Resultado da função logística para a comparação de classes CLL e HCL. Não há região de incerteza entre essas classes. As classes são altamente separáveis.



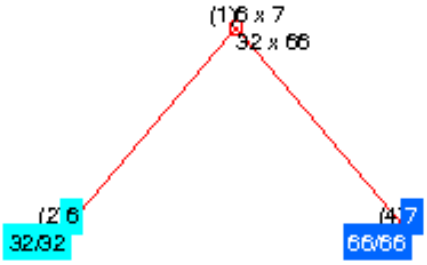
(d) Resultado da função logística para a comparação de classes FL e LPL+MZL. Região de incerteza extensa, mas pouco densa. A maior parte das observações é separável.

Figura 4.2: Resultado da função logística para comparação das categorias em pares. Identificação da região de incerteza e sua densidade. As figuras foram plotadas na dimensão de maior separabilidade x resultado da logística.

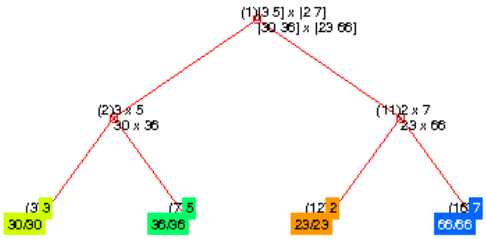
4.2 Metodologia utilizando conjuntos pré-determinados de marcadores (esquema de tubos)

A segunda implementação simula o procedimento do exame de citometria de fluxo, empregando parcialmente os marcadores em diferentes fases do processo de classificação, como explicado na Seção 3.4. O resultado dentro da amostra foi idêntico ao experimento anterior. No entanto, neste teste começou-se trabalhando em $\mathbb{R}^5$, aumentando até 5 dimensões (3 marcadores dos 8 incluídos são repetidos) por rodada. Logo, a dimensionalidade não era considerada alta para a quantidade de observações existentes, diminuindo a probabilidade de *overfitting*. Algumas das árvores construídas utilizando somente os marcadores dos tubos podem ser vistas na Figura 4.3. Os números representam a ordem em que as classes foram expostas na Tabela 3.1.

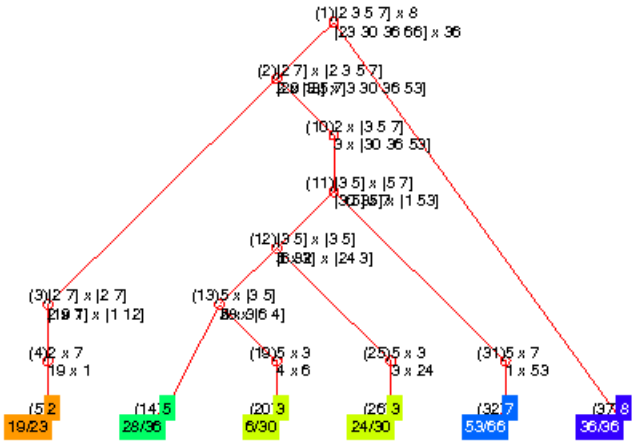
O teste fora da amostra foi realizado também com a técnica de *leave-one-out*, incluindo a construção da região de incerteza nas folhas. O resultado pode ser visto na Tabela 4.8. Cerca de 24.7% de todos os pacientes podem ser separados utilizando somente os 5 marcadores do tubo 1 e mais 21.2% com os marcadores dos tubos 1 e 2, totalizando 45.9% do total. Eles representam 82.7% dos pacientes considerando somente as classes BL, CLL, HCL e MCL, classes de maior separabilidade das demais. As classes que necessitam mais marcadores são as que apresentam uma quantidade maior de resultados múltiplos e cuja região de incerteza é grande. Portanto, seguindo o fluxo de trabalho, o método acaba por introduzir todos os marcadores para garantir a utilização de toda informação disponível antes de apresentar o resultado final.



(a) Árvore construída utilizando os marcadores dos tubos 1 e 3. Após o primeiro resultado, foram indicadas como possibilidades de diagnóstico as patologias HCL e LPL+MZL, representadas pelos números 6 e 7, respectivamente. Portanto, o algoritmo seguiu o caminho 2 do fluxograma (Figura 3.4), adicionando os marcadores do tubo 3 para chegar ao resultado.



(b) Árvore construída utilizando todos os marcadores. No entanto, 4 das 8 patologias foram eliminadas como possíveis diagnóstico em etapas anteriores do processo.



