Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação
Mestrado em Ciência da Computação

Dissertação de Mestrado

Comparação de Técnicas de *"Data Mining"* na Previsão da Precipitação

Gisele Azevedo de Araújo

Orientador: Prof. Dr. Fernando Carvalho Gomes

Fortaleza, Março de 2003

Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação
Mestrado em Ciência da Computação

Gisele Azevedo de Araújo

Comparação de Técnicas de *"Data Mining"* na Previsão da Precipitação

Dissertação apresentada ao curso de Mestrado em Ciência da Computação da Universidade Federal do Ceará como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Agradecimentos

Primeiro, agradeço a Deus, que sempre me guiou diante das dificuldades, iluminando sempre todos os caminhos.

Agradeço à minha família, que me apoiou durante este trabalho e que sempre tem me apoiado desde os princípios da educação. Agradeço também pelo interesse no que se referia à minha pesquisa, chegando a ler meus artigos me dando sugestões.

Ao Prof. Tavares, que me ajudou na formalização dos meus resultados e me motivou a continuar as pesquisas.

Ao meu orientador, pela indicação da área de pesquisa e pela boa orientação, que apesar do pouco tempo disponível conseguiu engrandecer os meus resultados.

Ao Prof. Stan Matwin, pelo incentivo dado a este meu aprimoramento.

À Prof. Sílvia, pelo apoio dado, principalmente, na fase de análise estaística dos resultados.

Abstract

This work describes and compares the performance of six Data Mining (DM) algorithms. It also describes a new method to reduce attributes for a precipitation forecasting problem. DM is a process to extract hidden, unknown and potentially useful information, in large databases, using it for decision making. The work shows possible solutions to forecast the rainy season of Boa Viagem, a city in the Ceará state. This region was selected because in its climate the precipitation variation affects the population and the investment results, mainly in the country side. Therefore, it would be very important to find a way to predict this variation. It could prevent possible problems because you get to know the information in advance. So, as a result, it intends to figure out which is the best suitable algorithm to obtain possible solutions and to develop a good technique to reduce attributes. In the previous related works, parametric methods were used, but DM had not been used in this specific region. If these works are performed very early in advance, the results will not accurate enough. The DM process was applied, using six Machine Learning techniques, with which solutions were compared; it was intended to increase the efficiency and decrease the forecasting delay. The algorithms used were C4.5, Naïve Bayes, Neural Networks, CART, One Rule and Support Vector Machine (SVM). The data set was formed by a temporal series from 1945 through 1989 (45 years). The attributes were precipitation, Sea Surface Temperature (SST), Sea Level Pressure (SLP), zonal (U3) and meridional (V3) wind components and sunspots. In each experiment, the available data was divided into training data and test data. The test set was composed of two ways: the first, with one third of the data set, and the second with cross validation. By reducing the attributes, 4232 values of SST are transformed into 6 values. Using all attributes the number of instances was 1.044. By comparing the methods, for the combined classes, the best method was SVM. A combination of values for predicted attributes defines a class, or, in other words, a class is defined by the condition of the attributes. A Ci class - in this context, a synonym for a precipitation occurred in the rainy quarter (February through May) - of an example is a subset of the training set S, consisting of all objects that satisfy the class condition: $C_i = \{o \in S | cond_i(o)\}$. Four sets of classes were used: C, with three classes, two classes and five classes. In the tests for three classes, the best methods were C4.5 and One Rule. To evaluate the performance of the individual solutions, we measured the relative absolute error, the root mean squared error, the root relative squared error and the percentage of correctly classified instances using ten-fold cross validation. Also, it was noticed the most important attributes for forecasting were SST groups. To obtain classes, the use of amplitude to divide the attribute into subsets was the best way. Summarizing, the forecasting results were good, but a certain bias was noticed towards the classes with the major number of examples.

Resumo

Este trabalho se propõe a comparar algoritmos de "Data Mining"(DM) e a desenvolver um método de redução de atributos para a previsão da precipitação pluviométrica. DM consiste no processo de extrair informação implícita, previamente desconhecida e potencialmente útil, a partir de grandes bases de dados, usando-as para tomada de decisão. O trabalho apresenta possíveis soluções sobre a previsão da estação chuvosa de Boa Viagem, cidade do Estado do Ceará. Esta região foi escolhida porque no semi-árido nordestino, a instabilidade da precipitação afeta a população local e os resultados de investimentos, principalmente na agricultura. Portanto, seria muito importante encontrar meios de prever estas variações. Isto diminuiria os possíveis problemas por antecipação. Então, como resultado do trabalho, visa-se averiguar qual dos algoritmos é o mais adequado na obtenção das possíveis soluções, assim como desenvolver uma boa técnica de redução de atributos. Nos trabalhos anteriores relacionados ao problema, foram usados métodos paramétricos e estatísticos, mas os métodos de DM (não-paramétricos) ainda não tinham sido usados nesta região específica. Os resultados dos trabalhos supracitados são deficientes quanto à precisão, em conseqüência da necessária antecedência da previsão. O processo de DM foi aplicado, usando seis técnicas de Aprendizagem Automática, com as quais compararam-se soluções, procurando-se aumentar a eficiência e diminuir o tempo de atraso da previsão. Os algoritmos utilizados foram C4.5, "Naïve Bayes", Redes Neurais, CART, Máquinas de Vetor Suporte (MVS) e "One Rule". A amostra foi formada por série de dados registrada no período de 1945 a 1989. Os atributos utilizados foram: nível de precipitação, temperatura na superfície do mar (TSM), pressão na superfície do mar, componente zonal e meridional do vento e manchas solares. As avaliações dos modelos resultantes foram efetuadas com conjuntos de testes construídos de duas formas distintas: a primeira, com 1/3 da amostra, e a segunda com validação-cruzada. Na redução realizada, 4232 valores de TSM são reduzidos a 6 valores. Na comparação feita entre os métodos com duas classes combinadas, a melhor aprendizagem foi feita pelo algoritmo MVS. A classe - aqui sinônimo de classificação da precipitação ocorrida na quadra chuvosa (fevereiro a maio) - de um exemplo é um subconjunto de um conjunto de treinamento S, consistindo de todos os objetos que satisfazem a condição de classe: $C_i = \{o \in S | cond_i(o)\}$. Foram usados quatro tipos de conjuntos de classes, são eles: C, contendo três classes, onde $C = \{seco, normal, chuvoso\}$, C_1 e C_2 contendo duas classes C, onde $C_1 = \{seco \cup normal, chuvoso\}$ e $C_2 = \{seco, normal \cup chuvoso\}$, e C_3 com cinco classes, onde $C_3 = \{muitoseco, seco, normal, chuvoso, muitochuvoso\}$. Para os testes com três classes, os melhores foram o CART e o C4.5. A avaliação baseou-se na análise da percentagem de acerto, erro quadrático médio, erro absoluto relativo e erro absoluto médio. Também verificou-se que os atributos mais importantes para as previsões foram os grupos de TSM. Em síntese, os resultados das previsões foram melhores do que os métodos utilizados atualmente, mas notou-se uma certa tendência nos resultados em direção às classes com mais exemplos.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

A disponibilidade de grandes volumes de dados em quase todas as áreas do conhecimento humano tem criado uma demanda por ferramentas. Estas devem ser capazes de utilizar estes dados para extrair conhecimento útil e inédito.

Para tentar satisfazer esta necessidade, métodos de Aprendizagem Automática, análises estatísticas de dados, reconhecimento de padrões, ferramentas de visualização de dados, entre outros, estão sendo utilizados. Estes esforços têm conduzido à emergência de uma área de pesquisa conhecida como *"Data Mining"*. *"Data Mining"* será melhor definido na Seção 2.1.

Os problemas aos quais *"Data Mining"* (DM) é passível de aplicação, podem ser oriundos de qualquer área, desde que se possuam uma quantidade considerável de dados confiáveis, ou seja, dados com um baixo percentual de ruído. Ruídos são incorreções ou falhas contidas nos dados. Estas falhas podem ser atribuídas a várias causas como, por exemplo, erro ou imprecisão na medição de valores das características. Outra causa de ruídos é em um mapeamento de um objeto do mundo real, para várias instâncias da linguagem de descrição da instância, pois, pode haver um erro de representação do objeto. Este tipo de erro não sistemático é usualmente referido como ruído. Algumas vezes, ruídos ocorrem nos valores de atributos ou nos valores de classes das instâncias [1]. Ruídos representam um problema para DM porque podem interferir na extração de um conhecimento válido de um conjunto de dados.

Considerando a abrangência do campo de atuação da nova área citada, o presente trabalho estabelece possíveis soluções para um problema de previsão de chuva no Nordeste brasileiro, mais especificamente, em Boa Viagem, uma cidade do Estado do Ceará, utilizando seis métodos de Aprendizagem Automática. Feito isto,

analisar e comparar as soluções respectivas.

Um problema de meteorologia foi escolhido, porque acredita-se que médias temporais mensais e sazonais de variáveis climáticas, possam ser previstas com precisão suficiente, de forma que se torne um benefício para aplicações sócio-econômicas, embora, detalhes diários de tempo não possam ser preditos com mais de alguns dias de antecedência [2]. Por exemplo, é impossível prever onde uma tempestade poderá ser gerada, mas é possível prever as mudanças de trajetória de tempestades no Atlântico [2]. Logo, é possível encontrar soluções para determinados problemas de previsão.

Esta informação deixa primariamente a premissa, que há um elemento determinístico com baixa freqüência de variabilidade nos atributos e interações de larga escala, entre estes atributos. Como em [3]: "Mesmo que sejam processos de pequena escala que governam os detalhes das interações ar - mar, é certamente um fenômeno de larga escala que no fim dirige em tempo e espaço trocas de calor, momentum (força ou produto da massa de um corpo e sua velocidade) e vapor d'água" [2].

A região "Sertão Central e Inhamuns" do Estado do Ceará foi selecionada devido à sua localização e ao seu clima. O clima é semi-árido, o qual é caracterizado por altas variações de precipitação ao longo do ano. Essa variação concentra-se em mudanças de espaço e quantidade, ou seja, os locais atingidos pelas chuvas e a quantidade de chuva verificada nestes locais, divergem muito. Historicamente, esta instabilidade tem afetado a população local, portanto, seria muito importante encontrar meios de prever estas variações. Isto diminuiria os possíveis problemas por antecipação da informação.

Adicionalmente, poderia ser útil prever secas ou inundações como uma maneira de evitar prejuízos. A estação chuvosa no Ceará é fortemente responsável por resultados positivos ou negativos de investimentos, principalmente na agricultura.

No escopo de DM, objetiva-se verificar a adequação e o comportamento de seis métodos não paramétricos, na solução do problema escolhido, analisando e comparando os resultados.

Em muitas áreas do conhecimento humano, ferramentas para gerar, extrair e armazenar dados têm sido desenvolvidas. Isto facilita o crescimento da quantidade de dados existentes. Por exemplo, em muitos supermercados, cada compra individual, gera automaticamente um registro em um repositório de informações. Logo, semanalmente, milhares de novos registros são gerados. Levando isto em conside-

ração, pode-se afirmar que grandes bases de dados já existem e estão aumentando de tamanho a cada minuto. Porém, na maioria destes repositórios, o potencial em conhecimento útil e escondido nos dados não é utilizado.

Objetivando extrair este potencial, uma demanda por ferramentas ou métodos tem surgido. Para satisfazer esta demanda, algumas soluções têm sido desenvolvidas, como por exemplo: métodos de Aprendizagem Automática, técnicas estatísticas e reconhecimento de padrões.

Um dos principais requisitos de DM para encontrar uma solução para um problema, é a existência de um repositório com dados confiáveis. Dados confiáveis são os que contém uma baixa porcentagem de ruídos. Entretanto, na maioria dos casos de aprendizagem, assume-se que o conjunto de exemplos fornecido é inteiramente correto. Porém, essa suposição é pouco provável de acontecer em dados do mundo real.

Define-se a seguir a estrutura da dissertação. No capítulo 2, descreve-se a fundamentação do trabalho e os conceitos necessários à sua realização. O capítulo 3 descreve a relevância do problema e relata trabalhos anteriores relacionados. O capítulo 4 apresenta o processo de *"Data Mining"* aplicado. No capítulo 5, os métodos que foram utilizados para resolver o problema são descritos. O capítulo 6 relata os resultados computacionais alcançados. Finalmente, o último capítulo, enumera as contribuições do trabalho e sugere trabalhos futuros.

Capítulo 2

Fundamentação Teórica e Estado da Arte

2.1 *"Data Mining"*

2.1.1 Descrição

"Data Mining" consiste no processo de extrair informação implícita, previamente desconhecida e potencialmente útil, a partir de grandes bases de dados, usando-as para tomada de decisão [4]. Basicamente, *"Data Mining"* é a pesquisa por descrições quando o conjunto de treinamento é um banco de dados (bd), ou o processo de derivar regras, onde um *bd* atua como o conjunto de treinamento. Em geral, este repositório é grande e grande parte do seu conteúdo não foi gerado com fins de aprendizagem. Estes dados tendem a possuir ruídos, e é freqüente a falta de valores para alguns atributos. Além disso, eles representam apenas um pequeno subconjunto do conjunto de estados possíveis, e o próprio sistema não pode manipular seu meio para gerar exemplos interessantes para melhorar a aprendizagem [1].

Um dos principais objetivos de DM é descobrir regras, relacionamentos e padrões globais inexplorados. DM propõe várias maneiras para se resolver um problema de classificação ou previsão, as quais combinam técnicas das seguintes áreas: Algoritmos, Inteligência Artificial, Aprendizagem Automática e Banco de Dados.

Nos dias de hoje, a quantidade de dados existente no mundo está aumentando rapidamente. Este crescimento rápido e demasiado pode dificultar a descoberta de informação valiosa, requerendo-se para esta descoberta, o auxílio de ferramentas cada vez mais poderosas. Portanto, DM está atraindo um interesse particular de

áreas científicas e comerciais, pois está ajudando a encontrar informação desconhecida. Resultados de extração usando DM têm sido encontrados com sucesso em diferentes campos como Biologia, Medicina, Telecomunicações, Computação, Climatologia, entre outros. Algumas das vantagens de DM são: rapidez na procura de uma boa solução, habilidade de combinar muitos atributos, uso de métodos não-paramétricos e suporte a grandes bases de dados. Por outro lado, há uma significativa variedade de métodos não-paramétricos e em muitos problemas, ainda é uma incógnita, qual método de DM é mais adequado para a sua solução.

"Data Mining" se diferencia de Aprendizagem Automática, porque as técnicas disponíveis nesta devem ser estendidas, para serem aplicáveis aos dados do mundo real. Em outras palavras, seus dados não possuem especificações de requisitos de dados sob a perspectiva dos objetivos da descoberta de conhecimento, antes que estes sejam coletados.

Esta informação válida escondida se constitui, basicamente, por relacionamentos e/ou padrões globais que existem em grandes bases de dados, mas que estão escondidos dentro da vasta quantidade deles. Por exemplo, relacionamentos entre dados de pacientes e diagnósticos médicos. Estes relacionamentos representam um valioso conhecimento sobre o banco de dados, os objetos contidos e o mundo real que o mesmo representa. Mas, isto só acontece, se esta base de dados, realmente, representar a realidade.

"Data Mining" tem evoluído como uma área de pesquisa importante e ativa, devido aos desafios teóricos e aplicações práticas, associados ao problema de descoberta ou extração de conhecimento útil e/ou interessante, e previamente desconhecido de grandes bases de dados do mundo real. Muitos aspectos têm sido investigados dentro de vários campos relacionados. Entretanto, existe a necessidade de se estender estes estudos, para incluir o domínio das bases de dados reais.

Um problema de DM é definido para enfatizar o desafio da pesquisa por conhecimento, em grandes bancos de dados e motivar pesquisadores e desenvolvedores de aplicação para resolver este desafio. Originou-se da idéia de que grandes bases de informações, podem ser vistas como verdadeiras "minas", contendo, possivelmente, informações valiosas que podem ser encontradas por eficientes técnicas de descoberta de conhecimento.

Estima-se que a quantidade de informação no mundo duplica a cada 20 meses [5]; ou seja, muitos sistemas de informações governamentais, científicos e

corporativos estão sendo inundados por um fluxo de informações que são geradas e armazenadas rotineiramente. Isto cresce dentro de grandes bancos de dados alcançando *"gigabytes"* e até mesmo *"terabytes"* de dados. Estes possuem uma "mina de ouro" de informações em potencial, mas, este potencial está além da capacidade de análise humana, por consistir em uma enorme massa de informação.

Dada a necessidade de uma análise destes dados, tem sido comum a prática da construção de aplicações *"on-line"* de bancos de dados ou a utilização de pacotes estatísticos em dados *"off-line"*, com um especialista do domínio para interpretação dos resultados. Aplicações *"on-line"* são as que requerem respostas imediatas e as *"off-line"* não possuem restrição de tempo de resposta. Nas aplicações *"on-line"* de bancos de dados, pode-se citar como exemplo as ferramentas OLAP (*"On-Line Analytical Processing"*). Ferramentas OLAP são aplicações de recuperação de informação para processamento analítico *"on-line"*, com a função de comparar e analisar padrões e tendências [6]. As principais desvantagens deste tipo de aplicação são a necessidade de um especialista, para analisar os resultados, e a incapacidade de descoberta de informação útil de forma automática. Porém, segundo a literatura, é melhor perseguir um objetivo geral em qualquer problema, do que desenvolver aplicações específicas para atender necessidades individuais de cada usuário.

O tipo de problema de DM definido anteriormente, simplesmente, combina todos os aspectos de descoberta de conhecimento no conceito de uma vasta quantidade de informações. Estas são representadas por relações, a qual é a estrutura predominante, tanto em Aprendizagem Automática quanto em Sistemas de Bancos de Dados. Cada tupla na relação corresponde a uma entidade, também conhecida como objeto ou instância. Entidades se constituem de atributos (também chamados campos ou características). O conjunto de dados é dividido dentro de um conjunto de treinamento e um conjunto de testes. O conjunto de treinamento é então usado para gerar algum conhecimento, e o conjunto de testes é usado para determinar a validade dos resultados e/ou refiná-los.

O processo de *"Data Mining"* consiste nas seguintes fases:

- definição de objetivos;
- seleção de dados;
- preparação de dados ou limpeza de dados, na qual dados incorretos, incompletos, esparsos ou redundantes são corrigidos ou descartados;

- transformação de dados, nesta fase os dados são transformados de forma a adequá-los à entrada dos algoritmos de aprendizagem;
- obtenção de padrões, cujo objetivo é descobrir o procedimento de classificação para os exemplos; e,
- análise de padrões, cujo modelo gerado é testado para estimar sua qualidade.

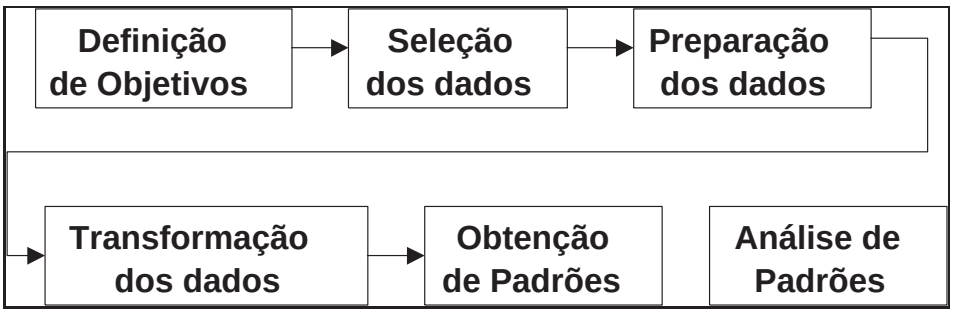

Figura 2.1: Fases do processo de *"Data Mining"*.

Um dos principais problemas de *"Data Mining"* é observado quando o número de relacionamentos possíveis é muito grande, e, portanto proíbe a pesquisa pelos relacionamentos corretos, apenas testando todas as possibilidades. Logo, existe a necessidade de estratégias de buscas heurísticas, o que é suprido por métodos de Aprendizagem Automática.

Aprendizagem Automática é um campo da Inteligência Artificial que estuda e desenvolve métodos automáticos capazes de aprender uma descrição útil de um conceito objetivo, quando são dadas instâncias do conceito [7].

Em um simples modelo de conceito de Aprendizagem, o mundo real fornece informações para o elemento de aprendizagem, que usa estas informações para melhorar a base de conhecimento. Finalmente, o elemento de performance usa a base de conhecimento para executar sua tarefa. O tipo de informação fornecida ao sistema pelo mundo real é usualmente imperfeita [7].

Uma desvantagem é que o elemento de aprendizagem não sabe previamente como completar detalhes perdidos, ou ignorar detalhes sem importância. Portanto, o sistema opera por suposição, recebendo respostas do elemento de performance. O mecanismo de retroalimentação habilita o sistema a avaliar suas hipóteses e revisá-las se necessário.

Aprendizagem Automática pode envolver dois tipos de processamento de informação: indutivo e dedutivo. No processamento de informação indutivo, padrões gerais e regras são obtidos através do conjunto de dados originais e da experiência. Por outro lado, no processamento dedutivo, regras gerais são usadas para determinar fatos específicos. A aprendizagem baseada na similaridade usa indução, enquanto que a prova de um teorema é a dedução de axiomas conhecidos e outros teoremas existentes. A aprendizagem baseada na explicação usa indução e dedução.

A importância das bases de conhecimento e as dificuldades experimentadas na aprendizagem têm conduzido para o desenvolvimento de vários métodos, para aumentar as bases de dados.

O tipo de aprendizagem usada neste trabalho foi a aprendizagem por exemplos, também conhecida como aprendizagem empírica ou aprendizagem baseada na similaridade.

Os métodos de aprendizagem por exemplos, possuem os seguintes aspectos fundamentais [7]:

- uma linguagem para representar as instâncias;
- uma linguagem para representar os conceitos;
- um algoritmo para formar a descrição do conceito a partir de um conjunto de instâncias classificadas, denominado algoritmo de aprendizagem; e
- um algoritmo para usar a descrição do conceito para classificar as instâncias desconhecidas, denominado algoritmo de classificação.

Considerando que o objetivo do presente estudo é o de "aprender procedimentos de classificação a partir de exemplos", dar-se-à agora a definição de cada uma dessas palavras. A linguagem humana sendo muito subjetiva, deixando margem à interpretações diferentes para um mesmo verbete, não poderia servir para definir os conceitos básicos de aprendizagem automática. Daí a utilização de uma linguagem matemática simples, como a que a seguir será empregada [8].

Um exemplo ou caso é um par (x, c) onde x representa a descrição de um caso, de um indivíduo, etc., e c representa uma classe. Os casos são descritos por um conjunto de atributos (ou sinais, temperatura do ar, etc.), representado por $V_p = \{x_1, x_2, ..., x_p\}$, onde cada um dos x_i tem seus valores num domínio D_i, podendo ser assimilado a $\{positivo, negativo\}$, $\{0, 1\}$, $\{seca, normal, chuvosa\}$, etc.

In casu, para descrever uma quadra chuvosa, foi usado um domínio diferente para cada atributo. A classe (aqui sinônimo de classificação da precipitação ocorrida na quadra chuvosa) do exemplo poderá ser representada por um inteiro $c \in E = \{1, 2, ..., C\}$, onde C designa o número de classes. A classe exprime a presença ou ausência de uma certa propriedade do objeto, etc. Quando existem apenas duas classes, diz-se que o exemplo é positivo ou negativo, de acordo com a classe.

Um exemplo binário é um exemplo descrito unicamente por atributos binários e cuja classe é uma variável binária. Seja $V_p = \{x_1, x_2, ..., x_p\}$ um conjunto de atributos binários, $L_p = \{x_1, x_2, ..., x_p\}$ é um conjunto de literais de V_p^2.

De um ponto de vista probabilístico, um exemplo (x, c) é a realização de uma variável aleatória (X, C) com valores em $D_1 \times D_2 \times ... \times ... \times D_p \times \{1, 2, ..., C\}$, onde p representa a dimensão dos exemplos. Em *"Data Mining"*, considera-se que não são conhecidas nem a regra de decisão $P(C/X)$, nem as propriedades do ambiente $P(X)$, nem mesmo a lei subjacente à distribuição $P(X, C)$. Para exprimir essa última restrição, localiza-se num contexto não paramétrico, ou *"distribution-free"*.

Uma amostra é um conjunto finito de exemplos obtidos de maneira independente, seguindo a distribuição $P(X, C)$. No curso deste trabalho, fala-se frequentemente de conjunto de aprendizagem ou de conjunto de teste. Trata-se de amostra de exemplos, como foi acima definido. [8]

Um procedimento de classificação (ou modelo) é uma aplicação F definida sobre X (descrição dos casos) e de valores em $\{1, 2, ..., C\}$. O objetivo da aprendizagem é descobrir o procedimento de classificação que governa os exemplos. De uma maneira ou de outra, os métodos de aprendizagem indutiva efetuam sempre uma exploração de um espaço previamente fixado de procedimentos de classificação viáveis. Esse espaço será representado por F. No caso dos exemplos binários, considerar-se-à que F é um conjunto de fórmulas lógicas, de forma predeterminada, expressas com ajuda dos operadores booleanos usuais $\wedge$ (e), $\vee$ (ou), $\neg$ (negação).

Aprender indutivamente, é ao mesmo tempo um processo de cálculo e uma exigência de capacidade preditiva, com respeito ao procedimento de classificação encontrado. O processo pode se resumir, como a seguir: dispõe-se de um conjunto de exemplos, ou conjunto de aprendizagem; define-se um conjunto F_4 de procedimentos de classificação; extrai-se de F um "bom" procedimento de classificação. O que se entende por bom? Por um lado, "bom" quer dizer que este procedimento classifica corretamente os exemplos do conjunto de aprendizagem. Mas isso não é tudo. Por

outro lado, deseja-se também que este procedimento tenha uma boa capacidade de predizer a classe de um exemplo que não está no conjunto de aprendizagem. Daí vem a exigência preditiva mencionada acima. Classificar corretamente um exemplo, através de um procedimento de classificação, significa que a classe do exemplo coincide com a classe dada pelo procedimento.

Resta-se definir o que se entende por: "boa capacidade de predizer". De fato, numerosas definições são possíveis. A que se dará a seguir é a mais simples, e uma das mais comuns.

A frequência de erro, ou taxa de erro, ou ainda taxa de erro aparente, de um procedimento de classificação F é dado por:

$$f_{err}(P) = \frac{err}{n},$$

onde, n representa o tamanho do conjunto de aprendizagem, e err é o número de exemplos do conjunto de aprendizagem que são incorretamente classificados por F.

A probabilidade de erro, ou taxa de erro real, é a probabilidade que um procedimento de classificação F classifique incorretamente um exemplo obtido de acordo com $P(X, C)$. Essa probabilidade será representada por $P_{err}(F)$. Portanto, aprender resume-se a:

1. Selecionar em F o procedimento de classificação que minimiza F_{opt}. Esse procedimento de classificação será representado por F_{opt}. A dificuldade dessa tarefa é de ordem algorítmica. Para certas classes de procedimentos de classificação, essa tarefa não será efetuada dentro de um tempo de cálculo aceitável. Dessa forma, contenta-se em encontrar um procedimento de classificação cuja freqüência de erro seja baixa, através de um método aproximativo de Inteligência Artificial (IA). Por uma questão de simplificação, continua-se representando essa aproximação por F_{opt}. Nessa dissertação mostra-se algorítmos indutivos de construção de árvores de decisão.

