Universidade Federal do Rio Grande do Norte

Centro de Tecnologia

Programa de Pós-Graduação em Engenharia Elétrica

Contribuições aos Processos de *Clustering* com Base em Métricas não-Euclidianas

Msc. Allan de Medeiros Martins

Orientador:
Dr. Adrião Duarte Dória Neto

co-orientador:
Dr. Jorge Dantas de Melo

Natal/RN - Brasil

Abril de 2005

Allan de Medeiros Martins

Contribuições aos Processos de *Clustering* com Base em Métricas não-Euclidianas

Orientador:

Adrião Duarte Dória Neto

Co-Orientador:

Jorge Dantas de Melo

Tese submetida ao programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Norte como parte dos requisitos necessários para obtenção do título de DOUTOR em CIÊNCIAS.

Natal/RN- Brasil

Abril de 2005

Catalogação da publicação na fonte. UFRN / Biblioteca Central "Zila Mamede"
Seção de Processos Técnicos

Martins, Allan de Medeiros

Contribuições aos Processos de *Clustering* com Base em Métricas não-Euclidianas / Allan de Medeiros Martins. Natal: 2005.
Orientador: Adrião Duarte Dória Neto
Co-Orientador: Jorge Dantas de Melo
Disertação (Mestrado) - Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica.

1. *Clustering* 2. Teoria da Informação 3. Métricas não-Euclidianas

RN/UF/BCZM CDU 123.456 : 789.1011 (121.3)

CONTRIBUIÇÕES AOS PROCESSOS DE *CLUSTERING* COM BASE EM MÉTRICAS NÃO-EUCLIDIANAS

Allan de Medeiros Martins

Tese apresentada à Coordenação do programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Norte como requisito parcial à obtenção do título de Doutor em Ciências. Aprovada, em 8 de Abril de 2005, pela Comissão Examinadora formada com os seguintes membros:

Composição da Banca Examinadora:

Adrião Duarte Dória Neto, (Doutor), Orientador

Jorge Dantas de Melo, (Doutor), Co-Orientador

José Alfredo Ferreira Costa, (Doutor), Examinador interno

Ananias Monteiro Mariz, (Doutor), Examinador interno

Benedito Guimarães Aguiar Neto, (Doutor), Examinador externo

Ricardo Tanscheit, (Doutor), Examinador externo

Natal/RN - Brasil
Abril de 2005

Agradecimentos

Meus agradecimentos vão em primeiro lugar para os meus orientadores professores Adrião Duarte Dória Neto e Jorge Dantas de Melo não só pela excelente orientação, sem a qual este trabalho não existiria, mas por servirem de exemplo para mim, muito obrigado aos dois!

Gostaria de agradecer ao professor José Alfredo F. Costa pela ajuda durante este trabalho.

Vai um agradecimento especial à Nina Maria e Emílio Silva pela oportunidade e revisão da aplicação em microscopia. Ao meu grande amigo Agostinho Brito também pela oportunidade e revisão da aplicação de reconstrução 3D e vetorização de imagens.

Agradeço de todo coração aos meus familiares, meus pais Ivan Martins e Maria Goretti, e meus irmãos Rafa e Lora pela paciência durante esta jornada.

Agradeço a minha querida namorada Bruna Brandão pela paciência, carinho e incentivo que teve em todo o tempo, sendo compreensiva e sempre me apoiando, um beijão enorme!

Gostaria de agradecer também aos meus amigos professores da UnP: Marcelo, Aarão, Gustavo, Oscar, Nadja, Antônio, Badiali e toda a turma da computação (engenharia e sistemas) pela força e incentivo!

Por fim, agradeço aos meus amigos e a todos que, de forma direta ou indireta, contribuíram para este trabalho, muito obrigado!

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

Neste trabalho apresentamos um novo método de *clustering* que agrupa pontos de um conjunto de dados em classes. O método baseia-se em um algoritmo para ligação de *clusters* auxiliares que são obtidos usando-se técnicas de quantização vetorial tradicionais. São descritas algumas abordagens durante o desenvolvimento do trabalho que baseiam-se em medidas de distância ou dissimilaridade (divergência) entre os *clusters* auxiliares. Este novo método utiliza apenas duas informações *a priori*, a saber: o número de centros auxiliares N_a e uma distância de limiar d_t que será utilizada para decidir sobre a ligação ou não dos *clusters* auxilares. O número de *clusters* pode ser automaticamente encontrado pelo método, que o faz com base na distância limiar d_t escolhida. Analogamente, o número de classes, pode ser fornecido como informação adicional para auxiliar na escolha do limiar correto. Algumas análises são feitas e os resultados são comparados com outros métodos tradicionais de *clustering*. Neste trabalho são analisadas diferentes métricas de dissimilaridade e uma nova métrica baseada no conceito de negentropia é proposta. Além de agrupar pontos de um conjunto de classes, é proposto um método para o modelamento estatístico das classes de modo a se obter uma expressão para a probabilidade de um ponto pertencer a uma das classes.

Experimentos com diversos valores de N_a e d_t são realizados em conjuntos de teste e os resultados são analisados de maneira a se estudar a robustez do método e propor heurísticas para a escolha do limiar correto. No trabalho são explorados os aspectos de teoria da informação aplicados ao cálculo das divergências. São exploradas em particular as diferenças medidas de informação e divergência utilizando a entropia de Rényi. Os resultados utilizando as diferentes métricas são comparados e comentados. O trabalho ainda conta com apêndices onde são expostas aplicações reais utilizando o método proposto.

Abstract

In this work we present a new *clustering* method that groups up points of a data set in classes. The method is based in a algorithm to link auxiliary clusters that are obtained using traditional vector quantization techniques. It is described some approaches during the development of the work that are based in measures of distances or dissimilarities (divergence) between the auxiliary clusters. This new method uses only two *a priori* information, the number of auxiliary clusters N_a and a threshold distance d_t that will be used to decide about the linkage or not of the auxiliary clusters. The number os classes could be automatically found by the method, that do it based in the chosen threshold distance d_t, or it is given as additional information to help in the choice of the correct threshold. Some analysis are made and the results are compared with traditional *clustering* methods. In this work different dissimilarities metrics are analyzed and a new one is proposed based on the concept of negentropy. Besides grouping points of a set in classes, it is proposed a method to statistical modeling the classes aiming to obtain a expression to the probability of a point to belong to one of the classes.

Experiments with several values of N_a e d_t are made in tests sets and the results are analyzed aiming to study the robustness of the method and to consider heuristics to the choice of the correct threshold. During this work it is explored the aspects of information theory applied to the calculation of the divergences. It will be explored specifically the different measures of information and divergence using the Rényi entropy. The results using the different metrics are compared and commented. The work also has appendix where are exposed real applications using the proposed method.

Capítulo 1

Introdução

Técnicas de classificação de padrões e *clustering* (normalmente citado em inglês na grande maioria da literatura) são utilizadas em muitas áreas do conhecimento ou em aplicações tais como mineração de dados, análise de dados de petróleo, geoprocessamento, processamento de sinais, imagens dentre outras. Por isto, o desenvolvimento de novos algoritmos para realizar a tarefa de classificar um conjunto de dados em padrões ou agrupamentos (*clusters*) é uma tarefa bastante importante. O campo de estudo nessa área está estabelecido a um certo tempo com aplicações em várias áreas científicas e tecnológicas, sempre requerendo o desenvolvimento de técnicas cada vez mais eficientes. Existe uma literatura básica extensa em classificação de padrões e *clustering* [16, 6, 17, 78] que trata de técnicas fundamentais nesta área. Podemos distinguir basicamente duas grandes classes de algoritmos e técnicas de classificação de padrões: a classificação supervisionada e a não-supervisionada. O grande desafio hoje está no desenvolvimento de técnicas não-supervisionadas, onde temos pouca ou nenhuma informação sobre às classes as quais as amostras de dados captadas pertencem. Muitas vezes não temos nem a informação sobre quantas classes existem.

O termo *clustering* se refere a tarefa de, dado um conjunto de pontos de D dimensões, separá-los em classes (agrupamentos) de acordo com certo critério. O grande desafio em *clustering* está em realizar a separação das classes quando os estas estão distribuídas de maneira complexa no espaço. Quando isto acontece dizemos que os dados são não-linearmente separáveis. O conjunto de pontos dados são, em

geral, amostras de algum processo como por exemplo a altura e peso de um conjunto de pessoas. Os pontos neste caso serão formados por um vetor de duas dimensões onde o conteúdo de uma delas será a altura e o da outra será o peso. Deseja-se então classificar as pessoas em, por exemplo, duas classes, sejam elas obesas e sadias. Normalmente são colhidas apenas um número N de amostras, que sejam representativas do processo como um todo. O resultado da utilização da técnica de *clustering* obviamente depende desta quantidade.

Em qualquer que seja a técnica utilizada, as amostras fornecidas servem para gerar um modelo. Este modelo será utilizado para classificar outros pontos que não faça parte da amostra inicial, obtendo desta maneira um classificador.

Um problema bastante freqüente na área de *clustering* é a alta dimensão dos dados. Alguns autores [16, 6] comentam que a complexidade do classificador aumenta muito com o aumento da dimensão dos dados a serem classificados[1]. Algumas técnicas porém, podem ser utilizadas para reduzir a dimensionalidade dos dados para que estes possam ser classificados utilizando-se um classificador mais simples. Jolliffe em 1986 [40] introduz uma técnica denominada *principal component analisys* (PCA) ou análise de componentes principais. Esta técnica consiste em promover uma transformação linear nos dados de modo que estes tenham suas componentes mais relevantes nas primeiras dimensões, em eixos denominados eixos principais.

Mesmo utilizando-se técnicas como a PCA, ainda existem problemas onde os dados, estando distribuídos em apenas um ou mais eixos principais, a separação entre as classes presentes não é feita. Fisher [20] e posteriormente outros autores [62] introduzem o conceito de discriminante de Fisher (mais tarde abordado computacionalmente em outros trabalhos [63]). Em sua análise, a transformação linear aplicada não visa mais proporcionar um eixo principal e sim um eixo onde a separação das médias de cada classe seja máxima.

Existe ainda uma abordagem não linear para a análise de componentes principais. Em seu livro [33], Haykin discute uma técnica de análise de *kernel* baseada em uma

[1]Muitos autores chamam esta dificuldade de *curse of dimensionality* ou maldição da dimensionalidade, devido a grande dificuldade causada pelo aumento da dimensão dos dados

arquitetura neural para o cálculo de projeções não lineares que promovem a redução de dimensionalidade dos dados em casos de distribuições não- linearmente separáveis.

Mesmo em baixas dimensões, é necessário estabelecer uma distinção entre os dados de uma classe e de outra. Na literatura, a ferramenta responsável por realizar esta distinção denomina-se função discriminante. Muitas abordagens são encontradas na literatura, a mais simples corresponde as funções discriminantes lineares. Em alguns livros [16, 33] são estudadas funções discriminante lineares. Haykin [33] trata uma abordagem através do uso de redes neurais (perceptrons de uma única camada). Bishop [6] e Claus [82], abordam discriminantes não lineares e descrevem técnicas para o cálculo dos mesmos. A decisão entre pertinência de um certo dado a uma classes também pode ser feita estatisticamente, utilizando-se análise bayesiana [68, 16].

Fazendo uso destas técnicas, existem muitos métodos para realizar a tarefa de classificação de padrões. Os métodos podem ser comparados basicamente quanto a presença de conhecimento *a priori* existente sobre o problema, obviamente sem considerar o conhecimento dos dados a serem classificados em si. Neste sentido, podemos dividir ou métodos de classificação de padrões em métodos supervisionados e métodos não-supervisionados, como já comentado.

1.1 Métodos supervisionados

Métodos supervisionados são aqueles em que amostras de cada classe são fornecidas ao classificador juntamente com a informação sobre a qual classe pertence cada amostra. De posse desta informação, os métodos realizam uma espécie de treinamento ou ajuste para que o classificador possa funcionar de maneira adequada, classificando corretamente amostras que não estejam no conjunto inicial de treinamento. Dentre os métodos supervisionados podemos citar os métodos paramétricos [16]. Classificadores que utilizam estes métodos, possuem funções discriminante parametrizadas $f_i(\mathbf{x}, \theta)$ para classificar um vetor de atributos $\mathbf{x}$ em uma classe i, representada pelo vetor de parâmetros θ. O objetivo desta análise é encontrar que parâmetros θ melhor descrevem as amostras dos dados. O cálculo

destes parâmetros pode se feito de modo a otimizar uma função denominada de função de verossimilhança [16, 76]. Duda e Hart [16] fazem uma análise detalhada sobre a utilização da função de verossimilhança para diversas funções discriminantes. O caso mais comum na literatura de utilização desta técnica de cálculo de parâmetros é onde a função discriminante é uma gaussiana com parâmetros de média, variância, e probabilidade *a priori*.

Outros métodos supervisionados utilizam apenas as informações das amostras para classificar um dado em um conjunto. Estes métodos são denominados na literatura como métodos não-paramétricos. Em 1970, Patrick e Fisher [71] apresentaram um algoritmo para classificação de padrões baseado apenas nos dados amostrados diretamente (sem estabelecer uma função discriminante parametrizada). Este algoritmo é conhecido como k-vizinhos mais próximos e é tratado em detalhes em grande parte da literatura atual.

Métodos supervisionados são normalmente métodos apenas de classificação, a separação dos dados em suas classes não precisa ser feita, pois já conhecemos que dados pertencem a que classe. Estes métodos são importantes pois servem de base para implementação de classificadores eficientes após a fase de separação.

1.2 Métodos não-supervisionados

O problema de classificação torna-se mais complexo quando não se tem a informação de classes nas amostras do conjunto de dados a ser classificado. Estas situações são tratadas através de métodos não-supervisionados de classificação. Em geral problemas de classificação não supervisionada podem ser vistos como problemas de *clustering* ou aglomeração. Este tipo de análise é muito importante, pois em problemas práticos como mineração de dados, segmentação de imagens ou reconhecimento de voz, nem sempre se tem "exemplos" de classes para se treinar o classificador. Este é o tipo de método explorado neste trabalho.

Técnicas de *clustering* tem sido utilizadas em diversas aplicações [74, 93, 75, 10, 83]. Dentre estas técnicas, podemos citar a estimação da máxima verossimilhança [17, 16] onde os parâmetros de uma função discriminante são

estimados sem a presença da informação das classes das amostras.

Em 1977, Dempster [15] introduziu um algoritmo denominado *Expectation Maximization* (EM) utilizado para o cálculo de parâmetros gaussianos utilizados para modelar um conjunto de dados. Este algoritmo é bastante famoso e muito utilizado para estimar os parâmetros de misturas de gaussianas, embora tenha como desvantagem o custo computacional. Posteriormente, alguns autores [14, 65] exploraram o algoritmo EM e em 1996 Xu e Jordan [89] fizeram uma análise da convergência do algoritmo. Ainda objetivando ó cálculo de parâmetros de uma dada função discriminante, Bishop [6] e Kosko [47] apresentam abordagens baseadas em redes neurais denominada redes bayesianas para estimação dos parâmetros em abordagens paramétricas para classificação de padrões. Ainda nesta linha, Figueiredo e Jain [18] desenvolveram uma técnica que inclui a seleção automática do número de gaussianas na mistura.

Linde, Buzo e Gray [53] descrevem uma técnica que tornou-se bastante famosa e é largamente utilizada em algoritmos de classificação não-supervisionada. Esta técnica, denominada na literatura por *k-means* corresponde a uma das principais técnicas de quantização vetorial e será explorada nos capítulos seguintes. Existem diversas outras técnicas de quantização vetorial e dentre estas, uma bastante comum que faz uso de uma abordagem neural são as redes neurais competitivas [33]. Estas redes consistem em uma arquitetura de uma única camada onde os neurônios que ligam a entrada com a saída representam padrões no espaço dos dados. Os pesos destes neurônios serão coordenadas para vetores que representarão todo o conjunto, quantizando o mesmo em poucos centros, algumas vezes denominados *code-vectors* e o conjunto completo dos centros denominado de *code-book*. O cálculo dos pesos dos neurônios é realizado de uma maneira iterativa. Em seu livro, Kohonen [46] realiza uma prova de convergência do algoritmo de treinamento competitivo utilizando o método do gradiente como ferramenta de otimização. Outras técnicas de *clustering* que utilizam redes neurais competitivas podem ser encontradas na literatura [55, 41]. Nestas abordagens, são utilizadas métricas não-Euclidianas para medidas de similaridade como a distância de Mahalanobis, que é uma das métricas utilizadas neste trabalho. Existem até trabalhos que utilizam algoritmos genéticos [39] para

realizar o a quantização vetorial.

Outro método de *clustering* consiste em utilizar funções critério (detalhadas e discutidas por Everitt [17]) a serem otimizadas. Nesta abordagem, os dados são modelados de uma maneira paramétrica e os parâmetros são estimados de maneira a minimizar ou maximizar uma função critério (ou custo). O Algoritmo EM (mencionado anteriormente) é um caso particular deste tipo de método onde a função a ser otimizada é a função de verossimilhança (discutida nos próximos capítulos). Recentemente Veenma et al. desenvolveram um método de *clustering* baseado na minimização de uma função erro quadrático [87]. Em seu artigo, mostraram como utilizar observações de tendências em medidas feitas em parâmetros do algoritmo para ajudar no resultado final. Este tipo de técnica será discutida também neste trabalho e consiste de uma ferramenta bastante importante na escolha correta dos parâmetros.

Um outro tipo de problema de classificação de padrões surge quando não temos disponível a informação sobre quantas classes existem no conjunto de dados ou quando a classificação tem que ser feita *on-line*. Classificação *on-line* ocorre quando não temos de imediato todas as amostras disponíveis e a classificação deve ser efetuada à medida que as amostras vão sendo apresentadas. Em 1980, Späth [81] apresenta uma abordagem denominada *leader-follower*. Nesta abordagem, uma medida é feita nos dados que vão sendo apresentados e, comparando-se com uma medida limiar, decide-se por classificar ou criar uma nova classe. Posteriormente, Carpenter e Grosberg [8] apresentam um modelo neural denominado *Adaptive Resonance Theory*(ART) baseando-se no comportamento de neurônios biológicos para realizar o agrupamento dos dados. O problema gerado por não se ter a informação sobre quantas classes existem, é bastante discutido na literatura. Em 1996, Hardy [32] compara diversos métodos de *clustering* utilizados como base para encontrar o número correto de classes. Logo após, em 1999, Kothari e Pitts [48] discutem um método de otimização para encontrar o número correto classes. Fraley e Raftery escreveram um relatório técnico [22] discutindo sobre o número de *clusters* em um conjunto e sobre métodos adequados para estimação do mesmo.

Recentemente, Kohonen propôs uma abordagem neural auto-organizada para

agrupar dados. Esta abordagem é bastante discutida em seu livro [46] e consiste em representar os dados originais por um mapa com topologia bem definida. De maneira análoga às redes neurais competitivas, os mapas auto-organizáveis (ou *Self-Organizing Maps*, SOM como são geralmente conhecidos na literatura) são arquiteturas neurais de apenas uma única camada. No caso dos mapas SOM, os neurônios de saída estão dispostos formando uma topologia bem definida. O treinamento do SOM é feito de maneira iterativa, calculando-se os valores dos pesos dos neurônios em cada iteração.

Existem também na literatura alguns métodos desenvolvidos para classificação de dados gerados artificialmente onde a estatística é bastante complexa. Lipson e Siegelmann [54] utilizam uma abordagem neural para modelar o classificador utilizando neurônios com métricas diferentes da Euclidiana (idéia central deste trabalho). Algumas idéias utilizando o Mapa de Kohonen com métricas baseadas em teoria da informação (como a divergência de Kullback-Leibler) foram propostas por Hollmen et al. Ultsch e Vetterem desenvolvem uma técnica de segmentação do mapa de Kohonen como objetivo de realizar a classificação de padrões mapeando-os para um espaço de dimensão menor [86]. Posteriormente, em 1999, Costa [13] desenvolveu métodos hierárquicos para trabalhar com o algoritmo de segmentação do Mapa de Kohonen.

1.3 Método proposto

O método proposto neste trabalho consiste de um método não- supervisionado para classificação de padrões. O objetivo é realizar a classificação de conjuntos de dados sem forma definida ou que tenham sido gerados por funções densidade de probabilidade muito complexas.

As principais contribuição propostas neste trabalho consistem no desenvolvimento de uma nova técnica de *clustering* e um método para estabelecer um modelo estatístico para o conjunto de dados analisado. O método de *clustering* proposto baseia-se em medidas de dissimilaridade não-Euclidianas que incorporam alguma informação sobre a estatística dos dados. Basicamente o método tem como

base a divisão do conjunto em um número excessivo de pequenas classes auxiliares para posterior agrupamento das mesmas em *clusters* maiores. As contribuições importantes a destacar no método e que serão descritas no decorrer do texto podem ser resumidas em: a simplicidade computacional que é conseguida devido a utilização de uma medida baseada na negentropia calculada analiticamente, o que livra o algoritmo de cálculos iterativos custosos como é o caso do algoritmo EM; utilização da entropia de Rényi para facilitar o cálculo analítico da negentropia; a obtenção de uma técnica extremamente simples de se obter modelos de misturas de gaussianas, dada a necessidade de se obter o modelo gaussiano de cada classe auxiliar, sendo estas posteriormente agrupadas formando um agrupamento de gaussianas; cálculo da negentropia baseando-se medidas de divergência, em particular a divergência de Kullback-Leibler.

A técnica de dividir os dados em classes ou centros auxiliares para depois agrupá-los foi inicialmente abordada por Tyree e Long [85]. A idéia neste artigo é utilizar a distância entre os centros das classes auxilares e após isso, classificar os pontos pela distância destes para os segmentos de retas gerados pela ligação dos centros. O grande problema que surge nesta abordagem é a utilização da distância Euclidiana para ligação dos centros, pois além de não ligar corretamente os mesmos em alguns casos existe uma dependência muito grande do limiar utilizado (distância a partir da qual se decide por ligar ou não dois centros) e os valores de escala dos dados em si. Neste trabalho, utilizamos métricas que levam em conta a estatística dos dados, e resolvemos o problema de ligação correta dos dados, além de não depender da escala dos mesmos. Além de sugerir um esquema de classificação bem mais eficiente, baseado em um modelo de misturas de gaussianas. A dependência da escala dos dados do conjunto na escolha do limiar é um fator crítico. Isto dificulta a escolha deste limiar, já que para o mesmo conjunto observado em uma unidade diferente (por exemplo comprimentos em centímetros ao invés de metros) o limiar será diferente.

A utilização de métricas não-Euclidianas em *clustering* já se mostrou bastante eficiente. Em [24] o autor desenvolve uma técnica de *clustering* que utiliza medidas de entropia para segmentar dados de imagens de ressonância magnética. Nesta

ocasião, o autor mostra a robustez das medidas estatísticas, que são objeto principal do presente trabalho. No capítulo 4 são feitos comentários comparando este método com o método proposto neste trabalho.

No decorrer do trabalho foram sendo desenvolvidas e aprimoradas técnicas, para realizar a tarefa de *clustering* baseadas na ligação de centros auxiliares, porém com medidas que incorporam a estatística local dos dados. Diversas métricas foram testadas até que se chegasse ao desenvolvimento de uma métrica bastante robusta. No decorrer do texto, serão explicadas cada métrica, seus problemas, e como estes foram sendo resolvidos com métricas cada vez mais adequadas.

1.4 Sumário

Muitos trabalhos se relacionam com as técnicas de *clustering*. Muitas técnicas utilizam combinações de outras técnicas e algoritmos para realizar a tarefa de separar *clusters* em um conjunto de dados. Nem todos os métodos e algoritmos utilizados neste trabalho são específicos para *clustering*. A idéia é utilizar-se de algoritmos com baixa complexidade computacional para que se obtenha um método rápido, porém robusto. No capítulo 2 serão apresentados os aspectos teóricos envolvidos no contexto deste trabalho onde as técnicas, métricas e abordagens são descritas de maneira a introduzir a teoria necessária para o desenvolvimento do mesmo. No capítulo 3 o método desenvolvido é apresentado. São detalhadas as etapas envolvidas na classificação de um conjunto de dados e fundamentada a utilização de cada aspecto teórico enfocado no capítulo 2. No capítulo 4 são mostrados e comentados alguns resultados obtidos. Os resultados apresentados são mostrados seguidos de análises teóricas feitas com base no que se esperava ou no que realmente se obteve, bem como sobre os parâmetros utilizados. Em seguida são apresentadas algumas conclusões e perspectivas para trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Classificar padrões significa atribuir uma classe ou grupo a cada um dos elementos de um conjunto de dados. Utilizamos classificação de padrões sempre que temos que realizar algum tipo de seleção como por exemplo escolher dentre frutas maduras ou verdes, separar livros velhos de novos, ou até mesmo selecionar quais objetos serão guardados em quais gavetas. Apesar de ser uma tarefa bastante corriqueira e simples de ser executada, realizá-la computacionalmente pode chegar a ser bastante complicado. Neste caso, classificar padrões significa: dada uma tabela onde cada linha corresponda a um objeto ou dado qualquer, e cada coluna seja um atributo qualquer deste objeto, atribuir a cada linha desta tabela uma classe ou grupo de maneira que, linhas com grupos ou classes iguais, terão atributos (colunas da tabela) semelhantes perante uma medida qualquer. Um exemplo clássico de classificação de padrões é dado por Duda e Hart [16]. Neste exemplo, os "dados" a serem classificados são duas espécies de peixes. Cada uma dessas espécies pode ser considerada uma classe. Cada peixe do conjunto a ser classificado será representado por uma linha da tabela e cada coluna representará uma dimensão física do peixe (largura e comprimento, por exemplo). O objetivo é classificar cada um dos peixes presentes no conjunto todo e identificá-los como pertencentes a uma das duas espécies, baseando-se apenas nas medidas dos tamanhos dos mesmos.

Um passo fundamental para o entendimento dos algoritmos de classificação é a representação gráfica dos dados. A representação em forma de tabela nos mostra pouco a respeito do conjunto. Uma forma bastante utilizada de representar

um conjunto de dados é a representação em forma de um diagrama de pontos denominado de *scatter* (Figura 2.1).

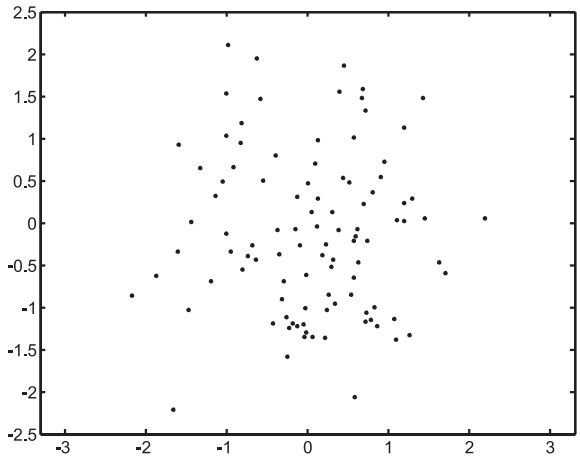


Figura 2.1: Diagrama de pontos representando um conjunto de dados

Neste diagrama, cada ponto representa um elemento do conjunto. As coordenadas de cada ponto representam os atributos e em geral não tem significado relevante. É muito comum o desenho desse tipo de diagrama sem o rótulo do nome dos eixos. No exemplo dos peixes, um peixe que tivesse comprimento 10cm e largura 4cm seria representado por um ponto de coordenadas $x = 10$ e $y = 4$. Claro que este tipo de diagrama funciona bem para duas (ou até três) dimensões, onde apenas dois (ou três) atributos são utilizados. Para dimensões elevadas outras técnicas de visualização podem ser utilizadas. Neste trabalho, para fins de exemplificação, serão utilizados conjuntos bi e tri-dimensionais, mesmo que o método ou técnica desenvolvido funcione para dimensões mais altas.

O termo "classificação de padrões" é muito utilizado para denominar a área de conhecimento onde se utilizam de diversas técnicas para resolver o problema de separar ou classificar pontos em conjuntos de dados. Muitas destas técnicas vêm sendo desenvolvidas e se aplicam em diversos casos de problemas de classificação. Para realizar a classificação dos dados de um conjunto de acordo com algum critério (frutas pela sua cor, carros pela sua velocidade ou células pela sua forma) é necessário obter informações sobre os dados em questão. A informação mais comum disponível para resolver um problema de classificação são amostras que serão classificadas. A partir destas amostras iremos encontrar um modelo matemático para que possamos

inferir sobre um elemento qualquer (pertencente ao conjunto de amostra ou não) atribuindo-lhe uma classe. Esta amostra de dados do problema consiste na principal informação a ser fornecida para o sistema que irá realizar a classificação, porém nem sempre é a única informação disponível. Outras informações a respeito do problema podem ser dadas, tais como: número de classes em que se deseja dividir o conjunto de dados, conhecimento *a priori* sobre as classes das amostras, etc. Dependendo do tipo de informação que é dada sobre o conjunto ou amostra utilizada, a classificação é dita supervisionada ou não-supervisionada como mencionado no capítulo 1.

O termo supervisionada ou não-supervisionada refere-se a existência ou não de exemplos (amostras) onde já se conhece sua classe. Podemos comparar o caso de problemas supervisionados com aquela classificação onde um especialista informa, em principio, quais dados utilizados como amostras pertencem a qual classe. No exemplo dos peixes já citado, o conjunto utilizado para "treinar" o classificador consistiria de uma tabela com os atributos das amostras (largura, comprimento, etc) e a classe de cada amostra (salmão, sardinha, etc). Com essas informações, um modelo é criado e então o classificador é implementado.

Já em classificações ditas não-supervisionadas apenas amostras são fornecidas. Nenhuma informação sobre a quais classes cada uma pertence são dadas. No caso dos peixes, apenas os atributos dos peixes são dados (largura, comprimento, etc...), não há a informação de classes na tabela. O problema de classificação não-supervisionada é em geral mais difícil do que o problema de classificação supervisionada. O problema torna-se tão mais difícil quanto menos informações são fornecidas. Uma informação normalmente fornecida aos classificadores não-supervisionados é o número de classes existentes. Outras informações podem ser dadas como por exemplo, a forma como os conjuntos de dados se agrupam no espaço dos atributos, ou alguma estatística sobre os mesmos.

Alguns métodos tentam utilizar, além das amostras, a menor quantidade de informações possível para desenvolver um classificador. Neste sentido, muitos métodos tem sido desenvolvidos e se baseiam nas mais diversas técnicas [8, 70, 5].

O termo *clustering* é normalmente utilizado para designar algoritmos que agrupam dados classificando-os como grupos genéricos (grupo 1, grupo 2, etc...).

Os grupos (ou *clusters* como é comum ser designado na literatura), são conjuntos de elementos que possuem atributos semelhantes. Segundo Everitt [17] a definição formal de *cluster* é difícil de ser elaborada e muitas vezes é formada de maneira a se adequar a uma situação particular. Neste trabalho, consideramos *cluster* como um conjunto de dados que possuem uma classificação genérica (grupo 1, grupo 2, etc...) e que foi estabelecido de acordo com uma medida de dissimilaridade para com os demais grupos. Desta maneira, classificar um elemento de um conjunto de dados significa associar este a um desses grupos. Halkidi et al. disponibilizam uma revisão [30, 31, 29] bastante interessante sobre as técnicas de *clustering* modernas.

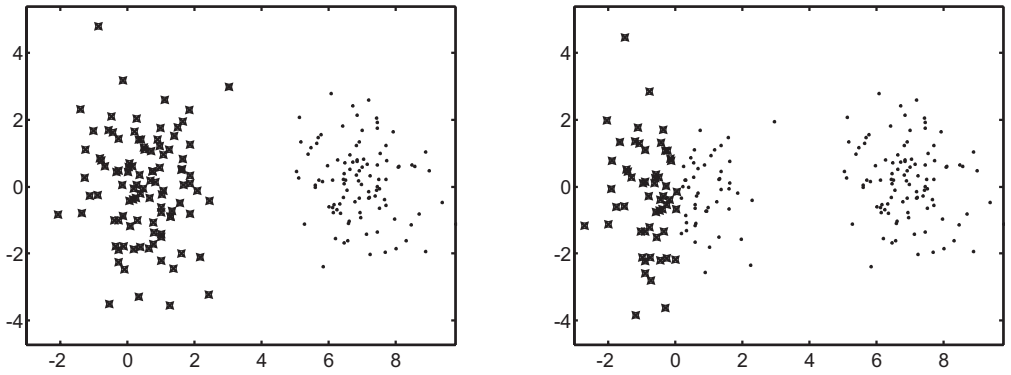
Nas seções seguintes, serão descritas algumas técnicas utilizadas e métodos para o desenvolvimento do algoritmo proposto neste trabalho para separar um conjunto de dados em *clusters* e associar um elemento qualquer a um dos *clusters* gerados.

2.1 Aglomeração (*Clustering*)

A tarefa de aglomerar dados de um conjunto qualquer é denominada geralmente de *clustering*. Ao se aglomerar dados de um conjunto qualquer, formamos classes ou *clusters*. Em geral a aglomeração pode ser vista como a rotulação de cada ponto ou dado do conjunto como sendo de uma das N_a classes existentes no conjunto todo. Esta rotulação deve obedecer um critério para que tenhamos uma boa aglomeração. Entende-se como uma boa aglomeração, a situação em que todos os pontos de uma mesma classe são semelhantes perante um critério de decisão estabelecido. Podemos observer como exemplo a Figura 2.2.

Na Figura 2.2(a), observamos que se considerarmos o critério "proximidade" dos vizinhos de mesma classe, o *cluster* representado com "x" (mais escuros) possui alguns pontos cujos vizinhos são distantes. Já na Figura 2.2(b), todos os pontos possuem distâncias parecidas para os vizinhos de mesma classe. Neste exemplo, o critério "distância para os vizinhos" foi utilizada para definir se uma rotulação esta correta ou incorreta.

Teoricamente podemos testar todas as possibilidades e escolher a rotulação que melhor se apresenta diante do critério escolhido. Esta técnica não é utilizada porque



(a) Pontos rotulados com um critério incorreto

(b) Pontos corretamente rotulados

Figura 2.2: Duas rotulações diferentes para o mesmo conjunto de pontos

o número de combinações para rotulação de N_a classes em um conjunto de N pontos é dado pelo número de Stirling de segunda ordem [12]

$$S(N, N_a) = \frac{1}{N_a!} \sum_{i=0}^{k-1} (-1)^i \begin{pmatrix} N_a \\ i \end{pmatrix} (N_a - i)^N. \tag{2.1}$$

Onde $\begin{pmatrix} N_a \\ i \end{pmatrix}$ representa a combinação de N_a tomados i a i. Por exemplo, para apenas 20 pontos dividindo em 4 classes, temos 45232115901 possibilidades a serem testadas. Isso torna extremamente proibitiva a busca exaustiva de cada combinação, visto que em aplicações normais é comum existirem conjuntos com tamanhos da ordem de mil ou até um milhão de pontos...

2.1.1 Critério de seleção das partições

Podemos analisar a "qualidade" de uma partição qualquer medindo o espalhamento dos dados nessa partição. Este procedimento equivale a medida de "proximidade" mencionada ainda a pouco. Uma maneira formal para realizar esta medida é através das matrizes de espalhamento de uma partição [37] (*scatter matrix* como é geralmente citado na literatura). Seja X um conjunto formado por N pontos $\mathbf{x}_i$, para $i = 1, 2, 3, ..., N$ dividido em N_a partições e cada partição possuindo n_i pontos. O vetor média de cada classe pode ser calculado como sendo

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_j^{(i)}, \tag{2.2}$$

onde $\mathbf{x}_j^{(i)}$ são os pontos do conjunto X que pertencem a partição i. A média total $\boldsymbol{\mu}$ do conjunto pode ser escrita como a soma ponderada das médias de cada partição

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N_a} n_i \boldsymbol{\mu}_i. \tag{2.3}$$

Define-se assim as matrizes de espalhamento, a matriz inter-grupo $\mathbf{W}$, entre-grupos $\mathbf{B}$ e total $\mathbf{T}$ que podem ser calculadas como mostram as equações de 2.4 a 2.6

$$\mathbf{W} = \sum_{i=1}^{N_a} \sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} - \boldsymbol{\mu}_i)(\mathbf{x}_j^{(i)} - \boldsymbol{\mu}_i)^t, \tag{2.4}$$

$$\mathbf{B} = \sum_{i=1}^{N_a} n_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^t, \tag{2.5}$$

$$\mathbf{T} = \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^t. \tag{2.6}$$

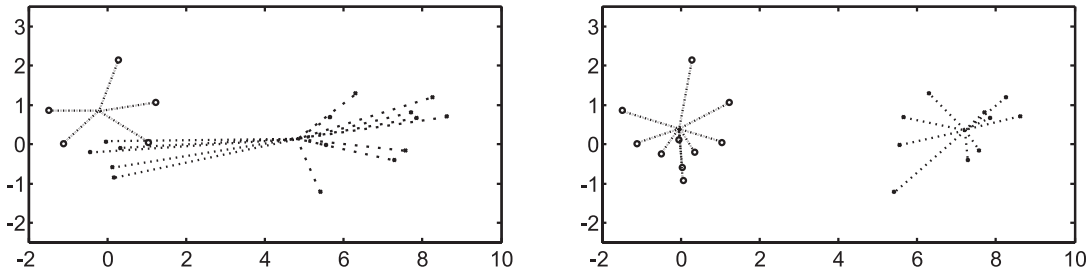
Pode-se mostrar que a variância total representada pela matriz T é o resultado da soma das variâncias inter-grupo e entre-grupos (representadas pelas matrizes $\mathbf{W}$ e $\mathbf{B}$), obedecendo a relação

$$\mathbf{T} = \mathbf{W} + \mathbf{B}. \tag{2.7}$$

Estas matrizes podem ser utilizadas para verificar se uma partição esta distribuída corretamente. Isto pode ser feito minimizando-se a distância entre os pontos de uma partição a média da mesma que equivale a minimizar o traço da matriz $\mathbf{W}$. A Figura 2.3 ilustra este critério.

Como pode-se observar, o particionamento inadequado gera diferenças maiores (maior variância) intra-grupos. Já no particionamento correto, as diferenças são menores. Outros critérios, como por exemplo maximizar o traço de $\mathbf{B}\,\mathbf{W}^{-1}$ [17] também são encontrados na literatura.

Nem todo tipo de agrupamento pode ser analisado em função das matrizes de



(a) Particionamento inadequado de um conjunto de dados

(b) Particionamento correto

Figura 2.3: Diferença entre diferentes particionamentos para o mesmo conjunto de pontos

espalhamento. Isto é fácil de se verificar quando temos conjuntos com espalhamentos mais complexos.

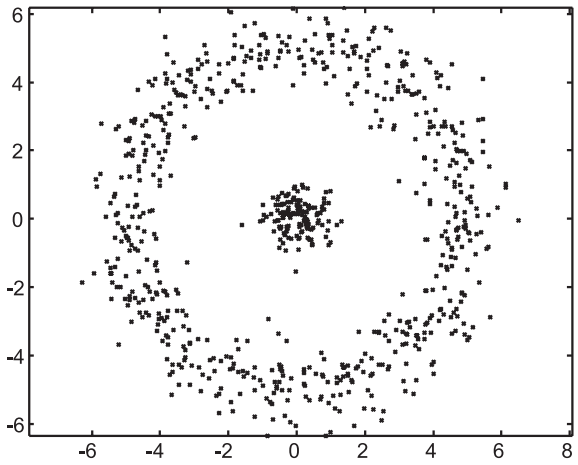


Figura 2.4: Conjunto com espalhamento complexo

Para o conjunto da Figura 2.4, podemos ver claramente que o particionamento correto seria agrupar os pontos do círculo central e os pontos do anel externo. Nesta situação, as médias das partições ficarão no mesmo lugar (centro do conjunto) e no mesmo lugar da média geral do conjunto. Isto faz com que o traço da matriz **W** seja um valor maior do que seria se posicionarmos o conjunto ao meio.

Com base neste exemplo, chegamos a conclusão de que um método de *clustering* para realizar o particionamento correto deve levar em consideração outros fatores que não sejam apenas as variâncias das partições. A utilização das matrizes de espalhamento ajudam no particionamento de conjuntos de dados simples, ou que tenham uma distribuição isotrópica de suas partições.

Quando temos o particionamento correto, as partições passam a representar *clusters* ou classes do conjunto de dados. A partir daí, é possível modelá-las estatisticamente e realizar a classificação de outros pontos fora do conjunto original. A este procedimento, dar-se o nome de classificação estatística. Existem muitas ferramentas que realizam esta tarefa. Na seção seguinte analisaremos o método utilizado neste trabalho.

2.2 Decisão Bayesiana e classificação estatística

Para realizar a tarefa de decidir a que classe um dado qualquer pertence, devemos utilizar uma função que nos permita comparar a pertinência deste dado a cada uma das classes. Como já citado no capítulo 1, esta função denomina-se função discriminante. Neste trabalho a decisão entre pertencer a uma classe ou outra será tomada com base na probabilidade de um dado pertencer a cada uma das classes. Para o caso de termos N_a classes denota-se $P(\mathbf{x}|w_i)$ a probabilidade de um vetor de dados $\mathbf{x}$ pertencer a classe w_i (dentre as N_a classes). Esta probabilidade condicional implica em conhecermos a função densidade de probabilidade que modela cada classe w_i. Para que possamos inferir sobre a pertinência de um elemento do conjunto à uma classe qualquer, temos que conhecer $P(w_i|\mathbf{x})$. Recorrendo a expressão de Bayes para o cálculo da probabilidade condiocionada (equação 2.8) temos:

$$P(w_i|\mathbf{x}) = \frac{P(\mathbf{x}|w_i)\,P(w_i)}{P(\mathbf{x})} \tag{2.8}$$

onde $P(w_i)$ é a probabilidade *a priori* da classe w_i e $P(\mathbf{x})$ a probabilidade de ocorrência de x, chamado de evidência que independe das classes, como podemos ver na equação 2.9 a seguir

$$P(\mathbf{x}) = \sum_{i=1}^{N} P(\mathbf{x}|w_i)\,P(w_i) \tag{2.9}$$

Utilizando a equação 2.8 podemos calcular a probabilidade de um elemento $\mathbf{x}$ pertencer a uma classe ou outra. Utilizando esta probabilidade podemos decidir a que classe este ponto pertence. Ou seja,

$$\text{Pertence a classe } w_i \text{ se} \quad P(w_i|\mathbf{x}) > P(w_j|\mathbf{x}) \quad \text{para todo } i \neq j \tag{2.10}$$

como $P(\mathbf{x})$ não depende da classe, podemos decidir por uma classe w_i se

$$P(\mathbf{x}|w_i)\,P(w_i) > P(\mathbf{x}|w_j)\,P(w_j) \quad \text{para todo } i \neq j. \tag{2.11}$$

Neste trabalho iremos estimar $P(w_i|\mathbf{x})$ para cada uma das classes. A classificação será feita aplicando-se a equação 2.11 para cada elemento do conjunto de dados e relacionando os mesmos a uma das classes existentes.

A Figura 2.5 mostra duas distribuições, cada uma correspondendo a distribuição de uma classe. O ponto onde as duas se igualam é o ponto de decisão onde uma amostra $\mathbf{x}$ passa de uma classificação para outra. As probabilidades *a priori* multiplicam cada função densidade de probabilidade, aumentando ou diminuindo a probabilidade de ocorrência de uma classe.

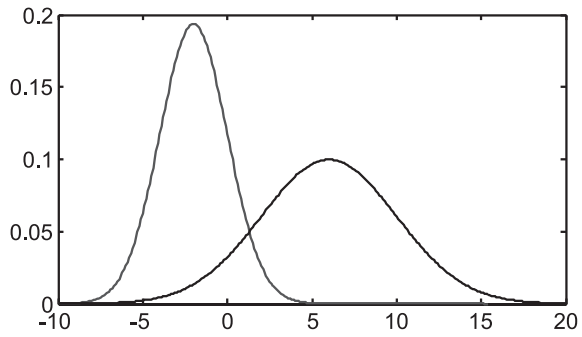


Figura 2.5: Distribuições de probabilidade de duas classes de uma mesma variável aleatória

A Figura 2.5 ilustra o caso de decisão entre duas classes com a variável aleatória (VA) uni-dimensional. Neste caso, o ponto de decisão é um valor específico da VA. Quando a VA é multi-dimensional, a "superfície" de decisão pode se tornar bastante complexa. A Figura 2.6 ilustra um caso bi-dimensional. A linha sinuosa na figura representa a linha de decisão (onde as probabilidades são iguais). A área ao lado direito da linha de decisão ficam os pontos (x, y) que pertencem a uma classe e ao lado esquerdo ficam os pontos que pertencem a outra classe.

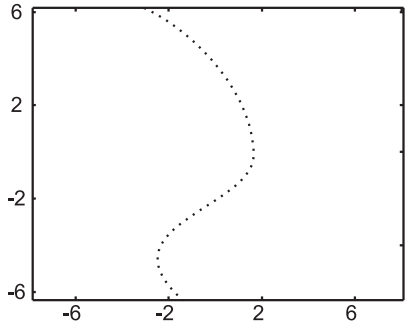


Figura 2.6: Superfície de decisão entre duas classes de uma VA bi-dimensional

2.3 Distribuições de probabilidade gaussianas

Em estatística, uma das distribuições de probabilidade mais importantes e utilizadas é a distribuição gaussiana. Sua importância se deve a um teorema denominado teorema do limite central [68]. Em linhas gerais este teorema nos diz que, em certas condições bastante comuns, a soma de variáveis aleatórias independentes é uma outra VA que tende para uma distribuição gaussiana. Isto faz com que a distribuição gaussiana seja uma distribuição bastante comum em processos práticos, que em geral são formados por combinações de diversas variáveis aleatórias independentes [1]. Uma distribuição gaussiana para um vetor aleatório $\mathbf{x}$ (também chamada de distribuição normal, ou gaussiana multivariada) é expressa como mostra a equação 2.12

$$N(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\,\pi)^D\,|\boldsymbol{\Sigma}|}}\,\exp(\frac{1}{2}\,(\mathbf{x}-\boldsymbol{\mu})\,\boldsymbol{\Sigma}^{-1}\,(\mathbf{x}-\boldsymbol{\mu})^t), \tag{2.12}$$

onde D é a dimensão do vetor de dados $\mathbf{x}$, neste caso, $x \in \mathbb{R}^D$. $\boldsymbol{\mu}$ e $\boldsymbol{\Sigma}$ são a média e a matriz de covariância respectivamente da distribuição.

A Figura 2.7 mostra o gráfico de uma distribuição gaussiana de duas variáveis ($D = 2$). O vetor média da distribuição é o ponto onde a probabilidade da VA é máxima e ocorre no topo da guassiana.

A matriz de covariância de uma gaussiana é uma matriz simétrica. Os elementos desta matriz medem a relação entre cada variável, sendo que os elementos da diagonal principal são o desvio padrão de cada variável da distribuição. Para o caso onde as variáveis envolvidas são independentes, obtemos uma matriz diagonal dada por

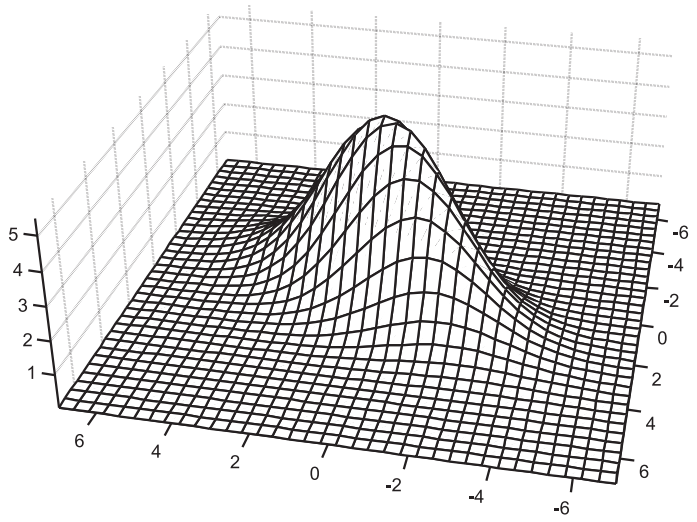


Figura 2.7: Gráfico de uma distribuição gaussiana

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_n^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_n^2 \end{bmatrix} \tag{2.13}$$

onde cada σ_i^2 representa a variância individual de cada variável.

2.3.1 Análise da covariância

Como vimos, os elementos da matriz de covariância de uma gaussiana representam o comportamento das variáveis em questão. Este comportamento pode ser bem ilustrado observando-se o gráfico de contorno de uma gaussiana, mostrado na Figura 2.8

Como podemos observar, os contornos da gaussiana são elipses e os eixos pontilhados representam os eixos principais da mesma. Quando a matriz de covariância não é diagonal, significa que existe uma dependência ou relação entre as variáveis. Na figura, isso corresponde a inclinação das elipses mostrando a relação entre as variáveis (quando uma aumenta, outra aumenta, etc...).

O fato da matriz de covariância ser simétrica faz com que esta tenha autovalores reais e não negativos. Neste caso, os autovalores representam os tamanhos dos eixos

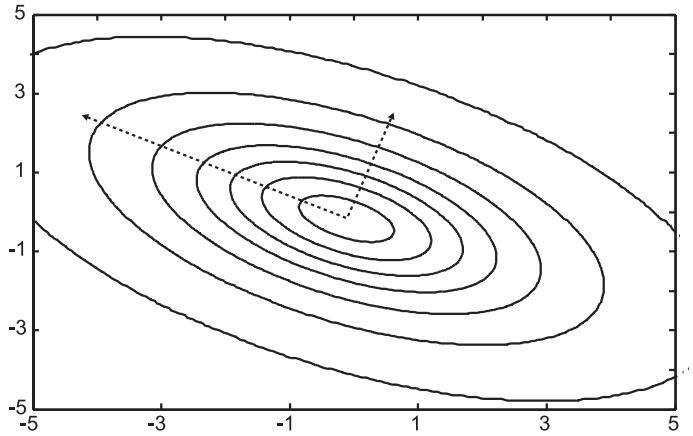


Figura 2.8: Gráfico de contorno de uma gaussiana

das elipses que formam os contornos da gaussiana. Os autovetores correspondentes a cada autovalor serão vetores nas direções de cada eixo. Este fato é importante porque nos fornece uma maneira de medir a "dispersão" dos dados de uma distribuição gaussiana. Esta dispersão pode ser medida pelo produto dos autovalores que, de fato, nos dá o determinante da matriz. Desta maneira o determinante da matriz de covariância nos da uma idéia da "dispersão" da VA em torno da sua média.

2.3.2 Mudança de escala

Será feita a seguir uma análise da estimativa da matriz de covariância mediante uma mudança de escala nos dados. Esta análise é importante porque, em análise de *clustering* é muito importante que o método seja invariante a mudanças de escala. Para que se possa testar esta invariância, é preciso observar o comportamento da matriz de covariância nesta situação de mudança de escala.

Uma mudança de escala pode ser vista como uma transformação linear do tipo

$$\mathbf{y} = \mathbf{A}\,\mathbf{x} + \mathbf{b} \tag{2.14}$$

onde $\mathbf{A}$ é uma matriz qualquer inversível e $\mathbf{b}$ um vetor de deslocamentos qualquer. A matriz $\mathbf{A}$ deve ser inversível porque, como será visto, alguns métodos requerem a inversão da matriz de covariância, e isto implicará na inversão também da matriz $\mathbf{A}$.

Para obtermos a matriz de covariância de um modelo gaussiano, precisamos

estimar seu valor. A partir de um conjunto de amostras de uma distribuição qualquer, é possível estimar o valor da matriz de covariância $\mathbf{\Sigma}$) de acordo com a equação 2.15 [68]

$$\mathbf{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu}) (\mathbf{x}_i - \boldsymbol{\mu})^t, \tag{2.15}$$

onde $\mathbf{x}_i$ representa cada uma das N amostras e $\boldsymbol{\mu}$ o vetor média do conjunto que é calculado como

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x_i}. \tag{2.16}$$

Utilizando esta estimativa, podemos observar como ficaria a estimativa da matriz de covariância da VA Y que é resultado da transformação linear em X. A média da VA Y pode ser calculada pela mesma estimativa dada para X, ou seja

$$\boldsymbol{\mu_y} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{A}\, \mathbf{x_i} + \mathbf{b}.$$

o que leva a

$$\boldsymbol{\mu_y} = \mathbf{A} \left(\frac{1}{N} \sum_{i=1}^{N} \mathbf{A}\, \mathbf{x_i} \right) + \mathbf{b}.$$

O termo ente parênteses é exatamente a média da VA X oque finalmente resulta em

$$\boldsymbol{\mu_y} = \mathbf{A}\, \boldsymbol{\mu_x} + \mathbf{b}, \tag{2.17}$$

ou seja, a média da VA Y sofre a mesma transformação que X.

Agora analisemos oque ocorre com a matriz de covariância. Da mesma forma, aplicamos a estimativa para a variável Y.

$$\mathbf{\Sigma_y} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{A}\, \mathbf{x}_i + \mathbf{b} - (\mathbf{A}\, \mu + \mathbf{b}))(\mathbf{A}\, \mathbf{x}_i + \mathbf{b} - (\mathbf{A}\, \mu + \mathbf{b}))^t.$$

Isolando $\mathbf{A}$ obtemos

$$\mathbf{\Sigma_y} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{A}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^t\mathbf{A}^t$$

$$\mathbf{\Sigma_y} = \mathbf{A}\left(\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^t\right)\mathbf{A}^t.$$

Novamente o termo entre parênteses corresponde a matriz de covariância de X, oque resulta em

$$\mathbf{\Sigma_y} = \mathbf{A}\,\mathbf{\Sigma_x}\,\mathbf{A}^t \tag{2.18}$$

Desta maneira, podemos escrever a média e matriz de covariância de uma VA Y como função linear de outra VA X que sofreu uma transformação linear. Como dito, isto será bastante útil posteriormente na análise de invariância do método proposto neste trabalho.

2.4 Misturas de Gaussianas

Uma maneira de modelar distribuições complexas consiste em utilizar misturas de gaussianas. Esta ferramenta vem sendo utilizada em muitas técnicas de classificação de padrões ou métodos estatísticos para tratamento de dados [28, 19]. Talvez a maneira mais tradicional de estimar os parâmetros de uma mistura de gaussianas seja utilizando o algoritmo EM [15] citado na introdução deste trabalho e detalhado nas seções seguintes. Na literatura, existem muitos trabalhos que utilizam misturas de gaussianas aplicadas à *clustering* [92, 72]. Em alguns trabalhos [66] um enfoque bayesiano é descrito, onde uma heurística de penalidades é aplicada para obtenção dos parâmetros da mistura. Outros trabalhos [50, 28] apresentam enfoques diferentes para solucionar o problema. É comum a utilização de um critério de informação para resolver o problema de modelamento com misturas. Bozdogan [7] propõe um método baseado em um critério de informação e da uma idéia de como estimar o número de gaussianas envolvidas estabelecendo limites superior e inferior para este número.

Uma mistura de gaussianas é definida como uma distribuição cuja função densidade de probabilidade é dada por uma soma ponderada de gaussianas como

mostrado na equação 2.19

$$p(\mathbf{x}) = \sum_{i=1}^{N_g} P_i \, \mathrm{N}(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{2.19}$$

onde $p(\mathbf{x})$ é a função densidade de probabilidade que se deseja modelar, N_g é o número de gaussianas presentes na mistura e P_i são as probabilidades *a priori* de cada gaussiana, satisfazendo a seguinte condição

$$\sum_{i=1}^{N_g} P_i = 1. \tag{2.20}$$

Uma distribuição normal com vetor média $\boldsymbol{\mu}_i$ e matriz de covariância $\boldsymbol{\Sigma}_i$, representada por $\mathrm{N}(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, é dada pela equação 2.12.

Para obtermos o modelo da mistura de gaussianas, um outro requisito necessário é o número de gaussianas que farão parte da mistura. No caso de modelar várias classes, também é importante o número de gaussianas em cada classe. Além desta quantidade, para modelar todas as classes presentes nos dados, precisamos dos parâmetros de cada gaussiana como média, probabilidade *a priori* e matriz de covariância.

2.5 Estimação de distribuições de probabilidade

Para que se possa realizar uma análise estatística de um conjunto de dados, precisamos obter a distribuição de probabilidade que os gerou. De posse dessa distribuição, podemos calcular a probabilidade de ocorrência de cada ponto do conjunto e até mesmo de pontos que não pertencem ao mesmo.

Existem os mais diversos métodos para estimar estas distribuições. Podemos dividir basicamente os métodos em métodos paramétricos e métodos não paramétricos. Nos métodos paramétricos, os dados são utilizados para ajustar os parâmetros de uma distribuição definida (uma gaussiana por exemplo). Nos métodos não paramétricos, os dados puros são utilizados diretamente como base para o cálculo da sua probabilidade.

2.5.1 Algoritmo EM

O algoritmo *Expectation Maximization* ou EM, como é comumente citado na literatura, consiste em um algoritmo para estimar parâmetros de uma mistura de gaussianas que melhor modele um conjunto qualquer de dados [15] (método paramétrico). Basicamente o algoritmo encontra um vetor de parâmetros $\mathbf{\Theta}$ de uma função densidade de probabilidade. Em resumo temos que encontrar $P(\mathbf{x}|\mathbf{\Theta})$, que é a probabilidade de ocorrer um vetor $\mathbf{x}$ dados os parâmetros $\mathbf{\Theta}$). No caso gaussiano, $\mathbf{\Theta}$ é formado por médias, matrizes de covariância e probabilidades *a priori*. O algoritmo EM encontra os parâmetros da distribuição maximizando a verossimilhança dos dados expressa pela equação 2.21.

$$L(\mathbf{\Theta}) = \prod_{i=1}^{N} P(\mathbf{x}_i|\mathbf{\Theta}) \tag{2.21}$$

Em geral não se maximiza diretamente $L(\Theta)$ e sim o seu logaritmo. Por ser uma função estritamente crescente, o logaritmo da verossimilhança terá um máximo no mesmo ponto que teria a verossimilhança em si, e torna o problema de maximização mais simples de se trabalhar, pois transforma o produtório em um somatório formando a expressão

$$\log(L(\mathbf{\Theta})) = \sum_{i=1}^{N} \log(P(\mathbf{x}_i|\mathbf{\Theta})). \tag{2.22}$$

Este critério de maximização da verossimilhança, não é o único critério que pode ser utilizado. Akaike [2] propõe uma alternativa para o critério da verossimilhança que leva em conta o número de parâmetros utilizados. Este critério é comumente denominado de critério de informação de Akaike (AIC). Posteriormente a este critério, outros autores desenvolveram outros critérios [7] que também podem ser utilizados para o cálculo dos parâmetros ideais.

