UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

ADRIANO SOARES DE OLIVEIRA BAILÃO

Reconhecimento de padrões por processos adaptativos de compressão

Goiânia
2020

SIBI
SISTEMA DE BIBLIOTECAS UFG

PRPG
PRÓ-REITORIA DE PÓS-GRADUAÇÃO

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o(a) autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico: **[ ] Dissertação** **[X] Tese**

2. Identificação da Tese ou Dissertação:

Nome completo do(a) autor(a): Adriano Soares de Oliveira Bailão

Título do trabalho: Reconhecimento de padrões por processos adaptativos de compressão

3. Informações de acesso ao documento:

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

Independente da concordância com a disponibilização eletrônica, é imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.

Adriano S. O. Bailão
Assinatura do(a) autor(a)[2]

Ciente e de acordo:

Andrea da Silva Soares
Assinatura do(a) orientador(a)[2] Data: 02 / 03 / 2020

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à) autor(a) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação. O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

[2] As assinaturas devem ser originais sendo assinadas no próprio documento. Imagens coladas não serão aceitas.

ADRIANO SOARES DE OLIVEIRA BAILÃO

Reconhecimento de padrões por processos adaptativos de compressão

Tese apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação.

Orientador: Prof. Dr. Anderson da Silva Soares
Co-Orientador: Prof. Dr. Alexandre Cláudio Botazzo Delbem

Goiânia
2020

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Soares de Oliveira Bailão, Adriano
Reconhecimento de padrões por processos adaptativos de compressão [manuscrito] / Adriano Soares de Oliveira Bailão. - 2020.
CLX, 160 f.: il.

Orientador: Prof. Dr. Anderson da Silva Soares; co-orientador Dr. Alexandre Cláudio Botazzo Delbem.
Tese (Doutorado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação em rede (UFG/UFMS), Goiânia, 2020.
Bibliografia.
Inclui siglas, fotografias, gráfico, tabelas, lista de figuras, lista de tabelas.

1. Inteligência Computacional. 2. Reconhecimento de Padrões. 3. Aprendizado de Máquina. 4. Análise de Agrupamentos. 5. Compressão de dados. I. da Silva Soares, Anderson, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE TESE

Ata Nº **01/2020** da sessão de Defesa de Tese de **Adriano Soares de Oliveira Bailão** que confere o título de Doutor em **Ciência da Computação**, na área de concentração em Ciência da Computação**.**

Aos dois dias do mês de março de dois mil e vinte, a partir das catorze horas, na sala 150 do Instituto de Informática, realizou-se a sessão pública de Defesa de Tese intitulada "**Reconhecimento de Padrões por Processos Adaptativos de Compressão**". Os trabalhos foram instalados pelo Orientador, Professor Doutor Anderson da Silva Soares (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Ronaldo Martins da Costa (INF/UFG), membro titular interno; Professora Doutora Nádia Félix Felipe da Silva (INF/UFG), membra titular externa, Professor Doutor Francisco José Mônaco (ICMC/USP), membro titular externo, Professor Doutor Cláudio Gottschalg Duque (FCI/UNB), membro titular externo, sendo que a participação dos três últimos ocorreu através de videoconferência. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Tese tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor Anderson da Silva Soares, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos dois dias do mês de março de dois mil e vinte.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Anderson Da Silva Soares**, **Professor do Magistério Superior**, em 02/03/2020, às 18:22, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Francisco José Monaco**, **Usuário Externo**, em 02/03/2020, às 18:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Ronaldo Martins Da Costa**, **Professor do Magistério Superior**, em 02/03/2020, às 18:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Nadia Felix Felipe Da Silva**, **Professor do Magistério Superior**, em 02/03/2020, às 18:28, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Claudio Gottschalg Duque**, **Usuário Externo**, em 02/03/2020, às 18:36, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Referência: Processo nº 23070.007378/2020-68 SEI nº 1164603

Agradecimentos

Agradeço ao meu orientador Anderson da Silva Soares (INF/UFG), aos meus co-orientadores Alexandre Cláudio Botazo Delbem (ICMC-USP), António Gaspar Cunha (UM/Univesidade do Minho) e Telma Woerle de Lima Soares (INF/UFG), por toda atenção, paciência e confiança.

Agradeço ao meu pai Alan Kardec, minha mãe Fátima Cáritas, e a minha tia Nara Maria, por todo carinho, incentivo e a oportunidade de estudar com tranquilidade desde o meu primeiro dia de aula, aos 4 anos de idade em 1981, até hoje (atualmente) aos 43 anos em 2020 concluindo este doutoramento.

"O começo da sabedoria é encontrado na dúvida; duvidando começamos a questionar, e procurando podemos achar a verdade."

Pierre Abelard,
Dialética.

Resumo

Bailão, Adriano Soares de Oliveira. **Reconhecimento de padrões por processos adaptativos de compressão**. Goiânia, 2020. 160p. Tese de Doutorado Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

A compressão de dados é um processo amplamente utilizado pela indústria no armazenamento e transporte de informações sendo aplicada a uma variedade de domínios como texto, imagem, áudio e vídeo. Os processos de compressão constituem-se em um conjunto de operações matemáticas que visam representar cada amostra de dados na forma comprimida, ou com menor tamanho. Técnicas de reconhecimento de padrões podem utilizar propriedades e métricas de compressão para conceber modelos de aprendizado de máquina a partir de algoritmos adaptativos que representam as amostras na forma comprimida. Uma vantagem dos modelos de compressão adaptativos, é que dispõem de técnicas de redução de dimensionalidade decorrentes das propriedades de compressão. Essa tese propõe um modelo de aprendizado não-supervisionado geral (para diversos domínios de problema e diferentes tipos de dados), que reúne as estratégias adaptativas de compressão em duas fases : a granulação, responsável pela percepção e representação do conhecimento necessário para resolver um problema de generalização, e a fase de codificação, responsável pela estruturação do raciocínio do modelo, a partir da representação e organização dos objetos do problema. O raciocínio expresso pelo modelo denota a capacidade de generalização dos objetos de dados no contexto geral. Métodos genéricos, baseados em compactadores (sem perda de informação), carecem de capacidade de generalização para alguns tipos de objetos de dados, sendo, nessa tese, utilizadas também, técnicas de compressão, com perdas, visando contornar o problema e aumentar a capacidade de generalização do modelo. Resultados demonstram que a utilização de técnicas e métricas baseadas em compressão adaptativa produzem uma boa aproximação das amostras de dados originais em fontes de dados com alta dimensionalidade. Testes apontam para bons modelos de aprendizado de máquina com boa capacidade de generalização derivados da abordagem baseada na redução de dimensionalidade oferecida por processos adaptativos de compressão.

Palavras–chave

Inteligência Computacional, Reconhecimento de Padrões, Aprendizado de Máquina, Análise de Agrupamentos, Compressão de dados.

Abstract

Bailão, Adriano Soares de Oliveira.
. Goiânia, 2020. 160p. PhD. Thesis Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

Data compression is a process widely used by the industry in the storage and transport of information and is applied to a variety of domains such as text, image, audio and video. The compression processes are a set of mathematical operations that aim to represent each sample of data in compressed form, or with a smaller size. Pattern recognition techniques can use compression properties and metrics to design machine learning models from adaptive algorithms that represent samples in compressed form. An advantage of adaptive compression models, is that they have dimensionality reduction techniques resulting from the compression properties. This thesis proposes a general unsupervised learning model (for different problem domains and different types of data), which combines adaptive compression strategies in two phases: granulation, responsible for the perception and representation of the knowledge necessary to solve a problem generalization, and the codification phase, responsible for structuring the reasoning of the model, based on the representation and organization of the problem objects. The reasoning expressed by the model denotes the ability to generalize data objects in the general context. Generic methods, based on compactors (without loss of information), lack generalization capacity for some types of data objects, and in this thesis, lossy compression techniques are also used, in order to circumvent the problem and increase the capacity of generalization of the model. Results demonstrate that the use of techniques and metrics based on adaptive compression produce a good approximation of the original data samples in data sources with high dimensionality. Tests point to good machine learning models with good generalization capabilities derived from the approach based on the reduction of dimensionality offered by adaptive compression processes.

Keywords

Computational Intelligence, Pattern Recognition, Machine Learning, Cluster Analysis, Data Compression.

Lista de Siglas

ASCII	American Standard Code for Information Interchange
BWT	Burrows-Wheeler Transformation
CAPTCHA	Comple-tely Automated Public Turing test to tell Computers and Humans Apart
DAMICORE	Data Mining of Code Repositories
FD	Fonte de Dados
FS	Fast Newman
GM	General Mining
HTML	Hypertext Markup Language
JPEG	Joint Photographics Experts Group
JSON	JavaScript Object Notation
K	Complexidade de Kolmogorov
KNN	k Nearest Neighbours
MPEG	Moving Picture Experts Group
LZ77	Lempel Ziv 77
LZW	Lempel Ziv Welch
MT	Máquina de Turing
NCD	Normalized Compression Distance
NID	Normalized Information Distance
NJ	Neighbor Joining
ORL	Olivetti Research Laboratory
PCA	Principal Component Analysis
PNG	Portable Network Graphics
RGB	Red, Green, Blue
RLE	Run-length Encoding
SGBD	Sistema de Gerenciamento de Banco de Dados
TCF	Teorema da Codificação de Fonte
TDC	Transformada Discreta de Cossen

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Reconhecimento de padrões é definido por [84] como a ciência do estudo da organização dos dados e pode ser realizada a partir de objetos representados em diversos tipos de dados como textos, áudios, sinais, vídeos, entre outros. Segundo [64], as técnicas de reconhecimento de padrões podem ser divididas, principalmente, em três abordagens: estatística, estrutural e neural. Em [35], os autores também consideram a abordagem de casamento de modelos (*template matching*, em inglês), mas argumentam que essas abordagens não são, necessariamente, independentes e que frequentemente um mesmo algoritmo existe sob diferentes interpretações. De acordo com [36], o casamento de modelos é tipo mais simples de abordagem para reconhecimento de padrões e é realizado determinando a similaridade entre duas entidades de mesmo tipo, podendo ser pontos, curvas, formas e outros. Segundo [64], na abordagem estatística são utilizadas fronteiras de decisão baseadas na distribuição estatística das características; a abordagem estrutural entende que a informação significante é descrita não somente pelas características, mas também pelas inter-relações, e os objetos são reconhecidos de acordo com a similaridade da sua descrição ou representação estrutural; e, na abordagem neural, as fronteiras de decisão são definidas com base na minimização do erro de classificação. Nessa perspectiva, acrescenta-se uma abordagem que explora o conceito de compressão de dados para reconhecer objetos semelhantes por meio da aproximação de suas informações mútuas [41].

A compressão é um processo que tem como objetivo reduzir a quantidade de dados contida em uma fonte partindo da hipótese da existência de informações redundantes que podem ser eliminadas de alguma forma [78, 73]. Os processos de compressão, geralmente, são implementados através de uma regra, chamada de código ou protocolo, que elimina os bits redundantes de informações, de modo a diminuir seu tamanho [73, 39]. Além da eliminação da redundância, os dados são comprimidos por diversas razões. Dentre as mais conhecidas razões para compressão, destacam-se a economia de espaço em dispositivos de armazenamento, como discos rígidos, ou ganhar desempenho (diminuir tempo) em transmissões [78]. Embora possam parecer sinônimos, compressão e compactação de dados são processos distintos. A compressão reduz a

quantidade de bits para representar algum dado enquanto a compactação tem a função de unir dados que não estejam unidos [65]. Um exemplo clássico de compactação de dados é a desfragmentação de discos. Da perspectiva do processo, a compressão de dados possui atividades semelhantes as atividades que compõem processos de reconhecimento de padrões, sendo assim, torna- -se possível utilizar técnicas e/ou atividades de compressão em processos de descoberta e aprendizado de padrões.

Existem diversas formas de se classificar os métodos de compressão de dados [37]. O mais conhecido é pela ocorrência ou não de perda de dados durante o processo [73]. Entretanto, diversas outras formas de classificação são úteis para se avaliar e comparar os métodos de compressão de dados e sua aplicação em problemas específicos. De acordo com [73], os métodos de compressão também podem ser classificados pela forma como operam, a saber: métodos estatísticos ou de aproximação de entropia; métodos baseados em dicionários ou de redução de redundância e; métodos baseados em transformação sem compressão direta.

1.1 Problema

A utilização da compressão de dados no contexto de reconhecimento de padrões foi proposta originalmente por [11]. O Modelo proposto em [11] não se restringe a um tipo de dado específico para as amostras de dados, porém, todas as amostras do conjunto possuem a mesma natureza e contém um rótulo identificador de classe. O rótulo identificador da classe divide a fonte de dados em subconjuntos disjuntos chamados de *grupos*. Cada amostra de uma fonte de dados é chamada de *objeto* e faz parte de um *grupo*. Para obter melhores níveis de generalização, em modelos de aprendizado não-supervisionado, o método de [11] baseou-se na extração de informações com semântica por meio da entropia dos objetos determinada pela Teoria da Complexidade de Kolmogorov utilizada e definida na secção 2.6.1 para a medida da complexidade. Segundo a teoria de Kolmogorov, a entropia algorítmica de uma cadeia de caracteres é o comprimento em bits do menor programa capaz de produzir essa cadeia [38]. Kolmogorov define o conceito de aleatoriedade intrínseca de um objeto x denotada por $K(x)$ e a complexidade condicional de dois objetos quaisquer x e y denotada por $K(x|y)$, Kolmogorov estabelece um limite inferior teórico, portanto incomputável [19], para essas descrições algorítmicas individual e condicional. Assim, os autores em [10] utilizaram a compressão de dados como aproximação para calcular um limite superior dessa complexidade algorítmica dos objetos e calcular essas semelhanças dentro de um conjunto de dados. A ideia básica é que dois objetos distintos podem ser agrupados como semelhantes se o conteúdo da informação de um deles explicar, de forma significante, o conteúdo da informação do outro. Nesse caso,

o tamanho resultante da compressão entre dois objetos mais similares tende a ser menor em relação aos outros, desde que normalizada.

Diversos trabalhos foram propostos utilizando medidas teóricas de similaridade baseadas na Teoria de Kolmogorov. A primeira foi a Distância da Informação (E) [11], posteriormente, sendo proposta uma versão normalizada para tratar arquivos de tamanhos diferentes, a Distância Normalizada da Informação (NID - *Normalized Information Distance*) [41]. A partir, dessas formulações, foram desenvolvidas métricas aproximadas de similaridade baseadas em compressão, sendo a Distância Normalizada de Compressão (NCD - *Normalized Compression Distance*) [22], a mais amplamente utilizada, e outras [14, 55, 61]. Os trabalhos citados anteriormente desenvolvem uma abordagem de aprendizagem não-supervisionada a partir da validação das métricas e aplicação em contexto específico. O processo de montagem e definição dos aglomerados é realizado por meio de heurísticas a partir da combinação de pares de objetos. Os modelos tradicionais, em grande parte, utilizam compressores comerciais de propósito geral e sem perda na reconstrução dos dados originais. Na abordagem tradicional, os resultados reportados na literatura são bem sucedidos para dados unidimensionais (texto), porém apresentam desempenho limitado para dados bidimensionais como no caso das imagens [60].

Dentre as desvantagens da abordagem tradicional, [34] afirma que os compressores comerciais não são ideais pois promovem uma aproximação ineficiente das informações mútuas ao tratarem diversos tipos de dados da mesma forma, sem se preocupar em como esses estão estruturados. Para problemas de agrupamento de imagens, ressalta-se que estas abordagens desprezam a correlação espacial dos dados — princípio fundamental da compressão de imagens [73] — e os compressores utilizados são sensíveis a ruídos uma vez que não aceitam variações mínimas entre dados similares ao não admitirem perda no processo de compressão [34]. Ainda como desvantagens dos métodos baseados nas derivações da NCD, são elencadas:

- Desprezo à semântica: apesar de ser uma das vantagens apontadas pelos autores, ao desenvolver um método livre de parâmetros, informações importantes de estruturas de dados bidimensionais, como a localidade em imagens, podem ser perdidas no processo de compressão dos arquivos byte-a-byte.
- *Lazy learning*: seu aprendizado é dito preguiçoso, pois a generalização do problema não é definida antes que um objeto desconhecido seja submetido à clusterização [32]. Como não possui a etapa de treinamento, todo o processamento é concentrado na predição, tornando-a computacionalmente dispendiosa. É necessária a preservação de todo o conjunto de dados e reconstrução da matriz de distâncias produzida pela NCD a cada novo objeto submetido ao *agrupamento*. Isso significa aumento progressivo do volume de dados armazenados e do tempo de processamento. Na

aplicação específica de recuperação de imagens, deve-se considerar o processamento computacionalmente mais oneroso.

- Rotulação dos *grupos*: como o processo de agrupamento de dados é um método não-supervisionado, não existe nenhum tipo de informação preexistente que promova uma classificação ótima para o conjunto de dados. O problema da rotulação de grupos é NP-difícil e necessita de uma heurística para encontrar uma solução próxima de uma solução ótima [42]. Para resolver esse problema, métodos adicionais, como a inclusão de uma etapa de aprendizado supervisionado [44], são necessários para rotulação automática dos *grupos*. Por se tratar de uma classificação executada sobre o resultado de uma descoberta de conhecimento automática, os acertos resultantes podem ser comprometidos pelo uso de métricas inadequadas ou falhas nas heurísticas de *agrupamento*. Quanto maior a semelhança entre os objetos contido no grupo, melhor o desempenho dos métodos de rotulação [7].

Os métodos baseados na NCD contribuíram de forma significativa para área de mineração de dados ao proporem uma solução livre de parâmetros e de propósito geral [22]. No trabalho de [75], por exemplo, foi possível demonstrar experimentalmente que a compressão é capaz não somente de tornar a representação de um objeto mais eficiente mas também de encontrar padrões para agrupamento de dados. Entretanto, também se mostrou limitada para avançar como soluções para determinados tipos de problemas, motivando o desenvolvimento de métodos específicos para cada tipo de dados como, por exemplo, as imagens. Explorando a aplicabilidade desse método, pode-se investigar em quais tipos de problemas de classificação de imagens seria possível obter melhores resultados.

De acordo com [4], um problema de similaridade de imagens pode ser entendido por meio de:

1. Um conjunto de características que descreva a assinatura das imagens;
2. Uma métrica adequada para calcular as distâncias entre as imagens.

Em grande parte dos problemas de classificação de imagens similares, são encontradas situações em que as características discriminantes do problema são desconhecidas e/ou as fronteiras das classes não são bem definidas. Nesses casos, quando há necessidade de um processo de tomada de decisão baseada em aprendizagem supervisionada, a aplicação de técnicas de reconhecimento de padrões tradicionais é difícil de ser implementada. Para [59], embora as escolhas mais usuais para avaliação de similaridade de imagens ainda envolvam a extração de características, a maior dificuldade está na escolha de quais características são discriminantes para o problema.

1.2 Agrupamentos e compreensão

A formação de agrupamentos de objetos utilizando técnicas de compressão de dados para a formulação de hipóteses, em grande parte, está associada às técnicas de redução de dimensionalidade em reconhecimento de padrões. Apesar da compressão de dados ter como motivação inicial o desempenho na transmissão de dados, resultados podem ser obtidos da técnica de compressão aplicada à medição de informações em objetos e extração de significado (semântica) desses objetos. Segundo [92], a compressão de dados pode ser utilizada para a representação de conhecimento como na abordagem PCA (*Principal Component Analysis*) [47]. Na PCA, a informação contida em um conjunto de dados é armazenada em uma estrutura computacional de dimensão reduzida a partir da projeção integral do conjunto de dados em um subespaço gerado por um sistema de eixos ortogonais. De acordo com [28], a PCA é um algoritmo de compressão com baixas perdas. O uso da técnica PCA para a compressão de dados justifica-se pela capacidade de representação de dados multidimensionais através da informação contida na matriz de covariância dos dados e pela fácil manipulação destas informações através de cálculos utilizando princípios da álgebra linear e estatística básica. Como visto na PCA, a compressão pode ser utilizada como um método de redução de dimensionalidade de um objeto para o tratamento do problema da dimensionalidade.

A compressão deve garantir que o objeto transformado (comprimido) possua a menor quantidade de informação possível. A informação de um objeto é descrita por símbolos ou combinação deles, concebendo, assim, as características de um objeto. A partir das propriedades de transformação contidas nas técnicas de compressão, Rudi em [12] pôde analisar relações expressas pelos objetos de dados na forma comprimida através de medidas estatísticas diferentes da variância utilizada na PCA. Rudi em [23] utiliza o conceito de compressão para representar objetos de dados em tarefas de agrupamento. Na abordagem de Rudi, os agrupamentos são obtidos através da análise das relações entre os objetos representados na forma comprimida utilizando o utilitário CompLearn [51]. A compressão de um objeto de dados resulta em novas características (combinações lineares ou não-lineares das características originais) e então as características resultantes da transformação de cada objeto podem não possuir um significado físico. Rudi associa o conceito de compressão ao conceito de aleatoriedade proposto por Kolmogorov [62, 17] visando extrair um valor numérico para a representação de cada objeto de dados comprimido. O valor numérico extraído do objeto de dados comprimido é denominado de complexidade K ou nível de incerteza de um objeto.

Kolmogorov propõe que a complexidade K de um objeto é uma medida absoluta da informação e obtida através da aferição da incerteza intrínseca de cada objeto individualmente. Através da álgebra proposta por Kolmogorov, os objetos podem ser trans-

formados em abstrações que são representadas em um modelo computacional, utilizando aproximações chamadas de compressores. Com a utilização de compressores, cada objeto pode ser expresso pelo tamanho do menor algoritmo capaz de representa-lo. [15].

Dentre as heurísticas que utilizam compressão para a formação de agrupamentos, o método quartet [82] objetiva a formação de uma estrutura hierárquica que representa o agrupamento onde objetos de dados semelhantes estão próximos, ao contrário daqueles que menos semelhanças apresentam. Nesta pesquisa, a estrutura hierárquica resultante da análise de semelhanças de um conjunto de objetos de dados é denominada hipótese H de agrupamento. Delbem em [75] implementa o fluxo de execução DAMICORE(*Data Mining of Code Repositories*) para a formação de hipóteses H em fontes de dados não-estruturados, que consiste de:

- compactadores para a representação dos objetos de dados de uma fonte O;
- a métrica NCD (Normalized Compression Distance) [24] para a comparação de objetos;
- a heurística NJ (Neighbor Joining) [72] baseada em parcimônia para a concepção do agrupamento de objetos;
- e um algoritmo de redes complexas FN (Fast Newman) [54] para a delimitação dos grupos particionais presentes dentro da estrutura de agrupamento hierárquico.

A metodologia de mineração de dados DAMICORE, introduzida em 2011, faz uso da NCD e tem possibilitado resultados em diversos domínios, como pode ser visto em [80, 52, 85, 20, 75, 43, 63]. Essa linha de investigação possibilita a concepção de modelos de aprendizado, baseados em reconhecimento de padrões, por meio de técnicas de compressão.

1.3 Objetivo

O objetivo geral da tese é conceber um método de aprendizado de máquina não-supervisionado, por meio de estratégias de compressão adaptativa que utilizam, além da abordagem sem perda de informação (utilizada em compactadores), a abordagem com perda de informação.

Os objetivos específicos são:

- mapear o conjunto adequado de características para objetos de dados de uma fonte;
- obter o conjunto de hipóteses de agrupamento para objetos de dados de fontes não--estruturadas;
- promover aumento da capacidade de generalização do modelo de aprendizado por meio da aplicação de técnicas de compressão com perdas.

A descrição detalhada de todas as técnicas utilizadas nesta tese se encontra no capítulo 4 - Materiais e métodos.

O capítulo 1 apresenta a motivação da tese. O capítulo 2 descreve o processo de compressão adaptativa, responsável pela codificação dos objetos de dados e sua medição. O capítulo 3 exibe as estruturas de dados do conjunto de objetos de dados. O capítulo 4 apresenta a metodologia de pesquisa proposta a partir dos materiais e métodos utilizados na pesquisa. O capítulo 5 apresenta os resultados dos experimentos utilizando objetos de dados de fontes com diferentes tipos de dados em diferentes domínios de problema. O capítulo 6 apresenta as conclusões da pesquisa e as considerações finais.

CAPÍTULO 2

Representação da informação e codificação de objetos

2.1 Definição de compressão

Seja $X = [e_0, e_1, e_2, \ldots, e_{t-1}]^T$ uma amostra, que representa um objeto de dados, composta de t eventos discretos com $0 < i < t$, $i \in \mathbb{N}$ sendo $e_i \in U = \{v_0, v_1, v_2, \ldots, v_{|U|-1}\}$. O conjunto U representa o conjunto dos possíveis símbolos ou valores dos eventos $e_i \in X$. Os eventos e_i possuem valores que fazem parte do domínio da informação do tipo de dados da amostra X, assim, $|\{X\}| \leq |U|$. A amostra X pode ser representada por um texto, onde cada símbolo do do texto equivale a um evento e_i. Caso X seja uma imagem digital, a função $i = c : R^n \rightarrow R$ indexa a sequência de eventos sendo c uma curva que percorre todos os pixels da função de imagem, realizando o preenchimento completo de espaço. Uma curva de preenchimento de espaço é uma curva cujo intervalo contém todo o quadrado unitário bidimensional generalizando-se para um hipercubo unitário n-dimensional[29].

Seja um experimento probabilístico que envolva a observação da saída de cada evento discreto da amostra X em unidades de tempo t, sendo $0 < i < t$. Esses eventos podem ser modelados como variáveis discretas aleatórias. Os valores emitidos pela saída de cada evento da amostra X são estatisticamente independentes e podem ser representados por uma fonte de dados H preditora discreta sem memória[6]. Em uma fonte H preditora, o símbolo ou valor emitido a qualquer tempo é independente de uma escolha prévia, sendo assim:

- antes de evento e_i ocorrer, há incerteza;
- quando o evento e_i ocorre, há surpresa;
- depois da ocorrência do evento e_i, há ganho na quantidade de informação, cuja essência pode ser interpretada como a resolução da incerteza.