2. Ao mesmo tempo, fica-se satisfeito com o resultado do algorítmo indutivo somente quando se possui garantias de que $P_{err}(F_{opt})$ seja próxima de 0. A seguir dá-se as principais noções estatísticas, assim como um conjunto de resultados, que permitem o melhor entendimento do problema em questão e as restrições que ele induz

2.1.2 Aspectos Estatísticos

Considere-se um procedimento de classificação F, fixado a priori. A lei dos grandes números indica que, quando o tamanho n da amostra aumenta (aqui, n representa o número de casos), a frequência de erro de F, $f_{err}(F)$ converge para probabilidade de erro de F,$P_{err}(F)$. Isso de maneira totalmente análoga àquela que se observa quando, no lançamento de uma moeda, obtém-se cara ou coroa: quanto maior o número de lançamentos, mais a freqüência de obtenção de cara se aproxima de 0,5. Noutros termos, $f_{err}(F)$ se traduz numa boa aproximação de $P_{err}(F)$, principalmente quando n aumenta [8].

Infelizmente, esse raciocínio não é verdadeiro quando se trata do procedimento de classificação ótimo F_{opt}, selecionado através do processo de minimização da freqüência de erro. De fato $f_{err}(F_{opt})$, fornece uma visão parcial e frequentemente otimista da probabilidade de erro $P_{err}(F_{opt})$, isto é, ela é claramente mais próxima de zero. Por que razão isso se produz? Por que não se pode ao mesmo tempo selecionar com uma amostra de aprendizagem e julgar com essa mesma amostra.

Um meio de remediar esse problema é a separação de um conjunto independente chamado de amostra ou conjunto de teste. Da mesma maneira, para um procedimento de classificação fixado a priori, a freqüência de erro de F_{opt}, medida a partir de um conjunto teste, dá uma boa estimativa de $P_{err}(F_{opt})$, notadamente quando o tamanho do conjunto de teste aumenta.

Na prática em *"Data Mining"*, dispõe-se de conjunto fixo de exemplos onde se deve separar o conjunto de aprendizagem do conjunto teste. Uma primeira idéia simples consiste em dividir a amostra em duas, aprender com metade e testar com a outra metade. Esta não é a melhor solução, salvo se o número de exemplos for muito grande. Uma melhor solução tentaria obter amostras onde a porcentagem das diferentes classes fossem iguais. Vários autores se voltaram para esse problema e propuseram diferentes soluções para estimar a probabilidade de erro, permitindo que se use toda a amostra para aprender [8].

O processo de validação cruzada proposto por Breiman et al. [9], sempre fornece uma boa estimativa da probabilidade de erro. Adicione-se à notação de um procedimento de classificação F um parâmetro k, que indica uma medida de complexidade para F, assim, notar-se-á F_k. Esse novo parâmetro depende de características do espaço F. No caso de árvores de decisão (vide definição detalhada adiante) k pode ser a profundidade das árvores, o número de folhas, etc.

A partir de trabalhos teóricos de Vapnik e Chervonenkis [10], chega-se à conclusão que, quanto maior o parâmetro k, menor será a freqüência de erro medida sobre o conjunto de aprendizagem. No entanto, quando se trata do conjunto de teste, num primeiro momento a probabilidade de erro diminui com o aumento de k. Em seguida entra num patamar onde k aumenta e a probabilidade se mantém inalterada, para finalmente começar a aumentar lentamente com o aumento de k.

Em consequência, para estimar a probabilidade de erro a partir de uma simples amostra, deve-se levar em consideração a complexidade k do procedimento de classificação ótimo, por exemplo, $F_{k,opt}$. Anteriormente foram apresentados dois aspectos importantes no conceito de aprendizagem indutiva: o problema algorítmico, que é o de encontrar um procedimento de classificação que minimize a freqüência de erro, e o problema estatístico, que é o dotar o procedimento de classificação de poder preditivo vis-à-vis da probabilidade de erro. Agora adicione-se um elemento novo: para resolver os dois problemas acima deve-se levar em consideração as características do espaço de busca, com a finalidade de descobrir a complexidade ideal do procedimento de classificação ótimo. De um lado, o problema estatístico está associado à complexidade do procedimento de classificação ótimo. Por outro lado, a complexidade dos procedimentos de classificação que compõem o espaço de busca, influencia a forma de percorrer esse espaço, i.e. o problema algorítmico. Gomes e Gascuel [11] trata detalhadamente do problema algorítmico em aprendizagem indutiva.

2.1.3 Grande Volume de Dados

Uma das maiores preocupações em *"Data Mining"* está relacionada com o volume de dados. Isto ocorre porque muitas técnicas de descoberta de conhecimento, envolvendo pesquisa exaustiva sobre o espaço de instância, são altamente sensitivas ao tamanho dos dados. Este tamanho é avaliado em termos de complexidade e indução de padrões de compactação. Por exemplo, o algoritmo de eliminação de candidato [12], uma técnica de aprendizagem orientada a tupla de exemplos, aponta para uma pesquisa no espaço da versão. O tamanho deste espaço é exponencial no número de atributos de exemplos de treinamento, para induzir um conceito generalizado. O conceito é satisfeito por todos os exemplos positivos e nenhum dos exemplos negativos. Portanto, as técnicas dirigidas a dados, ou dependem de heurísticas, para guiar sua pesquisa através do largo espaço de relações possíveis entre combinações

de valores de atributos e classes, ou reduzem sua pesquisa horizontalmente ou verticalmente no espaço.

A redução horizontal está relacionada à fusão de tuplas idênticas, ou à substituição de um valor de um atributo, por seu valor de nível mais alto - (em uma hierarquia de generalização pré-definida, de valores categóricos do atributo [13]) - ou a discretização de valores contínuos. A redução vertical é realizada, ou aplicando algum método de seleção de características, ou usando grafos de dependência de atributos. Considera-se também como redução vertical, uma parte dos métodos, para manipular dados redundantes.

O mais simples procedimento de discretização é a divisão do limite de uma variável contínua, dentro de intervalos do mesmo tamanho, tanto quanto for o número de intervalos definidos pelo usuário. Uma variação deste método é o uso da teoria de entropia de Shanon, onde o esquema de entropia determina os limites dos intervalos, pelo ganho de informação de ocorrências observadas em cada intervalo igual. Este procedimento é chamado de método de quantização de intervalos num conjunto de informações. Uma desvantagem deste método é que há uma grande quantidade de informação perdida, por causa dos pontos de corte que poderiam, não necessariamente, ser nos limites das classes pré-definidas. Em outras palavras, seu critério de discretização cai, para levar em consideração o relacionamento entre as classes pré-determinadas e os limites de intervalos. Vale ressaltar, que a idéia central é reduzir o número de valores de atributos, sem destruir o relacionamento de interdependência entre a classe e os valores dos atributos.

2.1.4 Ruído

Assume-se que a informação fornecida em todo o conjunto de exemplos é totalmente correta. Entretanto, em dados reais, os sistemas de aprendizagem automática não podem fazer esta forte suposição. Os exemplos podem conter atributos baseados em julgamentos subjetivos e medidas; ambos são passíveis de erros nos valores dos atributos. Alguns deles podem estar classificados erroneamente. Erros deste tipo nos valores dos atributos ou na informação da classe, são usualmente referidos como ruído. Ruídos são erros não sistemáticos que ocorrem durante a entrada de dados ou durante a coleta.

Ruído pode causar dois problemas: o primeiro, geração de descrições incorretas utilizando-se um conjunto de treinamento contendo ruídos, e o segundo,

classificações incorretas, quando objetos são classificados, usando estas descrições incorretas.

Infelizmente, ainda existe pouco suporte nos SGBDs comerciais para eliminar ou reduzir erros, que ocorrem durante a entrada de dados, embora exista potencial para prover tal capacidade a modelos de dados relacionais. Logo, dados errôneos podem constituir um significante problema em bancos de dados do mundo real.

Se um conjunto treinamento está corrompido com ruído, seria desejável que o sistema fosse capaz de identificá-lo e ignorá-lo. A presença de ruído na informação de classe do conjunto de treinamento afeta a precisão de regras geradas; daí, uma tentativa poderia ser feita para eliminar o ruído, que afeta a informação de classe de objetos no conjunto treinamento.

Quinlan [14], executou experimentos para investigar o efeito do ruído na classificação de exemplos do conjunto teste. A idéia básica é que uma pequena quantidade de dados excepcionais é considerada ser causada por ruído, e, portanto pode ser negligenciada. Os resultados experimentais indicaram que alguns sistemas com adição substancial de ruído, diminuíram a taxa de erro na classificação de exemplos não vistos. Uma vez que as descrições tenham sido construídas a partir do conjunto de treinamento, estas descrições podem ser usadas como regras de classificação para determinar a classe de exemplos previamente não vistos. Em experimentos com alguns sistemas, a adição de ruídos em um certo conjunto de dados, resultou em degradação. Em outro conjunto, a adição da mesma quantidade de ruído resultou em baixos índices de classificação errada, de exemplos não vistos.

Um interessante fenômeno é que estas regras, aprendidas a partir de um conjunto de treinamento corrompido, executam melhor na classificação de dados ruidosos, quando comparados à regras que são aprendidas no mesmo conjunto, sem a presença dos ruídos. Portanto, não é vantajoso eliminar ruído de valores atributos de objetos no conjunto de treinamento, se existir uma quantidade significante de ruído, quando a regra de classificação induzida é usada na prática [1].

2.1.5 Dados Incompletos

Outro problema que pode ocorrer, é que os valores de atributos podem estar incompletos.

Supondo que cada objeto no universo é descrito ou caracterizado por valores de um conjunto de atributos, se a descrição dos objetos individuais é suficiente e

precisa em relação a um dado conceito, esta poderá, de forma não ambígua, descrever a classe, ou seja, um subconjunto de objetos, representando o conceito.

Entretanto, o conhecimento disponível em muitas situações práticas é freqüentemente incompleto e impreciso. O dado tem sido organizado e coletado acerca das necessidades de atividades organizacionais, causando dados incompletos do ponto de vista de tarefa de descoberta de conhecimento. Sob as circunstâncias, o modelo de descoberta de conhecimento poderia ter a capacidade de prover decisões aproximadas com algum nível de confiança.

Muitos métodos são propostos para aproximação de um conceito. Por exemplo, a teoria de conjunto *"fuzzy"* caracteriza um conceito, aproximadamente, por uma função de associação com um intervalo de valores entre 0 e 1. Outra abordagem é baseada na teoria dos conjuntos, que provém aproximações superiores e inferiores de um conceito, dependendo em como o relacionamento entre duas diferentes partições de um universo finito é definido.

Wong e Ziarko [15] demonstraram que a noção generalizada de conjuntos, pode adicionalmente ser convenientemente descrita pelo conceito de conjuntos *"fuzzy"*, quando operações próprias do conjunto de lógica difusa são empregados. Wong e Yao [16] introduziram uma estrutura teórica de decisão bayesiana, que provém uma unificação plausível das abordagens do conjunto *"fuzzy"* e de conjunto para aproximar o conceito.

Exemplos com atributos perdidos podem ser simplesmente descartados, ou uma tentativa pode ser feita para substituir o valor perdido pelo valor mais provável. Quinlan [17] sugere a construção de regras que predizem o valor de um atributo perdido, baseado no valor de outros atributos do exemplo, e na informação de classe. Estas regras são então usadas para completar valores de atributos perdidos. O conjunto resultante é usado para construir as descrições.

Outra abordagem é analisar valores desconhecidos com um valor separado, por exemplo, adicionar o valor "desconhecido" para o domínio de cada atributo, e usar este valor nas descrições [1].

As regras construídas podem ser usadas para classificar novos exemplos, nos quais os valores de atributos estão perdidos. Quando uma regra contém condições em alguns destes atributos, a mesma não pode ser aplicada.

Para resolver o problema descrito acima, existe a técnica de calcular a probabilidade que a regra determinada aplica. Essa probabilidade é o produto de pro-

babilidades de cada valor de atributo perdido, requerido na condição da regra. A probabilidade de que um atributo possua um determinado valor, pode ser estimada através da análise da freqüência relativa de valores para este atributo em exemplos no conjunto de treinamento.

2.1.6 Valores Nulos

O valor nulo pode aparecer em SGBDs como valor de qualquer atributo que não faça parte da chave primária, e é tratado como um símbolo distinto de qualquer outro símbolo, incluindo outras ocorrências de valores nulos. O valor nulo não significa somente um valor desconhecido, mas também um valor não aplicável. Em bancos de dados relacionais, este problema ocorre freqüentemente porque o modelo relacional dita que, todas as tuplas em uma relação deve ter o mesmo número de atributos, mesmo se valores de alguns atributos são inaplicáveis para algumas tuplas.

Estudos de modelagem de incerteza ou consultas aproximadas podem não ser diretamente ligadas a um problema de *"Data Mining"*, mas, certamente provém uma base para o processo de descoberta de conhecimento. Por exemplo, identificar relacionamentos probabilísticos em dados, pode ser útil em descobertas funcionais, em produção de relacionamentos, ou ainda, em regras entre os exemplos.

Existe uma abordagem na qual o valor é manipulado subdividindo-o dentro de três casos: desconhecido, inaplicável e nulo. É a abordagem de Lee [18] que estende o modelo relacional para informação incerta e imprecisa. Na abordagem de [18], a incerteza associada a um atributo é representada por uma probabilidade, usando uma distribuição de probabilidade no conjunto de seu domínio, ao invés do valor atômico. Um conjunto de valores é permitido para representação de valores imprecisos. Para cada instância, há uma relação de atributo de sistema, a qual consiste de um par de valores de crédito e plausibilidade acoplado, para mostrar o nível de confiança existente em cada tupla.

Já Quinlan [17] sugeriu para árvores de decisão indutivas, que quando bancos de dados contivessem valores de atributos perdidos, a solução seria: descarte dos valores ou tentativa de substituição com valores mais prováveis. Quinlan [19] sugeriu construir regras que predizem o valor perdido de um atributo, baseado no valor dos outros atributos do exemplo e da sua classe. Estas regras são usadas para completar os valores perdidos de atributos e o conjunto de dados resultantes poderia ser utilizado para construir descrições.

Apesar de apresentarem soluções para o problema de valores nulos, as abordagens apresentadas acima possuem uma desvantagem, descrita em [20]. Elas podem transformar uma dada tabela de decisão com valores desconhecidos, em uma nova e possivelmente inconsistente tabela de decisão, na qual todo valor de atributo é conhecido, por substituição de um valor desconhecido de um atributo com todos os possíveis valores deste atributo perdido. Com isto, o problema de valores perdidos foi reduzido a um problema de aprendizagem com exemplos inconsistentes. O autor usou a teoria de conjunto para induzir possíveis regras.

Outras duas abordagens para o problema de valores nulos são: atribuição de valores randômicos e a associação de medidas de probabilidade de perda com os valores perdidos.

2.1.7 Redundância

O conjunto de dados pode conter atributos redundantes ou insignificantes com respeito ao problema. Esta situação surge de várias maneiras. Como por exemplo, combinar tabelas relacionais para aumentar o conjunto de dados, isto possibilita a formação de atributos redundantes. Porém existem muitas soluções quase-ótimas, com complexidade de tempo razoáveis. Estas eliminam atributos insignificantes de um dado conjunto de atributos, usando pesos para atributos individuais ou para combinação de atributos. Este tipo de algoritmo é conhecido como algoritmo de seleção ou redução de características [21].

Seleção de característica, um processo de pré-poda na aprendizagem indutiva, é o problema de escolher um pequeno subconjunto de características que é necessário e suficiente para descrever o conceito objetivo. A importância da seleção de característica é não somente reduzir o espaço de pesquisa, mas, também aumentar a velocidade dos processos de aprendizagem de conceitos e de classificação de objetos. Além disso, melhora a qualidade da classificação do classificador [22, 23, 24, 25].

Procurar o menor subconjunto de características no espaço de características demanda tempo. Este tempo é limitado por $O(2^l J)$, onde l é o número de características, e J é o esforço computacional requerido para avaliar cada subconjunto. Este tipo de pesquisa exaustiva é apropriado somente se l é pequeno e J é computacionalmente barato.

Para soluções quase-ótimas em casos especiais, pesos de características individuais ou combinações de características são computados, com respeito a algum

critério de seleção de característica, tal como coeficiente Bhattacharya, divergência, distância variacional de Kolmogorov, critério de entropia de Shanon, precisão de classificação, entre outras [21].

2.1.8 Métodos de *"Data Mining"*

Freqüentemente, algoritmos muito simples conseguem classificar surpreendentemente bem. Esta idéia é um dos fundamentos de muitos algoritmos de mineração [26].

Existem diferentes tipos de estruturas simples, são elas: um atributo pode ser o mais importante; todos os atributos podem contribuir independentemente com igual importância; uma combinação linear pode ser suficiente; uma representação baseada em instância pode trabalhar melhor, e simples estruturas lógicas podem resolver o problema.

É válido ressaltar que o sucesso do método depende do domínio.

O aprendizado supervisionado é um processo de aprendizagem através de exemplos, e é somente possível na presença de alguma forma de retroalimentação. A retroalimentação pode ser usada para quantificar e, portanto, comparar a performance de um programa.

Aprendizagem para classificar a precipitação, dado um conjunto de treinamento de exemplos classificados por um humano, é um exemplo de aprendizagem supervisionada.

Neste caso, a retroalimentação necessária para monitorar a performance pode ser obtida de uma taxa de erro, ou percentagem de resultados de classificações, diferentes das classificações fornecidas.

Russel e Norvig [27] determinam que aprendizagem supervisionada pode ser vista como inferência pura indutiva, que induz algumas funções que aproximam a realidade sendo aprendida.

No contexto de classificação, um bom algoritmo de aprendizagem supervisionada é aquele que faz um bom trabalho de predizer a classe de novos exemplos, ou seja, produz boas hipóteses [27].

A qualidade de um algoritmo de aprendizagem supervisionada que produz um classificador, pode ser determinada pelos seguintes passos:

- Dividir o conjunto de exemplos disponíveis dentro de dois conjuntos distintos: o conjunto de treinamento e o conjunto de teste;

- Usar um algoritmo sob o teste para produzir um classificador, usando exemplos de um conjunto de treinamento, ou seja, treinar o algoritmo;
- Usar o classificador resultante nos exemplos no conjunto de teste gerando uma classe para cada exemplo;
- Comparar as previsões resultantes contra as classes atuais dos exemplos e observar a percentagem de instâncias classificadas corretamente no conjunto teste.

Quando existe um grande número de características em um problema, há uma desvantagem de usar a liberdade resultante para aprender classificadores. Isto explica-se, por se utilizar atributos sem significado ou irregulares do conjunto de dados, na construção dos classificadores.

Esse problema é comum em todos os tipos de algoritmos de aprendizagem e é conhecido como *"overfitting"* ou *"overtraining"*.

Técnicas simples existem para prevenir *"overtraining"* para a maioria dos algoritmos de aprendizagem, tal como árvores de decisão podando árvores de decisão e *"optimal brain damage"* para redes neurais [27].

2.2 Técnicas Estatísticas

Nos últimos anos, têm-se notado um aumento do uso de técnicas estatísticas no processo de *"Data Mining"*, como por exemplo: seleção de atributos [28], dependência de dados [29, 30], classificação de objetos baseados em descrições, discretização de valores contínuos [29], redução de dados [30], previsão de valores incompletos, etc.

A motivação da utilização estatística deve-se ao fato de que as técnicas estatísticas para análise de dados já são bem desenvolvidas e amadurecidas, e, em alguns casos, consistem no único meio de análise [1]. No entanto, algumas das desvantagens do uso de métodos estatísticos para problemas de análise de dados são: fortes suposições estatísticas, inabilidade de reconhecer e generalizar relacionamentos (tal como a inclusão de conjunto), e a captura de características estruturais de um conjunto de dados.

Estatística é a ciência que dispõe de processos apropriados para recolher, organizar, classificar, apresentar e interpretar conjuntos de dados [31].

A Estatística fornece técnicas para extrair informação dos dados, os quais são muitas vezes incompletos, na medida em que oferece informação útil sobre o problema em estudo.

É objetivo da Estatística extrair informação dos dados para obter uma melhor compreensão das situações que representam.

No estudo do problema escolhido, a Estatística foi utilizada nas fases de transformação de dados e análise de resultados, envolvendo métodos e medidas estatísticas.

Sob a forma de amostra, os dados recolhidos sofreram redução e transformação, o que será melhor explicado nas Seção 5.3 e 5.4. Estes dados foram representados utilizando-se grupos. Objetivou-se com isto, a identificação de uma estrutura subjacente aos dados, deixando de lado a aleatoriedade presente, permitindo-se ainda reduzi-los, de modo que, pode-se extrair o máximo de informação relevante para o problema em estudo.

Também, uma parte da potencialidade da Estatística foi utilizada, de maneira que conclusões foram feitas acerca da população escolhida, baseando-se numa pequena amostra, dando ainda uma medida do erro cometido.

População é um conjunto, agregado ou coleção de unidades individuais, que podem ser pessoas ou resultados experimentais, com uma ou mais características comuns [31].

Amostra é um subconjunto de dados ou observações, recolhido a partir de uma determinada população, de forma a se obter conclusões sobre a população de origem. Para medir a amplitude de uma amostra, calcula-se a diferença entre o valor máximo e o valor mínimo do conjunto de dados [31].

As amostras devem ser selecionadas de modo a serem representativas para a população, do contrário, as amostras podem ser enviesadas, ou seja, podem não representar corretamente a população e a sua utilização poderia dar origem a interpretações erradas.

Numa análise estatística distinguem-se basicamente duas fases: Estatística Descritiva, onde se descreve e estuda a amostra - que são as características principais e as propriedades - e a Estatística Indutiva, em que se conclui sobre a população ou se imagina proposições mais gerais, que exprimam a existência de leis na população.

O estudo descritivo dos dados de uma amostra ou de uma população, resume-se na informação contida no conjunto de dados. Construindo tabelas e gráficos foram

encontradas algumas características do conjunto de dados [31].

Já na Estatística Indutiva, conhecidas certas propriedades, se imaginam proposições mais gerais, que exprimam a existência de leis obtidas a partir de uma análise descritiva de amostras, expressas por meio de proposições. No entanto, ao contrário das proposições deduzidas, não se pode dizer que são falsas ou verdadeiras, já que foram verificadas sobre um conjunto restrito de indivíduos. Portanto não são falsas, mas não foram verificadas para todos os indivíduos da população, pelo que também não se pode afirmar que são verdadeiras. Existe assim, um certo grau de incerteza (percentagem de erro), que é medido em termos de probabilidade [31].

Sobre os dados que constituem uma amostra, pode-se classificar em dois tipos fundamentais: dados qualitativos e dados quantitativos.

Os dados qualitativos representam informações que identificam alguma qualidade, categoria ou característica, não susceptível de medida, mas de classificação, assumindo várias modalidades. O estado civil de um indivíduo é um exemplo disso, pois, é capaz de variar dentro das seguintes categorias ou classes: solteiro, casado, viúvo e divorciado.

Os dados quantitativos representam a informação resultante de características susceptíveis de serem medidas, apresentando-se com diferentes intensidades, que podem ser de natureza discreta ou contínua. No caso de uma variável contínua, esta pode tomar todos os valores numéricos, inteiros ou não, compreendidos no seu intervalo de variação - temos por exemplo o peso, a altura, entre outros [31].

No caso das variáveis contínuas, pode-se discretizá-las definindo-se classes, de forma a transformar os valores contínuos em valores discretos. O processo compõe-se de certas etapas principais:

1. Definição das classes: determinar a amplitude da amostra - a diferença entre o valor máximo e o valor mínimo da amostra;

2. Calcular a amplitude de classe h: dividindo a amplitude do item anterior pelo número k de classes objetivo; essa amplitude de classe é um valor aproximado por excesso do valor anteriormente obtido.

3. Construir as classes de modo que tenham todas a mesma amplitude e cuja união contenha todos os elementos da amostra.

A primeira classe C_1 será $C_1 = [c_1, c_2[=$ [min. da amostra, min. da amostra+ h[.

As outras classes C_i serão C_i = [min amostra + $(i-1) \times h$, min amostra + $(i \times h)$], com $i = 2, ..., k$.

Um valor aproximado, para um número de classes a serem consideradas, pode ser obtida através da seguinte regra empírica: para uma amostra de dimensão n, k é o menor inteiro tal que: $2^k > n$.

Para a representação gráfica de dados contínuos, usa-se um diagrama de áreas ou histograma, formado por uma sucessão de retângulos adjacentes, tendo cada um por base um intervalo de classe e por área a freqüência relativa (ou a freqüência absoluta).

Freqüência absoluta representa o número de elementos de cada uma das categorias ou classes. Freqüência relativa é igual a freqüência absoluta dividida pela dimensão da amostra.

Se freqüências absolutas forem utilizadas para construção do histograma, a área total será igual a :

$$A = n_1 + n_2 + .. + n_k = n$$

No caso de se utilizarem as freqüências relativas:

$$A = f_1 + f_2 + .. + f_k = 1$$

Uma consideração sobre histograma, é que seu aspecto depende em grande parte do agrupamento que se tenha feito para os dados. Assim, a escolha de uma amplitude de classe muito pequena, traduz-se num grande número de classes, que não permite que se sobressaiam as características fundamentais dos dados, uma vez que se lhe pode sobrepor o aspecto aleatório dos dados. Por outro lado, um número muito pequeno de classes, pode não mostrar alguns aspectos importantes dos dados. Para representar graficamente as freqüências acumuladas considera-se a função cumulativa cuja construção, se exemplifica a seguir:

- Antes do limite inferior da primeira classe, a freqüência acumulada é nula, pelo que se traça um segmento de reta sobre o eixo x, até esse ponto.
- No limite inferior da segunda classe, a freqüência acumulada é a freqüência da classe anterior.
- No limite inferior da terceira classe, a freqüência acumulada é a soma das freqüências das duas classes anteriores.

- Quando se atinge a última classe, tem-se a garantia que a freqüência acumulada correspondente ao seu limite superior é igual a 1, nesse ponto marca-se 1, e continua-se com um segmento de reta paralelo ao eixo x.
- Pode-se chamar a atenção para algumas propriedades da função cumulativa, tal como foi construída: está definida para todo o x real;
- É sempre não decrescente;
- Só assume valores no intervalo $[0, 1]$.

Os dados são organizados na forma de uma tabela de freqüências. No entanto, em vez das categorias, apresentam-se os valores distintos da amostra, que vão constituir as classes. Uma freqüência acumulada de 50% é chamada de mediana. A mediana divide a distribuição das freqüências em duas partes iguais, já que 50% dos dados são menores ou iguais à mediana e os restantes 50% são maiores ou iguais. Vale lembrar que com esta técnica utilizada, obtém-se um valor aproximado para a mediana, e não o valor exato da mediana do conjunto de dados originais.

2.3 O Problema de Previsão de Chuva

Diversas metodologias têm sido utilizadas com propósitos operacionais no Instituto Nacional de Pesquisas Espaciais (INPE) e na Fundação Cearense de Meteorologia e Recursos Hídricos (FUNCEME), para predizer a estação chuvosa do Nordeste. Estas metodologias são baseadas em parâmetros empíricos atmosféricos e oceânicos, nos resultados de modelos de simulações, e em modelos estatísticos específicos [32].

Por muitos anos, o Sol foi visto somente como uma estrela produtora de uma quantidade fixa de calor e luz, ou seja, uma estrela constante. Porém, variações no tempo ou nas estações do ano são primariamente produzidas devido à inclinação da órbita da Terra. No verão, o sol é mais alto ao meio-dia, enquanto que no inverno é mais baixo e aparece por um menor espaço de tempo. Entretanto, nem todos os verões são iguais, e nem todos os invernos são iguais. Alguns anos trazem seca, outros podem trazer inundação. Esta é uma preocupação vital para muitas pessoas, particularmente para os que dependem da terra para sobreviver. A partir daí, começou-se a descobrir que o sol não é verdadeiramente uma estrela constante e muitos começaram a imaginar se estas variações da órbita solar podiam ter algo

relacionado com as mudanças no tempo. Logo, surgiu a questão de que as manchas solares ou "chamas" solares ou outros fenômenos relacionados controlavam ou não o tempo. (O tempo citado refere-se ao tempo da troposfera - o qual aparece nos primeiros 10 km ou então na nossa atmosfera).