Para distribuições formadas por misturas de gaussianas a maximização de $L(\Theta)$ deve ser feita respeitando-se a restrição nos valores das probabilidades *a priori* P_i como mostrado na equação 2.20. Para isto, alguma técnica de aplicação de restrições deve ser introduzida, como por exemplo o funcional de Lagrange.

A verossimilhança pode ser vista como a probabilidade de ocorrer todos os pontos dados se eles forem modelados por uma distribuição com parâmetros Θ e sua ocorrência for independente. Em estatística, a probabilidade de ocorrer dois eventos independentes é dada pelo produto das probabilidades

$$P(x_1, x_2) = P(x_1)\, P(x_2) \tag{2.23}$$

De maneira geral se tivermos N eventos temos

$$P(x_1, x_2, \ldots, x_n) = P(x_1)\, P(x_2)\, \ldots\, P(x_N). \tag{2.24}$$

Neste sentido, dado um vetor de parâmetros formados por exemplo por uma média e uma variância (no caso gaussiano), podemos formar uma distribuição de probabilidades. Após isso, dado um conjunto de observações X, a probabilidade destes pontos terem ocorrido desta distribuição é dada pela verossimilhança (o produtório significa exatamente a ocorrência de todos os pontos). Podemos então resumir da seguinte maneira; A função de verossimilhança $L(\mathbf{\Theta})$ é a probabilidade de ocorrerem os N pontos dados sabendo-se que a distribuição tem parâmetros $\mathbf{\Theta}$. Desta maneira, dados um conjuntos de pontos podemos especular que, como os pontos já são dados, a probabilidade destes ocorrerem é máxima (pois dentre todas as possibilidades, estes pontos foram os que ocorreram). Assim, se calcularmos quais parâmetros $\mathbf{\Theta}$ maximizam a verossimilhança, estamos encontrando a distribuição que provavelmente forneceu os dados em questão.

A solução analítica para este problema de maximização da verossimilhança consiste em encontrar o máximo de $\log(\mathbf{\Theta})$ (derivando e igualando a zero) solucionando o sistema de equações dado por

$$\frac{\partial L(\mathbf{\Theta})}{\partial \Theta_i} = 0. \tag{2.25}$$

Este sistema consiste de várias equações não lineares (uma para cara parâmetro), e no caso de mistura de gaussianas D-dimensionais, as equações são matriciais, sendo os parâmetros definidos pelas médias das gaussianas e pelas suas matrizes de

covariância (supondo as probabilidades *a priori* conhecidas).

Embora a solução para este sistema possa ser implementada utilizando o método numérico iterativo, ainda assim o custo computacional seria proibitivo. Neste sentido, o algoritmo EM é uma ferramenta alternativa para a solução do problema. Por se tratar de um método de otimização, o algoritmo EM está sujeito a encontrar máximos locais ou apresentar problemas de convergência. Recentemente, Figueiredo e Jain [38] propuseram um novo método de estimação baseado no algoritmo EM que procura resolver este problemas utilizando técnicas de *simulated annealing*[1].

2.5.1.1 Descrição do algoritmo

O algoritmo EM é um algoritmo iterativo utilizado para minimizar a função de verossimilhança. Partindo de uma solução inicial para o vetor de parâmetros e melhorar esta solução a cada iteração. A atualização dos parâmetros é feita calculando-se os valores que resolvem o sistema da equação 2.25. De fato, para misturas de gaussianas, obter uma expressão isolada para os parâmetros nestas equações é muito difícil. A solução é realizar uma aproximação (descrita detalhadamente na literatura [15, 14]) e calcular uma função mais simples que aproxime o valor das verossimilhança, dados os vetores de dados atuais. Este passo é denominado na literatura de *Expectation*. O próximo passo é encontrar quais os parâmetros que maximizam a função encontrada na fase de *expectation*. Este passo recebe o nome de *maximization*.

Em seu trabalho, Dempster et al. [15] mostram que, para o caso de misturas de gaussianas com parâmetros de média, matrizes covariância e probabilidades *a priori*, estes são calculados iterativamente utilizando as equações 2.26

[1] *annealing* ou recozimento é um termo utilizado em otimização para designar algoritmos motivados por teorias da área de termodinâmica onde a busca é feita inicialmente com uma alta entropia (buscas de longo alcance) e a medida que o algoritmo evolui, a busca vai sendo estreitada [45]

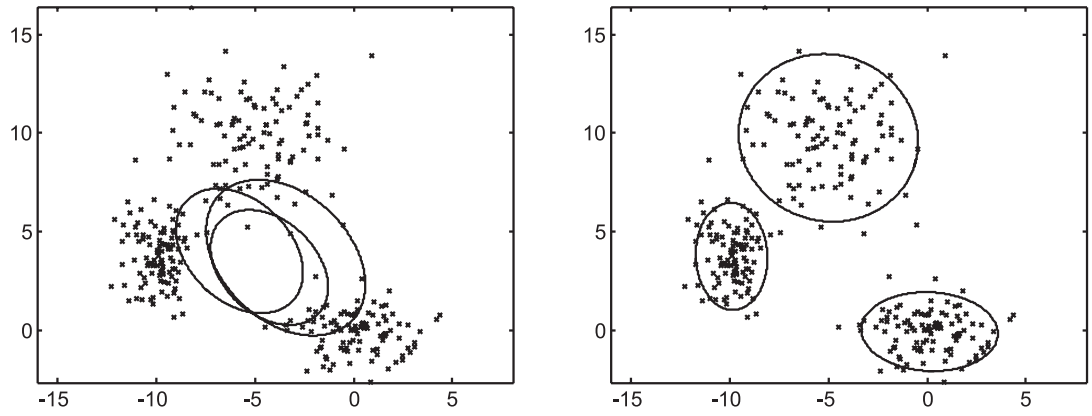
$$
\begin{aligned}
P_i^{(k+1)} &= \frac{\sum_{j=1}^{N} h_i^{(k)}(j)}{N} \\
\boldsymbol{\mu}_i^{(k+1)} &= \frac{\sum_{j=1}^{N} h_i^{(k)}(j)\mathbf{x}_j}{\sum_{j=1}^{N} h_i^{(k)}(j)} \\
\boldsymbol{\Sigma}_i^{(k+1)} &= \frac{\sum_{j=1}^{N} h_i^{(k)}(j)(\mathbf{x}_j-\boldsymbol{\mu}_i)^t(\mathbf{x}_j-\boldsymbol{\mu}_i)}{\sum_{j=1}^{N} h_i^{(k)}(j)}
\end{aligned}
\tag{2.26}
$$

onde os h_i representam as probabilidades a *posteriori* calculadas como

$$
h_i^{(k)}(j) = \frac{P_i^{(k)} p(\mathbf{x}_j|\boldsymbol{\mu}_i^{(k)}, \boldsymbol{\Sigma}_i^{(k)})}{\sum_{t=1}^{N_c} P_t^{(k)} p(\mathbf{x}_j|\boldsymbol{\mu}_t^{(k)}, \boldsymbol{\Sigma}_t^{(k)})}. \tag{2.27}
$$

Esta solução maximiza o logaritmo da verossimilhança sujeitas as restrições nas probabilidades *a priori* P_i mostradas na equação 2.20.

A Figura 2.9 mostra a inicialização dos parâmetros e a convergência em um exemplo de uso do algoritmo EM.



(a) Inicialização (b) Após 100 iterações

Figura 2.9: Exemplo de utilização do algoritmo EM

No exemplo são utilizados dados gerados por uma mistura de três gaussianas. As elipses representam o contorno de cada gaussiana. Observe que na Figura 2.9(a), as gaussianas são inicializadas de maneira aleatória, não modelando corretamente os dados. Ao final do processo, o algoritmo converge para os parâmetros da mistura e as gaussianas se ajustam de maneira a modelar corretamente os dados.

A idéia por traz do algoritmo EM, de realizar as fases de *expectation* e

maximization é utilizada na solução de diversos problemas de otimização de parâmetros de distribuições. Existem várias maneiras de se chegar na otimização dos parâmetros e o grande problema aparece quando se tem de manipular a expressão do logaritmo da verossimilhança, que no caso das misturas de gaussianas aparece na forma do logaritmo de uma soma. Em 1998, Minka [65] apresentou uma abordagem de limite inferior de modo a resolver este problema.

Mesmo com as simplificações envolvidas e devido ao fato de ser um algoritmo iterativo, o algoritmo EM possui um custo computacional considerável e nem sempre encontra um máximo global para verossimilhança, exigindo assim, outras técnicas de estimação de parâmetros de distribuições de probabilidades.

2.5.2 Janelas de Parzen

Os métodos de modelamento de dados como o EM, são conhecidos como métodos paramétricos. Nestes métodos, o objetivo é obter um conjunto de parâmetros de uma distribuição de probabilidade que melhor modela os dados fornecidos. Existe ainda uma abordagem denominada não-paramétrica que encontra ou estima a distribuição de probabilidade de um conjunto de dados utilizando os próprios pontos como formadores da distribuição [16]. Os métodos não-paramétricos possuem a vantagem de não necessitarem de algoritmos para cálculo de parâmetros de uma distribuição, porém exigem em geral esforço computacional no cálculo da probabilidade. No caso de métodos paramétricos, o esforço computacional é localizado no cálculo dos parâmetros e uma vez calculados os parâmetros, o cálculo da probabilidade de um dado $\mathbf{x}$ pode ser facilmente realizado apenas avaliando a expressão da probabilidade no ponto $\mathbf{x}$. No exemplo gaussiano, o esforço computacional se concentra no cálculo dos parâmetros de média e variância, o cálculo da probabilidade se resume então a calcular o valor da gaussiana no ponto $\mathbf{x}$. Os métodos não-paramétricos, não envolvem o cálculo de parâmetros, porém para o cálculo da probabilidade de $\mathbf{x}$, em geral é necessária a utilização do valor de todas as amostras, aumentando assim, o custo computacional.

Uma importante abordagem não-paramétrica para o cálculo de distribuições de probabilidade são as janelas de Parzen [16]. Outros métodos para estimação não

paramétrica utilizam janelas de Parzen [9, 4] e outras variações do método, que devem ser consideradas em situações de estimações específicas. As janelas de Parzen estimam a probabilidade baseando-se apenas em amostras da distribuição. O método parte do princípio de que a probabilidade uma variável aleatória $\mathbf{x}$ estar em uma região $\mathcal{R}$ é dada por

$$P = \int_{\mathcal{R}} p(\mathbf{x})\,\mathrm{d}\mathbf{x}. \tag{2.28}$$

Neste caso, podemos estimar essa probabilidade sorteando n pontos e medindo a taxa k/n sendo k o número de pontos que estão dentro da região $\mathcal{R}$. Se considerarmos n suficiente grande, obtemos

$$\int_{\mathcal{R}} p(\mathbf{x})\,\mathrm{d}\mathbf{x} \simeq p(\mathbf{x})\,V \tag{2.29}$$

onde V é o volume da região em questão [16]. Dessa maneira, temos como estimar a probabilidade $p(\mathbf{x})$ como

$$p(\mathbf{x}) \simeq \frac{k/n}{V} \tag{2.30}$$

De fato, quanto menor o V melhor a aproximação, porém, em casos práticos, quanto menor for o volume V, menos provável será de ocorrer um evento contado por k. A solução é usar um volume que seja uma função de n de maneira que quando n tender a infinito, V tenda para zero [16].

Esta abordagem pode ser utilizada para estimar a probabilidade de uma região contendo um ponto $\mathbf{x}$. Se considerarmos que cada ponto x_i de um conjunto de dados é o centro de uma região, podemos estimar a probabilidade de um outro ponto qualquer $\mathbf{x}$ utilizando a equação 2.30 calculando o valor de k para os n pontos dados.

Consideremos um hipercubo de dimensão D e volume $V = h^D$. Podemos calcular quantos pontos existem dentro deste hipercubo definindo uma função chamada de "janela" dada por

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |\mathbf{x}| \leq 1/2 \\ 0 & \text{caso contrário} \end{cases} \tag{2.31}$$

Nesta equação, $\varphi(\mathbf{x})$ será 1 se o ponto $\mathbf{x}$ cair dentro do hipercubo centrado na origem com lado 1 (janela com raio 1/2). Desta maneira a quantidade k de pontos dentre os dados $\mathbf{x}_i$ dentro de um hipercubo centrado em $\mathbf{x}$ e com lado h pode ser dada por

$$k = \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \tag{2.32}$$

Substituindo 2.32 em 2.30 obtemos uma estimativa da probabilidade de um ponto $\mathbf{x}$ como sendo

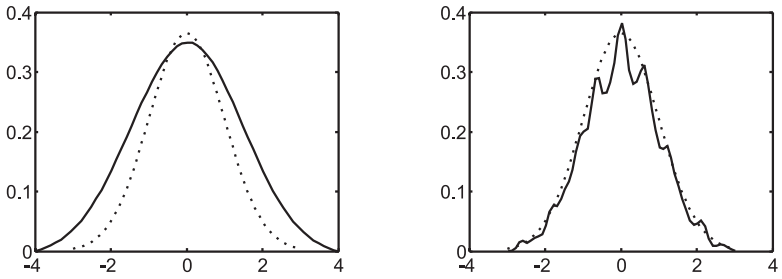
$$p(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \tag{2.33}$$

De maneira geral, a estimativa da densidade de probabilidade dada pela equação 2.33 pode utilizar outras funções janela que não sejam hipercubos. A condição é que a densidade de probabilidade resultante seja não negativa e tenha integral em todo o espaço igual a 1. Estas condições podem ser satisfeitas se $\varphi(\mathbf{x})$ também as satisfizer. A Figura 2.10 mostra três estimativas feitas para um conjunto de dados utilizando como funções janela, gaussianas com média zero e variância 1.

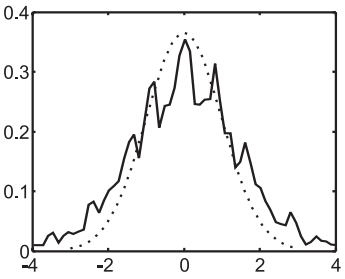
Como podemos observar, para valores maiores de h, obtém-se uma estimativa mais suave, enquanto que para valores muito baixos, obtemos estimativas mais pontuais.

2.5.2.1 Problema de dimensionalidade

As estimativas utilizando janelas de Parzen tem sua grande vantagem no fato de não termos que conhecer a forma da distribuição para estimá-la. Utilizando-se de um número suficiente de pontos, podemos aproximar qualquer tipo de distribuição. A grande desvantagem é que esse número de pontos pode vir a ser muito grande para que se obtenha uma boa estimativa. Este requisito sobe exponencialmente com a dimensão dos dados, levando as vezes a impossibilidade de se obter uma solução. Este problema é conhecido como "maldição da dimensionalidade" e está



(a) $h = 1$ (b) $h = 0.5$



(c) $h = 0.1$

Figura 2.10: Exemplo de densidades de probabilidades estimadas com valores diferentes de h

presente em problemas onde a dimensão dos dados envolvidos é alta. Ao se requerer um número de pontos muito grande para se obter uma boa aproximação para a distribuição em questão, nos deparamos com o problema do custo computacional. O custo envolvido para calcular uma simples probabilidade exige que seja realizado um somatório envolvendo todos os pontos. Em aplicações práticas e de altas dimensões este custo pode se tornar proibitivo.

2.6 Quantização vetorial

Quantização vetorial é uma técnica em que um conjunto de dados formado por N vetores de dimensão D é dividido em regiões em que, para cada região, é escolhido um único vetor representante [33]. Nesta técnica, o conjunto de dados $\mathbf{X}$ é organizado em forma de tabela onde cada coluna representa uma dimensão e cada linha representa um vetor completo. Na quantização vetorial, escolhe-se N_a representantes para quantizar todo o conjunto, e o resultado será um outro conjunto $\mathbf{W}$ com N_a elementos (vetores) de mesma dimensão D que os vetores do conjunto $\mathbf{X}$ em questão. Estes N_a vetores serão tidos como os centros das regiões em que o

conjunto foi dividido. A escolha dos N_a centros pode ser feita de diversas formas. As técnicas mais conhecidas são o *K-means* [53] e as redes neurais competitivas [23].

Obtidos os N_a centros, realiza-se a classificação de cada vetor do conjunto de dados. Classificar um vetor do conjunto de dados significa atribuir a este vetor uma única classe dentre as N_a escolhidas (representadas pelos N_a vetores quantizados). Esta atribuição pode ser feita com base em diversas regras. Neste trabalho, para fins de quantização vetorial, utilizamos o classificador de distância mínima (distância Euclidiana). Este tipo de classificador atribui-se uma classe w_j dentre todas as N_a classes, ao vetor $\mathbf{x}_i$ dentre todos os N vetores do conjunto de dados $\mathbf{X}$. Esta atribuição ocorre de modo que a distância $d(\mathbf{w}_j, \mathbf{x}_i)$ seja a menor dentre todos os elementos em $\mathbf{W}$ sendo $\mathbf{w}_j$ o vetor representante da classe w_j. Formalmente temos;

$$j = \min_k \{d(\mathbf{w}_k, \mathbf{x}_i)\} \tag{2.34}$$

onde j é o índice da classe atribuída ao vetor $\mathbf{x}_i$ e $d(\mathbf{x}, \mathbf{y})$ é a distância Euclidiana entre os vetores $\mathbf{x}$ e $\mathbf{y}$ dada por

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})(\mathbf{x} - \mathbf{y})^t} \tag{2.35}$$

O objetivo das técnicas de quantização vetorial são, em geral, minimizar alguma função custo relacionada ao conjunto de dados envolvido. Em muitos casos a função custo é o erro de quantização esperado que é definido por

$$e = \int ||\mathbf{x} - \mathbf{m}_c||\, p(\mathbf{x})\, d\mathbf{x} \tag{2.36}$$

onde $\mathbf{x}$ representa os pontos no espaço dos dados, $\mathbf{m}_c$ os centros, $p(\mathbf{x})$ a distribuição de probabilidade dos dados e a integral deve ser calculada considerando-se todo o espaço de x particionado nos $\mathbf{m}_c$ centros (por isso não é tão facilmente formalizada) [46].

2.6.1 Algoritmo *K-means*

O *k-means* [53] é um algoritmo que realiza a quantização vetorial em um conjunto de dados formado por vetores com descrito na seção 2.6. Em linhas gerais, o *K-means* consiste em gerar inicialmente N_a centros aleatoriamente distribuídos e classificar os dados de acordo com esses centros. Em seguida à classificação inicial, calcula-se os novos centros (fazendo a média dos pontos classificados) das classes atualmente calculadas. Este processo é repetido até que não se tenha variação na classificação dos dados ou uma quantidade suficiente de iterações tenha sido alcançada.

2.6.1.1 Cálculo dos centros

Os centros são calculados obtendo-se a média de todos os vetores pertencentes a classe. Isto faz com que o vetor média dos elementos pertencentes a classe seja utilizado como representante da mesma (seu centro). O cálculo formal é mostrado a seguir:

$$\mathbf{w}_j = \frac{1}{N_j} \sum_{i \in \mathbf{w}_j} \mathbf{x}_i \tag{2.37}$$

onde $\mathbf{w}_j$ é a média dos vetores e N_j é o número de elementos pertencentes a classe W. $\mathbf{x}_i$ é o i-ésimo elemento da classe.

O algoritmo completo é mostrado a seguir

1. Posiciona-se os N_a centros de maneira aleatória ($\mathbf{w}_j$, $j \in 1, 2, ..., N_a$)
2. Classificam-se cada um dos vetores $\mathbf{x}_i$ como pertencendo a uma classe $\mathbf{w}_j$
3. Calcula-se a média de cada classe $\mathbf{w}_j$
4. Atribui cada $\mathbf{w}_i$ à média de sua classe
5. Volta-se para o passo 2 (até que não ocorram mudanças na classificação)

Em geral o *K-means* requer que em cada iteração, todos os dados sejam classificados até que não se obtenha variações nesta classificação. Este procedimento torna o *k-means* um processo preciso porém lento e sujeito a mínimos locais.

2.6.2 Algoritmo de treinamento competitivo

Na quantização vetorial feita utilizando-se redes neurais competitivas [23], os N_a centros são encontrados de maneira adaptativa. Em cada iteração, os valores dos centros são ajustados de acordo com uma regra de atualização. Na topologia da rede neural competitiva os pesos da única camada existente na rede serão os N_a centros que serão calculados.

Dessa forma, a rede é constituída de uma única camada onde tem-se D entradas e N_a saídas (D é a dimensão dos dados e N_a o número de centros escolhido). Neste trabalho, N_a corresponde ao número de centros auxiliares escolhidos, cada um representando um centro auxiliar nos dados, como será visto nas seções seguintes. A topologia é mostrada na Figura 2.11. O número de entradas da rede é determinado pela dimensão dos vetores presentes no conjunto de dados, neste caso, a mesma dimensão dos dados a serem classificados.

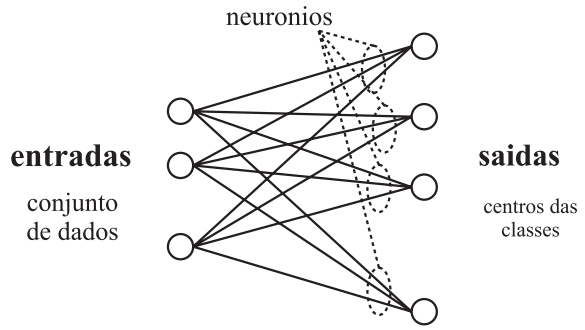


Figura 2.11: Exemplo de topologia de uma rede neural competitiva

O procedimento para treinamento da rede, e conseqüentemente cálculo dos centros, é o seguinte; Inicializa-se os pesos da rede aleatoriamente, em seguida, os padrões (vetores da tabela de dados) são apresentados a rede. Para cada padrão, é verificado qual neurônio possui menor distância Euclidiana entre seus pesos e o padrão apresentado. Este será o neurônio vencedor e será o único atualizado. Por esse motivo o processo é geralmente chamado de *"winner takes all"* (vencedor leva tudo). A atualização dos neurônios é feita de acordo com a expressão mostrada na equação 2.38.

$$\begin{aligned} \mathbf{w}_j(n+1) &= \mathbf{w}_j(n) + \Delta\mathbf{w}_j(n) \\ \Delta\mathbf{w}_j(n) &= \alpha\eta(\mathbf{x}_i - \mathbf{w}_j) \end{aligned} \tag{2.38}$$

onde $\mathbf{w}_j(n)$ é o neurônio vencedor da iteração n, $\Delta\mathbf{w}_c(n)$ é a diferença que será acrescentada ao neurônio vencedor, α é o coeficiente de aprendizado da rede e η um coeficiente utilizando para estabelecer a convergência. η é atualizado a cada iteração de forma a que tenhamos seu valor diminuído para evitar oscilações na convergência da rede. Em geral faz-se $\eta_{n+1} = \eta_n\tau$ onde τ é uma constante [33]. $\mathbf{x}_i$ é o padrão de entrada. O algoritmo é mostrado a seguir.

1. Posiciona-se os N_a centros de maneira aleatória
2. Escolhe-se um elemento $\mathbf{x}_i$ do conjunto de dados
3. Verifica-se qual o neurônio vencedor $\mathbf{w}_j$ para o elemento $\mathbf{x}_i$
4. Atualiza-se os pesos associados ao mesmo através da equação 2.38
5. Volta-se para o passo 2 (até que um critério de parada seja atingido)

Como mencionado anteriormente, Kohonen [46], em seu livro, mostra uma prova formal da convergência do algoritmo de treinamento competitivo. A regra de atualização dos pesos é baseada no gradiente da função custo mostrada na equação 2.36. O algoritmo de treinamento competitivo nem sempre converge de maneira ideal. O desempenho do algoritmo depende bastante da inicialização dos pesos da rede, ou seja da colocação dos centros iniciais. Uma ocorrência bastante comum durante a inicialização dos centros é a colocação de um centro distante do conjunto de maneira que o mesmo nunca seja vencedor durante o treinamento, ficando assim isolado e comprometendo o treinamento correto da rede. A Figura 2.12 ilustra uma situação onde este fato ocorre. Como podemos observar, o centro destacado esta isolado do conjunto de dados. Qualquer ponto do conjunto possui um centro mais próximo que não seja o centro do meio.

Haykin [33] sugere que os centros sejam inicializados em pontos aleatórios do próprio conjunto de dados, dessa maneira não há como existirem centros isolados e

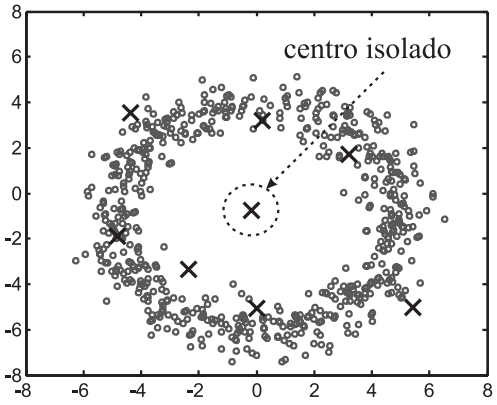
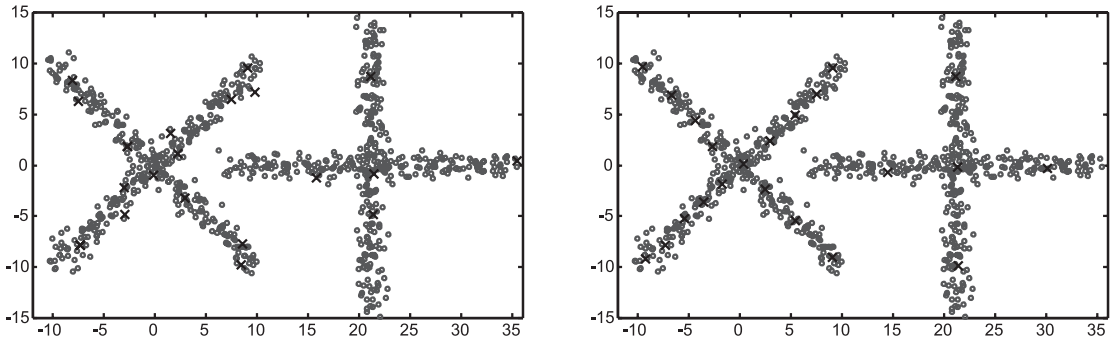


Figura 2.12: Exemplo de inicialização dos pesos onde ocorre isolamento

todos os centros vencem em alguma iteração, fazendo com que o algoritmo convirja ao final do processo.

Uma outra situação decorrente da inicialização dos centros com pontos do próprio conjunto é a ocorrência de *clusters* com muitos centros enquanto outros ficam com poucos, como mostrado na Figura 2.13.



(a) Inicialização (b) Após o treinamento

Figura 2.13: Exemplo de inicialização dos pesos onde ocorre *cluster* com excesso de centros

No contexto deste trabalho, são utilizadas técnicas de quantização vetorial para estabelecer centros auxiliares que serão mais tarde utilizados para estimar parâmetros de uma distribuição de probabilidades que irá modelar os dados. Em termos de resultado, tanto o *k-means* quanto as redes neurais competitivas fornecem valores de quantização bastante parecidos, podendo serem utilizados sem distinção.

2.7 Entropia e informação

Nas área do conhecimento que envolvem análise de dados, como é o caso de *clustering*, freqüentemente são utilizadas ferramentas de processamento da informação. Na análise de *clustering* temos que tratar com dados que representam ou contém alguma informação. Em teoria da informação, o conceito de entropia é bastante ligado ao conceito de medida de informação [68].

Entropia é uma medida que se aplica a uma variável aleatória (VA) ou a uma partição de uma variável aleatória. Ela nos dá de uma forma quantitativa, uma medida sobre a informação contida naquela partição ou VA. Uma forma de ilustrar este conceito consiste em imaginar uma moeda fictícia onde a probabilidade de se obter cara em uma jogada aleatória é p, logo a probabilidade de se obter coroa é de $1 - p$. A entropia desta VA pode ser dada por

$$H = -p \log(p) - (1 - p) \log(1 - p) \tag{2.39}$$

Esta medida corresponde a entropia de Shannon e será explicada mais a frente. Se fizermos o gráfico de $H(p)$ obtemos algo como mostrado na Figura 2.14.

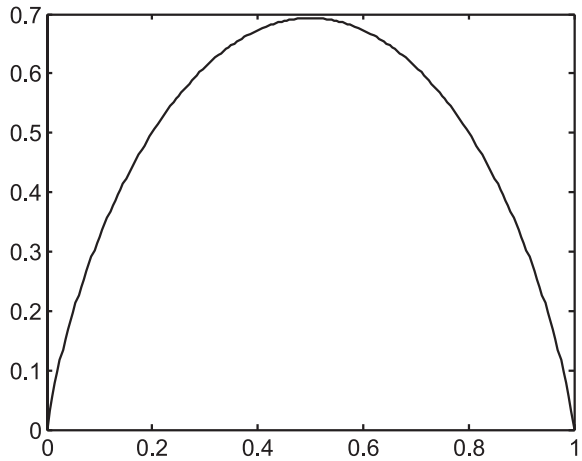


Figura 2.14: Gráfico de $H(p)$ em função de p

Como podemos observar, o máximo deste gráfico ocorre em $p = 0.5$. Este resultado nos diz que para uma moeda onde as probabilidades de se obter cara ou coroa seja igual a 0.5, temos a entropia máxima. De fato, esta é a situação onde temos menos informação sobre a VA. Se a probabilidade de ocorrer cara fosse 0.3

e a probabilidade de ocorrer coroa fosse de 0.7 teríamos uma informação sobre a moeda, sabendo que esta tende a dar coroa (ou tende a não dar cara). No limite, se tivermos probabilidade de dar cara igual a 0 e probabilidade de dar coroa igual a 1 (ou vice-versa), a entropia e 0 e nós temos informação máxima, ou seja a nossa incerteza sobre a VA é 0.

Utilizando o exemplo da moeda para ilustrar o significado da entropia podemos estabelecer uma ligação entre a entropia e a incerteza sobre a VA em questão. Utilizando este principio, surge um método muito utilizado em estatística denominado método da máxima entropia, muito utilizado em diversos problemas inclusive estimação de distribuições de probabilidade [64].

2.7.1 Entropia de Shannon

O conceito de entropia foi primeiro utilizado na termodinâmica e posteriormente dada uma interpretação probabilística por Boltzmann em 1877 seguido por Plank em 1906. Shannon [79] foi quem aplicou o conceito de entropia à estudos em teoria da codificação e serviu de base para a atual teoria da informação. A entropia de Shannon para uma VA discreta X é calculada como sendo

$$H(X) = -\sum_i p_i \log(p_i) \tag{2.40}$$

onde os p_i's são as probabilidades de cada evento i da partição ou VA X em questão. No caso de uma VA contínua, a entropia de Shannon é calculada, com base na distribuição de probabilidade $p(x)$ da VA como sendo

$$H(X) = -\int_{\mathfrak{L}} p(x) \log(p(x)) \, \mathrm{d}x \tag{2.41}$$

Neste caso a integração deve ser feita sobre todo o espaço $\mathfrak{L}$ da variável X.

No contexto deste trabalho, estamos interessados em medir a entropia de um *cluster*. Esta entropia irá de certa forma medir o quanto um *cluster* tem de informação comparado a outro, utilizando uma medida de divergência (detalhada na seção seguinte). O cálculo da entropia será realizado utilizando-se a probabilidade

de cada ponto pertencer a um *cluster*. Esta probabilidade será dada por algum dos métodos vistos nas seções 2.4 e 2.5.

Um dos modelos de distribuição utilizado neste trabalho é o modelo gaussiano visto na seção 2.4. Podemos calcular a entropia de uma variável gaussiana utilizando a equação 2.41. Para uma VA gaussiana com média μ e variância σ^2 a entropia é dada por

$$H(X) = \log\left(\sigma\sqrt{2\pi e}\right) \tag{2.42}$$

onde X é a VA com distribuição gaussiana. Podemos observar que a entropia H independe da média. Este fato é esperado já que a mudança da média não modifica a quantidade informação que temos sobre a VA. Já a variância, reflete a quantidade de informação contida na VA. Quanto mais dispersa for a distribuição (ou seja, maior a variância) maior é a entropia e maior é a nossa incerteza sobre que valores a VA pode assumir. Para valores pequenos de variância, temos gaussianas mais estreitas e os valores ocorrem em uma faixa menor, fazendo com que tenhamos mais informação sobre a VA, ou seja diminuindo assim a entropia.

No caso de uma gaussiana multivariada (de dimensão D) com média $\boldsymbol{\mu}$ e matriz de covariância $\boldsymbol{\Sigma}$ a entropia da VA X_g é dada por

$$H(X_g) = \frac{1}{2}\left(D + D\log(2\pi) + \log\left(|\boldsymbol{\Sigma}|\right)\right) \tag{2.43}$$

Novamente observamos a independência da média e a dependência da variância. Dessa vez o determinante de $\boldsymbol{\Sigma}$ nos informa que a entropia depende das variâncias em cada eixo (como visto na seção 2.3).

Um fato interessante na entropia de gaussianas multivariadas, é a dependência direta da dimensão D. Isto nos diz que o simples fato de aumentar a dimensão, já aumenta a incerteza sobre a VA. O aumento da dimensão corresponde à inclusão de mais uma variável ao conjunto de dados, o que obviamente aumenta nossa incerteza sobre a ocorrência de um vetor qualquer.

No caso de representarmos a distribuição de uma VA por uma mistura de gaussianas, o cálculo da entropia, embora possível, se torna bastante complicado

devido à necessidade de se integrar o logaritmo de uma soma ponderada. Intuitivamente espera-se que, desta vez, a entropia não dependa somente da dimensão e matriz de covariância, mas também das médias. A dependência das médias de uma mistura de gaussianas no cálculo da entropia se deve ao fato de que, em uma mistura, quanto mais separadas as distribuições individuais, maior é a dispersão da VA e maior é a nossa incerteza sobre a mesma.

2.7.2 Entropia de Rényi

A entropia de Rényi é uma medida de entropia generalizada parametrizada por uma constante α, e é definida (no caso discreto) como sendo [77]

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \left(\sum_i {p_i}^\alpha \right), \tag{2.44}$$

onde p_i são as probabilidades de ocorrência dos valores $\mathbf{x}_i$ da variável X. Para o caso contínuo, o somatório se torna uma integral e p_i passa a ser a densidade de probabilidade $p(x)$. Esta medida generaliza a entropia Shannon. Pode-se mostrar que para o caso de $\alpha = 1$ a entropia de Rényi torna-se igual a entropia de Shannon [94].

A entropia de Rényi é muito utilizada pelos Físicos em particular na área de Mecânica Quântica. Por ser parametrizada, a entropia de Rényi mostra propriedades interessantes para os diversos valores de α. Neste trabalho estamos interessados em particular no caso para o qual $\alpha = 2$, embora em qualquer caso (qualquer valor de α) as propriedades gerais para entropia de uma VA se mantém. No caso para $\alpha = 2$ temos bastante simplificado o cálculo para entropias de gaussianas e até misturas de gaussianas. Em particular, a entropia de Rényi para uma VA gaussiana X_g com média $\boldsymbol{\mu}$ e matriz de covariância Σ é dada por

$$H_\alpha(X_g) = \frac{1}{2} \left(D \log 4\,\pi + \log |\Sigma| \right). \tag{2.45}$$

Como podemos observar, a expressão para entropia de Rényi é parecida com a expressão da entropia de Shannon. De fato, o cálculo para entropia de Rényi de

ordem 2 é bem mais simples do que o cálculo para entropia de Shannon. Comparando a equação 2.45 com a equação 2.43 observamos as mesmas propriedades. Podemos observar a dependência direta da dimensão D e do determinante da matriz de covariância. Outra forma de compararmos as duas entropias, é refazendo o experimento com a moeda citado anteriormente, só que desta vez utilizando a entropia de Rényi. A Figura 2.15 mostra como varia a entropia de Rényi H_2 (entropia de Rényi para $\alpha = 2$) em função da probabilidade de ocorrência de uma das faces da moeda.

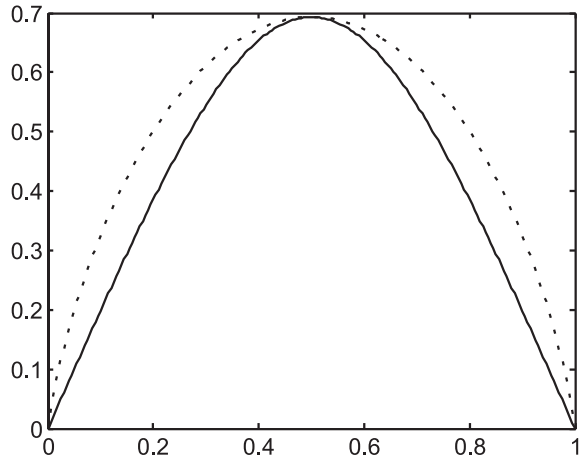


Figura 2.15: Gráfico de $H_2(p)$ e $H(p)$ em função de p

Como podemos observar, a entropia de Rényi é máxima quando temos eventos equiprováveis e é zero quando termos certeza da ocorrência (ou não-ocorrência de um dos eventos).

No contexto deste trabalho, a entropia de Rényi será utilizada para o cálculo da entropia de uma mistura de guassianas. A ordem 2 ($\alpha = 2$) será utilizada pelo fato de simplificar bastante o cálculo analítico da entropia. A simplificação acontece porque no cálculo da entropia de Rényi, a integral resultante ocorre somente no quadrado do somatório das gaussianas e não no logaritmo do somatório, como é o caso da entropia de Shannon. O uso da entropia no método de *clustering* apresentado, serve para realizar medidas de informação de um *cluster* com relação o outro. Por tratar-se de uma medida relativa, não estamos interessados nos valores numéricos da entropia em si, mas nas propriedades que estes valores apresentam quando utilizados como métrica de dissimilaridade. Este fato nos deixa livres para escolher uma medida de

entropia que não precisa necessariamente ter unidade definida ou ter uma definição padrão, como a entropia de Shannon. Isto justifica a liberdade na escolha do valor de α na entropia de Rényi.

2.8 Métricas de dissimilaridade

Existem várias maneiras de medir a similaridade entre dois padrões (ou amostras de um conjunto de dados) em um classificador. Medidas de similaridade (ou dissimilaridade) servem para comparar dois ou mais padrões em um processo de classificação ou medir a pertinência de um dado padrão a uma classe qualquer. Existem na literatura [17, 37] diversas medidas de similaridade ou dissimilaridade. Alguns autores costumam denominar algumas medidas de dissimilaridade como medidas de distância. A medida mais comum é a distância Euclidiana entre dois padrões (vetores) dada pela equação 2.35.

Em geral há sempre uma estatística envolvida na distribuição dos padrões em um conjunto de dados como um todo e cada classe possui sua distribuição. A utilização de uma métrica que incorpore esta estatística apresenta uma grande vantagem em relação a medida de distância Euclidiana comum. A seguir serão descritas métricas de dissimilaridade (ou similaridade) que levam em conta a estatística dos dados onde a medida será realizada. Como visto na seção 2.7, a ligação entre estatística e informação é bastante utilizada, por esse motivo, algumas métricas utilizarão o conceito de informação tanto quanto o conceito de estatística.

Um ponto muito importante a ser analisado, quando se trata de medidas de divergência em *clustering* é quanto a invariância à escala dos dados. O algoritmo de *clustering* não deve ser sensível a transformações lineares que venham a ocorrer nos dados. O motivo para que isto ocorra é que em geral, os conjuntos de dados utilizados estão em um espaço denominado de espaço de atributos. Como dito na introdução, cada padrão a ser analisado é representado por um vetor de dimensão D. Cada componente desse vetor é um atributo da amostra ou padrão. Estes atributos podem ter as mais variadas unidades; comprimento, tempo, intensidade luminosa, etc... Inclusive na maioria dos casos, um vetor de dados é composto por

atributos com unidades diferentes. A simples mudança de uma unidade para outra (por exemplo de metro para centímetro) causa uma mudança radical nos números que compõem o conjunto. Na grande maioria dos casos, uma mudança de unidade é tida como uma transformação linear (apenas um ajuste de escala). Não é admissível que um algoritmo de *clustering* seja sensível a mudança de escala, ou seja, que tenha seu resultado modificado por uma simples mudança de unidade. Por causa disso, é importante que as medidas de dissimilaridade utilizadas em algoritmos de *clustering* sejam invariantes pelo menos a transformações lineares, garantindo assim, que mudanças de unidade, ou escala, não interfiram no resultado final do algoritmo.

2.8.1 Distância de Mahalanobis

A distância de Mahalanobis entre dois pontos $\mathbf{x}_1$ e $\mathbf{x}_2$ é definida como mostra a equação 2.46

$$d_m(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 - \mathbf{x}_2)\mathbf{\Sigma}^{-1}(\mathbf{x}_1 - \mathbf{x}_2)^t \tag{2.46}$$

onde $\mathbf{\Sigma}$ é a matriz de covariância associada a estatística onde os dois vetores estão imersos. Pode-se observar que a distância de Mahalanobis é uma generalização da distância Euclidiana onde as direções são ponderadas. Na distância Euclidiana, todas as direções são ponderadas da mesma maneira e dizemos que trata-se de uma medida esférica ou isotrópica.

A distância de Mahalanobis pode medir a distância entre os centros de duas gaussianas considerando a matriz de covariância de cada uma. Esta medida fornece valores baixos quando os centros estão alinhados com a direção da variância dos dados. Na Figura 2.16 a distância de Mahalanobis d_2 é maior do que a distância d_1, embora com as mesmas distâncias medidas utilizando a métrica Euclidiana ocorra o contrário.

A distância de Mahalanobis pode ser utilizada como métrica alternativa em algoritmos que utilizem distâncias. Younis et al [91] desenvolveram uma técnica de quantização vetorial baseada na distância de Mahalanobis. Neste trabalho também será utilizada a distância de Mahalanobis como métrica alternativa à distância

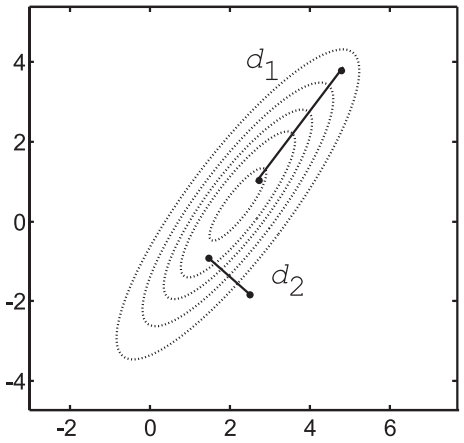


Figura 2.16: Exemplo de distâncias de Mahalanobis em um meio com estatística conhecida

Euclidiana em uma das abordagens descritas no próximo capítulo.

É importante observar que a distância de Mahalanobis, quando utilizada para medir a distância entre os centros de dois conjuntos de dados, é invariante a transformações lineares. Isto ocorre porque a diferença entre os centros é ponderada pelo inverso da matriz de covariância do *cluster* analisado. Quando ocorre uma transformação linear que aumente ou diminua a escala, os valores dos pesos das diferenças é compensado no inverso. Isto torna a distância de Mahalanobis uma candidata a medida de divergência a ser utilizada em algoritmos de *clustering*.

2.8.1.1 Matriz de covariância

A matriz de covariância de um conjunto qualquer pode ser obtida utilizando a equação 2.15 vista anteriormente. Para que esta matriz seja uma boa estimativa, é necessário a utilização de um número de pontos suficiente para representar estatisticamente os dados. Uma forma de testar a representatividade dos dados é ir aumentando o número de pontos e realizar um teste de hipótese [68], que em geral leva a uma estabilização dos valores estimados das matrizes de covariância.

2.8.2 Divergência de Kullback-Leibler

Até aqui, o termo divergência significou a diferença ou distância entre dois pontos ou vetores. Estas diferenças podem ser utilizadas para medir a divergência

entre *clusters*, bastando medir a distância entre seus centros ou médias. Embora a distância de Mahalanobis nos dê uma medida que leve em conta a estatística dos dados, ela não diz respeito a quantidade de informação contida nos *clusters*. Como vimos na seção 2.7 podemos medir a informação de uma variável aleatória utilizando a entropia como medida básica. Neste sentido, podemos utilizar a informação contida nos *clusters* como medida de divergência ou mesmo similaridade. Neste enfoque, a medida não é mais feita entre dois vetores e sim duas distribuições de probabilidade. Talvez a forma mais comum de calcular a divergência entre duas distribuições de probabilidade $p(x)$ e $q(x)$ denotada $D_{p|q}$ seja a divergência de Kullback-Leibler [33] definida como

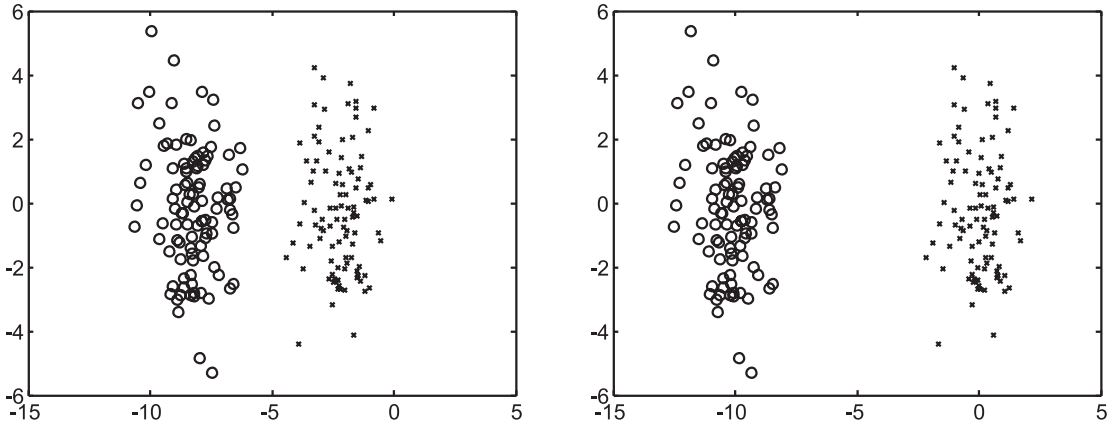
$$D_{p|q} = \sum_{x \in X} p(x) \log \left(\frac{p(x)}{q(x)} \right), \tag{2.47}$$

onde X é uma VA. Para o caso contínuo, temos uma integral no lugar do somatório e a VA em questão seria uma VA contínua. p e q neste caso, se tratariam de "densidades" de probabilidade.

Esta divergência é uma medida não-negativa e não simétrica, e calcula a diferença entre dois modelos de distribuições p e q para uma dada VA X. Se tivermos que p é a distribuição real da VA X, a divergência de Kullback-Leibler (KL) mede o quanto q está "errando" em modelar X. A divergência KL também pode ser utilizada para o cálculo de centróides em algoritmos de *clustering* [88]. Embora esta seja uma aplicação direta de *clustering*, não é o objetivo deste trabalho. No contexto deste trabalho, a divergência KL será utilizada como medida de "distância" entre *clusters*.

Para calcularmos a divergência KL entre dois *clusters*, temos que conhecer as suas distribuições de probabilidade. A única informação que temos a disposição são amostras do conjunto, e por isso um método de estimação de distribuições de probabilidade deve ser utilizado. A título de ilustração, a Figura 2.17 mostra o valor da divergência KL em duas situações diferentes. Neste exemplo, utilizou-se janelas de Parzen para estimar as distribuições $p(x)$ e $q(x)$. Como os dados são numéricos, o valor da divergência foi calculado utilizando-se integração de Monte Carlo.

Como podemos observar, quando os *clusters* estão mais próximos, a divergência é



(a) $D_{p|q} = 0.277$ (b) $D_{p|q} = 2.82$

Figura 2.17: Medidas da divergência KL em duas situações diferentes.

menor. Uma vantagem deste tipo de medida é que a mesma é totalmente invariante a transformações lineares ocorridas no conjunto de dados como um todo. Isto se deve ao fato da divergência operar diretamente nas probabilidades e não na posição dos pontos do conjunto (como é o caso das distâncias). Por outro lado, além de ser uma medida assimétrica, este tipo de medida de divergência, exige que se modele cada *cluster* com uma distribuição diferente, fazendo com que não se leve em conta a quantidade de pontos em cada *cluster*, o que refletiria em uma probabilidade *a priori* para ocorrência de cada um.

A medida da divergência KL está ligada com o conceito de informação mútua [33] que é um conceito muito utilizado em teoria da informação. A informação mútua entre duas variáveis aleatórias X e Y é definida como [68]

$$I(X,Y) = H(X) - H(X|Y) \tag{2.48}$$

onde $H(X)$ denota a entropia da VA X e $H(X|Y)$ denota a entropia da VA formada pela ocorrência de um elemento X dado que ocorreu um elemento de Y. A relação entre informação mútua e a divergência KL é dada pela equação 2.49

$$I(X,Y) = D_{p(x,y)|p_x(x)\,p_y(y)} \tag{2.49}$$

sendo $p(x,y)$ a distribuição de probabilidade conjunta das VAs X e Y. $p_x(x)$ e $p_y(y)$ são as distribuições individuais. No contexto deste trabalho um conjunto é

formado por amostras compostas de diversas VAs. Podemos ver a divergência KL como uma medida de quanto um modelo de distribuição $p(\mathbf{x})$ difere da distribuição de probabilidade da ocorrência de uma amostra, ou seja ocorrência de de cada VA simultaneamente (caso onde a distribuição seria o produto das distribuições de cada VA). Isso nos leva a perceber a divergência em função da entropia de um conjunto (ou *cluster*). Esta abordagem é adequada, pois podemos utilizar o conceito de entropia generalizada (Rényi, por exemplo) para o cálculo da divergência. Esta ultima abordagem é adequada pois devemos lembrar da dificuldade em se trabalhar com a entropia convencional de Shannon para o caso de misturas de gaussianas.

2.8.3 Divergência utilizando entropia de Rényi

Conforme exposto na seção anterior, podemos ver a divergência entre duas distribuições como uma medida baseada na entropia de cada distribuição. Neste sentido, pode-se utilizar a entropia de Rényi para calcular a divergência entre duas distribuições. Algumas aplicações já utilizam com sucesso esta técnica [34] em processamento de sinais e até mesmo para *clustering* [25]. Neste trabalho, consideramos a seguinte medida $H_r(X) - H_r(Y)$ onde H_r denota entropia de Rényi. Como apresentado na seção 2.7.2 a medida de entropia só depende da dimensão dos dados e da variância do conjunto. A diferença entre as entropias conseqüentemente só dependerá da diferença entre as variâncias. Esta medida é inadequada para ser utilizada como divergência, pois não contabiliza a distância para os centros de maneira que *clusters* distantes, porém com mesma variância, terão divergência zero.

A proposta do presente trabalho para utilização da entropia de Rényi como forma de medir a divergência entre *clusters* consiste em considerar um conjunto formado pelos dois *clusters*. A divergência proposta aqui é formulada como mostra a equação 2.50

$$D_r(p_x(x), p_y(y)) = H_r(\{X, Y\}) - H_r(X_g). \tag{2.50}$$

Nesta expressão, $\{X, Y\}$ é o conjunto formado pelos pontos tanto do *cluster* X quanto do *cluster* Y (que mais tarde serão modelados utilizando uma mistura

de gaussianas). X_g é um modelo de distribuição gaussiana estimada utilizando o conjunto inteiro $\{X, Y\}$. A Figura 2.18 mostra como age a comparação entre duas distribuições utilizando este tipo de medida.

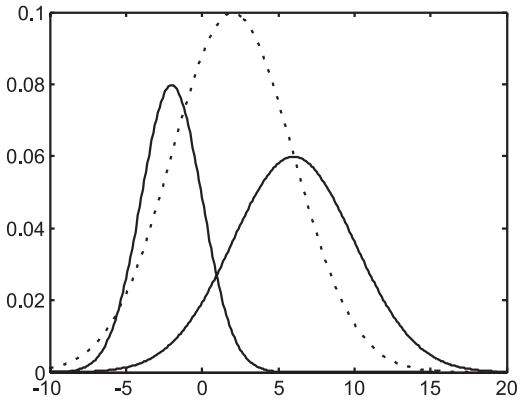


Figura 2.18: Comparação entre um conjunto de dados formado por dois *clusters* e o modelo gaussiano equivalente

Na Figura, os dois *clusters* são formados pelas distribuições com linha cheia. A linha pontilhada representa o modelo gaussiano que "se ajusta" no conjunto todo (considerando as duas distribuições). Desta forma compara-se a "quantidade de variância" presente no conjunto todo, com a variância de uma gaussiana que modela o mesmo. Se separar-mos os *clusters* mantendo sua variância, a variância da gaussiana equivalente aumentará, aumentando assim a divergência. No capítulo seguinte, estudaremos o caso onde utiliza-se uma mistura de gaussianas para modelar os dois *clusters*, e compararemos o caso não só de separação, mas rotação e mudança de escala dos modelos de gaussianas.

A grande vantagem da utilização desta métrica reside no fato de não precisarmos de distribuições diferentes para comparar. Podemos comparar uma distribuição composta por uma mistura de duas gaussianas (como mencionado a pouco) com a gaussiana equivalente. Isto faz com que *clusters* que tenham poucos pontos, e conseqüentemente uma probabilidade *a priori* pequena, contribuam pouco para a divergência. Outra grande vantagem a ser destacada é a facilidade do cálculo analítico quando se trata do modelo de mistura. Esta facilidade, como já explicado, se deve a utilização da entropia de Rényi como medida de informação.

Ainda dentre as vantagens da utilização da divergência utilizando entropia de

Rényi podemos citar a invariância quanto a transformações lineares. Mais adiante neste trabalho será mostrado formalmente o aspecto de invariância para o caso gaussiano.

2.8.4 Negentropia

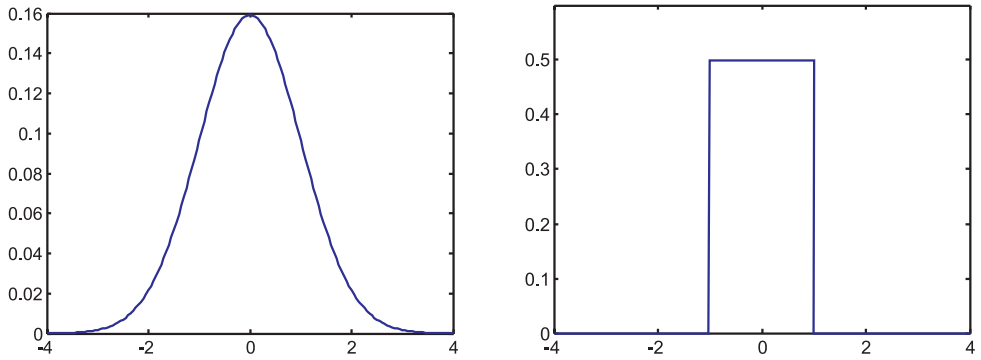
Nas seções anteriores, foram mostradas formas de calcular a divergência entre dois *clusters* de modo a medir a sua "separação". Um modelo proposto neste trabalho, como já citado, consiste em medir a divergência entre a distribuição de probabilidade de dois *clusters* juntos, com o modelo gaussiano equivalente. Esta abordagem é conhecida na literatura como cálculo da *negentropia* entre duas distribuições. A negentropia é bastante utilizada em métodos de Análise de Componentes Principais [36] e é conhecida como medida de *não-gaussianidade*. A negentropia é definida como [11]

$$J(X) = H(X_g) - H(X). \tag{2.51}$$

onde $H(X_g)$ é a entropia de uma VA com mesma média e matriz de covariância da VA X. A negentropia é uma medida não-negativa, e se torna igual a zero, quando a VA X tem distribuição gaussiana. Existem na literatura diversas maneiras de se estimar a negentropia de uma VA qualquer [35]. Classicamente, são utilizados momentos de alta ordem da VA (como *kurtosis* e *skewness*) para obter apenas uma estimativa da negentropia. Outras técnicas incluem estimativas baseadas em diferença quadrática entre os valores esperados da VA e de uma VA normalizada gaussiana [36]. Em termos gerais, a negentropia mede o quanto a distribuição de uma VA não se parece com uma gaussisna. Este fato é especialmente útil neste trabalho pois estamos interessados em medir o quanto dois *clusters* estão "juntos". A título de ilustração, a Figura 2.19 mostra dois conjuntos e suas respectivas medidas de negentropia.

É interessante notar que a negentropia de distribuições gaussianas possuem valor igual a zero 2.19(a), enquanto distribuições não gaussianas 2.19(b) possuem negentropia com valor maior que zero.

Estes exemplos foram calculados analiticamente, porém para casos de



(a) $J(X) = 0$ (b) $J(X) = 1.5628$

Figura 2.19: Medidas da negentropia para conjuntos de dados com distribuições diferentes.

distribuições desconhecidas, é necessário uma estimativa da entropia e um método de integração numérica ou a utilização de métodos aproximados (como já mencionado). Neste trabalho foi desenvolvida uma técnica para calcular a negentropia para o caso de distribuições bimodais que não envolve integração numérica, diminuindo o tempo de processamento e o erro cometido. Este método será exposto nos capítulos seguintes.

Neste trabalho, além de um método para calcular a negentropia, será proposta uma maneira de se calcular a divergência entre duas distribuições utilizando a medida de negentropia. Esta consiste em calcular a negentropia da distribuição de um conjunto de dados formado por ambos os conjuntos cujas distribuições queremos medir a divergência. O efeito desta medida é que, quanto mais separadas as distribuições, menos as mesmas, juntas, se parecem com uma gaussiana. Mesmo em situações onde as distribuições estão próximas, quanto menos "alinhadas" elas estiverem, menos se parecerão com uma gaussiana. Desta maneira, a negentropia pode ser utilizada para medir a divergência entre duas distribuições.

Segundo Hyvärinen e Oja [36] a negentropia é uma medida invariante a transformações lineares. Como já mencionado, este fato é muito importante em problemas de *clustering*. Por ser uma medida relacionada com informação, a mesma incorpora as vantagens vistas na seção 2.7.

