UM NOVO ALGORITMO PARA RESOLUÇÃO DE PROBLEMAS DE CLASSIFICAÇÃO

Alberto de Oliveira Moreno

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Dr. Adilson Elias Xavier, D.Sc.

Dr. Alexandre Pinto Alves da Silva, Ph.D.

Dr. Nelson Maculan Filho, D.Sc.

Dr. Luiz Fernando Loureiro Legey, Ph.D.

Dr. Nélio Domingues Pizzolato, Ph.D.

Dr. Luiz Mariano Paes de Carvalho, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

FEVEREIRO DE 2008

MORENO, ALBERTO DE OLIVEIRA

Um novo algoritmo para resolução de problemas de classificação [Rio de Janeiro] 2008

XIII, 84 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 2008)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Algoritmos de classificação

I. COPPE/UFRJ II. Título (série)

Dedico esta dissertação a meus pais,
cujo exemplo de honestidade e trabalho
tem sido um norteador para a minha vida,
e para minha esposa, que tem
me dado apoio nos momentos mais difíceis
e mostrado a simplicidade de ter esperança.

Agradecimentos

Agradeço ao estimado prof. Adilson Elias Xavier, orientador deste trabalho, pelas competentes contribuições ao longo de todo o desenvolvimento da tese de doutorado e pelas valiosas sugestões, que auxiliaram de forma expressiva na obtenção dos resultados. Em especial, é necessário destacar a contribuição do prof. Adilson na proposição de um algoritmo eficiente para a solução do problema, baseado na suavização hiperbólica e a sua dedicação e empenho para o desenvolvimento de um software adequado e eficiente para a implementação do algoritmo proposto.

Agradeço ao prof. Alexandre Pinto, co-orientador deste estudo, pelas valiosas sugestões para o desenvolvimento do trabalho.

Agradeço ao doutorando da COPPE, Michael Pereira de Souza pela colaboração e suporte na utilização do algoritmo para obter resultados confiáveis em diversos casos usados para a validação de nosso método.

Agradeço aos alunos do Mestrado da COPPE-UFRJ, Victor Strole e Patrícia Curvelo pelas sugestões no desenvolvimento do trabalho e em especial pela colaboração no desenvolvimento do sistema computacional para a implementação do algoritmo de solução proposto no desenvolvimento do trabalho.

Agradeço a minha mãe Maria de Lourdes de Oliveira Moreno, pelo apoio e confiança na obtenção de resultados alcançados com este trabalho.

Agradeço a minha esposa Nilza Carvalho Moreno pela paciência e compreensão da minha dedicação ao trabalho, em sacrifício da minha participação na solução de problemas de casa.

Agradeço aos meus filhos Patrícia Moreno Schuhmann e seu esposo Dieter Schuhmann e a Alexandre Carvalho Moreno e sua esposa Nathália Beserra de Souza Moreno pelo estimulo a realização do trabalho.

Agradeço ao meu neto recém-nascido Bernardo Moreno Schuhmann pela inspiração na realização deste trabalho.

Finalmente, não poderia deixar de expressar ao meu neto Victor de Souza Moreno os agradecimentos pelos momentos de extrema felicidade proporcionados pela convivência permanente, que ajudaram a aliviar as incertezas e tensões que ocorreram ao longo do desenvolvimento do trabalho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do Grau de Doutor em Ciências (D. Sc.)

UM NOVO ALGORITMO PARA RESOLUÇÃO DE PROBLEMAS DE CLASSIFICAÇÃO

Alberto de Oliveira Moreno

Fevereiro/2008

Orientadores: Adilson Elias Xavier
Alexandre Pinto Alves da Silva

Programa: Engenharia de Sistemas e Computação

Neste trabalho considera-se uma nova metodologia de solução para o problema de classificação associado à análise de agrupamentos. A formulação matemática clássica para este problema é baseada em modelos de otimização não - diferenciáveis, que são resolvidos por métodos que carecem de precisão e eficiência. Neste trabalho, o problema é aproximado por uma formulação suavizadora hiperbólica que conduz a um problema diferenciável, cuja solução pode ser obtida com maiores robustez e eficiência por métodos clássicos e mais poderosos de otimização. A eficiência e a precisão do método são comprovadas através de experiências numéricas aplicadas a vários problemas teste apresentados na literatura.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requeriments for the degree of Doctor of Science (D. Sc.)

A NEW ALGORITHM TO SOLVE THE CLASSIFICATION PROBLEMS

Alberto de Oliveira Moreno

February/2008

Advisors: Adilson Elias Xavier
Alexandre Pinto Alves da Silva

Department: Engineering of Systems and Computation

This work considers a new solution methodology for the classification problem associated to the cluster analysis. The classical mathematical formulation for this problem is based on non differentiable optimization problems that are solved by methods that have few precision and efficiency. In this work, by using the hyperbolic smoothing technique, the problem is formulated as completely differentiable problem, whose solution by classic and more powerful methods can be obtained with great robustness and efficiency. We hope the proposed methodology prove an improvement both reliability and the efficiency in comparison with other alternative procedures.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

Diariamente são produzidos e manipulados em todo o mundo um número explosivamente crescente de arquivos contendo dados sobre universos das mais diversas naturezas. No entanto, apenas a disponibilidade desses dados não é suficiente para dar-se um uso eficiente e eficaz aos mesmos. É necessário tratá-los, categorizá-los e classificá-los para promover o seu uso de forma eficiente, para afinal explorar ao máximo suas potencialidades. Assim, sobretudo nos últimos 30 anos, foram publicados inumeráveis artigos dedicados ao estudo do tratamento de informações visando o aprimoramento da sua utilização.

Dentre as técnicas estudadas para tais fins, serão consideradas duas de grande relevância: agrupamento e classificação.

Agrupamento Agrupamento ou análise de agrupamento envolve a identificação de subconjuntos de dados que são similares. Os subconjuntos usualmente correspondem intuitivamente a pontos que são mais similares entre si do que a pontos de outro subconjunto. O agrupamento é conduzido de forma dita não supervisionada, tentando encontrar subconjuntos de pontos que são similares, sem que se tenha uma noção prévia de qualquer tipo de vinculação entre os mesmos. Elementos pertencentes a uma determinado grupo devem ser o máximo possível homogêneos, mantendo grande similaridade entre si, enquanto que elementos pertencentes a grupos distintos devem guardar entre si maiores diferenças.

Classificação A classificação envolve a atribuição dita supervisionada de elementos de dados com vinculações conhecidas e pré-definidas. Aqui, há um conjunto de observações que possuem classes já rotuladas e o problema consiste em rotular uma nova observação sem rótulo a uma classe utilizando a estrutura intrínseca vigente ao conjunto original. Usualmente, as classes conhecidas de exemplos constituem um conjunto de treinamento e são usadas para caracterizar uma descrição destas classes. Outra possibilidade existente no problema de classificação é que uma nova informação a ser classificada não seja alocada em nenhuma das classes existentes,

mas que seja associada a uma nova classe a ser especificamente criada.

Diversos artigos foram dedicados ao estudo de tais problemas anteriormente apresentados. Dentre estes, destacam-se pelos bons resultados os artigos [Bagirov et al., 2002] e [Bagirov and Yearwood, 2003] que tratam, utilizando sofisticados métodos de programação matemática, tanto do problema de determinação de agrupamentos quanto do problema de classificação. No entanto, o assunto não se esgota em Bagirov, já que é possível aprimorar ainda mais tanto a precisão quanto o tempo de processamento do seu método.

As técnicas de agrupamento e classificação podem ser aplicadas em uma grande variedade de problemas advindos dos mais diferentes contextos. Por exemplo, suponha que um conjunto de doentes sofra de uma certa moléstia e que nesses doentes essa moléstia se manifesta com diferentes graus de intensidade. Em primeiro lugar, as técnicas de agrupamento poderiam ser utilizadas para agrupar os doentes de modo a categorizar os diferentes estágios da doença. Com base nessa categorização e através das técnicas de classificação, pode-se classificar um novo doente, ou seja, pode-se fazer um diagnóstico do estágio da doença em que esse doente se encontra. Essas medidas contribuiriam para definir a medicação mais adequada em cada estágio da doença ou até mesmo para subsidiar a criação de medicamentos mais eficientes para o tratamento.

Outro exemplo, suponha que uma empresa de cartões de créditos deseja potencializar seus lucros. Para isso, deve conhecer bem os seus clientes a fim de evitar perdas devido à inadimplência. Uma forma de alcançar este objetivo é classificar os atuais clientes em categorias que caracterizem adequadamente seu comportamento e, ao mesmo tempo, definir procedimentos de classificação de novos clientes, visando estabelecer os riscos dos créditos a eles concedidos.

É conveniente ressaltar que nos trabalhos de Bagirov, os problemas de agrupamento e classificação, são tratados de forma independente, embora haja o reconhecimento das fortes relações que eles guardam entre si. Existem autores que estudam os dois problemas de forma integrada, explorando intensamente as suas relações.

O objetivo principal deste trabalho é propor um novo e eficiente método para o problema específico de classificação. Esse novo método é fortemente motivado pelos trabalhos de Bagirov.

O conteúdo aqui exposto é divido em seis capítulos. No capítulo 2, são apresentados diferentes métodos de abordagem para os problemas associados a agrupamentos. No capítulo 3, são formalmente definidos os problemas de agrupamento e de classificação.

Além disso, ainda no capítulo 3, são apresentados os algoritmos propostos por Bagirov para a solução desses problemas. No capítulo 4, propõem-se uma nova metodologia para resolução do problema de classificação que é baseada nas técnicas de suavização e penalização hiperbólicas. Foram realizados experimentos numéricos com alguns problemas artificiais e com problemas clássicos apresentados na literatura. No capítulo 5, são apresentados os resultados obtidos nesses experimentos com a implementação em FORTRAN da metodologia proposta. O capítulo 6 destina-se às conclusões.

2 Problemas de agrupamento e classificação

A análise de dados dá apoio a diversas aplicações computacionais, seja numa fase de projeto ou numa fase operacional. Procedimentos de análise de dados podem ser classificados como exploratórios ou confirmatórios, baseados na disponibilidade de modelos apropriados para as bases de dados. Um elemento chave em ambos os procedimentos (seja para comprovação de hipóteses ou para uma tomada de decisão) é o agrupamento, ou a classificação de medidas baseadas em (i) bom ajustamento a um modelo postulado, ou (ii) grupos naturais revelados através da análise.

Análise de agrupamento é a organização de uma coleção de padrões (usualmente representados como vetores de medidas, ou pontos num espaço multidimensional) em grupos baseados na similaridade. Conceitualmente, um padrão que pertença a um determinado grupo é mais similar a padrões que pertençam a esse mesmo grupo que a padrões que pertençam a outros grupos. Um exemplo de agrupamento com similaridade baseada na distância é mostrado na Figura 1. Os padrões são mostrados na Figura 1(a) e os grupos identificados pelos rótulos (numeração) são mostrados na Figura 1(b).

A variedade de técnicas para a representação de dados, de medidas de proximidade (similaridade) entre elementos de dados e de critérios de agrupamento tem produzido um rico e muitas vezes até confuso conjunto de métodos de agrupamento.

É importante entender as diferenças entre procedimentos de agrupamento (classificação não supervisionada) e procedimentos de discriminação (classificação supervisionada). Na classificação supervisionada, tem-se um conjunto de rótulos de padrões (pré-classificados), o problema é rotular uma nova informação, recentemente obtida, num padrão. Tipicamente, em uma etapa denominada treinamento, os dados padrões são usados para aprender a descrição das classes, que por outro lado são usadas para rotular um novo padrão. No caso do agrupamento, o problema é agrupar uma dada coleção de padrões, ainda não rotulados, em grupos significativos. Em certo sentido, rótulos são

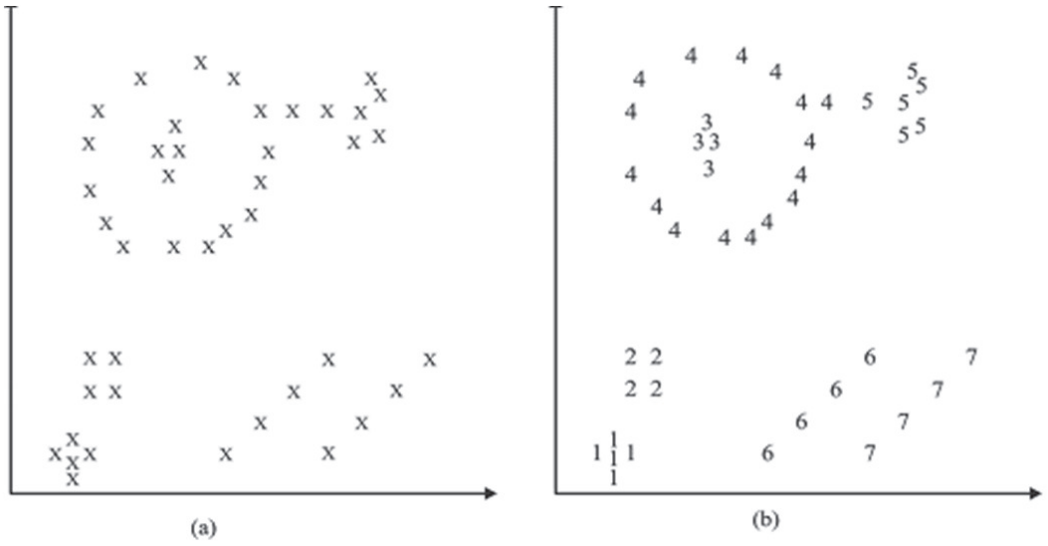

Figura 1: Agrupamento de dados.

associados a grupos, mas essa categorização de rótulos e de dados derivados são obtidos exclusivamente a partir dos dados disponíveis.

Agrupamentos são usados em diversas análises exploratórias de padrões, tomada de decisões e situações de aprendizagem por máquinas, incluindo mineração de dados, recuperação de documentos, segmentação de imagens e classificação de padrões. Entretanto, em muitos problemas desse tipo, há menos informação a *priori* (por exemplo, modelos estatísticos) disponíveis sobre os dados, e o decisor deve fazer algumas hipóteses sobre os dados. É com essas condições e restrições que a metodologia de agrupamento é particularmente apropriada para a exploração de relações entre os elementos de dados para fazer afirmações (ainda que preliminares) sobre a sua estrutura.

O termo agrupamento é usado por vários autores para descrever métodos de agrupamento de dados não rotulados. Esses autores possuem diferentes terminologias e hipóteses para a especificação do processo de agrupamento e para o contexto no qual os agrupamentos são usados. Assim, o escopo deste tópico se configura como um verdadeiro dilema. A produção de um resumo totalmente compreensível sobre métodos de agrupamento é uma tarefa muito difícil devido ao grande volume da literatura nesta área. A acessibilidade do resumo pode ser questionável dada a necessidade de reconciliar diferentes vocabulários e hipóteses relativas aos agrupamentos apresentados por diferentes grupos de autores ([Hansen and Jaummard, 1997], [Jain and Dubes, 1988],[Spath, 1980], etc).

2.1 Agrupamento

De acordo com Jain, a tarefa de gerar agrupamentos, em geral, envolve os seguintes passos [Jain et al., 1999]:

1. representação dos dados;
2. escolha de uma medida de similaridade, ou distância métrica, que é mais apropriada para a tarefa de agrupamento;
3. realização do agrupamento;
4. descrição dos grupos resultantes;
5. avaliação.

A representação dos dados consiste em definir o número de atributos disponíveis para utilização, sua natureza e escala, o tamanho do conjunto de dados e o número de grupos ou classes. Algumas dessas informações podem não ser controláveis na prática pelo decisor. Seleção de atributos, ou componentes do vetor de dados, é o processo de selecionar um subconjunto de atributos, que seja o mais efetivo para ser usado nos procedimentos de agrupamento. A seleção de atributos é usada para identificar os atributos mais informativos, para remover alguns atributos pouco informativos ou com ruído e para reduzir a dimensão do problema em análise. Extração de atributos ou combinação de atributos é usada para transformar o conjunto de atributos disponíveis, produzindo atributos mais expressivos, para serem usados eficientemente pelo algoritmo de classificação.

O conceito de proximidade é usualmente medido por uma função de distância definida nos pares de padrões. Uma variedade de medidas de distância é usada por vários autores ([Anderberg, 1973], [Jain and Dubes, 1988], [Diday and Simon, 1976]). Uma medida de distância simples, como a distância euclidiana, pode muitas vezes refletir a dissimilaridade entre dois padrões, enquanto outras medidas podem ser usadas para caracterizar a similaridade entre padrões [Michalski et al., 1983]. De acordo com Jain e Dubes, é muito importante selecionar uma métrica adequada para medir a similaridade ou a dissimilaridade entre indivíduos, assim como é necessário uma boa escolha para a representação justa das classes [Jain and Dubes, 1988].

O processo de realização de agrupamento pode ser realizado de várias formas. O resultado do agrupamento (ou agrupamentos) pode ser uma partição perfeita dos dados

em grupos completamente desconexos ou, alternativamente, em um particionamento *fuzzy*, onde cada padrão tem um grau variável de ser membro de cada grupo.

A abstração de dados é o processo de extrair uma representação simples e compacta do conjunto de dados. Aqui, busca-se a simplicidade sob a perspectiva de uma análise automática (de modo que uma máquina possa fazer processamentos posteriores eficientemente) ou sob a perspectiva de viabilizar a análise humana (de modo que a representação obtida é fácil de compreender, tendo um apelo intuitivo). No contexto do agrupamento, uma abstração de dados típica é uma descrição compacta de cada grupo, usualmente em termos dos protótipos de grupos ou de padrões de representatividade, como, por exemplo, os centróides [Diday and Simon, 1976].

Como o resultado do algoritmo de agrupamento é avaliado? O que caracteriza um bom agrupamento e um agrupamento pobre? Todos os algoritmos de agrupamento, quando trabalham com os dados, irão produzir grupos, independente dos dados já os conterem ou não. Se os dados não contêm grupos naturais, alguns algoritmos de agrupamento podem obter melhores resultados que os outros. A avaliação do resultado de um algoritmo de agrupamento tem várias facetas. Uma delas é uma avaliação do domínio dos dados ao invés do algoritmo em si mesmo. O estudo de tendências intrínsecas na constituição dos grupos para examinar se os dados possuem algum tipo de estrutura a *priori* é uma área onde a literatura ainda é insipiente. Os interessados podem consultar [Dubes, 1987] e [Cheng, 1995] para maiores informações.

A análise de validade dos grupos produzidos, ao contrário, é a avaliação dos resultados de um algoritmo de agrupamento. Muitas vezes, essa análise usa um critério específico de otimalidade, entretanto, estes critérios são usualmente subjetivos. Daí, pouco existe a respeito de bons padrões de formação de grupos, com exceção em domínios pré-prescritos. A validade dessas avaliações deve ser objetiva, tendo a perspectiva de verificar se o resultado é significativo sob o ponto de vista prático [Dubes, 1993]. Uma estrutura de grupos é válida se ela não pode razoavelmente ter ocorrido por acaso, ou como um artifício do algoritmo de agrupamento.

Quando métodos estatísticos são aplicados a problemas de agrupamento, a validação é acompanhada pela aplicação cuidadosa de métodos e testes de hipóteses. Há três tipos de estudo de validação. Uma avaliação externa compara a estrutura recoberta com uma estrutura dada a *priori*. Um exame interno de validade tenta determinar se a estrutura é intrinsecamente apropriada para os dados. Um teste relativo compara duas estruturas e mede os seus méritos relativos. Um conjunto de índices alternativos usados para essa

comparação é discutido em detalhes em [Jain and Dubes, 1988] e [Dubes, 1993].

A noção de agrupamento é relativamente flexível, assim como o objetivo para identificar e revelar grupos numa análise exploratória de dados. Um conceito importante de um grupo representativo é o chamado perfil de grupo, ou vetor de classificação, ou rótulo de grupo ou centróide. Esse deve ser próximo ou similar a cada objeto do grupo em um sentido médio. A similaridade dos objetos para outros objetos é medida por uma função de similaridade.

A noção de similaridade é crucial na definição de um agrupamento. Similaridade é usualmente medida por alguma medida contrária de dissimilaridade como, por exemplo, distâncias ou métricas definidas no conjunto de dados. Entretanto, há outros meios de explicitar a noção de similaridade. Por exemplo, através de relações de equivalência e, nesse caso, o conceito de similaridade é mais forte que o usual definido por distâncias.

2.2 O lema do usuário e o papel do especialista

A disponibilidade de uma vasta coleção de algoritmos de agrupamento na literatura pode facilmente confundir um usuário que tenta selecionar um algoritmo adequado para o problema a ser resolvido. Em [Dubes and Jain, 1976], um conjunto de critérios admissíveis definidos por Fisher em [Fisher and Van Ness, 1971] é usado para comparar algoritmos de agrupamento. Estes critérios são baseados em (1) a maneira pela qual o agrupamento é formado, (2) a estrutura dos dados e (3) a sensibilidade da técnica de agrupamento quanto a mudanças que não afetam a estrutura dos dados. Entretanto, não há nenhuma análise crítica de algoritmos de agrupamento que tratem questões importantes tais como

1. Como os dados devem ser normalizados;
2. Que medida de similaridade é apropriada para ser usada em cada situação;
3. Como o domínio do conhecimento deve ser usado em cada situação;
4. Como pode um grande conjunto de dados (por exemplo, um milhão de padrões) ser agrupado eficientemente.

Esses resultados motivaram a idéia de formalizar uma perspectiva sobre o estado da arte no que concerne às metodologias de agrupamento e aos algoritmos de agrupamento.

Com essa perspectiva, um especialista bem informado deve ser capaz de avaliar com confiabilidade os resultados de diferentes técnicas e, finalmente, tomar uma decisão competente sobre a técnica a ser empregada em cada aplicação.

Não há uma técnica que seja universalmente aplicável às diferentes estruturas da variedade de conjuntos de dados. Por exemplo, considere a Figura 1(a). Nem todas as técnicas de agrupamento podem tratar os grupos ali contidos com a mesma eficiência, porque os algoritmos muitas vezes contêm hipóteses implícitas, a respeito da forma dos grupos, baseadas nas medidas de similaridade e nos critérios de agrupamento utilizados.

O desempenho humano pode ser capaz de competir com procedimentos automáticos de agrupamento em problemas com duas dimensões, mas muitos problemas reais envolvem várias dimensões. É extremamente difícil para os humanos obter interpretações intuitivas de dados imersos num espaço de grande dimensão. Adicionalmente, os dados raramente seguem uma estrutura ideal (por exemplo, hiperesférica, linear) como mostrada na Figura 1. Isso explica o grande número de algoritmos de agrupamento que continuam a surgir na literatura, cada novo algoritmo trabalha melhor que os demais para uma dada distribuição de padrões.

É essencial para o uso de um algoritmo de agrupamento, não apenas ter um completo entendimento da técnica particular que está sendo utilizada, mas também conhecer os detalhes do processo de obtenção dos dados. Quanto mais informação o especialista tiver sobre os dados disponíveis, mais provável será a realização de uma análise bem sucedida do problema [Jain and Dubes, 1988]. Esse domínio da informação pode também ser usado para melhorar a qualidade da extração de atributos, da especificação do cálculo da similaridade, e da representação do grupo [Murty and Jain, 1995].

Restrições apropriadas na fonte de dados podem ser incorporadas num procedimento de agrupamento. Um exemplo disto é a resolução do problema da mistura [Titterington et al., 1985], onde é assumido que os dados são retirados de uma mistura de um número de funções densidades com parâmetros desconhecidos (muitas vezes essas funções são supostas serem Gaussianas). O problema de agrupamento aqui é identificar o número de componentes da mistura, bem como os parâmetros de cada componente. O conceito de densidade de um agrupamento e uma metodologia de decomposição do espaço de atributos [Bajcsy, 1997] têm sido incorporados em metodologias tradicionais de agrupamento, conduzindo a uma técnica de extração de grupos sobrepostos.

2.3 Histórico

Mesmo pensando que há um interesse crescente no uso de métodos de agrupamento em problemas de reconhecimento de padrões [Anderberg, 1973], proteção de informações ([Salton, 1991], [Rasmussen, 1992]) e processamento de imagens [Jain and Flynn, 1996], o agrupamento tem uma história rica em outras disciplinas tais como: taxonomia numérica [Sneath and Sokal, 1973]; biologia, geologia, geografia e marketing [Jain and Dubes, 1988]; quantização de vetores [Oehler and Gray, 1995]. Outras áreas mais ou menos associadas com agrupamentos incluem: aprendizagem por observação [Michalski et al., 1983] e aprendizagem não-supervisionada [Jain and Dubes, 1988]. O campo da análise espacial de padrões é também relacionado à análise de agrupamento [Ripley, 1989]. A importância e a natureza interdisciplinares do tema agrupamento são evidentes através da vasta literatura sobre o assunto.

Um grande número de livros sobre agrupamentos têm sido publicado ([Anderberg, 1973], [Duran and Odell, 1974], [Hartigan, 1975], [Spath, 1980], [Jain and Dubes, 1988], [Everitt, 1993], [Backer, 1995]), em adição a inúmeros artigos de revista. Um resumo do estado da arte em agrupamentos pode ser encontrado em [Dubes and Jain, 1980]. Uma comparação de vários algoritmos de agrupamento para construir a árvore geradora mínima e o caminho gerador mais curto foram dados em [Lee, 1981]. Análise de agrupamento foi também resumida em [Jain et al., 1986]. Uma revisão de segmentação de imagens por agrupamento foi reportada em [Jain and Flynn, 1996]. Comparações de diversos métodos combinatoriais de otimização aplicados à problemas de agrupamento, baseados em experimentos, foram também reportados em [Mishra and Raghavan, 1994] e [Al-Sultan and Khan, 1996].