Várias pesquisas têm sido conduzidas em muitas décadas, na tentativa de relacionar as manchas solares e outras formas de atividade solar, com o tempo. O assunto é com freqüência popular à mídia. Incontáveis horas têm sido perdidas na tentativa de convencer o mundo que as secas e as cheias são conseqüências de uma inesperada explosão da fúria solar. Infelizmente, a cada artigo publicado mostrando a relação entre as cheias de um ano e as manchas solares, pode ser freqüentemente encontrado um outro artigo contraditório, mostrando que não há relação entre manchas solares e a seca do mesmo ano.

Alguns cientistas acreditam que deve existir uma pequena conexão entre os distúrbios do tempo e a atividade solar. Outros acreditam numa pequeníssima conexão [33]. A razão pela qual muitos cientistas têm dificilmente aceitado que a atividade solar tem uma maior influência, é simples: mesmo a grande erupção solar, sendo uma gigante explosão para os padrões da Terra, a atividade solar somente libera uma quantidade de energia, comparável à energia que o Sol emite em poucos segundos. Em outras palavras, a atividade solar é apenas uma variação muito pequena na variação total de saída solar. Muitas das grandes variações na luz solar recebida na superfície da Terra são devidas à inclinação da Terra e à sua órbita elíptica.

Para pessoas que querem um método prático de previsão do tempo, segundo o eminente meteorologista australiano Barrie Pittak, poderiam se basear no pensamento: "existe, atualmente, pouca ou nenhuma evidência convincente de correlações úteis com significância estatística ou prática entre o ciclo de manchas solares e o clima em escalas de tempo de médio prazo".

Esta conclusão parece justificar a existência de massiva literatura sobre o assunto. A literatura possui sugestões evidentes que, se mais dados e melhores análises fossem feitas, se poderia obter resultados bem-sucedidos na verificação de relações significantes, entre o tempo e as manchas solares. Relações estas, que, provavelmente, poderão explicar a variância da precipitação, por exemplo.

Existe evidência de reduzidíssimos efeitos de pequeno prazo (dentro de dias) e também alguns efeitos climáticos de longo prazo (da ordem de séculos). Entretanto,

nenhum destes oferecem muita esperança na previsão do tempo.

Capítulo 3

O Problema da Previsão de Chuva no Ceará - PPCC

No Ceará, em termos gerais, devido às suas características climáticas, pode-se afirmar que existem apenas duas estações por ano: a estação seca e a estação chuvosa. Isto é decorrente do fato, do regime de precipitação, se concentrar em apenas um período do ano. As duas estações citadas são responsáveis pelos resultados positivos ou negativos dos investimentos agropecuários. Logo, dada a importância destas "estações" neste Estado, objetiva-se prever a classificação do período chuvoso.

Levando em consideração que a quantidade de chuva do período restante do ano, representa cerca de 30% em relação ao total anual, foi assumido que estes períodos representariam as estações.

Este período é chamado de *quadra chuvosa*, pois, as chuvas anuais se concentram principalmente no espaço temporal de quatro meses, sendo: fevereiro, março, abril e maio. No entanto, é válido ressaltar que, a precipitação do Nordeste é esparsa e, normalmente, inicia-se em outubro. A classe da quadra a ser prevista pertencerá a um certo conjunto C.

Considerando-se somente esta alta e estreita concentração de chuva de fevereiro a maio, pode-se afirmar que a previsão do comportamento da quadra chuvosa determina a classificação da precipitação anual em todo o Estado. Portanto, se existisse uma forma de prever a quadra chuvosa, seria muito vantajoso para o planejamento geral da agricultura e pecuária cearense.

Diversos autores já estudaram a relação da precipitação nordestina com outros eventos naturais, como por exemplo, a relação da chuva do Nordeste com a Temperatura da Superfície do Mar. Uma das conclusões encontradas nestes estudos,

foi que o gradiente de pressão meridional é fortemente associado com as anomalias de precipitação do Nordeste. Neste estudo, observando-se o Atlântico, encontrou-se que as secas do Nordeste coincidem com a baixa fase da Oscilação Sul. A baixa na Oscilação Sul é definida pelas seguintes situações anormais: alta pressão em Darwin e baixa pressão no Tahiti, ou, então águas superficiais do mar excessivamente quentes no Pacífico Equatorial [34]. Outro exemplo foi a descoberta através de modelos climatológicos que o principal sistema causador de chuvas no Ceará é a Zona de Convergência Intertropical (ZCIT), ou seja, a região da Terra onde os ventos do hemisfério norte e sul se encontram e sobem, facilitando a formação das nuvens.

O prognóstico das chuvas realizado pela FUNCEME para o Estado do Ceará, baseia-se na análise dos seguintes dados: condições da geografia física regional (chuva, relevo e vegetação do Ceará); resultados gerados pelos modelos dinâmicos, estatísticos e estocásticos de previsão de clima e de Temperatura da Superfície do Mar (TSM), nas bacias dos oceanos Atlântico e Pacífico; médias mensais relativas da TSM nos oceanos Pacífico e Atlântico; ventos na superfície do mar e em altitude; pressão atmosférica; e radiação de onda longa, parâmetro que fornece um indicativo sobre a cobertura de nuvens da região tropical.

Uma das dificuldades desta previsão é que o Nordeste exibe não somente uma alta variabilidade na quantidade anual de precipitação, mas também alta variabilidade temporal na precipitação dentro de sua estação chuvosa. Isto dificulta ainda mais o prognóstico das chuvas [32].

Levando em conta, as considerações acima, o trabalho propõe-se a apresentar possíveis soluções e comparar as soluções de seis métodos não paramétricos na previsão de classificação da chuva. Mais especificamente, prever a classe da quadra chuvosa na cidade de Boa Viagem,a qual se situa em uma das cinco regiões pluviometricamente homogêneas do Ceará [35], chamada Sertão Central e Inhamuns.

Os métodos não paramétricos combinaram diversas variáveis oceânicas e atmosféricas, para tentar descobrir padrões nos dados que definem a classe da chuva no Estado mencionado, objetivando encontrar novos relacionamentos e um nível de certeza maior. Em outras palavras, o objetivo é investigar o uso de métodos de Aprendizagem Automática em dados climatológicos para a previsão de chuva anual no Estado do Ceará através do processo de *"Data Mining"*. A previsão de chuva anual consiste em determinar previamente o comportamento da chuva na quadra chuvosa. Este comportamento pode ser classificado qualitativamente em uma das

seguintes classes: seco, normal e chuvoso. Assim, desejamos saber previamente em qual destas classes a chuva de um certo ano se encontra. Para isso, precisamos extrair um modelo de previsão de chuva a partir do conjunto de dados climatológicos. Um modelo consiste em um conjunto de regras que relacionam as variáveis climatológicas, como a Temperatura da Superfície do Mar (TSM) com a classificação de um determinado período.

O trabalho se propôs a verificar o comportamento de alguns métodos não-paramétricos para prever a quadra chuvosa. A região selecionada foi uma das cinco regiões homogêneas do Ceará, encontradas no artigo [36]. A região escolhida foi "Região Central e Inhamuns". As classes usadas foram classes nominais encontradas, usando-se três técnicas estatísticas: percentil, quantil e amplitude. Para as técnicas de percentil e amplitude foi utilizado um conjunto C de três classes, onde $C = \{seco, normal, chuvoso\}$ e dois conjuntos de duas classes $C_1 = \{seco + normal, chuvoso\}$ e $C_2 = \{seco, normal + chuvoso\}$. Para a técnica de quantil o conjunto C foi $C = \{\text{muito seco}, \text{seco}, \text{normal}, \text{chuvoso}, \text{muito chuvoso}\}$.

Os métodos utilizados serão descritas no capítulo que descreve o processo de *"Data Mining"* usado. O critério de escolha destas variáveis baseou-se nos trabalhos anteriores [35, 37, 32, 38], como determinantes para a precipitação e na disponibilidade das bases de dados. Os dados foram obtidos na FUNCEME (Fundação Cearense de Meteorologia) e no Observatório Real da Bélgica.

3.1 Definição do Problema

O principal objetivo é investigar o uso de métodos de Aprendizagem Automática em dados climatológicos, para a previsão de chuva anual em uma cidade no Estado do Ceará, através do processo de *"Data Mining"*. A previsão de chuva anual consiste em determinar previamente o comportamento da chuva em um determinado período do ano. Este comportamento é classificado qualitativamente em muito seco, seco, normal, chuvoso e muito chuvoso [36]. Assim, deseja-se saber previamente em qual destas classes a chuva de um certo ano se encontra. Para isso, precisa-se extrair um modelo de previsão de chuva a partir do conjunto de dados climatológicos. Um modelo consiste em um conjunto de regras que relacionam as variáveis climatológicas, como a Temperatura da Superfície do Mar, com as classes.

3.2 Trabalhos Anteriores

3.2.1 Métodos Paramétricos

O maior problema destes métodos é que a forma assumida é muito restritiva para diversos domínios. A maior vantagem dos métodos é simplicidade e a eficiência computacional.

Várias metodologias têm sido usadas com os objetivos operacionais no Instituto Nacional de Pesquisas Espaciais (INPE), na Fundação Cearense de Meteorologia e Recursos Hídricos (FUNCEME), dentre outras, para prever a estação chuvosa no Nordeste Brasileiro. Estas metodologias são baseadas em parâmetros atmosféricos e oceânicos, nos resultados de modelos de simulação, e em modelos estatísticos específicos [32].

O interesse em estabelecer um método para a previsão das chuvas no Ceará existe desde 1928. O primeiro trabalho relacionado foi feito por G. Walker em 1928, que previu o índice de chuva no Ceará e obteve um nível de acerto de aproximadamente 60%. Após isto, vários pesquisadores aplicaram técnicas estatísticas para prever chuva no Nordeste e encontrar variáveis preditoras.

As técnicas estatísticas mais utilizadas para previsão de chuva no Nordeste Brasileiro foram a análise de regressão e a análise de correlação. Estas técnicas realizam o inter-relacionamento entre duas ou mais variáveis contínuas. Além destas técnicas, também utilizou-se e utiliza-se a previsão de modelos acoplados. A análise de correlação realiza inferências estatísticas sobre três medidas de associação: o coeficiente de correlação simples, que mede a força de um relacionamento entre duas variáveis, o coeficiente de correlação múltipla, que mede a força de um relacionamento linear entre uma variável e um conjunto de variáveis, e o coeficiente de correlação parcial, que mede a associação linear entre duas variáveis, depois de remover o efeito linear de um conjunto de outras variáveis.

A análise de regressão estuda o relacionamento entre uma variável, denominada variável dependente, e diversas outras variáveis independentes. Este relacionamento é representado por uma equação, que associa uma variável dependente com variáveis independentes, considerando um conjunto de suposições relevantes a esta associação. Existe uma função que relaciona as variáveis independentes com a variável dependente, a qual envolve um conjunto de parâmetros desconhecidos. Quando esta função é linear nos parâmetros, tem-se um modelo de regressão linear.

Caso contrário, tem-se um modelo de regressão não-linear.

A técnica de análise de correlação foi utilizada por: Markham [39], que analisou uma função de autocorrelação no período de 1850 a 1970; Girardi e Teixeira [40], que usaram séries temporais da precipitação pluviométrica anual de Fortaleza e de outros seis postos de medição de pluviometria, para encontrar um coeficiente de correlação em torno de 0,74; Nobre [41] fazendo um teste de autocorrelação de Kolmogorov-Sminorv aplicado às séries temporais; Brito [42], que usou um modelo puramente estatístico baseado no modelo de Hastenrath [43]; e Alves e Repelli [38] para predizer a variabilidade espacial da chuva sobre o Nordeste Brasileiro. A técnica de análise de regressão foi utilizada por: G. Walker em 1928, que baseado na regressão múltipla do índice de chuva sazonal (janeiro a junho), de Fortaleza a Quixeramobim, desenvolveu uma fórmula estatística para a previsão de chuva no Estado do Ceará; e, Xavier [35] para a avaliar o papel da componente meridional do vento, na costa do Nordeste Brasileiro e de outras covariáveis, para prever a chuva no Ceará.

3.2.2 Métodos Não-Paramétricos

O método não paramétrico mais utilizado em previsões são as Redes Neurais. Redes Neurais ou Conexionismo objetivam investigar a possibilidade de simulação de comportamentos inteligentes através de modelos baseados na estrutura e funcionamento dos sistemas neurais biológicos.

Hall et al. [44] previram a probabilidade de precipitação (PP) e a precipitação quantitativa (PQ) em uma área do Texas (Dallas - Fort Worth), utilizando uma Rede Neural para cada previsão. A entrada consistiu em 19 variáveis meteorológicas e a precipitação observada em 36 medidores de chuva. Todas as variáveis tinham periodicidade diária. As redes foram treinadas utilizando dados de 1994 a 1995 e testadas com dados de 1 de março de 1996 a 30 de setembro de 1997 (579 dias).

Na rede PP; as classes foram: chuva e não chuva e o resultado era um percentual. Na rede PQ, não existia classificação. As redes previam a estação quente (abril a outubro) e a fria (novembro a março) e tinham a capacidade de mudar qualquer variável e re-executar o processo, como forma de antecipar qualquer mudança meteorológica. As previsões resultantes da rede PP tiveram uma correlação linear de 0,96, com a chuva observada e em todas as previsões maiores ou iguais a 38,5% houve precipitação.

Miyano e Girosi [45] previram as variações de temperatura global usando Redes Neurais. Este trabalho usou também os métodos paramétricos: técnicas de regressão linear e não-linear para a análise de séries temporais entre 1861 e 1984. Os dados utilizados foram as diferenças de temperatura do ar da superfície global entre anos sucessivos. As variações foram previstas usando quatro métodos: rede de regularização, *"perceptrons"* multi-camadas, auto-regressão linear e um modelo local conhecido como método de projeção simplex. O treinamento foi realizado com as séries de 1861-1909 e os testes: 1910-1944, 1910-1964 e 1910-1984. Para todos os modelos, os erros das previsões aumentam notavelmente após 1965. Estes resultados são consistentes com a hipótese de que houve uma mudança no clima neste período. O modelo da rede de regularização faz a melhor previsão, enquanto a auto-regressão linear faz a pior. Isto sugere, neste caso, que o comportamento dinâmico das séries temporais é não-linear.

Hastenrath e Greischar [34] desenvolveram um trabalho para predizer a precipitação do Nordeste Brasileiro de março a junho, usando informações referentes ao mês de janeiro. Os métodos usados foram dois métodos de previsão paramétricos (Regressão Múltipla Passo-a-Passo e Análise de Discriminante Linear) e um método não paramétrico (Redes Neurais). Serviram como variáveis preditoras precipitação acumulada regional do Nordeste, temperatura da superfície do mar do Atlântico e do Pacífico e a componente do vento meridional do Atlântico. Os dados de precipitação consistiam de medições em 27 postos espalhados pelo Nordeste. Para cada estação, os seguintes intervalos foram somados como preditores: outubro a dezembro (OND), outubro a janeiro (ONDJ), outubro a fevereiro (ONDJF) e outubro a março(ONDJFM). Como preditando, usou-se março a abril (MA), março a junho (MAMJ), abril a junho (AMJ) e março a setembro (MAMJJAS). Para cada ano, as variações normalizadas foram somadas sobre todas as 27 estações, para produzir séries temporais da variação normalizada média de todas as estações, para cada intervalo mencionado. O registro de previsão estende-se sobre 32 anos (1958-1989), enquanto que o conjunto de treinamento de 1921 a 1957 (excluindo 1943-1947) tendo sido utilizado para o treinamento dos métodos.

Capítulo 4

Métodos de Aprendizagem Abordados

O problema escolhido é uma instância do problema de previsão. Por isso, definições e conceitos são pertinentes ao problema citado e à literatura relacionada.

Como anteriormente dito, objetiva-se aprender procedimentos de classificação a partir de exemplos.

Um exemplo é uma tupla (x, c) onde x representa a descrição de uma situação, de um objeto ou de um indivíduo e c representa uma classe. No presente trabalho, considerou-se que os objetos são descritos por um conjunto de atributos $A = \{x_1, x_2.....x_n\}$, onde cada um destes x_i possui um valor dentro um certo domínio d_i, que pode ser associado a $\Re$,N, $\{0, 1\}$, $\{seco, normal, chuvoso\}$, entre outros. Para descrever uma carta a ser jogada, pode-se por exemplo, utilizar dois atributos a cor e o naipe, com os valores pertencendo respectivamente aos domínios: $D1 = \{\clubsuit, \diamondsuit, \heartsuit, \spadesuit\}$ e $D = \{\acute{A}, 2, ..., Reis\}$ [46].

A classe do exemplo poderá ser representada por um inteiro c e $\{1, 2..., c\}$, onde c determina o nome das classes. A classe exprime um diagnóstico médico a presença ou a ausência de uma determinada propriedade do objeto, entre outras. Quando somente há duas classes, freqüentemente o exemplo é positivo ou negativo, conforme pertença a uma classe ou a outra. O positivo será notado igual a 1 e negativo igual a 0 [46].

Um exemplo binário é um exemplo descrito unicamente por atributos binários e a classe e uma variável binária. Seja $A_n = \{x_1, x_2, ..., x_n\}$ um conjunto de atributos binários, $L_n = \{x_1, \overline{x}_1, x_2, ..., x_n, \overline{x}_n\}$ e um conjunto de literais resultantes de A_p ($\overline{x}_1$ representa a negação de x_1).

Do ponto de vista probabilístico, um exemplo (x, c) é uma realização de uma variável aleatória (X, C) com os valores em $D_1 \times D_2 \times ... \times D_p \times \{1, 2, ..., C\}$.

Uma amostra é um conjunto final de exemplos escolhidos de maneira independente, seguindo a distribuição $P(X, C)$.

Um procedimento de classificação (ou procedimento de decisão) é uma aplicação F definida sobre X e o valor de $\{1, 2, ..., C\}$. O objetivo de aprendizagem é descobrir um procedimento de classificação que direcione os exemplos. De uma maneira ou de outra, os métodos de aprendizagem, procedem sempre por exploração de um espaço de procedimentos possíveis pré-determinados. Este espaço será denominado F. No caso de exemplos binários, considera freqüentemente que F é um conjunto de fórmulas lógicas, de forma pré-determinada, descrito com a ajuda dos operadores booleanas usuais: $\vee(e), \wedge(ou), \neg(não)$. Neste caso, emprega-se freqüentemente o termo do conceito, para designar um procedimento de classificação.

Aprender é ao mesmo tempo um processo calculado e uma exigência da capacidade preditiva. O processo pode se resumir em: escolher um conjunto de exemplos, ou conjunto de aprendizagem; considerar um conjunto F de procedimentos de classificação; extrair F de um bom procedimento de classificação.

4.1 C4.5

C4.5 é um tipo de aprendizagem em árvores de decisão. Aprendizagem em árvores de decisão é um dos métodos mais utilizados e um dos mais práticos para inferência indutiva. Trata-se de um método para aproximar funções objetivo discreto-valoradas, no qual a função "aprendida" é representada por uma árvore de decisão. Um espaço de hipótese expressivo é pesquisado, evitando, portanto, a dificuldade de restrição de espaço de hipótese. É um método robusto para dados com ruídos e é capaz de aprender expressões disjuntas.

4.1.1 Descrição

Uma *Árvore de Decisão* [17] é constituída de uma hierarquia de nós, onde um nó pode ter um determinado número de filhos; em caso da árvore ser binária, o número máximo de filhos é dois. Os nós com filhos são chamados nós internos e cada um está associado a um teste em um atributo no conjunto de dados de entrada. Os nós sem filhos ou nós folhas definem as classes, ou seja, somente uma

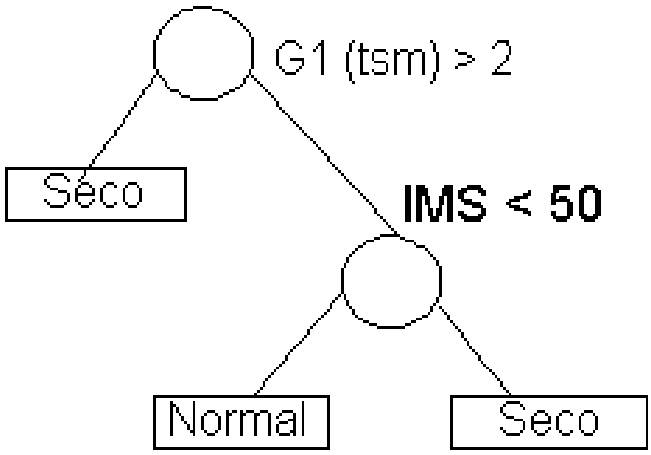

Figura 4.1: Exemplo de árvore binária.

classe é atribuída às instâncias contidas em cada nó folha. Cada teste feito por um nó interno divide o conjunto de dados em conjuntos disjuntos, um em cada ramo, ou seja, cada teste possui uma saída mutuamente exclusiva e exaustiva. Em árvores binárias de decisão, o conjunto é dividido em, no máximo, dois conjuntos disjuntos por nó interno. Este processo de teste e divisão continua até que um nó folha seja alcançado.

Resumindo, em uma árvore de decisão, um nó interno representa uma condição ou teste aplicado a um dos aspectos do problema a ser resolvido e um nó folha representa uma classe.

Uma árvore de decisão define um número determinado de classes para um conjunto de treinamento, filtrando padrões através dos testes na árvore. Conjunto de treinamento é o conjunto de dados usado para desenvolver os modelos de previsão. Neste conjunto as classes dos objetos devem ser previamente conhecidas.

Os testes podem ser multivariados, nos quais diversas características da entrada são testados simultaneamente; ou univariados, onde apenas uma das características é testada.

O processo de classificação por uma árvore de decisão envolve seguir o caminho partindo da raiz até uma folha, utilizando as decisões feitas em cada nó, para determinar que ramo ser seguido.

Existem vários algoritmos de aprendizagem usando árvores de decisão binárias, para classificação de novas instâncias. Alguns exemplos são: ID3 [17]; C4.5, uma nova versão do ID3; e CART [9].

Aprendizagem em árvores de decisão, mais especificamente, é um processo recursivo baseado na inclusão de nós na árvore, até que todos os dados de treinamento sejam divididos em classes. O algoritmo de aprendizagem seleciona uma característica para o ramo em cada nó e faz chamadas recursivas para construir subárvores para cada ramo criado. Para selecionar a condição do nó raiz, todas as instâncias de treinamento são utilizadas. Na seleção das condições dos nós subseqüentes utilizam-se os subconjuntos menores dos dados de treinamento, obtidos pela divisão imediatamente anterior. As árvores obtidas após o processo da aprendizagem podem ser também representadas como conjuntos de regras "se-então", para melhorar o entendimento humano.

Árvores de decisão classificam instâncias, ordenando-as da raiz da árvore até um nó folha, o qual determina a classificação da instância. Cada nó na árvore especifica um teste de um atributo da instância, e cada ramo descendente do nó corresponde a um dos possíveis valores para esse atributo.

Para classificar uma nova instância, inicia-se o processo observando-se o nó raiz e testando o atributo especificado para esse nó. Com isso, determina-se o ramo que será seguido pelo valor do atributo na nova instância. Repetem-se os passos anteriores para as subárvores subseqüentes, até que se alcance um nó folha. A classe da nova instância será a mesma do nó folha alcançado pela classe.

Em geral, árvores de decisão representam a disjunção de conjunções de restrições nos valores dos atributos de instâncias. Cada caminho da raiz da árvore para uma folha, corresponde a uma conjunção de testes de atributos, e árvore em si é a disjunção destas conjunções. [47]

A construção de uma árvore de decisão tem o intuito de dividir recursivamente os exemplos de um conjunto de aprendizagem, até obter subconjuntos de modelos que, preferencialmente, contenham exemplos de uma mesma classe, ou, até que todas as instâncias do conjunto treinamento sejam classificadas. Somente instâncias que alcançam um nó, são utilizadas para selecionar a condição do próprio nó.

O procedimento mais comumente utilizado na construção de árvores de decisão é o (*"top down"*), que segue a linha do mais geral para o mais específico.

O pseudocódigo para a construção de uma árvore de decisão encontra-se descrito no Algoritmo 1.

Existem algumas heurísticas para se escolher os melhores atributos para a construção de uma árvore de decisão. Elas objetivam obter árvores menores com

Algoritmo 1 Pseudo-código de construção de uma árvore de decisão.

```
1.  Pseudocódigo: Construção de uma árvore de decisão
2.  Entrada: L
3.  início
4.      Selecionar um atributo para nó raiz
5.      Criar um ramo para cada um dos valores possíveis do teste
6.      Dividir as instâncias dentro de cada ramo, formando subconjuntos disjuntos L_n
7.      Retornar ao passo 1, considerando como conjunto de entrada, o subconjunto de
8.      dados contidos no ramo, até que todas as instâncias tenham a mesma classe, ou
9.      um certo critério de parada seja alcançado
10. fim
```

mais precisão e nós mais puros. A pureza dos nós refere-se ao número de instâncias em um nó folha, de uma classe distinta da classe definida pelo nó.

O critério mais conhecido para medir o grau de impureza de um atributo é o ganho de informação. O ganho de informação aumenta de acordo com a pureza média de subconjuntos que um atributo produz.

Uma das heurísticas é escolher o atributo que resulte no melhor ganho de informação. Existe também outra medida, a entropia de distribuição, a qual é a informação requerida para prever um evento, dado uma distribuição de probabilidade.

O cálculo da entropia é dado por:

$$entropia(p_1, p_2, ..., p_n) = -p_1 log p_1 - p_2 log p_2 ... - p_n log p_n \tag{4.1}$$

No processo de formação da árvore de decisão, nem todos os nós folhas precisam ser puros; algumas vezes instâncias idênticas têm classes diferentes. Isto acontece porque, às vezes, existem instâncias com classes diferentes, que os atributos não conseguem separar em conjuntos disjuntos, a não ser que haja uma super especialização.

As principais propriedades requeridas da medida de pureza são: se o nó é puro, a medida é zero; quando a impureza é maximal (todas as classes são igualmente prováveis).

A entropia é a função que satisfaz todas as três propriedades citadas acima.

Existe uma problemática com atributos, os quais possuem um grande número de valores, um exemplo, o código de identificação de valores.

Os subconjuntos são mais prováveis para serem puros se há um grande número de valores. O ganho de informação é influenciado em direção a escolher atributos com uma larga variedade de valores. Porém, esta escolha pode resultar em uma super generalização, ou seja, a seleção de atributos que não são ótimos para previsão.

A razão de ganho é uma modificação de ganho de informação, que reduz sua parcialidade. A razão leva em conta o número e o tamanho de ramos quando escolhe um atributo, e corrige o ganho de informação por considerar a informação intrínseca de uma divisão.

A informação intrínseca é a entropia de distribuição de instâncias dentro de ramos. Um exemplo é a quantidade de informação necessária para saber se um ramo pertence a uma determinada instância.

A desvantagem da razão de ganho é a possibilidade de uma compensação muito alta, ou melhor, escolhe-se um atributo apenas porque sua informação intrínseca é muito baixa; ou somente considera-se atributos com um ganho de informação maior do que a média.

A tarefa de indução é desenvolver uma regra de classificação que determine a classe de um objeto, através de seus atributos. Essencialmente, objetiva-se construir uma árvore de decisão que classifique corretamente, não somente os elementos do conjunto de treinamento, mas, principalmente, objetos não vistos.

Métodos de Árvores de Decisão podem obter classificadores precisos através da comparação de pequenos conjuntos de exemplos de treinamento [9, 48]. Além disso, estes classificadores têm uma forma simples que pode ser armazenada e classificando eficientemente um novo dado.

Cada objeto no universo deste trabalho pertence a uma das classes a um determinado conjunto, cada classe deste conjunto é mutuamente exclusiva e representa a classificação da chuva, ocorrida no período chuvoso do ano seguinte. A classe de uma quadra pertence ao conjunto $C = \{seco, normal, chuvoso\}$ ou $C_1 = \{\text{muito seco}, \text{seco}, \text{normal}, \text{chuvoso}, \text{muito chuvoso}\}$, o qual foi obtido usando-se a técnica de quantis [36]. O tipo do conjunto de classes depende da técnica usada para obter as classes.