2.8.5 Distancia de Bhattacharyya

As medidas de divergência apresentadas até agora medem (ou podem ser usadas para medir) a distância entre duas distribuições de probabilidade quaisquer. Para o caso onde estas distribuições são gaussianas, uma métrica alternativa pode ser utilizada, a distância de Bhattacharya. Esta distância mede a diferença entre duas funções gaussianas e é definida como [42]

$$B(p,q) = \frac{1}{8}\left(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q\right)^t \left[\frac{\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_q}{2}\right]^{-1} \left(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q\right) + \frac{1}{2}\log \frac{\left|\frac{\boldsymbol{\Sigma}_p + \boldsymbol{\Sigma}_q}{2}\right|}{\sqrt{|\boldsymbol{\Sigma}_p|\ |\boldsymbol{\Sigma}_q|}} \tag{2.52}$$

onde $\boldsymbol{\mu}_q$, $\boldsymbol{\mu}_q$, $\boldsymbol{\Sigma}_\mathbf{p}$ e $\boldsymbol{\Sigma}_\mathbf{q}$ são as médias e matrizes de covariância das distribuições p e q. É interessante notar que a distância de Bhattacharya é composta por duas parcelas, a primeira delas representa a diferença entre as médias, enquanto a segunda representa a diferença entre as matrizes de covariância. Isto quer dizer que a distância de Bhattacharya só será efetivamente zero, se as distribuições tiverem mesma média e mesma matriz de covariância. Isto é importante porque mede, de certa forma, tanto a distância quanto o "alinhamento" entre as gaussianas.

Mais uma vez, devemos observar a questão da invariância quanto à transformações lineares. No caso da distância de Bhattacharya, trata-se de uma medida invariante. Embora tenhamos uma parcela dependendo da diferença entre os centros das gaussianas, a medida ainda é invariante a mudanças de escala. Isto acontece graças à ponderação com o inverso da média das matrizes de covariância, como ocorre com a distância de Mahalanobis.

2.8.6 Outras medidas de divergência

Existem muitas formas de se medir a divergência entre dois *clusters*. Foram apresentadas até agora, medidas bastante utilizadas na literatura e que de certa forma apresentam alguma vantagem no uso em problemas de *clustering* (como invariância a transformações lineares e incorporação da estatística dos dados). Por serem medidas estatísticas, nenhum comentário é feito sobre a unidade ou grandeza

de cada medida de divergência. Em particular para problemas de *clustering*, o que interessa é a quantidade relativa, apenas para fins de comparação com outras medidas da mesma natureza realizadas pelo algoritmo no mesmo conjunto de dados. Fora as medidas apresentadas, podemos citar ainda algumas medidas de divergência que, embora não utilizadas neste trabalho, podem servir de base para a elaboração de medidas mais simples e úteis.

2.8.6.1 Erro quadrático

Podemos medir a divergência entre duas distribuições de probabilidade $p(\mathbf{x})$ e $q(\mathbf{x})$ integrando o erro quadrático da diferença entre as duas, como mostra a equação 2.53

$$D(p,q) = \int \cdots \int \left(p(\mathbf{x}) - q(\mathbf{x})\right)^2 \mathrm{d}\mathbf{x}. \tag{2.53}$$

onde a integração deve ser feita em todo o espaço da variável $\mathbf{x}$. Esta abordagem é útil no caso gaussiano, em que a integração pode ser feita analiticamente de maneira acelerar o processo de cálculo.

Esta proposta para o cálculo da divergência, resulta em valor maior ou igual a zero, sendo zero apenas quando as duas distribuições forem iguais.

2.8.6.2 Verossimilhança

Outra maneira alternativa de se medir a divergência entre *clusters* é utilizando a verossimilhança. Podemos usar a equação 2.21 de maneira a testar o quanto os parâmetros de uma distribuição modelam a outra distribuição e vice versa. Esta medida pode ser utilizada para o caso gaussiano de maneira a simplificar o conjunto de parâmetros. O emprego do logaritmo da verossimilhança pode também ser utilizado para evitar erros numéricos que ocasionalmente possam aparecer devido à aplicação do produtório em números bastante pequenos gerados pelas exponenciais da gaussiana. A equação 2.54 apresenta uma formalização da medida

$$D(p,q) = \prod_{i=1}^{N_p} f(\mathbf{x}_i|\boldsymbol{\theta}_q) + \prod_{i=1}^{N_q} f(\mathbf{x}_i|\boldsymbol{\theta}_p) \tag{2.54}$$

em que o somatório é feito para todos os pontos de cada conjunto, ou seja, N_p pontos no conjunto cuja distribuição é $p(x)$ e N_q pontos no conjunto cuja distribuição é $q(x)$. A função $f(\cdot)$ é um modelo de distribuição escolhido. $\boldsymbol{\theta}_q$ e $\boldsymbol{\theta}_p$ são os conjuntos de parâmetros estimados para a distribuição q e p. A medida é formada por duas parcelas para que a mesma se torne simétrica e contabilize a diferença tanto para uma, quanto para a outra distribuição.

Neste tipo de abordagem, o valor medido é sempre maior ou igual a zero, e é máximo quando as duas distribuições forem iguais.

2.9 Sumário

Este capítulo apresentou as técnicas necessárias para o desenvolvimento deste trabalho. Obviamente existem muitas outras e a literatura é bastante extensa. No contexto deste trabalho as ferramentas são utilizadas, aproximadamente na ordem em que foram apresentadas neste capítulo, para dar origem a um novo método de *clustering* que seja robusto e que de certa forma venha a apresentar soluções para as situações onde os métodos tradicionais falham. Utilizaremos neste trabalho técnicas para realizar o particionamento correto de um conjunto de dados e após isso, técnicas para modelagem estatística das classes geradas a partir deste particionamento.

O capítulo seguinte mostrará o desenvolvimento do método proposto neste trabalho. Serão utilizadas as ferramentas apresentadas aqui, enfatizando pontos positivos e negativos de cada uma e como as mesmas contribuem para elaboração do método.

Capítulo 3

Método Proposto

3.1 Introdução

Neste capítulo será descrito o método e as técnicas propostas neste trabalho. Nas seções seguintes, serão apresentados os desenvolvimentos de cada item que compõe o método como um todo. O método proposto realiza a classificação (*clustering*) e modelamento estatístico de um conjunto de dados. Para isto são utilizados métodos de quantização vetorial e ferramentas e métodos estatísticos apresentados no capítulo 2

Classificar estatisticamente um dado qualquer de um conjunto, significa atribuir uma classe, dentre um conjunto de classes, a este dado, baseado na probabilidade de pertinência do mesmo a cada uma das classes. A classe que tiver maior probabilidade de que o dado pertença a ela, será a classe atribuída. Para que este tipo de classificação seja possível, precisamos obter as distribuições de probabilidades para cada classe, ou seja, precisamos de um modelo estatístico do conjunto que contém as classes. Obviamente esta informação não está disponível e em muitos casos estas distribuições podem vir a ser bastante complexas. Neste trabalho, sugerimos uma técnica para estimar os parâmetros de misturas de gaussianas (descritas nas seções anteriores) que modelarão cada classe. Desta forma estaremos estimando a probabilidade de um dado ou ponto pertencer a uma classe ou *cluster* ($P(\mathbf{x}|w_i)$) modelado por uma mistura de gaussianas conforma visto na seção 2.2 do capítulo anterior. A dificuldade desta tarefa reside no fato de não se ter *a priori* informação

de classes nas amostras conhecidas, apenas pontos desordenados que formam *clusters* em determinados lugares no espaço (também desconhecidos).

O método consiste em quantizar vetorialmente todo o conjunto com um número sobrestimado[1] de centros modelando cada um como uma gaussiana. Uma vez modelados os conjuntos iniciais, desejamos saber a quais *clusters* cada um pertence. isto é feito com o conceito de "ligação" dos conjuntos auxiliares. Os conjuntos que estiverem ligados entre si, formarão o mesmo *clusters*. Por exemplo, se o conjunto A estiver ligado ao conjunto B, que por sua vez estiver ligado ao conjunto C, estes três (A, B e C) formarão um *cluster*. Lembrando que cada conjunto é modelado por uma gaussiana, ao final do processo, cada *cluster* será modelado por uma mistura de gaussianas formada pelas gaussianas modeladas inicialmente.

3.2 Modelo dos dados

A técnica proposta inicia estabelecendo N_a gaussianas para modelar todo o conjunto. A Figura 3.1 ilustra um exemplo de conjunto de dados qualquer a ser classificado. As N_a gaussianas iniciais serão posteriormente agrupadas em classes (ou seja, *clusters*). cada classe será modelada por uma mistura de algumas destas gaussianas iniciais, de maneira que no final, todas as gaussianas estarão fazendo parte de uma mistura que modelará alguma classe.

No início do processo nenhuma gaussiana é atribuída a uma classe. O primeiro passo consiste em estimar os parâmetros individuais das gaussianas que separam os dados em conjuntos. Neste trabalho exploramos a utilização de duas técnicas para realizar esta tarefa. A primeira trata-se do algoritmo EM. Este algoritmo possui a vantagem de estimar não só as médias, mas os outros parâmetros como matriz de covariância e probabilidades *a priori*. A desvantagem na utilização do algoritmo EM é o alto custo computacional. Embora seja simples, este método requer a computação iterativa de expressões com alto custo em termos de processamento. A segunda forma abordada neste trabalho consiste em utilizar a quantização vetorial para estimar os parâmetros das misturas. Os primeiros parâmetros estimados são

[1] número de centros maior do que o número de *clusters* presentes

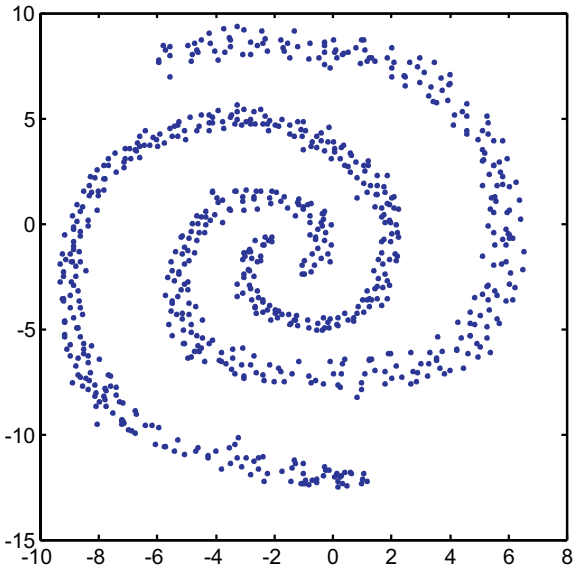


Figura 3.1: Exemplo de um conjunto de dados

as médias $\boldsymbol{\mu_i}$ para todos o conjuntos. As médias são estimadas utilizando-se uma redes neural competitiva [23] que encontrará N_a centros $w_1, w_2, ..., w_{N_a}$. Cada centro encontrado pelo algoritmo de quantização vetorial será tido como a média de uma das N_a gaussianas. A Figura 3.2 mostra o resultado da quantização vetorial. Após esta etapa, cada centro divide os dados em pequenos grupos que são separados utilizando a distância Euclidiana formando células de Voronoi.

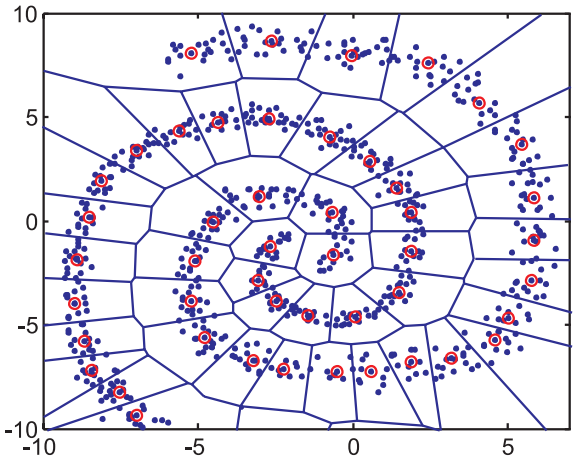


Figura 3.2: Conjunto de dados separados utilizando quantização vetorial

O segundo passo a ser realizado é a estimativa das matrizes de covariância $\mathbf{\Sigma_i}$ de cada gaussiana. Para esta estimativa, deve-se utilizar apenas os dados classificados para cada centro (médias $\boldsymbol{\mu_i}$) encontrados na etapa de quantização vetorial. Isto

classifica cada ponto do conjunto como pertencente a uma gaussiana apenas. Uma estimativa da matriz de covariância pode ser estimada por [68]

$$\boldsymbol{\Sigma}_i \approx \frac{1}{N_i} \sum_{j=1}^{N_i} (\mathbf{x}_j - \boldsymbol{\mu}_i) (\mathbf{x}_j - \boldsymbol{\mu}_i)^t, \tag{3.1}$$

onde N_i é quantidade de pontos associados a gaussiana i no processo de classificação após a quantização vetorial.

Este processo de estimativa das gaussianas não corresponde ao processo ótimo, ou seja, as gaussianas encontradas não minimizam o erro de classificação. Entretanto, escolhendo um número suficiente de centros, este erro pode chegar a ser desprezível e compensa quando observado a simplicidade computacional em relação a algoritmos que minimizariam completamente o erro como é o caso do algoritmo EM [15].

A Figura 3.3 ilustra como se dá o erro entre a classificação com o algoritmo EM e a classificação utilizada neste trabalho.

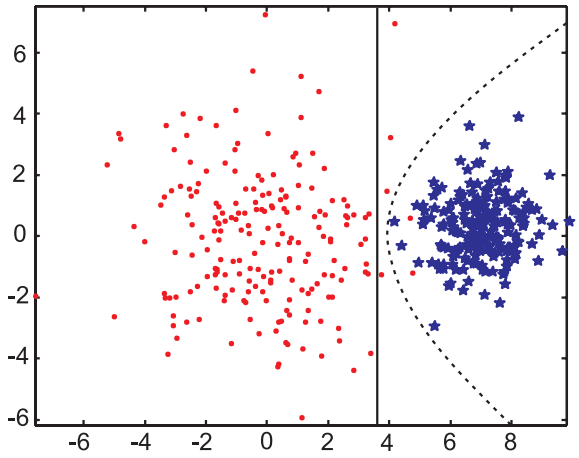


Figura 3.3: Erro cometido na classificação utilizando o algoritmo EM e a quantização vetorial

Na figura, a linha cheia representa a separação utilizada para a classificação com a quantização vetorial e a linha pontilhada representa a classificação utilizando o algoritmo EM. Como podemos observar, a linha cheia separa alguns pontos que pertencem a classe da esquerda. Isto porque a separação é linear. Já a classificação utilizando o algoritmo EM agrupa os pontos utilizando uma superfície que separa de maneira mais adequada os pontos das duas classes. O erro cometido entre as duas separações será considerado pequeno neste trabalho em benefício do alto ganho

em termos de custo computacional quando utilizamos a quantização vetorial como classificador.

Este tipo de classificação agrupa os pontos em células isotrópicas (devido a métrica Euclidiana utilizada para separar os dados), porém as gaussianas estimadas a partir dos pontos classificados nem sempre são isotrópicas. Não foram encontradas na literatura, referências à trabalhos que utilizem este tipo de abordagem, tratando-se assim, de uma das contribuições do mesmo.

A Figura 3.4 ilustra uma situação onde o modelo das gaussianas de cada classe encontrada pelo método de quantização vetorial leva a *clusters* não-isotrópicos.

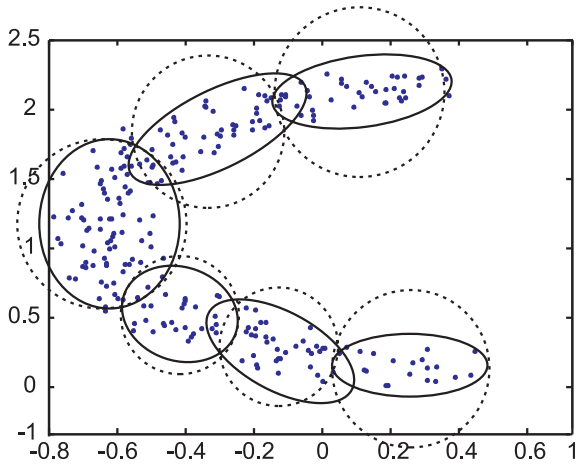


Figura 3.4: Classificação isotrópica resultando em uma gaussiana não isotrópica

Na figura, os círculos pontilhados representam a influência isotrópica (se estes fossem considerados isotrópicos) de cada *cluster* encontrado. Já as linhas cheias representam a influência real de cada um. Como podemos observar, embora a classificação tenha sido feita utilizando a distância Euclidiana, os *clusters* encontrados não são isotrópicos.

Esta forma de se modelar um conjunto, embora simples, se mostrou muito útil para utilização em conjunto com o método de *clustering* desenvolvido neste trabalho. Como a diferença entre o modelo encontrado desta maneira e o modelo encontrado pelo algoritmo EM é desprezível, resolveu-se optar por esta técnica.

O custo computacional deste algoritmo é composto pelo custo de cada técnica utilizada. O primeiro custo envolvido é o custo da quantização vetorial. Nesta etapa, a cada iteração são realizadas operações entre todos os pontos com todos os centros,

sendo assim um algoritmo de ordem $O(N\,N_a)$ (número de operações envolvidas é proporcional ao número de dados vezes o número de centros auxiliares). A outra etapa é a classificação dos dados que consiste em calcular as distâncias de cada ponto para cada centro, sendo assim uma etapa também de ordem $O(N\,N_a)$. A próxima e última etapa consiste no cálculo das matrizes de covariância. Nesta etapa, para cada uma das matrizes são realizadas operações com os pontos classificados por cada uma, de maneira que ao final o número é proporcional ao número de pontos, sendo assim de ordem $O(N)$.

Se o número de pontos for bem maior do que o número de centros auxiliares utilizado, $N \gg N_a$, este prevalece na ordem de operações, tornando este um algoritmo de ordem $O(N)$.

As Figuras 3.6 e 3.5 mostram os resultados de experimentos realizados para mostrar a relação entre o número de pontos em um conjunto e os tempos para calcular a matriz de covariância e os centros na quantização vetorial.

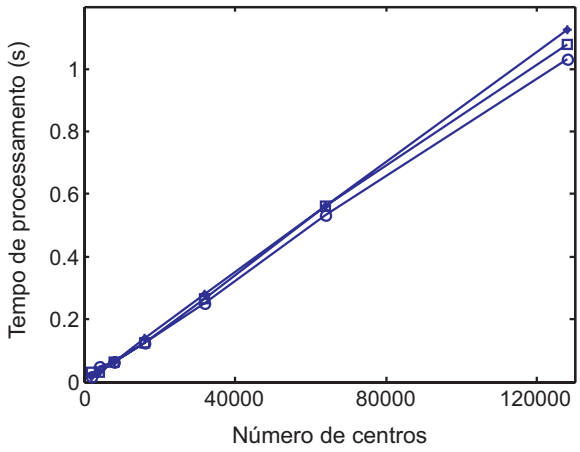


Figura 3.5: Tempo de processamento para o cálculo da matriz de covariância em função do número de pontos em um conjunto de dados

Na Figura 3.5 foram realizados 3 experimentos em dados com dimensões 2, 4 e 8 (círculo, quadrado e estrela, no gráfico). Como podemos observar, em todos os casos, o tempo de processamento cresce linearmente com o número de pontos, como previsto na análise ainda a pouco. Na Figura 3.6 foram realizados também 3 experimentos, só que com conjuntos de dados diferentes. Desta vez existe uma diferença entre um experimento e outro, porém em todos os casos, o tempo é linear com o número de

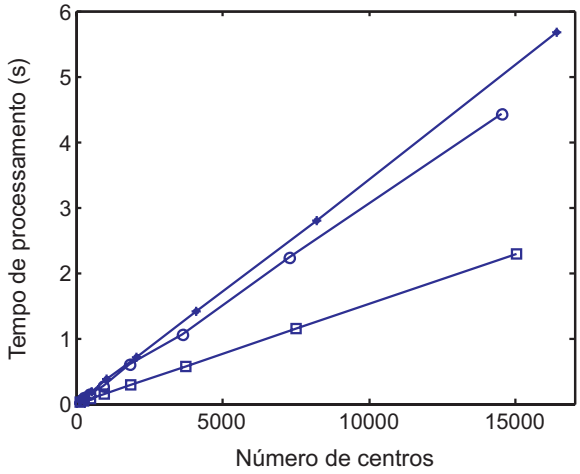


Figura 3.6: Tempo de processamento para o cálculo da quantização vetorial em função do número de pontos em um conjunto de dados

pontos. A utilização do tempo de processamento no lugar do número de operações foi uma escolha meramente prática. Na análise desejada, não importa a unidade do tempo, e sim a verificação da linearidade do processo, confirmando as expectativas sobre a ordem $O(N)$, já mencionada.

3.2.1 Número de conjuntos auxiliares

A escolha de um número de centros auxiliares para a quantização vetorial inicial não é crítica, porém é uma medida empírica. A escolha segura do número de centros pode se dar começando com uma quantidade elevada de centros (relativa ao número de pontos) de maneira a alocar, em média, um número maior do que a dimensão dos dados. Por exemplo, se um conjunto possui 100 pontos e tem dimensão 4, podemos utilizar 20 conjuntos auxiliares de modo a que cada conjunto fique com (em média) 5 pontos, que é um número maior do que a dimensão 4. O número deve ser maior que a dimensão dos dados, porque para que a matriz de covariância seja inversível, a mesma deve possuir autovalores todos diferentes de zero, o que, pela forma de estimar da equação 2.15 só é possível com um número de pontos maior que a dimensão D dos dados. Isto ocorre porque com menos pontos que a dimensão dos dados, os pontos são sempre "co-planres" (em uma dimensão elevada poderia-se chamaria de "co-hiperplanares"). Se isto ocorrer, a matriz será singular e deveremos optar por uma quantidade menor de conjuntos auxiliares.

3.3 Ligação dos conjuntos auxiliares

Estimados os parâmetros das gaussianas passamos para a associação de cada gaussiana à uma determinada classe. Esta é a etapa que constitui a contribuição mais importante do trabalho. Serão mostradas nas próximas seções, métodos e aperfeiçoamentos destes que levam ao algoritmo final que constitui o método proposto.

Para associar uma gaussiana a uma classe, formaremos o conceito de matriz de ligação **L**. Esta matriz é definida como

$$\begin{aligned} &\mathbf{L} = [l_{i\,j}]_{N_a \times N_a} \\ &l_{i\,j} = \begin{cases} 1 & \text{gaussiana } i \text{ está ligada à gaussina } j \\ 0 & \text{caso contrário} \end{cases} \end{aligned} \tag{3.2}$$

Esta matriz $N_a \times N_a$ possui diagonal preenchida com 1 e é simétrica. Os valores $L_{i,j}$ são sempre 0 ou 1, como visto na equação 3.2. O conceito de ligação entre gaussianas é utilizado aqui para denominar gaussianas que pertencem a mesma classe. Este conceito não é totalmente novo e já foi utilizado por Tyree e Long em [85]. De fato, uma classe é formada pela mistura de gaussianas "ligadas". Se a gaussiana **A** está ligada a gaussiana **B** e a gaussiana **B** estiver ligada a gaussiana **C**, todas estas farão parte da mesma mistura que representará a mesma classe. A Figura 3.3 ilustra como ficarão os centros após a etapa de ligação. Como podemos observar, existem apenas duas classes cujos centros estão ligados de maneira a formar cada uma das classes.

O objetivo, a partir da definição da matriz **L**, é encontrar uma matriz (ligações entre centros) que melhor modela o conjunto, ou seja, que produza misturas de gaussianas que se agrupem em classes correspondentes a distribuição espacial dos dados.

Uma vez associadas as gaussianas às suas classes, cada classe é então modelada pela mistura destas gaussianas como visto na seção 2.4. A Figura 3.8 mostra o conjunto totalmente modelado pelas misturas. As cores indicam o gradiente de probabilidade.

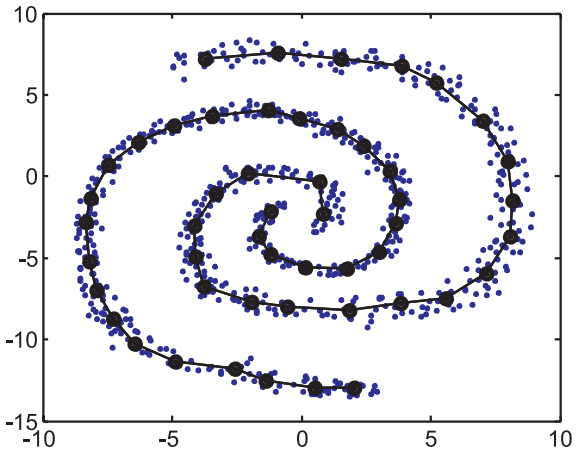


Figura 3.7: Centros dos conjunto ligados formando as misturas de gaussianas

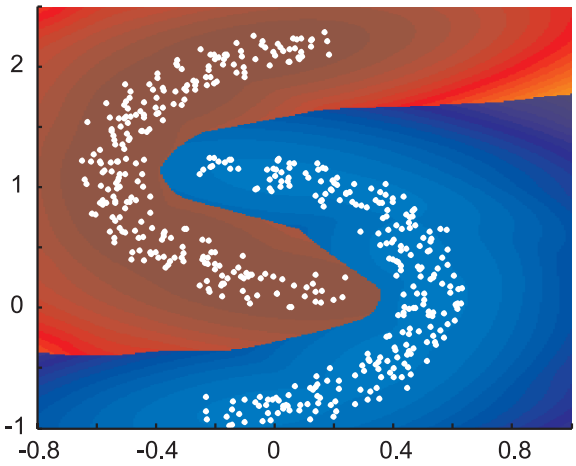


Figura 3.8: Resultado final do método com as probabilidades de cada classe

Com este resultado, podemos classificar pontos fora do conjunto medindo sua probabilidade de pertencer a uma classe ou a outra (cada uma tem um modelo de mistura de gaussianas) através da expressão

$$p(\mathbf{x}|w_i) = P_i \sum_{j \in w_i} P_j^{(i)} \, N(\mathbf{x}, \boldsymbol{\mu}_j^{(i)}, \boldsymbol{\Sigma}_j^{(i)}) \tag{3.3}$$

onde $p(\mathbf{x}|w_i)$ é a probabilidade de ocorrer $\mathbf{x}$, dado w_i, $N(\mathbf{x}, \boldsymbol{\mu}_j^{(i)}, \boldsymbol{\Sigma}_j^{(i)})$ é uma gaussiana com média $\boldsymbol{\mu}_j^{(i)}$ e matriz de covariância $\boldsymbol{\Sigma}_j^{(i)}$, o índice superior (i) indica que o valor esta associado a classe i.

Neste trabalho foi desenvolvido um método de ligação que encontra a matriz $\mathbf{L}$. Para se chegar a este método foram sendo desenvolvidos e estudados outros métodos, envolvendo outras técnicas, que foram sendo aperfeiçoados. Estes métodos

apresentaram vantagens e desvantagens que foram sendo analisadas e melhoradas até se chegar ao método final. Nas seções seguintes serão mostrados os métodos na seqüência, do mais simples ao método final que é o grande objetivo do trabalho. Antes de descrever o primeiro método, será descrita a forma geral de obtenção da matriz **L**. Será demonstrado também que, para o método proposto neste trabalho, a distância Euclidiana não é adequada para ser utilizada como métrica.

3.4 Obtenção da matriz L

Nesta seção será explicado como se dá a obtenção da matriz **L** para diferentes métricas. Nas seções posteriores serão detalhadas as utilizações de cada métrica utilizando a abordagem descrita nesta seção.

De maneira geral, o método inicia realizando uma medida de dissimilaridade entre cada par de conjuntos (todos os pares possíveis). Isso gera uma matriz de distâncias com diagonal igual a zero (distância de um conjunto a ele mesmo). Em seguida é definido um limiar d_t que irá ser utilizado para decidir se um par de conjuntos qualquer será ligado ou não. Cada par (C_i, C_j) é então testado; se sua dissimilaridade for maior do que o valor d_t, os conjuntos C_i e C_j serão ligados, ou seja $L_{i,j} = 1$, caso contrário teremos $L_{i,j} = 0$ e os conjuntos não serão considerados ligados. O valor de d_t depende obviamente da dissimilaridade utilizada e de fato é um valor de dissimilaridade limite para o qual os conjuntos serão considerados ligados. A escolha do limiar que produza uma matriz de ligação correta também vai depender de que medida de dissimilaridade está sendo utilizada. Neste trabalho foram utilizadas e testadas algumas medidas de dissimilaridade, cada uma dando origem a um método. Nas seções adiante serão detalhados os usos de cada uma dessas medidas. A seguir é mostrado um resumo do algoritmo.

Trabalhos já foram desenvolvidos utilizando medidas de divergência [3] para separar *clusters* em um conjunto de dados. Isto justifica a utilização de métricas de divergência com o objetivo de medir a dissimilaridade entre duas gaussianas, como é o caso utilizado neste trabalho. Em geral, o modelo gaussiano é o mais utilizado e tem se mostrado um modelo bastante útil.

O ponto crítico nesta abordagem é a escolha do limiar d_t. A utilização de métricas invariantes a transformações lineares ajuda bastante na escolha do limiar. Esta ajuda se dá no sentido de que, para vários problemas diferentes, o limiar apropriado tende a ser o mesmo. Neste trabalho foi realizada uma análise quantitativa da influência do limiar no resultado final obtido, que será mostrada no capítulo de resultados.

Antes de descrever a utilização das medidas de dissimilaridade utilizadas, será justificado porque a distância Euclidiana não é uma medida adequada para ser utilizada neste tipo de técnica.

3.4.1 Distância Euclidiana

A utilização da distância Euclidiana como métrica de dissimilaridade pode ser feita medindo-se a distância entre os centros dos conjuntos. Considere o conjunto da Figura 3.9. Se utilizarmos a distância Euclidiana para testar a ligação nos centros, ocorrerá que alguns centros estarão próximos entre si, mas não farão parte da mesma classe.

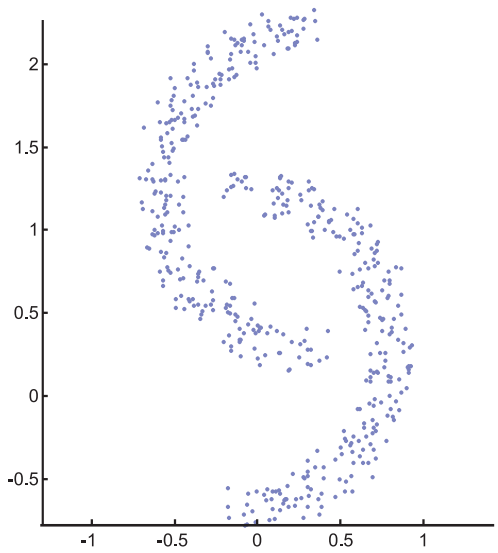
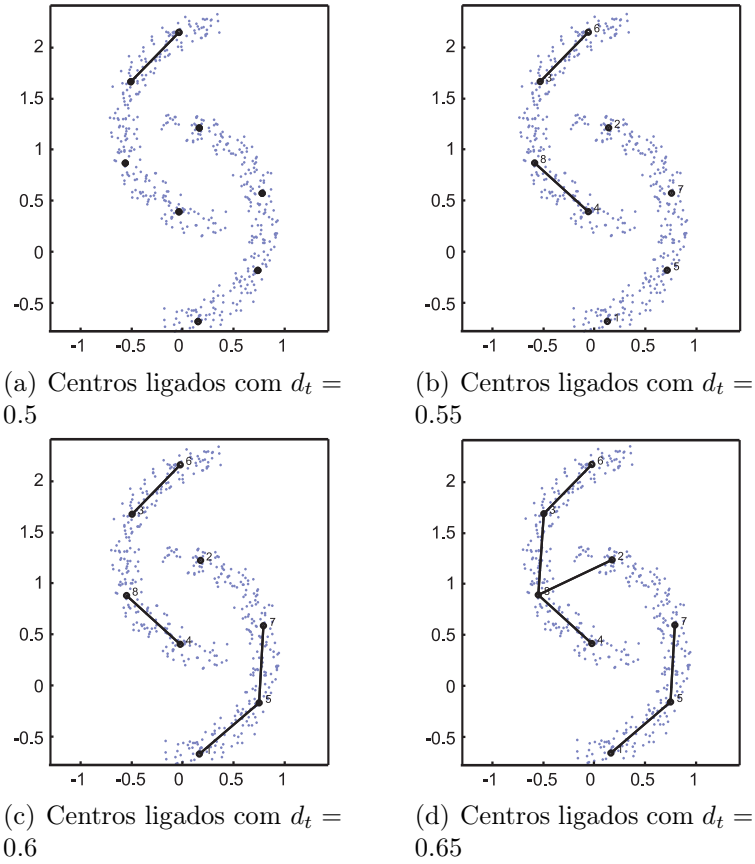


Figura 3.9: Exemplo de um conjunto de dados

A Figura 3.10 mostra as ligações que vão se formando em um conjunto qualquer quando se incrementa gradativamente o limiar d_t utilizando-se como medida a distância Euclidiana. Neste exemplo temos duas classes que devem ser formadas pela ligação correta dos centros que formam o semi-arco superior e pelos que formam

o semi-arco inferior.



(a) Centros ligados com $d_t = 0.5$

(b) Centros ligados com $d_t = 0.55$

(c) Centros ligados com $d_t = 0.6$

(d) Centros ligados com $d_t = 0.65$

Figura 3.10: Ligações feitas aumentando-se a distância Euclidiana limiar d_t

Como podemos observar, para distâncias pequenas, poucas ligações se formam, pois a distância entre os centros é alta em relação ao limiar escolhido. Com o limiar igual a 0.5 (Figura 3.10(a)) existem 7 classes, 6 correspondendo aos 6 centros que não foram ligados e uma correspondendo aos dois primeiros centros superiores ligados. A medida que aumentamos o limiar, mais ligações vão se formando e menos classes são criadas. Na Figura 3.10(d) temos finalmente duas classes (que é o número de classes correto) formadas. Entretanto, a primeira classe formada por 5 centros, está incorretamente formada, por possuir um centro que pertenceria a outra classe.

Para que a distância Euclidiana pudesse ser utilizada adequadamente, deveria existir uma grande quantidade de centros para que não ocorresse o problema de ligação incorreta. Isto gera uma dependência do número de centros que não facilita a utilização do método.

Ainda que a escolha do número de centros não fosse problemática, existe

a questão da invariância a transformações lineares. A utilização da distância Euclidiana torna o método dependente da escala, inviabilizando totalmente a sua utilização em problemas de *clustering*. Por exemplo, se os dados fossem medidas de comprimento e estivessem em metros, a distância limiar estaria em metros. A simples mudança para centímetros, faria com que a distância limiar aumentasse em 100 vezes. Lembrando que é bastante comum que os dados possuam unidades diferentes para cada variável, o problema torna-se crítico.

3.5 Método utilizando distância de Mahalanobis

O primeiro método implementado utiliza-se da distância de Mahalanobis descrita na seção 2.8.1 para ligar os conjuntos. Utilizaremos esta medida como métrica alternativa à distância Euclidiana para medir a distância entre os centros encontrados na fase de quantização vetorial.

Na abordagem de ligação descrita, uma distância limiar d_t é estabelecida, e todas as ligações são testadas. Quando uma ligação possui distância de Mahalanobis menor do que a distância limiar, os centros são considerados ligados, como já explicado na seção anterior. Este tipo de métrica já foi utilizado com sucesso em técnicas de *clustering* [84] e, como será mostrado mais adiante, suas características de invariância já são bem conhecidas.

O objetivo desta abordagem é medir a dissimilaridade entre as gaussianas e agrupar as que sejam similares em relação a distância d_t. A utilização da distância de Mahalanobis possibilita realizar uma medida que leve em consideração a estatística entre uma gaussiana e outra. A medida de divergência entre duas gaussianas que compõem uma VA X é feita como mostra a equação 3.4

$$d_m^{(x)}(C_1, C_2) = d_m(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2), \tag{3.4}$$

onde $d_m(\cdot)$ é a distância de Mahalanobis, descrita no capítulo anterior, entre o centro (vetor média) da primeira gaussiana (representada pelo conjunto C_1) até o centro da segunda (C_2) levando em consideração a matriz de covariância de ambos os conjuntos. Na equação3.4, podemos observar a dependência da diferença entre

as médias ponderada pelo inverso das matrizes de covariância. Isto quer dizer que quanto mais distante estiverem as gaussianas, maior a sua divergência. A ponderação com o inverso da matriz de covariância tem o efeito de diminuir a divergência se as gaussianas estiverem separadas na direção da maior variação e diminuir, caso contrário. Deste modo, duas gaussianas que estiverem distantes espacialmente de 3 unidades, porém com eixos principais (maior variância) paralelos, terão divergência maior que duas que estiverem com eixos principais alinhados. Como veremos na seção de testes e resultados, esta é uma característica que torna a distância de Mahalanobis muito apropriada para ser utilizada como métrica em problemas de *clustering*.

Esta abordagem encontra-se publicada [58] (incluindo uma versão utilizando modelamento estatístico [60]) e constitui uma das contribuições deste trabalho. A utilização da distância de Mahalanobis possibilita a implementação de um algoritmo rápido e simples, porém com ótimos resultados, como será visto no capítulo 4.

3.5.1 Invariância à transformações lineares

Será analisado agora a invariância à mudanças de escala para a métrica utilizando distância de Mahalanobis. Para esta análise, vamos substituir a VA X por uma variável Y que seja uma transformação linear da VA X dada por

$$\mathbf{y} = \mathbf{A}\,\mathbf{x} + \mathbf{b}. \tag{3.5}$$

Tanto o conjunto de amostras da VA X quanto a de Y são separados em duas distribuições com média $\boldsymbol{\mu}_1^{(x)}$, $\boldsymbol{\mu}_2^{(x)}$, $\boldsymbol{\mu}_1^{(y)}$ e $\boldsymbol{\mu}_2^{(y)}$. Estas médias correspondem as médias da primeira e segunda distribuição que compõem a VA X e as médias da primeira e segunda distribuição que compõem a VA Y. O mesmo se aplica para as matrizes de covariância.

A distância de Mahalanobis entre as distribuições C_1 e C_2 da VA X é dada por

$$d_m^{(y)}(C_1, C_2) = (\boldsymbol{\mu}_1^{(y)} - \boldsymbol{\mu}_2^{(y)})^t \left(\boldsymbol{\Sigma}_1^{(y)} + \boldsymbol{\Sigma}_2^{(y)}\right)^{-1} (\boldsymbol{\mu}_1^{(y)} - \boldsymbol{\mu}_2^{(y)}) \tag{3.6}$$

De acordo com a relação estabelecida na seção 2.3.2, podemos escrever as médias e

matrizes de covariância da VA Y como função das médias e matrizes de covariância da VA X. Fazendo esta relação obtemos a seguinte expressão para a distância de Mahalanobis para a variável Y.

$$d_m^{(y)}(C_1, C_2) = \left((\boldsymbol{A}\boldsymbol{\mu}_1^{(x)} + \boldsymbol{b}) - (\boldsymbol{A}\boldsymbol{\mu}_2^{(x)} + \boldsymbol{b})\right)^t \left(\boldsymbol{A}\boldsymbol{\Sigma}_1^{(x)}\boldsymbol{A}^t + \boldsymbol{A}\boldsymbol{\Sigma}_2^{(x)}\boldsymbol{A}^t\right) \left((\boldsymbol{A}\boldsymbol{\mu}_1^{(x)} + \boldsymbol{b}) - (\boldsymbol{A}\boldsymbol{\mu}_2^{(x)} + \boldsymbol{b})\right)$$

Isolando os termos comuns na equação obtemos

$$d_m^{(y)}(C_1, C_2) = \left(\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)}\right)^t \boldsymbol{A}^t \left(\boldsymbol{A}\left(\boldsymbol{\Sigma}_1^{(x)} + \boldsymbol{\Sigma}_2^{(x)}\right)\boldsymbol{A}^t\right)^{-1} \boldsymbol{A}\left(\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)}\right)$$

$$d_m^{(y)}(C_1, C_2) = \left(\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)}\right)^t \boldsymbol{A}^t\boldsymbol{A}^{-t} \left(\boldsymbol{\Sigma}_1^{(x)} + \boldsymbol{\Sigma}_2^{(x)}\right)^{-1} \boldsymbol{A}^{-1}\boldsymbol{A}\left(\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)}\right).$$

Para uma transformação linear seja aplicável a um conjunto, a mesma deve ser inversível. Isto garante que não serão perdidas informações quando o conjunto for submetido a tais transformações. Desta maneira, temos de fato que o termo que sobra é igual a distância de Mahalanobis para o conjunto X

$$d_m^{(y)} = d_m^{(x)}, \tag{3.7}$$

comprovando assim, que se um conjunto for submetido a uma transformação linear inversível, não observa-se mudanças na métrica. Isto torna a métrica de Mahalanobis uma boa candidata a ser utilizada como métrica em técnicas de *clustering*.

3.6 Método utilizando distância de Bhattacharyya

Uma alternativa à distância de Mahalanobis para medir a distância ou dissimilaridade entre duas gaussianas é a distância de Bhattacharyya (seção 2.8.5). Para o caso onde os dois *clusters* são considerados gaussianos, que é o caso deste trabalho, a distância de Bhattacharyya pode ser utilizada diretamente como medida de dissimilaridade.

Nesta abordagem também é utilizada uma distância limiar d_t para decidir se

as duas gaussianas em questão serão ligadas ou não. Na literatura, a distância de Bhattacharyya, embora não muito comum, tem sido utilizada com sucesso em outros algoritmos de *clustering* [90]. Os parâmetros utilizados para realizar esta medida são as médias e matrizes de covariância das duas gausssianas.

O objetivo, assim como na distância de Mahalanobis, é medir a "semelhança" entre dois *clusters* para decidir se estes fazem parte do mesmo conjunto. A expressão para o cálculo da distância de Bhattacharyya já foi mostrada na equação 2.52 na seção 2.8.5.

Podemos observar que a distância é composta por dois termos, o primeiro contribui para medir a distância espacial entre as gaussianas representada pela diferença entre as médias. Esta diferença é ponderada pelo inverso da média das matrizes de covariância, o que tem efeito de normalizar os valores da diferença. O outro termo é função apenas das matrizes de covariância e é responsável por relacionar as formas das gaussianas, direção de maior variância, etc.

Pode-se observar que esta medida é sempre maior ou igual a zero, sendo zero apenas quando as duas gaussianas forem iguais. Na seção seguinte iremos verificar a invariância à transformações lineares da distância de Bhattacharyya.

3.6.1 Invariância à transformações lineares

Voltamos a frisar que para uma métrica ser utilizada em algoritmos de *clustering*, esta deve ser invariante à transformações lineares, resultando assim em resultados iguais, independente da escala dos dados.

Sejam os conjuntos X e Y como definidos na seção anterior. Y representa o mesmo conjunto X a menos de uma transformação linear. A distância de Bhattacharyya entre as distribuições C_1 e C_2 da VA Y é dada por

$$d_B(C_1, C_2) = \frac{1}{8}(\boldsymbol{\mu}_1^{(y)} - \boldsymbol{\mu}_2^{(y)})^t \left[\frac{\boldsymbol{\Sigma}_1^{(y)} + \boldsymbol{\Sigma}_2^{(y)}}{2}\right] (\boldsymbol{\mu}_1^{(y)} - \boldsymbol{\mu}_2^{(y)}) + \frac{1}{2}\log \frac{\left|\frac{\boldsymbol{\Sigma}_1^{(y)} + \boldsymbol{\Sigma}_2^{(y)}}{2}\right|}{\sqrt{\left|\boldsymbol{\Sigma}_1^{(y)}\right| \left|\boldsymbol{\Sigma}_2^{(y)}\right|}}. \tag{3.8}$$

Da mesma maneira que foi feito para a distância de Mahalanobis, substituímos os parâmetros da VA Y como função dos parâmetros de X. A expressão pode ser escrita como

$$d_B(C_1,C_2) = \frac{1}{8}(\boldsymbol{A}\boldsymbol{\mu}_1^{(x)} + \boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu}_2^{(x)} - \boldsymbol{b})^t \left[\frac{\boldsymbol{A}^t\boldsymbol{\Sigma}_1^{(x)}\boldsymbol{A} + \boldsymbol{A}^t\boldsymbol{\Sigma}_2^{(x)}\boldsymbol{A}}{2}\right]^{-1} (\boldsymbol{A}\boldsymbol{\mu}_1^{(x)} + \boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu}_2^{(x)} - \boldsymbol{b}) + \\ + \frac{1}{2}\log \frac{\left|\frac{\boldsymbol{A}^t\boldsymbol{\Sigma}_1^{(x)}\boldsymbol{A}+\boldsymbol{A}^t\boldsymbol{\Sigma}_2^{(x)}\boldsymbol{A}}{2}\right|}{\sqrt{\left|\boldsymbol{A}^t\boldsymbol{\Sigma}_1^{(x)}\boldsymbol{A}\right|\ \left|\boldsymbol{A}^t\boldsymbol{\Sigma}_2^{(x)}\boldsymbol{A}\right|}} .$$

Simplificando e isolando as constantes da transformação obtemos

$$d_B(C_1,C_2) = \frac{1}{8}(\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)})^t \boldsymbol{A}^t \left[\frac{\boldsymbol{A}^t\left(\boldsymbol{\Sigma}_2^{(x)} + \boldsymbol{\Sigma}_1^{(x)}\right)\boldsymbol{A}}{2}\right]^{-1} \boldsymbol{A}(\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)}) + \\ + \frac{1}{2}\log \frac{\left|\boldsymbol{A}^t\right|\left|\frac{\left(\boldsymbol{\Sigma}_2^{(x)}+\boldsymbol{\Sigma}_1^{(x)}\right)}{2}\right|\ |\boldsymbol{A}|}{\left|\boldsymbol{A}^t\right|\sqrt{\left|\boldsymbol{\Sigma}_1^{(x)}\right|\ \left|\boldsymbol{\Sigma}_2^{(x)}\right|}\ |\boldsymbol{A}|}$$

que finalmente torna-se

$$d_B(C_1,C_2) = \frac{1}{8}(\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)})^t \left[\frac{\boldsymbol{\Sigma}_2^{(x)} + \boldsymbol{\Sigma}_1^{(x)}}{2}\right]^{-1} (\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)}) + + \frac{1}{2}\log \frac{\left|\frac{\boldsymbol{\Sigma}_2^{(x)}+\boldsymbol{\Sigma}_1^{(x)}}{2}\right|}{\sqrt{\left|\boldsymbol{\Sigma}_1^{(x)}\right|\ \left|\boldsymbol{\Sigma}_2^{(x)}\right|}} \tag{3.9}$$

que de fato é igual a distância de Bhattacharyya da variável X.

3.7 Método utilizando divergência de Kullback-Leibler

Neste método, a divergência utilizada para medir a separação entre as gaussianas foi a divergência de Kullback-Leibler apresentada na seção 2.8.2.

Nesta abordagem, assim como nas anteriores, d_t é escolhido para servir de limiar de ligação entre as gaussianas. Duas gaussianas as quais a divergência forem menores que o limiar d_t, serão consideradas ligadas. Esta métrica já foi utilizada na literatura [52] como base para métricas baseadas na entropia de Shannon. Sua aplicação neste trabalho será restrita ao caso gaussiano.

Para o caso onde as duas distribuições envolvidas no cálculo da divergência de Kullback-Leibler sejam gaussianas, a expressão para a divergência entre dois conjuntos C_1 e C_2 pode ser dada pela equação 3.10 [51] [49]

$$d_{KL}(C_1, C_2) = \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2^{-1} - \boldsymbol{I}\right) - \frac{1}{2}\log\left(\left|\boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2^{-1}\right|\right) \tag{3.10}$$

Nesta equação, as variáveis $\boldsymbol{\mu}$ e $\boldsymbol{\Sigma}$ (com seus respectivos índices) representam os parâmetros das gaussianas (médias e matrizes de covariância). $|.|$ é o operador de determinante e tr(.) representa o traço de uma matrix e $\mathbf{I}$ é uma matriz identidade de mesma ordem das matrizes de covariância.

Como pode-se observar, a medida da divergência KL é uma medida não simétrica. Para torná-la simétrica podemos somar a contribuição da divergência de C_1 para C_2 com a divergência de C_2 para C_1. Desta maneira, a forma simétrica da divergência KL pode ser dada por

$$d_{KL_S}(C_1, C_2) = \frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_1 \boldsymbol{\Sigma}_2^{-1} + \boldsymbol{\Sigma}_2 \boldsymbol{\Sigma}_1^{-1} - 2\boldsymbol{I}\right). \tag{3.11}$$

Esta é a métrica realmente utilizada nesta abordagem. De maneira semelhante a métrica de Mahalanobis, a divergência KL depende da separação espacial entre as

gaussianas, representada pela diferença das médias. Esta é ponderada pelo inverso das gaussianas. Nesta abordagem, o limiar é geralmente maior devido a ponderação com a soma dos inversos das matrizes de covariância (contra o inverso da soma na distância de Mahalanobis). Mesmo assim, a tendência a ponderar mais as separações que não estão na direção de maior variância ainda existe. Outra diferença em relação à métrica de Mahalanobis é a presença de um segundo termo que envolve o traço das matrizes de covariância. Este termo funciona como uma medida de diferença entre as matrizes de covariância, mesmo quando as mesmas estão na mesma posição (média zero). Esta diferença será zero quando as gaussianas estiverem "alinhadas" ou paralelas. Este termo exerce uma influência importante na decisão de ligar *clusters* próximos porém com matrizes de covariância diferentes.

Outra diferença muito importante é a presença da dependência direta da dimensão dos dados. Esta dependência está representada pela presença da matriz identidade no segundo termo da equação. Por estar contabilizada no traço das matrizes de covariância, esta resulta na soma de D unidades (D sendo a dimensão dos dados) no cálculo da divergência. Esta contribuição é pequena quando comparada com a diferença dada pela separação das gaussianas e não constitui um problema na escolha do limiar.

Assim como a abordagem utilizando a distância de Mahalanobis, o método utilizando a divergência de Kullback-Leibler também encontra-se publicado [61] e constitui mais uma contribuição deste trabalho. Foi publicada também uma análise comparativa [59] das duas técnicas, utilizando ambas as métricas. Esta comparação será detalhada no capítulo 4.

3.7.1 Invariância à transformações lineares

Assim como na análise de invariância da distância de Mahalanobis e Bhattacharyya, será calculada a divergência KL para dois conjuntos, sendo um deles uma transformação linear do outro. Seja a VA Y formada por uma transformação linear da VA X, como já mostrado na equação 3.5. A divergência KL da VA Y é dada por

$$d^{(y)}_{KL_S}(C_1,C_2) = \frac{1}{2}(\boldsymbol{\mu}_1^{(y)} - \boldsymbol{\mu}_2^{(y)})^t \left(\left(\boldsymbol{\Sigma}_1^{(y)}\right)^{-1} + \left(\boldsymbol{\Sigma}_2^{(y)}\right)^{-1} \right) (\boldsymbol{\mu}_1^{(y)} - \boldsymbol{\mu}_2^{(y)}) + \\ + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{(y)} \left(\boldsymbol{\Sigma}_2^{(y)}\right)^{-1} + \boldsymbol{\Sigma}_2^{(y)} \left(\boldsymbol{\Sigma}_1^{(y)}\right)^{-1} - 2\boldsymbol{I} \right)$$

Substituindo os parâmetros da VA Y pelos seus equivalente em X (como feito para a distância de Mahalanobis) obtemos

$$d^{(y)}_{KL_S}(C_1,C_2) = \frac{1}{2}(\boldsymbol{A}\boldsymbol{\mu}_1^{(x)} + \boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu}_2^{(x)} - \boldsymbol{b})^t \left(\left(\boldsymbol{A}\boldsymbol{\Sigma}_1^{(x)}\boldsymbol{A}^t\right)^{-1} + \left(\boldsymbol{A}\boldsymbol{\Sigma}_2^{(x)}\boldsymbol{A}^t\right)^{-1} \right) (\boldsymbol{A}\boldsymbol{\mu}_1^{(x)} + \boldsymbol{b} - \boldsymbol{A}\boldsymbol{\mu}_2^{(x)} - \boldsymbol{b}) + \\ + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{A}\boldsymbol{\Sigma}_1^{(x)}\boldsymbol{A}^t \left(\boldsymbol{A}\boldsymbol{\Sigma}_2^{(x)}\boldsymbol{A}^t\right)^{-1} + \boldsymbol{A}\boldsymbol{\Sigma}_2^{(x)}\boldsymbol{A}^t \left(\boldsymbol{A}\boldsymbol{\Sigma}_1^{(x)}\boldsymbol{A}^t\right)^{-1} - 2\boldsymbol{I} \right)$$

que simplificando e rearrumando alguns termos obtem-se

$$d^{(y)}_{KL_S}(C_1,C_2) = \frac{1}{2}(\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)})^t \boldsymbol{A}^t \left(\boldsymbol{A}^{-t} \left(\left(\boldsymbol{\Sigma}_1^{(x)}\right)^{-1} + \left(\boldsymbol{\Sigma}_2^{(x)}\right)^{-1} \right) \boldsymbol{A}^{-1} \right) \boldsymbol{A}(\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)}) + \\ + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{A}\boldsymbol{\Sigma}_1^{(x)} \left(\boldsymbol{\Sigma}_2^{(x)}\right)^{-1} \boldsymbol{A}^{-1} \right) + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{A}\boldsymbol{\Sigma}_2^{(x)} \left(\boldsymbol{\Sigma}_1^{(x)}\right)^{-1} \boldsymbol{A}^{-1} \right) - \mathrm{tr}\left(\boldsymbol{I}\right).$$

Como estamos tratando de transformações inversíveis, podemos simplificar obtendo

$$d^{(y)}_{KL_S}(C_1,C_2) = \frac{1}{2}(\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)})^t \left(\left(\boldsymbol{\Sigma}_1^{(x)}\right)^{-1} + \left(\boldsymbol{\Sigma}_2^{(x)}\right)^{-1} \right) (\boldsymbol{\mu}_1^{(x)} - \boldsymbol{\mu}_2^{(x)}) + \\ + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{(x)} \left(\boldsymbol{\Sigma}_2^{(x)}\right)^{-1} \right) + \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}_2^{(x)} \left(\boldsymbol{\Sigma}_1^{(x)}\right)^{-1} \right) - \mathrm{tr}\left(\boldsymbol{I}\right),$$

que de fato é igual a divergência entre os conjuntos na VA X.

3.8 Método baseado em medidas de negentropia

Na seção 2.8.4 foi introduzido o conceito de negentropia de um sinal. Nesta seção será mostrado como utilizar a negentropia de um conjunto formado por duas

distribuições para calcular a dissimilaridade entre estas distribuições. Este método constitui mais uma contribuição deste trabalho e será explorado nas próximas seções. O método tem como base a divergência entre a distribuição do conjunto e um modelo de uma única gaussiana que descreve todo o conjunto de dados. Este é o conceito já introduzido na seção 2.8.3, que será formalizado para o caso onde temos um conjunto modelado por uma mistura de duas gaussianas.

A negentropia do conjunto nos diz o quanto a distribuição do mesmo se aproxima de uma distribuição gaussiana. Para utilizar este conceito como medida de divergência, consideraremos aqui um conjunto formado por duas distribuições gaussianas. Este conjunto porém, pode ser modelado com apenas uma gaussiana com média e matriz de covariância. Este raciocínio equivale a modelar duas gaussianas utilizando apenas uma média e uma matriz de covariância. A Figura 3.11 ilustra diversos casos bi-dimensionais. As linhas cheias representam as duas gaussianas originais que modelam conjunto e as linhas tracejadas, uma gaussiana que modela o conjunto todo.

Como pode-se observar, nos casos onde as gaussianas originais são próximas ou estão alinhadas, a gaussiana equivalente modela bem o conjunto todo. Quanto mais diferem as gaussianas originais, pior é o modelo com uma única gaussiana. Quanto mais bem modelado por uma gaussiana o conjunto todo for, menor a sua negentropia, ou seja, quanto mais parecidas ou alinhadas forem as gaussianas originais, menor será a negentropia total do conjunto de dados.

Comparando um modelo de duas gaussianas com o modelo de uma gaussiana apenas (medindo assim a negentropia do conjunto) propomos uma métrica para media a divergência entre duas gaussianas. A métrica é bastante simples. Dados dois conjuntos de dados, modela-se cada um como uma gaussiana e gera-se uma mistura para o conjunto como um todo. A divergência medida entre estes dois conjuntos será a negentropia do conjunto formado por ambos.

Definida a nova métrica, resta agora encontrar uma forma prática e eficiente de calcular a negentropia do conjunto para que esta possa compor uma abordagem para o problema de *clustering*. Nas próximas seções, serão tratadas diversas formas para o cálculo da negentropia, ressaltando as suas vantagens e desvantagens.

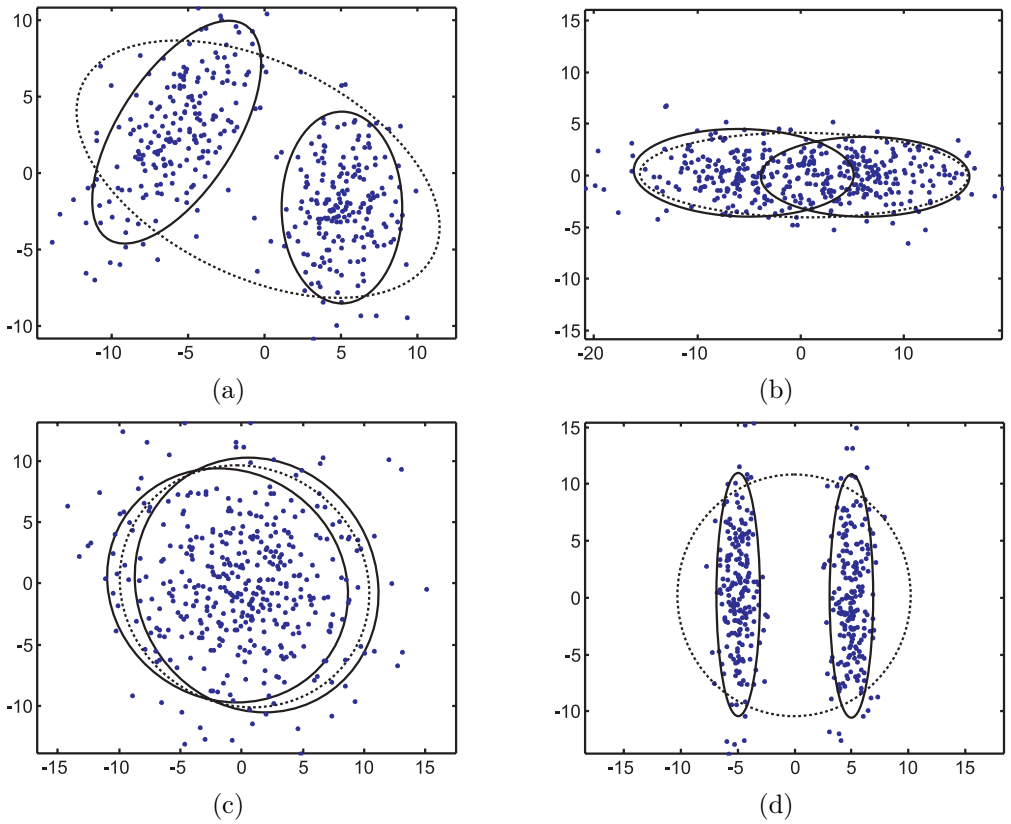


Figura 3.11: Conjuntos modelados como misturas de gaussianas e como uma única gaussiana.