2.4 Definições e notação

Será usada a seguinte notação neste texto:

- Padrão (ou vetor de atributos, observações ou dados) é um item de dado simples usado num algoritmo de agrupamento, que, tipicamente, consiste de um vetor com n medidas ou componentes;
- n é a dimensão do padrão ou o espaço do padrão;
- Um conjunto padrão é denotado por $H = \{x_1, \ldots, x_m\}$. O i-ésimo padrão em

H é denotado por $x_i = \{x_{i1}, \ldots, x_{in}\}$. Em muitos casos o conjunto padrão a ser agrupado é visto como uma matriz $m \times n$;

- As componentes escalares individuais x_{ik}, $k = 1, \ldots, n$ de um padrão x_i são chamadas atributos;

- Uma classe, no abstrato, refere-se a um estado da natureza que governa certo processo de geração de padrões. Mais concretamente, uma classe pode ser vista como uma fonte de padrões cuja distribuição no espaço de atributos é governada por uma específica densidade de probabilidades para a classe. As técnicas de agrupamento tentam agrupar padrões de modo que as classes obtidas reflitam as diferenças e semelhanças entre os padrões envolvidos;

- Técnicas de agrupamento associam um rótulo l_i a cada padrão x_i identificando a sua classe. O conjunto de todos os rótulos para um conjunto padrão H é $L = \{l_1, \ldots, l_m\}$ com $l_i \in \{1, \ldots, k\}$, onde k é o número de grupos;

- Procedimentos para agrupamentos *fuzzy* atribuem a cada padrão x_i um grau fracionário de pertinência f_{ij} a cada grupo j, obedecendo normalmente a relação:

$$\sum_{j=1}^{k} f_{ij} = 1, \quad i = 1, \ldots, m.$$

- Uma medida de distância (uma especialização da noção de proximidade) é uma métrica (ou quase métrica) no espaço de atributos usada para quantificar a similaridade entre padrões.

Um mapeamento $d : U \times U \to \mathbb{R}$ (que atribui um número real a cada par de elementos de U) é chamado de função distância se, para arbitrários $x, y \in U$, tem-se:

$$d(x, y) \geq d_0, \tag{2.1}$$

$$d(x, x) = d_0, \tag{2.2}$$

$$d(x, y) = d(y, x), \tag{2.3}$$

onde d_0 é um número arbitrário real finito. Das relações (2.1) e (2.2) conclui-se que d é mínima quando os pares de valores são iguais. A equação (2.3) expressa a propriedade de simetria.

Uma função distância definida como acima será uma métrica se, em adição, as seguintes condições forem verdadeiras:

$$d(x,y) = d_0 \Longrightarrow x = y \tag{2.4}$$

$$d(x,z) \leq d(x,y) + d(y,z), \quad \forall x,y,z \in U. \tag{2.5}$$

Além disso, se d_0 for zero, obtém-se o conceito de métrica da análise funcional.

2.5 Representação de padrões, seleção e extração de atributos

Não há orientações teóricas que determinem quais padrões e/ou atributos são apropriados em situações específicas. De fato, o processo de geração de padrões muitas vezes não é diretamente controlável. O papel do usuário no processo de representação de padrões é obter fatos e conjecturas sobre os dados, opcionalmente realizar seleção de atributos e extração, e projetar os próximos elementos do sistema de agrupamento. Devido às dificuldades que cercam a representação de padrões, é convenientemente assumido que a representação de padrões está disponível antes de realizar o agrupamento.

Uma cuidadosa investigação dos atributos disponíveis e a realização de transformações adequadas sobre os mesmos (mesmo que sejam simples) podem gerar melhorias significativas nos resultados do agrupamento. Uma boa representação de padrões pode muitas vezes resultar num simples e compreensível agrupamento. Uma representação pobre pode resultar em grupos complexos cuja verdadeira estrutura pode ser difícil ou impossível explicitar. A Figura 2 mostra um exemplo simples e representativo dessa problemática. Os pontos neste espaço de 2 dimensões são dispostos num grupo curvilíneo com distância aproximadamente constante até a origem.

Se forem escolhidas coordenadas cartesianas para representar os padrões, muitos algoritmos de agrupamento irão fragmentar o grupo em dois ou mais grupos, visto que o conjunto de padrões não apresenta uma estrutura muito compacta. Se, entretanto, for usado um sistema de coordenadas polares para a representação dos padrões, a distribuição da coordenada raio exibirá uma estrutura compacta e uma solução com um único grupo será provavelmente obtida.

Um padrão pode medir seja um objeto físico simples (por exemplo, uma cadeira) ou uma noção abstrata (por exemplo, o estilo de escrever), tal como notado acima. Padrões são representados convencionalmente como vetores multidimensionais, onde cada

Figura 2: Um grupo curvilíneo cujos pontos são aproximadamente equidistantes da origem. Diferentes representações de padrões (sistemas de coordenadas) podem levar algoritmos de agrupamento a gerarem diferentes resultados para estes dados.

dimensão é um atributo simples [Duda and Hart, 1973]. Esses atributos podem ser quantitativos ou qualitativos. Por exemplo, se peso e cor são dois atributos utilizados, então (20, preto) é uma representação de um objeto preto com 20 unidades de peso. Os atributos podem ser subdivididos nos seguintes tipos [Gowda and Diday, 1992]:

(1) atributos quantitativos , i.e.,

- **(1.a)** valores contínuos (por exemplo, peso),
- **(1.b)** valores discretos (por exemplo,o número de computadores),
- **(1.c)** valores de intervalos (i.e., a duração de um evento),

(2) atributos qualitativos,

- **(2.a)** nominais ou não ordinais (por exemplo, cor),
- **(2.b)** ordinais (por exemplo, ordenamento militar ou grau de instrução).

Atributos quantitativos podem ser medidos numa escala de razões (com valores de referência, tal como temperatura) ou escalas nominais ou ordinais.

Pode-se usar também atributos estruturados [Michalski et al., 1983], os quais são representados como árvores, onde os nós pais num nível superior representam uma generalização dos nós descendentes. Por exemplo, um nó pai 'veículo' pode ser uma generalização de nós descendentes 'carro', 'ônibus', 'caminhões', e 'motocicletas'. Além disso, os nós 'carros' podem ser uma generalização de carros do tipo 'Toyota', 'Ford', 'Mercedes-Benz', etc. Uma representação generalizada de padrões chamada de objetos simbólicos foi proposta em [Diday, 1988]. Objetos simbólicos são definidos por uma conjunção simbólica

de eventos. Estes eventos ligam valores e atributos nos quais os atributos podem assumir um ou mais valores e todos os objetos não necessitam ser definidos no mesmo conjunto de atributos.

Pode ser muitas vezes interessante isolar somente os atributos mais descritivos e discriminativos do conjunto de dados de entrada, e utilizar somente esses atributos na análise subsequente. Técnicas de seleção de atributos identificam um subconjunto de atributos existentes para uso subsequente, enquanto que técnicas de extração de atributos determinam novos atributos a partir do conjunto original. Em qualquer caso, o objetivo é melhorar a classificação original e ou a eficiência computacional. A seleção de atributos é um tópico bem explorado em estatística para reconhecimento de padrões [Duda and Hart, 1973]. Entretanto, no contexto de análise de agrupamento, isto é, na ausência de classes de rótulos padrões, o processo de seleção de padrões é de necessidade quase obrigatória e pode envolver um processo de "tentativa e erro", onde vários subconjuntos de atributos são selecionados, os padrões resultantes são agrupados e os resultados são avaliados usando um índice de validade. Em oposição, alguns dos processos populares de extração (por exemplo, análise de componentes principais [Fukunaga, 1990]) não dependem dos dados rotulados e podem ser usados diretamente. A redução do número de atributos possui um benefício adicional, que é a possibilidade de produzir resultados que podem ser entendidos com mais facilidade e até mesmo visualmente representados.

2.6 Medidas de similaridade

Visto que a similaridade é fundamental para a definição de um agrupamento, a medida da similaridade entre dois padrões do mesmo espaço de atributos é essencial na maioria dos procedimentos de agrupamento. Devido à variedade dos tipos de atributos e escalas, a medida de distância deve ser escolhida cuidadosamente. É muito comum calcular as dissimilaridades entre dois padrões usando uma medida de distância definida no espaço de atributos. Neste trabalho, a atenção será concentrada nas bem conhecidas medidas de distância usadas em padrões cujos atributos são contínuos.

A mais popular métrica para atributos contínuos é a distância euclidiana:

$$d_2(x_i, x_j) = \|x_i - x_j\|_2 = \left(\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2 \right)^{1/2}, \tag{2.6}$$

que é um caso especial ($p = 2$) da distância de Minkowski:

$$d_p(x_i, x_j) = \|x_i - x_j\|_p = \left(\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^p \right)^{1/p} . \tag{2.7}$$

A distância euclidiana tem um apelo intuitivo, pois é comumente usada na avaliação da proximidade de objetos em espaços de duas ou três dimensões. Essa distância trabalha bem quando o conjunto de dados contém grupos compactos ou isolados [Mao and Jain, 1996]. A consequência direta do uso de qualquer métrica de Minkowski é a tendência dos atributos de maior porte dominarem os demais atributos. Soluções para essa dificuldade incluem a normalização dos atributos contínuos (para um intervalo comum de variância) ou a adoção de outros esquemas de pesos.

A presença de correlação linear entre atributos pode distorcer as medidas de distância. Essa distorção pode ser aliviada aplicando uma transformação simples nos dados ou usando a distância quadrática de Mahalanobis.

$$d_M(x_i, x_j) = \Sigma^{-1}(x_{i,k} - x_{j,k})^T \tag{2.8}$$

onde os padrões, x_i são assumidos serem vetores linha e Σ é a matriz de covariâncias entre os padrões. A distância $d_M(\cdot, \cdot)$ atribui diferentes pesos aos diferentes atributos baseando-se tanto na variância dos atributos quanto nas correlações existentes entre eles. É implicitamente assumido que as densidades condicionais das classes são unimodais e caracterizadas por uma dispersão multidimensional, e, ademais, que as densidades são Gaussianas multidimensionais. A distância regularizada de Mahalanobis foi usada por [Mao and Jain, 1996] para extrair grupos hiperelipsoidais.

Recentemente, alguns pesquisadores usaram a distância de Hausdorff no contexto de um conjunto de pontos ([Huttenlocher et al., 1993], [Dubuisson and Jain, 1994]). Sendo A e B dois conjuntos, essa distância é definida como:

$$h(A, B) = \max_{a \in A} \left\{ \min_{b \in B} \|a - b\| \right\} .$$

Alguns algoritmos trabalham numa matriz de valores de proximidades ao invés do conjunto original de padrões. Isto é útil em situações onde se pré-calculam os $n(n-1)/2$ pares de distâncias para os n padrões e elas são guardadas em uma matriz (simétrica).

O cálculo de distâncias entre padrões com alguns ou todos os atributos não sendo contínuos é problemático, visto que diferentes tipos de atributos não são comparáveis.

Nessa situação, a noção de proximidade é efetivamente valorada por binários para atributos com escala nominal. A despeito deste fato, alguns estudiosos, especialmente aqueles associados a aprendizagem por máquinas onde padrões mistos são comuns, desenvolveram medidas de proximidade para tipos de padrões heterogêneos. Um exemplo recente é [Wilson and Martinez, 1997], que propõe uma combinação da métrica modificada de Minkowski para atributos contínuos e uma distância baseada em contagens para atributos nominais. Uma variedade de outras métricas foi reportada em [Diday and Simon, 1976] e [Ichino and Yaguchi, 1994] para computar as similaridades entre padrões representados por atributos quantitativos e qualitativos.

Padrões podem ser representados usando cadeias ou estruturas de árvores [Knuth, 1973]. Cadeias são usadas em agrupamentos sintáticos [Fu and Lu, 1977]. Várias medidas de similaridade entre cadeias são descritas em [Baeza-Yates, 1992]. Um bom resumo de medidas de similaridades entre árvores é dada em [Zhang, 1995]. Uma comparação de métodos estatísticos e sintáticos para o reconhecimento de padrões foi apresentado em [Tanaka, 1995], onde foi concluído que os métodos sintáticos são inferiores em todos os aspectos. Portanto, os métodos sintáticos não mais serão considerados neste texto.

Uma função de similaridade é um mapeamento $s : U \times U \rightarrow \mathbb{R}$ com as seguintes propriedades:

$$s(x, y) \leq s_0, \tag{2.9}$$

$$s(x, x) = s_0, \tag{2.10}$$

$$s(x, y) = s(y, x), \tag{2.11}$$

onde s_0 é um número real . A distinção entre as medidas de distância d e similaridade s está nas expressões (2.1) e (2.9).

Uma função de similaridade é chamada uma métrica se as seguintes condições são verdadeiras

$$\text{Se} \quad s(x, y) = s_0, \quad \text{então} \quad x = y, \tag{2.12}$$

$$s(x, z) \leq s(x, y) + s(y, z), \quad \forall x, y, z \in U. \tag{2.13}$$

As relações (2.9) e (2.10) correspondem à proposição de que a máxima similaridade pode apenas ser verdadeira se os dois pontos são iguais. A relação (2.11) é análoga à relação (2.3) e daí não há maior diferença entre função de distância e função de similaridade.

É possível provar que (ver [Bock, 1970], [Deichsel, 1972], [Fritsche, 1973] e [Soergel, 1967]) que se d é uma função (métrica) distância tal que o seus valores estão em $\mathbb{R}^+$ ou $\mathbb{R}^-$, então $1/d$ é uma função (métrica) de similaridade. Além disso, se d assume apenas valores finitos, então

$$\begin{aligned} s(x,y) &= [\max_{(w,z)\in U^2} d(w,z)] - d(x,y), \\ s(x,y) &= \sqrt{[\max_{(w,z)\in U^2} d(w,z)] - d(x,y)}, \\ s(x,y) &= \left[\max_{(w,z)\in U^2} d(w,z)\right] - d^2(x,y), \\ s(x,y) &= \exp(-d(x,y)) \end{aligned}$$

são funções (métricas) de similaridade.

Há certas medidas de distância reportadas na literatura [Gowda and Krishna, 1977], [Jarvis and Patrick, 1973] que levam em conta o efeito de pontos da vizinhança ou próximos dela. Estes pontos são chamados em [Michalski et al., 1983] de contextos.

A similaridade entre dois pontos x_i e x_j, dado o contexto, é dada por

$$s(x_i - x_j) = f(x_i, x_j, E) \tag{2.14}$$

onde E é o contexto (o conjunto de pontos da vizinhança). Uma métrica definida usando o contexto é a distância de vizinhança recíproca (MND) proposta em [Gowda and Krishna, 1977] que é dada por

$$MND(x_i, x_j) = NN(x_i, x_j) + NN(x_j, x_i) \tag{2.15}$$

onde $NN(x_i, x_j)$ é o número da vizinhança de x_i com relação a x_j.

As Figuras 3 e 4 dão um exemplo. Na Figura 3, a vizinhança mais próxima de A é B e a vizinhança mais próxima de B é A. Assim, tem-se os números de vizinhos $NN(A,B) = NN(B,A) = 1$ e a vizinhança recíproca $MND(A,B) = 2$. Entretanto, $NN(B,C) = 1$, $NN(C,B) = 2$ e, então, $MND(B,C) = 3$.

A Figura 4 foi obtida da Figura 3 adicionando três novos pontos D, E e F. Agora $MND(B,C) = 3$ (como anteriormente), mas $MND(A,B) = 5$. A MND entre A e B cresceu introduzindo pontos adicionais, mesmo considerando que A e B não se moveram a partir das suas posições adicionais. A MND não é uma métrica (ela não satisfaz a desigualdade triangular) [Zhang and Michalski, 1995]. Apesar desse fato, MND tem sido aplicada com sucesso em diversas aplicações de agrupamentos [Gowda and Diday, 1992].

Essa observação conduz à conclusão que a dissimilaridade não necessita ser uma métrica.

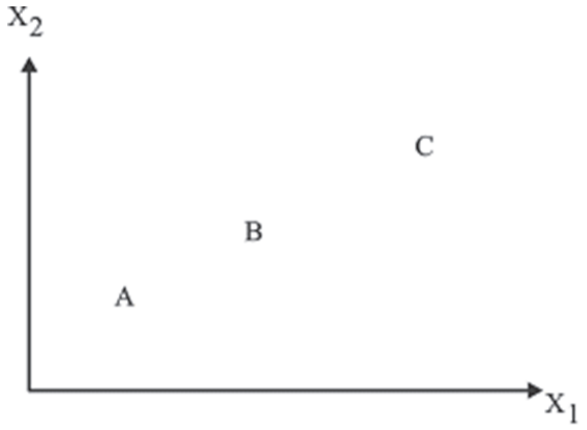

Figura 3: A e B são mais similares que A e C.

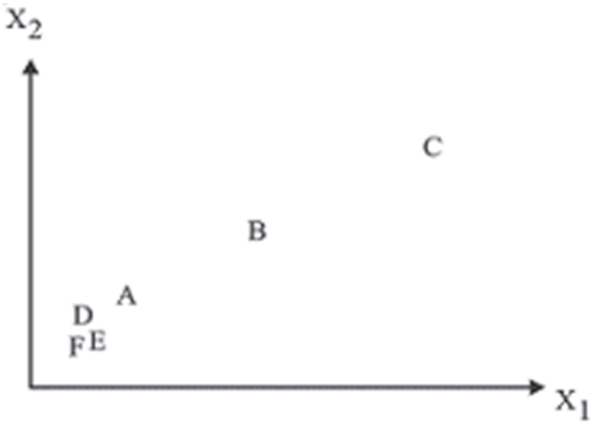

Figura 4: Após uma mudança no contexto, B torna-se mais similar a C.

Isto implica que é possível fazer dois padrões arbitrários igualmente similares considerando um número suficientemente grande de atributos. Como uma consequência, dois padrões quaisquer arbitrários são igualmente similares, a menos que se use alguma informação adicional. Por exemplo, no caso do agrupamento conceitual [Michalski et al., 1983] a similaridade entre x_i é x_j e definida como

$$s(x_i - x_j) = f(x_i, x_j, C) \tag{2.16}$$

onde C é um conjunto pré-definido de conceitos. Na Figura 5, a noção é ilustrada considerando em destaque 3 pontos: A, B e C do conjunto de pontos. Aqui, a distância euclidiana entre os pontos A e B é menor que a distância entre B e C. Entretanto, B e C podem ser vistos como mais similares que A e B porque B e C pertencem ao mesmo conceito (elipse) e A pertence a um conceito diferente (retângulo). A similaridade conceitual é a medida mais geral de similaridade.

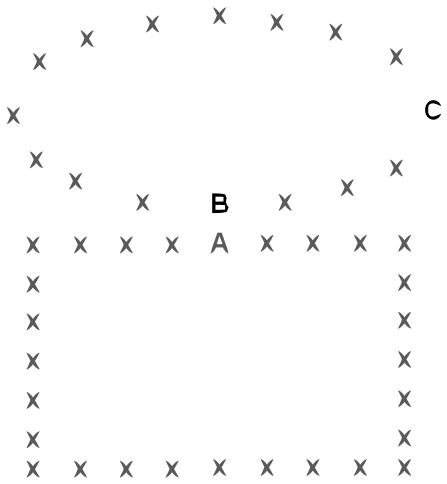

Figura 5: Similaridade conceitual entre pontos.

Muitos algoritmos de agrupamento usam parâmetros empiricamente determinados tais como:

- o número de grupos;
- o tamanho mínimo e máximo de cada grupo;
- um limite para a função de similaridade, abaixo do qual um objeto não será incluído no grupo;
- um controle sobre o espaço entre grupos;
- uma função objetivo a ser otimizada.

2.7 Técnicas de agrupamento

Diferentes métodos podem ser usados para descrever os dados em problemas de agrupamento. A Figura 6, mostra a taxonomia das metodologias de agrupamento, originalmente apresentada em [Jain and Dubes, 1988].

No nível mais elevado, há uma distinção entre métodos hierárquicos e métodos particionais. Essa distinção é de certa forma universal, pois amplamente adotada na literatura ([Hansen and Jaummard, 1997], [Jain et al., 1999]).

A taxonomia mostrada na Figura 6 deve ser suplementada por uma discussão dos princípios dos critérios de corte ou de separação, que, em principio, afetam todos os diferentes métodos, independentemente da sua posição na árvore de taxonomia.

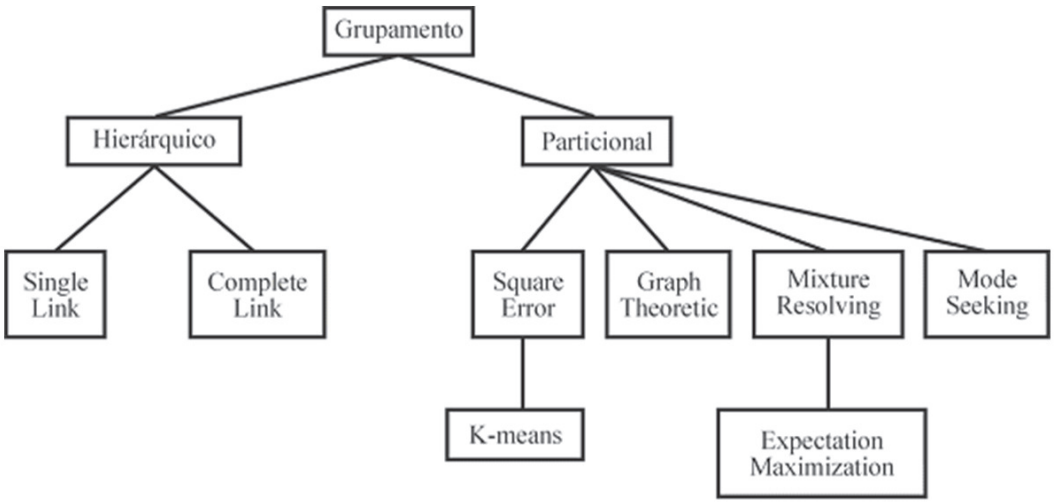

Figura 6: Taxonomia dos agrupamentos.

2.7.1 Agrupamentos hierárquicos

A operação de algoritmos hierárquicos para agrupamentos é ilustrada usando o conjunto de dados bidimensionais mostrados pela Figura 7. Essa figura representa sete padrões rotulados A, B, C, D, E, F, e G em três grupos.

Um algoritmo hierárquico conduz a um dendrograma, que é essencialmente uma estrutura em árvore cujos diferentes níveis podem ser usados para a especificação do agrupamento. Um dendrograma correspondente a sete pontos (obtidos pelo algoritmo de conexão simples) [Jain and Dubes, 1988] é mostrado na Figura 8. Esse dendrograma pode ser quebrado em diferentes níveis, para se obter diferentes agrupamentos dos dados.

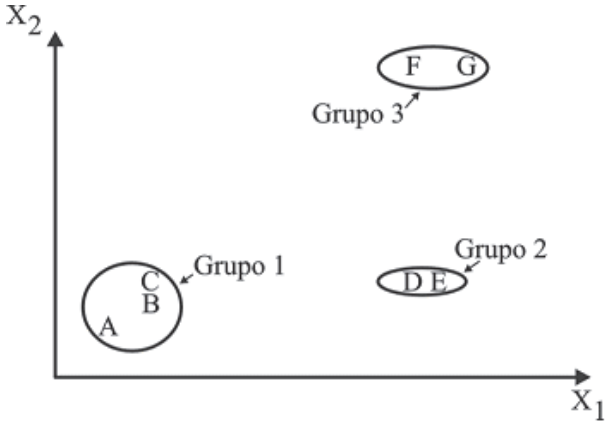

Figura 7: Pontos pertencentes a três grupos.

A maioria dos algoritmos de aglomeração hierárquica são variantes dos métodos de conexão simples [Sneath and Sokal, 1973], de conexão completa [King, 1967], e de variância mínima [Ward, 1963], [Murtagh, 1984]. Destes, os algoritmos de conexão simples e de

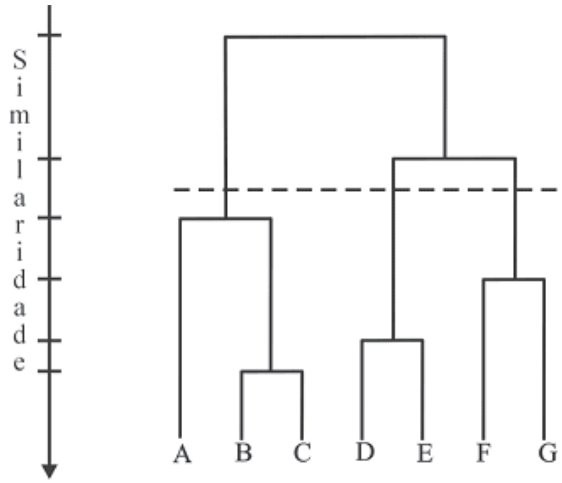

Figura 8: O dendrograma obtido usando o algoritmo de conexão simples.

conexão completa são os mais populares. Estes dois algoritmos diferem na forma que caracterizam a similaridade entre pares de grupos. No método de conexão simples, a distância entre dois grupos é a mínima das distâncias entre todos os pares de padrões retirados dos dois grupos (um padrão do primeiro grupo, o outro do segundo grupo). No algoritmo de conexão completa, a distância entre dois grupos é o máximo de todas as distâncias entre padrões dos dois gruposs. Em cada caso, dois grupos são misturados para formar um grupo maior com base no critério da mínima distância.

Os algoritmos de conexão completa produzem grupos mais compactos [Baeza-Yates, 1992]. Os algoritmos de conexão simples, ao contrário, sofrem de um efeito cadeia [Nagy, 1968] com a tendência de produzir grupos retos e alongados. Os algoritmos de variância mínima consideram a minimização da soma da variância intra de todos os grupos como critério de particionamento.

As Figuras 9 e 10 mostram dois grupos separados por uma ponte de padrões estranhos. O algoritmo de conexão simples produz os grupos mostrados na Figura 9, enquanto que o algoritmo de conexão completa obtém os grupos mostrados na Figura 10.

2.7.2 Algoritmos particionais

Um algoritmo particional para agrupamentos obtém uma partição simples dos dados ao invés de promover uma estrutura dos dados, tal como num dendrograma produzido por uma técnica hierárquica. Métodos particionais têm vantagens em aplicações envolvendo grandes conjuntos de dados para os quais o dendrograma é computacionalmente proibitivo. Um problema associado ao uso do algoritmo particional é a escolha do número de grupos.