A base do processo de classificação em uma Árvore de Decisão é a indução. Para realizá-la, precisa-se de um universo de objetos que são descritos em termos de uma coleção de atributos. Cada atributo mede alguma característica importante

de um objeto. No presente trabalho, os objetos são períodos do ano, a tarefa de classificação envolve condições do tempo e os atributos serão descritos nos capítulo subseqüentes.

Um exemplo bem-sucedido do C4.5 foi a classificação de clientes encontrando boas regras [49].

4.2 CART

4.2.1 Descrição

O Método CART (Classification and Regression Trees), proposto por Breiman et al. [9], lida de forma elegante com dois problemas supracitados. Os procedimentos de classificação são representados por estruturas chamadas de árvore de decisão. CART é um algoritmo que, a grosso modo, é executado em três passos: construção de uma árvore de complexidade máxima, poda radical da árvore e estimativa da probabilidade de erro através de validação cruzada.

Uma das vantagens do uso de árvores de decisão para aprendizagem sobre métodos estatísticos é que: padrões induzidos podem ser baseados em um fenômeno local, enquanto métodos estatísticos controlam somente condições pertencentes à população inteira.

O uso de árvores binárias em probabilidade de regressão e classificação, como forma de ver os dados é bastante indicada, pois tem vantagem de ser uma ferramenta flexível não-paramétrica, para analisar dados.

Um exemplo de classificação é o da identificação de pacientes cardíacos de alto risco, baseado nos dados das 24h iniciais, usando-se 19 atributos. O problema da Classificação se constitui em: predizer a classe do caso analisado, baseado em medições, ou seja, encontrar uma maneira sistemática de predizer a classe [9].

No exemplo citado, temos as seguintes classes: classe 1, paciente não é de alto risco, e, classe 2, paciente é de alto risco.

O classificador ou regra de classificação é uma maneira sistemática de se predizer a classe em que o caso se encontra, para qualquer problema.

Para representar formalmente os casos, com suas variáveis e a classe em que se encontra, utiliza-se:

- Instância: um vetor x de medidas $(x_1, x_2, ..., x_n)$;

- χ é o espaço de medida multidimensional, onde $\forall x(x \in \chi)$, cada variável é uma dimensão e qualquer caso é um ponto no espaço χ;
- As classes estão contidas em $C = 1, ..., J$;
- Um classificador ou uma regra de classificação é uma função $d(x)$ definida em χ, onde, para todo x, $d(x)$ é uma das classes de $1, ..., J$;

Um classificador também pode ser notado, se for definido A_j como o subconjunto de χ no qual $d(x) = j$, ou seja, $A_j = x|d(x) = j$. Os conjuntos $A_1, ..., A_j$ são disjuntos e $\chi = \cup_j A_j$. Logo, A_j forma uma partição de χ.

Outra definição de classificador é uma partição de χ dentro dos J subconjuntos disjuntos $A_1, ..., A_j$, $\chi = \cup_j A_j$, tal que para todo $x(Aj$ a classe predita j é J [9].

Na construção do classificador é utilizada a aprendizagem por exemplo. Isto consiste de dados medidos observados no passado, junto com a respectiva classificação.

Outro exemplo é que pacientes cardíacos com baixa pressão geralmente são de alto risco.

A amostra de aprendizagem consiste de dados $(x_1, j_1), ..., (x_N, j_N)$ em n casos onde $x_N \in \chi$ e $,j_N \in 1, ..., J$, com $n = 1, ..., N$. A amostra de aprendizagem é representada por:

$$L = (x_1, j_1), ..., (x_N, j_N)$$

Existem tipos gerais de variáveis: numérica ou ordenada, se o valor é um número real, e, categórica: se o valor está contido em conjunto finito não tendo uma ordem natural.

Se todos os vetores de medida x_N tem a mesma dimensionalidade, então os dados têm uma estrutura padrão.

Com a análise da classificação objetiva-se entender quais as variáveis ou interações de variáveis dirigem o fenômeno, ou melhor, dar uma simples caracterização de condições que determinam quando um objeto está em uma classe ou em outra.

As características dos conjuntos de dados adequados: grande volume de dados, alta dimensionalidade, mistura de tipos de dados, e, estrutura de dados não-padronizadas.

O número de parâmetros em M dimensões binária $O(2M)$ e normal $O(M^2)$. Porém, se todas as variáveis forem independentes o número de parâmetros é O(M).

Dada uma função $d(x)$ definida em χ usando valores de C, denota-se $R^*(d)$ a taxa de classificação errada. Para estimar $R^*(d)$ testa-se o classificador em casos subseqüentes, os quais a classificação correta tem sido observada.

Pode-se definir $R^*(d)$ também como a probabilidade que d classificará erroneamente um novo exemplo, ou $R^*(d) = P(d(\chi) \neq Y)$.

As estimativas de R^* são referidas como estimativas internas. Os tipos de estimativas relevantes são [9]:

- Resubstituição: os casos que estão em L são utilizados para estimar R;
- Conjunto de teste: os casos são divididos em dois conjuntos L_1 e L_2 , L_1 para construir d e L_2 para estimar R;
- Validação cruzada ou *V-fold cross validation*: para V classificadores é construído usando um tamanho de amostra de aprendizagem de $N(1 - \frac{1}{V})$.

Supondo que (X, Y), $X \in \chi$, $Y \in C$, é um exemplo randômico de probabilidade de distribuição $P(A, J)$ em $\chi \times C$. Temos que $dB(X)$ é uma regra de *"Bayes"* se para qualquer outro classificador $d(X)$,

$$P(dB(X) \neq Y) \leq P(d(X) \neq Y)$$

A regra de *"Bayes"* é denominada de regra de probabilidade máxima.

Os procedimentos de classificação mais utilizados são análise de discriminante; estimativa de densidade de *"Kernel"*, e, o k-ésimo vizinho mais próximo.

Os classificadores estruturados em árvores binárias, são construídos por repetidas divisões de subconjuntos de χ, dentro de dois subconjuntos descendentes, iniciando com o próprio χ.

Os subconjuntos terminais (folhas) formam uma partição de χ.

Cada subconjunto terminal é designado por uma classe. Pode haver mais de um subconjunto terminal denominado pela mesma classe.

As divisões são formadas por condições nas condições de $x = (x_1, x_2, ...)$. Por exemplo, a divisão 1 de χ dentro de χ^2 e χ^3 poderia ser da forma: $\chi^2 = x; x_4 \leq 7$, $\chi^3 = x; x_4 < 7$.

Um nó t é um subconjunto de χ e o nó raiz é sempre χ.

Os elementos da construção da árvore são: a seleção de divisões (splits), as decisões de quando declarar um nó terminal ou continuar dividindo-o, e, a atribuição de cada nó terminal para uma classe.

Como utilizar o conjunto L de casos para determinar as divisões de χ dentro de pedaços cada vez menores? A principal idéia é selecionar cada divisão do subconjunto, até que os dados de cada um subconjunto descendente sejam mais puros do que os dados no subconjunto pai.

No exemplo de aprendizagem L para a probabilidade da classe J, deixa N_j ser o número de casos na classe J. Freqüentemente as probabilidades anteriores $\pi(j)$ são levadas a ser as proporções $\frac{N_j}{N}$.

Os quatros elementos do procedimento de crescimento da árvore inicial são [9]:

1. O conjunto *sigma* de questões binárias da forma $x \in A?$, $A \subset X$;
2. A vantagem do critério de divisão $\phi(s,t)$ que podem ser avaliados por qualquer divisão s de qualquer nó t;
3. A regra de parada de divisão;
4. A regra para a atribuição de um nó final para a classe.

O conjunto padronizado de questões σ é definido como segue:

1. Cada divisão depende do valor de somente uma variável;
2. Para cada variável ordenada X_m, σ inclui todas as questões da fórmula: $\{\acute{E}\ X_m \leq C?\}$ para todo C com domínio $(-\infty, \infty)$;
3. Se X_m é categórica tomando os valores em $b_1, b_2, ..., b_L$, então ∞ inclui todas as questões da forma: $\acute{E} X_m \in S?$ Como S limita todos os subconjuntos de $b_1, b_2, ..., b_L$.

A vantagem do critério de divisão foi originalmente derivada da função de impureza. A regra de atribuição de classe e estimativa de resubstituição. Suponha que uma árvore t, tenha sido construída e tenha nós terminais β.

A regra de atribuição de classe atribui a classe $j \in 1, ..., J$ para todo nó terminal $t \in \beta$. A classe atribuída para o nó $t \in \beta$ é denotada por j(t).

A dificuldade mais significante foi que as árvores freqüentemente davam resultados irreais. As estimativas da resubstituição R(t) eram falsamente baixas e as árvores maiores que a informação, na garantia de dados.

A solução satisfatória veio somente depois da mudança fundamental do foco. Ao invés de tentar parar a divisão no conjunto certo de nós terminais, continuar a divisão até que todos os nós terminais estejam muito pequenos, resultando em uma árvore maior. Seletivamente a poda (recombinação) desta árvore maior para cima, dando uma diminuição da seqüência de subárvores. Então usa-se a validação cruzada ou uma estimativa de uma amostra de teste

Uma divisão é selecionada, se reduz o nível de impureza em relação a outras divisões.

Uma regra usa o índice de Gini de diversidade como a medida de impureza de nó, isto é, a regra twoing: no nó t dividindo t dentro de t_L e t_R, escolha a divisão s que maximiza

$$(P_L \frac{P_L}{4})[\sum_j p(j|t_L) - p(j|t_R)|]2$$

.

Outra deficiência nas árvores usando a estrutura padrão é que todas as divisões são em variáveis simples. O programa de árvore padrão poderia dividir muitas vezes na tentativa de aproximar o hiperplano, separado por retângulos.

Existem dois aspectos para problemas de valores perdidos: primeiro, alguns casos em L podem ter alguns valores de medição perdidos, segundo, pode-se querer a árvore completa para predizer o nome da classe para o vetor de medidas, dos quais os valores são perdidos.

Uma das vantagens de aproximação de árvore estruturada é ser um procedimento recursivo e interativo, que requer as especificações de somente poucos elementos, os quais são:

- Conjunto σ de questões;
- A regra de seleção da melhor divisão em qualquer nó;
- O critério para escolha de árvore de tamanho certo;

O uso da resubstituição resultou em árvores muito grandes. A resubstituição é representada por R(T) e no caso de nó terminal com um caso $R(T) = 0$.

Se o tempo computacional fosse ilimitado, então, a divisão continuaria até todo nó terminal conter somente um nó. Como não é possível, a solução é usar um método para crescimento de T_{max}: estabelecer N_{min} e cada nó terminal. O nó

terminal pode ser um nó puro ou satisfazer $N(t) \leq N_{min}$ ou conter vetores idênticos [9].

Se o nó t' é descendente do nó mais alto t então t é antecessor de t'.

Considere o ramo T_t, sendo o nó $t \in T$, o qual é a raiz da árvore T e todos os outros nós são descendentes do nó t. Para podar ramo T_t : exclui-se de T e todos descendentes.

A medida do custo-complexidade se dá por: $R\alpha(t)$ como $R\alpha(t) = R(t) + \alpha|\beta|\beta$, número de nós terminais. Sendo $\alpha \geq 0$, parâmetro de complexidade.

A melhor subárvore a ser podada é a que apresente um menor $R'(T_k)$.

Quando o conjunto de exemplos não é muito grande, a validação cruzada é a melhor estimativa para analisar a aprendizagem do algoritmo.

Um pseudocódigo para a validação cruzada ou *"V-fold cross validation"* está representado no Algoritmo 2.

Algoritmo 2 Pseudo-código do processo de Validação Cruzada.

```
1.  Pseudocódigo: Validação Cruzada
2.  início
3.      Dividir o conjunto de exemplos em subconjuntos v, com o mesmo número de casos
4.      Treinar o método com um certo conjunto v_i e testar com v_j, sendo j ≠ i
5.  fim
```

A divisão em novos ramos naturalmente reduz mais a estimativa de erro do que resubstituição.

O critério de divisão é deficiente, pois pode reduzir a taxa de classificação errada, com um erro de super generalização (*"overfitting"*).

Se existem apenas duas classes para a previsão, o problema da divisão é solucionada utilizando-se a função impureza do nó, ou seja, a taxa de classificação errada do nó.

No caso de mais de duas classes para a previsão, utiliza-se o critério de *Gini* e o de *Twoing* para escolher a divisão.

O índice de *Gini* representa o número de categorias que podem ser guardadas.

No critério de *Twoing*, as seguintes características são requeridas: um pequeno número de classes, uma superclasse fixada e que as variáveis sejam categóricas.

A melhor superclasse é escolhida.

Outra dificuldade que existe em um problema do mundo real é a existência de diferentes estruturas de dados. Algumas das soluções usadas no *CART* são dividir as variáveis em combinações de variáveis usando: combinação linear, combinações booleanas e combinações *"ad hoc"* por exame dos dados.

Para manipular dados perdidos são verificadas partições substitutas e suas medições. Para isto é feito um *"ranking"* da importância de variáveis e uma detecção de mascaramento. O propósitos do algoritmo com os dados perdidos é fazer o uso máximo dos casos de dados e construir uma árvore que classifique qualquer caso, mesmo havendo perda.

O *"ranking"* de variáveis é feito da seguinte forma: se a melhor divisão do nó t, no caso não variável, está em X_{m_1}, e se X_{m_2} tem sido mascarado em t, isto é, se X_{m_2} pode gerar uma divisão similar $S \times m_1$, mas, não completamente bom, então em t , será muito grande.

Nas árvores de regressão ou *"least squares regression"* existe a construção de um preditor $d(x)$. Este preditor tem a função de predizer a variável responsável pela classificação e ajudar no entendimento do relacionamento das variáveis.

Os requisitos para um preditor são: a maneira de selecionar divisões, a regra para nó terminal e a regra para a saída $y(t)$ para nó terminal.

Nas árvores de regressão existem dificuldades na poda, devido ao tamanho da árvore e das estimativas muito otimistas. A poda nas árvores de regressão tem como característica podar somente dois nós terminais por vez.

Na regras de *"Bayes"* $R(d)$ é o risco da perda esperada usando $d(x)$ e o seu objetivo é minimizar $R(d)$, o que diminui o risco. A função partição na regra de *"Bayes"*: associa cada caso a um nó terminal.

A regra de divisão para redução de Risco calcula redução de risco de pares t e t' e divide, se houver redução de risco.

Definição: árvore consiste de um conjunto finito não vazio T de inteiros positivos e duas funções esquerda(.) e direita(.) de t para $T0$.

Uma árvore possui um conjunto finito de subárvores e para ser "ótima" minimiza $R(T)$ [9].

O pseudocódigo de algoritmo de poda ótimo segue no Algoritmo 3.

O pseudocódigo para construção de uma árvore de aprendizagem está representado no Algoritmo 4.

Algoritmo 3 Pseudo-código do Algoritmo de Poda.

```
1.  Pseudocódigo: Algoritmo de Poda
2.  início
3.      Para todos os casos, calcular comparadores
4.      Repita até N(1) =1
5.  fim
```

Algoritmo 4 Pseudo-código do construção de uma árvore de aprendizagem.

```
1.  Pseudocódigo: Construção de uma árvore de aprendizagem
2.  início
3.      Particionar o conjunto de dados em V subconjuntos
4.      Particionar um subconjunto v_i de treinamento com a regra de "Bayes" estimada
5.      Escolher a poda ótima
6.      Testar a árvore com o conjunto v_j de exemplos de teste
7.      Fazer validação cruzada repetindo os passos anteriores para o restante
8.      dos subconjuntos
9.      Selecionar a melhor árvore construída
10. fim
```

As principais vantagens da classificação através de árvores de decisão são: eficiência em tempo de processamento e ser um método intuitivo de analisar os resultados.

As principais desvantagens estão na dificuldade de manipulação de dados perdidos e na super generalização (*"overfitting"*).

O método CART é muito empregado, em estudos comparativos. CART aparece como um dos melhores métodos de construção de árvores de decisão, como por exemplo em ([9]; J. Mingers, 1987; J. Mingers, 1989). Muitos autores propuseram modificações no método CART no processo de seleção da divisão de um nó (Loh e Vanichsetakul, 1988; Chou, 1991), e no processo de validação cruzada (Crawford, 1990). Entretanto, ainda é difícil superar os resultados do CART [46].

A técnica foi usada com sucesso para predizer a qualidade do vinho, de acordo com características da terra [50]. Outra aplicação bem sucedida é a previsão de entonação de frases de um texto para reconhecimento e síntese de voz. A taxa de sucesso foi de mais de 90%, representando o melhor resultado sobre as outras tentativas de previsão de texto sem restrições [51].

4.3 Redes Neurais

Redes Neurais ou Conexionismo ou Sistemas de Processamento Paralelo e Distribuído objetivam investigar a possibilidade de simulação de comportamentos inteligentes, através de modelos baseados no funcionamento e nas estruturas neurais biológicas. Em 1943, foi desenvolvido o primeiro trabalho na área com Warren McCulloch e Walter Pitts, que concentrou-se principalmente em descrever um modelo artificial de neurônio e de apresentar suas capacidades computacionais. Depois deste trabalho, Donald Hebb em 1949, mostrou como a plasticidade da aprendizagem é obtida, através da variação dos pesos de entrada dos nós e propôs uma teoria que explica o aprendizado em nós biológicos, o qual se baseia no reforço das ligações sinápticas entre nós excitados. A Regra de Hebb, denominação do trabalho de Donald Hebb, vem sendo utilizada por vários algoritmos de aprendizado. Depois disso, Widrow e Hoff sugeriram uma regra de aprendizagem, a regra Delta ou regra Widrow-Hoff, baseada no método do gradiente para minimização do erro na saída de um neurônio, com resposta linear [52].

Frank Rosemblatt demonstrou com um modelo novo chamado *"perceptron"*, que, se fossem acrescidas de sinapses ajustáveis, as Redes Neurais poderiam ser

treinadas para classificar certos tipos de padrões. O autor descreveu uma topologia de rede, estruturas de ligações entre os nós e um algoritmo para treinar a rede. O *"perceptron"* mais simples de Rosemblatt é composto de três camadas: a primeira recebe as entradas do exterior e possui conexões fixas; a segunda recebe impulsos da primeira camada através de conexões, cujo peso é ajustável e, por sua vez, envia saídas para a terceira camada. Inicialmente, no modelo de Rosemblatt, a rede é aleatória, mas, pelo ajuste gradual dos pesos o *"perceptron"* é treinado para fornecer saídas de acordo com os dados de treinamento [52].

Em 1969, Minsky e Papert descobriram algumas desvantagens do *"perceptron"*, por exemplo, que não consegue detectar paridade, simetria e conectividade, em problemas que são linearmente separáveis [52].

Em 1986, Rumehart et al mostraram que a visão de Minsky e Papert sobre o *"perceptron"* era pessimista, pois, Redes Neurais com múltiplas camadas são, sem dúvida, capazes de resolver problemas que são difíceis de aprender. Porém, só a partir de meados da década de 80 é que houve uma explosão de interesse por Redes Neurais pela comunidade científica internacional [52].

Um dos atrativos da solução de problemas através de Redes Neurais é a possibilidade de um desempenho superior ao dos modelos convencionais, pela forma de representação interna da solução dos problemas e pelo paralelismo natural inerente à sua arquitetura [53].

Nesta técnica, o procedimento usual na solução de problemas passa, inicialmente, por uma fase de aprendizagem. Um conjunto de exemplos é apresentado pela rede, a qual extrai as características necessárias para representar a informação fornecida.

4.3.1 Descrição

Uma rede neural é um grafo dirigido consistindo de nós com interconexão sináptica e ligações (*"links"*) de ativação, o qual é caracterizado por quatro propriedades [54]:

1. Cada neurônio é representado por um conjunto de ligações sinápticas linear, um limiar aplicado externamente, e uma ligação de ativação não-linear.O limiar é representado por uma ligação sináptica com um sinal de entrada fixado no valor de -1.

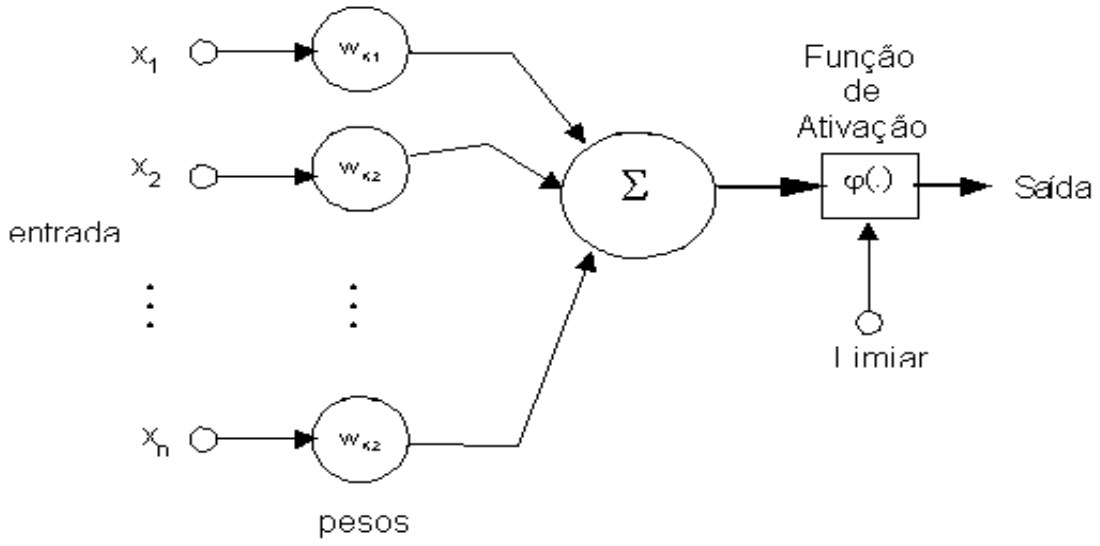

Figura 4.2: Exemplo de um neurônio de uma rede neural.

2. As ligações sinápticas de um neurônio modificam seus respectivos sinais de entrada.

3. A soma dos pesos dos sinais de entrada define o nível de atividade interna do neurônio em questão.

4. A ligação de ativação divide o nível de atividade interna do neurônio, para produzir uma saída, que representa a variável de estado do neurônio.

Uma rede neural pode possuir um sistema dinâmico, no qual a saída de um elemento no sistema, influencia em parte a entrada aplicada para este elemento particular. Isto denomina-se retroalimentação ou *"feedback"*. Um exemplo deste tipo de sistema, podemos ver na figura 4.2 [54]:

Em uma rede neural é dito existir um sistema dinâmico, se uma saída de um elemento no sistema, influencia em parte a entrada aplicada para este elemento particular. Isto denomina-se retroalimentação ou *"feedback"*. Um exemplo deste tipo de sistema, podemos ver na figura 4.2 :

Redes Neurais [53] são sistemas paralelos distribuídos compostos por unidades, denominadas neurônios artificiais, que computam determinadas funções matemáticas, geralmente não lineares.

Os neurônios artificiais estão dispostos em uma ou mais camadas e interligados por um grande número de conexões, geralmente unidirecionais. Na maioria

dos modelos, estas conexões estão associadas a pesos. Elas armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede. Esta forma de computação não algorítmica é caracterizada por sistemas que, em um certo nível de abstração, relembram a estrutura do cérebro humano.

A estrutura do neurônio artificial proposto por McCulloch e Pitts - modelo MCP - é baseada no neurônio biológico. Um neurônio biológico, de maneira resumida, é formado por um corpo celular que contém o núcleo da célula, vários dendritos (através dos quais impulsos elétricos são recebidos) e um axônio (através do qual impulsos elétricos são enviados). A propagação de um impulso elétrico ao longo de um Dendrite ou de um axônio, se dá através da alteração da concentração dos íons K^- e Na^+ em ambos os lados da membrana. As interligações entre neurônios são efetuadas através de sinapses, pontos de contatos entre dendrites e axônios, controlados por impulsos elétricos e reações químicas. O efeito das sinapses é variável, e é esta variável que dá ao neurônio a capacidade de adaptação. Um neurônio biológico dispara, quando a soma de impulsos que recebe ultrapassa o seu limiar de excitação [52].

A ativação do neurônio artificial de McCulloch e Pitts ou modelo MCP é obtida aplicando-se uma função de ativação, que ativa a saída ou não, dependendo do valor da soma ponderada das entradas. Na descrição original do modelo MCP, a função de ativação é dada pela seguinte função de limiar:

$$\sum_{i=1}^{n} x_i w_i \geq \theta$$

sendo n o número de entradas do neurônio, w_i o peso associado à entrada x_i, e θ o limiar do neurônio. As principais limitações deste modelo são: sua natureza binária, ou seja, em redes com camada única, somente implementam funções linearmente separáveis; os pesos são fixos; e existem pesos negativos, os quais são mais adequados a representar disparos inibidores [54].

A partir do modelo citado, foram derivados vários outros modelos. O modelo geral mais utilizado generaliza o MCP nos seguintes aspectos [54]:

- O nível de ativação é definido como uma função qualquer g das atividades dos neurônios da rede: $\sigma_i = g(x_1, .., x_n)$.
- A função de ativação $\mathcal{F}$, que determina a atividade de um neurônio pode ser

uma função limitada qualquer. Sendo interessante que $\mathcal{F}$ não seja linear, para evitar as restrições do modelo binário MCP.

- Foi introduzido um valor de polarização $\theta \in \Re$, de modo que a atividade de um neurônio é calculada por $x_i = \mathcal{F}(\sigma_i + \theta)$.

Em geral, as redes neurais artificiais são organizadas em camadas. Estas camadas são denominadas: camada de entrada, camada de saída e camadas internas.

Do ponto de vista funcional, uma rede pode ser homogênea, se todos os neurônios se comportarem da mesma forma, ou heterogênea, caso contrário.

A maneira na qual os neurônios de uma rede neural são estruturados é intimamente ligado com o algoritmo de aprendizagem, usado para treinar a rede. Podendo-se portanto, falar que algoritmos (regras) de aprendizagem usados na construção de uma Rede Neural, são estruturados.

Basicamente, pode-se identificar quatro classes de arquitetura para redes neurais, são elas [54]:

1. Redes com retroalimentação de camada única ou *"Single-Layer feedforward networks"*: nesta rede tem-se uma camada de nós de entrada que se projeta dentro de uma camada de saída de neurônios, mas, não vice-versa;

2. Redes com retroalimentação de multicamada ou *"MultiLayer feedforward networks"*: esta classe diferencia-se da anterior pela presença de uma ou mais camadas escondidas, nas quais os nós presentes são denominados neurônios escondidos ou unidades escondidas, cuja função é intervir entre a entrada externa e a saída da rede. Com isto, a rede adquire uma perspectiva global em relação à conectividade local, devido ao conjunto extra de conexões sinápticas e a dimensão extra de interações neurais;

3. Redes recorrentes: a característica principal desta classe é possuir pelo menos um ciclo de retroalimentação ou *"feedback loop"*. Em outras palavras, uma rede recorrente pode consistir de uma simples camada neuronal, com cada neurônio alimentando seu sinal de saída para as entradas de todos os outros neurônios.

4. Estruturas *"Lattice"*: uma *"lattice"* constitui-se de um vetor de uma ou mais dimensões de neurônios, com um conjunto correspondente de nós de origem

que fornecem sinais de entrada para o vetor; a dimensão de uma *"lattice"* refere-se ao número de dimensões de espaço no qual o grafo possui.

A definição da arquitetura de uma Rede Neural é um parâmetro muito importante na sua concepção, por restringir o tipo de problema que pode ser tratado pela rede. Por exemplo, redes com uma camada única de nós MCP, só conseguem resolver problemas linearmente separáveis. A arquitetura é definida, de um modo geral, pelos seguintes parâmetros: o número de nós em cada camada, o número de camadas, o tipo de conexão entre os nós e a topologia da rede.