3.8.1 Cálculo da negentropia

O cálculo da negentropia de um conjunto de dados, envolve o conhecimento da distribuição do conjunto e a estimação de momentos de alta ordem como visto na seção 2.8.4 do capítulo 2. Neste trabalho, estudaremos o caso particular para o uso no método de *clustering* proposto. Para este caso, serão propostas formas para o cálculo da negentropia levando em consideração as informações *a priori* que temos sobre o conjunto.

Podemos utilizar o conhecimento que temos sobre o conjunto para facilitar o cálculo da negentropia. A primeira informação importante é que o conjunto em questão está dividido em dois subconjuntos. Este fato decorre da natureza do método proposto. Lembramos que inicialmente os dados originais são quantizados em pequenos *clusters* auxiliares e depois, cada par é submetido a uma medida de dissimilaridade. Este "par" de *clusters* são justamente os dois subconjuntos que formam o conjunto no qual estamos interessados em medir a negentropia. O

fato de termos dois subconjuntos, facilita o cálculo da negentropia no sentido de podermos modelar diretamente o conjunto por uma mistura de duas gaussianas (cada uma estimada a partir de um subconjunto). Outra informação importante é que quantidade de pontos em cada subconjunto é conhecida, o que possibilita a estimativa da probabilidade *a priori* de cada gaussiana da mistura.

Uma vez modelado o conjunto, com uma mistura de gaussianas, temos disponível os parâmetros da mistura: P_1 e P_2 que são as probabilidades *a priori* de cada gaussiana, $\boldsymbol{\mu}_1$ e $\boldsymbol{\mu}_2$ que são as médias e $\boldsymbol{\Sigma}_1$ e $\boldsymbol{\Sigma}_2$ que são as matrizes de covariância. O próximo passo é estimar a média e matriz de covariância do conjunto todo (supondo que o mesmo seja modelado por apenas uma gaussiana).

Com os parâmetros da mistura e os parâmetros da gaussiana associada ao conjunto resultante da união, a negentropia pode ser calculada medindo-se a divergência (por alguma das técnicas já utilizadas) entre a mistura e a gaussiana individual.

3.8.1.1 Média e covariância de uma mistura

As estimativas da média e da matriz de covariância no conjunto modelado pela mistura de gaussianas podem ser obtidas calculando-se analiticamente o segundo momento de uma distribuição formada pela mistura. Este procedimento visa reduzir o custo computacional envolvido já que a forma analítica consiste no cálculo direto da média e da matriz de covariância do conjunto, como função dos parâmetros da mistura. Em outras palavras, podemos calcular a média e matriz de covariância "equivalentes" de uma mistura, dadas as suas médias, matrizes de covariância e probabilidades *a priori*. O objetivo desta seção é proceder este cálculo, para que o resultado seja utilizado nas seções que se seguem.

A média de uma distribuição é dada pelo seu primeiro momento, que por sua vez é dado pela equação [68] [73]

$$\boldsymbol{\mu} = \int \cdots \int \boldsymbol{x}\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}. \tag{3.12}$$

Onde $p(\boldsymbol{x})$ é a distribuição da VA X e a integração deve ser feita em todo o espaço

da VA.

Para uma distribuição composta por uma mistura de gaussianas (ver seção 2.4), o primeiro momento é calculado utilizando a seguinte expressão

$$\boldsymbol{\mu} = \int \cdots \int \boldsymbol{x} \left(\sum_{i=1}^{N_g} P_i \, k_i \, e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^t \, \boldsymbol{\Sigma}_i^{-1} \, (\boldsymbol{x}-\boldsymbol{\mu}_i)} \right) \mathrm{d}\boldsymbol{x}.$$

Onde k_i é uma constante de normalização da integral gaussiana (ver seção 2.3). Como não há variáveis de integração nos índices do somatório, podemos permutá-lo com a integral, resultando em

$$\boldsymbol{\mu} = \sum_{i=1}^{N_g} P_i \int \cdots \int \boldsymbol{x} \, k_i \, e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^t \, \boldsymbol{\Sigma}_i^{-1} \, (\boldsymbol{x}-\boldsymbol{\mu}_i)} \mathrm{d}\boldsymbol{x}.$$

Observa-se que as integrais restantes são o primeiro momento de cada gaussiana individual da mistura, correspondendo de fato às suas médias. Com isto, o primeiro momento ou média da mistura é dada por

$$\boldsymbol{\mu} = \sum_{i=1}^{N_g} P_i \, \boldsymbol{\mu}_i. \tag{3.13}$$

Observando a equação 3.13 percebe-se que a média resultante de uma mistura é a soma ponderada pelas probabilidades *a priori* das médias de cada gaussiana.

O mesmo pode ser feito para o cálculo da matriz de covariância, que é dada pelo segundo momento central definido pela equação 3.14

$$\boldsymbol{\Sigma} = \int \cdots \int (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^t \, p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{3.14}$$

onde $\boldsymbol{\mu}$ é a média da distribuição. Com $p(\boldsymbol{x})$ sendo uma mistura de gaussianas temos

$$\boldsymbol{\Sigma} = \int \cdots \int (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^t \left(\sum_{i=1}^{N_g} P_i \, k_i \, e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^t \, \boldsymbol{\Sigma}_i^{-1} \, (\boldsymbol{x}-\boldsymbol{\mu}_i)} \right) \mathrm{d}\boldsymbol{x}.$$

Trocando a ordem do somatório com a integração obtemos

$$\boldsymbol{\Sigma} = \sum_{i=1}^{N_g} P_i \int \cdots \int (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^t \, k_i \, e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^t \, \boldsymbol{\Sigma}_i^{-1} \, (\boldsymbol{x}-\boldsymbol{\mu}_i)} \mathrm{d}\boldsymbol{x}.$$

Fazendo $\boldsymbol{y}_i = \boldsymbol{x} - \boldsymbol{\mu}_i$ e substituindo, obtemos

$$\boldsymbol{\Sigma} = \sum_{i=1}^{N_g} P_i \int \cdots \int (\boldsymbol{y}_i + \boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{y}_i + \boldsymbol{\mu}_i - \boldsymbol{\mu})^t k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i,$$

que expandindo torna-se

$$\boldsymbol{\Sigma} = \sum_{i=1}^{N_g} P_i \left[\begin{array}{l} \int \cdots \int \boldsymbol{y}_i \boldsymbol{y}_i^t k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i + \\ + \int \cdots \int \boldsymbol{y}_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})^t k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i + \\ + \int \cdots \int (\boldsymbol{\mu}_i - \boldsymbol{\mu}) \boldsymbol{y}_i^t k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i + \\ + \int \cdots \int (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^t k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i \end{array} \right]$$

$$\boldsymbol{\Sigma} = \sum_{i=1}^{N_g} P_i \left[\begin{array}{l} \int \cdots \int \boldsymbol{y}_i \boldsymbol{y}_i^t k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i + \\ + (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^t \left(\int \cdots \int k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i \right) + \\ + \left(\int \cdots \int \boldsymbol{y}_i k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i \right) (\boldsymbol{\mu}_i - \boldsymbol{\mu})^t + \\ + (\boldsymbol{\mu}_i - \boldsymbol{\mu}) \left(\int \cdots \int \boldsymbol{y}_i^t k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i \right) \end{array} \right].$$

Nesta última expressão, os dois últimos termos correspondem ao primeiro momento da uma VA Y que tem média zero, sendo portanto igual a zero. Resta então a integral

$$\boldsymbol{\Sigma} = \sum_{i=1}^{N_g} P_i \left[\int \cdots \int \boldsymbol{y}_i \boldsymbol{y}_i^t k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i + (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^t \left(\int \cdots \int k_i e^{-\frac{1}{2}\boldsymbol{y}_i^t \boldsymbol{\Sigma}_i^{-1} \boldsymbol{y}_i} \mathrm{d}\boldsymbol{y}_i \right) \right].$$

Por observação, vemos que o primeiro termo desta integral corresponde ao segundo momento individual de cada gaussiana, sendo portanto igual as suas matrizes de covariância. O segundo termo envolve a integral de cada gaussiana, que de fato é igual a unidade[2]. Segue portanto que a matriz de covariância de uma mistura é dada por

[2]Propriedade fundamental das distribuições de probabilidade

$$\boldsymbol{\Sigma} = \sum_{i=1}^{N_g} P_i \left[\boldsymbol{\Sigma}_i + (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^t\right]. \tag{3.15}$$

É interessante notar que a matriz equivalente, é resultado da soma ponderada das matrizes de covariância e uma estimativa envolvendo a diferença das médias para a média da mistura. Isto é importante porque quanto mais afastadas as gaussianas estiverem, maior será a covariância equivalente.

Neste trabalho estamos interessados na média e matriz de covariância de uma mistura de duas gaussianas. Para o caso com apenas duas gaussianas, a média e matriz de covariância equivalente são dadas pelas equações 3.16 e 3.17.

$$\boldsymbol{\mu} = P_1 \boldsymbol{\mu}_1 + P_2 \boldsymbol{\mu}_2 \tag{3.16}$$

$$\boldsymbol{\Sigma} = P_1 \boldsymbol{\Sigma}_1 + P_2 \boldsymbol{\Sigma}_2 + P_1 P_2 (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t \tag{3.17}$$

3.8.2 Divergência KL e Entropia de Shannon

Uma das maneiras de calcular a negentropia pela abordagem proposta aqui é utilizar uma medida de divergência entre a distribuição original e a distribuição modelada com uma gaussiana. Nesta seção será discutido o uso da divergência de Kullback-Leibler baseada na entropia de Shannon. Para que se possa medir a divergência entre o conjunto e seu modelo gaussiano, devemos obter o modelo da distribuição original. Serão discutidos dois métodos para o cálculo da divergência, utilizando duas abordagens para o modelo dos dados originais (que serão comparados com o modelo de uma única gaussiana).

Em ambas as abordagens temos dois conjuntos X e Y que formarão um único conjunto que denominaremos XY. O objetivo é medir a divergência KL entre a distribuição "real" de XY e seu modelo gaussiano, como mostra a equação 3.18

$$J = \int p_{XY}(\boldsymbol{x}) \log \left(\frac{p_{XY}(\boldsymbol{x})}{g_{XY}(\boldsymbol{x})} \right) \mathrm{d}\boldsymbol{x}. \tag{3.18}$$

$p_{XY}(\boldsymbol{x})$ é o modelo da distribuição "real" do conjunto e $g_{XY}(\boldsymbol{x})$ é o modelo gaussiano.

A integração deve ser feita em todo o espaço da variável $\boldsymbol{x}$.

3.8.2.1 Modelo utilizando janelas de Parzen

Para o cálculo da divergência entre os dois conjuntos e seu modelo gaussiano equivalente, a distribuição do conjunto formado por ambos é estimada utilizando Janelas de Parzen [16]. A vantagem deste tipo de abordagem encontra-se na facilidade para o cálculo da estimativa da distribuição real. Outra vantagem reside no fato de que o modelo de Parzen não pressupõe uma forma para a distribuição, sendo possível a mesma apresentar qualquer forma. Embora estas sejam vantagens importantes, a grande desvantagem da utilização das janelas de Parzen é a necessidade de informar a variância utilizada no modelo (ver seção 2.5.2) e a maldição da dimensionalidade. Para que o modelo com janelas de Parzen apresente uma boa representação da distribuição dos dados, é necessário que se tenham muitos pontos, o que nem sempre é possível e, quando o é, torna o algoritmo lento.

O modelo utilizando janelas de Parzen é uma alternativa apenas para o caso onde os conjuntos não são gaussianos ou possuem matrizes de covariância mal condicionadas (com determinante próximo de zero). Casos não gaussianos envolvem distribuições com formas não elípticas enquanto matrizes de covariância mal condicionadas acontecem nos casos onde temos pontos coplanares ou com variância zero em uma das dimensões. Este aspecto foi explorado em trabalhos recentes [61] como um dos resultados parciais deste trabalho.

Os casos citados, onde é necessária a utilização de janelas de Parzen, são raros e alguns podem ser evitados sobre-estimando o número de conjuntos auxiliares na fase de quantização vetorial.

3.8.2.2 Modelo utilizando mistura de gaussianas

Uma alternativa à utilização do modelo utilizando janelas de Parzen é a utilização de misturas de gaussianas. Esta abordagem supõe que ambos os conjuntos X e Y são bem modelados por uma gaussiana cada um e que o conjunto completo XY seja modelado pela mistura destas duas gaussianas. Esta suposição é bastante razoável, já que na fase de quantização vetorial são utilizados uma quantidade sobre estimada

de centros. Além do mais, mesmo que o conjunto não seja totalmente elíptico (ou hiper-elíptico para o caso de altas dimensões) uma aproximação elíptica é suficiente.

Nesta abordagem a divergência KL é calculada entre a mistura de gaussianas formada pelas gaussianas individuais dos conjuntos X e Y e o modelo gaussiano de todo o conjunto XY. A equação 3.19 expressa a divergência a ser calculada

$$J = \int \left(P_X \mathrm{N}(\boldsymbol{x}, \boldsymbol{\mu}_X, \boldsymbol{\Sigma}_Y) + P_Y \mathrm{N}(\boldsymbol{x}, \boldsymbol{\mu}_Y, \boldsymbol{\Sigma}_Y)\right) \log \left(\frac{P_X \mathrm{N}(\boldsymbol{x}, \boldsymbol{\mu}_X, \boldsymbol{\Sigma}_X) + P_Y \mathrm{N}(\boldsymbol{x}, \boldsymbol{\mu}_Y, \boldsymbol{\Sigma}_Y)}{\mathrm{N}(\boldsymbol{x}, \boldsymbol{\mu}_{XY}, \boldsymbol{\Sigma}_{XY})} \right) \mathrm{d}\boldsymbol{x} \tag{3.19}$$

Onde $\mathrm{N}(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ representa uma gaussiana com média e matriz de covariância dadas.

A vantagem deste método está no fato de que o modelo é paramétrico e os parâmetros são facilmente calculados. Este método possui custo computacional reduzido quando comparado com o método utilizando janelas de Parzen [61] já que não são necessários muitos pontos para realizar uma boa estimativa. Além do mais, não existe parâmetro livre a ser escolhido.

A desvantagem deste método está no fato de que o cálculo analítico torna-se difícil devido ao logaritmo da soma, e soluções aproximadas requerem a suposição de que as probabilidades sejam bem distintas, o que na maioria dos casos não é verdade.

Novamente precisamos integrar em todo o espaço da variável x para encontrar o valor da divergência. Este procedimento torna o algoritmo menos eficiente que as abordagens que não utilizam medidas de negentropia como o uso da distância de Mahalanobis ou a divergência KL direta (entre as duas gaussianas).

A vantagem da utilização da negentropia está no fato desta considerar os dois conjuntos que estão sendo testados como uma única distribuição, levando em conta assim, a diferença entre a quantidade de pontos de cada um. Esta consideração é garantida com a utilização das probabilidades *a priori* P_x e P_y. Isto é importante porque nos casos em que temos, por exemplo, um conjunto X com apenas 10 pontos e um conjunto Y com 400 pontos o peso da probabilidade *a priori* do segundo conjunto fará com que a divergência seja determinada predominantemente por este como se todo o conjunto fosse formado por ele. Em geral este é um caso onde a divergência

deve ter valor baixo, pois os 10 pontos do primeiro conjunto provavelmente são pontos fora da estatística e devem ser unidos ao conjunto com mais pontos.

3.8.2.3 Integração de Monte Carlo

Os métodos para calcular a dissimilaridade entre dois conjuntos utilizando negentropia utilizados até o momento necessitam de um método de integração numérica. O método de integração utilizado ara testes neste trabalho é o método de Monte Carlo. Trata-se de um método estocástico para o cálculo de integrais definidas da forma [80] [76]

$$I = \int_a^b f(x)\mathrm{d}x, \tag{3.20}$$

que obviamente se aplica para integrais múltiplas, como o é o objetivo neste trabalho.

No método de Monte Carlo, a integral é aproximada utilizando n pontos aleatórios dentro do volume de integração, conforme a seguinte expressão

$$I \approx \frac{V}{n}\sum_{i=1}^{n} f(x_i) \tag{3.21}$$

onde V é o volume do espaço de integração, e os valores de x_i são escolhidos aleatoriamente e uniformemente distribuídos dentro deste volume. Trata-se de um método muito simples de integração, o que torna o método bastante prático.

Infelizmente os limites de integração para o caso das integrais utilizadas neste trabalho são impróprios $(-\infty, \infty)$ e é impossível calcular numericamente as integrais. A solução para este problema está em se utilizar da propriedade de decaimento rápido das distribuições de probabilidade e estipular um espaço limitado por um hiper-volume finito que englobe maior parte dos valores relevantes da distribuição em questão. A escolha dos limites deste hiper-volume pode ser feita empiricamente e normalmente se utiliza de 3 a 4 vezes a maior variância presente dos dados.

A desvantagem deste método de integração, também presente em qualquer método de integração numérica, é a quantidade de pontos utilizados para o cálculo. Para que se tenha uma boa aproximação do valor real da integral, devem ser

utilizados muitos pontos. Em compensação, quanto mais pontos, mais lento o cálculo se torna. Este problema é mais acentuado quando a dimensão dos dados aumenta. Para que se tenha um valor representativo da integral em altas dimensões, devem ser utilizados muitos pontos.

A Figura 3.12 ilustra como se dá o processo de integração para um caso bi-dimensional. Os pontos abaixo da função são os pontos utilizados para a estimativa da integral. O volume V neste caso será área quadrada onde os pontos estão concentrados.

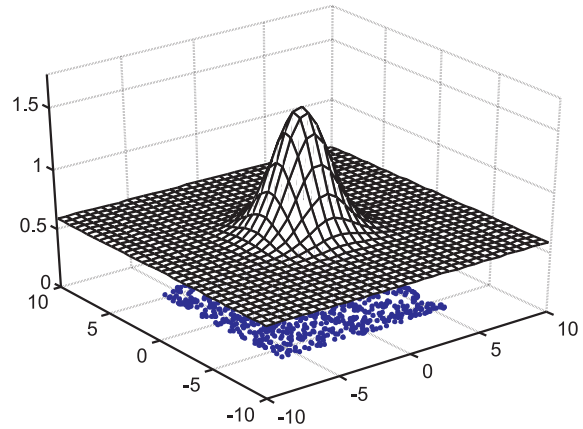


Figura 3.12: Ilustração do método de Monte Carlo

3.8.3 Entropia de Rényi

Uma alternativa ao uso da divergência KL para calcular, utilizando a negentropia, a dissimilaridade de um conjunto consiste na utilização da entropia quadrática de Rényi como medida direta na expressão da negentropia. Como resultado não obtemos uma medida direta da negentropia, porem conseguimos uma métrica alternativa baseada na forma como a negentropia é definida.

Neste caso, utilizaremos a abordagem onde comparamos a mistura de gaussianas geradas pelos conjuntos separados X e Y com o modelo gaussiano do conjunto completo XY. Esta abordagem para o cálculo baseado na negentropia e sua utilização para medir a divergência entre duas distribuições consiste em mais uma contribuição deste trabalho e de fato é uma das mais importantes. Em resumo, estamos interessados em medir diferença entre uma distribuição gaussiana e uma

mistura de duas distribuições também gaussianas. Os dados fornecidos são os parâmetros de uma mistura de duas gaussianas com médias $\boldsymbol{\mu}_1$ e $\boldsymbol{\mu}_2$, matrizes de covariância $\boldsymbol{\Sigma}_1$ e $\boldsymbol{\Sigma}_2$ e probabilidades *a priori* P_1 e P_2 e um modelo gaussiano da distribuição completa $\boldsymbol{\mu}_q$ e $\boldsymbol{\Sigma}_q$. Mostraremos que esta medida pode ser calculada pela expressão 3.22

$$\log\left(P_1^2\sqrt{\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_1|}} + \frac{2P_1P_2\sqrt{|\boldsymbol{\Sigma}_q|}e^{-\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t(\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)}}{\sqrt{\left|\frac{\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2}{2}\right|}} + P_2^2\sqrt{\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_2|}}\right) \tag{3.22}$$

Podemos notar diversos pontos interessantes nesta medida, que a tornam uma boa candidata a ser utilizada como medida de divergência. Podemos citar:

- Simetria. Esta medida é uma medida simétrica e não necessita de ser somada ao seu complemento, como é o caso da divergência KL. A simetria pode ser observada facilmente trocando-se os índices dos parâmetros das gaussianas,
- Forma analítica definida. Como a integração já foi feita analiticamente, o cálculo da divergência é direto e não necessita de algoritmo de integração numérica. Isto torna o algoritmo bastante rápido,
- Considera as probabilidades *a priori*. Como explicado anteriormente, é importante considerar a quantidade de pontos de cada conjunto na medida de divergência,
- Invariante a transformações lineares. Como será mostrado adiante, esta métrica é invariante a transformações lineares, o que torna possível seu uso em algoritmos de *clustering*,
- Sem problemas quanto a dimensão dos dados. Ao contrário do modelo utilizando janelas de Parzen, esta métrica não tem problemas quanto a dimensão dos dados, bastando que estes sejam suficientes para uma boa estimativa das matrizes de covariância.

A princípio, temos como desvantagem a complexidade da expressão quando comparada com outras medidas como a distância de Mahalanobis ou distância de Bhattacharyya. Porem, o aumento do custo computacional para uso no algoritmo de *clustering* proposto é muito pequeno em relação ao custo das outras fases.

É interessante notar que a nova métrica é composta por três termos. O primeiro e o último termo, contabilizam a diferença entre as matrizes de covariância de cada gaussiana que compõe a mistura($\mathbf{\Sigma}_1$ e $\mathbf{\Sigma}_2$), com a gaussiana equivalente do conjunto ($\mathbf{\Sigma}_q$). Lembrando que o termo $\mathbf{\Sigma}_q$ é a matriz de covariância da mistura vista como uma única gaussiana e pode ser calculado como função das duas variâncias e das médias das gaussianas individuais como visto na seção 3.17.

Analisando o termo central, observamos que o mesmo possui uma ponderação que tende a diminuir quando as médias se afastam (termo negativo na exponencial). Este efeito (que depende da dimensão dos dados) altera a forma com que a negentropia cresce quando as médias são separadas, aumentando ou diminuindo a intensidade do crescimento. Este termo logo se torna desprezível quando as médias são afastadas.

A expressão para esta nova métrica depende somente dos parâmetros da mistura, já que $\mathbf{\Sigma}_q$ pode ser escrita como função destes parâmetros. Embora possa ser escrita como função apenas dos parâmetros da mistura, em alguma aplicação, pode ser mais vantajoso estimar estes parâmetros diretamente a partir do conjunto. Por este motivo a expressão é deixada em função de $\mathbf{\Sigma}_q$.

A seguir será demonstrada a obtenção da métrica descrita nesta seção. Algumas passagens são bastante extensas e são deixadas para um apêndice no final do trabalho.

3.8.3.1 Demonstração 1

A forma proposta neste trabalho para o cálculo da nova métrica com base no calculo da negentropia utiliza a entropia de Rényi. A negentropia consiste na diferença entre as entropias da duas distribuições de probabilidade, portanto é dada pela expressão

$$J = H_g - H_p = \frac{1}{1-\alpha} \log \frac{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(\boldsymbol{x})^{\alpha} \mathrm{d}\boldsymbol{x}}{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} p(\boldsymbol{x})^{\alpha} \mathrm{d}\boldsymbol{x}} \tag{3.23}$$

Onde H_g é a entropia de um modelo gaussiano dos dados e H_p é a entropia real da mistura. Para $\alpha = 2$ (entropia quadrática) e para o caso onde temos p e g definidas como

$$p(\boldsymbol{x}) = P_1 N\left(\boldsymbol{x}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\right) + P_2 N\left(\boldsymbol{x}, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\right)$$

e,

$$g(\boldsymbol{x}) = N\left(\boldsymbol{x}, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right)$$

onde $N\left(\boldsymbol{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right)$ representa uma função densidade de probabilidade gaussiana

$$N\left(\boldsymbol{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) = k_i \, e^{-(\boldsymbol{x}-\boldsymbol{\mu_i})^t \, {\boldsymbol{\Sigma}_i}^{-1} \, (\boldsymbol{x}-\boldsymbol{\mu})}.$$

Sendo k_i a constante de normalização da gaussiana dada por

$$k_i = \frac{1}{\sqrt{(2\,\pi)^D \, |\Sigma_i|}},$$

temos,

$$\log\left(\frac{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left(P_1 N_1\left(\boldsymbol{x}, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\right) + P_2 N_2\left(\boldsymbol{x}, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\right)\right)^2 \mathrm{d}\boldsymbol{x}}{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} N_q\left(\boldsymbol{x}, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right)^2 \mathrm{d}\boldsymbol{x}}\right) =$$

$$\log\left(\frac{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left(P_1 k_1 e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^t \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1)} + P_2 k_2 e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^t \boldsymbol{\Sigma}_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2)}\right)^2 \mathrm{d}\boldsymbol{x}}{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left(k_q e^{-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}_q\right)^t \boldsymbol{\Sigma}_q^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}_q\right)}\right)^2 \mathrm{d}\boldsymbol{x}}\right).$$

Definindo:

$$\alpha_1 = \left(\boldsymbol{x} - \boldsymbol{\mu}_1\right)^t \boldsymbol{\Sigma}_1^{-1} \left(\boldsymbol{x} - \boldsymbol{\mu}_1\right)$$
$$\alpha_2 = \left(\boldsymbol{x} - \boldsymbol{\mu}_2\right)^t \boldsymbol{\Sigma}_2^{-1} \left(\boldsymbol{x} - \boldsymbol{\mu}_2\right)$$
$$\alpha_q = \left(\boldsymbol{x} - \boldsymbol{\mu}_q\right)^t \boldsymbol{\Sigma}_q^{-1} \left(\boldsymbol{x} - \boldsymbol{\mu}_q\right)$$

e substituindo temos:

$$\log\left(\frac{\int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty}\left(P_1 k_1 e^{-\frac{1}{2}\alpha_1}\right)^2 + 2P_1 k_1 P_2 k_2 e^{-\frac{1}{2}\alpha_1} e^{-\frac{1}{2}\alpha_2} + \left(P_2 k_2 e^{-\frac{1}{2}\alpha_2}\right)^2 \mathrm{d}\boldsymbol{x}}{\int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} k_q^2 e^{-\alpha_q}\mathrm{d}\boldsymbol{x}}\right) \cdot \log\left(\frac{\int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} P_1^2 k_1^2 e^{-\alpha_1} + 2P_1 k_1 P_2 k_2 e^{-\frac{1}{2}\alpha_1 - \frac{1}{2}\alpha_2} + P_2^2 k_2^2 e^{-\alpha_2}\mathrm{d}\boldsymbol{x}}{\int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} k_q^2 e^{-\alpha_q}\mathrm{d}\boldsymbol{x}}\right)$$

Separando as integrais, obtemos:

$$\log\left(\frac{\int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} P_1^2 k_1^2 e^{-\alpha_1}\mathrm{d}\boldsymbol{x} + \int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} 2P_1 k_1 P_2 k_2 e^{-\frac{1}{2}\alpha_1 - \frac{1}{2}\alpha_2}\mathrm{d}\boldsymbol{x} + \int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} P_2^2 k_2^2 e^{-\alpha_2}\mathrm{d}\boldsymbol{x}}{\int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} k_q^2 e^{-\alpha_q}\mathrm{d}\boldsymbol{x}}\right).$$

Fazendo:

$$C = 2P_1 k_1 P_2 k_2 e^{-\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)}$$

e resolvendo as integrais[3], obtemos:

$$\log\left(\frac{P_1^2 k_1^2 \sqrt{\pi^D \left|\boldsymbol{\Sigma}_1\right|} + C\sqrt{\frac{(2\pi)^D}{\left|\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right|}} + P_2^2 k_2^2 \sqrt{\pi^D \left|\boldsymbol{\Sigma}_2\right|}}{k_q^2 \sqrt{\pi^D \left|\boldsymbol{\Sigma}_q\right|}}\right).$$

Fazendo,

[3]A solução da integral que forma o termo central é detalhada na seção seguinte. As demais podem ser encontradas no apêndice.

$$w = e^{-\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t\,(\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)^{-1}\,(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)}$$

e simplificando temos:

$$\log\left(\frac{P_1^2\,\frac{\sqrt{\pi^D\,|\boldsymbol{\Sigma}_1|}}{(2\pi)^D\,|\boldsymbol{\Sigma}_1|} + 2\,w\,\frac{P_1}{\sqrt{(2\pi)^D\,|\boldsymbol{\Sigma}_1|}}\,\frac{P_2}{\sqrt{(2\pi)^D\,|\boldsymbol{\Sigma}_2|}}\sqrt{\frac{(2\pi)^D}{\left|\boldsymbol{\Sigma}_1^{-1}+\boldsymbol{\Sigma}_2^{-1}\right|}} + P_2^2\,\frac{\sqrt{\pi^D\,|\boldsymbol{\Sigma}_2|}}{(2\pi)^D\,|\boldsymbol{\Sigma}_2|}}{\frac{\sqrt{\pi^D\,|\boldsymbol{\Sigma}_q|}}{(2\pi)^D\,|\boldsymbol{\Sigma}_q|}}\right)$$

$$\log\left(\frac{P_1^2\,\frac{\sqrt{|\boldsymbol{\Sigma}_1|}}{|\boldsymbol{\Sigma}_1|} + 2\,w\,P_1\,P_2\sqrt{\frac{2^D}{|\boldsymbol{\Sigma}_1|\left|\boldsymbol{\Sigma}_1^{-1}+\boldsymbol{\Sigma}_2^{-1}\right||\boldsymbol{\Sigma}_2|}} + P_2^2\,\frac{\sqrt{|\boldsymbol{\Sigma}_1|}}{|\boldsymbol{\Sigma}_2|}}{\frac{\sqrt{|\boldsymbol{\Sigma}_q|}}{|\boldsymbol{\Sigma}_q|}}\right)$$

o que resulta em

$$\log\left(P_1^2\sqrt{\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_1|}} + \frac{2\,P_1\,P_2\sqrt{|\boldsymbol{\Sigma}_q|}\,e^{-\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t\,(\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)^{-1}\,(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)}}{\sqrt{\left|\frac{\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2}{2}\right|}} + P_2^2\sqrt{\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_2|}}\right). \tag{3.24}$$

Lembramos que esta expressão não se destina ao cálculo da negentropia diretamente, pois como veremos nos capítulo de resultados, a presença de valores negativos invalida a definição formal de negentropia. A negentropia foi apenas utilizada como inspiração para a medida da não-gaussianidade da mistura.

3.8.3.2 Demonstração 2

Para calcular a integral

$$I = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^t\,\boldsymbol{\Sigma}_1^{-1}\,(\boldsymbol{x}-\boldsymbol{\mu}_1)-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^t\,\boldsymbol{\Sigma}_2^{-1}\,(\boldsymbol{x}-\boldsymbol{\mu}_2)}\mathrm{d}\boldsymbol{x}$$

utilizaremos uma estratégia de fatoração polinomial para reescrever a integral na forma

$$I = K \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\boldsymbol{x}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{x}} \mathrm{d}\boldsymbol{x}$$

de maneira que possamos resolvê-la utilizando substituições de variáveis. Inicialmente expandimos o polinômio da exponencial (lembrando-se tratar de polinômios matriciais, portando importando a ordem das multiplicações) fazendo:

$$I = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\boldsymbol{x}^t \boldsymbol{\Sigma}_1^{-1} \boldsymbol{x} - \boldsymbol{x}^t \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_1^t \boldsymbol{\Sigma}_1^{-1} \boldsymbol{x} + \boldsymbol{\mu}_1^t \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 + \boldsymbol{x}^t \boldsymbol{\Sigma}_2^{-1} \boldsymbol{x} - \boldsymbol{x}^t \boldsymbol{\Sigma}_2^{-1} \boldsymbol{\mu}_2 - \boldsymbol{\mu}_2^t \boldsymbol{\Sigma}_2^{-1} \boldsymbol{x} + \boldsymbol{\mu}_2^t \boldsymbol{\Sigma}_2^{-1} \boldsymbol{\mu}_2\right)} \mathrm{d}\boldsymbol{x}.$$

Escrevendo a parte polinomial na forma $\boldsymbol{x}^t \boldsymbol{A} \boldsymbol{x} + \boldsymbol{x}^t \boldsymbol{B} + \boldsymbol{B}^t \boldsymbol{x} + \boldsymbol{C}$ obtemos:

$$I = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\boldsymbol{x}^t \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right) \boldsymbol{x} - \boldsymbol{x}^t \left(\boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1} \boldsymbol{\mu}_2\right) - \left(\boldsymbol{\mu}_1^t \boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\mu}_2^t \boldsymbol{\Sigma}_2^{-1}\right) \boldsymbol{x} + \boldsymbol{\mu}_1^t \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^t \boldsymbol{\Sigma}_2^{-1} \boldsymbol{\mu}_2\right)} \mathrm{d}\boldsymbol{x}.$$

O objetivo agora é fatorar o polinômio da exponencial. Fazendo

$$\boldsymbol{x}^t \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right) \boldsymbol{x} - \boldsymbol{x}^t \left(\boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1} \boldsymbol{\mu}_2\right) - \left(\boldsymbol{\mu}_1^t \boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\mu}_2^t \boldsymbol{\Sigma}_2^{-1}\right) \boldsymbol{x} + \boldsymbol{\mu}_1^t \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^t \boldsymbol{\Sigma}_2^{-1} \boldsymbol{\mu}_2 = (\boldsymbol{x} - \boldsymbol{m})^t \boldsymbol{\alpha}^{-1} (\boldsymbol{x} - \boldsymbol{m}) + \boldsymbol{C}$$

e expandindo-se o lado direito da igualdade obtemos:

$$\boldsymbol{x}^t \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right) \boldsymbol{x} - \boldsymbol{x}^t \left(\boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1} \boldsymbol{\mu}_2\right) - \left(\boldsymbol{\mu}_1^t \boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\mu}_2^t \boldsymbol{\Sigma}_2^{-1}\right) \boldsymbol{x} + \boldsymbol{\mu}_1^t \boldsymbol{\Sigma}_1^{-1} \boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^t \boldsymbol{\Sigma}_2^{-1} \boldsymbol{\mu}_2 = \boldsymbol{x}^t \boldsymbol{\alpha}^{-1} \boldsymbol{x} - \boldsymbol{x}^t \boldsymbol{\alpha}^{-1} \boldsymbol{m} - \boldsymbol{m}^t \boldsymbol{\alpha}^{-1} \boldsymbol{x} + \boldsymbol{m}^t \boldsymbol{\alpha}^{-1} \boldsymbol{m} + \boldsymbol{C}$$

Aplicando-se a igualdade de polinômios, podemos formar um sistema de equações onde cada coeficiente de cada grau é igual ao seu correspondente no outro lado da igualdade. Dessa maneira obtemos o sistema linear de equações matriciais

$$\begin{cases} \boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1} = \boldsymbol{\alpha}^{-1} \\ \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 = \boldsymbol{\alpha}^{-1}\boldsymbol{m} \\ \boldsymbol{\mu}_1^t\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^t\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 = \boldsymbol{m}^t\boldsymbol{\alpha}^{-1}\boldsymbol{m} + \boldsymbol{C} \end{cases}$$

Resolvendo o sistema obtemos

$$\begin{aligned} &\boldsymbol{\alpha} = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1} \\ &\boldsymbol{m} = \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1} \left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right) \\ &\boldsymbol{C} = \boldsymbol{\mu}_1^t\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2^t\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 - \left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right)^t \left(\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1}\right)^{-1} \left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2\right) \end{aligned}$$

onde $\boldsymbol{C}$ pode ser escrito ainda como [4]

$$\boldsymbol{C} = \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right)^t \left(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2\right)^{-1} \left(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right).$$

Desta maneira a integral pode ser escrita como:

$$I = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left((\boldsymbol{x}-\boldsymbol{m})^t\boldsymbol{\alpha}^{-1}(\boldsymbol{x}-\boldsymbol{m})+\boldsymbol{C}\right)} \mathrm{d}\boldsymbol{x} = e^{-\frac{1}{2}\boldsymbol{C}} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left((\boldsymbol{x}-\boldsymbol{m})^t\boldsymbol{\alpha}^{-1}(\boldsymbol{x}-\boldsymbol{m})\right)} \mathrm{d}\boldsymbol{x}.$$

Realizando uma troca de variáveis simples fazendo

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{x} - \boldsymbol{m} \\ \mathrm{d}\boldsymbol{y} &= \mathrm{d}\boldsymbol{x} \end{aligned}$$

obtemos

$$I = e^{-\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t(\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\boldsymbol{y}^t\left(\boldsymbol{\Sigma}_1^{-1}+\boldsymbol{\Sigma}_2^{-1}\right)\boldsymbol{y}\right)} \mathrm{d}\boldsymbol{y}.$$

Finalmente, a integral restante pode ser resolvida normalmente (ver apêndice).

[4] Esta passagem não é trivial, e pode ser encontrada em apêndice no final desta tese.

3.8.3.3 Invariância à transformações lineares

Nesta seção iremos mostrar a invariância à transformações lineares da métrica apresentada. Como fizemos com as métricas de Mahalanobis, Kullback-Leibler e Bhattacharyya, vamos considerar uma transformação linear como mostrada na equação 3.5. Como mostrado nas seções anteriores, a divergência entre duas gaussianas para uma VA Y pode ser calculada como

$$d_r^{(y)}(C_1, C_2) = \log\left(P_1^2 \sqrt{\frac{\left|\boldsymbol{\Sigma}_q^{(y)}\right|}{\left|\boldsymbol{\Sigma}_1^{(y)}\right|}} + \frac{2P_1P_2\sqrt{\left|\boldsymbol{\Sigma}_q^{(y)}\right|}e^{-\frac{1}{2}\left(\boldsymbol{\mu}_1^{(y)}-\boldsymbol{\mu}_2^{(y)}\right)^t\left(\boldsymbol{\Sigma}_1^{(y)}+\boldsymbol{\Sigma}_2^{(y)}\right)^{-1}\left(\boldsymbol{\mu}_1^{(y)}-\boldsymbol{\mu}_2^{(y)}\right)}}{\sqrt{\left|\frac{\boldsymbol{\Sigma}_1^{(y)}+\boldsymbol{\Sigma}_2^{(y)}}{2}\right|}} + P_2^2\sqrt{\frac{\left|\boldsymbol{\Sigma}_q^{(y)}\right|}{\left|\boldsymbol{\Sigma}_2^{(y)}\right|}}\right). \tag{3.25}$$

Substituindo os parâmetros da VA Y pelo seu equivalente em função de X obtermos

$$d_r^{(y)}(C_1, C_2) = \log\left(\begin{array}{l} P_1^2\sqrt{\frac{\left|\boldsymbol{A}\boldsymbol{\Sigma}_q^{(x)}\boldsymbol{A}^t\right|}{\left|\boldsymbol{A}\boldsymbol{\Sigma}_1^{(x)}\boldsymbol{A}^t\right|}} + \\ + \frac{2P_1P_2\sqrt{\left|\boldsymbol{A}\boldsymbol{\Sigma}_q^{(x)}\boldsymbol{A}^t\right|}e^{-\frac{1}{2}\left(\boldsymbol{A}\boldsymbol{\mu}_1^{(x)}+\boldsymbol{b}-\boldsymbol{A}\boldsymbol{\mu}_2^{(x)}-\boldsymbol{b}\right)^t\left(\boldsymbol{A}\boldsymbol{\Sigma}_1^{(x)}\boldsymbol{A}^t+\boldsymbol{A}\boldsymbol{\Sigma}_2^{(x)}\boldsymbol{A}^t\right)^{-1}\left(\boldsymbol{A}\boldsymbol{\mu}_1^{(x)}+\boldsymbol{b}-\boldsymbol{A}\boldsymbol{\mu}_2^{(x)}-\boldsymbol{b}\right)}}{\sqrt{\left|\frac{\boldsymbol{A}\boldsymbol{\Sigma}_1^{(x)}\boldsymbol{A}^t+\boldsymbol{A}\boldsymbol{\Sigma}_2^{(x)}\boldsymbol{A}^t}{2}\right|}} + \\ + P_2^2\sqrt{\frac{\left|\boldsymbol{A}\boldsymbol{\Sigma}_q^{(x)}\boldsymbol{A}^t\right|}{\left|\boldsymbol{A}\boldsymbol{\Sigma}_2^{(x)}\boldsymbol{A}^t\right|}} \end{array}\right).$$

Após algumas simplificações obtemos

$$d_r^{(y)}(C_1,C_2) = \log\left(\begin{array}{l} P_1^2\sqrt{\dfrac{|\boldsymbol{A}|^2\left|\boldsymbol{\Sigma}_q^{(x)}\right|}{|\boldsymbol{A}|^2\left|\boldsymbol{\Sigma}_1^{(x)}\right|}}+ \\ +\dfrac{2\,P_1\,P_2\,|\boldsymbol{A}|\sqrt{\left|\boldsymbol{\Sigma}_q^{(x)}\right|}e^{-\frac{1}{2}\left(\boldsymbol{\mu}_1^{(x)}-\boldsymbol{\mu}_2^{(x)}\right)^t\boldsymbol{A}^t\boldsymbol{A}^{-t}\left(\boldsymbol{\Sigma}_1^{(x)}+\boldsymbol{\Sigma}_2^{(x)}\right)^{-1}\boldsymbol{A}^{-1}\boldsymbol{A}\left(\boldsymbol{\mu}_1^{(x)}-\boldsymbol{\mu}_2^{(x)}\right)}}{|\boldsymbol{A}|\sqrt{\left|\frac{\boldsymbol{\Sigma}_1^{(x)}+\boldsymbol{\Sigma}_2^{(x)}}{2}\right|}}+ \\ +P_2^2\sqrt{\dfrac{|\boldsymbol{A}|^2\left|\boldsymbol{\Sigma}_q^{(x)}\right|}{|\boldsymbol{A}|^2\left|\boldsymbol{\Sigma}_2^{(x)}\right|}} \end{array}\right).$$

Como nas análises anteriores, sendo **A** uma transformação inversível, podemos escrever

$$d_r^{(y)}(C_1,C_2) = \log\left(P_1^2\sqrt{\frac{\left|\boldsymbol{\Sigma}_q^{(x)}\right|}{\left|\boldsymbol{\Sigma}_1^{(x)}\right|}}+\frac{2\,P_1\,P_2\sqrt{\left|\boldsymbol{\Sigma}_q^{(x)}\right|}e^{-\frac{1}{2}\left(\boldsymbol{\mu}_1^{(x)}-\boldsymbol{\mu}_2^{(x)}\right)^t\left(\boldsymbol{\Sigma}_1^{(x)}+\boldsymbol{\Sigma}_2^{(x)}\right)^{-1}\left(\boldsymbol{\mu}_1^{(x)}-\boldsymbol{\mu}_2^{(x)}\right)}}{\sqrt{\left|\frac{\boldsymbol{\Sigma}_1^{(x)}+\boldsymbol{\Sigma}_2^{(x)}}{2}\right|}}+P_2^2\sqrt{\frac{\left|\boldsymbol{\Sigma}_q^{(x)}\right|}{\left|\boldsymbol{\Sigma}_2^{(x)}\right|}}\right)$$

que de fato é a divergência da VA X.

3.8.3.4 Estudo de casos para divergência de Rényi

Nesta seção serão analisados alguns casos particulares de medidas de divergência entre duas gaussianas utilizando a métrica proposta. Estes casos serão analisados para mostrar a coerência da métrica diante de situações onde os resultados são esperados.

3.8.3.5 Médias e covariâncias iguais

A divergência utilizando a medida de negentropia de uma mistura formada por duas gaussianas iguais deve ser igual a zero. Esta imposição deixa coerente o fato de que um *cluster* deve ser "ligado" a ele mesmo, fazendo assim parte de um mesmo conjunto.

Seja $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$ e $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{\mu}$ a média e matriz de covariância para ambas as gaussianas. A gaussiana equivalente é dada por (ver seção 3.8.1.1)

$$\boldsymbol{\Sigma}_q = P_1\,\boldsymbol{\Sigma} + P_2\,\boldsymbol{\Sigma} + P_1\,P_2\,(\boldsymbol{\mu} - \boldsymbol{\mu})\;(\boldsymbol{\mu} - \boldsymbol{\mu})^t \tag{3.26}$$

Simplificando temos

$$\boldsymbol{\Sigma}_q = (P_1 + P_2)\,\boldsymbol{\Sigma}.$$

Como se trata de uma mistura de gaussianas, temos que $P_1 + P_2 = 1$, oque leva a

$$\boldsymbol{\Sigma}_q = \boldsymbol{\Sigma}.$$

Substituindo os valores definidos na expressão da divergência obtemos

$$\log\left(P_1^2\sqrt{\frac{|\boldsymbol{\Sigma}|}{|\boldsymbol{\Sigma}|}} + \frac{2\,P_1\,P_2\sqrt{|\boldsymbol{\Sigma}|}\,e^{-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{\mu})^t\,(\boldsymbol{\Sigma}+\boldsymbol{\Sigma})^{-1}\,(\boldsymbol{\mu}-\boldsymbol{\mu})}}{\sqrt{\left|\frac{\boldsymbol{\Sigma}+\boldsymbol{\Sigma}}{2}\right|}} + P_2^2\sqrt{\frac{|\boldsymbol{\Sigma}|}{|\boldsymbol{\Sigma}|}}\right), \tag{3.27}$$

que simplificando leva à

$$\log\left(P_1^2 + \frac{2\,P_1\,P_2\sqrt{|\boldsymbol{\Sigma}|}}{\sqrt{|\boldsymbol{\Sigma}|}} + P_2^2\right)$$

$$\log\left(P_1^2 + 2\,P_1\,P_2 + P_2^2\right)$$

Fatorando o termo dentro do logaritmo obtemos

$$\log\left((P_1 + P_2)^2\right),$$

que leva à log(1) que é igual a zero.

3.8.3.6 Médias iguais

O caso onde as médias são iguais, nos leva à uma medida de divergência que contabiliza apenas a diferença entre as matrizes de covariância. Sendo $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{\mu}$ a média das gaussianas, obtemos o seguinte para a matriz de covariância equivalentes

da mistura

$$\begin{aligned}\boldsymbol{\Sigma}_q &= P_1\boldsymbol{\Sigma}_1 + P_2\boldsymbol{\Sigma}_2 + P_1P_2\,(\boldsymbol{\mu}-\boldsymbol{\mu})\;(\boldsymbol{\mu}-\boldsymbol{\mu})^t \\ \boldsymbol{\Sigma}_q &= P_1\boldsymbol{\Sigma}_1 + P_2\boldsymbol{\Sigma}_2\end{aligned}.$$

Para a divergência, obtemos

$$\log\left(P_1^2\sqrt{\frac{|P_1\boldsymbol{\Sigma}_1+P_2\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}}+\frac{2P_1P_2\sqrt{|P_1\boldsymbol{\Sigma}_1+P_2\boldsymbol{\Sigma}_2|}e^{-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{\mu})^t(\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu})}}{\sqrt{\left|\frac{\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2}{2}\right|}}+P_2^2\sqrt{\frac{|P_1\boldsymbol{\Sigma}_1+P_2\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_2|}}\right), \tag{3.28}$$

que leva à

$$\log\left(P_1^2\sqrt{\frac{|P_1\boldsymbol{\Sigma}_1+P_2\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}}+\frac{2P_1P_2\sqrt{|P_1\boldsymbol{\Sigma}_1+P_2\boldsymbol{\Sigma}_2|}}{\sqrt{\left|\frac{\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2}{2}\right|}}+P_2^2\sqrt{\frac{|P_1\boldsymbol{\Sigma}_1+P_2\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_2|}}\right). \tag{3.29}$$

Observando a expressão 3.29, vemos que a soma ponderada das matrizes de covariância são comparadas com a matriz de covariância da primeira gaussiana, com a segunda e com a média das duas. Para o caso onde as probabilidades *a priori* são iguais temos

$$\log\left(\frac{1}{4}\sqrt{\frac{\left|\frac{\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2}{2}\right|}{|\boldsymbol{\Sigma}_1|}}+\frac{1}{2}+\frac{1}{4}\sqrt{\frac{\left|\frac{\boldsymbol{\Sigma}_1+\boldsymbol{\Sigma}_2}{2}\right|}{|\boldsymbol{\Sigma}_2|}}\right). \tag{3.30}$$

3.8.3.7 Covariâncias iguais

O caso onde as médias são iguais, nos leva à uma medida de divergência que contabiliza apenas a diferença entre as matrizes de covariância. Sendo $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$ a matriz de covariância das gaussianas, obtemos o seguinte para a matriz de covariância equivalente da mistura

$$\begin{aligned}\boldsymbol{\Sigma}_q &= P_1\boldsymbol{\Sigma} + P_2\boldsymbol{\Sigma} + P_1P_2\ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\ \ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t \\ \boldsymbol{\Sigma}_q &= (P_1+P_2)\,\boldsymbol{\Sigma} + P_1P_2\ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\ \ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t. \\ \boldsymbol{\Sigma}_q &= \boldsymbol{\Sigma} + P_1P_2\ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\ \ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t\end{aligned} \tag{3.31}$$

A divergência então é calculada como

$$\log\left(\begin{array}{l} P_1^2\sqrt{\dfrac{\left|\boldsymbol{\Sigma}+P_1P_2\ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\ \ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t\right|}{|\boldsymbol{\Sigma}|}}+ \\ +\dfrac{2\,P_1P_2\sqrt{\left|\boldsymbol{\Sigma}+P_1P_2\ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\ \ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t\right|}\,e^{-\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t(\boldsymbol{\Sigma}+\boldsymbol{\Sigma})^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)}}{\sqrt{\left|\frac{\boldsymbol{\Sigma}+\boldsymbol{\Sigma}}{2}\right|}}+ \\ +P_2^2\sqrt{\dfrac{\left|\boldsymbol{\Sigma}+P_1P_2\ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\ \ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t\right|}{|\boldsymbol{\Sigma}|}} \end{array}\right). \tag{3.32}$$

Simplificando e rearrumando alguns termos obtemos

$$\log\left(\sqrt{\frac{\left|\boldsymbol{\Sigma}+P_1P_2\ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)\ \ (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t\right|}{|\boldsymbol{\Sigma}|}}\left(P_1^2+2\,P_1P_2e^{-\frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^t(2\,\boldsymbol{\Sigma})^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)}+P_2^2\right)\right).$$

Como já comentado, o termo que contabiliza a diferença entre as médias tende a elevar o valor da divergência. Neste caso particular, este efeito é bem mais simples de ser visualizado. O termo com a diferença entre as médias que está na exponencial tende a diminuir e cair para zero, enquanto o termo anterior tende a subir com o aumento da diferença. Este comportamento é adequado, pois quanto mais distante estiverem as gaussianas, maior será o valor da divergência.

3.8.4 Relação entre as métricas

É interessante observar que os casos particulares para a divergência proposta neste trabalho têm comportamento parecido nas outras métricas analisadas. Em todos os casos, a divergência é igual a zero quando as gaussianas são iguais. Como já mencionado, este fato é importante pois assume que um *cluster* está "ligado" a

ele mesmo e faz parte da mesma classe. Outro requisito importante é o aumento no valor da divergência com a separação das gaussianas (distanciamento entre as médias). Este fato também é observado em todas as métricas e é fundamental para garantir que quanto mais distantes os *clusters*, maior a divergência e menor a probabilidade de pertencerem à mesma classe. Uma análise também possível é quanto a divergência entre gaussianas com a mesma média. Nesta análise a distância de Mahalanobis não apresenta diferença entre as gaussianas e resulta sempre em zero, independente da "forma" das gaussianas. As demais medidas, inclusive a proposta neste trabalho, resultam em um pequeno resíduo de divergência, mesmo quando as médias são iguais, contabilizando assim, a diferença entre as "formas" das gaussianas. Este fator não é importante pois os casos onde as médias são iguais não ocorrem no algoritmo de *clustering* proposto neste trabalho.

O ponto chave para a justificativa do uso das métricas apresentadas até agora é a sensibilidade ao alinhamento entre as gaussianas. Todas as métricas analisadas, inclusive a proposta, levam em consideração a "direção" de variação dos dados. Este fato é bastante importante para o algoritmo de ligação dos *clusters*. Isto evita a ligação de *clusters* que, embora possuindo centros próximos, estão paralelos entre si e não fazem parte da mesma classe. A diferença entre o comportamento das métricas analisadas em relação a esta situação é apenas quantitativo. No capítulo seguinte serão realizados testes para medir esta diferença.

Por fim, em todas as métricas analisadas, observamos invariância à transformações lineares, oque valida o uso das mesmas para o problema de *clustering* proposto. Em todos os casos, a distância limiar para decidir se um *cluster* está ligado a outro é estável e pode ser determinada utilizando algumas heurísticas que serão apresentadas no capítulo seguinte.

Na seção seguinte será mostrado o algoritmo final proposto neste trabalho e alguns detalhes práticos nas implementações dos métodos propostos até agora.

3.9 Implementação

O método proposto neste trabalho consiste basicamente de um algoritmo de ligação de *clusters* que, ligados, formarão classes modeladas por misturas de gaussianas. A seguir será detalhado o algoritmo geral para realizar o agrupamento e modelagem de um conjunto de dados. Em seguida serão feitos comentários sobre a implementação das métricas descritas nas seções anteriores.

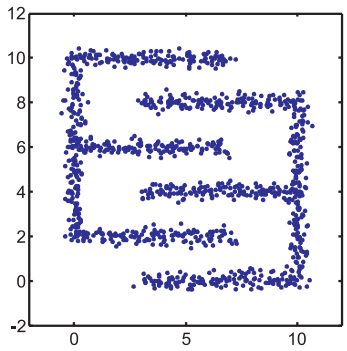
3.9.1 Algoritmo geral

Neste trabalho são propostos basicamente dois algoritmos que dividem o método em duas etapas. O primeiro é um algoritmo de *clustering*, que agrupa os pontos de um conjunto de dados em classes. O segundo consiste de um algoritmo para modelar as classes definidas na etapa de *clustering* utilizando mistura de gaussianas. Na etapa de *clustering*, um conjunto de dados é fornecido para o algoritmo que rotula cada um dos pontos com os rótulos das classes encontradas. O algoritmo inicia realizando uma quantização vetorial em todo o conjunto, separando-o em N_a *clusters* denominados de *clusters* auxiliares. O valor de N_a é o primeiro parâmetro a ser fornecido ao algoritmo. Após a quantização vetorial, é gerada uma matriz $\mathbf{U}$ de distâncias ou dissimilaridades que contabiliza a distância entre cada par dentre os N_a *clusters* auxiliares, formando assim uma matriz $N_a \times N_a$. Esta matriz é simétrica e tem diagonal igual a zero (divergência entre um *cluster* e ele mesmo). Desta maneira, o elemento $u_{i,j}$ ou $u_{j,i}$ representa a divergência entre o *cluster* i e o *cluster* j. Após calculada a matriz de distâncias $\mathbf{U}$, é gerada uma matriz de ligação $\mathbf{L}$ de mesma ordem de $\mathbf{U}$ contendo apenas 0 e 1. Os valores da matriz $\mathbf{L}$ são gerados, comparando-se cada valor de $\mathbf{U}$ com um limiar d_t escolhido. Caso o valor de $u_{i,j}$ seja menor ou igual ao valor de d_t, o elemento $l_{i,j}$ será 1, caso contrário será zero. Este limiar é então o segundo parâmetro a ser fornecido ao algoritmo. Após gerada a matriz de ligação, serão formadas as classes. A formação se dá associando classes iguais à *clusters* ligados, de maneira que se o *cluster* A está ligado ao *cluster* B e o *cluster* C ligado à B, então todos fazem parte da mesma classe. Após separadas as classes, cada ponto é rotulado com a classe atribuída ao seu *cluster*. A seguir é apresentado

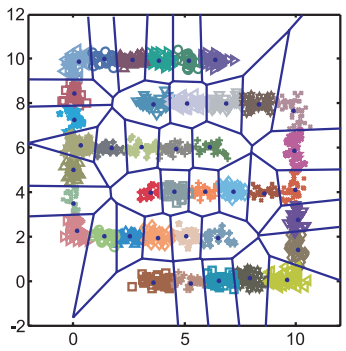
um resumo do algoritmo.

1. Realizar a quantização vetorial, separando os N_a *clusters*.
2. Gerar a matriz de distâncias **U**.
3. Gerar a matriz de ligação **L** baseado no limiar d_t
4. Agrupar os *clusters* ligados em classes
5. Rotular cada ponto de cada *cluster* com sua respectiva classe

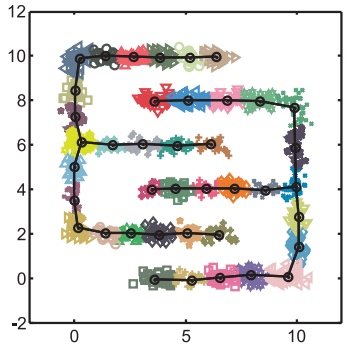
As Figuras de 3.13(a) a 3.13(d) ilustram a seqüência de eventos durante que ocorrem com um conjunto de dados durante a execução do algoritmo.



