Maria das Graças da Silva Oliveira

Sistema para Agrupamento de Dados baseado no Comportamento Superparamagnético do Modelo de Potts

Recife

2004

Maria das Graças da Silva Oliveira

Sistema para Agrupamento de Dados baseado no Comportamento Superparamagnético do Modelo de Potts

Dissertação apresentada ao Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Orientador:
Alejandro C. Frery

Programa de Pós-graduação em Ciência da Computação
Centro de Informática
Universidade Federal de Pernambuco

Recife

2004

Tese de Mestrado sob o título *Sistema para Agrupamento de Dados baseado no Comportamento Superparamagnético do Modelo de Potts*, defendida por Maria das Graças da Silva Oliveira e aprovada em 26 de março de 2004, em Recife, Estado de Pernambuco, pela banca examinadora constituída pelos professores:

Prof. Dr. Alejandro C. Frery
Orientador
Universidade Federal de Alagoas

Prof. Dr. Fernando da Fonseca de Souza
Universidade Federal de Pernambuco

Prof. Dr. Luciano da Fontoura Costa
Universidade de São Paulo

Agradecimentos

Agradecer é uma forma de retribuir um sentimento maior do que qualquer bem material, por isso só agradecemos com sinceridade a quem realmente nos é importante e a quem gostamos muito.

É uma satisfação poder iniciar esta parte da escrita da dissertaçãoo agradecendo e homenageando todas as pessoas que contribuiram para a realização desse sonho, que não é apenas meu, mas de todos que me acompanharam desde o início e também daqueles que "pegaram o bonde andando" e que resolveram, sem motivo algum a priori, me apoiar, incentivar e ajudar.

Inicialmente, agradeço a Deus, como não podia deixar de ser, pela saúde, paz e força essencialmente necessárias.

Agradeço também à minha família por toda ajuda e paciência nos momentos mais difíceis e pela compreensão ao abandoná-los para "correr atrás desse sonho". Um muito obrigada também a todos os colegas do `Cin` que passaram a ser amigos queridos e que, sempre que possível, deixavam seus trabalhos de lado apenas para ajudar uma pessoa que eles conheciam há pouco tempo, mas que já tinham um grande carinho, que era transmitido numa brincadeira, numa conversa e até mesmo nos momentos de ajuda. Graças a essas pessoas pude transformar o `Cin` em uma segunda casa e eles em uma segunda família.

Uma eterna gratidão a uma pessoa que me ensinou muito mais do que um professor é capaz de ensinar. Ensinou coisas que apenas amigos são capazes de ensinar e me proporcionou momentos que desde já sinto saudades. Ao professor Alejandro, minha adimiração, amizade e carinho.

Por último, mas não menos importante do que os demais e sim tão importante quanto, agradeço ao meu grande amigo e namorado, companheiro de todos os momentos, Wagner Menezes, pelo empenho, carinho, dedicação e ajuda nos momentos mais difíceis.

À minha família

e ao meu amor

Wagner,

com muito carinho. . .

Resumo

O agrupamento de dados é uma das operações mais importantes na análise de informações. Esta operação consiste em, dada uma massa de informações a respeito de uma população de indivíduos, procurar grupos de elementos semelhantes entre si e diferentes dos outros. Essa técnica encontra aplicações em praticamente todas as áreas, desde análise de imagens até bioinformática. Quando o volume de dados é considerável, o problema se torna computacionalmente muito difícil.

Recentemente foi estabelecida uma analogia entre o problema de agrupamento e a procura de configurações típicas de um modelo físico, o modelo de Potts. Dado que existem algoritmos eficientes para a localização dessas configurações, como por exemplo a dinâmica de Swendsen-Wang, é possível aplicar essas técnicas para um grande volume de dados e em uma grande diversidade de situações.

Para verificar essa analogia foi desenvolvido o programa `SPC`, em linguagem `C`, pelo Professor Eytan Domany, do Departamento de Física de Sistemas Complexos, do Instituto de Ciência de Weizmann, em Israel. A função principal desse programa é a geração de agrupamentos de dados utilizando uma nova técnica de agrupamento baseada na analogia citada. Essa técnica ficou conhecida como "superparamagnética" e se baseia na procura de ocorrências de um certo modelo de Potts não-homogêneo em um estágio intermediário entre duas fases do magnetismo, a ferromagnética e a paramagnética. O programa SPC utiliza a dinâmica de Swendsen-Wang para simular os estados "típicos" do modelo de Potts.

Esta dissertação estuda essa técnica de agrupamento e apresenta proposta, construção e avaliação de um sistema amigável para sua aplicação em diversas situações. O resultado deste trabalho é uma interface amigável, desenvolvida em `IDL`, que permite tanto a especificação dos parâmetros que determinam o funcionamento do algoritmo SPC quanto a análise dos resultados por ele produzidos. Esta análise permite a visualização dos agrupamentos superparamagnéticos através de gráficos hierárquicos (dendrogramas). Esses dendrogramas oferecem ao usuário mecanismos de interação para descoberta de informações, bem como análises quantitativas (média, variância, mediana, curtose, coeficiente e assimetria, entre outras) e qualitativas (*Brushplots*) dos dados. A junção desse sistema com o programa `SPC` foi empregada com sucesso na análise de dados.

`Palavras-chaves`: Agrupamento de dados, interface, interação, dendrograma, modelo físico.

Abstract

Data clustering is one of the most important operations in data analysis. This operation consists of, given a mass of information regarding a population of individuals, looking for groups of similar elements. This technique finds applications in almost every scientific field, from image analysis to bioinformatics. When volume of data is considerable, the problem becomes computationally very difficult.

Recently, an analogy was established between the problem of grouping and the search of typical configurations of a physical model, the Potts model. Efficient algorithms for finding these configurations exist, as for example the Swendsen-Wang dynamics, so it is possible to apply this technique for a great volumes of data and in a great diversity of situations.

To illustrate this analogy the program SPC was developed, in language C, by Professor Eytan Domany (Department of Physics of Complex Systems, Institute of Science of Weizmann, Israel). The main purpose of this program is the generation of data groupings using the new technique. This technique is known as "superparamagnetic" since it is based on simulating outcomes of a non-homogeneous Potts model in an intermediate regime between the two phases of magnetism, the ferromagnetic and the paramagnetic. The SPC program uses the dynamics of Swendsen-Wang to simulate the typical states of the model of Potts.

This work consistes of studying the SPC algorithm in order to propose, to implement and to evaluate an user-friendly system for applying this technique on a diversity of fields. The result of this study is an amenable interface, developed in the IDL programming language, which allows both the specification of the required parameters for the use of the algorithm and the analysis of the results. This analysis includes dendrogram visualization and interaction, quantitative (mean, variance, median, kurtosis, skewness, among others) and qualitative (brushplots) analysis of data and groups.

The use of this system was validated with real data.

Keywords: Data clustering, interface, interaction, dendrogram, physical model.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

O agrupamento de dados é uma das operações mais importantes na análise de informações. Esta operação consiste em, a partir de um conjunto de dados que caracterizam populações de indivíduos, procurar grupos de elementos semelhantes entre si e diferentes dos outros. Alguns dos textos que tratam este problema detalhadamente são os de Duda, Hart & Stork (2001), de Fukunaga (1990), de Kaufman & Rousseeuw (1990) e de Ripley (1996).

Para ilustrar a operação de agrupamento definida acima, consideremos um problema simples: o agrupamento de oito indivíduos, onde cada um deles é caracterizado por uma medida bi-dimensional. Exemplos disto seriam a coleta de estatura e peso, de idade e estatura, entre outras, em oito indivíduos. Deseja-se saber se esses indivíduos, quando caracterizados pelas duas medidas observadas, formam mais de um "grupo natural".

A Figura 1 (página 16) mostra as oito observações na forma de pontos no plano, e é intuitivo afirmar que os indivíduos formam dois grupos distintos de objetos, os formados pelas observações rotuladas {A,B,C,D} e pelas observações rotuladas {E,F,G,H}.

Os objetos que pertecem ao mesmo grupo apresentam a característica *altura-peso* com valores aproximados, sendo assim temos a formação de dois grupos, que são chamados de *clusters* . A descoberta destes *clusters* é o principal objetivo da *análise de agrupamento*, onde objetos que pertencem ao mesmo grupo (*cluster*) são similares entre si e diferentes de objetos que pertencem a outros grupos (*clusters*).

O problema de agrupamento de objetos semelhantes é uma importante atividade humana. Uma criança, por exemplo, aprende a distinguir entre homens e mulheres, gatos e ratos, mesas e cadeiras, através do processo contínuo de melhoramento dos esquemas de classificação do subconsciente.

No século XVIII, em astronomia, Hertzsprung e Russell (ver Lang 1992) classificaram estrelas em diversas categorias baseadas em atributos como intensidade de luz e temperatura da superfície. Uma vez que eles realizaram classificações, isto implica que não apenas agrupamentos foram realizados, mas estes foram ainda rotulados e os grupos resultantes desse processo de agrupamento e rotulação (processo também conhecido como classificação), foram chamados de categorias. Hertzsprung e Russell observaram ainda que todas as estrelas têm propriedades que são facilmente

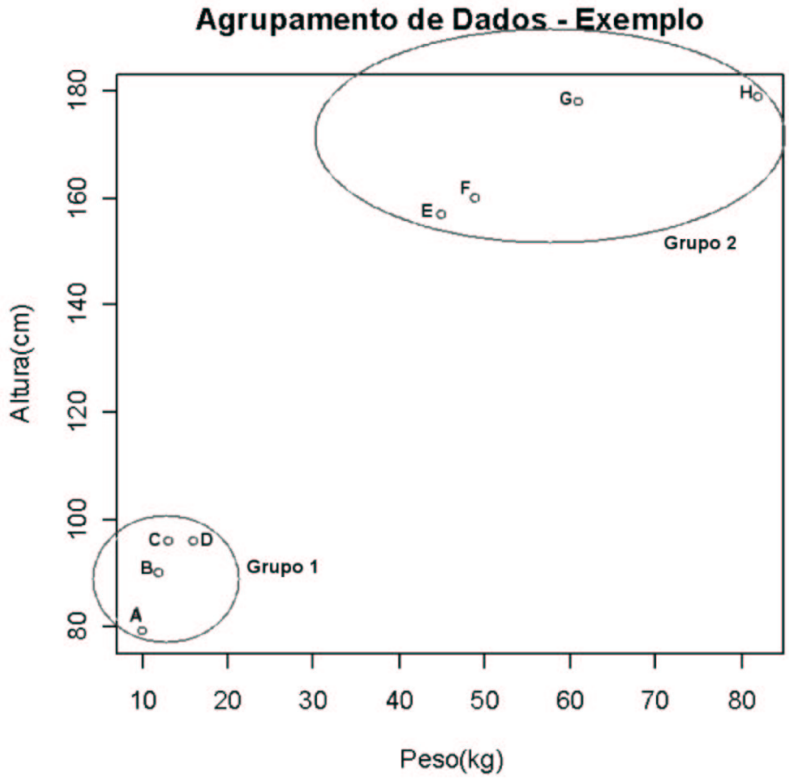


Figura 1: Agrupamento de Dados com Oito objetos.

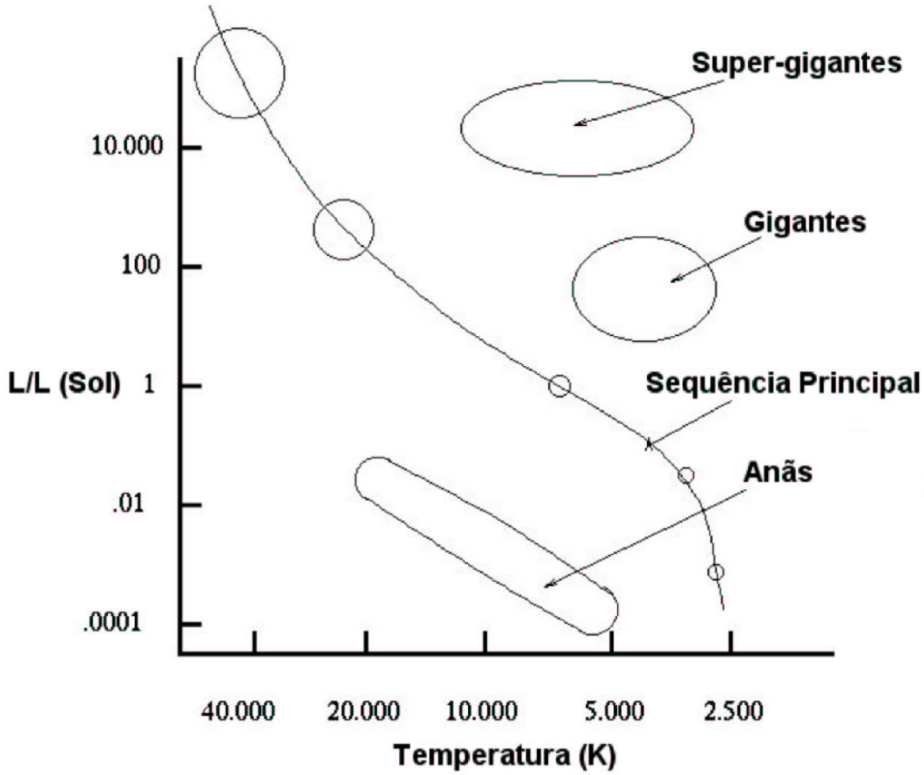


Figura 2: Diagrama HR.

visíveis, tais como: brilo e cor. Algumas estrelas apresentam a cor avermelhada, algumas outras apresentam a cor amarelada (como o Sol, por exemplo), e outras apresentam a cor azulada ou ainda azul claro (como é o caso da estrela *Sirius*). É sabido que a cor tem uma relação direta com a temperatura, ou seja, objetos estelares que apresentam uma cor avermelhada são relativamente frios, se comparados a objetos estelares que apresentam uma cor amarelada, e estes, por sua vez, são mais frios do que os que apresentam uma cor azulada.

A Figura 2 representa o sistema solar e mostra um gráfico da temperatura versus o brilho, onde a temperatura é incrementada da direita para a esquerda por razões históricas. Esse gráfico é conhecido como diagrama HR ou ainda diagrama Hertzsprung-Russell. Estes diagramas foram originalmente baseados na cor, sendo a orientação da esquerda para a direita representada pelas cores de azul a vermelho.

Esse diagrama é importante porque ele é a interface gráfica entre observação e teoria na evolução estelar, ou seja, ele representa a temperatura da superfície e o brilho de uma estrela baseado em propriedades fisicas da estrutura estelar. Nesse diagrama ainda podemos verificar que deriva uma cor baseada na temperaturada, ou ainda pode-se observar que o brilho e a cor derivam da temperatura da superfície da cor observada.

Na Figura 2 pode-se observar ainda que o diagrama não é aleatório, ou seja, se

várias estrelas forem observadas e graficamente representadas, o que se tem são pontos que não estão livremente dispersos, como se poderia pensar. Ao invés disso, um padrão específico emerge. As estrelas tendem a se concentrar ao longo de uma curva que vai decaindo através do gráfico, da esqueda para a direita, que é chamada de sequência principal. Ela é chamada assim porque nela Hertzsprung e Russell encontraram a maioria das estrelas. Existem massas de estrelas de brilho avermelhado na parte superior direita do diagrama e também estrelas opacas azuladas abaixo da sequência principal. As primeiras são chamadas gigantes vermelhas, porque elas são muito grandes para serem simultaneamente frias e terem brilho. Da mesma forma, as estrelas opacas azuladas que estão localizadas abaixo da sequência principal são muito pequenas para serem simultaneamente quentes e opacas.

Outros aglomerados de estrelas ainda podem ser observados na Figura 2 (página 17), e a partir dessa observação é facilmente verificável que Hertzsprung e Russell utilizaram os conceitos de agrupamento e classificação em suas pesquisas e dessa forma utilizaram clusters para representar seus agrupamentos de estrelas.

Em ciências sociais, frequentemente pessoas são classificadas de acordo com seu comportamento e preferências. Muitos outros exemplos ainda podem ser considerados, tal como em geografia, onde agrupamento de regiões é uma atividade interessante, ou ainda em medicina, onde o agrupamento de incidências de tipos específicos de câncer é um fator importante.

No passado, o agrupamento de dados era feito de maneira subjetiva, isto é, através da confiança na percepção e julgamento de um pesquisador. Apesar do sistema olho-cérebro humano ter evoluído ao longo de milênios, ele está muito bem adaptado apenas para efetuar o agrupamento de indivíduos caracterizados por observações de até dimensão três, e sempre que os grupos não sejam em número excessivo e que estejam separados espacialmente. Em casos com mais dimensões, com muitos indivíduos ou com grupos pouco separados (situações bastante comuns na prãtica) essa técnica não é aplicável.

Observando a necessidade de classificação em situações realistas, a ciência moderna tem notado um considerável aumento nos **procedimentos de agrupamento automáticos**.

Nos últimos anos, uma grande quantidade de algoritmos e programas de computador têm sido desenvolvidos para o agrupamento de dados. Isto deve-se à diversidade de áreas onde as técnicas de agrupamento são necessárias. Essa diversidade de áreas gera a necessidade de diferentes tipos de algoritmos que se adequem aos diferentes tipos de aplicações.

Atualmente existem sistemas aptos a agrupar dados de forma simples e rápida, porém a qualidade dos resultados nem sempre é garantida por resultados teóricos de validade geral. A Ciência da Computação, mais especificamente as áreas de Inteligência Artificial e Aprendizagem de Máquina, aliada à Estatística e à Probabilidade,

mais especificamente à área de Teoria Estatística de Reconhecimento de Padrões (ver Devroye, Györfi & Lugosi 1996), formam um conjunto que pode ajudar a melhorar esses resultados.

Existem diversas ferramentas voltadas para a integração das ciências da Computação e da Estatística; são programas que permitem classificar dados através da execução de simples comandos e da passagem de parâmetros, por exemplo o conjunto de dados a ser classificado e suas variáveis.

Além do advento dos programas, outro fator que contribui para facilitar a solução do problema de agrupamento de dados é o hardware de computador. Na maioria das vezes o conjunto de dados e variáveis que se deseja classificar é muito grande.

Sendo assim, computadores com poucos recursos de memória e processador podem retardar ou dificultar o processo de agrupamento, aumentando o tempo requerido para efetuá-lo. Dependendo do tamanho do conjunto de dados, um hardware apropriado pode ser escolhido e dessa forma o problema do tempo de classificação pode ser consideravelmente reduzido.

É bom notar que, mesmo utilizando-se um hardware que gere um tempo de atraso na classificação grande, ainda é claramente mais vantajoso usar um computador para classificar dados do que o sistema olho-cérebro humano com seus milênios de evolução quando o problema é de tamanho e/ou complexidade pelo menos moderados.

A próxima seção descreve brevemente alguns dos principais algoritmos de agrupamento de dados existentes.

1.1 Algoritmos de agrupamento de dados

Para tentar solucionar o problema do agrupamento de dados temos diversos algoritmos que, dependendo do conjunto de dados, apresentam um grau de eficiência relativamente bom, ou seja, com base em análises e comparações com os resultados de outros algoritmos pode-se verificar se o algoritmo utilizado no agrupamento de um determinado conjunto de dados apresenta resultados compatíveis com a realidade destes dados. Escolher, entre esses diversos algoritmos, o melhor para agrupar um determinado conjunto de dados é uma tarefa difícil, uma vez que essa escolha deve estar baseada em algumas características dos dados, tal como o tipo de dado a ser agrupado. Uma possível solução para esse dilema é testar vários desses algoritmos, analisando e comparando os resultados obtidos através de gráficos ou ferramentas que permitam uma fácil visualização desses resultados.

Um ponto chave nessa solução é a interpretação dos resultados, uma vez que ela deve ser baseada em uma percepção do significado dos dados originais em conjunto com alguma experiência sobre os algoritmos usados.

Existem muitos algoritmos de agrupamento na literatura e seria impraticável descrever todos eles neste documento, mas podemos considerar cinco categorias e algumas de suas principais técnicas. São elas:

- Métodos por partição
- Métodos hierárquicos
- Métodos baseados em densidade
- Métodos baseados em grade
- Métodos baseados em modelos

Os *métodos por partição*, ou métodos não hierárquicos, procuram dividir n objetos em k grupos, ou seja, dado um conjunto de dados e um critério de particionamento, esses dados são separados em grupos com base na similaridade (dissimilaridade) que esses dados têm uns com os outros, de acordo com o critério especificado. Alguns exemplos desses métodos são: PAM (*Partitioning Around Medoids*), CLARA (*Clustering Large Applications*) e K-médias, todos descritos no Capítulo 2, página 27.

Já os *métodos hierárquicos* são realizados por sucessivas fusões ou por sucessivas divisões, ou seja, criam uma decomposição do conjunto de dados, usando para isso algum critério. Existem dois tipos de técnicas hierárquicas: a aglomerativa e a divisiva. A ténica aglomerativa inicia com cada observação formando um grupo separado e em seguida observações próximas são sucessivamente agrupadas. Já a técnica divisiva realiza o processo inverso da técnica aglomerativa, ou seja, o grupo inicial é composto por todas as observações do conjunto de dados e em seguida esse único grupo é dividido sucessivamente em dois outros menores até que cada um contenha exatamente uma única observação. Alguns exemplos desses métodos são: Algoritmo de Junção Simples e Algoritmo de Junção por Centróide.

Nos *métodos baseados em densidade* a idéia principal é que para cada ponto de um grupo, sua vinzinhança dentro de um determinado raio tem que conter um número mínimo de pontos, isto é, a densidade na vizinhaça não pode exceder um derterminado limite. Nesses métodos normalmente são encontrados grupos com formatos arbitrários, uma vez que o formato de uma vizinhança é determinado pela escolha de um função de distância para dois pontos, p e q, definida por $dist(p, q)$. Por exemplo, quando a disntância Manhattan no espaço $2D$ é escolhida, o formato da vinzinhança é retangular. Outra maneira de entender a definição é observando a Figura 3 (página 21), onde podem ser vistos conjuntos de pontos. Nessa figura, grupos de pontos e pontos dispersos são facilmente detectados sem que haja ambiguidade. A principal razão para que esses grupos sejam reconhecidos tão facilmente é que dentro de cada um existe uma densidade típica de pontos que é consideravelmente maior do que fora deles. Além disso, a densidade dentro das áreas dispersas é menor do que a densidade dentro de qualquer grupo.

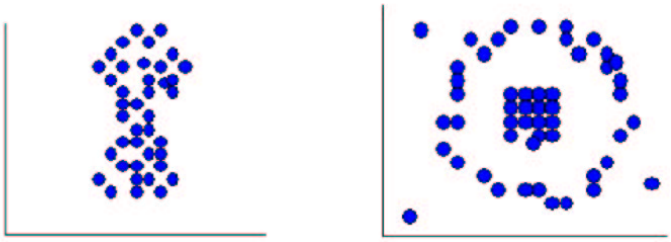


Figura 3: Agrupamentos baseados em densidade.

Os *métodos baseados em grade* manipulam dados espaciais e quantizam o espaço em um número finito de células e então realizam todas as operações no espaço quantizado.

Um algoritmo que representa esses métodos é o STING (*Statistical Information Grid-based method*), ele divide a área espacial em células retangulares usando uma estrutura hierárquica. O STING passa através do conjunto de dados e computa parâmetros estatísticos (como por exemplo média, variância, valor mínimo, valor máximo e o tipo de distribuição) de cada característica numérica dos objetos dentro das células. Em seguida ele gera uma estrutura hierárquica das células de forma a representar informações sobre os agrupamento em diferentes níveis.

Nos *métodos baseados em modelo* os dados são visualizados como amostras de uma população, que consiste de um número de subpopulações (grupos ou clusters). Cada grupo pode ser descrito por uma distribuição de probabilidade. Nesses métodos a inferência estatística é utilizada para descobrir, por exemplo, a probabilidade de um objeto pertencer a um determinado grupo ou então descobrir o número mais provável de grupos presentes na população.

Os métodos por partição e os métodos hierárquicos são os mais conhecidos e utilizados e também os que apresentam maior número de algoritmos. Eles serão descritos em maiores detalhes no Capítulo 2 (página 27), bem como suas principais ténicas de agrupamento de dados, citadas nesta seção.

1.2 Agrupamento Superparamagnético

O problema do agrupamento de dados descrito no início do capítulo 1 (página 15) levou ao desenvolvimento de técnicas e implementações que sempre tiveram como objetivo principal a corretude dos resultados obtidos. Por uma técnica 'correta' entende-se, neste contexto, aquela que produz como saída os grupos 'naturais' presentes nos dados de entrada.

Descobrir a qual grupo um determinado indivíduo pertence não é uma tarefa simples, uma vez que existem diversas técnicas para agrupamento (ver seção 1.1,

página 19) e cada uma delas apresenta resultados potencialmente diferentes.

Algumas das técnicas discutidas exigem um conhecimento mais profundo do conjunto de dados a ser agrupado, isto por causa da quantidade de parâmetros que algumas técnicas necessitam para que o agrupamento possa ser realizado satisfatoriamente. Nem sempre poderemos fornecer todos esses parâmetros, logo, o processo de agrupamento poderá ficar comprometido e os resultados obtidos poderão não representar a realidade do conjunto de dados.

Descobrir qual é a melhor técnica equivale a perguntar se os agrupamentos resultantes estão realmente corretos, pergunta que nem sempre sabemos responder. As diversas abordagens mais formais ao problema de agrupamento têm, dentre outros objetivos, a procura de métricas capazes de medir a 'qualidade' de agrupamentos de dados.

Com base neste problema foi desenvolvido um novo método para agrupamento de dados baseado não apenas em características estatísticas e matemáticas como são os métodos clássicos, mas também em propriedades fisicas. Esse método é conhecido como **Agrupamento Superparamagnético** e foi desenvolvido pelos professores Domany, Blatt, Gdalyahu & Weinshall (1999) do *Department of Physics of Complex Systems*, *Weizmann Institute of Science*, Israel.

A técnica de agrupamento superparamagnético é baseada em um modelo fisico-magnético, o modelo de *Potts não-homogêneo*, que procura associar configurações deste modelo, baseado na temperatura, a grupos de dados. Essa associação é a chave do método de agrupamento superparamagnético e será descrita no capítulo 3 (página 50).

Para verificar a eficiência do método superparamagnético foi desenvolvido, pelo professor Eytan Domany (Domany et al. 1999), um programa que realiza o agrupamento superparamagnético de dados. Esse programa foi desenvolvido em linguagem `C` e recebe como parâmetros de entrada um *arquivo de dados* e um *arquivo contendo informações a respeito desses dados*, como por exemplo número de indivíduos contidos no arquivo de dados, temperatura a partir da qual os agrupamentos começam a ser realizados, temperatura final dos agrupamentos, entre outros que serão melhor descritos na seção 5.1 (página 76).

A passagem desses parâmetros de entrada é feita pelo próprio usuário que, inicialmente, tem a inconveniência de gerá-los em algum software matemático a parte e em seguida passá-los ao programa atráves de linha de comando, o que não é uma tarefa tão simples para a maioria dos usuários que tenham a pretensão de utilizar o programa, uma vez que grande parte deles não pertencerá à area de Computação.

Como mencionado no parágrafo anterior, o programa `SPC` descrito não apresenta interação alguma com o usuário; a única maneira de interagir com ele é através da passagem de parâmetros através de um arquivo de configuração antes de sua

execução. Os arquivos resultantes dos agrupamentos são de difícil entendimento por parte de qualquer usuário, o que restringe o uso do novo método apenas aos desenvolverdores ou especialistas no assunto.

Procurando amenizar os problemas de interação acima apontados, foi desenvolvida uma interface que realiza a comunicação entre o usuário e o SPC, proporcionando ao usuário não só a passagem dos parâmetros de entrada para o programa, mas também a criação dos mesmos.

Com esta proposta, o usuário pode criar os parâmetros de entrada dentro da própria interface e escolher as informações apropriadas para a realização dos agrupamentos. Além disso, os arquivos resultantes dos agrupamentos foram substituídos por apenas dois, os quais contêm informações que são utilizadas para representar graficamente os agrupamentos. Essas representações gráficas estão sob a forma de **dendrogramas**, gráficos em forma de árvore. A interação do usuário com esses dendrogramas é total e permite ao usuário visualizar os clusters individualmente, além de outras informações como histogramas das variáveis associadas aos agrupamentos.

Enquanto as tarefas acima eram desenvolvidas, foram feitas modificações substanciais ao código original com o objetivo de aperfeiçoar as funcionalidades já existentes e ajustá-las às necessidades da interface desenvolvida, ou seja, era preciso "enxugar" o aglomerado de informações que eram geradas e armazenadas pelo programa original e, além disso, gerar outros dados necessários às novas funções da interface desenvolvida.

Uma descrição mais detalhada da interface desenvolvida é descrita na seção 5.5 (página 85). Esta interface foi implementada em linguagem IDL (*Interactive Data Language*) e os algoritmos matemáticos utilizados nela foram baseados nos algoritmos existentes na plataforma R, ambas, linguagem e plataforma, especificadas na próxima seção.

1.3 Plataformas de trabalho: IDL e R

1.3.1 Plataforma IDL

O IDL (*Interactive Data Language*) é um ambiente de trabalho completo para análise e visualização interativa de dados de qualquer natureza. Ele integra o poder de uma linguagem de quarta geração com ferramentas avançadas de análise matemático-estatístico, representação gráfica e visualização em forma de funções e rotinas facilmente acessíveis ao usuário (*IDL: Interactive Data Language* 2003). Com o IDL se tem uma maior flexibilidade, podendo os dados serem processados inteiramente no ambiente do IDL, agregar também rotinas Fortran ou C, ou ainda, chamar funções do IDL num programa Fortran ou C , com livre escolha da plataforma de trabalho.

1.3.2 Plataforma R

R é um conjunto integrado de facilidades de software para manipulação de dados, cálculo e exibição gráfica (ver Venables & Smith 2001). Entre outras coisas ele apresenta:

- uma manipulação efetiva de dados e facilidade de armazenamento;
- um conjunto de operadores para cáluculos com arrays, em particular matrizes;
- uma coleção grande e integrada de ferramentas intermediárias para análise de dados;
- facilidades gráficas para análise de dados e
- uma linguagem de programação simples que inclui: comandos condicionais, loops, funções recursivas definidas pelo usuário e facilidades de entrada/saída.

1.4 Objetivos do trabalho

A proposta inicial era o desenvolvimento de uma interface gráfica amigável que recebesse como entrada principal um conjunto de dados. A partir deste conjunto de dados seriam gerados dois arquivos, um contendo sempre uma matriz de dissimilaridade e o outro informações a respeito dessa matriz. Esses dois arquivos são os parâmetros de entrada do programa SPC, como mencionado na Seção 1.2, página 21. Além da geração desses arquivos, um dos objetivos da interface é a comunicação com o próprio SPC, ou seja, visualmente o usuário poderá invocar o programa sem a necessidade de usar linha de comando.

Após a realização dos agrupamentos, outro objetivo importante é permitir ao usuário uma fácil compreensão dos resultados, através de gráficos e funções, tal como a interação do usuário com alguns desses gráficos. A interface implementa os gráficos conhecidos como **dendrogramas**, que se apresentam sob a forma de árvore hierárquica e permitem a interação do usuário com os agrupamentos realizados. Dessa forma, os problemas mencionados na Seção 1.2 (página 21) seriam eliminados, uma vez que o ambiente gráfico permite uma fácil compreensão das funções fornecidas pelo programa e baseadas nas técnicas utilizadas.

A interface foi desenvolvida em IDL e recebe como entrada não só o conjunto de dados, para gerar a matriz de dissimilaridade, mas também algumas das informações contidas no segundo arquivo parâmetro, necessárias à execução do programa.

Além da criação de gráficos, como os dendrogramas, também está incluída nos resultados a geração de uma *análise multivariada de dados*, *análise descritiva* e outros recursos, como *técnicas de busca e representação de indivíduos*, necessários a uma melhor apresentação dos resultados aos usuários do sistema.

1.5 Principais contribuições

Este projeto tem como principais objetivos o desenvolvimento de uma ferramenta que disponibilize ao seu usuário recursos novos e eficientes para geração, manipulação e visualização de agrupamentos de dados; e promover a integração e utilização da técnica SPC, além de proporcionar uma melhor apresentação e compreensão dos resultados obtidos a partir dela.

A partir dos objetivos expostos no parágrafo anterior, pode-se verificar que as principais contribuições deixadas por este trabalho são a utilização e integração de uma nova técnica de agrupamento de dados a um sistema amigável, que procura resolver os problemas encontrados em outras ferramentas e melhorar os recursos considerados úteis à compreensão dos resultados.

Outra contribuição igualmente importante é a disseminação dessa nova técnica de agrupamento, que se mostrou eficiente quando aplicada a conjuntos de dados grandes e complexos.

1.6 Estrutura do documento

Este trabalho procura esclarecer não só os objetivos do projeto, mas também os conceitos utilizados para atingí-los. Dessa maneira, o documento apresentará as seguintes informações:

Capítulo 1: Este capítulo faz uma introdução dos principais conceitos, objetivos e ferramentas que serão discutidos ao longo de todo o documento.

Capítulo 2: A base de todo o trabalho é discutida neste capítulo que procura definir o problema de agrupamento de dados e exibir exemplos das diversas técnicas de agrupamentos existentes.

Capítulo 3: Este capítulo descreverá a técnica de Agrupamento Superparamagnético através de seções que tratam dos conceitos físicos usados na construção do seu algoritmo, tais como: o modelo de Potts e o método de Swendsen-Wang.

Capítulo 4: A nova técnica de agrupamento de dados é eficiente, principalmente, quando aplicada a grandes e complexos conjuntos de dados. Para demonstrar tal situação são exibidos neste capítulo o problema em agrupar esses tipos de dados e alguns trabalhos relacionados com agrupamentos de dados em geral.

Capítulo 5: Este capítulo descreverá o uso da nova técnica de agrupamento, bem como alterações realizados em sua implementação. Além disso, a ferramenta desenvolvida neste projeto é detalhada através de uma análise de requisitos e de sua implementação.

Capítulo 6: Neste capítulo será analisado um conjunto de dados, que será utilizado nos testes da ferramenta desenvolvida. Os resultados obtidos para esse conjunto de teste também serão exibidos neste capítulo.

Capítulo 7: As conclusões de todo o projeto e trabalhos futuros são descritos neste capítulo.

Apêndice A: Este apêndice exibe os principais códigos fontes gerados neste trabalho.

2 O problema do agrupamento

O capítulo anterior descreveu o problema de agrupamento de dados como uma das operações mais importantes na análise de informações e definiu que esta operação consiste em, dado um grande conjunto de informações a respeito de uma população de indivíduos, procurar grupos de elementos (indivíduos) semelhantes entre si e diferentes dos outros. Essa operação de agrupamento é ilustrada no exemplo da Figura 1 (página 16), no capítulo anterior. Alguns dos textos que tratam este problema detalhadamente são Devroye et al. (1996), Duda et al. (2001), Fukunaga (1990), Kaufman & Rousseeuw (1990) e Ripley (1996).

O problema geral de agrupamento de dados será formalizado adiante e essa formalização será adotada ao longo de todo o documento.

Considerando $\Omega = \{\omega_1, \ldots, \omega_n\}$ um conjunto finito de indivíduos. A cada indivíduo, $\omega \in \Omega$, estará associado um vetor de atributos $\boldsymbol{x}_\omega$ de dimensão $p \geq 1$, $p \in \mathbb{N}$. Esses vetores de atributos $\boldsymbol{x}_\omega = (x_{\omega,1}, \ldots, x_{\omega,p})$, $\omega \in \Omega$, serão as entradas primárias do problema de agrupamento. Como exposto na Seção 2.1.1 (página 28), há uma vasta literatura a respeito da utilização simultânea dos valores reais dos p atributos para fazer o agrupamento dos n indivíduos (ver Kaufman & Rousseeuw 1990), mas neste trabalho o foco será colocado em técnicas que empregam medidas de dissimilaridade multivariadas.

Uma vez definida a medida de dissimilaridade de interesse, o problema de agrupamento dos dados consiste em procurar partições de Ω de $g \in \mathbb{N}$ elementos não vazios chamados 'grupos', da forma $\mathcal{P} = (\Omega_1, \ldots, \Omega_g)$, onde $\Omega_i \cap \Omega_j = \emptyset$ se $i \neq j$ e $\cup\Omega_i = \Omega$, para as quais são satisfeitas simultaneamente as condições:

1. A variação das dissimilaridades dentro dos grupos é mínima, e
2. A variação das dissimilaridades entre grupos é máxima.

A primeira condição tenta preservar a propriedade de coesão dos indivíduos de um grupo, ou seja, quanto menor a variação das dissimilaridades, mais similares são os indivíduos pertencentes a um mesmo grupo.

A segunda condição está associada à propriedade de isolamento entre grupos, isto é, quanto maior for a variação das dissimilaridades entre grupos, mais distantes eles estarão.

É possível formalizar de diversas maneiras os critérios que os grupos devem satisfazer. Dependendo do critério de agrupamento utilizado na formação de novos grupos, mais ou menos aglomerados podem ser formados. Caso esse critério seja 'muito frouxo', as diferenças entre os indivíduos não são percebidas e nesse caso apenas um grupo contendo todos os indivíduos é formado. Já se o critério utilizado for 'muito rigoroso', as semelhanças entre os indivíduos não são percebidas e assim cada um deles estaria em um único grupo. Bons algoritmos de agrupamento devem oferecer ao usuário o controle do rigor com que o grupos serão compostos.

Para solucionar o problema geral de agrupamento surgem diversos métodos e seus algoritmos. Dependendo das características do conjunto de dados a ser agrupado, vários desses algoritmos podem ser eficientes para um mesmo conjunto de dados, ou todos eles podem apresentar resultados insatisfatórios.

Para realizar uma boa escolha é preciso conhecer o algoritmo que se está utilizando, além de analisar e comparar, com outros algoritmos, os resultados obtidos. Com base nessa necessidade, este capítulo define os principais métodos de agrupamentos de dados e descreve alguns de seus algoritmos, são eles:

- Métodos de agrupamento por partição, que são os mais desenvolvidos e estudados, por apresentar única resposta a cada saída, tornando-os métodos diretos e práticos de se trabalharem;
- Métodos de agrupamento hierárquico, que apresentam várias soluções, constituindo uma árvore hierárquica. Esta variedade de respostas pode se tornar uma ferramenta poderosa na análise de informações.

2.1 Definições Básicas

2.1.1 Medidas de Proximidade

O esforço para produzir uma simples estrutura de agrupamento a partir de um conjunto de dados complexos requer necessariamente uma medida de 'proximidade', que é usualmente indicada por alguma ordem de distância. Esta medida define o quão próximos estão os dados do conjunto e pode ser classificada em duas categorias: similaridade ou dissimilaridade. A escolha de uma medida de proximidade envolve um pouco de subjetividade, mas algumas considerações importantes incluem: a natureza das variáveis (discreta, contínua, binária), escalas de medida (nominal, ordinal, intervalar, proporção) e o conhecimento do assunto.

A maioria dos métodos de agrupamento assumem que as relações em um conjunto de n objetos são descritas através de uma matriz de tamanho $n \times n$ contendo uma medida de similaridade s_{ij} ou dissimilaridade d_{ij} entre o i-ésimo e j-ésimo objeto para cada par de objetos (i, j), onde $i = j = 1, \ldots, n$. Assim, pode-se dizer, que

dois indivíduos estão 'próximos' quando sua dissimilaridade d_{ij} é pequena, ou sua similaridade s_{ij} é grande.

Existe uma grande diversidade de medidas de proximidade, mas nossa atenção será restrita a medidas de dissimilaridade. Como estas medidas definem objetos próximos quando seus valores são mínimos, elas são associadas às métricas de distância, cujas propriedades, entre pares de objetos de d_{ij}, satisfazem as seguintes condições:

1. a função é não negativa: $d_{ij} \geq 0$;
2. a função é nula apenas quando avaliada no mesmo elemento: $d_{ii} = 0$;
3. a função é simétrica: $d_{ij} = d_{ji}$.

Uma matriz de dissimilaridade $\mathcal{D} = (d_{ij})_{1 \leq i,j \leq n}$ é considerada uma métrica se satisfaz a desigualdade triangular, $d_{ij} \leq d_{ik} + d_{kj}$, para todos os grupos de três indivíduos $1 \leq i, j, k \leq n$.

Uma variedade de medidas já foram propostas para obter uma matriz de dissimilaridade de um conjunto de observações multivariadas contínuas. As medidas mais comuns são: a distância Euclidiana, a distância de Manhattan, distância de Minkowski (que generaliza as duas anteriores) e a distância de Canberra. Em Gower (1985) e em Gower & Legendre (1986), encontram-se maiores informações sobre outras medidas multivariadas contínuas.