Quanto ao número de camadas, pode-se ter: redes com uma única camada, nas quais, existe somente um nó entre qualquer entrada e qualquer saída da rede; ou, redes com múltiplas camadas, nas quais existem mais de um neurônio entre uma entrada e uma saída.

Quanto ao tipo de conexões de uma rede, tem-se os tipos: *"feedforward"* ou acíclica ou alimentação para a frente, na qual a saída de um neurônio na i-ésima camada da rede, não pode ser usada como entrada de nós em camadas de índice menor ou igual a i, e, *"feedback"* ou cíclica, na qual a saída de algum neurônio é utilizada como entrada de nodos, em camadas de índice menor ou igual a i.

Quanto à conectividade, uma rede pode ser: fracamente (parcialmente) ou completamente conectada.

Na fase de aprendizagem, na qual a rede extrai informações relevantes de padrões de informação apresentados pela mesma, criando desta forma uma representação própria para o problema. Esta etapa consiste em um processo iterativo de ajuste de parâmetros da rede, como por exemplo, o ajuste dos pesos das conexões entre as unidades de processamento que, ao final do processo, guardam o conhecimento que a rede adquiriu do ambiente em que está operando.

Aprendizagem em Redes Neurais [53], de uma forma mais geral, é o processo pelo qual os parâmetros de uma rede são ajustados, através de uma forma continuada de estímulo pelo ambiente no qual a rede está operando, sendo o tipo específico de aprendizagem realizada definido, pela maneira particular como ocorrem os ajustes realizados nos parâmetros.

Existem diversos métodos desenvolvidos para o treinamento de Redes Neurais, pode-se agrupá-los em dois paradigmas principais: Aprendizado Supervisionado e Aprendizado não Supervisionado [52].

O *Supervisionado* é o tipo de treinamento mais comumente usado para neurônios com peso ou sem peso, e é denominado *Aprendizado Supervisionado* porque a entrada e a saída desejadas pela rede, são fornecidas por um supervisor externo, ou seja, existe a disponibilidade de um conjunto de treinamento composto por pares de vetores de entrada e de saída, chamados pares de treinamento. O objetivo é ajustar os parâmetros da rede, de forma a encontrar uma ligação entre os pares de entrada e os pares de saída. No treinamento, compara-se a resposta desejada com a resposta calculada a cada padrão de entrada submetido à rede, e ajusta-se os pesos das conexões para minimizar o erro entre as respostas. A soma dos erros quadráticos de todas as saídas é normalmente utilizada como medida de desempenho da rede e também como função de custo a ser minimizada pelo algoritmo de treinamento. A desvantagem deste método é que na ausência do professor, novas estratégias não poderão ser aprendidas, para situações não cobertas pelos exemplos de treinamento. Os exemplos de algoritmo supervisionado mais comuns são regra delta e o algoritmo *"backpropagation"* [53].

No *Aprendizado Não-supervisionado*, não existe um supervisor no acompanhamento do processo de aprendizagem, como o próprio nome sugere. Nestes algoritmos, somente padrões de entrada são apresentados para a rede, então, a rede estabelece uma harmonia com as regularidades estatísticas da entrada de dados, desenvolvendo-se na rede uma habilidade de formar representações internas, para codificar características da entrada e criar novas classes ou grupos, automaticamente. Este aprendizado requer redundância de dados para se tornar possível, ou seja, sem redundância não é possível que quaisquer características ou padrões sejam encontrados. Outra característica deste aprendizado, é que a estrutura do sistema pode ter uma variedade de formas diferentes, como, por exemplo, pode consistir de uma camada de saída, conexões *"feed-forward"*, com múltiplas camadas da entrada para a saída e conexões laterais entre os neurônios da camada de saída [53].

Redes neurais têm sido aplicadas com sucesso em diversas áreas. Entre as aplicações bem-sucedidas pode-se citar: aprendizagem de pronúncia de textos em inglês (Sejnowski e Rosemberg, 1987), reconhecimento de voz (Cohen et al. 1993; Renals et al 1992), reconhecimento de caracteres óticos (Guyon, 1990), identificação de sistemas (Narendra e Parthasarathy, 1990), classificação e detecção de alvos em radares (Haykin e Bhattacharya, 1992; Haykin e Deng, 1991; Orlando et al., 1990); diagnóstico médico de ataques do coração (Baxt, 1993; Harrison et al., 1991) e

modelagem do controle do movimento dos olhos (Robinson, 1992) [54].

Este método é empregado também com sucesso na previsão da alta freqüência da taxa de câmbio de dólares americanos e dólares canadenses, no Banco do Canadá. Neste caso, redes neurais executam melhor do que outros modelos em termos de percentagem de previsões corretas de mudanças da taxa de câmbio [55].

Outros exemplos bem-sucedidos são aplicações em processamento de sinal [56].

4.4 *"Naive Bayes"*

Em Aprendizagem Automática, existe o interesse de se determinar a melhor hipótese de algum espaço de descrição H, considerando o conjunto de treinamento D. Uma maneira de se encontrar a melhor hipótese é escolher a hipótese mais provável dentro de um conjunto de hipóteses. Tendo-se para isto, dados iniciais D e conhecimento prévio sobre probabilidade anterior de várias hipóteses em H. O teorema de *"Bayes"* provém de um método direto para calcular tais probabilidades [57].

Usando-se o teorema de *"Bayes"*, calcula-se a probabilidade de uma hipótese baseada na sua probabilidade anterior. Denomina-se *P(h)* a probabilidade anterior de h. Se não houver conhecimento anterior, determina-se a mesma probabilidade anterior para cada hipótese candidata. Similarmente, escreve-se $P(D)$ para denominar a probabilidade anterior dos dados D, que serão observados. Para representar a probabilidade de x dado y utiliza-se $P(x|y)$. $P(h|D)$ é chamado de probabilidade posterior de h, a probabilidade reflete a influência dos dados de treinamento D, em contraste com a probabilidade anterior P(h), que é independente de D.

O teorema de *"Bayes"* calcula a probabilidade posterior $P(h|D)$, da seguinte forma:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{4.2}$$

Em muitos cenários, o sistema de aprendizagem considera algumas hipóteses candidatas H e tenta encontrar a hipótese mais provável $h \in H$ considerando o dado observado D (ou no mínimo a hipótese que apresenta a maior probabilidade se existir várias). Qualquer uma das hipóteses maximalmente prováveis é chamada hipótese máxima a posteriori (MAP). As hipóteses MAPs podem ser determinadas

através do teorema de *"Bayes"*, para calcular a probabilidade posterior de cada hipótese candidata [47].

O teorema de *"Bayes"* fornece uma maneira para calcular a probabilidade posterior de cada hipótese dos dados de treinamento. Por isso, o teorema pode ser a base para um algoritmo de aprendizagem, que calcula a probabilidade para cada hipótese possível, tendo como resultado a hipótese mais provável [47].

Assume-se que o algoritmo considera um espaço finito de hipóteses definida sobre o espaço de instância. Sua tarefa é aprender conceitos c, com $c : X \rightarrow 0, 1$. Assume-se também que é dado uma seqüência de exemplos de treinamento $x_1, d_1, ..., x_n, d_n$, onde x_i é alguma instância de X e onde d_i é o valor objetivo de x_i. A seqüência de instâncias $x_1, ..., x_m$ é considerada fixa para simplificar, então os dados de treinamento D podem ser como uma seqüência $D = d_1, ..., d_m$.

O pseudocódigo de aprendizagem MAP usando força bruta segue no Algoritmo 5.

Algoritmo 5 Pseudo-código de Bayes.

```
1.  Pseudocódigo: BAYES
2.  início
3.      Para cada hipótese h em H, calcular a probabilidade posterior
4.      Dentro do conjunto de probabilidades calculadas, escolher a hipótese
5.      h_map com a probabilidade posterior mais alta.
6.      h_MAP =_{h∈H} argmaxP(h|D)
7.  fim
```

Dependendo do número de instâncias, este algoritmo pode requerer significante computação, porque aplica o teorema de *"Bayes"* para cada hipótese em H para calcular $P(h|D)$. Portanto, em grandes espaços de hipóteses, sua performance pode ser comprometida.

4.4.1 Descrição

Como já foi dito, a aprendizagem de classificação calcula a probabilidade da classe, dada uma instância. O valor da classe dada uma instância é denominada evento H e uma instância também é denominada evidência E.

Em *"Naive Bayes"* assume-se que a evidência tem a capacidade de ser dividida dentro de partes independentes, por exemplo, atributos de instância.

$$Pr[H|E] = \frac{Pr[E_1|H] \times Pr[E_2|H] \times ... \times Pr[E_n|H] \times Pr[H]}{Pr[E]} \tag{4.3}$$

Neste método, existe a possibilidade de um determinado valor de um atributo, não acontecer em nenhum caso de um valor de uma classe. Neste caso, a probabilidade anterior e a probabilidade posterior serão zero, porém, se o cálculo destas probabilidades é feito através de contadores, é adicionado 1 ao contador para toda combinação classe-valor.

O resultado da solução acima é que as probabilidades nunca serão 0. Além disso, ocorre uma estabilização das estimativas de probabilidade.

Em alguns casos, adicionar uma constante diferente de 1 poderia ser mais apropriado. Os pesos não necessitam ser iguais. A instância não é incluída no contador de freqüência para combinação de classe-valor.

Assume-se nesta técnica que os atributos têm uma distribuição de probabilidade gaussiana ou normal, que todos têm a mesma importância e todos os atributos são independentes. Com isto, esta técnica permite que todos os atributos contribuam de um modo uniforme.

A função de densidade de probabilidade para a distribuição normal é definida por dois parâmetros.

A média de exemplo ϕ é dada por:

$$\phi = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{4.4}$$

O desvio padrão σ é calculado por:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \phi)^2}. \tag{4.5}$$

Já a função de densidade é definida por [58]:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\phi)^2}{2\sigma^2}}. \tag{4.6}$$

Em *"Naive Bayes"*, os valores perdidos durante o treinamento não são incluídos no cálculo da média e do desvio padrão.

Calcula-se o relacionamento entre probabilidade e densidade, da seguinte forma:

$$Pr\left[\ -\tfrac{\epsilon}{2}\ \right] < x < Prc + \frac{\epsilon}{2} = \epsilon * f(c) \tag{4.7}$$

Isso não muda o cálculo de probabilidades posteriores porque ϵ cancela.

Em um problema de classificação de uma nova instância $(a_1, a_2, ..., a_n)$ dentro de um número finito de categorias de um conjunto V, a abordagem bayesiana determina a categoria V_{MAP} mais provável dada a instância $(a_1, a_2, ..., a_n)$.

$$\begin{aligned} V_{MAP} &= arg_{v \in V} max P[v|a_1, a_2, ..., a_n] \\ V_{MAP} &= arg_{v \in V} max \frac{P[a_1, a_2, ..., a_n|v].P[v]}{P[a_1, a_2, ..., a_n]} \\ V_{MAP} &= arg_{v \in V} max P[a_1, a_2, ..., a_n|v].P[v] \end{aligned}$$

Uma vantagem da técnica de *"Naive Bayes"* é que trabalha surpreendentemente bem, mesmo se a suposição de independência é claramente violada. Isto se explica por não requerer estimativas de probabilidade precisas, tanto quanto a probabilidade máxima é determinada para a classe correta.

É valido considerar que adicionar vários atributos redundantes causarão problemas e que muitos atributos numéricos não são normalmente distribuídos.

O método *"Naive Bayes"* usa o método padrão "dividir para conquistar", para tratar toda possível combinação de valores de um atributo como classes separadas.

Neste método, dois problemas devem ser citados: a complexidade computacional e o número resultante de regras, as quais poderiam ser podadas utilizando algum critério, como por exemplo, o critério de definição da confiança.

As principais vantagens dos classificadores bayesianos são a inerência a ruídos e a boa base estatística, o que garante uma performance relativamente boa em domínios que envolvam muitos atributos irrelevantes.

A literatura experimental é consistente com estas vantagens e nos testes os pesquisadores relatam que o classificador *"Naive Bayes"* possui uma alta exatidão em muitos domínios naturais. Por exemplo, Cestnik, Kononenko e Bratko (1987) encontraram que os classificadores bayesianos executaram melhor do que técnicas mais sofisticadas. Langley et al. (1992) obteve que *"Naive Bayes"* executou melhor

do que um algoritmo com indução em árvores de decisão em quatro de cinco domínios naturais.

"Naive Bayes" obteve sucesso na detecção de mascaradores usando seqüência de comandos *"Unix"* [59].

4.5 Uma Regra (*"One Rule" ou 1R*)

4.5.1 Descrição

"One Rule" ou *1R* é um classificador simples que extrai um conjunto de regras baseada em um simples atributo, introduzido por [26].

O processo de aprendizagem usado por 1R testa regras associadas a um único atributo, classifica-as de acordo com a freqüência nos dados de treinamento, avalia a taxa de erro para cada atributo e escolhe a regra de melhor precisão. 1R manipula os valores desconhecidos tratando-os como um valor legítimo. Todos atributos numéricos são tratados como valores contínuos e são discretizados, ou seja, os valores-limite de um atributo são divididos dentro de vários intervalos disjuntos. Porém, o procedimento de discretização é sensível a ruído. Por exemplo, uma única instância com uma classe incorreta resultará muito provavelmente em um intervalo separado.

Uma solução simples para o problema de discretização é forçar um número mínimo de instâncias na classe majoritária por intervalo. O número mínimo de instâncias mais utilizado é 6 (seis) instâncias por intervalo. Este número foi encontrado empiricamente por alguns pesquisadores.

"One Rule" tem uma boa performance em muitos casos, porque, geralmente, em bancos de dados do mundo real, as estruturas nos dados são muito rudimentares e, assim, apenas um atributo é suficiente para classificar uma instância.

O objetivo final do algoritmo é encontrar uma regra para um atributo em particular que maximize a aprendizagem no conjunto de treinamento.

1R conta quão freqüentemente cada classe aparece para cada atributo, encontra a classe mais freqüente para cada um, constrói a regra que determina a classe para o seu valor, calcula a taxa de erro das regras, e escolhe a regra com a menor taxa de erro.

Em outras palavras, para alcançar o resultado final, *"One Rule"* calcula o número de instâncias que cada atributo classifica, discretiza valores numéricos, cal-

cula a precisão de cada regra, escolhe a regra mais precisa para cada um e, finalmente, escolhe a melhor regra dentre as regras encontradas. Um pseudocódigo mostrando todas as características do algoritmo *1R* é mostrado no Algoritmo 6.

Uma aplicação bem-sucedida de *"One Rule"* é a previsão de terremotos [60].

4.6 Máquina de Vetores Suporte (MVS - *"Support Vector Machine"*)

4.6.1 Descrição

Máquina de Vetores Suporte ou *MVS* [61] é um tipo de algoritmo de aprendizagem, originalmente introduzidos por Vapnik e colaboradores, e sucessivamente estendido por um número de outros pesquisadores. Sua notável performance robusta com respeito a dados esparsos e ruidosos é fazer deles um sistema de escolhas em várias aplicações, desde categorização de texto às classificações na área biológica.

MVS é um sistema para treinar eficientemente máquinas de aprendizagem linear em espaços de características induzidos no núcleo. *MVS* é uma eficiente maneira de aprender usando bons hiperplanos de separação em vastos espaços dimensionais.

Na aprendizagem linear, classificação binária é freqüentemente executada por usar uma função real valorada $f : X \subseteq \Re^n \rightarrow \Re$ da seguinte maneira: a entrada $x = (x_1, ..., x_n)'$ é determinado para a classe positivo, se $f(x) \geq 0$, e caso contrário para a classe negativa. Considerando-se o caso onde $f(x)$ é uma função linear de $x \in X$, portanto, pode-se representar como:

$$\begin{aligned} f(x) &= \langle w \times x \rangle + b \\ &= \sum_{i=1}^{n} w_i x_i + b \end{aligned}$$

, onde $(w, b) \in \Re^n \times \Re$ são os parâmetros que controlam a função e a regra de decisão é dada por $sgn(f(x))$. A interpretação geométrica desse tipo de hipótese é que o espaço de entrada X, é dividido em duas partes pelo hiperplano definido pela equação $\langle w.x \rangle + b = 0$. Um hiperplano é um subespaço de dimensão $n-1$ que divide o espaço em dois espaços que correspondem as entradas de duas classes distintas.

Definição. Usa-se X para denominar o espaço de entrada e Y para denominar o domínio de saída. Usualmente tem-se $X \subseteq \Re^n$, enquanto para a classificação binária $Y = -1, 1$, para classificação de m classes $Y = 1, 2, ..., m$ e para regressão $Y \subseteq \Re$. O conjunto treinamento é uma coleção de exemplos de treinamento, que

Algoritmo 6 Algoritmo do 1R.

1. **Algoritmo:** 1R
2. **Entrada:** L
3. **início**
4. Selecionar um atributo para nó raiz
5. No conjunto de treinamento, contar o número de exemplos na classe C,
6. sendo V o valor do atributo A. Armazenar esta informação em uma matriz de
7. 3 dimensões: $M[C, V, A]$.
8. Assumir que a classe *"default"* é a classe que possui mais exemplos no
9. conjunto treinamento. A precisão da classe *"default"* é dada por:
10. $P = \frac{N}{T}$, onde N é o número de exemplos da classe *default*
11. e T é o número total de exemplos.
12. Para cada atributo numérico A, criar uma versão nominal de A por definir
13. um número finito de intervalos de valores. Estes intervalos se tornam valores
14. de uma versão nominal de A. Por exemplo, se os valores numéricos de A são
15. particionados dentro de 3 intervalos, a versão nominal de A terá 3 valores:
16. intervalo1, intervalo2 e intervalo3
17. $M[C, \text{“}intervalo1'', A] = \sum_{i=1}^{n} M(C, V, A)$, onde $V \in intervalo1$
18. Definição:
19. Classe C é ótima para o atributo A, valor V, se maximiza
20. $M(C, V, A)$.Classe C é ótima para o atributo A, $intervaloI$,
21. se maximiza $M(C, \text{“}intervaloI'', A)$.
22. Para cada atributo A, usar a versão nominal dos atributos numéricos.
23. Construir uma hipótese envolvendo o atributo A por selecionar, para cada valor
24. V de A (também para valores não-conhecidos), uma classe ótima para V. Se
25. várias classes ótimas forem encontradas para um certo valor V, escolher
26. randomicamente.
27. Adicionar a hipótese construída para um conjunto chamado H. Este conjunto
28. conterá uma hipótese para cada atributo.
29. **fim**

são também chamados dados de treinamento, pode-se representar como:

$$S = ((x_1, y_1), ..., (x_\ell, y_l)) \subseteq (X \times Y)^\ell \tag{4.8}$$

onde ℓ é o número de exemplos, x_i representa um exemplo e y_i a classe do exemplo [62].

Representações de núcleo projetam os dados de entrada dentro de um espaço de característica, ou seja, mapeiam os dados em outro espaço o que pode representar uma simplificação na tarefa de aprendizagem. Nessa projeção, para aprender relações não lineares com uma máquina linear é necessário aplicar um mapeamento fixo não-linear dos dados para o espaço de característica, ou seja, selecionar um conjunto de características não lineares e reescrever os dados na nova representação. O conjunto de hipóteses serão funções do tipo

$$f(x) = \sum_{i=1}^{N} w_i \phi_i(x) + b,$$

onde $\phi : X \to F$ é um mapa não-linear do espaço de entrada para algum espaço de característica. Baseado nisso, máquinas não lineares podem ser construídas em dois passos: primeiro, um mapeamento não-linear fixo transforma os dados dentro de um espaço de característica F e, depois, uma máquina linear é usada para classificar esses dados no espaço de característica.

A hipótese pode ser expressa como uma combinação linear de pontos de treinamento, então a regra de decisão pode ser avaliada usando apenas produtos internos entre o ponto de teste e os pontos de treinamento:

$$f(x) = \sum_{i=1}^{\ell} \alpha_i y_i \langle \phi(x_i).\phi(x) \rangle + b. \tag{4.9}$$

Se existe uma maneira de calcular o produto interno $\langle \phi(x_i).\phi(x) \rangle$ no espaço de característica diretamente como uma função de pontos originais, torna-se possível combinar dois passos necessários para construir uma máquina de aprendizagem não-linear. Esse método computacional direto é chamado de função *"kernel"* ou núcleo.

Um *"kernel"* ou núcleo é uma função K, tal que para todo $x, z \in X$

$$K(x, z) = \langle \phi(x).\phi(z) \rangle. \tag{4.10}$$

O uso de *"kernel"* torna possível mapear os dados implicitamente dentro de um espaço de característica e treinar uma máquina linear em tal espaço. A

chave dessa abordagem é encontrar uma função *"kernel"* que possa ser avaliada eficientemente.

Quando o limite de decisão no espaço de entrada correspondendo a um hiperplano nos espaços de característica é uma curva polinomial de grau d, então estes *"kernels"* são freqüentemente chamados *"kernels"* polinomiais.

O teorema de *Mercer* fornece uma caracterização de quando a função $K(x, z)$ é um *"kernel"*. Considerando um finito espaço de entrada $X = x_1, ..., x_n$, e suponha $K(x, z)$ é uma função simétrica em X. Considere a matriz

$$K = (K(x_i, x_j))_{i,j=1}^{n}$$

desde que K é simétrico há uma matriz ortogonal V tal que $\mathbf{K} = \mathbf{V}\Lambda\mathbf{V}$', onde Λ é uma matriz diagonal contendo os *"eigenvalues"* $v_t = (v_{ti})_{i=1}^{n}$ as colunas de V [62]. Considere que $T : V \to V$ seja uma transformação linear. Se λ é um número e $v \neq 0$ é um vetor não zero tal que:

$$T(v) = \lambda v, \tag{4.11}$$

então λ é um *"eigenvalues"* e v é um *"eigenvector"* [63].

MVS, quando usado para classificação, separa o conjunto de treinamento dado com um hiperplano que é maximalmente distante dos dados, esse hiperplano é conhecido como o hiperplano margem maximal. O conjunto de dados de treinamento já deve estar previamente classificado em somente duas classes. Para casos nos quais nenhuma separação linear é possível, *MVS* pode trabalhar em combinação com a técnica de núcleos, que automaticamente realiza um mapeamento não linear para um espaço de característica. O hiperplano encontrado por *MVS* no espaço de característica corresponde ao limite de decisão linear no espaço de entrada.

Considere o ponto de entrada j-ésimo $x^j = (x_1^j, ..., x_n^j)$ seja a realização do vetor randômico X^j. Considere esse ponto de entrada ser classificado pela variável randômica $Y^j \in -1, +1$.

Considere $\phi : I \subseteq \Re^N \to F \subseteq \Re^N$ seja um mapeamento do espaço de entrada $I \subseteq \Re^N$ para o espaço de característica F. Assume-se que temos um exemplo S de m pontos de dados classificados: $S = (x^1, y^1), ..., (x^m, y^m)$. O algoritmo de aprendizagem de *MVS* encontra um hiperplano (w,b) tal que a quantidade

$$\gamma = \min_i y^i((w, \phi(x^1)) - b)$$

é maximizado, onde , denota um produto interno, o vetor w tem a mesma dimensionalidade de F, $||w||_2$ é mantido constante, b é um número real, e γ é chamado à margem. A quantidade $(\{w, \phi(x^i)\} - b)$ corresponde a distância entre o ponto x^i e o limite de decisão. Quando multiplicado pela etiqueta y^i, dá um valor positivo para todas as classificações corretas e um valor negativo para os incorretos. O mínimo desta quantidade sobre todos os dados é positivo se os dados são linearmente separáveis, e é chamado de margem. Dado um novo ponto de dados x para classificar, uma etiqueta é determinada de acordo com o seu relacionamento para o limite de decisão, e a função de decisão correspondente é

$$f(x) = sign(\{w, \phi(x)\} - b\} \tag{4.12}$$

É fácil provar [61] que, para o hiperplano maximal de margem,

$$w = \sum_{i=1}^{m} \alpha_i y^i \phi(x^i)$$

onde α_i são números reais positivos que maximiza

$$\sum_{i=1}^{m} \alpha_i - \sum_{ij=1}^{m} \alpha_i \alpha_j y^i y^j \phi(x^i), \phi(x^j)$$

sujeito a

$$\sum_{i=1}^{m} \alpha_i y^i = 0, \alpha_i > 0 \tag{4.13}$$

A abordagem de *MVS* tem sido aplicada com considerável sucesso em dois problemas em bioinformática: a classificação de padrões de expressão de genes [64] e a detecção da homologia da proteína em casos de similariedade de baixa seqüência [65].

Capítulo 5

O Processo de *"Data Mining"* na Solução do PPCC

5.1 Introdução

O problema de previsão de chuva é um problema prático que se deseja prever em muitas partes do globo. Por este motivo tem sido objeto de estudo sob vários aspectos. Uma instância deste problema foi o foco de atenção. A instância escolhida foi o Problema de Previsão de Chuva em Boa Viagem no Ceará (PPCBVC). Neste capítulo será apresentado todo processo de *"Data Mining"* realizado, descrevendo as fases para obtenção das soluções.

Este capítulo encontra-se dividido como segue. Na seção 5.2 será feita a descrição da seleção de variáveis. A seção 5.3 é dedicada ao pré-processamento dos dados. A transformação dos dados será descrita na seção 5.4. Na seção 5.5 será descrita a mineração dos dados. Finalmente na seção 5.6 será discutida a interpretação dos modelos gerados.

A seleção de variáveis consiste em selecionar variáveis relevantes, segundo a literatura de trabalhos anteriores e segundo a disponibilidade das fontes.

O pré-processamento dos dados é baseado na identificação das fontes de dados. Nesta etapa, estes terão de passar por uma fase de filtragem ou limpeza, buscando-se evitar ruídos.

A próxima etapa é denominada transformação dos dados. Nesta etapa, os dados são transformados no modelo requerido pelo algoritmo de mineração, os quais são pré-processados e realizam a padronização do banco de dados.

A quarta fase é a mineração dos dados que consiste na aplicação dos seis

algoritmos de mineração escolhidos. Como forma de avaliar a qualidade dos resultados e verificar a necessidade de uma nova execução dos algoritmos, será utilizada a metodologia de validação cruzada que estima a qualidade da generalização dos classificadores. Para a especificação das técnicas de Aprendizagem Automática é necessário representar e modelar detalhadamente as técnicas que serão utilizadas: Árvores Binárias de Decisão, Árvores de Decisão, 1R, Máquinas de Vetor de Suporte, *"Naive Bayes"* e Redes Neurais. Além disto, existe a geração dos modelos que se caracteriza pela extração de conhecimento dos dados, isto é, pela classificação de cada instância utilizando os modelos de previsão de chuva.

Na interpretação dos modelos gerados na etapa anterior, os dados obtidos são analisados e interpretados objetivando-se um entendimento do resultados face ao problema. Também, assimila-se o conhecimento que constitui-se na aplicação dos modelos em novas instâncias, ou melhor, etapa em que a previsão é realmente realizada.

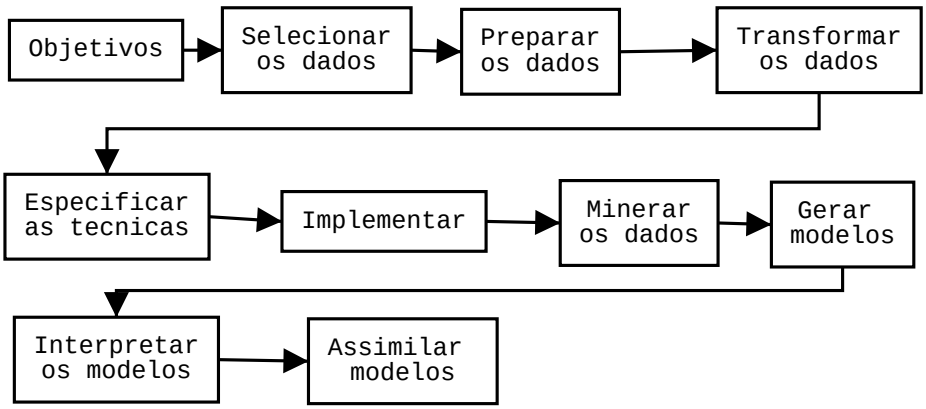

Figura 5.1: Fases da Dissertação.