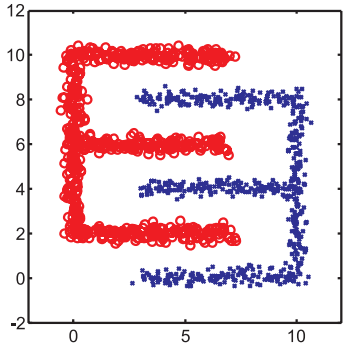
(a) Conjunto original



(b) Conjunto após a quantização vetorial



(c) Ligação baseada no limiar fornecido



(d) Resultado da rotulação final

Figura 3.13: Etapas do algoritmo de *clustering*

A Figura 3.13(a) mostra o conjunto original de dados. Como vemos, todos os pontos são rotulados igualmente, de maneira que não temos informação sobre a que classe cada ponto pertence. Em seguida (Figura 3.13(b)) o conjunto é quantizado em

pequenos *clusters* auxiliares. As linhas azuis são os limites de cada *cluster* formando as células de Voronoi. Na Figura 3.13(c) seguinte, temos os *clusters* ligados de acordo com o limiar fornecido. Observamos que *clusters* ligados entre si farão parte da mesma classe. Finalmente na Figura 3.13(d) os pontos são rotulados de acordo com a classe a qual seu *cluster* inicial está associado.

Uma forma alternativa para o algoritmo mostrado pode ser definida quando se tem o número de classes que se deseja agrupar. Neste caso os passos 3, 4 e 5 são repetidos para todos os limiares possíveis até que se encontre um limiar que produza o número de classes desejado. Os limiares possíveis são os próprios valores existentes na matriz **U**. Por exemplo, considere a matriz **U** a seguir

$$\boldsymbol{U} = \begin{bmatrix} 0 & 0.14 & 0.65 & 4.6 \\ 0.14 & 0 & 0.89 & 1.7 \\ 0.65 & 0.89 & 0 & 0.07 \\ 4.6 & 1.7 & 0.07 & 0 \end{bmatrix} \tag{3.33}$$

Para esta matriz, os únicos limiares que necessitam ser testados são 0.14, 0.65, 4.6, 0.89, 1.7 e 0.07. por exemplo, para um limiar de 0.14, a matriz de ligação obtida é dada por

$$\boldsymbol{L} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \tag{3.34}$$

e para qualquer limiar entre 0.14 e 0.65 (exclusive) a matriz **L** será a mesma.

O algoritmo modificado para esta situação é resumido a seguir

1. Realizar a quantização vetorial, separando os N_a *clusters*.
2. Gerar a matriz de distâncias **U**.
3. Para cada valor de **U**, fazer $d_t = u_{i,j}$
4. Gerar a matriz de ligação **L** baseado no limiar d_t

5. Agrupar os *clusters* ligados em classes

6. Se o número de classes for igual ao número desejado, vá para para o passo 7, se não, retorne para o passo 3

7. Rotular cada ponto de cada *cluster* com sua respectiva classe

Estes dois algoritmos oferecem alternativas para o problema quanto à informação que temos sobre o número de classes. É possível que, para algum valor de limiar d_t, o algoritmo forneça o número correto de classes, porém com o agrupamento errado. Como veremos na seção de resultados e testes, este caso raramente ocorre. Isto se deve à utilização das métricas que levam em conta a distribuição estatística e espacial dos conjuntos quantizados.

A etapa de quantização vetorial requer a utilização de um coeficiente de aprendizado, coeficiente de convergência e número máximo de iterações. Embora sejam parâmetros que devem ser fornecidos ao algoritmo, estes não constituem um problema na sua escolha. O algoritmo é bastante robusto quanto a estes parâmetros, podendo os mesmos assumirem uma larga faixa de valores sem que o algoritmo produza resultados diferentes. No próximo capítulo, serão analisados estes parâmetros e a sua escolha.

Ambas as formas do algoritmo proposto para realizar a etapa de *clustering* requerem a utilização de uma métrica para o cálculo da matriz $\mathbf{U}$. A métrica utilizada pode ser uma das métricas apresentadas e propostas neste trabalho. Nas seções seguintes serão feitos comentários sobre a utilização de cada uma das métricas analisadas. A diferença entre tais métricas será explorada no capítulo 4.

O outro parâmetro necessário para o funcionamento do algoritmo é o número de *clusters* auxiliares N_a. A escolha deste parâmetro implica diretamente no custo computacional envolvido na fase de *clustering*. Algumas heurísticas para sua escolha serão apresentadas também no capítulo de resultados e testes.

3.9.2 Modelamento dos dados

Após a segmentação do conjunto, o que se obtém são classes formadas por *clusters* que foram inicialmente quantizados e que são formados por pontos do conjunto de

dados inicial. A etapa de modelamento consiste em encontrar um modelo de mistura de gaussianas para cada classe agrupada na fase de *clustering*. Cada *cluster* auxiliar irá dar origem a um modelo gaussiano, de maneira que ao final, cada classe será formada por várias gaussianas, formando assim uma mistura.

Os parâmetros de cada gaussiana da mistura que modelará uma classe serão encontrados com base nos *clusters* auxiliares que formam a classe. A média e matriz de covariância de cada gaussiana da mistura serão estimadas utilizando-se apenas os pontos de cada um dos *clusters* auxiliares. A Figura 3.14 ilustra a formação da mistura para uma classe de um conjunto de dados.

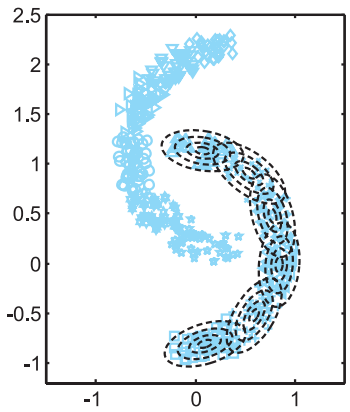


Figura 3.14: Modelo de misturas para uma classe

Na figura, as linhas tracejadas representam os contornos de cada gaussiana de cada *cluster* auxiliar. Os *clusters* são compostos pelos pontos representados por símbolos. Cada símbolo diferente representa pontos de *clusters* diferentes. Como podemos observar, a classe composta pelos *clusters* que formam o "semi-círculo" inferior, é modelada pelas gaussianas de maneira que a classe como um todo é modelada pela mistura das mesmas.

Para que a mistura possa ser totalmente modelada, é necessário a estimação das probabilidades *a priori* de cada gaussiana. Para calcular os valores estimados de cada probabilidade, basta que se divida o número de pontos presentes no *cluster* pelo total existente na classe, como mostra a equação 3.35

$$P_i = \frac{N_i}{\sum_{j \in w} N_j}, \tag{3.35}$$

onde P_i é a probabilidade *a priori* para a gaussiana i, N_i é o número de pontos no *cluster* i e o somatório deve ser feito para todos os *clusters* pertencentes na classe w que se deseja modelar.

Este procedimento é realizado para cada classe agrupada na etapa de *clustering*. Ao final, o modelo será composto por várias misturas de gaussianas (uma para cada classe). Cada mistura modela a probabilidade $p(\mathbf{x}|w_i)$ que é a probabilidade de uma ponto x pertencer a classe w_i. O modelo final é então gerado pela "mistura das misturas", como mostra a equação 3.36

$$p(\mathbf{x}) = \sum_{i=1}^{N_w} P_{w_i} p(\mathbf{x}|w_i), \tag{3.36}$$

onde $p(x)$ é o modelo de distribuição de probabilidade para a VA X que gerou os pontos do conjunto como um todo. N_w é o número de classes presentes e cada P_{w_i} são as probabilidades *a priori* de ocorrência de cada classe w_i. P_{w_i} pode ser calculado de maneira semelhante as probabilidades *a priori* das misturas de gaussianas de cada classe. Neste caso

$$P_{w_i} = \frac{N_{w_i}}{\sum_{j=1}^{N_w} N_{w_j}} \tag{3.37}$$

onde N_{w_j} é o número de pontos na classe w_j.

3.9.2.1 Utilização do modelo

O objetivo de se obter o modelo dos dados é possibilitar o cálculo das probabilidades de um ponto $\mathbf{x}$ dado que o mesmo pertence a uma classe w_i. Na seção 2.2 do capítulo 2, foi descrito um método para classificar estatisticamente um ponto como pertencente a uma classe ou a outra. A ferramenta descrita foi o classificador bayesiano. Para classificar um ponto como pertencente a uma classe w_i dentre N_w classes, calculamos as probabilidades deste ponto pertencer a cada classe e atribuímos ao mesmo, a classe com maior probabilidade.

Uma vez estimado o modelo dos dados, podemos a partir dai, classificar pontos fora do conjunto de dados utilizando a classificação bayesiana. Este é o objetivo de

se obter o modelo dos dados.

3.10 Aplicações

Métodos de *clustering* tem aplicações em qualquer área que necessite agrupar dados em classes específicas. Em particular, o método proposto neste trabalho foi testado em três aplicações bem distintas: Análise de dados de microscopia eletrônica, Segmentação e vetorização de imagens e Reconstrução 3D de superfícies. Os detalhes sobre a utilização do método para cada uma das aplicações encontram-se nos apêndices no final deste trabalho. A seguir será feito um breve resumo sobre a utilização do método nas aplicações descritas.

3.10.1 Análise de dados de microscopia eletrônica

Nesta aplicação, são captados dados de um microscópio de varredura eletrônica sobre amostras de minerais (em particular amostras de cimento). Estes dados correspondem às concentrações de átomos analisados no material. O conjunto é formado por pontos ou vetores cujas dimensões representam as concentrações dos átomos. Cada ponto consiste das concentrações em um determinado local no espaço na amostra em questão. Por exemplo na primeira dimensão podemos ter a concentração de cálcio, na segunda de oxigênio, etc. Este procedimento denomina-se na literatura de ZAP [26].

O problema consiste em classificar as regiões do material analisado em classes de acordo com as concentrações dos átomos neste material. A partir do conjunto de pontos amostrados de um certo material, aplicamos o método utilizando a abordagem da escolha do número de classes. Escolhe-se inicialmente um número N_w de classes e seleciona-se o limiar até que este número seja atingido, segmentando em seguida o conjunto todo. Uma vez segmentado o conjunto, obtemos o modelo estatístico do mesmo. Com o modelo estatístico em mãos, procede-se a classificação de todos os pontos varridos pelo microscópio, formando assim uma imagem onde cada ponto representa a classe selecionada pelo algoritmo para aquele local da amostra. A imagem resultante consiste de uma imagem com regiões que

correspondem aos diferentes tipos de minerais presentes na amostra do material.

A classificação química dos materiais ou minerais presentes nas amostras não é, obviamente, realizada pelo método de *clustering*. O método apenas separa o material em classes de 1 a N_w que devem ser posteriormente analisadas pelo especialista da área. Mais detalhes sobre o procedimento para esta aplicação encontram-se no apêndice A.

3.10.2 Segmentação de imagens

Esta aplicação consiste em segmentar uma imagem digital de acordo com a textura. Segmentar uma imagem significa separá-la em regiões não sobrepostas onde os píxeis nestas regiões possuam atributos semelhantes [27]. Pela definição de segmentação, podemos tratar o problema como sendo um problema de *clustering*. Para isto, devemos considerar os pixeis da imagem como sendo os pontos que formarão o conjunto a ser agrupado. Os atributos são obtidos baseados nas componentes R, G e B de cada pixel, que são as componentes vermelho, verde e azul que formam as cores na imagem. Cada pondo do conjunto tem portanto três dimensões, correspondendo cada uma a uma componente de cor (R, G e B).

O objetivo é agrupar os pontos amostrados da imagem em classes que irão corresponder às regiões de texturas diferentes na mesma. Nesta aplicação, tanto a abordagem da escolha do limiar quanto da escolha do número de classes pode ser escolhida. Uma vez agrupados os pontos e gerado o modelo dos dados, cada pixel da imagem é classificado e a imagem segmentada é gerada. Cada pixel na imagem segmentada possui o rótulo da classe que foi designada pelo classificador.

3.10.3 Vetorização de imagens

Uma imagem digital é uma representação rasterizada de uma cena real. Sua representação mais comum é através de matrizes bidimensionais, onde a posição de cada elemento da matriz representa uma posição espacial relativa e ao valor do elemento naquela posição é associada uma intensidade de cor. Vetorizar uma imagem significa representar os desenhos existentes na mesma de forma vetorial, ou

seja, ao invés do mapa de bits convencional (matriz de pontos), utilizar primitivas geométricas, tais como retas, polígonos, círculos, elipses, etc para representar as regiões de coloração semelhante nesta imagem [69].

O procedimento de vetorização com o método proposto aproveita o conceito de ligação dos *clusters* auxiliares desenvolvido neste trabalho e não utiliza a técnica de *clustering* em si. O dado de entrada para o algoritmo de vetorização é uma imagem previamente segmentada, ou seja, com as regiões de coloração semelhante rotuladas com identificadores específicos.

Para utilizar o método proposto neste trabalho, um conjunto de dados é gerado a partir da imagem segmentada. A imagem deve conter apenas duas regiões; uma região de "fundo" e uma região que representa o objeto a ser vetorizado. O conjunto é formado por vetores bi-dimensionais em que cada vetor corresponde às coordenadas (x, y) dos pixeis que fazem parte da região do objeto. Se a imagem contiver mais de dois tipos de regiões segmentadas, basta executar o algoritmo para cada tipo região de modo individual para cada uma.

Uma vez extraído o conjunto da imagem, é realizada a quantização vetorial do mesmo, separando-o em N_a pequenos *clusters*. Estes *clusters* corresponderão a pequenas regiões de pixeis pertencentes aos objetos segmentados na imagem. Em seguida, um limiar é escolhido e os *clusters* auxiliares são ligados de acordo com o algoritmo já apresentado. Neste ponto, não interessa a classificação ou o agrupamento das regiões. A informação desejada será obtida das ligações realizadas entre os *clusters*. Cada ligação corresponde a um vetor na imagem vetorizada. Este vetor é calculado observando-se os centros de cada par de *cluster* ligados.

3.10.4 Reconstrução 3D

A reconstrução de uma superfície é um procedimento que converte um conjunto de pontos no espaço ou seções transversais em uma representação de superfície, identificando-a e representando-a através de primitivas geométricas [43], tais como polígonos, triângulos, ou superfícies contínuas.

Neste tipo de aplicação, os dados são fornecidos, em geral, por *scanners 3D*. Estes equipamentos são capazes de realizar varreduras tridimensionais em um

objeto e capturar as posições espaciais de pontos de sua superfície. Um algoritmo de reconstrução utiliza estas posições amostradas como espaço de entradas para obtenção da superfície representativa do objeto escaneado.

O método proposto neste trabalho é capaz de obter, para um dado conjunto de posições espaciais, um conjunto de arestas interligando os vértices de polígonos candidatos à realizar a reconstrução completa da superfície.

Inicialmente o conjunto é quantizado em N_a pequenos *clusters*, que serão compostos por pontos da superfície do objeto (já adquiridos com o *scanner* ou outro equipamento). Em seguida, um limiar é definido e os *clusters* são ligados com base neste limiar. Como na vetorização de imagens, as "ligações" darão origens a vetores.

Os vetores gerados a partir do conjunto de dados são as arestas dos polígonos candidatos. Operações posteriores de filtragem e seleção de elementos permitirão obter representações poligonais adequadas para a superfície reconstruída.

3.11 Sumário

Neste capítulo foi descrito e desenvolvido o método proposto neste trabalho. Foram descritas as métricas utilizadas bem como o algoritmo final de *clustering* e classificação. Sobre as métricas utilizadas, procurou-se explorar o conceito de invariância a transformações lineares, com o objetivo de validar seu uso no algoritmo de *clustering* proposto. Foi proposta uma nova métrica de divergência baseada no conceito de negentropia utilizando a entropia de Rényi. Uma forma analítica para o cálculo da divergência proposta foi desenvolvido, sendo possível com a utilização da entropia de Rényi. Foram analisados também os aspectos sobre o custo computacional envolvido e, ao final, um resumo sobre algumas aplicações foi apresentado.

Capítulo 4

Testes e Resultados

4.1 Introdução

Neste capítulo serão apresentados alguns testes e resultados obtidos com o método desenvolvido neste trabalho. Os testes realizados destinam-se a ilustrar o comportamento de cada ítem ou etapa apresentada no capítulo 3. Em seguida serão apresentados alguns resultados juntamente com uma análise qualitativa sobre o comportamento do método em cada situação apresentada. Ao final, serão feitas análises comparativas com outros métodos de *clustering*.

4.2 Testes

Para ilustrar o funcionamento de cada etapa que compõe o método desenvolvido foram realizados alguns testes. A seguir serão apresentados os testes realizados em cada etapa desenvolvida. Inicialmente foram realizados testes para verificar o comportamento das medidas de divergência para diferentes distribuições. Em seguida os testes objetivaram estudar o algoritmo de quantização vetorial, observando alguns parâmetros dos *clusters* auxiliares gerados. Por fim, será demonstrado o comportamento do algoritmo completo, visando obter algumas heurísticas para estimação dos parâmetros necessários para o funcionamento do mesmo.

4.2.1 Medidas de Divergência

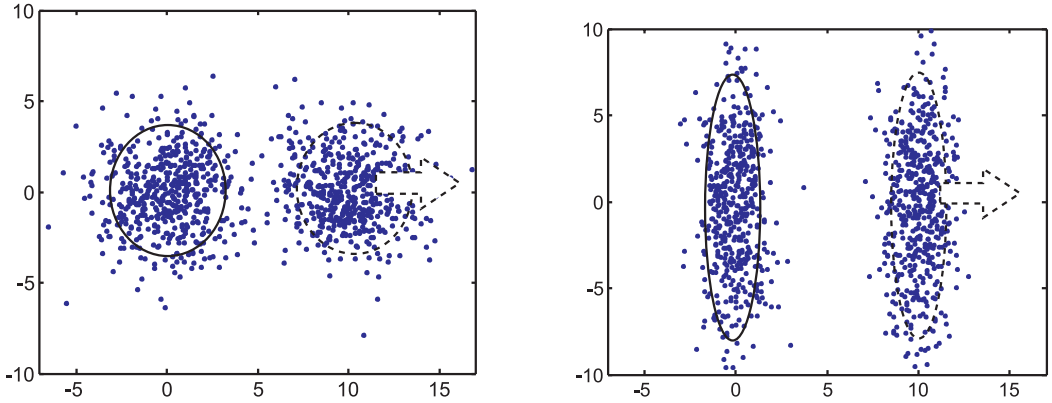
Nesta seção será observado o comportamento dos valores das divergências apresentadas neste trabalho para situações úteis em problemas de *clustering*. Em particular estamos interessados em saber como varia o valor das medidas de divergência entre *clusters* cujas médias estão se distanciando ou cujas matrizes de covariância diferem entre si. Isto é importante pois, no algoritmo desenvolvido, é preciso tomar a decisão de ligar ou não um *cluster* a outro.

4.2.1.1 Variações nas médias

Para verificar o comportamento entre *clusters* distantes entre si, podemos montar um gráfico que relacione a medida de divergência com a distância Euclidiana entre as médias. O comportamento esperado é de aumento da divergência com o aumento da distância entre as médias. Para realizar este teste foram geradas quatro situações distintas. A primeira situação consiste em variar a distância entre dois *clusters* isotrópicos (Figura 4.1(a)). A segunda situação consiste em variar a distância entre dois *clusters* não-isotrópicos e paralelos, sendo esta variação contrária à variância dos *clusters* (Figura 4.1(b)). Na terceira situação, a variação da média ocorre também entre *clusters* não-isotrópicos, porém na direção da variância, com os *clusters* alinhados (Figura 4.1(c)). Finalmente o último caso corresponde a uma variação da distância entre as médias em uma direção qualquer entre dois *clusters* não-isotrópicos com direções quaisquer (Figura 4.1(d)). As figuras correspondem a situações de dados bi-dimensionais, porém o resultado é totalmente válido para dimensões mais altas. Neste caso, a diferença está no fato de haver um número maior maior de graus de liberdade para a separação das médias.

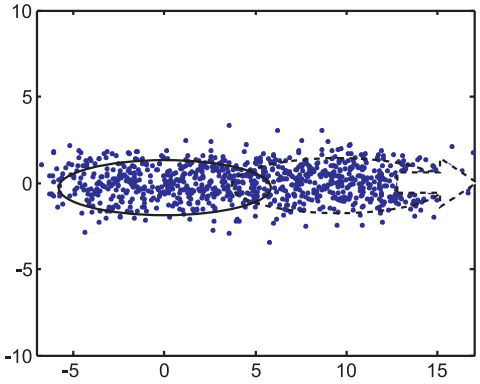
Nestas figuras, os conjuntos circulados com linhas pontilhadas são distanciados do conjunto a esquerda na direção das setas. Para cada passo, calcula-se a divergência. As Figuras 4.2(a) à 4.2(d) mostram os gráficos gerado pelas medidas de divergência em função da distância entre as médias.

Como podemos observar pelos gráficos mostrados e pelas expressões das divergências, para todas elas a tendência é aumentar quando a separação aumenta

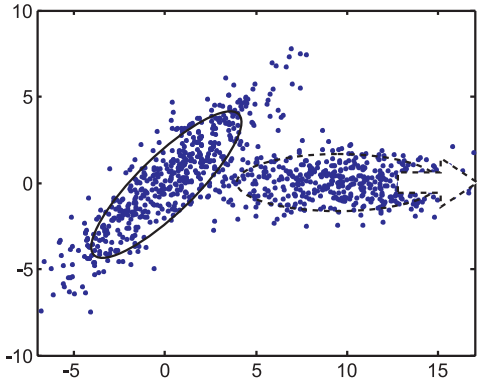


(a) *Clusters* isotrópicos

(b) *Clusters* paralelos



(c) *Clusters* alinhados

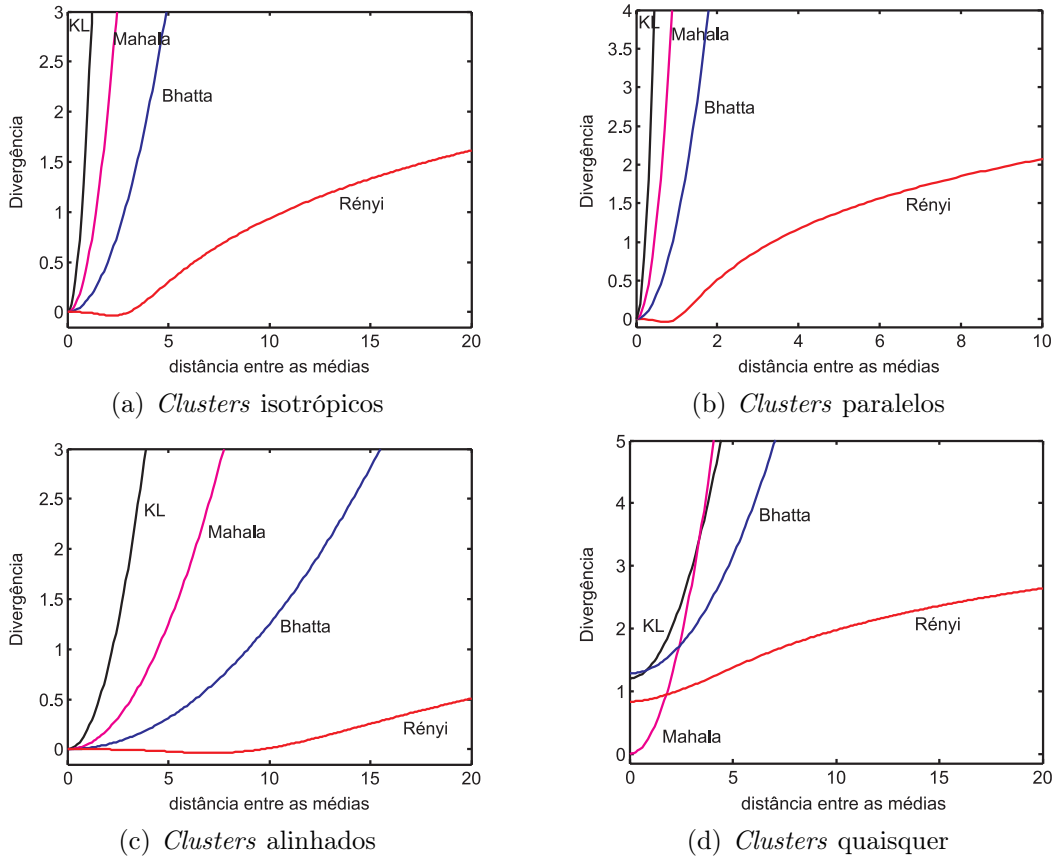


(d) *Clusters* quaisquer

Figura 4.1: Conjuntos de dados utilizados para realizar o teste de variação da distância entre as médias

(distância entre as médias). Em todos os casos há uma diferença para a divergência utilizando a entropia de Rényi (desenvolvida neste trabalho). Neste caso, existe uma tendência inicial a permanecer baixa a divergência para pequenas separações entre os *clusters*. Isto é bem razoável já que uma pequena separação entre dois *clusters* isotrópicos os tornam parecidos com um único *cluster* um pouco "não-isotrópico" até que a separação seja tal que haja realmente diferença ou um pequeno "vale" entre ambos os *clusters*. O caso interessante ocorre por exemplo quando os *clusters* estão alinhados. Neste caso, a divergência de Rényi demora para começar a aumentar, o que de fato deve acontecer, pois já que os mesmos são alinhados, mesmo separados ambos parecem um único *cluster*[1]. Para visualizar este efeito, a Figura 4.3 reúne, em um único gráfico, as divergências em função da distância das médias para os casos onde temos os *clusters* paralelos, isotrópicos e alinhados (linhas ponto-traçejada,

[1]Este fato pode ser observado na Figura 4.1(c). Nesta figura, embora não pareça, existem dois agrupamentos, porém como os mesmos estão alinhados, só percebe-se um



(a) *Clusters* isotrópicos

(b) *Clusters* paralelos

(c) *Clusters* alinhados

(d) *Clusters* quaisquer

Figura 4.2: Gráficos gerados variando-se as distancias entre as médias dos conjuntos

contínua e tracejada respectivamente).

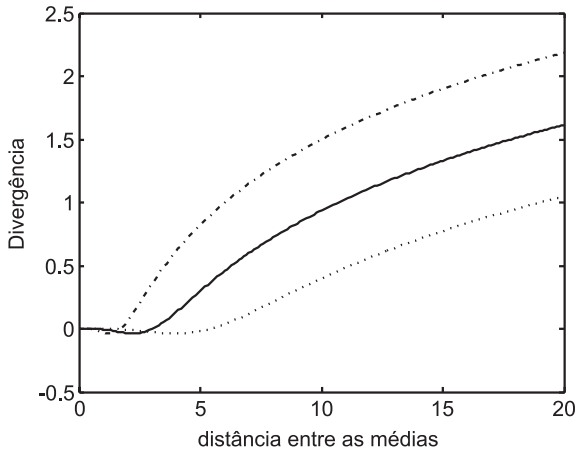


Figura 4.3: Crescimento da divergência em função da separação entre os *clusters*

Como podemos observar, os *clusters* paralelos, logo se tornam *clusters* separados e a divergência cresce rapidamente com a distância entre as médias. Para os demais esta só vem a crescer quando a separação é um pouco maior, sendo mais crítico o

caso para os *clusters* alinhados.

Outro fato interessante é quanto ao crescimento mais lento da divergência de Rényi. Este crescimento é lento devido à presença do logaritmo na expressão da divergência. Como veremos nos resultados, este fato não atrapalha o algoritmo já que o crescimento em si é o que importa para decidir se dois *clusters* estão ligados.

Podemos ainda observar um detalhe importante sobre o comportamento da divergência no caso onde os *clusters* são alinhados de maneira aleatória como na Figura 4.1(d). Os gráficos que descrevem o comportamento nesta situação (Figura 4.2(d)) mostram que a divergência não é zero quando a distância entre as médias é zero (exceto para o caso da distância de Mahalanobis). Isto se deve a diferença natural entre os alinhamentos das matrizes de covariância.

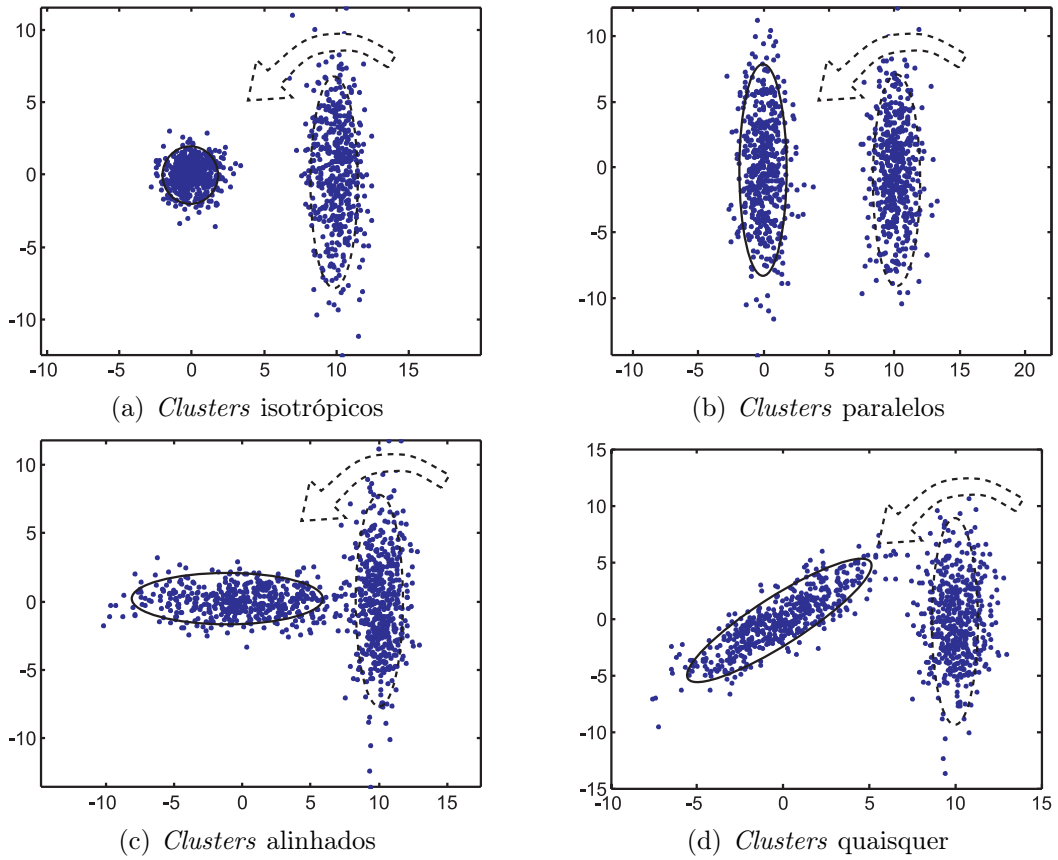
Por fim, observa-se a ocorrência de valores negativos na divergência de Rényi. Para efeito de comparação, estes valores estão de acordo com a lógica utilizada de maneira que uma divergência negativa é menor que uma divergência, por exemplo, igual a zero, significando que os *clusters* auxiliares devem ser ligados.

4.2.1.2 Variações nas covariâncias

De forma análoga ao observado na variação das distâncias entre as médias, podemos observar as mudanças na divergência com a mudança nas matrizes de covariância entre os *clusters*. Para verificar este comportamento, foram feitas medidas quanto ao "alinhamento" dos dados, o que leva a rotação dos mesmos em torno de sua média. Nas Figuras de 4.4(a) à 4.4(d) são mostradas as situações testadas. As linhas pontilhadas novamente indicam o *cluster* que se modifica. Desta vez o *cluster* é rotacionado de um ângulo entre 0 e $2\,\pi$.

Em cada situação, foi observado o comportamento da divergência para as métricas analisadas neste trabalho. As Figuras de 4.5(a) à 4.5(d) mostram os gráficos gerados. Os gráficos estão dispostos em um eixo semi-logaritmo para facilitar a visualização.

Neste tipo de situação, obtemos comportamentos um pouco mais parecidos para todas as divergências. Podemos notar que as divergências KL e Rényi são mais parecidas entre si e as divergências de Bhattacharyya e Mahalanobis também são



(a) *Clusters* isotrópicos

(b) *Clusters* paralelos

(c) *Clusters* alinhados

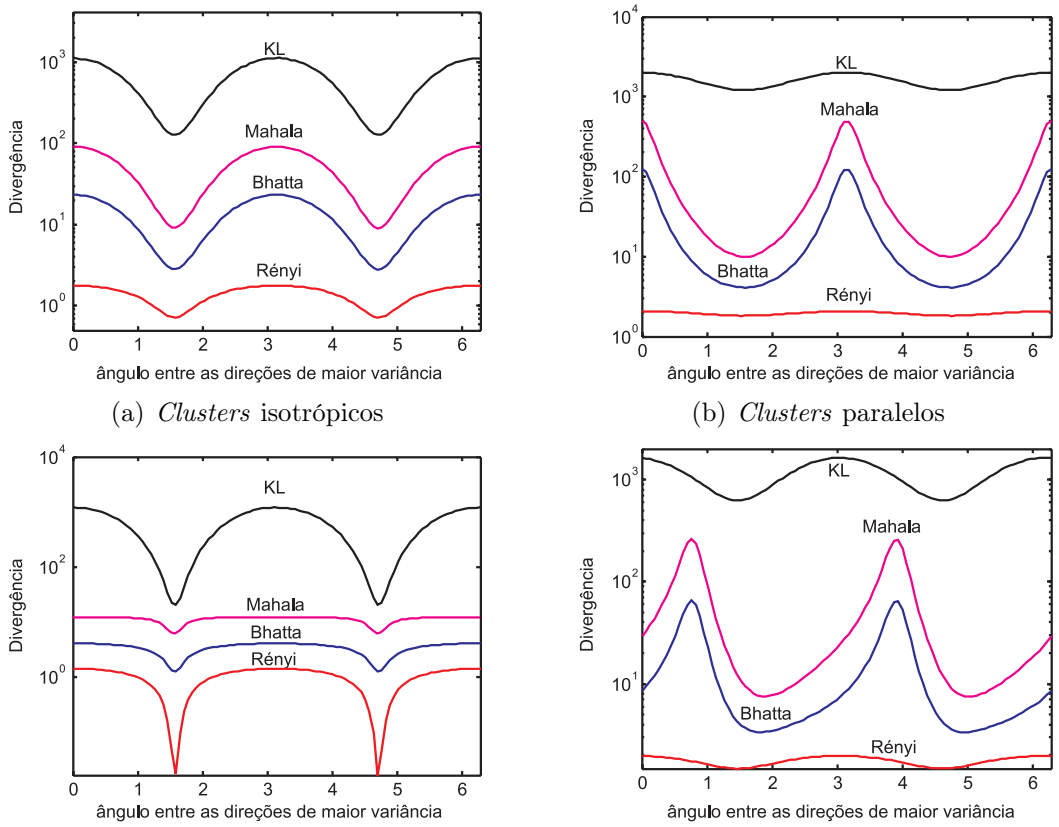
(d) *Clusters* quaisquer

Figura 4.4: Conjuntos de dados utilizados para realizar o teste de variação do "alinhamento" entre as médias

parecidas entre si. No caso das divergências de Mahalanobis e Bhattacharyya, observamos que as mesmas ponderam a distância entre as médias com o inverso da soma das matrizes de covariância. No caso da divergência KL a ponderação se dá com a soma dos inversos. Na divergência de Rényi esta ponderação é distribuída entre os termos dentro do logaritmo, e por isso os gráficos para esta divergência tem forma ligeiramente diferente dos gráficos da divergência KL.

Novamente podemos observar um comportamento diferenciado quando se tem um *cluster* orientado aleatoriamente (Figura 4.5(d)). De fato para este tipo de situação, a diferença se concentra nas divergências de Mahalanobis e Bhattacharyya.

Em geral, para o teste de "alinhamento" realizado, observamos que a divergência oscila entre um valor máximo e mínimo. Este comportamento é esperado e de fato corresponde a realidade, de maneira que *clusters* que estejam "alinhados" terão divergência baixa (vale nos gráficos) e *clusters* que estejam paralelos possuem



(a) *Clusters* isotrópicos

(b) *Clusters* paralelos

(c) *Clusters* alinhados

(d) *Clusters* quaisquer

Figura 4.5: Gráficos gerados variando-se o alinhamento entre os conjuntos

divergência alta (picos nos gráficos).

4.2.2 Quantização vetorial

Os testes para a etapa de quantização vetorial foram realizados para observar o que ocorre com esta etapa diante da variação do número de *clusters* auxiliares (N_a) iniciais. O ponto principal observado neste teste é o fato de que a escolha de um número elevando para N_a separa o conjunto inicial em *clusters* muito pequenos e com poucos elementos, gerando matrizes de covariância singulares e portanto não inversíveis.

Embora este problema não seja crítico, pode-se utilizar uma heurística para resolver o mesmo. Esta heurística consiste em iniciar com um valor alto de N_a e reduzir o mesmo até que as matrizes de covariância geradas sejam inversíveis. Na prática uma escolha para N_a como sendo uma porcentagem do número total de

pontos evita a ocorrência de matrizes singulares. Neste trabalho todos os exemplos utilizam entre 1 e 5 porcento do número total de pontos do conjunto.

O algoritmo utilizado para a quantização vetorial é o algoritmo de treinamento de uma rede neural competitiva já apresentado. Este algoritmo requer basicamente três parâmetros: o coeficiente de aprendizado α, constante de estabilização τ e o número máximo de épocas M_e. O algoritmo apresenta-se bastante robusto quanto a escolha destes parâmetros. Para todos os resultados e testes realizados, utilizou-se sempre os valores

$$\begin{aligned} \alpha &= 0.8 \\ \tau &= 0.99541 \\ M_e &= 200 \end{aligned} \tag{4.1}$$

Onde uma época é considerada quando todos os exemplos são apresentados a rede neural, de maneira que o número de iterações consistirá do produto do número de épocas pelo número de pontos do conjunto.

4.2.3 Estabilidade do número de classes

O algoritmo completo do método proposto foi submetido a testes para observar o comportamento do número de classes encontradas pelo algoritmo em função do limiar escolhido. Lembrando que, de acordo com a abordagem mostrada na seção 3.9.1 do capítulo 3, dado um limiar, as divergências entre cada par de *clusters* são medidas e, baseado nesse limiar, estes *clusters* são ligados ou não. *Clusters* ligados fazem parte de uma mesma classe de maneira que ao final, dependendo do limiar, o número de classes é estabelecido automaticamente pelo algoritmo.

Estes testes foram realizados em todos os conjuntos utilizados como exemplos neste trabalho. A Figura 4.6 mostra um exemplo de um destes testes.

Neste tipo de teste, para cada limiar testado, o número de classes é computado. O limiar é variado de zero até o primeiro valor que corresponda a apenas uma classe. Em todos os testes realizados o gráfico apresenta uma área de "estabilidade" onde o número de classes não varia com o aumento do limiar. Na maioria dos casos

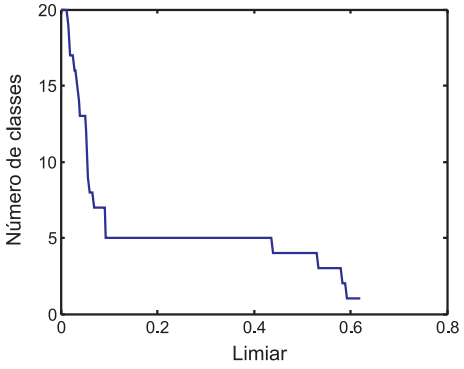


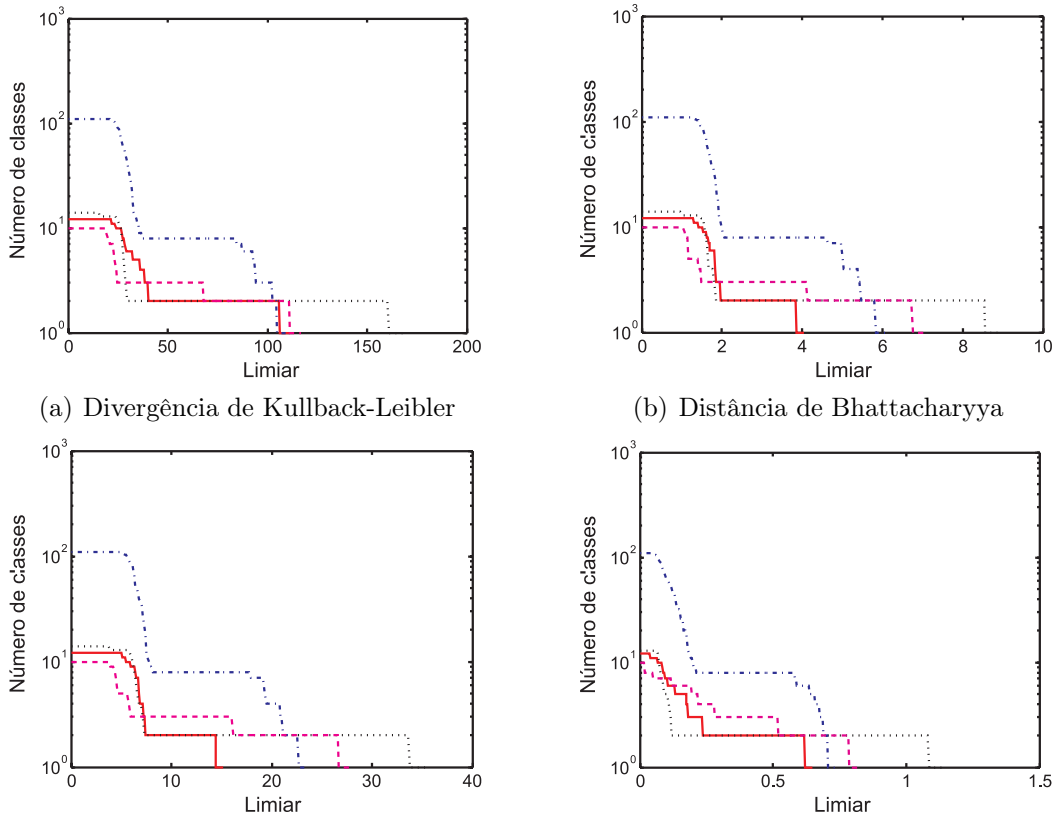
Figura 4.6: Número de classes encontradas pelo algoritmo em função do limiar utilizando a divergência de Rényi

testados esta estabilidade corresponde exatamente ao número de classes presentes no conjunto. Em alguns casos, existem mais de uma região de estabilidade, porém a região com maior estabilidade é a que provavelmente corresponde ao número correto de classes. Caso o conjunto não apresente região de estabilidade, leva-se a crer que haja apenas uma classe e o conjunto não deve ser separado.

A Figura 4.7 mostra alguns testes realizados para conjuntos diferentes para cada uma das medidas de divergência analisadas. Os gráficos mostram a divergência em um eixo logaritmo apenas por questões de visualização.

Nestas figuras, cada conjunto de dados é representado por um gráfico de uma cor e tipo de linha diferente. Cada gráfico corresponde a um teste realizado com uma medida de divergência diferente, conforme indicado no rótulo de cada uma das figuras. Uma observação muito importante que deve ser feita diz respeito a semelhança entre os gráficos que utilizam as divergências de Kullback-Leibler, Mahalanobis e Bhattacharyya. Para um mesmo conjunto de dados, o número de classes praticamente segue o mesmo padrão, mudando apenas a escala do limiar. Isto indica que para o algoritmo proposto, qualquer uma destas medidas pode ser utilizada sem que se tenha mudanças significativas no resultado final do agrupamento. Por outro lado, a divergência de Rényi mostra um comportamento diferente porém apresentando o mesmo comportamento geral.

Outro ponto importante a ser observado é a robustez das medidas quanto a escolha do limiar. Observando por exemplo a Figura 4.7(c) observamos que um limiar entre 9 e 14 contem a região de maior estabilidade para todos os conjuntos.



(a) Divergência de Kullback-Leibler

(b) Distância de Bhattacharyya

(c) Distância de Mahalanobis

(d) Divergência de Rényi

Figura 4.7: Gráficos do número de classes em função dos limiares

O mesmo ocorre com as demais divergências (mudando-se a faixa do limiar).

A estabilidade observada no número de classes pode ser utilizada como um passo adicional no algoritmo de *clustering* proposto neste trabalho. A escolha automática deste limiar baseado na estabilidade do número de classes pode ser trivialmente implementada de maneira que o algoritmo dispensaria a escolha do parâmetro d_t (limiar). Desta maneira o algoritmo proposto exigiria apenas o número de *clusters* auxiliares iniciais como parâmetro, além dos parâmetros de treinamento da etapa de quantização que, como já mencionado, não são críticos.

Esta etapa de escolha automática do limiar visa fazer com que o algoritmo selecione automaticamente o número de classes correto existente no conjunto. Esta abordagem funciona na maioria dos casos, porém, falha em alguns casos, gerando um número de classes errado. Em geral, para os casos onde as classes são bem separadas espacialmente o algoritmo funciona fornecendo o número correto de classes, mesmo que o conjunto esteja entrelaçado ou misturado. Os casos em que esta abordagem

falha são casos especiais e, como veremos na seção de resultados, a escolha manual do limiar ou número de classes compensa.

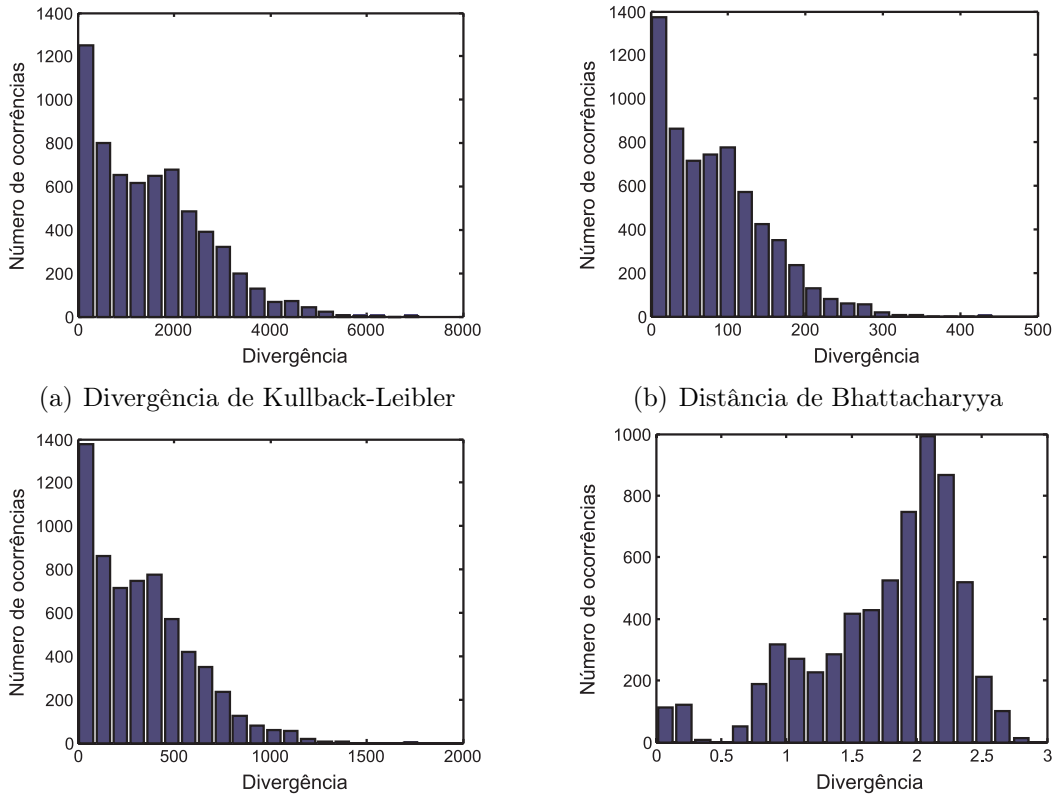
4.2.3.1 Custo computacional

Antes de passar para a próxima análise, é importante comentar sobre o custo computacional envolvido na construção dos gráficos do número de classes em função do limiar. A princípio esta etapa pode parecer custosa computacionalmente, porem este não é o caso.

Como já mencionado, para a construção dos gráficos do número de classes, devemos variar o limiar e obter a quantidade de classes fornecidas pelo algoritmo para cada um dos limiares. Para todos os testes realizados neste trabalho foram utilizados 200 limiares entre 0 e o máximo (limiar que retorne apenas uma classe). Para contar o número de classes, devemos comparar o limiar com as divergências entre cada par de centros. Esta tarefa praticamente não possui custo computacional nesta fase do algoritmo pois as divergências foram calculadas e já estão disponíveis na matriz de distâncias **U**. O que se faz é gerar uma nova matriz de ligação **L** para cada um dos 200 limiares e contar o número de classes apresentadas por estas matrizes de ligação. A contagem do número de classes é feita, como já mencionado, agrupando pares de centros (nesta etapa, representados por linhas e colunas na matriz **U** e **L**) que estejam ligados entre si. Esta etapa de agrupamento possui custo computacional muito baixo, fazendo com que seja possível a análise de uma quantidade bastante grande de limiares sem que isto afete o custo computacional do algoritmo final.

4.2.4 Histograma de divergências

Outra análise interessante a ser feita é computar o histograma das divergências ocorridas para um dado conjunto de teste. Este histograma conta a freqüência de ocorrência dos valores de divergências medidas entre cada par de centros. Basicamente esta contagem é feita observando-se os valores presentes na matriz **U**. A Figura 4.8 ilustra os histogramas de um conjunto agrupado utilizando cada uma das divergências.



(a) Divergência de Kullback-Leibler

(b) Distância de Bhattacharyya

(c) Distância de Mahalanobis

(d) Divergência de Rényi

Figura 4.8: Histogramas das divergências para um conjunto de teste

Os histogramas apresentados nas Figuras de 4.8(a)a 4.8(d) são histogramas de um conjunto de testes específico. Embora existam mudanças de um histograma para outro, a forma geral dos mesmos é igual para todos os conjuntos testados.

É interessante notar que novamente as divergências de Kullback-Leibler, Bhattacharyya e Mahalanobis apresentam comportamentos muito semelhantes. A divergência de Rényi porém, apresenta um comportamento diferente. A forma do histograma utilizando a divergência de Rényi apresenta uma região de pouca ocorrência de valores, em geral, centrados em 0.5. Este valor de divergência está situado em um vale que divide o histograma e coincide exatamente com os valores de divergência que estabilizam o número correto[2] de classes da maioria dos conjuntos. A Figura 4.9 mostra alguns histogramas da divergência de Rényi juntamente com o gráfico do número de classes para limiares iguais a cada valor de divergência

[2]Há uma estabilidade em uma classe para todos os conjuntos, porém, esta não é considerada pois qualquer valor de dt maior que o limiar que gera uma classe continuará gerando apenas uma classe

mostrado no histograma[3].

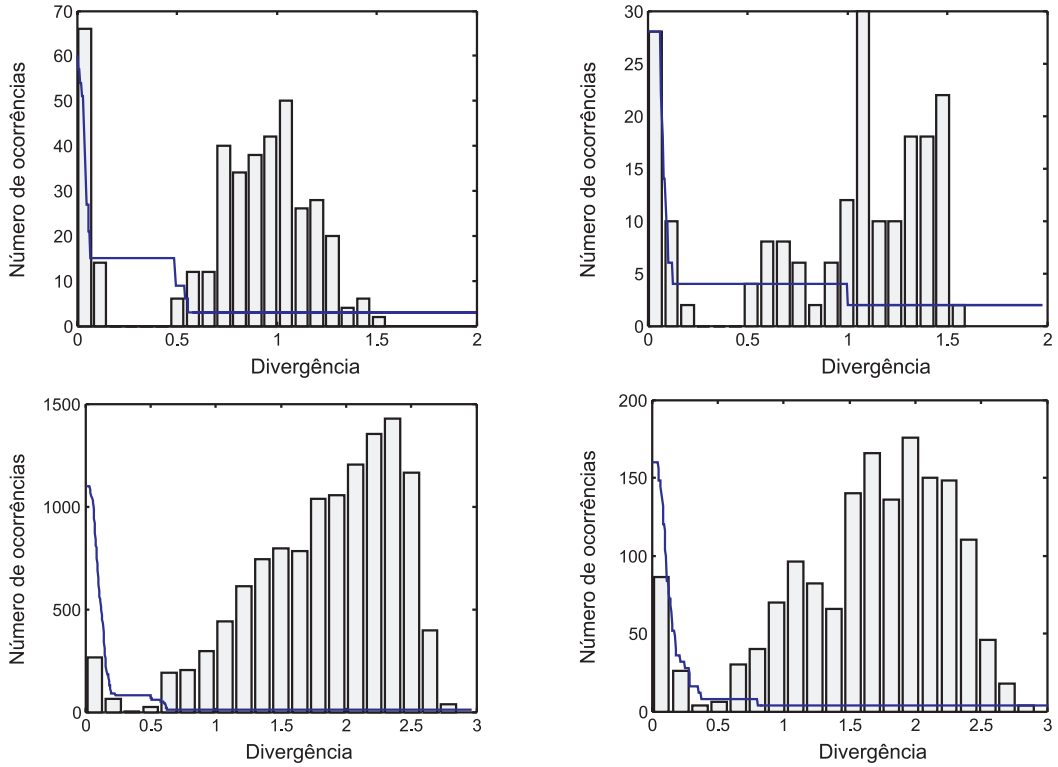


Figura 4.9: Comparação do histogramas de diversos conjuntos de dados com o gráfico do número de centros

Na figura, cada gráfico corresponde a um conjunto diferente de dados. Em todos, a região de estabilidade do número de classes ocorre nos limiares onde a divergência se localiza perto do vale que é comum a todos os histogramas que utilizam a divergência de Rényi. Os histogramas que não são totalmente suaves foram gerados com poucos valores (poucos *clusters* auxiliares), porém o mesmo comportamento se verifica.

O histograma das divergências de Rényi é importante pois serve como base para a escolha correta do limiar e funciona como uma forma de visualizar como os limiares estão distribuídos. Embora os valores ótimos para o limiar não sejam claramente escolhidos de acordo com o histograma, este fornece uma boa base para esta escolha e consiste de uma outra alternativa ao gráfico da estabilidade do número de classes.

[3]O gráfico do número de classes teve sua escala alterada para questões de comparação. O Eixo Y do gráfico corresponde apenas aos valores de freqüência de ocorrência do histograma

4.3 Resultados

Nesta seção serão apresentados resultados de *clustering* e modelamento utilizando o algoritmo proposto com base em conjuntos de dados descritos a seguir. Em cada resultado apresentado, será feita uma análise discutindo a os aspectos de dificuldade, qualidade da solução etc. Serão analisados casos particulares de *clustering* que representam situações um pouco mais delicadas para os algoritmos tradicionais e para o algoritmo proposto. Por fim será comentado sobre a comparação de algoritmos de *clustering* e de modelamento com o algoritmo proposto.

4.3.1 Conjuntos analisados

Antes de apresentar resultados, nesta seção serão descritos os conjuntos utilizados. Cada conjunto possui uma particularidade que será explorada e servirá para testar a qualidade e eficiência do algoritmo proposto. Em sua maioria os conjuntos foram gerados artificialmente para estabelecer a dificuldade desejada para testar o algoritmo. Os conjuntos testados possuem dimensões variadas, porém a aplicação do algoritmo não se limita as dimensões mostradas aqui. Conjuntos de qualquer dimensão podem ser utilizados. A lista a seguir descreve cada conjunto utilizado para posteriores citações no texto.

(arcos) Este é um conjunto bi-dimensional simples e testa a capacidade de aglomeração do algoritmo em uma situação não-linearmente separável. É composto de dois arcos posicionados um acima do outro.

(X+) Este conjunto possui duas classes bi-dimensionais, uma em forma de "X" e outra em forma de um sinal de "+". Este é um conjunto também não-linearmente separável que explora a capacidade do algoritmo de ligar *clusters* ramificados.

(espirais) Este conjunto de duas classes em espiral explora muito bem o conceito de classes não-linearmente separáveis. Trata-se de um conjunto bi-dimensional complexo onde as distâncias Euclidianas muitas vezes são menores entre

clusters que não podem ser ligados. Isto testa bem a utilização das divergências apresentadas neste trabalho.

(círculo 2D) Este exemplo (também bi-dimensional) apresenta duas classes. Uma destas classes é uma classe composta de um círculo isotrópico dentro de um anel que forma a outra classe. Este conjunto explora a capacidade do algoritmo de aglomerar dados nesta situação (uma classe circundada por outra).

(EE) Neste conjunto (bi-dimensional), é explorado novamente a não linearidade da separação formando misturas ramificadas e entrelaçadas como no conjunto (X+). Desta vez as ramificações contribuem para o conjunto ser não-linearmente separável.

(EEE) Esta é uma versão tri-dimensional do conjunto (EE). Aqui, ramificações formam duas classes entrelaçadas em três dimensões também não-linearmente separáveis.

(círculos 3D) Este conjunto tri-dimensional consiste de 8 anéis (formando 8 classes) que se atravessam entre si formando uma corrente fechada. Este conjunto explora a capacidade de gerar superfícies de separação muito complexas.

(molas) Este exemplo é composto por duas classes tri-dimensionais em forma de espirais entrelaçadas. Esta é uma versão 3D da espiral e testa a capacidade de separação de dados bem entrelaçados em três dimensões.

(iso 3D) Este conjunto é formado por 5 esferas tri-dimensionais (conjuntos isotrópicos) posicionadas nos vértices de uma pirâmide. Este conjunto é muito simples e serve apenas para demonstrar a capacidade do algoritmo de informar o número de classes automaticamente quando as classes são bem separadas.

(simples) Como o próprio nome diz, este é um conjunto bi-dimensional simples de 3 classes, porém com uma das classes próxima da outra. Este conjunto mostra que nesta situação, nem sempre o algoritmo consegue informar o número de classes automaticamente.

(notas) Este conjunto de seis dimensões é composto de duas classes. Os pontos deste conjunto correspondem a medidas reais de distâncias feitas em notas extraídas do *Swiss Bank* [21]. As duas classes correspondem a notas falsas e verdadeiras. Este conjunto corresponde a uma aplicação real do algoritmo.

(iris) Este é um conjunto de teste reais de quatro dimensões muito utilizado em métodos de *clustering*. Trata-se das medidas de comprimento e largura das pétalas e sépalas de 3 tipos de flores. Este conjunto foi utilizado por Fisher (1936) entre outros como padrão de testes em algoritmos de *clustering*.

Os primeiros 10 conjuntos foram gerados artificialmente para promover o teste do algoritmo em situações de distribuição complexa. Os dois últimos são dados reais utilizados como teste do algoritmo em situações práticas. A tabela 4.1 apresenta um resumo das características dos conjuntos apresentados.