A distância Euclidiana é a medida de distância mais comumente utilizada, e pode ser descrita da seguinte forma,

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2},$$

onde x_{ik} e x_{jk} são, respectivamente, o k-ésimo valor da variável para os indivíduos i e j, no espaço p-dimensional. Esta medida apresenta uma propriedade interessante: d_{ij} pode ser interpretado como a distância física entre dois pontos $X_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ e $X_j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ no espaço Euclidiano de dimensão p.

A distância Manhattan,

$$d_{ij} = \sum_{k=1}^{p} |x_{ik} - x_{jk}|,$$

descreve uma configuração retilínea, ou seja, todas as distâncias serão medidas em linha reta (Larson & Sadiq 1983).

Uma outra medida de distância é a métrica de Minkowski,

$$d_{ij} = \left(\sum_{k=1}^{p} |x_{ik} - x_{jk}|^r \right)^{1/r},$$

com $r \geq 1$. Pode-se perceber que as distâncias Euclidiana e Manhattan são casos especiais desta métrica para $r = 2$ e $r = 1$, respectivamente.

Finalmente, a distância de Canberra, que é dada por

$$d_{ij} = \begin{cases} 0 & x_{ik} = x_{jk} = 0 \\ \sum_{k=1}^{p} \frac{|x_{ik} - x_{jk}|}{(|x_{ik}| + |x_{jk}|)} & x_{ik} \neq 0 \quad \text{ou} \quad x_{jk} \neq 0 \end{cases}$$

é sensível a pequenas mudanças próximas de $x_{ik} = x_{jk} = 0$ (Lance & Willians 1966). Ela é frequentemente vista como uma generalização de uma medida de dissimilaridade para dados binários (ver Everitt, Landau & Leese 2001). Dessa forma, a mesma pode ser padronizada, dividindo o seu resultado pelo número de variáveis, p, para assegurar que o coeficiente de dissimilaridade permaneça no intervalo $[0, 1]$.

Vale ressaltar que existem medidas para outros tipo de variáveis, tais como: variáveis binárias, variáveis nominais e ordinais e variáveis mistas, podendo estas serem encontradas em Everitt et al. (2001), Kaufman & Rousseeuw (1990), Gower & Legendre (1986) e Johnson & Wichern (1992).

2.1.2 Dendrogramas

Agrupamentos hierárquicos produzidos através de rotas aglomerativas ou divisivas podem ser representados por um diagrama bidimensional conhecido por *dendrograma*, que ilustra as fusões ou divisões que acontecem em cada estágio da análise. O dendrograma, diagrama na forma de árvore, é uma representação matemática e ilustrativa do procedimento de agrupamento completo, como define Everitt et al. (2001) (ver Figura 4, página 31).

Os *nós* de um dendrograma representam grupos e os tamanhos das hastes ou *alturas* representam as distâncias nas quais os grupos são formados. As hastes podem ser desenhadas de forma que elas não se prolonguem para a linha zero do diagrama, para indicar a ordem na qual os objetos inicialmente se agrupam. A maioria dos dendrogramas tem duas bordas emanando de cada nó (árvores binárias). Essa disposição dos nós e das hastes é a *topologia* da árvore. Os nomes dos objetos ligados a cada nó terminal são conhecidos como *rótulos*. Os nós internos não são usualmente rotulados.

Existem diversas formas de representação de dendrogramas, como pode ser visto na Figura 5 (página 32). Os formatos mais comuns são os mostrados nas representações 5(a) (página 32) e 5(b) e suas versões rotacionadas de 90 ou 180 graus. A Figura 5(c) mostra um dendrograma justificado à direita e a Figura 5(d) mostra uma *stick tree*, sendo os grupos indicados pela extensão de uma linha horizontal.

Mesmo após a escolha do formato do dendrograma, a representação dos agrupamentos hierárquicos ainda apresenta uma certa indeterminação, uma vez que existem 2^{n-1} representações equivalentes de uma mesma árvore binária, dependendo da

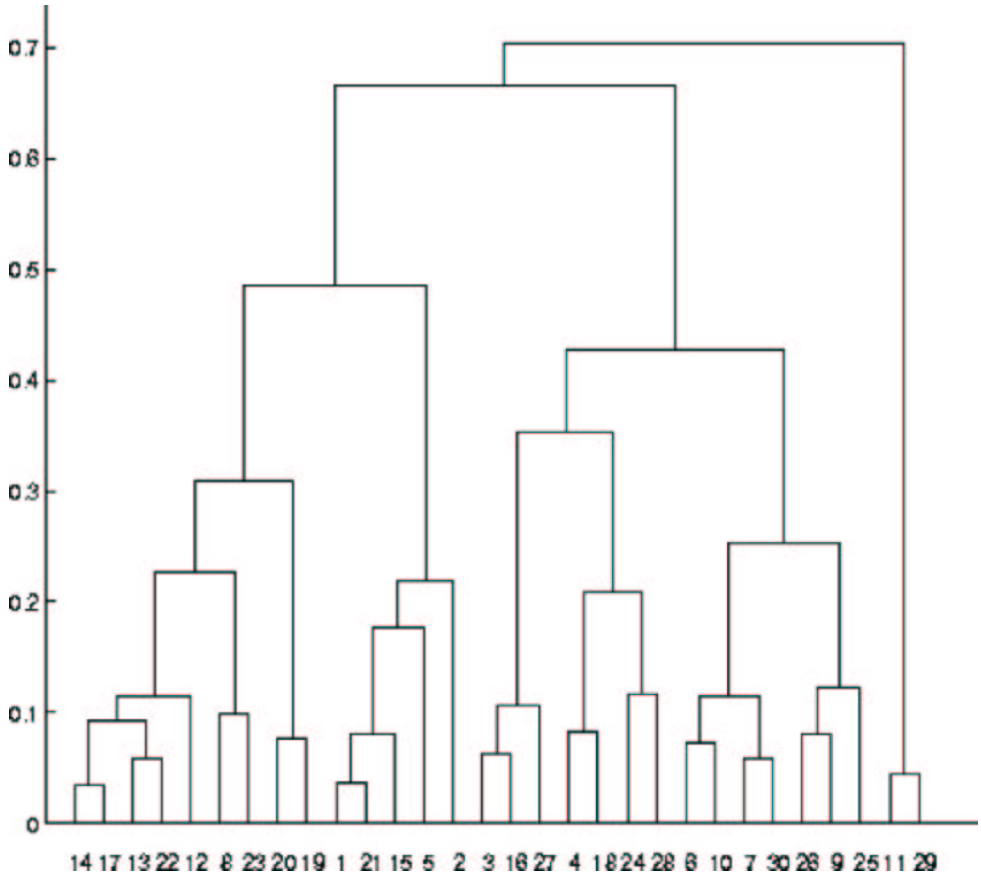


Figura 4: Ilustração de um dendrograma.

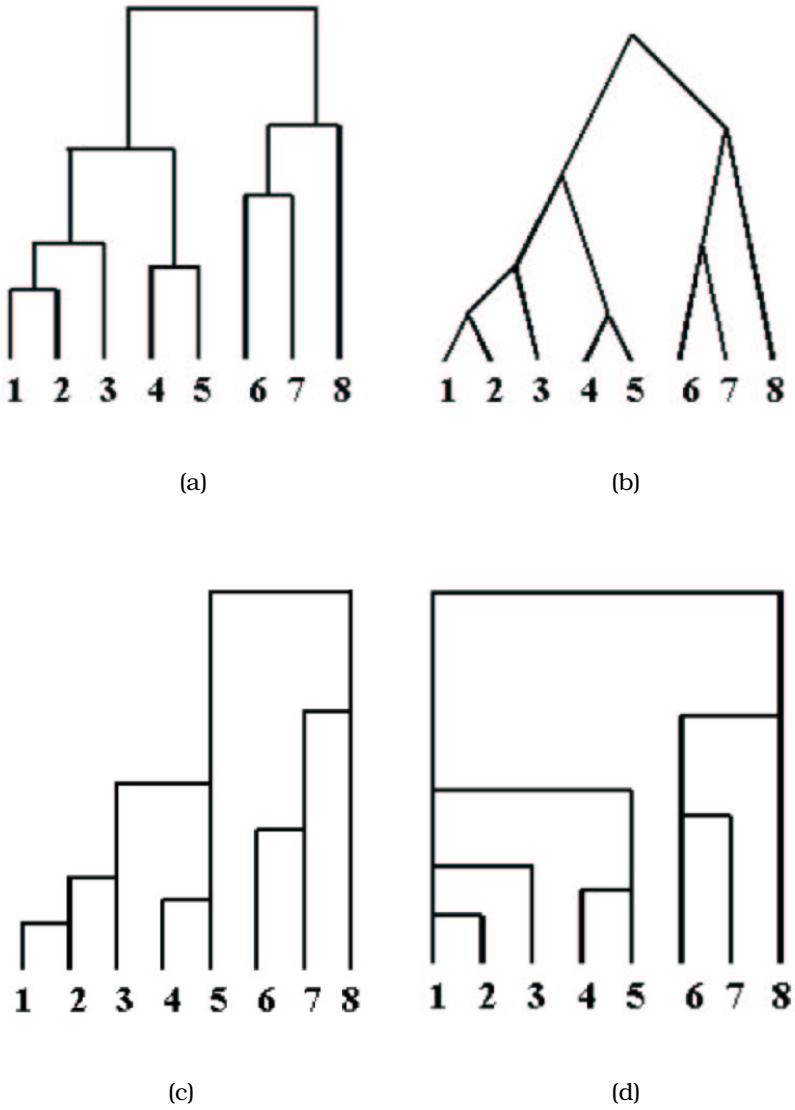


Figura 5: Quatro formatos de dendrograma para representar o mesmo agrupamento.

ordem na qual os nós são exibidos.

Diferentes tipos de representação podem ainda ser encontradas na literatura, (Everitt et al. 2001) e (Gordon 1999), tais como *pirâmide* e *árvore aditiva*, entre outras.

2.2 Agrupamento por partição

Este método de agrupamento tem como objetivo a construção de uma partição $\mathcal{P} = \{\Omega_1, \ldots, \Omega_g\}$ com cada Ω_i representando um grupo, onde todos eles devem apresentar as seguintes características:

- Cada grupo deverá conter no mínimo um objeto ($\nexists \Omega_i = \emptyset$)
- Cada objeto deverá pertencer exatamente a um grupo ($\Omega_i \cap \Omega_j = \emptyset$ se $i \neq j$).

Os métodos de agrupamento por partição tipicamente começam o processamento com base em uma partição inicial do conjunto de dados a ser agrupado, e então usam uma estratégia de controle iterativo para otimizar uma função objetivo. Cada grupo pode ser representado, por exemplo, pelo centro de gravidade do grupo (algoritmos baseado em k-médias) ou por um dos objetos do grupo localizado próximo do seu centro (algoritmos baseados em k-medóides).

A otimização da função objetivo é reportada como um 'critério de agrupamento', pois fornece uma idéia intuitiva dos grupos através de uma fórmula matemática (ver Peña, Lozano & Larrañaga 1999). O valor da função usualmente depende da partição atual do conjunto de dados $\mathcal{P}_i = \{\Omega_1, \ldots, \Omega_g\}$, isto é:

$$F: P_g(\Omega) \rightarrow \mathbb{R},$$

onde $P_g(\Omega)$ é o conjunto de todas as partições da base de dados $\Omega = \{\omega_1, \ldots, \omega_n\}$ em g grupos não vazios. Cada ω_i dos n objetos da base de dados Ω é um vetor p-dimensional.

As seções a seguir explicam alguns dos métodos de partição mais utilizados.

2.2.1 K-Médias

O Algoritmo de k-médias (k-Means) foi apresentado originalmente por McQueen (1967), e utiliza os centróides de cada grupo como seus pontos representantes. Em uma grande escala de dados estes algoritmos são mais eficientes do que os algoritmos hierárquicos tradicionais, porém seu desempenho depende da seleção inicial dos centróides (Bradley & Fayyad 1998).

Como a maioria das técnicas de agrupamento, os centróides finais dos grupos não representam uma solução ótima global, mas apenas uma solução ótima local.

Além disso, os resultados obtidos podem ser completamente diferentes por causa da variedade de escolhas a serem atribuídas aos centóides iniciais. Para tanto, existem diversas técnicas propostas para a escolha dos centróides, sendo algumas delas aleatórias (ver McQueen 1967).

O algoritmo k-médias começa selecionando uma partição inicial dos dados. Para esta partição inicial, calcula-se um centróide a cada grupo. Em seguida, cada objeto do conjunto de dados é realocado para o grupo que tiver o centro mais próximo em uma tentativa de reduzir a dissimilaridade dentro dos grupos. Se um objeto muda de grupo, por exemplo do Ω_s para o Ω_t, os centróides destes grupos são modificados, sendo necessário atualizar o valor desses centróides e recalcular as distâncias entre os objetos a estes novos centróides. Este processo continua até que haja convergência, ou seja, os objetos permanecem fixos em seus grupos. Os principais passos do algoritmo são descritos a seguir:

Algorithm 1

1. Selecionar uma partição inicial do conjunto de dados em g grupos $\mathcal{P} = \{\Omega_1, \ldots, \Omega_g\}$
2. Calcular os centróides dos grupos , considerando n_k o número total de objetos num grupo Ω_k, tem-se $\overline{\omega_k} = \frac{1}{n_k} \sum_{j=1}^{n_k} \omega_{kj}$, $k = 1, \ldots,$ g
3. Para todo ω_k e seguindo a seqüência dos casos faça

   a)Realoque o objeto ω_k para o grupo com o centróide mais próximo, $\omega_k \in \Omega_s$ é movido de Ω_s para Ω_t se $\|\omega_k - \overline{\omega_t}\| \leq \|\omega_k - \overline{\omega_s}\|$, $\forall j = 1, \ldots,$g $t \neq s$

   b)Recalcule os centróides para os grupos Ω_s e Ω_t
4. Se os membros dos grupos estão estabilizados então pare, senão volte ao passo 3.

Apesar do algoritmo k-médias ser utilizado em uma variedade de aplicações, ele não está isento de desvantagens, algumas das quais tem sido amplamente apresentadas na literatura. As mais importantes são listadas abaixo,

- Como muitos métodos de agrupamento, o k-média requer a especificação do número de grupos g a priori, o que, obviamente, não é necessariamente verdade em aplicações reais;
- Como uma técnica iterativa, o algoritmo k-médias é especialmente sensível às condições iniciais, grupos iniciais e ordem dos objetos;
- O algoritmo k-médias converge de forma limitada a um mínimo local da função de custo associada.

2.2.2 Partition Around Medoids (PAM)

Este algoritmo é baseado na busca, entre todos os objetos do conjunto de dados, de k objetos que serão os representantes de cada grupo. Esses objetos poderão representar vários aspectos da estrutura dos dados. Na literatura da análise de clusters

esses objetos representativos são conhecidos como *medoides* dos grupos, ou ainda objetos *centrais*. A idéia principal desse algoritmo é que, depois de serem encontrados esses k objetos representativos, os k grupos serão construídos através da atribuição de cada objeto do conjunto de dados ao grupo que possui o objeto representante mais próximo.

Uma maneira de definir o objeto representante é encontrar o objeto que possui a menor dissimilaridade média em relação a todos os outros objetos do grupo. Como ilustração, considere a Tabela 1.

Objeto	Variável 1	Variável 2
1	1	4
2	5	1
3	5	2
4	5	4
5	10	4
6	25	4
7	25	6
8	25	7
9	25	8
10	29	7

Tabela 1: Conjunto de dados ilustrativo.

Suponha que os objetos 1 e 5 sejam selecionados arbitrariamente como representantes em um processo de agrupamento em $k = 2$ grupos. A Tabela 2 (página 36) atribui objetos aos grupos associados a esses representantes. Ela também contém, na quarta coluna, a menor dissimilaridade entre os objetos do conjunto de dados e os dois objetos representantes escolhidos. A dissimilaridade média é 9.37, e este valor pode ser utilizado para medir a qualidade dos agrupamentos com base na coesão dos objetos.

A Tabela 3 (página 37) possui as mesmas informações contidas na Tabela 2 (página 36), mas agora considerando os objetos representantes como sendo 4 e 8. A média obtida é 2.30, que é consideravelmente menor do que a obtida ao considerar como objetos representativos os de índices 1 e 5.

Utilizando o exemplo dado anteriormente, podemos optar como representantes, para o conjunto de dados da Tabela 1, os objetos 4 e 8, pois eles apresentaram a menor média das dissimilaridades encontradas. É dessa maneira que o algoritmo PAM escolhe os objetos representantes. Existem variações dessa escolha, por exemplo, pode-se utilizar a soma das dissimilaridades ao invés da média.

Objeto	Diss. do Objeto 1	Diss. do Objeto 5	Diss. Mínima	Objeto Representativo mais próximo
1	0.00	9.00	0.00	1
2	5.00	5.83	5.00	1
3	4.47	5.39	4.47	1
4	4.00	5.00	4.00	1
5	9.00	0.00	0.00	5
6	24.00	15.00	15.00	5
7	24.08	15.13	15.13	5
8	24.19	15.30	15.30	5
9	24.33	15.52	15.52	5
10	28.16	19.24	19.24	5
			Média: 9.37	

Tabela 2: Dissimilaridades entre os objetos do conjunto de dados e os representantes 1 e 5.

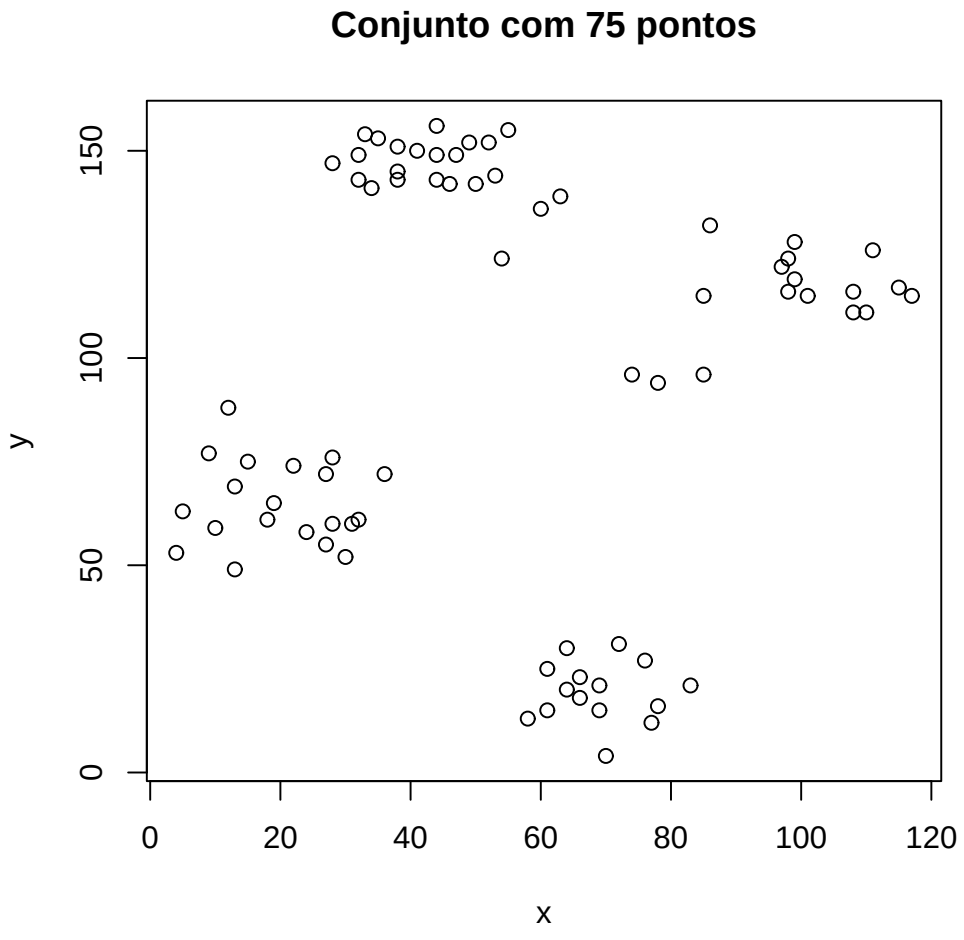


Figura 6: Representação do conjunto de dados de Ruspini.

Objeto	Diss. do Objeto 4	Diss. do Objeto 8	Diss. Mínima	Objeto Representativo mais próximo
1	4.00	24.19	4.00	4
2	3.00	20.88	3.00	4
3	2.00	20.62	2.00	4
4	0.00	20.22	0.00	4
5	5.00	15.30	5.00	4
6	20.00	3.00	3.00	8
7	20.10	1.00	1.00	8
8	20.22	0.00	0.00	8
9	20.40	1.00	1.00	8
10	24.19	4.00	4.00	8
			Média: 2.30	

Tabela 3: Dissimilaridades entre os objetos do conjunto de dados e os representantes 4 e 8.

Para ilustrar este algoritmo considere o conjunto de dados da Tabela 4 (página 48), também conhecidos como dados de Ruspini, que apresenta 75 pontos. Este conjunto de dados pode ser visualizado ainda na Figura 6 (página 36), onde são reconhecidos quatro grupos.

Suponha que o conjunto de pontos da Tabela 4 (página 48) deve ser dividido em 5 grupos, ou seja, $k = 5$. Dessa maneira é necessário encontrar cinco pontos representantivos de tal forma que estes pontos possuam as menores dissimilaridades médias em relação a todos os outros do conjunto.

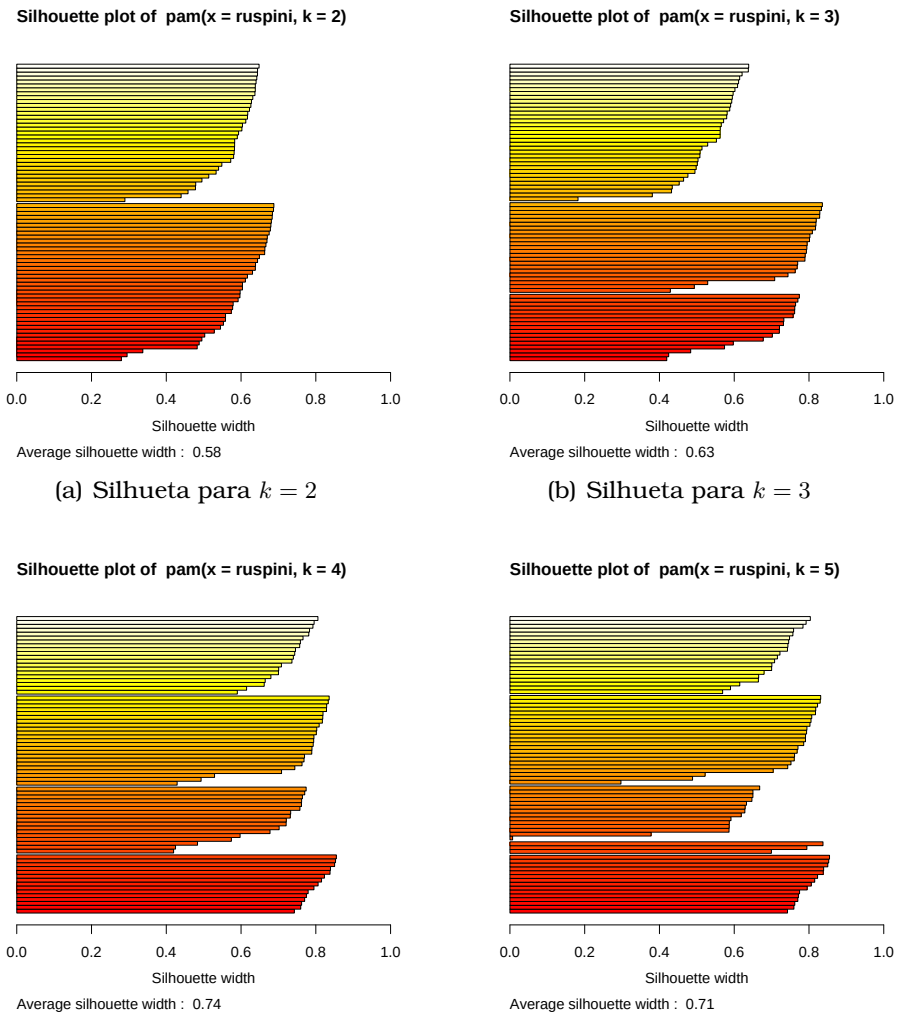
Utilizando a ferramenta R (ver Seção 1.3.2, página 24), é possível obter uma matriz de dissimilaridade para o conjunto de dados de **Ruspini** e utilizá-la como parâmetro no algoritmo PAM, como mostrado a seguir:

```
> matriz_dissimilaridade_Pontos <- dist(ruspini)
> resultado_pam <- pam(matriz_dissimilaridade_Pontos,k=2,
                       diss=TRUE)
```

No R, a função $dist$ calcula e retorna uma matriz de distâncias, usando uma medida de distância específica, tais como Euclidiana, Canberra ou Manhattan. Já a função pam retorna um agrupamento dos dados em k grupos utilizando o algoritmo PAM descrito nesta seção. O primeiro parâmetro dessa função é a matriz de dissimilaridade dos dados, calculada através de $dist$; o segundo parâmetro é um inteiro positivo indicando o número de grupos e o terceiro parâmetro é um valor lógico que, se verdadeiro, indica que o primeiro parâmetro passado é uma matriz de dissimilaridade, se falso, indica que é uma matriz de observações por variáveis. Esta função apresenta ainda outros parâmetros não utilizados nesses exemplos.

Os resultados apresentados na Figura 7 são obtidos através da seguinte função do R:

```
> plot(resultado_pam)
```



(a) Silhueta para $k = 2$

(b) Silhueta para $k = 3$

(c) Silhueta para $k = 4$

(d) Silhueta para $k = 5$

Figura 7: Gráficos de silhueta dos agrupamentos realizados com PAM.

Os gráficos apresentados na Figura 7 são gráficos de silhueta que auxiliam na determinação do melhor número de grupos. Todos eles foram gerados a partir dos resultados obtidos com o algoritmo PAM. Nesses gráficos, para cada objeto i é definido um índice $s(i) \in [-1, 1]$, que mede a diferença entre $b(i)$ e $a(i)$, onde $a(i)$ é a dissimilaridade média do objeto i a todos os outros objetos do grupo que ele pertence, e $b(i)$ é a dissimilaridade média do objeto i a todos os objetos que pertencem aos outros grupos próximos. Quando $s(i)$ está próximo de 1, o objeto i está mais perto do seu próprio grupo do que dos grupos vizinhos e dessa forma tem-se uma boa classificação. Da mesma forma, quando $s(i)$ está perto de -1 a relação contrária é obtida e é considerada uma má classificação. Quando $s(i)$ está próximo de 0 (zero), significa que não

está claro se o objeto deveria ter sido associado ao seu próprio grupo ou a algum grupo vizinho.

No gráfico de silhueta, os $s(i)$ são mostrados como barras horizontais, colocadas em ordem descrescente para cada grupo. Esses gráficos são um meio de avaliar a qualidade de uma solução para agrupamentos, permitindo identificar os grupos mais fortes e os fracos.

Quando se tem várias soluções obtidas de várias escolhas para o número de grupos, como é o caso da Figura 7 (página 38), é possível realizar uma comparação entre esses gráficos e obter o número de grupos que representa a solução ótima dos agrupamentos. Isto pode ser feito através da largura média da silhueta (*average silhouette width*), que representa a média de $s(i)$ sobre o conjunto de dados inteiro.

Na Figura 7(a) (página 38), pode-se observar a formação de dois grupos, uma vez que o parâmetro $k = 2$ foi escolhido. Nesse caso, silhuetas curtas penalizam as fusões artificiais. Para a Figura 7(b) um outro grupo foi encontrado. Este grupo é na verdade a divisão do grupo inferior da Figura 7(a). Na Figura 7(c) a solução "ótima" foi encontrada. Nela quatro grupos são formados e sua largura média da silhueta ou $\bar{s}(i)$ é a mais próxima de 1 se comparada com as dos outros três gráficos. A Figura 7(d) apresenta 5 agrupamentos, mas seu $\bar{s}(i)$ é inferior ao da Figura 7(c), logo concluímos que esta é a melhor solução para o agrupamento realizado com PAM.

A seção seguinte descreve o CLARA, um algoritmo de agrupamento de dados baseado no PAM, descrito nesta seção.

2.2.3 Clustering Large Application (CLARA)

O algoritmo PAM descrito na seção anterior apresenta bons resultados em situações onde o conjunto de dados utilizado é relativamente pequeno. Se PAM for utilizado com um conjunto de dados grande, ele poderá não ser tão útil, uma vez que o seu tempo de processamento e os requerimentos de memória tornam-se inviáveis.

O algoritmo CLARA (*Clustering Large Application*) é baseado no algoritmo PAM, e é mais indicado quando o conjunto de dados é grande.

O agrupamento de um conjunto de objetos com CLARA é realizado em dois passos. No primeiro, uma amostra é extraída do conjunto de objetos original aleatoriamente e agrupada em k sub-grupos usando o algoritmo PAM, que irá fornecer k objetos representativos. Utilizando o PAM apenas nessas amostras, de tamanhos relativamente pequeno em relação ao conjunto total de dados, ele poderá apresentar bons resultados.

O tamanho das amostras depende do número de grupos. Para um agrupamento em k grupos, o tamanho das amostras recomendado (ver Kaufman & Rousseeuw 1990) é $40 + 2k$. Se o número de grupos varia entre 1 e 30, as amostras deverão conter

entre 42 e 100 objetos, respectivamente. O uso de uma função do número de grupos para calcular o tamanho da amostra é motivado pelo objetivo de ter uma probabilidade razoável de encontrar objetos de todos os grupos existentes em pelo menos uma das amostras geradas.

Para a construção da primeira amostra o número desejado de objetos é selecionado do conjunto total através de uma amostragem simples sem reposição. Para este fim, pode ser empregado um gerador de números pseudoaleatórios (Bustos & Frery 1992*b*).

No segundo passo, cada objeto que não pertencer à amostra extraída inicialmente, é associado ao objeto representativo mais próximo, isto produz um agrupamento do conjunto de objetos original. A medida da qualidade dos agrupamentos, semelhantemente ao PAM, é obtida através do cálculo da distância média entre cada objeto do conjunto de dados original e seu objeto representativo. Depois que um determinado número de amostras tenham sido extraídas e agrupadas, a que apresentar a menor distância média será selecionada.

As três últimas seções apresentaram exemplos de algoritmos de agrupamento por partição e suas aplicações. A próxima seção abordará os algoritmos hierárquicos, suas classificações e exemplos.

2.3 Agrupamento hierárquico

O resultado obtido por um algoritmo hierárquico é um conjunto de partições para os dados de entrada e todos os possíveis valores de $g = \{1, \ldots, n\}$, tipicamente organizado na forma de uma árvore. O conjunto de partições será definido como $P = \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$. Quando $g = 1$, tem-se $\mathcal{P}_1 = \{\Omega\}$ sendo $\Omega = \{\omega_1, \ldots, \omega_n\}$, ou seja, tem-se uma partição formada por um único grupo contendo todos os n indivíduos. Quando $g = n$, tem-se $\mathcal{P}_n = \{\Omega_1, \ldots, \Omega_n\}$ sendo $\Omega_i = \{\omega_i\}$, ou seja, tem-se uma partição formada por n grupos unitários. Os resultados obtidos por valores intermediários, $g = \{2, 3, \ldots, n-1\}$, formam uma transição gradual, onde a diferença entre $g = k$ e $g = k + 1$ é a divisão de um cluster em dois para se obter $k + 1$ grupos. Analisando a operação numa direção oposta, encontra-se a diferença entre $g = k + 1$ e $g = k$ através da junção entre dois clusters para se obter k grupos.

Em muitas definições clássicas, a passagem de $g = k$ para $g = k + 1$ ($g = k + 1$ para $g = k$, respectivamente) se efetua pela divisão (pela fusão, respectivamente) de exatamente um grupo (dois grupos, respectivamente) em dois (em um, respectivamente). O algoritmo SPC, que será explorado neste trabalho, não procede necessariamente dessa forma. Para este algoritmo, o agrupamento hierárquico não requer a especificação da passagem de $g = k$ para $g = k + 1$ (nem de $g = k + 1$ para $g = k$). No entanto, será mantida esta especificação no restante desta seção introdutória.

A forma com que os resultados são obtidos define dois tipos de técnicas: a aglomerativa, gerando partições $\mathcal{P}_i$ com $g = \{n, \ldots, 1\}$, por ter a propriedade de agrupar dois clusters a cada iteração; e a divisiva, formando partições $\mathcal{P}_i$ com $g = \{1, \ldots, n\}$, por ter a propriedade de dividir um cluster em dois a cada iteração. A Figura 8 ilustra os resultados dos agrupamentos para um conjunto de dados com $n = 5$ indivíduos. As direções seguidas pelos métodos aglomerativo e divisivo encontram-se, respectivamente, na parte superior e inferior. No exemplo, os resultados dos dois tipos de métodos coincidem, mas usualmente eles são diferentes (Kaufman & Rousseeuw 1990). Diferentes também são os seus respectivos custos computacionais, tipicamente.

As divisões e uniões realizadas pelos métodos hierárquicos são irreversíveis, ou seja, se um algoritmo aglomerativo juntar dois indivíduos, estes não poderão ser separados nas próximas iterações. O mesmo ocorre com os métodos divisíveis, quando ocorre a divisão de grupo em dois, estes não poderão ser unidos novamente. Este fato faz com que uma possível falha nos resuldados obtidos em uma iteração anterior nunca seja reparada.

As aplicações dos métodos hierárquicos se destinam à biologia, em especial na construção de árvores taxonômicas, estudos de sistemas sociais, biblioteconomia, arqueologia, entre outras.

Como ilustrado na Figura 8 o conjunto de partições, $P = \{\mathcal{P}_1, \ldots, \mathcal{P}_n\}$, pode ser representado por um diagrama bidimensional, chamado 'dendrograma'.

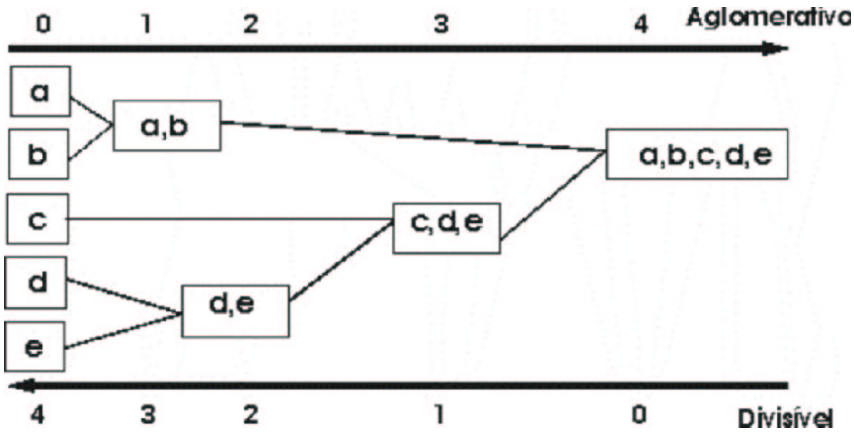


Figura 8: Resultados das técnicas aglomerativas e divisivas.

O dendrograma tem a estrutura de uma árvore: os nós representam os grupos formados; os níveis representam as partições $\mathcal{P}_i$, e as folhas estão associadas aos grupos formados pela primeira partição $\mathcal{P}_1$. Como exemplo, considere o dendrograma formado pelo método aglomerativo da Figura 8, tem-se as n folhas representando os n grupos unitários de $\mathcal{P}_1$, e quatro nós cujas associações são para os grupos: $\{a, b\}$ no nível 1; $\{d, e\}$ no nível 2; $\{c, d, e\}$ no nível 3 e $\{a, b, c, d, e\}$ no nível 4.

Normalmente, os dendrogramas possuem a topologia de árvores binárias por

conta das limitações encontradas nas técnicas clássicas. Uma ilustração dessa estrutura pode ser vista na Figura 9 (pagina 44). Mas, como explicado no início dessa seção, sua estrutura pode variar em qualquer tipo de árvore quando aplicada a métodos 'generalizados' de agrupamento hierárquico. Quando o volume de dados é considerável, a construção dos dendrogramas pode se tornar computacionalmente custosa, e de difícil, ou até impossível, análise dos dados. Este fato se deve à ordem com que os dados são dispostos nas folhas do dendrograma, que pode influenciar diretamente na análise dos resultados, dado que existem 2^{n-1} formas de construção. Este problema é chamado de 'ordenação do dendrograma' (ver Morris, Asnake & Yen 2003).

As duas subseções subsequentes, explicam esses dois tipos de métodos hierárquicos em detalhes, mostrando alguns dos algoritmos que os implementam.

2.3.1 Métodos Aglomerativos

Os métodos aglomerativos são, provavelmente, os mais utilizados entre os métodos hierárquicos. Eles iniciam seus algoritmos com uma partição contendo n grupos unitários, $\mathcal{P}_1 = \{\Omega_1, \ldots, \Omega_n\}$, com $\Omega_i = \{\omega_i\}$. A cada iteração dois grupos são unidos, até chegar a uma partição contendo um único grupo que conterá todos os indivíduos, definida por $\mathcal{P}_n = \{\Omega\}$ com $\Omega = \{\omega_1, \ldots, \omega_n\}$. O critério de agrupamento utilizado depende das medidas de proximidade calculadas sobre os dados de entrada.

As operações básicas de todos os métodos aglomerativos são similares, diferenciando-se pelas medidas de dissimilaridade (ou similaridade) escolhidas. Assim sendo, só precisam ser explicados em detalhes dois exemplos específicos para ilustrar estas técnicas e, para tanto, foram escolhidos a junção simples (*single linkage*) e a junção por centróide (*centroid linkage*).

2.3.1.1 Algoritmo de Junção Simples

O algoritmo de *junção simples* também é conhecido como técnica do vizinho mais próximo (*nearest-neighbour*). É um dos métodos hierárquicos mais simples. A definição do método se faz pelas distâncias mais próximas (medidas de dissimilaridades) que formam o critério de união entre dois grupos ou indivíduos. Esta técnica também pode ser aplicada a métodos divisivos, o critério de divisão em dois grupos é a escolha das distâncias de máximo valor (Everitt et al. 2001).

O critério de escolha dos grupos a serem unidos é definido por: dada uma matriz de dissimilaridade (distância) de tamanho $n \times n$ contendo valores representados por d_{ij}, com $I, J \in N$, onde N é um conjunto constituído por todos os indivíduos e subconjuntos obtidos até a iteração anterior, juntar dois grupos ou indivíduos denotados por I e J se e somente se $\min_{\{I,J \in N | I<J\}}(d_{IJ})$.

Para o algoritmo de junção será apresentado um exemplo cuja matriz de dissimilaridade é descrita abaixo:

$$D_1 = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0.0 & & & & \\ 2.0 & 0.0 & & & \\ 6.0 & 5.0 & 0.0 & & \\ 10.0 & 9.0 & 4.0 & 0.0 & \\ 9.0 & 8.0 & 5.0 & 3.0 & 0.0 \end{bmatrix}$$

No exemplo, os indivíduos escolhidos na fase inicial são 1 e 2, porque possuem o menor valor de dissimilaridade diferente de 0 (zero), no caso $d_{12} = 2.0$. Redefine-se os valores de dissimilaridades de acordo com os novos dados:

$$\begin{aligned} d_{(12)3} &= \min[d_{13}, d_{23}] = d_{23} = 5.0 \\ d_{(12)4} &= \min[d_{14}, d_{24}] = d_{24} = 9.0 \\ d_{(12)5} &= \min[d_{15}, d_{25}] = d_{25} = 8.0 \end{aligned}$$

Uma nova matriz é definida de acordo com as informações atualizadas.

$$D_2 = \begin{array}{c} \{1,2\} \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0.0 & & & \\ 5.0 & 0.0 & & \\ 9.0 & 4.0 & 0.0 & \\ 8.0 & 5.0 & 3.0 & 0.0 \end{bmatrix}$$

De acordo com D_2, os indivíduos 4 e 5 são agrupados. Redefinindo-se novamente os valores de dissimilaridades e a matriz, como ilustra D_3.

$$D_3 = \begin{array}{c} \{1,2\} \\ 3 \\ \{4,5\} \end{array} \begin{bmatrix} 0.0 & & \\ 5.0 & 0.0 & \\ 8.0 & 4.0 & 0.0 \end{bmatrix}$$

Em seguida são agrupados o indivíduo 3 com o grupo $\{4,5\}$. Finalmente os grupos $\{1,2\}$ e $\{3,4,5\}$ são agrupados em um único grupo. O dendrograma ilustrado na Figura 9 (página 44) retrata o exemplo citado acima.

2.3.1.2 Algoritmo de Junção por Centróide

O método de junção por centróide, também conhecido como UPGMC (*unweighted pair-group method using centroid approach*), baseia-se no método de junção simples, porém ao invés de operar diretamente em uma matriz de dissimilaridade, ele requer o acesso dos dados de entrada. Com a informação dos dados de entrada, defini-se uma matriz de dissimilaridade através de métricas estabelecidas, como a distância Euclidiana.