(c) Árvore construída utilizando marcadores dos tubos 1 e 2. Após o primeiro resultado, 3 das 8 patologias foram eliminadas como possíveis diagnósticos. A árvore seguiu então para o caminho 1 do fluxograma (Figura 3.4), construindo uma segunda árvore (a desta Figura) somente com as 5 patologias restantes.

Figura 4.3: Exemplos de árvores utilizando marcadores em tubos.

		BL	CD10-	CD10+	CLL	FL	HCL	LPL + MZL	MCL	Casos resolvidos por tubo
Tubo 1	Classificação correta	13 (86.6%)	0	0	32 (74.4%)	0	20 (60.6%)	0	4 (11.1%)	70 (24.7%)
	Classificação errada	0	0	1 (3.3%)	0	0	0	0	0	
Tubos 1+2	Classificação correta	0	3 (12.5%)	6 (20%)	10 (23.3%)	4 (11.1%)	0	6 (9.1%)	26 (72.2%)	60 (21.2%)
	Classificação errada	0	0	2 (6.7%)	0	2 (5.6%)	0	1 (1.5%)	0	
Tubos 1+3	Classificação correta	0	0	0	0	0	9 (27.3%)	0	0	12 (4.3%)
	Classificação errada	0	1 (4.2%)	1 (3.3%)	0	0	0	1 (1.5%)	0	
Tubos 1+2+3	Classificação correta	0	0	0	0	0	3 (9.1%)	4 (6.1%)	1 (2.8%)	8 (2.8%)
	Classificação errada	0	0	0	0	0	0	0	0	
Todos os tubos	Classificação correta	0	4 (16.7%)	8 (26.7%)	0	7 (19.4%)	1 (3%)	26 (39.4%)	0	133 (47%)
	Classificação múltipla	1 (6.7%)	13 (54.1%)	11 (36.7%)	0	20 (55.6%)	0	28 (42.4%)	3 (8.3%)	
	Classificação errada	1 (6.7%)	3 (12.5%)	1 (3.3%)	1 (2.3%)	3 (8.3%)	0	0	2 (5.6%)	
Total de casos por tipo de linfoma		15 (100%)	24 (100%)	40 (100%)	43 (100%)	36 (100%)	33 (100%)	66 (100%)	36 (100%)	283 (100%)

Tabela 4.8: Números de casos por tipo de linfoma resolvidos em cada grupo de tubos

Os resultados obtidos mostram que essa metodologia é completamente viável, uma vez que apresenta resposta equivalente a da Tabela 4.3. Esse resultado indica que é possível chegar a uma conclusão utilizando marcadores de maneira parcial, adicionando-os somente conforme a necessidade.

Capítulo 5

Conclusão

Nesta dissertação se propôs e se implementou um modelo de apoio ao diagnóstico de pacientes com linfomas. A arquitetura proposta realiza a construção de árvores de decisão para classificação de pacientes em uma ou mais categorias, oferecendo como resposta uma ou mais opções de linfoma. A representação dos dados em uma estrutura de árvore se encaixa muito bem com a ideia de interpretabilidade da solução oferecida, conforme mostrado nos estudos do capítulo 2. Ainda, a árvore associada à analise dos atributos relevantes em cada divisão, oferece ao médico uma visão mais detalhada do que acontece em cada etapa do processo. Em contrapartida, normalmente a utilização de mais de um atributo por nó torna a compreensão do resultado um pouco mais complicada. Porém, como este modelo está sendo proposto em um contexto específico de medicina, o modelo linear proveniente da utilização de múltiplos atributos por nó pode ser interpretado como um conjunto de marcadores que se possuírem valores altos simultaneamente caracterizam uma doença, oferecendo assim uma representação de fácil compreensão do modelo.

A utilização do Lasso como método de regressão logística se mostrou uma escolha apropriada, principalmente quando se chega mais perto das folhas, onde há menos observações e a seleção de atributos é essencial para diminuir o *overfitting*. O fato deste método estimar os pesos e realizar a seleção de atributos em conjunto, diminui a complexidade do algoritmo (não é preciso fazer cada passo separadamente) e produz pesos mais coerentes entre si, pois os mesmos critérios que levam a selecionar ou zerar determinados atributos são os critérios utilizados para estimar quais atributos terão os maiores pesos. O processo é feito em uma só etapa.