Em uma fonte H, o ganho de informação é um conjunto de alternativas, entre os possíveis valores de U, que levam a emissão de um evento e_i. Se não há surpresa não há ganho de informação.

Supondo que uma fonte H possa *escolher* entre os valores alternativos de U, para cada evento $e_i \in X$, com igual probabilidade, a cada período de tempo i, a quantidade de informação, em bits (0 ou 1), que representa este conjunto de escolhas pode ser expressa por:

$$n = \log_2 m \quad [\text{bits}] \tag{2-1}$$

A partir da equação 2-11, conclui-se que n bits é a informação necessária para se escolher entre m alternativas equiprováveis, analogamente, *1 bit* é a ganho de informação necessária para escolher entre duas alternativas equiprováveis como mostra a equação 2-2.

$$n = \log_2 2 = 1\,\text{bit} \tag{2-2}$$

As escolhas realizadas pela fonte preditiva H podem ser comparadas aos eventos e_i correspondentes na amostra X original. As escolhas equivalentes realizadas pela fonte H com os eventos e_i na amostra original podem ser representadas com o número "1", caso contrário, com o número "0", e então, acrescentadas em uma lista L. A fonte H continua o processo emitindo uma escolha sobre o próximo evento e_{i+1} da amostra X. Caso valor escolhido e emitido pela fonte H não corresponda ao evento e_i, insere-se o numero "0"na lista L e então a fonte H pode realizar outra escolha até que o processo se encerre por meio da escolha do evento $e_i \in X$ correspondente na amostra original. A lista L pode ser dada pela expressão regular L com alfabeto $\Sigma = \{0, 1\}$:

$$L = 0^{q_0}1 \quad . \quad 0^{q_1}1 \quad . \quad ,..., \quad . \quad 0^{(q_{t-1})}1 \tag{2-3}$$

O vetor $Q = [q_0, q_1, q_2, \ldots, q_{t-1}]^T$ indica a quantidade de incerteza contida da amostra X representada pela quantidade de "0"s que compõe cada evento e_i :

- No melhor caso, escolhendo de imediato e corretamente cada evento $e_i \in X$, tem-se $q_0 = q_1 = ... = q_{t-1} = 0$ e $|L| = |X|$, incerteza mínima (I_{min});
- No pior caso, esgotando-se todas as alternativas de escolha para todas as tentativas de eventos $e_i \in X$, tem-se $q_0 = q_1 = q_{t-1} = |U|$ -1 e $|L| = |U|^t$, incerteza máxima (I_{max}).

A partir dos limites assintóticos de incerteza da informação tem-se:

$$|X| \leq |L| \leq |U|^t \tag{2-4}$$

Quanto mais escolhas emitidas pela fonte H, equivalentes aos eventos $e_i \in X$, mais reduzida (comprimida) torna-se a lista L, menos incerteza tem-se na amostra X.

Desta forma, define-se compressão K de uma amostra X como o menor tamanho possível da incerteza expressa por L.

$$min\{|L| : K(x) = I\} \quad para \quad x = X, \quad I \in \mathbb{N} \tag{2-5}$$

Uma forma de satisfazer o critério de otimização expresso na equação 2-5, ou seja, minimizar L, é por meio do calculo de ganho de informação de cada $e_i \in X$. A quantidade de informação $I(e_t)$ obtida por um evento e_t, a fim de minimizar L, pode ser calculada por:

$$I(e_t) = \log_2(\frac{1}{P(e_t)}) \quad [\text{bits}], \tag{2-6}$$

sendo $P(e_t)$ a probabilidade da ocorrência de e_t na amostra X. A equação 2-6 descreve as possibilidades de escolhas dicotômicas do valor de saída correspondente a cada evento $e_t \in X$ a partir de sua probabilidade. A incerteza de cada evento $e_t \in X$ é expressa em bits, onde cada bit representa uma escolha realizada pela fonte H preditora.

2.2 Compressão e redução de dimensionalidade

Conceitos de compressão, como na equação 2-5 podem ser utilizados para reduzir o espaço de dimensões que representa um objeto [18]. A separabilidade entre classes diminui monotonicamente à medida que o número de dimensões aumenta.

Os algoritmos atuais de agrupamento, devido a sua especificidade, são técnicas aplicadas a domínios específicos, limitados a determinados tipos de dados e dependentes de formatos pré-definidos. Dentre as dificuldades de implementação de uma técnica que concilie capacidade de generalização dos algoritmos com a concepção de um modelo genérico para a representação dos objetos de dados, destacam-se [40]:

- não há uma técnica de agrupamento universal capaz de revelar toda a variedade de estruturas que podem estar presentes para fontes de dados genéricos;
- na definição da medida de similaridade e dos critérios de agrupamento, os algoritmos de agrupamento geralmente dependem implicitamente da imposição de certas hipóteses a respeito da forma dos grupos ou da configuração dos múltiplos grupos;
- os dados dificilmente estão estruturados “idealmente”, ou seja, não formam configurações hiperesféricas, hiperelipsoidais, lineares, etc., de modo que cada novo algoritmo de agrupamento pode apresentar um comportamento diferente aos já existentes para uma dada formação específica dos dados no espaço de características.

As técnicas de formação de agrupamentos utilizando compressão de objetos de dados podem prover maior abstração dos algoritmos de análise de agrupamento

através do tratamento genérico das fontes de dados. A figura 2.1 mostra um conjunto de compactadores e formatos de compressão disponíveis e utilizados por diversos sistemas operacionais. Outros formatos de arquivos podem ser convertidos para os apresentados na figura 2.1.

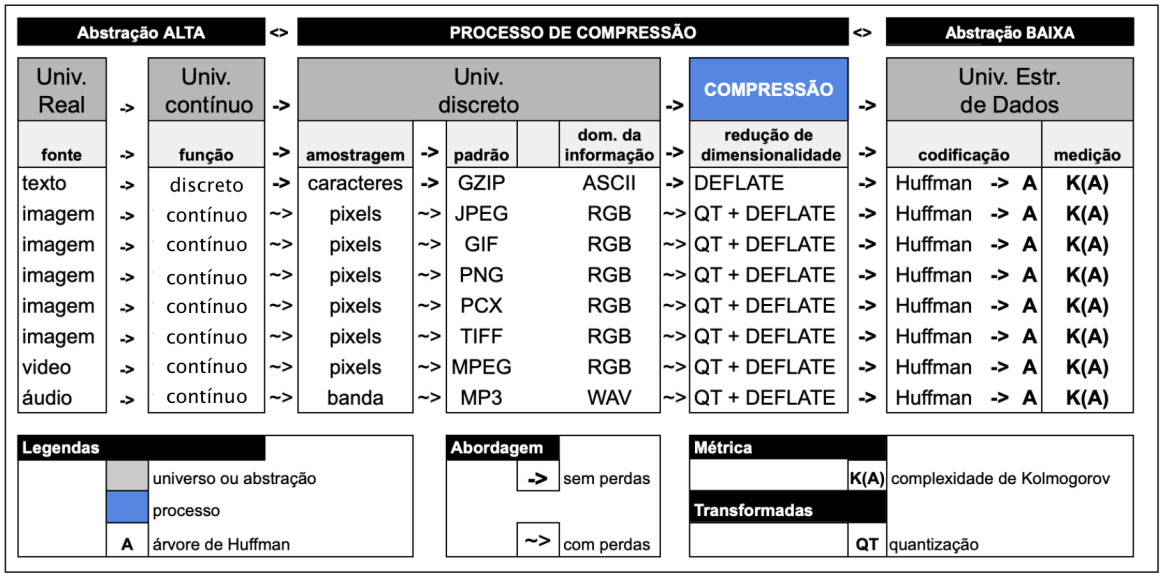

Figura 2.1: *Processos de compressão*

Os algoritmos, contidos no processo de compressão dos compactadores e compressores, compartilham basicamente a mesma codificação. Isso se deve ao fato da codificação Huffman ser ótima e sem perdas. Outras técnicas, como por exemplo, a técnica DEFLATE[1], são responsáveis pela identificação de estruturas elementares de informação que compõem os objetos de dados. Essas técnicas são implementadas por métodos de redução de redundância e transformadas que utilizam algum sistema de representação. Transformadas são métodos que por si só não comprimem os dados, mas são capazes de transformar dados que não seriam comprimidos ou seriam pouco comprimidos pelos métodos convencionais. Um exemplo é a Transformada Discreta de Cosseno (TDC), utilizada pelos padrões JPEG e MPEG. A granulação utiliza um sistema de representações, por exemplo, no compressor PNG o sistema é RGB e no compactador gzip ASCII. Na granulação são utilizadas técnicas de compressão de informação [66].

Nas abordagens propostas por [75] e [23], os compactadores utilizados para representação dos objetos alcançam generalização para o tratamento das fontes de dados não-estruturados, porém os compactadores utilizados nessas abordagens têm como prioridade a maximização de suas taxas de compressão objetivando somente a representação

[1] O algoritmo DEFLATE é usado nos utilitários que comprimem ou descomprimem arquivos no padrão ZIP ou no padrão gzip. A base do algoritmo é uma compressão usando LZ77 e a saída é codificada usando-se codificação de Huffman. Esse algoritmo também é usado para comprimir imagens no formato PNG.

do conhecimento estatístico com base no sistema de representação. Os compactadores não realizam a representação da informação semântica como nos processos cognitivos de percepção humana. Dentre as dificuldades das abordagens de medição da informação, que utilizam compactadores, destacam-se [67, 74]:

- os compactadores executam suas tarefas sempre com a mesma resolução da informação;
- falta de representação da localidade para objetos de dados do tipo imagem.

2.3 Nível de estruturação de uma fonte de dados

Um empecilho para a utilização de um único algoritmo de reconhecimento de padrões é a dependência dos algoritmos em geral com a fonte de dados cuja a natureza possui diversas formas de estruturação. As fontes de dados podem ser estruturadas da seguinte forma [71]:

- Dados estruturados: são organizados em blocos semânticos (relações) de um mesmo grupo e possuem as mesmas descrições (atributos). As descrições para todas as classes de um grupo possuem o mesmo formato (esquema). Os dados mantidos em um SGBD (Sistema Gerenciador de Banco de Dados) são chamados de dados estruturados por manterem a mesma estrutura de representação (rígida), previamente projetada. Os dicionários são fontes estruturadas de informação.
- Dados semiestruturados: atualmente, muitos dados não são mantidos em SGBDs. Dados Web, por exemplo, apresentam uma organização heterogênea. A alta heterogeneidade dificulta consultas a estes dados. Assim, estes dados são classificados como semiestruturados.
- Dados não-estruturados: são os dados que não possuem uma estrutura definida. Normalmente, caracterizados por documentos textos, imagens, vídeos, etc. As estruturas são descritas implicitamente com base na detecção e análise estatística das associações dos elementos internos que compõem o objeto de dados.

Os principais formatos utilizados como fonte de dados para compressão podem ser enumerados como segue:

- .txt para fontes de objetos de dados não-estruturados do tipo texto;
- .wav para fontes de objetos de dados não-estruturados do tipo áudio;
- .bmp, .png, .jpg, .mpeg para fontes de objetos de dados não-estruturados do tipo imagem;
- .xml para fontes de objetos de dados semiestruturados;
- .csv para fontes de objetos de dados estruturados do tipo conjunto de tuplas numéricas.

2.4 Sistema de representação de uma fonte

Um sistema de representação estabelece os possíveis valores para cada característica do objeto. Os sistemas de representação utilizados, nesta pesquisa, são os mesmos utilizados por grande parte das técnicas de compressão e descritas na tabela 2.1. Observa-se a distribuição dos sistemas de representação a partir da estrutura da fonte e tipo de dados dos objetos na figura 2.1.

Estrutura	**Tipo de dados da fonte**	**Sistema de representação**
Não-estruturado	Texto	ASCII
	Audio	WAV
	Imagem	RGB
	Video	RGB
Semiestruturado	XML (bem formado), HTML, JSON, ...	UTF-8 e TAGs
Estruturado	Tuplas numéricas: Linhas das tabelas de SGBDs e planilhas eletrônicas.	Dicionário

Tabela 2.1: *Sistema de representação*

A identificação do sistema de representação torna-se essencial na determinação do algoritmo de compressão adequado para uma tarefa de reconhecimento de padrões. A partir da determinação do sistema de representação da fonte, pode-se determinar se um objeto é estruturado ou não e, então, realizar a estruturação de fontes não estruturadas pela granulação. A estruturação de uma fonte de dados é essencial para a operação do modelo de reconhecimento de padrões.

2.5 Granulação

Nesta pesquisa, a granulação é um conjunto de técnicas algorítmicas adaptativas de compressão utilizadas para o processo de estruturação utilizado em fontes de dados não-estruturadas. As estruturas elementares presentes em uma fonte de dados sem estruturação são chamadas de grânulos. Os grânulos são entidades de informação, que surgem no processo de abstração e derivação dos dados de uma fonte. Os grânulos são organizados em conjunto, devido a sua semelhança, adjacência física (ou funcional) e coerência.

Dessa forma, os grânulos são padrões elementares que formam um padrão maior. A unidade elementar de informação é dada pelo sistema de representação adotado. A partir das representações em ASCII, WAV ou RGB, por exemplo, outras entidades são concebidas formando grânulos mais complexos que contribuem para o ganho de informação semântica.

A definição do processo de granularidade está relacionada com o objetivo de classificação do conjunto de objetos, ou seja, com o viés (*bias*) a ser aprendido pela aplicação através da definição de estruturas semânticas elementares e essenciais para a representação e comparação dos objetos [88]. A escolha dos tipos de grânulos pode nortear o algoritmo de concepção do modelo de aprendizado, na direção adequada, de forma a tomar as decisões alinhadas com a identificação de semelhanças e diferenças entre os objetos. Idealmente, a granulação precisa organizar seu conjunto de grânulos de forma a não tornar o modelo sensível a ruídos ou a dados específicos que levam em conta características que não são importantes para as tarefas como comparação e classificação dos objetos em reconhecimento de padrões.

O objetivo da granulação é conceber uma representação estruturada do objeto de dados a partir de uma máquina de estados. Este tipo de formalismo possibilita versatilidade na representação dos objetos e a utilização de métricas de compressão para comparar os objetos. A granulação pode ser dividida em duas etapas, a análise léxica e a análise sintática.

2.5.1 Análise léxica de uma fonte de dados texto

A análise léxica, também conhecida como *scanner* ou leitura, visa agrupar os grânulos de uma fonte de dados [8]. A sequência de grânulos é processada pela análise sintática. Torna-se necessário a construção de uma tabela de grânulos que possibilite que o analisador sintático consiga inserir informações como, por exemplo, a probabilidade daquele grânulo dentro da fonte de dados, ou seja, informações quantificadas em geral. Em termos de implementação, a análise léxica normalmente é uma sub-rotina da análise sintática formando um único passo. Porém, a divisão conceitual promove a modularizarão da granulação. A tabela de grânulos com a definição de todos os possíveis lexemas de um objeto de dados pode ser concebida por uma linguagem do *tipo 3* (ver seção 2.6.6), por exemplo, uma expressão regular [9]. Os grânulos elementares de uma fonte compõem o nível atômico dos objetos de dados como: caractere para texto (ASCII), pixel para imagem (RGB) e amplitude para áudio (WAV).

Geração de dicionários de *strings*

Os algoritmos de compressão LZ77 [53] e LZW [53] realizam buscas a "substrings"que são repetidas dentro de uma *string*. Defini-se que o termo "janela móvel"que significa que, para qualquer ponto dos dados de entrada, há um registro dos caracteres que vieram antes na *string*. Uma janela móvel de 32K significa que o algoritmo tem um registro dos últimos 32768 (32×1024) caracteres. Quando a próxima sequência de caracteres a ser comprimida é idêntica a uma que pode ser encontrada dentro da janela móvel, a sequência de caracteres é substituída por dois números: uma distância (que representa a distância em direção contrária da *string* de entrada da janela móvel da sequência inicial) e um comprimento (que representa o número de caracteres cuja a sequência é idêntica).

No caso de uma fonte de dados texto por exemplo, a granulação pode conter grânulos na forma de palavras, sendo assim, análise sintática a partir das transições de estados do modelo pode "saltar" por unidades de palavras (*string*), como ocorre no modelo de analisador léxico dos algoritmos LZ77 e LZW. Neste caso, a granularidade aumenta seu nível de resolução para grânulos de informação mais complexos e dependentes.

2.5.2 Análise léxica de uma fonte de dados imagem

A análise léxica de um objeto de imagem envolve propriedades importantes como a localização espacial dos pixels. A figura 2.2 exibe uma imagem contínua f que é convertida em formato digital. Para isso, é necessário realizar a amostragem da função de imagem tanto nas coordenadas x e y quanto na amplitude. A digitalização dos valores de coordenadas é chamada de amostragem. A digitalização dos valores de amplitude é chamada de quantização [2]. A quantização é uma técnica utilizada na compressão de imagens, áudios e vídeos.

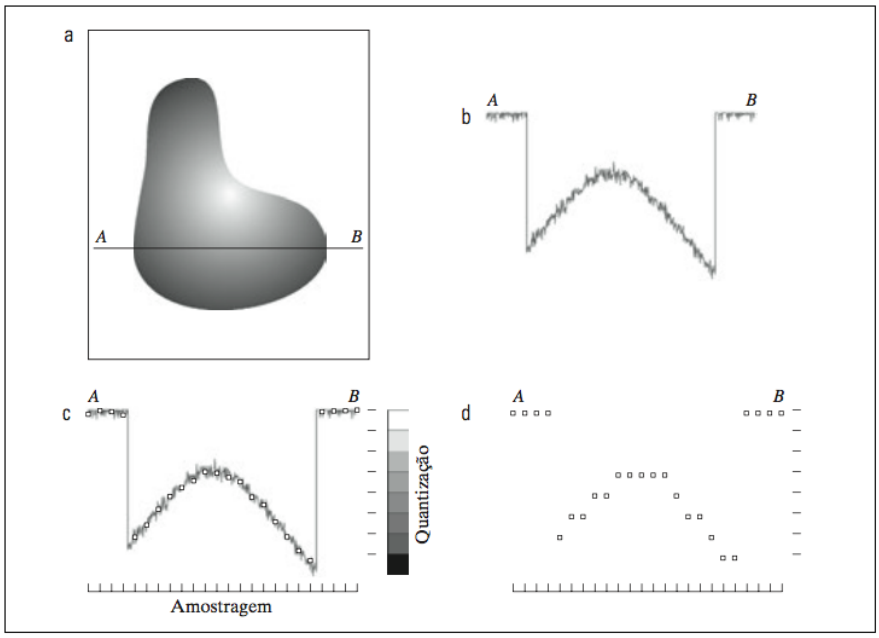

Figura 2.2: *Quantização como técnica de compressão*

A figura 2.2(b) mostra o gráfico que representa os valores de amplitude da imagem contínua ao longo do segmento de reta AB da figura 2.2(a). Para realizar a amostragem, foram coletadas amostras igualmente espaçadas ao longo da linha AB, exibidas na figura 2.2(c). A posição de cada amostra no espaço é indicada por uma pequena marca vertical na parte inferior da figura. O conjunto, dessas localizações discretas, concede a função de amostragem, porém os valores das amostras ainda cobrem uma faixa contínua de valores de intensidade.

Para formar uma função digital, é necessário converter também os valores de intensidade (quantizados). O lado direito da figura 2.2(c) mostra uma escala de intensidade divido em oito intervalos discretos, variando do preto ao branco. As marcas verticais indicam o valor específico atribuído a cada um dos oitos níveis de intensidade. Os níveis de intensidade contínuos são quantizados atribuindo um dos oito valores para cada amostra. Essa atribuição é feita dependendo da proximidade vertical de uma amostra a uma marca indicadora. As amostras digitais, resultantes da amostragem e da quantização, podem ser vistas na figura 2.2(d). Esse processo é feito linha por linha começando na parte superior resultando em uma imagem digital bidimensional. O número de níveis de quantização é discreto.

A figura 2.3(a) apresenta uma imagem contínua projetada sobre o plano de uma matriz. A figura 2.3(b), exibe a imagem após a amostragem e a quantização.

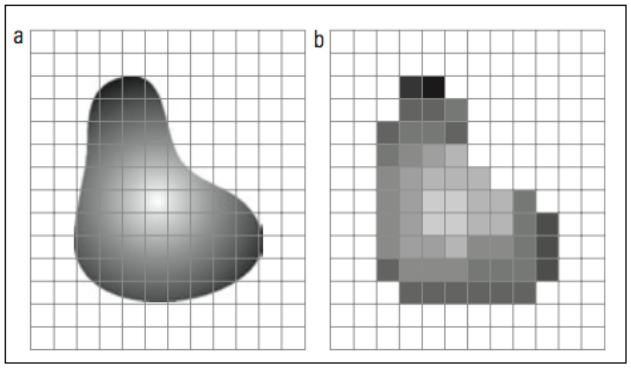

Figura 2.3: *Resultado da quantização*

2.5.3 Análise sintática de uma fonte de dados

A análise sintática, também conhecida como parser [45], tem como tarefa principal, determinar se o fluxo de grânulos identificados pela análise léxica possui as sentenças válidas para o contexto de classificação desejado. Além disso, a análise sintática pode quantificar a recorrência dos grânulos e atribuir pesos para os grânulos. A análise sintática pode utilizar linguagens de *tipo 2* (ver seção 2.6.6) como, por exemplo, as gramáticas livres de contexto para especificar a sintaxe e a atribuição de pesos semânticos para grânulos de um objeto.

Na perspectiva da representação dos objetos de dados por máquinas de estados, o analisador léxico concebe o conjunto de estados da máquina e o analisador sintático verifica as transições de estados definidas pelas dependências entre os grânulos do objeto.

Codificação "Run-length"

A codificação "Run-length"(ou RLE) [91] é uma técnica para compactação de objetos na forma de cadeias de caracteres onde existem sequências longas de caracteres repetidos. A compactação de objetos de dados é realizada por um compactador e consiste da síntese de informação contida dentro do objeto expressa por um valor numérico. A principal diretiva do compactador é realizar o processo de estruturação de objetos transformando fontes de dados não-estruturados ou semiestruturados em fontes de objetos estruturados. As características dos objetos de dados podem ser as mesmas para cada um dos objetos, mas os seus valores nominais $v_i \in V$ podem diferir. Se o conjunto de características $p \in P$ de cada objeto $o \in O$ contiver somente valores $x \in \mathbb{N}$ para V, então os objetos são ditos discretizados ou quantizados. Caso $x \in \mathbb{R}$ para V, os objetos são ditos contínuos.

Modelos de dependência de informação

Os modelos de dependência de informação são processos estocásticos capazes de capturar as probabilidades e as dependências estatísticas presentes entre os grânulos de uma fonte de dados. Essas dependências são listadas por Shannon na forma de *transições* em [79] e exibidas logo a seguir.

Dependência de ordem 0

O modelo de dependência de ordem 0 (zero) pode representar objetos com características equiprováveis. Suponha cinco grânulos A, B, C, D, E, que são identificados em um objeto, cada um com probabilidade 0.2 de ocorrência dentro do objeto. Uma sequência de escolhas desta fonte é : A B B D A E E C A C E E B A E E C B C E A D (construída com o uso de um gerador de números aleatórios).

Dependência de ordem 1

Os modelos de dependência de primeira ordem representam as características de um objeto de dados com suas respectivas frequências. A figura 2.4 apresenta um modelo de objeto de dados texto com os grânulos A, B, C, D, E que representam características com probabilidades 0.4, 0.1, 0.2, 0.2, 0.1, respectivamente.

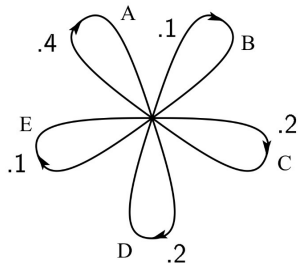

Figura 2.4: *Granulação com dependência de ordem 1*

Um objeto de dados texto que representa a fonte de dados da figura 2.4 pode ser: E A D C A B E D A D D C E C A A A A A D.

Como exemplo de um modelo de ordem 1, assuma um objeto não-estruturado do tipo texto (sist. de representação: ASCII) como: x_i = "AAAAAABBBBBCCC-CDDDEEF" ($i \leq n$) e o projeto de granulação com grânulos na forma de letras da fonte de dados que contém o objeto x_i. O conjunto de características é dado por $P = \{"A","B","C","D","E","F"\}$. As ocorrências de cada característica são contabilizadas como mostra o dicionário da tabela 2.2.

«Características» = Σ	A	B	C	D	E	F
«valores» = Ocorrências	6	5	4	3	2	1

Tabela 2.2: *Contabilização das características*

O vetor de dados $o_i(x_i) = [6,5,4,3,2,1]^T$ (figura 2.5) cujas características são o vetor: $[”A”,”B”,”C”,”D”,”E”,”F”]^T$ é o resultado da compactação do objeto x_i concebendo o dicionário que representa o objeto de dados discretizado o_i. A apresentação do objeto o_i é realizada por um gráfico estrela com as características representadas e valoradas, proporcionalmente, de acordo com os valores da tabela 2.2.

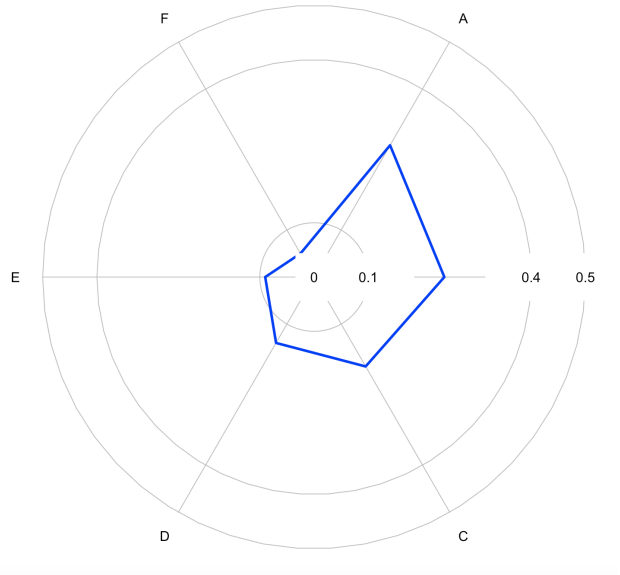

Figura 2.5: *Apresentação do objeto de dados o_i*

Observe outro texto: x_j = "a minha mãe pegou a vassoura e limpou a casa". O vetor de dados $o_j(x_j) = [7,3,2,1,1,2,2,1,3,3,1,3,1,1,2,1,9]^T$ (figura 2.6) cujas características são o vetor $[”a”,”m”,”i”,”n”,”h”,”e”,”p”,”g”,”o”,”u”,”v”,”s”,”r”,”l”,”c”,””,” ”]^T$ é o resultado da compactação do objeto x_j com dependência da informação 1.

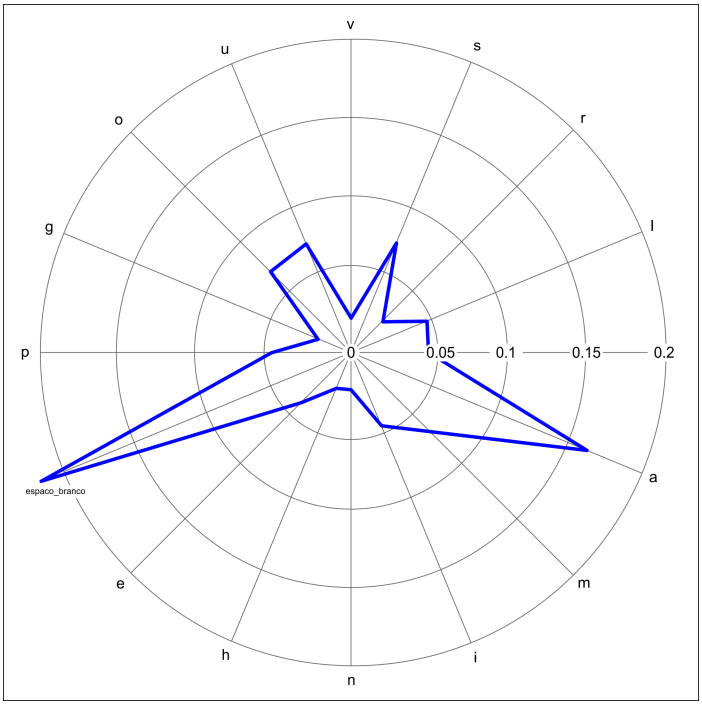

Figura 2.6: *Apresentação do objeto de dados o_j*

A apresentação do objeto o_j é realizada por um gráfico estrela com as características representadas e valoradas, proporcionalmente, de acordo com os valores do objeto.

Outro exemplo de estruturação pode ser ilustrado por um projeto diferente de granulação que modela cada grânulo da sequência como uma palavra (análise léxica). Essa abordagem utiliza como estrutura elementar de informação a palavra ("string"), como pode ser vista na tabela 2.3. A sequência: x_j = "a minha mãe pegou a vassoura e limpou a casa" representa um objeto e cada lexema, agora mapeado não mais por letras, mas por palavras, como sendo, o grânulo de informação que representa uma característica p_i. Cada característica p_i pode ser determinada pela execução do algoritmo do analisador léxico do compactador LZ77. Esse princípio de compactação é utilizado pelo compactador *gzip*.

«Características» = Σ	a	minha	mãe	pegou	vassoura	e	limpou	casa
«valores» = Ocorrências	3	1	1	1	1	1	1	1

Tabela 2.3: *Contabilização das características em strings*

O vetor $o_3 = [3,1,1,1,1,1,1,1]^T$ para $o_3 \in U$ é um objeto cujo os grânulos são: *p = {"a", "minha", "mãe", "pegou", "vassoura", "e", "limpou", "casa"}*, sendo $p : U \rightarrow V_p$, tal que $V_p \in \mathbb{N}$.

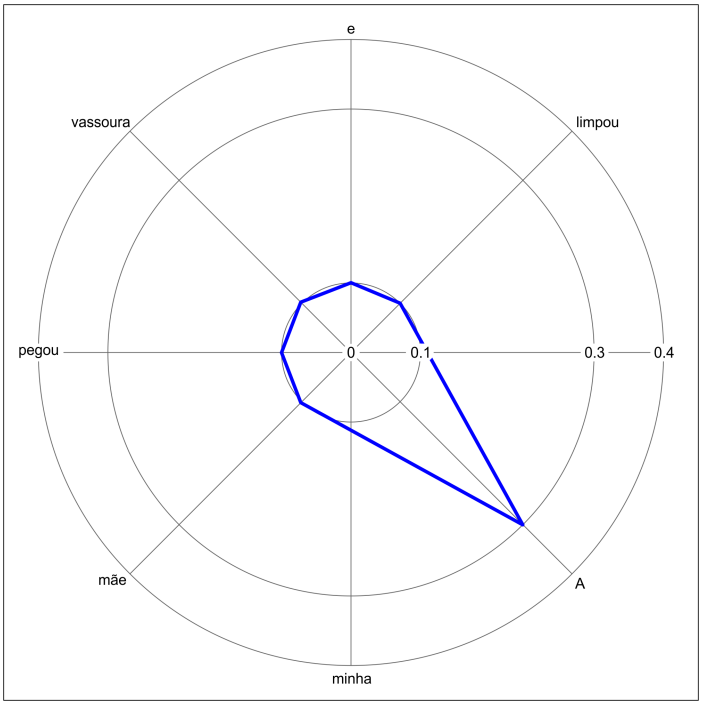

Figura 2.7: *Apresentação do objeto de dados o_3*

Uma fonte de objetos de dados do tipo imagem é apresentada na figura 2.8. O objeto não-estruturado x_i do tipo imagem que possui 256 x 256 pixels e um sistema de representação RGB. O conteúdo da imagem contém um quadrado de lado l = 85 pixels e espessura da linha de 1 (um) pixel como exibido na figura 2.8.

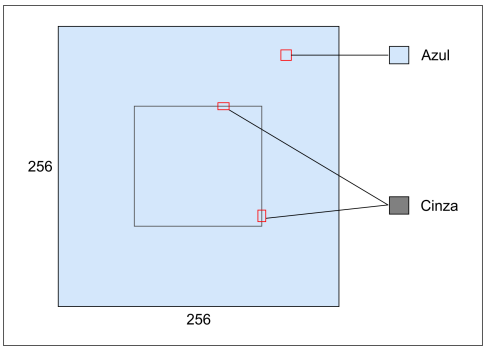

Figura 2.8: *Imagem com dependência da informação de ordem 1*

O conjunto de características da imagem da figura 2.8 é dado por P = {"azul","cinza"} para granulação com dependência da informação de ordem 1. As ocorrências de cada característica com dependência de ordem 1 são contabilizadas como mostra o dicionário da tabela 2.4.

«Características» = Σ	cinza	azul
«valores» = Ocorrências	340	65196

Tabela 2.4: *Características de uma imagem com dependência da informação = 1*

Dependência de ordem 2

Os modelos de dependência da informação de segunda ordem caracterizam-se pela identificação de *digramas* e pelas frequências de ocorrência de cada digrama. Uma estrutura aproximada mais complexa é obtida se as características em objetos texto não forem escolhidas de forma independente, como caracteres isolados, mas se suas probabilidades dependerem de representações anteriores. No caso mais simples, uma escolha depende apenas da representação anterior, e não de outros antes. A estrutura estatística pode, então, ser descrita por um conjunto de probabilidades de transição $P_i(j)$, sendo a probabilidade de que a representação i seja seguida pela representação j como mostrado na figura 2.9.

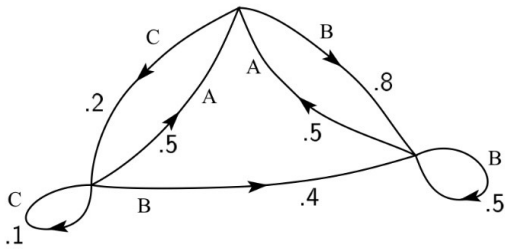

Figura 2.9: *Granulação com dependência de ordem 2*

Os índices i e j variam em todas as representações possíveis. Uma segunda maneira equivalente de especificar a estrutura de probabilidades é através de um digrama onde $p(i,j)$ é a frequência relativa dos *digramas* ij. As frequências das representações $p(i)$, (a probabilidade da representação i), as probabilidades de transição $Pi(j)$ e as probabilidades de diagrama $p(i,j)$ estão relacionadas pelas seguintes fórmulas:

$$p(i) = \sum_j \mathrm{p}(i,j) = \sum_j \mathrm{p}(j,i) = \sum_j \mathrm{p}(j)p_j(i) \tag{2-7}$$

$$p(i,j) = p(i)p_i(j) \tag{2-8}$$

$$\sum_j \mathrm{p}_i(j) = \sum_i \mathrm{p}(i) = \sum_{i,j} \mathrm{p}(i,j) = 1 \tag{2-9}$$

A dependência da informação de segunda ordem é característica do Método de *Burrows-Wheeler Transformation* (BWT) [46]. Trata-se de um processamento estatístico de um bloco de dados que aumenta a redundância espacial, facilitando a aplicação de técnicas de compressão de dados. Diferente de outras técnicas usadas para a compressão de dados que trabalham com um fluxo contínuo de bytes tratados um a um, o método de Burrows-Wheeler trabalha com blocos geralmente grandes de dados que são processados em conjunto antes de serem comprimidos.

Por exemplo, a partir do objeto x_i = "AAAAAABBBBBCCCCDDDEEF" ($i \leq n$) e o modelo de dependência de informação de ordem 2, onde a estrutura elementar de informação é o caractere o vetor de dados $o_v(x_i) = [5,1,4,1,3,1,2,1,1,1]^T$ cujas características são o vetor: ["A $\rightarrow$ A", "A $\rightarrow$ B", "B $\rightarrow$ B", "B $\rightarrow$ C", "C $\rightarrow$ C", "C $\rightarrow$ D", "D $\rightarrow$ D", "D $\rightarrow$ E", "E $\rightarrow$ E", "E $\rightarrow$ F"] é o resultado da compactação do objeto x_i com dependência da informação 2 e grânulos na forma de caracteres.

Para o exemplo utilizando o objeto x_j = "a minha mãe pegou a vassoura e limpou a casa" e a dependência de informação de ordem 2, onde a estrutura elementar de informação é a palavra, o vetor de dados $o_y(x_j) = [1,1,1,1,1,1,1,1]^T$ cujas características são: ["A $\rightarrow$ minha", "minha $\rightarrow$ mãe", "pegou $\rightarrow$ a", "a $\rightarrow$ vassoura", "vassoura $\rightarrow$ e", "e $\rightarrow$ limpou", "limpou $\rightarrow$ a", "a $\rightarrow$ casa"] é o resultado da compactação do objeto x_j com dependência da informação 2 e grânulos na forma de palavras.

No exemplo utilizando imagens digitais, a partir de cada pixel do objeto de imagem da figura 2.10, pode-se visualizar sua vizinhança e contabilizar as relações de localidade que são importantes na definição de uma imagem. Suponha que a varredura da imagem seja feita pelas linhas da imagem (de cima para baixo e da esquerda para a direita). O conjunto de características é dado por $P = \{$"*azul* $\rightarrow$ *azul*","*azul* $\rightarrow$ *cinza*","*cinza* $\rightarrow$ *cinza*","*cinza* $\rightarrow$ *azul*"$\}$ para dependência da informação de ordem 2.

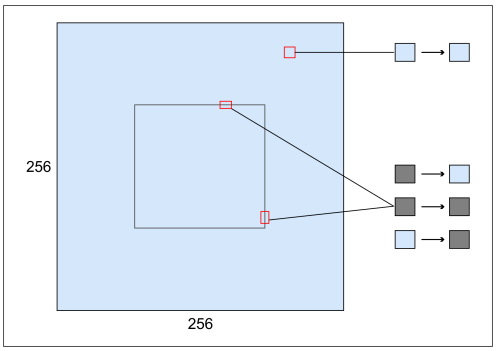

Figura 2.10: *Imagem com modelo de dependência de ordem 2*

As ocorrências de cada característica, com dependência 2, são contabilizadas

como mostra o dicionário da tabela 2.5.

«Características» = Σ	azul→cinza	cinza→cinza	cinza→azul	azul→azul
«valores» = Ocorrências	168	168	168	65032

Tabela 2.5: *Características de uma imagem com dependência = 2*

Dependência de ordem N

As dependências de ordem posterior capturam as dependências entre *n-gramas* com as frequências de ocorrência de cada *n-gramas*. A próxima escala de complexidade envolve as frequências de *trigramas*. A escolha de uma representação dependeria das duas representações anteriores, mas não das representações antes desse ponto. Para um conjunto de frequências de trigrafos $p(i, j, k)$, seria necessário um conjunto de probabilidades de transição $p_{ij}(k)$. Continuando assim, obtém-se sucessivamente processos estocásticos mais complexos que podem ser modelados para o caso geral do *n-grama* através de um conjunto de probabilidades $p(i_1, i_2, ..., i_n)$ ou probabilidades das transições $p_{i_1,i_2,...,i_{n-1}}(i_n)$.

Sistemas L

Além das dependências de ordem N já listadas, a análise sintática pode ser expressa por um sistema de reescrita (sistema L). Um sistema L ou Lindenmayer [86] é um sistema de reescrita paralelo e um tipo de gramática formal. Um sistema L consiste em um alfabeto de símbolos que podem ser usados para conceber palavras, uma coleção de regras de produção que expandem cada símbolo em uma sequência maior de símbolos e uma sequência inicial de *axiomas* a partir do qual iniciam a concepção de um mecanismo para tradução das palavras gerando estruturas geométricas. Lindenmayer utilizou sistemas L para descrever o comportamento das células vegetais e modelar os processos de crescimento do desenvolvimento das plantas. Os sistemas L também foram usados para modelar a morfologia de uma variedade de organismos e podem ser usados para gerar fractais auto-similares.

Por exemplo, em imagens, um sistema de reescrita (sistema L) pode descrever uma curva de Hilbert [21]. A curva de Hilbert é uma curva de preenchimento do espaço e pode ser muito útil no reconhecimento de padrões pelo fato de possuir propriedades de agrupamento. Se "esticarmos" a curva de Hilbert, os pontos que estão próximos no layout bidimensional também tenderão a estar próximos na sequência linear (Apenas tendem a estar próximos porque não se pode fixar esta premissa como uma solução ótima).

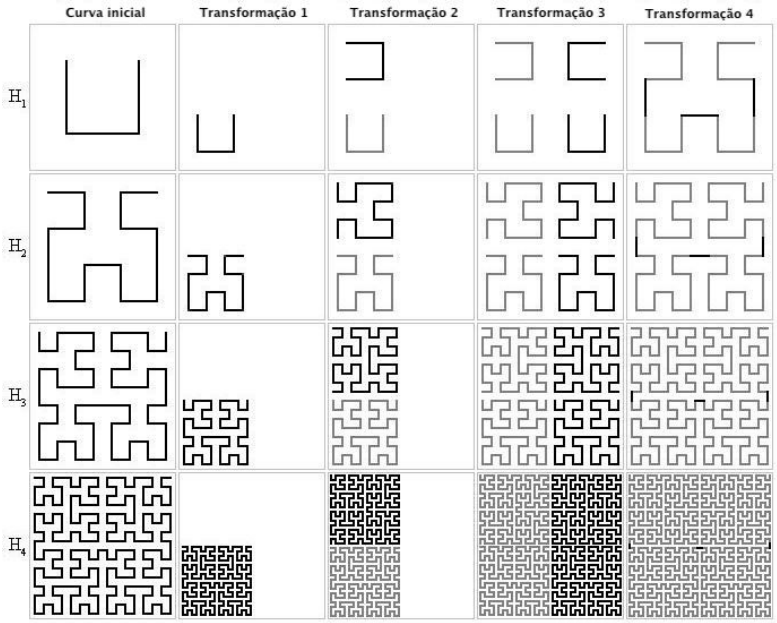

Figura 2.11: *Conjunto de curvas de Hilbert de ordem H1 a H4.*

Pode-se mostrar que qualquer curva desse tipo terá alguns pontos próximos uns dos outros espacialmente, mas longe um do outro na curva. O comportamento de agrupamento concebido pela curva de Hilbert (figura 2.11) pode promover boa capacidade de separação de objetos em uma cena de imagem. Para um exemplo de como essa propriedade pode ser útil, imagine que temos um banco de dados com dois índices: x e y. Sabe-se que são realizadas consultas frequentes sobre esses índices, requisitando registros em que x e y estejam dentro dos intervalos especificados. Pode-se reduzir esse problema a recuperar regiões retangulares de um espaço bidimensional. Dado este cenário, como pode-se dispor os registros no disco para minimizar o acesso ao disco? As informações no disco são armazenadas sequencialmente, portanto, o que se deseja é uma esquema que maximize a probabilidade de que os registros em qualquer região retangular específica também estejam adjacentes no disco. Em outras palavras, o que se deseja é uma maneira de ordenar o espaço bidimensional de registros, de modo que os registros próximos uns dos outros em duas dimensões também tendam a estar próximos uns dos outros na ordem sequencial (fita). Essa é exatamente a propriedade valiosa da curva de Hilbert, portanto uma solução é armazenar os registros em disco na ordem expressa pela curva de Hilbert.

A curva de Hilbert pode ser expressa por um sistema de reescrita (sistema L) da seguinte forma [86]:

Alfabeto: A, B

Constantes: F + -

Axioma: A

Regras de produção:

A $\rightarrow$ - B F + A F A + F B -

B → + A F - B F B - F A +

Aqui, "F" significa "avançar" , -" significa "virar à esquerda 90°" , "+" significa "virar à direita 90°" , conforme os ângulos descritos na figura 2.12, e "A" e "B" são ignorados durante o percurso da curva no desenho.

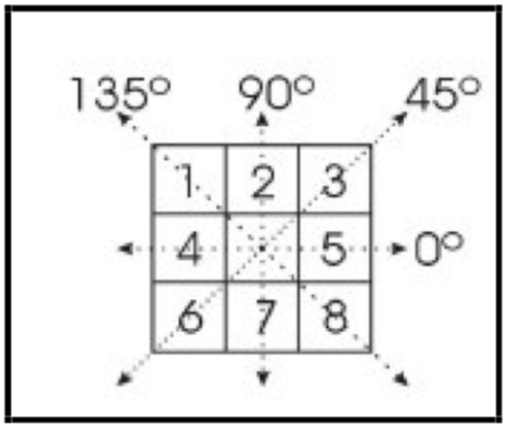

Figura 2.12: *Ângulos de direção para percorrimento de uma imagem diigtal*

A figura 2.13 mostra o percurso de uma curva de Hilbert de ordem 6 (seis) por um sistema L em uma imagem de resolução (64 X 64) pixels. Nota-se na imagem da figura 2.13 alguns aglomerados de pixels representando objetos em uma cena. Vale a pena notar que as relações de localidade da imagem ficam mais preservadas no processo de transformação do objeto de 2 (duas) dimensões pra sua representação linear em 1 (uma) dimensão.

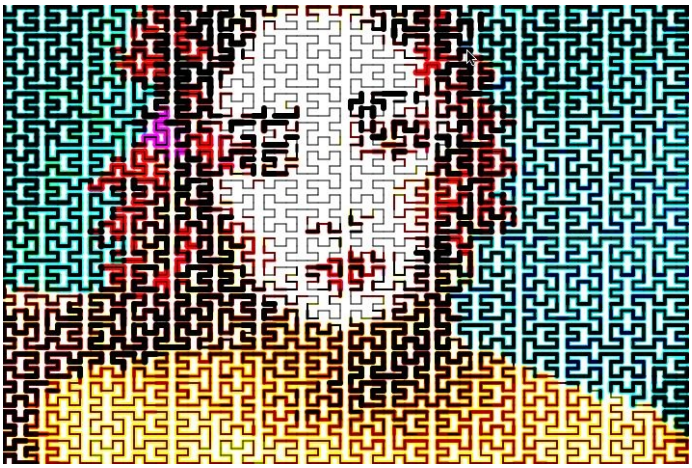

Figura 2.13: *Curva de Hilbert de ordem 6 (seis) percorrida por um sistema L em uma imagem de resolução (64 X 64) pixels.*

Utilizando analisadores sintáticos como sistemas L ou fractais em geral pode-se mapear melhor as propriedades de localidade que uma imagem possui e utilizar estas características para o reconhecimento de padrões utilizando as técnicas de compressão.

2.5.4 Compressão de objetos de dados

Um compressor é um processo composto de práticas para compressão de dados capazes de gerar representações de menor tamanho, em bits, para cada objeto de dados. Na literatura, a motivação inicial do uso de compressores foi a redução de custos na transmissão de dados. Por outro lado, esta pesquisa propõe utilizar compressores para a representação e medição de objetos a fim de realizar o reconhecimento de padrões na fonte de dados. Do ponto de vista da transmissão de dados, o processo de reduzir o tamanho de um objeto é a compressão, já na perspectiva do reconhecimento de padrões, a compressão é utilizada como forma de abstração. A abstração é importante em processos de generalização de objetos. Utilizando técnicas de compressão em reconhecimento de padrões, pode-se representar os objetos de dados de forma mais abstrata, desprezando detalhes irrelevantes ou eventuais ruídos em uma análise de dados. A tabela 2.6 exibe alguns processos de compressão a partir de fontes com objetos não-estruturados do tipo texto e imagem:

Tipo	Dependência	Léxico	Sintático	Processo
texto	1	caractere	RLE	compactação RLE
texto	2	digrama	RLE	compactação BWT
texto	1	palavra	RLE	compactação Gzip
imagem	1	pixel	RLE	compactação RLE de imagem
imagem	1	quantização	RLE	compressão de imagem com compactação RLE
imagem	1	quantização	RLE + Sistema L	compressão de imagem com compactação RLE por localidade

Tabela 2.6: *Processos de compressão*

Os processos de compressão podem ser implementados por um algoritmo na forma de uma máquina de Turing (MT). A MT, que representa cada processo de compressão é descrita na tabela 2.6, pode ser implementada, tendo seu conjunto de estados finitos definidos a partir do analisador léxico que retorna os grânulos equivalentes as transições da máquina e o analisador sintático determinado como as transições são realizadas entre

os estados. A concepção da MT, sua codificação e o conjunto de métricas disponíveis são descritas na seção 2.6.

2.5.5 Concatenação de objetos de dados

A concatenação (+) é a operação realizada em objetos de dados texto capaz de encadear o conteúdo de dois textos sequencialmente. Por exemplo, considerando as palavras "casa"e "mento"a concatenação da primeira com a segunda, gera a palavra "casamento". A operação de concatenação é realizada entre objetos de dados. Diversas linguagens de programação fornecem operadores binários para a concatenação. A linguagem Python, por exemplo, oferece o operador + para a concatenação. Uma forma válida em Python de se representar o exemplo anterior seria: print "casa"+ "mento".

Vale a pena lembrar que a operação de concatenação não é simétrica, ou seja, x + y $\neq$ y + x para x $\neq$ y.

2.5.6 Combinação de objetos de dados

A operação de combinação ($\oplus$) é realizada entre objetos compactados. Dados 2 (dois) objetos compactados, um exemplo de combinação é apresentado pela tabela 2.7.

Conjunto de objetos			
«grânulos» = Σ	A	B	C
Objeto o_1	1	25	0
Objeto o_2	14	2	6
combinação ($o_1 \oplus o_2$)	15	27	6

Tabela 2.7: *Objetos o_1 e o_2*

A operação de combinação é simétrica, ou seja, x $\oplus$ y = y $\oplus$ x para x $\neq$ y.

2.6 Codificação de objetos

A codificação é o processo pelo qual um objeto compactado é transformado em uma chave única de identificação (representação única), que serve como medição para a distinção do próprio objeto perante os demais [1]. Os processos de medição são realizados a partir das propriedades emergentes dos processos de compressão.

2.6.1 Princípios para representação e medição de informações

Diversas heurísticas e princípios podem ser utilizados na representação de informação. Um princípio muito utilizado é a evolução mínima [56]. Trata-se de um princípio científico e filosófico que propõe que entre hipóteses formuladas sobre as mesmas evidências, é mais racional acreditar na mais simples. Ou seja, diante de várias explicações para um problema, a mais simples tende a ser a mais correta. O filósofo inglês William de Occam (1285-1347) não foi o primeiro a formular isso. Aristóteles já fazia o mesmo no século 4 a.C.. Já o termo “navalha” ou “lâmina” é uma metáfora que surgiu muito depois dele: sugere que, com o uso da parcimônia, a hipótese mais complicada seja desconsiderada.

A navalha de Occam é também chamada de princípio da parcimônia e interpretada como "a explicação mais simples é a melhor" ou "não multiplique hipóteses desnecessariamente". A navalha de Occam é o princípio utilizado da ontologia, isto é, por filósofos da ciência num esforço de estabelecer critérios para escolher entre várias teorias com igual valor explicatório. Ao fornecer razões explicativas para algo, não postule mais que o necessário. As representações dos objetos e as comparações par a par são calculadas e norteados com base no princípio de evolução mínima.

2.6.2 Unidade básica de informação compressão (”bit”)

A fim de entender de forma intuitiva o que significa dizer 1 bit de informação comprimida, imagine a seguinte situação: um rota é definida partindo do ponto marcado com a letra ”*A*” na figura 2.14 até o destino no ponto "D".

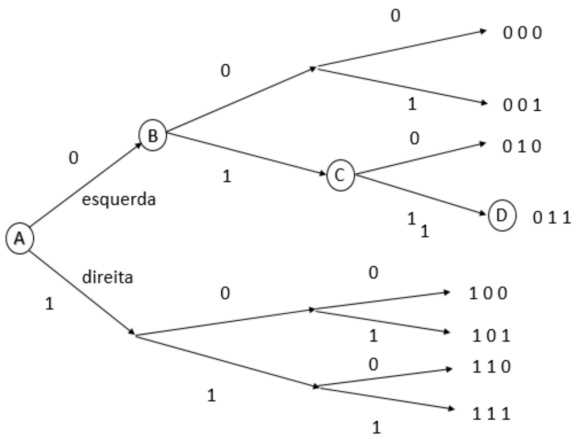

Figura 2.14: *Conjunto de caminhos. Em cada bifurcação recebe-se a instrução esquerda (0) direita (1) até alcançar seu destino final.*

O caminho entre ”*A*” e ”*D*”, possui várias bifurcações (como os pontos ”*B*” e ”*C*”). Como desconhece-se o caminho de cada ponto percorrido (representado pelas

bifurcações), torna-se necessário a busca de informação, para a decisão de seguir à direita ou à esquerda. Na figura 2.14, dizer que se deve seguir à esquerda é o mesmo que mostrar o dígito binário 0, e a direita o dígito 1.

Dessa forma, deve-se obter (pedir) informação nos pontos "A", "B" e "C" para alcançar o destino "D". Note que independente do destino ($\{000, 001, 011, ...\}$) o número de perguntas realizadas para alcançá-lo, nesse caso, é sempre 3 (três). Em outras palavras, escolher entre oito destinos requer três perguntas: 2^3 destinos.

Note que o expoente do número dois é igual ao número de perguntas feitas. Defini-se então que, para escolher entre um dos oito possíveis destinos, é necessário uma quantidade de informação igual a 3 bits.

Aplicando o logaritmo de base dois na expressão anterior, temos:

$$3 = \log_2 8 \quad [\text{bits}] \tag{2-10}$$

É importante salientar que se pode escolher qualquer um dos destinos finais possíveis, mas são todos equiprováveis, com probabilidade $p = \frac{1}{8}$.

De forma análoga, para o caso em que se tem m possíveis destinos, e supondo que o viajante possa escolher qualquer um deles com igual probabilidade, a quantidade de informação, em "bits", para alcançar um dos possíveis destinos é dada pela relação a seguir:

$$n = \log_2 m \quad [\text{bits}] \tag{2-11}$$

A partir da equação 2-11, conclui-se que n bits é a informação necessária para se escolher entre m alternativas equiprováveis, ou $1bit$ é a quantidade de informação necessária para escolher entre duas alternativas equiprováveis.

$$n = \log_2 2 = 1 \, [\text{bit}] \tag{2-12}$$

2.6.3 Unidade de medida da informação

Uma fonte de informação discreta pode representar um processo de Markov e vice-versa. Partindo da análise de uma fonte de informação discreta como um processo de Markov, pode-se definir uma quantidade que mede quanta informação é produzida por esse processo, e a que taxa a informação é produzida.

Suponha um conjunto de eventos possíveis representados por objetos compactados cujas probabilidades de ocorrência são p_1, p_2, .., p_n. Essas probabilidades são as únicas informações sobre os eventos, desta forma, pode-se encontrar uma *medida* de quanta *escolha* E_s está envolvida na seleção do evento ou de quão incerto é a informação do evento. Seja E_s (p_1, p_2, .., p_n) tal *medida* com as seguintes propriedades:

- E_s deve ser contínua em p_i.
- Se todos os p_i forem iguais, $p_i = \frac{1}{n}$, então, E_s deve ser uma função de aumento monotônico de n. Dados n eventos equiprováveis há mais opções, ou incerteza, quanto há mais eventos possíveis.
- Se uma escolha E_s for dividida em duas escolhas sucessivas, a E_s original deve ser a soma ponderada dos valores individuais de E_s como ilustrado na figura 2.15.

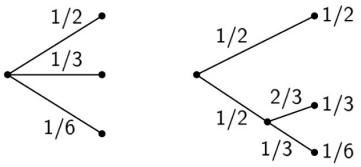

Figura 2.15: *Decomposição de uma escolha de três possibilidades*

À esquerda, tem-se três probabilidades $p_1 = \frac{1}{2}$, $p_2 = \frac{1}{3}$, $p_3 = \frac{1}{6}$. À direita, escolhemos duas possibilidades cada uma com probabilidade $\frac{1}{2}$, e se a segunda ocorrer, faça outra escolha com probabilidades $\frac{2}{3}$, $\frac{1}{3}$. Os resultados finais têm as mesmas probabilidades que antes. Seja I_n a incerteza dada pelo conjunto de escolhas E_s, então:

$$I_n = E_s(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}) = E_s(\frac{1}{2}, \frac{1}{2}) + \frac{1}{2}E_s(\frac{2}{3}, \frac{1}{3}) \tag{2-13}$$

O coeficiente $\frac{1}{2}$ reflete a segunda escolha que ocorre apenas metade do tempo.

2.6.4 Incerteza da informação de um objeto

Shannon em [79] associa o conceito de redundância ao de incerteza da informação de forma a relacioná-los como métricas complementares uma da outra. Do ponto de vista abstrato, a redundância é tudo o que não é fundamental para o entendimento de uma determinada mensagem, ou seja, trata-se de uma extensão da incerteza [58].

As informações são representadas e encapsuladas em objetos através de características que contêm os grânulos. As ocorrências dos grânulos e suas dependências podem ser contabilizadas em um objeto, gerando, assim, distribuições de probabilidade que caracterizam sua redundância intrínseca. As distribuições de probabilidade de cada objeto são convertidas para valores em uma escala de incerteza a partir dos seguintes princípios:

- a incerteza I_n de um objeto é máxima quando existirem probabilidades equivalentes de ocorrência para todas as características;
- a predição de uma análise de agrupamento não depende unicamente da contagem do número de grânulos de um objeto mas também da contagem de suas dependências;

- um conjunto de objetos com alta incerteza I_n dispõe de um conjunto rico, com grânulos diferenciados, que demostram o poder das combinações. Um conjunto de objetos como pouca incerteza é pobre e repetitivo com relação a seus grânulos;
- a imprevisibilidade total é correspondente à informação nula, ou seja, não há informação;
- sendo um objeto de classificação $A = [a_1, a_2, a_3, \ldots, a_n]^T$, a medida da informação contida em uma característica a_i pode ser calculada a partir da fórmula de Hartley [33]:

$$i(a_i) = \log_b(r) \tag{2-14}$$

onde r é o número de resultados possíveis da variável aleatória a_i representada por um grânulo e b é a unidade da informação (Por exemplo, no caso de informação binária, b = 2);

- o conteúdo de informação e_s (em escolhas ou "bits") de cada grânulo a_i com probabilidade p_i não nula é:

$$e(a_i) = -log_2 p_i \tag{2-15}$$

- já a medida de incerteza de um objeto é dada por [77]:

$$I_n(A) = E_s(A) = -\sum_{i=1}^{n} p_i \log_2(p_i) \tag{2-16}$$

onde p_i é a probabilidade do i-ésimo resultado;

- uma maneira simples de medir a incerteza é atentar-se para duas possibilidades de ocorrência de um evento: p e q, sendo q = 1 - p. Então, a incerteza do sistema é calculada como:

$$I_n(A) = -(p\log_2(p) + q\log_2(q)). \tag{2-17}$$

Veja figura 2.16 .

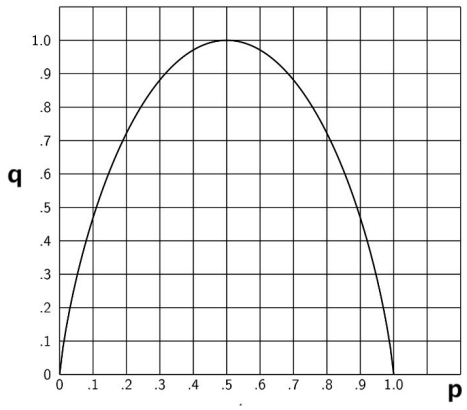

Figura 2.16: *Incerteza I_n de uma fonte de informação*

Outra forma de medir a informação de um objeto pode ser realizada através da descoberta do conteúdo da informação. A partir da medição do conteúdo de informação de um objeto é possível utilizar esta metainformação como métrica para distinção entre objetos.

2.6.5 Certeza da informação de um objeto

A complexidade de Kolmogorov é uma teoria matemática que possui modelos da Teoria da Informação capazes de representar objetos. Dentre esses modelos, destacam-se as máquinas de estado [49] utilizadas para representar conteúdo de informação. A complexidade de Kolmogorov [16] utiliza o conceito de certeza para representar o conteúdo da informação de objetos individuais e sua medição através do tamanho de sua menor descrição algorítmica para um objeto, formando, assim, o conceito de certeza. Através desta complexidade, atribui-se a um objeto um valor numérico pontual numa escala que representa seu respectivo grau de certeza. A complexidade de Kolmogorov é uma forma direta de medida que gera conteúdo da informação para medição de objetos a partir da teoria algorítmica da informação, pois as representações são algoritmos. A complexidade de Kolmogorov é uma subárea derivada da Teoria da Informação de Claude Shannon, embora atualmente possa ser considerada uma área autônoma. A figura 2.17 exibe o arcabouço teórico utilizado pela complexidade de Kolmogorov.

Figura 2.17: *Arcabouço teórico da Complexidade de Kolmogorov*

A Máquina de Turing (MT) [81] é usada como modelo para descrever os objetos e para definir os algoritmos. Desta forma, os objetos são estruturados por algoritmos do tipo máquinas de estado que contêm o poder de representação disponível em linguagens descritas pela Teoria da Computabilidade [27]. A arquitetura da MT é concebida pela complexidade Estado-Simbolo descrita pelos modelos de dependência da informação apresentados nas figuras 2.4 e 2.9.

Para a representação do conteúdo da informação dos objetos são utilizados os seguintes conceitos da complexidade de Kolmogorov:

- a complexidade de Kolmogorov está definida sobre o conjunto de palavras binárias definidas pelas expressões regulares cujo alfabeto $\Sigma = \{0,1\}^*$ e associa a cada palavra binária um valor numérico que é sua complexidade (grau de certeza);
- a complexidade de Kolmogorov pode ser definida como o tamanho do menor programa ou descrição algorítmica, que computa na MT uma determinada sequência binária;
- exige-se que o conjunto de descrições (algoritmos) forme um conjunto livre de prefixo. Denomina-se esta complexidade de *complexidade de prefixo*, representada por $K(x)$;
- visando enfatizar a máquina que está sendo utilizada para definir a complexidade de prefixo pode-se escrever $K_M(x)$ ao invés de $K(x)$, onde M é uma Máquina de Turing em particular ou qualquer uma de suas especializações (Linguagens de tipo 0, 1, 2 ou 3 (Ver seção 2.6.6));
- pode-se definir a *complexidade condicional*, $K(x|y)$, através do tamanho do programa que computa x a partir de y, que é dado como entrada do programa. Intuitivamente, $K(x|y)$ representa a quantidade de informação que se deve adicionar à informação em y para computar x.

2.6.6 Representação de algoritmos por máquinas computacionais

Os objetos com suas respectivas características podem ser representados por máquinas computacionais como proposto pela complexidade de kolmogorov. Na teoria da computabilidade, a MT é uma máquina de estados finitos de representatividade genérica, podendo estabelecer um conjunto de máquinas na forma de especializações que concebem linguagens formais descritivas para algoritmos. Dentre as especializações da MT destacam-se: Máquina NORMA, Máquina POST, Automato de Pilha (AP), Gramática Livre de Contexto (GLC), Automato Finito Determinístico (AFD), Automato Finito Não Determinístico (AFND), Automato Finito com Saída (AFD), Gramática (GR) e Expressões Regulares (ER). Cada uma das máquinas quanto ao seu poder de descrição e representação são classificadas segundo a hierarquia de Chomsky em:

- tipo 0 - Linguagens Enumeráveis Recursivamente;
- tipo 1 - Linguagens Sensíveis ao Contexto;
- tipo 2 - Linguagens Livres de Contexto;
- tipo 3 - Linguagens Regulares.

A Hierarquia de *Chomsky* [50] é a classificação das máquinas de computação pelo linguista *Noam Chomsky*. Esta classificação possui 4 níveis (de 0 a 3), sendo que os dois últimos níveis (os níveis 2 e 3) são amplamente utilizados na descrição de linguagens de programação e na implementação de interpretadores e compiladores. Mais especificamente, o nível 2 é utilizado em análise sintática e o nível 3 em análise léxica.

A classificação das máquinas começa pelo tipo 0, com maior nível de liberdade em suas regras, e aumentam as restrições até o tipo 3. Cada nível é um super conjunto do próximo nível. Logo, uma máquina de tipo n é consequentemente uma máquina de tipo n - 1, ou seja, mais genérica com relação a poder de representação.

A figura 2.18 apresenta as máquinas computacionais de acordo com seu nível de representatividade e poder de descrição.

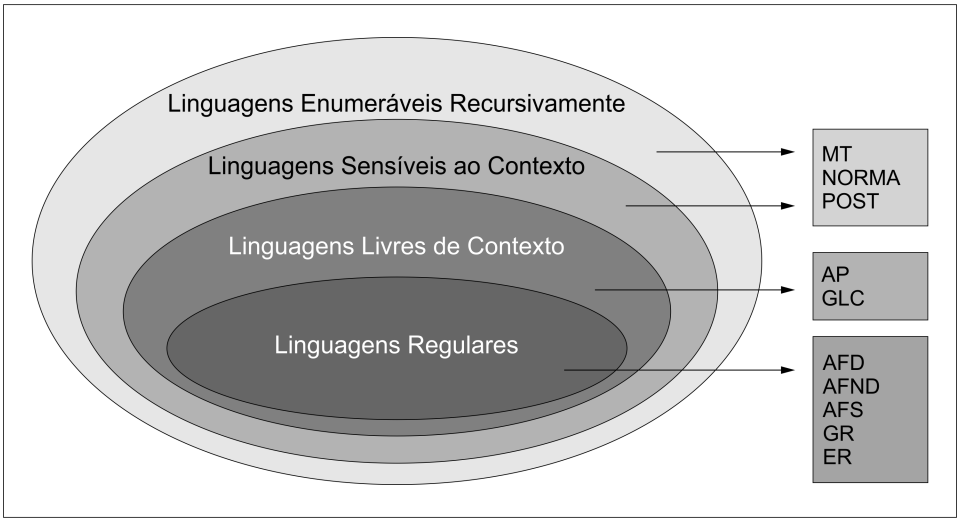

Figura 2.18: *Hierarquia de linguagens de representação X Máquinas computacionais*

2.6.7 Representação formal de um objeto por MT

Para $\Delta = \{0, 1\}$, considere a bijeção de Δ^* com $\mathbb{N}$, onde cada $n \in \mathbb{N}$ corresponde a n-ésima sequência binária na ordem lexicográfica (crescente): (0, ε), (1, 0), (2, 1), (3, 00), (4, 01), (5, 10), (6, 11), (7, 000), ... , (n, 000...). Desta forma, cada sequencia de ”bits” x é indexada por n. O comprimento ou tamanho (número de bits) de x é dado por $|x|$.

Seja f_{bij} uma função bijetora que converte objetos para uma representação na forma de incerteza (Sequências de "bits"), e expressa por um número, o resultado da medição do tamanho sequência x. Sendo assim, de acordo com $K_M(x)$, são necessárias formas fechadas para a bijeção $f_{bij} : \mathbb{N} \rightarrow \Delta$ e para a sua inversa f_{bij}^{-1}. Assim, uma máquina de estados descrita por uma linguagem de tipo 3 pode ser capaz de desempenhar a função de um reconhecedor de linguagens (Autômatos Finito - AF) ou de um tradutor de linguagens (Autômato Finito com Saída - AFS).

Como exemplo da representação de um objeto, seja um AFS na forma de uma máquina de Mealy [50] cujo o formalismo contenha o capacidade computacional de tradução de uma linguagem para outra. Por definição, a máquina de Mealy é um AFS que gera uma palavra de saída, que pode ser vazia, para cada transição da máquina. Numa máquina de Mealy, a saída depende do estado atual da máquina e do valor das entradas, ou seja, das transições da máquina. Sendo Σ um o conjunto de grânulos dos objetos, a máquina de Mealy pode ser definida como uma 6-upla $m = (S, S_0, \Sigma, \Delta, \delta, \delta_s)$ consistindo de:

- um conjunto finito de estados S;
- o estado inicial S_0 que é elemento de S;
- um conjunto finito de grânulos de informação Σ;

- um conjunto chamado de alfabeto de saída $\Delta = \{0, 1\}$;
- a função de transição $\delta : S \times \Sigma \rightarrow S$ mapeando um estado e um grânulo do alfabeto de entrada para o próximo estado;
- a função de saída de dados $\delta_s : S \times \Sigma \rightarrow \Delta^*$ mapeando uma transição para o alfabeto de saída. A função de saída δ_s é uma função bijetora f_{bij}.

Um representação mais estável de objeto pode ser implementada através de uma máquina de Moore. Trata-se também de um AFS que gera uma palavra de saída, que pode ser vazia para cada estado da máquina. Numa máquina de Moore, a saída só depende do estado atual da máquina. A máquina de Moore pode ser definida da mesma forma descrita pela 6-upla da máquina de Mealy, porém com a função de saída de dados δ_s, definida da seguinte forma:

- uma função de saída $\delta_s : S \rightarrow \Delta^*$ mapeando um estado para o alfabeto de saída. A função de saída δ_s é uma função bijetora f_{bij}.

Segundo Kolmogorov, a complexidade de um objeto de dados pode ser expressa por uma função $K_M(o_i)$, onde $o_i = [p_1, p_2, p_3, \ldots, p_n]^T$ $o_i \in U$ é um objeto de agrupamento mapeado a partir de uma fonte de dados, m é algum algoritmo formalizado como uma MT ou uma de suas especializações e $p_1, \ldots, p_n$ são os valores de cada característica do objeto. Em outras palavras, o comprimento da sequência gerada por uma máquina de estados a partir do algoritmo m é a complexidade de Kolmogorov $K_M(o_i)$ para $M = (S, S_0, \Sigma, \Delta, \delta, \delta_s) = f_{bij}$), onde m pode ser uma máquina Mealy ou Moore.

Para a utilização de uma máquina Moore, é necessário que os estados alcancem sua estabilidade. Para adquirir a estabilidade probabilística de um conjunto de estados, torna-se necessário a convergência dos valores de probabilidade de cada estado calculado a partir das transições.

2.6.8 Conteúdo da informação

O conteúdo da informação de um objeto consiste da discriminação dos *bits* que compõem o objeto dispostos em ordem numa fita formando um código C. O primeiro passo para a geração do conteúdo de informação de um objeto é a obtenção da função δ_s de uma forma fechada para a bijeção $f_{bij} : \mathbb{N} \rightarrow \Delta^*$ e para a sua inversa f_{bij}^{-1}. A codificação de prefixo pode ser utilizada para satisfazer as propriedades exigidas pela bijeção f_{bij}.

Um código é classificado como código de prefixo se nenhuma de suas palavras código é prefixo de qualquer outra de suas palavras código [15]. Dado o código binário para os objetos com dependência de ordem 1 apresentados na Tabela 2.8, o objeto <iaaoa> seria codificado como <110111100111>. Percorrendo o objeto codificado da esquerda para a direita, pode-se decodificá-la, por exemplo, como <iaaoa> ou <auaoa>. Diz-se,

então, que este código não é unicamente decodificável. Isto acontece porque a palavra de código que representa a característica do objeto <a> → (11) é também o início da palavra-código que representa a característica <i> → (110). Diz-se que a palavra-código 11 é um prefixo da palavra-código 110 como na tabela 2.8.

«grânulos» = Σ	Palavra de Código
a	11
e	000
i	110
o	001
u	011

Tabela 2.8: *Codificação binária não prefixada*

Considere um código C com palavras de código $c_0, c_1, ..., c_{m-1}$. Se, para $i \neq j$, c_i não é um prefixo de c_j, com $i, j = 0, 1, ..., m$–1, diz-se que C é um código de prefixo. Códigos de prefixo permitem uma decodificação simples e sem ambiguidades. O código apresentado na tabela 2.9 é um código de prefixo.

«grânulos» = Σ	Palavra de Código
a	10
e	000
i	110
o	001
u	011

Tabela 2.9: *Codificação binária de prefixo*

Através da codificação de prefixo, qualquer cadeia de palavras de código pode ser univocamente decodificada, percorrendo-se a cadeia da esquerda para a direita e substituindo-se cada palavra-código encontrada no processo pelo grânulo correspondente. A operação de decodificação equivale a inversa f_{bij}^{-1}.

2.6.9 Geração do conteúdo da informação

A partir do teorema da Codificação de Fonte (TCF) [6], os limites do tamanho de um código de prefixo são delimitados pela entropia da fonte de dados. De acordo com o TCF, o tamanho do código de prefixo deve ser o menor possível para que se aproxime da entropia da fonte. Sendo assim, a geração de conteúdo se reduz a construir um código de prefixo C com o objetivo de diminuir o tamanho de um objeto composto por um conjunto de características $\Sigma = \{p_0, p_1, ..., p_{j-1}\}$, onde a característica p_j aparece n_j vezes. Se a

i-ésima palavra de código possui l_j bits, com $l_j \in \mathbb{N}$, deve-se minimizar o comprimento total em bits do objeto codificado de modo que a diferença, entre o comprimento médio dos elementos, na sequência codificada, aproxime-se da incerteza máxima $I_n max$.

$$L = n_0.l_0 + n_1.l_1 + ... + n_{j-1}.l_{j-1} \tag{2-18}$$

O algoritmo de Huffman [57] representa uma maneira sistemática de construir códigos de prefixo que efetivamente minimize L. Por este motivo, o código de Huffman é denominado código de redundância mínima. Observe que foi colocada na formulação do problema a restrição de utilização de palavras de código com um número inteiro de bits, e este foi o problema específico solucionado por Huffman, porém códigos sem essa restrição podem apresentar redundância menor que o código de Huffman.

O algoritmo de codificação de Huffman associa uma árvore ponderada a cada característica do conjunto de objetos. Inicialmente, cada árvore possui um único nó, com peso igual à probabilidade de ocorrência da característica a ela associada. A cada iteração do algoritmo, as duas árvores de menor peso são substituídas por uma nova árvore cujo peso é a soma dos pesos das primeiras. A árvore de menor peso se torna a subárvore esquerda e a outra se torna a subárvore direita da nova árvore. Na ordenação dos pesos, empates são resolvidos por qualquer regra sistemática. O procedimento para quando resta apenas uma única árvore. A palavra de código para qualquer característica é obtida percorrendo-se esta árvore desde a raiz até a folha correspondente à característica em questão, registrando 0 para cada ramo esquerdo e 1 para cada ramo direito. O código de Huffman é o melhor código de comprimento inteiro possível, e o comprimento médio de suas palavras-código se aproxima da incerteza da informação de um objeto.

Veja um exemplo, passo a passo de f_{bij} = "Huffman", utilizando os seguinte configuração:

- x_i = "AAAAAABBBBBCCCCDDDEEF"
- p = {"A", "B", "C", "D", "E", "F"}, sendo p: O $\rightarrow$ Vp, tal que Vp $\in \mathbb{N}$
- $o_i = [6,5,4,3,2,1]^T$ para $o_i \in O$

No diagrama da figura 2.19, vê-se os nós que representam cada característica e marca-se os dois nós que serão unidos no primeiro passo. No caso os nós E e F:

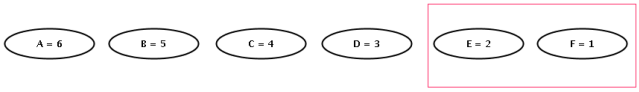

Figura 2.19: *primeiro passo de* f_{bij}

Os nós E e F são unidos num nó que chamamos de E+F, e que tem o peso igual à soma dos nós E e F. Este novo nó é inserido no conjunto de nós dos quais escolhe-se

os próximos nós a serem unidos. Coincidentemente, neste exemplo, os próximos nós são este mesmo novo nó E+F e o nó D, como se vê na figura 2.20:

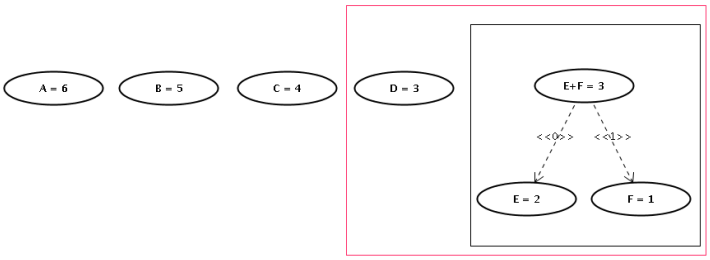

Figura 2.20: *segundo passo de* f_{bij}

Continuando o processo uni-se agora os nós B e C conforme figura 2.21:

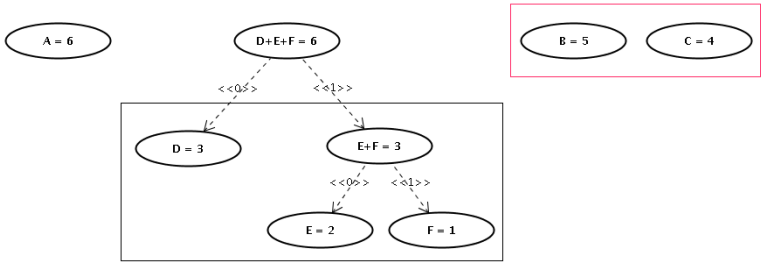

Figura 2.21: *terceiro passo de* f_{bij}

O nó A finalmente será unido com o nó D+E+F e mostrado na firgura 2.22:

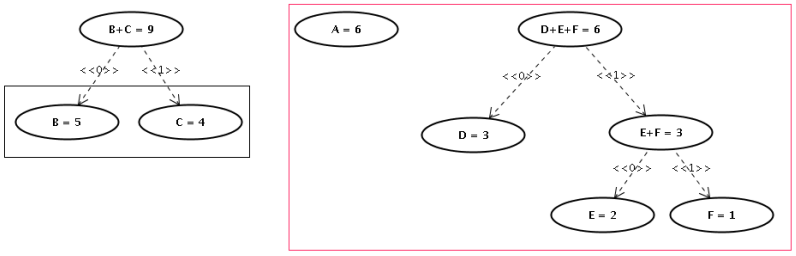

Figura 2.22: *quarto passo de* f_{bij}

Na figura 2.23 a última junção dos nós A+D+E+F com o nó B+C:

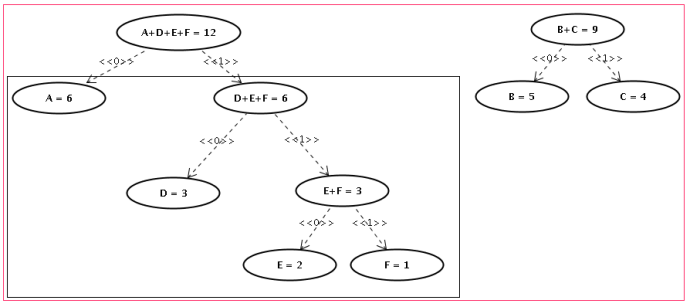

Figura 2.23: *quinto passo de* f_{bij}

Gerando a árvore de Huffman que agora é uma árvore estritamente binária na figura 2.24:

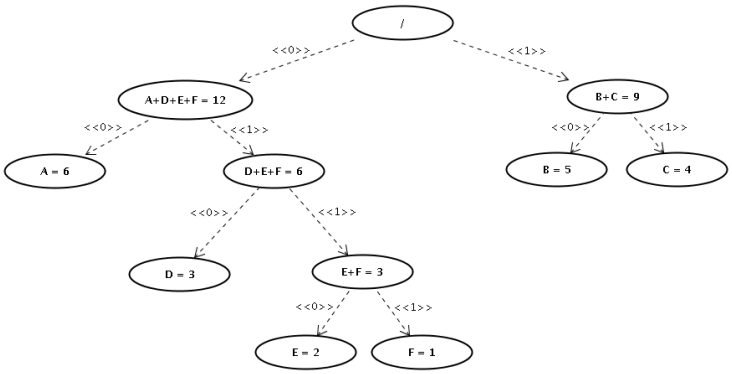

Figura 2.24: *sexto passo de* f_{bij}
Fonte da figura: Wikipédia

Na árvore da figura 2.24, pode-se identificar os códigos para cada característica. Para isto, basta percorrer a árvore até o símbolo e registrar o bit correspondente às arestas que foram percorridas. Por exemplo, para chegar a característica *D* percorremos os bits 0 até o nó A+D+E+F, depois o bit 1 para chegar em D+E+F e depois o bit 0 novamente, chegando a característica *D*. Assim, o código Huffman para a característica *D* será 010. A tabela 2.10 lista os códigos de prefixo ótimo para cada uma das características:

Codificação de prefixo ótima - f_{bij} = (Huffman)						
«palavra de código» $\Delta = \{0,1\}^*$	00	10	11	010	0110	0111
«características» = Σ	A	B	C	D	E	F

Tabela 2.10: *Códigos de prefixo ótimo para cada uma das característica*

Ao codificar o objeto bruto original x_i, tem-se:
000000000000101010101011111111010010010011001100111 totalizando 51 bits, ou seja: $K_M(o_i) = 51$.

2.6.10 Função δ_s da MT

Seja o_i um objeto que contém a probabilidade de cada a característica dada por uma distribuição estacionária τ. Seja uma máquina de Moore definida pela 6-upla (S, S_0, Σ, Δ, δ, δ_s) onde δ_s é a função de saída que retorna palavras em codificação por prefixo.

- Entrada: Objeto o_i sendo $o_i = \{p_1, p_2, \cdots, p_n\}$, o conjunto de a características de tamanho n.
  Sendo $W = \{w_1, w_2, \cdots, w_n\}$, o conjunto de valores positivos (usualmente probabilidades de ocorrência), i.e. $w_i = \tau(o_i), 1 \leq i \leq n$.

- Saída: Código $C(o_i, W) = (c_1, c_2, \cdots, c_n)$, é a tupla de palavras-chave (binárias), onde c_i é a código binário para $p_i, 1 \leq i \leq n$.
- Meta: Seja $Le(C) = \sum_{i=1}^{n} w_i \times \text{length}(c_i)$ o comprimento do caminho ponderado do código C. Condição: $Le(C) \leq Le(T)$ para qualquer código $T(o_i, W)$.
- A complexidade de Kolmogorov de um objeto pode ser representada por K_m (C) = $Le(C)$, onde m é uma máquina Mealy cuja função f_{bij} é uma codificação de prefixo que minimiza Le.
- A incerteza $I_n(A)$ pode ser uma aproximação da complexidade de Kolmogorov de um objeto, onde m é uma máquina Mealy cuja a função f_{bij} é representada por: $\sum_{i=1}^{n}(-w_i \log_2 w_i)$.

A tabela 2.11 mostra a relação entre a métrica de conteúdo (Complexidade de Kolmogorov) e sua aproximação (Incerteza) $Le(C) \approx I_n(A)$.

Entrada (o_i,τ)	Característica (p_i)	a	b	c	d	e	Sum
	W (τ_i)	0.10	0.15	0.30	0.16	0.29	= 1
Saída KC	Código binário (c_i) - fita de saída	010	011	11	00	10	
	Comprimento de código (in bits) (le_i)	3	3	2	2	2	
	Contribuição da probabilidade no tamanho (le_i * w_i)	0.30	0.45	0.60	0.32	0.58	$Le(C)$ = 2.25
Saída I_n	Orçamento de probabilidade (2^{-le_i})	1/8	1/8	1/4	1/4	1/4	= 1
	Conteúdo da informação (in bits) (-$log_2 w_i$) ≈	3.32	2.74	1.74	2.64	1.79	
	Contribuição para a incerteza (-$w_i log_2 w_i$)	0.332	0.411	0.521	0.423	0.518	$I_n(A)$ = 2.205

Tabela 2.11: *Complexidade de Kolmogorov e aproximações*

A complexidade de Kolmogorov (KC) e a Incerteza (I_n) possuem maneiras diferentes embora consigam representar os objetos fornecendo propriedades semelhantes. A complexidade de Kolmogorov, em termos concretos, é o pedaço mínimo de código / programa que se pode escrever para gerar uma string particular. A diferença básica entre a KC e a I_n é que a KC possui foco no conteúdo da informação, enquanto a I_n concentra–se na informação faltando, ilustrando: suponha que haja uma partida entre a equipe A e a

equipe B, com possíveis resultados sendo vitória ou perda somente. Alice está assistindo a partida, enquanto que Bob não tem acesso a ela, mas, no entanto, conhece as equipes envolvidas e outros detalhes essenciais. Agora, a aleatoriedade associada a tal evento é capturada pela I_n. Ele diz que esse evento tem aleatoriedade equivalente a 1 bit. Alice pode usar com segurança 1 bit para transmitir o resultado da partida (a informação faltando). Também chamada de informação perdida porque Bob teve todas as informações sobre a partida, exceto o resultado que foi o resultado de um evento aleatório.

Agora, suponha que Alice queira armazenar um pedaço de corda (resultados das partidas de uma equipe) como, por exemplo, "11110000". Poucas formas possíveis de armazenar o conjunto de resultados:

- "11110000"usando 8 caracteres;
- "1*40*4"usando menor número de caracteres.

A menor dessas possibilidades, em termos de tamanho, é o que é conhecida como KC. Note que a KC está associada a todo o conteúdo (certeza) da informação.

2.6.11 Relações entre objetos

Seja R o conjunto das relações $R = (O, B)$, em que: O é um conjunto não-vazio de objetos, denominado conjunto universo, e B é um conjunto de relações binárias $R \subseteq O \times O$. As relações possuem propriedades como mostra a tabela 2.12.

Propriedade	descrição
Reflexiva	$\forall x \in O\,(x,x) \in R$
Irreflexividade	$\forall x \in O\ \neg R(x,x)$;
Simétrica	$\forall x,y \in O\,(x,y) \in R \Rightarrow (y,x) \in R$;
Antissimétrica	$\forall x,y \in O\,(x,y) \in R \wedge (y,x) \in R \Rightarrow (x,y) = (y,x)$
Assimetria	$\forall x,y \in O\,(R(x,y) \Rightarrow \neg R(y,x))$
Transitiva	$\forall x,y,z \in O\,(x,y) \in R \wedge (y,z) \in R \Rightarrow (x,z) \in R$

Tabela 2.12: *Propriedades das relações.*

Essas propriedades caracterizam vários tipos de relações. Um tipo de relação é a relação de equivalência. As propriedades de uma relação de equivalência garantem uma série de condições, dentre elas, a representação dos objetos em um espaço métrico e sua apresentação em gráficos por exemplo. As relações de equivalência são importantes para a concepção de estruturas capazes de auxiliar na formação das relações de ordem. Uma relação de equivalência deve possuir as seguintes propriedades: reflexiva, simétrica e transitiva.

A relação de ordem é representada através do percurso de um caminho em uma estrutura hierárquica (árvore) composta de uma sequência de objetos consecutivos (o_1, o_2, ..., o_{k-1}, o_k) tal que existe sempre a relação: "o_i<predicado>o_{i+1}". Uma lista de possíveis predicados que caracterizam uma relação de ordem são mostrados na tabela 2.13.

Predicados binários	Predicados unários
o_i é o anterior de o_{i+1}	o_1 é o primeiro
o_{i+1} é o posterior de o_i	o_k é o último
o_i é menor que o_{i+1}	o_k é o menor
o_i é menor ou igual a o_{i+1};	o_k é o maior
o_{i+1} é maior que o_i	
o_{i+1} é maior ou igual a o_i	

Tabela 2.13: *Predicados presentes em uma relação de ordem.*

Também, nas relações de ordem, torna-se possível a utilização de predicados unários do tipo O_i<predicado>.

Um percurso pode ser formado entre k objetos ordenados de forma a conceber $k-1$ pares de relações de ordem consecutivas e um caminho de comprimento também igual a $k-1$. Tal propriedade se torna útil no processo de identificação de intervalos através de cortes na árvore de agrupamento que são responsáveis pela delimitação das fronteiras entre as classes e subclasses destes objetos.

2.6.12 Relações Emergentes entre os Objetos

Seja E o conjunto das relações emergentes $E = (O,R)$, em que: O é um conjunto não-vazio de objetos, denominado conjunto universo, e R é um conjunto de relações binárias $R \subseteq O \times O$.

As relações emergentes são representadas da seguinte forma: Sejam x e y dois objetos de agrupamento pertencentes a O, então xR_iy é a relação entre x e y do tipo i, sendo $i \in N^*$. O valor de i indica qual o tipo de relação binária relaciona os dois objetos, por exemplo, a indiscernibilidade R_i para $i = 1$ pode ser definida da seguinte forma: dados os objetos x, $y \in$ O, se xR_1y então x e y são indiscerníveis em A, ou seja, a classe de equivalência definida por x é a mesma que a definida por y, i.e., $[x]R_1 = [y]R_1$. A indiscernibilidade de dois objetos cresce monotonicamente com o aumento de correspondências e a redução de diferenças.

2.6.13 Similaridade e dissimilaridade das relações emergentes

As métricas de comparação (semelhança ou diferença) entre objetos que caracterizam as relações emergentes podem ser classificadas por dois tipos:

- *similaridade*: mede quão semelhante são dois objetos (maior valor escalar, maior a semelhança);
- *dissimilaridade*: mede quão diferente são dois objetos (menor valor escalar, maior semelhança).

Em geral, os algoritmos de análise de agrupamentos têm como base métricas de dissimilaridade. Quanto maior for a medida de dissimilaridade menor será a semelhança entre os objetos. A classificação dos tipos de relações emergentes por similaridade e dissimilaridade visa o estabelecimento de escalas de valores para a comparação entre os objetos.

2.6.14 Métricas de semelhança com base em incerteza

Pode-se quantificar a incerteza dos pares de objetos através da quantidade de informação associada as características expressas pelas variáveis aleatórias conjuntamente (a incerteza conjunta, $I_n(X;Y)$), a quantidade de informação de uma variável aleatória dado que outra variável aleatória é conhecida (a incerteza condicional, $I_n(X|Y)$) e também a quantidade de informação que uma variável aleatória contém acerca da outra (incerteza mútua, $I_n(X;Y)$). As relações entre essas métricas são expressas na figura 2.25.

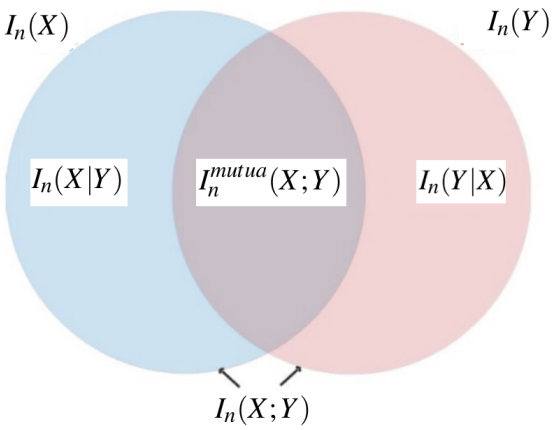

Figura 2.25: *Métricas com base em incerteza.*

A incerteza conjunta é dada por:

$$I_n(X;Y) = \sum_x \sum_y p_{x;y} * log_2 \frac{1}{p_{x;y}} \tag{2-19}$$

No caso, os objetos $X;Y$ podem ser considerados uma variável aleatória única, com uma distribuição probabilística de eventos x_i e y_i combinados, na forma $x_i.y_i$.

Dessa forma, a equação da incerteza para uma variável aleatória pode ser usada para quantificação da incerteza conjunta, sendo:

$$I_n(X;Y) = I_n(X.Y) = \sum p_{x.y} * 1/p_{x.y} \tag{2-20}$$

A incerteza condicional é dada por:

$$I_n(X|Y) = \sum_x \sum_y p_{x;y} * log_2 \frac{1}{p_{y|x}} \tag{2-21}$$

Uma estratégia simples de cálculo é $I_n(X|Y)$ = $I_n(X;Y)$ - $I_n(Y)$, em que os valores $I_n(X;Y)$ e $I_n(Y)$ podem ser calculados pela equação da incerteza para uma variável aleatória. Note que $I_n(X|Y)$ difere de $I_n(Y|X)$.

No entanto, a partir da propriedade:

$$I_n(X) - I_n(X|Y) = I_n(Y) - I_n(Y|X) \tag{2-22}$$

obtém-se a informação mútua entre as distribuições. A informação mútua é dada por:

$$I_n^{mutua}(X;Y) = \sum_x \sum_y p_{x;y} * log_2 \frac{p_{x;y}}{p_x * p_y} \tag{2-23}$$

Um artifício de cálculo é:

$$I_n^{mutua}(X;Y) = I_n(X) + I_n(Y) - I_n(X;Y) \tag{2-24}$$

Os valores $I_n(X)$, $I_n(Y)$ e $I_n(X;Y)$ podem ser calculados pela equação da incerteza para uma variável aleatória.

2.6.15 Distância entre objetos

Uma distância xD_iy é um tipo de relação emergente xR_iy, agregada de um conjunto de restrições. Para que uma métrica, também, seja uma distância D, torna-se necessário que xR_iy seja uma extensão de uma relação de equivalência. Uma relação de equivalência é um conjunto de restrições que garante a representação dos objetos e suas semelhanças em um espaço métrico. A condição de desigualdade triangular da geometria euclidiana é essencial para a representação de objetos em um espaço métrico, sendo assim, esta condição é garantida com as restrições em xR_iy satisfeitas. As restrições são:

$$\forall x \in U, xR_ix(reflexividade) \tag{2-25}$$

$$\forall x, y \in U, xR_i y \Rightarrow yR_i x(simetria) \tag{2-26}$$

$$\forall x, y, z \in U, xR_i y \wedge yR_i z \Rightarrow xR_i z(transitividade) \tag{2-27}$$

A distância euclidiana (ou distância métrica) é a distância entre dois pontos, que pode ser provada através da aplicação recorrente do teorema de Pitágoras. Utilizando a distância euclidiana, o espaço euclidiano torna-se o espaço métrico.

A distância euclidiana entre os objetos $p = (p_1, p_2, \ldots, p_n)$ e $q = (q_1, q_2, \ldots, q_n)$, num espaço euclidiano n-dimensional, é definida como:

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}. \tag{2-28}$$

A distância de informação entre objetos codificados por prefixo também pode ser medida por métricas de dissimilaridade ou similaridade. Uma métrica de dissimilaridade é a Distância Normalizada de Compressão (NCD)[24]. A *NCD* , entre duas informações, é expressa por um valor numérico compreendido entre 0 e 1, sendo que um valor *NCD* próximo a 0 indica que as informações são semelhantes e um valor próximo a 1 indica que as informações não são semelhantes.
Para m = Máquina de Mealy = $(S, S_0, \Sigma, \Delta, \delta, \delta_s = f_{bij})$,

$$NCD_m(x, y) = \frac{K_m(x \oplus y) - \min\{K_m(x), K_m(y)\}}{\max\{K_m(x), K_m(y)\}} \tag{2-29}$$

onde $x \oplus y$ é a combinação dos objetos x e y.

A aplicação sucessiva da função $NCD_m(x, y)$, para cada par de objetos, gera o conjunto de relações emergentes.

CAPÍTULO 3

Agrupamentos de objetos de dados

Agrupamento é um conjunto de objetos de dados reunidos segundo algum grau de semelhança [3]. O critério de semelhança faz parte da definição do problema e depende do tipo de algoritmo responsável pela formação dos agrupamentos. Como exemplo, um procedimento algorítmico pode ser aplicado a bases de texto, agrupando objetos textuais que possuam como critério o mesmo assunto podendo ser separados por diferentes conteúdos.

Os agrupamentos podem ser de vários tipos podendo assumir características comuns. No agrupamento particionado, os objetos são divididos em grupos do mesmo nível, ou seja, sem sobreposição. No agrupamento hierárquico, os grupos de objetos são alinhados e organizados na forma de uma árvore. O agrupamento é dito exclusivo quando cada objeto é atribuído a um único grupo. Denomina-se agrupamento sobreposto, no caso da coexistência de objetos em vários grupos. Outra forma de agrupamento é o fuzzy, no qual cada objeto pertence a um grupo segundo um grau de pertinência variando entre 0% a 100%. No agrupamento completo, todos os objetos são atribuídos necessariamente a um grupo, caso contrário, são denominados parciais existindo assim objetos que não pertencem a nenhum grupo [83].

A análise de agrupamentos é a reunião de objetos em diferentes grupos, cada um dos quais deve conter os objetos semelhantes segundo alguma função de distância estatística [31]. Essa reunião deve ser realizada de maneira automática, sem intervenção do usuário, sem considerar previamente propriedades características dos grupos e sem o uso de grupos de teste previamente conhecidos para direcionar a formação do aglomerado. A análise de agrupamentos pode ser aplicada a diversas áreas de atuação, destacando- -se no contexto de modelagem de sistemas. Nesta pesquisa, a análise dos agrupamentos de uma fonte de dados produz uma hipótese H que pode ser utilizada para modelos de aprendizado de máquina.

3.1 Agrupamento hierárquico

Um Dendrograma (dendro = árvore) é um tipo específico de diagrama ou representação icônica que organiza determinados fatores e variáveis na forma de um agrupamento hierárquico [31]. Resulta de uma análise estatística de determinados dados, em que se emprega um método quantitativo que leva a agrupamentos e à sua ordenação hierárquica ascendente, o que em termos gráficos se assemelha aos ramos de uma árvore que se divide em outros sucessivamente. O arranjo de agrupamentos na forma de um dendrograma derivado da aplicação de um algoritmo de agrupamento (*clustering*) pode ser visualizado na figura 3.1.

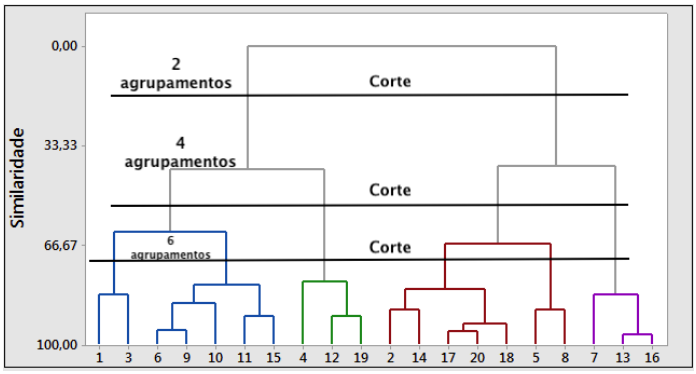

Figura 3.1: *Dendrograma*

Um cladograma é um diagrama semelhante ao dendrograma que representa as relações filogenéticas entre os seres vivos. Em biologia computacional, o cladograma mostra as relações evolutivas entre diferentes clados biológicos(árvore filogenética), após a análise estatística dos dados genéticos. Um cladograma não difere na hierarquização de informações com um dendrograma, apenas em sua geometria(figura 3.2).

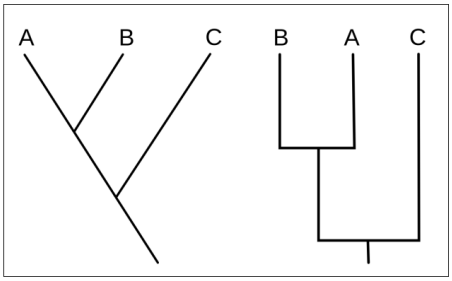

Figura 3.2: *Cladograma (a), dendrograma(b)*

3.1.1 Representação formal do agrupamento hierárquico

O agrupamento hierárquico tradicional parte de um universo original do conhecimento, que se divide e subdivide-se para criar uma única estrutura classificatória[42]. Cada item ou tópico do conhecimento representado por um elemento do tipo objeto é

alocado em um ponto(nó) terminal desta estrutura. O universo de objetos completo interligado pelas estruturas de ramos e nós internos é chamado de *Agrupamento Principal* ou *Contexto de Agrupamento* (figura 3.3). Cada objeto é disposto em grupos segundo alguma característica de divisão em um caminho único e lógico.

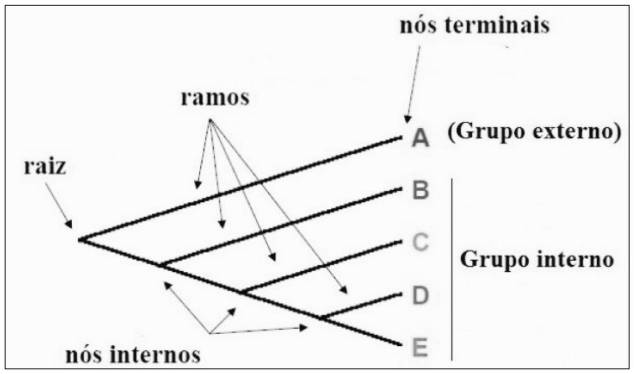

Figura 3.3: *Hierarquia de objetos*

O contexto de agrupamento ou simplesmente contexto C é definido como uma 5-upla $C = (O, P, E, N, A)$ onde O é um conjunto finito e não-vazio de objetos chamado de universo, P é um conjunto finito e não-vazio formado pelos atributos condicionais desses objetos, E é um conjunto finito e não-vazio das relações emergentes entre os objetos pertencentes a O, N é o conjunto de nós internos e A é o conjunto de ramos(arcos) que conectam os nós internos de N com os objetos do conjunto O. O contexto de agrupamento é uma arquitetura que indexa os objetos de um contexto de informação estabelecendo uma relação de ordem entre eles.

3.1.2 Nós internos e ramos do agrupamento hierárquico

Seja $O = \{o_0, o_1, o_2, .., o_{i-1}, o_i\}$ para $i, m \in \mathbb{N}$ o conjunto dos objetos do contexto de agrupamento, $N = \{n_0, n_1, n_2, .., n_{i-1}\}$ os nós internos e $A = \{a_0, a_1, a_2, .., a_m\}$ os ramos ou arcos de conexão entre os nós e objetos. Tanto os nós internos N como os ramos A são abstrações matemáticas utilizadas em conjunto para especificar as relações classificatórias entre os objetos do conjunto O, de modo a organizá-los em uma hierarquia de elementos e acomodá-los em uma relação de ordem, possibilitando assim, uma instância de contexto de agrupamento. Os conjuntos de nós internos e os ramos compõem uma estrutura de dados capaz de dispor os objetos em subconjuntos que hierarquicamente são subordinados a outros subconjuntos disjuntos de objetos.

Defini-se, desta forma, um agrupamento hierárquico como um conjunto finito de um ou mais objetos, um conjunto finito de nós, e um conjunto de finito de ramos, tais que:

- o conjunto de ramos possui três partições formando três conjuntos: o conjunto de ramos que conectam os nós internos A_{in}, o conjunto de ramos que conectam os nós

internos aos objetos A_{io} e o conjunto de ramos que conectam os objetos de A_{oo}. Os conjuntos A_{in} e A_{io} e A_{oo} são disjuntos;

- existe um nó n_0 particular em N chamado raiz(nó principal);
- os nós internos restantes(denominados de filhos) estão divididos em n > 0 conjuntos disjuntos $n_1, n_2, \ldots, n_m$, os quais estão ligados à raiz através dos ramos formando um caminho;
- cada conjunto de nós internos n_k, $k = 1, \ldots, M$ é uma hierarquia, denominada sub--hierarquia da raiz;
- cada nó filho é também raiz de uma sub-hierarquia;
- um nó sem filhos é denominado de objeto (ou nó terminal). Ou seja, um nó sem filhos não pertence ao conjunto N e sim ao conjunto O.

As fases anteriores do processo de agrupamento são caracterizadas pela composição da informação dos objetos e suas relações emergentes através de uma estrutura de dados. O modelo de arquitetura de informação que comporta e gerencia os objetos para a classificação utiliza uma hierarquia. As hierarquias são concebidas a partir de um conjunto de relações emergentes que contém os dados disponíveis sobre os objetos e suas inter-relações. Estes dados são comparados, e os objetos agrupados em clados ou ramos de acordo com suas inter-relações.

3.1.3 Algoritmos para a formação de agrupamentos hierárquicos

Busca binária

A busca binária é um algoritmo de busca em vetores que segue o paradigma de divisão e conquista. Ela parte do pressuposto de que o vetor está ordenado e realiza sucessivas divisões do espaço de busca comparando o elemento buscado (chave) com o elemento no meio do vetor. Se o elemento do meio do vetor for a chave, a busca termina com sucesso. Caso contrário, se o elemento do meio vier antes do elemento buscado, então a busca continua na metade posterior do vetor. E finalmente, se o elemento do meio vier depois da chave, a busca continua na metade anterior do vetor.

Árvore k-d

Uma árvore $k-d$(abreviação para a árvore k-dimensional) [5] é uma estrutura de dados de particionamento do espaço para a organização de pontos em um k-dimensional espaço. Árvores $k-d$ são estruturas úteis para uma série de aplicações, tais como pesquisas envolvendo pesquisa multidimensional de chaves(busca de abrangência e busca do vizinho mais próximo). Árvores k-d são um caso especial de árvores de particionamento binário de espaço.

Vizinhos próximos

Grande parte dos algoritmos de agrupamento analisa um conjunto de relações emergentes de objetos e formam uma hierarquia capaz de representar cada objeto do agrupamento a partir de uma estrutura de dados na forma de uma árvore (grafo acíclico). A análise do conjunto de relações emergentes, em alguns casos, pode ser conduzida por heurísticas, que otimizam o espaço de buscas do algoritmo. Podem ser utilizados vários algoritmos aglomerativos [25] para esta função. A filogenética computacional concebe um modelo de agrupamento capaz de implementar uma hierarquia (conjunto de táxons). A organização, desta hierarquia, é baseada em uma unidade taxonômica, essencialmente associada a um sistema de classificação científica. O táxon pode indicar uma unidade de divisão (corte) em qualquer nível de um sistema de classificação organizado por uma estrutra de dados do tipo árvore. O objetivo da filogenética computacional é prover uma árvore evolutiva T para um dado conjunto de n táxons, onde a distância Mi j entre táxons i e j é conhecida. T é uma árvore ponderada sem raiz, cujos pesos de aresta representam a distância de um táxon para seu ancestral. Ela possui n folhas, correspondente aos táxons. Cada nó interno representa o ancestral comum mais recente entre seus filhos; se a árvore for binária, existem n - 2 ancestrais. Para escolher uma filogenia para um conjunto de táxons, precisamos ser capazes de comparar árvores e decidir qual é melhor. O princípio da máxima parcimônia determina que dentre duas árvores, é preferível aquela que supõe o menor número de modificações de cada ancestral para seus descendentes. Isto é similar à navalha de Occam, que escolhe a explicação mais simples entre duas igualmente poderosas. Se as mudanças entre ancestrais e descendentes é mensurada pelo comprimento de aresta, esse princípio é equivalente a minimizar a soma total dos comprimentos da árvore.

Simplesmente, enumerar todas as árvores possíveis e compará-las é inviável, exceto para conjuntos muito pequenos de dados, já que o número de árvores com n folhas cresce exponencialmente. Métodos filogenéticos precisam confiar em heurísticas para buscar o espaço de árvores, preferencialmente, evitando hipóteses não promissoras. Os algoritmos mais avançados, como MrBayes [68], realizam mutações em uma árvore para explorar o espaço e encontrar árvores que sejam mais parcimoniosas. Esses métodos são muito caros computacionalmente e precisam de um passo inicial bom para serem bem-sucedidos.

O método de Junção de Vizinhos (*Neighbor Joining*) é um algoritmo que busca minimizar a soma total de comprimentos de aresta com uma abordagem bottom-up gulosa [72]. Este método encontra a árvore mais parcimoniosa se assumir que a métrica entre elementos é aditiva, isto é, $M_{ij} + M_{jk} = M_{ik}$.

Neste trabalho, buscando compreender o mecanismo completo do método de Junção de Vizinhos, são demonstradas as passagens do algoritmo a seguir.

- Primeiro passo:
  Com base no conjunto de relações emergentes dispostos na forma de uma matriz de distâncias, calcule a matriz Q.

$$Q(i,j) = (n-2)d(i,j) - \sum_{k=1}^{n} d(i,k) - \sum_{k=1}^{n} d(j,k), \tag{3-1}$$

  onde $d(i,j)$ é o valor da relação emergente de equivalência d entre o objeto i e o objeto j.
- Segundo passo:
  Encontre o par de objetos distintos i e j (isto é, com $i \neq j$) para o qual $Q(i,j)$ tem seu valor mais baixo. Esses objetos estão unidos a um nó interno recém-criado, que está conectado ao nó central. Para cada um dos objetos no par que está sendo unido, utilize as seguintes fórmulas para calcular a distância para o novo nó interno:

$$\delta(f,u) = \frac{1}{2}d(f,g) + \frac{1}{2(n-2)}\left[\sum_{k=1}^{n} d(f,k) - \sum_{k=1}^{n} d(g,k)\right] \tag{3-2}$$

  e

$$\delta(g,u) = d(f,g) - \delta(f,u). \tag{3-3}$$

  Os objetos f e g são os pares de nós e u é o nó interno recém-criado. Os ramos conectando f, u e g, u, e seus comprimentos, $\delta(f,u)$ e $\delta(g,u)$ fazem parte da hierarquia que está sendo criada gradualmente. Eles não afetam nem são afetados por etapas de junção posteriores.
- Terceiro passo:
  Calcule a distância de cada um dos objetos no par para este novo nó interno.
- Quarto passo:
  Calcule a distância de cada um dos objetos fora deste par para o novo nó interno.
  Para cada objeto não considerado no passo anterior, calculamos a distância ao novo nó interno da seguinte maneira:

$$d(u,k) = \frac{1}{2}[d(f,k) + d(g,k) - d(f,g)] \tag{3-4}$$

Onde u é o novo nó interno, k é o nó interno que deseja-se calcular a distância e f e g são os objetos do par que foram incorporados a hierarquia.

- Quinto passo:
  Comece o algoritmo novamente, substituindo o par de objetos vizinhos unidos com o novo nó interno e usando as distâncias calculadas no passo anterior.

Em resumo, o algoritmo inicia com um conjunto de relações emergentes de equivalência , cuja topologia corresponde a de uma rede estrela e itera todos os passos descritos anteriormente até que a árvore (hierarquia) esteja completa com todos os nós internos e ramos calculados e conhecidos. Um exemplo completo é mostrado na figura 3.4.

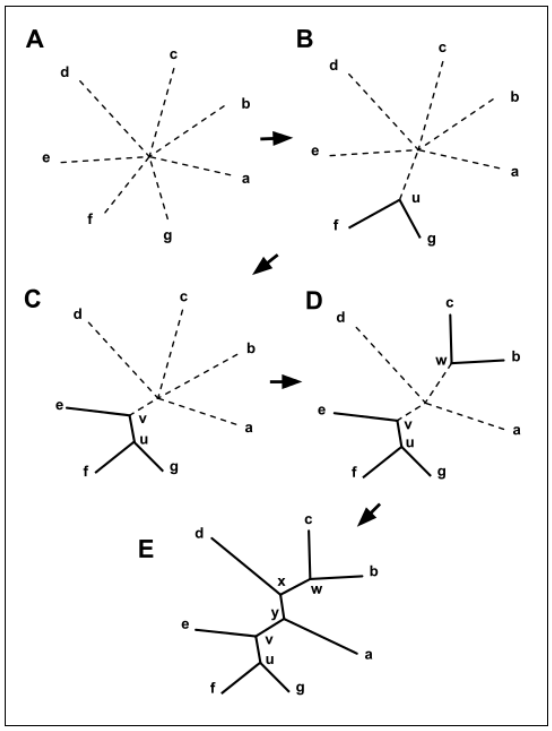

Figura 3.4: *Exemplo dos passos da formação do agrupamento*

Na figura 3.4, ao iniciar os objetos na forma de uma hierarquia com topologia estrela (A), a matriz Q é calculada e utilizada para escolher um par de objetos para junção, neste caso f e g. Estes são unidos a um nó interno recém-criado u como mostrado em (B). A parte da hierarquia mostrada como linhas contínuas agora está corrigida e não será alterada nas etapas de junção subsequentes. As distâncias do nó interno u para os objetos $a-e$ são calculadas a partir da equação descrita no passo (4) do algoritmo. Esse processo é, então, repetido, usando uma matriz com apenas as distâncias entre os objetos, a,b,c,d,e e o nó interno u, e uma matriz Q derivada dele. Neste caso, u e e estão unidos ao nó interno v recentemente criado, como mostrado em (C). Mais duas iterações levam

primeiro a (D) e, em seguida, a (E), neste ponto o algoritmo é concluído, e a hierarquia torna-se a representação do contexto de agrupamento.

3.1.4 Métrica qualitativa para agrupamento hierárquico

Uma escala contém o intervalo de valores que compõem um sistema de medida incluindo os limites mínimo e máximo. Além da definição do domínio de valores da escala, a associação dos valores numéricos aos objetos de medição é essencial.

Nos agrupamentos hierárquicos concebidos pelo protótipo desta tese, deseja-se medir, a partir de uma escala, quanto a hipótese(figura 3.5) de agrupamento descrita pelo dendrograma aproxima-se da expectativa do experimento.

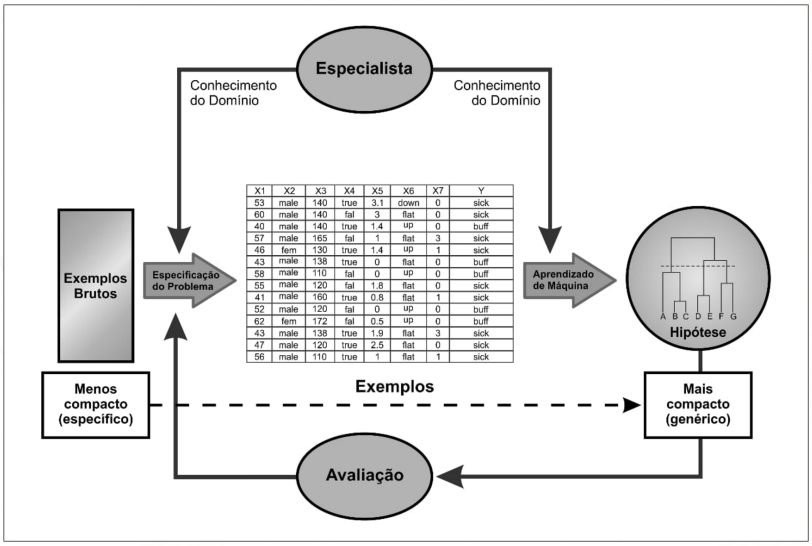

Figura 3.5: *Hipótese de agrupamentos*

Objetiva-se então a concepção de uma métrica de qualidade cujo propósito é quantificar a capacidade de generalização do modelo expressa pelo agrupamento descrito no dendrograma. Em outras palavras, a medida da generalização expressa a capacidade de aprendizado do modelo a partir da informação extraída de uma fonte de dados. A generalização é expressa pela ordem dos objetos de dados dentro do dendrograma. A ideia geral é que objetos de dados semelhantes estejam situados de forma contígua no dendrograma que representa a hipótese.

3.1.5 Métrica quantitativa para agrupamento hierárquico

A análise quantitativa do agrupamento hierárquico pode ser realizada por uma métrica de robustez a partir da contagem do número de congruências em associações par a par de elementos entre diversos agrupamentos (dendrogramas). O número de congruência, também, pode quantificar a correlação de similaridades encontradas pelas análises de

diferentes fontes de dados. Para quantificar o nível de correspondência dos agrupamentos, é importante determinar uma métrica consistente, capaz de avaliar todas as congruências obtidas por diferentes análises.

Sejam G_1 e G_2 dois grupos encontrados por dois experimentos diferentes. Se há alguma correlação entre os dois aspectos, é esperado que muitos objetos de dados que sejam agrupados juntos em G_1, também estejam agrupados juntos em G_2. O número de congruências em agrupamentos par a par contados nos dois agrupamentos, pode representar uma estimativa quantitativa da correspondência entre ambos.

Índice de congruência par a par

O índice de congruência par a par pode ser expresso pela equação 3-5.

$$\rho = \frac{Ne}{Nt} \; Nt = max\{N1, N2\} \tag{3-5}$$

Onde Ne é o número efetivo de congruências encontradas, e $Nt > 0$ é a máxima quantidade de congruências possíveis para o conjunto em análise. Se Nk é o número de associações (par a par) em um agrupamento k, então o máximo número de congruência irá ocorrer se todas as associações(agrupamentos) se repetirem no outro conjunto de dados.

Seja J o conjunto de grupos encontrados a partir da análise de uma determinada fonte de dados e seja i o i-ésimo grupo de J, se n_{ij} é o número de elementos do grupo, então a máxima combinação de elementos será dada por $C_2^{n_{ij}}$. Somando-se todas as combinações de cada grupo do conjunto, obtemos Nj, que representa a máxima congruência possível para o conjunto j. Logo, para dois conjuntos (1,2), o valor máximo de Nt será max$\{N1,N2\}$, o que pode ser estendido para qualquer quantidade de conjuntos $m, m > 2$, onde $j = 1, 2, ..., m$.

Considere os dois conjuntos de agrupamentos, $G1$ e $G2$, obtidos a partir de duas análises independentes e descritos na tabela 3.1:

$G1$	$G2$
K_0^1 = {a,b,c,d,e,f}	K_0^2 = {a}
K_1^1 = {g,h}	K_1^2 = {b,c,d,e,f}
K_2^1 = {i,j}	K_2^2 = {g,h}
	K_3^2 = {i,j}

Tabela 3.1: *Análises independentes de grupo 1 (G1) e grupo 2 (G2)*

Seja {a,b,c,d,e,f,g,h,i,j}, um conjunto de objetos de dados. Em $G1$, obteve-se três grupos, K_0^1, K_1^1 e K_2^1, os grupos K_1^1 e K_2^1 possuem apenas dois elementos, o que caracteriza apenas uma única associação par a par em cada: (g, h) e (i, j). Já o grupo K_0^1

possui seis elementos, portanto permitirá $C_2^6 = 15$ associações par a par: (a,b),(a,c),...(a, f),(b,c),(b,d),...(e, f). Somando todas as associações, têm-se $N1 = 17$.

Esta análise é aplicada exatamente da mesma forma para $G2$, onde foram identificados quatro grupos, K_0^2, K_1^2, K_2^2 e K_3^2. Os grupos K_2^2 e K_3^2 possuem apenas dois elementos, totalizando apenas duas associações possíveis. Já o grupo K_0^2 possui apenas o elemento a, logo nenhuma associação par a par é possível. O grupo K_1^2 possui cinco elementos, totalizando $C_2^5 = 10$ associações possíveis. Portanto, $N2 = 10 + 1 + 1 = 12$.

Seja U_{G1} o conjunto que contém todas as associações par a par possíveis de $G1$ e V_{G2} o conjunto que contém todas as associações possíveis de $G2$. O número efetivo de congruências entre ambos(Ne) será dado por $Ne = n(U_{G1} \cap V_{G2})$, ou seja, a quantidade de elementos comuns aos dois conjuntos.

Em resumo, para o exemplo apresentado, têm-se:

$$Ne = 12 \quad Nt = max\{17, 12\} = 17 \quad \rho = \frac{Ne}{Nt} = \frac{12}{17} = 0.7059 \tag{3-6}$$

Vale a pena lembrar que assim como diversos outros índices da literatura, o valor de ρ não é mapeado em uma escala linear de acordo com o número de congruências. No exemplo apresentado, a única diferença entre $G1$ e $G2$ é a ausência do elemento a no grupo K_1^2, sendo agrupado sozinho em K_0^2. Apenas essa diferença é suficiente para reduzir de 100% (congruência total) para 70%. A exclusão de um único elemento de um grupo de tamanho n irá reduzir o número de associações par a par possíveis do grupo em n-1. Por exemplo, se o elemento b do grupo K_1^2 também for excluído (ficando em um grupo sozinho, assim como o elemento a, ρ cairá para aproximadamente 47%. Esta informação deve sempre ser considerada durante a análise do índice ρ obtido em qualquer experimento.

A figura 3.6 exibe os agrupamentos $G3$ e $G4$ monofiléticos com corte na raiz resultado da execução do protótipo para a fonte de dados 2(dois) a partir de máquinas de Turing RLE e BWT.

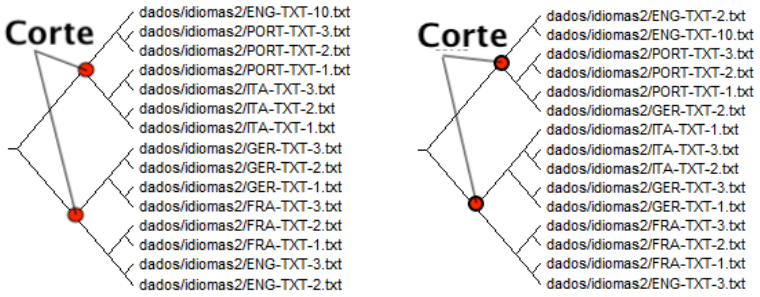

Figura 3.6: *Dendrograma de 15 objetos comprimidos por MT : Dep. 1(Agrup. à esquerda) G3 e Dep. 2((Agrup. à direita) G4*

A partir dos agrupamentos G3 e G4 da figura 3.6 observa-se os seguintes subdivisões K_y^x apresentadas na tabela 3.2.

$G3$	$G4$
K_0^1 = {eng10}	K_0^1 = {eng2,eng10}
K_1^1 = {por3,por2}	K_1^1 = {por3,por2,por1}
K_2^1 = {por1}	K_2^1 = {ger2}
K_3^1 = {ita3,ita2,ita1}	K_3^1 = {ita1,ita3,ita2}
K_4^1 = {ger3,ger2,ger1}	K_4^1 = {ger3,ger1}
K_5^1 = {fra3}	K_5^1 = {fra3,fra2,fra1}
K_6^1 = {fra2,fra1}	K_6^1 = {eng3}
K_7^1 = {eng3,eng2}	

Tabela 3.2: *Grupos formados*

Realizando os calculos:
N3 = 0 + 1 + 0 + 3 + 3 + 0 + 1 + 1 = 9
N4 = 1 + 3 + 0 + 3 + 1 + 3 + 0 = 11
Ne = 9 Nt = max{11,9} = 11 $\rho = \frac{Ne}{Nt} = \frac{9}{11} = 0.82$

O G_{max} (Grau de generalização máximo) é encontrado quando cada subconjunto K_i^j contiver somente objetos da mesma classe. Esta condição estabelece que cada subconjunto possui sua quantidade máxima de elementos. Uma forma de conceber uma métrica de comparação de árvores filogenéticas é a partir da seguinte condição Nt = G_{max} e Ne relativo a árvore filogenética, a qual se deseja medir. A combinação par a par de objetos é dada por:

$$C_2^n = \frac{n!}{s! \cdot (n-s)!} = \frac{n!}{2 \cdot (n-2)!} \tag{3-7}$$

Deseja-se maximizar C_2^n de K_i^j para todos agrupamentos de objetos rotulados a partir de uma determinada classe de forma a obter G_{max}. Maximizando C_2^n, o numerador da equação (1-7) sempre será maior que o denominador, e C_2^n é máximo quando n é máximo. Como o modelo é representado por uma árvore filogenética parcimoniosa que ordena em seus percursos as amostras por similaridade assumindo que cada amostra é ligada a outra amostra de sua vizinhança mais próxima. Sendo assim, da perspectiva qualitativa, a unidade básica de agrupamento pode ser descrita da seguinte forma: se a amostra contém um rótulo designando sua classe e o ancestral mais próximo desta amostra, na árvore filogenética parcimoniosa, possui o mesmo rótulo, então tem-se uma aresta de vizinhança $a_i = 1$, caso contrário $a_i = 0$. Desta forma, a medida de qualidade

(generalização) do modelo M pode ser representada a partir do balanceamento da árvore filogenética e expressa por:

$$G = \frac{\sum_{i=0}^{m-1} \mathrm{a}_i}{m - m_{classes}}, \tag{3-8}$$

onde m é o número de amostras e $m_{classes}$ é o número de classes existentes no conjunto de dados D.

3.2 Geração de hipóteses H de agrupamento

Uma hipótese H de agrupamento é apresentação de uma hierarquia a partir dos objetos de dados de E, sendo E um conjunto de objetos de dados. Uma expectativa E_x é a formulação da hipótese H de agrupamento ideal para determinado conjunto de objetos de dados de E. Uma hipótese H de agrupamento pode ser descrita pela sua representação gráfica (hierárquica e particionada). Os resultados, na forma gráfica, são dendrogramas (árvores) e planos de coordenadas polares.

Os processos de granulação transformam cada objeto de uma fonte de dados em um SI que, então, é codificado por uma MT, que MT define um espaço E de agrupamento. Com relação a E, trata-se de um espaços comprimido, assim, a compressão é a técnica de redução de dimensionalidade utilizada o método proposto (que visa combater o problema da dimensionalidade).

O espaço E possui uma diagramação na forma de um dendrograma ou cladograma para a representação das hipóteses H de uma fonte de dados a partir da MT configurada com um analisador léxico e sintático específicos. As hipóteses podem ser comparadas par a par, com base na referência dada, pela expectativa E_x do agrupamento. A expectativa E_x é dada por:

$$E_x = G_{max}; \tag{3-9}$$

em que G_{max} é o valor máximo possível de G dada uma fonte de dados.

O espaço E_{Cl} é a representação dos objetos de dados em um espaço métrico através de um plano de coordenadas retangulares, e outro com coordenadas polares. A partir destes planos, pode-se utilizar métricas de agrupamento particionado e avaliar se a hipótese H de agrupamento, que inclui a escolha do conjunto P de características, é a mais adequada para determinada fonte de dados.

Para a caracterização dos objetos de dados de uma hipótese H de agrupamento em espaços E e E_{Cl}, uma fonte de dados não-estruturada deve ser convertida em uma fonte

estruturada. Determinar o *SI* de uma fonte não-estruturada equivale a converter a fonte não-estruturada em uma fonte estruturada. No processo de estruturação da fonte de dados, ou determinação do *SI*, a descoberta das características é realizada pela granulação através do mapeador e o domínio da informação de cada característica definido pela codificação através da técnica de compressão.

A compressão realizada pela MT é necessária, pois, em objetos de dados de fontes não-estruturadas, somente existe o nível elementar de estruturação, ou seja, o conjunto P de características de um objeto de dados é definido a partir das menores unidades de informação que compõem o objeto. Além dos valores de dimensão d variarem de objeto para objeto de dados, a baixa resolução da informação eleva o valor de d sendo então necessária, além da padronização do valor de d, alguma técnica de redução de dimensionalidade que reduza o volume de U buscando o mínimo valor de $|U|$ possível. A compressão para os tipos não-estruturados reduz a dimensionalidade dos objetos de dados de forma a combater o problema da dimensionalidade.

3.3 Agrupamento particionado

No agrupamento particionado os objetos são divididos em grupos do mesmo nível, ou seja, sem sobreposição. Nas próximas seções, o modelo, para agrupamento particionado utilizado, nesta pesquisa, é apresentado.

3.3.1 Representação formal do agrupamento particionado

Seja $O = \{o_1, o_2, o_3, \ldots, o_n\}$ o conjunto de objetos representados por $X = [\overrightarrow{x_1}, \overrightarrow{x_2}, \overrightarrow{x_3}, \ldots, \overrightarrow{x_n}]^T$ um conjunto de vetores que representa cada objeto de O, onde $\overrightarrow{x_j}$ é um vetor de p coordenadas e n é o número de elementos do conjunto de dados.

$$\overrightarrow{x_1} = [x_{1,1} \;\; x_{1,2} \;\; \ldots \;\; x_{1,p}]$$
$$\overrightarrow{x_2} = [x_{2,1} \;\; x_{2,2} \;\; \ldots \;\; x_{2,p}]$$
$$\ldots = \ldots$$
$$\overrightarrow{x_n} = [x_{n,1} \;\; x_{n,2} \;\; \ldots \;\; x_{n,p}]$$

Cada vetor representa um objeto do conjunto O e cada coordenada desse vetor representa um atributo descritivo do objeto. O conjunto de dados X reside no espaço $\mathbb{R}^p$, e este espaço é referenciado pelos algoritmos de análise de dados como *espaço dos dados*, *espaço de entrada* ou *espaço vetorial*.

Formalmente, dado um conjunto de objetos representados por um conjunto de vetores X onde $x_i \in \mathbb{R}^p$, tem-se então uma função $G : \mathbb{R}^p \times W \rightarrow C$ onde W é um vetor

de parâmetros ajustáveis, por meio de um algoritmo de aprendizado não supervisionado, que determina c-grupos em X, $C = C_1, ..., C_c (c \leq n)$, tal que:

- $C_i \neq \emptyset$, $i = 1, ..., c$;
- $U_{i=1}^{c} = 1\ Ci = X$;
- $C_i \cap C_j = \emptyset$, $i, j = 1, ..., c$ e $i \neq j$, assumindo a abordagem de agrupamento clássica.

3.3.2 Redução de dimensionalidade dos objetos

Seja (0,0) o ponto de origem de um plano cartesiano $\mathbb{R}^2$ (figura 3.7) e *PC* o plano descrito pelo círculo $x^2 + y^2 \leq 1$ denominado *Plano Polar de Certeza*. Seja *Cr* a circunferência $x^2 + y^2 = 1$ subdividida em p partes iguais. Em outras palavras, existem p pontos em *Cr* descritos por vetores Cr_i para $i = 1, ..., p$ com $i \in \mathbb{N}$. Cada vetor Cr_i parte da origem, possui módulo = 1 e angulo $\theta_j = j\frac{2\pi}{p}$ sendo $\theta_j \in \theta \ \ com \ \ 0 \leq j \leq p-1$ e $j \in \mathbb{N}$.

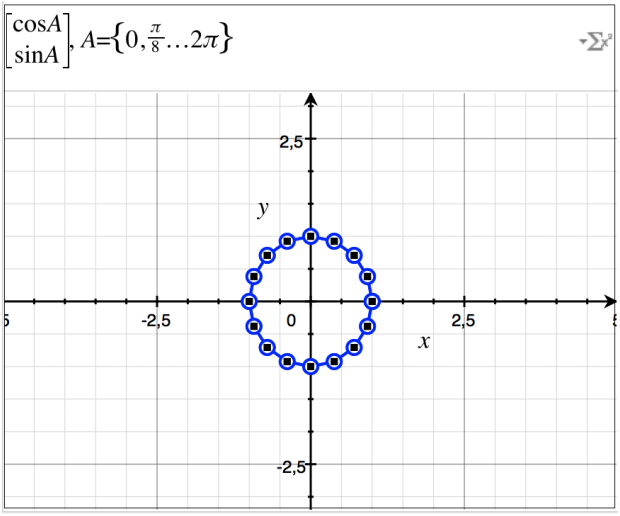

Figura 3.7: *Plano de Certeza com 16 direções (j = 16)*

Seja a desigualdade de Kraft que estabelece a condição necessária e suficiente de existência de um código instantâneo formado por palavras de comprimento variável, definida da seguinte forma:

$$\sum_{i=1}^{p} r^{-l_i} \leq 1 \tag{3-10}$$

Onde r é o número símbolos do alfabeto do código produzido pela fase de cofificação, p o total de palavras-código ou características e l_i é o comprimento associado ao i-ésimo grânulo-código. Para r = 2 devido a complexidade de Kolmogorov, então a desigualdade de Kraft pode ser descrita pelo somatório dos termos: $\frac{1}{2^{li}}$, neste caso, cada termo da desigualdade de Kraft é um multiplo de $\frac{1}{2}$.

Seja um agrupamento particionado representado por um conjunto de vetores X que representa cada objeto de O sendo $x_i \in X$ onde cada $x_{n,p} \in x_i$ possui um módulo que tem como valor um termo $\frac{1}{2^{li}}$ da desigualdade de Kraft. Ou seja, para cada $\overrightarrow{x_n} = [x_{n,1}, x_{n,2}, \ldots, x_{n,p}]$ existe uma sequência $KRAFT_n = [x_{n,l_1}, x_{n,l_2}, \ldots, x_{n,l_i}] = [\frac{1}{2^1}, \frac{1}{2^2}, \ldots, \frac{1}{2^i}]$.

A combinação de cada elemento $(Kraft_i, \theta_i)$ forma uma coordenada polar. Cada coordenada polar corresponde a um vetor onde $KRAFT$ é a lista dos valores de módulos e θ corresponde a um operador de direção horário aplicado a cada módulo de KRAFT.

$\overrightarrow{x_n} = \overrightarrow{Kraft_n} = (\text{Kraft}_1, \theta_1) + (Kraft_2, \theta_2) + ... + (Kraft_i, \theta_i)$

3.3.3 Métricas para agrupamento particionado

Medidas de avaliação externa para agrupamentos podem ser aplicadas quando os rótulos das classes para cada ponto de dados em algum conjunto de avaliação podem ser determinados a priori. A tarefa de agrupamento particionado é então atribuir a estes pontos de dados rótulos numéricos de grupos ou classes, de modo que cada grupo contenha todos e somente os pontos de dados que são membros da mesma classe. Dados os rótulos verdadeiros de cada objeto, é trivial determinar se o agrupamento perfeito foi alcançado. No entanto, avaliar quão uma solução de agrupamento é incorreta, torna-se uma tarefa difícil, pois muitas abordagens carecem de rigor [48].

Para os propósitos da discussão a seguir, suponha um conjunto de dados compreendendo N pontos de dados e duas partições destes: um conjunto de classes, $C = \{c_i | i, ..., n\}$ e um conjunto de agrupamentos, $K = \{k_i | i, ..., m\}$. Seja A a tabela de contingência produzida pelo algoritmo de formação de grupos que representa a solução de agrupamento, de modo que $A = \{a_{ij}\}$ onde a_{ij} é o número de pontos de dados que são membros da classe c_i e elementos do agrupamento k_i.

Para satisfazer os critérios estabelecidos das métricas de agrupamento particionado utilizadas nesta pesquisa, um grupo deve atribuir apenas os pontos de dados que são membros de uma única classe a um único grupo. Ou seja, a distribuição de classes dentro de cada grupo deve ser inclinada para uma única classe, isto é, incerteza zero.

Homogeneidade

Pode-se determinar quão próximo um determinado agrupamento é do ideal, examinando a incerteza condicional da distribuição de classes, dado o agrupamento proposto . No caso perfeitamente homogêneo, esse valor, $I_n(C,K)$ é 0. Entretanto, em uma situação imperfeita, o tamanho desse valor, em bits, depende do tamanho do conjunto de dados e da distribuição dos tamanhos das classes. Portanto, ao invés de representar a incerteza condicional bruta, normaliza-se esse valor pela redução máxima na incerteza que a informação de agrupamento poderia fornecer, especificamente, $I_n(C)$.

Observe que $I_n(C,K)$ é maximal (e é igual a $I_n(C)$ quando a tarefa de agrupamento não fornece nenhuma nova informação, ou seja, a distribuição de classes dentro de cada agrupamento é igual à distribuição geral da classe. $I_n(C,K)$ é 0 quando cada agrupamento contém apenas membros de uma única classe, um agrupamento perfeitamente homogêneo. No caso degenerado em que $I_n(C)$ quando há apenas uma única classe, definimos a homogeneidade como sendo 1. Para uma solução perfeitamente homogênea, a normalização, $\frac{I_n(C,K)}{I_n(C)}$, igual a 0. Assim, adotando a convenção de $I_n(C)$ como sendo 1 para desejável e 0 indesejável, defini-se a homogeneidade como [69]:

$$homo = \begin{cases} 1, & se \;\; I_n(C,K) = 0 \\ 1 - \frac{I_n(C,K)}{I_n(C)}, & caso \;\; contrário \end{cases} \tag{3-11}$$

Completude

A completude é simétrica à homogeneidade. Em uma solução de agrupamento perfeitamente completa, cada uma das distribuições será completamente inclinada para um único grupo. Pode-se avaliar este grau de inclinação calculando a incerteza condicional da distribuição de agrupamentos proposta, dada a classe dos componentes de dados, $I_n(K,C)$. No caso perfeitamente completo, $I_n(K,C) = 0$. No entanto, no pior cenário, cada classe é representada por cada agrupamento com uma distribuição igual à distribuição dos tamanhos de grupos, $I_n(K,C)$ é máxima e é igual a $I_n(K)$. Finalmente, no caso degenerativo em que $I_n(K) = 0$, quando há um único grupo, define-se que a completitude é 1. Portanto, simétrico ao cálculo acima, define-se a completude como [69]:

$$compl = \begin{cases} 1, & se \;\; I_n(K,C) = 0 \\ 1 - \frac{I_n(K,C)}{I_n(K)}, & caso \;\; contrário \end{cases} \tag{3-12}$$

V-medição

Com base nos cálculos de homogeneidade e completude, pode-se calcular um agrupamento V-medição da solução, calculando o peso da média harmônica de homogeneidade e completude, $V_\beta = \frac{(\beta * h) + c}{(1+\beta) * h * c}$ [69]. A V-medição pode ser definida também da seguinte forma:

$$Vmed = 2 * (homo * compl)/(homo + compl) \tag{3-13}$$

Similarmente para a F-medição, se *beta* for maior que 1, a completude será mais fortemente ponderada no cálculo, se for menor que 1, a homogeneidade será mais fortemente ponderada.

O leitor pode observar que os cálculos de homogeneidade, completude e V-medição são completamente independentes do número de classes, do número de grupos,

do tamanho do conjunto de dados e do algoritmo de agrupamento usado. Assim, essas medidas podem ser aplicadas e comparadas em qualquer solução de agrupamento, independentemente do número de pontos de dados(n-invariância), do número de classes ou do número de grupos. Além disso, calculando a homogeneidade e completude separadamente, uma avaliação, mais precisa do desempenho do agrupamento pode ser obtida.

Ajuste aleatório

O índice *Rand* [76] ou a medida de *Rand* (nomeada por William M. Rand) em estatística e, em particular, em *clustering* de dados, é uma medida da similaridade entre dois agrupamentos de dados. Uma variante da medida *Rand* pode ser definida e ajustada para o agrupamento aleatório de elementos, este é o índice *Rand* ajustado. Do ponto de vista matemático, o índice *Rand* está relacionado à precisão, mas é aplicável mesmo quando os rótulos de classe não são usados.

Dado um conjunto de n elementos $S = \{o_1, \ldots, o_n\}$ e duas partições de S para comparar sendo $X = \{X_1, \ldots, X_r\}$, uma partição de S em r subconjuntos e $Y = \{Y_1, \ldots, Y_s\}$, uma partição de S em s subconjuntos, define o seguinte:

- a, o número de pares de elementos em S que estão no mesmo subconjunto em X e no mesmo subconjunto em Y;
- b, o número de pares de elementos em S que estão em subconjuntos diferentes em X e em diferentes subconjuntos em Y;
- c, o número de pares de elementos em S que estão no mesmo subconjunto X e em diferentes subconjuntos em Y;
- d, o número de pares de elementos em S que estão em subconjuntos diferentes em X e no mesmo subconjunto Y.

O índice *Rand* é:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{3-14}$$

Intuitivamente a + b pode ser considerado como o número de acordos entre X e Y e c + d como o número de discordâncias entre X e Y.

Como o denominador é o número total de pares, o índice *Rand* representa a frequência de ocorrência de acordos sobre o total de pares, ou a probabilidade de que x e Y concordará com um par escolhido aleatoriamente. O valor $\binom{n}{2}$ é calculado como $\frac{n(n-1)}{2}$.

O índice *Rand* tem um valor entre 0 e 1, com 0 indicando que os dois agrupamentos de dados não concordam em nenhum par de pontos e 1 indicando que os agrupamentos de dados são exatamente os mesmos. Em termos matemáticos, a, b, c, d são definidos da seguinte forma:

- $a = |S^*|$, *onde* $S^* = \{(o_i, o_j) | o_i, o_j \in X_k, o_i, o_j \in Y_l\}$
- $b = |S^*|$, *onde* $S^* = \{(o_i, o_j) | o_i \in X_{k_1}, o_j \in X_{k_2}, o_i \in Y_{l_1}, o_j \in Y_{l_2}\}$
- $c = |S^*|$, *onde* $S^* = \{(o_i, o_j) | o_i, o_j \in X_k, o_i \in Y_{l_1}, o_j \in Y_{l_2}\}$
- $d = |S^*|$, *onde* $S^* = \{(o_i, o_j) | o_i \in X_{k_1}, o_j \in X_{k_2}, o_i, o_j \in Y_l\}$

para algum $1 \leq i, j \leq n, i \neq j, 1 \leq k, k_1, k_2 \leq r, k_1 \neq k_2, 1 \leq l, l_1, l_2 \leq s, l_1 \neq l_2$

O índice *Rand* ajustado é a versão corrigida do índice *Rand*. Embora o índice Rand possa apenas gerar um valor entre 0 e +1, o índice *Rand* ajustado pode produzir valores negativos se o índice for menor que o índice esperado.

Dado um conjunto S de n elementos e dois agrupamentos ou partições (por exemplo, agrupamentos) desses elementos, $X = \{X_1, X_2, \ldots, X_r\}$ e $Y = \{Y_1, Y_2, \ldots, Y_s\}$, a sobreposição entre X e Y pode ser resumida em uma tabela de contingência $\left[n_{ij}\right]$ onde cada entrada ij denota o número de objetos em comum entre X_i e $n_{ij} = |X_i \cap Y_j|$.

X \ Y	Y_1	Y_2	$\ldots$	Y_s	Somas
X_1	n_{11}	n_{12}	$\ldots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\ldots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\ldots$	n_{rs}	a_r
Somas	b_1	b_2	$\ldots$	b_s	

A forma ajustada do Índice *Rand*, o Índice *Rand* Ajustado(IRA), é:

$$\overbrace{IRA}^{\text{Índice ajustado}} = \frac{\overbrace{\sum_{ij}\binom{n_{ij}}{2}}^{\text{Índice}} - \overbrace{[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}^{\text{Índice Esperado}}}{\underbrace{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}]}_{\text{Índice Max}} - \underbrace{[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}_{\text{Índice Esperado}}} \tag{3-15}$$

Informação mútua ajustada

Na teoria da probabilidade e na teoria da informação, Informação Mútua Ajustada(IMA) é uma variação da informação mútua [89]. A IMA corrige o efeito da concordância apenas devido ao acaso entre agrupamentos, semelhante à maneira como o índice de Rand ajustado corrige o índice de Rand. Está intimamente relacionado à variação de informações. Quando um ajuste similar é feito no índice I_n^{mutua}, torna-se equivalente ao IMA. A medida ajustada, entretanto, não é mais métrica.

Dado um conjunto S de N elementos $S = \{s_1, s_2, \ldots s_N\}$, considere duas partições de S, $U = \{U_1, U_2, \ldots, U_R\}$ com agrupamentos R e $V = \{V_1, V_2, \ldots, V_C\}$ com agrupamentos C. Presume-se aqui que as partições são chamadas de agrupamentos rígidos. As partições são separadas por pares:

- $U_i \cap U_j = V_i \cap V_j = \emptyset$
- para todo $i \neq j$ e completo:
- $\cup_{i=1}^{R} U_i = \cup_{j=1}^{C} V_j = S$

As informações mútuas de sobreposição de agrupamentos entre U e V podem ser resumidas na forma de uma tabela de contingência $RxC\ M = [n_{ij}]_{j=1C}^{i=1\ldots R}$, onde n_{ij} denota o número de objetos comuns aos agrupamentos U_i e V_j. Isso é, $n_{ij} = \left|U_i \cap V_j\right|$.

Como o índice Rand, o valor da linha de base de informações mútuas entre dois agrupamentos aleatórios não assume um valor constante e tende a ser maior quando as duas partições possuem um número maior de agrupamentos (com um número fixo N de elementos do conjunto). Ao adotar um modelo hipergeométrico de aleatoriedade, pode-se mostrar que a informação mútua esperada entre dois agrupamentos aleatórios é:

$$E\{I_n^{mutua}(U;V)\} = \sum_{i=1}^{R}\sum_{j=1}^{C}\sum_{n_{ij}=(a_i+b_j-N)^+}^{\min(a_i,b_j)} \frac{n_{ij}}{N}\log\left(\frac{N\cdot n_{ij}}{a_i b_j}\right) \times \frac{a_i!b_j!(N-a_i)!(N-b_j)!}{N!n_{ij}!(a_i-n_{ij})!(b_j-n_{ij})!(N-a_i-b_j+n_{ij})!} \tag{3-16}$$

Onde $(a_i + b_j - N)^+$ denota $\max(1, a_i + b_j - N)$. As variáveis a_i e b_j são somas parciais da tabela de contingência; isso é, $a_i = \sum_{j=1}^{C} n_{ij}$ e $b_j = \sum_{i=1}^{R} n_{ij}$.

A medida ajustada para a informação mútua pode então ser definida como:

$$IMA(U,V) = \frac{I_n^{mutua}(U,V) - E\{I_n^{mutua}(U,V)\}}{\max\{I_n(U), I_n(V)\} - E\{I_n^{mutua}(U,V)\}} \tag{3-17}$$

A IMA recebe um valor de 1 quando as duas partições são idênticas e 0 quando o I_n^{mutua} entre duas partições é igual ao valor esperado devido apenas ao acaso.

Silhuetas

Silhueta refere-se a um método de interpretação e validação de consistência dentro de agrupamentos de dados. A técnica fornece uma representação gráfica e sucinta de quão bem cada objeto está dentro de seu agrupamento [70]. O valor da silhueta é a medida de como um objeto é semelhante ao seu próprio agrupamento(coesão) comparado a outros grupos(separação). A silhueta varia de 1 a +1, onde um valor alto indica que

o objeto é bem compatível com seu próprio agrupamento e mal combinado com os agrupamentos vizinhos. Se a maioria dos objetos tiver um valor alto, a configuração do agrupamento é apropriada. Se muitos pontos tiverem um valor baixo ou negativo, a configuração do agrupamento poderá ter muitos ou poucos grupos.

Suponha que os dados foram agrupados por meio de qualquer técnica, como k-médias, em k grupos. Para cada dado, $a(i)$ é a distância média entre i e todos os outros dados dentro do mesmo grupo. Pode-se interpretar $a(i)$ como uma medida de quão bem i é atribuído ao seu grupo (quanto menor o valor, melhor a atribuição). Em seguida, define-se a diferença média de pontos i para um grupo c como a média da distância entre i para todos os pontos em c.

Seja $b(i)$ a menor distância média de i para todos os pontos. O grupo, com menor dissimilaridade média, é dito "agrupamento vizinho"de i porque é o próximo melhor conjunto de ajuste para o ponto i. Dessa forma, define-se silhueta assim:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{3-18}$$

Como consequência da definição acima, conclui-se que $-1 \leq s(i) \leq 1$.

Para $s(i)$ estar perto de 1 precisamos $a(i) \ll b(i)$. Como $a(i)$ é uma medida de quão diferentes i é para o seu próprio grupo, um pequeno valor significa que é bem correspondido. Além disso, um grande $b(i)$ implica que i é mal correspondido ao seu agrupamento vizinho. Assim, um $s(i)$ próximo de 1(um) significa que os dados estão apropriadamente agrupados. E se $s(i)$, está perto de negativo, então, pela mesma lógica, conclui-se que i seria mais apropriado se fosse agrupado em seu agrupamento vizinho. O $s(i)$ próximo de zero significa que o dado está na fronteira de dois aglomerados naturais.

A média $s(i)$, sobre todos os dados de um agrupamento, é uma medida de quão, fortemente agrupados são todos os dados no agrupamento. Assim, a média $s(i)$, sobre todos os dados de todo o conjunto de dados, é uma medida de quão apropriadamente os dados foram agrupados. Se houver muitos ou poucos grupos, como pode ocorrer quando uma má escolha de k é utilizada no algoritmo de agrupamento(por exemplo, k-médias), alguns dos agrupamentos geralmente exibem silhuetas muito mais estreitas do que o restante. Assim, gráficos e médias de silhueta podem ser usados para determinar o número natural de grupos dentro de um conjunto de dados. Também, é possível aumentar a probabilidade de a silhueta ser maximizada no número correto de grupos redimensionando os dados usando pesos de recursos que são específicos do agrupamento.

CAPÍTULO 4

Método proposto para geração de hipóteses *H* - *General Mining* (*GM*)

Este capítulo apresenta o método *General Mining* (*GM*) utilizado no processo de geração de hipóteses *H* de agrupamento.

O método *GM* baseia-se na formação de agrupamentos hierárquicos e particionados. O fluxo de execução para a formação dos agrupamentos pode ser visualizado na figura 4.1 e descrito pelas seguintes atividades:

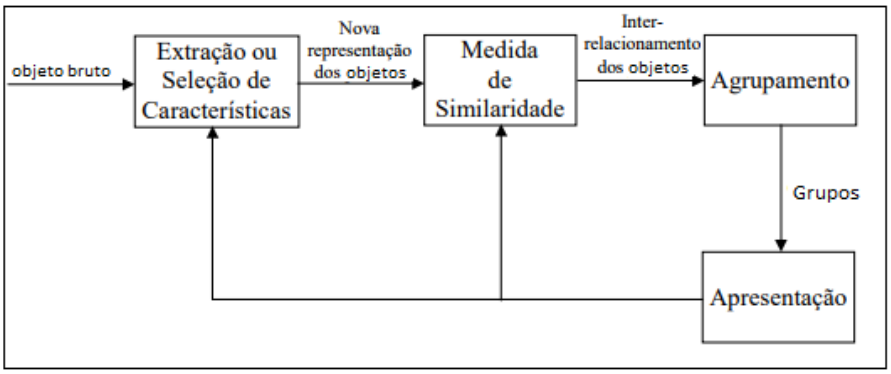

Figura 4.1: *Formação de agrupamentos*

- representação dos objetos: envolve definição do número, tipo e modo de apresentação das características que descrevem cada objeto;
- seleção de características: processo de identificação do subconjunto mais efetivo das características disponíveis para descrever cada objeto;
- extração de características: uso de uma ou mais transformadas junto as características de entrada de modo a salientar uma ou mais característica dentre aquelas que estão presentes nas fontes de dados;
- medida de similaridade: fornecida por uma função de distância definida entre pares de objetos. É possível incluir, na medida de distância, aspectos conceituais (qualitativos) ou, então, numéricos (quantitativos);
- agrupamento: um processo recursivo de junções ou separações de grupos;

- apresentação: deve permitir que um computador possa utilizar o resultado de forma direta ou, então, deve ser orientada ao usuário, permitindo a visualização gráfica dos grupos e a compreensão de suas inter-relações, através da proposição de protótipos ou outras descrições compactas para os grupos.

4.1 Protótipo

A função do protótipo é demostrar o fluxo de execução dos experimentos realizados por meio do método *GM* a partir de um computador. O protótipo contém os seguintes componentes:

- Mapeador de padrões: concebe uma representação da informação para os objetos de dados.
- Combinador de padrões: estabelece as relações entre os objetos de dados através de métricas de similaridade baseadas em compressão.
- Redutor de padrões: resume a informação do conjunto de todos os objetos de dados em uma única estrutura hierárquica.

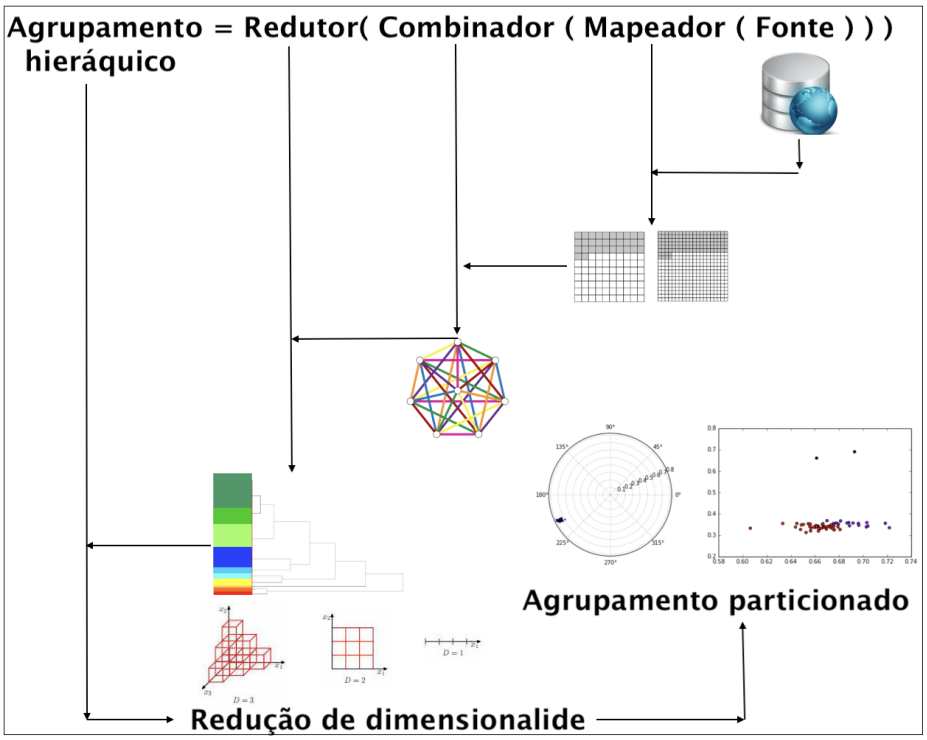

Figura 4.2: *Modelo para agrupamentos*

O *Mapeador*, o *Combinador* e o *Redutor* são funções componentes do protótipo e realizam de forma coordenada a formação dos agrupamentos de objetos. A partir de uma fonte de dados contendo um conjunto de objetos, a execução dos componentes

do protótipo pode ser vista na figura 4.2. Nas próximas subseções, é detalhado cada componente do protótipo.

4.1.1 Mapeador

O componente *Mapeador* é responsável pelas seguintes funções:

- identificar o grau de estruturação da informação contida nos objetos de dados. As opções são: não-estruturado (por ex.: texto, som e imagens), semi-estruturado (por ex.: conjunto de strings ou palavras ,HTML e XML) ou estruturado (por ex.: vetores e dicionários);
- estabelecer o sistema de representação (ASCII, RGB, etc) dos objetos de dados, em outras palavras, identificar o domínio de informação de cada característica que compõe o objeto de dados;
- identificar os grânulos de informação de cada objeto de dados a partir de um sistema de representação;
- padronização do número de dimensões d para todos os objetos de dados de uma fonte.

O *Mapeador* disponibiliza, ao final de sua execução, um conjunto de grânulos endereçados na forma de um dicionário chave-valor para cada objeto.

4.1.2 Combinador

A Máquina de Turing é utilizada como modelo de algoritmo de compressão e utilizada neste estudo para implementar os objetos de dados e suas relações. Segundo Kolmogorov [16], a complexidade de um objeto é um atributo intrínseco do próprio objeto, sendo assim, independente da máquina universal escolhida para representá-lo. A máquina de Mealy possui formalismo tradutor capaz de representar um objeto sendo também um subtipo de MT. O componente *Combinador* possui uma máquina Mealy como modelo de representação dos objetos e relações. Vale a pena lembrar que uma máquina Mealy é definida como uma 6-upla $M = (S, S_0, \Sigma, \Delta, \delta, \delta_s)$ consistindo de:

- um conjunto finito de estados S;
- o estado inicial S_0 que é elemento de S;
- um conjunto finito de grânulos de informação Σ;
- um conjunto chamado de alfabeto de saída $\Delta = \{0, 1\}$;
- a função de transição $\delta : S \times \Sigma \rightarrow S$ mapeando um estado e um grânulo do alfabeto de entrada para o próximo estado;

- a função de saída de dados $\delta_s : S \times \Sigma \rightarrow \Delta^*$ mapeando uma transição para o alfabeto de saída. A função de saída δ_s é uma função bijetora f_{bij}.

A função de saída de dados δ_s é definida a partir de um algoritmo de compressão. Grande parte dos algoritmos de compressão utilizados em compactadores e compressores da literatura, utilizam a codificação Huffman.

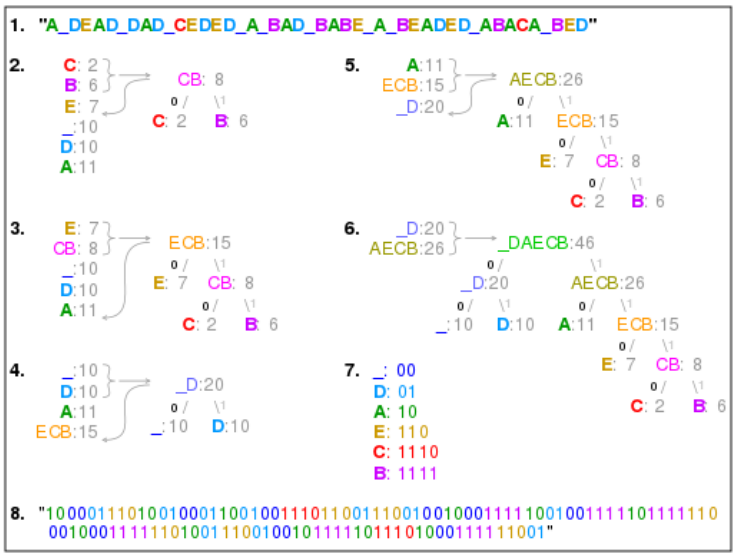

Figura 4.3: *Codificação de objetos de dados*

A figura 4.3 apresenta um exemplo da execução do algoritmo de codificação Huffman a partir de passos numerados de 1 a 8.

- 1 - Entrada: string D de dados puros de um objeto de acordo com algum sistema de representação. No caso, o sistema de representação da figura 4.3 é ASCII.
- de 1 para 2 : formação do conjunto de características P a partir do processo de granulação do objeto de dados.
- de 2 até 6 : estrutura recorrente de comandos (estrutura de repetição) responsável pela formação de uma hierarquia (árvore) de informação (codificação). Tratam-se de instruções que evolvem etapas intermediárias capazes de organizar, a informação contida em um objeto de dados, numa estrutura de dados do tipo árvore binária. A heurística de formação, dessa hierarquia, tem como fundamento a execução de um algoritmo guloso que escolhe a melhor solução, no caso a mais compacta, em cada iteração e agrega a solução global. A recorrência termina quando todas as características P do objeto de dados estão representadas como nós folha de uma árvore binária, onde ao percorrer as subárvores da esquerda, desta árvore binária, imprime-se ou traduz-se "0"e a direita "1".
- 7 : codificação das características P extraídas do objeto de dados a partir do percorrimento top-down (da raiz para as folhas) na árvore binária. O processo se desenvolve pela concatenação de "0"ou de "1"extraídos do percurso das subárvores e

então acrescentados a string C de codificação. O processo sucessivo de concatenações termina ao encontrar no percurso o nó folha que mantém a informação de uma característica $p_i \in P$ específica.

- 8 - Saída: conversão de cada característica de P da string D de dados puros, em sua codificação, correspondente seguido de sua concatenação com a string C de codificação do objeto de dados.

Cada elemento presente na codificação Huffman pode ser mapeado para uma 6-upla, que implementa uma MT do tipo 3. Utiliza-se o modelo RLE de compressão como projeto elementar de uma MT, pois os processos de compressão RLE estão presentes na maioria dos processos de compressão em geral. A implementação do algoritmo de codificação Huffman, como uma máquina de Turing é exibida na tabela 4.1 e o projeto na forma de diagrama de estados apresentados na figura 4.4.

Tabela 4.1: *Compressor implementado como MT*

Implementação						
	S	S_0	Σ	Δ	δ	δ_s
MT(*Mealy*)	$\mathbb{N}$	$n_0 \in \mathbb{N}$	P	$\{0,1\}$	$S \times P \rightarrow S$	$S \times P \rightarrow \Delta^*$
Huffman	$\{P\}^*$	$p_i \in \{P\}^*$ (raiz)	$\{0,1\}$	$\{\varepsilon,0,1\}$	$\{P\}^* \times \Delta \rightarrow \{P\}^*$	$\{P\}^* \times \Delta \rightarrow \Delta$

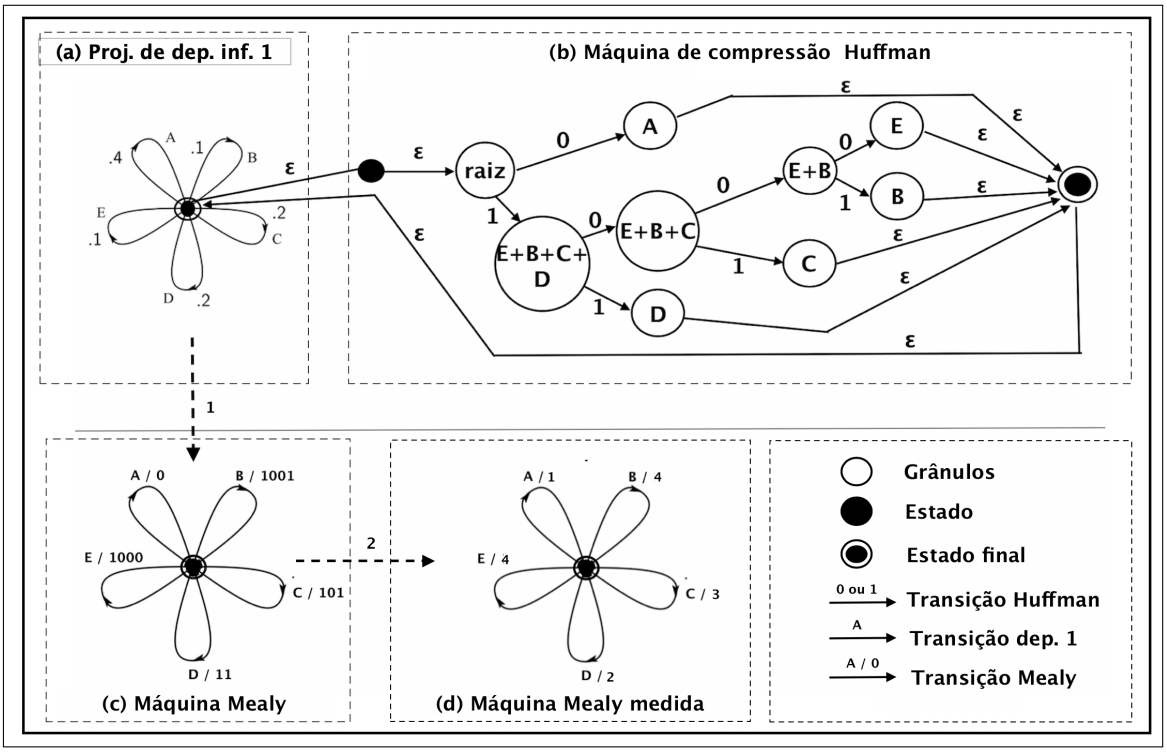

Figura 4.4: *Compressor implementado como uma máquina Mealy*

A figura 4.4 (a) representa um projeto de compressão com dependência da informação nível 1(um) , contendo, um estado inicial e final ligando todas as transições que carregam as probabilidades de ocorrência de cada grânulo: A,B,C,D e E. A figura 4.4 (b) demostra a estrutura de cada transição que compõe a MT. AS transições são concebidas a partir da árvore de codificação huffman. Na figura 4.4 (c), o processo de codificação produz uma representação C do objeto, pronta para medição. A figura 4.4 (d) representa a máquina de medição de informações. A complexidade de Kolmogorov dada por $|C|$ pode ser calculada por esta máquina.

4.1.3 Redutor

O componente *Redutor* tem a função de realizar a síntese das informações representando os objetos de dados a partir de uma estrutura de dados hierárquica. Nesta pesquisa, o modelo de arquitetura de informação que comporta e gerencia os objetos não-estruturados de dados utiliza uma árvore filogenética. O Neighbor Joining (NJ) é o algoritmo utilizado, nesta pesquisa, para a construção da árvore filogenética. Trata-se de uma heurística, para reconstrução de árvores filogenéticas, baseada no princípio da evolução mínima (minimum evolution) equivalente ao princípio da parcimônia.

4.2 Geração de hipóteses *H* com *GM*

O protótipo é utilizado para a geração de agrupamentos hierárquicos e particionados a partir de uma fonte de dados. Os experimentos baseiam-se na comparação dos agrupamentos de objetos produzidos pelo protótipo com o agrupamento ideal esperado. O protótipo possui diferentes processos de compressão representados pelo modelo de máquina de Turing. Nos experimentos realizados, confere-se ao analista de dados a aquisição de dados, a definição das classes esperadas, a rotulagem dos objetos em cada experimento e a análise dos resultados exibidos.

O protótipo[1] é escrito em linguagem Python e incorpora algoritmos pertencentes a abordagem DAMICORE (Data Mining Code Repositories) figura 4.5. Delbem em [75], implementa o fluxo de execução DAMICORE para a formação de hipóteses em fontes de dados não-estruturados, que consiste de:

- compactadores para a representação dos objetos de dados de uma fonte O;
- a métrica R = NCD [24] para a comparação de objetos;

[1] O protótipo, incluindo as fontes de dados texto e imagem, e as árvores filogenéticas que representam os agrupamentos hierárquicos de cada fonte estão disponíveis em: http://adrianobailao.com.br/tese/

- a heurística Junção de Vizinhos - NJ (Neighbor Joining) [72] baseada em parcimônia para a concepção do agrupamento de objetos;
- e um algoritmo de redes complexas FN (Fast Newman) [54] para a delimitação dos grupos particionais presentes dentro da estrutura de agrupamento hierárquico.

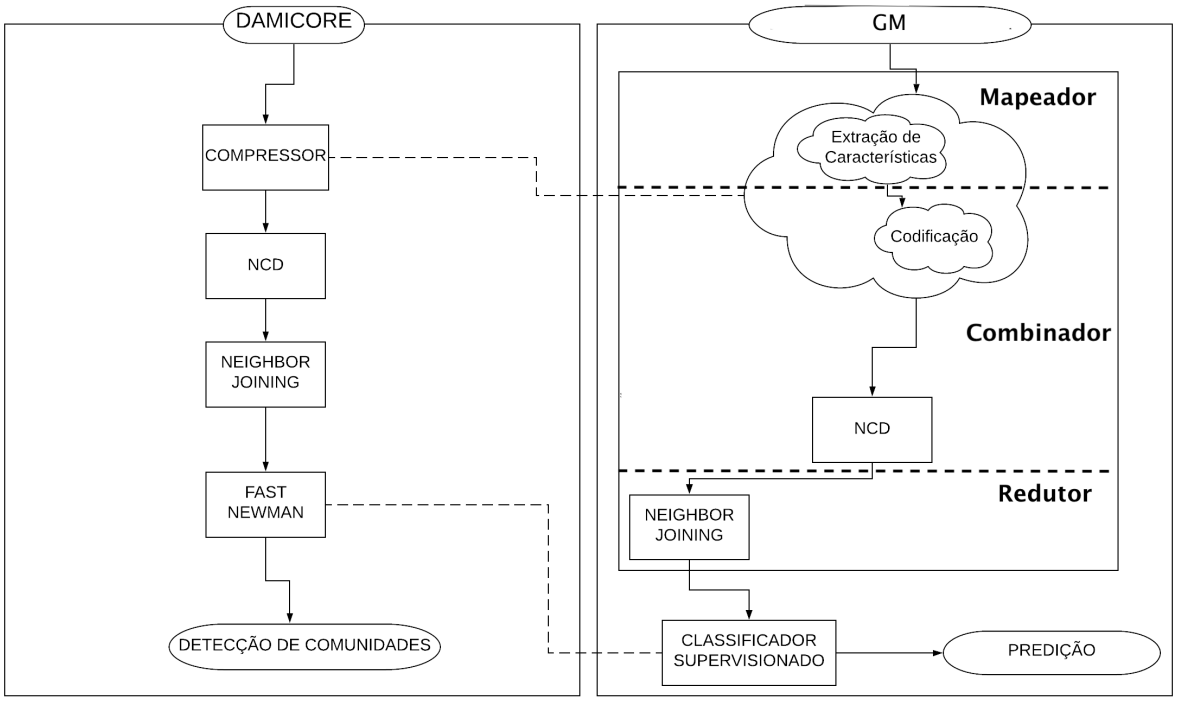

Figura 4.5: *Formação de agrupamentos*

A partir de uma fonte de dados, a ordem do fluxo de execução dos experimentos segue as setas que fazem parte da figura 4.5. A figura 4.5 ilustra também o método GM a partir dos elementos:

- extração de características realizada pelo Mapeador e orientada pelo processo adaptativo de Granulação;
- representação de cada objeto de dados como uma máquina de Turing (compressão) e sua medição efetuada através de métricas de compressão. Ambos os processos realizados pelo Combinador;
- I: formulação do agrupamento hierárquico a partir da heurística executada pelo Redutor;

As métricas utilizadas pelo protótipo são descritas na tabela 4.2.

Protótipo		
		Métrica
Objeto de dados		Complexidade de Kolmogorov
Relação entre objetos		Incerteza (conjunta e condicional) e Informação mútua
Distância entre objetos		NCD
Agrupamento particionado de objetos		Homogeneidade, Completude, V-medição, Ajuste aleatório, Informação mútua ajustada e Silhueta.
Agrupamento hierárquico de objetos		$G = \frac{\sum_{i=0}^{m-1} a_i}{m - m_{classes}}$ (ver seção 3.1.5)

Tabela 4.2: *Métricas utilizadas nos experimentos*

4.3 Fontes de dados

Cada fonte de dados possui uma quantidade n de objetos. As fontes de dados foram preparadas visando a execução de experimentos com diferentes tipos de estruturação para os dados: texto e imagem. As fontes de dados apresentam características diferentes como por exemplo o número de classes e a quantidade de objetos por classe. Todas estas variações, na concepção das fontes de dados, visam enriquecer os experimentos a fim de buscar a maior quantidade possível de cenários de classificação, e então, demostrar a genericidade do método.

4.4 Fontes de dados de texto

As fontes de dados, no formato de texto, que são utilizadas para os experimentos estão disponíveis na *web* a partir dos seguintes repositórios: *UCI Machine Learning Repository*, *Twitter*, *arxiv.org*, IMDB, com sinopses de filmes e outros diversos sites nacionais, contendo notícias de diversas categorias. Os dados coletados podem ser divididos em resumos, sentimentos e notícias, na íntegra, os quais serão explanados com mais detalhes nas subseções seguintes.

No total, foram coletados 1.162 textos, os quais pertencem as seguintes categorias:

- Sinopses de Filmes IMDB (45).

- Resumos de artigos do Arxiv.org (250).
- Comentários do e-commerce Amazon (200).
- Emoções no Twitter (300).
- Notícias do Brasil (Twitter) (300).
- Artigos Noticiários (107).
- Detecção de Idiomas (260).

São descritas nas próximas seções a coleta de dados e a categorização dos textos.

4.4.1 Análise Léxica

A Análise Léxica é o processo de analisar a entrada de linhas de caracteres (tal como o código-fonte de um programa de computador), sendo assim, os experimentos que utilizam estas fontes podem reconhecer tokens (palavras) de uma determinada linguagem e assim categorizar os textos.

Detecção de Idiomas

Para os experimentos envolvendo agrupamentos particionados, 8 (oito) fontes de dados foram preparadas visando testes com o protótipo proposto na tese através da variação, na fonte de dados, da:

- quantidade de objetos total;
- quantidade de classes;
- quantidade de objetos por classe.

Foram utilizadas bases de dados texto e os objetos rotulados pelo seu idioma correspondente. Este problema de agrupamento foi escolhido devido sua boa distinção entre os objetos de dados, sendo bem separado, pelos algoritmos da literatura. As fontes 1 a 8 são de cunho pedagógico visando a demostração de experimentos no próximo capítulo.

Fontes de dados			
Fonte	Descrição fonte	N de objetos	Expectativa
1	Arquivos texto de 2 (duas) classes de linguagem.	55	Arquivos separados em 2 (duas) classes: Francês e Alemão.
2	Arquivos texto de 5 (cinco) classes de linguagem.	15	Arquivos separados em 5 (cinco) classes: Francês, Alemão,Inglês,Português e Italiano.
3	Arquivos texto de 3 (três) classes de linguagem.	30	Arquivos separados em 3 (três) classes: Francês, Alemão e Português.
4	Arquivos texto de 4 (quatro) classes de linguagem. Característica particular do experimento: Número variado de objetos por classe.	38	Arquivos separados em 4 (quatro) classes: Francês, Alemão, Português e Italiano.
5	Arquivos texto em português de 2 (duas) classes de linguagem.	8	Arquivos separados em de 2 (duas) classes: Política Brasileira e Política Internacional.
6	Arquivos texto em português de 4 (quatro) classes de linguagem e 3 (três) subclasses para cada classe.	38	Arquivos separados em 4 (quatro)classes: Francês, Alemão, Italiano e Português e 3 (três) subclasses para cada classe: Biologia, Política Brasileira e Política Internacional.

Fonte de dados			
Fonte	Descrição fonte	N de objetos	Expectativa
7	Arquivos texto de 4 (quatro) classes de linguagem.	37	Arquivos separados em 4 (quatro) classes: Francês, Alemão, Italiano e Português.
8	Arquivos texto de 4 (quatro) classes de linguagem.	39	Arquivos separados em 4 (cinco) classes: Francês, Alemão, Italiano e Português.

4.4.2 Sumarização

Esta seção descreve a coleta de dados para textos da categoria de resumos de um arquivo textual. Os experimentos envolvendo as fontes de dados desta seção testam a capacidade de resumo (sumarização) que o algoritmo do protótipo possui.

Sinopses de Filmes IMDB

Coletou-se um total de 45 resumos de filmes da base de dados do IMDB, divididos em 7 classes, conforme mostrado na tabela 4.3.

Classe	**Objetos**
AÇÃO	9
BIOGRAFIA	7
COMÉDIA	6
DRAMA	7
HISTÓRIA	4
ROMANCE	6
GUERRA	6
Total	**45**

Tabela 4.3: *Dados coletados do IMDB*

Resumos de artigos do Arxiv.org

Foi coletado, do repositório Arxiv.org, um conjunto de 250 amostras de *abstracts* de artigos separados em 5 classes, o qual se encontra descrito na tabela 4.4.

Classe	Objetos
BIOLOGIA	50
QUÍMICA	50
CIÊNCIA DA COMPUTAÇÃO	50
FÍSICA	50
CIÊNCIAS SOCIAIS	50
Total	**250**

Tabela 4.4: *Dados coletados do Arxiv.org*

4.4.3 Análise de Sentimentos

As tabelas 4.5, 4.6 e 4.7 são relacionadas as quantidades de objetos coletados com suas respectivas classes de sentimento. A partir das fontes de dados de sentimentos pode-se avaliar a capacidade de generalização que o algoritmo do protótipo possui, referente a categorizar textos com características psicológicas.

Comentários do e-commerce Amazon

Realizou-se coletas dos comentários de compradores, os quais demostraram seu nível de satisfação após adquirir produtos da loja virtual Amazon, um total de 200 comentários separados em 2 classes, conforme ilustra a tabela 4.5.

Classe	Objetos
NEGATIVO	100
POSITIVO	100
Total	**200**

Tabela 4.5: *Dados coletados da Amazon*

Emoções no Twitter

Coletou-se 300 tweets formando uma base de dados composta por 6 classes distintas capazes de expressar emoções no conteúdo das postagens. As informações dos textos coletados são descritas na tabela 4.6.

Classe	Objetos
TRISTEZA	50
SURPRESA	50
ALEGRIA	50
DESGOSTO	50
RAIVA	50
MEDO	50
Total	**300**

Tabela 4.6: *Dados coletados do Twitter*

Notícias do Brasil (Twitter)

Notícias do cenário político e econômico brasileiro do ano de 2016 foram coletadas a fim de expressar a relação existente entre o sentimento de positividade, negatividade e neutro, com relação a estes temas, alcançando um total de 300 textos coletados. A tabela 4.7 descreve como foi dividida a coleta.

Classe	Objetos
POSITIVO	100
NEGATIVO	100
NEUTRO	100
Total	**300**

Tabela 4.7: *Dados coletados do Twitter*

Artigos Noticiários

Foram coletados um total de 107 artigos de notícias, em diversos portais brasileiros, categorizados cada um com sua especialidade. No total foram 5 classes, mais detalhes podem ser observados na tabela 4.8.

Classe	Objetos
ECONOMIA	15
ESPORTE	21
POLÍTICA	17
SAÚDE	27
TECNOLOGIA	27
Total	**107**

Tabela 4.8: *Dados coletados de sites de notícias nacionais*

4.5 Fontes de dados de imagem

As fontes de dados de imagem são coletadas visando testes envolvendo problemas de reconhecimento de padrões, em particular, visão computacional. Através da extensão da metodologia de mineração de dados DAMICORE ilustrada na figura 4.5, uma aplicação (protótipo) em *python* foi implementada com capacidade de gerar de forma não supervisionada, um modelo hierárquico de informação. Para isso, também, foram coletadas imagens na internet para realização de experimentos.

Para a avaliação da qualidade dos modelos de agrupamento, são testados diversos conjuntos de dados $D(n)$ de imagens, cuja característica é inerente a algum problema de agrupamento encontrado, na literatura, da seguinte forma:

- Frutas, Animais e Captcha: detecção de objetos na cena de uma imagem digital;
- Pinturas Abstratas: detecção da estilo artístico em pinturas;
- Plantações: detecção da textura de uma imagem digital;
- Faces humanas: detecção de faces humanas em uma imagem digital;
- Objetos variados: detecção de objetos na cena e detecção da textura de uma imagem digital;

Grande parte das imagens foram coletadas através de serviços de busca como o *google search*, outras pertencem a conjuntos de dados disponibilizados por terceiros, e para uma categoria em especial, a categoria de imagens do teste CAPTCHA, as imagens foram coletadas através da captura de tela em sites que fazem uso do mesmo. Foram coletadas um total de 3435 imagens, sendo elas pertencentes as seguintes categorias:

- Frutas (102).
- Animais (1232).
- Plantações (1322).
- CAPTCHA (124).
- Faces Humanas (400).
- Pinturas (255).

A seguir, tem-se uma descrição de todos os conjuntos de imagens descritos por diferentes categorias de imagens coletadas. As categorias estão organizadas em subseções que possuem informações do conjunto de dados, além de uma tabela com as classes e quantidade de instâncias coletada por classe.

4.5.1 Conjunto de imagens de frutas

Foram coletadas um total de 102 imagens de frutas através de buscas na internet em provedores, como, *google search, pixabay, flickr e shutterstock*. As imagens estão divididas em 5 classes.

Tabela 4.9: *Conjunto de dados 1 - Frutas*

CLASSE	OBJETOS
BANANA	20
LARANJA	20
LIMÃO	20
MAÇA	20
MORANGO	22
TOTAL	102

As imagens estão em diversas resoluções, com diferentes perspectivas de capturas, e podem conter uma ou várias unidades de alguma fruta. A figura 4.6 mostra exemplos de imagens coletadas nessa categoria.

Figura 4.6: *Exemplos de imagens coletadas na categoria de frutas*

4.5.2 Conjunto de Imagens de Plantações

Foram coletadas um total de 1322 imagens de plantações através de provedores de busca na internet, como, *google search, pixabay, flickr e shutterstock*. As imagens estão divididas em 12 classes.

Tabela 4.10: *Conjunto de dados 2 - Plantações*

CLASSE	OBJETOS
ALGODÃO	109
ARROZ	116
MILHO	110
GIRASSOL	101
MANDIOCA	100
CANAVIAL	106
MAMÃO	110
BANANA	113
JABUTICABA	111
ABACAXI	117
TOMATE	130
ALFACE	110
TOTAL	1322

Figura 4.7: *Exemplos de imagens coletadas na categoria de plantações*

As imagens de plantações são distribuídas em 12 classes conforme a tabela 4.10 e estão em diversas resoluções e com diferentes perspectivas de captura. As imagens possuem grandes ou pequenas plantações (que podem ter frutos visíveis ou não), e algumas (uma pequena parcela) apenas frutos provenientes da plantação. A figura 4.7 apresenta alguns exemplos de imagens que foram coletadas nessa categoria.