As principais justificativas para a aplicação do processo de *"Data Mining"* para a previsão de chuva são: diminuir a taxa de erros e encontrar novos preditores. Além disso, caracterizar dependências entre variáveis em um nível conceitual e abstrato, produzir uma explicação causal da razão de existências das dependências e produzir uma descrição qualitativa de regularidades. Isto determina as dependências encontradas em fatores que não são explicitamente providos nos dados e representa uma analogia entre a regularidade descoberta e a regularidade em outro domínio.

5.2 Seleção de Variáveis

Primeiramente trabalhos anteriores foram analisados. Segundo a literatura de Climatologia, vários relacionamentos foram identificados entre a precipitação no nordeste e componentes meridional e zonal do vento, temperatura na superfície do mar, pressão ao nível do mar no Atlântico, posição da Zona de Convergência Intertropical (ZCIT) sobre o Atlântico e frentes frias; os quais são eventos da Oscilação Sul [1] (*El Niño* [2]).

O objetivo foi reunir o máximo possível de variáveis oceânicas, atmosféricas, entre outras, para tentar descobrir padrões nos dados em relação ao nível de chuva, na localidade chamada Boa Viagem no Estado do Ceará.

Foi verificado que em alguns modelos climatológicos, o principal sistema causador de chuvas no Ceará é a ZCIT (ver definição no capítulo 3).

Baseado em pesquisas realizadas no campo de relacionamentos entre fenômenos da natureza e trabalhos anteriores de previsão, como por exemplo em [35, 37, 32, 38], foram selecionadas as variáveis utilizadas na construção do conjunto de dados. As variáveis escolhidas foram:

- Componente Meridional do Vento na Superfície do Atlântico e do Pacífico (V);
- Componente Zonal do Vento na Superfície do Atlântico e no Pacífico (U);
- Índice de Manchas Solares ou Índice de Wolf (IMS);
- Precipitação em várias localidades do Ceará - SUDENE ($Precip_1$);
- Precipitação em várias localidades do Ceará - FUNCEME ($Precip_2$);
- Pressão ao nível do mar no Atlântico e no Pacífico (PSM);
- Temperatura da superfície do mar no Atlântico e no Pacífico (TSM) [38, 35];

Estes conjuntos são compostos de observações de navio e satélite no Atlântico e no Pacífico e registros de chuva no Ceará.

[1] A Oscilação Sul caracteriza-se por uma flutuação de pressão de grande escala observada sobre a Bacia do Pacífico Tropical.

[2] O *El Niño* é um fenômeno que se caracteriza pelo aquecimento acima do normal das águas oceânicas do centro-leste do oceano Pacífico Tropical, compreendendo desde à costa da América do Sul (nas proximidades do Peru e Equador) até aproximadamente uma longitude de 180^o).

Os dados foram obtidos na FUNCEME (Fundação Cearense de Meteorologia e Recursos Hídricos), na SUDENE (Superintendência do Desenvolvimento do Nordeste) através da FUNCEME e no Observatório Real da Bélgica (*"Royal Observatory of Belgium" - ORB*).

Os registros de precipitação são compostos por dois conjuntos distintos: o primeiro, consiste em observações feitas em 623 postos de medição da própria FUNCEME no período de 1974-2000, e o segundo é formado por observações em 445 postos da SUDENE no período 1912-1985. Porém, nos dois conjuntos ocorre um problema comum em dados reais; as dados são incompletos. No caso da FUNCEME, isto deve-se ao fato de alguns postos terem sido criados após 1974 ou terem sido desativados antes de 2000. Nos dados oriundos da SUDENE, provavelmente, problemas similares ocorreram gerando também séries incompletas.

Porém as estações da FUNCEME e da SUDENE não eram equivalentes, ou seja, algumas estações da FUNCEME não existem nos dados da SUDENE, e vice-versa. A resolução deste problema é descrito na Seção 5.2.

Também foram obtidos na FUNCEME os dados de temperatura na superfície do mar, pressão ao nível do mar, componente meridional e zonal do vento verificados de 1945 a 1989. Estas observações são dados mensais, possuem resolução espacial de 2^o de latitude $\times 2^o$ de longitude, tem como ponto inicial $0,5^o$ Oeste e $31,5^o$ Sul e o ponto final nas coordenadas $177,5^o$ Leste e $30,5^o$ Norte. Cada um desses contendo uma grade de 180 colunas por 32 linhas, resultando em um total de 5.760 pontos, onde 4.232 são pontos válidos de medições mensais. Estes dados abrangem áreas de superfície terrestre e marítima, nas quais os dados são realmente medidos, para fazer esta diferenciação, os dados inválidos (referentes à "terra") possuem um valor pré-determinado de -1×10^{10}. Os valores medidos são anomalias, ou seja, representam diferenças entre o valores absolutos medidos e uma média calculada no período.

Os índices de manchas solares (IMS) foram obtidos no *"Sunspot Index Data Center"* do Observatório Real da Bélgica e são observações mensais.

De acordo com os artigos [32],[34], [38] e [66] foram escolhidas duas áreas para análise: uma área no Atlântico e outra no Pacífico que mais se relacionam com a precipitação no Nordeste. Estas áreas delimitam os seguintes dados: componente zonal e meridional do vento, Temperatura da Superfície do Mar (TSM) e Pressão ao Nível do Mar (PSM).

O critério de escolha destas variáveis baseou-se nos trabalhos anteriores [35,

37, 32, 38] e na disponibilidade das bases de dados.

As bases de dados utilizadas foram:

1. Temperatura da Superfície do Mar - fonte: FUNCEME - periodicidade: mensal - período: 1945-1989 - total de pontos: 5760 pontos (32 linhas × 180 colunas);
2. Vento na Superfície do Mar - fonte: FUNCEME - periodicidade: mensal - período: 1945-1989 - total de pontos: 5760 pontos (32 linhas × 180 colunas);
3. Pressão na Superfície do Mar - fonte: FUNCEME - periodicidade: mensal - período: 1945-1989 - total de pontos: 5760 pontos (32 linhas × 180 colunas);
4. Precipitação - fonte: SUDENE (através da FUNCEME) - periodicidade: diário - total de pontos: 445- período: 1910-1985;
5. Precipitação - fonte: FUNCEME - periodicidade: diário - total de pontos: 445 - período: 1974-2000;
6. Índice de Manchas Solares - fonte: Observatório Real da Bélgica - periodicidade: mensal - total de pontos: 1 - período: 1949-2001;

A temperatura da superfície do mar (TSM), componente meridional do vento (U) sobre o mar, componente zonal do vento (V) sobre o mar e pressão no nível do mar (PSM) estão na grade 2^o latitude × 2^o longitude com 32 linhas e 180 colunas. Estas bases de dados possuem 5760 pontos mensais numa série de 45 anos, totalizando 540 meses no período de 1945 a 1989. Dentre estes 5760 pontos, 4232 pontos são sobre a superfície marítima e, portanto, apenas 4232 foram usados.

A série de precipitação escolhida compõe-se da precipitação diária em uma cidade da região do Sertão Central e Inhamuns, chamada Boa Viagem. Esta série foi combinada para obter uma série completa de 1945 até 1989.

5.3 Pré-processamento dos Dados

Nesta fase, a tarefa alvo é analisar e pré-processar os dados obtidos.

As séries de precipitação selecionadas e obtidas eram diferentes. Para unificar estes conjuntos foram comparadas a latitude e a longitude (localização geográfica) dos postos de medição para identificar postos equivalentes. A identificação nominal do município e do posto de cada estação foi utilizada apenas como um indicativo

de uma possível equivalência. As estações presentes em apenas um dos conjuntos foram desconsideradas. Esta combinação foi feita da seguinte forma: cada posto de dados da SUDENE teve sua série de dados adicionada a série existente no posto equivalente do outro conjunto. O objetivo desta mesclagem foi a obtenção de séries completas no período de 1945 a 2000. A série foi obtida por:

$$q^a = \left(\sum_{m=2}^{5} p^{m,a} \right) \tag{5.1}$$

Então, formou-se uma série completa de precipitação, criando-se um conjunto P com as variáveis $Precip_1$ e $Precip_2$, as seguintes características:

$$P_c = \begin{cases} p_c^{d,m,a} = precip_{1,c}^{d,m,a} & \text{se } 1910 \leq a \leq 1973 \\ p_c^{d,m,a} = precip_{2,c}^{d,m,a} & \text{se } 1973 < a \leq 1989 \end{cases}$$

onde c é o ponto da medida, d é o dia em que a medida foi realizada e m representa o mês referente, $a \in A = \{1910, ..., 1989\}$ representa os anos contidos no intervalo de tempo de 1910 à 1989 e $P_c \subset P$.

Para analisar a nova série de precipitação foi selecionada uma amostra dos dados. A amostra para análise constituíu-se da escolha de uma cidade de cada sub-região das cinco regiões homogêneas descritas em [36] (ver Tabela 5.7). O critério de escolha foi a homogeineidade e a série estar completa.

Região	Sub-região	Cidades
Litoral	(L1) Litoral Norte	Acaraú
	(L2)Litoral Pecém (Trairi-Pecém)	Itapajé
	(L3) Litoral Fortaleza (Caucaia-Beberibe)	Fortaleza
	(L4) Baturité	Pacoti
Ibiapaba	Ibiapaba	Tianguá
Jaguaribana	Jaguaribana	Jaguaribe
Cariri	Cariri	Caririaçu
Sertão Central e Inhamuns	Sertão Central e Inhamuns	Boa Viagem

Tabela 5.1: Cidades escolhidas para análise de precipitação.

Desta amostra, construíram-se vários gráficos de precipitação diária e mensal para verificar a existência de algum indicativo de padrão de comportamento (ver figura 5.2).

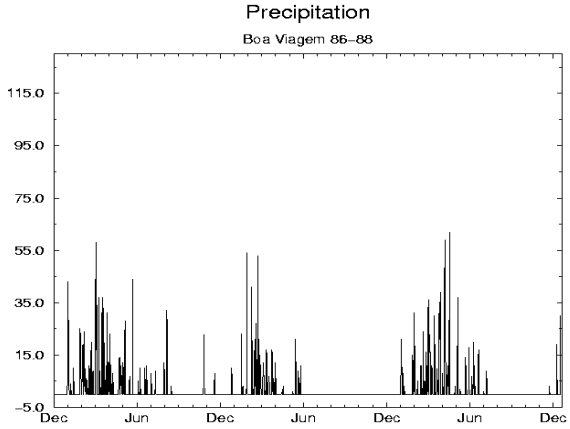

Figura 5.2: Precipitação em Boa Viagem de janeiro de 1986 a 1988.

Com a precipitação nestas cidades, uma análise foi feita para verificar a relação entre a quantidade de chuva no período de novembro a janeiro (chamada estação pré-chuvosa) e a quantidade de chuva de fevereiro a maio (chamada estação chuvosa).

$$P^n_{\text{estação pré-chuvosa}} = P^{n,\text{nov}} + P^{n,\text{dez}} + P^{n,\text{jan}}, \tag{5.2}$$

com $N = 1910, ..., 2000$

$$P^n_{\text{estação chuvosa}} = P^{n,\text{fev}} + P^{n,\text{mar}} + P^{n,\text{abr}} + P^{n,\text{mai}}, \tag{5.3}$$

com $N = 1910, ..., 2000$

Para classificar a estação pré-chuvosa, usou-se:

$$C^a_c = \begin{cases} classe3 & \text{se } P^a_c \geq mediana(P^a_{\text{estação pré-chuvosa,c}}) + \epsilon \\ classe2 & \text{se } mediana(P^a_{\text{estação pré-chuvosa,c}}) - \epsilon \leq P^a_c < \\ & \quad mediana(P^a_{\text{estação pré-chuvosa,c}}) + \epsilon \\ classe1 & \text{se } P^a_c \leq mediana(P^a_{\text{estação pré-chuvosa,c}}) - \epsilon \end{cases}$$

onde c é o ponto da medida e $a \in A = \{1910, ..., 1989\}$ representa os anos contidos no intervalo de tempo de 1910 à 2000 e $P_c \subset P$.

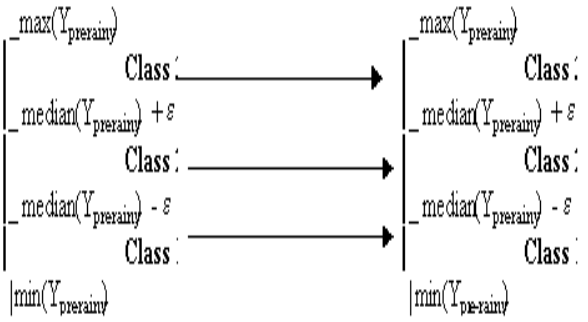

Figura 5.3: Classificação muito relacionada.

A classificação da estação chuvosa foi similar, como se nota a seguir:

$$C_c^a = \begin{cases} classe3 & \text{se } P_c^a \geq mediana(P^a_{\text{estação chuvosa,c}}) + \epsilon \\ classe2 & \text{se } mediana(P^a_{\text{estação chuvosa,c}}) - \epsilon \leq P_c^a < \\ & mediana(P^a_{\text{estação pré-chuvosa,c}}) + \epsilon \\ classe1 & \text{se } P_c^a \leq mediana(P^a_{\text{estação chuvosa,c}}) - \epsilon \end{cases}$$

onde c é o ponto da medida e $a \in A = \{1910, ..., 1989\}$ representa os anos contidos no intervalo de tempo de 1910 à 2000 e $P_c \subset P$.

Para mapear a classificação da estação chuvosa e da pré-chuvosa, foram encontrados três tipos de relacionamentos entre as classes: muito relacionadas, relacionadas e não relacionadas. O relacionamento "muito relacionadas" está representado na figura 5.3. Em todos os casos analisados, a maioria dos relacionamentos eram "muito relacionadas". O resultado desta análise serviu para se certificar que o atributo precipitação poderia servir como um indicativo para o mesmo.

As variáveis de TSM, PSM, V e U também precisavam de um pré-processamento, pois era necessário saber que pontos da superfície marítima compunham estes conjuntos e como se comportavam. A visualização teria de ser diferente da precipitação pelo número de pontos. A solução foi desenvolver um programa simples escrito na linguagem C baseado em código já existente, para visualizar estas medidas em tons de cinza e em cores. Um dos programas simples criados está descrito

em forma de algoritmo no Apêndice A no Algoritmo 8. Entre as representações testadas, a melhor foi obtida discretizando-se os valores em 16 classes (tons de cinza) e usando o preto para representar superfície terrestre e falta de valores. As menores temperaturas correspondiam aos tons mais escuros e as maiores, aos tons mais claros (ver figura 5.4).

É válido ressaltar que os conjuntos de TSM, PSM, V e U eram anomalias do período de 1945 a 1989. Anomalia em climatologia é a diferença entre o valor absoluto e um valor médio relativo a um certo período. Foram construídos gráficos de anomalia e de valores absolutos. Os arquivos de média foram fornecidos pela FUNCEME.

Produziu-se também animações anuais contendo 12 quadros, mas, nada de conclusivo foi notado.

O conjunto de IMS foi analisado, através de gráficos. Esta variável foi identificada ser a mais aleatória.

Um grande problema ainda tinha de ser analisado. O problema era o grande número de atributos em relação à quantidade de exemplos. A proporção pode ser notada na Tabela 5.2.

$\overbrace{TSM}^{4232\times12\ pontos}$	$\overbrace{U}^{4232\times12\ pontos}$	$\overbrace{V}^{4232\times12\ pontos}$	$\overbrace{PSM}^{4232\times12\ pontos}$	$\overbrace{Precip}^{12}$	$\overbrace{IMS}^{12}$
$t_1^{1,1945}$ ...	$u_1^{1,1945}$ ...	$v_1^{1,1945}$ ...	$l_1^{1,1945}$ ...	$p^{1,1945}, ...$	$s^{1,1945}, ...$
$t_1^{1,1946}$ ...	$u_1^{1,1946}$ ...	$v_1^{1,1946}$ ...	$l_1^{1,1946}$ ...	$p^{1,1946}, ...$	$s^{1,1946}, ...$
$t_1^{1,1947}$ ...	$u_1^{1,1947}$ ...	$v_1^{1,1947}$ ...	$l_1^{1,1947}$ ...	$p^{1,1947}, ...$	$s^{1,1947}, ...$
$t_1^{1,1948}$ ...	$u_1^{1,1948}$ ...	$v_1^{1,1948}$ ...	$l_1^{1,1948}$ ...	$p^{1,1948}, ...$	$s^{1,1948}, ...$
$\vdots$ ...	$\vdots$ ...	$\vdots$ ...	$\vdots$ ...	$\vdots$	$\vdots$
$t_1^{1,1989}$ ...	$u_1^{1,1989}$ ...	$v_1^{1,1989}$ ...	$l_1^{1,1989}$ ...	$p^{1,1989}, ...$	$s^{1,1989}, ...$

Tabela 5.2: Conjunto de instâncias com os atributos

A Tabela 5.2 mostra que a proporção é de $|A| = (4232 \times 12) \times 4 + 2 = 203.138$ atributos para $|I| = 45$ instâncias. Entretanto, com essa proporção, os métodos teriam um espaço de aprendizagem demasiadamente restrito. Este problema foi denominado, neste trabalho, Problema da Dimensionalidade dos Atributos (PDA).

Os dados de TSM, PSM e componentes U e V do vento pertenciam a uma grade de 180 colunas por 32 linhas, resultando em 5.760 pontos, dos quais, 4.232 eram relevantes por serem medições sobre a superfície do mar.

Para realizar uma redução nestas quantidades de atributos decidiu-se avaliar uma estratégia de *"ranking"*, ou seja, classificação na ordem decrescente de variação absoluta de anomalia. Os *"rankings"* obtidos eram listas mensais de classificação. Para cada mês, os primeiros 10 pontos foram escolhidos e marcados na grade original dos dados para a verificação da existência de padrões. O mesmo processo foi feito para 10, 40, 423 e 1270 pontos.

Em outras palavras, os dados de anomalia TSM, PSM, U e V foram analisados usando a variação absoluta dos valores de anomalia. Essa variação foi calculada, individualmente, em cada ponto, durante os 45 anos. Construindo-se gráficos a partir desses totais mensais, verificou-se a existência de agrupamentos com características similares. Os gráficos construídos mostravam os pontos que tiveram a maior variação absoluta durante o período analisado. Foi feita uma análise de agrupamentos formados com diferentes quantidades de pontos. Usando este conhecimento, os gráficos contendo 40 pontos foram os escolhidos. A figura mostra três exemplos contendo 40, 423 e 1270 primeiros pontos do *"ranking"* de TSM, respectivamente.

No tipo de visualização escolhida, encontrou-se 6 agrupamentos mais definidos nos gráficos de TSM. Para cada grupo desse, analisou-se o comportamento dos pontos do *"ranking"* em histogramas feitos de amostras construídas a cada três meses e a cada três anos (ver figura 5.3). Os histogramas apresentaram um certo padrão, validando assim, a análise dos *"rankings"* para encontrar os grupos. Nestes histogramas, os valores de variância, média, desvio padrão, coeficiente de variabilidade, mínimo e máximo serviram de parâmetros para medir a homogeneidade dos pontos em cada grupo. Estes valores estão na Tabela 5.3 para valores de anomalia de TSM e na Tabela 5.4 para temperatura absoluta do mar nos pontos.

Tabela 5.3: Análise do grupos com valores de anomalia de TSM

Grupo	Qtde Pontos	$\overline{X}$	s	CV	Mín	Máx
Grupo 1	12	0.82	0.04	0.05	0.75	0.92
Grupo 2	18	0.84	0.02	0.02	0.81	0.86
Grupo 3	61	0.99	0.07	0.07	0.81	1.11
Grupo 4	79	1.00	0.05	0.05	0.91	1.11
Grupo 5	115	0.91	0.10	0.11	0.75	1.18
Grupo 6	138	0.96	0.05	0.05	0.88	1.11

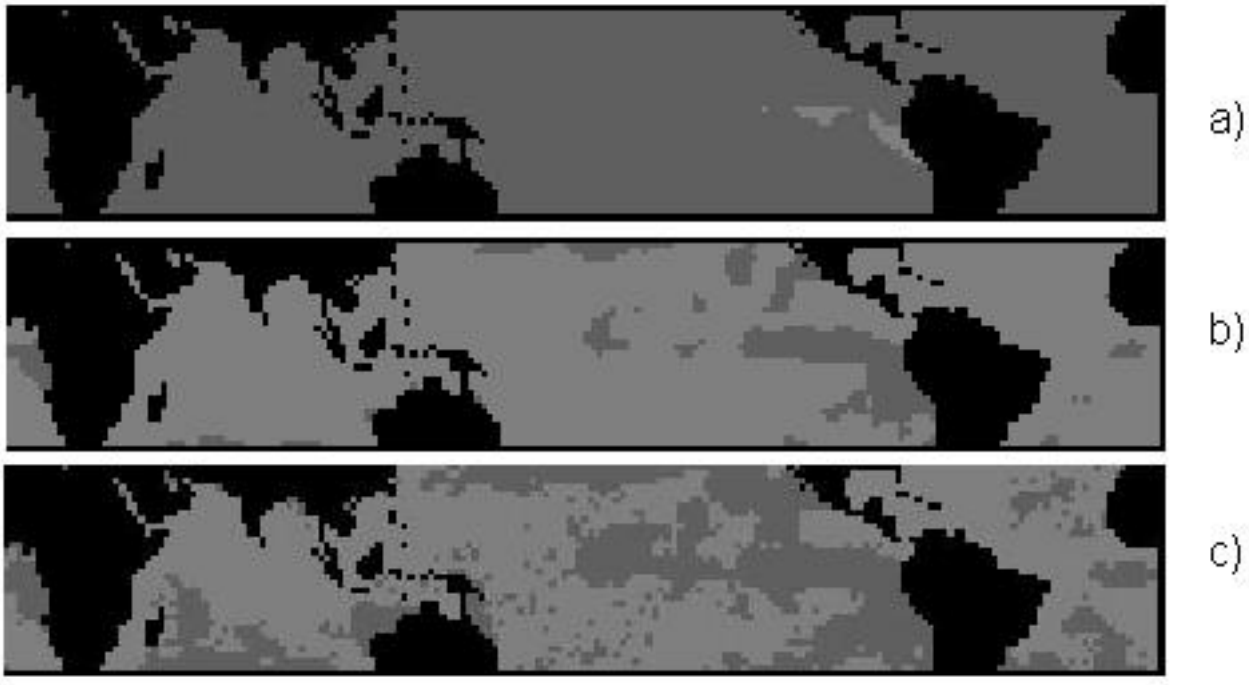

Figura 5.4: Gráficos do *"ranking"* de TSM de junho de 1945 a 1989 ((a) 40 primeiros pontos, (b) 423 primeiros pontos e (c) 1.270 primeiros pontos).

Tabela 5.4: Análise do grupos com valores de TSM absoluto dos pontos

Grupo	Qtde Pontos	$\overline{X}$	s	CV	Mín	Máx
Grupo 1	12	22.43	1.45	0.06	19.93	24.07
Grupo 2	18	25.03	0.63	0.03	23.80	25.86
Grupo 3	61	26.81	0.68	0.03	25.24	27.75
Grupo 4	79	23.07	0.96	0.04	21.44	25.24
Grupo 5	115	23.83	2.25	0.09	15.77	27.76
Grupo 6	138	21.90	2.08	0.10	16.30	26.05

A figura 5.4 mostra quais os grupos escolhidos de janeiro a junho e a figura 5.5 mostra os grupos do segundo semestre.

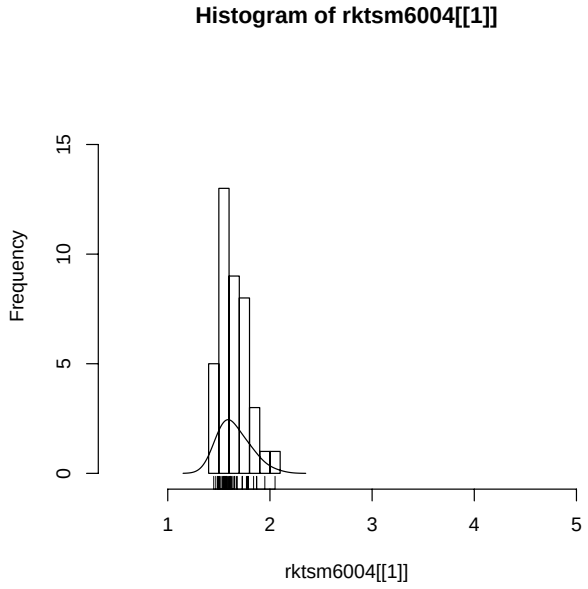

Figura 5.5: Histograma de *"ranking"* de 40 pontos de TSM de abril de 1960.

5.3.1 Método de Redução de Atributos Meteorológicos

Esta subseção tem como objetivo expor uma proposta de método de redução para o Problema da Dimensionalidade dos Atributos (PDA), definido anteriormente. O PDA visa minimizar o número de atributos de uma mesma variável, em um conjunto de instâncias. O objetivo do problema é especificar uma redução $T = \{t_1^{1,1945}, ..., t_{4232}^{12,1989}\} \mapsto \phi(T) = \left(g_1^{1,1945}, g_2^{1,1945}, ..., g_n^{12,1989}\right)$, de modo que padrões contidos nos dados não sejam perdidos.

Baseado nas análises dos dados de TSM através de diversos gráficos, encontrou-se os grupos descritos anteriormente.

Para formalizar o uso destes grupos, um método heurístico foi proposto. O método está descrito no Algoritmo 7.

Esse problema não possui até o momento nenhum algoritmo que encontre uma redução com garantia de performance e sem perda de informações. Este método não foi totalmente formalizado e uma comparação com outros métodos é sugerido como trabalho futuro.