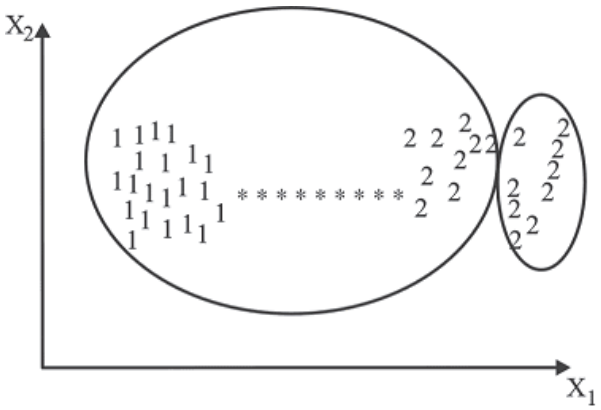

Figura 9: Agrupamento obtido pela aplicação de um algoritmo de conexão simples em um conjunto formado por duas classes (1 e 2) conectadas por uma cadeia de padrões estranhos.

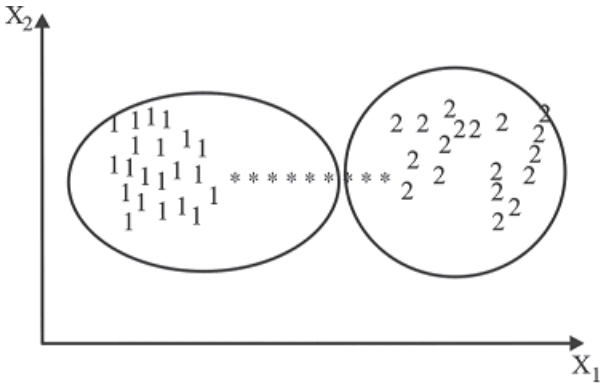

Figura 10: Agrupamento obtido pela aplicação de um algoritmo de conexão completa em um conjunto que formado por duas classes (1 e 2) conectadas por uma cadeia de padrões estranhos.

Um artigo fundamental [Dubes, 1987] dá informações sobre essa importante questão.

A técnica particional usualmente produz grupos pela otimização de uma função critério definida localmente (definida por um subconjunto de padrões) ou globalmente (definida por todos os padrões). Pesquisa combinatorial do conjunto de rótulos possíveis para um valor ótimo do critério é claramente proibitiva sob o ponto de vista computacional. Na prática, entretanto, o algoritmo é executado várias vezes, com diferentes pontos de partida, e a melhor configuração obtida de todas as execuções é usada como o melhor agrupamento.

2.7.3 Algoritmo do erro quadrático

A mais intuitiva e frequentemente utilizada função critério em técnicas de aglomeração particional é o critério do erro quadrático. O erro quadrático para um agrupamento L de

um padrão H (contendo k grupos) é dado por

$$e^2(H, L) = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left\| a_i^{(j)} - x^j \right\|^2 \tag{2.17}$$

onde $a_i^{(j)}$ é o i-ésimo padrão pertencente ao j-ésimo grupo, n_j é o número de padrões do grupo j e x^j é o centróide do j-ésimo grupo.

O k-média é o mais simples e mais comum algoritmo empregado utilizando o critério do erro quadrático [McQueen, 1967]. Esse algoritmo parte de uma partição inicial aleatória e mantém as reassociações de padrões aos grupos baseados na similaridade entre padrões e os centros dos grupos, até que um critério de convergência é alcançado (isto é, não há nenhuma reassociação de padrões de um grupo a outro, ou o erro quadrático torna-se abaixo de um nível significativo). O algoritmo k-média é popular porque é fácil de implementar, e a sua complexidade em tempo de cada iteração é $O(n)$, onde n o número de padrões. Uma dificuldade com esse algoritmo é sua sensibilidade à partição inicial, pois pode convergir para um mínimo local com alto valor da função critério, portanto inadequado, se a partição inicial calhar muito desfavorável.

A Figura 11 mostra sete padrões bidimensionais a ser agrupados em grupos. Se forem utilizados os padrões A, B e C como valores iniciais em torno dos quais os grupos são construídos, então produz-se a partição $((A), (B, C), (D, E, F, G)))$ representada por elipses.

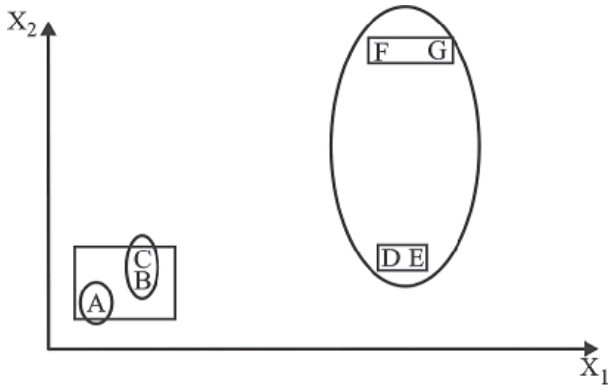

Figura 11: O algoritmo k-means é sensível à partição inicial.

O critério do erro quadrático é bem mais elevado para essa partição do que para a sua partição ótima $((A, B, C), (D, E), (F, G))$, representada por retângulos, que conduz ao valor mínimo da função erro quadrático para um agrupamento com três grupos. Uma solução correta para o problema com três grupos é obtida escolhendo, por exemplo, A, D e F como ponto de partida.

Várias variantes [Anderberg, 1973] do algoritmo k-média tem sido apresentadas na literatura. Algumas deles tentam selecionar uma boa partição inicial de modo que o algoritmo convirja com mais certeza, senão ao mínimo global, a um mínimo local profundo.

Uma possibilidade de variação do algoritmo k-média envolve a seleção de uma outra função critério. O algoritmo de agrupamento dinâmico (que permite representações de grupos que não os centróides) está proposto em [Diday, 1973]. [Symon, 1977] descreve um método de agrupamento dinâmico obtido pela formulação do problema de agrupamento no contexto da estimação por máxima verossimilhança. A distância regularizada de Mahalanobis foi usada em [Mao and Jain, 1996] para obter grupos hiperelipsoidais.

2.7.4 Agrupamentos via teoria dos grafos

O melhor algoritmo para agrupamentos baseado em teoria dos grafos é apoiado na construção de árvores geradoras mínimas (MST - *minimun spanning tree*) dos dados [Zahn, 1971]. A seguir, são retirados os arcos da MST de maior comprimento para se obter novos grupos. A Figura 12 apresenta a MST obtida a partir de nove pontos bidimensionais. Quebrando o arco CD com um comprimento de 6 unidades (o arco de maior comprimento), dois grupos (A, B, C) e (D, E, F, G, H, I) são obtidos. O segundo grupo pode ser dividido em dois grupos quebrando o arco EF, com comprimento de 4,5 unidades e assim por diante.

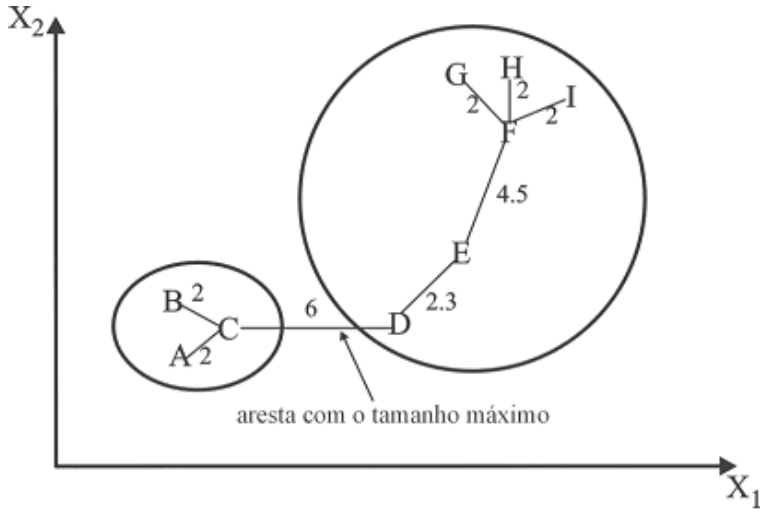

Figura 12: Usando a árvore geradora mínima para formar grupos.

2.7.5 Agrupamentos de vizinhança mais próxima

Visto que a proximidade tem um papel importante na noção intuitiva de agrupamento, distâncias mais próximas da vizinhança podem servir como uma base para o agrupamento. Um processo interativo foi proposto em [Lu and Fu, 1978], em que se atribui cada padrão não rotulado ao grupo do padrão da sua vizinhança mais próxima, desde que a distância a essa vizinhança seja menor que um dado limite. O processo prossegue até que todos os padrões são rotulados ou nenhum rótulo adicional ocorre. O valor da vizinhança (descrito anteriormente no contexto do cálculo da distância) pode ser usado para aumentar os grupos a partir de vizinhanças mais próximas.

2.7.6 Algoritmo de agrupamento *fuzzy*

Métodos de agrupamento tradicionais ou complexos geram partições ou grupos onde cada ponto pertence a um e somente um grupo. Agrupamento *fuzzy* estende este conceito para o domínio multi-rótulo, onde cada ponto pode estar simultaneamente em vários grupos. Esse propósito é alcançado usando-se uma função de pertinência, que dá a cada ponto o grau de sua vinculação a cada grupo.

Nos métodos de agrupamento tradicionais, o resultado é uma partição perfeita do conjunto inicial de padrões. Nos métodos de agrupamento *fuzzy* o resultado é uma matriz de pertinência P com dimensão (m, k), onde os elementos genéricos p_{ij} são os valores da vinculação do padrão i ao grupo j.

A Figura 13 mostra um conjunto de 9 pontos agrupados em dois grupos segundo o enfoque tradicional, referenciados por $H_1 = \{1, 2, 3, 4, 5\}$ e $H_2 = \{6, 7, 8, 9\}$. Os mesmos pontos são agrupados sob o enfoque *fuzzy* nos grupos F_1 e F_2. Observa-se que nesse enfoque os pontos 4, 5, 6 e 7 pertencem simultaneamente aos grupos F_1 e F_2.

2.7.7 Agrupamentos por redes neurais artificiais

Redes neurais artificiais (ANNs - *artificial neural networks*) [Hertz et al., 1991] são motivadas por redes biológicas neurais. ANN's têm sido usadas extensivamente ao longo das três últimas décadas tanto para o agrupamento quanto para a classificação [Sethi and Jain, 1991], [Jain and Mao, 1994].

Redes neurais competitivas (ou consideradas vencedoras) [Jain and Mao, 1996] são muitas vezes usadas para agrupar dados de entrada. Em aprendizagem competitiva,

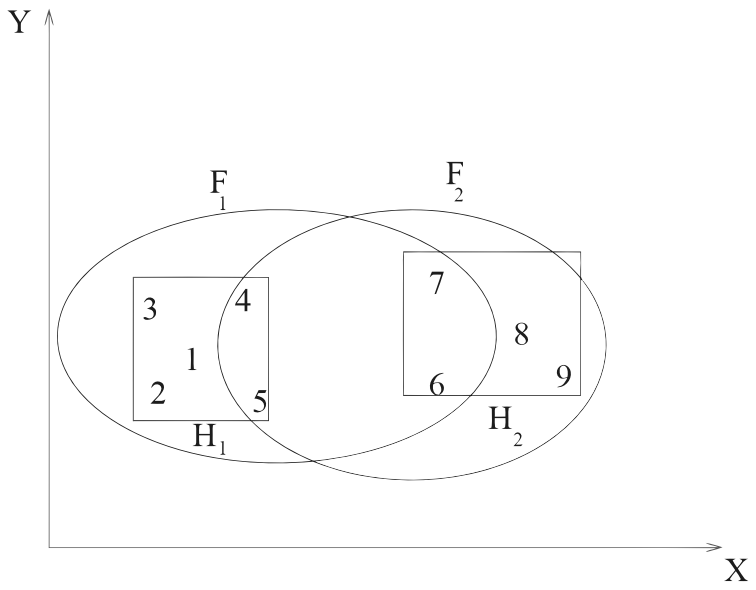

Figura 13: Agrupamento *fuzzy*.

padrões similares são agrupados pela rede e representados por uma unidade simples (neurônio). Esse agrupamento é feito automaticamente baseado em correlação de dados.

Exemplos bem conhecidos de redes neurais para agrupamentos incluem a quantificação do vetor de aprendizagem de (LVQ - *learning vector quantization*) e mapas de auto organização (SOM - *self-organized map*) [Kohonen, 1989], e modelos de teoria de ressonância adaptativa [Carpenter and Grossberg, 1990].

A arquitetura destas ANN's são simples. Elas são simplesmente estratificadas. Padrões são representados na entrada e são associadas com os nós de saída. Os pesos entre os nós de entrada e os nós de saída são iterativamente alterados (isto é conhecido como aprendizagem) até que um critério de terminação é satisfeito.

Aprendizagem competitiva foi inspirada em redes neurais biológicas. Entretanto, a aprendizagem ou os processos de atualização de pesos são bastante similares àqueles usados em alguns métodos clássicos de agrupamento. Por exemplo, a relação entre o algoritmo k-média e LVQ é explicitada em [Pal et al., 1993]. O algoritmo de aprendizagem em modelos ART é similar ao algoritmo de agrupamento líder [Moor, 1988].

2.7.8 Agrupamentos por Simulated Annealing

O método Simulated Annealing é uma busca aleatória emulada nos processos físicos associados ao recozimento de cristais. Algoritmos de Simulated Annealing são projeta-

dos tendo como objetivo a obtenção de um ponto de mínimo global de um problema de otimização. Para esse fim, adotam uma estratégia que consiste em aceitar com certa probabilidade uma nova solução de menor qualidade para a próxima iteração. A probabilidade de aceitação é governada por um parâmetro chamado de temperatura, que varia a partir de um valor de partida na primeira iteração até um valor final na última iteração.

Um algoritmo de Simulated Annealing seleciona aleatoriamente um ponto de partida, valores para as temperaturas inicial e final e calcula o erro quadrático. Pontos são realocados em grupos com base no valor do erro quadrático e com certa probabilidade dependente da temperatura da iteração. A cada iteração, o valor da temperatura é reduzido.

Simulated Annealing pode ser lento para alcançar a solução ótima. Soluções ótimas globais requerem temperaturas decrescentes bem vagarosamente no processo iterativo. [Selim and Al-Sultan, 1991] estudaram os efeitos dos parâmetros de controle. Existe um resultado teórico que garante estatisticamente a obtenção de um ótimo global pelo algoritmo de Simulated Annealing [Aarts and Korst, 1989].

2.8 Classificação de dados

Classificação é uma atribuição supervisionada de dados a classes pré-definidas. Aqui, há uma coleção de padrões com rótulos associados a um conjunto de classes. O problema consiste em rotular uma nova observação ou dado como pertencente a uma ou mais classes. O objetivo em muitos casos, onde os dados pertencem a um espaço real n-dimensional, é determinar uma função $f : \mathbb{R}^n \rightarrow y_i,\ i = 1, \ldots, l$ a partir dos dados de treinamento. Aqui, as l classes conhecidas são representadas pelos rótulos y_i. Essa função pode ser usada para atribuir novos exemplos às classes. O problema é estimar f de modo que o rótulo previsto $f(x)$ seja o verdadeiro rótulo y para exemplos (x, y), que sejam gerados a partir da mesma distribuição de probabilidades. O problema, como representado acima, determina a função de classificação ou *classifier* f que prediz muitas classes, mas associa uma classe simples a cada ponto de dados.

Há problemas de classificação que requerem uma predição e estes são usualmente denominados problemas de classificação multi-nível. Além disso, classificadores são usualmente binários, mas o caso mais geral de problemas multi-nível pode ser resolvido pela aprendizagem de vários classificadores binários.

O campo da classificação é muito amplo e cobre uma larga variedade de áreas, incluindo biologia, ciência da informação e bio-informática. Fisher apresenta um trabalho

pioneiro usando classificadores lineares [Fisher, 1936]. Uma boa revisão de aprendizagem por máquina, métodos neurais e estatísticos, incluindo algoritmos de árvore de decisão podem ser encontrados em [Michie et al., 1994].

2.8.1 Máquina de vetores suporte

A classificação por máquina de vetores suporte (SVM - *support vector machine*) é baseada na definição de classes através de um hiperplano:

$$\langle w, x \rangle + b = 0, \tag{2.18}$$

$w, x \in \mathbb{R}^n$, $b \in \mathbb{R}$, correspondendo à função binária de decisão:

$$f(x) = sign(\langle w, x \rangle + b). \tag{2.19}$$

É possível provar que o hiperplano ótimo é o que maximiza a margem de separação entre duas classes [Vapnik, 1995]. Esse hiperplano pode ser construído resolvendo um problema de otimização quadrática restrito, cuja solução tem w como uma combinação linear do subconjunto de dados de treinamento que estão próximos da fronteira. Estes pontos de treinamento, chamados de vetores suporte, carreiam todas as informações relevantes acerca do problema de classificação. É fácil mostrar que a margem é inversamente proporcional a $\|w\|$ e que o problema é

$$\begin{aligned} \min \quad & \langle w, w \rangle \\ \text{s. a} \quad & y_i[\langle w, x_i \rangle + b] \geq 1 \end{aligned} \tag{2.20}$$

A função final de decisão pode ser escrita como

$$f(x) = \langle w, x \rangle + b = \sum_i y_i \alpha_i \langle x_i, x \rangle + b \tag{2.21}$$

onde o índice i cobre apenas os vetores suportes. Isto é, se todos os dados, além dos vetores suportes forem removidos, o algoritmo encontra a mesma solução. Essa propriedade de esparsidade é importante na implementação e análise de algoritmo. O problema de programação quadrática e a função final de decisão dependem apenas dos produtos internos entre dados e isto permite a generalização deste método para o caso não-linear via funções Kernel.

A idéia principal dos métodos Kernel é a imersão dos dados num espaço vetorial de

maior dimensão em que haja separabilidade entre as duas classes. Isto envolve o uso de técnicas de álgebra linear para identificar estrutura nos dados, [DeCoste and Scholkopf, 2002].

O uso de técnicas de máquina de vetores suporte tem obtido importantes resultados nas tarefas de classificação incluindo, categorização de textos, [Joachins, 1969], reconhecimento de dígitos, [DeCoste and Scholkopf, 2002] e com expressão genética [Brown et al., 2000]. SVM têm demonstrado melhor desempenho que outros métodos em diversas áreas.

Esse método, entretanto, nem sempre conduz a um bom resultado. De forma genérica, podem ser feitos os seguintes comentários negativos sobre a abordagem SVM, centrados basicamente em críticas à definição do Kernel:

1. Existem várias alternativas para se efetuar o Kernel;
2. Não existe procedimento de definição a *priori* da melhor alternativa;
3. A escolha da transformação é feita empiricamente;
4. Definida a transformação, a escolha dos parâmetros intrínsecos a essa transformação, também é feita por ajuste manual-empírico.

3 Procedimentos para a análise de agrupamento e classificação

3.1 Análise de agrupamento via otimização não-diferenciável

3.1.1 Introdução

Agrupamento é uma classificação não supervisionada de padrões. Na análise de agrupamento, tem-se um conjunto A finito de pontos do espaço n-dimensional $\mathbb{R}^n$, isto é

$$A = \left\{a^1, \ldots, a^m\right\}, \text{ onde } a^i \in \mathbb{R}^n, \;\; i = 1, \ldots, m.$$

Há diferentes tipos de agrupamentos, como particionais, packing, coberturas e hierárquicos. Nesta seção, serão considerados agrupamentos particionais.

O objetivo da análise de agrupamento é a partição do conjunto A em k subconjuntos disjuntos $A^i,\ i = 1, \ldots, k$ com respeito a um critério predefinido tal que

$$A = \bigcup_{i=1}^{k} A^i.$$

Os conjuntos A^i são chamados grupos.

Conforme já relatado, existem diferentes procedimentos de agrupamento. Descrições de vários desses algoritmos podem ser encontradas, por exemplo, em [Dubes and Jain, 1976], [Hawkins et al., 1982] e [Spath, 1980]. Um excelente resumo atualizado de métodos disponíveis é proporcionado por [Jain et al., 1999].

O problema de agrupamento é denominado complexo se cada ponto pertence a um e somente um grupo. Fora dos agrupamentos complexos é permitido aos grupos formar interseções (*overlaps*) com outros grupos (agrupamentos fuzzy). Neste texto, será

considerado o problema de agrupamento complexo não restrito. Isto é, é assumido que:

$$A^i \cap A^j = \emptyset, \quad \forall i, j = 1, \ldots, k$$

e nenhuma restrição é imposta aos grupos $A^i,\ i = 1, \ldots, k$. Então, cada ponto $a \in A$ está contido em exatamente um e somente um conjunto A^i.

Muitos autores reduzem o problema de agrupamento ao seguinte problema de otimização (ver, por exemplo, [Bock, 1974], [Bock, 1998] e [Spath, 1980]).

$$\begin{aligned} \min \quad & \varphi(C, x) = \frac{1}{m} \sum_{i=1}^{k} \sum_{a \in A^i} \left\| x^i - a \right\|_2^2 \\ \text{s. a} \quad & C \in \mathcal{C}, \\ & x = (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k}, \end{aligned} \tag{3.1}$$

onde $\|x\|_2$ é a norma euclidiana, $C = \left\{A^1, \ldots, A^k\right\}$ é um particular conjunto de grupos, $\mathcal{C}$ é o conjunto de todas as k-partições possíveis do conjunto A e, além disso, cada x^i representa o centro do grupo A^i com $i = 1, \ldots, k$.

Visto que o conceito de agrupamento é flexível, pode-se usar diferentes normas segundo as conveniências específicas. O seguinte problema pode ser considerado como alternativa ao problema (3.1):

$$\begin{aligned} \min \quad & \varphi_1(C, x) = \frac{1}{m} \sum_{i=1}^{k} \sum_{a \in A^i} \left\| x^i - a \right\| \\ \text{s. a} \quad & C \in \mathcal{C}, \\ & x = (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k} \end{aligned} \tag{3.2}$$

O problema (3.2) depende da escolha da norma e, consequentemente, diferentes normas podem conduzir a diferentes grupos.

A formulação (3.1) pode ser colocada sob o seguinte problema de programação não-linear inteira mista:

$$\begin{aligned} \min \quad & g(x, w) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} w_{ij} \left\| x^j - a^i \right\|^2 \\ \text{s. a} \quad & x = (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k}, \\ & \sum_{j=1}^{k} w_{ij} = 1, \quad i = 1, \ldots, m, \\ & w_{ij} \in \{0, 1\}, \quad \forall i = 1, \ldots, m, \quad \forall j = 1, \ldots, k \end{aligned} \tag{3.3}$$

onde são dados: o número k de grupos e o número m de padrões disponíveis, e se quer determinar: o vetor $x \in \mathbb{R}^{n \times k}$ de centros dos grupos e os pesos w_{ij} das associações dos padrões a^i com os grupos j dados por

$$\begin{aligned} w_{ij} &= \begin{cases} 1, & \text{se o padrão } i \text{ é alocado no grupo } j; \\ 0, & \text{caso contrário,} \end{cases} \\ \forall i &= 1, \ldots, m, \quad \forall j = 1, \ldots, k. \end{aligned}$$

Como w é uma matriz booleana $m \times k$ e $x \in \mathbb{R}^{n \times k}$ é um vetor real, então esse é um problema misto (contém tanto variáveis contínuas como inteiras).

A função objetivo $g(x, w)$ de (3.3) tem um número extremamente grande de mínimos locais. Diferentes métodos de programação matemática, como, por exemplo, programação dinâmica, branch and bound, planos cortantes e algoritmos k-média podem ser aplicados para resolver esse problema. Uma revisão desses algoritmos pode ser encontrada em [Hansen and Jaummard, 1997].

O método da programação dinâmica, vide, por exemplo, [Jensen, 1969], pode ser efetivamente aplicado ao problema de agrupamento quando o número de observações $m \leq 20$, o que significa que este método não é eficiente para resolver problemas reais.

Algoritmos de *branch and bound*, vide, por exemplo, [Diehr, 1985], [Hansen and Jaummard, 1997] e [Koontz et al., 1975], são efetivos quando o conjunto de dados contém somente centenas de registros e o número de grupos não é muito grande (menor que 5). Para esses métodos, a solução de problemas de agrupamento de grande porte está fora de propósito. Isto leva ao uso de técnicas locais e diferentes heurísticas, como o algoritmo k-média, que na essência procuram um mínimo local de um problema equivalente a (3.1).

Resultados bem melhores têm sido obtidos com metaheurísticas para otimização global, tais como Simulated Annealing, busca Tabu e algoritmos genéticos [Reeves, 1993]. O método de Simulated Annealing para agrupamento tem sido estudado, por exemplo, em [Brown and Entail, 1992], [Selim and Al-Sultan, 1991] e [Sun et al., 1994]. Aplicações de busca Tabu para resolver problemas de agrupamento estão descritas em [Al-Sultan, 1995]. Algoritmos genéticos para agrupamentos foram estudados em [Reeves, 1993].

Os resultados de experimentos numéricos apresentados em [Al-Sultan and Khan, 1996] mostram que mesmo para problemas pequenos de análise de agrupamento, com número de observações $m \leq 100$ e número de grupos $k \leq 5$, esses algoritmos consomem de 500 a

700 vezes o tempo de CPU do algoritmo k-média. Para conjuntos de dados maiores pode-se esperar uma otimização global ineficiente. Portanto, os algoritmos de metaheurística para otimização global são ineficientes para resolver muitos problemas de agrupamento de grande porte.

Uma metodologia para análise de agrupamento baseada em técnicas de programação bilinear foi apresentada em [Mangasarian, 1997]. Se uma distância poliédrica tal como a distância de norma-1 é usada, o problema de análise de agrupamento pode ser formulado como um problema de minimização do produto de duas funções lineares num conjunto que satisfaz um sistema de desigualdades lineares.

O artigo [Bagirov and Rubinov, 2001] descreve um processo de otimização global para o problema de agrupamento (3.1) e demonstra como o problema de classificação de dados supervisionados pode ser resolvido via agrupamentos. Uma apresentação detalhada das idéias principais desse artigo pode ser encontrada em [Bagirov et al., 2002]. A função objetivo do problema de classificação definida em [Bagirov et al., 2002] é tanto não-convexa como não-diferenciável e possui um grande número de minimizadores locais. Problemas desse tipo são desafiadores para as técnicas de otimização global. Como uma regra geral, devido ao grande número de variáveis e a complexidade da função objetivo, técnicas de otimização global de propósito geral falham para resolver esses problemas.

Funções objetivo de problemas de otimização que são equivalentes a (3.1) usualmente têm vários mínimos locais, que não provêem uma boa descrição do conjunto de dados. Entretanto, como mencionado acima, as técnicas de solução global consomem tempo de execução exagerado. Sendo assim, é muito importante desenvolver algoritmos de agrupamento baseados em técnicas de otimização que calculam adequadamente mínimos locais profundos.

Tais minimizadores profundos devem naturalmente proporcionar uma boa e suficiente descrição do conjunto de dados em análise. Assim, como o agrupamento é uma noção intrinsecamente flexível, assume-se que um mínimo local profundo descreve de forma satisfatória a estrutura de dados para o agrupamento.

Um método diferente, que pode ser usado com sucesso para a classificação supervisionada, é trocar o problema de otimização em análise por uma série de problemas mais simples (método passo a passo). Alguns destes métodos passo a passo podem ser usados no agrupamento.

Usualmente, os padrões em análise contêm dois tipos de atributos (coordenadas):

contínuos ou categóricos. Atributos contínuos frequentemente refletem resultados de alguma medição. Um atributo é chamado categórico, se é nominal ou ordinal. Uma variável é nominal se seus valores são códigos de possíveis estados do correspondente atributo. Se os estados de uma variável nominal podem ser arranjados numa ordem significativa, então ela é uma variável ordinal.

A experiência demonstra que os algoritmos de otimização trabalham muito melhor quando o conjunto de dados contém vetores com variáveis contínuas. Entretanto, algoritmos de otimização podem também ser usados para alguns conjuntos de dados com atributos categóricos, havendo, em geral, necessidade de uma quantificação criteriosa desses últimos.

3.1.2 Otimização não-diferenciável para agrupamentos

Aqui, descreve-se a abordagem do problema de agrupamento apresentada em [Bagirov et al., 2002]. Essa abordagem leva à formulação de um problema de otimização não-convexo e não-diferenciável equivalente a (3.2), no entanto mais simples do ponto de vista computacional.

Considera-se um espaço real n-dimensional com uma norma $\|\cdot\|$. Como regra, assume-se que $\|\cdot\| = \|\cdot\|_p$, $1 \leq p \leq \infty$, onde

$$\|x\|_p = \left(\sum_{l=1}^{n} |x_l|^p \right)^{1/p}.$$

Considere um conjunto A de m vetores n-dimensionais $a = (a_1, \ldots, a_m)$. O propósito do agrupamento é representar este conjunto como a união de k classes. Aceita-se a hipótese de que cada classe pode ser descrita por um ponto que pode ser considerado como seu centro. Assim deve-se definir os centros a fim de descrever adequadamente as classes $A^1, \ldots, A^k$. Primeiro, é necessária uma definição formal dos centros de um sistema finito de conjuntos de classes disjuntas $A^1, \ldots, A^k$, tendo em conta que estes conjuntos são desconhecidos e deseja-se apenas a sua união.

Considere um conjunto arbitrário X. A distância $d(a, X)$ de um ponto $a \in A$ a este conjunto é definida por

$$d(a, X) = \min_{x \in X} \|x - a\| \tag{3.4}$$

O desvio $d(A, X)$ do conjunto A para o conjunto X é definido como

$$d(A, X) = \sum_{a \in A} d(a, X)$$

O conjunto $\overline{X} = \{\overline{x}^1, \ldots, \overline{x}^k\}$ é dito conjunto ótimo de centros das k classes do conjunto A se $d(A, \overline{X}) \leq d(A, X)$, para qualquer $X = \{x^1, \ldots, x^k\}$. Então, a busca pelos centros das classes (daí, a busca pelos grupos) pode ser reduzida ao seguinte problema de minimização

$$\begin{array}{ll} \min & C_k(x^1, \ldots, x^k) \\ \text{s. a} & (x^1, \ldots, x^k) \in \mathbb{R}^{n \times k} \end{array} \tag{3.5}$$

onde

$$C_k(x^1, \ldots, x^k) = \frac{1}{m} \sum_{a \in A} \min_{s=1,\ldots,k} \|x^s - a\| \tag{3.6}$$

A função C_k definida em (3.6) será chamada função agrupamento. A Figura 14 ilustra um gráfico da função agrupamento C_2 em $\mathbb{R}^2$ para um conjunto de 20 dados.

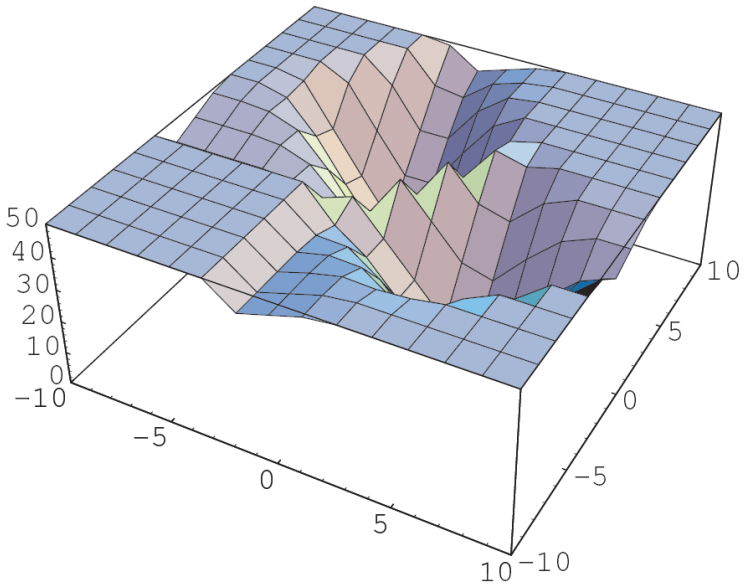

Figura 14: Função agrupamento em $\mathbb{R}^2$.

Se $k > 1$, a função agrupamento é não-convexa e não-diferenciável. Pode ser mostrado que esta função possui vários mínimos locais, que normalmente são próximos entre si.

Note que o número de variáveis no problema de otimização (3.5) é $n \times k$. Se o número k de classes e o número de atributos são grandes, tem-se um problema de otimização global de grande porte. No entanto, a forma da função objetivo nesse problema ainda é

bastante complexa para a aplicação direta de métodos de otimização global. Portanto, dentro de uma perspectiva de natureza prática, o uso de métodos de otimização local se configura como a única alternativa adequada.

Obviamente, o uso de métodos de otimização local não garante uma solução global para o problema (3.5). Entretanto, como o agrupamento é uma noção flexível, não é necessário ter uma solução exata para (3.5), é suficiente ter uma boa aproximação para esta solução. Esta aproximação está associada à profundidade do mínimo local da função agrupamento. É, assim, natural admitir que um mínimo local profundo provê uma boa descrição do conjunto de dados em consideração. Dentro desta perspectiva, bucam-se mínimos locais profundos. O método de suavização, que será apresentado mais a frente, tem exatamente esse propósito.

É mostrado em [Bock, 1974] que os problemas (3.1), (3.3) e (3.5) são equivalentes. O número de variáveis no problema (3.3) é $(m+n) \times k$, enquanto no problema (3.5) este número é apenas $n \times k$, ou seja, o número de variáveis não depende do número de observações. Deve ser notado que em muitas bases de dados reais o número de observações m é substancialmente maior que o número de atributos n. Além disso, os coeficientes w_{ij} no problema (3.3) são inteiros, enquanto que o problema (3.5) possui apenas variáveis contínuas. Todas estas circunstâncias podem ser consideradas como vantagens da formulação não-diferenciável (3.5) sobre a formulação (3.1) e sua versão (3.3).

3.1.3 A função agrupamento como uma ferramenta para medir a qualidade de um ajuste

Muitas vezes, diferentes vetores $X = (x^1, \ldots, x^k)$ podem ser considerados como centros das classes. É uma tarefa importante comparar esses candidatos. Agora, será descrita uma situação bastante simples onde se faz necessária tal comparação.

Assuma que um certo método local será utilizado para achar os centros das classes. Esse método pode ser utilizados muitas vezes a partir de diferentes pontos iniciais. Como regra geral, diferentes soluções serão obtidas. Qual é a melhor dessas soluções?

Pode-se usar uma função agrupamento para a comparação. Assuma que a norma $\|\cdot\|$ é fixada e que a função agrupamento C_k é gerada por esta norma. Segue diretamente da definição da função agrupamento, que o candidato $X_* = (x_*^1, \ldots, x_*^k)$ será melhor ajustado para o papel de centro do que o candidato $X = (x^1, \ldots, x^k)$ no sentido da

norma $\|\cdot\|$, se

$$C_k(x_*^1, \ldots, x_*^k) < C_k(x^1, \ldots, x^k).$$

3.1.4 O algoritmo *k*-média

Nesta seção, apresenta-se o algoritmo k-média que é um dos mais eficientes para resolver problemas de agrupamento de larga escala. Esse algoritmo é baseado na minimização das variâncias dentro das classes e da maximização simultânea das variâncias entre as classes. A variância depende da norma escolhida e diferentes versões desse método foram estudadas (ver [Spath, 1980]). Usualmente, a norma $\|\cdot\|_2$ é usada para esse propósito, entretanto versões do k-média com diferentes normas podem também ser usadas [McQueen, 1967].

Em sua versão mais simples, o algoritmo k-média assume como entrada uma k-partição arbitrária do conjunto das m observações e segue os seguintes passos:

Passo 1: Determina o centróide de cada um dos grupos da partição.

Passo 2: Gera uma nova partição associando cada uma das observações ao centróide mais próximo no sentido da norma escolhida.

Passo 3: Se nenhuma das observações muda de grupo, finaliza. Caso contrário, retorna ao passo 1.

O algoritmo k-média é muito rápido e portanto adequado para resolver problemas com grande dimensão. Esse algoritmo dá bons resultados quando há poucas classes, mas se deteriora quando são várias classes [Hansen and Jaummard, 1997]. O algoritmo k-média encontra um mínimo local do problema (3.1) (ver [Selim and Ismail, 1984]), sendo que certos experimentos mostram que o melhor agrupamento encontrado com k-média pode ser 50% pior do que a melhor solução [Hansen and Jaummard, 1997]. Isto é porque o k-média, tal como a maioria dos métodos locais, é muito sensível ao ponto inicial. Dessa forma, esses experimentos mostram que o algoritmo k-média usualmente conduz a mínimos locais de (3.1) que não descrevem bem a estrutura dos dados.

Devido à sua rápida convergência, o método k-média é muitas vezes utilizado para geração de soluções iniciais de métodos mais complexos para a solução de problemas de agrupamento.

3.1.5 Um algoritmo de otimização para o agrupamento

Será apresentado a seguir o algoritmo proposto por Bagirov e Yearwood em [Bagirov and Yearwood, 2003] para resolver o problema de agrupamento. Trata-se de um procedimento iterativo para resolver o problema (3.5) no qual, em cada iteração, é resolvido um problema de otimização de grandes dimensões, que é não-convexo e não-diferenciável. Portanto, não necessariamente será determinada uma solução global. Em geral, determina-se uma solução local, idealmente profunda.

Conforme já apresentado anteriormente, o problema a ser resolvido consiste em particionar o conjunto de n pontos em k classes de modo que a função agrupamento seja minimizada. Essas k classes estão associadas a k variáveis que representam os seus centros. Como cada variável é um ponto do $\mathbb{R}^n$, o total de variáveis do problema é $n \times k$.

Uma escolha adequada do número de classes é muito importante para a análise de agrupamento. É difícil definir *a priori* quantas classes representam um conjunto A em análise. A fim de aumentar o conhecimento a respeito do agrupamento, o decisor pode partir de um pequeno número de classes k e, gradualmente, aumentar essa quantidade até que certo critério de terminação seja satisfeito. Ou seja, se a solução do problema (3.5) com k classes não for satisfatória, o decisor deve considerar o problema (3.5) com $k+1$ classes e assim por diante. Neste caso, será necessário resolver repetidamente o problema de otimização (3.5) com diferentes e cada vez maiores valores de k. A fim de evitar essa dificuldade, Bagirov e Yearwood propõem uma determinação de classes passo a passo.

A idéia central do algoritmo proposto por Bagirov e Yearwood em [Bagirov and Yearwood, 2003] é usar os resultados obtidos num certo passo para encontrar um boa solução inicial para o próximo passo. Note que a função agrupamento para $k = 1$ é convexa e, portanto, nesse caso, as técnicas de programação convexa podem ser utilizadas.

Esse algoritmo tem dois atributos importantes:

- permite ao decisor manipular o conjunto de dados para reduzir com sucesso o número de observações (registros) do conjunto em análise sem perda de informação disponível;
- provê a capacidade de calcular grupos passo a passo, gradualmente aumentando o número de grupos até que uma condição de parada é satisfeita, isto é, permite

calcular tantos grupos quanto os contidos no conjunto de dados com relação a alguma especificada tolerância.

Algoritmo 3.1 Um algoritmo para resolver o problema de análise de agrupamento.

Passo 1 (Inicialização) Selecione uma tolerância $\epsilon > 0$ e um inteiro positivo k_0 como número de partida do total de classes. Faça $k = k_0$, selecione um ponto de partida em $\mathbb{R}^{n \times k_0}$ e resolva o problema de minimização (3.5) com $k = k_0$. Seja $x^{1*} \in \mathbb{R}^{n \times k_0}$ uma solução desse problema e C_{1*} o correspondente valor da função objetivo.

Passo 2 (Cálculo do próximo centro de classe) Selecione um ponto $x^0 \in \mathbb{R}^n$ e resolva o seguinte problema de minimização de dimensão n.

$$\begin{aligned} \min \quad & \overline{C}_k(x) \\ \text{s. a} \quad & x \in \mathbb{R}^n, \end{aligned} \tag{3.7}$$

onde

$$\overline{C}_k(x) = \sum_{a \in A} \min \left\{ \left\| x^{1*} - a \right\|, \ldots, \left\| x^{k*} - a \right\|, \|x - a\| \right\}.$$

Passo 3 (Refinamento dos centros das classes). Seja $\overline{x}^{k+1,*}$ a solução do problema (3.7). Com função objetivo C_{k+1}, resolva o problema (3.5) tomando $x^{k+1,0} = (x^{1*}, \ldots, x^{k*}, \overline{x}^{k+1,*})$ como ponto de partida.

Passo 4 (Critério de Parada) Seja $x^{k+1,*}$ a solução obtida no passo anterior e $C_{k+1,*}$ o correspondente valor da função objetivo. Se

$$\frac{C_{k*} - C_{k+1,*}}{C_{1*}} < \epsilon$$

então pare, caso contrário faça $k = k + 1$ e vá para o Passo 2.

No Passo 1, os centros das primeiras k_0 classes são calculados. Em particular, se for tomado $k_0 = 1$, então o centro do conjunto total A será calculado. No Passo 2, calcula-se um centro da próxima classe $(k + 1)$, assumindo conhecidos os centros das k classes anteriores. Note que o número de variáveis do problema (3.7) é n, o qual é substancialmente menor que o número de variáveis quando todas as classes são calculadas simultaneamente. No Passo 3, o refinamento de todos os centros é efetuado. É bem possível que o ponto $x^{k+1,0}$ calculado no passo anterior, Passo 2, não esteja distante daquele obtido na solução de (3.5). Neste caso, um número moderado de iterações será

suficiente para obter uma nova solução. Esta estratégia permite uma significativa redução do tempo de cálculo para resolver o problema (3.5)

Tem-se que $C_{k*} \geq 0$ para todo $k \geq 1$ e que a seqüência $\{C_{k*}\}$ é monótona decrescente, ou seja,

$$C_{k+1,*} \leq C_{k,*}, \ \ \forall k \geq 1.$$

Daí, após certo número iterações, o critério de parada no Passo 4 será satisfeito.

3.1.6 Redução da complexidade para conjuntos de dados de grande porte via funções de agrupamento generalizadas

Devido a natureza combinatória do problema de agrupamento, duas características do conjunto de dados podem afetar severamente o desempenho da ferramenta de agrupamento: o número de registros de dados (observações) e o número de atributos de dados. Em muitos casos o desenvolvimento de ferramentas eficientes requer a redução de ambos os números, isto é, observações e atributos, sem perda da capacidade de conhecimento. Primeiro considera-se a redução do número de observações. A redução do número de atributos será discutida posteriormente.

Grandes bancos de dados usualmente contém um imenso número de pontos localizados dentro de um conjunto limitado. Então, vários pontos do conjunto de dados estão muito próximos uns dos outros. Seja $A \subset \mathbb{R}^n$ um conjunto finito. Assuma que uma certa vizinhança de um ponto $b \in \mathbb{R}^n$ contém m_b pontos de A. Pode-se aproximar cada um destes pontos por b e substituir a parte correspondente da função agrupamento pelo termo $m_b \times \min_i \|x_i - b\|$.

Mais precisamente, para um dado conjunto A e para uma dada tolerância ϵ considere um conjunto $B \subset \mathbb{R}^n$, tal que para cada $a \in A$ exista $b \in B$ com a propriedade $\|b - a\| < \epsilon$. A coleção $(A_b)_{b \in B}$ de subconjuntos de A é dita uma ϵ-cobertura disjunta de A se

$$\begin{aligned} \|a - b\| &< \epsilon, \ \ (a \in A_b), \\ A_b \cap A_{b'} &= \emptyset, \ \ (b \neq b'), \\ A &= \bigcup_{b \in B} A_b. \end{aligned}$$

Seja m_b a cardinalidade de A_b. Substituindo cada $a \in A_b$ por b na apresentação da função agrupamento C_k, obtém-se a seguinte função de agrupamento

generalizado:

$$\tilde{C}_k(x^1, \ldots, x^k) = \frac{1}{m} \sum_{b \in B} m_b \min \left\{ \left\| x^1 - b \right\|, \ldots, \left\| x^k - b \right\| \right\}$$

A Figura 15 ilustra uma função de uma variável $f(x) = \tilde{C}_2(x, \tilde{x}_2)$ com $\tilde{x}_2$ fixo, onde $\tilde{C}_2$ é uma função de agrupamento generalizada em $\mathbb{R}^1$ para um conjunto de dados de 23 pontos. A figura 15 ilustra a multiplicidade de pontos de mínimo local.

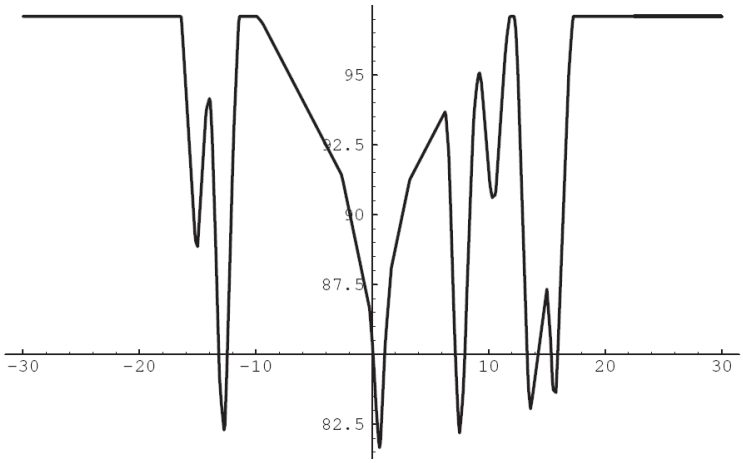

Figura 15: Função agrupamento generalizada em $\mathbb{R}^1$.

Pode-se usar funções de agrupamento generalizadas para efetuar uma adequada aproximação de funções de agrupamento. E, visto que a noção de agrupamento é intrinsecamente flexível, pode-se utilizar uma aproximação apropriada da função de agrupamento, mesmo que essa aproximação não seja exata.

3.1.7 Redução da complexidade para conjuntos de dados de grande porte via seleção de atributos

Os métodos de solução para os problemas de agrupamento sugeridos por vários autores tornam-se ineficientes em conjuntos de dados de grande porte. Para contornar esse problema, foram propostos algoritmos de seleção de atributos. Esses algoritmos visam determinar um conjunto tão pequeno quanto possível de atributos que descreva adequadamente o conjunto das observações sob o ponto de vista da classificação. Alguns métodos estatísticos são usados para a seleção de atributos, por exemplo, análise de componentes principais, ver [Mirkin, 1996] e referências.

Uma abordagem para efetuar a seleção de atributos supõe que a informação não é uma propriedade individual do atributo, ela depende da estrutura de classificação dos dados

em consideração. Em princípio, um conjunto de atributos de informação deve ajudar a diferenciar as classes. É melhor considerar atributos de informação não individuais, mas subconjuntos de atributos de informações. Um exemplo simples: se os valores de x_1 são proporcionais aos valores de x_2, pode-se considerar x_1 como informativo e x_2 como supérfluo ou vice-versa.

Assumindo que os centros dos grupos fornecem alguma informação sobre a estrutura das classes, a abordagem acima pode ser implementada através de um algoritmo de seleção de atributos que parta do agrupamento no interior de cada classe. Então, usando um conjunto de regras, alguns atributos são removidos, passo a passo, até que a estrutura das classes começa a se alterar. Mais especificamente, se os grupos gerados após a remoção de algum atributo são diferentes dos anteriores, então o atributo eliminado é informativo (sua eliminação levou a um resultado diferente) e o processo é finalizado. Caso contrário, o atributo não é informativo e pode ser eliminado.

3.2 Classificação supervisionada via agrupamento

3.2.1 Introdução

O objetivo da classificação de dados supervisionada é estabelecer regras para a classificação de um conjunto de observações, assumindo que as classes já são conhecidas. Para encontrar essas regras, um pesquisador pode usar conhecimento sobre subconjuntos de treinamento. A construção do procedimento de classificação pode também ser um procedimento de reconhecimento de padrões, um procedimento de discriminação ou um procedimento de aprendizagem supervisionada. Esses problemas ocorrem num amplo campo das atividades humanas.

Há vários métodos para a classificação de dados, que são baseados em diferentes metodologias: estatística, redes neurais, métodos de teoria da informação, etc. Uma excelente revisão destes métodos, incluindo a parte computacional e comparações, pode ser encontrada em [Michie et al., 1994]. Métodos estatísticos para a classificação são descritos em [McLachlan, 1992].

Um dos mais promissores enfoques para a classificação de dados é baseado em métodos de otimização matemática. Existem duas diferentes formas de aplicação de técnicas de otimização para a classificação supervisionada de uma base de dados que consiste de pelo menos duas classes e em que há um conjunto treinamento para cada classe.

A primeira, dita externa, é baseada na separação do conjunto treinamento por meio de uma certa (não necessariamente linear) função. O método externo é atualmente o mais popular. Em [Bradley and Mangasarian, 2000], problemas de programação quadráticos e bilineares são usados para definir o processo de classificação.

A segunda metodologia, dita interna, consiste em descrever o agrupamento para um dado conjunto treinamento. Os vetores de dados são atribuídos ao grupo mais próximo e, consequentemente, ao conjunto que contém esse grupo. Para a implementação desse método é necessário resolver um problema complexo, não-convexo e não-diferenciável de otimização, seja local ou globalmente.

3.2.2 O método interno para a classificação

Assuma que tem-se um conjunto de dados consistindo de l classes $A_i, \ldots, A_l$. Assuma que a classe A_j seja composta por k_j grupos. Assim, pode-se usar a minimização da função agrupamento C_{k_j} a fim de encontrar os centros destes grupos e, então, usá-los para a classificação. A primeira vista, este método não é adequado. De fato, na verdade reduz-se o problema mais simples de classificação supervisionada a uma série de problemas mais complicados de classificação não supervisionada. Entretanto, a presença de classes conhecidas facilita substancialmente a busca de grupos.

A função agrupamento C_k depende de $n \times k$ variáveis, onde n é o número de atributos e k é o número de grupos. Se as classes são conhecidas, pode-se aplicar um certo procedimento de seleção para diminuir o número de atributos, e daí pode-se usar somente $n_1 < n$ atributos. Segundo, pode-se determinar os centros dos grupos passo a passo. Isto significa que será considerada uma série de k problemas de dimensão n_1 ao invés de um problema de dimensão $n_1 \times k$. A solução dessa série é muito mais fácil do que a solução de um único problema com dimensão maior. A seguir, é apresentado o procedimento passo a passo usado para a classificação supervisionada.

Seja A um conjunto composto de duas classes A_1 e A_2 . Assumindo que cada classe A_j $(j = 1, 2)$ consiste de um único grupo, pode-se calcular seus centros pela solução do seguinte problema de programação convexa.

$$\begin{aligned} \min \quad & C_1(x^j) = \sum_{a \in A_j} \left\| x^j - a \right\|. \\ \text{s. a} \quad & x^j \in \mathbb{R}^n \end{aligned} \tag{3.8}$$

A influência de todos os pontos de um dado conjunto, como por exemplo A_1, está nele refletida. Tendo o centro de A_2, pode-se definir pontos não classificados de A_1 como pontos que são mais próximos do centro A_2 do que do centro A_1. Removendo todos os pontos não classificados e resolvendo novamente o problema (3.1) para $j = 1$, pode-se encontrar uma solução mais precisa para o centro x_1 do conjunto A^1. Pode-se considerar x_1 como o centro do primeiro grupo (principal) do conjunto A_1. Então, pode-se olhar para o centro do segundo grupo com o conhecimento do centro do primeiro grupo e assim por diante. Uma idéia similar pode ser usada para a redução da complexidade do processo.

3.3 Procedimento passo a passo para encontrar os centros dos agrupamentos

Nesta seção, será feito um preâmbulo para o algoritmo de classificação. Será apresentado um método refinado, baseado na presença de classes conhecidas, que permite encontrar grupos do conjunto de dados por um processo passo a passo.