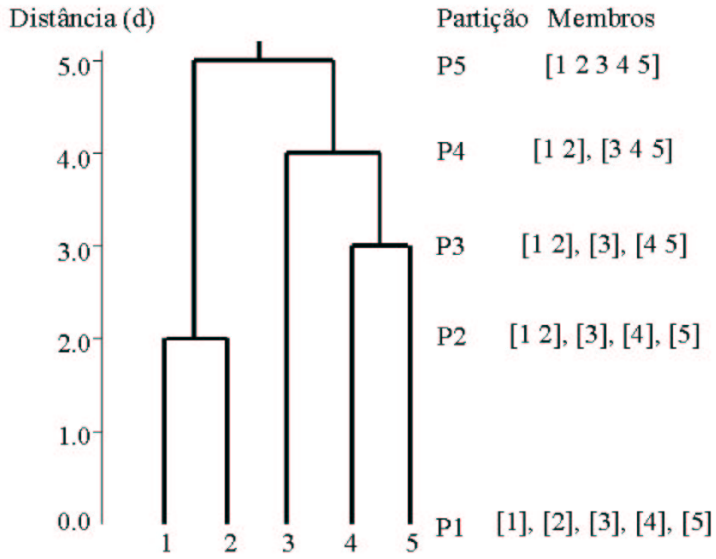


Figura 9: Dendrograma para o exemplo da junção simples, mostrando partições em cada passo.

Em seguida, define-se os grupos/indivíduos que possuem a menor dissimilaridade. Recalcula-se a matriz com as informações obtidas pelo dado de entrada (e não pela matriz de dissimilaridade anterior, como ocorre na junção simples). Repete-se este processo até a definiçãao do grupo final, que conterá todos os n indivíduos.

Uma ilustração do método de junção por centróide está definida na Tabela 5 (página 49). Dada uma matriz de $n = 5$ indivíduos contendo duas variáveis. Estabeleceu-se a distância Euclidiana para encontrar as matrizes de dissimilaridade. A partir dos dados de entrada, a matriz de dissimilaridade inicial é:

$$C_1 = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} 0.0 & & & & \\ 1.0 & 0.0 & & & \\ 5.39 & 5.10 & 0.0 & & \\ 7.07 & 7.0 & 2.24 & 0.0 & \\ 7.07 & 7.28 & 3.61 & 2.0 & 0.0 \end{bmatrix}$$

Através de C_1, une-se os indivíduos com menor dissimilaridade, 1 e 2. O vetor médio (centróide) do grupo é calculado, no caso $(1, 1.5)$:

$$centroide_{12} = ((1.0 + 1.0)/2, (1.0 + 2.0)/2) = (1.0, 1.5)$$

A nova matriz de dissimilaridade será calculada com este novo valor, que corresponde ao centróide para o grupo $\{1, 2\}$. Ela está definida em C_2.

$$C_2 = \begin{matrix} \{1,2\} \\ 3 \\ 4 \\ 5 \end{matrix} \begin{bmatrix} 0.0 & & & \\ 5.22 & 0.0 & & \\ 7.02 & 2.24 & 0.0 & \\ 7.16 & 3.61 & 2.0 & 0.0 \end{bmatrix}$$

Segundo a matriz C_2, o próximo grupo a ser formado conterá $\{4,5\}$. O vetor médio encontrado para este grupo foi $(8.0, 1.0)$. Este vetor influenciará diretamente na próxima matriz.

$$C_3 = \begin{matrix} \{1,2\} \\ 3 \\ \{4,5\} \end{matrix} \begin{bmatrix} 0.0 & & \\ 5.22 & 0.0 & \\ 7.02 & 2.83 & 0.0 \end{bmatrix}$$

Em C_3, o indivíduo 3 se agrupa com o grupo $\{4,5\}$. O estágio final une os dois grupos $\{1,2\}$ e $\{3,4,5\}$.

Existem outros algoritmos aglomerativos que se assemelham a estes dois métodos. O diferencial se encontra nas métricas que definem os critérios de agrupamento. Por exemplo, o método de junção completa (conhecido também como método do vizinho mais distante) tem a idéia oposta ao de junção simples: a medida de proximidade é definida pelas maiores distâncias entre grupos. Na junção média, também conhecida como UPGMA (*unweighted pair-group method using average approach*), as distâncias entre dois grupos são definidas pelas médias das distâncias entre todos os indivíduos dos dois grupos. Uma ilustração destes dois métodos encontra-se na Figura 10.

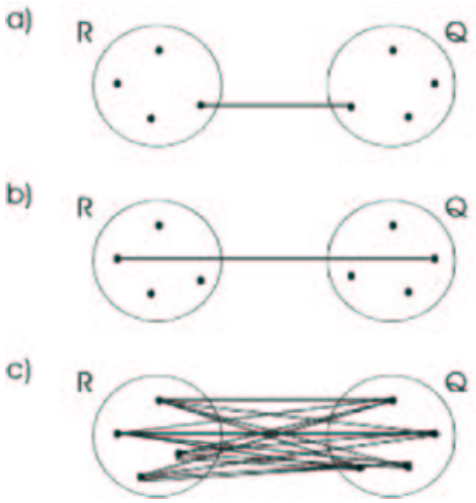


Figura 10: Representação das medidas de dissimilaridade definidas entre os métodos: (a) Junção simples. (b) Junção Completa. (c) Junção média: $d_{RQ} = \frac{\sum_{i \in R, j \in Q} d_{i,j}}{n}$.

2.3.2 Métodos Divisivos

Por outro lado, os métodos divisivos iniciam com uma partição formada por um único grupo contendo todos os indivíduos, $\mathcal{P}_1 = \{\Omega\}$ com $\Omega = \{\omega_1, \ldots, \omega_n\}$. A cada iteração, um grupo é dividido em dois outros grupos até ser obtida uma partição formada por n grupos unitários, definida por $\mathcal{P}_n = \{\Omega_1, \ldots, \Omega_n\}$, com $\Omega_i = \{\omega_i\}$.

Estes métodos operam na direção oposta aos métodos aglomerativos (ver Figura 8, página 41). Eles iniciam com uma partição formada por um único grupo contendo todos os indivíduos, $\mathcal{P}_1 = \{\Omega\}$ com $\Omega = \{\omega_1, \ldots, \omega_n\}$. A cada iteração, um grupo é dividido em dois outros grupos até ser obtida uma partição formada por n grupos unitários, definida por $\mathcal{P}_n = \{\Omega_1, \ldots, \Omega_n\}$, com $\Omega_i = \{\omega_i\}$.

Esses métodos exigem, computacionalmente, no máximo $2^{k-1}-1$ divisões possíveis em cada nível, ou seja, divisão de um grupo com k objetos em dois subgrupos. Entretanto, para dados binários (dados que possuem apenas dois valores, 0 e 1) existem algoritmos relativamente simples e computacionalmente eficientes. Um exemplo desses algoritmos é o MONA (*Monothetic Analysis*). Este algoritmo é completamente diferente dos vistos até agora, pois sua idéia básica é selecionar uma das variáveis do conjunto de objetos original e dividir este em objetos com e sem o atributo correspondente. Dessa maneira, para cada variável do conjunto de objetos, serão obtidos dois subconjuntos, sendo que o processo continua com a escolha de uma outra variável que, da mesma maneira, divide esse subconjunto em dois grupos menores. Esse processo continua até que o subconjunto contenha um único objeto ou então que as variáveis restantes não consigam mais separar os objetos. Esta última situação só ocorre quando cada variável contém valores constantes para todos os objetos no subconjunto. Para entender melhor essa última situação considere a Tabela 6 (página 49), onde os objetos **B** e **C** não podem ser separados.

Pelo fato de que o conjunto de objetos é dividido em subconjuntos e este processo é contínuo dentro de cada subconjunto, o algoritmo é hierárquico. Mais precisamente ele é divisivo. Além disso, por ser a separação realizada usando uma única variável por vez, ela é chamada de *monotética*.

A parte mais importante do algoritmo é escolher a variável que irá dividir o conjunto de objetos. A idéia principal é escolher a variável mais centralmente localizada, ou seja, escolher a variável que apresentar a maior soma das 'similaridades' a todas as outras variáveis.

Uma maneira simples e utilizada por MONA para obter uma medida de similaridade entre duas variáveis é calcular o produto entre o número de objetos para os quais as duas variáveis possuem o valor 0 e o número para os quais possuem o valor 1. Em seguida, calcular o produto entre o número de objetos para os quais a primeira variável apresenta valor 0 e a segunda 1 e o número de objetos para os quais a primeira variável apresenta valor 1 e a segunda 0. A medida de similaridade

é definida como sendo o valor absoluto da diferença entre esses dois produtos. Para uma melhor compreensão do cálculo dessa medidade de similaridade considere novamente a Tabela 6 (página 49). As variáveis 1 e 2 são idênticas e dessa forma deveriam apresentar uma alta similaridade. Os dois produtos são 3 (1 x 3) e 0 (0 x 0), então a medida de similaridade é $|3 - 0| = 3$. Já as variáveis 1 e 3 são muito diferentes, logo devem apresentar uma baixa similaridade. Os produtos para elas são 0 (0 x 1) e 2 (1 x 2). A medida de similaridade para as variáveis 1 e 3 é $|0 - 2| = 2$.

A medida de similaridade apresentada só é recomendada se as duas variáveis utilizadas na análise fornecem divisões similares do conjunto de dados. As variáveis 1 e 2 por ser idênticas não apresentam problemas, mas as variáveis 1 e 3, por exemplo, apresentam informações muito diferentes e mesmo assim o valor da similaridade foi próximo do valor calculado para as variáveis 1 e 2. Logo, pode-se concluir que se duas variáveis apresentam valores diferentes para todos os objetos de um conjunto de dados, elas dão informações idênticas e a similaridade torna-se alta. Como um exemplo desta conclusão considere as variáveis 1 e 4, que são completamente diferentes. Calculando os produtos temos 0 (0×0) e 3 (1×3). A medida de similaridade para as variáveis 1 e 4 é $|0 - 3| = 3$, que é exatamente o mesmo valor obtido quandos duas variáveis são iguais.

Como o objetivo é encontrar a variável que é mais similar a todas as outras, a soma das similaridades a todas as outras variáveis é então maximizada. A variável para a qual a soma é máxima, ou seja, a variável que apresentar a maior soma entre todas as calculadas será a escolhida para dividir o conjunto ou subconjunto de dados.

No próximo capítulo será discutida a técnica de agrupamento superparamagnético e seus principais conceitos. Suas principais vantagens e uso serão apresentados no capítulo 6 (página 115).

Ponto	coordenada x	coordenada y	**Ponto**	coordenada x	coordenada y
1	4	53	**46**	85	96
2	5	63	**47**	78	94
3	10	59	**48**	74	96
4	9	77	**49**	97	122
5	13	49	**50**	98	116
6	13	69	**51**	98	124
7	12	88	**52**	99	119
8	15	75	**53**	99	128
9	18	61	**54**	101	115
10	19	65	**55**	108	111
11	22	74	**56**	110	111
12	27	72	**57**	108	116
13	28	76	**58**	111	126
14	24	58	**59**	115	117
15	27	55	**60**	117	115
16	28	60	**61**	70	4
17	30	52	**62**	77	12
18	31	60	**63**	83	21
19	32	61	**64**	61	15
20	36	72	**65**	69	15
21	28	147	**66**	78	16
22	32	149	**67**	66	18
23	35	153	**68**	58	13
24	33	154	**69**	64	20
25	38	151	**70**	69	21
26	41	150	**71**	66	23
27	38	145	**72**	61	25
28	38	143	**73**	76	27
29	32	143	**74**	72	31
30	34	141	**75**	64	30
31	44	156	-	-	-
32	44	149	-	-	-
33	44	143	-	-	-
34	46	142	-	-	-
35	47	149	-	-	-
36	49	152	-	-	-
37	50	142	-	-	-
38	53	144	-	-	-
39	52	152	-	-	-
40	55	155	-	-	-
41	54	124	-	-	-
42	60	136	-	-	-
43	63	139	-	-	-
44	86	132	-	-	-
45	85	115	-	-	-

Tabela 4: Conjunto de dados - Pontos.

Indivíduo	Variável 1	Variável 2
1	1.0	1.0
2	1.0	2.0
3	6.0	3.0
4	8.0	2.0
5	8.0	0.0

Tabela 5: Exemplificação de um dado de entrada.

Objetos/Variáveis	1	2	3	4
A	0	0	1	1
B	1	1	0	0
C	1	1	0	0
D	1	1	1	0

Tabela 6: Exemplo de conjunto de dados com variáveis binárias.

3 Agrupamento Superparamagnético

Este capítulo descreverá o algoritmo de agrupamento de dados baseado no comportamento superparamagnético do modelo de Potss não homogêneo. Essa descrição será feita partindo-se do problema de agrupamento e analisando-se o modelo fisico utilizado, além do método de *Swendsen-Wang* (responsável pela simulação de ocorrências deste modelo). Outros pontos abordados na descrição são: a localização das regiões superparamagnéticas e a identificação dos grupos de dados.

No Capítulo 2 (página 27) foi descrito o problema de agrupar dados com base em alguns requisitos, são eles:

- indivíduos semelhantes entre si são atribuídos a um mesmo grupo;
- indivíduos que pertencem a grupos distintos são diferentes entre si;
- todo indivíduo deve pertencer a algum grupo, e
- cada indivíduo faz parte de apenas um único grupo.

Recentemente foi estabelecida uma analogia entre o problema de agrupamento de dados e a procura de configurações típicas de um modelo físico, o modelo de Potts (Wiseman, Blatt & Domany 1998). Dado que existem algoritmos eficientes para a localização dessas configurações, como o método de Swendsen-Wang, por exemplo, é possível aplicar essas técnicas a uma grande diversidade de situações. As seções seguintes descrevem essa analogia e os métodos e técnicas citados anteriormente.

3.1 Definição

O método superparamagnético é uma técnica para agrupamento de dados baseada em um sistema que exibe comportamento magnético em função de um parâmetro, mais especificamente essa técnica de agrupamento de dados é baseada nas propriedades físicas de um sistema magnético e o parâmetro citado é conhecido como 'temperatura'; esta técnica foi proposta por Blatt, Wiseman & Domany (1996). O termo 'superparamagnético' se deve ao fato deste modelo exibir três tipos de comportamento associados a outras tantas 'regiões' da temperatura, são eles:

- ferromagnético (a temperaturas baixas);
- paramagnético (a temperaturas altas), e
- superparamagnético (a temperaturas intermediárias).

O último comportamento citado, que ocorre a temperaturas intermediárias entre altas e baixas, é capaz de fornecer informações que torna evidente o agrupamento de dados.

Como citado na seção anterior, o modelo físico tomado como base do agrupamento superparamagnético é o modelo de Potts. Uma maneira simples de entender o funcionamento deste método é através do modelo de Ising, que é uma especialização do modelo Potts, ou seja, uma vez entendido o modelo de Ising, restarão poucos conceitos até o entendimento do modelo de Potts (Seção 3.3, página 54), descrito em seguida.

Uma vez definido o modelo de Potts, o algoritmo de Swendsen-Wang (Seção 3.4, página 54) será descrito. Este permite gerar eficientemente ocorrências deste modelo e, portanto, é central para a implementação da técnica de agrupamento superparamagnético.

3.2 O Modelo de Ising

A proposta inovadora para solucionar o problema de agrupamento através do método superparamagnético é utilizar o modelo de Potts, que é uma generalização do modelo de Ising (ver, por exemplo, Bustos, Frery & Ojeda 1998, Pickard 1987, Wu 1982). Esta seção descreve as principais propriedades do modelo de Ising e na próxima seção o modelo de Potts.

Consideremos uma rede bi-dimensional $S = \{-M, \ldots, M\} \times \{-N, \ldots, N\}$, onde é associado uma partícula ou um átomo a cada um dos $(2M+1)(2N+1)$ pontos $s \in S$ da malha (ver Figura 11, página 52). No que segue será tratado o caso finito, mas as propriedades interessantes do modelo emergem ao considerar o caso infinito, isto é, quando $M, N \to \infty$ (ver também Horta 2004).

O átomo associado a cada coordenada $s \in S$ da malha poderá adotar um entre dois estados possíveis (*spin*). Estes estados podem ser entendidos como representativos da orientação do campo magnético que, por sua vez, pode ter sido criado pela rotação ou deslocamento de uma partícula carregada eletricamente, pois podemos imaginar elétrons em cada coordenada s da rede S. Dado que essa orientação poderá flutuar entre os dois estados possíveis sem que o observador tenha controle sobre ela, a mesma poderá ser convenientemente descrita por uma variável aleatória $X_s: \Omega \to \{-1, +1\}$ (algumas representações utilizam $\{0, 1\}$ como espaço de estados, sem conseqüência para o resto da discussão). O conjunto de todos os estados adotados pelas variáveis aleatórias $(X_s)_{s \in S}$ é chamado 'configuração' ou 'estado'; é usual

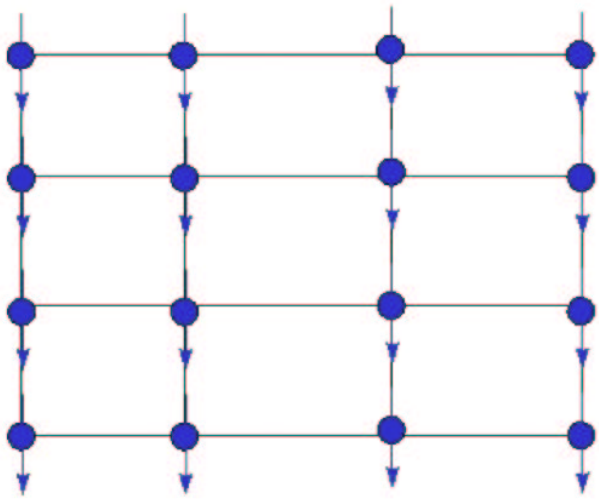


Figura 11: Grade Bidimensional.

a notação $\xi = (x_1, \ldots, x_{(2M+1)(2N+1)})$ para simbolizar um estado, como também pode ser visto em Horta (2004). O conjunto de todos os estados possíveis costuma ser denotado Ξ, e é imediato que ele possui $2^{(2M+1)(2N+1)}$ elementos.

O modelo de Ising consiste em associar uma medida de energia (ver equação (3.1)) a cada configuração $\xi \in \Xi$. A natureza, buscando o equilíbrio, tenderá a escolher configurações que possuam uma baixa energia, em detrimento das que possuam uma alta energia.

A energia total do sistema no estado ξ pode ser escrita da seguinte forma:

$$E(\xi) = -J \sum_{\langle u,v \rangle}^{N} x_u x_v - H \sum_{s \in S} x_s. \tag{3.1}$$

Na equação (3.1), x_s depende do sentido do *spin* do átomo. Se o *spin aponta* para cima, podemos considerar que x_s é igual a $+1$, caso contrário a -1. A constante $J \in \mathbb{R}$ parametriza o acoplamento entre os *spins*, ou seja, J representa a força de interação entre dois *spins*, e $H \in \mathbb{R}$ é um campo magnético externo. O primeiro termo da equação (3.1) é um somatório responsável pela contabilização das contribuições associadas à interação entre uma dada partícula e suas vizinhas. Para os propósitos do nosso trabalho, poderemos considerar $H = 0$ mas as interações J irão variar conforme as partículas que interagem. A notação '$\langle u, v \rangle$' indica que devem ser somados todos os pares de vizinhos $u, v \in S$.

Caso sejam consideradas interações de primeira ordem, um átomo que esteja na posição $u = (i, j)$ da malha pode interagir apenas com seus vizinhos nas posições $(i+1, j)$, $(i-1, j)$, $(i, j+1)$, $(i, j-1)$ (esta é chamada também interação de vizinhança-4).

O primeiro termo da equação faz com que a energia do sistema seja mínima (máxima) quando os *spins* estiverem alinhados paralelamente (antiparalelamente) no caso em que J é positivo, ou seja, J positivo favorece o **ferromagnetismo** (ver Horta 2004). A constante H do segundo termo é utilizada na descrição dos fenômenos de **Diamagnetismo** e **Paramagnetismo**, quando $H < 0$ e $H > 0$, repectivamente.

No caso deste projeto o campo externo poderá ser considerado nulo, isto é, $H = 0$, uma vez que ele é empregado quando o modelo de Potts assume um contexto diferente do exposto neste trabalho. Tal como apresentado pela equação (3.1), onde as interações entre partículas são descritas por uma única constante J, temos o chamado 'modelo de Ising homogêneo'. Para as aplicações de interesse neste trabalho, as interações irão depender das posições e, com isso, o modelo de Ising (agora não homogêneo) estará caracterizado pela energia

$$E(\xi) = -\sum_{\langle u,v \rangle} J(u,v) x_u x_v. \tag{3.2}$$

Um parâmetro muito conveniente para descrever o comportamento do modelo de Ising é a temperatura, que aparece escrevendo a energia da configuração ξ como

$$E_T(\xi) = -\frac{1}{T} \sum_{\langle u,v \rangle} J(u,v) x_u x_v, \tag{3.3}$$

com $T \in \mathbb{R}$; para os propósitos deste trabalho será suficiente supor $T > 0$.

Será conveniente no decorrer do texto empregar uma formulação equivalente à utilizada na equação (3.2): ao invés de somar produtos de *spins* vizinhos e, com isso, termos um somatório de '$+1$' e '-1', podemos mapear este somatório para um outro de somandos '1' e '0', respectivamente. Para isto, a equação (3.2) pode ser escrita

$$E_T(\xi) = -\frac{1}{T} \sum_{\langle u,v \rangle} J(u,v) \mathbb{I}(x_u, x_v), \tag{3.4}$$

onde a função $\mathbb{I}(a,b)$ vale 1 se $a = b$ e zero caso contrário. A diferença entre as formulações dadas pelas equações (3.2) e (3.4) é uma constante na energia da configuração, constante essa que não terá influência nenhuma no modelo.

O modelo de Ising é um modelo estocástico, pois ele associa uma probabilidade a cada configuração $\xi \in \Xi$:

$$\Pr_T(\xi) = \frac{1}{Z_T} \exp\{-E_T(\xi)\},$$

onde $Z_T = \sum_{\xi \in \Xi} \exp\{-E_T(\xi)\}$ é chamada 'constante de partição' ou ainda 'constante de normalização'.

A próxima seção detalhará o modelo de Potts a partir do modelo de Ising, exposto nesta seção.

3.3 O Modelo de Potts

Uma generalização para o modelo de Ising, que é limitado a apenas dois estados, é o modelo de Potts, onde cada elemento pode adotar um de k estados possíveis $\{\zeta_1, \ldots, \zeta_k\}$. Este modelo descreve o comportamento de partículas que interagem entre si a uma temperatura $T \in \mathbb{R}$. Sejam as n partículas $\mathbf{x} = \{x_{i,j}\}_{i,j=1}^{n}$ interagindo todas entre si. A partícula x_u interage (atrativamente) com a partícula x_v com intensidade $\beta(u, v) \geq 0$. A distribuição da configuração ζ é dada por

$$\Pr_T(\mathbf{x} = \zeta) = \frac{1}{Z_T} \exp\Big\{\frac{1}{T} \sum_{u,v} \beta(u, v)\mathbb{I}_{u,v}\Big\} \tag{3.5}$$

onde

$$\mathbb{I}_{u,v} = \begin{cases} 1 & \text{se} \quad \zeta_u = \zeta_v \\ 0 & \text{caso contrário,} \end{cases}$$

onde $Z_T = \sum_{\zeta \in \Xi} \exp\{T^{-1} \sum_{u,v} \beta(u, v)\mathbb{I}_{u,v}\}$ é uma constante de normalização e Ξ é o conjunto de todas as configurações possíveis. Para $T > 0$, que é o caso que iremos considerar neste trabalho, esta distribuição sempre irá favorecer as configurações de partículas do mesmo tipo, mais intensamente quanto menor for a temperatura e quanto mais atrativa for a interação.

No modelo clássico de Ising, temos que as interações consideradas dependem apenas da distância entre as partículas, e estas estão localizadas sobre uma grade regular. Já no modelo não homogêneo, que é um caso mais geral aqui considerado, a localização das partículas não é necessariamente levada em conta, pois todas podem interagir entre si.

3.4 O Método de Swendsen-Wang

Existem na literatura vários métodos para gerar ocorrências do modelo de Potts, como visto na seção anterior. O método de Swendsen-Wang foi o escolhido pelos pesquisadores da nova técnica por apresentar algumas vantagens significativas em relação aos outros métodos.

O método de Swendsen-Wang é uma dinâmica que realiza simulações de ocorrências do modelo de Potts de forma iterativa. Essas simulações, por sua vez, são importantes para estimar a esperança de variáveis aleatórias definidas sobre um sistema de Potts, uma vez que computâ-las de forma analítica é inviável, ou ainda, impossível em geral.

Um dos fatores que levaram à escolha do método de Swendsen-Wang é a sua capacidade de diminuir significativamente o decremento da eficiência computacional, quando o tamanho do sistema aumenta. Esta situação foi identificada, por exemplo, quando o algoritmo de Metropolis (Metropolis, Rosembluth, Rosembluth, Teller &

Teller 1953) foi empregado para gerar as ocorrências. Este é um algoritmo de Monte Carlo tradicional (Metropolis & Ulam 1949) que, por sua vez, são métodos largamente utilizados em estudos de mecânica estatística e, mais geralmente, em estudos que envolvem simulação estocástica (Bustos & Frery 1992*b*).

Segundo Blatt, Wiseman & Domany (1997), foi verificado que o algoritmo de Metropolis, quando empregado para simular o modelo de Potts, apresentava um notável decremento da eficiência computacional, em função do tamanho do sistema. A dinâmica de Swendsen-Wang ameniza este problema reduzindo significativamente esse decremento.

O algoritmo desenvolvido por Swendsen e Wang (Swendsen 1991, Swendsen & Wang 1987) realiza mudanças globais (não locais) nas configurações de *spins*, diferentemente do algoritmo de Metropolis, que é um algoritmo local. O algoritmo de Swendsen-Wang reduz muito o decremento da eficiência computacional para certos tipos de modelo.

Este método faz parte das etapas do algoritmo de agrupamento de dados (descrito na próxima seção) e seu algoritmo está descrito nos passos seguintes (esses passos também podem ser encontrados, sob o enfoque de imagens, em Horta (2004)):

1. Considere uma configuração de *spins* gerada aleatoriamente, ou seja, nesse primeiro passo, a configuração inicial a partir da qual o algoritmo começa o seu processamento é gerada de maneira arbitrária. Em particular, e sem perda de generalidade, considere a configuração onde cada *spin* é gerado independentemente dos outros com probabilidade uniforme nos k estados possíveis $\{\zeta_1, \ldots, \zeta_k\}$. Esta situação é exibida na Figura 12, onde são considerados $n = 10$ indivíduos e $k = 3$ estados possíveis.

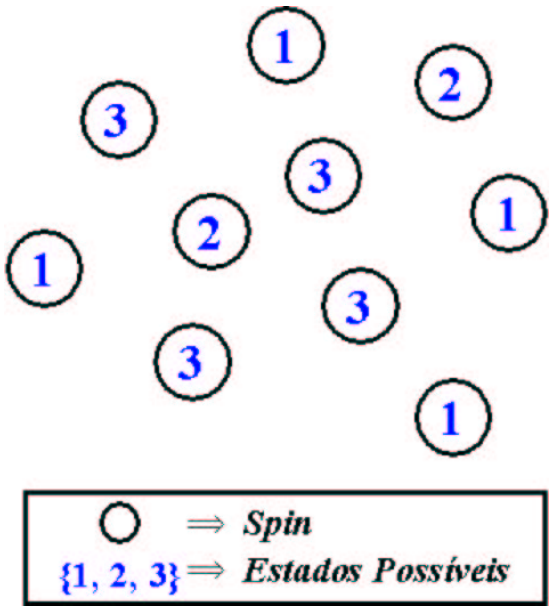


Figura 12: Associação entre *spins* e estados.

2. Neste passo, todos os pares de *spins* $\langle u, v\rangle$ que interagem devem ser visitados e, em seguida, se os dois *spins* sendo analisados estiverem no mesmo estado, isto é se $\zeta_u = \zeta_v$, então é calculada a probabilidade de ser estabelecido um **arco** (ou ligação) entre esses dois *spins*. Esta probabilidade é calculada a partir da equação (3.5) (página 54) do modelo de Potts, restringido-a à interação entre as partículas de interesse, digamos $p_{u,v}$.

3. Após calculada a probabilidade do passo anterior, é gerada aleatoriamente uma ocorrência de $w = U(\omega)$, com distribuição uniforme no intervalo $(0, 1)$, isto é, $U \sim \mathcal{U}(0, 1)$. Em seguida, é verificado se $w < p_{u,v}$, caso isto aconteça, será estabelecido uma arco representando uma ligação entre o par de *spins* u e v. Caso contrário, nenhum arco é estabelecido.

   Para ilustrar os passos 2 e 3, considere o exemplo da Figura 12 (página 55). Se inicialmente e ao acaso escolhermos o *spin* destacado da Figura 13 para iniciar o passo 2, teremos os *spins* destacados da Figura 14 (página 57) que satisfazem a condição $\zeta_u = \zeta_v$, ou seja, apresentam-se no mesmo estado (ver também Horta 2004).

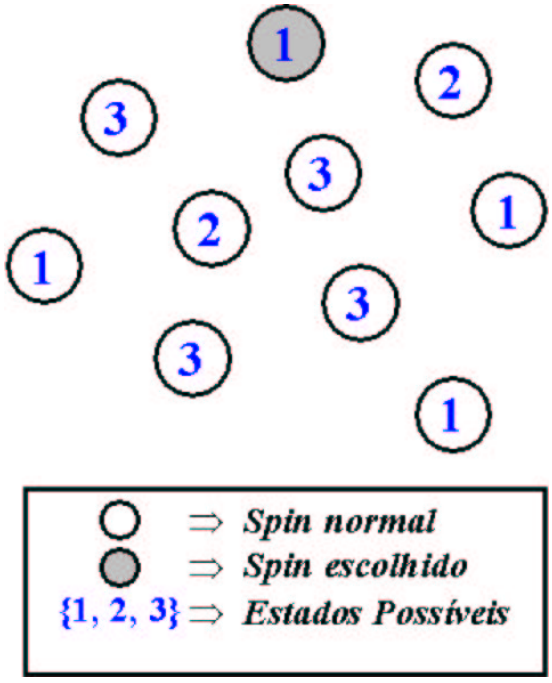


Figura 13: Escolha do *spin* inicial.

Uma vez que os *spins* satisfazem a codição de estarem no mesmo estado, são calculadas as probabilidades de formar arcos entre eles e, em seguida, são observadas ocorrências independentes da variavel aleatória $U \sim \mathcal{U}(0, 1)$, uma para cada par de partículas. Um arco será estabelecideo entre os pares de *spins* onde a ocorrência u for menor do que a respectiva probabilidade. Supondo, para o caso do exemplo dado, que a probabilidade obtida para o par de *spins* destacado na Figura 15 (página 58) seja 0.8 e que, aleatoriamente, o valor de w obtido seja

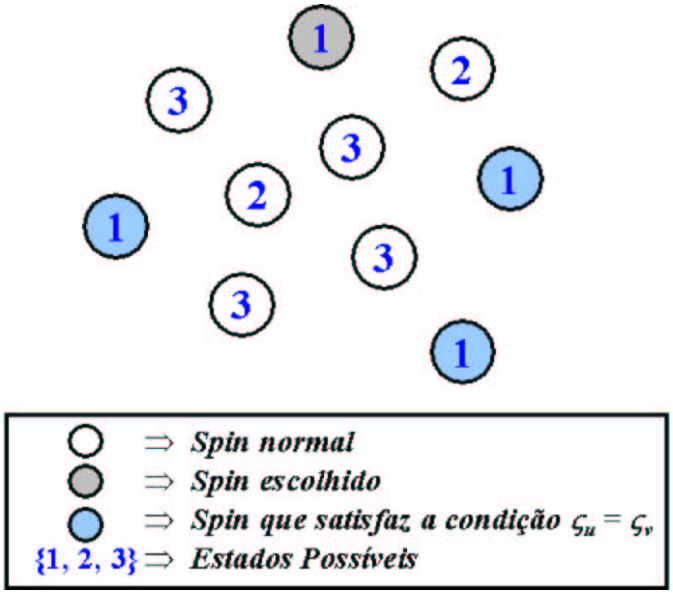


Figura 14: *Spins* que satisfazem a condição $\zeta_u = \zeta_v$.

0.6, de acordo com o passo 3 será estabelecida uma ligação (arco) entre o par.

4. Após os estabelecimentos de todas as ligações entre os pares de *spins* que satisfazem a condição de forma arco, o passo seguinte é a identificação dos grupos de *spins* de Swendsen-Wang (SW). Um grupo SW contém todos os *spins* que possuem uma ligação entre eles e, pelo já definido, somente os *spins* com os mesmos valores, ou seja, que estão no mesmo estado, poderão fazer parte do mesmo grupo SW. Uma possível configuração e a representação de grupos de Swendsen-Wang podem ser vistas na Figura 16 (página 58).

5. A etapa final do procedimento consiste em escolher, para cada grupo de Swendsen-Wang, um estado uniforme e independentemente dentre o conjunto de estados possíveis. Com isso uma nova configuração é gerada. A Figura 17 (página 59) ilustra este último passo. Nela pode ser observado que um dos *spins* que estava no estado 1 e que fazia parte de um grupo SW, contendo outros dois *spins*, foi escolhido e o estado de todos que pertenciam a esse grupo foi substituído pelo novo estado escolhido aleatoriamente, no caso do exemplo o estado escolhido foi o 2.

6. Retorna-se ao passo 2 enquanto não for satisfeito algum critério de convergência, como por exemplo o total de iterações.

Os trabalhos citados anteriormente (Swendsen 1991, Swendsen & Wang 1987) mostram que, seja qual for a configuração inicial e para toda temperatura T, após um certo número de iterações a configuração observada pode ser considerada oriunda do modelo de Potts.

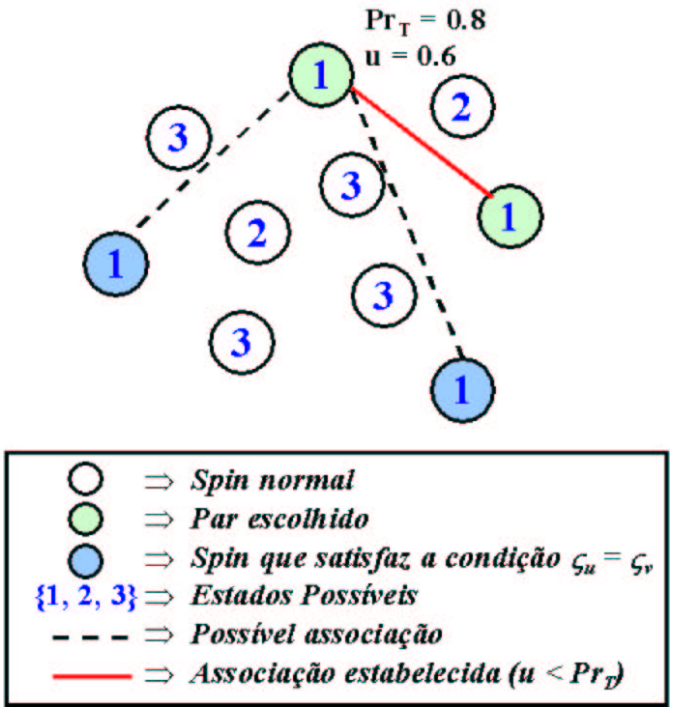


Figura 15: Estabelecimento de arco entre *spins* - Passo 3.

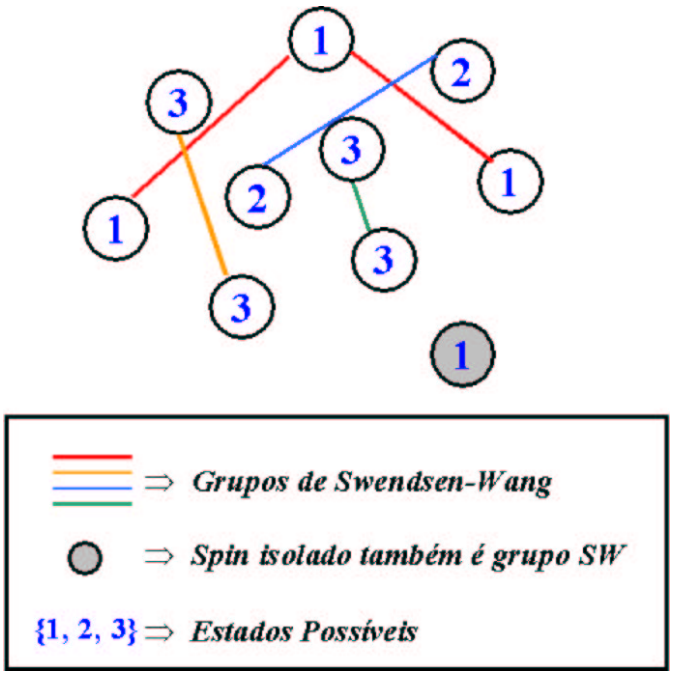


Figura 16: Configuração com grupos de Swendsen-Wang.

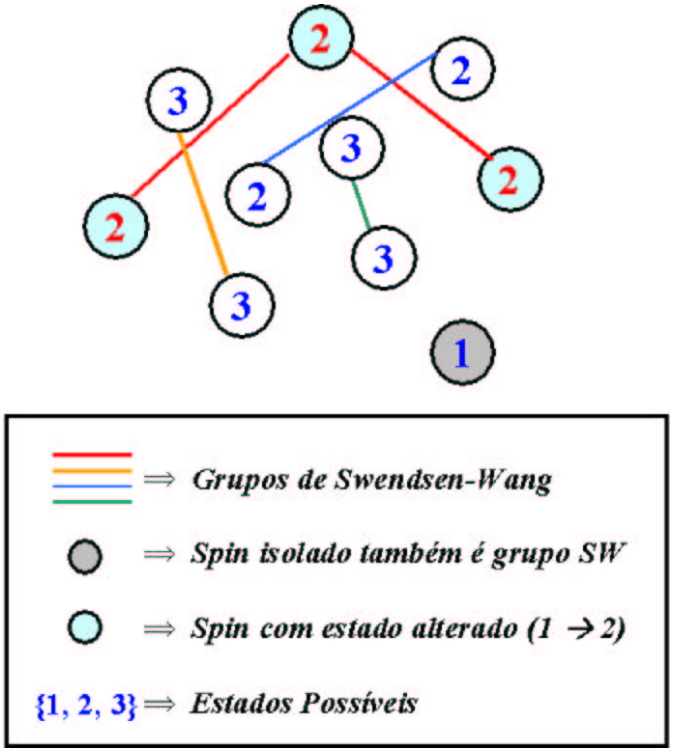


Figura 17: Nova configuração gerada.

3.5 Visão Geral do Algoritmo de Agrupamento de Dados

As seções anteriores discutiram e detalharam os métodos que servem de base ao algoritmo de agrupamento de dados discutido nesta seção. Esses métodos foram: o modelo de Potts não homogêneo (ver Seção 3.3, página 54) e o método de Swendsen-Wang (ver Seção 3.4, página 54). A seguir é exibido de uma maneira geral o funcionamente deste novo algoritmo.

Considere que o conjunto de dados consiste de N indivíduos, especificados por vetores d-dimensionais. Podemos considerar ainda para cada par de indivíduos i, j que sua dissimilaridade é denotada $d_{i,j}$. Será observada a evolução do modelo de Potts formado por N partículas (cada uma representando um dos N indivíduos). As partículas i e j interagem com uma força $J_{i,j}$ que é inversamente proporcional à dissimilaridade $d_{i,j}$. Reescrevendo a equação (3.5) (página 54), o Hamiltoniano (a energia) do modelo de Potts assim definido é dado por

$$\Pr_T(\mathbf{x} = \zeta) = 1 - \frac{1}{Z_T} \exp\Big\{-\frac{1}{T} \sum_{u,v} J_{i,j} \mathbb{I}_{u,v}\Big\}.$$

Dado que as interações $J_{i,j}$ são todas não negativas e observando o somatório da equação, a configuração mais provável, isto é, de menor energia é aquela que atribui o mesmo estado a todas as partículas. Analogamente, a configuração menos provável é aquela que atribui um estado diferente para cada partícula. Nenhum desses estados é interessante para o problema em questão, uma vez que a importância maior é

dada nas fases intermediárias, onde espera-se que indivíduos (partículas) similares se agrupem entre si e, simultaneamente, se diferenciem de indivíduos distintos.

De outra forma é facilmente observado que a temperaturas muito baixas o modelo de Potts ficará muito frequentemente "congelado", isto é, com a maioria das partículas no mesmo estado (ver Horta 2004). A temperaturas muito altas, o mesmo modelo frequentemente exibirá estados onde a maioria das partículas adota estados diferentes.

Em termos de agrupamento, a primeira situação corresponde àquela onde todos os indivíduos foram associados a um único grupo, enquanto na última cada indivíduo equivale a um grupo isolado. Como comentado anteriormente, nenhum desses dois extremos é interessante na maioria das aplicações práticas.

A proposta de Blatt et al. (1996), refinada nos artigos posteriores (Blatt et al. 1997, Domany 1999, Domany et al. 1999, Wiseman et al. 1998), consiste em adotar uma seqüência de temperaturas entre uma temperatura muito baixa e uma temperatura muito alta. Para cada uma dessas temperaturas, é observada uma seqüência de ocorrências do modelo de Potts, com o objetivo de estimar a correlação entre duas partículas. Duas observações serão consideradas no mesmo grupo se a correlação entre os respectivos estados for superior a um limiar previamente estabelecido.

A dinâmica de Swendsen-Wang é empregada para gerar essas seqüências de ocorrências do modelo de Potts. A vantagem de empregar esta dinâmica em relação a usar as dinâmicas de Metropolis (Metropolis & Ulam 1949, Metropolis et al. 1953) ou o *Gibbs sampler* (Geman & Geman 1984) reside nas seguintes propriedades:

1. esta dinâmica converge rapidamente seja qual for a configuração inicial;
2. uma vez atingida a convergência, são necessárias poucas iterações entre duas configurações para que as mesmas possam ser consideradas independentes (a independência é essencial para poder fazer estimativas confiáveis de, por exemplo, a correlação entre estados);
3. é possível implementá-la de forma eficiente com linguagens de programação usuais.

Este algoritmo pode ser dividido em três etapas, cujos passos são detalhados a seguir e em Horta (2004):

1. Construir uma analogia física com o problema de *spins* de Potts
   (a) Associar a cada ponto v_i um *spin* de Potts caracterizado por um de q estados possíveis $s_i = 1, 2, \ldots, q$.
   (b) Identificar os vizinhos de cada ponto v_i de acordo com um critério pré-selecionado; no problema de agrupamento de dados todos os indivíduos são potencialmente vizinhos.

   (c) Calcular a interação J_{ij} entre os vizinhos v_i e v_j de forma que seja inversamente proporcional à dissimilaridade entre esses indivíduos.

2. Localizar a fase superparamagnética, ou para vários valores da temperatura entre uma temperatura baixa e uma temperatura alta

   (a) Estimar a magnetização média para diferentes temperaturas.

   (b) Usar a susceptibilidade para identificar a fase superparamagnética.

3. Determinar dentro da fase superparamagnética as seguintes medidas:

   (a) Medir a correlação spin-spin para todos os pontos vizinhos v_i,v_j.

   (b) Construir os grupos de dados.

As seções seguintes detalham essas etapas do algoritmo de agrupamento.

3.5.1 Analogia física com o problema de *spins* de Potts

Nesta fase é especificado o Hamiltoniano (ver equação (3.5), página 54) para realizar a analogia física com o problema de *spins* de Potts.

Uma vez definido q, o número de estados possíveis que um *spin* de Potts pode assumir, é atribuído a cada ponto de dado um *spin* de Potts, através da escolha de um dentre os q valores possíveis com probabilidade q^{-1} independentemente das outras posições.

Uma característica importante a ser definida é o conceito de "vizinhança", utilizada na execução do algoritmo para auxiliar na geração dos grupos de dados. Neste projeto, onde o foco principal é o agrupamento de "dados", todos os indivíduos são vizinhos pontenciais e interagem entre si. Já no caso específico de segmentação de imagens, como pode ser visto em Horta (2004), só são consideradas as interações entre posições próximas espacialmente.

Por fim é fornecida a dependência funcional da força da interação J_{ij} na distância entre os *spins* vizinhos.

Nesta seção são discutidas as possíveis escolhas para esses atributos do Hamiltoniano e suas influências na performance do algoritmo. Com base nessas influências, uma observação importante deve ser feita. O algoritmo considerado é suficientemente robusto para produzir resutados bons para uma grande classe de especificações, ou seja, não há uma especificação "ótima" em detrimento de outras especificações. O algoritmo irá produzir bons resultados sempre que uma escolha razoável for feita, e a faixa de escolhas razoáveis é muito ampla.

3.5.1.1 Associação dos *Spins* de Potts a cada ponto v_i

Como comentado em seções anteriores, é necessária inicialmente e ao longo das iterações do algoritmo a associação de *spins* Potts (estados) a cada ponto v_i do conjunto de dados. O número de estados possíveis, q, determina principalmente a sutileza das transições e as temperaturas nas quais elas ocorrem. Por outro lado, à medida que o valor de q aumenta torna-se necessário executar simulações muito longas com a finalidade de manter uma dada precisão estatística para os resultados. A partir de testes realizados por Domany et al. (1999), e verificados neste trabalho e em Horta (2004), foi possível concluir que a influência de q na classificação resultante é fraca, e por isso ao longo deste trabalho foi atribuído o valor $q = 20$ para todos os exemplos.

Uma vez escolhido o número de estados a ser utilizado no modelo de Potts, o passo seguinte é associar a cada spin um possível estado, como pode ser visto na Figura 18.

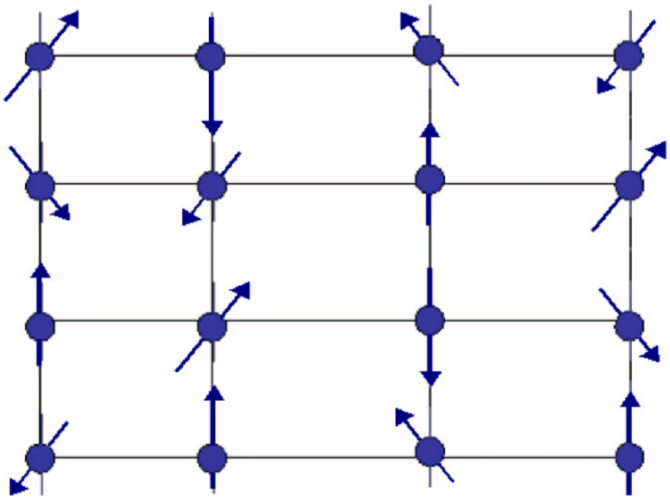


Figura 18: Associação de estados do modelo de Potts a *Spins*.

3.5.1.2 Identificação dos vizinhos

A identificação de vizinhos se faz necessária como uma maneira de solucionar ou amenizar o problema de decremento da eficiência computacional do algoritmo sendo utilizado. Ao mesmo tempo que ela surge como um meio solucionador, ela também cria um novo problema, que é a descoberta dos vizinhos. Este último problema poderia ser solucionado se todos os indivíduos de um conjunto de dados fossem vizinhos potenciais, ou seja, a necessidade para a identificação dos vizinhos do ponto x_i poderia ser eliminada se todos os pares i, j de *spins* de Potts interagissem com cada outro via uma interação de curta distância J_{ij} = $f(d_{ij})$, que decresce exponencialmente com a distância entre os dois pontos de dados. As fases e as propriedades do modelo não serão fortemente afetadas pela escolha da forma precisa de f. Aplicando essa solução foi verificado que computacionalmente esse decrescimento exponencial exi-

giria um grande gasto, então foi verificado que esse problema seria resolvido se fossem mantidas somente as interações de um *spin* com um número limitado de vizinhos, e atribuído a todas as outras J_{ij} igual a zero. Uma interação $J_{ij} = 0$ é equivalente a afirmar que a similaridade entre as observações i e j é nula, ou que a sua dissimilaridade é infinita; neste caso essas observações jamais serão agrupadas, a não ser que todas as observações sejam colocadas na mesma classe. Dessa maneira, é possível concluir que a identificação de vizinhos é uma solução mais viável do que o problema criado por sua geração.

Uma vez que os dados não formam uma grade regular, é preciso fornecer alguma definição razoável para "vizinhos". Sendo assim, para dimensões pequenas, onde $d \leq 3$, foram utilizadas características da triangularização sobre estruturas de grafos em agrupamentos de dados (Ahuja 1982). Os conjuntos de dados usados nos nossos exemplos possuem grandes dimensões, logo, maior atenção é dada a dimensões onde $d > 3$.

Para grandes dimensões, é usado o valor de vizinhança mútua, onde v_i e v_j têm um valor de vizinhança mútua K, se e somente se v_i é um dos K vizinhos mais próximos de v_j e v_j é um dos K vizinhos mais próximos de v_i. O valor de K foi escolhido de tal maneira que as interações conectem todos os pontos de dados a um grafo conectado. Claramente K crescre com a dimensão. Por motivos computacionais, tornou-se conveniente, em casos onde a dimensão é muito alta ($d > 100$, por exemplo) fixar $K = 10$.

3.5.1.3 Cálculo das interações locais

Para ter um modelo com as propriedades físicas de um imã granular não-homogêneo, é preciso obter fortes interações entre *spins* que correspondam a dados de uma região de alta densidade e fracas interações entre *spins* que estejam em regiões de baixa densidade. Para este fim, e em comum com outros métodos locais, será suposto que existe uma quantidade local a, definida pelas regiões de alta densidade e menor do que a distância típica entre pontos nas regiões de baixa densidade. Este valor a é a quantidade característica sobre a qual nossas interações de curta distância decaem. Uma boa escolha para calcular J_{ij} é a seguinte:

$$J_{i,j} = \begin{cases} \frac{1}{\hat{K}} \exp\left(\frac{-d_{ij}^2}{2a^2}\right) & \text{se } v_i \text{ e } v_j \text{ forem vizinhos} \\ 0 & \text{caso contrário,} \end{cases}$$

onde, segundo Domany et al. (1999), a quantidade a será a média de todas as distâncias d_{ij} entre a vizinhança dos pares v_i e v_j, e $\hat{K}$ é o número médio de vizinhos. Todos os detalhes vistos até agora podem ser facilmente implementados quando ao invés do fornecimento de x_i para todos os dados, for obtida uma matriz $N \times N$ de dissimilaridades d_{ij}.

3.5.2 Localização da fase superparamagnética

Para identificar uma fase superparamagnética é preciso inicialmente estimar um parâmentro de ordem do sistema, no caso deste projeto a magnetização, e em seguida calcular uma medida que indica as mudanças de fases ocorridas no sistema. Os vários intervalos de temperatura nos quais o sistema se auto-organiza em diferentes partições para grupos são identificados através da medida de susceptibilidade χ, que é função da temperatura. A susceptibilidade é a variância da magnetização média para cada temperatura, e deve ser estimada a partir das simulações Monte Carlo.

O método de Monte Carlo foi utilizado com o seguinte procedimento, que inclui a dinâmica de Swendsen-Wang e que também pode ser consultado em Horta (2004):

1. Escolher o número de iterações M a ser executada.
2. Gerar uma configuração inicial através da atribuição de um valor aleatório a cada *spin*.
3. Associar um arco entre os pontos vizinhos v_i e v_j com probabilidade p_{ij} (ver equação (3.5), página 54).
4. Encontrar os subgrafos conectados, os grupos de SW.
5. Atribuir novos valores aleatórios aos *spins* (aos *spins* que pertencem ao mesmo grupo SW são associados o mesmo valor). Isto é a nova configuração do sistema.
6. Calcular o valor assumido pelas quantidades físicas de interesse na nova configuração de *spin*.
7. Voltar para o passo 3 a menos que o número de iterações, M, tenha sido alcançado.
8. Calcular a magnetização média, dada pelo tamanho do maior grupo dividido pelo número total de observações.

A fase superparamagnética pode conter muitos e diferentes subgrafos com propriedades diferentes. A susceptibilidade χ foi medida para diferentes temperaturas com o objetivo de localizar diferentes regimes. A meta é identificar as temperaturas nas quais o sistema muda sua estrutura.

Observando a Figura 19 (página 65) é preciso destacar duas características básicas da susceptibilidade que possuem uma grande importância no trabalho.

A primeira é um pico na susceptibilidade, que sinaliza uma transição da fase ferromagnética para superparamagnética, na qual um grupo grande é quebrado em pequenos grupos. A segunda característica é uma queda brusca da susceptibilidade, correspondendo a uma transição da fase superparamagnética para a paramagnética, na qual um ou mais grupos grandes são quebrados em muitos grupos pequenos.

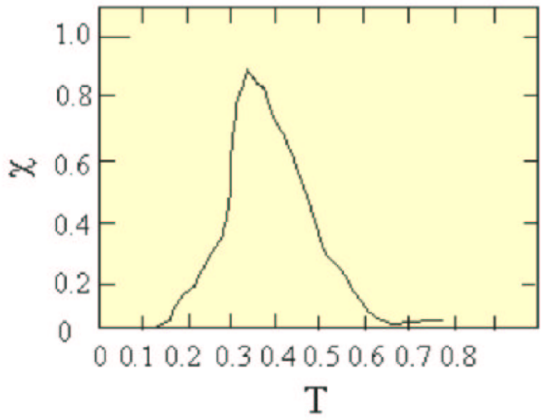


Figura 19: Exemplo de uma curva de susceptibilidade.

Como a susceptibilidade é obtida para cada temperatura, pode-se concluir que com os valores obtidos junto ao pico e a queda da curva da Figura 19 é que a identificação da fase superparamagnética é feita.

3.5.3 Determinação de medidas na fase superparamagnética

Esta fase tem início após a identificação da fase superparamagnética e de suas subfases. O primeiro passo desta fase é a seleção de uma temperatura em cada região de interesse. Em seguida cada subfase caracteriza um tipo particular de partição dos dados, de tal forma que novos grupos estarão sendo formados ou divididos. Outro ponto a ser considerado é que dentro de uma determinada fase a temperatura é variada e, dessa maneira, espera-se somente que a expansão ou o encolhimento dos grupos já existentes, mudando somente a classificação dos pontos nas fronteiras (limites) dos grupos.

3.5.3.1 A Correlação *Spin-Spin*

Como comentado anteriormente, uma medida de correlação é calculada para determinar se dois *spins* pertencem a um mesmo grupo. Assim podemos representá-la por uma função G_{ij}, que é obtida para dois indivíduos v_i e v_j.

O método Monte Carlo permite calcular estimadores da função de correlação *spin-spin*, em particular pelo fato de empregar a dinâmica de Swendsen-Wang que, como já mencionado, é de rápida execução.

Um estimador simples para a correlação entre os pontos i e j pode ser obtido a partir de um estimador da probabilidade desses indivíduos estarem no mesmo grupo (ver também Horta 2004). Este estimador pode ser a média das funções indicadoras calculada sobre M ocorrências (idealmente independentes) do modelo de

Potts, isto é, estimaremos G_{ij} calculando $\widehat{C}_{ij} = \sum_{\ell=1}^{M} \mathbb{I}_{i,j}(\ell)/M$, com

$$\mathbb{I}_{i,j} = \begin{cases} 1 & \text{se } v_i \text{ e } v_j \text{ pertencem ao mesmo grupo} \\ 0 & \text{caso contrário,} \end{cases}$$

que é um estimador da probabilidade de encontrar os indivíduos v_i e v_j no mesmo grupo, ℓ denota a iteração, e finalmente computando

$$\widehat{G}_{ij} = \frac{(q-1)\widehat{C}_{ij} + 1}{q},$$

que é usado como estimador da correlação G_{ij}.

3.5.3.2 Os grupos de dados

Os grupos são identificados em três passos:

1. Construir o centro dos grupos usando um procedimento de limiarização, ou seja, é estabelecido um valor, 0.5 por exemplo, onde se $\widehat{G}_{ij} > 0.5$, uma ligação (arco) é estabelecida entre os pontos de dados vizinhos v_i e v_j. O grafo conectado resultante depende fracamente do valor escolhido para este procedimento (0.5), desde que este limiar esteja no intervalo $(q^{-1}, 1 - 2q^{-1})$. A razão para isto é que a distribuição das correlações entre os dois *spins* vizinhos chega ao ponto máximo fortemente nesses dois valores e é muito pequena entre eles. Em altas temperaturas, por exemplo, onde o sistema é paramagnético ou ainda desordenado, a função de correlação G_{ij} decai até q^{-1} quanto a distância entre os pontos v_i e v_j é grande; esta é a probabilidade de encontrar dois *spins* de Potts completamente independentes no mesmo estado.

2. Capturar pontos que estejam situados na periferia dos grupos através da ligação de cada ponto v_i a seu vizinho v_j de correlação máxima estimada $\widehat{G}_{ij}$. Pode acontecer que pontos v_i e v_j já tenham sido ligados no passo anterior.

3. Os grupos de dados são identificados como os componentes ligados dos grafos obtidos nos passos 1 e 2.

Este capítulo descreveu além da nova técnica de agrupamento de dados utilizada neste projeto, conceitos utilizados por esta técnica e necessários a um maior entendimento sobre o algoritmo. A próxima seção detalha o domínio da aplicação, ou seja, expõe o problema em agrupar conjuntos de dados grandes e complexos e maneiras de contornar este problema.

4 O Domínio da Aplicação

O progresso da tecnologia permitiu o armazenamento de um grande volume de dados que anteriormente era impossível guardar. Isto aconteceu graças ao avanço do *hardware* de computadores, que foi o grande responsável pelo espaço disponível, cada vez maior, para armazenamento desses dados.

Trabalhar com grandes volumes de dados atualmente é um desafio com o qual diversos profissionais de várias áreas frequentemente se deparam. Para os seres humanos, interpretar esses grandes volumes sob a forma textual ou numérica é uma tarefa impossível de ser realizada sem a ajuda de uma máquina. Percebe-se também que quando grandes conjuntos de dados são apresentados na forma gráfica, a percepção humana torna-se muito boa.

Quando esses grandes volumes de dados são apresentados graficamente a seres humanos existe ainda uma necessidade de atingir uma relação entre os dados, ou seja, identificar os dados que apresentam características semelhantes, exibindo ao ser humano uma lógica no conjunto de informações. Uma forma de obter tal resultado é através de agrupamentos, que buscam identificar em quais níveis os dados se relacionam e de que maneira.

Mas até mesmo nesses agrupamentos existem dificuldades que devem ser consideradas. Quando um conjunto de dados relativamente pequeno é agrupado em diversos níveis, é simples para um ser humano identificar os passos dos agrupamentos, ou seja, extrair as informações necessárias que resultam desses agrupamentos, seja na forma textual, numérica ou gráfica. Se o conjunto de dados a ser agrupado apresentar uma dimensão consideravelmente grande, essa mesma capacidade humana de extrair visualmente as informações relevantes torna-se inútil, uma vez que, mesmo graficamente, a quantidade de dados exibidos de uma só vez supera a expectativa humana.

Um dos principais objetivos deste trabalho é integrar a característica humana de análise de dados em formato gráfico com os recursos computacionais de armazenamento de grandes volumes de dados, proporcionando desta forma um ganho na interpretação dos resultados e auxiliando o usuário nessa interpretação.

As próximas seções descrevem o problema em agrupar dados complexos e com grandes volumes, além de soluções e representações para tal problema, e faz uma

comparação entre alguns trabalhos desenvolvidos na área.

4.1 Agrupamento no domínio de aplicação

Trabalhar com grandes volumes de dados é uma tarefa árdua que exige, acima de tudo, paciência. Isto porque muitas vezes a quantidade de informações que se deseja extrair de tais conjuntos de dados é grande e envolve o relacionamento de várias de suas variáveis. Uma ilustração de tal afirmação pode ser visualizada nas Tabelas 7 e 8.

Índice	Pessoa	Peso	Altura
1	Maria	35	190
2	André	40	190
3	Wagner	35	160
4	Sandra	40	160

Tabela 7: Conjunto de dados com 4 indivíduos

Índice	Pessoa	Peso	Altura
1	Maria	35	190
2	André	40	190
3	Wagner	35	160
4	Sandra	40	160
5	João	70	170
6	Eduardo	65	175
7	Graça	72	168
8	Josefa	80	190
9	Jonathan	70	169
10	Paulo	66	163
11	Ricardo	78	180
12	Cristiano	80	195
13	Roan	88	189
14	Carmem	75	176
15	Michelle	61	159
16	Michael	62	160
17	Robson	78	180
18	Carla	92	196
19	Jean	73	174
20	Rômulo	90	198
⋮	⋮	⋮	⋮
4000	Erick	65	175

Tabela 8: Conjunto de dados com 4000 indivíduos

Para encontrar os indivíduos mais altos na Tabela 7 através de uma observação humana, a procura da informação é bastante simples, uma vez que a tabela apresenta apenas quatro indivíduos, logo os indivíduos mais altos são Maria e André. Para realizar a mesma busca na Tabela 8, a operação se tornaria uma tarefa cansativa

e na maioria das vezes inexata, uma vez que humanos erram. Descobrir dentro de um conjunto contendo 4000 indivíduos os que são mais altos, ainda é uma tarefa pouco complicada, se comparada com a tarefa de descobrir quais indivíduos estão dentro do peso e altura ideais através da inclusão de mais uma variável, como idade por exemplo. Para realizar essa tarefa seria necessário calcular para cada indivíduo o seu `IMC` (Índice de Massa Compórea) e fazer uma relação com a idade do mesmo para descobrir se ele é obeso ou não.

As tarefas citadas anteriormente seriam facilmente realizadas com a ajuda de uma máquina. Ao utilizar uma máquina para executar tais tarefas surgem alguns pontos que precisam ser discutidos. Dois deles são:

- a escolha de uma técnica para agrupar o conjunto de dados, de forma que as informações que são procuradas sejam realmente obtidas;
- a escolha de uma forma de representação para os resultados obtidos, de tal maneira que todas as informações extraídas possam ser visualizadas.

Escolher a melhor ténica de agrupamento de dados é um problema que está muito ligado ao próprio conjunto de dados. Portanto, não é uma operação genérica. Na literatura uma grande variedade de técnicas foi proposta para diferentes aplicações e diferentes tamanhos dos conjuntos de dados. A aplicação de uma dessas técnicas a um conjunto de dados tem como objetivo, assumindo que o conjunto de dados oferece uma certa tendência para agrupamento, a descoberta de suas partições naturais (Halkidi & Vazirgiannis 2001). Entretanto, o processo de agrupamento é visto como um processo não supervisionado, uma vez que não existem classes predefinidas e nenhum exemplo que poderia mostrar que tipo de relações desejáveis poderiam ser válidas dentro dos dados.

Esses vários algoritmos de agrupamento são baseados em alguns parâmetros para definir o particionamento de um conjunto de dados. Como uma consequência, eles podem se comportar de maneiras diferentes dependendo de:

- as características do conjunto de dados (geometria e distribuição da densidade dos grupos) e
- os valores dos parâmetros de entrada.

Os algoritmos que realizam agrupamentos por partição (ver Seção 2.2, página 33), como o K-médias (ver Seção 2.2.1, página 33) por exemplo, não manipulam bem dados que se distanciam muito do padrão do conjunto de dados a que ele pertence, bem como não são adequados para descobrir grupos com formas não convexas. Além disso, eles são baseados em certos parâmetros para particionar o conjunto de dados. Eles precisam especificar o número de grupos em progresso. Já as técnicas

hierárquicas (ver Seção 2.3, página 40) procedem sucessivamente à fusão de grupos pequenos, formando grupos maiores, ou à divisão de grupos grandes, formando grupos menores. Os resultados destas ténicas são árvores de grupos. Dependendo do nível no qual o corte da árvore for realizado, diferentes agrupamentos dos dados serão obtidos.

Outras técnicas que podem ser consideradas também são as baseadas em densidade e em grade (ver Seção 1.1, página 19), que adequadamente manuseiam coleções de pontos com formas arbitrárias (elipse, espiral, cilindro, entre outras.) e também grupos de diferentes tamanhos. Além disso, elas podem eficientemente separar indivíduos que são muito diferentes do restante do conjunto de dados. Apesar de todas essas vantagens, a maioria dessas téncnicas são sensíveis a alguns parâmetros de entrada de forma que é necessária uma seleção de valores bastante cuidadosa.

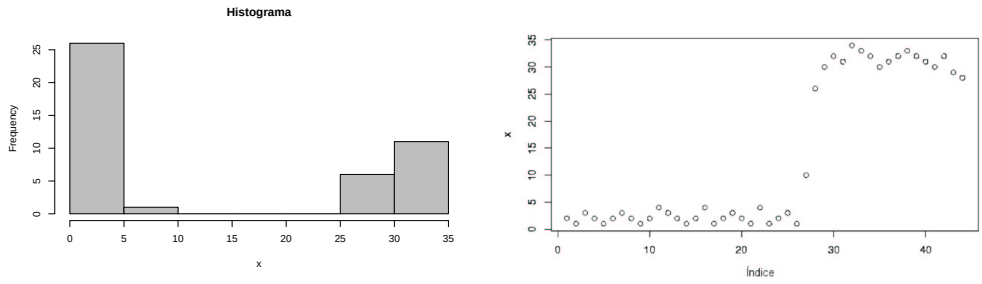
A técnica de agrupamento de dados apresentada no Capítulo 3 (página 50) é um dos pontos principais deste trabalho e é indicada para grandes e complexos conjuntos de dados. Segundo Blatt et al. (1997), os resultados obtidos com essa técnica, quando comparados a outras como `junção simples` e `junção completa`, foram superiores, o que leva a acreditar que a técnica `SPC` é a mais indicada para tais conjuntos de dados. A análise dos resultados para essa ténica será melhor descrita no Capítulo 6.2 (página 116).

Dessa discussão pode-se concluir que todas as ténicas de agrupamento não são eficientes para todas as aplicações, e isto é o porquê de existir uma grande diversidade de algoritmos de agrupamento. Dependendo do critério de agrupamento e da habilidade de manusear requerimentos especiais de uma aplicação, uma ténica de agrupamento pode ser considerada mais eficiente em um certo contexto (dados espaciais, medicina, entre outros).

Escolhida uma técnica para realizar o agrupamento dos dados e obtidos os resultados, estes precisam ser apresentados ao usuário de maneira objetiva e completa. Para representar os agrupamentos obtidos, na maioria das vezes são utilizados gráficos diversos, mas as formas textuais e numéricas também são muito utilizadas. A maneira de representação ideal dos agrupamentos, principalmente para grandes conjuntos de dados, seria a união das formas textuais, numéricas e gráficas, escolhendo-se, principalemente para a forma gráfica, as melhores ferramentas que fazem parte delas.

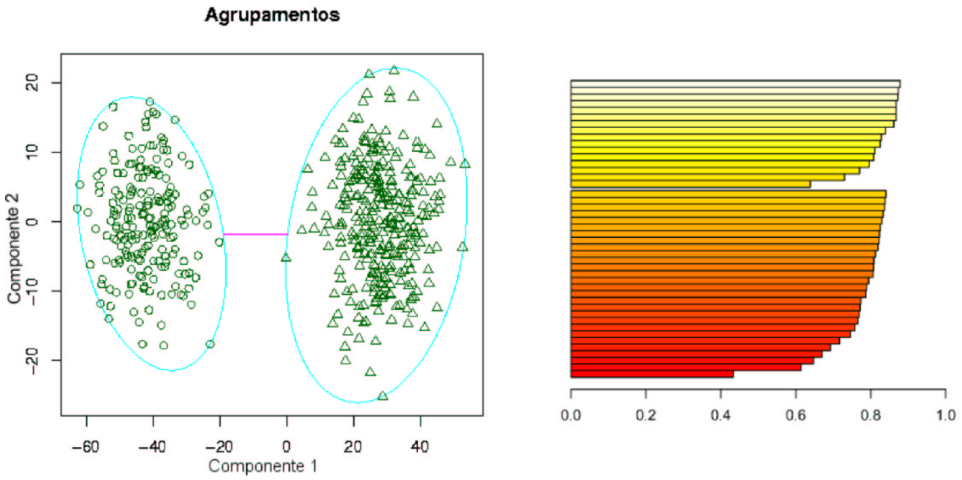
Algumas das formas gráficas mais usadas para representar conjuntos de dados são histogramas em uma, duas ou três dimensões, scatterplots e gráficos de perspectiva. A Figura 20 (página 71) mostra algumas dessas formas.

Para representar agrupamentos de dados existem formas gráficas mais adequadas e que exibem os grupos obtidos de maneira simples. Exemplos dessas representações gráficas estão na Figura 21 (página 71), onde é apresentada a formação de grupos de um determinado conjunto de dados.

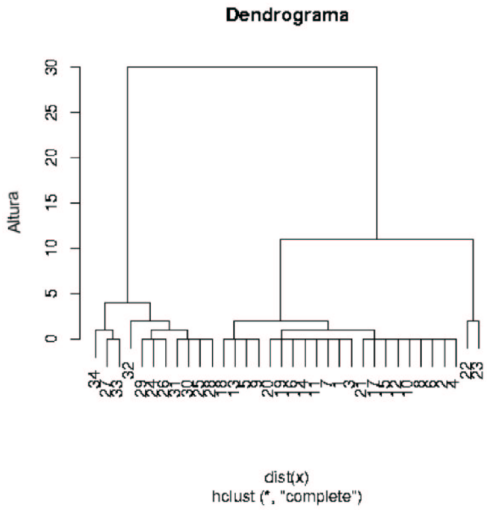


(a) Histograma (b) ScatterPlot

Figura 20: Representações gráficas de conjuntos de dados.



(a) Gráfico de componentes (b) Gráfico de Silhueta



(c) Dendrograma

Figura 21: Representações gráficas para agrupamentos de dados.

Para pequenos conjuntos de dados todas as formas gráficas apresentadas exibem ótimos resultados, ou seja, exibem grupos visíveis e distintos. Para grandes conjuntos de dados, os mesmos resultados não serão obtidos, uma vez que a representação de uma grande quantidade de dados, mesmo graficamente, na maioria das vezes pode gerar uma confusão visual, dependendo do tipo de gráfico utilizado. Este problema é comum e inevitável quando se tem pouco espaço para representação e um grande volume de informação.

A ferramenta apresentada no Capítulo 5 (página 76) foi desenvolvida com o objetivo de amenizar o problema citado. Ela exibe os agrupamentos para o usuário através de gráficos como o da Figura 21(c) (página 71). Esse tipo de gráfico foi escolhido por exibir um nível de detalhamento dos agrupamentos elevado, uma vez que ele mostra os passos desses agrupamentos em forma de uma árvore. Juntamente com os dendrogramas, a aplicação do Capítulo 5 (página 76) apresenta os dados ao usuário na forma textual e numérica, para que juntas permitam uma interpretação fácil e correta ao mesmo. Além dessas formas de representação, a aplicação permite ao usuário uma interação simples com o dendrograma, o que valoriza bastante a sua interface. O Capítulo 5 descreve detalhadamente todas essas opções.

A próxima seção aborda trabalhos relacionados com o tema de representação dos agrupamentos, mas especificamente, trabalhos que utilizam dendrogramas e possuem características semelhantes às da aplicação apresentada neste projeto.

4.2 Trabalhos Relacionados

A ferramenta desenvolvida neste projeto e detalhada no próximo capítulo tem como objetivos promover a utilização da técnica SPC e proporcionar uma melhor visualização e compreensão dos resultados obtidos a partir dela.

Dentre os recursos disponíveis ao usuário na ferramenta desenvolvida estão:

1. Geração de matriz de dissimilaridade, optando-se por uma de duas métricas, Minkowski ou Canberra;
2. Padronização do conjunto de dados;
3. Análise multivariada dos dados: conjunto de dados exibido sob a forma de brushplot e informações estatísticas, tais como: matriz de correlação, médias, mediana, quartis, variância, entre outras;
4. Execução da técnica SPC visualmente;
5. Exibição dos resultados sob a forma de dendrogramas interativos e legenda textual, além dos valores estatísticos já citados;
6. Armazenamento dos gráficos gerados em arquivos do tipo *postscript*.

Estes são os principais recursos da ferramenta desenvolvida e que são detalhados no Capítulo 5 (página 76).

Existem ferramentas que também realizam agrupamentos e possuem opções para exibição dos resultados de forma gráfica, tais como as ferramentas R (ver Seção 1.3.2, página 24) e HCE.

A ferramenta R permite que um determinado conjunto de dados possa ser agrupado utilizando-se uma das várias técnicas implementadas pela ferramenta. Praticamente todas as técnicas citadas neste documento estão implementadas no R. A técnica SPC não está implementada no R. Ela foi implementada apenas neste projeto e em Horta (2004).

Os agrupamentos resultantes no R podem ser exibidos sob a forma dos gráficos apresentados na Figura 21 (página 71). Essa ferramenta, apesar de possuir tantos recursos, entre eles a representação na forma de dendrograma e a realização de agrupamentos, não é tão simples de ser manipulada, uma vez que todos os seus recursos são acessados através da linha de comando da ferramenta. Isto porque o R além de ferramenta também é uma linguagem, que permite não só a execução de comandos mas também o desenvolvimento de programas.

Realizando uma comparação entre a ferramenta R e a desenvolvida neste projeto, existem pontos importantes que devem ser considerados e distinguidos. O primeiro deles é a facilidade de manuseio da ferramenta desenvolvida, que é totalmente voltada para o usuário, ou seja, a interface foi desenvolvida com o objetivo principal de minimizar ao máximo a digitação do usuário e permitir que ele realize praticamente todas as operações de forma visual. No R, como descrito anteriormente, tudo é realizado através de programação, através de linha de comando. Outro ponto interessante é a interação que a ferramenta desenvolvida permite que o usuário realize com o dendrograma. Mesmo não tendo outros tipos de gráficos disponíveis para a exibição dos agrupamentos, a ferramenta desenvolvida associa diversos recursos à representação do dendrograma, tais como os nós sensíveis ao clique do mouse (ver serção 5.5.2.3, página 106), que permitem uma visualização dos grupos individualmente e suas informações estatísticas, além de uma análise multivariada através de brushplots. Outras características relacionadas com os dendrogramas referem-se à exibição desses gráficos com base nas temperaturas e indivíduos escolhidos.

A ferramenta R também permite a interação do usuário com os dendrogramas, mas isto só é possível através de uma programação precisa e longa do próprio usuário, o que recai na mesma dificuldade. A tela dessa ferramenta é exibida na Figura 22 (página 74).

Outra ferramenta interessante é a HCE (*Hierarquical Clustering Explorer*), que foi desenvolvida na universidade de *Maryland* por Seo & Shneiderman (2002), com o objetivo de realizar agrupamentos sem a necessidade da especificação do número de grupos como parâmetro para os agrupamentos e permitindo que o próprio usuário de-

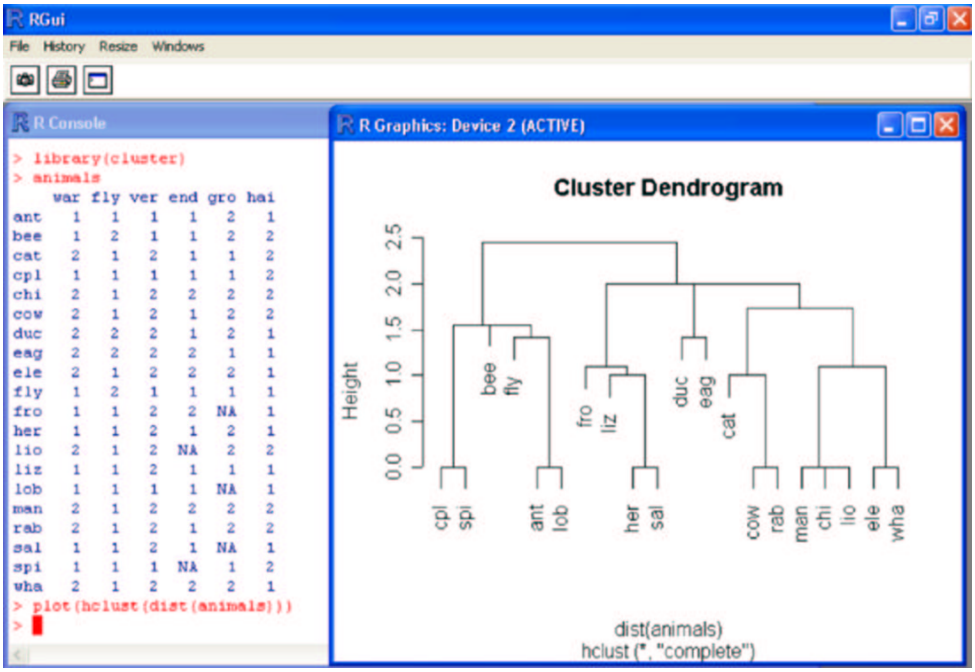


Figura 22: Ferramenta R.

termine o agrupamento natural de forma visual, através de dendrogramas e mosaicos de cores. Essa ferramenta (ver Figura 23, página 75) disponibiliza cinco técnicas de agrupamento hierárquicas para o usuário e alguns outros recursos tais como: visualização de scatterplots e interação do usuário com eles.

A ferramenta da Figura 23 (página 75), assim como a ferramenta desenvolvida neste projeto, é totalmente visual, ou seja, voltada para o usuário. Ela também permite a interação do usuário com os scatterplots, mas não permite com os dendrogramas. É uma ferramenta que tem como principal função realizar o agrupamento dos dados e exibí-los, sem maiores detalhes. Apesar de ser uma ferramenta visual, o que deveria auxiliar o seu usuário, ela possui uma interface confusa e em fase de manutenção.

Existem diversas outras ferramentas que utilizam dendrogramas para representar seus agrupamentos, mas em sua maioria elas não apresentam maiores detalhes.

O próximo capítulo descreve a ferramenta desenvolvida em detalhes, desde a sua análise até a implementação, bem como as alterações realizadas no algoritmo da técnica SPC necessárias à integração do programa à ferramenta.

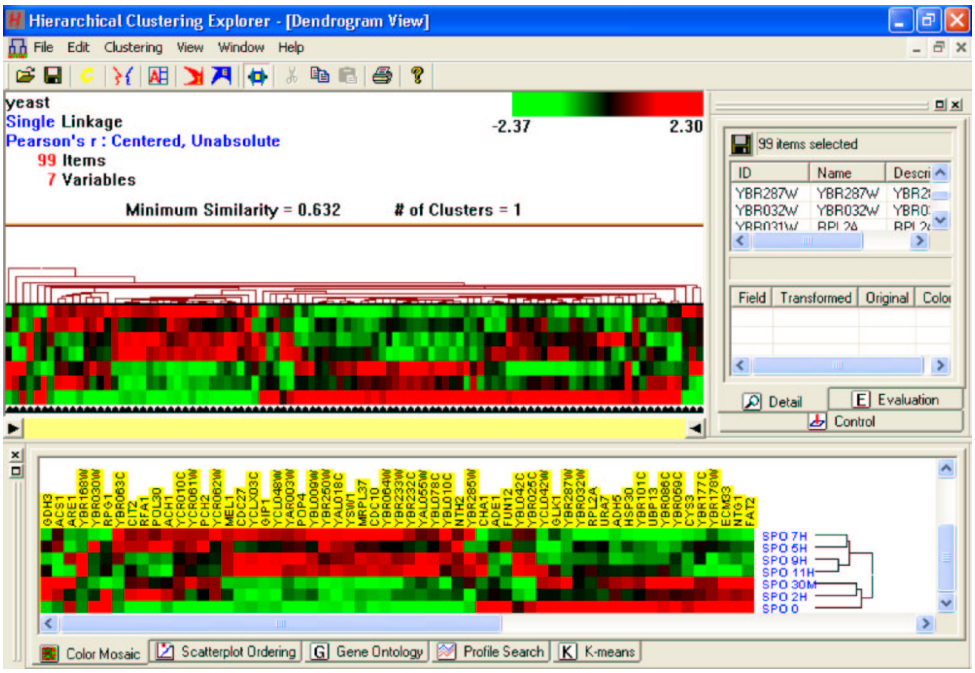


Figura 23: Ferramenta HCE.

5 SPC - Programa e Interface Desenvolvida

O programa SPC é a base dos trabalhos realizados nesta dissertação. Este capítulo descreve todos os procedimentos realizados por este programa. Inicialmente, o programa original é explicado através de seus arquivos de entrada e saída. Em seguida, são descritas as adaptações e melhorias realizadas visando a aplicação proposta e diante das análises feitas no programa original. Por fim, a interface desenvolvida na dissertação é apresentada.

5.1 O Programa SPC.EXE

SPC (Superparamagetic Clustering) é um programa que realiza o agrupamento de dados utilizando pra isto a técnica de agrupamento superparamagnético. Seu autor (ver Domany et al. 1999) procurou desenvolver uma aplicação que satisfizesse suas necessidades imediatas, ou seja, ele precisava apenas de algum método automatizado para executar o algoritmo de agrupamento superparamagnético em grandes conjuntos de dados, e esta era uma tarefa impossível de ser realizada manualmente, devido ao tamanho dos conjuntos de dados e à complexidade dos algoritmos utilizados. Sendo assim, ele desenvolveu um programa, em linguagem C, que executava o algoritmo superparamagnético e que salvava em arquivos, mais precisamente cinco, várias informações, inclusive os próprios agrupamentos.

Para um usuário comum de programas de agrupamentos de dados ou ainda de programas matemáticos ou estatísticos, as informações armazenadas em arquivos pelo SPC não faziam nenhum sentido, a não ser que uma pesquisa mais aprofundada fosse realizada nos dados obtidos, além disso o SPC era executado a partir de linha de comando, o que tornava seu uso ainda mais complicado. O objetivo do autor do SPC não era comercializá-lo ou distribuí-lo, uma vez que o grande objetivo de programas desses tipos é facilitar a "vida" do usuário, no sentido de permitir que o mesmo realize quase nenhum trabalho. Dessa forma, tínhamos uma ferramenta que realizava um bom trabalho e, da mesma maneira, gerava bons resultados, mas que era difícil de ser manipulada.

Tendo como base a qualidade da pesquisa e dos resultados obtidos por Domany

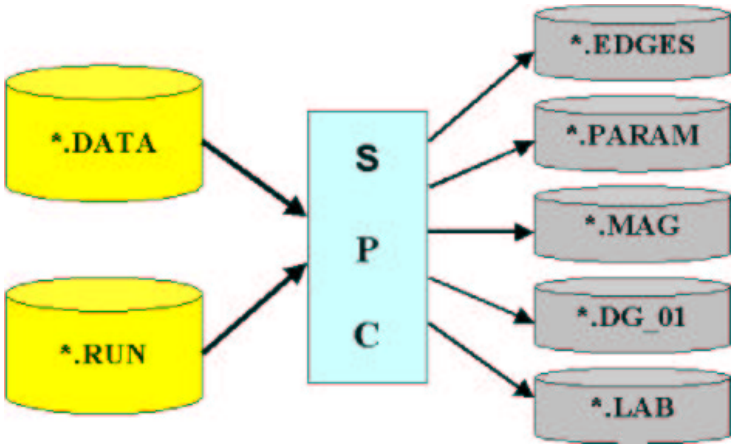


Figura 24: Estrutura original do SPC.EXE.

et al. (1999) e as dificuldades na compreensão desses resultados, descritas no parágrafo anterior, uma solução foi proposta para minimizar essas dificuldades e ter um maior aproveitamento da técnica implementada.

As seções seguintes descrevem a estrutura original do programa executável desenvolvido por Domany et al. (1999) e as otimizações realizadas nele para que a interface desenvolvida em IDL pudesse executar eficientemente. As próximas seções também exibem uma análise dos eventos do sistema e os resultados finais, ou seja, a interface implementada.

5.2 Estrutura dos Arquivos de Entrada

O SPC original recebia como parâmetros de entrada dois arquivos, dos quais um era um arquivo de dados, podendo conter uma matriz de dissimilaridade ou uma matriz de observações por variáveis, e o outro um arquivo contendo informações sobre os dados contidos no primeiro arquivo e outros parâmetros necessários à execução do programa. A estrutura do SPC é exibida na Figura 24.

Pode-se observar na Figura 24 que os dois arquivos de entrada do SPC apresentam as extensões .DATA e .RUN, respectivamente para o arquivo de dados e para o arquivo de informações gerais. O arquivo de dados, como dito antes, pode conter tanto uma matriz de dissimilaridade quanto uma matriz de observações por variáveis. Neste projeto, apenas matrizes de dissimilaridade são utilizadas, logo, quando algo for dito a respeito do arquivo de dados, esse tipo matriz deve ser considerado. A matriz de dissimilaridade é obtida a partir de um conjunto inicial de dados e é calculada com a ajuda de algum software comercial, por exemplo a plataforma R. O SPC não realiza o cálculo dessa matriz, ele apenas a utiliza. Para ilustrar essa parte inicial consideremos o conjunto de dados da Tabela 9 (página 78).

O conjunto de dados da Tabela 9 apresenta 28 espécies, onde são considerados o

	Corpo	**Cérebro**
Castor da Montanha	1.350	8.1
Vaca	465.000	423.0
Lobo cinza	36.330	119.5
Cabra	27.660	115.0
Preá	1.040	5.5
Dipliodocus	11700.000	50.0
Elefante asiático	2547.000	4603.0
Burro	187.100	419.0
Cavalo	521.000	655.0
Macaco Potar	10.000	115.0
Gato	3.300	25.6
Girafa	529.000	680.0
Gorila	207.000	406.0
Humano	62.000	13.20
Elefante africano	6654.000	5712.0
Tricerátopes	9400.000	70.0
Macaco da Índia	6.800	179.0
Canguru	35.000	56.0
Rato dourado da Índia	0.120	1.0
Rato	0.023	0.4
Coelho	2.500	12.1
Carneiro	55.500	175.0
Jaguar	100.000	157.0
Chipanzé	52.160	440.0
Ratazana	0.280	1.9
Braquiosauros	87000.000	154.5
Topeira	0.122	3.0
Porco	192.000	180.0

Tabela 9: Conjunto de dados - Animais.

peso médio do corpo dos animais em kilogramas e o peso médio do cérebro em gramas. Se desejarmos realizar agrupamentos com esse conjunto de dados, usando para isto o programa SPC, teremos que passar como parâmetro a matriz de dissimilaridade do conjunto de dados e não o próprio conjunto. Uma matriz de dissimilaridade parcial para este conjunto de dados é apresentada na Tabela 10.

	Castor da Montanha	Vaca	Lobo cinza	Cabra
Castor da Montanha	**0.000**	622.184	116.763	110.090
Vaca	622.184	**0.000**	525.233	534.911
Lobo cinza	116.763	525.233	**0.000**	9.76826
Cabra	110.090	534.911	9.76826	**0.000**

Tabela 10: Matriz de Dissimilaridade Parcial para o conjunto de dados Animais.

A Tabela 10 é uma representação do arquivo de entrada `.DATA` do SPC. Esta é apenas uma tabela parcial representando os quatro primeiros animais do conjunto de dados original.

O segundo arquivo de entrada para o SPC possui a extensão .RUN e contém informações necessárias à execução do programa, além de informações sobre a matriz de dissimilaridade do arquivo .DATA (ver Tabela 11).

NumberOfPoints	28
DataFile	C:\ Animals\ Animals.data
Dimensions	0
MinTemp	0.00
MaxTemp	0.15
TempStep	0.005
OutFile	yout
SWCycles	4000
KNearestNeighbours	5
MSTree	|
DirectedGrowth	|
SaveSuscept	|
WriteLabels	|
DataIsMatrix	|
WriteCorFile	~

Tabela 11: Campos do arquivo RUN.

Alguns dos campos da Tabela 11 podem ser facilmente identificados, tais como NumberOfPoints, que representa o número de indivíduos presentes no conjunto de dados, neste exemplo são 28 animais. DataFile contém o local e o nome do arquivo contendo a matriz de dissimilaridade. Dimensions indica a dimensão dos vetores que descrevem os pontos ou indivíduos, se seu valor é 0 (zero) então uma matriz de dissimilaridade é esperada, por enquanto este é o único valor considerado. MinTemp contém a temperatura inicial a partir da qual os agrupamentos serão realizados. MaxTemp contém a temperatura final até onde os agrupamentos serão realizados. TempStep representa o incremento da temperatura, ou seja, a partir da temperatura inicial o valor de TempStep será adicionado ao valor atual da temperatura até que a temperatura final seja atingida. OutFile é um prefixo para os arquivos de saída. SWCycles é o número de ciclos a ser considerado no algoritmo de Swendsen-Wang. KnearestNeighbours é o número máximo de vizinhos próximos usado no algoritmo KNN. MSTree é uma função que adiciona as bordas de ligação da árvore de spanning mínima. Assim como MSTree - DirectedGrowth, SaveSuscept, WriteLabels, DataIsMatrix e WriteCorFile - são funções que geram os arquivos de saída do SPC e que manipulam os dados obtidos do algoritmo superparamagnético.

5.3 Estrutura dos Arquivos de Saída

Os arquivos de saída do SPC são cinco, mas aqui consideraremos apenas dois, que contêm informações interessantes que serão utilizadas na construção da interface para o SPC. Esses dois arquivos apresentam as extensões $.DG\_01$ e .LAB, que

armazenam os dados resultantes dos agrupamentos, tais como os grupos e os indivíduos pertencentes a cada grupo.

O arquivo com extensão $.DG\_01$ apresenta a estrutura ilustrada na Tabela 12:

0	0.00000	0	1	28	0	0	0	0	0	0	0	0	0	0	0	0
1	0.00500	0	5	23	2	1	1	1	0	0	0	0	0	0	0	0
2	0.01000	0	5	23	2	1	1	1	0	0	0	0	0	0	0	0
3	0.01500	0	5	23	2	1	1	1	0	0	0	0	0	0	0	0

Tabela 12: Arquivo com extensão DG_01.

O primeiro campo da Tabela 12 indica o índice da temperatura sendo considerada e está variando de 0 a 3. O segundo campo indica o valor da temperatura utilizada e varia, neste caso, de 0.00000 a 0.01500. O terceiro campo, que contém apenas valores 0 (zero), indica o número mínimo de indivíduos que estão fazendo parte de algum grupo àquela temperatura, nesse caso existem grupos que não contêm indivíduos. O quarto campo, que possui os valores $\{1,5\}$, representa o total de grupos que possuem pelo menos um indivíduo. Os outros campos da tabela, 13 no total, são os grupos obtidos em cada temperatura. Então, analisando a Tabela 12 tem-se um total de 13 grupos, onde na temperatura 0.00000 o grupo 0 possui 28 indivíduos e os outros 12, nenhum. Na temperatura 0.00500, o grupo 0 possui 23 indivíduos, o grupo 1 possui 2 indivíduos, os grupos 2, 3 e 4 possuem 1 indivíduo cada, e os outros 8 grupos não contêm indivíduos. O resultado obtido na temperatura 0.00500 repete-se nas temperaturas 0.01000 e 0.01500, ou seja, não aconteceram mais agrupamentos após a temperatura 0.00500.

Os arquivos com extensão `.LAB` assemelham-se com os arquivos com extensão $.DG\_01$, pois eles também armazenam os agrupamentos, mas de um ponto de vista diferente. A Tabela 13 ilustra este ponto de vista.

0	0.00000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0.00500	0	0	0	0	0	1	2	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	4	0	0	
2	0.01000	0	0	0	0	0	1	2	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	4	0	0	
3	0.01500	0	0	0	0	0	1	2	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	4	0	0	

Tabela 13: Arquivo com extensão `.LAB`.

A Tabela 13 representa o arquivo com extensão `.LAB` correspondente ao arquivo com extensão $.DG\_01$. Pode-se verificar que os dois primeiros campos desse arquivo são idênticos aos do arquivo com extensão $.DG\_01$, enquanto os outros, no total de 28, representam agora os indivíduos e não mais os grupos. Analisando os outros 28 campos da Tabela 13, temos que na temperatura 0.000 todos os indivíduos pertencem ao grupo 0 (zero), por isso a representação 0 (zero) nos campos, ou seja, as posições representam os indivíduos, e os valores dos campos representam os grupos aos quais os indivíduos pertencem. Na temperatura 0.005, temos que os cinco primeiros indivíduos pertencem ao grupo 0, o sexto indivíduo ao grupo 1, o sétimo ao grupo 2, os próximos 7 (posições de 8 a 14) pertencem ao grupo 0, o indivíduo 15 pertence ao grupo 3, o 16

ao grupo 1, os próximos 9 (posições de 17 a 25) ao grupo 0 novamente, o indivíduo 26 pertence ao grupo 4 e os últimos dois indivíduos ao grupo 0. Calculando-se o total de indivíduos por grupo no arquivo com extensão `.LAB`, obtém-se exatamente o arquivo com extensão $.DG\_01$.

O funcionamento do programa `SPC` descrito nesta seção corresponde ao programa original, desenvolvido por Domany et al. (1999) . Pelas razões apresentadas anteriormente, com relação à interação usuário-programa, algumas alterações foram realizadas nos arquivos de saída apresentados nesta seção e foi implementada uma interface para trabalhar em conjunto com o programa `SPC`. Essas alterações e a interface são descritas nas seções seguintes.

5.4 Otimizações

O programa `SPC`, como descrito na seção anterior, apesar de apresentar bons resultados e ser eficiente na realização dos agrupamentos de dados, não era orientado ao usuário, ou seja, não permitia uma interação simples e amigável com qualquer outra pessoa que não fosse o seu desenvolvedor. Sendo assim, foi desenvolvida uma interface amigável que procura extrair o melhor, em termos de informações relacionadas com os agrupamentos, desse programa. A interface foi desenvolvida em `IDL` (ver Seção 1.3.1, página 23). Esta linguagem foi escolhida para a implementação porque apresentou vários recursos matemáticos que facilitaram a implementação de muitas opções disponibilizadas no programa e que envolviam cálculos, inclusive com matrizes.

Ao iniciar o desenvolvimento da interface foi verificado que os cinco arquivos de saída do programa `SPC` (ver Figura 24, página 77) não eram necessários ao desenvolvimento da mesma, mas apenas os dois arquivos mencionados na seção anterior, ou seja, o $.DG\_01$ e o `.LAB`. Estes arquivos ainda sofreram alterações para que contivessem informações relevantes para o desenvolvimento dos dendrogramas (ver Seção 2.1.2, página 30).

Os outros três arquivos de saída armazenam informações de controle para o desenvolvedor, talvez ele os tenha criado para verificar a corretude de suas operações ou outro motivo particular. O arquivo com extensão `.EDGE` é opcional e é usado para forçar o programa `SPC` a considerar bordas específicas, além das selecionadas pelo método `KNN`. Cada borda é descrita por um alinhamento no formato:

```
i1  i2
```

onde i1 e i2 são os índices dos pontos.

Um outro arquivo de saída não utilizado é o `.PARAM` que contém uma lista de todos os parâmetros usados pelo programa e seus valores. A Tabela 14 exibe um exemplo

desse tipo de arquivo.

AverageInteraction	0.074233
ClustersReported	12
CharDist	13.567979
DataFile	yoram.data
Dimentions	0
DirectedGrowth	
DataIsMatrix	
Dimensions	0
KNearestNeighbours	10
MinTemp	0.00
MaxTemp	0.15
MSTree	
NumberOfPoints	90
NumberOfEdges	375
NearestNeighbrs	8.333333
OutFile	yout
PottsSpins	20
RandomSeed	951312526
SusceptColors	4
SaveSuscept	
SWCycles	4000
SWFraction	0.800000
ThresholdTheta	0.500000
TempStep	0.005
WriteLabels	

Tabela 14: Exemplo de arquivo com extensão `.PARAM`.

Muitos dos parâmetros exibidos no exemplo da Tabela 14, os mais importantes para efeito deste projeto, já foram discutidos na Seção 5.2 (página 77). O terceiro arquivo não utilizado é um arquivo de `log`, que armazena os passos de execução do `SPC`.

Os dois arquivos utilizados no projeto foram os com extensão `.LAB` e o $.DG\_01$. O primeiro tipo de arquivo sofreu duas mudanças, a primeira foi a ordem de armazenamento das temperaturas e agrupamentos, ou seja, antes os agrupamentos eram exibidos da temperatura inicial até a final, como pode ser visto na Tabela 13 (página 80). Com a mudança, a ordem inversa foi adotada (ver Tabela 15, página 83). Esta primeira alteração foi necessária por causa do algoritmo utilizado para a construção dos dendrogramas, uma vez que ele monta os gráficos da temperatura mais alta para a mais baixa. Com esta mudança, ganha-se tempo de processamento, uma vez que não é mais necessário o armazenamento de todos os agrupamentos em arquivo para em seguida montar o dendrograma, dessa maneira, à medida que os agrupamentos são lidos, os dendrogramas são montados. A geração dos agrupamentos pelo `SPC` ainda é feita da temperatura mais baixa para a mais alta, sendo que estes agrupamentos não são mais armazenados em arquivos e sim em uma estrutura de dados dinâmica.

Uma lista duplamente encadeada foi utilizada para armazenar os agrupamentos e as informações que serão definitivamente salvas em arquivos e lidas posteriormente pela interface. Essa implementação pode ser observada no apêndice A (página 128), na seções A.2.1 e A.2.2.

0.01500	0	0	0	0	0	1	2	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	4	0	0
0.01000	0	0	0	0	0	1	2	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	4	0	0
0.00500	0	0	0	0	0	1	2	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	4	0	0
0.00000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela 15: Arquivo com extensão .LAB com ordem invertida.

A segunda alteração melhorou a performance do programa e permitiu que os arquivos gerados ocupassem menos espaço em disco. Como foi visto na Tabela 13 (página 80), em algumas temperaturas não ocorrem novos agrupamentos e assim os agrupamentos anteriores são repetidos. A segunda alteração encarrega-se de eliminar estas repetições e armazenar no arquivo apenas as linhas em cujas temperaturas aconteceram novos agrupamentos (ver Tabela 16). Como pode ser observado na Tabela 16, uma repetição ainda é mantida, isto acontece porque, por convenção, preferiu-se manter nos arquivos de saída os agrupamentos das temperaturas mais baixa e mais alta, para facilitar a representação durante o processamento da interface. Mesmo mantendo-se essa repetição, o ganho no armazenamento é considerável e a simplicidade da estrutura do arquivo de saída é mantida. A implementação da eliminação de agrupamentos repetidos pode ser observada na Seção A.2.3 (página 146) do Apêndice A.

0.01500	0	0	0	0	0	1	2	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	4	0	0
0.00500	0	0	0	0	0	1	2	0	0	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	4	0	0
0.00000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela 16: Arquivo com extensão .LAB atualizado.

O SPC apresentava uma limitação que também foi retirada, ela refere-se ao número de grupos armazenados nos arquivos. Este número de grupos era sempre o mesmo, ou seja, era uma constante configurada no código fonte do programa, o que não era interessante para os objetivos da interface, que era mostrar todos os grupos e todos os indivíduos pertencentes a estes grupos em uma forma gráfica. Com o número reduzido de grupos isso não era possível, uma vez que a hierarquia dos agrupamentos não era mostrada. Essa mudança tornou os arquivos de saída mais completos, ou seja, eles passaram a armazenar as informações de todos os agrupamentos realizados.

O segundo arquivo, com extensão *.DG_01*, também teve sua ordem de armazenamento invertida e suas linhas repetidas eliminadas, mas acrescentou outras mudanças, informações a respeito dos agrupamentos e a respeito das propriedades usadas na geração do arquivo com extensão .DATA (ver Apêndice A, página 128). Cinco novos parâmetros foram introduzidos nas primeiras linhas desse arquivo, são eles:

- Número de variáveis utilizadas na geração do arquivo com extensão `.DATA`;
- Índices das variáveis utilizadas na geração do arquivo com extensão `.DATA`;
- Número de temperaturas onde ocorreram novos agrupamentos;
- Total de grupos gerados durante o agrupamento;
- Número de indivíduos utilizados nos agrupamentos.

Todo conjunto de dados apresenta pelo menos uma variável, que determina alguma característica específica de cada indivíduo. Um conjunto de dados pode apresentar N variáveis. Porém, o algoritmo de geração da matriz de dissimilaridade utilizado neste projeto permite que sejam escolhidas quais variáveis farão parte da geração da matriz. Com isso, a identificação das variáveis utilizadas e sua quantidade foram armazenadas no arquivo de dados para que, ao longo do programa, o reconhecimento, utilização e apresentação das variáveis realmente usadas na geração de uma determinada matriz de dissimilaridade fosse considerado e até mesmo exposto ao usuário como uma informação.

Dessa maneira, o primeiro parâmetro a considerar, o `número de variáveis`, indica que das N variáveis que fazem parte do conjunto de dados, apenas um número determinado foi utilizado na geração da matriz. O segundo parâmetro corresponde aos índices dessas variáveis, ou seja, se um conjunto de dados apresenta $N = 5$ variáveis, mas apenas três delas foram utilizadas na geração da matriz, o primeiro parâmetro do arquivo com extensão `.DATA` será 5 e na sua segunda linha constarão os índices dessas variáveis, por exemplo as variáveis com os índices 0, 3 e 4 foram utilizadas na geração da matriz de dissimilaridade correspondente. Um exemplo desse arquivo é exposto na Tabela 17.

4											
0	2	3	6								
17	10	50									
80	0.40000	23	11	2	2	2	2	2	2	2	2
41	0.20500	21	13	2	2	2	2	2	2	2	2
...											

Tabela 17: Exemplo de um arquivo com extensão $.DG\_01$ com parâmetros.

O terceiro parâmetro, representado pelo valor 17 na Tabela 17, representa o número de temperaturas onde aconteceram alguma modificação nos agrupamentos, ou seja, indica o número de linhas que faltam antes do fim do arquivo. O quarto parâmetro, representado pelo valor 10 na mesma tabela, equivale à quantidade de grupos formados e o último parâmetro indica a quantidade total de indivíduos do conjunto de dados.

Após essas alterações no `SPC`, o projeto e implementação da interface pôde ser iniciado. Estas fases são descritas nas próximas seções.

5.5 A Interface do SPC

O desenvolvimento de uma interface amigável para o programa SPC é uma maneira de verificar a nova técnica de agrupamento de dados e, mais ainda, de permitir um maior aproveitamento dos resultados obtidos e da maneira mais simples possível. Buscando antigir essa simplicidade e esse maior aproveitamento, foi realizada uma análise dos resquisitos do sistema, com o objetivo de definir a priori tudo o que seria necessário para o desenvolvimento de uma interface amigável, completa e simples de usar por qualquer usuário. Para representar essa análise foi utilizada uma abordagem mista, envolvendo ferramentas de modelagem de objetos (Furlan 1998) e de análise essencial (Pompilho 2002, Yourdan 1990), para exibir de maneira mais clara o significado de cada função componente do sistema.

A próxima seção exibe a análise dos requisitos e suas funções, enfocando os eventos do sistema. A Seção 5.5.2 (página 93) descreve a implementação do sistema com base na seção de análise.

5.5.1 Análise e Projeto dos Requisitos

A especificação de requisitos para o desenvolvimento de uma interface é uma tarefa complicada, uma vez que interfaces são orientadas por eventos, que é um mecanismo conhecido como estímulo/resposta, ou seja, o usuário produz um estímulo dependendo de sua necessidade e dos recursos disponibilizados pela interface e esta, por sua vez, emite uma resposta ou realiza algum processamento. A parte complicada de se trabalhar com sistemas baseados em eventos é especificar o que será e o que não será útil para os usuários que utilizarão o sistema, sem confundi-lo com a inserção de opções desnecessárias.

Podemos classificar os eventos em três tipos diferentes, são eles:

Evento orientado por fluxo de dados: é aquele em que o estímulo é a chegada ao sistema de um fluxo de dados enviado por uma entidade externa.

Evento orientado por controle: é aquele em que o estímulo é a chegada ao sistema de um fluxo de controle, por exemplo a ativação de alguma variável binária.

Evento orientado por tempo: é aquele em que o estímulo é a chegada ao sistema da informação de haver passado um determinado intervalo de tempo.

A implementação da interface foi baseada na análise de seus eventos e no controle do sistema. Como dito antes, a metodologia utilizada para realizar a análise corresponde a técnicas e ferramentas de análise essencial e de modelagem orientada a objetos. A análise essencial envolve três abordagens, das quais apenas duas são usadas aqui, são elas as abordagens **funcional** e de **controle**. No início da análise do sistema

aqui proposto foi utilizada uma sobreposição das técnicas de análise essencial e de modelagem de dados, definindo o sistema a partir de suas funções e representando esse comportamento através de diagramas de respostas a eventos e diagramas de caso de uso. Uma visão mais geral do sistema pôde ser obtida a partir da modelagem de objetos, exibindo as interações do usuário com os módulos/funções componentes do sistema e as interações entre elas próprias.

A Tabela 18 exibe uma lista de eventos do sistema classificando-os da seguinte maneira:

- (F) - Evento orientado por fluxo de dados
- (T) - Evento temporal
- (C) - Evento orientado por controle

Estes eventos são considerados os requisitos principais do sistema e é a partir deles que decorre todo desenvolvimento do projeto.

Número do Evento	Nome do Evento	Tipo do Evento
(1)	Abrir arquivo de dados	(F)
(2)	Fechar arquivo de dados	(C)
(3)	Padronizar dados	(C)
(4)	Gerar MDA (Multivariate Data Analysis)	(C)
(5)	Gerar Matriz de Dissimilaridade	(F)
(6)	Gerar arquivo .run e Executar SPC	(F)
(7)	Gerar Dendrograma	(F)
(8)	Procurar Grupo	(F)

Tabela 18: Principais eventos do sistema

Os eventos da Tabela 18 representam as principais funções do sistema, dentre estes destacam-se os eventos (4), (7) e (8), que são responsáveis pela integração do módulo principal do sistema com os módulos de análise multivariada, dendrogramas e pesquisa de grupos, repectivamente. De uma maneira geral, o sistema pode ser representado como na Figura 25 (página 87).

De acordo com o diagrama de caso de uso apresentado na Figura 25 (página 87), pode-se verificar que o usuário e o programa SPC são as únicas entidades externas (atores) que se comunicam com a interface do sistema. Investigando esta interface, podemos representar, de uma forma mais detalhada, as funções do sistema de acordo com a Figura 26 (página 88).

As funçõs apresentadas na Figura 26 são descritas nas próximas seções, bem como as ações geradas para acioná-las e suas respostas aos estímulos gerados.

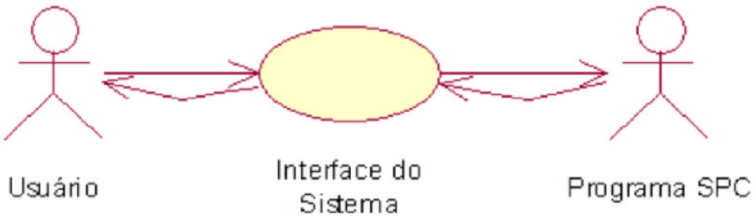


Figura 25: Caso de uso geral para o Sistema Proposto.

5.5.1.1 Abrir Arquivo de Dados

Esta função dá início a todas as operações fornecidas pela interface, uma vez que ela permite a abertura do arquivo de dados para a interface. Esses dados seguem o modelo visto na Tabela 9 (página 78), ou seja, uma matriz de observações por variáveis.

Após a abertura do arquivo, os dados serão exibidos ao usuário que então poderá manipulá-los. Como apresentado no diagrama de resposta ao evento da Figura 27 (página 89), os dados são armazenados temporariamente em uma estrutura para melhorar a eficiência de sua manipulação.

No diagrama da Figura 27, podemos observar dois depósitos de dados. O ARQUIVO 1 representa o arquivo em disco, ou seja, o arquivo que está sendo aberto pelo usuário. O ARQUIVO 2 representa o armazenamento temporário do conteúdo de ARQUIVO 1, para facilitar a manipulação e exibição para o usuário, uma vez que todas as operações realizadas a partir da abertura do arquivo de dados serão sobre o armazenamento temporário.

Segundo a análise essencial, a ativação e a reação deste evento será na forma:

Estímulo: Escolha da opção abrir.

Ações: Ler do arquivo selecionado.

Respostas: Carga do conteúdo do arquivo selecionado para uma grade.

5.5.1.2 Fechar Arquivo de Dados

A função de fechar arquivo de dados é responsável pelo fechamento do arquivo ativo e pela liberação de memória das unidades logicamente alocadas pelo sistema.

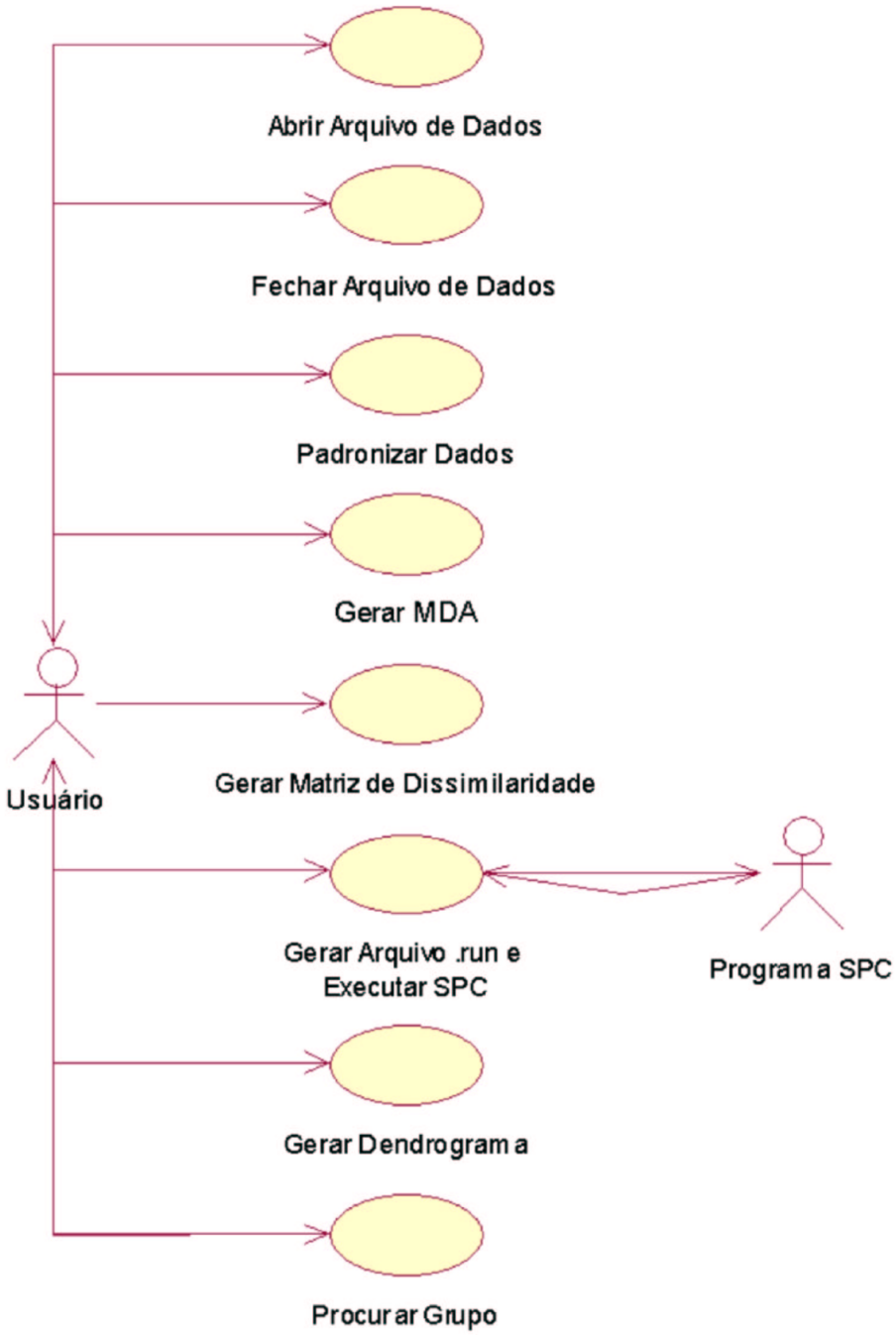


Figura 26: Caso de uso detalhado do sistema.

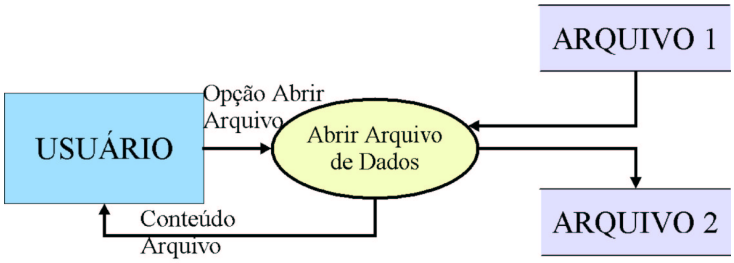


Figura 27: Diagrama de Resposta ao Evento “Abrir Arquivo de Dados”.

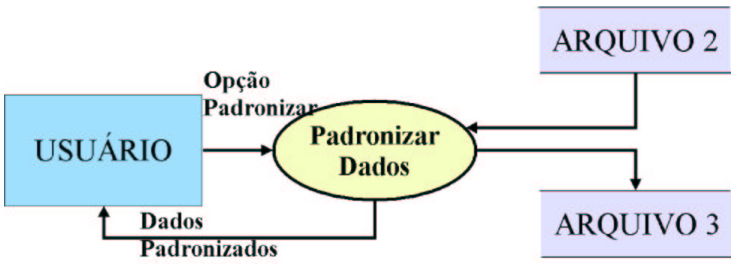


Figura 28: Diagrama de Resposta ao Evento “Padronizar Dados”.

5.5.1.3 Padronizar Dados

Algumas vezes as unidades utilizadas para medir os dados não são adequadas e fazem com que os agrupamentos sejam imprecisos, sendo assim, dependendo das unidades utilizadas podemos visualizar vários agrupamentos diferentes, o que causa confusão.

Esta função é responsável pela padronização dos dados ativos, ou seja, através dela é retirada a dependência dos dados em relação às suas unidades.

A Figura 28 apresenta o diagrama de resposta ao evento “Padronizar Dados”, com base nas seguintes informações.

Estímulo: Checagem da opção de padronização.

Ações: Calcular dados não padronizados.

Respostas: Exibição dos dados padronizados na grade.

No diagrama da Figura 28, podemos observar dois repositórios de dados, o ARQUIVO 2 representa o armazenamento temporário do arquivo aberto pelo usuário, enquanto o ARQUIVO 3 representa o armazenamento temporário dos dados padronizados de ARQUIVO 2. Dessa maneira, o cálculo para padronização é realizado apenas uma vez e armazenado para futura manipulação, tal como a geração de uma matriz de dissimilaridade.

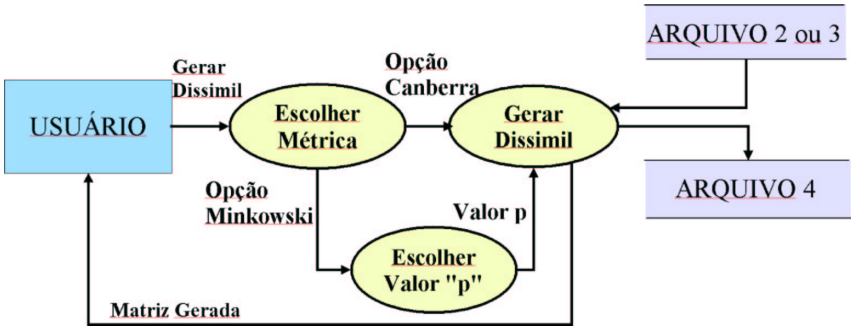


Figura 29: Diagrama de Resposta ao Evento “Gerar Dissimilaridades”.

5.5.1.4 Gerar Dissimilaridades

A partir desta função é possível a geração de uma matriz de dissimilaridades para os dados armazenados em ARQUIVO 2 ou em ARQUIVO 3, dependendo da opção de padronização.

Para calcular uma matriz de dissimilaridade serão disponibilizadas duas opções de métricas a serem consideradas nos cálculos, são elas: Canberra e Minkowski (ver Seção 2.1.1, página 28).

O diagrama de resposta ao evento é apresentado na Figura 29. Nele podemos verificar que, dependendo da escolha de padronização ou não, os ARQUIVOS 2 ou 3, respectivamente, serão utilizados como entrada da função. Além disso, antes da geração da matriz a métrica deve ser escolhida. Se Canberra for escolhida, então a matriz de dissimilaridade será gerada imediatamente após a escolha. Caso Minkowski seja escolhida, é necessária a seleção de um valor para o parâmetro p. Este parâmetro é utilizado na equação da distância de Minkowski e especifica a equação da distância a ser utilizada para a obtenção das dissimilaridades.

O ARQUIVO 4 armazena fisicamente e permanentemente a matriz de dissimilaridades gerada, tal como a apresentada na Tabela 10 (página 78).

5.5.1.5 Gerar Agrupamentos (Funções “Gerar Arquivo .run” e “Executar SPC”)

O diagrama da Figura 30 (página 91) mostra que a função de geração de agrupamentos envolve outras duas funções, a de geração do arquivo de informação `.RUN` e a execução do `SPC`.

A primeira função, “Gerar Arquivo .run”, recebe dados do usuário (ver Tabela 11, página 79) para montar o arquivo de informações, que é armazenado fisicamente em disco e servirá como parâmetro para o `SPC`. Após a criação do arquivo com extensão `.RUN`, a função chama o procedimento que executa o `SPC`, que se encarrega de ler as informações armazenadas no `.RUN` e ler a matriz de dissimilaridade armazenada no

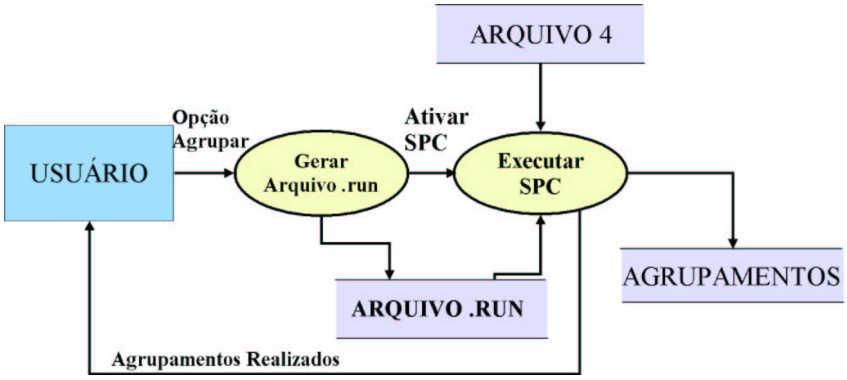


Figura 30: Diagrama de Resposta ao Evento “Gerar Agrupamentos”.

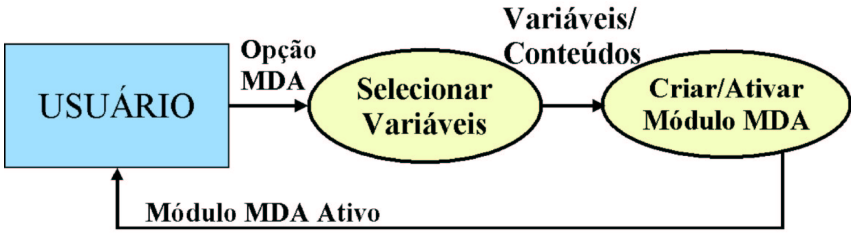


Figura 31: Diagrama de Resposta ao Evento “Gerar MDA”.

ARQUIVO 4 (ver Seção 5.5.1.4, página 90).

O usuário é informado da realização com sucesso ou não dos agrupamentos e ele pode ainda, em caso de sucesso, visualizar esses agruapamentos de forma gráfica através de dendrogramas, que podem ser acessados por meio do módulo “Dendrogramas” (ver Seção 5.5.1.7, página 92).

5.5.1.6 Gerar MDA (`Multivariate Data Analysis`)

`MDA` significa “Análise de Dados Multivariados” e esta função faz uma ligação entre o módulo principal e o módulo de análise multivariada, onde serão disponibilizadas funcionalidades como visualização dos dados através de histogramas e `scatterplots`, análise descritiva dos dados e armazenamento dos gráficos em arquivos.

O diagrama de resposta ao evento “Gerar MDA” é apresentado na Figura 31. Esta é uma função simples que realiza o armazenamento das variáveis que serão utilizadas na análise dos dados e cria o módulo MDA. Ele retorna ao usuário o acesso ao módulo de análise multivariada, que contém funções específicas do gráfico exibido. A Tabela 19 (página 92) exibe um resumo dessas funções.

Função	Descrição
Selecionar Variáveis	Seleção das variáveis que farão parte dos gráficos
Configurar Gráficos	Configuração do modo de exibição do gráfico, por exemplo um scatterplot pode ter seus dados logaritmicamente transformados para uma melhor visualização.
Salvar Gráfico	Salva o gráfico corrente em um arquivo com formato .EPS
Análise Descritiva	Exibe valores estatísticos calculados com base no conjunto de dados

Tabela 19: Sub-funções da função “Gerar MDA”.

5.5.1.7 Gerar Dendrogramas

Semelhantemente à função “Gerar MDA”, “Gerar Dendrogramas” possui um diagrama de resposta ao evento muito simples, pois também retorna ao usuário o acesso a um outro módulo, o de “Dendrogramas”. Este módulo exibe, sob a forma de dendrogramas (ver Seção 2.1.2, página 30), o resultado de agrupamentos realizados através da chamada ao evento “Gerar Agrupamentos” (ver Seção 5.5.1.5, página 90).

As funções deste módulo são exibidas na Tabela 20.

Função	Descrição
Escoher Tipo Dendrograma	A exibição do dendrograma pode ser feita de três maneiras: completa, por temperatura e por indivíduo
Ativar Sensibilidade do Gráfico	Ativa a sensibilidade do gráfico ao clique do mouse
Salvar Gráfico	Salva o gráfico corrente em um arquivo com formato .EPS

Tabela 20: Sub-funções da função “Gerar Dendrogramas”.

A sub-função “Escolher Tipo Dendrograma” controlará a exibição do dendrograma ao usuário. Isto poderá ser feito de três maneiras:

completa: esta forma apresentará todo o dendrograma, sem restrições.

por temperatura: nesta forma, temperaturas são escolhidas e apenas os agrupamentos realizados nessas temperaturas serão exibidos no gráfico.

por indivíduo: semelhantemente à forma por temperatura, indivíduos são escolhidos e o dendrograma exibido será o completo, mas com destaque para os agrupamentos que contêm os indivíduos escolhidos.

Já a sub-função “Ativar Sensibilidade do Gráfico” é uma das funções mais interessantes disponibilizadas na interface, pois ela permitirá que o usuário interaja com o dendrograma e consequentemente com os agrupamentos nele exibidos.

Figura 32: Diagrama de Resposta ao Evento “Procurar Grupo”.

O objetivo dessa sub-função é fazer com que o usuário, ao mover o ponteiro do mouse através da região do gráfico, verifique sua mudança para a forma de seta. Isto acontecerá em quase toda região do gráfico, pois nos pontos onde acontecerem os agrupamentos, ocorrerá outra mudança da forma do ponteiro do mouse para uma cruz. Esta mundaça indicará ao usuário que naquela região, sob as condições de temperatura exibidas no dendrograma, aconteceu um agrupamento. O usuário pode ainda clicar nessa região do gráfico, onde o ponteiro está sob a forma de cruz, e obter informações sobre a quantidade de indivíduos contidos naquele agrupamento e quais são aqueles indivíduos, além de uma análise estatística envolvendo os dados e as variáveis do conjunto de dados. Essa função é bastante útil em situações onde o conjunto de dados utilizado no agrupamento é muito grande, impossibilitando uma visão apropriada das informações contidas no dendrograma.

5.5.1.8 Procurar Grupo

Esta função localizará e exibirá o agrupamento mais recente que contenha todos os indivíduos selecionados na pesquisa. É apenas mais uma opção para localização de grupos, que poderá ser utilizada sempre que não for preciso a geração de dendrogramas e cujo único interesse sejam os valores estatísticos calculados sobre o agrupamento.

A Figura 32 exibe o diagrama de resposta a este evento. Uma vez escolhida a opção de pesquisa de grupo, o usuário deverá escolher os indivíduos que necessariamente deverão fazer parte do grupo que ele procura. Após esta escolha, os indivíduos selecionados são passados para a função de pesquisa que devolverá ao usuário o agrupamento mais recente que contém os indivíduos que ele escolheu, além da análise descritiva mencionada no parágrafo anterior.

5.5.2 Implementação da Interface

A Seção 5.5.1 (página 85) analisou e descreveu de uma maneira textual o funcionamento da interface proposta, nesta seção será exibida essa interface, suas funções correspondentes e as implementações realizadas.

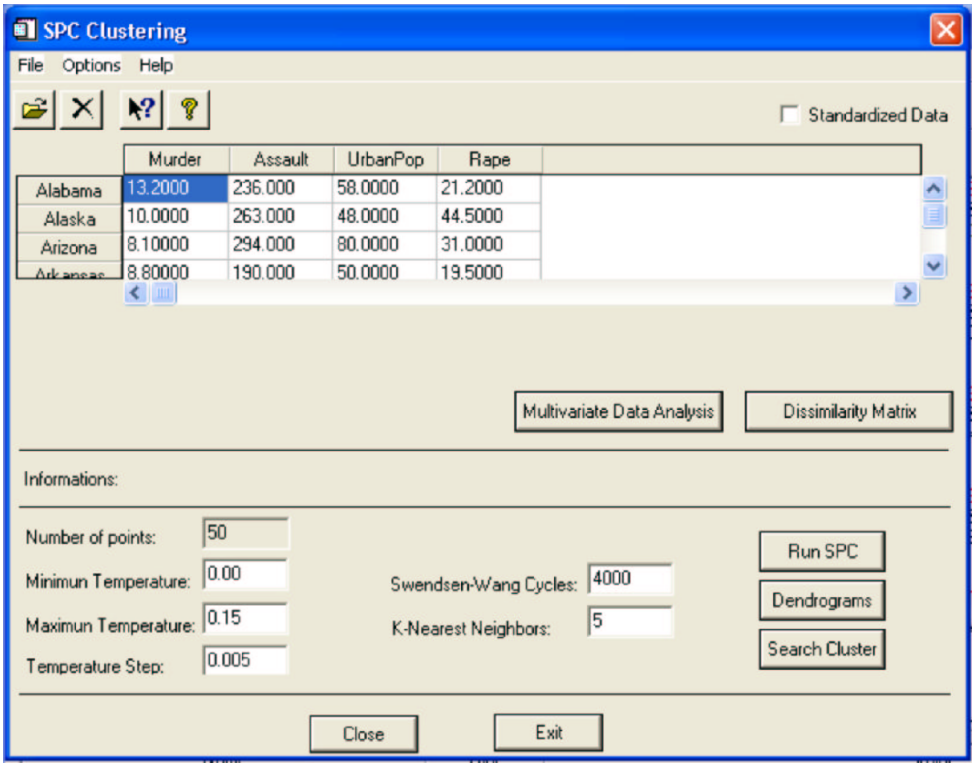


Figura 33: Tela Principal da Interface.

O conjunto de dados utilizado para exibir a interface nas próximas seções possui 50 indivíduos, que correspondem a estados dos *Estados Unidos da América* e a alguns outros países. Possui ainda 4 variáveis que são utilizadas tanto na exibição da interface quanto na realização dos cálculos do sistema. Três dessas variáveis representam a quantidade de prisões feitas nesses estados e países de acordo com os crimes de assassinato, assalto e estupro. A quarta variável representa a população urbana de cada estado/país. Esse conjunto de dados é exibido na Tabela 21 (página 113).

5.5.2.1 Módulo Principal

O módulo principal, responsável por todas as funções ilustradas na Figura 26 (página 88), possui a aparência exibida na Figura 33. Estas funções podem ser acessadas através de menus e botões para facilitar a interação com o usuário.

A interface principal apresenta outras poucas funções além das principais descritas na Seção 5.5.1 (página 85), algumas delas redundantes, para melhorar a interação usuário-interface.

A seguir as principais opções fornecidas por este módulo são explicadas.

Menu File: Este menu apresenta três opções, `Open`, `Close` e `Exit`, como ilustra a Figura 34. A opção `Open` permite abrir um arquivo de dados e carregá-lo

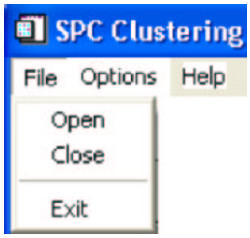


Figura 34: Opção “File” do Menu Principal.

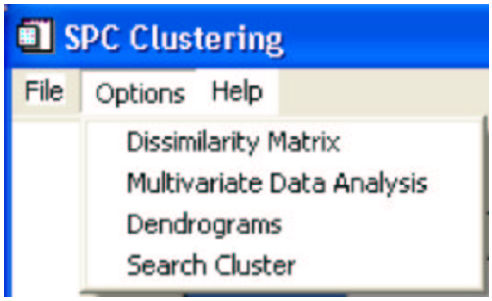


Figura 35: Opção “Options” do Menu Principal.

para a grade principal da interface. Sua implementação pode ser conferida na Seção A.1.1 (página 128). A opção `Close` fecha o arquivo de dados atualmente carregado na grade principal e libera as unidades lógicas ocupadas por ele (ver implementação na Seção A.1.2, página 131). A última opção deste menu é a `Exit`, que fecha todo o programa.

Menu Options: É composto por quatro opções: `Dissimilarity Matrix`, `Multivariate Data Analysis`, `Dendrograms` e `Search Cluster`, ver Figura 35. Todas essas opções são também disponibilizadas através de botões que serão detalhados mais adiante.

Opção Standardized Data: Esta opção corresponde à função “Padronizar Dados” vista na Seção 5.5.1.3 (página 89).

Para ilustrar o processo realizado por ela, considere os dados da Tabela 22 (página 114).

Para padronizar os dados da Tabela 22, inicialmente calcula-se o valor médio da variável f, dada por

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_{if} \tag{5.1}$$

para cada $f = 1, 2, \ldots, q$. Utilizando a Tabela 22 (página 114) como referência, a variável f representa os atributos “Idade” e “Altura”. Em seguida calcula-se uma medida de dispersão para cada f. Mais comumente usa-se o desvio padrão para

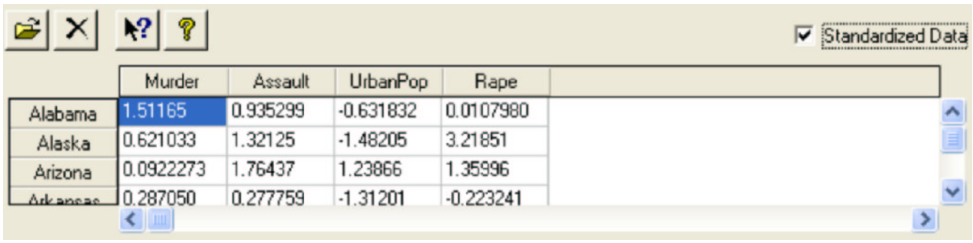


Figura 36: Opção “Standardized Data” do Módulo Principal.

este propósito.

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{if} - \bar{x})^2} \tag{5.2}$$

Entretanto, o desvio padrão passa a não ser uma medida precisa quando consideramos valores muito dispersos no conjunto de dados, ou seja, valores muito distantes da média do conjunto, isto porque $x_{if} - \bar{x}$ é um valor ao quadrado. Por causa desse problema, precisamos de uma medida de dispersão que não seja tão sensível a valores muito dispersos, é o caso do `desvio absoluto médio`

$$s_f = \frac{1}{n}\sum_{i=1}^{n} \mid x_{if} - \bar{x} \mid \tag{5.3}$$

onde a contribuição de cada medida x_{if} é proporcional ao valor absoluto $|x_{if} - \bar{x}|$.

Essa medida é mais robusta no sentido que valores muito dispersos não terão uma grande influência em s_f. Assumindo que s_f seja diferente de zero (caso contrário a variável f é constante em todos os objetos e por causa disto deve ser removida). As medidas padronizadas são definidas por

$$z_{if} = (x_{if} - \bar{x})/s_f \tag{5.4}$$

e algumas vezes é chamada de `z-scores`. Todas estas medidas estão sem unidades, porque tanto o numerador quanto o denominador possuem a mesma unidade.

Para ilustrar a função, vamos padronizar os dados da Tabela 22 (página 114). Seguindo as equações (5.1) (página95), (5.3) e (5.4), temos:

Calculadas as médias e os desvios absolutos médios para os atributos “Idade” ($\bar{x}_1$ e s_1) e “Altura” ($\bar{x}_2$ e s_2), podemos obter os valores de z_{if}, ou seja, os dados padronizados (ver Tabela 24, página 114).

O usuário do sistema pode optar por visualizar na tela ou o conjunto de dados original ou seus dados padronizados, como detalhado anteriormente. A implementação dessa função em `IDL` é exibida na Seção A.1.3 (página 132) e o seu resultado, ou seja, a interface gráfica gerada a partir desse código mostrada na Figura 36.

Figura 37: Campos “Informations” do Módulo Principal.

Figura 38: Opção “Multivariate Data Analysis” do Módulo Principal.

Campos Informations: Esses campos são uma éspecie de representação visual do arquivo com extensão `.RUN`, ver Seção 5.2 (página 77). Com base nas alterações realizadas nesses campos (ver Figura 37), o arquivo com extensão `.RUN` é gerado quando a opção `Run SPC` é executada.

Os campos exibidos na Figura 37 são detalhados na Seção 5.2 e a implementação da geração do arquivo com extensão `.RUN` é citada juntamente com a explicação da opção `Run SPC`.

Opção Multivariate Data Analysis: Esta opção realiza a ligação entre o módulo principal e o módulo MDA. Ela pode ser acessada através do botão exibido na Figura 38. Este módulo será detalhado na Seção 5.5.2.2.

Opção Dissimilarity Matrix: Esta opção pode ser acessada através do componente exibido na Figura 39 (página 98). Ele tem como objetivo realizar os cálculos necessários à geração da matriz de dissimilaridade.

Esta opção permite que o usuário escolha entre duas métricas: Minkowski e Canberra (ver Seção 2.1.1, página 28). Ao clicar no componente da Figura 39 (página 98), a janela 40 (página 98) é exibida para que essa escolha seja feita. Se a opção Canberra for escolhida, a matriz é gerada imediatamente após à escolha do nome do arquivo que irá armazenar a matriz de dissimilaridade. Se a opção Minkowski for a escolhida, a janela da Figura 41 (página 98) será exibida ao usuário para que este determine o valor do parâmetro p. Este parâmetro é qualquer número real maior ou igual a 1. A escolha do parâmetro p é necessário para a geração da matriz de dissimilaridades. Este parâmetro é utilizado na distância de Minkowski e especifica a equação da distância a ser utilizada para a obtenção das dissimilaridades. Essa também é chamada métrica L_p, com os

Figura 39: Opção “Dissimilarity Matrix” do Módulo Principal.

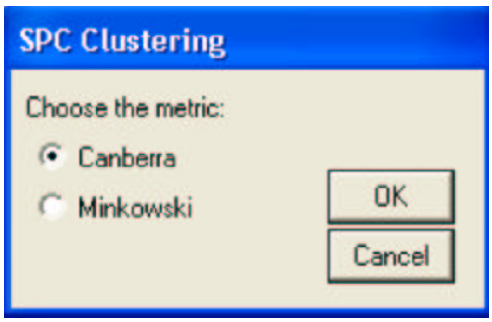


Figura 40: Métricas para a geração da matriz de dissimilaridade.

casos especiais de distância Euclidiana ($p = 2$) e distância Manhattan ($p = 1$).

A implementação desta função é exibida na Seção A.1.4 (página 133).

Opção Run SPC: Realiza a ligação entre o módulo principal e o programa `SPC`. Esta opção pode ser acessada pelo usuário através do componente exibido na Figura 42 (página 99). Após acioná-lo, é solicitado ao usuário que escolha um arquivo com extensão `.DATA` que contenha uma matriz de dissimilaridade válida. Uma vez selecionada a matriz de dissimilaridade, o usuário aguarda alguns segundos enquanto o sistema cria um outro arquivo, com extensão `.RUN`, que contém informações sobre o conjunto de dados sendo usado e a matriz de dissimilaridade escolhida. Em seguida, o sistema passa esses dois arquivos como parâmetros para o programa `SPC`.

A janela 43 (página 99) é exibida em seguida. Ela mostra a chamada feita ao programa `SPC` pelo sistema e os seus resultados, ou seja, os agrupamentos sendo realizados.

A chamada ao programa `SPC` é feita pelo sistema, após a escolha da matriz de dissimilaridade pelo usuário e a criação do arquivo de informações pelo próprio

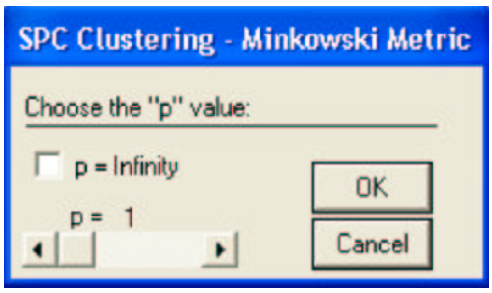


Figura 41: Escolha do parâmetro p para Minkowski.

Figura 42: Opção “Run SPC” do Módulo Principal.

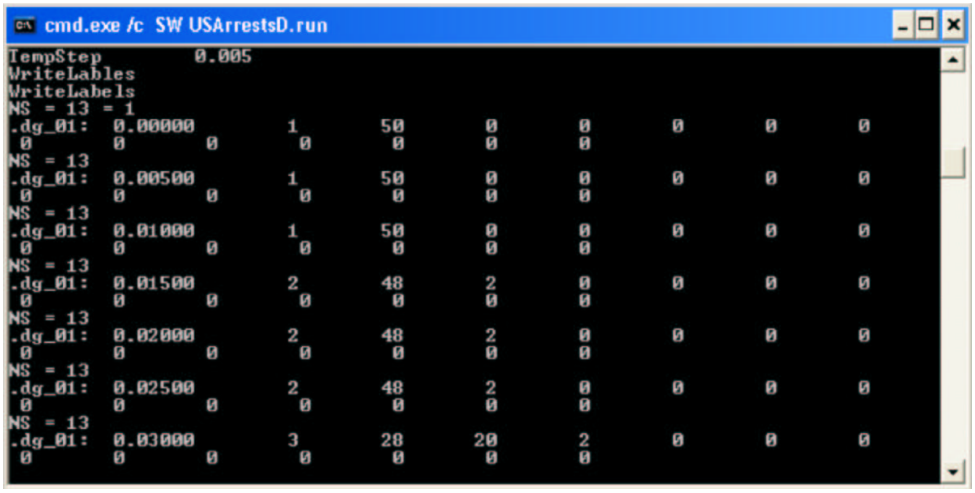


Figura 43: Programa `SPC` em execução.

sistema. Essa chamada é feita através do `IDL` em uma única linha de código:

```
spawn , 'SW ' + file
```

Onde `SW` é o nome do executável do programa `SPC` e `file` é o nome do arquivo com extensão `.RUN` que será executado. O procedimento do `IDL`, `spawn`, executa um comando ou uma série de comandos e tem como opção a possibilidade de exibir uma janela `shell` ou não. No exemplo apresentado o `shell` foi exibido para ilustração.

A implementação dessa função é exibida na Seção A.1.5 (página 138).

Opção Dendrograms: Esta opção pode ser executada a partir do componente exibido na Figura 44. Ela realiza a ligação entre o módulo principal e o módulo dendrograma, que será detalhado mais adiante.

Opção Search Cluster: Esta opção pode ser executada a partir do componente exibido na Figura 45 (página 100). Ela realiza a ligação entre o módulo principal e o módulo procurar grupo, que será detalhado mais adiante.

Figura 44: Opção “Dendrograms” do Módulo Principal.

Figura 45: Opção “Search Cluster” do Módulo Principal.

Figura 46: Opção “Close” do Módulo Principal.

Opção Close: Esta opção fecha o arquivo atualmente ativo na grade principal da interface e libera as unidades lógicas ocupadas por ele. Ela pode ser acessada através do componente apresentado na Figura 46 e sua implementação é exibida na Seção A.1.2 (página 131).

Opção Exit: Fecha todo o programa. É mostrada na figura 47.

Nas próximas seções serão apresentados os outros três módulos do sistema, o `Módulo MDA`, o `Módulo Dendrogramas` e o `Módulo Procurar Grupo`, suas descrições e implementações.

5.5.2.2 Módulo MDA

A Figura 48 (página 101) apresenta a tela do módulo `MDA` e suas sub-funções (ver Tabela 19, página 92).

O gráfico apresentado na Figura 48 é um `brushplot` em sua forma triangular, que exibe em sua diagonal principal os histogramas correspondentes às variáveis selecionadas na Figura 49 (página 101).

Nas outras posições do `brushplot` são apresentados `scatterplots`, que são gráficos de correlação entre as variáveis selecionadas.

São disponibilizadas também opções de configuração para esses gráficos. Para os histogramas é possível configurar o tipo e o valor do `bin` a ser apresentado; o tipo e a largura da linha do gráfico (pontilhada, tracejada, entre outras); o tipo, a largura e o tamanho da fonte; e o tipo e a largura dos eixos dos histogramas (ver Figura 50, página 102). Para os `scatterplots` também é possível configurar o tipo e a largura

Figura 47: Opção “Exit” do Módulo Principal.

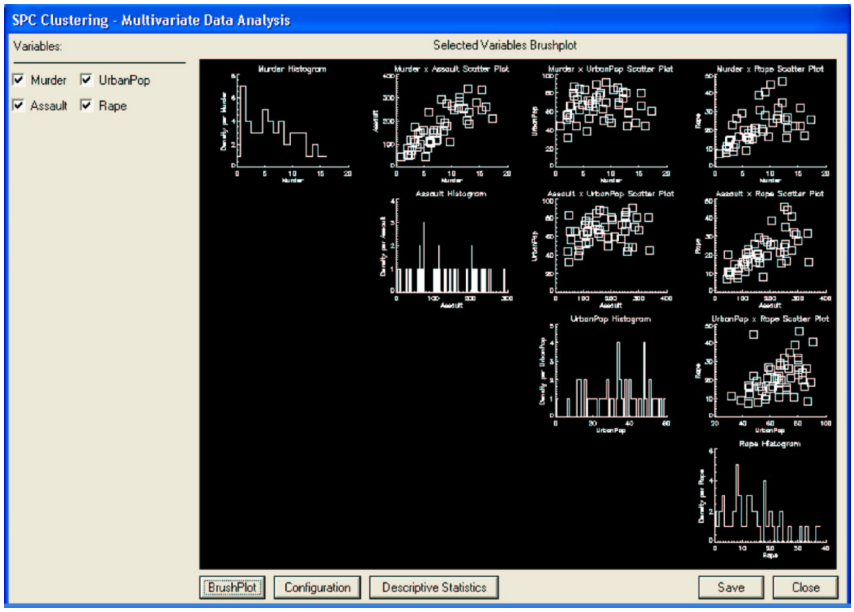


Figura 48: Módulo MDA.

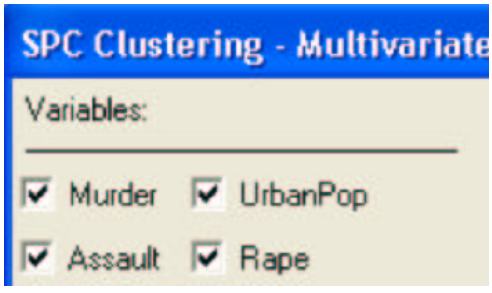


Figura 49: Seleção de Variáveis no Módulo MDA.

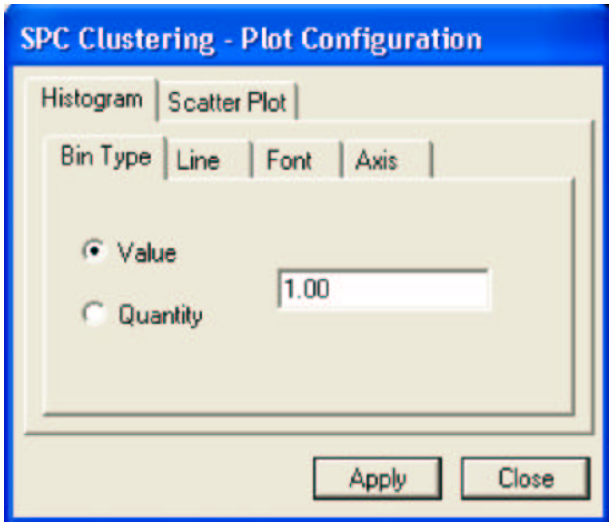


Figura 50: Configuração de Histogramas no Módulo MDA.

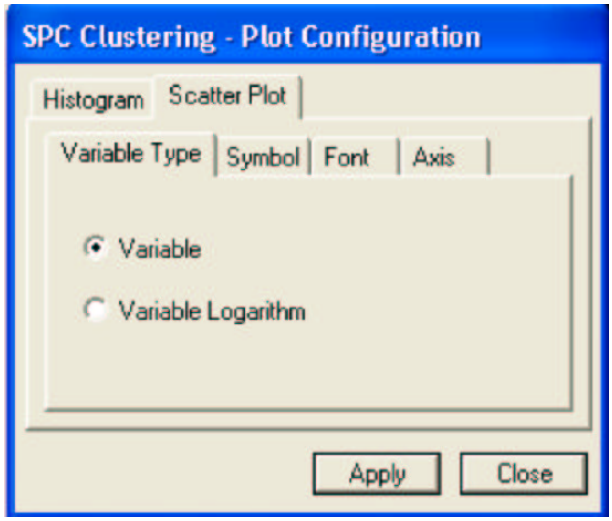


Figura 51: Configuração de ScatterPlots no Módulo MDA.

dos eixos; o tipo, a largura e o tamanho da fonte; o tipo e o tamanho do símbolo a ser exibido nos gráficos, tais como ponto, asterisco, sinal de somar, diamante, entre outros; e o tipo da variável, na forma normal ou logaritmica (ver Figura 51).

Outra opção disponibilizada é a de análise descritiva (ver Figura 52, página 105). Através dessa opção são exibidas várias quantidades estatísticas para as variáveis selecionadas na Figura 49 (página 101), que pertencem ao conjunto de dados da Tabela 21 (página 113). Essas quantidades são (ver Bustos & Frery 1992*a*):

Média Aritmética: É uma medida de localização que informa o valor central da amostra e obtém-se a partir da seguinte expressão: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, onde x_i representa o elemento da amostra e n a sua dimensão.

Média Geométrica: é a raiz enésima do produto enésimo dos elementos de um conjunto de dados. $M_G = \sqrt[n]{\prod_{i=1}^{n} x_i}$.

MAD: Define-se desvio médio absoluto (MAD) para uma série de n elementos da amostra como sendo a média aritmética simples dos módulos dos desvios desses n elementos. É dado por: $MAD = \frac{1}{n} \sum_{i=1}^{n} \mid x_i - \bar{x} \mid$.

Mediana: Dado que elementos da amostra estejam ordenados, a mediana é o valor (pertencente ou não à amostra) que a divide ao meio, isto é, 50% dos elementos da amostra são menores ou iguais à mediana e os outros 50% são maiores ou iguais à mediana. Dada a notação $x_{1:n}, x_{2:n}, \ldots, x_{n:n}$ para os elementos da amostra ordenada, tem-se a seguinte expressão para o cálculo da mediana:

$$mediana = \begin{cases} x_{\frac{n+1}{2}:n} & \text{se } n \text{ é ímpar,} \\ \frac{1}{2}(x_{\frac{n}{2}:n} + x_{\frac{n}{2}+1:n}) & \text{se } n \text{ é par.} \end{cases}$$

Primeiro Quartil: é o ponto que indica que 25% da amostra é necessariamente menor que ele, sendo os restantes 75% necessariamente maiores.

Terceiro Quartil: é o menor valor entre os 25% maiores valores do conjunto de dados.

Máximo e Mínimo: o maior e menor valor, respectivamente, do conjunto de dados.

Variância: é uma medida da variabilidade dos dados em torno da média. É dada por: $S^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$.

Desvio Padrão: é uma medida da variabilidade ou dispersão com as mesmas unidades que os dados. É dada por: $S = \sqrt{S^2}$. Quanto maior for o valor de S, mais dispersos são estes dados. Se $S = 0$, então não existe variabilidade, isto é, os dados são todos iguais.

CV: O fato do desvio padrão ser expresso na mesma unidade dos dados limita o seu emprego quando se deseja comparar duas ou mais séries de valores, relativamente à sua dispersão ou variabilidade, quando expressas em unidades diferentes. Para contornar esta dificuldade, pode-se caracterizar a dispersão ou variabilidade dos dados em termos relativos a seu valor médio. Esta medida é denominada de CVP (Coeficiente de Variação de Pearson), sendo definida pela expressão: $CV = \frac{S}{\bar{x}}$.

Coeficiente de Assimetria: mede o grau de simetria entre os pontos à esquerda e à direita do ponto central. É obtida a partir do terceiro momento central. É dada por: $Skewness = \frac{1}{n} \sum_{i=1}^{n} (\frac{x_i - \bar{x}}{S})^3$.

Curtose: mede o grau de achatamento de uma distribuição. É obtida a partir do quarto momento central. Dada por: $Kurtosis = \frac{1}{n} \sum_{i=1}^{n} \left[(\frac{x_i - \bar{x}}{S})^4 - 3 \right]$.

Matriz de correlação: é uma matriz quadrada, simétrica que informa dispersão entre dados multivariados. Em cada posição (i, j) da matriz, o coeficiente de correlação entre dados bivariados (x_k, y_k) é informado. O coeficiente de correlação entre duas amostras, denominado r, é dado por:

$$r = \frac{S_{XY}}{\sqrt{S_{XX}}\sqrt{S_{YY}}},$$

onde $S_{XY} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$. Para as posições localizadas na diagonal da matriz, este valor é 1, visto que será um dado do tipo (x_k, x_k).

Ainda é disponibilizada na interface uma opção para salvar os `brushplots` gerados no formato .eps. Em `IDL` esse procedimento é realizado da seguinte maneira:

```
thisDevice = !D.Name
Set_Plot, 'PS'
Device,FileName=file,xsize=8,ysize=8,/inches,xoffset=2.25 $
      ,yoffset=3.5,/ENCAPSULATED

IlustraBrushplot, Nstruct, event.id, val3,0

Device, /Close_File
Set_Plot, thisDevice
```

A primeira linha desse procedimento armazena na variável `thisDevice` o nome do dispositivo ativo. `!D` é uma variável do sistema e equivale a uma estrutura que contém informações sobre os gráficos correntes do dispositivo de saída. O seu campo `Name` guarda uma cadeia de caracteres que representa o nome do dispositivo ativo.

O procedimento `Set_Plot`, cuja sintaxe é : `SET_PLOT, Device`, determina o dispositivo de saída usado pelos procedimentos gráficos do `IDL`. Para salvar no formato .eps, o dispositivo especificado é o `'PS'`, representando o formato `PostScript`. O passo seguinte é definir as características do dispositivo `'PS'` especificado através do procedimento:

```
Device,FileName=file,xsize=8,ysize=8,/inches,xoffset=2.25 $
      ,yoffset=3.5,/ENCAPSULATED+
```

Este procedimento define características do dispositivo como o nome do arquivo onde os gráficos serão salvos, o seu tamanho e o seu formato, nesse caso específico `ENCAPSULATED`, para o formato .eps. Em seguida o `brushplot` é ilustrado, mas agora em outro dispositivo, no caso em um arquivo .eps e não mais no dispositivo de saída padrão, a tela do computador.

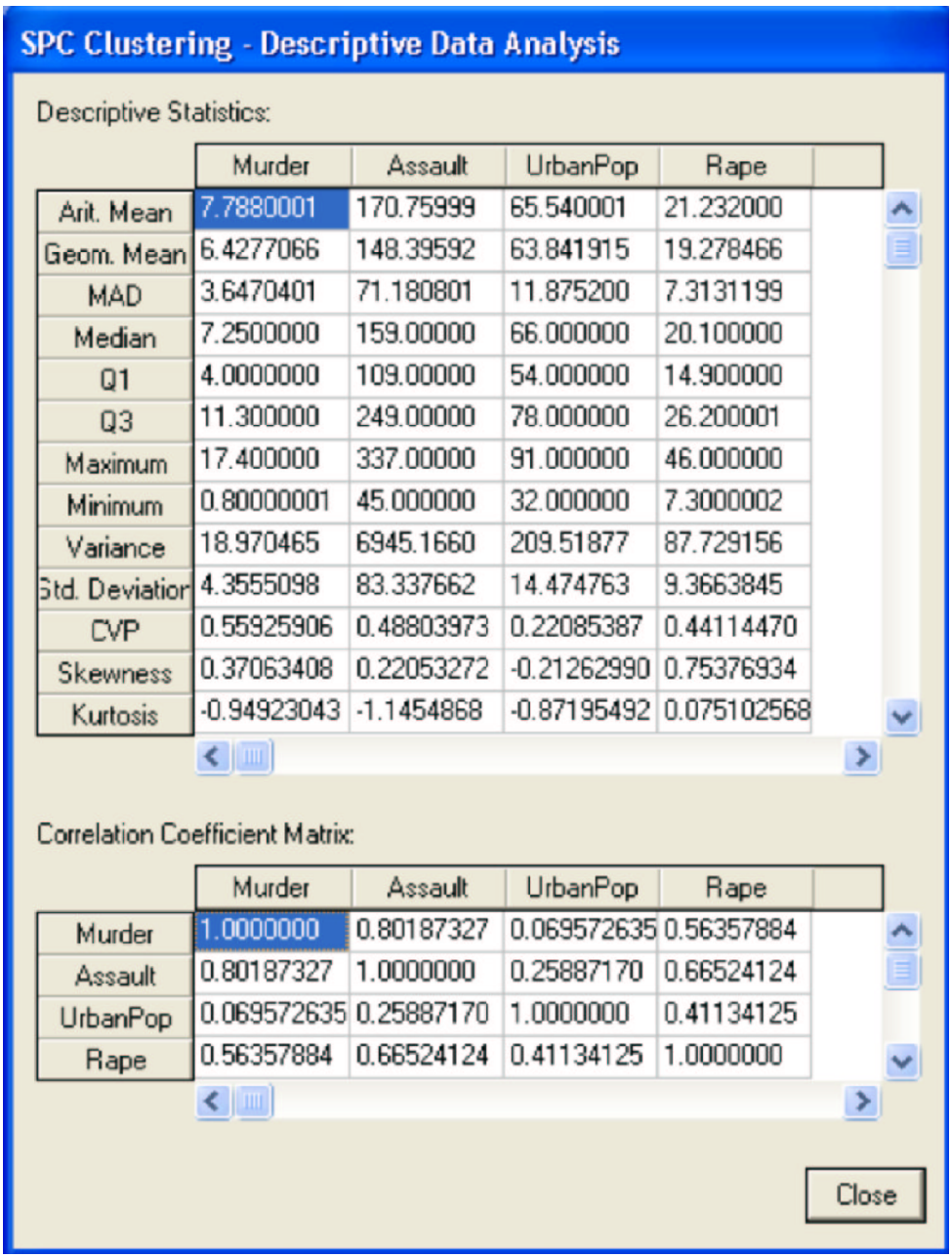


Figura 52: Análise Descritiva do Módulo MDA.

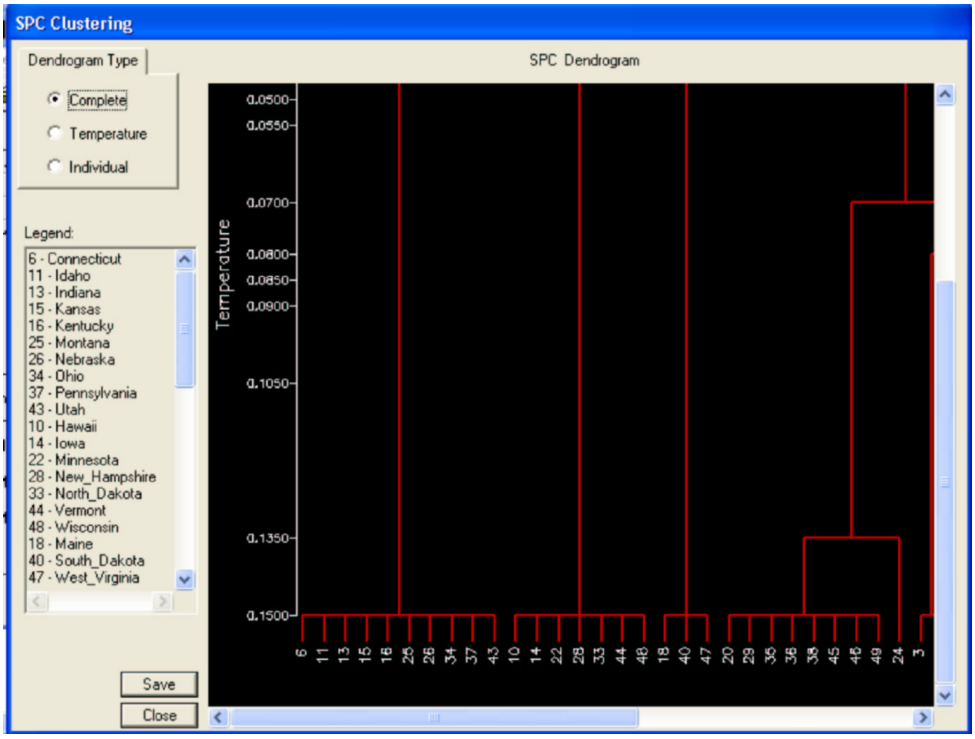


Figura 53: Tela do Módulo Dendrograma.

As duas últimas linhas do procedimento retornam o controle para o dispositivo de saída padrão, anteriormente armazenado na variável `thisDevice`.

A implementação das opções detalhadas nesta seção são exibidas na seção A.1.6 (página 139).

5.5.2.3 Módulo Dendrograma

O módulo Dendrograma é representado pela tela da Figura 53, que também apresenta suas principais sub-funções (ver Tabela 20, página 92).

Este módulo apresenta algumas das mais importantes funções da interface desenvolvida. Seu objetivo principal é exibir, na forma de **dendrogramas**, os resultados obtidos com o programa `SPC`, ou seja, exibir hierarquicamente os agrupamentos.

Como visto na Seção 2.1.2 (página 30), dendrogramas são diagramas bidimensionais que ilustram as fusões ou divisões que acontecem com o conjunto de dados, baseado em alguma técnica. Os dendrogramas desse projeto apresentam no eixo horizontal os indivíduos que serão agrupados e no eixo vertical as temperaturas nas quais ocorrerem os agrupamentos.

A legenda apresentada na Figura 53 exibe os indivíduos na ordem em que eles aparecem no dendrograma. Isto é necessário porque se os verdadeiros nomes dos indivíduos fossem exibidos no gráfico haveria uma grande confusão em qualquer con-

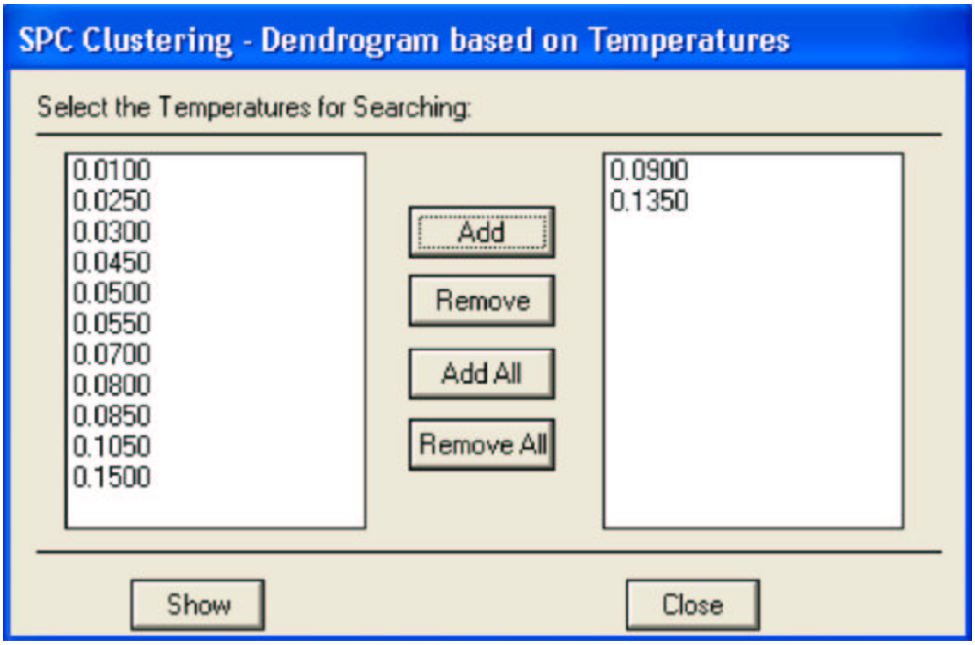


Figura 54: Seleção de temperaturas no módulo dendrograma.

junto de dados consideravelmente grande. Para resolver esse problema, índices foram atribuídos aos indivíduos e estes são mostrados no dendrograma.

Uma outra característica desse módulo é a disponibilidade de três tipos de opções para a visualização dos dendrogramas, são elas: `Complete`, `Temperature` e `Individual`.

Na opção `Complete` (ver Figura 53, página 106), o dendrograma é completamente exibido sem restrições, ou seja, ele é exibido com todos os agrupamentos e todas as temperaturas nas quais ocorreram agrupamentos. A opção `Temperatura` permite a visualização dos agrupamentos que aconteceram a determinadas temperaturas. Ao escolher esta opção, a janela da Figura 54 é exibida para que o usuário selecione as temperaturas que ele deseja que façam parte do seu dendrograma. A opção `Show` da mesma figura aplica as condições, ou seja, as temperaturas escolhidas na construção do dendrograma e o resultado é o gráfico apresentado na Figura 55 (página 108).

A última opção é a `Individual` que exibe a janela da Figura 56 (página 108) para que o usuário selecione um ou mais indivíduos. O dendrograma da Figura 57 (página 109) é um dendrograma completo que destaca todos os agrupamentos que contenham os indivíduos selecionados.

Este módulo, assim como o módulo MDA, também apresenta uma opção "salvar", sendo que esta opção armazena dendrogramas e a outra `brushplots`. O procedimento para salvar esses gráficos é o mesmo apresentado no módulo MDA (ver Seção 5.5.2.2, página 100, para maiores informações).

Uma das opções mais interessantes que este módulo possui é a de nó sensível

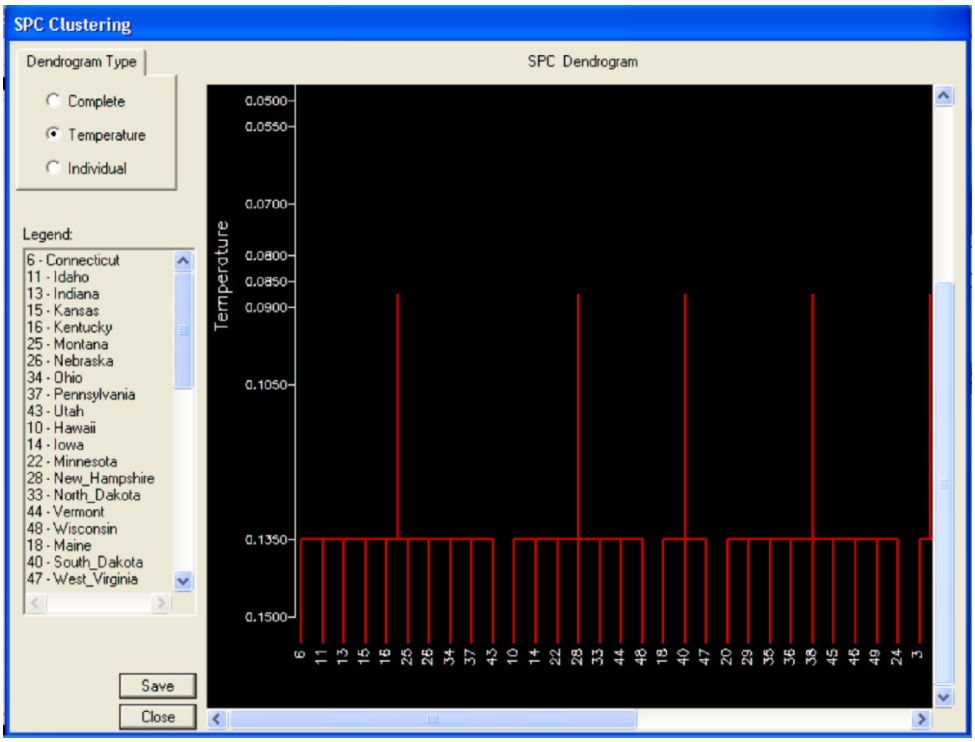


Figura 55: Dendrograma com temperaturas selecionadas.

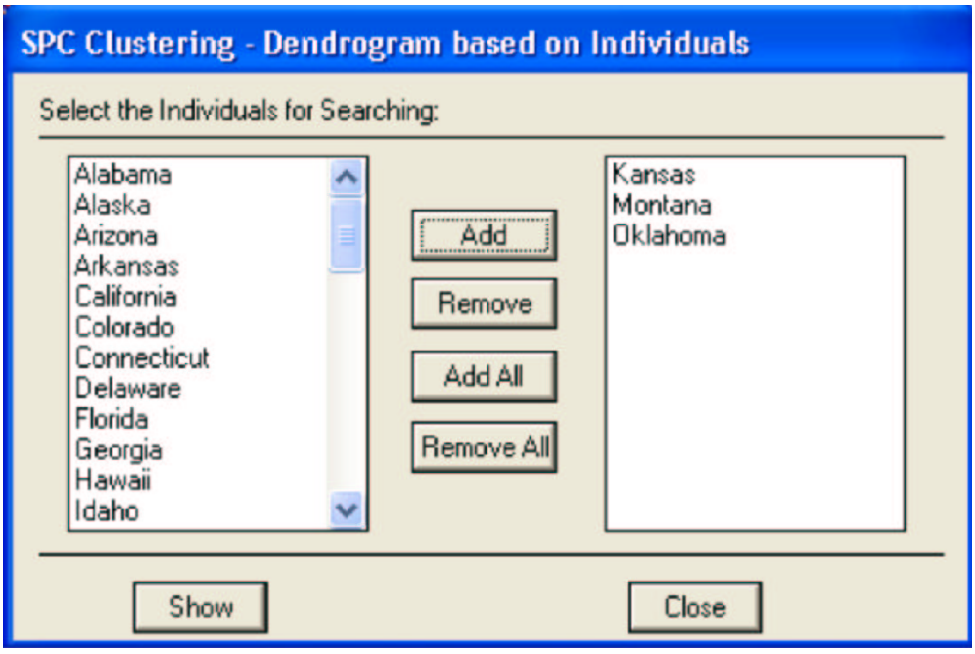


Figura 56: Seleção de indivíduos no módulo dendrograma.

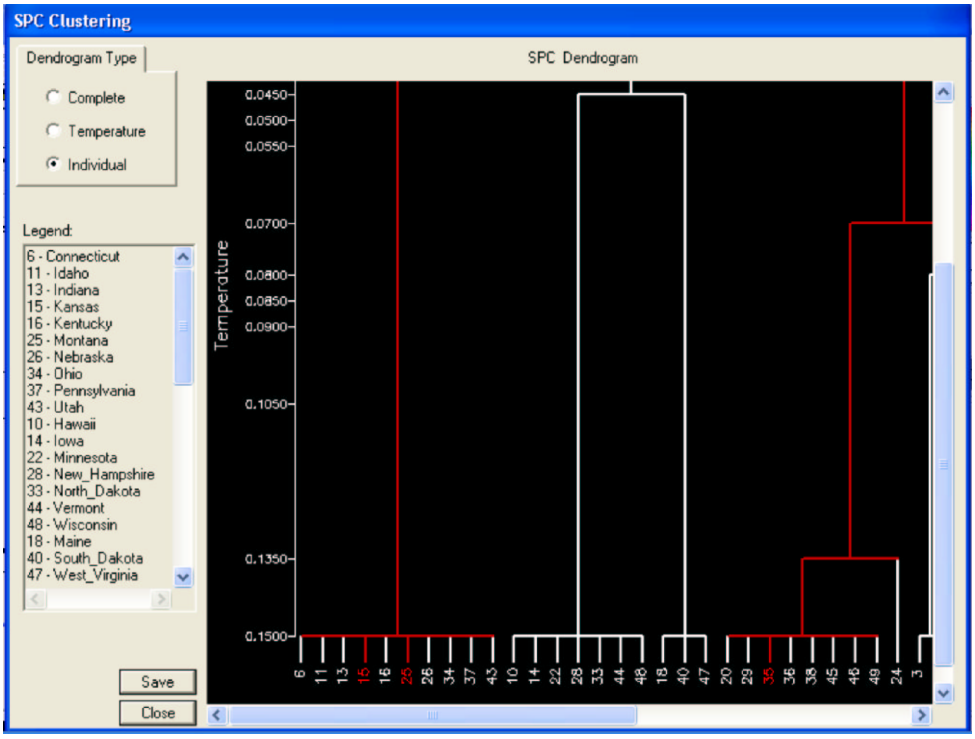


Figura 57: Dendrograma com indivíduos selecionados.

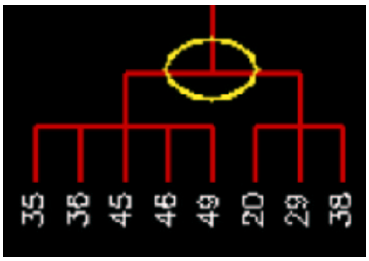


Figura 58: Área do dendrograma sensível ao clique do mouse.

ao clique do mouse. O ponto sensível representado na Figura 58 é um nó do dendrograma e indica que um agrupamento aconteceu. Quando um evento de clique do mouse é efetuado na área exibida são apresentados ao usuário todos os indivíduos que fazem parte daquele agrupamento e uma análise descritiva do conjunto de dados, como exibido anteriormente na Figura 52 (página 105) do módulo MDA. Essas informações podem ser vistas na Figura 59 (página 110), que também disponibiliza ao usuário a opção de visualizar o `brushplot` para o conjunto de dados originado do clique do mouse sobre uma região do dendrograma.

Para construir os dendrogramas foram utilizadas as informações geradas nos arquivos de agrupamento com extensões *.DG_01* e `.LAB`. Para o armazenamento e a geração dos dendrogramas foi utilizada uma representação que permite um acesso mais rápido às informações desses gráficos e que economize memória, respeitando dessa maneira limitações que uma máquina possa ter. As informações extraídas

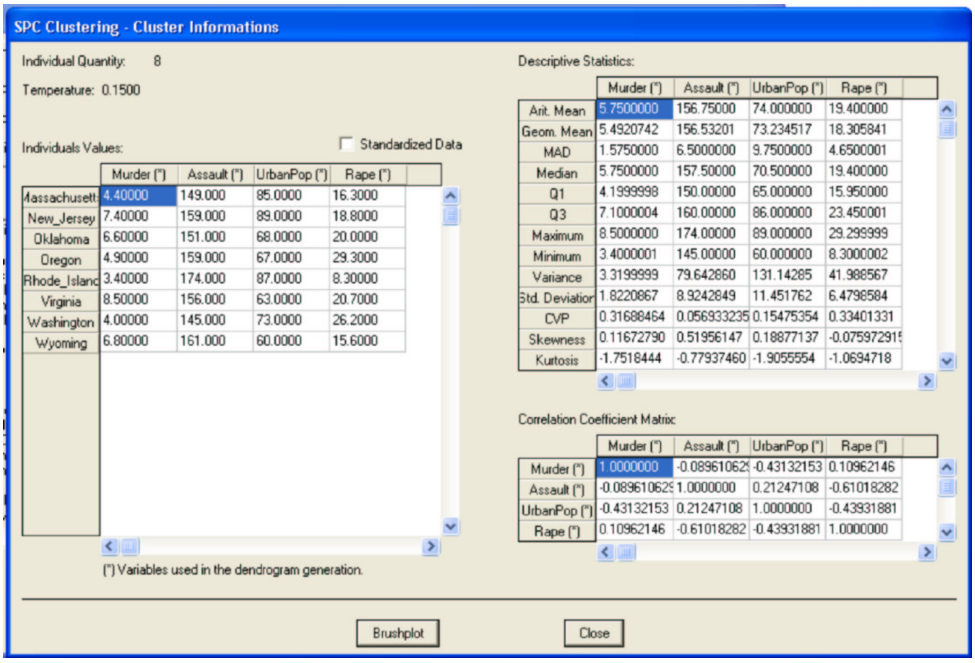


Figura 59: Análise descritiva no módulo dendrograma.

desses arquivos foram armazenadas em estruturas do `IDL`, que seguem os formatos:

ArrTemp: Contém o array de informações por temperatura e apresenta os seguintes campos:

NTemp: índice da temperatura.

VTemp: valor da temperatura.

VMaxClus: grupo de maior valor (com mais indivíduos).

QtdIndClus: quantidade de indivíduos por grupo.

VIndClus: individuo de menor valor por grupo.

e

VIndiv: Contém todos os indivíduos ordenados e apresenta os seguintes campos:

VCluster: todos os grupos ordenados.

QTD_TEMP: quantidade de temperaturas.

QTD_CLUSTER: quantidade total de grupos.

QTD_IND: quantidade de indivíduos.

Dessa maneira, os indivíduos ordenados são armazenados apenas uma vez em VIndiv, na ordem em que aparecem no dendrograma, e para cada temperatura, o índice específico armazena as informações da temperatura em ArrTemp.

Com essas estruturas, é possível trabalhar em todas os módulos que manipulam os resultados, tais como: dendrograma, procura grupo e outros.

O passos seguintes representam o algoritmo utilizado na geração dos dendrogramas.

Algorithm 2

1. Para a primeira temperatura, montar todos os grupos que tenham a quantidade de indivíduos maior que 2 e armazená-los em VIndiv.
2. Verificar se um determinado indivíduo faz parte de um grupo.
3. Verificar se o indivíduo do passo anterior se juntou a outro grupo. Se SIM então unir grupos. Se NÃO, permanecem os grupos separados. O processo de unir grupos: verificar o grupo que está em primeiro lugar na ordem de VIndiv. Ao que estiver em primeiro, acrescenta-se o grupo seguinte e realoca-se o restante dos grupos que estiverem depois dele (inserção em fila).
4. Se todos os indivíduos foram verificados termina o passo, caso contrário volta-se ao passo 3.

A montagem na tela dos dendrogramas fica simples, uma vez que ele foi montado em uma estrutura de dados pelo algoritmo anterior. Dessa forma, a cada tempertatura e grupo lidos são projetadas na tela as linhas verticais e horizontais do dendrograma.

A próxima seção detalha o último dos módulos da interface, o módulo "procura grupo", que é um conjunto de funções auxiliares que objetiva ajudar o usuário na exploração dos agrupamentos obtidos com o `SPC`.

5.5.2.4 Módulo Procura Grupo

Este módulo apresenta algumas funções que são típicas do módulo dendrograma, mas que são usadas aqui para que o usuário não seja forçado a gerar um dendrograma toda vez que quiser informações a respeito de um derteminado agrupamento. Ele funciona exatamente igual ao clique do mouse em uma região de agrupamento no módulo dendrograma, mas especificamente no dendrograma da opção `Individual`.

Quando a opção "Search Cluster" é ativada a janela 56 (página 108) é apresentada ao usuário para que este selecione os indivíduos que estarão presentes no agrupamento que ele procura. Em seguida, a tela de análise descritiva da Figura 59 (página 110) é apresentada ao usuário com informações sobre o último agrupamento contendo os indivíduos selecionados. Dessa maneira, nenhum dendrograma é exibido

e as mesmas informações que o usuário iria obter a partir dele são exibidas de forma mais simples.

O próximo capítulo apresenta uma análise dos dados de teste, onde a eficiência do programa SPC e a simplicidade da interface serão discutidas.

Estado	Assassinato	Assalto	População	Estupro
Alabama	13.2	236	58	21.2
Alasca	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arcansas	8.8	190	50	19.5
Califórnia	9.0	276	91	40.6
Colorado	7.9	204	78	38.7
Connecticut	3.3	110	77	11.1
Delaware	5.9	238	72	15.8
Flórida	15.4	335	80	31.9
Georgia	17.4	211	60	25.8
Havaí	5.3	46	83	20.2
Idaho	2.6	120	54	14.2
Illinois	10.4	249	83	24.0
Indiana	7.2	113	65	21.0
Iowa	2.2	56	57	11.3
Kansas	6.0	115	66	18.0
Kentucky	9.7	109	52	16.3
Louisiana	15.4	249	66	22.2
Maine	2.1	83	51	7.8
Maryland	11.3	300	67	27.8
Massachusetts	4.4	149	85	16.3
Michigan	12.1	255	74	35.1
Minnesota	2.7	72	66	14.9
Mississippi	16.1	259	44	17.1
Missouri	9.0	178	70	28.2
Montana	6.0	109	53	16.4
Nebrasca	4.3	102	62	16.5
Nevada	12.2	252	81	46.0
Nova Hampshire	2.1	57	56	9.5
Nova Jersey	7.4	159	89	18.8
Novo México	11.4	285	70	32.1
Nova Yorque	11.1	254	86	26.1
Carolina do Norte	13.0	337	45	16.1
Dakota do Norte	0.8	45	44	7.3
Ohio	7.3	120	75	21.4
Oklahoma	6.6	151	68	20.0
Oregon	4.9	159	67	29.3
Pensilvânia	6.3	106	72	14.9
Ilha de Rhode	3.4	174	87	8.3
Carolina do Sul	14.4	279	48	22.5
Dakota do Sul	3.8	86	45	12.8
Tennessee	13.2	188	59	26.9
Texas	12.7	201	80	25.5
Utah	3.2	120	80	22.9
Vermont	2.2	48	32	11.2
Virgínia	8.5	156	63	20.7
Washington	4.0	145	73	26.2
Oeste da Virgínia	5.7	81	39	9.3
Wisconsin	2.6	53	66	10.8
Wyoming	6.8	161	60	15.6

Tabela 21: Total de prisões e população urbana em estados dos EUA e outros países.

Pessoa	**Idade**	**Altura**
Mary	35	190
Andrew	40	190
Wagner	35	160
Sandra	40	160

Tabela 22: Idade (em anos) e Altura (em cm) de Quatro Pessoas.

Medida	Valor
$\bar{x}_1$	37.5
s_1	$(2.5 + 2.5 + 2.5 + 2.5)/4 = 2.5$
$\bar{x}_2$	175
s_2	$(15 + 15 + 15 + 15)/4 = 15$

Tabela 23: Resultados Intermediários da Padronização.

Pessoa	Variável 1	Variável 2
Mary	$(35 - 37.5)/2.5$ = **-1**	$(190 - 175)/15$ = **1**
Andrew	$(40 - 37.5)/2.5$ = **1**	$(190 - 175)/15$ = **1**
Wagner	$(35 - 37.5)/2.5$ = **-1**	$(160 - 175)/15$ = **-1**
Sandra	$(40 - 37.5)/2.5$ = **1**	$(160 - 175)/15$ = **-1**

Tabela 24: Idade e Altura Padronizadas para as Pessoas da Tabela 22 (página 114)

6 Análises e Resultados

Este capítulo descreve o comportamento de um conjunto de dados que é utilizado para a obtenção de resultados através da ferramenta desenvolvida. Esses resultados serão analisados e comparados com os de outras duas ferramentas, R e HCE.

6.1 Análise dos dados de teste

Os conceitos, técnicas e ferramentas apresentadas neste trabalho foram aplicados e testados em vários conjuntos de dados. Estes conjuntos possuiam tamanhos variados, desde 20 objetos a 4000 objetos.

Para realizar a análise proposta foi escolhido na literatura o conjunto de dados **Íris**, conhecido também como conjunto de dados de Anderson-Fisher, um popular conjunto de dados que consiste na medida de quatro quantidades em centímetros: comprimento da sépala, largura da sépala, comprimento da pétala e largura da pétala, para cada um dos 150 pontos (flores). Esse conjunto divide ainda seus pontos por espécies, mas essa informação não será considerada nessa análise, apenas as quantidades citadas serão.

O objetivo desta análise é verificar o comportamento do conjunto de dados Íris, utilizando a ferramenta desenvolvida neste projeto, para que comparações e conclusões possam ser estabelecidas na próxima seção.

O conjunto de dados Íris (ver Tabela 25, página 116) possui valores que aproximam-se de dois pontos centrais, ou seja, existem dois grupos que destacam-se quando o conjunto é apresentado na forma gráfica, como pode ser visto no brushplot da Figura 60 (página 117), gerado com a ferramenta desenvolvida neste projeto.

Pode-se verificar nos scatterplots do brushplot a aglomeração dos pontos em duas partes distintas dos gráficos, o que leva a concluir que agrupar esses dados em mais de dois grupos (mais especializados) é uma tarefa complexa e que depende das quatro variáveis que fazem parte do conjunto de dados. Esses agrupamentos serão descritos mais adiante.

Antes de agrupar os dados com a ferramenta desenvolvida é necessária a geração de sua matriz de dissimilaridades, que é um parâmetro obrigatório para a técnica SPC

Objeto	Comp. Sépala	Larg. Sépala	Comp. Pétala	Larg. Pétala
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4
7	4.6	3.4	1.4	0.3
8	5.0	3.4	1.5	0.2
9	4.4	2.9	1.4	0.2
10	4.9	3.1	1.5	0.1
⋮	⋮	⋮	⋮	⋮
140	6.9	3.1	5.4	2.1
141	6.7	3.1	5.6	2.4
142	6.9	3.1	5.1	2.3
143	5.8	2.7	5.1	1.9
144	6.8	3.2	5.9	2.3
145	6.7	3.3	5.7	2.5
146	6.7	3.0	5.2	2.3
147	6.3	2.5	5.0	1.9
148	6.5	3.0	5.2	2.0
149	6.2	3.4	5.4	2.3
150	5.9	3.0	5.1	1.8

Tabela 25: Conjunto de dados Íris

considerada. Para gerar essa matriz foi utilizada a métrica Euclidiana.

Após a obtenção da matriz de dissimilaridades, os agrupamentos são gerados passando-se como parâmetros, além da matriz citada, as seguintes informações: temperatura mínima, temperatura máxima, incremento da temperatura, ciclos de Swendsen-Wang e número de vizinhos mais próximos. O dendrograma da Figura 61 (página 118) é obtido como resultado.

A próxima seção analisa os agrupamentos obtidos para o conjunto de dados Íris e realiza uma comparação entre a ferramenta desenvolvida neste projeto e as ferramentas `R` (Venables & Smith 2001) e `HCE` (Seo & Shneiderman 2002), incluindo as técnicas e as representações gráficas.

6.2 Resultados Obtidos

O conjunto de dados descrito na seção anterior foi utilizado para a obtenção dos resultados que serão apresentados nesta seção e comparados com outros resultados obtidos com o auxílio das ferramentas `R` e `HCE`.

Os parâmetros utilizados na ferramenta desenvolvida para obtenção de resultados foram também empregados na geração de novos resultados junto com as outras duas

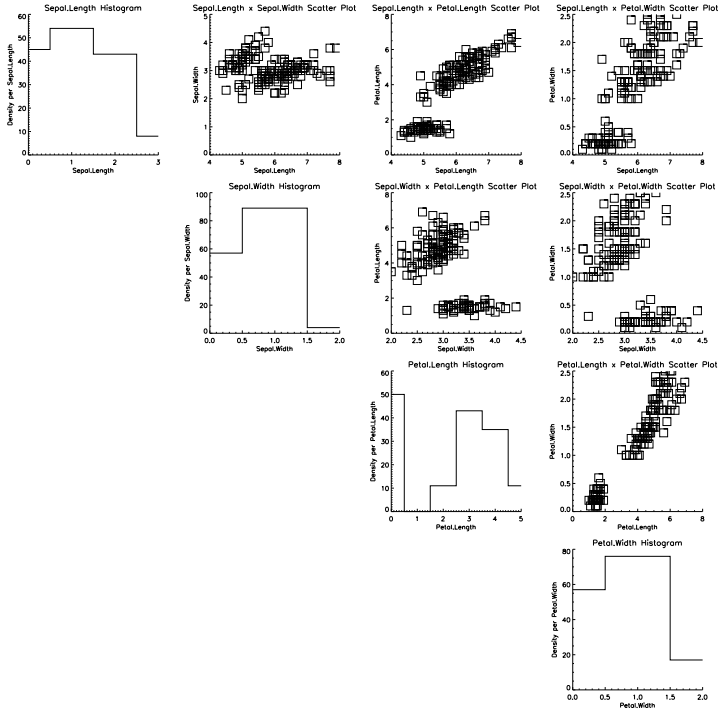


Figura 60: Brushplot do conjunto de dados Íris.

ferramentas em análise. Esses resultados serão exibidos na forma de dendrogramas para todas as ferramentas.

Os resultados obtidos para o conjunto de dados Íris, utilizando a ferramenta desenvolvida neste projeto, foram satisfatórios, assim como em outros conjuntos de dados testados. Quando esses resultados são comparados com os das outras duas ferramentas mencionadas, a parte gráfica visualizada pelo usuário da ferramenta desenvolvida apresenta uma qualidade superior, uma vez que todas as informações exibidas puderam ser facilmente localizadas e armazenadas pelo usuário, além de sua interação com a ferramenta ser maior e mais simples.

Os resultados obtidos com a ferramenta desenvolvida podem ser visualizados no dendrograma da Figura 62(a) (página 119), onde os agrupamentos obtidos são vistos de forma objetiva, ou seja, todos os agrupamentos são bem visualizados e em todos os níveis do gráfico. Ao contrário do dendrograma gerado pela ferramenta desenvolvida, os gerados pelas ferramentas R (Figura 62(b), página 119) e HCE (Figura 62(c), página 119) não exibem os dados de forma clara, uma vez que os dendrogramas gerados com a primeira ferramenta apresentam o erro de sobreposição dos dados (muito comum em outras ferramentas) quando o conjunto de dados a ser agrupado possui um tamanho considerável, impedindo a correta visualização dos resultados; e os dendrogramas gerados com a segunda ferramenta apresentam as alturas dos níveis muito pequenas, impendindo dessa forma que o próprio dendrograma seja visualizado.

A ferramenta desenvolvida também poderá exibir os resuldados apresentados no

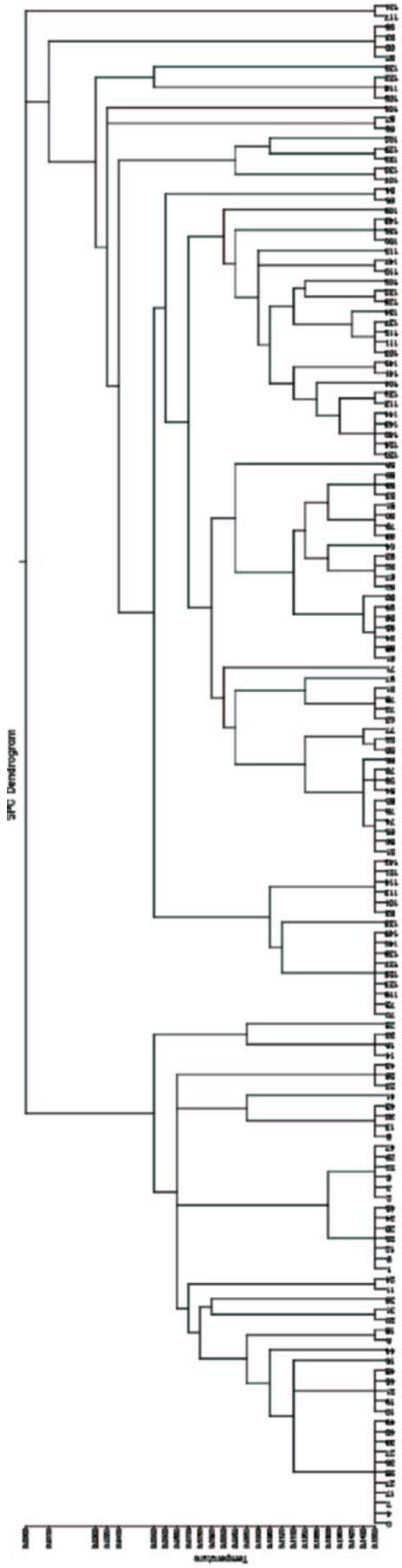
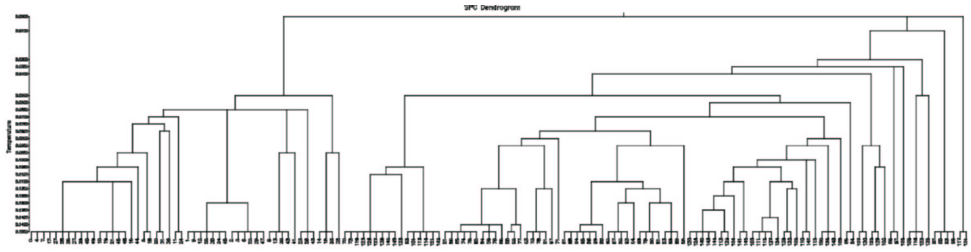
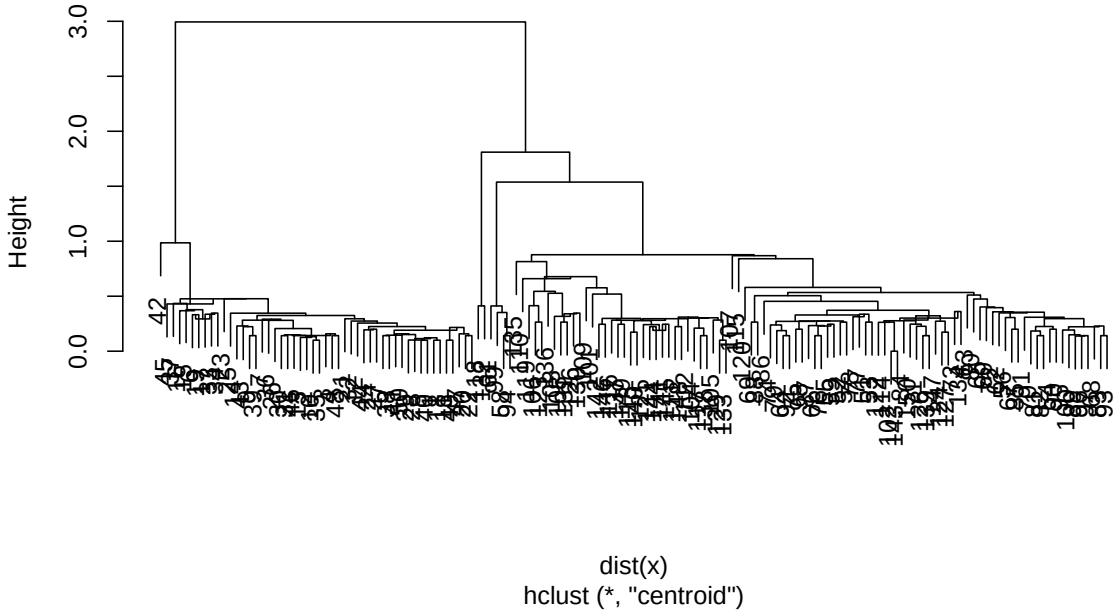


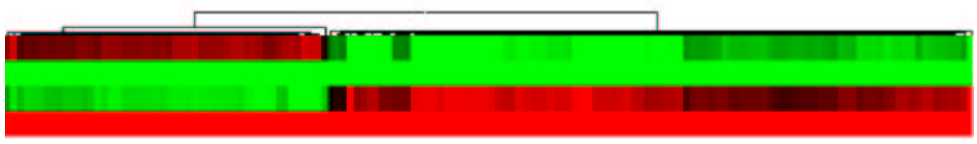
Figura 61: Dendrograma dos agrupamentos do conjunto de dados Íris.



(a) Ferramenta Desenvolvida



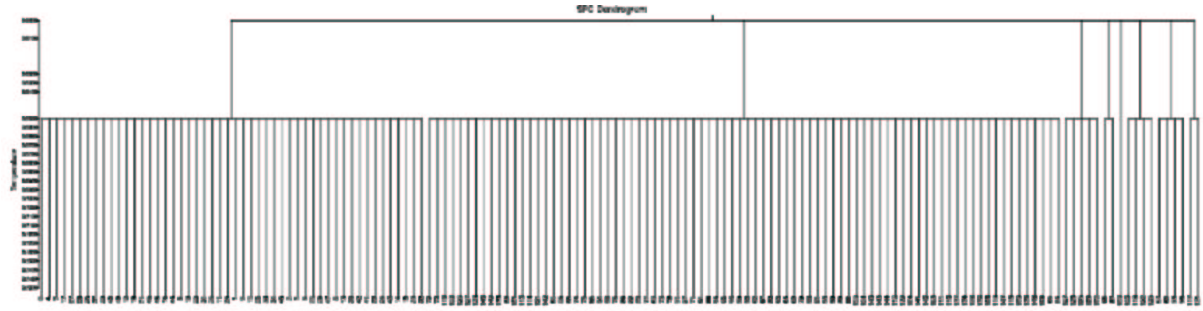
(b) Ferramenta R



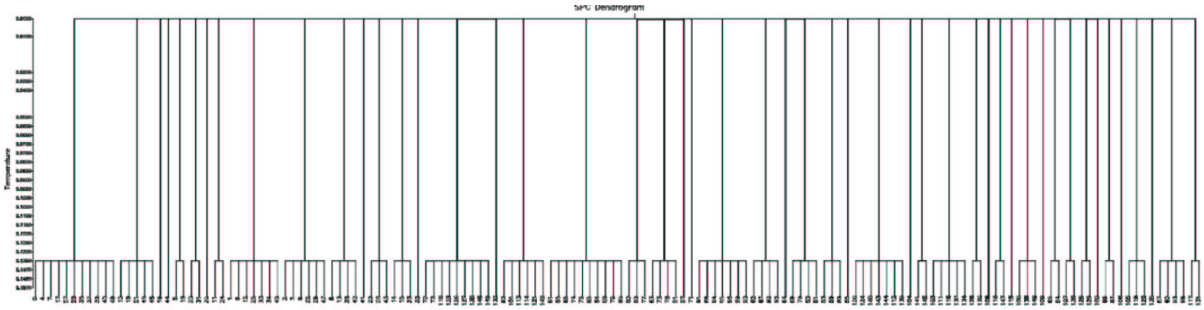
(c) Ferramenta HCE

Figura 62: Dendrogramas dos agrupamentos do conjunto de dados Íris.

dendrograma da Figura 62(a) (página 119) separados por níveis, como pode ser visto na Figura 63. Esses cortes na árvore do dendrograma são úteis por permitirem que níveis não importantes possam ser ocultados e apenas os níveis necessários à analise do usuário sejam exibidos e, dessa forma, não haja confusão por causa da quantidade de informação apresentada.



(a) Níveis 0 e 5



(b) Níveis 0 e 21

Figura 63: Diferentes níveis para um mesmo dendrograma.

Outros resultados obtidos a partir dos dendrogramas da ferramenta desenvolvida são os que fazem parte da análise descritiva dos dados pertencentes aos agrupamentos. Quando algum nó do dendrograma é selecionado, através do clique do mouse, é exibida ao usuário essa análise descritiva contendo os indíviduos que fazem parte do agrupamento, em que nível eles se agruparam, algumas medidas estatísticas (médias, quartis, variância, máximo, mínimo, curtose, entre outras) e uma matriz de correlação das váriaveis do conjunto de dados, ver Figura 64 (página 121). Essas medidas permitem que o usuário tenha um conhecimento maior sobre os dados que estão sendo analisados. A ferramenta R também permite que tais medidas sejam calculadas, mas o usuário terá que calcular cada uma delas individualmente e através de funções disponíveis na ferramenta que deverão ser executadas através de linha de comando.

Para cada grupo formado no dendrograma é possível, além da análise descritiva mostrada, visualizar um brushplot para cada grupo selecionado. De acordo com o resultado exibido na Figura 64 (página 121), o brushplot equivalente para tal grupo seria o apresentado na Figura 65 (página 121).

Dessa maneira é possível obter uma visão completa e geral do conjunto de dados e de seus grupos, uma vez que para cada grupo os resultados das Figuras 64

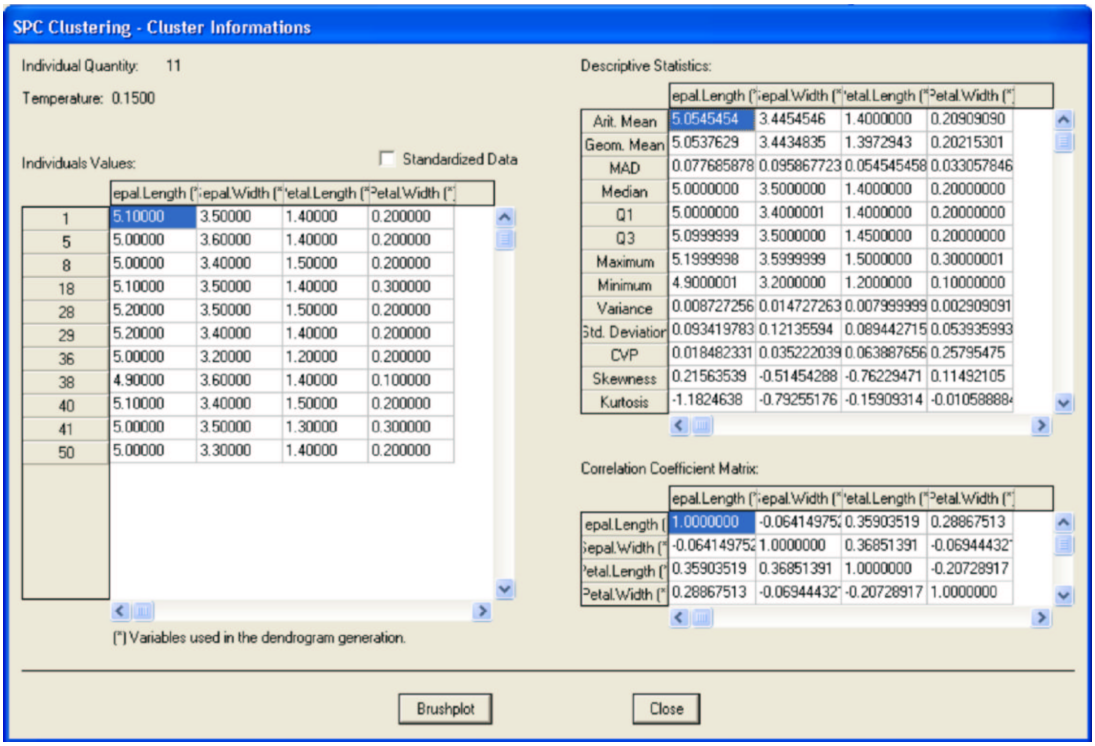


Figura 64: Análise Descritiva para um grupo do conjunto de dados Íris.

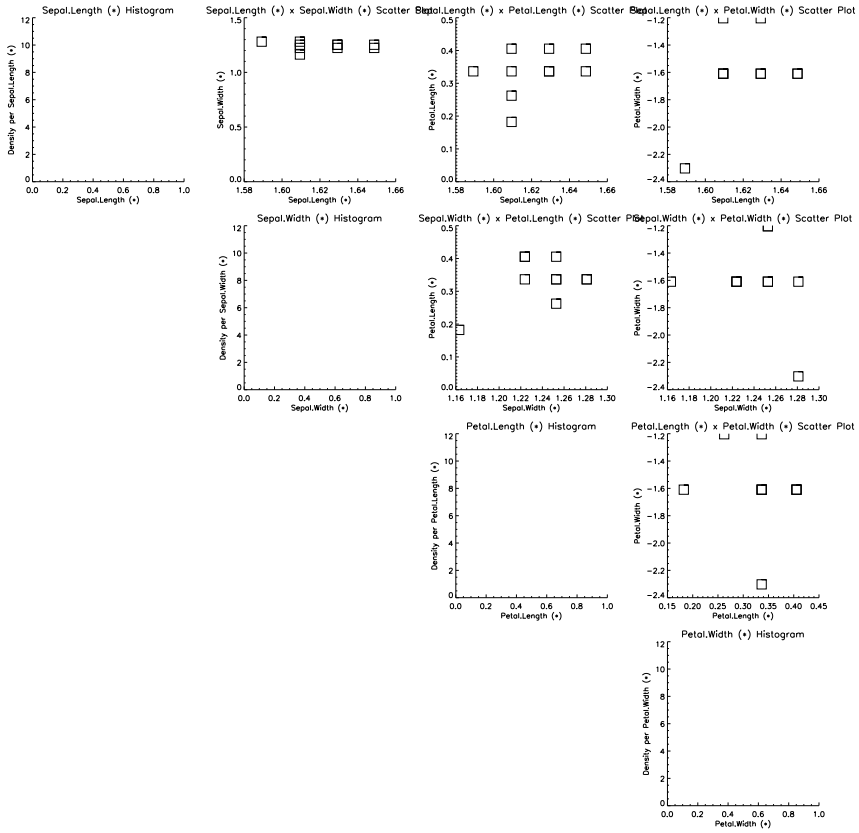


Figura 65: Brushplot para o grupo da Figura 64 (página 121).

(página 121) e 65 (página 121) são fáceis e interativamente exibidos aos usuários do sistema.

Após a verificação dos resultados obtidos para a interface desenvolvida é interessante discutir também os resultados obtidos para a nova técnica de agrupamento SPC. Segundo Blatt et al. (1997), foram realizados testes para verificação da eficiência no agrupamento de seis conjuntos de dados, entre eles o Íris, incluindo sua característica de espécies. Os 150 pontos desse conjunto de dados foram divididos em três espécies, sendo que 50 deles pertencentes à espécie *setosa*, outros 50 pertencentes à espécie *virgínica* e outros 50 à espécie *multicores*.

A partir dos resultados obtidos por Blatt et al. (1997), foi possível verificar que o conjunto de dados Íris, durante o processo de agrupamento, é divido em grupos em dois estágios. Isto reflete o fato de duas das três espécies possuirem valores mais aproximados uma da outra do que da terceira. A técnica SPC trabalha com tal organização hierárquica muito bem, visto que 125 pontos do conjunto de dados foram agrupados corretamente (como comparado com agrupamento manual) e 25 pontos não foram classificados. Pode ser observado no dendrograma da Figura 62(a) (página 119) que um agrupamento muito semelhante ao gerado por Blatt et al. (1997) foi obtido, uma vez que 21 dos 150 pontos não foram agrupados (grupos isolados com apenas um indíviduo). Isto ocorreu por causa dos diferentes parâmetros utilizados na realização dos agrupamentos.

Os melhores resultados de todos os algoritmos de agrupamento utilizados por Blatt et al. (1997) na verificação da eficência da técnica SPC estão sumarizados na Tabela 26.

Técnica	Maior Grupo	Grupo Médio	Menor Grupo
Árvore de Cobertura Mínima	50	50	50
SPC	45	40	38
Procura do vale	67	42	37
Junção Completa	81	39	30
Grafo Direcionado	90	30	30
K-Vizinhos compartilhados	90	30	30
Junção Simples	101	30	19
Valor de vizinhança mútua	101	30	19

Tabela 26: Resultados dos agrupamentos para as oito técnicas testadas.

Apenas as técnicas de Árvore de Cobertura Mínima e a SPC retornaram grupos onde os pontos pertencentes a diferentes espécies não foram misturados, ou seja, ou os pontos foram agrupados corretamente ou não foram agrupados.

Uma vez apresentados os resultados obtidos para a ferramenta desenvolvida neste projeto e a nova técnica de agrupamento utilizada, o próximo capítulo aponta as principais conclusões verificadas neste projeto.

7 Conclusões e Trabalhos Futuros

Como visto nos capítulos anteriores, o agrupamento de grandes conjuntos de dados é uma tarefa complexa e que exige dedicação e paciência. Analisar esses dados e seus resultados é uma tarefa praticamente impossível sem a ajuda de uma máquina. Os diversos algoritmos de agrupamento de dados encontrados na literatura, aliados ao avanço tecnológico, permitiram automatizar essa análise e encontrar resultados precisos e de maneira rápida.

Mesmo existindo uma diversidade grande de algoritmos de agrupamento, nem todos eles podem ser aplicados a qualquer conjunto de dados. É preciso que os dados possuam algumas características que a técnica a ser empregada no agrupamento exija. Dessa forma, quando o conjunto de dados apresenta um grande volume, poucos algoritmos podem ser empregados em seu agrupamento, e alguns desses não realizam sua tarefa de maneira satisfatória.

Nos testes realizados durante o desenvolvimento do projeto e os realizados por Blatt et al. (1997), foi verificado que a nova técnica SPC apresentou resultados satisfatórios para todos os conjuntos de dados que foram utilizados, sejam eles pequenos ou grandes. Nos testes, algumas vezes essa nova técnica não apresentou o melhor dos resultados, mas ficou entre as três melhores técnicas que apresentaram. Devido a esses resultados, é interessante que tal técnica continue sendo pesquisada e melhorada para que se torne conhecida e solucione uma variedade maior de problemas.

Utilizar algo novo na maioria das vezes é uma tarefa complicada, pois é algo que é preciso aprender bem para que não haja dúvidas sobre os passos a serem seguidos. Para solucionar esse problema, a interface desenvolvida foi utilizada para disseminar a nova técnica e, principalmente, auxiliar o seu usuário de tal forma que ele faça o menor esforço possível pra obtenção dos resultados. É evidente as vantagens que o sistema desenvolvido neste projeto e o desenvolvido em Horta (2004) proporcionaram às pessoas interessadas nas áreas de agrupamentos de dados e imagens.

O trabalho desenvolvido por Horta (2004) possui a mesma base deste trabalho, ou seja, o algoritmo de agrupamento de dados baseado no comportamento super-paramagnético do modelo de Potts. O primeiro trabalho, especializou-se na área de segmentação de imagens, enquanto o segundo, na área de agrupamento de dados propriamente descrita. Os sistemas desses dois projetos são os únicos a implementar esse novo algoritmo e a abordar funcionalidades diferentes com base nos tipos de

dados utilizados.

A parte visual do sistema deste projeto permite que o usuário praticamente não digite, mas que apenas faça escolhas visuais. Ela ainda fornece diversos recursos para manipulação dos dados e um, em particular, que não é comumente visto em ferramentas desse tipo, a sensibilidade ao clique do mouse proporcionada pelos gráficos dendrogramas. Essa característica quase nunca é encontrada em ferramentas e, quando é, apresenta-se de forma precária ou de difícil acesso, como é o caso da ferramenta R, que exige um conhecimento profundo de sua linguagem, pelo usuário, para realizar essa função. Neste trabalho a ferramenta R foi utilizada como uma fonte de recursos, onde os conjuntos de dados, usados nos exemplos, foram retirados dessa ferramenta facilmente, devido às suas funções de carga de dados. Resultados obtidos com ela também ajudaram na comparação realizada com a ferramenta desenvolvida neste projeto.

A ferramenta desenvolvida apesar de apresentar várias funções já implementadas, deve ainda ser melhorada e acrescentar novas funcionalidades, como por exemplo a inclusão de novos gráficos, de novas técnicas de agrupamento e de uma opção que permita a comparação entre os resultados obtidos entre duas ou mais técnicas.

Desenvolver esta ferramenta e aplicá-la à técnica SPC permitiu o desenvolvimento e a aprendizagem de conhecimentos de áreas como Estatística, Física, Análise de dados e Computação. Além disso, contribuiu para o crescimento profissional das pessoas que participaram do seu projeto.

Referências Bibliográficas

Ahuja, N. (1982), 'Dot pattern processing using Voronoi neighborhood', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **4**, 336–343. PAMI.

Blatt, M., Wiseman, S. & Domany, E. (1996), 'Superparamagnetic clustering of data', *Physical Review Letters* **76**(18), 3251–3254.

Blatt, M., Wiseman, S. & Domany, E. (1997), 'Data clustering using a model granular magnet', *Neural Computation* **9**, 1805–1842.

Bradley, P. S. & Fayyad, U. M. (1998), Refining initial points for k-means clustering, *in* 'Proceedings 15th International Conf. on Machine Learning', Morgan Kaufmann, San Francisco, CA, pp. 91–99.

Bustos, O. H. & Frery, A. C. (1992*a*), 'Reporting Monte Carlo results in statistics: suggestions and an example', *Revista de la Sociedad Chilena de Estadística* **9**(2), 46–95.

Bustos, O. H. & Frery, A. C. (1992*b*), *Simulação Estocástica: Teoria e Algoritmos (versão completa)*, Monografias de Matemática, 49, CNPq/IMPA, Rio de Janeiro, RJ.

Bustos, O. H., Frery, A. C. & Ojeda, S. (1998), 'Strong Markov processes in image modelling', *Brazilian Journal of Probability and Statistics – REBRAPE* **12**(2), 149–194.

Devroye, L., Györfi, L. & Lugosi, G. (1996), *A Probabilistic Theory of Pattern Recognition*, number 31 *in* 'Applications of Mathematics', Springer-Verlag, New York.

Domany, E. (1999), 'Superparamagnetic clustering of data: The definitive solution of an ill-posed problem', *Physica A* **263**, 158–169.

Domany, E., Blatt, M., Gdalyahu, Y. & Weinshall, D. (1999), 'Superparamagnetic clustering of data: Application to computer vision', *Computer Physics Communications* **121–122**, 5–12.

Duda, R. O., Hart, P. E. & Stork, D. G. (2001), *Pattern Classification*, 2 edn, John Wiley & Sons, New York.

Everitt, B. S., Landau, S. & Leese, M. (2001), *Cluster Analysis*, 4 edn, Arnould, New York.

Fukunaga, K. (1990), *An Introduction to Statistical Pattern Recognition*, Computer Science and Scientific Computing, 2 edn, Academic, San Diego.

Furlan, J. D. (1998), *Modelagem de Objetos através da UML - The Unified Modeling Language*, Makron Books, São Paulo.

Geman, D. & Geman, S. (1984), 'Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(6), 721–741.

Gordon, A. D. (1999), *Classification*, Chapman & Hall/CRC, New York.

Gower, J. C. (1985), *Measures of Similarity, Dissimilarity and Distance*, Encyclopaedia of Statistical Sciences, Volume 5, Wiley, New York.

Gower, J. & Legendre, P. (1986), 'Metric and euclidean properties of dissimilarity coeficients', *Journal of Classification* **5**, 5–48.

Halkidi, M. & Vazirgiannis, M. (2001), A data set oriented approach clustering algorithm selection, *in* 'Proceedings of PKDD Conference', Freiburg, Germany.

Horta, M. M. (2004), Sistema para Segmentação de Imagens por Agrupamento Hierárquico baseado no Comportamento Superparamagnético do Modelo de Potts, Dissertação de mestrado em ciência da computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, PE.

IDL: Interactive Data Language (2003). Última consulta em janeiro de 2004. *http://www.rsinc.com/idl

Johnson, R. A. & Wichern, D. W. (1992), *Applied Multivariate Statistical Analysis*, 3 edn, Prentice Hall, New Jersey.

Kaufman, L. & Rousseeuw, P. J. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley & Sons, New York.

Lance, G. & Willians, W. (1966), 'Computer programs for hierarchical polithetic classification', *Computer Journal* **9**, 60–64.

Lang, K. R. (1992), *Astrophysical Data: Planets and Stars*, Springer Verlag Berlin and Heidelberg GmbH & Co. KG, Massachusetts, USA.

Larson, R. & Sadiq, G. (1983), 'Facility locations with the Manhattan metric in the presence of barriers to travel', *Operation Research* **31**, 652–699.

McQueen, J. (1967), 'Some methods for classification and analysis of multivariate observations', *5-th Berkeley Symposium on mathematics* **1**, S. 281–298.

Metropolis, N., Rosembluth, A. W., Rosembluth, M. N., Teller, A. H. & Teller, E. (1953), 'Equations of state calculations by fast computing machines', *Journal of Chemical Physics* **21**, 1087–1092.

Metropolis, N. & Ulam, S. (1949), 'The Monte Carlo method', *Journal of the American Statistical Association* **44**, 335–341.

Morris, S. A., Asnake, B. & Yen, G. G. (2003), 'Dendrogram seriation using simulated annealing', *Information Visualization* **2**(2), 95–104.

Peña, J. M., Lozano, J. A. & Larrañaga, P. (1999), 'An empirical comparison of four initialization methods for the k-means algorithm', *Pattern Recognition Letters* **20**, 1027–1040.

Pickard, D. K. (1987), 'Inference for discrete Markov fields: The simplest nontrivial case', *Journal of the American Statistical Association* **82**(1), 90–96.

Pompilho, S. (2002), *Análise Essencial - Guia prático de Análise de Sistemas*, Editora Ciência Moderna, Rio de Janeiro.

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge.

Seo, J. & Shneiderman, B. (2002), 'Interactively exploring hierarchical clustering results', *IEEE Computer* **35**(7), 80–86.

Swendsen, R. (1991), 'Acceleration methods for Monte Carlo computer simulations', *Computer Physics Communications* **65**(1/3), 281–288.

Swendsen, R. & Wang, J. (1987), 'Nonuniversal critical dynamics in Monte Carlo simulations', *Physical Review Letters* **58**(2), 86–88.

Venables, W. N. & Smith, D. M. (2001), *An introduction to R*, Network Theory Limited, UK.

Wiseman, S., Blatt, M. & Domany, E. (1998), 'Superparamagnetic clustering of data', *Physical Review E* **57**(4), 3767–3783.

Wu, F. Y. (1982), 'The Potts model', *Reviews of Modern Physics* **54**(1), 235–268.

Yourdan, E. (1990), *Análise Estruturada Moderna*, Editora Campus, Rio de Janeiro.

APÊNDICE - Anexos

Neste capítulo estão apresentados os principais trechos de códigos-fonte utilizados nas alterações realizadas no programa SPC e no desenvolvimento da interface apresentada nesta dissertação.

A.1 Códigos em IDL (*Interactive Data Language*)

A.1.1 Procedimento Abrir Arquivo de Dados

Procedimento utilizada para abrir o arquivo de dados escolhido, no formato .txt, e carregá-lo para a grade principal, além de armazenar temporariamente todas as informações necessárias a respeito do conjunto de dados sendo aberto.

```
pro btOpen, Event
  ;Dialog Open File
  IF (file = DIALOG_PICKFILE(/READ, FILTER = '*.txt'))
    THEN BEGIN
    a = ''
    cont = 0
    pos = 0
    NV = 0

    aux = ''

    ;Open File  TXT
    IF (strpos(STRUPCASE(file),'.TXT') NE -1) THEN BEGIN
      OPENR, fp, file, /GET_LUN
      WIDGET_CONTROL,Event.ID, /HOURGLASS
      READF, fp, a

      IF (a EQ 'mgtxt') THEN BEGIN

        WHILE NOT EOF(fp) DO BEGIN
```

```
    READF, fp, a
    a = STRCOMPRESS(a)
    IF (cont EQ 0) THEN BEGIN ;Firts line of data set
        ;get number of variables
        pos=STRPOS(a, ' ')
        NV = FIX(STRMID(a, 0, pos))
        a = STRMID(a, pos+1, STRLEN(a))

        variaveis = STRARR(NV)

        variaveis = obtemVar(NV, a)
        valores    = FLTARR(1,NV)
    ENDIF ELSE BEGIN ;just individuals and values

        a = STRCOMPRESS(a)
        WHILE (STRMID(a, 0, 1) EQ ' ')  DO BEGIN
            a = STRMID(a, 1, STRLEN(a))
        ENDWHILE

        aux = aux + STRMID(a, 0 , STRPOS(a, ' ')) + ' '
        a = STRMID(a, STRPOS(a, ' ')+1, STRLEN(a))

        obtemIndiVal , a, NV, cont, valores

        valores = CONGRID(valores, cont+1, NV)

    ENDELSE

    cont = cont + 1
ENDWHILE
individuos = STRARR(cont)
obtemInd ,individuos, cont, aux

StrNVal = STRARR(cont,NV)
j = 0
k = 0
WHILE (j LT cont) DO BEGIN
    WHILE (k LT NV) DO BEGIN
        IF ((valores(j,k)) EQ -9999) THEN BEGIN
            StrNVal(j,k) = ''
        ENDIF ELSE BEGIN
            StrNVal(j,k) = $
```

```
            STRCOMPRESS(STRING(valores(j,k))$
                 , /REMOVE_ALL)
        ENDELSE
        k = k + 1
    ENDWHILE
    k = 0
    j = j + 1
ENDWHILE

WIDGET_CONTROL, Event.top, GET_UVALUE=stash

structval = {valores:valores, StrNVal:StrNVal}
WIDGET_CONTROL, stash.table, SET_VALUE=StrNVal  $
             ,SET_UVALUE=structval  $
             ,ROW_LABELS=individuos $
             ,COLUMN_LABELS = variaveis, XSIZE=NV $
             ,YSIZE=cont-1,/TABLE_DISJOINT_SELECTION
WIDGET_CONTROL,stash.text0 $
             ,SET_VALUE=STRCOMPRESS(STRING(cont-1)$
             ,/REMOVE_ALL)

struct = {var:variaveis, NI: cont, NV:NV$
 , valores:valores$
 , indiv: individuos$
 , filename:obtemFileName(file)}

WIDGET_CONTROL, stash.DM, SET_UVALUE=struct

WIDGET_CONTROL, stash.DM, SENSITIVE=1
WIDGET_CONTROL, stash.imCloseF, SENSITIVE=1
WIDGET_CONTROL, stash.imMD, SENSITIVE=1
WIDGET_CONTROL, stash.imMDA, SENSITIVE=1
WIDGET_CONTROL, stash.imDendro, SENSITIVE=1
WIDGET_CONTROL, stash.imSearchC, SENSITIVE=1

WIDGET_CONTROL, stash.btadm, SENSITIVE=1
WIDGET_CONTROL, stash.btCloseFile, SENSITIVE=1
WIDGET_CONTROL, stash.btnClose, SENSITIVE=1
WIDGET_CONTROL, stash.btGo, SENSITIVE=1
WIDGET_CONTROL, stash.btStand, SENSITIVE=1
WIDGET_CONTROL, stash.btDendro, SENSITIVE=1
WIDGET_CONTROL, stash.btSearch, SENSITIVE=1
```

```
        CLOSE, fp
      ENDIF ELSE BEGIN
        a = 0
        a = Dialog_Message("Invalid File Format!"$
            , /INFORMATION)
      ENDELSE
    ENDIF
  ENDIF
END
```

A.1.2 Procedimento Fechar Arquivo de Dados

Procedimento utilizada para fechar o arquivo de dados atualmente em uso. Este procedimento libera ainda todas as unidades lógicas em uso pelo programa.

```
pro CloseFile , Event

    WIDGET_CONTROL, Event.top, GET_UVALUE=stash

    WIDGET_CONTROL, stash.DM, SENSITIVE=0
    WIDGET_CONTROL, stash.imCloseF, SENSITIVE=0
    WIDGET_CONTROL, stash.imMD, SENSITIVE=0
    WIDGET_CONTROL, stash.imMDA, SENSITIVE=0
    WIDGET_CONTROL, stash.imMDA, SENSITIVE=0
    WIDGET_CONTROL, stash.imDendro, SENSITIVE=0
    WIDGET_CONTROL, stash.imSearchC, SENSITIVE=0
    WIDGET_CONTROL, stash.btadm, SENSITIVE=0
    WIDGET_CONTROL, stash.btCloseFile, SENSITIVE=0
    WIDGET_CONTROL, stash.btnClose, SENSITIVE=0
    WIDGET_CONTROL, stash.btGo, SENSITIVE=0
    WIDGET_CONTROL, stash.btStand, SENSITIVE=0
    WIDGET_CONTROL, stash.btDendro, SENSITIVE=0
    WIDGET_CONTROL, stash.btSearch, SENSITIVE=0
    WIDGET_CONTROL, stash.text0, SET_VALUE='0'
    WIDGET_CONTROL, stash.text1, SET_VALUE='0.00'
    WIDGET_CONTROL, stash.text2, SET_VALUE='0.15'
    WIDGET_CONTROL, stash.text3, SET_VALUE='0.005'
    WIDGET_CONTROL, stash.text4, SET_VALUE='4000'
    WIDGET_CONTROL, stash.text5, SET_VALUE='5'
    WIDGET_CONTROL, stash.table, SET_VALUE=['']$
     , ROW_LABELS=['0']$
```

```
                  , COLUMN_LABELS = ['0'], XSIZE=1$
                  , YSIZE=1$
                  , /TABLE_DISJOINT_SELECTION
end
```

A.1.3 Função Padronizar Dados

Função utilizada para realizar os cálculos necessários à padronização do conjunto de dados em uso.

```
function PadronizaDados,valores,NV,NI
   indicesNAN = WHERE(valores EQ -9999)
   tam = size(indicesNAN,/N_ELEMENTS)

   means = FLTARR(NV)
   meansabsdev = FLTARR(NV)
   novosvalores = FLTARR(NI,NV)
   novosvalorestr = STRARR(NI,NV)
   i=0
   WHILE i LT NV DO BEGIN
      means(i) = MEAN(valores(0:NI-1,i),/NaN)
      meansabsdev(i) = MEANABSDEV(valores(0:NI-1,i),/NaN)
      i = i + 1
   ENDWHILE

   i= 0
   WHILE (i LT tam-1) DO BEGIN
      valores(indicesNAN(i)) = -9999
      i = i + 1
   ENDWHILE

   i=0
   WHILE i LT NV DO BEGIN
      j=0.0
      WHILE j LT  NI DO BEGIN
         IF (valores(j,i) NE -9999) THEN BEGIN
            IF (meansabsdev(i) NE 0) THEN BEGIN
                novosvalores(j,i) = (valores(j,i) $
                                     - means(i))/meansabsdev(i)
                novosvalorestr(j,i) = $
                  STRCOMPRESS(String(novosvalores(j,i))$
                  , /REMOVE_ALL)
```

```
            ENDIF ELSE BEGIN
                novosvalores(j,i) = -9999
                novosvalorestr(j,i) = ''
            ENDELSE
         ENDIF ELSE BEGIN
            novosvalores(j,i) = -9999
            novosvalorestr(j,i) = ''
         ENDELSE
         j = j + 1
      ENDWHILE
      i = i + 1
   ENDWHILE

   structnovos = {nvalores:novosvalores $
                  ,strnvalores:novosvalorestr}
   return, structnovos
end
```

A.1.4 Procedimentos para Gerar Matriz de Dissimilaridade

```
PRO CalculaCanberra, Event
    WIDGET_CONTROL, Event.top, GET_UVALUE=stash
    WIDGET_CONTROL, stash.DM, GET_UVALUE=struct

    NVsel = INTARR(struct.NV)

    i = 0
    ; Retrieve the table's selection mode and selection.
    disjoint = WIDGET_INFO(stash.table$
             , /TABLE_DISJOINT_SELECTION)
    selection = WIDGET_INFO(stash.table, /TABLE_SELECT)

    IF (selection(0) EQ -1) THEN begin
        WIDGET_CONTROL, stash.table, SET_TABLE_SELECT=[[0,0]
                      ,[struct.NV-1,struct.NI-2]]
    endif

    WIDGET_CONTROL, stash.table, GET_VALUE=value$
                  , /USE_TABLE_SELECT

    i = 0
```

```
dimensaoSel = size(selection)
valsel = INTARR(struct.NV)

i = 0
WHILE (i LT struct.NV) DO BEGIN
    valsel(i) = -1
    i = i + 1
ENDWHILE

valsel(0) = selection(0,0)

i = 1
x = 0

IF (dimensaoSel(0) NE 1) THEN BEGIN

    o = 0
    while (o LT struct.NV) DO BEGIN
        NVsel(o) = -1
        o = o + 1
    ENDWHILE

    o = 0

    totsel = 0
    result = WHERE(selection(0,0) EQ NVsel)
    WHILE (result(0) EQ (-1)) DO BEGIN
        NVsel(o) = selection(0,o)
        o = o + 1
        result = WHERE(selection(0,o) EQ NVsel)
    endwhile

    totsel = o

    WHILE (i LT struct.NV) DO BEGIN
        x = WHERE(valsel EQ selection(0,i))
        IF (x(0) EQ -1) THEN BEGIN
                ;print, 'entrou'
            valsel(i) = selection(0,i)
        ENDIF ELSE BEGIN
            break;
        ENDELSE
```

```
        i = i + 1
    ENDWHILE

    novoval = 0
    i = 0
    WHILE (valsel(i) NE -1) DO BEGIN
        novoval = novoval + 1
        i = i + 1
        IF (i GE struct.NV) THEN break;
    ENDWHILE
        ;print, valsel

        ;print, size(selection)

    novosval = FLTARR(struct.NI-1, novoval)

        ;print, 'novoval = ' + string(novoval)
    i = 0
    j = 0
    WHILE (i LT novoval) DO BEGIN
        WHILE (j LT struct.NI-1) DO BEGIN
            novosval(j,i) = struct.valores(j,valsel(i))
            j = j + 1
        ENDWHILE
        j = 0
        i = i + 1
    ENDWHILE
ENDIF

IF (dimensaoSel(0) NE 1) THEN BEGIN
    widget_control,event.id,/hourglass
    NI=struct.NI
    valores=novosval
    p=novoval
    totsel=totsel
    NVsel=NVsel
ENDIF ELSE BEGIN
    WIDGET_CONTROL, stash.table,TABLE_DISJOINT_SELECTION=0
    WIDGET_CONTROL, stash.table,SET_TABLE_SELECT=[0,0 $
                  ,struct.NV-1,struct.NI-2]
    WIDGET_CONTROL, stash.table,TABLE_DISJOINT_SELECTION=1
    totsel = struct.NV
```

```
    i = 0
    while (i LT struct.NV) DO BEGIN
        NVSel(i) = i
        i = i +1
    ENDWHILE
    widget_control,event.id,/hourglass
    NI=struct.NI
    valores=struct.valores
    p=struct.NV
    totsel=totsel
    NVsel=NVsel
ENDELSE

;CALCULA A DISTÂNCIA E SALVA NUM ARQUIVO COM EXTENSÃO DATA
indGrid = WHERE(NVsel EQ -1)

IF (NVsel(n_elements(NVsel)-1) EQ -1) THEN BEGIN
    qtdVar  = n_elements(NVsel)$
              -n_elements(WHERE(NVsel EQ -1))
ENDIF ELSE BEGIN
    qtdVar = n_elements(NVsel)
ENDELSE

varGrid = INTARR(qtdVar)
i = 0
while (i lt qtdVar) do begin
    varGrid(i) = NVsel(i)
    i = i + 1
endwhile

rNI= NI-1
DM = FLTARR(10*(rNI),2)
DM[*,0]=-1

IF (file = DIALOG_PICKFILE(/WRITE, FILTER = '*.data')) $
   THEN BEGIN

IF (STRPOS(file,'.data') EQ -1) THEN BEGIN
    file = file + '.data'
ENDIF

OPENW, fp, file, /GET_LUN
```

```
printf, fp, string(qtdVar)
printf, fp, string(varGrid)
nome = ObtemFileName(file)

val = TRANSPOSE(valores)
FOR a=0L,rNI-1 DO BEGIN
    FOR b=0L,rNI-1 DO BEGIN
        CASE 1 OF
            (a LT b): BEGIN
                    indice = WHERE(val[*,a] NE -1)
                    IF (indice[0] GE 0) THEN BEGIN
                        indice1 = WHERE(val[indice,b]$
                                  NE -1,count1)
                        IF (indice1[0] GE 0) THEN BEGIN
                            x=abs(val[indice1,a] $
                              -val[indice1,b])$
                              /(abs(val[indice1,a])$
                              +abs(val[indice1,b]))
                        ENDIF
                    ENDIF
                    vDM=TOTAL(x,/NAN)
                    PRINTF, fp, vDM
                    indice=WHERE(DM[*,0] EQ -1)
                    DM[indice[0],0]=a*rNI+b
                    DM[indice[0],1]=vDM
                END
            (a GT b): BEGIN
                    indice=WHERE(DM[*,0] EQ b*rNI+a)
                    PRINTF, fp, DM[indice[0],1]
                    DM[indice[0],0]=-1
                END
            (a EQ b): PRINTF, fp, 0.0000
        ENDCASE
    ENDFOR
ENDFOR

a = 0
CLOSE, fp
a = Dialog_Message("Sucessfully Generated$
 Dissimilarity Matrix!"$
     , /INFORMATION)
```

```
    ENDIF
END
```

A.1.5 Função Executar SPC

```
pro SalvaRun, Event
    IF (filedata = DIALOG_PICKFILE(/READ, FILTER = '*.data')) $
       THEN BEGIN

    OPENR, fp1, filedata, /GET_LUN
    path = ''
    path = FILEPATH(filedata)
    pathAtual = ''
    pathAtual = obtemPathReal(path)
    cd , pathAtual

    file = obtemFileName(filedata) + '.run'
    widget_control,event.id,/hourglass

    WIDGET_CONTROL, Event.top, GET_UVALUE=stash

    OPENW, fp2, file, /GET_LUN

    WIDGET_CONTROL, stash.text0, GET_VALUE=val
    PRINTF, fp2, 'NumberOfPoints:  ' + val
    PRINTF, fp2, 'DataFile:  ' + filedata
    PRINTF, fp2, 'Dimentions: 0'
    WIDGET_CONTROL, stash.text1, GET_VALUE=val2
    PRINTF, fp2, 'MinTemp: ' + val2
    WIDGET_CONTROL, stash.text2, GET_VALUE=val3
    PRINTF, fp2, 'MaxTemp: ' + val3
    WIDGET_CONTROL, stash.text3, GET_VALUE=val4
    PRINTF, fp2, 'TempStep: ' + val4
    PRINTF, fp2, 'OutFile:          miga'
    WIDGET_CONTROL, stash.text4, GET_VALUE=val5
    PRINTF, fp2, 'SWCycles: ' + val5
    WIDGET_CONTROL, stash.text5, GET_VALUE=val6
    PRINTF, fp2, 'KNearestNeighbours: ' + val6
    PRINTF, fp2, 'MSTree|                    '
    PRINTF, fp2, 'DirectedGrowth| '
    PRINTF, fp2, 'SaveSuscept|'
    PRINTF, fp2, 'WriteLables|'
```

```
    PRINTF, fp2, 'DataIsMatrix| '
    PRINTF, fp2, 'WriteCorFile~ '


    CLOSE, fp1
    CLOSE, fp2

    widget_control,event.id,/hourglass
    spawn , 'SW ' + file
    a = Dialog_Message("Sucessfully Realized Grouping!",$
        /INFORMATION)
    a = Dialog_Message("Generate Dendrogram?", /QUESTION)

    ;cria dendrograma imediatamente
    IF (a EQ 'Yes') THEN BEGIN
        ChamaDendro,event,1,pathAtual,file
    ENDIF
    ENDIF
end
```

A.1.6 Módulo MDA

```
pro wADM,Event,type,current_window
    i = 0

    WIDGET_CONTROL, Event.top, GET_UVALUE=stash
    WIDGET_CONTROL, stash.table, GET_VALUE=valorest,$
       /TABLE_DISJOINT_SELECTION
    ;tentar descobrir a posicao de cada valor
    WIDGET_CONTROL, stash.DM, GET_UVALUE=struct

    ; Retrieve the table's selection mode and selection.
    disjoint = WIDGET_INFO(stash.table$
             , /TABLE_DISJOINT_SELECTION)
    selection = WIDGET_INFO(stash.table, /TABLE_SELECT)
    IF (selection(0) EQ -1) THEN begin
        WIDGET_CONTROL, stash.table, SET_TABLE_SELECT=[[0,0]$
                      ,[struct.NV-1,struct.NI-2]]
    endif

    WIDGET_CONTROL, stash.table, GET_VALUE=value$
                  , /USE_TABLE_SELECT
```

```
dimensaoSel = size(selection)

valsel = INTARR(struct.NV)
i = 0
WHILE (i LT struct.NV) DO BEGIN
    valsel(i) = -1
    i = i + 1
ENDWHILE
valsel(0) = selection(0,0)

i = 1
x = 0

IF (dimensaoSel(0) NE 1) THEN BEGIN
    WHILE (i LT struct.NV) DO BEGIN
        x = WHERE(valsel EQ selection(0,i))
        IF (x(0) EQ -1) THEN BEGIN
            valsel(i) = selection(0,i)
        ENDIF ELSE BEGIN
            break;
        ENDELSE
        i = i + 1
    ENDWHILE

    novoval = 0
    i = 0
    WHILE (valsel(i) NE -1) DO BEGIN
        novoval = novoval + 1
        i = i + 1
        IF (i GE struct.NV) THEN break;
    ENDWHILE
ENDIF

checks = INTARR(struct.NV)

xWindow = WIDGET_BASE(TITLE="SW Clustering - " $
          +"Multivariate Data Analysis", TLB_FRAME_ATTR=2$
          ,XSIZE=790, YSIZE=540,XOFFSET=5,YOFFSET=5)
WIDGET_CONTROL, /MANAGED, xWindow

If type then BEGIN
    btDesc = WIDGET_BUTTON(xWindow$
```

```
            , VALUE='Descriptive Statistics'$
            , UVALUE='Desc',XOFFSET=340,YOFFSET=510$
            , XSIZE=120, YSIZE=20,FRAME=1)
ENDIF

xBase = WIDGET_BASE(xWindow, COLUMN=2,/NONEXCLUSIVE$
       ,YOFFSET=30)
wInfoItem = WIDGET_BUTTON(xWindow, VALUE='Close'$
           , UVALUE='EXIT', XOFFSET=720,YOFFSET=510$
           ,XSIZE=60, YSIZE=20,FRAME=1)
wInfoItem2 = WIDGET_BUTTON(xWindow, VALUE='Close'$
            , XOFFSET=720,YOFFSET=510,XSIZE=60$
            , YSIZE=20)
btAux  = WIDGET_BUTTON(xWindow, VALUE='Close'$
        ,XOFFSET=720,YOFFSET=510,XSIZE=60, YSIZE=20)
btSalvar = WIDGET_BUTTON(xWindow, VALUE='Save'$
          ,UVALUE='Salvar',XOFFSET=650,YOFFSET=510$
          , XSIZE=60, YSIZE=20,FRAME=1)
WID_DRAW_0 = Widget_Draw(xWindow, UNAME='WID_DRAW_0'$
            ,XOFFSET=180,YOFFSET=24 ,SCR_XSIZE=600 $
            ,SCR_YSIZE=480 ,RETAIN=1)

WID_LABELvar = Widget_Label(xWindow,VALUE='Selected' $
              +'Variables Brushplot',YOFFSET=4$
              ,XOFFSET=400);
btBrush = WIDGET_BUTTON(xWindow, VALUE='BrushPlot'$
         ,UVALUE ='Brush',XOFFSET=180,YOFFSET=510$
         ,XSIZE=60, YSIZE=20,FRAME=1)
btBrush2 = WIDGET_BUTTON(xWindow, VALUE='BrushPlot2'$
          ,XOFFSET=180,YOFFSET=510, XSIZE=60, YSIZE=20$
          ,FRAME=1)
btConf = WIDGET_BUTTON(xWindow, VALUE='Configuration'$
        ,UVALUE='Conf',XOFFSET=250,YOFFSET=510,XSIZE=80$
        ,YSIZE=20,FRAME=1)

labelvar   = WIDGET_LABEL(xWindow, VALUE='Variables:'$
            ,UVALUE='EXIT', XOFFSET=5,YOFFSET=5,XSIZE=60$
            ,YSIZE=20,/ALIGN_LEFT)
labeltraco = WIDGET_LABEL(xWindow, VALUE='_____________'$
            +'_____________', UVALUE='EXIT', XOFFSET=5$
            ,YOFFSET=15, YSIZE=20,/ALIGN_LEFT)
```

```
i = 0
IF (dimensaoSel(0) NE 1) THEN BEGIN
    WIDGET_CONTROL, btBrush2, SET_UVALUE=novoval

    WHILE i LT novoval DO BEGIN
        checks (i) = Widget_Button(xBase$
                    , UNAME='WID_BUTTON_'$
                    +STRING(i),/ALIGN_LEFT $
                    ,VALUE=struct.var(valsel(i))$
                    , UVALUE=0)


        WIDGET_CONTROL, checks(i), /SET_BUTTON
        WIDGET_CONTROL, checks(i), SET_UVALUE=Event.Select
        i = i + 1
    ENDWHILE
    totsel=i
    ;Armazena variáveis para visulização automática
    valsel1=valsel
    novoval1 = novoval

ENDIF ELSE BEGIN
    WIDGET_CONTROL, btBrush2, SET_UVALUE=struct.NV

    WHILE i LT struct.NV DO BEGIN

        checks(i) = Widget_Button(xBase$
                    ,UNAME='WID_BUTTON_'$
                    +STRING(i),/ALIGN_LEFT $
                    ,VALUE=struct.var(i),UVALUE=0)

        WIDGET_CONTROL, checks(i), /SET_BUTTON
        WIDGET_CONTROL, checks(i), SET_UVALUE=Event.Select
        i = i + 1
    ENDWHILE
    totsel=i
    i = 0
    WHILE (i LT struct.NV) DO BEGIN
        valsel(i) = -1
        i = i + 1
    ENDWHILE
```

```
    ;Armazena variáveis para visulização automática
    valsel1=INDGEN(struct.NV)
    novoval1 = struct.NV

ENDELSE

EstPlotConfig = {HistBin:1, HistBinTxt:1.0$
      , HistLineType:0,$
      HistLineThick:1.0, HistFontSize:1.0, $
      HistFontType:'!3', HistFontThick:1.0$
      ,HistAxisType:0,$
      HistAxisThick:1.0,ScatterVariable: 1,$
      ScatterSymbolType:6, ScatterSymbolSize:1.0,$
      ScatterFontSize:1.0, ScatterFontType:'!3',$
      ScatterFontThick:1.0, ScatterAxisType:0,$
      ScatterAxisThick:1.0}
WIDGET_CONTROL, btAux, SET_UVALUE = EstPlotConfig

IF type THEN BEGIN
    NStruct = {btB2:btBrush2,checks:checks,$
            valores:struct.valores, NI:struct.NI,$
            NV:struct.NV,variaveis:struct.var,$
            valsel:valsel,dimensaoSel:dimensaoSel,$
            table:stash.table,draw:WID_DRAW_0,$
            btAux:btAux,type:type,totsel:totsel}
 ENDIF ELSE BEGIN
    NStruct = {btB2:btBrush2,checks:checks,$
            valores:struct.valores, NI:struct.NI,$
            NV:struct.NV,variaveis:struct.var,$
            valsel:valsel,dimensaoSel:dimensaoSel,$
            table:stash.table,btAux:btAux,$
            current_window:current_window,type:type,$
            totsel:totsel}
 ENDELSE

WIDGET_CONTROL, xWindow, SET_UVALUE=NStruct

IF (dimensaoSel(0) EQ 1) THEN BEGIN
    WIDGET_CONTROL, stash.table$
                  ,TABLE_DISJOINT_SELECTION=0
    WIDGET_CONTROL, stash.table$
                  ,SET_TABLE_SELECT=[0,0,struct.NV-1$
```

```
                    ,struct.NI-2]
    ENDIF
    WIDGET_CONTROL, stash.table,TABLE_DISJOINT_SELECTION=1

    WIDGET_CONTROL, /REALIZE, xWindow
    XManager, "examples", xWindow$
            , EVENT_HANDLER="WindowEventHdlr", /NO_BLOCK
end
```

A.2 Códigos implementados em C

Os principais trechos de código fonte incluídos nos arquivos do programa SPC são listados abaixo.

A.2.1 Estrutura de dados utilizada no arquivo Sw.c

```
typedef struct no {
    int nT;
    float T;
    int nc;
    unsigned int change;
    unsigned int *Bloco;
    unsigned int *dgOldBloco;
    long *ClusterSize;
    struct no *next;
    struct no *prior;
} l_no;

l_no *cabeca,*novo, *atual,*auxiliar;
```

A.2.2 Armazenamento dos agrupamentos em Sw.c

```
if (nT == 0) {
  printf("nT = %d", nT);
  cabeca->next = cabeca->prior = NULL;
  cabeca->nT = nT;
  cabeca->nc = nc;
  cabeca->T = T;
  cabeca->Bloco = InitUIVector(N);
  cabeca->dgOldBloco = InitUIVector(N);
  cabeca->ClusterSize = InitUIVector(N);
```

```
  for (i=0;i < N;i++) {
    cabeca->Bloco[i] = Block[i];
    cabeca->dgOldBloco[i] = dgOldBlock[i];
    cabeca->change = 0;
  }

  i=0;
  while (ClusterSize[i])
    i++;

  for (i=0;i < nc;i++) {
    cabeca->ClusterSize[i] = ClusterSize[i];
  }
}
else {
  if ((novo = malloc(sizeof(l_no)))==NULL)
    printf("erro de malloc! - nT <> 0");
  novo->Bloco  = InitUIVector(N);
  novo->dgOldBloco = InitUIVector(N);
  novo->ClusterSize = InitUIVector(N);
  for (i=0;i < N;i++) {
    novo->Bloco[i] = Block[i];
    novo->dgOldBloco[i] = dgOldBlock[i];
    if (Block[i] != dgOldBlock[i]) {
      novo->change = 1;
    }
  }

  if (novo->change==1) ++nchanges;

  for (i=0;i < nc;i++) {
    novo->ClusterSize[i] = ClusterSize[i];
  }
  novo->nT = nT;
  novo->nc = nc;
  novo->T = T;
  atual = cabeca;
  if (cabeca->next != NULL) {
    while (atual->next) {
      atual=atual->next;
    }
```

```
  }

  novo->prior  = atual;
  novo->next   = NULL;
  atual->next = novo;
}
```

A.2.3 Geração dos arquivos de saída

```
//foi para o final dos agrupamentos
atual = cabeca;
while (atual->next) {
  atual=atual->next;
}

mudancas = 1;

myfiledg  = fopen("dg_01.txt","a");      //clustersize
myfilelab = fopen("dg_01.lab.txt","a");  //block

fprintf(myfiledg,"%s  ", qtdVar);
fprintf(myfiledg,"\n");
for (i=0; i < (atoi(qtdVar));i++) {
  fprintf(myfiledg,"%d ", indVar[i]);
}

fprintf(myfiledg,"\n");
fprintf(myfiledg,"%3d  ", nchanges+1);
nchanges = 0;
fprintf(myfiledg,"%3d  ", nc);
fprintf(myfiledg,"%3d  ", N);
fprintf(myfiledg,"\n");

//segue da temperatura mais alta para a mais baixa
while (atual->prior) {
  if (mudancas == 1) {
    fprintf(myfiledg,"%3d  ", atual->nT);
    fprintf(myfiledg,"%8.5f  ", atual->T);
    for(i = 0; i < nc; i++) {
      if (atual->ClusterSize[i] < 0) atual->ClusterSize[i] = 0;
        fprintf(myfiledg, "%4d ", atual->ClusterSize[i]);
    }
```

```
    fprintf(myfiledg,"\n");
    for(i = 0; i < N; i++) {
      fprintf(myfilelab, "%4d ", atual->Bloco[i]);
    }
    fprintf(myfilelab,"\n");
  }
  if (atual->change == 1) mudancas = 1;
  else mudancas = 0;

  atual = atual->prior;
}

fprintf(myfiledg,"%3d  ", atual->nT);
fprintf(myfiledg,"%8.5f  ", atual->T);

for(i = 0; i < nc; i++) {
  if (atual->ClusterSize[i] < 0) atual->ClusterSize[i] = 0;
  fprintf(myfiledg, "%4d ", atual->ClusterSize[i]);
}

for(i = 0; i < N; i++) {
  fprintf(myfilelab, "%4d ", atual->Bloco[i]);
}

fclose(myfiledg);
fclose(myfilelab);
```