Tabela 4.1: Resumo das características dos conjuntos utilizados como teste

Conjunto	N	Dimensão	Classes	Tipo
(arcos)	502	2	2	Artificial
(X+)	804	2	2	Artificial
(espirais)	772	2	2	Artificial
(círculo 2D)	759	2	2	Artificial
(EE)	1168	2	2	Artificial
(EEE)	1604	3	2	Artificial
(círculo 3D)	5032	3	8	Artificial
(molas)	2000	3	2	Artificial
(iso 3D)	2400	3	5	Artificial
(simples)	1400	3	3	Artificial
(notas)	200	4	2	Real
(iris)	150	6	3	Real

Nesta tabela a coluna "N" indica o número de pontos presentes no conjunto. A dimensão dos dados é mostrada na coluna "Dimensão" e o número de classes na coluna "classes". A coluna "Tipo" indica se os dados foram gerados artificialmente ou correspondem a dados reais de conjuntos extraídos de problemas apresentados na literatura.

4.3.2 Descrição dos resultados

Os resultados da classificação são apresentados basicamente de duas maneiras. A primeira consiste na representação gráfica dos conjuntos (*scatter plot*) separando as classes por cores ou símbolos juntamente com uma representação dos centros dos *clusters* auxiliares e suas ligações. Um exemplo deste tipo de representação de resultado pode ser visto na Figura 4.10.

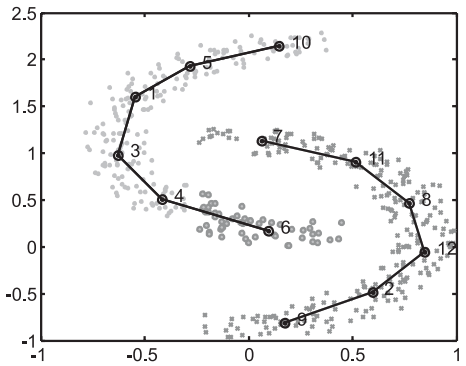


Figura 4.10: Exemplo de apresentação de um resultado de classificação

Os números que aparecem próximos aos centros dos *clusters* são apenas números de identificação e as linhas representam as ligações entre os *clusters*. Obviamente esta representação só pode ser utilizada em conjuntos bi ou tri-dimensionais. A segunda forma de apresentação dos resultados consiste na apresentação da matriz de confusão do conjunto classificado. Esta matriz possui tamanho $N_c \times N_c$ (sendo N_c o número de classes). Cada linha e coluna representa um classe. Os valores de cada elemento representam a quantidade de pontos que são realmente da classe "linha" e que foram rotulados como classe "coluna". Idealmente esta matriz deve ser diagonal e cada elemento deve conter o número total de pontos em cada classe. Pode ocorrer que a matriz não seja diagonal, porém a maioria dos elementos sejam classificados como uma classe sendo na verdade de outra (ver matriz a seguir)

$$\mathbf{C} = \begin{bmatrix} 95 & 0 & 5 \\ 3 & 0 & 97 \\ 1 & 98 & 1 \end{bmatrix}$$

Nesta matriz, observamos na linha 2, por exemplo, que 97 elementos deveriam

ser classificados como sendo de classe 2 (segunda linha da matriz) porém foram classificados como sendo classe 3 (terceira coluna). A princípio somos levados a entender esta classificação como erro, porém, na linha 3 da matriz ocorre o contrário, elementos que seriam de classe 3 foram classificados como classe 2. Isto significa que o algoritmo que classificou os dados desta maneira trocou apenas o rótulo da classe 2 com a classe 3, porém a classificação está coerente em sua maioria.

Para os dados gerados artificialmente (que possuem dimensão até 3) apresentamos os resultados em forma de gráfico. Para os dados reais do conjunto "notas" e "iris" são apresentadas as matrizes de confusão.

Ainda sobre a apresentação dos resultados. Serão apresentados também resultados do modelamento. Serão basicamente três formas de apresentação. A primeira será utilizada para apresentar os resultados de modelamento para *clusters* bi-dimensionais. Um exemplo desta é mostrado na Figura 4.11.

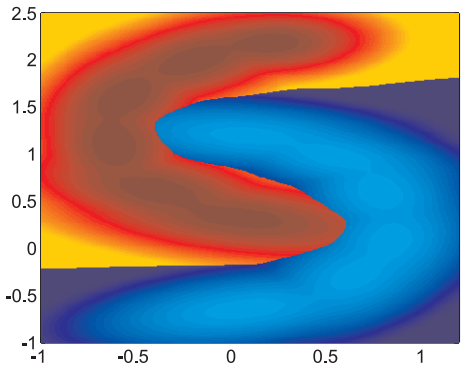


Figura 4.11: Exemplo do primeiro tipo de apresentação de um resultado de modelamento

Esta apresentação consiste de uma imagem onde cada ponto de coordenadas (x, y) é utilizado tendo sua probabilidade calculada pelo modelo. Na imagem cada ponto é colorido de acordo com sua probabilidade, em um gradiente de cores que indica como estão distribuídos os valores das probabilidades.

O outro tipo de representação consiste do gráfico de $p_i(x, y)$ sendo estas as distribuições de probabilidade de cada classe. Um exemplo desta representação é mostrada na Figura 4.12

Neste tipo de representação, os eixos X e Y são os eixos das variáveis do conjunto de dados ou seja, atributos (x_1, x_2) da tabela de dados utilizada pelo algoritmo

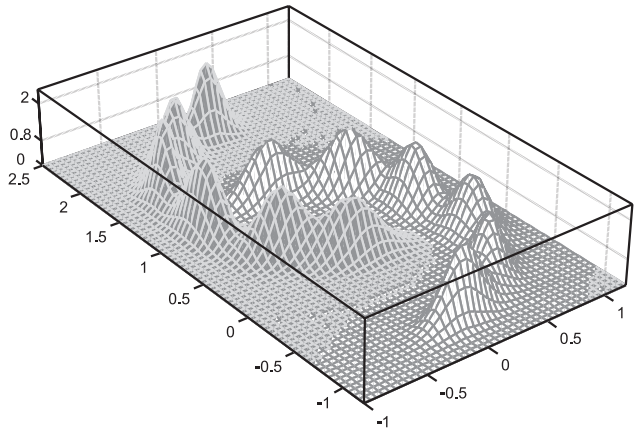


Figura 4.12: Exemplo do segundo tipo de apresentação de um resultado de modelamento

(altura e largura de um objeto por exemplo). O eixo Z corresponde a probabilidade de ocorrência deste vetor de atributos. As probabilidades de cada classes são representadas por funções de cores diferentes.

O último tipo de apresentação de resultados de modelamento tem como objetivo ilustrar o modelo de $p_i(x, y, z)$ para conjuntos tri-dimensionais. Para dar idéia da distribuição $p_i(x, y, z)$, são construídos gráficos de equipotenciais para um determinado valor de p_i. A Figura 4.13 ilustra um caso.

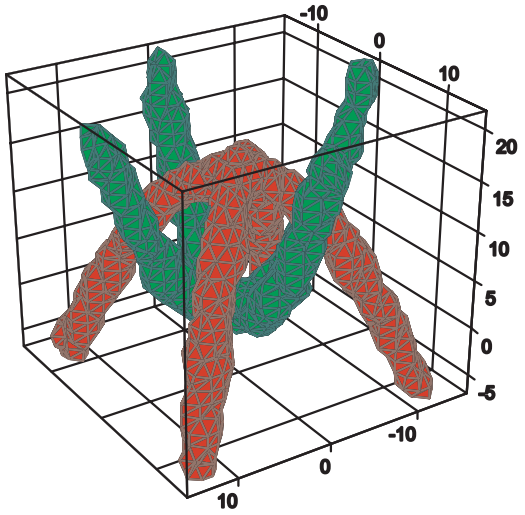


Figura 4.13: Gráfico de equipotencial para $p_i(x, y, z)$

Nesta figura, as cores indicam as classes. As figuras são formadas por uma superfície equipotencial de probabilidades, ou seja são o conjunto de pontos (x, y, z) que possuem mesmo valor de p_i.

4.3.3 Resultados de *clustering*

Nesta seção serão apresentados os resultados obtidos com o método de *clustering* desenvolvido. Serão utilizadas as representações descritas na seção anterior. Serão mostrados e descritos em detalhes os resultados de cada etapa realizada apenas para o primeiro exemplo (arcos), para os demais exemplos serão apenas mostrados e comentados os resultados finais.

A tabela 4.2 mostra os parâmetros utilizados para cada conjunto. Os parâmetros de treinamento da rede competitiva não são mostrados pois, como já mencionado, foram utilizados o mesmos valores para todos os conjuntos.

Tabela 4.2: Parâmetros utilizados nos exemplos de *clustering*

Conjunto	N_a	d_t (Rényi)	d_t (Mahala)	d_t (Bhatta)	d_t (KL)
(arcos)	8	0.3 - 0.5	7 - 10	2 - 2.8	47 - 65
(X+)	20	0.4 - 0.61	10 - 20	2.8 - 5	49 - 105
(espirais)	45	0.57 - 0.73	9 - 17	2.7 - 4.7	90 - 96*
(círculo 2D)	14	0.17 - 1.1	8.6 - 31.5	2.2 - 8	39 - 155
(EE)	40	0.37 - 0.8	11 - 37	2.9 - 9.4	59 - 158
(EEE)	50	0.63 - 1.5	11.6 - 90	2.7 - 22	77 - 370
(círculo 3D)	110	0.22 - 0.35*	9 - 13	2.2 - 3.3	40 - 54.5*
(molas)	80	0.31 - 0.53	8.7 - 18.5	2.2 - 4.7	48 - 82
(iso 3D)	20	0.1 - 0.42	5.6 - 12.7	1.4 - 3.3	25 - 65
(simples)	10	0.21 - 0.59	6.4 - 21	1.58 - 5.3	26 - 90
(notas)	4	1.025 - 1.08**	5 - 17	1.6 - 4.7	40 - 97
(iris)	6	0.44 - 0.56**	***	***	***

* Número de classes não foi encontrado automaticamente, este valor teve que ser informado para ocorrer a classificação correta.
** Erro de mais de 10% na classificação.
*** Não foi possível encontrar a classificação correta.

Nesta tabela, cada linha corresponde a um conjunto de dados. A coluna 2 apresenta o número de *clusters* auxiliares iniciais. As colunas de 3 a 6 apresentam as faixas de divergências utilizadas que correspondem a classificação correta e a definição automática do número de classes (exceto para os casos marcados com asteriscos). Podemos perceber que na maioria dos casos o número de classes é encontrado automaticamente.

Analisando os parâmetros dos conjuntos reais (notas) e (iris) percebemos que estes são os mais difíceis de se estabelecer a classificação correta. Embora

modificando os parâmetros cuidadosamente possamos alcançar a classificação correta, procurou-se demonstrar que em geral esta classificação é difícil. Isto de deve a pouca quantidade de pontos e a ocorrência de pontos sobrepostos entre as classes (no caso do conjunto (iris)), como será mostrado mais adiante. os parâmetros foram estabelecidos de acordo com as regras gerais utilizadas para os demais conjuntos. O ajuste cuidadoso dos parâmetros embora consigam fazer com que o algoritmo chegue na classificação correta, corresponde a uma situação pouco prática. Em geral não se tem informação suficiente para realizar este ajuste fino, portanto os resultados de classificação são deixados fracos propositalmente.

4.3.3.1 Análises dos resultados

A seguir, serão mostrados os resultados passo-a-passo da classificação do conjunto (arcos). Os demais conjuntos realizam os mesmos passos, de maneira que apenas os resultados são diferentes. A Figura 4.14 mostra a seqüência de passos e as representações gráficas em cada passo.

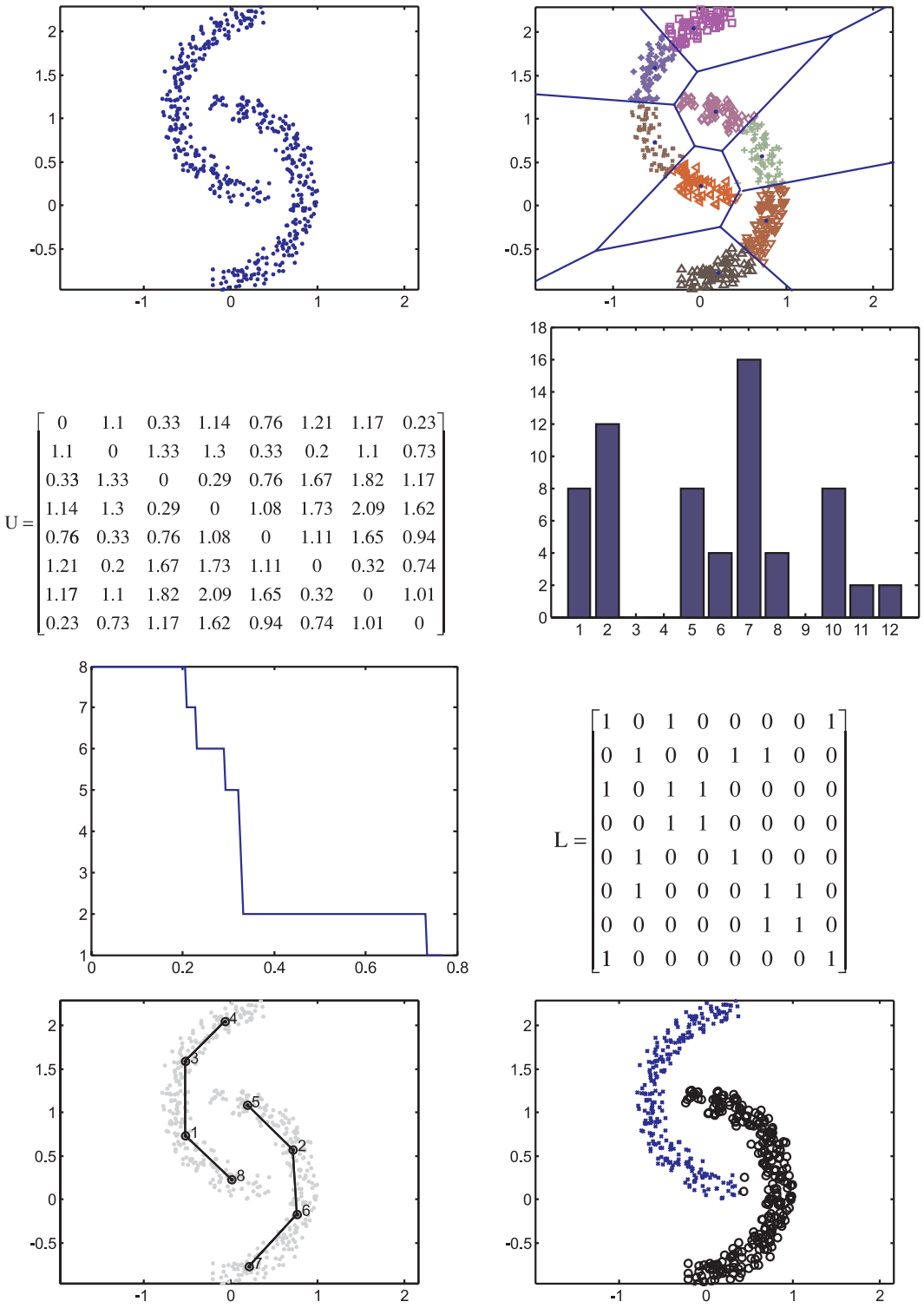


Figura 4.14: Seqüência de passos realizados para realizar o agrupamento total de um conjunto

Inicialmente o conjunto original é quantizado com N_a *clusters*. Em seguida, é calculada a matriz de divergências **U** que relaciona cada par de *clusters* auxiliares.

Logo após são gerados os histogramas e os gráficos do número de classes como apresentado nas seções 4.2.4 e 4.2.3. Estes gráficos servem para decidir (manual ou automaticamente) pelo limiar correto. Escolhido o limiar, a matriz de ligação **L** é gerada. Ao final os *clusters* são agrupados em classes e os dados são rotulados. O histograma e o gráfico do número de classes em função do limiar foram construídos utilizando a divergência de Rényi, porém o procedimento é exatamente o mesmo para qualquer uma das divergências analisadas neste trabalho.

A seguir são mostrados os resultados dos demais conjuntos. Como já mencionado, não serão apresentados novamente os resultados por etapas como foi feito para o conjunto (arcos). Para cada conjunto são mostrados os histogramas e gráficos do número de classes para todas as quatro divergências analisadas. Também serão mostrados o conjunto original e o conjunto final classificado.

Os gráficos do histograma e do número de classes são apresentados sem os rótulos dos eixos por questões visualização. Em ambos o eixo X corresponde a divergência. No histograma, o eixo Y corresponde a freqüência de ocorrência e no gráfico do número de classes o eixo Y corresponde a quantidade de classes encontradas para cada divergência limiar. O gráfico do número de classes é apresentado na maioria dos casos na forma de gráfico semi-logaritmo para facilitar a visualização.

A matriz de confusão para este caso (conjunto (arcos)) é mostrada a seguir

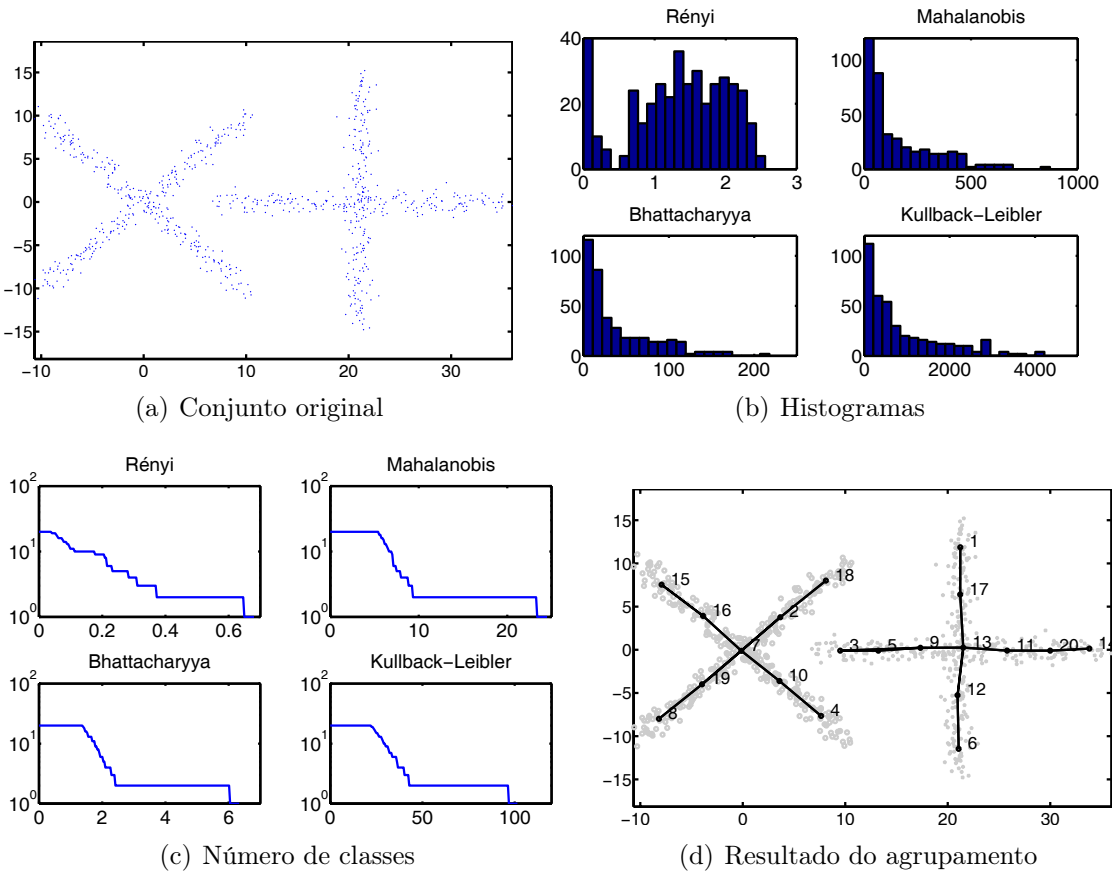
$$\boldsymbol{C} = \begin{bmatrix} 252 & 0 \\ 2 & 248 \end{bmatrix}. \tag{4.2}$$

Existem alguns poucos pontos classificados erroneamente. Este fato decorre da quantização vetorial com poucos centros. A presença de 1 ou 2 centros a mais na etapa inicial do algoritmo levaria a uma classificação de 100%. Em todos os casos, a matriz de confusão é a mesma para cada número de classes estabelecida, não dependendo da métrica utilizada. Isto porque a classificação dos pontos do conjunto já foi realizada na etapa de quantização vetorial.

4.3.3.2 Conjunto (X+)

Os resultados para o conjunto (X+) são apresentados na Figura 4.15. Observando o gráfico do número de classes em função do limiar, podemos perceber que o número de classes pode ser encontrado automaticamente pelo algoritmo em qualquer uma das medidas de divergência. Para este caso, a distância de Mahalanobis foi a medida de divergência que mais estabilizou em duas classes, embora todos as divergências propiciem a determinação do número de classes automaticamente. A matriz de confusão para este conjunto é dada por

$$\boldsymbol{C} = \begin{bmatrix} 402 & 0 \\ 0 & 402 \end{bmatrix}. \tag{4.3}$$



(a) Conjunto original (b) Histogramas

(c) Número de classes (d) Resultado do agrupamento

Figura 4.15: Resultados para o conjunto (X+)

4.3.3.3 Conjunto (espirais)

Os resultados para o conjunto (espirais) são apresentados na Figura 4.16. Este é um conjunto bastante complexo de ser agrupado. Embora os resultados apresentados sejam bons, e em todas as medidas de divergência o número de classes seja alcançado automaticamente, nota-se algumas regiões de estabilidade falsas. Estas regiões embora pequenas, podem aparecer maiores em outras execuções do algoritmo.

Um ponto importante a ser observado neste conjunto é a proximidade dos centros. Pode-se ver claramente que os centros 37 e 20 estão mais próximos entre si que os centros 8 e 3 por exemplo. Os centros 4 e 9 fazem parte de classes diferentes e, embora estejam mais próximos, não podem ser ligados. De fato utilizando as medidas de divergência apresentadas neste trabalho, os mesmos não são ligados e a classificação segue corretamente.

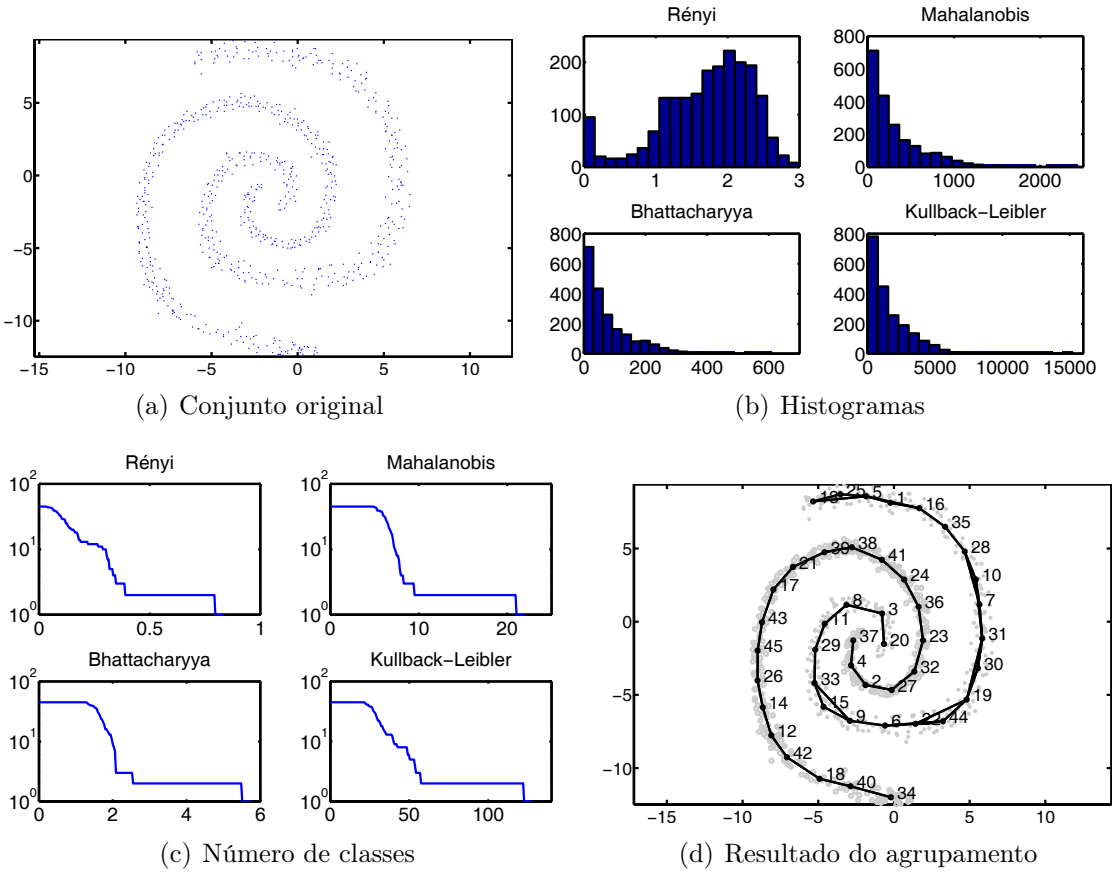
A matriz a seguir corresponde a matriz de confusão calculada para este conjunto.

$$\boldsymbol{C} = \begin{bmatrix} 367 & 0 \\ 0 & 405 \end{bmatrix}. \tag{4.4}$$

4.3.3.4 Conjunto (círculo 2D)

Os resultados para o conjunto (círculo 2D) são apresentados na Figura 4.17. Este conjunto, embora tenha uma disposição espacial bastante peculiar (uma classe dentro de outra) se mostrou bastante fácil de ser separado utilizando qualquer uma das métricas. Em todos os casos há uma estabilização forte no número correto de classes, como pode ser observado no gráfico do número de classes, sendo este determinado automaticamente pelo algoritmo. A seguir é apresentada a matriz de confusão para este conjunto

$$\boldsymbol{C} = \begin{bmatrix} 0 & 130 \\ 629 & 0 \end{bmatrix}. \tag{4.5}$$



(a) Conjunto original (b) Histogramas

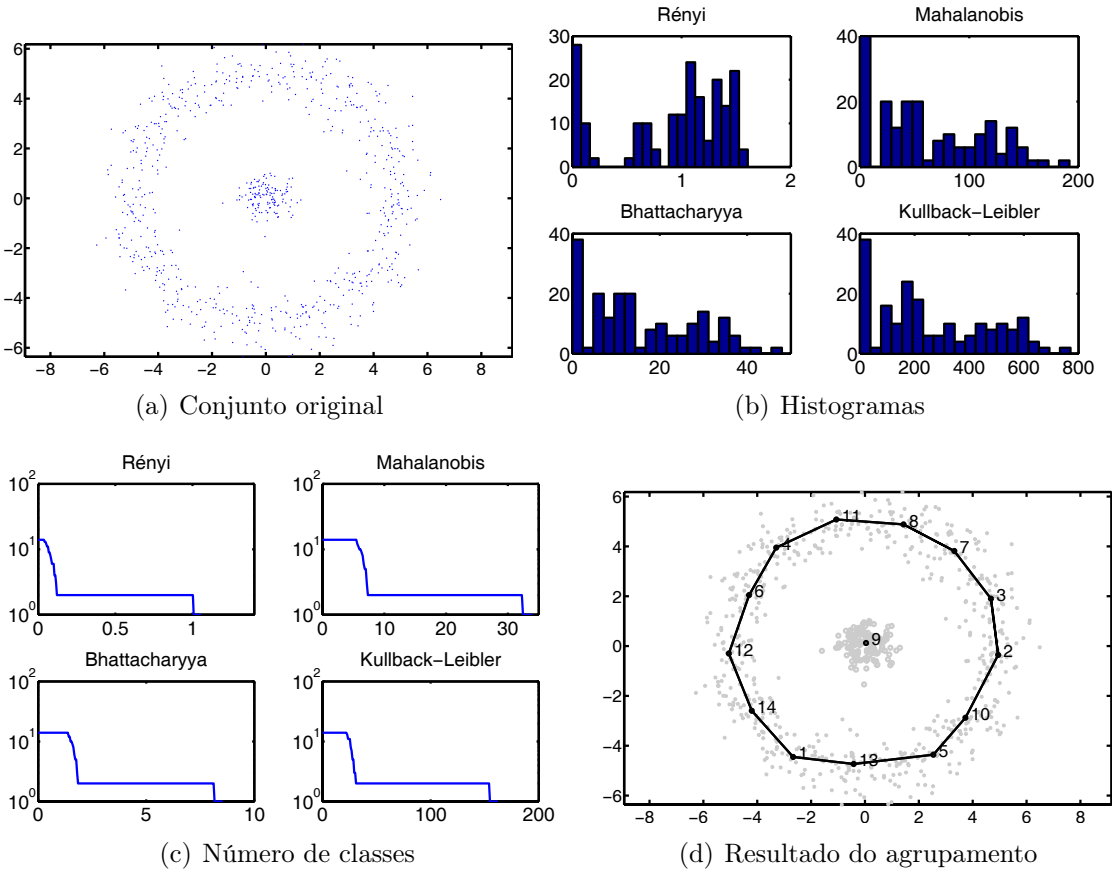
(c) Número de classes (d) Resultado do agrupamento

Figura 4.16: Resultados para o conjunto (espirais)

4.3.3.5 Conjunto (ee)

Este é um conjunto bastante entrelaçado e com ramificações. Os resultados para este conjunto são apresentados na Figura 4.18. Para todas as divergências, houve uma estabilização forte em duas classes, que é o número correto de classes. Conforme podemos observar no último gráfico, as ligações são feitas de maneira não só a separar corretamente as classes, mas a definir e modelar as ramificações corretamente. O número de classes também foi determinado pelo algoritmo automaticamente. A matriz de confusão é mostrada a seguir

$$\boldsymbol{C} = \begin{bmatrix} 584 & 0 \\ 0 & 584 \end{bmatrix}. \tag{4.6}$$



(a) Conjunto original (b) Histogramas

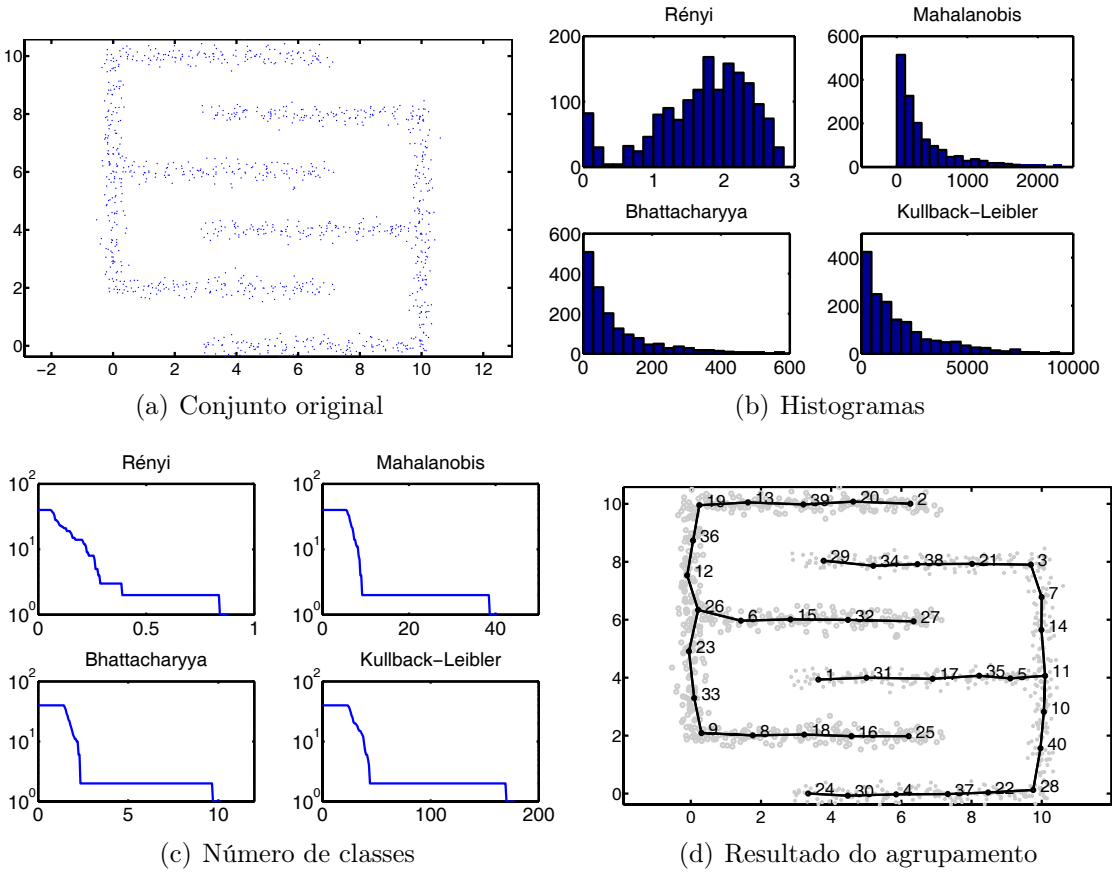
(c) Número de classes (d) Resultado do agrupamento

Figura 4.17: Resultados para o conjunto (círculo 2D)

4.3.3.6 Conjunto (eee)

Esta é uma versão tri-dimensional do conjunto (ee). Os resultados para este conjunto são apresentados na Figura 4.19. Devido a grande separação espacial entre as classes, mesmo que entrelaçadas, podemos observar uma área muito grande de estabilização no gráfico do número de classes. Como no exemplo do conjunto (ee), este conjunto foi separado corretamente e as ramificações foram bem estabelecidas pelas ligações entre os *clusters* auxiliares. Como nos anteriores, o número de classes foi estabelecido pelo algoritmo automaticamente. A matriz de confusão deste conjunto é dada por

$$\boldsymbol{C} = \begin{bmatrix} 0 & 802 \\ 802 & 0 \end{bmatrix}. \tag{4.7}$$



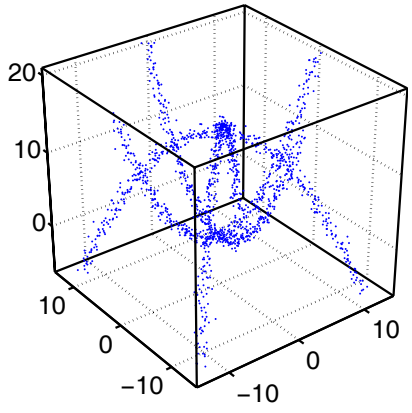
(a) Conjunto original (b) Histogramas

(c) Número de classes (d) Resultado do agrupamento

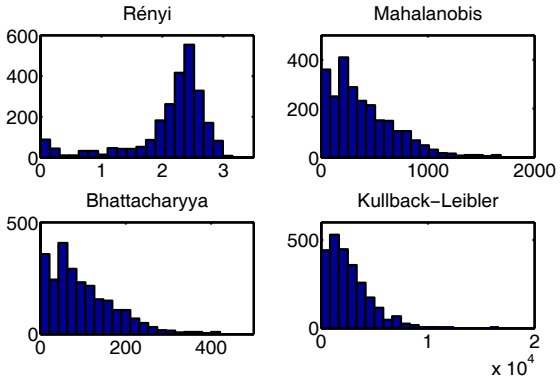
Figura 4.18: Resultados para o conjunto (ee)

4.3.3.7 Conjunto (círculos 3D)

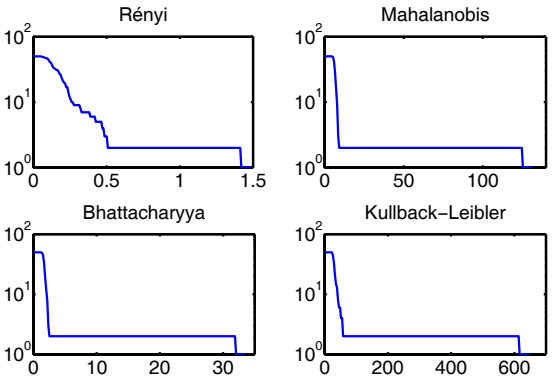
Os resultados para o conjunto (círculos 3D) são apresentados na Figura 4.20. Observando o gráfico do número de classes para todas as divergências, percebemos a presença de uma região de estabilidade em 8 classes. Deste modo, o algoritmo obteve sucesso na classificação e determinou o número de classes automaticamente, gerando a seguinte matriz de confusão



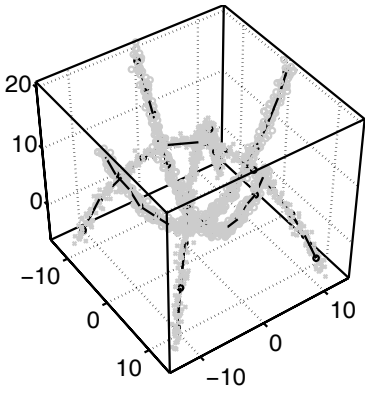
(a) Conjunto original



(b) Histogramas



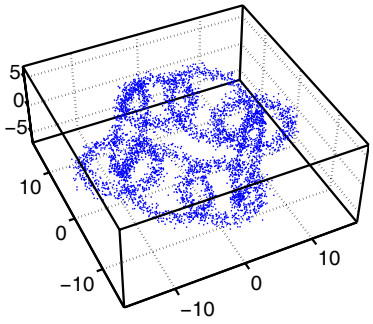
(c) Número de classes



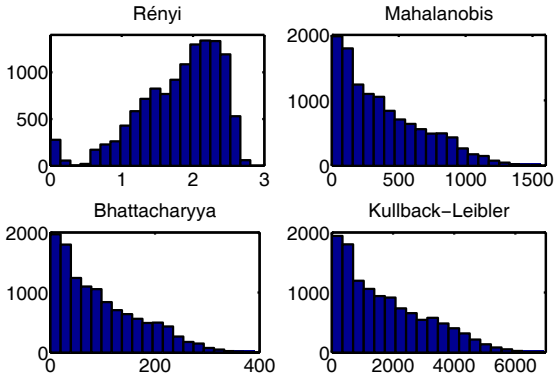
(d) Resultado do agrupamento

Figura 4.19: Resultados para o conjunto (eee)

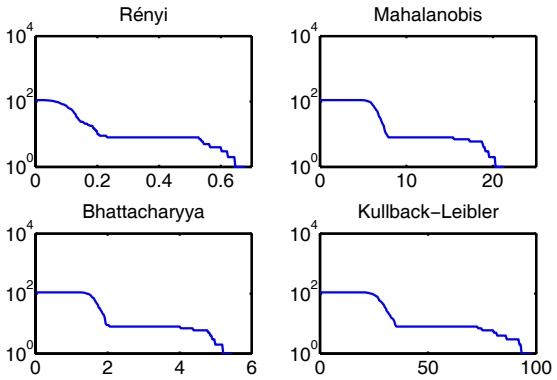
$$\boldsymbol{C} = \begin{bmatrix} 0 & 628 & 0 & 0 & 1 & 0 & 0 & 0 \\ 629 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 629 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 627 & 0 \\ 0 & 0 & 0 & 0 & 629 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 629 \\ 1 & 0 & 0 & 628 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 629 & 0 & 0 \end{bmatrix} \tag{4.8}$$



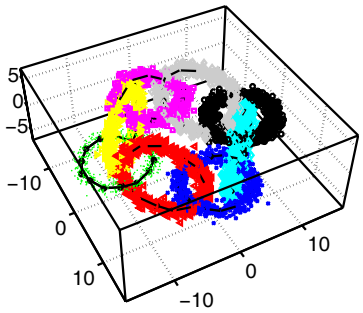
(a) Conjunto original



(b) Histogramas



(c) Número de classes



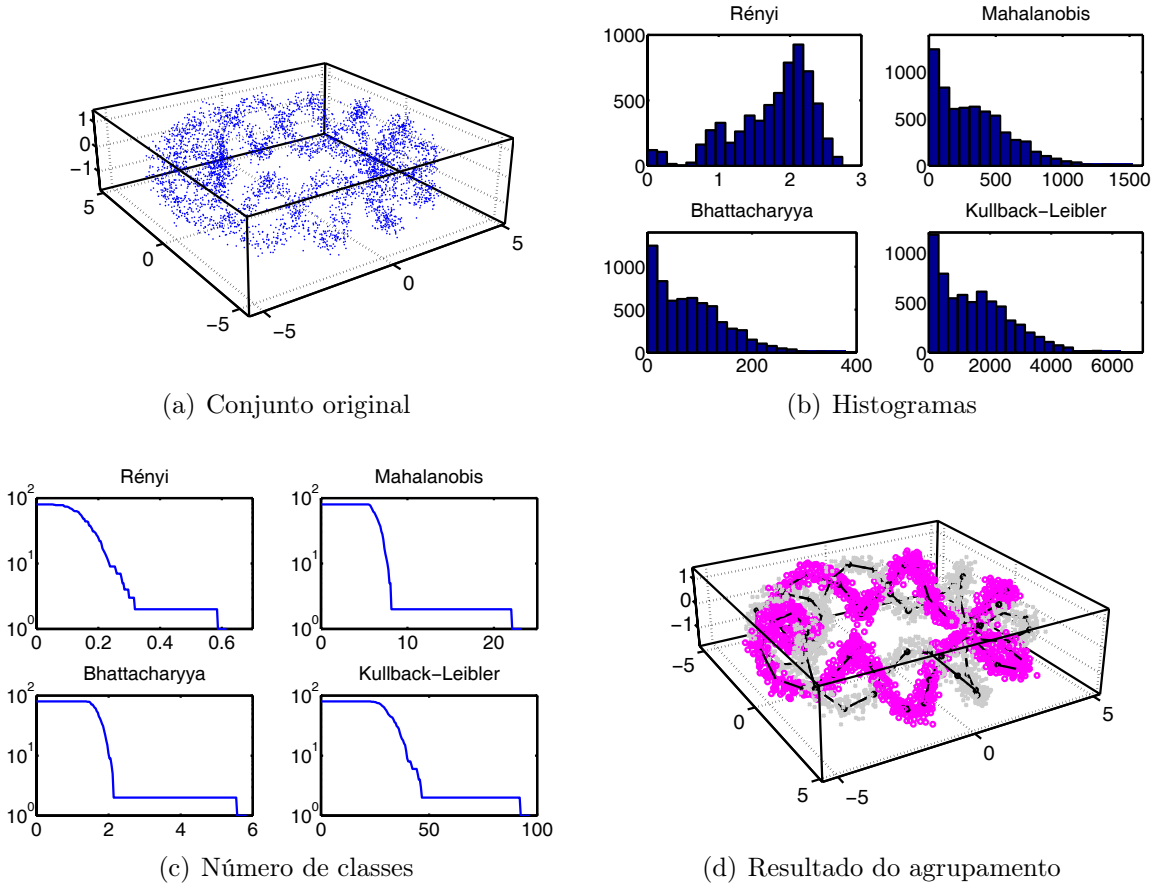
(d) Resultado do agrupamento

Figura 4.20: Resultados para o conjunto (círculos 3D)

4.3.3.8 Conjunto (molas)

Os resultados para o conjunto (molas) são apresentados na Figura 4.21. Neste conjunto as duas classes estão bastante entrelaçadas e próximas. Mesmo nestas condições, o algoritmo foi capaz de (em todas as divergências) estabelecer uma região de estabilidade no número correto de classes. Com esta estabilização a classificação foi realizada com sucesso. A matriz de confusão é apresentada a seguir

$$\boldsymbol{C} = \begin{bmatrix} 998 & 2 \\ 1 & 999 \end{bmatrix}. \tag{4.9}$$



(a) Conjunto original

(b) Histogramas

(c) Número de classes

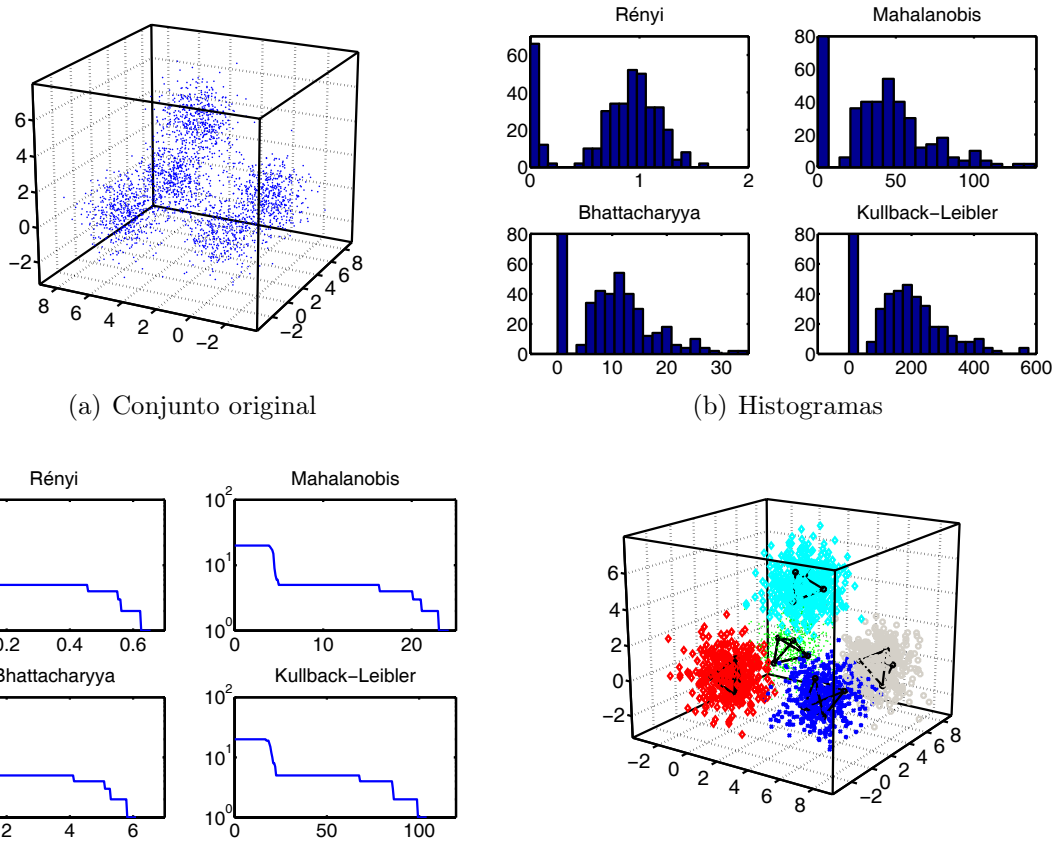
(d) Resultado do agrupamento

Figura 4.21: Resultados para o conjunto (molas)

4.3.3.9 Conjunto (iso 3D)

Os resultados para o conjunto (iso 3D) são apresentados na Figura 4.22. Este exemplo serve para ilustrar que em casos simples de *clusters* isotrópicos, o algoritmo se comporta como a maioria dos algoritmos convencionais (*k-means*, etc...) e ainda é capaz de encontrar automaticamente o número de classes. Como podemos observar no gráfico do número de classes, em qualquer medida de divergência, há uma estabilização em 5 classes, que é o número correto. A matriz de confusão calculada para este conjunto é dada por

$$\boldsymbol{C} = \begin{bmatrix} 1 & 398 & 0 & 0 & 1 \\ 1 & 0 & 1 & 498 & 0 \\ 0 & 0 & 0 & 0 & 500 \\ 0 & 0 & 498 & 0 & 2 \\ 500 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.10}$$



(a) Conjunto original (b) Histogramas

(c) Número de classes (d) Resultado do agrupamento

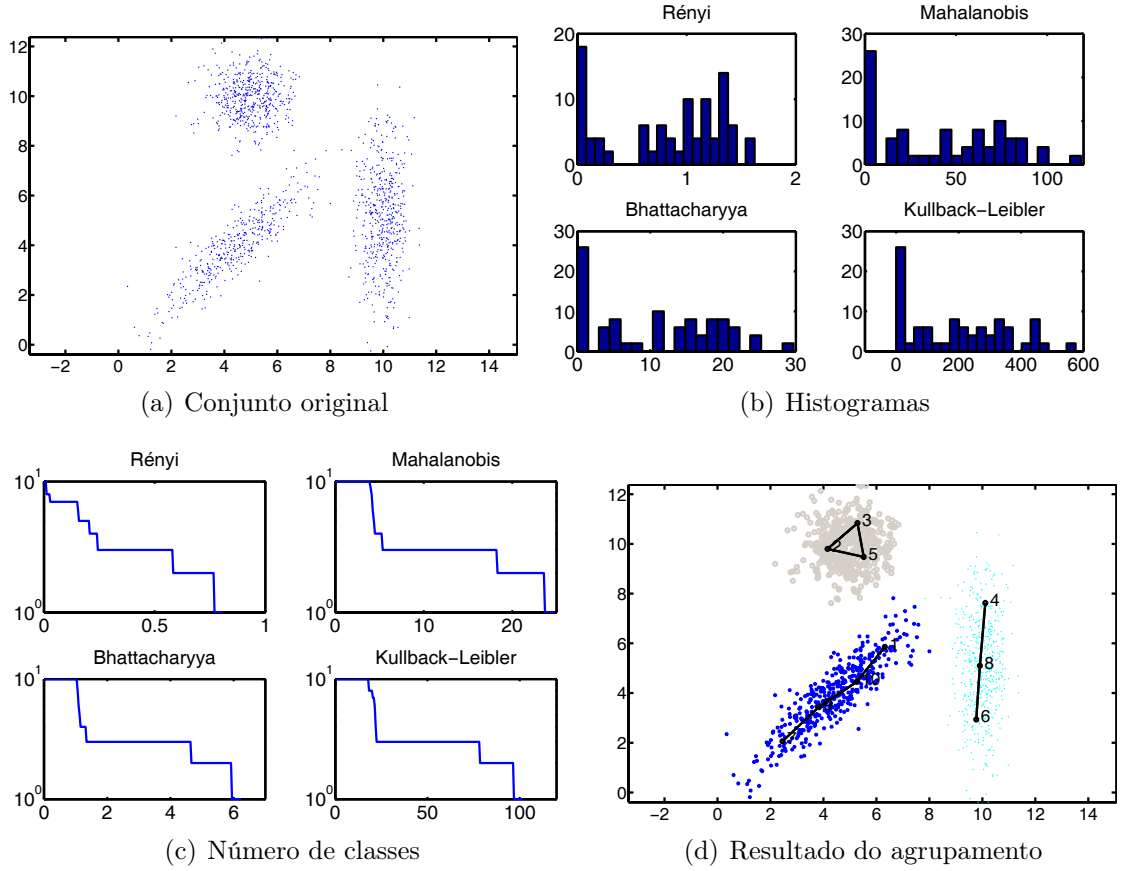
Figura 4.22: Resultados para o conjunto (iso 3D)

4.3.3.10 Conjunto (simples)

Este conjunto é composto de 3 classes simples, porém não-linearmente separáveis. Os pontos "dispersos" presentes nas classes fazem com que as medidas de divergência sejam próximas entre os *clusters*. Este fato pode ser observado analisando-se o gráfico do número de classes na Figura 4.23. Embora todas as medidas de divergência possuam uma região de estabilidade em 3 classes (que é o número correto), existem

pequenos degraus em duas classes, reflexo do efeito da dispersão dos dados. A matriz de confusão para este conjunto é dada por

$$\boldsymbol{C} = \begin{bmatrix} 0 & 500 & 0 \\ 0 & 0 & 500 \\ 396 & 0 & 4 \end{bmatrix} \tag{4.11}$$



(a) Conjunto original (b) Histogramas

(c) Número de classes (d) Resultado do agrupamento

Figura 4.23: Resultados para o conjunto (simples)

4.3.3.11 Conjunto (notas)

O conjunto (notas) é um conjunto de dados bastante utilizado na literatura. Trata-se de um conjunto de dados de medidas feitas em notas do *Swiss Bank* como mostrado na Figura 4.24 [21]

Estas medidas são utilizadas para decidir se a nota é verdadeira ou falsa. Os dados possuem 6 dimensões e portanto não é possível visualizá-los me forma de um

Figura 4.24: Exemplo das medidas extraídas de notas utilizadas como conjunto de teste

scatter plot. Os resultados apresentados consistem nos gráficos do número de classes (Figura 4.25) e da matriz de confusão (mostrada a seguir).

$$\boldsymbol{C} = \begin{bmatrix} 99 & 1 \\ 0 & 100 \end{bmatrix}. \tag{4.12}$$

Como podemos observar, a classificação foi feita quase totalmente correta, errando apenas a classificação de uma amostra em apenas uma das classes.

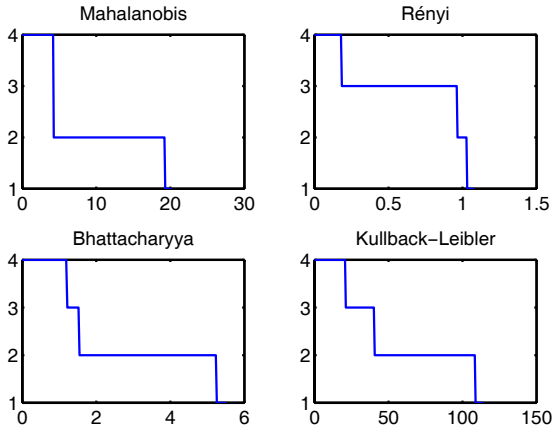


Figura 4.25: Número de centros em função do limiar

Observando a Figura 4.25 podemos observar que na divergência de Rényi, se faz necessário informar o número de classes, pois não ha estabilização suficiente no número correto de classes (duas). Nas demais medidas, o número de classes é automaticamente informado e a classificação ocorre corretamente.

4.3.3.12 Conjunto (iris)

O conjunto (iris) é um conjunto de dados também bastante utilizado na literatura. Trata-se de um conjunto de dados de medidas feitas em um certos tipos de flores (chamadas iris). Existem três classes de flores: a iris versicolor, iris setosa e iris virginica. Os dados possuem 4 dimensões que são a largura e altura da pétala e largura e altura da sépala. Como existem 4 dimensões, não é possível visualizar em um *scatter plot* (como no caso das notas). A Figura 4.26 mostra duas, as quatro dimensões presentes no conjunto.

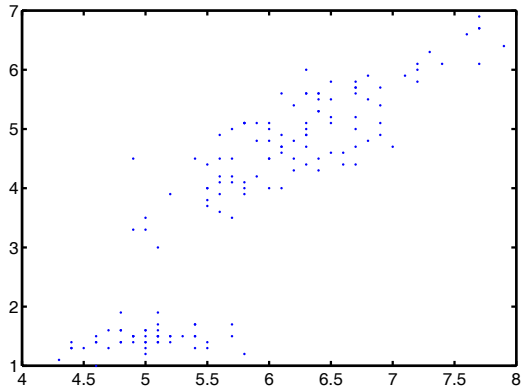


Figura 4.26: Visualização de duas das quatro dimensões do conjunto (iris)

Observando a Figura 4.26, só observamos a presença de dois possíveis *clusters*. De fato duas das três classes estão bastante próximas, havendo inclusive sobreposição. Na Figura 4.27 observamos os gráficos do número de classes. Como podemos observar, de fato apenas a divergência de Rényi foi capaz de separar (informando o número correto de classes) os dados. Mesmo assim, a sobreposição entre as duas classes causou um erro de classificação muito grande, como podemos observar na matriz de confusão a seguir

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 50 \\ 0 & 50 & 0 \\ 12 & 38 & 0 \end{bmatrix} \tag{4.13}$$

Em todas as medidas de divergência, observa-se a estabilização em duas classes. De fato para este tipo de dados que não possui separação espacial entre as classes, o algoritmo não é adequado.

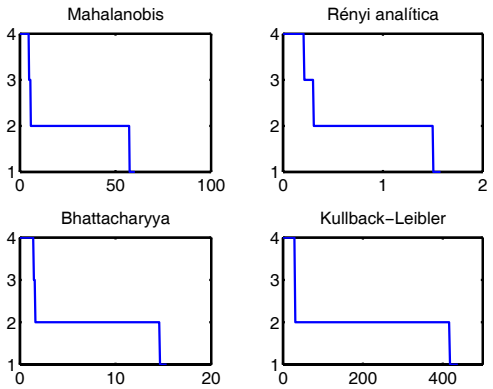


Figura 4.27: Número de classes em função do limiar para o conjunto (iris)

4.3.4 Resultados de modelamento

Nesta seção serão apresentados os resultados de modelamento para cada um dos conjuntos testados. Como já explicado na seção 4.3.2 os resultados de modelamento correspondem a obtenção de funções $p_i(\mathbf{x})$ que calculam a probabilidade de um ponto $\mathbf{x}$ pertencer a classe i. Os resultados são mostrados em forma de gráficos de p_i. Como já citado, quando se tratar de um conjunto tri-dimensional, os gráficos correspondem a equipotenciais em algum valor de p_i.

As Figuras de 4.28 a 4.36 mostram os resultados. Apenas os conjunto artificiais são apresentados pois possuem dimensão até 3.

Podemos perceber em todos os resultados que os modelos são bem complexos. Isto é possível graças ao modelo de misturas utilizado. Com o correto agrupamento dos *clusters* auxiliares, estes compõem as misturas de gaussianas e podem formar padrões bem complexos.