Seja A_j uma das classes do conjunto A. Assumindo que cada classe é formada por um único grupo, pode-se determinar o centro de cada classe, isto é, o centro de cada subconjunto A_j, resolvendo o problema (3.8). Um refinamento evidente pode ser feito removendo todos os pontos não classificados corretamente e, em seguida, resolvendo novamente o problema (3.8), mas desta vez considerando somente os pontos corretamente classificados. Cada solução x_j^1 obtida pode ser considerada como centro do primeiro grupo de cada classe A_j.

A fim de determinar o centro do segundo grupo de cada classe, resolve-se para cada A_j o seguinte problema:

$$\begin{aligned} \min f_{2j}(x) &= \sum_{a \in A_j} \min \left\{ \left\| x_j^1 - a \right\|, \|x - a\| \right\}. \\ \text{s. a} \quad & x \in \mathbb{R}^n, \end{aligned} \tag{3.9}$$

Assuma que já foram calculados os centros $x_j^1, \ldots, x_j^{t-1}$ de $t-1$ grupos em cada classe A_j. Então, o centro x_j^t do t-ésimo grupo é definido como a solução do seguinte problema:

$$\begin{aligned} \min \quad & f_{tj}(x) \\ \text{s. a} \quad & x \in \mathbb{R}^n, \end{aligned} \tag{3.10}$$

onde

$$f_{tj}(x) = \sum_{a \in A_j} \min \left\{ \left\| x_j^1 - a \right\|, \ldots, \left\| x_j^{t-1} - a \right\|, \|x - a\| \right\}. \tag{3.11}$$

Note que a minimização da função agrupamento requer muito mais tempo do que a solução de uma série de problemas (3.10), já que o número de variáveis em (3.10), i.e., n, é substancialmente menor que em (3.5).

Seja $X_j^t = \left\{ x_j^1, \ldots, x_j^t \right\}$ o conjunto formado pelos centros dos t primeiros grupos da classe A_j. Então, pode-se apresentar a função f_{tj} definida por (3.11) na seguinte forma:

$$f_{tj}(x) = \sum_{a \in A_j} \min \left\{ d(a, X_j^{(t-1)}), \|x - a\| \right\}, \quad t \geq 2.$$

Assim, tem-se o seguinte problema de otimização global:

$$\begin{aligned} \min f_{tj}(x) &= \sum_{a \in A_j} \min \left\{ d(a, X_j^{(t-1)}), \|x - a\| \right\} \\ \text{s. a} \quad & x \in \mathbb{R}^n. \end{aligned} \tag{3.12}$$

Está claro que

$$0 \leq f_{tj}(x) \leq f_{(t-1)j}(x) \leq \ldots \leq f_{1j}(x)$$

para todo $x \in \mathbb{R}^n$ e que portanto esta seqüência é convergente. Assim, denotando por x_j^t a solução do problema (3.12) na iteração t, pode-s assumir como critério de parada

$$f_{(t-1)j}(x_j^{t-1}) - f_{tj}(x_j^t) < \epsilon,$$

onde $\epsilon > 0$ é alguma tolerância, pois a solução do problema (3.12) na próxima iteração não irá melhorar significativamente a descrição da classe A_j.

Segue-se imediatamente da definição da função f_{tj} , que para todo $x \in \mathbb{R}^n$ vale

$$\begin{aligned} f_{tj}(x) &= \sum_{a \in A_j} \min \left\{ d(a, X_j^{t-1}), \|x - a\| \right\} \\ &\leq \sum_{a \in A_j} d(a, X_j^{t-1}) \\ &= d(A_j, X_j^{t-1}) \end{aligned}$$

Ou seja, é válida a seguinte afirmativa (ver [Bagirov et al., 2002]):

Proposição 3.1

$$f_{tj}(x) \leq d(A_j, \left\{ x_j^1, \ldots, x_j^{t-1} \right\}), \ \forall x \in \mathbb{R}^n.$$

Feito o preâmbulo, será apresentado o algoritmo principal de classificação.

3.4 O principal algoritmo de classificação

Nesta seção, será apresentado uma descrição do algoritmo de classificação para a solução do problema de classificação.

As p-normas com $p = 1$ ou $p = 2$ podem ser usadas indistintamente. Por simplicidade, será considerada uma base de dados que contém unicamente as classes A_1 e A_2.

Sejam $\epsilon > 0$, $N_1 = \{1, \ldots, |A_1|\}$, $N_2 = \{|A_1| + 1, \ldots, |A_1| + |A2|\}$.

Algoritmo 3.2 Algoritmo de Classificação.

Passo 1 (Inicialização) Determinação dos centros dos grupos assumindo que A_1 e A_2 contêm um único grupo.

$$\begin{aligned} \min \quad & \sum_{i \in N_1} \left\| x^1 - a^i \right\| && (3.13) \\ \min \quad & \sum_{i \in N_2} \left\| x^2 - a^i \right\| && (3.14) \\ \text{s. a} \quad & x^j \in \mathbb{R}^n, \ \ j = 1, 2. \end{aligned}$$

Faça $k = 1$.

Sejam x^*_{1k} e x^*_{2k} as soluções do problema (3.13) e (3.14) e sejam f^*_{1k} e f^*_{2k} os valores destes problemas respectivamente.

Passo 2 Encontre o conjunto de pontos que não foram classificados corretamente nos grupos atuais. Ou seja, calcule os conjuntos:

$$\begin{aligned} N^*_{1k} &= \left\{ i \in N_1 : \min_{t=1,\ldots,k} \left\| x^*_{2t} - a^i \right\| \leq \min_{t=1,\ldots,k} \left\| x^*_{1t} - a^i \right\| \right\}, \\ N^*_{2k} &= \left\{ i \in N_2 : \min_{t=1,\ldots,k} \left\| x^*_{1t} - a^i \right\| \leq \min_{t=1,\ldots,k} \left\| x^*_{2t} - a^i \right\| \right\}. \end{aligned}$$

Passo 3 Calcule os seguintes conjuntos:

$$\begin{aligned} K_1 &= \left\{ i \in N_1 \backslash N^*_{1k} : \left\| x^*_{1k} - a^i \right\| \leq \min_{t=1,\ldots,k-1} \left\| x^*_{1t} - a^i \right\| \right\}, \\ K_2 &= \left\{ i \in N_2 \backslash N^*_{2k} : \left\| x^*_{2k} - a^i \right\| \leq \min_{t=1,\ldots,k-1} \left\| x^*_{2t} - a^i \right\| \right\}. \end{aligned}$$

Passo 4 Refine os centros dos grupos usando apenas pontos que são próximos dos centros desses grupos, ou seja, os pontos que pertencem aos conjuntos K_1 e K_2.

Resolva o seguinte problema de programação convexa:

$$\begin{aligned} \min \quad & \sum_{i \in N_1} \left\| x^1 - a^i \right\| && (3.15) \\ \min \quad & \sum_{i \in N_2} \left\| x^2 - a^i \right\| && (3.16) \\ \text{s. a} \quad & x^j \in \mathbb{R}^n, \ \ j = 1, 2. \end{aligned}$$

Sejam x^{01} e x^{02} as soluções dos problemas (3.15) e (3.16) respectivamente. Faça $x^*_{1k} = x^{01}$ e $x^*_{2k} = x^{02}$.

Passo 5 Determine o próximo grupo.

Resolva os seguintes problemas de otimização:

$$\begin{aligned} \min \quad & \sum_{i \in N_1} \min \left\{ \left\| x^1 - a^i \right\|, \left\| x^*_{11} - a^i \right\|, \ldots, \left\| x^*_{1k} - a^i \right\| \right\} && (3.17) \\ \min \quad & \sum_{i \in N_2} \min \left\{ \left\| x^2 - a^i \right\|, \left\| x^*_{21} - a^i \right\|, \ldots, \left\| x^*_{2k} - a^i \right\| \right\} && (3.18) \\ \text{s. a} \quad & x \in \mathbb{R}^n, \ \ j = 1, 2. \end{aligned}$$

Passo 6 Seja x^{11} e x^{12} as soluções, e $f_{1,k+1}$ e $f_{2,k+1}$ os valores dos problemas (3.17) e (3.18), respectivamente. Faça $x^*_{1,k+1} = x^{11}$ e $x^*_{2,k+1} = x^{12}$.

Passo 7 Verificando o critério de parada.

Se

$$\max \left\{ \frac{|f_{1,k+1} - f_{1k}|}{f_{11}}, \frac{|f_{2,k+1} - f_{2k}|}{f_{21}} \right\} < \epsilon,$$

então o algoritmo termina. Caso contrário, faça $k = k + 1$ e vá para o Passo 2.

Nota: Visto que os problemas (3.17) e (3.18) tem a forma (3.12), a fim de aplicar este algoritmo para conjuntos de dados reais, é necessário resolver eficientemente o problema de minimização (3.12). Note que o algoritmo pode ser abordado com dois diferentes enfoques: um, utilizando técnicas de otimização local, e outro, o ferramental de otimização global, na resolução do problema (3.12). Então, diferentes métodos de otimização numérica conduzem a diferentes versões desse algoritmo.

Observações: Essa estratégia proposta por Bagirov apresenta as seguintes vantagens em

relação às demais alternativas de classificação:

1. o enfoque é geral;
2. o enfoque é adaptativo à conformação das classes;
3. pode ser facilmente estendido para efetuar a classificação com qualquer número de classes;
4. Não carece de transformações mais elaboradas.

4 Proposta de solução do problema de classificação

Nesta seção, propõem-se uma alternativa diferenciável para a resolução de problemas da forma (3.12), entre os quais encontram-se os principais problemas presentes no algoritmo de classificação (3.2). Significa dizer que, neste trabalho, a preocupação principal é com uma proposta alternativa ao algoritmo (3.2), de modo que os demais algoritmos não serão objeto de análise mais profunda.

Nesta proposta, sugere-se que o principal problema de otimização do algoritmo (3.2), seja resolvido através de um processo de suavização hiperbólica que garante a diferenciabilidade do problema. Com esta proposta, embora não seja possível garantir um mínimo global, espera-se obter mínimos locais profundos, já que como será visto o processo de suavização pode reduzir o número de mínimos locais.

Na solução do problema de classificação utilizando o algoritmo (3.2), tem-se como maior desafio a solução dos problemas de otimização (3.17) e (3.18) do passo 5, explicitamente,

$$\begin{aligned} \min \quad & \sum_{i \in N_1} \min \left\{ \left\| x^1 - a^i \right\|, \left\| x_{11}^* - a^i \right\|, \ldots, \left\| x_{1k}^* - a^i \right\| \right\} \\ \min \quad & \sum_{i \in N_2} \min \left\{ \left\| x^2 - a^i \right\|, \left\| x_{21}^* - a^i \right\|, \ldots, \left\| x_{2k}^* - a^i \right\| \right\} \\ \text{s. a} \quad & x^j \in \mathbb{R}^n, \;\; j = 1, 2. \end{aligned}$$

É considerada uma base de dados que contém duas classes: A_1 e A_2, sendo

$$\begin{aligned} N_1 &= \{1, \ldots, |A_1|\}, \\ N_2 &= \{|A_1| + 1, \ldots, |A_1| + |A_2|\}. \end{aligned}$$

Com o sentido de simplificar a definição dos problemas de otimização (3.17) e (3.18),

define-se

$$D_{ji} = \min\left\{\left\|x_{j1}^* - a^i\right\|, \ldots, \left\|x_{jk}^* - a^i\right\|\right\},\ i \in N_j, \tag{4.1}$$

onde $x_{j1}^*, \ldots, x_{jk}^*$ são constantes que representam as coordenadas dos k centros da classe j gerados da primeira até a k-ésima iteração do algoritmo (3.2).

Para a obtenção de uma forma mais conveniente do problema à aplicação da suavização, serão definidas as variáveis

$$Z_{ji}(x^j) = \min\left\{\left\|x^j - a^i\right\|, D_{ji}\right\},\ i \in N_j. \tag{4.2}$$

Usando as definições (4.1) e (4.2), tem-se:

$$\begin{aligned}\min_{x^j} \sum_{i\in N_j} \min\left\{\left\|x^j - a^i\right\|, \left\|x_{j1}^* - a^i\right\|, \ldots, \left\|x_{jk}^* - a^i\right\|\right\} &= \min_{x^j} \sum_{i\in N_j} \min\left\{\left\|x^j - a^i\right\|, D_{ji}\right\} \\ &= \min_{x^j} \sum_{i\in N_j} Z_{ji}(x^j).\end{aligned}$$

4.1 Transformação do problema

Para $j = 1, 2$, serão considerados os problemas equivalentes a (3.17) e (3.18)

$$\begin{aligned}\min \quad & \sum_{i\in N_j} Z_{ji} \\ \text{s. a} \quad & Z_{ji} = \min\left\{\left\|x^j - a^i\right\|, D_{ji}\right\},\ \ i \in N_j.\end{aligned} \tag{4.3}$$

Considerando esta definição, cada Z_{ji} deve necessariamente satisfazer as seguintes restrições de desigualdade:

$$Z_{ji} \leq \left\|x^j - a^i\right\|,\ \ \forall i \in N_j; \tag{4.4}$$

$$Z_{ji} \leq D_{ji},\ \ \forall i \in N_j. \tag{4.5}$$

Substituindo as restrições de igualdade do problema (4.3) pelas desigualdades acima, para cada $j = 1, 2$, obtém-se o seguinte o problema relaxado:

$$\begin{aligned}\min \quad & \sum_{i\in N_j} Z_{ji} \\ \text{s. a} \quad & Z_{ji} \leq \left\|x^j - a^i\right\|, \\ & Z_{ji} \leq D_{ji}, \\ & i \in N_j.\end{aligned} \tag{4.6}$$

Uma vez que as variáveis Z_{ji} são inferiormente ilimitadas, então o problema relaxado (4.6) também será inferiormente ilimitado. Segue que, para se obter a equivalência desejada entre os problemas (4.3) e (4.6), é necessário modificar este último. Para tal, primeiro define-se $\varphi(y) = \max\{y, 0\}$ e observa-se que para o conjunto de desigualdades do problema (4.6) vale

$$\varphi(Z_{ji} - \left\|x^j - a^i\right\|) + \varphi(Z_{ji} - D_{ji}) = 0, \quad i \in N_j. \tag{4.7}$$

Para j, i fixos, são definidos

$$\begin{aligned} d_1 &= \min\left\{\left\|x^j - a^i\right\|, D_{ji}\right\}, \\ d_2 &= \max\left\{\left\|x^j - a^i\right\|, D_{ji}\right\}. \end{aligned}$$

Pela definição, tem-se $d_1 \leq d_2$. A Figura 16 mostra as duas parcelas de (4.7) em função de Z_{ji}, considerando a situação em que prevalece a relação mais frequente em que se manifesta a desigualdade estrita $d_1 < d_2$.

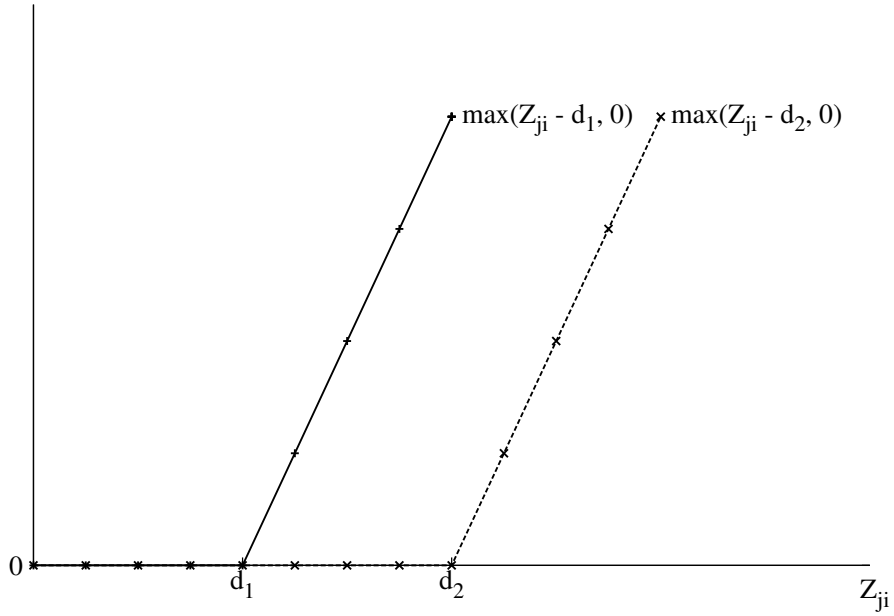

Figura 16: Os valores das parcelas do lado esquerdo da equação (4.7).

Usando as igualdades (4.7) no lugar das restrições de desigualdade do problema (4.6), seria obtido um novo problema onde Z_{ji} continuaria inferiormente ilimitado. Note que a função objetivo do problema (4.6) força a redução de cada Z_{ji}, portanto pode se pensar em limitar estas variáveis inferiormente substituindo o sinal "=" em (4.7) pelo sinal

">". Obtém-se assim o problema não canônico, i.e., com restrições de desigualdade estrita

$$\begin{aligned} \min \quad & \sum_{i \in N_j} Z_{ji} \\ \text{s. a} \quad & \varphi(Z_{ji} - \left\|x^j - a^i\right\|) + \varphi(Z_{ji} - D_{ji}) > 0, \;\; i \in N_j. \end{aligned} \tag{4.8}$$

A formulação canônica pode ser recuperada a partir de (4.8) pela adição de um termo perturbação ϵ. Considera-se assim o problema modificado

$$\begin{aligned} \min \quad & \sum_{i \in N_j} Z_{ji} \\ \text{s. a} \quad & \varphi(Z_{ji} - \left\|x^j - a^i\right\|) + \varphi(Z_{ji} - D_{ji}) \geq \epsilon, \;\; i \in N_j. \end{aligned} \tag{4.9}$$

para $\epsilon > 0$.

Uma vez que a região viável do problema (4.8) é o limite da região viável de (4.9) quando $\epsilon \to 0_+$, pode-se então solucionar o problema (4.8) resolvendo uma sequência de problemas da forma (4.9) gerada por uma sequência de valores de ϵ que tende a zero.

4.2 Suavização do problema

Analisando o problema (4.9), vê-se que a definição da função φ envolve uma estrutura rígida e não-diferenciável, o que torna a sua solução computacionalmente muito complexa. Em vista disso, o método numérico que será adotado para resolver as restrições do problema (4.3) faz uso de uma estratégia de suavização [Xavier, 1993]. Dessa perspectiva, define-se a função

$$\phi(y, \tau) = (y + \sqrt{y^2 + \tau^2})/2; \tag{4.10}$$

para $y \in \mathbb{R}$ e $\tau > 0$.

A função ϕ tem as seguintes propriedades:

(a) $\phi(y, \tau) > \varphi(y), \;\; \forall \tau > 0$;

(b) $\lim_{\tau \to 0} \phi(y, \tau) = \varphi(y)$;

(c) $\phi(\cdot, \tau)$ é uma função C^∞ crescente e convexa.

A função ϕ é uma boa aproximação da função φ. O parâmetro τ indica o nível da aproximação, pois à medida que τ tende 0, a função suavizadora ϕ se aproxima da função original φ. Adotando-se as mesmas hipóteses da Figura 16, as parcelas das

restrições do problema (4.9) e suas correspondentes aproximações suavizadas, dadas por (4.10), são exibidas na Figura 17.

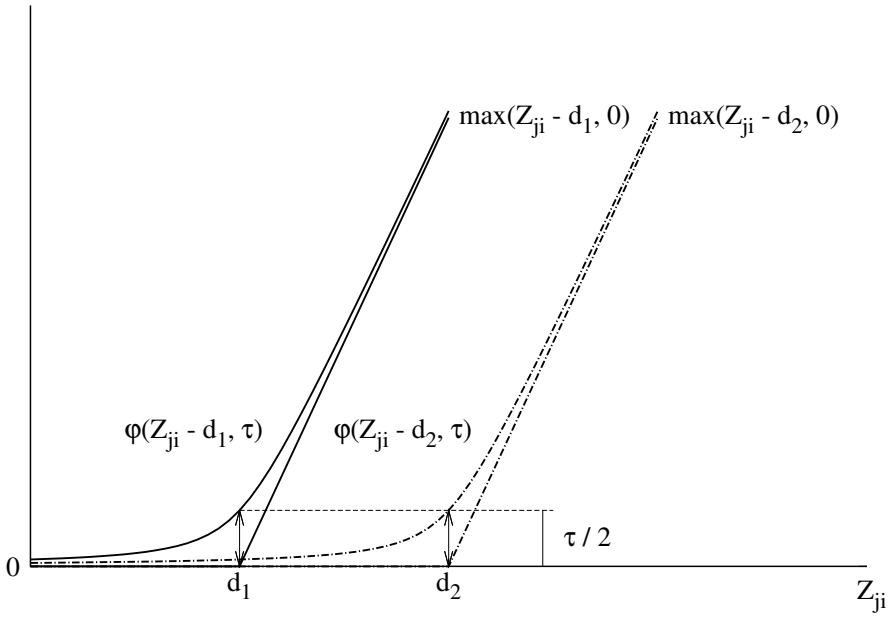

Figura 17: O valor das parcelas das restrições do problema (4.9) e suas respectivas aproximações suavizadas dadas pela função ϕ.

Substituindo a função φ por ϕ no problema (4.9) obtém-se

$$\begin{aligned} \min \quad & \sum_{i \in N_j} Z_{ji} \\ \text{s. a} \quad & \phi(Z_{ji} - \left\|x^j - a^i\right\|, \tau) + \phi(Z_{ji} - D_{ji}, \tau) \geq \epsilon, \quad i \in N_j. \end{aligned} \tag{4.11}$$

para $\epsilon > 0$.

Para se obter um problema completamente diferenciável é necessário, ademais, suavizar a função distância euclidiana $\|x^j - a^i\|$. Com este propósito, define-se a função

$$\theta(x^j, a^i, \gamma) = \sqrt{\|x^j - a^i\|_2^2 + \gamma^2}, \quad \text{para } \gamma > 0.$$

A função θ tem as seguintes propriedades:

(a) $\lim_{\gamma \to 0} \theta(x^j, a^i, \gamma) = \left\|x^j - a^i\right\|$;

(b) θ é uma função C^∞.

A função $\theta(x^j, a^i, \gamma)$ é uma boa aproximação para a distância euclidiana $\|x^j - a^i\|$. O parâmetro γ indica a qualidade da aproximação. Na medida em que γ se aproxima

de 0, a função θ tende para o valor exato da distância, conforme visualizado na figura 18.

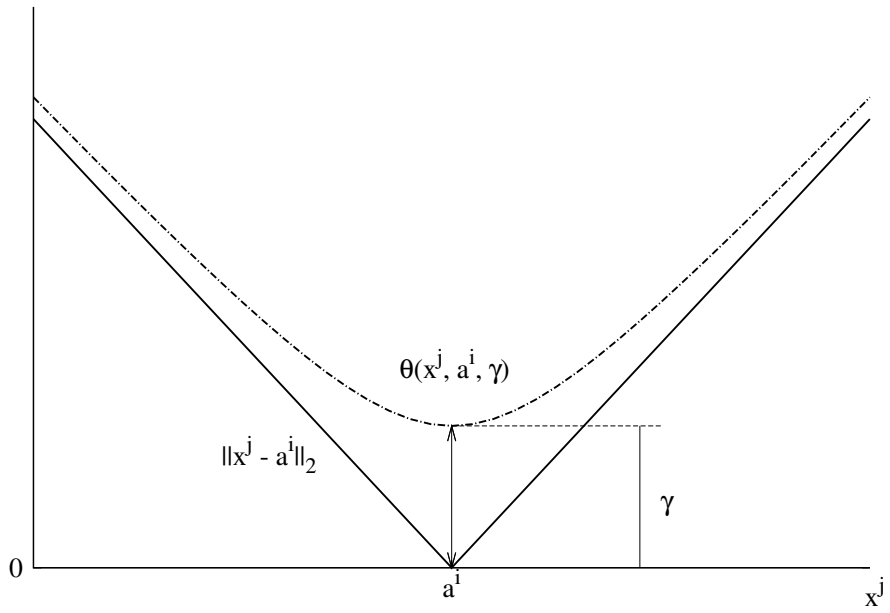

Figura 18: Distância euclidiana entre os pontos x^j e a^i e sua suavização dada pela função $\theta(x^j, a^i, \gamma)$.

Substituindo a distância $\|x^j - a^i\|$ pela função θ no problema (4.11), obtém-se o problema diferenciável

$$\begin{aligned} \min \quad & \sum_{i \in N_j} Z_{ji} \\ \text{s. a} \quad & \phi(Z_{ji} - \theta(x^j, a^i, \gamma), \tau) + \phi(Z_{ji} - D_{ji}, \tau) \geq \epsilon, \;\; i \in N_j. \end{aligned} \tag{4.12}$$

As propriedades das funções ϕ e θ permitem encontrar a solução do problema (4.8) resolvendo uma sequência de subproblemas da forma (4.12) produzida pelo decréscimo dos parâmetros $\gamma,\ \tau,\ \epsilon$.

Dado um centróide x^j, a propriedade (c) da função hiperbólica ϕ implica que as restrições do problema (4.12) serão ativas. Portanto, o problema (4.12) é equivalente a

$$\begin{aligned} \min \quad & \sum_{i \in N_j} Z_{ji} \\ \text{s. a} \quad & h_{ji}(Z_{ji}, x^j) = \phi(Z_{ji} - \theta(x^j, a^i, \gamma), \tau) + \phi(Z_{ji} - D_{ji}, \tau) - \epsilon = 0, \;\; i \in N_j. \end{aligned} \tag{4.13}$$

A dimensão do espaço das variáveis do problema (4.13) é $|N_j| + n$. Como em geral, o número de observações dado por $|N_j|$ é grande, o problema (4.13) tem um grande número

de variáveis. Todavia, ele possui uma estrutura separável, pois cada variável Z_{ji} aparece em uma única restrição de igualdade. Além disso, a derivada parcial de $h_{ji}(Z_{ji}, x^j)$ com respeito a Z_{ji} é estritamente positiva, pois $h_{ji}(Z_{ji}, x^j)$ é crescente com respeito a Z_{ji} (vide Figura 17). Portanto, é possível usar o Teorema da Função Implícita para calcular cada componente Z_{ji} como função do centróide x^j.

Dessa forma, o problema irrestrito

$$\min f(x) = \sum_{i \in N_j} Z_{ji}(x^j). \tag{4.14}$$

é obtido, onde cada $Z_{ji}(x^j)$ resulta do cálculo de uma raiz de cada equação

$$h_{ji}(Z_{ji}, x^j) = \phi(Z_{ji} - \theta(x^j, a^i, \gamma), \tau) + \phi(Z_{ji} - D_{ji}, \tau) - \epsilon = 0, \quad i \in N_j. \tag{4.15}$$

Devido a propriedade (a) da função hiperbólica ϕ, as duas primeiras parcelas da equação acima são estritamente crescentes com respeito a variável Z_{ji} e, portanto, a equação acima possui raiz única.

Novamente, devido ao Teorema da Função Implícita as funções $Z_{ji}(x^j)$ têm todas as derivadas com respeito a variável x^j. Assim, é possível calcular o gradiente da função objetivo do problema (4.14) da seguinte forma:

$$\nabla f(x^j) = \sum_{i \in N_j} \nabla Z_{ji}(x^j), \tag{4.16}$$

onde

$$\nabla Z_{ji}(x^j) = -\nabla h_{ji}(Z_{ji}) / \frac{\partial h_{ji}(Z_{ji}, x^j}{\partial Z_{ji}}. \tag{4.17}$$

A metodologia acima emprega as mesmas idéias de [Abadie and Carpenter, 1969] utilizadas no desenvolvimento do algoritmo do gradiente reduzido dedicado à resolução do problema geral de programação não-linear sujeito a restrições de igualdade. Dessa forma, é fácil resolver o problema (4.14) utilizando qualquer método baseado em informações da derivada de primeira ordem. É necessário enfatizar que o problema (4.13) é definido em um espaço n-dimensional, ou seja, trata-se de um problema de pequeno porte.

Em suma, pelo uso da suavização hiperbólica, conforme acima apresentado para $j = 1, 2,$ deve ser resolvida uma sequência de problemas da forma abaixo, obtida pela manipulação conveniente dos parâmetros τ, ϵ, γ:

$$\min \quad \sum_{i \in N_j} Z_{ji} \tag{4.18}$$

$$\text{s. a} \qquad h_{ji}(Z_{ji}, x^j) = \phi(Z_{ji} - \theta(x^j, a^i, \gamma), \tau) + \phi(Z_{ji} - D_{ji}, \tau) - \epsilon = 0, \;\; i \in N_j,$$

onde

$$\phi(y, \tau) = (y + \sqrt{y^2 + \tau^2})/2, \tag{4.19}$$

$$\theta(x^j, a^i, \gamma) = \sqrt{\|x^j - a^i\|_2^2 + \gamma^2}. \tag{4.20}$$

Ademais, esse problema pode ser resolvido pela forma irrestrita (4.14).

Agora, será feita uma digressão sobre os efeitos adicionais engendrados pelo uso da suavização. Para efeito de uma ilustração bem singela, será considerado um problema com unicamente três pontos num espaço de uma única dimensão.

Na Figura 19, são representadas as parcelas $Z_{j1},\ Z_{j2},\ Z_{j3}$ da função objetivo do problema original como função de x^j. A Figura 20 é a representação suavizada do mesmo caso representado pela Figura 19. As parcelas suavizadas representadas na Figura 20 foram geradas a partir das soluções $Z_{j1},\ Z_{j2},\ Z_{j3}$ das restrições de igualdade do problema suavizado (4.12) em função de x^j e um conjunto de três valores pré-fixados dos parâmetros τ, γ, ϵ.

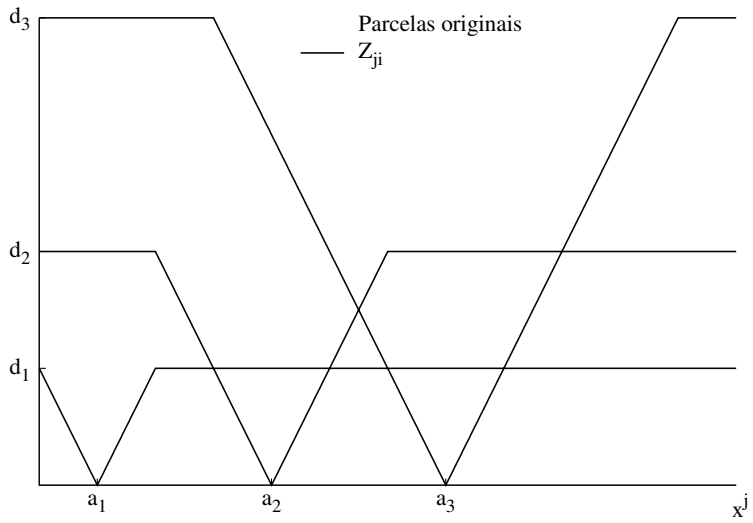

Figura 19: Parcelas originais Z_{ji} do problema (4.3).

Na Figura 21, foram representadas a soma das três parcelas da função objetivo, bem como das respectivas funções suavizadas. Observa-se na Figura 21, a redução do número de mínimos locais. Em particular, os mínimos locais existentes em a_1 e a_2 na função original são eliminados na função objetivo suavizada. Isso mostra o poder convexificador da proposta de suavização hiperbólica. Constata-se que a soma das parcelas suavizadas resultou numa função unimodal com mínimo global próximo ao da função original. Em-

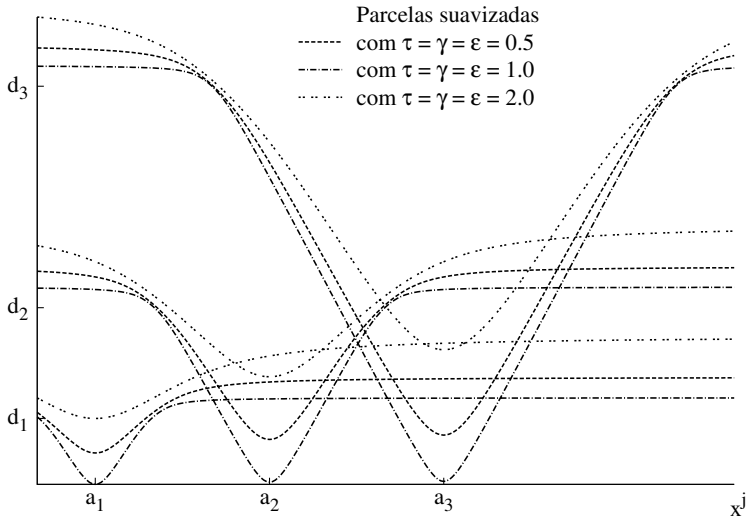

Figura 20: Parcelas do problema suavizado (4.12) com diferentes valores dos parâmetros τ, γ, ϵ.

bora esse fato seja extremamente promissor, não se pode garantir que, para qualquer caso, o mesmo torne a ocorrer.

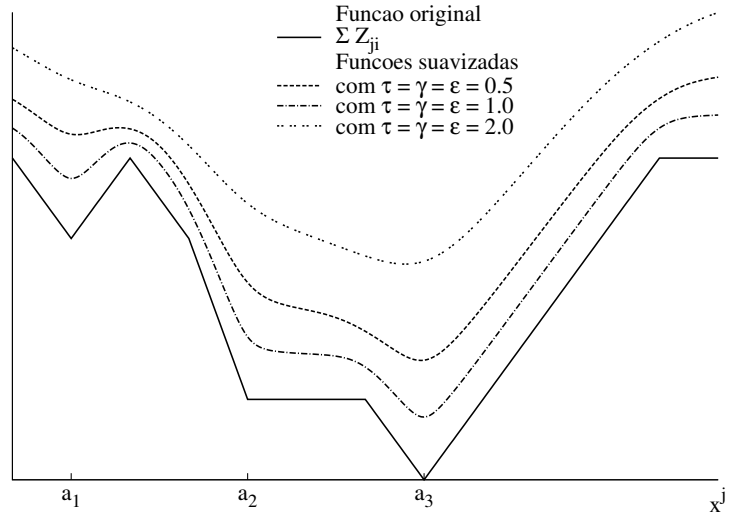

Figura 21: Função objetivo original Z_{ji} do problema (4.3) e funções objetivo do problema suavizado (4.12) com diferentes valores dos parâmetros τ, γ, ϵ.

A transformação do problema original não-diferenciável e fortemente não-convexo em um problema diferenciável e menos não-convexo, i.e., com um menor número mínimos locais é uma das principais vantagens da suavização hiperbólica. As Figuras 20 e 21 ilustram essa afirmação no caso em que $x^j \in \mathbb{R}$ e $i \in \{1, 2, 3\}$.

Assim, de acordo com a apresentação acima o processo de solução do problema (4.18) será implementado através do seguinte algoritmo geral:

Algoritmo SH (suavização hiperbólica)

Inicialização

Escolha ponto inicial x^0 para parâmetros $\gamma^1, \tau^1, \epsilon^1$

Escolha valores de redução ρ_j, $j = 1, 2, 3$ no intervalo aberto $(0, 1)$.

Faça $k = 1$

Passo Principal

Enquanto algum critério de parada não for satisfeito, repita:

Resolva para $j = 1, 2$ o problema (4.18) com

$$\gamma = \gamma^k,$$
$$\tau = \tau^k,$$
$$\epsilon = \epsilon^k,$$

tomando como ponto inicial x^{k-1} e obtendo como x^k como solução.

Faça

$$\gamma^{k+1} = \rho_1 \gamma^k,$$
$$\tau^{k+1} = \rho_2 \tau^k,$$
$$\epsilon^{k+1} = \rho_3 \epsilon^k,$$
$$k = k + 1.$$

Como em outros métodos de suavização, a solução para o problema (4.18) será obtida através da resolução de uma seqüência de subproblemas de minimização irrestritos ($k = 1, 2, \ldots, m$) conforme indicado no Algoritmo SH.

Note que o Algoritmo SH faz τ e γ se aproximarem de zero, logo as funções ϕ e θ presentes nas restrições do problema (4.18) aproximam-se das funções φ e $\|\cdot\|$ respectivamente. Adicionalmente, faz ϵ se aproximar de zero, portanto, em um movimento simultâneo, o problema resolvido se aproxima gradativamente do problema original. É implicitamente assumido que o algoritmo encontra uma solução de mínimo global, x^k, do subproblema suavizado ou, pelo menos, um mínimo local profundo, ou seja, com valor muito baixo.

Sob essa hipótese e devido às propriedades de continuidade de todas as funções envolvidas, a seqüência $x^1, x^2, \ldots$ de valores ótimos deve tender igualmente a um valor ótimo profundo do problema original.

5 *Experimentos computacionais*

Neste capítulo, serão apresentados os resultados de experimentos computacionais realizados com o intuito de avaliar e validar proposta de solução deste trabalho. Foram primeiramente utilizadas cinco bases de dados sendo duas delas sintéticas e as demais reais.

As bases de dados reais utilizadas nos experimentos utilizando o método aqui proposto fazem parte do repositório *UCI - Machine Learning Repository* [Asuncion and Newman, 2007]. Dentre as bases de dados disponíveis nesse repositório, foram selecionadas a base de dados australiana de aprovação de crédito (Australian Credit Approval), a base de dados de diagnóstico de câncer de mama do estado de Wisconsin (Breast Cancer Wisconsin - Diagnostic) e a base de dados de doenças do fígado (Liver Disorders Data Set), aqui referidas, respectivamente, por BDCA, BDCW e BDDF.

Para implementar o algoritmo de solução, foi escrito um código fonte em FORTRAN 77 e compilado no compilador Compaq Visual Fortran v6.1. Os experimentos computacionais foram realizados em um computador com sistema operacional Windows XPSP2, processador Intel Pentiun Dual Core 2 Duo E-6320 - 1,86GHz e 2GB de memória RAM. Os procedimentos de minimização incondicional foram efetuados através do método Quase-Newton com atualização da matriz hessiana dada pela fórmula BFGS, obtida na Harvell Subroutine Library (HSL - *http://hsl.rl.ac.uk/archive/hslarchive/hslarchive.html*).

5.1 Experimentos computacionais com bases de dados sintéticas

As duas bases de dados sintéticas ART1 e ART2 foram criadas objetivando tão somente um teste inicial do algoritmo, em que fosse possível um acompanhamento visual e lógico dos procedimentos do mesmo. Não é feita qualquer comparação de desempenho, inclusive pela inexistência de quaisquer outros resultados. Os testes foram realizados

adotando-se como único critério de parada o número máximo de iterações do algoritmo.

O método proposto é recursivo e em cada iteração cria dois novos grupos, um para cada classe, e determina seus respectivos centros. Em seguida, cada uma das observações da base de dados é associada ao grupo de cujo centro está mais próxima. Após essa etapa, ocorre um refinamento na posição dos últimos centros gerados, levando-se em conta apenas as observações corretamente classificadas. Mesmo após o refinamento dos centros, podem ocorrer erros em que uma observação pertencente a uma dada classe é alocada, de forma equivocada, em um agrupamento pertencente à outra classe. A despeito dos esforços do algoritmo ao longo das iterações para redução desses erros, tendência empiricamente constatada, não há garantias de que eles sejam completamente eliminados até o final do processo.

Observa-se ainda que, devido às características do algoritmo, pode ocorrer em certos casos um ligeiro acréscimo do valor da função objetivo entre duas iterações consecutivas, mas se mantém a tendência de decréscimo do valor da função objetivo ao longo das iterações na quase totalidade das iterações desses experimentos.

5.1.1 Base de dados sintética ART1

A base de dados sintética ART1 é formada por 148 observações divididas igualmente entre duas classes. As observações de ART1 possuem dois atributos. Nesse experimento, as classes são relativamente simétricas e formadas por subgrupos completamente separados, ou seja, bem definidos (ver Figura 22). Na geração da solução, o número máximo de iterações foi arbitrariamente fixado em quatro.

Na Tabela 1, a coluna K corresponde ao número da iteração, as colunas EA1 e EA2 correspondem ao percentual de observações mal classificadas das classes A_1 e A_2 respectivamente, a coluna EA indica o percentual total de observações mal classificadas, as colunas FO1 e FO2 correspondem respectivamente ao valor da função objetivo nas classes A_1 e A_2 e as colunas TP1 e TP2 correspondem aos tempos de processamento em segundos consumidos pelo algoritmo em cada uma das classes por iteração.

Observa-se claramente na Tabela 1 a tendência de redução tanto do número de observações mal classificadas quanto do valor da função objetivo a cada iteração em ambas as classes. Nota-se também que os tempos de processamento são baixos e relativamente próximos entre iterações. A proximidade do tempo de processamento nas diferentes iterações é um comportamento esperado, já que a cada iteração é resolvido um problema

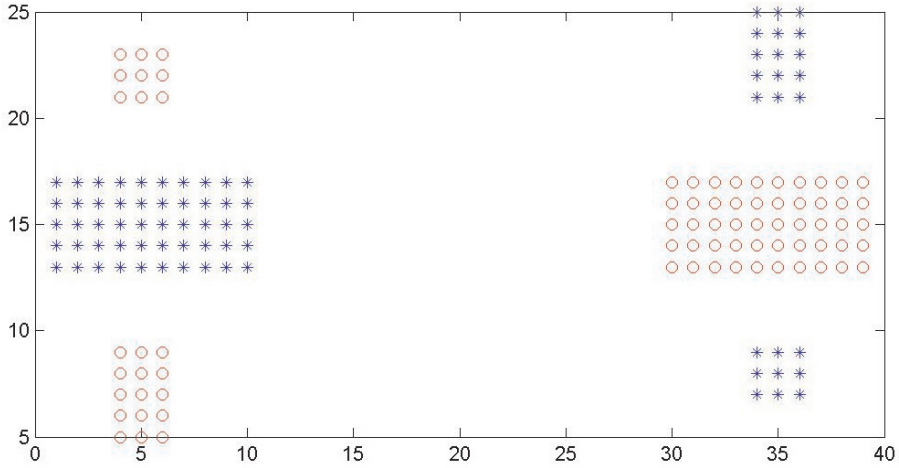

Figura 22: Base de dados ART1 - as observações pertencentes à classe A_1 são representadas em azul pelo símbolo $(*)$ e as pertencentes à classe A_2, em vermelho pelo símbolo $(\circ)$.

de otimização ligeiramente diferente do resolvido na iteração anterior, mas exatamente do mesmo porte.

K	EA1	EA2	EA	FO1	FO2	TP1	TP2
1	32,4%	32,4%	32,4%	324,95	324,95	0,01	0,04
2	12,2%	12,2%	12,2%	302,67	302,67	0,01	0,02
3	12,2%	12,2%	12,2%	151,48	151,48	0,03	0,03
4	0,0%	0,0%	0,0%	137,19	137,19	0,01	0,02

Tabela 1: Base de dados ART1: erros percentuais, valores da função objetivo e os tempos de processamento em segundos em cada classe por iteração.

As Figuras 23-26 ilustram o comportamento do algoritmo proposto em cada uma das quatro iterações.

Na primeira iteração do algoritmo proposto sempre gera-se dois grupos iniciais. Nesse caso, os dois grupos possuem observações erradamente vinculadas, mas observa-se que a maioria das observações já está corretamente classificada (ver Figura 23).

Na segunda iteração, são formados dois novos centros. Os grupos associados a esses centros são formados apenas por observações corretamente classificadas (ver Figura 24).

Na terceira iteração (Figura 25), devido aos ditames da minimização da função objetivo, valeu a pena dividir os grupos mais numerosos ao invés de criar dois novos grupos em cima dos subconjuntos de observações mal classificadas.

O resultado final, a classificação sem erros de todas as observações da base de dados, é exibido na Figura 26.

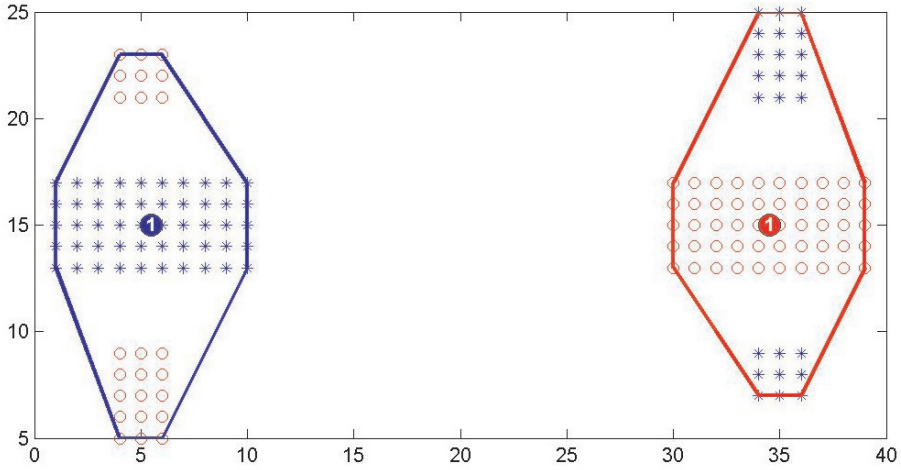

Figura 23: Base de dados ART1, na primeira iteração são formados dois grupos iniciais.

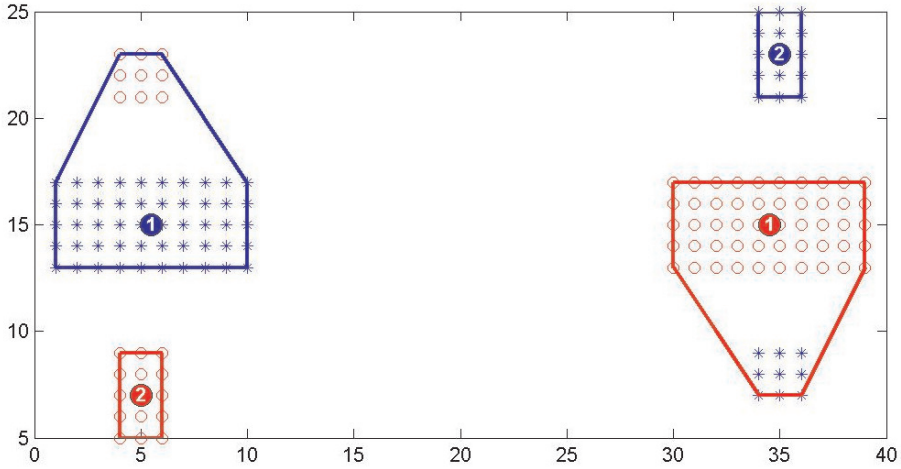

Figura 24: Base de dados ART1, na segunda iteração os novos grupos formados possuem apenas classificações corretas.

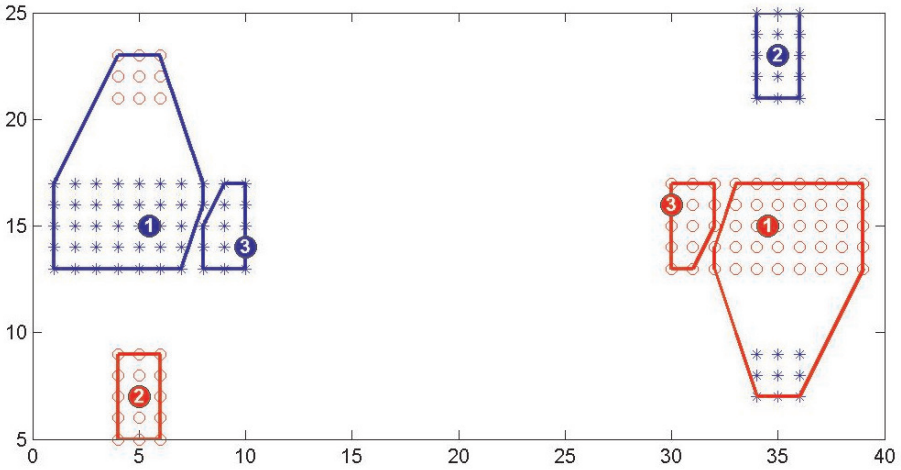

Figura 25: Base de dados ART1, na terceira iteração o algoritmo dividiu os grupos mais numerosos.

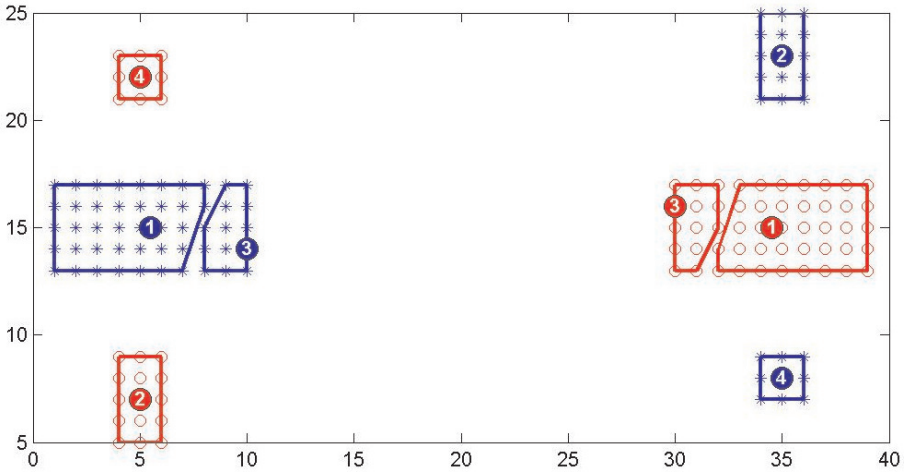

Figura 26: Base de dados ART1, configuração final com todas as observações corretamente classificadas.

Cabe observar que a solução obtida captura a simetria da distribuição das classes o que reflete a racionalidade implícita nos procedimentos do algoritmo.

5.1.2 Base de dados sintética ART2

A base de dados ART2 é formada por 1000 observações dividas igualmente entre duas classes. As observação da base ART2 possuem somente dois atributos. Aqui, as classes são distribuídas de forma quase que contínua, não ocorrendo a formação de subgrupos bem definidos no interior de cada classe. Além disso, a classe A_1 está distribuída ao redor da classe A_2 de tal forma que a interseção de seu fecho convexo com a classe A_2 não é vazia (ver Figura 27). Por isso, esse problema teste apresenta dificuldades de resolução a despeito da sua separabilidade visualmente evidente.

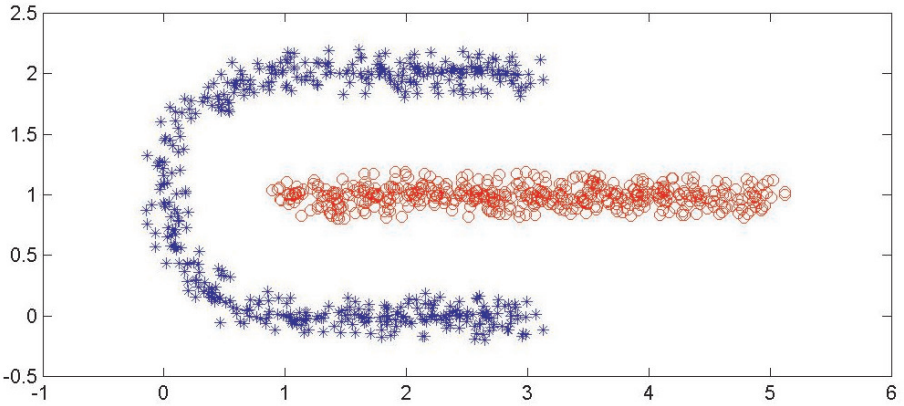

Figura 27: Base de dados ART2 - as observações pertencentes à classe A_1 são representadas em vermelho pelo símbolo $(*)$ e as pertencentes à classe A_2, em vermelho pelo símbolo $(\circ)$.

Na Tabela 2, a coluna K corresponde ao número da iteração, as colunas EA1 e EA2 correspondem ao percentual de observações mal classificadas das classes A_1 e A_2 respectivamente, EA indica o percentual total de observações mal classificadas, as colunas FO1 e FO2 correspondem respectivamente ao valor da função objetivo nas classes A_1 e A_2 e as colunas TP1 e TP2 correspondem aos tempos de processamento em segundos consumidos pelo algoritmo em cada uma das classes por iteração. Nesse experimento o número máximo de iterações foi arbitrariamente fixado em sete.

Novamente, observa-se claramente a tendência de redução do percentual de erros a cada iteração e que os tempos de processamento são baixos e relativamente próximos.

K	EA1	EA2	EA	FO1	FO2	TP1	TP2
1	25,0%	33,0%	29,0%	529,59	261,03	0,13	0,14
2	23,6%	31,8%	27,7%	434,28	194,35	0,14	0,13
3	10,6%	31,8%	21,2%	339,96	140,47	0,15	0,16
4	0,8%	23,2%	12,0%	282,88	121,18	0,15	0,18
5	0,8%	18,0%	9,4%	227,69	104,04	0,15	0,19
6	0,8%	18,0%	9,4%	200,02	88,04	0,16	0,15
7	0,8%	13,2%	7,0%	179,46	81,35	0,16	0,23

Tabela 2: Base de dados ART2: erros percentuais, valores da função objetivo e os tempos de processamento em segundos em cada classe por iteração.

As Figuras 28-34 ilustram a evolução do algoritmo ao longo das sete iterações.

A primeira iteração do algoritmo proposto sempre gera dois grupos iniciais (ver Figura 28). Nesse caso, devido à distribuição das observações da classe A_1 e aos critérios definidos pela função objetivo, o algoritmo posicionou o primeiro centro da classe A_1 sobre observações da classe A_2.

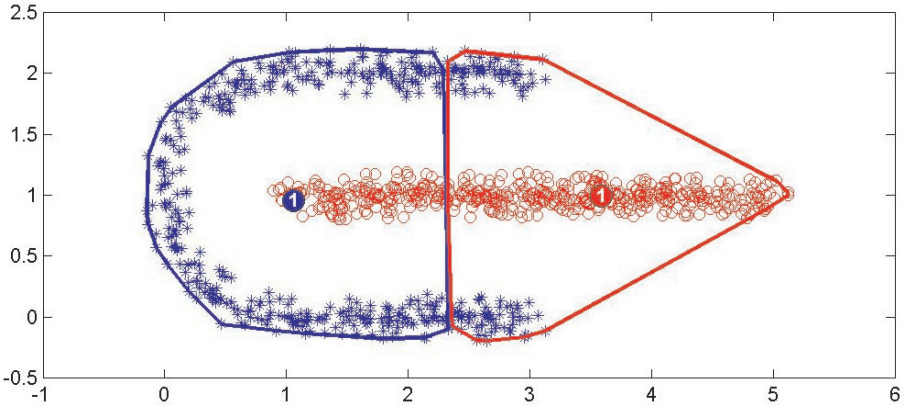

Figura 28: Base de dados ART2, os primeiros centros gerados pelo algoritmo.

Na segunda iteração, o algoritmo posicionou dois novos centros para as classes, me-

lhorando a classificação segundo os critérios estabelecidos pela função objetivo. Contudo o novo centro da classe A_1 também está localizado sobre observações da classe A_2 (ver Figura 29).

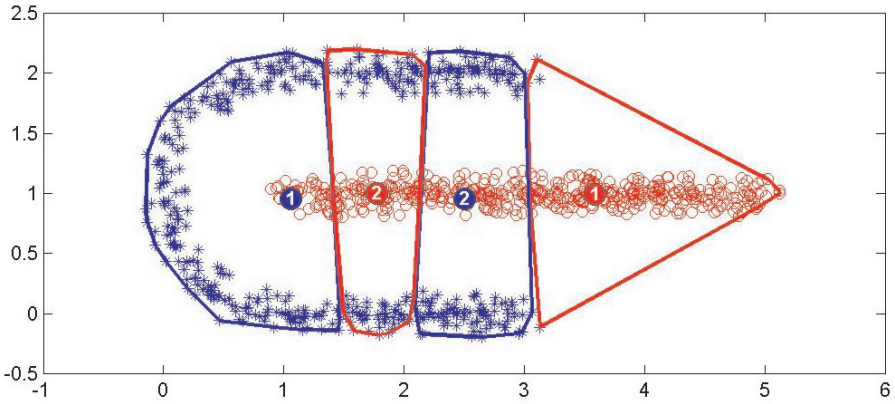

Figura 29: Base de dados ART2, segunda iteração: os grupos gerados e seus centros.

Da terceira à sexta iteração, o algoritmo gera para ambas as classes novos centros que refinam os grupos no interior de cada uma delas (ver Figuras 30-33).

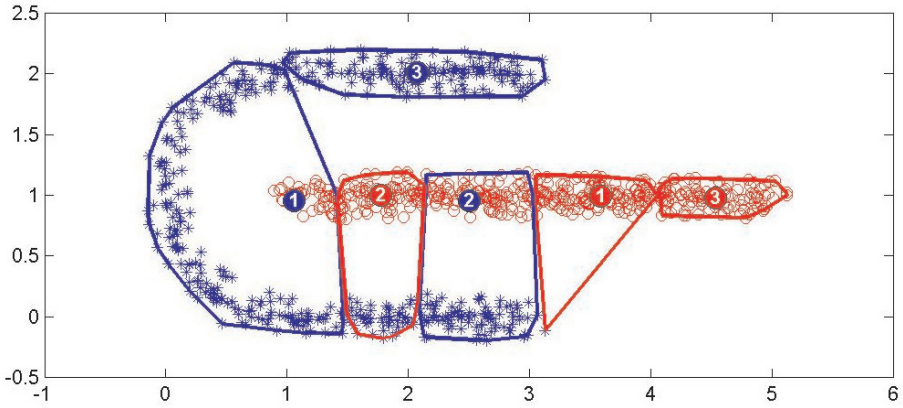

Figura 30: Base de dados ART2, terceira iteração.

A Figura 34, ilustra o resultado final alcançado pelo algoritmo com sete centros em cada classe. Os dois primeiros centros da classe A_1 (azul) estão erroneamente posicionados sobre observações da classe A_2 (vermelha). O grupo quatro da classe A_2 (vermelha) contém equivocadamente uma pequena parcela de pontos da classe A_1 (azul). Apesar desses comentários, constata-se que o resultado final obtido pelo algoritmo tem boa qualidade, uma vez que o percentual total de pontos equivocadamente classificados se situa em torno de apenas 7%.

Vê-se pelas figuras que a interseção do fecho convexo da classe A_1 com a classe A_2 ocasionou a definição equivocada dos dois primeiros centros da classe A_1. Assim, é possível

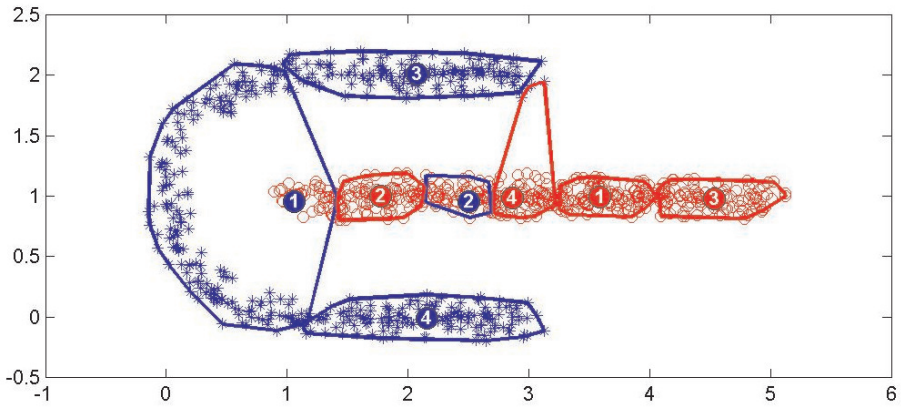

Figura 31: Base de dados ART2, quarta iteração.

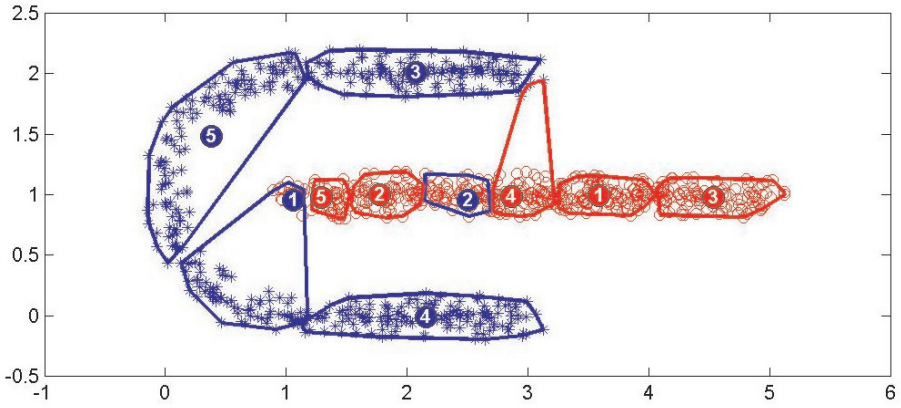

Figura 32: Base de dados ART2, quinta iteração.

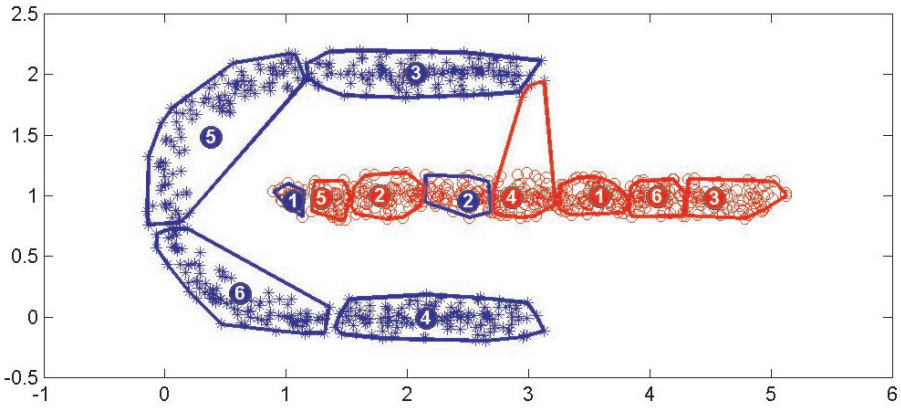

Figura 33: Base de dados ART2, sexta iteração.

ver uma certa fragilidade do algoritmo em lidar com superposições de fechos convexos das classes consideradas, sobretudo nas primeiras iterações. No entanto, a resultado final apresenta erro relativamente baixo em ambas as classes como é exibido na Figura 34. Vale ressaltar que os grupos definidos pelos dois primeiros centros da classe A_1 são formados apenas por observações da classe A_2, portanto uma alternativa para aumentar a eficiência

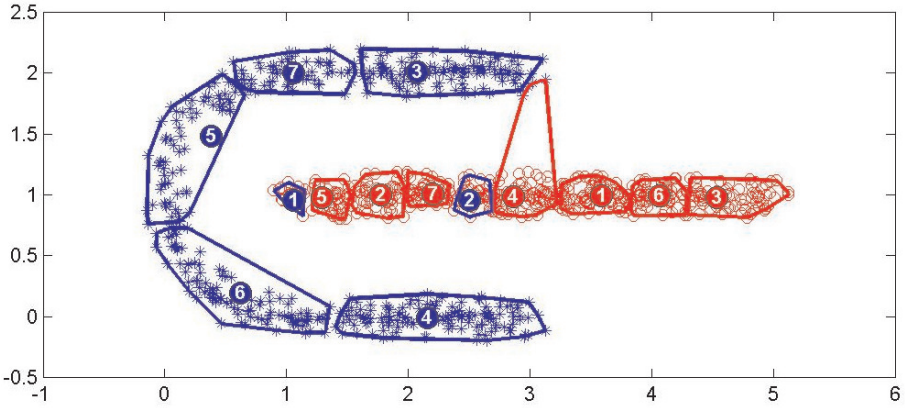

Figura 34: Base de dados ART2, configuração final.

do método é considerar a possibilidade de exclusão dos centros de grupos que ao final do algoritmo apresentem alto grau de impureza, i.e., um percentual elevado de observações equivocadamente classificadas.

5.2 Bases de dados reais

A seguir serão analisados os resultados obtidos com o algoritmo proposto para três bases de dados estudadas também por Bagirov em [Bagirov and Yearwood, 2003]. Especificamente, será analisado o desempenho do método proposto na classificação utilizando a base de dados australiana de aprovação de crédito (Australian Credit Approval), a base de dados de diagnóstico de câncer de mama do estado de Wisconsin (Breast Cancer Wisconsin - Diagnostic) e a base de dados de doenças do fígado (Liver Disorders Data Set), aqui referidas, respectivamente, por BDCA, BDCW e BDDF, todas contidas no repositório *UCI - Machine Learning Repository* [Asuncion and Newman, 2007].

Nos experimentos computacionais realizados com as bases de dados reais utilizou-se como critério de parada uma tolerância na variação da função objetivo entre duas iterações consecutivas. A parada do algoritmo ocorre quando o módulo da diferença de valores da função objetivo, em duas iterações sucessivas, dividido pelo valor inicial da função objetivo for inferior a uma dada tolerância, conforme definido no passo 7 do algoritmo (3.2). Em todos os experimentos foi adotada a tolerância $\epsilon = 10^{-2}$.

As bases de dados aqui analisadas têm características bem distintas das bases artificiais estudadas no item anterior. Em primeiro lugar, no caso das bases artificiais, cada observação tinha apenas dois atributos que assumiam valores contínuos e comparáveis, ao passo que, nas bases reais, as observações têm vários atributos, alguns contínuos com

valores bem díspares e outros discretos. Tal fato gera um problema de natureza prática que é a necessidade de harmonizar as diferentes escalas em que os atributos são expressos. Visando a comparabilidade dos diferentes atributos, é usual adotar um procedimento de homogeneização dos dados.

Nesse estudo, foram aplicados três procedimentos de homogeneização encontrados no software MATLAB: *meanvar*, *softmax* e *minmax*. Denotando, respectivamente, por x_{old} , x_{new} , $\bar{x}$, σ_x , $\min(x)$ e $\max(x)$. Os valores original e homogeneizado do atributo, a média, o desvio padrão, o mínimo e o máximo do atributo considerando todas as observações da base de dados, esses procedimentos de homogeneização são definidos da seguinte forma:

a) meanvar

$$x_{new} = \frac{x_{old} - \bar{x}}{\sigma_x}, \tag{5.1}$$

b) softmax

$$x_{new} = \left(1 + \exp\left(\frac{\bar{x} - x_{old}}{\sigma_x}\right)\right)^{-1}, \tag{5.2}$$

c) minmax

$$x_{new} = \frac{x - \min(x)}{\max(x) - \min(x)}. \tag{5.3}$$

Para cada base de dados, foram realizados testes de comparação entre os três métodos de homogeneização visando escolher em cada caso o mais adequado. Para os testes com as bases de dados BDCA, BDCW e BDDF, as homogeneizações que produziram os melhores resultados foram, respectivamente, *meanvar*, *softmax* e *minimax*(ver Tabela 3). O critério de julgamento utilizado baseou-se no erro percentual cometido pelo método de classificação proposto após a aplicação de cada um dos procedimentos de homogeneização.

BD	none	meanvar	softmax	minmax
BDCA	35,1%	13,6%	14,9%	19,1%
BDCW	10,5%	5,3%	3,2%	3,7%
BDDF	44,3%	37,7%	35,7%	33,0%

Tabela 3: Erros percentuais obtidos pelo método de classificação proposto com as diferentes estratégias de homogeneização. A coluna *none* apresenta os erros percentuais obtidos quando os dados não sofrem qualquer homogeneização.

Além disso, em foi adotada a estratégia de validação dos resultados denominada *ten-folder*. Esta estratégia cíclica é dividida em duas etapas: *treinamento* e *teste*. Inicialmente, os dados são aleatoriamente divididos em dez subconjuntos com cardinalidades aproximadamente iguais. Em seguida, inicia-se a fase de treinamento, onde obtém-se uma solução

(centros das classes) para o problema de classificação utilizando nove dos dez subconjuntos. Na fase de teste, o subconjunto não utilizado é classificado com base na solução obtida na fase de treinamento. Esse processo é realizado dez vezes, deixando-se, em cada caso, outro dos dez subconjuntos de fora. Ao final considera-se a média dos resultados obtidos na fase de teste para julgar a eficiência do método.

Os resultados obtidos por três diferentes métodos apresentados em [Bagirov and Yearwood, 2003] serão utilizados para comparação. Esses métodos serão denotados por BAG1, BAG2 e BAG3 e são aplicados para resolução do problema de otimização do passo 5 do algoritmo (3.2). O método BAG1 utiliza um algoritmo de otimização local baseado em aproximações contínuas do subdiferencial de Clarke, o método BAG2 aplica um algoritmo de otimização local do problema baseado em aproximações contínuas do quasidiferencial de Demyanov-Rubinov e o método BAG3 utiliza um algoritmo de otimização global.

Os resultados dos experimentos de Bagirov foram gerados utilizando-se códigos escritos em FORTRAN 90 que foram processados em um PC Pentium-S com CPU de 150 MHz.

Conforme assinalado no artigo [Bagirov and Yearwood, 2003], os métodos BAG1, BAG2 e BAG3 foram aplicados após uma estratégia de homogeneização e seleção de atributos que permitiu uma considerável redução do número de atributos dos problemas. Assim, foi possível obter uma redução no número de atributos de 14 para 3 na base BDCA e de 30 para 3 na base BDCW. Não houve redução do número de atributos da base de dados BDDF. Em todos os experimentos numéricos apresentados em [Bagirov and Yearwood, 2003], o critério de parada definido pela variação relativa do valor da função objetivo estabelecido pelo parâmetros $\epsilon = 10^{-2}$. Esses autores também laçam mão do método *ten-folder* para validação dos procedimentos.

Os erros encontrados utilizando a formulação aqui proposta são relativamente próximos aos obtidos por Bagirov, embora o número de grupos seja bem diferente. Isso deve-se a uma série de fatores, entre os quais destacam-se: as diferentes estratégias de homogeneização de dados, particularidades de implementação, número de grupos, e, principalmente, o fato de não ser utilizado nenhum procedimento de seleção de atributos nos experimentos computacionais realizados. Outro aspecto relevante é que os problemas tratados apresentam diversos mínimos locais, podendo a convergência de cada algoritmo específico ocorrer em um particular mínimo local diferente do obtido por outro método.

Ressalva-se que existe conjunto de fatores intervenientes que difere nos experimentos

computacionais, tais como: homogeneização, seleção de atributos e escolha do número de grupos. Tais diferenças limitam destarte o alcance das comparações aqui realizadas. A despeito dessas limitações são incluídas comparações para dar idéia mínima a respeito da performance relativa do método aqui proposto.

Observou-se que em todos os experimentos, os tempos de processamento obtidos com o método aqui proposto foram bastante inferiores aos obtidos por Bagirov podendo ser de 1,2% chegando no máximo a 30% destes. Além disso, os tempos foram obtidos sem a utilização de qualquer procedimento de seleção de atributos, o que, naturalmente, propiciaria a diminuição do tempo de processamento, na medida em que reduziria a dimensão dos problemas considerados.

Destaca-se a relevância desse desempenho como um indicativo da potencialidade de aplicação do método proposto para resolução de problemas concretos de classificação, problemas de grande porte, intrínsecas às aplicações consentâneas sobre grandes bancos de dados.

Convém ressaltar que nos experimentos realizados e aqui apresentados foi utilizado um equipamento de informática mais moderno que o utilizado por Bagirov. A velocidade extremamente maior dos experimentos aqui realizados em relação ao tempos obtidos por Bagirov em seus estudos se deve basicamente a dois fatores: a eficiência do método aqui proposto e a utilização de equipamentos mais modernos de informática. As experiências conduzidas em estudos equivalentes realizados por [Xavier, 1993] mostram que é inegável a redução do tempo de processamento obtida com a aplicação da suavização hiperbólica na solução de problemas de otimização. Assim, a natural velocidade de evolução dos equipamentos de informática não diminuem o mérito do presente trabalho.

Serão exibidos os resultados de dois experimentos computacionais utilizando cada uma das bases de dados reais. No primeiro experimento, o método aqui proposto foi utilizado para classificar toda a base de dados e foi registrado o percentual de acerto, o valor da função objetivo e o tempo de processamento por iteração em cada classe. No segundo experimento, foram comparados o erro médio, o tempo de processamento e número de grupos gerados pelos métodos de classificação BAG1, BAG2, BAG3 e o método aqui proposto no procedimento de validação *ten-folder*.

5.2.1 Base de dados BDCA

A base de dados BDCA (Australian Credit Approval) foi provida por J. R. Quinlan [Quinlan, 1993]. O propósito dessa base de dados é o de fornecer regras que decidam o risco de concessão de crédito. A BDCA é formada por 690 observações com 14 atributos dividas em duas classes. A primeira classe possui 307 observações e a segunda classe possui 383 observações. A base BDCA é interessante pois suas observações possuem tanto atributos contínuos quanto discretos. Além disso, os atributos discretos assumem 2, 3, 9 e até 14 valores distintos.

Na Tabela 4, exibe-se o desempenho do método proposto, denotado por MCSH (método de classificação com suavização hiperbólica), na classificação da base de dados BDCA em cada iteração. A coluna K corresponde às iterações, as colunas EA1 e EA2 correspondem ao percentual de observações equivocadamente classificadas em cada uma das classes, a coluna EA indica o percentual total de classificações equivocadas, as colunas FO1 e FO2 indicam o valor da função objetivo alcançado em cada uma das classes e, finalmente, as colunas TP1 e TP2 os tempos em segundos consumidos em cada iteração para geração dos centros de cada classe.

K	EA1	EA2	EA	FO1	FO2	TP1	TP2
1	21,2%	9,4%	14,6%	1.066,81	1.188,32	0,11	0,14
2	12,4%	13,3%	12,9%	1.037,18	1.166,41	0,11	0,14
3	14,3%	13,3%	13,8%	1.022,49	1.149,84	0,11	0,14
4	13,7%	13,3%	13,5%	1.008,59	1.136,87	0,09	0,13
5	13,0%	14,1%	13,6%	997,04	1.121,54	0,10	0,13
6	13,0%	14,1%	13,6%	997,04	1.114,63	0,12	0,13

Tabela 4: Resultados obtidos em cada iteração pelo método MCSH utilizando a base de dados BDCA.

Percebe-se na Tabela 4 a clara tendência de redução do valor da função objetivo (FO1 e FO2) em ambas as classes a cada iteração. No entanto, cabe ressaltar que a redução na valor da função objetivo não produz necessariamente redução no percentual de erro da classificação das observações (EA1 e EA2), ainda que esteja clara a conexão existente entre esses valores.

Na Tabela 5, são comparados os desempenhos dos métodos citados no procedimento de validação (*ten-folder*) de diferentes métodos de classificação utilizando a base de dados BDCA. As colunas TP, TE, NG, NORMA e N representam, respectivamente: o tempo médio em segundos de processamento na fase de treinamento, o erro percentual médio cometido na fase de teste, o número de grupos gerados, a norma utilizada e o

número de atributos considerados. São exibidos os resultados dos métodos BAG1, BAG2 e BAG3 anteriormente citados e também os resultados obtidos pelo método MCSH proposto. Destaca-se que o método MCSH convergiu muito mais rapidamente e apresentou o menor percentual de erro dentre todos os métodos citados que se baseiam em otimizações locais, perdendo apenas para o método BAG3, que se utiliza de otimizações globais, mas em compensação, com consumo de tempo computacional expressivamente maior.

Método	TP	ET	NG	Norma	N
BAG1	7,51	14,4%	2	2	3
BAG2	20,50	14,4%	2	2	3
BAG3	87,90	12,8%	3	2	3
MCSH	1,47	13,3%	6	2	14

Tabela 5: Resultados obtidos via *ten-folder* para a base de dados BDCA pelos diferentes métodos de classificação.

5.2.2 Base de dados BDCW

A base de dados BDCW (Breast Cancer Wisconsin - Diagnostic) foi criada por W. H. Wolberg e O. L. Magasarian, membros, respectivamente, do departamento de Cirurgia Geral e do departamento de Ciência da Computação da Universidade de Wisconsin. A base BDCW contém 2 classes, 569 pontos e 30 atributos. Todos os atributos da base BDCW são contínuos. Os valores dos atributos são gerados a partir de imagens digitalizadas de uma pequena amostra da mama e descrevem características dos núcleos das células presentes nessas imagens.

Na Tabela 6, exibe-se o desempenho do método proposto na classificação da base de dados BDCW em cada iteração. A coluna K corresponde às iterações, as colunas EA1 e EA2 correspondem ao percentual de observações equivocadamente classificadas em cada uma das classes, a coluna EA indica o percentual total de classificações equivocadas, as colunas FO1 e FO2 indicam o valor da função objetivo alcançado em cada uma das classes e, finalmente, as colunas TP1 e TP2 com os tempos em segundos consumidos em cada iteração para geração dos centros de cada classe.

Percebe-se na Tabela 6 que os resultados obtidos para a base de dados BDCW são bastante satisfatórios. O percentual de acerto alcançado foi de aproximadamente 97% na classificação das observações dessa base de dados.

Na Tabela 7, são comparados os desempenhos no procedimento de validação (*ten-folder*) de diferentes métodos de classificação utilizando a base de dados BDCW. As

K	EA1	EA2	EA	FO1	FO2	TP1	TP2
1	12,3%	3,1%	6,5%	179,83	253,54	0,19	0,30
2	7,5%	2,8%	4,6%	165,20	244,64	0,17	0,38
3	5,7%	3,4%	4,2%	156,50	227,83	0,17	0,34
4	6,1%	2,8%	4,0%	148,66	220,68	0,18	0,34
5	3,3%	5,3%	4,6%	142,44	216,08	0,20	0,41
6	3,3%	5,3%	4,6%	138,48	208,29	0,20	0,33
7	3,3%	3,9%	3,7%	136,57	202,68	0,22	0,35
8	3,8%	2,8%	3,2%	134,57	199,50	0,24	0,33
9	2,8%	3,4%	3,2%	132,50	197,64	0,23	0,38
10	2,8%	3,4%	3,2%	130,91	194,78	0,27	0,37
11	2,8%	3,4%	3,2%	129,25	190,95	0,24	0,40
12	2,8%	3,4%	3,2%	126,67	188,27	0,20	0,38
13	3,3%	3,1%	3,2%	124,92	186,22	0,22	0,53

Tabela 6: Resultados obtidos em cada iteração pelo método MCSH utilizando a base de dados BDCW.

colunas TP, ET, NG, NORMA e N representam respectivamente: o tempo médio em segundos de processamento na fase de treinamento, o erro percentual médio cometido na fase de teste, o número de grupos gerados, a norma utilizada e o número de atributos considerados.

Método	TP	ET	NG	NORMA	N
BAG1	27,10	3,2%	3	2	3
BAG2	125,33	1,6%	3	2	3
BAG3	235,41	1,4%	3	2	3
MCSH	7,47	3,7%	13	2	30

Tabela 7: Resultados obtidos via *ten-folder* para a base de dados BDCW pelos diferentes métodos de classificação.

A Tabela 7 mostra que os resultado obtido pelo método MCSH é coerente com os encontrados na literatura ainda que ligeiramente maior. É conveniente observar que o crescimento do número de atributos pode influenciar no erro total obtido. Por exemplo, em um problema com dois atributos pode se ter uma solução evidente, sem erros, bem conformada com classes disjuntas de informações, enquanto que, se for considerado um atributo adicional, as classes podem ser modificadas de tal que produzam erros na classificação.

O tempo de processamento foi consideravelmente menor que o obtido pelo algoritmo BAG1, o mais rápido dentre os apresentados em [Bagirov and Yearwood, 2003], equivalendo a menos de 1/3 deste. O número de grupos gerados pelo método MCSH é maior que o gerado pelos outros métodos. E, uma vez que a geração de um novo grupo está

condicionada à solução de um problema de otimização, ressalta-se que o método MCSH resolveu um número muito maior de problemas de otimização com um número N de atributos dez vezes maior. A despeito disso, consumiu um tempo total inferior ao dos demais métodos.

5.2.3 Base de dados BDDF

A base de dados BDDF (*Liver Disorders Data Set*) foi disponibilizada por Richard S. Forsyth (BUPA Medical Research Ltd). A base BDDF contém 2 classes, 345 observações e 6 atributos. As primeiras 5 variáveis são resultados de exames sanguíneos supostamente sensíveis a doenças do fígado advindas do consumo excessivo de álcool. As observações da base BDDF são formadas tanto por atributos contínuos quanto discretos.

Na Tabela 8, exibe-se o desempenho do método proposto na classificação da base de dados BDDF em cada iteração. A coluna K corresponde às iterações, as colunas EA1 e EA2 correspondem ao percentual de observações equivocadamente classificadas em cada uma das classes, a coluna EA indica o percentual total de classificações equivocadas, as colunas FO1 e FO2 indicam o valor da função objetivo alcançado em cada uma das classes e, finalmente, as colunas TP1 e TP2 os tempos em segundos consumidos em cada iteração para geração dos centros de cada classe.

K	EA1	EA2	EA	FO1	FO2	TP1	TP2
1	29,0%	54,5%	43,8%	38,93	59,73	0,05	0,09
2	43,4%	40,0%	41,4%	35,53	54,59	0,06	0,08
3	36,6%	49,0%	43,8%	33,70	50,91	0,06	0,08
4	35,2%	45,0%	40,9%	31,63	47,91	0,10	0,09
5	42,1%	42,5%	42,3%	30,44	45,16	0,06	0,08
6	41,4%	41,0%	41,2%	29,99	44,23	0,06	0,10
7	41,4%	37,0%	38,8%	29,79	43,64	0,07	0,08

Tabela 8: Resultados obtidos em cada iteração pelo método MCSH utilizando a base de dados BDDF.

O resultado final não é muito satisfatório, no entanto na Tabela 9 vê-se que esse baixo percentual de acerto acontece também ocorre com outros métodos de classificação. Essa concordância nos resultados sugere que as observações/atributos dessa base não fornecem informações suficientes para a caracterização das classes. Os resultados obtidos pelo método MCSH não foram tão precisos quanto os obtidos por Bagirov. No entanto, cabe ressaltar que os estudos de Bagirov envolveram transformações de escala nos atributos das informações dos problemas, que não foram claramente explicitadas em seu trabalho.

Esse fato por si só justifica os erros mais elevados obtidos pelo método MCSH. Assim, os erros mais baixos obtidos pelos outros métodos podem não ser decorrentes do mérito de tais métodos, mas, provavelmente, decorrem do fato de se trabalhar com informações mais organizadas, após o seu tratamento.

Novamente, o tempo computacional consumido pelo método MCSH foi consideravelmente menor que os dos demais métodos, mesmo tendo gerado quase o dobro de grupos.

Método	TP	ET	NG	NORMA	N
BAG1	47,19	30,9%	6	2	6
BAG2	103,71	29,4%	6	2	6
BAG3	142,26	27,4%	6	2	6
MCSH	1,07	39,8%	11	2	6

Tabela 9: Resultados obtidos via *ten-folder* para a base de dados BDDF pelos diferentes métodos de classificação.

5.3 Comentários

Os experimentos realizados com as bases de dados sintéticas demonstraram a racionalidade implícita do método proposto e também suas limitações. Destacam-se positivamente a velocidade de convergência do método e seu comportamento quase monótono tanto na redução do valor da função objetivo quanto do percentual de classificações equivocadas. Cabem críticas com respeito a sensibilidade do método à sobreposição do fecho convexo das classes, sobretudo nas primeiras iterações. Como dito anteriormente, esse contratempo pode ser dirimido excluindo-se os centros dos grupos com alto grau de impureza, i.e., com elevado número de observações mal classificadas e recalculando, a seguir, mais precisamente os novos centros.

Outra característica favorável é o caráter interno do método proposto, pois em sua definição não se faz qualquer hipótese sobre a distribuição das observações e, portanto, não requer transformações na base de dados. O método é intrinsecamente adaptativo, i.e., conforma-se naturalmente às características subjacentes à distribuição dos dados.

Nos experimentos com as bases reais, o método proposto apresentou alta velocidade de convergência e acurácia comparável a dos demais métodos apresentados. Portanto, fica clara a potencialidade de aplicação do método proposto na resolução de problemas de classificação em bases de dados de grande porte, consentâneas às aplicações que surgem na realidade atual.

6 Conclusões

Neste trabalho, baseamo-nos principalmente nos trabalhos [Jain and Dubes, 1988] e [Bagirov and Yearwood, 2003]. De início, em uma revisão bibliográfica, apresentou-se uma consolidação dos conceitos a respeito dos problemas de agrupamento e classificação.

No problema de agrupamento, dado um conjunto de informações para as quais são definidos um conjunto de atributos, busca-se agrupar essas informações em grupos, de modo que informações contidas num mesmo grupo sejam semelhantes, isto é, sejam caracterizadas por atributos similares, enquanto que informações em grupos diferentes devam ser as mais distintas possíveis.

O problema de classificação está associado a um conjunto de informações para as quais, além do conjunto de atributos, existe adicionalmente um atributo de pertinência a um conjunto finito de classes. A problemática intrínseca à classificação diz respeito a discriminar as observações segundo às classes, de modo que um ponto externo possa ser alocado a uma das classes da forma mais criteriosa possível.

Foram enumeradas diferentes técnicas de agrupamento e de classificação. Além disso, procurou-se trazer à baila o caráter flexível do conceito de similaridade que é o grande indutor.

Os problemas de agrupamento foram analisados sob o ponto de vista conceitual e, em seguida, foram analisados os algoritmos propostos por Bagirov para resolvê-los. Partindo de um algoritmo para resolução de problemas de agrupamento, Bagirov desenvolveu um método de classificação com bom desempenho computacional. No entanto, esse método apóia-se essencialmente na resolução de problemas de otimização não-diferenciável com diversos mínimos locais.

Propõe-se em nosso trabalho um método alternativo para resolução do problema de classificação que se baseia no método proposto por Bagirov. Por esse método alternativo, os problemas não-diferenciáveis formulados por Bagirov são substituídos por problemas equivalentes suavizados, nos quais as restrições dos problemas originais são representadas

de forma aproximada por funções hiperbólicas. Observa-se que essa estratégia tem como vantagem, não só a diferenciabilidade dos problemas considerados, que possibilita o uso de técnicas consagradas e eficientes de otimização, mas também, a redução da quantidade de mínimos locais. Em consequência, nosso algoritmo pode conduzir a soluções que são mínimos locais mais profundos e, portanto, de boa qualidade.

A fim de validar a metodologia proposta, foram realizados experimentos computacionais utilizando duas bases de dados sintéticas e três bases de dados da biblioteca *UCI - Machine learning repository*. Especificamente, foram consideradas a base de dados australiana de aprovação de crédito, a base de dados de diagnóstico de câncer de mama da cidade de Wisconsin e a base de dados de doenças do fígado. O algoritmo proposto foi implementado em linguagem FORTRAN 77. Os problemas de otimização considerados foram resolvidos utilizando o método Quase-Newton com atualização da aproximação da matriz hessiana dada pela fórmula BFGS. Especialistas interessados em utilizar o método aqui proposto poderão consultar o professor Adilson Elias Xavier através do email *adilson@cos.ufrj.br*.

No que tange a acurácia, o método proposto obteve desempenho comparável ao dos métodos apresentados por Bagirov em [Bagirov and Yearwood, 2003]. Com relação ao tempo de processamento, o método aqui proposto mostrou-se muito mais veloz, convergindo para soluções comparáveis a dos demais algoritmos considerados em tempo expressivamente menor. Portanto, os resultados obtidos são plenamente satisfatórios e, dada a velocidade de convergência, fica clara a possibilidade de aplicação do método proposto na resolução de problemas de classificação em bases de dados de grande porte, consentâneas às aplicações reais que concretamente surgem hoje em dia.

Os problemas relacionados a agrupamentos e classificação não se encerram na proposta desta tese. Diversas extensões deste trabalho estão abertas para novas investigações. Em particular, o desempenho do método aqui proposto pode ser aperfeiçoado aplicando-se, pelo menos, dois procedimentos: a exclusão de clusters que ao final do procedimento apresentem alto grau de impureza e a implementação de técnicas de redução do porte do problema, tanto através da seleção de observações quanto da seleção de atributos.

Conclui-se a partir das análises realizadas que os problemas de classificação de dados em bases com grande número de observações e atributos constituem-se ainda grandes desafios. Assim, há uma expectativa de que a pesquisa e a busca de algoritmos mais eficientes para resolução desses problemas devam ser uma preocupação permanente no futuro próximo.

Referências

[Aarts and Korst, 1989] Aarts, E. and Korst, J. (1989). Simulated annealing and boltzmann machines a stochastic approach to combinatorial optimization and neural computing. In *Wiley-Interscience series in discrete mathematics and optimization*. John Wiley and Sons.

[Abadie and Carpenter, 1969] Abadie, J. and Carpenter, J. (1969). *Generalization of the wolfe reduced gradient method to the case of nonlinear constraints*. Academic Press.

[Al-Sultan, 1995] Al-Sultan, K. S. (1995). A tabu search approach to the clustering problem. *Pattern Recognition*, 28:1443–1451.

[Al-Sultan and Khan, 1996] Al-Sultan, K. S. and Khan, M. M. (1996). Computational experience on four algorithms for the hard clustering problem. *Pattern Recognition*, 17(3):295–308.

[Anderberg, 1973] Anderberg, M. R. (1973). *Cluster analysis for applications*. Academic Press.

[Asuncion and Newman, 2007] Asuncion, A. and Newman, D. (2007). UCI machine learning repository.

[Backer, 1995] Backer, E. (1995). *Computer-assisted reasoning in cluster analysis*. Prentice-Hall.

[Baeza-Yates, 1992] Baeza-Yates, R. A. (1992). Introduction to data structures and algorithms related to information retrieval. In *Information retrieval data structures and algorithms*, pages 13–27. Prentice-Hall.

[Bagirov and Rubinov, 2001] Bagirov, A. M. and Rubinov, A. M. (2001). Modified versions of the cutting angle method. In *Advances in Convex Analysis and Global Optimization*. Kluver Academic Publishers.

[Bagirov et al., 2002] Bagirov, A. M., Rubinov, A. M., and Yearwood, J. (2002). A global optimization approach to classification. *Optimization and Engineering*, 22:65–74.

[Bagirov and Yearwood, 2003] Bagirov, A. M. and Yearwood, J. (2003). A new nonsmooth optimization algorithm for clustering problems. Technical report, University of Ballarat.

[Bajcsy, 1997] Bajcsy, P. (1997). *Hierarquical segmentation and clustering using similarity analysis*. PhD thesis, University of Illinois.

[Bock, 1970] Bock, H. H. (1970). Automatische klassification. In *Lecture Notes in Operations Research and Mathematical Systems*, 39. Springer Verlag.

[Bock, 1974] Bock, H. H. (1974). *Automatic Klassification*. Vasndenhoevk and Ruprecht.

[Bock, 1998] Bock, H. H. (1998). Clustering and neural networks. In *Advances in Data Science and Classification*. Springer Verlag.

[Bradley and Mangasarian, 2000] Bradley, P. S. and Mangasarian, O. L. (2000). Feature selection via concave minimization and support vector machines. In *Machine Learning Proceedings of the Fifteenth International Conference*, pages 82–90.

[Brown and Entail, 1992] Brown, D. E. and Entail, C. L. (1992). A practical application of simulated annealing to the clustering problem. *Pattern Recognition*, 25:401–412.

[Brown et al., 2000] Brown, M. G., Lin, D. C. W., Sugest, C. M., Furey, T., Ares, M., and Haussler, D. (2000). Knowledge based analysis of microarray gene expression data mining using support vector machines. In *Proceedings of National Academy of Sciences*, volume 97, pages 262–267.

[Carpenter and Grossberg, 1990] Carpenter, G. and Grossberg, S. (1990). Hierarquical search using chemical transmitters in self-organitazion pattern recognition architectures. *Neural Networks*, 3:129–152.

[Cheng, 1995] Cheng, Y. (1995). Mean shift, mode seeking and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(7):790–799.

[DeCoste and Scholkopf, 2002] DeCoste, D. and Scholkopf, S. (2002). Training invariant support vector machines. *Machine Learning*, 46:161–190.

[Deichsel, 1972] Deichsel, G. (1972). Verfahren der automatischen klassification durch cluster-analyse und anwendung bie morphologischen untersuchungan amoeben. Technical report, Universitart of Stuttgasrt.

[Diday, 1973] Diday, E. (1973). The dynamic cluster method in non-hierarchical clustering. *J. Comput. Inf. Sci.*, 2:61–88.

[Diday, 1988] Diday, E. (1988). The symbolic approach in clustering. In *Classification and related methods*. North-Holland Publishing.

[Diday and Simon, 1976] Diday, E. and Simon, J. C. (1976). Clustering analysis. In *Digital Pattern Recognition*, pages 47–94. Springer Verlag.

[Diehr, 1985] Diehr, G. (1985). Evaluation of a branch and bound algorithm for clustering. *SIAM Journal of Scientific and Statistical Computing*, 6:268–284.

[Dubes, 1987] Dubes, R. C. (1987). How many clusters are best? - an experiment. *Pattern Recognition*, 20(6):645–663.

[Dubes, 1993] Dubes, R. C. (1993). Cluster analysis and related issues. In *Handbook of Pattern Recognition & Computer Vision*, pages 3–22. World Scientific Publishing.

[Dubes and Jain, 1976] Dubes, R. C. and Jain, A. K. (1976). Clustering techniques: The user's dilemma. *Pattern Recognition on Neural Networks*, 8:247–260.

[Dubes and Jain, 1980] Dubes, R. C. and Jain, A. K. (1980). Clustering methodology in exploratory data analysis. In *Advances in Computers*, pages 113–125. Academic Press.

[Dubuisson and Jain, 1994] Dubuisson, M. P. and Jain, A. K. (1994). A modified hausdorff distance for object matching. In *Proceedings of the International Conference on Pattern Recognition*, pages 566–568.

[Duda and Hart, 1973] Duda, R. O. and Hart, P. E. (1973). *Pattern classification and scene analysis*. John Wiley and Sons.

[Duran and Odell, 1974] Duran, B. S. and Odell, P. L. (1974). *Cluster analysis: A survey*. Springer Verlag.

[Everitt, 1993] Everitt, B. S. (1993). *Cluster analysis*. Edward Arnold.

[Fisher and Van Ness, 1971] Fisher, L. and Van Ness, J. W. (1971). Adimissible clustering procedures. *Biometrika*, 58:91–104.

[Fisher, 1936] Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188.

[Fritsche, 1973] Fritsche, M. (1973). Automatic clustering techniques on information retrieval. Technical report, University of Stuttgart.

[Fu and Lu, 1977] Fu, K. S. and Lu, S. Y. (1977). A clustering procedure for syntactic patterns. *IEEE Transactions on Systems Mans and Cybernectics*, 7:734–742.

[Fukunaga, 1990] Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. Academic Press, 2 edition.

[Gowda and Diday, 1992] Gowda, K. C. and Diday, E. (1992). Symbolic clustering using a new dissimilarity cluster measure. *IEEE Trans. Pattern Syst. Man Cybern.*, 22:368–378.

[Gowda and Krishna, 1977] Gowda, K. C. and Krishna, G. (1977). Aglomerative clustering using the concept of mutual nearest neighborhood. *Pattern Recognition*, 10:105–112.

[Hansen and Jaummard, 1997] Hansen, P. and Jaummard, B. (1997). Cluster analysis and mathematical programming. *Matematical Programming*, 79:191–215.

[Hartigan, 1975] Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley and Sons.

[Hawkins et al., 1982] Hawkins, D. M., Muller, M. W., and Krooden, J. A. T. (1982). Cluster analysis. In *Topics in Applied Multivariate Analysis*. Cambridge University Press.

[Hertz et al., 1991] Hertz, J., Krogh, A., and Palmer, R. G. (1991). Introduction to the theory of neural computation. In *Sciences of Complexity lecture notes*. Addison-Wesley.

[Huttenlocher et al., 1993] Huttenlocher, D. P., Klanderman, G. A., and Rucklidge, W. J. (1993). Comparing images using the hausdorff distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(9):850–863.

[Ichino and Yaguchi, 1994] Ichino, M. and Yaguchi, H. (1994). Generalized minkowski metrics for mixed feature-type data analysis. *IEEE Transactions on Systems Mans and Cybernectics*, 24:698–708.

[Jain and Dubes, 1988] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Advanced references series. Prentice-Hall.

[Jain and Flynn, 1996] Jain, A. K. and Flynn, P. (1996). Image segmentation using clustering. In Ahuja, N. and Bowyer, K., editors, *Advances in Image Understanding*, pages 65–83. IEEE Press.

[Jain and Mao, 1994] Jain, A. K. and Mao, J. (1994). Neural networks and pattern recognition. In Zurada, J. M., Marks, R. J., and Robinson, C. J., editors, *Proceedings of the IEEE World Congress on Computational Intelligence*.

[Jain and Mao, 1996] Jain, A. K. and Mao, J. (1996). Artificial neural networks: A tutorial. *IEEE Computer*, 29:31–44.

[Jain et al., 1999] Jain, A. K., Murty, M., and Flynn, P. (1999). Data clustering a review. *ACM Computing Surveys*, 31:264–323.

[Jain et al., 1986] Jain, N. C., Indrayan, A., and Goel, L. R. (1986). Monte carlo comparison of six hierarquical clustering methods on random data. *Pattern Recognition*, 19(1):95–96.

[Jarvis and Patrick, 1973] Jarvis, R. A. and Patrick, E. A. (1973). Clustering using a similarity method based on shared near neighbors. *IEEE Trans. Comput.*, 22(8):1025–1034.

[Jensen, 1969] Jensen, R. E. (1969). A dynamic programming algoritm for cluster analysis. *Operations Research*, 17:137–142.

[Joachins, 1969] Joachins, T. (1969). A dynamic programming algorithm for cluster analysis. *Operations Research*, 17:1034–1057.

[King, 1967] King, B. (1967). Step-wise clustering procedures. *J. Am. Stat. Assoc.*, 69:86–101.

[Knuth, 1973] Knuth, D. (1973). *The art of computer programming*. Addison-Wesley.

[Kohonen, 1989] Kohonen, T. (1989). *Self-Organitazion and Associative Memory*. Information science series. Springer Verlag, 3 edition.

[Koontz et al., 1975] Koontz, W. L. G., Narendra, P. M., and Fukunaga, K. (1975). A branch and bound clustering algorithm. *IEEE Transactions on Systems Mans and Cybernectics*, 8:381–389.

[Lee, 1981] Lee, R. C. T. (1981). Cluster analysis and its applications. In *Advances in information systems science*. Plenum Press.

[Lu and Fu, 1978] Lu, S. Y. and Fu, K. S. (1978). A sentence-to-sentence clustering procedure for pattern analysis. *IEEE Transactions on Systems Mans and Cybernectics*, 8:381–389.

[Mangasarian, 1997] Mangasarian, O. L. (1997). Mathematical programming in data mining. *Data Mining and Knowledge Discovery*, 1:183–201.

[Mao and Jain, 1996] Mao, J. and Jain, A. K. (1996). A self-organizing network for hyperellipsoidal clustering (hec). *IEEE Trans. Neural Networks*, 7:16–29.

[McLachlan, 1992] McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley and Sons.

[McQueen, 1967] McQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Firth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.

[Michalski et al., 1983] Michalski, R., Stepp, R. E., and Diday, E. (1983). Automated construction of classifications conceptual clustering versus numerical taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 5:396–409.

[Michie et al., 1994] Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994). *Machine Learning and Statistical Classification*. Artificial intelligence. Ellis Horwood.

[Mirkin, 1996] Mirkin, B. (1996). *Mathematical Classification and Clustering*. Kluver Academic Publishers.

[Mishra and Raghavan, 1994] Mishra, S. K. and Raghavan, V. V. (1994). *An empirical study of the performance of heuristic methods for clustering*. PhD thesis, University of Southwestern Louisiana.

[Moor, 1988] Moor, B. K. (1988). Art 1 and pattern clustering. In *Connectionist Summer School*, pages 174–185. Morgan Kaufman.

[Murtagh, 1984] Murtagh, F. (1984). A survey of recent advances in hierarquical clustering algorithms wich use cluster centers. *Comput. J.*, 26:354–359.

[Murty and Jain, 1995] Murty, M. N. and Jain, A. K. (1995). Knowledge-based clustering scheme for collection management and retrieval of library books. *Pattern Recognition*, 28:949–964.

[Nagy, 1968] Nagy, G. (1968). State of the art in pattern recognition. *Proc. IEEE*, 56:836–862.

[Oehler and Gray, 1995] Oehler, K. L. and Gray, R. M. (1995). Combining image compression and classification using vector quantization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17:461–473.

[Pal et al., 1993] Pal, N. R., Bezdek, J. C. D., and Tsao, E. C. K. (1993). Generalized clustering networks and kohonen's self-organizing scheme. *IEEE Trans. Neural Networks*, 4:549–557.

[Quinlan, 1993] Quinlan, J. R. (1993). *Programs for Machine Learning*. Morgan Kaufman.

[Rasmussen, 1992] Rasmussen, E. (1992). Clustering algorithms. In *Information retrieval data structures and algorithms*, pages 419–442. Prentice-Hall.

[Reeves, 1993] Reeves, C. R. (1993). *Modern Heuristic Techniques for Combinatorial Problem*. Blackwell.

[Ripley, 1989] Ripley, B. D. (1989). *Statistical inference for Spatial Processes*. Cambridge University Press.

[Salton, 1991] Salton, G. (1991). Developments in automatic text retrieval. *Science*, 253:974–980.

[Selim and Al-Sultan, 1991] Selim, S. Z. and Al-Sultan, K. (1991). A simulated annealing algorithm for the clustering problem. *Pattern Recognition*, 24(10):1003–1008.

[Selim and Ismail, 1984] Selim, S. Z. and Ismail, M. A. (1984). k-means type algorithm: generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:81–87.

[Sethi and Jain, 1991] Sethi, I. and Jain, A. K. (1991). *Artificial Neural Networks and Pattern Recognition Old and New Connections*. Elservier Science.

[Sneath and Sokal, 1973] Sneath, P. H. A. and Sokal, R. R. (1973). *Numerical Taxonomy*. Freeman.

[Soergel, 1967] Soergel, D. (1967). Mathematical analysis of documentation systems. *Information Stor. Retr.*, 3:129–173.

[Spath, 1980] Spath, H. (1980). *Cluster analysis algorthms for data reduction and classification*. Ellis Horwood.

[Sun et al., 1994] Sun, L. X., Xie, Y., Song, X. H., Wang, J. H., and Yu, R. Q. (1994). Cluster analysis by simulated annealing. *Computers and Chemistry*, 18:103–108.

[Symon, 1977] Symon, M. J. (1977). Clustering criterion and multi-variate normal mixture. *Biometrics*, 77:35–43.

[Tanaka, 1995] Tanaka, E. (1995). Theoretical aspects of syntactic pattern recognition. *Pattern Recognition*, 28:1053–1061.

[Titterington et al., 1985] Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical analysis of finite mixture distributions*. John Wiley and Sons.

[Vapnik, 1995] Vapnik, V. N. . (1995). *The nature of statistical learning theory*. Springer Verlag.

[Ward, 1963] Ward, J. H. J. (1963). Hierarquical grouping to optimize an objective function. *J. Am. Stat. Assoc.*, 58:236–244.

[Wilson and Martinez, 1997] Wilson, D. R. and Martinez, T. R. (1997). Improved heterogeneous distance functions. *J. Artif. Intell. Res.*, 6:1–34.

[Xavier, 1993] Xavier, A. E. (1993). Solução de problemas de programação não-diferenciáveis via suavização. Technical report, Universidade Federal do Rio de Janeiro.

[Zahn, 1971] Zahn, C. T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, 20:68–86.

[Zhang and Michalski, 1995] Zhang, J. and Michalski, R. S. (1995). An integration of rule induction and exemplar-based learning for graded concepts. *Mach. Learn.*, 21(3):235–267.

[Zhang, 1995] Zhang, K. (1995). Algorithms for the constrained editing distance between ordered labeled trees and related problems. *Pattern Recognition*, 28:463–474.