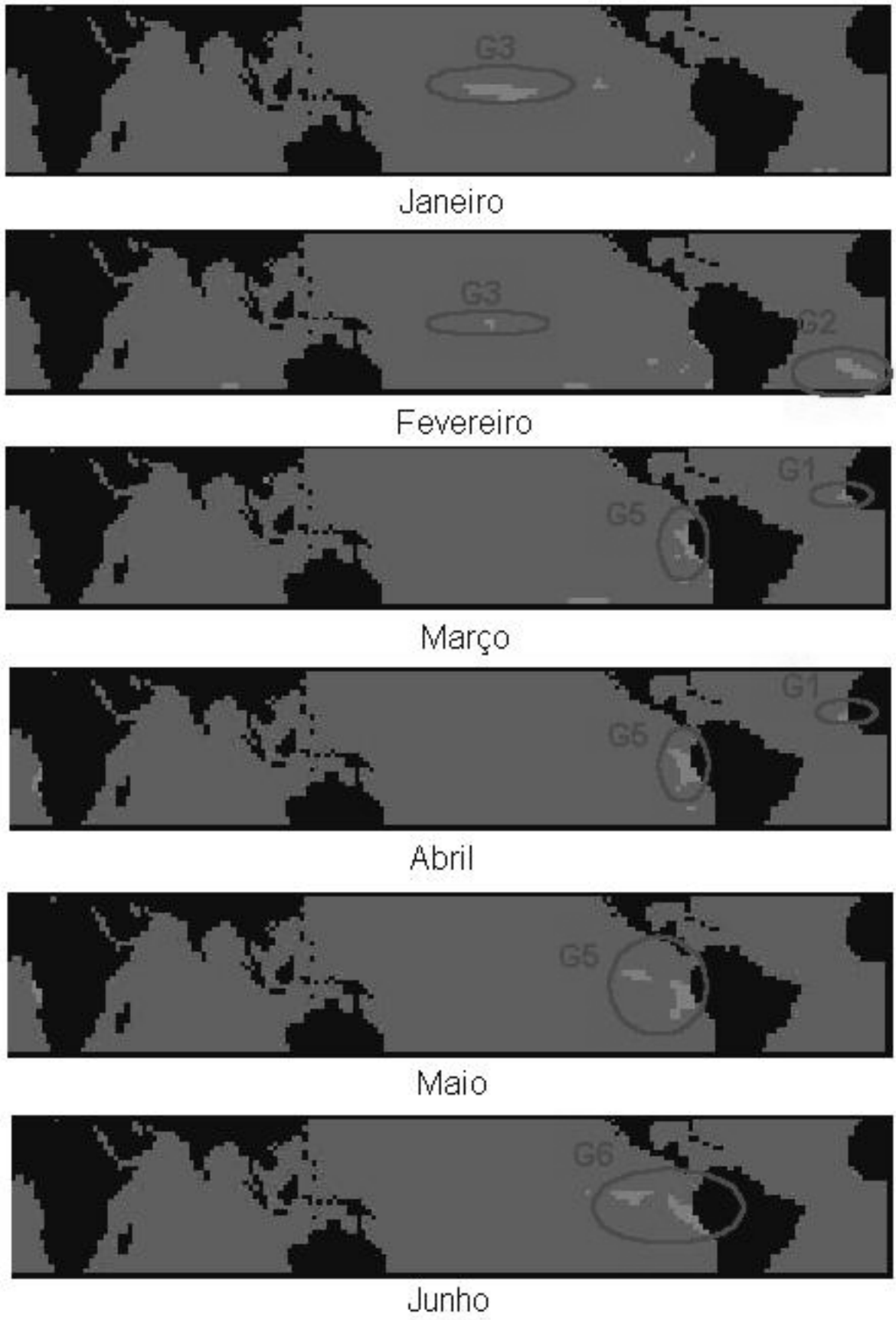

Figura 5.6: Gráficos dos grupos de *"ranking"* de TSM de janeiro a junho.

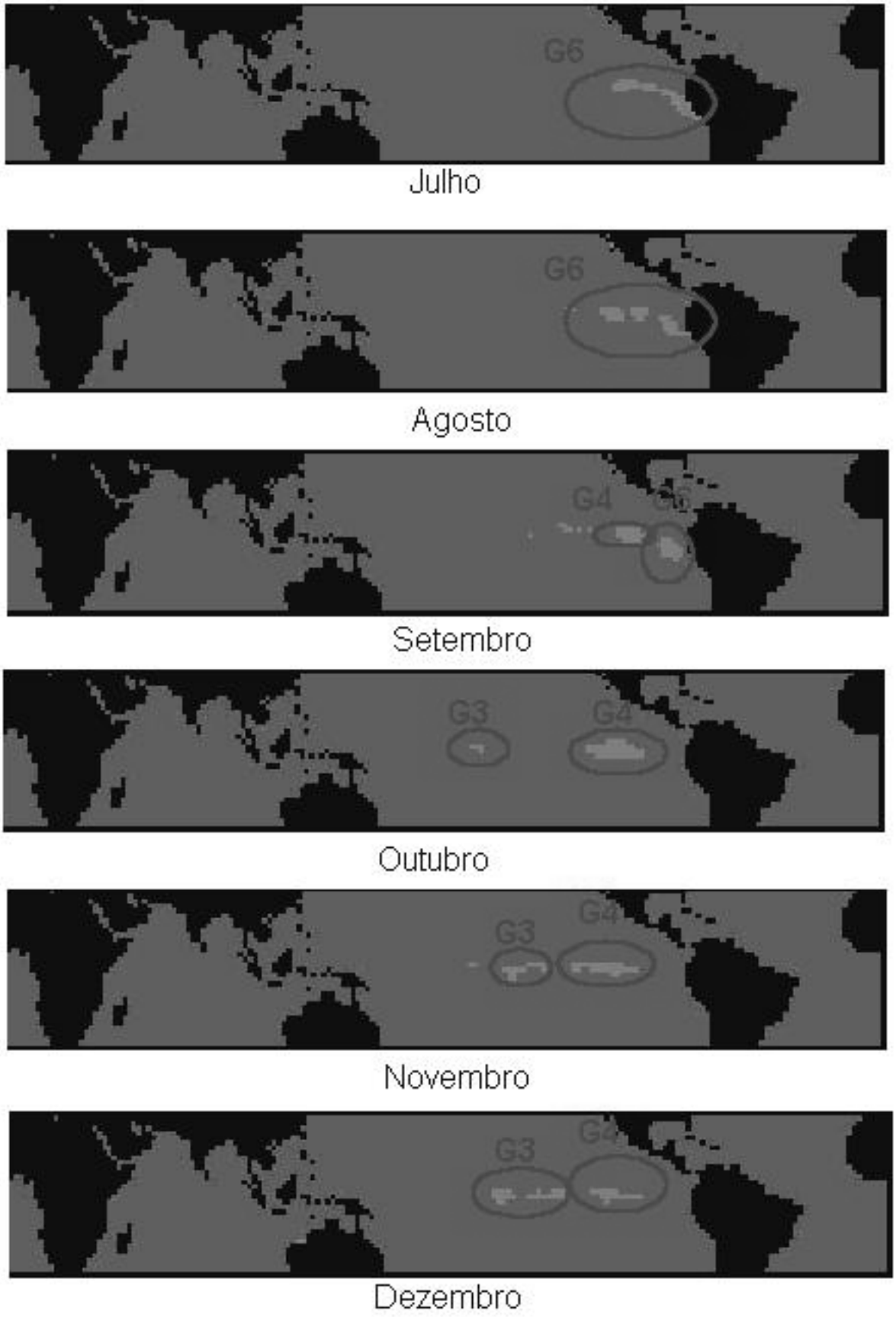

Figura 5.7: Gráficos dos grupos de *"ranking"* de TSM de julho a dezembro.

Algoritmo 7 Método de redução de atributos.

1. **Método:** REDUÇÃO DE ATRIBUTOS
2. **início**
3. **Entrada:** L
4. // Um vetor L de n posições de dados absolutos
5. Calcular a variação absoluta mensal de todos os N pontos da série temporal
6. Classificar em ordem decrescente de variação absoluta
7. Escolher $\frac{N}{100}$ primeiros pontos de cada ranking mensal
8. Agrupar estes pontos usando a comparação dos menores: desvio padrão, CV, média
9. aritmética, amplitude, distância máx e o CV da distância
10. Escolher melhores grupos
11. Calcular a média de cada grupo escolhido como atributo
12. **fim**

5.4 Transformação dos Dados

Na fase de transformação dos dados estes são adequados ao modelo analítico requerido pelos algoritmos de mineração.

A primeira característica considerada para transformação foi a periodicidade. As variáveis pertencentes ao conjunto P_c possuíam uma periodicidade diária, enquanto que as demais eram mensais. Para transformar estas instâncias diárias em mensais foi utilizada o seguinte mapeamento:

$$P_c = (p_c^{1,1,1910}, p_c^{2,1,1910}, ..., p_c^{31,12,1989}) \mapsto \phi(P_c) = \left(\sum_{d=1}^{31} p_c^{d,1,1910}, ..., \sum_{d=1}^{31} p_c^{d,12,1989} \right) \tag{5.4}$$

Para restringir o espaço da variável $p_c \in P$, foi escolhido o subconjunto P_c com $c = \{'BoaViagem'\}$, pelas características do subconjunto de dados completos e por pertencer a uma região homogênea chamada "Sertão Central e Inhamuns" [35].

De acordo com os histogramas e com as outras medidas, os grupos apresentaram uniformidade com pequenas variações nos padrões. Portanto, escolheu-se estes grupos para reduzir o número de atributos. As reduções foram dadas pelas equações:

$$T^{m,a} = \begin{cases} \sum t_i^{m,a} & \text{,se } i \in G_1 \\ \sum t_i^{m,a} & \text{,se } i \in G_2 \\ \sum t_i^{m,a} & \text{,se } i \in G_3 \\ \sum t_i^{m,a} & \text{,se } i \in G_4 \\ \sum t_i^{m,a} & \text{,se } i \in G_5 \\ \sum t_i^{m,a} & \text{,se } i \in G_6 \end{cases}$$

ou,

$T = \{t_1^{1,1945}, ..., t_{4232}^{12,1989}\} \mapsto \phi(T) = \left(g_1^{1,1945}, g_2^{1,1945}, ..., g_5^{12,1989}, g_6^{12,1989}\right)$

O mesmo mapeamento foi aplicado para as variáveis de PSM, U e V, usando os grupos de TSM.

Depois desse mapeamento, a proporção é $|A| = (6 \times 12) \times 4 + 2 = 290$ atributos para $|I| = 45$ instâncias. Entretanto, essa quantidade de instâncias ainda era reduzida em relação ao tamanho de A.

Essa restrição foi resolvida, adicionando-se dois novos atributos ao conjunto A, $mes \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, representando o mês associado à instância, e $ano \in \{1, 2\}$, que representa o ano associado, sendo o valor 1 para ano anterior e 2 para o ano anterior ao ano anterior. De outra forma, foram desmembradas as instâncias que eram anuais, para instâncias mensais. Além de usar para previsão não só um ano anterior, mas os dois anos anteriores a classe objetivo (quadra chuvosa). O formato da nova instância é mostrado na Tabela 5.5.

$\overbrace{TSM}^{6}$	$\overbrace{U}^{6}$	$\overbrace{V}^{6}$	$\overbrace{PSM}^{6}$	$\overbrace{Precip}^{1}$	$\overbrace{IMS}^{1}$	$\overbrace{mes}^{1}$	$\overbrace{ano}^{1}$
...	...	...	...	$p^{1,45}$	$s^{1,45}$	1	1
...	...	...	...	$p^{2,45}$	$s^{2,45}$	2	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
...	...	...	...	$p^{12,89}$	$s^{12,89}$	12	2

Tabela 5.5: Conjunto de instâncias com os atributos

Essa adição, aumentou o tamanho do conjunto I para $|I| = 1044$ instâncias e diminuiu A para $|A| = (6 \times 4) + 4 = 28$ atributos. Resolvido assim, o problema de espaço de aprendizagem, os esforços foram voltados para a formação das classes o que será apresentado no item 5.3.

Os mesmos seis grupos ou regiões encontrados nos gráficos de TSM foram utilizados para as variáveis PSM, V e U, para construir a primeira base de dados

para treinamento e teste. A segunda base de dados foi constituída de grupos de pontos encontrados e calculados, baseados nas respectivas variáveis.

Cada grupo desse foi transformado em apenas um valor, calculando-se a média de grupo correspondente, como descrito na fórmula:

$$G_i = \frac{\sum_{j=1}^{n_i} v_j}{n_i}, onde \;\; i = 1, 2, ..., 6 \tag{5.5}$$

Este processo reduziu de 4232 pontos para 6 resultados dos grupos, para cada variável.

Resumindo, as variáveis usadas foram 6 grupos de TSM, PSM, V e U, precipitações mensais, índices de manchas solares. A escolha deve-se à importância da temperatura da superfície do mar (TSM), em relação à precipitação, à pressão e aos componentes do vento. As instâncias foram construídas com os 12 meses anteriores e os 24 meses anteriores destes atributos para calcular a quadra chuvosa. Os 24 meses anteriores à quadra chuvosa a ser calculada. Ao todo, com todos os atributos formou-se 1.044 instâncias

5.5 Mineração dos Dados

Para minerar os dados, escolheu-se os seguintes métodos de Aprendizagem Automática *1R*, *C4.5*, Redes Neurais, *"Naive Bayes"*, Máquina de Vetor de Suporte e *CART*, descritos no capítulo anterior. A implementação destes métodos tornou-se desnecessária, devido a utilização de dois ambientes de aprendizagem já implementados o *"Weka"* e o CART [9]. O *"Weka"* é um ambiente baseado em java que permite a aplicação de alguns algoritmos de classificação. CART implementa árvores de decisão.

A indução em árvores de decisão foi um dos métodos escolhidos, porque sua utilização em probabilidade de regressão e classificação para visualização dos dados, é bastante indicada, por consistir em uma ferramenta flexível na análise destes. Além disso, árvores de decisão permitem estimar a probabilidade de erro na classificação de um novo objeto.

Algumas das vantagens de uma rede neural são: possuir boa fundamentação teórica, ser uma abordagem robusta para aproximação de funções objetivo com valores reais ou discretos e ser resistente a ruídos presentes nos dados. Mesmo apresentando uma avaliação muito rápida de uma nova instância, requer porém um tempo de treinamento relativamente longo [47, 67].

"One Rule" é um dos métodos mais simples e foi selecionado pela boa performance em domínios naturais. *"Support Vector Machine"* produz bons resultados, é rápido e preciso em muitos domínios.

Vários testes foram realizados com diversas combinações dos atributos, diferentes modos de definição de classes (ver Tabela 5.2).

Tabela 5.6: Definição de conjuntos de atributos

Todos os atributos	
Qtde. Atributos	Atributos
28	$G_1(t^{m,a})$,$G_2(t^{m,a})$,$G_3(t^{m,a})$,$G_4(t^{m,a})$, $G_5(t^{m,a})$, $G_6(t^{m,a})$, $G_1(l^{m,a})$,$G_2(l^{m,a})$,$G_3(l^{m,a})$,$G_4(l^{m,a})$, $G_5(l^{m,a})$,$G_6(l^{m,a})$, $G_1(u^{m,a})$,$G_2(u^{m,a})$,$G_3(u^{m,a})$,$G_4(u^{m,a})$, $G_5(u^{m,a})$,$G_6(u^{m,a})$, $G_1(v^{m,a})$,$G_2(v^{m,a})$,$G_3(v^{m,a})$,$G_4(v^{m,a})$, $G_5(v^{m,a})$,$G_6(v^{m,a})$, S, Ano,Mes, $Precip$
Atributos Reduzidos	
Qtde. Atributos	Atributos
10	$G_1(t^{m,a})$,$G_2(t^{m,a})$,$G_3(t^{m,a})$,$G_4(t^{m,a})$,$G_5(t^{m,a})$,$G_6(t^{m,a})$, S, Ano,Mes, $Precip$
Atributos TSM	
Qtde. Atributos	Atributos
8	$G_1(t^{m,a})$,$G_2(t^{m,a})$,$G_3(t^{m,a})$,$G_4(t^{m,a})$, $G_5(t^{m,a})$,$G_6(t^{m,a})$, Mes

Para que as técnicas pudessem ser aplicadas o problema, teve-se que optar por algumas especificações de técnicas. Estas escolhas estão resumidas nas Tabela 5.7 para três classes e 5.8 para duas classes.

No algoritmo C4.5 com melhor resultado, algumas especificações precisaram ser escolhidas. O limiar de confiança para o procedimento de poda escolhido foi de 0.25. Não forçou-se o uso de árvores sem poda e nem suprimiu-se o crescimento de subárvores. Executou-se o procedimento padrão de poda de erro reduzido, o qual poda a árvore para otimizar a performance no conjunto treinamento. O número de conjuntos para a poda de erro reduzido usado foi 3 e o número mínimo de instâncias em nó folha foi 2. Construi-se árvores com nós internos possuindo mais de dois ramos e não usou-se o estimador de *"Laplace"*.

Para o algoritmo *"One Rule"* apenas uma opção foi especificada o número

mínimo de instâncias por intervalo. Utilizou-se o valor de 6 (seis) instâncias por intervalo.

O método *"Naïve Bayes"* teve como configuração o uso do estimador de *"kernel"*.

A Máquina de Vetor de Suporte (MVS) usada possui as seguintes configurações: limite superior de pesos igual a 1.0, o grau dos polinomiais igual a 1, tolerância de 0.0010, uso de dados normalizados e epsilon com valor de $1.0E-12$.

As Redes Neurais tiveram como algoritmo de aprendizagem *"Backpropagation"*, taxa de aprendizagem de 0.3, quantidade de nós na camada de saída igual a 3 e de entrada igual a 20, limiar de validação de 20, o uso de atributos normalizados, momentum com o valor de 0.2 e a normalização das classes numéricas. A diferenciação estava na quantidade de camadas intermediárias. A primeira Rede Neural testada era baseada no trabalho anterior [34] e tinha duas camadas escondidas, com 2 e 3 nós, respectivamente. A segunda Rede Neural escolhida de forma arbitrária tinha uma camada escondida com a nós, sendo a o limite inferior inteiro de $a' = \frac{(N^o atributos + N^o classes)}{2}$, logo $a = 11$.

No CART, optou-se pela árvore de classificação, a validação cruzada (VC) de *"ten-fold"*, o critério de escolha da melhor árvore e o critério de Gini para classificação.

Os equipamentos usados para estes testes foram dois Pentium, com 32 e 128 MB de memória. Todos os testes apresentados nesta seção foram executados por 10 vezes, sendo cada execução realizado durante o período médio de meia hora.

5.6 Interpretação

Nesta seção será apresentada a interpretação dos experimentos computacionais realizados durante o desenvolvimento da disertação, usando as instâncias que foram geradas de acordo com o método descrito na Seção 6.1. O principal objetivo foi verificar qual método daria a melhor previsão. Além disso, descobrir que atributos influenciavam, positiva ou negativamente nos resultados e qual era a melhor forma de construir as classes. Foram geradas instâncias com classes construidas usando-se as técnicas de quantil, percentil e amplitude. As tabelas 5.10, 5.11 e 5.12 mostram os melhores resultados obtidos para duas e três classes construídas com a técnica da amplitude.

A partir dos resultados, observou-se que a técnica de amplitude levou aos

Tabela 5.7: Especificação das técnicas escolhidas para teste com 3 classes.

Técnica	**Característica**	**Valor**
C4.5	Usa árvores sem poda	Não
	Permite crescimento de subárvores	Sim
	Limiar de confiança para poda	0.25
	Procedimento de poda: erro reduzido	Não
	N^o de conj. p/ poda de erro reduzido	3
	Divisões binárias	Não
	N^o mínimo de instâncias em um nó folha	Não
	Usa o estimador de *"Laplace"*	Não
"One Rule"	Tamanho mínimo de instâncias por intervalo	6
"Naive Bayes"	Usa estimador de *"kernel"*	Sim
Rede Neural	Quantidade de nós (camada de entrada)	20
	Quantidade de nós (camadas internas)	11
	Quantidade de nós (camada de saída)	3
	Limiar de validação	20
	Aprendizagem	*"Backpropagation"*
	Atributos normalizados	sim
	Momentum	0.2
	Classe numérica normalizada	sim
	Taxa de aprendizagem	0.3
Rede Neural 1	Quantidade de nós (camada de entrada)	20
	Quantidade de nós (camadas internas)	2, 3
	Quantidade de nós (camada de saída)	3
	Limiar de validação	20
	Aprendizagem	*"Backpropagation"*
	Atributos normalizados	sim
	Momentum	0.2
	Classe numérica normalizada	sim
	Taxa de aprendizagem	0.3
CART	Tipo de árvore	classificação
	Melhor árvore	custo mínimo
	Classificação de árvore	Gini
	Teste	VC 10-fold

Tabela 5.8: Especificação das técnicas escolhidas para teste com 2 classes.

Técnica	**Característica**	**Valor**
C4.5	Usa árvores sem poda	Não
	Permite crescimento de subárvores	Sim
	Limiar de confiança para poda	0.25
	Procedimento de poda: erro reduzido	Não
	N^o de conj. p/ poda de erro reduzido	3
	Divisões binárias	Não
	N^o mínimo de instâncias em um nó folha	Não
	Usa o estimador de *"Laplace"*	Não
"One Rule"	Tamanho mínimo de instâncias por intervalo	6
"Naive Bayes"	Usa estimador de *"kernel"*	Sim
MVS	Limite superior dos pesos	1.0
	ϵ	1.0E-12
	Dados normalizados	sim
	Grau dos polinomiais	1
	Tolerância	0.0010
Rede Neural 2	Quantidade de nós (camada de entrada)	20
	Quantidade de nós (camadas internas)	11
	Quantidade de nós (camada de saída)	2
	Limiar de validação	20
	Aprendizagem	*"Backpropagation"*
	Atributos normalizados	sim
	Momentum	0.2
	Classe numérica normalizada	sim
	Taxa de aprendizagem	0.3
Rede Neural 3	Quantidade de nós (camada de entrada)	20
	Quantidade de nós (camadas internas)	2, 3
	Quantidade de nós (camada de saída)	2
	Limiar de validação	20
	Aprendizagem	*"Backpropagation"*
	Atributos normalizados	sim
	Momentum	0.2
	Classe numérica normalizada	sim
	Taxa de aprendizagem	0.3
CART	Tipo de árvore	classificação
	Melhor árvore	custo mínimo
	Classificação de árvore	Gini
	Teste	VC 10-fold

melhores resultados. Testou-se instâncias de três tipos, sendo esses tipos: todos os atributos, atributos reduzidos e atributos de TSM.

Pelos resultados dos testes, vê-se que, os grupos TSM são os atributos mais significativos, levando aos melhores resultados. Entretanto, o melhor resultado foi obtido com instâncias contendo os atributos reduzidos (ver Tabela 5.6).

Os métodos foram comparados usando-se as seguintes medidas: percentagem de acertos das previsões, erro quadrático médio, erro absoluto médio e erro absoluto relativo.

O erro quadrático médio é a principal e mais utilizada medida de performance; algumas vezes a raiz quadrada é calculada para dar as mesmas dimensões do valor previsto. A desvantagem é tender a exagerar no efeito de anomalias, ou seja, instâncias cuja previsão de erro é maior do que as outras [58]. Este erro é calculado por:

$$\frac{(p_1 - a_1)^2 + ... + (p_n - a_n)^2}{n} \tag{5.6}$$

Outra medida usada é o erro absoluto médio que calcula a média da magnitude dos erros individuais sem levar em conta seu sinal [58]. O erro absoluto médio é representado por

$$\frac{|p_1 - a_1| + ... + |p_n - a_n|}{n} \tag{5.7}$$

O erro absoluto relativo é simplesmente o erro absoluto total [58] e calcula-se por:

$$\frac{|p_1 - a_1| + ... + |p_n - a_n|}{|a_1 - \bar{a}| + ... + |a_n - \bar{a}|} \tag{5.8}$$

, onde $\bar{a} = \frac{1}{n}\sum_i a_i$

Para os testes, os dados foram testados de duas formas: a primeira, com um terço dos dados só para testes, e a segunda, usando validação cruzada. Neste processo, decide-se o número de partições do conjunto inicial. O número de partições testados foi 10.

Os seis métodos tiveram seus resultados comparados de acordo com a taxa de acerto, erro quadrático médio, erro absoluto médio e erro absoluto relativo.

A taxa de acerto usando validação cruzada é um bom padrão, mas não é confiável pela variação substancial que existe nesta técnica de teste.

Tabela 5.9: Teste com 10 atributos com 528 instâncias de 1945-1989 usando validação cruzada para 3 classes.

Método	Acertos(%)	erro absoluto médio	erro quadrático médio	erro absoluto relativo
CART	70.45	0.23	0.30	90.32
C4.5	69.51	0.27	0.39	95.02
1R	66.86	0.22	0.47	77.07
"Naive Bayes"	66.86	0.26	0.38	90.66
Rede Neural	64.20	0.25	0.44	87.46
Rede Neural 1	69.89	0.26	0.39	92.33

Tabela 5.10: Teste com 10 atributos com 528 instâncias de 1945-1989 usando validação cruzada para 2 classes ($C_1, C_2 \cup C_3$).

Método	Acertos(%)	erro absoluto médio	erro quadrático médio	erro absoluto relativo
MVS ($C_1, C_2 \cup C_3$)	58.52	0.45	0.50	94.00
C4.5 ($C_1, C_2 \cup C_3$)	61.74	0.41	0.56	85.37
1R ($C_1, C_2 \cup C_3$)	60.98	0.39	0.62	80.69
"Naive Bayes" ($C_1, C_2 \cup C_3$)	61.36	0.43	0.48	89.12
Rede Neural 2 ($C_1, C_2 \cup C_3$)	56.63	0.45	0.59	92.98
Rede Neural 3 ($C_1, C_2 \cup C_3$)	54.54	0.46	0.51	95.40

Tabela 5.11: Teste com 10 atributos com 528 instâncias de 1945-1989 usando validação cruzada para 2 classes ($C_1 \cup C_2, C_3$).

Método	Acertos(%)	erro absoluto médio	erro quadrático médio	erro absoluto relativo
MVS ($C_1 \cup C_2, C_3$)	97.73	0.28	0.29	605.14
C4.5 ($C_1 \cup C_2, C_3$)	97.73	0.04	0.15	96.30
1R ($C_1 \cup C_2, C_3$)	97.73	0.02	0.15	49.26
"Naive Bayes" ($C_1 \cup C_2$ e C_3)	97.35	0.05	0.17	100.53
Rede Neural 2 ($C_1 \cup C_2, C_3$)	96.97	0.04	0.17	93.92
Rede Neural 3 ($C_1 \cup C_2, C_3$)	97.16	0.04	0.16	95.74

O erro quadrático médio é um bom e fácil indicativo de performance. Considerando-se esta medida, o melhor método foi o *MVS*.

A magnitude de erros individuais é medida pelo erro absoluto médio. A melhor performance foi obtida pelo *C4.5*.

O erro absoluto relativo é o erro absoluto total com o mesmo tipo de normalização e nesse critério o melhor método foi *C4.5*.

Em termos gerais, considerando os critérios citados o método mais preciso foi o *MVS*. Porém, o *MVS* apresentou um erro absoluto relativo muito significativo no teste usando duas classes ($C_1 \cup C_2, C_3$), devido ao desbalanceamento dos exemplos, ou seja, a alta concentração de exemplos da primeira classe em relação à segunda classe resultou em alto valor do erro absoluto relativo.

Capítulo 6

Resultados Computacionais

Nesta dissertação, métodos não-paramétricos são propostos para o Problema de Previsão de Chuva no Ceará. Pela impossibilidade de garantir matematicamente a qualidade das previsões encontradas por esses métodos, a experimentação torna-se uma parte importante do trabalho. Neste capítulo, serão apresentados os resultados obtidos com os métodos escolhidos.

Primeiramente, será descrito na Seção 6.1 o conjunto de instâncias usadas para o treinamento dos métodos. Uma parte destas instâncias serviu como conjunto de treinamento e outra para ser o conjunto de teste. Na Seção 6.2 será descrito como foram escolhidas e construídas as instâncias. Na Seção 6.3 será descrito, como as classes são formadas. Finalmente, serão apresentados os resultados dos testes com os métodos na Seção 6.4.

6.1 Instâncias

Durante o desenvolvimento desta dissertação uma das maiores dificuldades e preocupação foi a obtenção de um conjunto de dados que fosse representativo para o problema. Nesta seção, serão descritas as instâncias que foram usadas para a construção do conjunto de treinamento e do conjunto de testes.

Com o propósito de realização de testes para este trabalho, foram utilizadas vários conjuntos de instâncias. Os tipos de instâncias utilizados como conjunto de treinamento e conjunto de teste dos métodos escolhidos, foram os seguintes:

- Foi criado um regra de formação de conjunto de instâncias T_1 contendo todos os 29 atributos dos dados: mes, TSM_1, TSM_2, TSM_3, TSM_4, TSM_5, TSM_6, PSM_1, PSM_2, PSM_3, PSM_4, PSM_5, PSM_6, U_1, U_2, U_3, U_4, U_5, U_6, V_1, V_2,

V_3, V_4, V_5, V_6, IMS, A, $Precip$ e C. As instâncias serão descritas na Seção 6.1.

- Um segundo tipo de conjunto de instâncias criado foi T_2 contendo 11 atributos dos dados: mes, TSM_1, TSM_2, TSM_3, TSM_4, TSM_5, TSM_6, IMS, A, $Precip$ e C. As instâncias serão descritas na Seção 6.1.

- Um terceiro tipo de conjunto de instâncias criado foi T_3 contendo todos os 6 atributos dos dados: mes, TSM_1, TSM_2, TSM_3, TSM_4, TSM_5, TSM_6 e C. As instâncias serão descritas na Seção 6.1.

Os tipos de conjuntos descritos acima serviram como regra de formação para os conjuntos abaixo:

- Baseado no tipo T_1, foi criado o conjunto C_1 de instâncias com todas as instâncias no período de 1945 a 1989. As instâncias serão descritas na Seção 6.1.

- Baseado no tipo T_1, um segundo conjunto C_2 de instâncias com todas as instâncias no período de 1945 a 1970.

- Baseado no tipo T_1, um conjunto C_3 de instâncias com todas as instâncias no período de 1971 a 1989.

- Baseado no tipo T_2, foi criado o conjunto C_4 de instâncias com todas as instâncias no período de 1945 a 1989. As instâncias serão descritas na Seção 6.1.

- Baseado no tipo T_2, um conjunto C_5 de instâncias com todas as instâncias no período de 1945 a 1970.

- Baseado no tipo T_2, um conjunto C_6 de instâncias com todas as instâncias no período de 1971 a 1989.

- Baseado no tipo T_3, foi criado o conjunto C_7 de instâncias com todas as instâncias no período de 1945 a 1989. As instâncias serão descritas na Seção 6.1.

- Baseado no tipo T_3, um conjunto C_8 de instâncias com todas as instâncias no período de 1945 a 1970.

- Baseado no tipo T_3, um conjunto C_9 de instâncias com todas as instâncias no período de 1971 a 1989.

6.2 Geração das Instâncias

Uma das dificuldades enfrentadas durante todo o trabalho foi o pequeno número de instâncias do mundo real obtidas para treinar e testar os algoritmos. Os conjuntos de instâncias foram obtidos nas seguintes fontes: FUNCEME (Fundação Cearense de Meteorologia e Recursos Hídricos), SUDENE (Superintendência de Desenvolvimento do Nordeste) através da FUNCEME e Observatório Real da Bélgica (*"Royal Observatory of Belgium"*). Apesar disso, o número de instâncias completo era somente 45 anuais. Um problema maior surge porque o número de atributos é superior ao de exemplos e os atributos possuem periodicidades diferentes.

A grande maioria dos trabalhos anteriores combinam os atributos e transformam o número de instâncias para aumentar o conjunto de dados. Desta maneira, duas das tarefas realizadas no decorrer da pesquisa foram a transformação das instâncias e a combinação delas.

Para realizar estas tarefas, foram usadas informações advindas dos trabalhos de Hastenrath e Greischar [34] e Hall [44]. No primeiro artigo citado, o conjunto de dados usado é semelhante e também foi obtido na Funceme. Entretanto, as instâncias foram formadas diferentemente. As variáveis que foram usadas encontram-se descritas na tabela 6.1.

Tabela 6.1: Atributos

Atributo	Período	Periodicidade
1. Temperatura na superfície do mar (TSM)	1945-1989	mensal
2. Componente U do vento na superfície do mar (U)	1945-1989	mensal
3. Componente V do vento na superfície do mar (V)	1945-1989	mensal
4. Pressão na superfície do mar (PSM)	1945-1989	mensal
5. Precipitação 1 ($Precip_1$)	1910-1985	diária
6. Precipitação 2 ($Precip_2$)	1974-1989	diária
7. Índice de manchas solares (IMS)	1949-2001	mensal

As variáveis de TSM, U, V e PSM citadas na tabela 6.1 se encontravam em uma grade 2^o latitude $\times$ 2^o longitude com 32 linhas e 180 colunas na seguinte

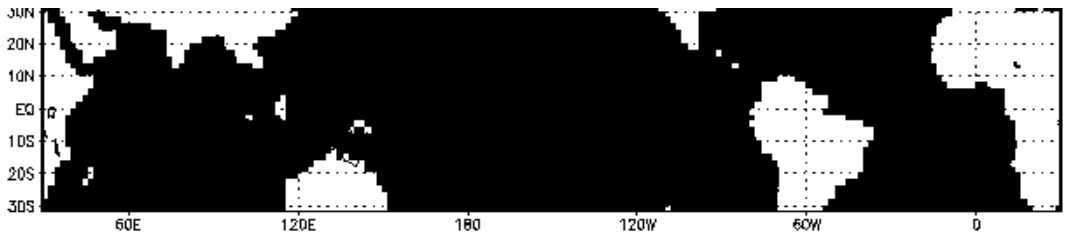

Figura 6.1: Grade das variáveis de TSM, PSM, U e V.

resolução espacial de 2×2 graus. O ponto inicial é $0,5^o$ O e $31,5^o$ S e o ponto final nas coordenadas $177,5^o$ L e $30,5^o$ N, demonstrados na figura 6.1.

Dos atributos contidos na tabela anterior, todos foram combinados para a composição das instâncias e possuem as seguintes características:

1. Temperatura da Superfície do Mar (TSM), representado por $t_i^{m,a} \in T$, $T = t_1^{1,1945}, ..., t_i^{m,a}$, onde $i = 1, ..., 4232$ pontos, $m = 1, ..., 12$ meses e $a = 1945$, ..., 1989;

2. Componente U do vento na Superfície do Mar (U), representado por $u_i^{m,a} \in U$, $U = u_1^{1,1}, ..., u_i^{m,a}$, onde $i = 1, ..., 4232$ pontos, $m = 1, ..., 12$ meses e $a = 1945, ..., 1989$;

3. Componente V do vento na Superfície do Mar (V), representado por $v_i^{m,a} \in V$, $V = v_1^{1,1945}, ..., v_i^{m,a}$, onde $i = 1, ..., 4232$ pontos, $m = 1, ..., 12$ meses e $a = 1945, ..., 1989$;

4. Pressão na Superfície do Mar (PSM), representado por $l_i^{d,m,a} \in L, L = l_1^{1,1}, ..., l_i^{d,m,a}$, onde $i = 1, ..., 4232$ pontos, $m = 1, ..., 12$ meses e $a = 1945, ..., 1989$;

5. Precipitação 1 ($Precip_1$), representado por $p_{1,c}^{d,m,a} \in P_{1,c}$, $P_{1,c} = p_{1,c}^{1,1,1910}, ..., p_{1,c}^{d,m,a}$, onde $d = 1, ..., 31$ dias, $d = 1, ..., 12$ meses, $a = 1910, ..., 1985$, c é o local da medição e $P_{1,c} \subset P_1$;

6. Precipitação 2 ($Precip_2$), representado por $p_{2,c}^{d,m,a} \in P_{2,c}, P_{2,c} = p_{2,c}^{1,1,1974}, ..., p_{2,c}^{d,m,a}$, onde $d = 1, ..., 31$ dias, $d = 1, ..., 12$ meses,$a = 1974, ..., 1989$ meses, c é o local da medição e $P_{2,c} \subset P_2$;

7. Índice de Manchas Solares (IMS), representado por $s^{m,a} \in S, S = s^{1,1949}, ..., s^{m,a}$, onde $m = 1, ..., 12$ meses e $a = 1949, ..., 2001$;

Com os dados citados, foram feitas algumas transformações para possibilitar a criação das instâncias.

Primeiro, em relação à formação de uma série completa de precipitação, preparou-se um novo conjunto P com as variáveis $Precip_1$ e $Precip_2$, as seguintes características citadas na Seção 5.3.

As variáveis de TSM, PSM, V e U também precisavam de um pré-processamento. São 4232 pontos mensais, o que comparado ao número de instâncias era muito alto. Então, um estudo nestes conjuntos de dados foi feito para diminuir o número de atributos em relação ao número de instâncias.

Para resolver este problema de dimensões, foram feitos *"rankings"* da variação absoluta de todos os pontos de TSM, PSM, U e V, separadamente, durante os 45 anos. Os *"rankings"* construídos tiveram 10, 20, 40 e 1270 pontos respectivamente. Estes pontos foram plotados usando a mesma grade dos dados originais, com um destaque apenas para os pontos do ranking. Exemplos dos gráficos gerados seguem abaixo:

Os gráficos descritos anteriormente foram comparados em relação à formação de regiões homogêneas ou agregados. Os gráficos mais representativos foram aqueles contendo 40 pontos, por ter apresentado uma maior homogeneidade entre as regiões.

As regiões encontradas nestes gráficos foram agrupadas em 6 grupos para cada variável. Dentro de cada um dos 6 grupos, a correlação entre os pontos foi medida com o cálculo da média, variância, maior e menor valor. Os valores obtidos foram satisfatórios com indicativos de homogeneidade entre os pontos.

Também, foram feitos histogramas dos 40 pontos mensais em amostras de 1 a cada 3 meses e de 1 a cada 3 anos para analisar se o comportamento (distribuição) dos pontos, apresentava algum padrão ou era aleatório. Nos histogramas, um certo comportamento padrão nas curvas foi verificado. O padrão pode ser verificado nas figuras que se seguem.

6.3 Geração das Classes

A classe objetivo deste trabalho é a quadra chuvosa Q, que segundo definição da literatura, constitui-se das chuvas verificadas nos meses de fevereiro, março, abril e maio. O local de medição escolhido foi $c = \{'BoaViagem'\}$.

Para a construção das classes, foram usados totais da variável $Precip$ da obtidas pela fórmula 5.1.

Usando a fórmula citada obteve-se resultados com valores contínuos. Porém, a aplicação de alguns métodos requeria a discretização destes valores. Estes valores foram discretizados utilizando-se três técnicas estatísticas distintas já definidas anteriormente, foram: amplitude, percentil e quantil. Além destes métodos, após testes, dois divisores com valores fixos foram usados na discretização. Esses valores eram 600 e 800. No caso do método de *MVS* foram usados dentro de cada técnica mencionada, a combinação das classes de modo a formar duas classes.

Com amplitude, valores fixados e percentil, classificou-se os valores em três classes distintas formando o conjunto $Q = \{seco,\ normal,\ chuvoso\}$. A técnica de quantil aplicada resultou no conjunto de classes $Q' = \{\ muitoseco,\ seco,\ normal,\ chuvoso,\ muitochuvoso\}$.

6.4 Testes

Usando as instâncias que foram geradas de acordo com as reduções descritas na Seção 13.2, os métodos de aprendizagem escolhidos foram treinados e testados. Os conjuntos de treinamento usados foram descritos na Seção 6.1. Os tipos de testes utilizados foram um conjunto de teste fornecido contendo $\frac{1}{3}$ dos exemplos e validação cruzada. Os testes executados para cada um dos 4 tipos de construção de classes estão descritos na tabela 6.2.

Tabela 6.2: Estrutura dos testes realizados para cada um dos 4 tipos de construção de classes discretizadas

Testes			
Período (trein.)	Qtde. instâncias(trein.)	Qtde. atributos	Período (testes)
1945-1989	1044	28	Validação cruzada
1945-1970	624	28	1971-1989
1945-1989	528	10	Validação cruzada
1945-1970	312	10	1971-1989
1945-1989	528	8	Validação cruzada
1945-1970	312	8	1971-1989

Vários testes foram realizados com diversas combinações dos atributos, diferentes modos de definição de classes, diferentes técnicas de aprendizagem e modos de teste. Como descreveu-se na Seção 5.5.

O melhor resultado destes conjuntos de treinamento e teste foi o treinamento feito com 10 atributos, com 528 instâncias de 1945-1989, usando como teste validação cruzada. Este resultado está resumido na Tabela 5.9.

É válido notar que avaliação é a chave do processo de *"Data Mining"*, já que existem muitas maneiras de inferência a partir dos dados. Logo, performance no conjunto de treinamento não é uma medida significativa. Porém, um bom indicador de performance pode ser encontrado usando um conjunto de dados independente para o cálculo da performance [58].

A proporção de exemplos classificados corretamente é uma medida imparcial e precisa da probabilidade de erro, quando o tipo de teste utilizado é a validação cruzada. No entanto, para avaliar o sucesso dos resultados obtidos outras medidas são necessárias. As outras medidas foram o erro quadrático médio, erro absoluto médio e erro absoluto relativo (ver equações 5.5, 5.6 e 5.7).

Pode-se observar (ver Tabela 5.9,5.10 e 5.11), que os grupos de TSM conseguem ajudar na previsão, comprovando que uma boa redução foi feita nos dados. Além disto, o método que conseguiu se adaptar e aprender mais foi MVS possuindo um nível de acerto bom, prevendo bem se a classificação da quadra for "seca".

Capítulo 7

Conclusão

Neste trabalho foram apresentadas soluções não-paramétricas para a previsão da quadra chuvosa em Boa Viagem, uma cidade do Ceará, situada numa região caracterizada por chuvas escassas. Este é um problema do mundo real, que se tenta resolver anualmente. As soluções propostas baseiam-se na adequação do problema para utilização de métodos computacionais já existentes, na busca de uma melhor previsão.

Pode-se citar algumas contribuições à teoria e à prática introduzidas nesta dissertação:

1. Foi feita uma pesquisa geral de métodos paramétricos e não-paramétricos aplicados ao problema, o que pode ser de grande interesse para o embasamento de trabalhos futuros.

2. Foi desenvolvido algoritmos simples para visualização de dados em escalas de cinza e visualização de grupos, o que pode ser reutilizado em trabalhos futuros.

3. Foi proposto um método de redução de atributos meteorológicos específicos ao tipo de problema.

4. Finalmente, foi feita na dissertação uma comparação entre os métodos utilizados.

Na comparação feita entre os métodos, para os experimentos com duas classes combinadas $K_2 = \{Seca \cup Normal, Chuvosa\}$ e $K_3 = \{Seca, Normal \cup Chuvosa\}$, a melhor aprendizagem foi feita pelo algoritmo MVS. Para os testes com três classes, os melhores foram o CART e o C4.5. A avaliação baseou-se na análise da percenta-

gem de acerto, no erro médio quadrático, no erro relativo absoluto e no erro absoluto médio.

Nos experimentos realizados, verificou-se que os atributos mais importantes para as previsões foram os grupos de TSM. Também verificou-se que o uso da amplitude para discretizar o atributo-classe consistiu no melhor método para construção das classes.

Em síntese, os resultados das previsões foram bons, mas notou-se uma certa tendência nos mesmos. Isto pode ter sido causado pelo desbalanceamento do número de exemplos de cada classe. Portanto, seria importante que em um possível trabalho futuro, comparações dos algoritmos fossem feitas usando conjuntos de exemplos com pesos atribuídos às instâncias de menor número. Além desta sugestão, também podem constituir possíveis futuras direções, os seguintes tópicos:

1. *Treinamento e testes com mais instâncias* Um dos maiores problemas encontrado no trabalho foi a obtenção de instâncias reais do problema. Portanto, um tópico futuro possível seria a aplicação e validação destes resultados, usando uma massa de dados maior.

2. *Grupos de PSM, U e V.* Neste trabalho, os agrupamentos analisados se restringiram a variável de TSM. Entretanto, as variáveis de PSM, U e V possuem um potencial ainda inexplorado. Como trabalho futuro, pode-se explorar estes atributos e aproveitá-los também para a redução de variáveis.

Referências Bibliográficas

[1] HOLSHEIMER, M., SIEBES, A. P. J. M., *Data Mining: The Search for Knowledge in Databases*, Report CS-R9406, P.O. Box 94079, 1090 GB Amsterdam, 1994.

[2] RODIONOV, S. N., MARTIN, J. H., "An Expert System-Based Approach to Prediction of Year-to-Year Climatic Variations in the North Atlantic Region", v. 4, pp. 88–93, 1988.

[3] NAMIAS, J., CAYAN, D. R., "Large-scale air-sea interactions and short period climate fluctuations", *Science*, pp. 869–876, 1981.

[4] CABENA, P., HADJINIAN, P., STADLER, R., *et al.*, *Discovering Data Mining From Concept To Implementation*. USA, IBM, 1997.

[5] FRAWLEY, W. J., PIATETSKY-SHAPIRO, G., MATHEUS, C. J., "Knowledge discovery databases: An overview". In: Piatetsky-Shapiro, G., Frawley, W. J. (eds.), *Databases*, pp. 1–27, Cambridge-MA-USA: AAAI/MIT, 1991.

[6] HAN, J., "OLAP Mining: Integration of OLAP with Data Mining". In: *DS-7*, pp. 1–11, 1997.

[7] CRAVEN, M., SHAVLIK, J., "Machine Learning Approaches to Gene Recognition", *Artificial Intelligence and Molecular Biology*, v. 3, 1994.

[8] CARVALHO, F. F. L. D., "Auxílio ao diagnóstico de tumores do ângulo ponto-cerebelar, com a utilização de técnicas de inteligência artificial", 2000.

[9] BREIMAN, L., FRIEDMAN, J., OLSHEN, R., *et al.*, *Classification and Regression Trees*. Belmont, California, USA, Wadsworth Int. Group, 1984.

[10] VAPNIK, V. N., CHERVONENKIS, Y., "On the uniform convergence of relative frequencies of events to their probability", *Theory of Probability and its Applications*, pp. 264–280, 1971.

[11] GOMES, F. C., GASCUEL, O., "SDL, a stochastic algorithm for learning decision lists with limited complexity", *Annual Math. Artificial Intelligence*, v. 10, pp. 281–302, 1994.

[12] MITCHELL, T., "Generalization as search", *Artificial Intelligence*, v. 37, pp. 203–226, 1982.

[13] HAN, J., CAI, Y., CERCONE, N., "Knowledge discovery in databases: An attribute-oriented approach". pp. 547–559, Vancouver, British Columbia, Canada, 1992.

[14] QUINLAN, J. R., "The effect of noise on concept learning,".

[15] WONG, S. K. M., ZIARKO, W., "Comparison of the probabilistic approximate classification and fuzzy set model", *Fuzzy Stes and Systems*, , n. 21, pp. 357–362, 1982.

[16] YAO, Y. Y., WONG, K. M., "A decision theoretic framework for approximating concepts", *International Journal Man-Machine Studies*, v. 37, pp. 793–809, 1992.

[17] QUINLAN, J. R., "Induction of Decision Trees". In: Shavlik, J., Dietterich, T. (eds.), *Readings in Machine Learning*, M. Kaufmann, 1990. Originally published in *Machine Learning* 1:81–106, 1986.

[18] LEE, S. K., "An extended relational database model for uncertain and imprecise information". In: *18th VLDB conference*, pp. 211–218, Vancouver, British Columbia, Canada, 1992.

[19] QUINLAN, J. R., "Uknown attribute values in induction". In: Segre, A. M. (ed.), *Proceedings of the Sixth International Machine Learning Workshop*, San Mateo, CA, USA, pp. 164–168, 1989.

[20] GRZYMAALA-BUSSE, J. W., "On the unknown attribute values in learning from examples". In: Ras, Z. W., Zemankowa, M. (eds.), *Lectures Notes in AI*, pp. 368–377, New York: Springer Verlag, 1991.

[21] DEOGUN, J., RAGHAVAN, V., SARKAR, A., *et al.*, "Data Mining: Trends in Research and Development", 1996.

[22] DEOGUN, J. S., RAGHAVAN, V. V., SEVER, H., "Exploiting upper approximations in the rough set methodology". In: Fayyad, U., Uthurusamy, R. (eds.), *The First International Conference on Knowledge Discovery and Data Mining*, Montreal, Quebec, Canada, pp. 69–74, 1995.

[23] KIRA, K., RENDELL, L., "The feature selection problem: Traditional method and a new algorithm". In: *Proceedings of AAAI-92*, AAAI Press, pp. 129–134, 1992.

[24] ALMUALLIM, H., DIETTERICH, T., "Learning with many irrelevant features". In: *Proceedings of AAAI-91*, Menlo Park, CA - USA, AAAI Press, pp. 547–552, 1991.

[25] PAWLAK, Z., SLOWINSKI, K., SLOWINSKI, R., "Rough classification of patients after highly selective vagotomy for duodenal ulcer", *International Journal of Man-Machine Studies*, v. 37, pp. 413–433, 1986.

[26] HOLTE, R. C., "Very simple classification rules perform well on most commonly used datasets", *Machine Learning*, v. 11, pp. 63–90, 1993.

[27] RUSSELL, S. J., NORVIG, P., *Artificial Intelligence A Modern Approach*. New York, Prentice-Hall, Inc. Upper Saddle River, 1995.

[28] C. CORINNA, H. DRUCKER, D. H., VAPNIK, V., "Capacity and complexity control in predicting the spread between barrowing and lending interest rates", *The First International Conference on Knowledge Discovery and Data Mining*, pp. 51–76, 1995.

[29] ZHONG, N., OHSUGA, S., "Discovering concept clusters by decomposing databases", *Data and Knowledge Engineering*, v. 12, pp. 223–244, 1994.

[30] DIETTERICH, T. G., MICHALSKI, R. S., "A comparative review of selected methods for learning from examples". pp. 41– 81.

[31] MARTINS, M. E. G., "Noções de Estatística".

[32] UVO, C., REPELLI, C., ZEBIAK, S., *et al.*, "A Study on the Influence of the Pacific and Atlantic SST on the Northeast Brazil Monthly Precipitation Using Singular Value Decomposition". pp. 210–216.

[33] GOLDBERG, R. A., "Solar Activity and the weather - is there a connection?", 1999.

[34] HASTENRATH, S., GREISCHAR, L., "Further Work on the Prediction of the Northeast Brazil Rainfall Anomalies", *Journal of Climate*, pp. 743–758, Abr 1993.

[35] XAVIER, T., XAVIER, A., "Papel da componente meridional do vento na costa do nordeste brasileiro e de outras covariáveis para prever a chuva no estado do Ceará", *Revista Brasileira de Recursos Hídricos*, v. 3, n. 4, pp. 121–139, Out/Dez 1998.

[36] XAVIER, T., XAVIER, A., "Caracterização de períodos secos ou excessivamente chuvosos no estado do Ceará através da técnica dos quantis: 1964-1998", *Revista Brasileira de Meteorologia*, v. 14, n. 2, pp. 63–78, 1999.

[37] AZEVEDO, P., SILVA, B., RODRIGUES, F., "Previsão Estatística das Chuvas de Outono no Estado do Ceará", *Revista Brasileira de Recursos Hídricos*, v. 13, n. 1, pp. 19–30, 1998.

[38] REPELLI, C., ALVES, J., "Use of Canonical Correlation Analysis to Predict the Spatial Rainfall Variability over Northeast Brazil". pp. 225–230.

[39] MARKHAN, C., "Apparent periodicities in rainfall at Fortaleza", *Journal of Applied Meteorology*, v. 13, pp. 176–179, 1974.

[40] GIRARDI, C., TEIXEIRA, L., "Prognóstico a longo prazo para o Nordeste brasileiro", *CTA/IAE(Relatório Técnico ECA -06/78)*, v. 6, 1978.

[41] NOBRE, C., YANASSE, H., YANASSE, C., "Previsão das secas no Nordeste pelo método das periodicidades: usos e abusos", *INPE - 2344RPE/407*, v. 1, 1982.

[42] BRITO, J., NOBRE, C., ZARANZA, A., "Precipitação da pré-estação chuvosa do norte do Nordeste", *Climanálise*, v. 6, pp. 39–53, 1991.

[43] HASTENRATH, S., HELLER, L., "Dynamics of climatic hazards in Northeast Brazil", *Quart. J. Roy. Meteor. Soc.*, v. 103, pp. 77–92, 1977.

[44] HALL, T., BROOKS, H., DOSWELL, C., "Precipitation Forecasting Using a Neural Network", *Weather and Forecasting*, , 1997.

[45] MIYANO, T., GIROSI, F., "Forecasting Global Temperature Variations by Neural Networks", *Massachusetts Institute of Technology -Artificial Intelligence Laboratory*, v. 3, 1994.

[46] GOMES, F. A. D. C., "Utilisation d'algorithmes stochastiques en apprentissage", 1992.

[47] MITCHELL, T., *Machine Learning*. USA, McGraw-Hill, 1997.

[48] QUINLAN, J., "An Empirical Comparison of Genetic and Decision-Tree Classifiers". In: Kaufmann, M. (ed.), *Proceedings Fifth International Machine Learning Conference*, pp. 135–141, 1988.

[49] NG, K., LIU, H., "Customer Retention via Data Mining", 1999.

[50] SCHWARZ, R., "Predicting Wine Quality from Terrain Characteristics by Regression Trees", 1996.

[51] WANG, M. Q., HIRSCHBERG, J., "PREDICTING INTONATIONAL PHRASING FROM TEXT", 1996.

[52] BITTENCOURT, G., *Inteligência Artificial: Ferramentas e Teorias*. Campinas, São Paulo, Brasil, Campinas: Instituto de Computação - UNICAMP, 1996.

[53] BRAGA, A., CARVALHO, A., LUDERMIR, T., *Fundamentos das Redes Neurais Artificiais*. Rio de Janeiro, DCC/IM - Núcleo de Computação Eletrônica, COPPE/Sistemas - UFRJ, 1998.

[54] HAYKIN, S., *Neural Networks (Computer Science)*. New Jersey, USA, Prentice-Hall, 1994.

[55] GRADOJEVIC, N., YANG, J., "The Application of Artificial Neural Networks to Exchange Rate Forecasting: The Role of Market Microstructure Variables", 2000.

[56] C. LEE GILES, STEVE LAWRENCE, A. C. T., "Noisy Time Series Prediction using a Recurrent Neural Network and Grammatical Inference", *Machine Learning*, v. 44, pp. 161–183, 2001.

[57] HECKERMAN, D., "Bayesian Networks for Data Mining", *Data Mining and Knowledge Discovery*, v. 1, pp. 79–119, 1997.

[58] WITTEN, I. H., FRANK, E., *Data Mining: practical machine learning tools and techniques with Java implementations*. California, USA, Morgan Kaufmann, 2000.

[59] JIA, P., "Detect Masqueraders Using UNIX Command Sequences", 2000.

[60] SHENGKUI, Z., CHENGMIN, W., LI, M., "Application of artificial intelligence in earthquake forecasting", 1995.

[61] CRISTIANINI, N., SHAWE-TAYLOR, J., "An Introduction to Support Vector Machines". In: *www.support-vector.net*, Cambridge, MA - USA, Cambridge University Press, 2000.

[62] CRISTIANINI, N., SHAWE-TAYLOR, J., *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge, United Kingdom, The Cambridge University Press, 2000.

[63] YAN, M., "Eigenvalue and Eigenvector", 2002.

[64] BROWN, M., GRUNDY, W., LIN, D., *et al.*, "Support vector machine classification of microarray gene expression data", 1999.

[65] JAAKKOLA, T., DIEKHANS, M., HAUSSLER, D., "A discriminative framework for detecting remote protein homologies", 1998.

[66] HASTENRATH, S., WU, M., CHU, P., "Towards the Monitoring and Prediction of North-east Brazil Droughts", *Quart. J. Roy. Meteorology Society*, v. 110, pp. 411–425, 1984.

[67] FU, L., "Knowledge discovery based on neural networks", *Communications of the ACM*, pp. 47–50, Nov 1999.

[68] WEIGEND, A., GERSHENFELD, N., *Time Series Predition: Forecasting the Future and Understanding the Past*. USA, Addison Wesley, 1994.

Apêndice A

Algoritmo de Visualização

Algoritmo 8 Algoritmo Mostra Variação.

```
1.  Algoritmo: MOSTRA VARIAÇÃO
2.  Entrada: L, U,n_intervalos
3.  início
4.      // Uma matriz L m × n de dados de anomalia
5.      // Uma matriz U m × n de dados absolutos
6.      min ← +∞
7.      max ← -∞
8.      Laço para m iterações
9.          Laço para n iterações
10.             R[m,n] ← L[m,n] + U [m,n]
11.             se min > R[m, n] então
12.                 min ← R[m,n]
13.             se max < R[m, n] então
14.                 max ← R[m,n]
15.         fim_laco
16.     fim_laco
17.     amp ← (min−max) / n_intervalos
18.     Laço para m iterações
19.         Laço para n iterações
20.             D[m,n] ← G[m,n]−min / amp
21.         fim_laco
22.     fim_laco
23.     Plote D
24. fim
```