Os métodos de poda para evitar o *overfitting* da árvore foram de pré-poda, ou seja, critérios mais rígidos de parada da árvore, sendo o maior a exigência de um grupo puro. Como a árvore resultante possui somente uma folha por classe, não foi possível aplicar métodos pós-poda. Os resultados fora da amostra utilizando métodos de validação cruzada e *leave-one-out* foram compatíveis com os resultados utilizando observações não vistas no treinamento. A acurácia caiu de 78.4% dentro

da amostra para aproximadamente 70% nos três experimentos fora da amostra. Foram desenvolvidas duas alternativas para se evitar falsos negativos: criação de comparações individuais nas folhas das árvores e utilização do resultado da logística para criação de regiões de confiança e incerteza. Com essas medidas, foi possível indicar a categoria correta em aproximadamente 95% dos casos.

O resultado mais impactante deste trabalho é a possibilidade de construir modelos utilizando parcialmente os atributos em diferentes fases do processo. Foi mostrado que esta abordagem não só é viável, como produz resultados similares aos obtidos utilizando todos os atributos desde o início. Portanto, utilizando este método como apoio ao diagnóstico, é factível o uso de menos tubos no exame laboratorial, diminuindo o tempo de resposta do diagnóstico (o procedimento manual de separação e sumarização das células é encurtado) e o barateamento do exame. Com esta abordagem seriam economizados cerca de 36% dos tubos em comparação com o painel completo. Ainda, este método é de menor custo computacional, criando modelos finais mais simples e humanamente interpretáveis, além de iniciar com regressões em $\mathbb{R}^8$, aumentando a complexidade a cada rodada.

O trabalho apresentado cumpriu os requisitos delineados para esta aplicação. Como investigação futura é proposto verificar se as metodologias utilizadas aqui podem ser extrapoladas para outros ambientes e problemas. Os critérios de escolha da melhor separação de classes no nó foram escolhidos empiricamente, baseados em critérios encontrados na literatura. É possível que haja outros critérios que se encaixem melhor neste problema ou que sejam mais apropriados para a generalização do método. Outra opção de investigação consiste em explorar a curva ROC para definir a região de incerteza e de confiança do resultado da logística, de modo a maximizar a sensibilidade geral do algoritmo.

Referências Bibliográficas

[1] ROBINSON, J. P., ROEDERER, M., "Flow cytometry strikes gold", *Science*, v. 350, n. 6262, pp. 739–740, 2015.

[2] PEDREIRA, C. E., "Citometria de fluxo e outras Aplicações, Inteligência computacional a serviço da medicina", *Ciência Hoje*, v. 49, n. 291, pp. 40, 2012.

[3] ROBINSON, J. P., "Flow cytometry", *Encyclopedia of biomaterials and biomedical engineering*, v. 3, 2004.

[4] PEDREIRA, C. E., COSTA, E. S., LECREVISSE, Q., et al., "Overview of clinical flow cytometry data analysis: recent advances and future challenges", *Trends in biotechnology*, v. 31, n. 7, pp. 415–425, 2013.

[5] COSTA, E., PEDREIRA, C. E., BARRENA, S., et al., "Automated pattern-guided principal component analysis vs expert-based immunophenotypic classification of B-cell chronic lymphoproliferative disorders: a step forward in the standardization of clinical immunophenotyping", *Leukemia*, v. 24, n. 11, pp. 1927–1933, 2010.

[6] PEDREIRA, C. E., COSTA, E. S., ARROYO, M. E., ALMEIDA, J., ORFAO, A., "A multidimensional classification approach for the automated analysis of flow cytometry data", *Biomedical Engineering, IEEE Transactions on*, v. 55, n. 3, pp. 1155–1162, 2008.

[7] "EuroFlow", http://euroflow.org, Accessed: 2016-01-26.

[8] SZWARCFITER, J. L., *Estruturas de Dados e seus Algoritmos*. 2nd ed. Editora Campus, 1986.

[9] ROKACH, L., MAIMON, O., *Data Mining with Decision Trees: Theory and Applications*. 2nd ed., v. 81. World Scientific, 2014, Series in Machine Perception Artificial Intelligence.

[10] LOH, W.-Y., "Classification and regression trees", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, v. 1, n. 1, pp. 14–23, 2011.

[11] PILTAVER, R., LUŠTREK, M., GAMS, M., et al., "Comprehensibility of classification trees - survey design". In: *Proceedings of 17th International multiconference Information Society*, pp. 70–73, 2014.

[12] HUYSMANS, J., DEJAEGER, K., MUES, C., et al., "An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models", *Decision Support Systems*, v. 51, n. 1, pp. 141–154, 2011.

[13] HALL, M., FRANK, E., HOLMES, G., et al., "The WEKA data mining software: an update", *ACM SIGKDD explorations newsletter*, v. 11, n. 1, pp. 10–18, 2009.

[14] DEMŠAR, J., CURK, T., ERJAVEC, A., et al., "Orange: data mining toolbox in Python", *The Journal of Machine Learning Research*, v. 14, n. 1, pp. 2349–2353, 2013.

[15] HYAFIL, L., RIVEST, R. L., "Constructing optimal binary decision trees is NP-complete", *Information Processing Letters*, v. 5, n. 1, pp. 15–17, 1976.

[16] HANCOCK, T., JIANG, T., LI, M., et al., "Lower bounds on learning decision lists and trees", *Information and Computation*, v. 126, n. 2, pp. 114–122, 1996.

[17] MURTHY, S. K., "Automatic construction of decision trees from data: A multi-disciplinary survey", *Data mining and knowledge discovery*, v. 2, n. 4, pp. 345–389, 1998.

[18] ZANTEMA, H., BODLAENDER, H. L., "Finding small equivalent decision trees is hard", *International Journal of Foundations of Computer Science*, v. 11, n. 02, pp. 343–354, 2000.

[19] NAUMOV, G., "NP-completeness of problems of construction of optimal decision trees". In: *Soviet Physics Doklady*, v. 36, p. 270, 1991.

[20] BARROS, R. C., CERRI, R., JASKOWIAK, P., et al., "A bottom-up oblique decision tree induction algorithm". In: *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*, pp. 450–456, 2011.

[21] QUINLAN, J. R., "Induction of decision trees", *Machine learning*, v. 1, n. 1, pp. 81–106, 1986.

[22] QUINLAN, J. R., "C4. 5: Programming for machine learning", *Morgan Kauffmann*, 1993.

[23] BREIMAN, L., FRIEDMAN, J., OLSHEN, R., et al., "CART: Classification and regression trees", *Wadsworth: Belmont, CA*, v. 156, 1983.

[24] FAYYAD, U. M., IRANI, K. B., "On the handling of continuous-valued attributes in decision tree generation", *Machine learning*, v. 8, n. 1, pp. 87–102, 1992.

[25] FRIEDMAN, J. H., "A Recursive Partitioning Decision Rule for Nonparametric Classification", *IEEE Trans. Comput.*, v. 26, n. SLAC-PUB-1573-REV, pp. 404–408, 1976.

[26] ABU-MOSTAFA, Y. S., MAGDON-ISMAIL, M., LIN, H.-T., *Learning from data*. AMLBook, 2012.

[27] QUINLAN, J., *Decision trees and multi-valued attributes*. Oxford University Press, Inc., 1988.

[28] QUINLAN, J. R., "Simplifying decision trees", *International journal of man-machine studies*, v. 27, n. 3, pp. 221–234, 1987.

[29] PATIL, D. D., WADHAI, V., GOKHALE, J., "Evaluation of decision tree pruning algorithms for complexity and classification accuracy", *International Journal of Computer Applications (0975–8887)*, v. 11, n. 2, 2010.

[30] POLIKAR, R., "Ensemble based systems in decision making", *Circuits and Systems Magazine, IEEE*, v. 6, n. 3, pp. 21–45, 2006.

[31] POWELL, W. B., *Approximate Dynamic Programming: Solving the curses of dimensionality*. v. 703. John Wiley & Sons, 2007.

[32] TIBSHIRANI, R., "Regression shrinkage and selection via the lasso", *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[33] VAN DONGEN, J., LHERMITTE, L., BÖTTCHER, S., et al., "EuroFlow antibody panels for standardized n-dimensional flow cytometric immunophenotyping of normal, reactive and malignant leukocytes", *Leukemia*, v. 26, n. 9, pp. 1908–1975, 2012.

[34] PEDREIRA, C. E., COSTA, E. S., BARRENA, S., et al., "Generation of flow cytometry data files with a potentially infinite number of dimensions", *Cytometry Part A*, v. 73, n. 9, pp. 834–846, 2008.

[35] DAVIES, A. J., ROSENWALD, A., WRIGHT, G., et al., "Transformation of follicular lymphoma to diffuse large B-cell lymphoma proceeds by distinct oncogenic mechanisms", *British journal of haematology*, v. 136, n. 2, pp. 286–293, 2007.