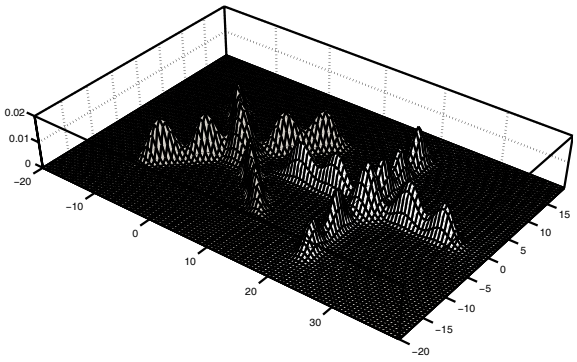


Figura 4.28: Gráfico de p_i para o conjunto (X+)

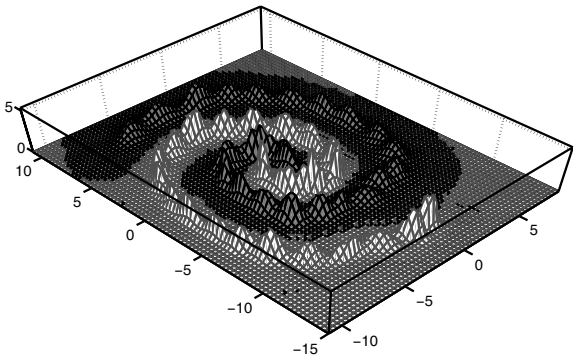


Figura 4.29: Gráfico de p_i para o conjunto (espirais)

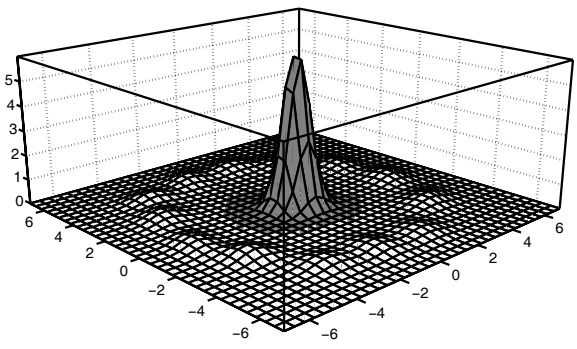


Figura 4.30: Gráfico de p_i para o conjunto (círculo 2D)

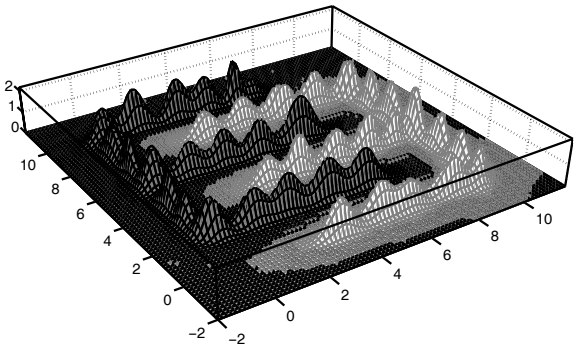


Figura 4.31: Gráfico de p_i para o conjunto (ee)

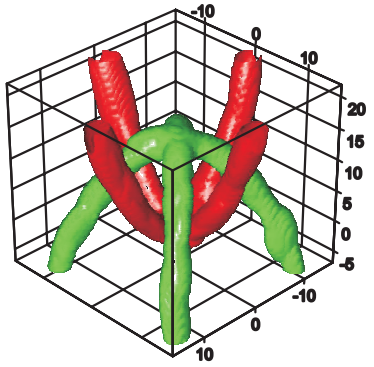


Figura 4.32: Gráfico de p_i para o conjunto (eee)

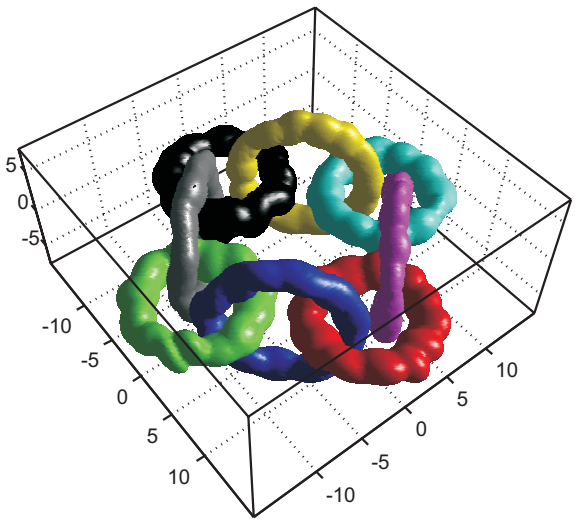


Figura 4.33: Gráfico de p_i para o conjunto (círculos 3D)

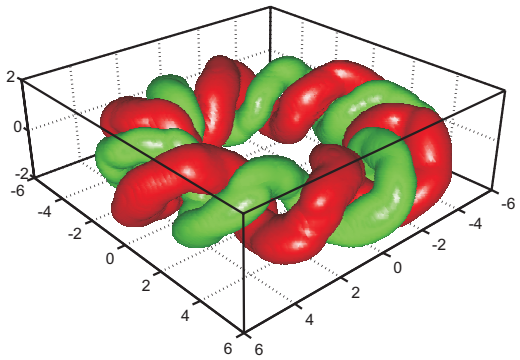


Figura 4.34: Gráfico de p_i para o conjunto (molas)

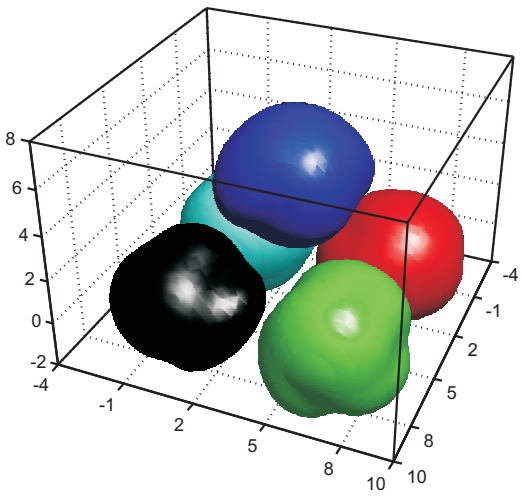


Figura 4.35: Gráfico de p_i para o conjunto (iso 3D)

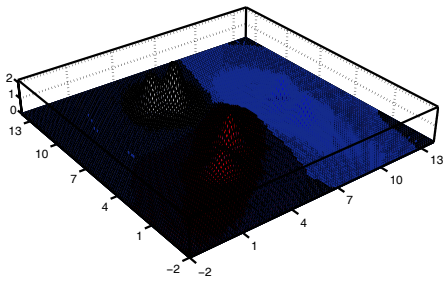


Figura 4.36: Gráfico de p_i para o conjunto (simples)

4.3.5 Estudo de casos

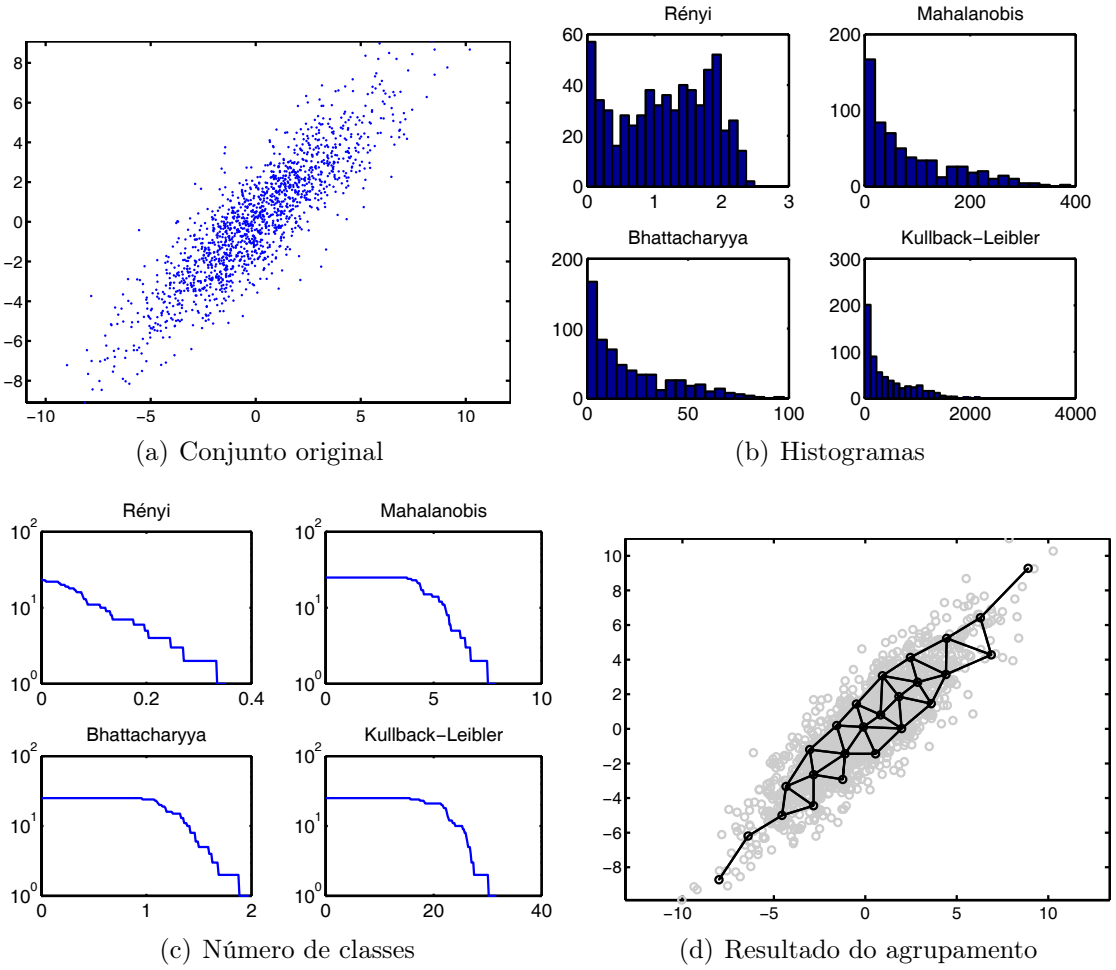
Nesta seção será feito um breve estudo de dois casos particulares de utilização do método proposto. O primeiro caso corresponde a situação onde os dados estão distribuídos em apenas uma classe. O outro conjunto corresponde a classes separadas porém com ruído em forma de pontos uniformemente distribuídos em torno dos pontos que formam as classes. Estes casos são interessantes porque ilustram situações diferentes que podem ocorrer em casos práticos e testa a robustez do algoritmo.

4.3.5.1 *Classe* simples

No primeiro teste o algoritmo recebeu um conjunto com 1500 pontos de duas dimensões gerados por uma distribuição gaussiana não isotrópica. Os pontos são mostrados na Figura 4.37(a). O algoritmo rodou com 25 *clusters* auxiliares e com os parâmetros de treinamento da rede competitiva iguais aos utilizados em todo trabalho. Os resultados são mostrados na Figura 4.37.

Observando os gráficos do número de classes em função do limiar, pode-se verificar que nas métricas de Mahalanobis, Bhattacharyya e Kullback-Leibler, a única região de estabilidade está em 25 classes. Após esta estabilização, o número de classes cai bruscamente. Isto quer dizer que aumentando-se o limiar, existe um momento em que as ligações ocorrem quase todas de uma só vez. Este é um bom indício da presença de uma classe apenas, pois todos os *clusters* auxiliares estão mais ou menos eqüidistastes em relação a estas três métricas. Na métrica de Rényi, observa-se pequenas regiões de estabilidade, mesmo com a queda constante do número de classes. Em compensação, o vale presente no histograma das divergências não é bem acentuado e não coincide com a região de maior estabilidade no gráfico do número de classes.

A conclusão que se pode chegar é que, embora o algoritmo não possa decidir diretamente sobre a presença de uma classe apenas, podemos supor esta situação observando os comportamentos anormais nos gráficos do histograma e no número de classes. Uma heurística poderia ser pensada para que esta decisão fosse tomada



(a) Conjunto original

(b) Histogramas

(c) Número de classes

(d) Resultado do agrupamento

Figura 4.37: Resultados para o conjunto de testes com apenas uma classe

automaticamente, porém este assunto não foi tratado neste trabalho e fica como sugestão para trabalhos futuros.

4.3.5.2 Presença de ruído

A outra situação a que o método foi submetido foi aquela onde existe ruído distribuído nos dados. O conjunto possui 4400 pontos bi-dimensionais e 4 classes distribuídas como mostra a Figura 4.38.

O algoritmo foi executado utilizando-se 100 *clusters* auxiliares e os mesmos parâmetros para a etapa de quantização vetorial utilizado em todo o trabalho. A Figura 4.39 mostra os gráficos do número de classes em função do limiar e o histograma para todas as divergências.

Como podemos observar, para as métricas de Bhattacharyya, Kullback-Leibler e

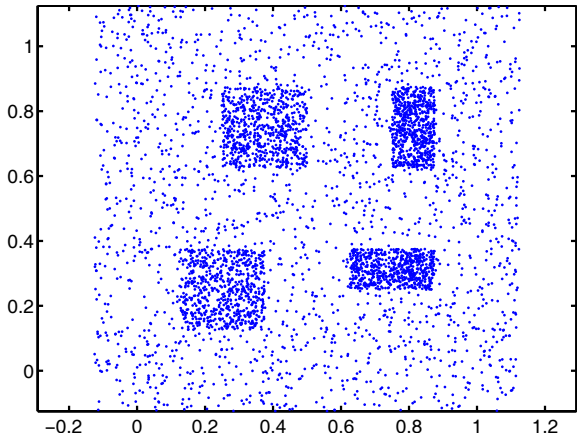
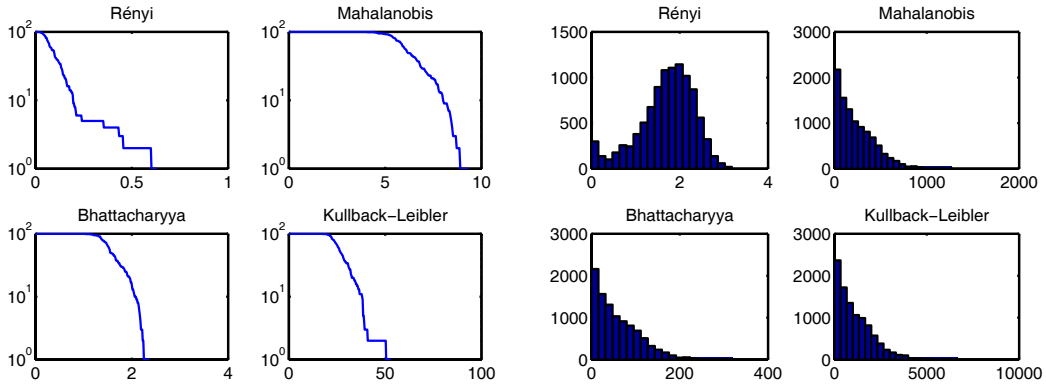


Figura 4.38: Conjunto apresentando ruido uniforme dentre as classes

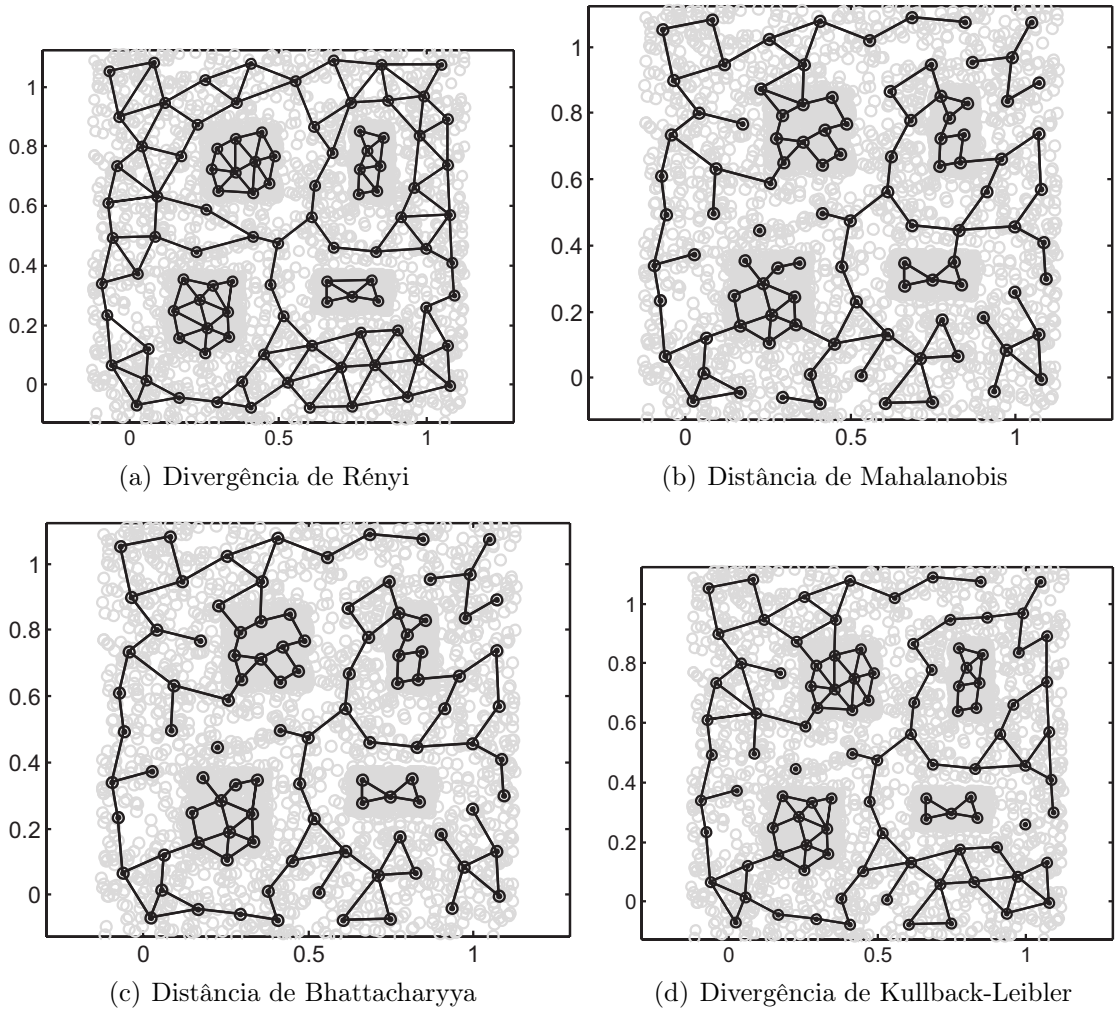


(a) Número de classes em função do limiar (b) Histogramas

Figura 4.39: Resultados para o conjunto com ruído entre as classes

Mahalanobis, não houve estabilização concreta e a situação pareceu bastante com o ocorrido para o conjunto com uma classe. Entretanto, a métrica Rényi apresentou estabilizações próximas ao vale do histograma de divergências. Não há uma região predominante de estabilização porém o gráfico mostra uma primeira região ocorrendo em 5 classes. A presença de 5 classes pode ocorrer se o método conseguir separar o ruído como sendo uma classe extra, o que vem a ser bastante conveniente em um algoritmo de *clustering*. Isto é exatamente o que o ocorre quando se utiliza a divergência de Rényi, proposta neste trabalho. As demais métricas não conseguem segmentar corretamente o ruído mesmo escolhendo-se o limiar que separa o conjunto em 5 classes. A Figura 4.40 mostra o resultado da classificação para 5 classes utilizando todas as divergências.

Como podemos observar, a divergência de Rényi consegue separar as classes



(a) Divergência de Rényi

(b) Distância de Mahalanobis

(c) Distância de Bhattacharyya

(d) Divergência de Kullback-Leibler

Figura 4.40: Ligações estabelecidas para cada medida de divergência

e isolar o ruído em uma outra classe. Para que isto seja possível, é necessária informação do número de classes, pois a estabilização no número correto não é sempre a maior ocorrência no gráfico do número de classes.

A Figura 4.41 mostra o resultado da classificação utilizando a divergência de Rényi. Na figura podemos observar alguns poucos pontos pertencentes ao ruído sendo classificados como sendo de alguma classe nas interfaces entre as mesmas e o ruído. Isto se deve a quantidade pequena de pontos na fase de quantização vetorial, que segmenta alguns pontos de ruído juntamente com pontos de alguma classe. Este problema pode ser ainda minimizado utilizando-se mais *clusters* auxiliares.

Este resultado só é possível devido a contabilização da probabilidade *a priori* no cálculo da divergência de Rényi. Neste tipo de conjunto, a quantização vetorial

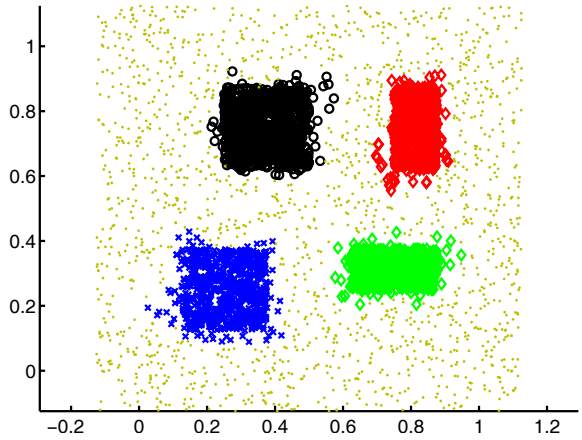


Figura 4.41: Conjunto com ruído classificado utilizando a divergência de Rényi

tende a gerar *clusters* auxiliares isotrópicos devido a uniformidade da distribuição dos pontos. Para as métricas de Mahalanobis, Bhattacharyya e Kullback-Leibler, quando temos *clusters* isotrópicos, o que diferencia um *cluster* de outro é apenas a sua distância Euclidiana (normalizada pela matriz de covariância). Desta maneira é muito difícil distinguir se dois *clusters*, um de uma classe e outro de ruído devem ou não ser ligados quando estes estão nas bordas de uma classe. Na métrica desenvolvida neste trabalho, o *cluster* de ruído terá probabilidade *a priori* um pouco menor que um *cluster* pertencente a uma classe. Nesta situação, a divergência é maior que a divergência obtida quando as probabilidades *a priori* são iguais.

A separação entre *cluster* próximos às bordas das classes promovida pela divergência de Rényi não é suficiente para gerar uma estabilização muito grande no gráfico do número de classes. Isto porque o algoritmo de quantização vetorial tente a deixar as probabilidades *a priori* iguais, colocando mais pontos nas regiões mais densas. Isto porém não é suficiente para equalizar totalmente as probabilidades, deixando uma pequena folga que é utilizada para permitir a separação utilizando a métrica de Rényi.

4.3.6 Sensibilidade às condições iniciais

Em todos os testes realizados com o método proposto, foram observados resultados para diferentes inicializações. Estes resultados não apresentam variação considerável e portanto não afetam a qualidade do método. A diferença que ocorre

na inicialização se deve a etapa de quantização vetorial. Nesta etapa os *clusters* auxiliares podem ser dispostos em locais diferentes, gerando limiares diferentes para se chegar ao mesmo resultado. Em raras situações há a ocorrência de mínimos locais na etapa de quantização vetorial, que faz com que alguns *clusters* auxiliares fiquem em uma posição que compromete a classificação do conjunto. Este é um problema da etapa de quantização vetorial e pode ser resolvido utilizando alguma técnica para evitar os mínimos locais. Não foi objetivo deste trabalho aprofundar-se em técnicas mais robustas de quantização vetorial, visto que estas situações são raras.

4.4 Comparação com outros métodos

Nesta seção serão feitos comentários a respeito do desempenho do método proposto neste trabalho quando comparado com outros métodos. Basicamente a comparação deve ser feita para o método de *clustering* e para o método de modelamento estatístico. Cada um deles corresponde a uma contribuição deste trabalho.

4.4.1 Método de *clustering*

A comparação entre resultados de *clustering* pode ser feita observando-se a matriz de confusão gerada para cada método. Idealmente, a matriz deve conter dispersão nula por linha, ou seja, em cada linha, apenas uma coluna deve conter valores e estes devem corresponder ao número de pontos presentes na classe correspondente. Esta forma de comparação nos mostra o erro de classificação cometido pelo método ao associar um ponto a uma classe incorretamente. Um algoritmo básico de *clustering* que realiza o agrupamento de pontos em classes é o *k-means* que corresponde a um algoritmo de quantização vetorial (visto no capítulo 2). Trata-se de um algoritmo simples que só funciona bem para classes que sejam linearmente separáveis umas das outras. A Figura 4.42 mostra o que ocorre com a classificação de um conjunto não-linearmente separável quando submetido ao algoritmo *k-means*.

A matriz de confusão gerada pelo *k-means* é dada por

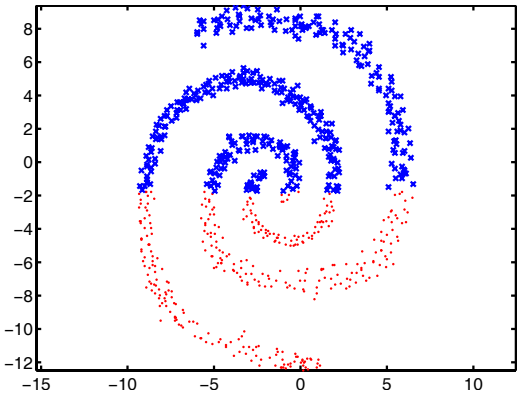


Figura 4.42: Separação de duas classes utilizando o *k-means*

$$\boldsymbol{C} = \begin{bmatrix} 302 & 470 \\ 450 & 322 \end{bmatrix}. \tag{4.14}$$

Podemos perceber que a classificação ocorre com um erro muito grande. Este erro irá ocorrer em qualquer algoritmo que realize a classificação utilizando quantização vetorial. Além disto, o número de classes deve sempre ser informado nestes tipos de algoritmos.

Em relação a outros algoritmos mais avançados de *clustering*, por não dispormos de um resultado numérico para comparação podemos comparar em termos de utilização e desempenho. Em algoritmos como o apresentado por Tyree e Long [85] que se também baseiam em ligações de *clusters* auxiliares a métrica utilizada para ligação é a distância Euclidiana. Isto torna o algoritmo dependente da escala e torna a escolha do limiar adequado muito difícil. O algoritmo apresentado neste trabalho, por apresentar métricas invariantes, possibilita a escolha do limiar muitas vezes automaticamente, sendo o número de classes estabelecido pelo próprio algoritmo. Gokcay [24] utilizam um algoritmo que realizada o agrupamento de um conjunto de dados utilizando métricas invariantes para agrupar os pontos do conjunto. Neste caso, o custo computacional é grande quando o tamanho do conjunto aumenta. Cada ponto é considerado um centro para a estimativa utilizando janelas de Parzen da distribuição de probabilidade. Comparando com o método proposto neste trabalho, o agrupamento também é feito utilizando métricas invariantes, porém a quantização vetorial inicial ameniza bastante o problema do custo computacional e da maldição

da dimensionalidade na estimativa das probabilidades (caso que ocorre com a estimativa utilizando janelas de Parzen).

4.4.2 Modelamento estatístico

O método proposto neste trabalho para gerar o modelo estatístico de um conjunto baseia-se em realizar a quantização vetorial deste e modela o mesmo como uma mistura de gaussianas. Existem muitas técnicas de modelamento presentes na literatura. Para efeito de comparação, serão analisados os métodos clássicos que são as janelas de Parzen e o algoritmo EM.

O método utilizando janelas de Parzen apresenta uma grande desvantagem no cálculo da probabilidade de um ponto utilizando o modelo gerado. O cálculo da probabilidade de um simples ponto, necessita da contabilização da contribuição de **todos** os pontos do conjunto, como mostrado no capítulo 2. Embora simples, o método de Parzen é ineficiente do ponto de vista computacional para modelar conjuntos grandes. De maneira contrária, para se ter um bom modelo é preciso de muitos pontos para evitar a maldição da dimensionalidade. Comparando com o algoritmo proposto, o custo inicial da quantização vetorial e do cálculo das matrizes de covariância compensam quando se tem muitos pontos em um conjunto a ser modelado. O cálculo da probabilidade fica resumido à contabilização de cada gaussiana da mistura, ao invés de cada ponto do conjunto.

Outro método bastante comum para modelamento estatístico de conjuntos de dados consiste na utilização do algoritmo EM. O algoritmo EM, assim como no algoritmo proposto, modela um conjunto utilizando mistura de gaussianas. A sua grande desvantagem é o custo computacional para se chegar no modelo. Na Figura 3.3 do capítulo 3 podemos observar uma comparação entre o modelo estabelecido com o algoritmo EM e o algoritmo proposto neste trabalho. O erro cometido no modelo utilizando a quantização vetorial (este trabalho) é compensado pelo custo computacional e simplicidade do algoritmo.

4.5 Sumário

Neste capítulo foram apresentados testes e resultados utilizando o método de *clustering* e método de modelamento propostos neste trabalho. Os testes realizados tiveram o objetivo de mostrar o comportamento de cada etapa do algoritmo desenvolvido diante de situações particulares. Foram apresentados diversos conjuntos de dados diferentes com o objetivo de testar o algoritmo de *clustering* e modelamento diante de casos como entrelaçamento das classes, não-linearidade da separação, entre outros. Conjuntos clássicos com dados reais foram testados. Os resultados foram apresentados em forma de gráficos de evolução do número de classes em função do limiar escolhido. Analisando estes gráficos foi possível a implementação de uma heurística de escolha automática do limiar, baseado na estabilidade do número de classes. Com esta escolha, o número de classes é automaticamente encontrado pelo método. Foi mostrado que para a divergência de Rényi, o histograma de distâncias possui um vale que coincide com a região de estabilização do número de classes.

Os modelos de probabilidade gerados são bem complexos. Isto mostra a capacidade do método de absorver a estatística presente no conjunto e representá-la em forma de um modelo $p_i(\mathbf{x})$. Este modelo pode ser utilizado para calcular a probabilidade de pontos que não pertencem ao conjunto bem como classificá-los.

Capítulo 5

Conclusões e Perspectivas

A literatura da área de *clustering* é vasta em termos de aplicações. Muitos métodos com diferentes abordagens são propostos. Algumas técnicas são baseadas em teorias clássicas, porém algumas abordagens sugerem a utilização conjunta de técnicas distintas para realizar a tarefa de *clustering*. Esta é a abordagem utilizada neste trabalho. No capítulo 2 foram expostas técnicas e ferramentas utilizadas em problemas mais gerais. No capítulo 3 estas técnicas foram utilizadas em conjunto para desenvolver um algoritmo de *clustering* capaz de agrupar conjuntos complexos de dados. Ainda no capítulo 3 foi proposta uma técnica para modelamento estatísticos dos dados agrupados.

A técnica baseia-se na quantização vetorial do conjunto e na posterior ligação dos *clusters* formados. Esta técnica requer a utilização de alguns parâmetros. Os primeiros parâmetros utilizados são os parâmetros da etapa de quantização que não se mostraram críticos, sendo utilizados os mesmos parâmetros para **todos** os conjuntos e aplicações deste trabalho. O algoritmo se mostrou mais sensível aos parâmetros N_a e d_t. O número de *clusters* auxiliares N_a foi sempre estabelecido com base no número de pontos do conjunto. Todos os exemplos aqui apresentados utilizando-se entre 1% e 5% do número total de pontos. Para o valor do limiar d_t foram estabelecidas heurísticas baseadas no histograma de divergências e no gráfico do número de classes em função do próprio limiar. Como mostrado, esta gráfico além de guiar a escolha do limiar, pode ser utilizado para inferir automaticamente um limiar ótimo, que, em muitos casos, forneceu o número correto de classes do

conjunto. Entretanto, o método não se propõe a ser totalmente automático, e a escolha do limiar ou do número de classes é um parâmetro opcional para o usuário.

As divergências utilizadas para medir a dissimilaridade entre os *clusters* auxiliares foram apresentadas, e foram provadas as invariâncias à transformações lineares em todas elas. As métricas utilizadas foram a distância de Mahalanobis, distância de Bhattacharyya, divergência de Kullback-Leibler e uma métrica proposta neste trabalho denominada aqui de divergência de Rényi. Uma das principais contribuições do trabalho foi a utilização de métricas não-Euclidianas e baseadas em teoria da informação para realizar a ligação entre os *clusters* auxiliares. Em todas elas, pode-se perceber a ponderação das distâncias entre as médias pelo inverso das matrizes de covariância de cada conjunto. Este foi o fato que possibilitou a invariância das métricas.

Em particular uma das métricas constituiu outra contribuição importante deste trabalho sendo esta desenvolvida e proposta neste trabalho. A divergência proposta aqui baseia-se na diferença de entropia entre duas distribuições de probabilidade. Testes numéricos utilizando foram realizados utilizando-se integração de Monte Carlo e modelos utilizando janelas de Parzen. O ponto fundamental no desenvolvimento da nova divergência foi a utilização da entropia de Rényi. Com esta o cálculo da diferença de entropias pode ser feito baseando-se no conceito de negentropia e calculado analiticamente. A obtenção da expressão analítica para o cálculo da divergência tornou o processo rápido e compatível com as outras divergências apresentadas. A prova quanto a invariância à transformações lineares da nova divergência também foi apresentada. O conceito de negentropia foi utilizado para que fosse possível medir a divergências entre duas gaussianas e seu modelo equivalente de apenas uma gaussiana. Neste modelo são contabilizadas as probabilidades *a priori* de cada conjunto a ser comparado. Isto constituiu um diferencial muito importante na nova métrica proposta.

Foram realizados testes para analisar o comportamento do método em situações diferentes. Em geral, as divergências de Kullback-Leibler, Mahalanobis e Bhattacharyya se comportam de maneira semelhante. A divergência proposta por este trabalho, a divergência de Rényi, apresenta propriedades interessantes quanto

a separabilidade dos *clusters* auxiliares. Os testes possibilitaram uma comparação entre o comportamento das métricas nas situações apresentadas.

Os resultados mostrados ainda no capítulo 4 mostram que o método desenvolvido consegue classificar dados distribuídos de maneira complexa no espaço dos atributos. A utilização de divergências invariantes à transformações lineares como métrica para ligação dos *clusters* auxiliares conseguiu recuperar a forma espacial de cada classe, agrupando corretamente os dados. O modelo gerado por misturas de gaussianas é facilmente calculado e constitui um modelo simples, porém representativo do conjunto analisado. Os resultados para conjuntos com a presença de ruído mostram a capacidade de separação do método proposto, separando os dados das classes e gerando uma outra classe que contem apenas o ruído presente nos dados. Esta separação só é possível, como mostrado, com a utilização da nova métrica proposta, a divergência de Rényi. Nos demais conjuntos artificialmente gerados, o erro cometido foi de menos de 1% em todos os casos.

Foram analisados dois casos de conjuntos de dados reais: o conjunto (notas) e (iris). Para o conjunto (notas) o erro cometido foi de apenas 0.5% em uma das classes. Já o conjunto (iris) apresentou um erro de mais de 50% em uma das classes, mesmo acertando 100% nas demais. Isto se deve a existência de duas classes muito próximas. Isto mostra que o método proposto deve ser aplicado quando existam classes separadas espacialmente, mesmo que esta separação seja complexa e entrelaçada.

Ao final, foram selecionadas três aplicações que utilizaram o método proposto neste trabalho. A primeira consistiu de uma aplicação na área de materiais, visando classificar regiões de amostras de minerais. A segunda consiste de um novo método de segmentação automática de imagens baseado em texturas. A ultima consiste em uma aplicação do método em uma área aparentemente não-relacionada à *clustering* que é a vetorização e reconstrução 3D de imagens e objetos. Estas aplicações mostram o potencial o método para atuar em áreas bem distintas. Na análise de materiais, o método se mostrou adequado para atuar como uma ferramenta de auxilio, sendo o especialista responsável pela rotulação final dos elementos. A técnica de segmentação apresentada se mostrou robusta e, em alguns casos,

inferindo automaticamente sobre o número de texturas presentes. A aplicação de vetorização e reconstrução utilizou o conceito de ligação dos *clusters* auxiliares para gerar vetores e arestas em aplicações de computação gráfica. Não se encontrou na literatura, aplicações de algoritmos de *clusters* sendo utilizados como ferramentas de vetorização e reconstrução, o que torna esta mais uma contribuição do trabalho.

5.1 Perspectivas

O método de *clustering* apresentado e desenvolvido neste trabalho pode ser utilizado em qualquer problema de agrupamento e modelamento de dados se estes estiverem devidamente separados. Muitas áreas do conhecimento e tecnologia necessitam de uma ferramenta de *clustering*, como mineração de dados, análise de dados geológicos, clima, mercado financeiro, etc...

As ferramentas utilizadas para o desenvolvimento deste trabalho são ferramentas clássicas de teoria da informação. Algumas abordagens inteligentes podem ser utilizadas para melhorar o método proposto. A etapa de quantização vetorial pode ser melhorada com a utilização de ferramentas de quantização vetorial mais eficientes. Heurísticas para encontrar o número de *clusters* auxiliares ou o limiar podem ser propostas baseadas na minimização de um funcional. A determinação de um funcional que reflita a qualidade das ligações constitui a maior contribuição que pode ser agregada ao trabalho. Algoritmos de minimização como algoritmos genéticos ou até mesmo métodos clássicos como método do gradiente poderiam ser utilizados para encontrar a matriz de ligação ideal, ou selecionar o melhor limiar.

Uma outra perspectiva para este trabalho seria a análise de entropias de Rényi de ordem mais elevada. Esta abordagem aumentaria a complexidade da expressão da divergência, porém poderia proporcionar uma medida com características de agrupamento mais adequadas. Nesta linha, uma outra possibilidade seria a utilização de entropias diferentes como a entropia de Tsallis. O cálculo da negentropia que se baseia nestas medidas de entropia poderia também ser alterado, utilizando-se aproximações para os momentos de alta ordem da distribuição, simplificando assim a expressão fina da divergência.

Por fim, poderia-se realizar um estudo para determinar heurísticas para a escolha do número de *clusters* auxiliares que depende-se apenas do conjunto de dados inicial. Técnicas baseadas nas estimativas das matrizes de covariância poderiam ser testadas. Testes estatísticos poderiam ser utilizados para se chegar em um número ótimo de *clusters* iniciais.

Apêndice A

Aplicações

A.1 Introdução

Neste apêndice serão descritas algumas aplicações do método proposto neste trabalho. Estas aplicações ilustram a utilidade do método e sua aplicabilidade. Como exemplo serão mostradas três aplicações reais em problemas de três áreas diferentes.

A.2 Análise de dados de microscopia eletrônica

A primeira das aplicações consiste em utilizar o método proposto neste trabalho para classificar dados de minerais provenientes de microscopia eletrônica [26]. Conforme descrito no capítulo 3, os dados a serem classificados são obtidos através de microscopia eletrônica de amostras de um tipo de mineral. Estes minerais são formados por átomos e moléculas que se agrupam em substâncias que compõem o material do mineral. Os dados da microscopia consistem das concentrações dos átomos analisados em diversos pontos do material. Com isto forma-se um conjunto de N pontos, cada um extraído de um local diferente da amostra de mineral. Cada ponto possui dimensão D correspondendo ao número de átomos analisados, sendo que cada dimensão corresponde à uma concentração de um dos átomos analisados. Cada classe corresponderá a uma possível substância do material analisado.

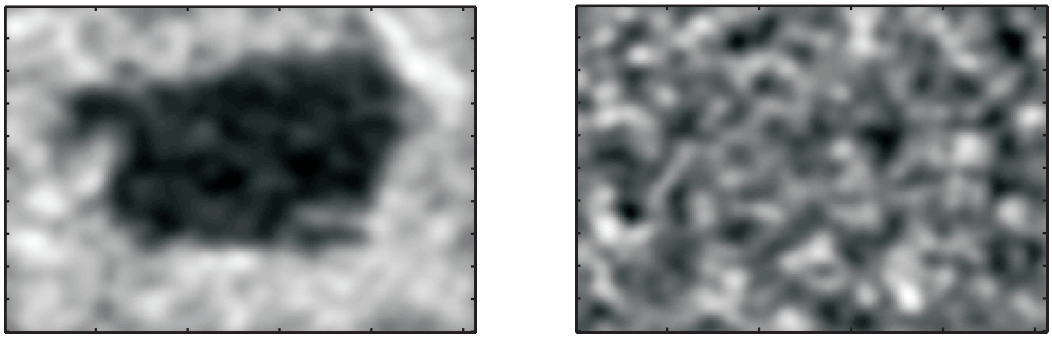
Os dados utilizados para esta aplicação foram extraídas de amostras de cimento

Tabela A.1: Tabela de dados das amostras utilizadas nesta aplicação

Conjunto	D	N_c
NE01	10	2
NE02	13	2
NE03	8	3
NE04	7	2

utilizadas em um trabalho recente na área de materiais [67]. Diversas amostras de diferentes tipos de cimento foram utilizados. Para cada amostra de material o método proposto neste trabalho foi aplicado e a classificação foi realizada.

Os dados originais são disponibilizados em forma de uma matriz onde cada elemento corresponde a um local no espaço onde foi realizada a medida. Para cada átomo temos uma matriz de concentrações. A Figura A.1 mostra 2 matrizes, uma correspondendo a concentração de alumínio e a outra de sódio em um dos materiais analisados.



(a) Concentração de Al (b) Concentração de Na

Figura A.1: Exemplo de imagens de concentrações de dois átomos de uma mesma amostra

As amostras foram nomeadas e organizadas como mostra a tabela A.1. A coluna D corresponde a dimensão dos dados, ou seja, quantos átomos foram analisados. Todos os exemplos rodaram com 20 *clusters* auxiliares e o número de classes foi escolhido pelo usuário. A coluna N_c mostra a quantidade de classes utilizadas. A divergência utilizada para todos os casos foi a divergência de Rényi. Os parâmetros para a etapa de quantização vetorial foram os mesmos para todas as situações analisadas neste trabalho.

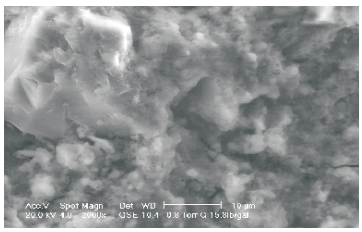
Uma análise inicial foi realizada em cada conjunto utilizando análise de

componentes principais (PCA). Verificou-se que em todos os conjuntos, existe uma dimensão que não possui variância significativa. Esta dimensão foi descartada e os dados utilizados no algoritmo foram resultado da descorrelação dos conjuntos originais descartando-se a dimensão de menor variância. Desta forma os dados efetivamente apresentados ao método possuíram dimensão $D - 1$.

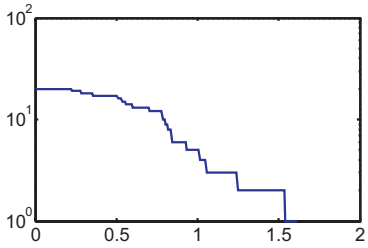
As Figuras A.2 a A.5 mostram os resultados obtidos. A imagens apresentadas no inicio de cada figura correspondem a uma imagem de espectro visível do material na área de coleta dos dados. O gráfico apresentado em cada resultado corresponde ao gráfico do número de classes em função do limiar. A segunda imagem apresentada corresponde a uma imagem montada a partir dos dados descorrelacionados utilizando-se como componentes RGB os 3 planos de maior variância. Por fim, a ultima imagem corresponde a imagem classificada. As duas primeiras imagens apresentadas em cada resultado (espectro visível e imagem das componentes de maior variância) não fazem parte do processo e servem apenas para ilustração e auxílio na escolha do limiar.

As classes segmentadas não são a princípio associadas a nenhum material ou substância. O objetivo do método é apenas separar as amostras em regiões que serão posteriormente rotuladas pelo especialista. Como podemos ver nos resultados, pequenos cristais que aparecem na imagem de espectro visível são separados porém sua composição deve ser analisada pelo especialista. O método proposto neste trabalho atua, neste caso, como uma ferramenta de auxilio à análise dos materiais.

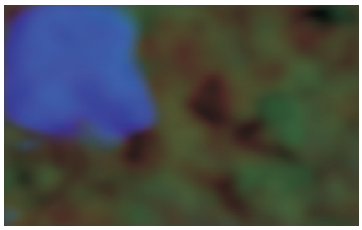
O número de classes neste caso foi escolhido pelo usuário que decidiu baseado nas informações do gráfico do número de classes e da imagem com as principais componentes. Em alguns casos, o número de classes mais estável não proporciona uma visualização das substâncias presentes e o usuário pode decidir visualizar a segmentação com mais ou menos classes.



(a) Imagem de espectro visível



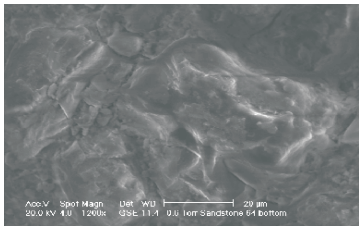
(b) Número de classes em função do limiar



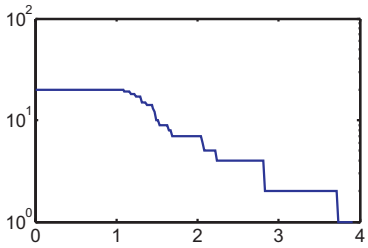
(c) Imagem dos planos de maior variância

(d) Imagem classificada

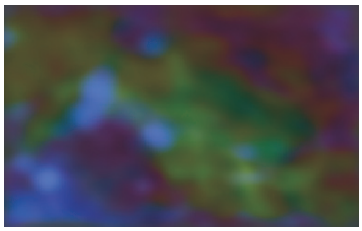
Figura A.2: Resultado para o conjunto NE01



(a) Imagem de espectro visível



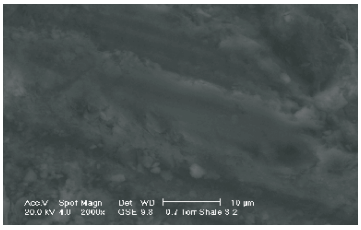
(b) Número de classes em função do limiar



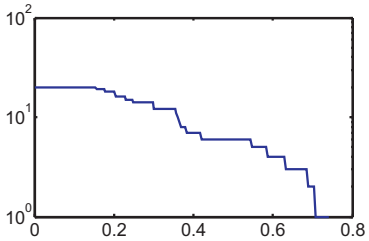
(c) Imagem dos planos de maior variância

(d) Imagem classificada

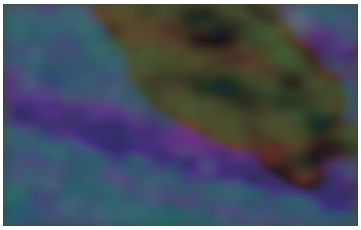
Figura A.3: Resultado para o conjunto NE02



(a) Imagem de espectro visível



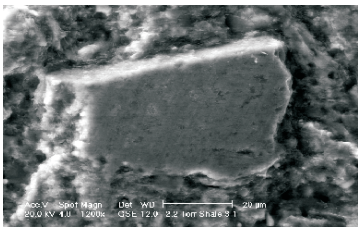
(b) Número de classes em função do limiar



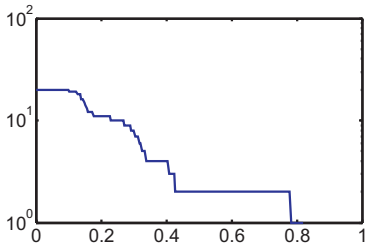
(c) Imagem dos planos de maior variância

(d) Imagem classificada

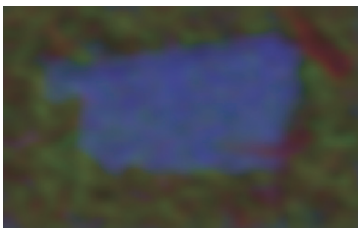
Figura A.4: Resultado para o conjunto NE03



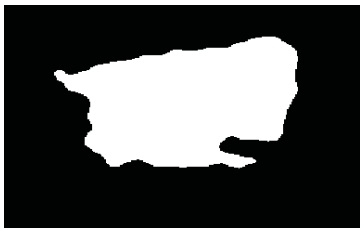
(a) Imagem de espectro visível



(b) Número de classes em função do limiar



(c) Imagem dos planos de maior variância



(d) Imagem classificada

Figura A.5: Resultado para o conjunto NE04

A.3 Segmentação de imagens

Nesta aplicação, o método foi utilizado para realizar a segmentação de imagens. Como já comentado no capítulo 3, o método de segmentação consiste em gerar um conjunto de dados baseado nos valores dos pixeis presentes na imagem e depois

agrupá-los em classes que representarão as texturas presentes na imagem.

O conjunto gerado consistiu de N pontos extraídos da imagem. Cada ponto consiste de um vetor de três dimensões correspondendo as médias dos valores R, G e B dos pixeis contidos em uma janela $n \times n$. A Figura A.6 ilustra como são extraídos os pontos em uma imagem.

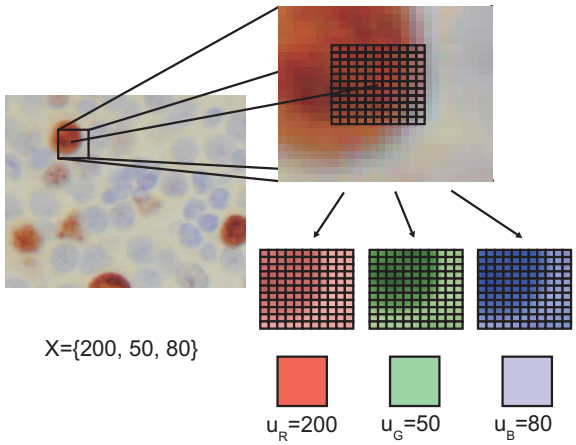
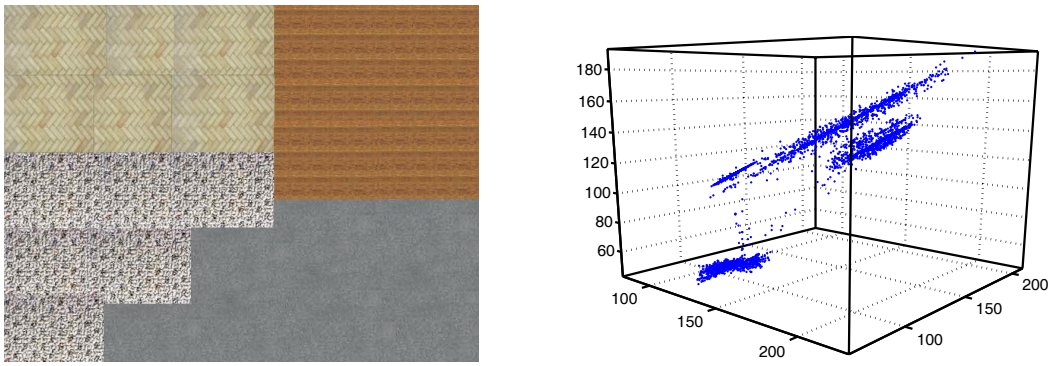


Figura A.6: Extração dos dados de uma imagem para formar o conjunto a ser agrupado.

Cada ponto do conjunto é formado como ilustra a figura. Um pixel é escolhido aleatoriamente na imagem e em torno do mesmo, uma janela de $n \times n$ pixeis é definida. O dado que fará parte do conjunto é formado pelas médias dos pixeis das componente R, G e B dentro da janela. Desta maneira, a primeira coordenada dos dados que formam o conjunto é a média da componente R, a segunda da componente G e a terceira da componente B. A Figura A.7 ilustra uma imagem e o conjunto formado a partir desta com $N = 4000$ e $n = 9$.

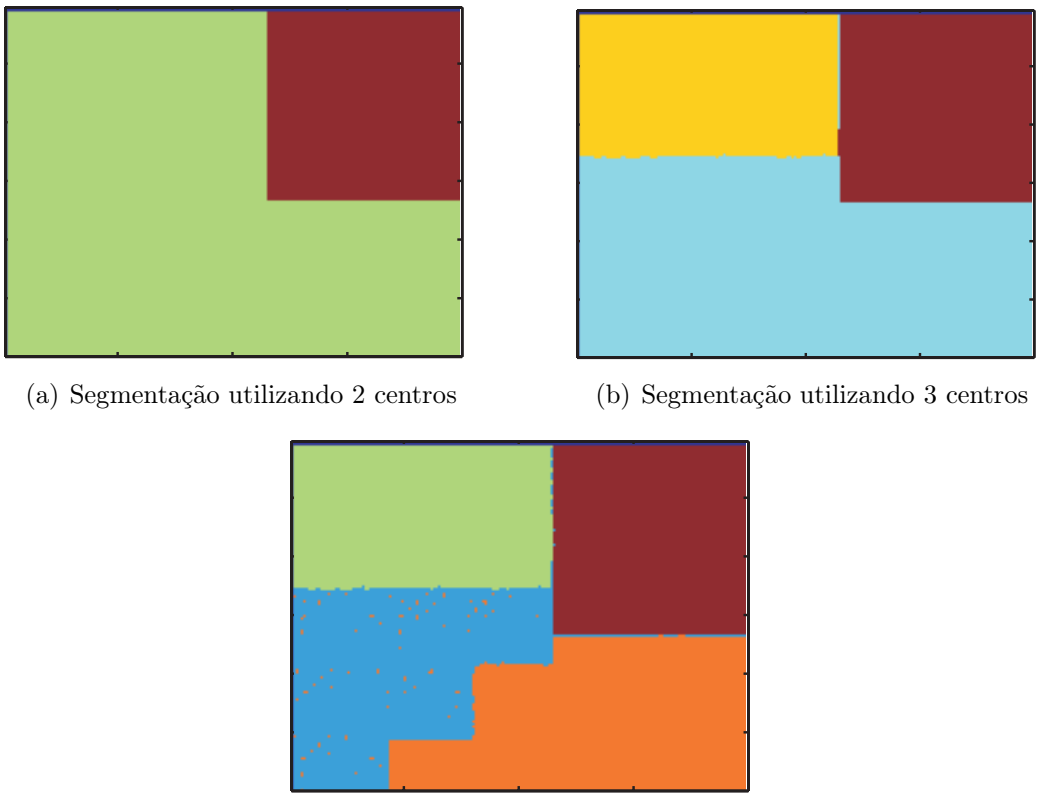
O resultado da segmentação utilizando o algoritmo para esta imagem é mostrado na Figura A.8. Foram utilizados $N_a = 20$ e os parâmetros de treinamento da rede neural competitiva foram os mesmos utilizados em todo o trabalho. A abordagem escolhida para agrupar os dados foi a abordagem escolhendo o número de classes. A divergência utilizada foi a divergência de Rényi, embora todas tenham sido testadas e tenham produzido um resultado muito parecido.

A imagem segmentada é gerada classificando-se cada pixel da mesma como pertencente a uma das classes geradas pelo método de *clustering*. Para cada pixel



(a) Imagem original (b) Imagem na forma de dados

Figura A.7: Exemplo de imagem e um conjunto gerado correspondente



(a) Segmentação utilizando 2 centros (b) Segmentação utilizando 3 centros

(c) Segmentação utilizando 4 centros

Figura A.8: Exemplo de segmentação

na imagem, o mesmo processo de colocação da janela e cálculo das médias R, G e B é realizado. O ponto resultante é então classificado e sua classe é atribuída ao pixel que lhe deu origem.

Métodos de *clustering* aplicados à segmentação de imagens não é uma abordagem nova. Existem diversas aplicações na literatura que utilizam algoritmos de *clustering*

para realizar a segmentação. A técnica utilizando a média das componentes RGB para geração de um conjunto de dados a ser agrupado foi desenvolvida por Martins et al. [57, 56]. Nesta técnica, o método de *clustering* utilizado baseava-se em quantização vetorial simples e nem sempre consegue separar bem todas as classes. A Figura A.9 mostra o resultado utilizando este método.

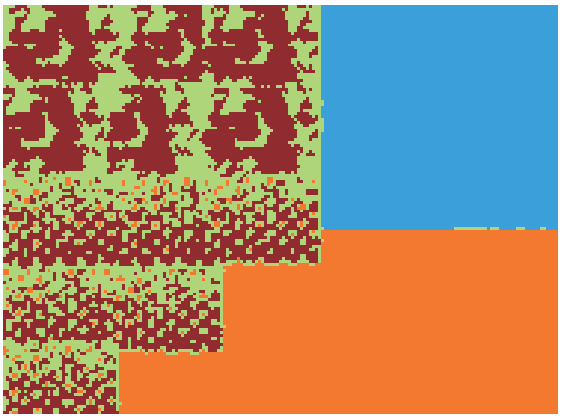


Figura A.9: Resultado da segmentação utilizando Redes Neurais competitivas simples.

Algumas outras imagens foram testadas e os resultados das segmentações são mostrados nas Figuras A.10 e A.11. Os resultados das segmentações são mostrados nas próprias imagens utilizando bordas coloridas. Como pode-se observar, as regiões com texturas diferentes são separadas separadas, segmentando a imagem em regiões contendo estas texturas. O método funciona bem para imagens com características bem diferentes, compreendendo de texturas artificiais à células e imagens aéreas.

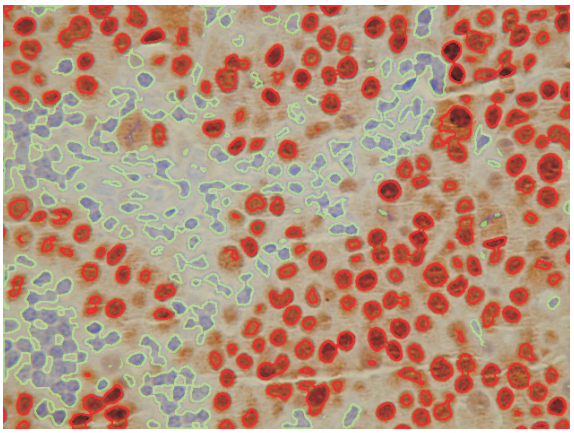


Figura A.10: Resultado da segmentação para imagem de células.

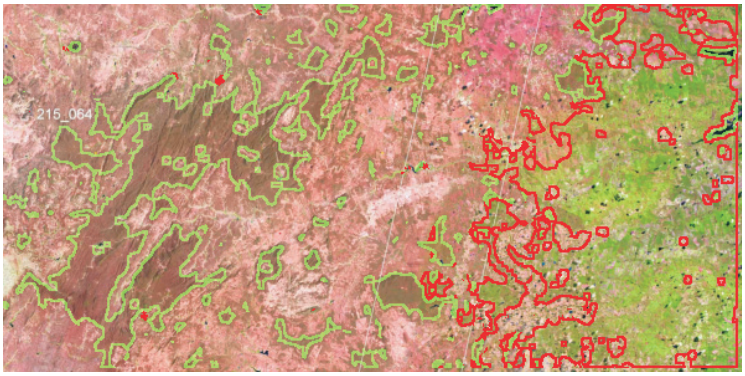


Figura A.11: Resultado da segmentação utilizando para uma imagem aérea

A.4 Vetorização e reconstrução 3D

A ultima aplicação a ser descrita neste apêndice consiste em realizar a vetorização de imagens bi-dimensionais e a reconstrução 3D a partir de pontos no espaço. Conforme já apresentado na seção 3.10.3 do capítulo 3, estes problemas não utilizam normalmente técnicas de *clustering* para serem resolvidos. Entretanto, o conceito de ligação de *clusters* auxiliares utilizado neste trabalho, pode ser aproveitado para resolver problemas tanto de vetorização quanto de reconstrução 3D.

A.4.1 Vetorização de imagens

O processo de vetorização de imagens consiste em representar uma imagem utilizando vetores para definir os contornos dos objetos nela presentes. Normalmente uma imagem é descrita por uma matriz de pixeis. Para o caso tratado neste trabalho, consideramos uma imagem binária, ou seja, cada pixel só pode assumir valores 0 ou 1. A grande desvantagem da representação matricial está no fato de que esta, a ser ampliada, produz um efeito de "serrilhamento". Este efeito acontece porque ao se aumentar a imagem, oque se faz na realizada é representar um pixel por vários, formando um "pixel maior". A Figura A.12 ilustra esse efeito.

Alguns métodos podem ser aplicados à imagem durante a sua ampliação como interpolação de pixeis, porém, esta não é adequada para imagens binárias. Uma imagem vetorial possui vetores ao invés de pixeis para formar os contornos dos objetos. Por serem vetores, estes podem ser escalonados a vontade sem que ocorra o

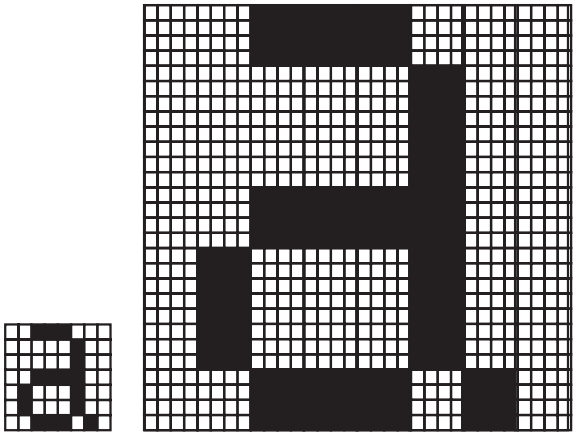


Figura A.12: Resultado da ampliação de uma imagem binária

efeito de serrilhamento. A Figura A.13 ilustra a ampliação de uma imagem vetorial.

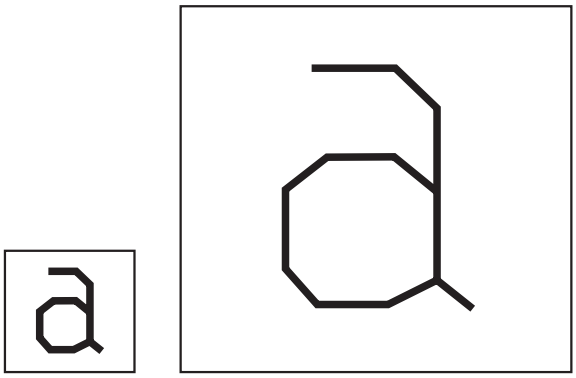


Figura A.13: Resultado da ampliação de uma imagem vetorial

Existem diversas técnicas clássicas de vetorização de imagens na literatura, algumas destas técnicas são utilizadas inclusive em softwares tradicionais de manipulação de gráficos como Corel Trace®.

A técnica de *clustering* proposta neste trabalho apresenta o conceito de ligação entre *clusters* auxiliares. Para utilizar este conceito em problemas de vetorização, a imagem vai ser considerada como um conjunto de dados bi-dimensional. Cada pixel com rótulo "1" será um ponto do conjunto. As coordenadas (x, y) do ponto serão os atributos de cada ponto (dimensão 1 será o valor de x e a dimensão 2 será o valor de y). Desta maneira, o algoritmo irá proceder a segmentação em N_a *clusters* iniciais que agruparão os pixeis da imagem em pequenos *clusters* como mostra a Figura A.14.

Após este passo, um limiar é escolhido e os *clusters* auxiliares são ligados. Cada

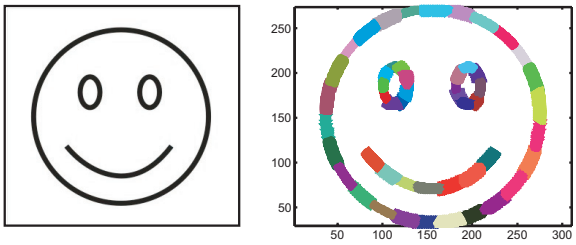


Figura A.14: Etapa de agrupamento dos pixeis da imagem em *clusters* auxiliares

ligação corresponderá a um vetor na imagem vetorizada. A Figura A.15 mostra o resultado final.

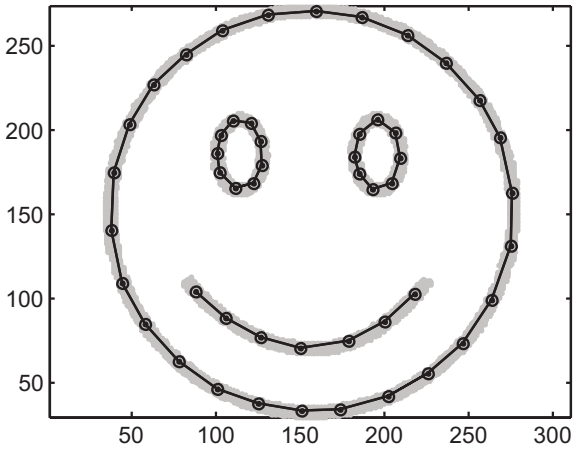
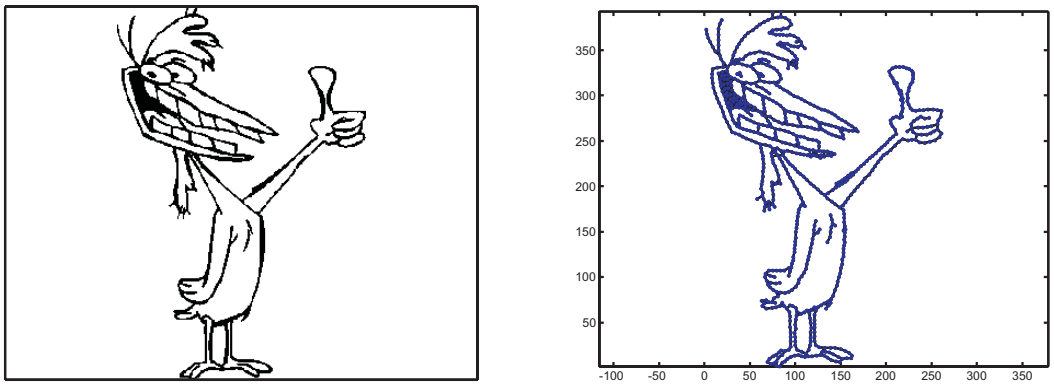


Figura A.15: *Clusters* auxiliares ligados

Neste tipo de problema, agrupamento em classes não é utilizado. O objetivo final são apenas as ligações entre os *clusters* auxiliares que formarão os vetores. O limiar escolhido terá efeito de aumentar ou diminuir a quantidade de ligações e promover a conexão entre os centros dos *clusters* auxiliares. O número de *clusters* auxiliares controlará o nível de detalhamento da figura, poucos *clusters* irão gerar imagens com poucos vetores, enquanto muitos *clusters* darão origem a imagens com superfícies mais detalhadas. Obviamente este número não pode ser muito grande sob pena de dificultar a etapa de quantização vetorial. O número de *clusters* auxiliares e o limiar são parâmetros que regulam o resultado fornecido pelo método, sendo estes a critério do usuário.

A seguir serão mostrados alguns resultados de vetorização utilizando o método proposto neste trabalho. As Figuras A.16 e A.17 mostram alguns dos resultados.

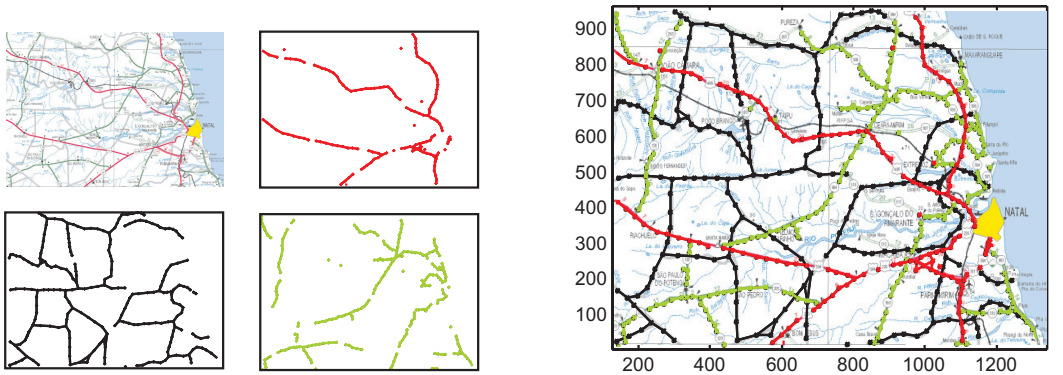
No primeiro exemplo (Figura A.16) o método é testado em uma imagem



(a) Imagem original (b) Imagem vetorizada

Figura A.16: Resultado da vetorização

monocromática simples consistindo de um desenho. O resultado ao lado é mostra a imagem vetorizada (linhas escuras) e os centros encontrados (pontos azuis). Nesta imagem, o desenho é representado pelo vetores, e não mais pelos pontos acesos ou não de uma matriz de pixeis.



(a) mapa original e as versões vetorizadas das rodovias (b) Imagem do mapa com as rodovias já vetorizadas

Figura A.17: Resultado da vetorização

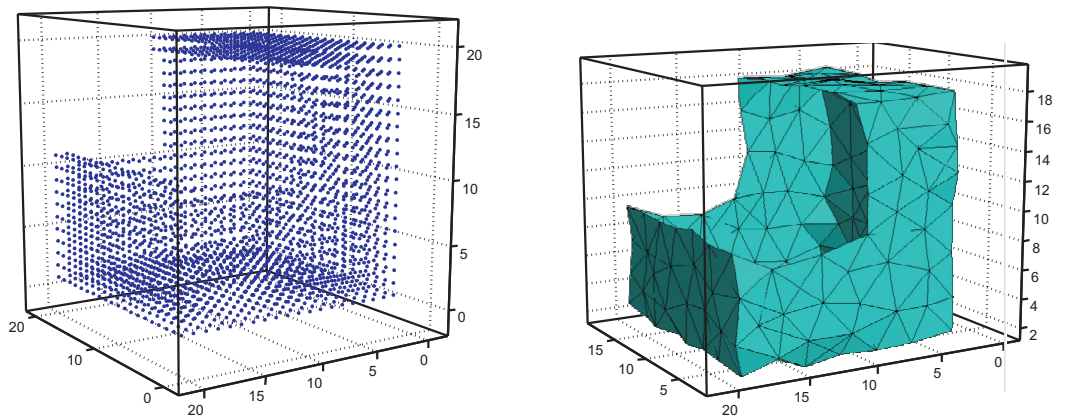
No segunda exemplo (Figura A.17) , utilizou-se uma imagem de mapa colorida. Apenas as regiões cinza, verde e vermelhas (correspondentes às rodovias no mapa) foram utilizadas como entrada para o método. Em cada um uma versão vetorizada foi encontrada. Desta maneira, o mapa pode ser ampliado sem que ocorra serrilhamento nas rodovias processadas. Isto é bastante útil em geo-processamento onde dados de aparelhos de GPS são utilizados para descrever rotas ou caminhos.

A divergência utilizada para estes exemplos foi a divergência de Mahalanobis, porém as divergências de Bhattacharyya e Kullback-Leibler produzem os mesmos

resultados. A divergência de Rényi não é muito adequada para esta aplicação. O efeito da quantidade de pontos de uma região da imagem implica na faz com que as probabilidades *a priori* mudem, gerando uma quantidade de ligação não uniforme em áreas com muitos pontos formando por exemplo uma linha mais grossa.

A.4.2 Reconstrução 3D

Conforme mencionado no capítulo 3 seção 3.10.4, o objetivo da reconstrução é representar a superfície de um objeto através de primitivas geométricas. Os dados utilizados para realizar esta tarefa são um conjunto de pontos na superfície do objeto. Esta representação auxilia na visualização do objeto e fornece um meio de representá-lo computacionalmente. A Figura A.18 mostra um objeto representado com pontos e com faces triangulares. Podemos perceber que a representação com faces, produz uma visualização melhor, além de dispor um modelo computacional mais adequado (por exemplo para simulações de propriedades mecânicas, etc...).



(a) Objeto representado com pontos no espaço

(b) Objeto representado com faces triangulares

Figura A.18: Representações de objetos utilizando pontos e faces

A tarefa de reconstruir tri-dimensionalmente um objeto a partir apenas de pontos de sua superfície não é trivial. A reconstrução pode utilizar-se somente de alguns pontos da superfície produzindo um objeto aproximado. Estes pontos escolhidos serão vértices dos polígonos (em geral triângulos) que comporão a superfície do objeto. O primeiro problema a ser enfrentado é a escolha destes pontos. Em seguida temos o problema da ausência de informação sobre as arestas que ligam cada ponto.

Por último a informação sobre que arestas formam cada triângulo (ou polígono).

Podemos utilizar novamente o conceito de quantização vetorial e ligação do algoritmo de *clustering* proposto neste trabalho. Normalmente os problemas de reconstrução 3D não utilizam métodos de *clustering* como ferramenta. Entretanto, o conceito de ligação entre os *clusters* auxiliares será novamente útil para definir um vetor, que desta vez representará uma aresta de uma face do objeto.

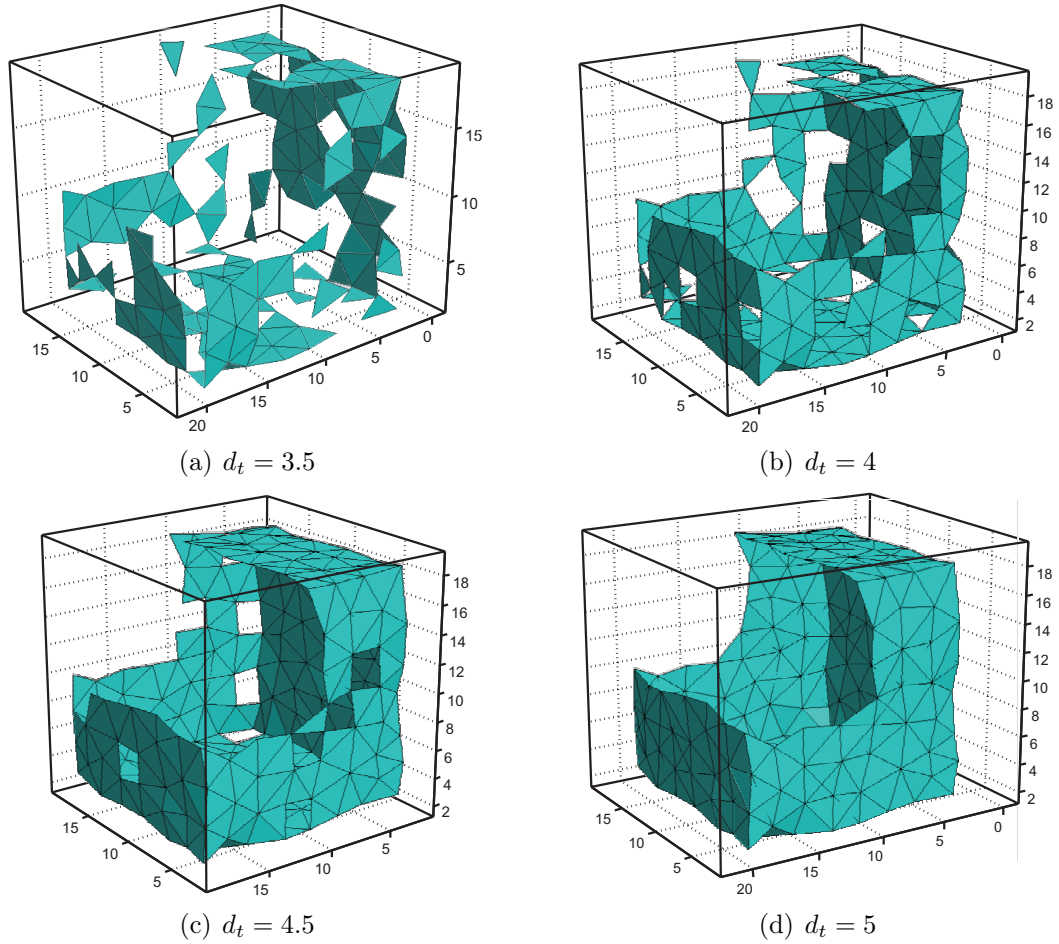
O conjunto de dados utilizado para o método será gerado de maneira semelhante ao processo de vetorização de imagens. Cada pondo da superfície do objeto será um ponto do conjunto que desta vez será tri-dimensional. Cada dimensão será composta por uma coordenada espacial do ponto (x, y, z).

Inicialmente o método realiza a quantização vetorial do conjunto selecionando apenas N_a dos N pontos iniciais. Esta etapa já consiste na primeira etapa de reconstrução que é selecionar um número menor de pontos da superfície para representar a superfície do objeto. Em seguida, define-se um limiar e realiza as ligações entre os centros dos *clusters* auxiliares. Estas ligações serão as possíveis arestas das faces. Esta corresponde a segunda etapa do processo de reconstrução. A última consiste em agrupar arestas de e 3 a 3 de maneira que estas formem os triângulos. Esta etapa não é realizada pelo método de *clustering*, porém é bastante simples de ser executada.

Neste tipo de aplicação, o limiar define também o número de ligações existentes. Este parâmetro pode fazer com que alguns centros não sejam ligados e apareçam espaços abertos na superfície. A Figura A.19 ilustra este efeito.

Como na vetorização de imagens, a divergência de Rényi não é muito apropriada pois proporciona regiões densas de ligações, fazendo com que o processo de seleção de arestas para formação das faces gere faces desnecessárias. A Figura A.20 ilustra o resultado de uma reconstrução com diferentes quantidade de *clusters* auxiliares[1].

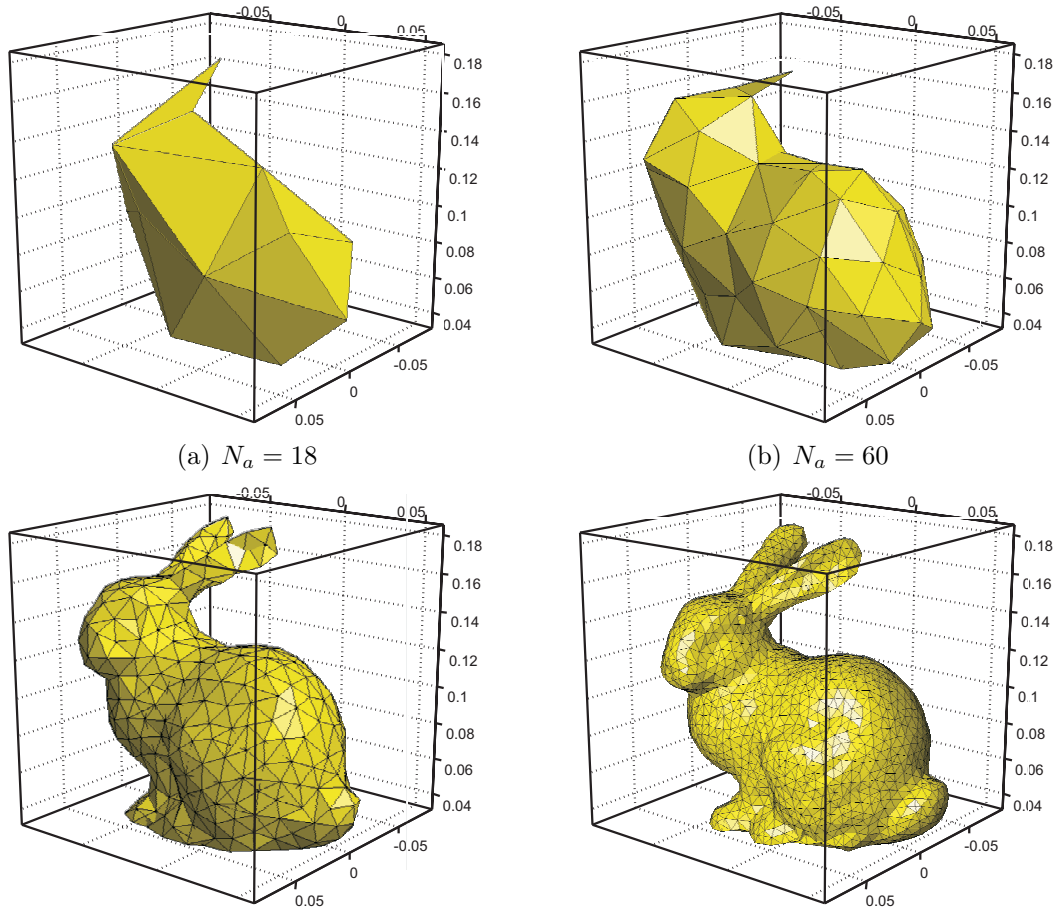
[1]Agradecimentos ao grupo "*The Stanford 3D Scanning Repository*" pelo modelo cedido



(a) $d_t = 3.5$ (b) $d_t = 4$

(c) $d_t = 4.5$ (d) $d_t = 5$

Figura A.19: Efeito de diferentes limiares na reconstrução de um objeto



(a) $N_a = 18$ (b) $N_a = 60$

(c) $N_a = 800$ (d) $N_a = 1800$

Figura A.20: Reconstrução dos dados “coelho de Stanford”

Apêndice B

Deduções

B.1 Introdução

Este apêndice tem como finalidade mostrar as deduções de algumas expressões utilizadas no trabalho que levam a resultados já conhecidos na literatura. As deduções são realizadas apenas para efeito de ilustração e não são essenciais ao trabalho.

B.2 Integral de uma gaussiana multivariada

Nesta seção será desenvolvida a solução para a integral de uma gaussiana multivariada. Esta solução é utilizada para o desenvolvimento da divergência proposta no capítulo 3. Após esta demonstração, será mostrada a solução para a integral gaussiana uni-dimensional, necessária durante a solução para a versão multi-dimensional, que o objetivo desta seção.

B.2.0.1 Gaussiana Multivariada

Para resolver a integral

$$I = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\mathbf{y}^t \boldsymbol{\Sigma}^{-1} \mathbf{y}} \mathrm{d}\mathbf{y}$$

começamos com uma substituição simples de variáveis. Seja $\mathbf{K}$ uma matriz definida como sendo

$$\mathbf{K} = \mathbf{\Sigma}^{-1}.$$

Como $\mathbf{K}$ é uma matriz real e simétrica, esta pode ser decomposta como

$$\mathbf{K} = \mathbf{T}^{-1}\mathbf{\Lambda}\mathbf{T}$$

onde $\mathbf{T}$ é a matriz dos autovetores de $\mathbf{K}$ e $\mathbf{\Lambda}$ uma matriz diagonal com os auto valores de $\mathbf{K}$. Desta forma a integral torna-se

$$I = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\mathbf{y}^t\mathbf{T}^{-1}\mathbf{\Lambda}\mathbf{T}\mathbf{y}} \mathrm{d}\mathbf{y}.$$

Seja $\mathbf{z}$ definido como

$$\mathbf{z} = \mathbf{T}\mathbf{y}.$$

Calculando d$\mathbf{z}$ obtemos [44]

$$\mathrm{d}\mathbf{z} = |\mathbf{T}|\, \mathrm{d}\mathbf{y}$$

onde $|\mathbf{T}|$ é o determinante de $\mathbf{T}$. Desta forma temos que

$$I = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\mathbf{z}\mathbf{\Lambda}\mathbf{z}} |\mathbf{J}|^{-1}\, \mathrm{d}\mathbf{z}.$$

Como $\mathbf{T}$ é formada pelos autovetores de $\mathbf{T}$, temos que

$$|\mathbf{J}| = 1$$

o que resulta em

$$I = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\mathbf{z}\mathbf{\Lambda}\mathbf{z}} \mathrm{d}\mathbf{z}.$$

Como dito anteriormente, $\mathbf{\Lambda}$ é uma matriz da forma

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$$

Desta maneira, o produto $\mathbf{z}\,\mathbf{\Lambda}\,\mathbf{z}$ pode ser escrito como um somatório. Como o somatório está no expoente, a integral então pode ser expressa como um produtório de integrais unidimensionais como mostrado abaixo

$$\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\mathbf{z}\mathbf{\Lambda}\mathbf{z}}\,\mathrm{d}\mathbf{z} = \prod_{i=1}^{n} \int_{-\infty}^{\infty} e^{-\frac{1}{2}z_i^2\lambda_i}\,\mathrm{d}z.$$

Podemos mostrar que

$$\int_{-\infty}^{\infty} e^{-\frac{1}{2}z_i^2\lambda_i}\,\mathrm{d}z = \sqrt{\frac{2\pi}{\lambda_i}}$$

o que leva a

$$\prod_{i=1}^{n} \int_{-\infty}^{\infty} e^{-\frac{1}{2}z_i^2\lambda_i}\,\mathrm{d}z = \prod_{i=1}^{n} \sqrt{\frac{2\pi}{\lambda_i}}$$

$$I = \sqrt{\frac{(2\pi)^n}{|\mathbf{K}|}}$$

que resulta em

$$I = \sqrt{(2\,\pi)^n\ |\mathbf{\Sigma}|}$$

B.2.1 Integral uni-dimensional

Para resolver a integral

$$I = \int_{-\infty}^{\infty} e^{-\frac{x^2}{2\sigma^2}}\,\mathrm{dx}$$

utilizaremos o seguinte artifício: Como trata-se de um valor positivo qualquer, vamos escrever I como sendo $\sqrt{I^2}$ calculados nas seguintes integrais

$$\sqrt{\int_{-\infty}^{\infty} e^{-\frac{x^2}{2\sigma^2}}\,\mathrm{dx}\int_{-\infty}^{\infty} e^{-\frac{y^2}{2\sigma^2}}\,\mathrm{dy}}.$$

Como as funções são separáveis, podemos agrupar da seguinte maneira

$$I = \sqrt{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{y^2}{2\sigma^2}}\, e^{-\frac{x^2}{2\sigma^2}}\,\mathrm{dy}\mathrm{dx}}$$

que reescrevendo fica

$$I = \sqrt{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}(x^2+y^2)}\,\mathrm{dy}\mathrm{dx}}.$$

Agora procedemos a seguinte mudança de variáveis

$$x^2 + y^2 = r^2$$

e

$$\mathrm{dxdy} = \mathrm{r\,dr\,d}\theta.$$

Esta mudança corresponde ainda a integrar em todo o $\mathbb{R}^2$, porém em coordenadas polares. Este procedimento resulta na seguinte integral

$$\mathrm{I} = \sqrt{\int_{0}^{2\pi}\int_{0}^{\infty} r\, e^{-\frac{1}{2\sigma^2}r^2}\,\mathrm{dr}\mathrm{d}\theta}$$

. Deste modo, esta é facilmente calculada e resulta em

$$I = \sqrt{2\pi}\sigma$$

B.3 Segundo momento para uma misturas de duas gaussianas

No capítulo 3 mostramos que a matriz de covariância de uma mistura de gaussianas é dada por

$$\boldsymbol{\Sigma} = \sum_{i=1}^{N_g} P_i \left[\boldsymbol{\Sigma}_i + (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^t\right]. \tag{B.1}$$

Na ocasião, foi estabelecido que para duas gaussianas apenas, a matriz de covariância da mistura tornava-se

$$\boldsymbol{\Sigma} = P_1\boldsymbol{\Sigma}_1 + P_2\boldsymbol{\Sigma}_2 + P_1P_2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t. \tag{B.2}$$

Nesta seção será demonstrada esta proposição. A expressão para a matriz de covariância para duas misturas pode ser expressa limitando-se o somatório da equação B.1 a dois termos. O objetivo é mostrar que a expressão resultante pode ser expressa como na equação B.2 que é mais intuitiva.

A expressão para equação B.1 para duas gaussianas é escrita como

$$\boldsymbol{\Sigma} = P_1\left[\boldsymbol{\Sigma}_1 + (\boldsymbol{\mu}_1 - \boldsymbol{\mu})(\boldsymbol{\mu}_1 - \boldsymbol{\mu})^t\right] + P_2\left[\boldsymbol{\Sigma}_2 + (\boldsymbol{\mu}_2 - \boldsymbol{\mu})(\boldsymbol{\mu}_2 - \boldsymbol{\mu})^t\right] \tag{B.3}$$

onde $\boldsymbol{\mu}$ é a média da mistura. Expandindo-se o produto das médias obtemos

$$\boldsymbol{\Sigma} = P_1\left[\boldsymbol{\Sigma}_1 + \boldsymbol{\mu}_1\boldsymbol{\mu}_1^t - \boldsymbol{\mu}_1\boldsymbol{\mu}^t - \boldsymbol{\mu}\boldsymbol{\mu}_1^t + \boldsymbol{\mu}\boldsymbol{\mu}^t\right] + P_2\left[\boldsymbol{\Sigma}_2 + \boldsymbol{\mu}_2\boldsymbol{\mu}_2^t - \boldsymbol{\mu}_2\boldsymbol{\mu}^t - \boldsymbol{\mu}\boldsymbol{\mu}_2^t + \boldsymbol{\mu}\boldsymbol{\mu}^t\right]$$

$$\begin{aligned}\boldsymbol{\Sigma} = {} & P_1\boldsymbol{\Sigma}_1 + P_2\boldsymbol{\Sigma}_2 + P_1\boldsymbol{\mu}_1\boldsymbol{\mu}_1^t - P_1\boldsymbol{\mu}_1\boldsymbol{\mu}^t - P_1\boldsymbol{\mu}\boldsymbol{\mu}_1^t + P_1\boldsymbol{\mu}\boldsymbol{\mu}^t + P_2\boldsymbol{\mu}_2\boldsymbol{\mu}_2^t - P_2\boldsymbol{\mu}_2\boldsymbol{\mu}^t - \\ & - P_2\boldsymbol{\mu}\boldsymbol{\mu}_2^t + P_2\boldsymbol{\mu}\boldsymbol{\mu}^t\end{aligned} \tag{B.4}$$

Substituindo a média da mistura pela expressão mostrada no capítulo 3 obtemos

$$\boldsymbol{\Sigma} = P_1\boldsymbol{\Sigma}_1 + P_2\boldsymbol{\Sigma}_2 + P_1\boldsymbol{\mu}_1\boldsymbol{\mu}_1^t - P_1^2\boldsymbol{\mu}_1\boldsymbol{\mu}_1^t - P_1P_2\boldsymbol{\mu}_1\boldsymbol{\mu}_2^t - P_1^2\boldsymbol{\mu}_1\boldsymbol{\mu}_1^t - P_1P_2\boldsymbol{\mu}_2\boldsymbol{\mu}_1^t + \\ + P_1^3\boldsymbol{\mu}_1\boldsymbol{\mu}_1^t + P_1^2P_2\boldsymbol{\mu}_1\boldsymbol{\mu}_2^t + P_1^2P_2\boldsymbol{\mu}_2\boldsymbol{\mu}_1^t + P_1P_2^2\boldsymbol{\mu}_2\boldsymbol{\mu}_2^t + P_2\boldsymbol{\mu}_2\boldsymbol{\mu}_2^t - P_2P_1\boldsymbol{\mu}_2\boldsymbol{\mu}_1^t - \\ - P_2^2\boldsymbol{\mu}_2\boldsymbol{\mu}_2^t - P_2P_1\boldsymbol{\mu}_1\boldsymbol{\mu}_2^t - P_2^2\boldsymbol{\mu}_2\boldsymbol{\mu}_2^t + P_2P_1^2\boldsymbol{\mu}_1\boldsymbol{\mu}_1^t + P_1P_2^2\boldsymbol{\mu}_1\boldsymbol{\mu}_2^t + P_1P_2^2\boldsymbol{\mu}_2\boldsymbol{\mu}_1^t + P_2^3\boldsymbol{\mu}_2\boldsymbol{\mu}_2^t \tag{B.5}$$

Isolando os termos comuns obtemos

$$\boldsymbol{\Sigma} = P_1\boldsymbol{\Sigma}_1 + P_2\boldsymbol{\Sigma}_2 + (P_1 - P_1^2 - P_1^2 + P_1^3 + P_2P_1^2)\boldsymbol{\mu}_1\boldsymbol{\mu}_1^t + \\ + (P_1^2P_2 - P_1P_2 - P_2P_1 + P_1P_2^2)\boldsymbol{\mu}_1\boldsymbol{\mu}_2^t + (P_1^2P_2 - P_1P_2 - P_2P_1 + P_1P_2^2)\boldsymbol{\mu}_2\boldsymbol{\mu}_1^t + \\ + (P_1P_2^2 + P_2 - P_2^2 - P_2^2 + P_2^3)\boldsymbol{\mu}_2\boldsymbol{\mu}_2^t \tag{B.6}$$

Rearrumando os termos com as probabilidades *a priori*, chegamos a

$$\boldsymbol{\Sigma} = P_1\boldsymbol{\Sigma}_1 + P_2\boldsymbol{\Sigma}_2 + P_1(1 - P_1 - P_1 + P_1^2 + P_2P_1)\boldsymbol{\mu}_1\boldsymbol{\mu}_1^t - P_1P_2(1 - P_1 + 1 - P_2)\boldsymbol{\mu}_1\boldsymbol{\mu}_2^t - \\ - P_1P_2(-P_1 + 1 + 1 - P_2)\boldsymbol{\mu}_2\boldsymbol{\mu}_1^t + P_2(P_1P_2 + 1 - P_2 - P_2 + P_2^2)\boldsymbol{\mu}_2\boldsymbol{\mu}_2^t$$

$$\boldsymbol{\Sigma} = P_1\boldsymbol{\Sigma}_1 + P_2\boldsymbol{\Sigma}_2 + P_1(1 - P_1 - P_1(1 - P_1 - P_2))\boldsymbol{\mu}_1\boldsymbol{\mu}_1^t - P_1P_2(2 - P_1 - P_2)\boldsymbol{\mu}_1\boldsymbol{\mu}_2^t - \\ - P_1P_2(-P_1 + 2 - P_2)\boldsymbol{\mu}_2\boldsymbol{\mu}_1^t + P_2(P_2(P_1 - 1 + P_2) + 1 - P_2)\boldsymbol{\mu}_2\boldsymbol{\mu}_2^t \tag{B.7}$$

Como só há duas gaussianas, as probabilidades *a priori* devem satisfazer a $P_1 + P_2 = 1$. Portanto, simplificando, obtemos

$$\boldsymbol{\Sigma} = P_1\boldsymbol{\Sigma}_1 + P_2\boldsymbol{\Sigma}_2 + P_1P_2\boldsymbol{\mu}_1\boldsymbol{\mu}_1^t - P_1P_2\boldsymbol{\mu}_1\boldsymbol{\mu}_2^t - P_1P_2\boldsymbol{\mu}_2\boldsymbol{\mu}_1^t + P_2P_1\boldsymbol{\mu}_2\boldsymbol{\mu}_2^t, \tag{B.8}$$

que fatorando resulta em

$$\boldsymbol{\Sigma} = P_1\boldsymbol{\Sigma}_1 + P_2\boldsymbol{\Sigma}_2 + P_1P_2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^t. \tag{B.9}$$

B.4 Entropia de Rényi para uma VA gaussiana

Nesta seção será demonstrado o cálculo da entropia de Rényi para uma VA gaussiana com média $\boldsymbol{\mu}$ e matriz de covariância $\boldsymbol{\Sigma}$.

A entropia de Rényi de uma VA X com densidade de probabilidade $p(\mathbf{x})$ é dada pela equação a seguir

$$H_\alpha = \frac{1}{1-\alpha} \log \left(\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} p(\boldsymbol{x})^\alpha \mathrm{d}\boldsymbol{x} \right) \tag{B.10}$$

Para a VA gaussiana em questão a expressão torna-se

$$H_\alpha = \frac{1}{1-\alpha} \log \left(\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left(k \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})} \right)^\alpha \mathrm{d}\boldsymbol{x} \right) \tag{B.11}$$

onde k é a constante de normalização da distribuição e vale

$$k = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}}. \tag{B.12}$$

Realizando uma substituição de variáveis simples como

$$\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{\mu} \tag{B.13}$$

obtemos o seguinte

$$H_\alpha = \frac{1}{1-\alpha} \log \left(k^\alpha \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \mathrm{e}^{-\frac{\alpha}{2} \boldsymbol{y}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{y}} \mathrm{d}\boldsymbol{y} \right). \tag{B.14}$$

Fazendo

$$\boldsymbol{\Sigma} = \alpha \boldsymbol{\Sigma}_q, \tag{B.15}$$

e substituindo, obtemos

$$\begin{aligned} H_\alpha &= \frac{1}{1-\alpha}\log\left(k^\alpha \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \mathrm{e}^{-\frac{\alpha}{2}\boldsymbol{y}^t(\alpha\boldsymbol{\Sigma}_q)^{-1}\boldsymbol{y}}\mathrm{d}\boldsymbol{y}\right) \\ H_\alpha &= \frac{1}{1-\alpha}\log\left(k^\alpha \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} \mathrm{e}^{-\frac{1}{2}\boldsymbol{y}^t\boldsymbol{\Sigma}_q^{-1}\boldsymbol{y}}\mathrm{d}\boldsymbol{y}\right) \end{aligned}. \tag{B.16}$$

Resolvendo a integral (como mostrado na seção B.2) obtemos

$$H_\alpha = \frac{1}{1-\alpha}\log\left(k^\alpha\sqrt{(2\pi)^D\left|\boldsymbol{\Sigma}_q\right|}\right) \tag{B.17}$$

$$H_\alpha = \frac{1}{1-\alpha}\log\left(k^\alpha\sqrt{(2\pi)^D\alpha^{-D}\left|\boldsymbol{\Sigma}\right|}\right) \tag{B.18}$$

Substituindo o valor de k na expressão, temos

$$H_\alpha = \frac{1}{1-\alpha}\log\left(\frac{\sqrt{(2\pi)^D\alpha^{-D}\left|\boldsymbol{\Sigma}\right|}}{\left((2\pi)^D\left|\boldsymbol{\Sigma}\right|\right)^{\frac{\alpha}{2}}}\right) \tag{B.19}$$

que simplificando leva a

$$\begin{aligned} H_\alpha &= \frac{1}{1-\alpha}\log\left(\sqrt{\alpha^{-D}}\left((2\pi)^D\left|\boldsymbol{\Sigma}\right|\right)^{\frac{1}{2}(1-\alpha)}\right) \\ H_\alpha &= \frac{1}{2}\frac{1}{1-\alpha}\left[-D\log\left(\alpha\right)+(1-\alpha)D\log\left(2\pi\right)+(1-\alpha)\log\left(\left|\boldsymbol{\Sigma}\right|\right)\right] \end{aligned} \tag{B.20}$$

onde finalmente temos

$$H_\alpha = \frac{1}{2}\left[-D\frac{\log\left(\alpha\right)}{1-\alpha}+D\log\left(2\pi\right)+\log\left(\left|\boldsymbol{\Sigma}\right|\right)\right]. \tag{B.21}$$

Para o caso da entropia quadrática ($\alpha = 2$), obtemos a seguinte expressão

$$H_2 = \frac{1}{2}\left[D\log\left(4\pi\right)+\log\left(\left|\boldsymbol{\Sigma}\right|\right)\right], \tag{B.22}$$

que já foi mostrada no capítulo 2.

Para o caso onde $\alpha = 1$, devemos obter a medida da entropia que corresponde a entropia de Shannon, como citado no capítulo 2. Calculando o limite

$$H_1 = \lim_{\alpha \to 1} \left[\frac{1}{2} \left(-D \frac{\log(\alpha)}{1-\alpha} + D \log(2\pi) + \log(|\mathbf{\Sigma}|) \right) \right], \tag{B.23}$$

que leva a

$$H_1 = \frac{1}{2} \left[-D \lim_{\alpha \to 1} \frac{\log(\alpha)}{1-\alpha} + D \log(2\pi) + \log(|\mathbf{\Sigma}|) \right], \tag{B.24}$$

obtemos, resolvendo o limite

$$H_1 = \frac{1}{2} \left[D + D \log(2\pi) + \log(|\mathbf{\Sigma}|) \right], \tag{B.25}$$

que de fato é a entropia de Shannon para uma VA gaussiana.

B.5 Segundo momento de uma gaussiana multivariada

Nesta seção será demonstrado o que o segundo momento central de uma gaussiana multivariada corresponde a sua matriz de covariância. Este resultado embora trivial, possui uma demonstração que ajuda a compreender algumas propriedades de integrais de gaussianas multivariadas. Isto é particularmente útil no cálculo das divergências apresentadas neste trabalho.

O segundo momento de uma VA Y com distribuição $p(\mathbf{y})$ é uma matriz $\mathbf{C}$ definida por

$$\boldsymbol{C} = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} (\boldsymbol{y} - \boldsymbol{\mu}) \ (\boldsymbol{y} - \boldsymbol{\mu})^t \ p(\boldsymbol{y}) \mathrm{d}\boldsymbol{y} \tag{B.26}$$

Para uma VA com distribuição gaussiana (média $\boldsymbol{\mu}$ e matriz de covariância $\mathbf{\Sigma}$), a expressão torna-se

$$\boldsymbol{C} = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} (\boldsymbol{y} - \boldsymbol{\mu}) \ (\boldsymbol{y} - \boldsymbol{\mu})^t \ k e^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^t \mathbf{\Sigma}^{-1} (\boldsymbol{y}-\boldsymbol{\mu})} \mathrm{d}\boldsymbol{y} \tag{B.27}$$

onde k é a constante de normalização da distribuição.

Fazendo

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{\Sigma}^{-1} \\ \boldsymbol{x} &= \boldsymbol{y} - \boldsymbol{\mu} \end{aligned} \tag{B.28}$$

e substituindo na integral temos

$$\boldsymbol{C} = k \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \boldsymbol{x}\,\boldsymbol{x}^t e^{-\frac{1}{2}\boldsymbol{x}^t\,\boldsymbol{A}\,\boldsymbol{x}} \mathrm{d}\boldsymbol{x}. \tag{B.29}$$

Utilizando o mesmo artifício usado na seção B.2 realizamos a seguinte substituição de variáveis

$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{T}\,\boldsymbol{x} \\ \mathrm{d}\boldsymbol{x} &= |\boldsymbol{J}|\,\mathrm{d}\boldsymbol{z} \end{aligned} \tag{B.30}$$

onde $\mathbf{T}$ é a matriz dos autovetores da matriz $\mathbf{A}$ e $\mathbf{\Lambda}$ uma matriz diagonal com os autovetores de $\mathbf{A}$. $\mathbf{J}$ é a matriz Jacobiano de $\mathbf{z}$, que, como $|\mathbf{T}| = 1$, esta também será. Isto nos leva a seguinte integral

$$\boldsymbol{C} = k \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \boldsymbol{T}^{-1}\,\boldsymbol{z}\,\boldsymbol{z}^t\,\boldsymbol{T}\,e^{-\frac{1}{2}\boldsymbol{z}^t\,\boldsymbol{\Lambda}\,\boldsymbol{z}}\,|\boldsymbol{J}|\,\mathrm{d}\boldsymbol{z}. \tag{B.31}$$

Rearanjando os temos na integral obtemos

$$\boldsymbol{C} = k\boldsymbol{T}^{-1} \left(\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \boldsymbol{z}\,\boldsymbol{z}^t\,e^{-\frac{1}{2}\boldsymbol{z}^t\,\boldsymbol{\Lambda}\,\boldsymbol{z}} \mathrm{d}\boldsymbol{z} \right) \boldsymbol{T}. \tag{B.32}$$

Podemos mostrar que a matriz resultante da integral

$$\boldsymbol{B} = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \boldsymbol{z}\,\boldsymbol{z}^t\,e^{-\frac{1}{2}\boldsymbol{z}^t\,\boldsymbol{\Lambda}\,\boldsymbol{z}} \mathrm{d}\boldsymbol{z} \tag{B.33}$$

possui as seguintes características

$$b_{ij} = 0,\ \forall\ i \neq j. \tag{B.34}$$

Como a matriz $\mathbf{\Lambda}$ é diagonal, os elementos da diagonal de $\mathbf{B}$ podem ser calculados

como sendo

$$b_{ii} = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} z_i^2 e^{-\frac{1}{2}\left(\sum_{j=1}^{n} \lambda_j z_j^2\right)} \mathrm{d}z_1 \mathrm{d}z_2 ... \mathrm{d}z_\mathrm{n}. \tag{B.35}$$

Separando as integrais obtemos

$$b_{ii} = \int_{-\infty}^{\infty} z_i^2 e^{-\frac{1}{2}\lambda_i z_i^2} \mathrm{dz_i} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\sum_{j=1, j\neq i}^{n} \lambda_j z_j^2\right)} \mathrm{dz_1 ... dz_{i\,-\,1} dz_{i\,+\,1} ... dz_n} \tag{B.36}$$

Resolvendo cada integral, chegamos a expressão

$$b_{ii} = \frac{1}{2}\sqrt{\frac{\pi}{\left(\frac{\lambda_i}{2}\right)^3}}\sqrt{\frac{(2\pi)^{n-1}}{\frac{|\mathbf{\Lambda}|}{\lambda_i}}} \tag{B.37}$$

$$b_{ii} = \frac{1}{\lambda_i}\sqrt{\frac{(2\pi)^n}{|\mathbf{\Lambda}|}}. \tag{B.38}$$

Sendo

$$\boldsymbol{A} = \mathbf{\Sigma}^{-1} \rightarrow \frac{1}{\lambda_i} = \lambda_i', \tag{B.39}$$

Obtemos, para a matriz $\mathbf{B}$ a expressão

$$\boldsymbol{B} = \sqrt{(2\pi)^n \, |\mathbf{\Sigma}|}\, \mathrm{diag}(\lambda_1', \lambda_2', ..., \lambda_n'). \tag{B.40}$$

Substituindo na expressão B.33 obtemos

$$\boldsymbol{C} = k\boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{T} \tag{B.41}$$

$$\boldsymbol{C} = \boldsymbol{T}^{-1}\, \mathrm{diag}(\lambda_1', \lambda_2', ..., \lambda_n')\boldsymbol{T}. \tag{B.42}$$

De fato, os autovalores que formam a matriz diagonal da equação anterior são os autovalores de $\mathbf{\Sigma}$. Como $\mathbf{T}$ é a matriz dos autovetores de $\mathbf{\Sigma}$, a matriz C torna-se portanto

$$\boldsymbol{C} = \boldsymbol{\Sigma}. \tag{B.43}$$

Referências Bibliográficas

[1] Peter Ahrendt. The multivariate gaussian probability distribution. Technical report, Technical University of Denmark, 2002.

[2] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *In 2nd International Symposium on Information Theory*, pages 267–281, 1973.

[3] H. Beigi, S. Maes, and J. Sorensen. A distance measure between collections of distributions and its application to speaker recognition. In *Proceedings of ICASSP 1998*, volume 2, pages 753–7756, 1998.

[4] Yoshua Bengio and Pascal Vincent. Manifold parzen windows. Technical report, CIRANO, 2004.

[5] Pavel Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[6] Christopher M. Bishop. *Neural Networks for pattern recognition*. Oxford, 2002.

[7] Hamparsum Bozdogan. Mixture-model cluster analysis using model selection criteria and a new informational measure of complexity. In *Proceedings of the First US/Japan Conference on the frontiers of Statistical Modeling: An informational Approach*, pages 69–113. Kluler Academic Publishers, 1994.

[8] G. A. Carpenter and S. Grossberg. The art of adaptive pattern recognition by a self-organizing neural network. *IEEE Computer*, 21:77–88, 1988.

[9] O. Chapelle. Active learning for parzen window classifier. *AI STATS*, 2005.

[10] C. W. Chen, J. Luo, and K. J. Parker. Image segmentation via adaptive k-means clustering and knowledge-based morphological operations with biomedical applications. *Elsevier Engineering Information*, 7, 1998.

[11] Pierre Comon. Independent component analysis, a new concept? *Signal Process.*, 36(3):287–314, 1994.

[12] Louis Comtet. *Advanced Combinatorics: The Art of Finite and Infinite Expansions*. D. Reidel Publishing Company, 1974.

[13] José Alfredo Costa. *Calssificação Automática e Análise de Dados por Redes Neurais Auto-organizáveis*. PhD thesis, Universidade Estadual de Campinas, 1999.

[14] Frank Dellaert. *The Expectation Maximization Algorithm*. PhD thesis, College of Computing, Georgia Institute of Technology, 2002.

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

[16] R. O. Duda and P. E. Hart. *Pattern Classification*. John Wiley & sons, 1998.

[17] Brian S. Everitt. *Cluster Analysis*. Arnold, 1993.

[18] Mário A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 24:381–396, 2002.

[19] Mário A. T. Figueiredo, José M. N. Leitão, and Anil K. Jain. On fitting mixture models. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 54–69, 1999.

[20] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, pages 179–188, 1936.

[21] B. Flury and H. Riedwyl. *Angewandte multivariate Statistik*. Gustav Fischer Verlag, Stuttgart, 1983.

[22] Chris Fraley and Adrian E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[23] A. Gersho. On the structure of vector quantizers. *IEEE Transactions on Information Theory*, 28:157–166, 1982.

[24] Erhan Gokcay. *A New Clustering Algorithm for Segmentation of Magnetic Resonance Images*. PhD thesis, University Of Florida, 2000.

[25] Erhan Gokcay and José Príncipe. A new clustering evaluation function using rényi's information potential. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2000.

[26] Joseph Goldstein, Dale E. Newbury, David C. Joy, Charles E. Lyman, Patrick Echlin, Eric Lifshin, L.C. Sawyer, and J.R. Michael. *Scanning Electron Microscopy and X-ray Microanalysis*. Plenum US, 3 edition, 2003.

[27] Rafael C. Gonzales. *Digital Image Processing*. Addison-Wesley, 1999.

[28] Graham L. Goodman and Daniel W. McMichael. A gaussian mixture model classifier using supervised and unsupervised learning. *Annals IEEE International Symposium on Signal Processing and its applications*, 1996.

[29] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Intelligent Information Systems Journal*, pages 107–145, 2001.

[30] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Cluster validity methods: Part i. *SIGMOD Record*, 2002.

[31] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Cluster validity methods: Part ii. *SIGMOD Record*, 2002.

[32] André Hardy. On the number of clusters. *Computacional Statistics & Data Analysis*, 23:83–96, 1996.

[33] Simon Haykin. *Neural Networks a Comprehensive Foundation*. Prentice Hall, second edition, 1999.

[34] Kenneth E. Hild, Deniz Erdogmus, and José Príncipe. Blind source separation using rényi's mutual information. *IEEE Signal Processing Letters*, 8:174–176, 2001.

[35] Aapo Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[36] Aapo Hyvärinen and Erkki Oja. Independent component analysis: A tutorial. Technical report, 1999.

[37] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1988.

[38] Anil K. Jain and Mário A. T. Figueiredo. Unsupervised selection and estimation of finite mixture models. In *International Conference on Pattern Recognition (ICPR'00)*, volume 2, 2000.

[39] Jianmin Jiang and Darren Butler. A genetic algorithm for vector quantization. In *Genetic Algorithms in Engineering Systems: Innovations and Applications*, 1995.

[40] Ian T. Jolliffe. *Principal Component Analisys*. Springer Verlag, 1986.

[41] Chang Jou, Quen Zong Wu, Shuh Chuan Tsay, Yuh Jiuan Tsay, and Shih Shien Yu. A hyperellipsoid neural network for pattern classification. In *IEEE International Symposium on Circuits and Systems*, volume 2, pages 1176–1179, 1991.

[42] T. Kailath. The divergence and the bhattacharyya distance measures in signal selection. *IEEE Trans. Commun. Technol*, COM-15:52–60, 1967.

[43] Arie Kaufman. *Volume Visualization*. IEEE Computer Society Press Tutorial, 1991.

[44] André I. Khuri. *Advanced Calculus with Applications in Statistics*. John Wiley & Sons, Inc., 2 edition, 2002.

[45] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983.

[46] Teuvo Kohonen. *Self-Organizing Maps*. Springer, 3 edition, 2001.

[47] Bart Kosko. *Neural Network for signal processing*. Prentice Hall, 1992.

[48] Ravi Kothari and Dax Pitts. On finding the number of clusters. *Pattern Recognition Letters*, 20:405–416, 1999.

[49] Solomon Kullback. *Information Theory and Statistics*. Dover, 1 edition, 1997.

[50] Sukhan Lee and Shunichi Shimoji. Self-organization of gaussian mixture model for learning class pdfs in pattern classification. *IEEE International Joint Conference on Neural Networks*, 1993.

[51] X. Q. Li and I King. Gaussian mixture distance for information retrieval. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN1999)*, volume 4, pages 2544–2549, 1999.

[52] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 1:145–151, 1991.

[53] Yoseph Linde, Andrés Buzo, and Robert M. Gray. An algorithm for vetor quantizer design. *IEEE Transactions on comunications*, 28, 1980.

[54] H. Lipson and H. T. Siegelmann. Clustering irregular shapes using high-order neurons. *Neural Computation*, 12:2331–2353, 1999.

[55] Jianchang Mao and Anil K. Jain. A self-organing network for hiperellipsoidal clustering. *IEE Transactions on neural networks*, 7, 1996.

[56] Allan M. Martins, Adrião D. D. Neto, Wedson T. de A. Filho, and Agostinho de M. B. Junior. A new method for mu ltitexture segmentation using neural networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN2002)*, 2002.

[57] Allan M. Martins, Adrião D. D. Neto, Agostinho de M. B. Junior, Alexandre Sales, and Sara Jane. Texture bases segmentation of cell images using neural networks and mathematical morphology. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN2001)*, 2001.

[58] Allan M. Martins, Adrião D. D. Neto, and Jorge D. Melo. Neural networks applied to classification of data based on mahalanobis metrics. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN2003)*, volume 4, pages 3071–3076, 2003.

[59] Allan M. Martins, Adrião D. D. Neto, and Jorge D. Melo. Comparison between mahalanobis distance and kullback-leibler divergence in clustering analysis. *WSEAS Transactions on SYSTEMS*, 3:501–505, 2004.

[60] Allan M. Martins, Adrião D. D. Neto, and Jorge D. Melo. A neural networks based algorithm for complex pattern classification problems. *Learning and Nonlinear Models*, 1:195–207, 2004.

[61] Allan M. Martins, Adrião D. D. Neto, Jorge D. Melo, and José A. F. Costa. Clustering using neural networks and kullback-leibler divergency. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN2004)*, 2004.

[62] Geoffrey J. McLachan. *Discriminant analisys and statistical pattern recognition.* Wiley, 1992.

[63] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[64] G. Miller and D. Horn. Maximum entropy approach to probability density estimation. In *Proceedings of Second International Conference on Knowledge-Based Intelligent Electronic Systems*, 1998.

[65] Thomas P. Minka. Expectation maximization as lower bound maximization. *Internet published*, 1998.

[66] Dirk Ormoneit and Volker Tresp. Improved gaussian mixture density estimates using bayesian penalty terms and network averaging. In *Advances in Neural Information Processing Systems*, volume 8, pages 542–548. The MIT Press, 1996.

[67] Maria D. M. Paiva, Dulce M. A. Melo, and Antonio E. Martinelli. Microstructural evaluation of oil well cementing slurries using alternative materials. In *Proceedings of the Rio Oil & Gas Conference*, 2004.

[68] Athanasios Papoulis. *Probability, Random Variables and Stochastic Processes*. Mc Graw Hill, 3 edition, 1991.

[69] J. R. Parker. Extracting vectors from raster images. *Computer & Graphics*, page 12, 1988.

[70] G. Patanè and M. Russo. Fully automatic clustering system. *IEEE Transactions on Neural Networks*, 13(6):1285–1298, Novembro 2002.

[71] Edward A. Patrick and Frederich P. Fisher. A generalized k-nearest neighbor rule. *Information and control*, pages 128–152, 1970.

[72] H. Permuter, J. Francos, and I. H. Jermyn. Gaussian mixture models of texture and colour for image database retrieval. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 3:III569–III572, 2003.

[73] Kaare Brandt Petersen and Michael Syskind Pedersen. The matrix cookbook. Technical report, Technical University of Denmark, 2004.

[74] Dzung L. Pham and Jerry L. Prince. An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities. *Pattern Recognition Lettes*, 20, 1999.

[75] Robert Porter and Nishan Canagarajah. Robust automatic clustering scheme for image segmentation using wavelets. *Elsevier Engineering Information*, 5, 1996.

[76] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Internet Published, 2 edition, 1992.

[77] A. Rényi. On measures of entropy and information. *Selected Papers of Alfred Rényi*, 2:565–580, 1976.

[78] Robert Schalkoff. *Pattern recognition*. John Willey & sons, 1992.

[79] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27, 1948.

[80] Ilya M. Sobol. *A primer for Monte Carlo Methods*. CRC Press, 1 edition, 1994.

[81] H. Späth. *Cluster Analysis Algorithms: for data reduction and classification of objects*. Ellis Horwood, Chichester, West Sussex, England, 1980.

[82] Claus Svarer. *Neural Networks for Signal Processing*. PhD thesis, Technical University of Denmark, 1994.

[83] A. Szymkowiak, J. Larsen, and L. Hansen. Hierarchical clustering for datamining, 2001.

[84] M. E. Tipping. Deriving cluster analytic distance functions from gaussian mixture models. In *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN99)*, volume 2, pages 815–820, 1999.

[85] Eric W. Tyree and J. A. Long. The use of linked line segments for cluster representation and data reduction. *Pattern Recognition Letters*, 20:21–29, 1999.

[86] A. Ultsch and C. Vetter. Self-organining-feature-maps versus statistical clustering methods: A benchmark. *FG Neuroinformatik & Künstliche Intelligenz*, 1994.

[87] C. J. Veenman, M. J. T. Reinders, and E. Backer. A maximum variance cluster analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9), 2002.

[88] Raymond Veldhuis. The centroid of the symmetrical kullback-leibler distance. *IEEE Signal Processing Letters, no. 3*, 9:96–99, 2002.

[89] Lei Xu and Michael I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural Computation*, 8(1):129–151, 1996.

[90] Dongsuk Yook. Decision tree based clustering. In *IDEAL*, pages 487–492, 2002.

[91] K. S. Younis, S. K. Rogers, and M. P. DeSimio. Vector quantization based on dynamic adjustment of mahalanobis distance. 2:858–862, 1996.

[92] Y. Zhang, M. Alder, and R. Togne. Using gaussian mixture modeling in speech recognition. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1:I613–I616, 1994.

[93] Zheng and Yong-Jian. Feature extraction and image segmentation using self-organizing networks. *Elsevier Engineering Information*, 8, 1995.

[94] K. Zyczkowski. Rényi extrapolation of shannon entropy. *Open Syst. Inf. Dyn*, 10:297–310, 2003.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo