Técnicas de projeção para identificação de grupos e comparação de dados multidimensionais usando diferentes medidas de similaridade

Paulo Joia Filho

Paulo Joia Filho

Técnicas de projeção para identificação de grupos e comparação de dados multidimensionais usando diferentes medidas de similaridade

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Luis Gustavo Nonato

USP – São Carlos
Dezembro de 2015

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

J74t

Joia, Paulo
Técnicas de projeção para identificação de grupos e comparação de dados multidimensionais usando diferentes medidas de similaridade / Paulo Joia; orientador Luis Gustavo Nonato. -- São Carlos, 2015.
170 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2015.

1. Projeção de dados multidimensionais. 2. Visualização de dados. 3. Agrupamento de dados. 4. Busca por similaridade. 5. Modelagem de incerteza. I. Nonato, Luis Gustavo, orient. II. Título.

Paulo Joia Filho

Projection techniques for group identification and multidimensional data comparison by using different similarity measures

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Luis Gustavo Nonato

USP – São Carlos
December 2015

À minha amada esposa Andréia,
que pela confiança e apoio sempre
me impulsionou a seguir adiante.

Agradecimentos

Agradeço a Deus pela oportunidade de concluir mais este trabalho.

À minha amada esposa Andréia Chudrik Jóia, pela paciência e cooperação incessante nesta árdua tarefa. Ainda que eu me esforce, palavras são insuficientes para traduzir sua imensa ajuda. Ter você ao meu lado é, de fato, um grande privilégio. Que nossos caminhos sigam para sempre juntos.

Meus sinceros agradecimentos ao Prof. Dr. Luis Gustavo Nonato, pela orientação e seriedade com que considerou todas as atividades que realizamos. Sempre disposto a esclarecer todas as dúvidas, jamais esquecendo-se de qualquer compromisso.

Ao colega Fabiano Petronetto do Carmo (UFES), pelo apoio nas tarefas que realizamos, pelas críticas construtivas visando melhorar a qualidade do trabalho, pela amizade e palavras de incentivo.

Aos professores Dra. Carla Maria Dal Sasso Freitas (UFRGS), Dr. Fernando Vieira Paulovich (ICMC-USP), Dr. Hélio Pedrini (UNICAMP) e Dr. Hugo Alexandre Dantas do Nascimento (UFG), pelos comentários e sugestões que contribuiram para a versão final desta tese.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pela concessão da bolsa de doutorado (Processo 2010/07367-9), cujo apoio financeiro foi essencial para a realização desta pesquisa.

"Se uma inteligência pudesse saber a posição de todas as partículas de matéria em dado momento, todas as nossas dúvidas se dissipariam e o futuro e o passado se descortinariam diante de nossos olhos".

Pierre Simon Laplace

Resumo

JOIA, P. **Técnicas de projeção para identificação de grupos e comparação de dados multidimensionais usando diferentes medidas de similaridade**. 2015. 170f. Tese (Doutorado) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP, 2015.

Técnicas de projeção desempenham papel importante na análise e exploração de dados multidimensionais, já que permitem visualizar informações muitas vezes ocultas na alta dimensão. Esta tese explora o potencial destas técnicas para resolver problemas relacionados à: *1) identificação de agrupamentos* e *2) busca por similaridade* em dados multidimensionais. Para identificação de agrupamentos foi desenvolvida uma técnica de projeção local e interativa que, além de projetar dados com ótima preservação de distâncias, permite que o usuário modifique o *layout* da projeção, agrupando um número reduzido de amostras representativas no espaço visual, de acordo com suas características. Os mapeamentos produzidos tendem a seguir o *layout* das amostras organizadas pelo usuário, facilitando a organização dos dados e identificação de agrupamentos. Contudo, nem sempre é possível selecionar ou agrupar amostras com base em suas características visuais de forma confiável, principalmente quando os dados não são rotulados. Para estas situações, um novo método para identificação de agrupamentos baseado em projeção foi proposto, o qual opera no espaço visual, garantindo que os grupos obtidos não fiquem fragmentados durante a visualização. Além disso, é orientado por um mecanismo de amostragem determinístico, apto a identificar instâncias que representam bem o conjunto de dados como um todo e capaz de operar mesmo em conjuntos de dados desbalanceados. Para o segundo problema: busca por similaridade em dados multidimensionais, uma família de métricas baseada em classes foi construída para projetar os dados, com o objetivo de minimizar a dissimilaridade entre pares de objetos pertencentes à mesma classe e, ao mesmo tempo, maximizá-la para objetos pertencentes a classes distintas. As métricas classes-específicas são avaliadas no contexto de recuperação de imagens com base em conteúdo. Com o intuito de aumentar a precisão da família de métricas classes-específicas, outra técnica foi desenvolvida, a qual emprega a teoria dos conjuntos *fuzzy* para estimar um valor de incerteza que é transferido para a métrica, aumentando sua precisão. Os resultados confirmam a efetividade das técnicas desenvolvidas, as quais representam significativa contribuição na tarefa de identificação de grupos e busca por similaridade em dados multidimensionais.

Palavras-chave: Projeção de dados multidimensionais. Visualização de dados. Agrupamento de dados. Busca por similaridade. Modelagem de incerteza.

Abstract

JOIA, P. **Projection techniques for group identification and multidimensional data comparison by using different similarity measures**. 2015. 170s. Thesis (Doctoral) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP, 2015.

Projection techniques play an important role in multidimensional data analysis and exploration, since they allow to visualize information frequently hidden in high-dimensional spaces. This thesis explores the potential of those techniques to solve problems related to: *1) clustering* and *2) similarity search* in multidimensional data. For clustering data, a local and interactive projection technique capable of projecting data with effective preservation of distances was developed. This one allows the user to manipulate a reduced number of representative samples in the visual space so as to better organize them. The final mappings tend to follow the layout of the samples organized by the user, therefore, the user can interactively steer the projection. This makes it easy to organize and group large data sets. However, it is not always possible to select or group samples visually, in a reliable manner, mainly when handling unlabeled data. For these cases, a new clustering method based on multidimensional projection was proposed, which operates in the visual space, ensuring that clusters are not fragmented during the visualization. Moreover, it is driven by a deterministic sampling mechanism, able to identify instances that are good representatives for the whole data set. The proposed method is versatile and robust when dealing with unbalanced data sets. For the second problem: similarity search in multidimensional data, we build a family of class-specific metrics to project data. Such metrics were tailored to minimize the dissimilarity measure among objects from the same class and, simultaneously to maximize the dissimilarity among objects in distinct classes. The class-specific metrics are assessed in the context of content-based image retrieval. With the aim of increasing the precision of the class-specific metrics, another technique was developed. This one, uses the fuzzy set theory to estimate a degree of uncertainty, which is embedded in the metric, increasing its precision. The results confirm the effectiveness of the developed techniques, which represent significant contributions for clustering and similarity search in multidimensional data.

Keywords: Multidimensional data projection. Data visualization. Clustering. Similarity search. Uncertainty modeling.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Abreviaturas e Siglas

ACC	*Accuracy Coefficient*
ANSI	*American National Standards Institute*
auc	*area under the curve*
CBIR	*Content-Based Image Retrieval*
CPU	*Central Processing Unit*
CSE	*Classifier Subset Evaluation*
CSM	*Column Selection Method*
CSMP	*Class-Specific Multidimensional Projection*
CSWIRe	*Class-Specific with Weight Image Retrieval*
CV	*Coefficient of Variation*
DS t-SNE	*Doubly Supervised t-SNE*
DWT	*Discrete Wavelet Transform*
EM	*Expectation Maximization*
FAE	*Filtered Attribute Evaluation*
FARG	*Fuzzy Attributed Relational Graph*
FF	*Farthest First*
FIRE	*Flexible Image Retrieval Engine*
FSE	*Filtered Subset Evaluation*
FSM	*Fuzzy Similarity Measures*
FTN	*Fuzzy Triangular Number*
GA-CBIR	*Genetic Algorithm CBIR*
HSI	*Hue, Saturation, and Intensity*
I2CDR	*Image-to-class distance ratio*
iLAMP	*inverse-LAMP*
IRP-Kmeans	*Iterative Random Projections K-means*
Isomap	*Isometric Feature Mapping*
JPEG	*Joint Photographics Experts Group*
JSE	*Jensen–Shannon Embedding*
k-NN	*k-Nearest Neighbors*
KL	*Kullback-Leibler*

KTP	*Kernel-based Transition Probability*
L-Isomap	*Landmark Isomap*
LAMP	*Local Affine Multidimensional Projection*
LAS2	*Lanczos Algorithm in SVD*
LIRe	*Lucene Image Retrieval*
LLE	*Locally Linear Embedding*
LMDS	*Landmark MDS*
LMT	*Logistic Model Tree*
LoCH	*Local Convex Hull*
LSA	*Latent Semantic Analysis*
LSP	*Least Square Projection*
MDBC	*Make Density Based Cluster*
MDS	*Multidimensional Scaling*
MS	*Multi-Scale*
NeRV	*Neighbourhood Retrieval and Visualisation*
NNG	*Nearest Neighbors Graph*
ONPP	*Orthogonal Neighborhood Preserving Projections*
PAn	*Projection Analyzer*
PCA	*Principal Component Analysis*
PLMP	*Part-Linear Multidimensional Projection*
PLP	*Piecewise Laplacian-based Projection*
PSO	*Particle Swarm Optimization*
QDA	*Quadratic Discriminant Analysis*
RAM	*Random Access Memory*
RF	*Relevance Feedback*
RL-Sim	*Ranked Lists Similarities*
RP	*Random Projection*
SC	*Star Coordinates*
Silh	*Overall Average Silhouette Width*
SIMPLIcity	*Semantics-sensitive Integrated Matching for Picture Libraries*
SKM	*Simple K-Means*
SM	*Sammon's Mapping*
SNE	*Stochastic Neighbor Embedding*
SSFS	*Subset Size Forward Selection*
SVD	*Singular Value Decomposition*
t-SNE	*t-Distributed Stochastic Neighbor Embedding*
UTOPIAN	*User-driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization*
WCDS	*Wholesale Customers Data Set*
WEKA	*Waikato Environment for Knowledge Analysis*
WSE	*Wrapper Subset Evaluation*
XM	*XMeans*

Capítulo

1

Introdução

O avanço da computação tem permitido armazenar quantidades cada vez maiores de informações. Encontrar formas de organizar tais informações, de modo a extrair características e agrupá-las segundo suas semelhanças é um fator requerido aos sistemas computacionais modernos. A exploração visual da informação desempenha um papel importante neste processo, graças à capacidade de percepção visual do sistema cognitivo humano, assim como técnicas de projeção, as quais permitem visualizar informações muitas vezes ocultas na alta dimensão. Assim sendo, esta tese explora o potencial das técnicas de visualização de informação com ênfase em projeção para auxiliar na identificação de agrupamentos e busca por similaridade em dados multidimensionais.

1.1 Contextualização

Visualização de informação é a área da visualização responsável em auxiliar na representação e entendimento de dados abstratos que podem ter ou não relação com o espaço físico que nos cerca (dimensão espacial) (Mazza, 2009).

Neste trabalho, *dado* é qualquer objeto descrito por um ou mais atributos. Quando o objeto tem um único atributo, estamos nos referindo a dados unidimensionais. Quando o objeto tem múltiplos atributos (normalmente acima de três), estamos nos referindo a dados multidimensionais. Os atributos também são chamados de variáveis ou características.

Em um conjunto de dados multidimensionais, o número de atributos indica sua dimensão. Além disso, os atributos constituem o *espaço de atributos* ou *espaço de características* do conjunto de dados. Nesta tese, cada elemento do conjunto de dados poderá ser tratado

indistintamente como um objeto, uma instância, um vetor de atributos (ou de características) ou simplesmente um ponto. Quando um conjunto de dados apresenta um atributo para identificação de classes, categorias ou grupos, é indicado como rotulado.

Uma questão que surge neste momento é como visualizar dados multidimensionais se os dispositivos de saída mais comuns restringem-se ao domínio bidimensional? Muitas técnicas de visualização de informação foram criadas para auxiliar nesta tarefa, entre as quais estão: 1) *geométricas*: matriz de *scatter plots* (Andrews, 1972), coordenadas paralelas (Inselberg e Dimsdale, 1990), conjuntos paralelos (Kosara et al., 2006); 2) *iconográficas*: faces de *Chernoff* (Chernoff, 1973), *star plots* (Chambers et al., 1983); 3) *orientadas por pixels* (Keim, 2000); 4) *hierárquicas*: empilhamento de dimensões (LeBlanc et al., 1990), *treemaps* (Shneiderman, 1991), *cone trees* (Robertson et al., 1991), árvores hiperbólicas (Lamping et al., 1995). Outras taxionomias e técnicas podem ser encontradas em Ferreira de Oliveira e Levkowitz (2003), Masseglia et al. (2007) e Mazza (2009).

Outra possibilidade é reduzir a dimensionalidade do espaço de origem, transformando os dados com base em um novo espaço de dimensão 1, 2 ou 3 (usualmente 2), de modo a preservar tanto quanto possível *proximidades* entre instâncias nos dois espaços. Proximidade corresponde à medida usada para expressar a *similaridade* ou *dissimilaridade* entre pares de instâncias (Härdle e Simar, 2007). O novo espaço obtido é chamado *espaço visual* e esta abordagem denomina-se *projeção ou mapeamento de dados multidimensionais*, assunto de interesse desta pesquisa.

Medida de similaridade é usada para indicar quão semelhantes são dois objetos. Ao contrário desta, a medida de dissimilaridade indica quão diferentes são os objetos do domínio em estudo. Métricas podem ser usadas como medida de similaridade, no entanto, restringem o conjunto de possíveis medidas a satisfazerem os *postulados de espaço métrico* (Definição 2.11). Uma busca por similaridade corresponde ao processo de busca onde o único critério de comparação é a medida de similaridade entre pares de objetos (Yu, 2002). Estes conceitos são detalhados no Capítulo 2.

Técnicas de projeção se aplicam a diferentes propósitos, mas em geral, ajudam a visualizar dados multidimensionais, confirmar hipóteses e revelar informações ocultas na alta dimensão. Existem várias técnicas capazes de projetar dados multidimensionais, que vão desde a tradicional *Principal Component Analysis* (PCA) (Pearson, 1901), passando por um conjunto de técnicas baseadas em *Multidimensional Scaling* (MDS) (Borg e Groenen, 2005), até técnicas mais atuais que permitem a interação do usuário no processo da projeção. Outras subdivisões são propostas por Maaten et al. (2009). Vale lembrar ainda que, a escolha da técnica de projeção tem implicações diretas na interpretação dos resultados (Tejada et al., 2003) e no tipo de aplicação a que se destina.

Quanto à natureza da projeção, as técnicas podem ser locais ou globais. Abordagens locais tentam preservar a geometria local dos dados. Em outras palavras, isto significa que o mapeamento de cada instância depende exclusivamente das amostras em sua vizinhança,

ao passo que abordagens globais tentam preservar a geometria em todas as escalas (De Silva e Tenenbaum, 2003; Joia et al., 2011).

Técnicas de projeção interativas permitem a intervenção do usuário no processo. Em geral, os recursos interativos são explorados via representação visual da informação, a qual tende a facilitar a identificação de padrões, agrupamentos, comportamentos ou correlações nos dados. No entanto, poucas técnicas viabilizam mecanismos de interação verdadeiramente versáteis, resumindo-se a operações mais simples, como seleção de regiões no plano de projeção. Técnicas capazes de reposicionar e/ou reagrupar interativamente os dados no espaço visual fazem parte de um universo mais restrito, alguns representantes desta categoria são a LSP (Paulovich et al., 2008), a PLMP (Paulovich et al., 2010b) e a PLP (Paulovich et al., 2011). Nestas técnicas, a interação se dá pela manipulação de amostras representativas ou pontos de controle, projetados a priori, logo nos primeiros passos da técnica. A Figura 1.1 ilustra, de modo generalizado, os passos desse processo interativo.

Figura 1.1: *Pipeline* das técnicas de projeção que permitem a interação do usuário no *layout* da projeção, através da manipulação de amostras representativas.

A quantidade de pontos de controle representa a maior limitação das técnicas que seguem o esquema interativo mostrado na Figura 1.1. Em geral, essas técnicas requerem um grande número de pontos de controle para realizar o mapeamento com qualidade, dificultando a interação do usuário.

Projeções locais e interativas são particularmente úteis neste contexto. Estes recursos são utilizados na LAMP, uma técnica capaz de projetar dados com base em uma pequena quantidade de pontos de controle, detalhada no Capítulo 4.

1.2 Motivação

Consideráveis avanços têm sido observados nas técnicas de projeção nos últimos anos, com aplicações em diferentes domínios: interação e visualização de campos vetoriais (Daniels II et al., 2010), análise visual de redes sociais (Martins et al., 2012), exploração de espaços tridimensionais complexos (visualização de fibras neuronais) (Poco et al., 2012), simulação de partículas (Santos et al., 2013), visualização de sequências genômicas (RNA) (Demiralp et al., 2013), visualização de coleções de música (Soriano et al., 2014), visualização de matrizes (Behrisch et al., 2014), simulação com dados variantes no tempo (Molchanov e Linsen, 2014), classificação de coleções de imagens (Paiva et al., 2015), entre outras.

Apesar da contínua evolução, alguns problemas relacionados ao tema ainda são frequentemente abordados, tais como:

1. Identificação de agrupamentos.
2. Busca por similaridade em dados multidimensionais.

Algumas técnicas propõem soluções para estes problemas, contudo, longe da solução ideal. Por exemplo, identificar grupos é uma tarefa complexa e na maioria das vezes os grupos obtidos não correspondem à verdadeira natureza dos dados, geralmente organizados com base exclusiva na geometria. Quanto à busca por similaridade, existem várias abordagens, todavia, poucas empregam medidas de similaridade realmente aptas a discriminar objetos de acordo com as classes existentes.

1.3 Objetivos

Este trabalho tem como objetivo principal, explorar o potencial das técnicas de projeção com o auxílio de recursos visuais e interativos para resolver problemas relacionados à *identificação de agrupamentos* e *busca por similaridade em dados multidimensionais*. A partir daí, os seguintes objetivos específicos foram propostos:

- Desenvolver uma técnica de projeção local, guiada por um pequeno subconjunto de pontos de controle a fim de permitir operações efetivamente interativas, capaz de gerar projeções com qualidade e eficiência.
- Identificar instâncias representativas em conjuntos de dados não rotulados que correspondam a um padrão específico dos dados, com base em um critério determinístico.
- Utilizar as instâncias representativas obtidas para detectar agrupamentos de dados.
- Criar uma família de métricas baseada em classes para otimizar buscas por similaridade em dados multidimensionais, de modo que instâncias pertencentes à mesma classe sejam projetadas próximas umas das outras e instâncias pertencentes à classes distintas fiquem separadas no *layout* da projeção.
- Calcular a incerteza decorrente da utilização da família de métricas classes-específicas.
- Inserir informações de incerteza na família de métricas classes-específicas, a fim de obter melhores resultados nas buscas por similaridade.

As possíveis soluções usadas para guiar o processo de investigação desta pesquisa são apresentadas a seguir, por meio de hipóteses.

Hipótese 1 *Uma técnica de projeção interativa, que utilize poucos pontos de controle para guiar a projeção e que tenha a mesma flexibilidade mostrada na Figura 1.1, isto é, capaz de reagir a diferentes entradas do usuário com boa precisão, à medida que ele modifica o layout dos pontos de controle arrastando-os de modo a agrupá-los, segundo algum critério de classificação ou associação, é de fundamental importância para a análise visual de dados e identificação de agrupamentos.*

Uma técnica de projeção com as características acima é adequada para análise de dados rotulados como mostrado na Figura 1.2(a), onde cada classe é representada por uma cor diferente, ou quando os objetos analisados apresentam características visuais que permitam agrupamento por similaridade como, por exemplo, imagens, vídeos, formas geométricas e outros, conforme ilustrado na Figura 1.2(b), caracterizando desta forma um processo supervisionado (estes conceitos são discutidos na Seção 2.2). Mas, e no caso de dados não rotulados, como ilustrado na Figura 1.2(c)?

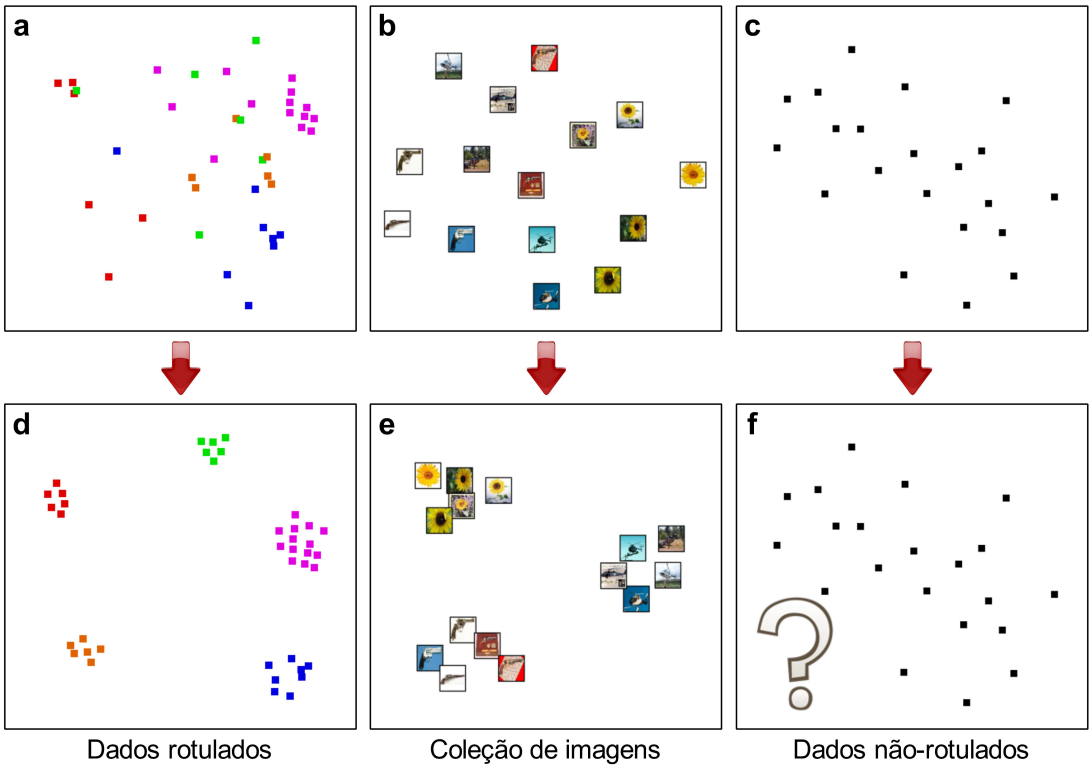


Figura 1.2: Exemplos de pontos de controle para diferentes tipos de dados: (a), (b) e (c) antes da intervenção do usuário; (d), (e) e (f) após a intervenção do usuário.

A questão acima sugere a seguinte hipótese:

Hipótese 2 *Um método de amostragem determinístico, capaz de selecionar instâncias em conjuntos de dados não rotulados que correspondam a um padrão específico dos dados (variação ou correlação, por exemplo), pode contribuir significativamente na tarefa de*

identificação de agrupamentos. Se o novo método for baseado em projeção multidimensional, a qual opera no espaço visual, isto pode garantir que os grupos obtidos não fiquem fragmentados durante a visualização.

Em grupos não fragmentados, os elementos aparentemente dispersos são agregados e posicionados em torno de seu centro (Palumbo et al., 2008), implicando melhor coesão e separação (ver Definição 2.8).

Para o segundo problema mencionado, isto é, busca por similaridade em dados multidimensionais, considere um conjunto de dados onde as classes não são conhecidas, mas cujos objetos podem ser identificados visualmente como, por exemplo, uma coleção de imagens. Neste caso:

Hipótese 3 *Um pequeno subconjunto de amostras escolhido, interativamente, a partir deste conjunto, de modo a conter objetos de diferentes categorias, pode ser classificado e os melhores atributos de cada classe utilizados para compor uma família de métricas classes-específicas. A nova métrica (medida de similaridade) construída, pode ser expandida para comparar quaisquer pares de objetos do conjunto original, de modo que instâncias pertencentes à mesma classe fiquem próximas e, instâncias pertencentes a diferentes classes fiquem afastadas durante a projeção.*

A próxima hipótese se refere à incerteza inerente ao processo de classificação:

Hipótese 4 *A incerteza inerente ao processo de classificação e seleção de atributos do subconjunto de amostras, sugerido na hipótese anterior, pode ser calculada e introduzida na família de métricas classes-específicas para aumentar a acurácia do processo de busca por similaridade.*

1.4 Contribuições

Este projeto de doutorado produziu resultados significativos na área de visualização de informação, com ênfase em projeção multidimensional. Resultados que podem ser comprovados pelas novas técnicas desenvolvidas e respectivas metodologias empregadas no desenvolvimento, conforme sumarizado a seguir.

- **Técnica de projeção local interativa**. *Local Affine Multidimensional Projection* (LAMP) (Joia et al., 2011) permite manipular pontos de controle no espaço visual de modo a organizá-los, possibilitando ao usuário guiar a projeção, porém com uma grande vantagem sobre as demais técnicas que se apoiam em subconjunto de amostras: requer um número muito reduzido de pontos de controle como entrada, tornando-se ideal para aplicações interativas. Tem formulação matemática baseada

em mapeamentos ortogonais, garantindo ótima preservação de distâncias durante a projeção multidimensional, e não depende de grafos de vizinhança para construir o mapeamento. É altamente precisa, com baixo custo computacional, apropriada para aplicações interativas envolvendo grandes volumes de dados. LAMP destaca-se como técnica do estado da arte em relação à preservação de distâncias e eficiência computacional, além de permitir projetar dados explorando tanto relações globais como locais entre instâncias, de maneira efetiva. A Figura 1.3 mostra o potencial da LAMP ao estabelecer uma correlação visual entre conjuntos de dados, a princípio, sem qualquer conexão.

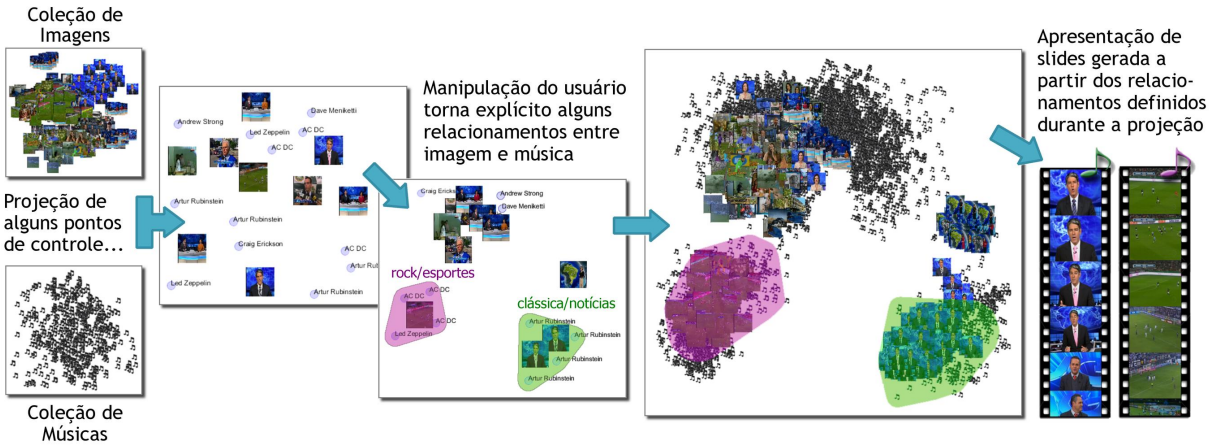


Figura 1.3: Utilizando a LAMP para correlacionar dados de diferentes naturezas. Inicialmente, uma projeção é criada para cada conjunto de dados, a partir de algumas amostras. A correlação entre as amostras é definida pelo usuário, agrupando objetos no espaço visual (imagens e músicas). Em seguida, os dados são projetados segundo as associações criadas pelo usuário. Por fim, as listas de objetos associados são usadas para criar uma apresentação de *slides* onde imagens e músicas são reproduzidas de forma sincronizada.

- **Método para identificação de grupos com base em projeção**. *Column Selection Method* (CSM) (Joia et al., 2015), um método de visualização apoiado em projeção multidimensional que permite agrupar dados. CSM opera no espaço visual, garantindo que os grupos obtidos não fiquem fragmentados durante a visualização. É orientado por um mecanismo de amostragem determinístico, capaz de identificar instâncias representativas que correspondem a um certo padrão nos dados. O mecanismo de amostragem é baseado em decomposição matricial (SVD) e capaz de operar mesmo em conjuntos de dados desbalanceados (Definição 2.2). Além de identificar instâncias representativas, o mecanismo de amostragem pode ser modificado para identificar os atributos mais relevantes de cada agrupamento obtido. Portanto, em um único *framework*, três tarefas são contempladas: *amostragem de dados*, *detecção de agrupamentos* e *seleção de atributos*. A Figura 1.4 ilustra a metáfora visual utilizada pela CSM para representar grupos e atributos, por meio de superfícies e nuvens de palavras.

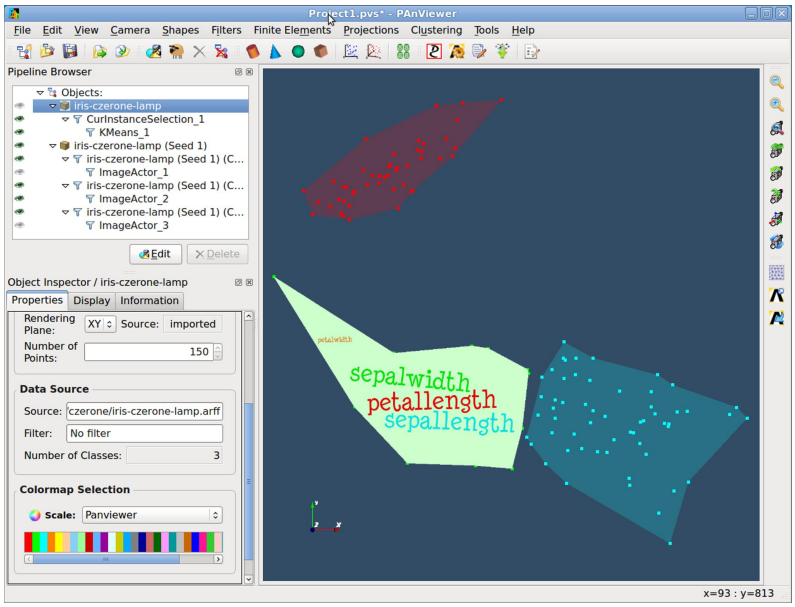


Figura 1.4: Metáfora visual utilizada pela CSM para representar grupos e atributos.

- **Família de métricas classes-específicas**. Muitas técnicas propõem medidas de similaridade para comparar dados multidimensionais, mas nenhuma diretamente relacionada às classes de objetos existentes no conjunto de dados. A *Class-Specific Multidimensional Projection* (CSMP) (Joia et al., 2012) é uma técnica de projeção baseada em uma família de métricas específicas por classe para projetar e comparar dados multidimensionais. As métricas são obtidas pela seleção dos atributos que melhor representam cada classe do conjunto de dados, de modo a minimizar a dissimilaridade entre pares de objetos pertencentes à mesma classe e, ao mesmo tempo, maximizá-la para objetos pertencentes a classes distintas. As métricas classes-específicas são avaliadas no contexto de recuperação de imagens por conteúdo para encontrar imagens similares a uma dada imagem de consulta. A lista de imagens similares pode ser retornada pelo sistema ou selecionada diretamente pelo usuário, a partir do *layout* da projeção, conforme exemplificado na Figura 1.5.

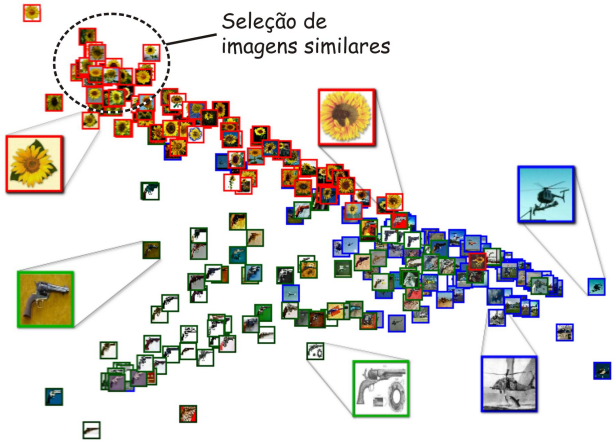


Figura 1.5: Seleção de imagens similares com o uso da CSMP.

- **Cálculo de incerteza na família de métricas classes-específicas**. Com o intuito de aumentar a precisão da família de métricas classes-específicas empregada na CSMP, uma nova técnica denominada *Class-Specific with Weight Image Retrieval* (CSWIRe) foi desenvolvida. Nesta abordagem, o usuário constrói um modelo a partir de um subconjunto de imagens, denominado "*modelo de classes*". A seguir, um classificador é aplicado sobre este modelo, retornando as melhores características e pesos que representam cada classe do modelo. Utilizando a teoria dos conjuntos *fuzzy*, um valor de incerteza é então calculado e associado à resposta do classificador para derivar uma família de métricas classes-específicas com pesos utilizada para comparar imagens com maior precisão. A Figura 1.6 ilustra o processo de recuperação de imagens da CSWIRe, utilizando uma interface gráfica apropriada.

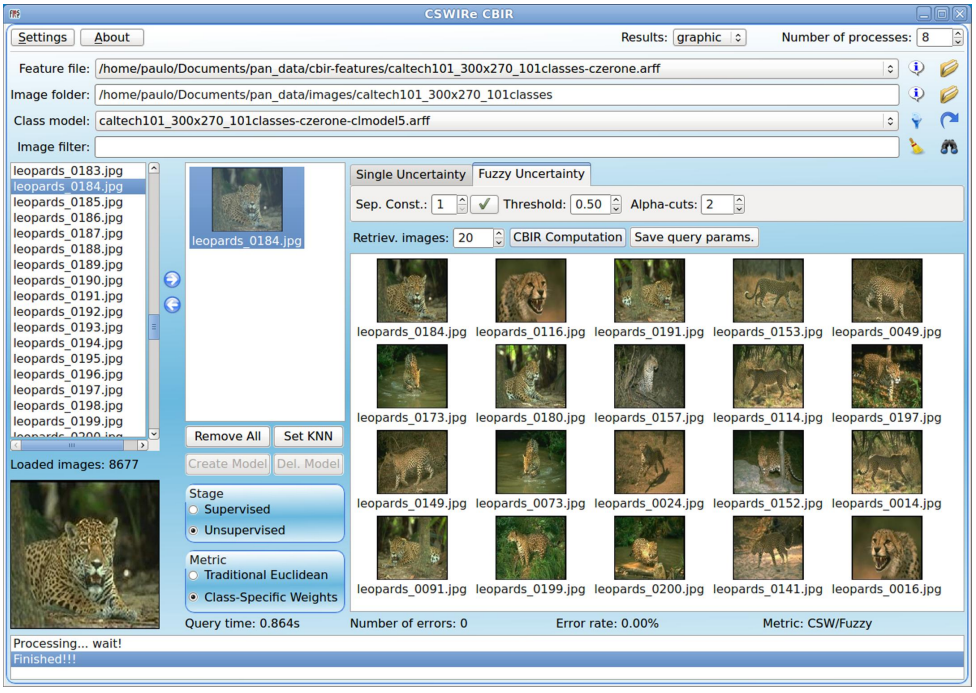


Figura 1.6: Interface gráfica da CSWIRe, mostrando o processo de recuperação de imagens por conteúdo.

As contribuições listadas acima são discutidas em detalhes nesta tese. Além destas, merece destaque o trabalho desenvolvido em colaboração com outro grupo de pesquisa[1], visando a exploração de espaços multidimensionais via projeção inversa, com uso da técnica intitulada *inverse-LAMP* (iLAMP). Esta técnica executa a projeção inversa através de mapeamentos locais afins que preservam a distância entre as novas amostras de modo preciso, já que ela segue os mesmos preceitos da LAMP. Desse modo, o usuário pode interativamente criar instâncias no conjunto de dados original, gerando assim, dados multidimensionais sintéticos além dos já existentes na disposição inicial. Para maiores detalhes sobre iLAMP, consulte Dos Santos Amorim et al. (2012).

[1] *Interactive Reservoir Modeling and Visualization Group*, Universidade de Calgary, Alberta, Canadá.

A listagem completa dos artigos científicos publicados durante este projeto de doutorado pode ser consultada no Apêndice A.

Para atender os requisitos do projeto, foram desenvolvidas algumas ferramentas computacionais. *Projection Analyzer* (PAn), um conjunto de bibliotecas em ANSI C, cuja versão inicial está disponível a partir de `http://sites.google.com/site/paulojoiafilho/tools`. Também foi implementado um módulo em Python, para facilitar a execução das tarefas. Este módulo, além de reutilizar o código em C, permite integração com pacotes de matemática numérica conhecidos (Langtangen, 2008). As interfaces gráficas mostradas nas Figuras 1.4 e 1.6 foram desenvolvidas a partir destas ferramentas.

1.5 Organização

Esta tese foi dividida em oito capítulos. Os demais capítulos seguem a estrutura abaixo:

- **Capítulo 2**: discute os conceitos que fazem parte do universo deste estudo, importantes para a compreensão do restante do documento. Mais especificamente, explora aspectos relacionados a: dados multidimensionais, classificação e detecção de agrupamentos, métricas e medidas de similaridade, redução de dimensionalidade, projeção de dados multidimensionais e cálculo de incerteza, além de algumas medidas de avaliação da qualidade dos resultados obtidos.

- **Capítulo 3**: apresenta uma revisão bibliográfica relacionada ao tema desta tese, com foco particular em técnicas de visualização envolvendo projeção de dados multidimensionais, esquemas de agrupamento de dados e emprego de diferentes medidas de similaridade, destacando as contribuições de cada técnica revisada.

  Os quatro capítulos seguintes constituem as principais contribuições deste projeto de doutorado. Correspondem às novas técnicas desenvolvidas e procuram destacar suas características, funcionamento, resultados, aplicações, vantagens e limitações.

- **Capítulo 4**: descreve a técnica de projeção local LAMP, uma técnica interativa, com sólida formulação matemática que se destaca pela qualidade dos mapeamentos produzidos, associados à sua eficiência computacional.

- **Capítulo 5**: apresenta a CSM, uma técnica de visualização de dados multidimensionais apoiada nos mapeamentos precisos produzidos pela LAMP. Orientada por um robusto mecanismo de seleção de amostras representativas, esta técnica permite identificar agrupamentos de dados com grande precisão.

- **Capítulo 6**: aborda a técnica de projeção denominada CSMP, a qual emprega uma "família de métricas" baseada em classes para comparar dados de alta dimensão. Com ênfase em dados extraídos a partir de coleções de imagens, CSMP é avaliada no contexto de recuperação de imagens com base em conteúdo.

- **Capítulo 7**: com o intuito de aumentar a eficácia da CSMP, este capítulo analisa outra técnica desenvolvida, denominada CSWIRe, a qual introduz informações de incerteza na família de métricas classes-específicas construída por sua antecessora.

- **Capítulo 8**: finaliza com algumas conclusões e trabalhos futuros.

Capítulo

2

Conceitos Fundamentais

Este capítulo apresenta os principais conceitos relacionados ao tema desta pesquisa, as ferramentas matemáticas utilizadas, e as medidas empregadas para avaliar a qualidade das técnicas desenvolvidas. O objetivo é, exclusivamente, dar suporte aos capítulos seguintes, sem a pretensão de esgotar todos os tópicos aqui apresentados, para isto a bibliografia referente aos assuntos pode ser consultada, muitas das quais são citadas no decorrer de cada seção.

2.1 Dados Multidimensionais

Dados multidimensionais (de alta dimensão, multivariados ou multivalorados) são aqueles que apresentam múltiplos atributos (variáveis ou características) (Mazza, 2009).

Em um conjunto de dados multidimensionais, os elementos são usualmente representados por vetores. Considere, por exemplo, o conjunto de dados X, contendo n elementos. O primeiro elemento de X será indicado por $x_1 = (x_{11}, x_{12}, \ldots, x_{1m})$, o segundo elemento por $x_2 = (x_{21}, x_{22}, \ldots, x_{2m})$, e assim por diante, onde o número de atributos m representa a dimensão do conjunto de dados X, também indicado como $X_{n \times m}$. Os m atributos constituem o *espaço de atributos* ou *espaço de características* do conjunto de dados. Neste contexto, cada elemento x_i será indicado como um objeto, uma instância, um vetor de atributos (ou de características) ou simplesmente um ponto do espaço m-dimensional.

Alguns conjuntos de dados apresentam um atributo C chamado atributo de *classes*, usado para rotular os dados. Este atributo assume uma quantidade discreta de valores, isto é, $C = \{c_1, c_2, \ldots, c_k\}$, onde k é o número de classes, com $k \geqslant 2$; e os valores $c_i,\ i = 1, \ldots, k$ são os rótulos de classe.

Definição 2.1 (Conjunto de Dados Rotulado (Liu, 2011)) *Um conjunto de dados que apresenta um ou mais atributos para identificação de classes diz-se rotulado, caso contrário diz-se não rotulado.*

Nesta tese, os conjuntos de dados utilizados – quando rotulados – apresentam um único atributo para identificação de classes, ou seja, cada instância está associada a um único rótulo.

Quando o conjunto de dados é rotulado, o termo balanceado é usado para indicar a frequência das classes, conforme segue.

Definição 2.2 (Conjunto de Dados Balanceado (Larose, 2006)) *Um conjunto de dados diz-se balanceado quando o número de instâncias em cada classe é aproximadamente o mesmo, caso contrário, diz-se desbalanceado ou não balanceado.*

Muitas vezes um conjunto de dados pode conter valores discrepantes ou atípicos, conhecidos como *outliers*. Tais ocorrências podem ser causadas por muitas razões, tais como erros humanos ou erros técnicos, ou podem ocorrer naturalmente em um conjunto de dados devido a eventos extremos (Olson e Delen, 2008).

Definição 2.3 (*Outliers* (Izenman, 2008)) *Outliers, dados atípicos ou dados discrepantes são valores nos dados que, por uma razão ou outra, parecem não obedecer um padrão comum à maioria dos valores; visualmente, eles estão localizados longe do restante dos dados, indicando sempre a presença de uma anomalia ou evento extremo.*

Para estimar a variação de um conjunto de dados, os coeficientes mais usados são a variância e desvio-padrão da população. Se uma população consiste de um conjunto de valores $x_1, x_2, \ldots, x_n$, então a variância da população, usualmente denotada por σ^2, é definida como:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}, \tag{2.1}$$

onde μ é a média da população.

Extraindo-se a raiz quadrada a partir da Equação (2.1), obtém-se o desvio-padrão, indicado por σ. O desvio-padrão tem uma vantagem sobre a variância porque ele é expresso na mesma unidade de medida do dado original, enquanto que a variância é expressa em unidades de medida ao quadrado, mesmo assim ele ainda é afetado por valores extremos (Clark-Carter, 2005). Outra medida estatística útil, baseada no desvio-padrão, é o coeficiente de variação de uma distribuição.

Definição 2.4 (Coeficiente de Variação (Borg e Groenen, 2005)) *Coeficiente de variação (CV), também conhecido como variabilidade relativa é a medida que indica a dispersão com respeito à média, definido como:*

$$CV = \frac{\sigma}{\mu},$$

onde σ é o desvio-padrão e μ é a média da distribuição.

O coeficiente de variação é, essencialmente, uma comparação relativa do desvio-padrão com a média. É particularmente útil ao comparar valores de desvio-padrão calculados a partir de diferentes médias (Black, 2009). Além disso, o CV apresenta algumas características favoráveis que o tornam ideal para uso neste trabalho: ele é independente de unidade de medida e da escala dos dados (Lovie, 2005).

2.2 Classificação e Detecção de Agrupamentos

A comunidade que estuda problemas relacionados à *aprendizagem de máquina* divide os problemas de aprendizagem em várias categorias. Em análise de dados, duas categorias são particularmente importantes: os métodos baseados em *aprendizagem supervisionada* e os baseados em *aprendizagem não supervisionada*. Classificação, normalmente, é uma tarefa supervisionada, enquanto que detecção de agrupamentos é, em geral, não supervisionada (Han et al., 2011; Maimon e Rokach, 2010). Tarefas de classificação e detecção de agrupamentos fazem parte de algumas técnicas apresentadas nesta tese, por este motivo, os conceitos serão apresentados abaixo.

Métodos supervisionados tentam descobrir um relacionamento entre valores conhecidos (por exemplo, um conjunto de dados com rótulos de classe pré-definidos) e valores desconhecidos (instâncias de dados onde os rótulos de classe não são conhecidos). Os dados contendo os valores conhecidos são chamadas de *dados de treinamento*, e o relacionamento procurado é chamado de *modelo*. Usualmente os modelos descrevem ou explicam fenômenos, os quais estão ocultos no conjunto de dados e, principalmente, são utilizados para predizer valores futuros nos dados (por exemplo, rotular futuras instâncias de dados fornecidas, com base nas classes existentes).

O processo supervisionado descrito acima caracteriza um processo de *classificação* e o modelo constitui um *modelo de classificação* ou *classificador*. Depois que o modelo é treinado ou construído a partir do conjunto de treinamento, ele deve ser avaliado usando um *conjunto de testes* para verificar sua acurácia.

Definição 2.5 (Classificação (Han et al., 2011)) *Classificação é o processo de encontrar um modelo (ou função) que descreve e distingue classes de objetos. Os modelos são baseados na análise de um conjunto de treinamento, onde os rótulos de classe são conhecidos. O modelo é então, usado para predizer o rótulo de classe de novos objetos.*

Definição 2.6 (Classificador) *O modelo (ou função) empregado na classificação chama-se modelo de predição, modelo de classificação ou simplesmente um classificador.*

Em métodos não supervisionados, os dados usados para aprendizagem não possuem valores conhecidos. Neste caso, o próprio algoritmo de aprendizagem precisa encontrar estruturas ocultas ou regularidades nos dados. Caso típico de aprendizagem não supervisionada são os algoritmos de detecção de agrupamentos, os quais organizam instâncias em grupos segundo suas similaridades (ou diferenças).

Definição 2.7 (Detecção de Agrupamentos (Tan et al., 2005)) *É um método de aprendizagem não supervisionado que tenta encontrar grupos de objetos, tal que os objetos em um mesmo grupo sejam similares (ou relacionados) entre si e, diferentes de (ou não relacionados a) objetos de outros grupos.*

A Figura 2.1 ilustra a ideia de agrupamento de dados do ponto de vista geométrico, enfatizando a proximidade entre objetos do mesmo grupo, bem como o afastamento entre objetos pertencentes a diferentes grupos. Note que as instâncias são organizadas de modo a representar a população que está sendo amostrada, ou seja, se S é um conjunto de amostras organizado em k grupos, $C_1, ..., C_k$, então:

$$S = \bigcup_{i=1}^{k} C_i \quad \text{e} \quad C_i \cap C_j = \emptyset, \text{ para } \; i \neq j.$$

Como consequência, qualquer instância de S pertence a um e somente um grupo C_i.

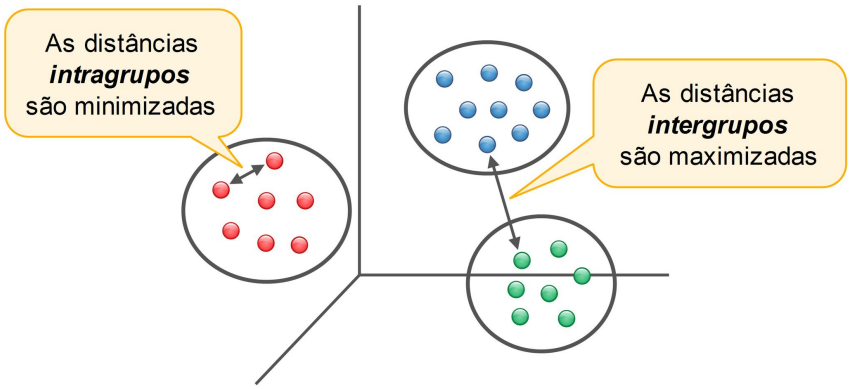


Figura 2.1: Agrupamentos de dados, mostrando as distâncias intra e intergrupos (Modificado de Tan et al. (2005)).

Além dos modelos de aprendizagem supervisionado e não supervisionado apresentados nesta seção, existe também o modelo que se apoia em um pequeno número de instâncias rotuladas e em um grande número de instâncias não rotuladas. Este modelo é conhecido como *aprendizagem semissupervisionada* (Liu, 2011).

2.2.1 Qualidade dos Agrupamentos

Uma das medidas mais usadas para avaliar a qualidade dos agrupamentos obtidos é a medida da silhueta. A silhueta combina duas medidas conhecidas: coesão e separação.

Definição 2.8 (Medida da Silhueta (Rousseeuw, 1987)) *Seja i um objeto contido no grupo A de um conjunto de dados, o valor da silhueta s_i pode ser calculado do seguinte modo:*

1. *Quando o grupo A contém outros objetos além do objeto i, calcula-se a_i = dissimilaridade média de i a todos os outros objetos de A (coesão);*
2. *Para todo grupo C diferente de A, calcula-se $d(i, C)$ = dissimilaridade média de i a todos os objetos de C (separação), obtendo-se $b_i = \min_{A \neq C} d(i, C)$;*
3. *O número s_i é obtido combinando-se a_i e b_i, conforme segue:*

$$s_i = \frac{b_i - a_i}{max(a_i, b_i)}.$$

A Figura 2.2 ilustra o cálculo da medida da silhueta para o objeto $i \in A$. Quando o grupo A contém somente um objeto, o cálculo de a_i é incerto, logo, s_i deve ser considerado igual a zero. Nesta tese, os resultados da silhueta apresentados, indicados por `Silh`, correspondem à média das silhuetas para todos os objetos i do conjunto de dados, ou seja:

$$\texttt{Silh} = \frac{1}{n} \sum_{i=1}^{n} s_i.$$

Esta medida é conhecida como *largura média de silhueta* (Rousseeuw, 1987).

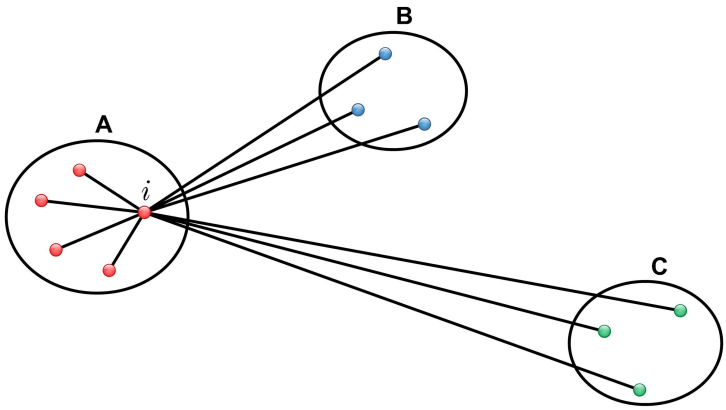


Figura 2.2: Elementos envolvidos no cálculo da silhueta para o objeto i pertencente ao grupo A (Retirado de Rousseeuw (1987)).

O valor da silhueta pode variar entre $[-1, 1]$. Valores negativos são indesejados porque correspondem ao caso em que a_i, a média das dissimilaridades para objetos no grupo, é

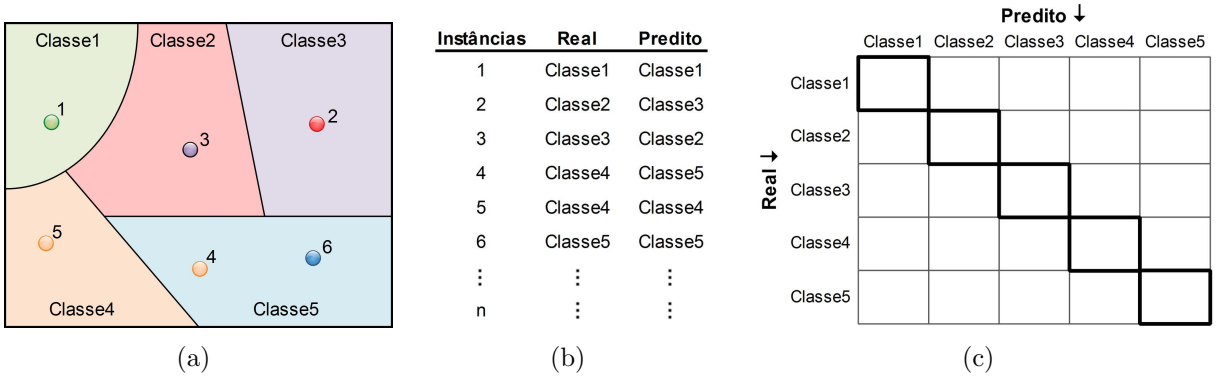
maior do que b_i, o mínimo das dissimilaridades médias para objetos em outros grupos. Em outras palavras, quanto maior o valor da silhueta, melhor será a coesão e a separação, ou seja, instâncias pertencentes ao mesmo grupo estarão mais próximas umas das outras, e ainda, grupos distintos estarão mais afastados. É um coeficiente sensível a pequenas variações, isto significa que um pequeno aumento em seu valor (mesmo na 3^a ou 4^a casa decimal) pode implicar em agrupamentos muito mais separados e coesos. Para mais informações sobre a medida da silhueta veja Rousseeuw (1987) e Tan et al. (2005).

Outra medida de avaliação empregada neste trabalho é a matriz de confusão, comumente utilizada para verificar a acurácia em tarefas de classificação de dados.

Definição 2.9 (Matriz de Confusão (Camps-Valls e Bruzzone, 2009))
Matriz de confusão é uma simples tabulação que cruza os rótulos das classes reais e preditas, observados para as instâncias contidas em um conjunto de testes, tal que a diagonal principal da matriz indica o número de instâncias alocadas corretamente em cada classe, enquanto que as demais posições indicam o número de instâncias alocadas incorretamente.

A Figura 2.3 ilustra o processo de cálculo da matriz de confusão a partir de instâncias previamente classificadas.

Uma matriz de confusão apropriadamente construída pode fornecer um simples sumário da acurácia da classificação e destacar dois tipos de erros de classificação que podem ocorrer: omissão (instâncias de uma classe incorretamente alocadas em outra classe, portanto, omitidas da classe de interesse) e concessão (instâncias de outra classe incorretamente alocadas na classe de interesse, portanto, concedidas para a classe de interesse) (Camps-Valls e Bruzzone, 2009).



(a) (b) (c)

Figura 2.3: Processo de cálculo da matriz de confusão: (a) Visualização de grupos e classificação das instâncias; (b) Rótulos das classes reais/preditas para as instâncias do conjunto de dados; (c) Tabulação cruzando os rótulos das classes reais e preditas para produzir uma matriz de confusão (Modificado de Camps-Valls e Bruzzone (2009)).

A matriz de confusão pode ser usada para derivar uma variedade de medidas que expressam a acurácia de uma classificação (Liu et al., 2007; Stehman, 1997; Trodd, 1995) como, por exemplo, a acurácia global da classificação, conforme definido abaixo.

Definição 2.10 (Acurácia da Classificação (Camps-Valls e Bruzzone, 2009)) *O coeficiente de acurácia (ACC) da classificação representa a proporção de instâncias que foram classificadas corretamente. Pode ser calculada pela soma das instâncias contidas na diagonal principal da matriz de confusão, dividida pelo número total de instâncias usadas para construir a matriz de confusão.*

A acurácia é algumas vezes multiplicada por cem para produzir a porcentagem de instâncias classificadas corretamente. A medida da acurácia não trabalha bem quando os conjuntos de dados são altamente desbalanceados (Han et al., 2011).

2.3 Medidas de Similaridade

A ideia de distância é natural e intuitiva, e desempenha papel fundamental na projeção e comparação de dados multidimensionais, portanto, os principais conceitos serão apresentados a seguir.

Definição 2.11 (Métrica (Lima, 1977)) *Uma métrica num conjunto M é uma função $d : M \times M \to \mathbb{R}$, que associa a cada par ordenado de elementos $x, y \in M$ um número real $d(x, y)$, chamado a distância de x a y, de modo que sejam satisfeitas as seguintes condições para quaisquer $x, y, z \in M$:*

p1) $d(x, y) \geqslant 0$;
p2) $d(x, y) = 0 \Leftrightarrow x = y$;
p3) $d(x, y) = d(y, x)$;
p4) $d(x, z) \leqslant d(x, y) + d(y, z)$.

Os postulados acima são conhecidos como *postulados de espaço métrico*. Os Postulados p1 e p2 afirmam que a distância é um valor sempre positivo, ou nulo quando $x = y$. O Postulado p3 afirma que a distância $d(x, y)$ é uma função simétrica das variáveis x, y. O Postulado p4 chama-se *desigualdade triangular* e tem origem no fato de que, no espaço Euclidiano de dimensão finita, o comprimento de um dos lados de um triângulo nunca excede a soma dos outros dois.

Definição 2.12 (Espaço Métrico (Lima, 1977)) *Um espaço métrico é um par (M, d), no qual $M \neq \emptyset$ é um conjunto e d é uma métrica em M.*

Cada elemento de um espaço métrico é referido como um ponto desse espaço, seja ele um ponto, um número, um vetor ou uma função, situações que se verificam comumente.

Também é comum, salvo quando houver possibilidade de dúvida, nos referirmos apenas ao "espaço métrico M", ficando subentendida a métrica que está sendo considerada, usualmente, a Euclidiana.

Quanto às métricas existentes, cada uma delas tem diferentes características e aplicações. Sua escolha deve levar em consideração a natureza dos dados envolvidos e o tipo de análise a ser realizada (Zezula et al., 2005). Um típico exemplo é a família de métricas de *Minkowski*, assim definida:

$$d(\mathrm{x},\, y) = \left| \left(\sum_{i=1}^{n} |x_i - y_i|^k \right)^{\frac{1}{k}} \right|, \quad k = 1, ..., \infty, \tag{2.2}$$

onde k é um parâmetro que modifica a métrica, gerando uma família de medidas de distância. Quando $k = 1$, por exemplo, obtém-se a *métrica Manhattan ou métrica City block*:

$$d(\mathrm{x},\, y) = \sum_{i=1}^{n} |x_i - y_i|. \tag{2.3}$$

Quando $k = 2$, ela se torna a *métrica Euclidiana* clássica, a qual é tipicamente usada para descrever a distância entre dois objetos no espaço Euclidiano:

$$d(\mathrm{x},\, y) = \left| \left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{\frac{1}{2}} \right|. \tag{2.4}$$

Quando $k = \infty$, obtém-se a *métrica do Máximo*:

$$d(\mathrm{x},\, y) = \max_{i=1}^{n} |x_i - y_i|. \tag{2.5}$$

Dependendo da quantidade de valores retornados pela função de distância, as métricas podem ser divididas em dois grupos (Zezula et al., 2005):

- *Discretas* - quando a função de distância retorna somente alguns valores pré-definidos. Um representante desta categoria é a métrica "zero-um", uma das mais simples existentes, definida como:

  $$d : M \times M \to \mathbb{R}, \text{ tal que } d(x,\, x) = 0 \text{ e } d(\mathrm{x},\, y) = 1 \text{ se } x \neq y.$$

- *Contínuas* - quando a cardinalidade do conjunto de valores retornado é muito alta ou infinita. Como é o caso da distância Euclidiana sobre os pontos de uma reta, por exemplo.

Esta tese não pretende explorar os vários tipos de métricas existentes. Para este fim, os trabalhos de Lima (1977), Zezula et al. (2005) e Zhang (2008) podem ser consultados.

No entanto, é importante destacar que, dependendo da natureza dos dados envolvidos, nem sempre uma métrica, satisfazendo os Postulados p1 a p4 da Definição 2.11, é a melhor forma de expressar a semelhança ou a diferença entre objetos do domínio em estudo.

Um exemplo típico é a *distância edit* (Levenshtein, 1965), usada para medir a proximidade entre cadeias de caracteres, a qual não satisfaz a propriedade da simetria (Postulado p3). Neste caso, a função de distância é conhecida como uma *quasi-métrica*, ou seja:

Definição 2.13 (Quasi-Métrica (Lima, 1977)) *Uma quasi-métrica num conjunto M é uma função real $d : M \times M \to \mathbb{R}$ que satisfaz as condições de uma métrica, salvo o fato de que pode ocorrer $d(x, y) \neq d(y, x)$.*

De modo similar, quando a função de distância não satisfaz o Postulado p2, a função é conhecida como uma *pseudo-métrica*:

Definição 2.14 (Pseudo-Métrica (Lima, 1977)) *Uma pseudo-métrica num conjunto M é uma função real $d : M \times M \to \mathbb{R}$ que satisfaz as condições de uma métrica, exceto que pode ocorrer $d(x, y) = 0$ com $x \neq y$.*

Neste último caso, M é conhecido como um *espaço pseudo-métrico*. Para mais detalhes sobre espaços métricos, veja Domingues (1982) e Lima (1977).

Com base na discussão anterior, surge naturalmente um novo conceito, mais abrangente que o de distância para comparar dois objetos, o de *similaridade*. Neste contexto, os objetos podem ser quaisquer tipos de dados: elementos de um espaço vetorial, elementos de um espaço métrico, dados relativos a texto, dados de imagem, ou quaisquer dados abstratos, tais como:

(a) O número de sintomas compartilhados por dois pacientes.
(b) O grau de parentesco entre duas pessoas.
(c) O custo de transporte de mercadorias entre duas cidades.
(d) A frequência relativa de palavras compartilhadas por dois documentos.

Definição 2.15 (Medida de Similaridade (Zhang, 2008)) *É um valor numérico usado para indicar quão semelhantes ou similares são dois objetos, segundo algum critério, normalmente uma função ou classe de comparação. Assim, quanto maior for a semelhança entre eles, maior será seu grau de similaridade.*

Em contraste à medida de similaridade define-se a *medida de dissimilaridade*, usada para indicar quão diferentes são os objetos do domínio em estudo. É comum referir-se à medida de dissimilaridade simplesmente como distância, ficando subentendido que um ou mais postulados da Definição 2.11 podem não ser satisfeitos.

Outra definição relevante neste estudo é a busca por similaridade, apresentada a seguir.

Definição 2.16 (Busca por Similaridade (Zezula et al., 2005)) *Busca por similaridade corresponde ao processo de obtenção de objetos de dados ordenados pela distância ou dissimilaridade de um dado objeto de consulta. É um tipo de ordenação de objetos com respeito ao objeto de consulta, onde o critério de ordenação é a medida da distância.*

Vale lembrar ainda que, determinadas técnicas requerem como entrada as dissimilaridades entre os n objetos do conjunto de dados. Quando isto ocorre, os valores calculados são comumente armazenados em uma estrutura matricial, denominada matriz de dissimilaridades.

Definição 2.17 (Matriz de Dissimilaridades (Han et al., 2011)) *Nome dado à estrutura que armazena a coleção de dissimilaridades avaliada para todos os pares de n objetos de um conjunto de dados. É frequentemente representada por uma matriz $n \times n$.*

2.4 Redução de Dimensionalidade

Redução de dimensionalidade é um recurso utilizado para representação compacta de grandes conjuntos de dados e redução do tempo de processamento computacional. Além disso, reduzir a dimensão de um conjunto de dados, permite observar e agrupar características importantes, dando uma visão da natureza dos dados muitas vezes não percebida na alta dimensão.

Existem, essencialmente, duas formas de se reduzir a dimensionalidade de um conjunto de dados (Webb, 2002). A primeira forma é identificar as variáveis que não contribuem com a tarefa de análise dos dados. Estas variáveis podem então ser negligenciadas, fazendo com que o espaço final fique automaticamente reduzido. Em determinadas circunstâncias, selecionar um subconjunto a partir de um grande número de variáveis ou características já é suficiente para a análise dos dados. Esta abordagem é conhecida como *seleção de características (feature selection)*.

A segunda abordagem é aquela que transforma o espaço inicial m-dimensional em outro espaço de menor-dimensão[1] p, empregando uma classe (ou função) de transformação f. Esta abordagem corresponde à *transformação do espaço de características*[2]. Esta transformação pode ser uma combinação linear ou não linear das variáveis originais, sendo indicada apenas como *redução de dimensionalidade*, e a classe de transformação como uma *técnica de redução de dimensionalidade*.

Em alguns casos, as duas abordagens podem ser combinadas, como é o caso da CSMP (ver Capítulo 6). Um diagrama comparativo entre as duas abordagens utilizadas para reduzir a dimensionalidade de um conjunto de dados pode ser observado na Figura 2.4.

[1] Muitas vezes denotado como espaço reduzido/transformado, espaço de destino ou espaço-alvo.
[2] Alguns autores preferem o termo *extração de características*.

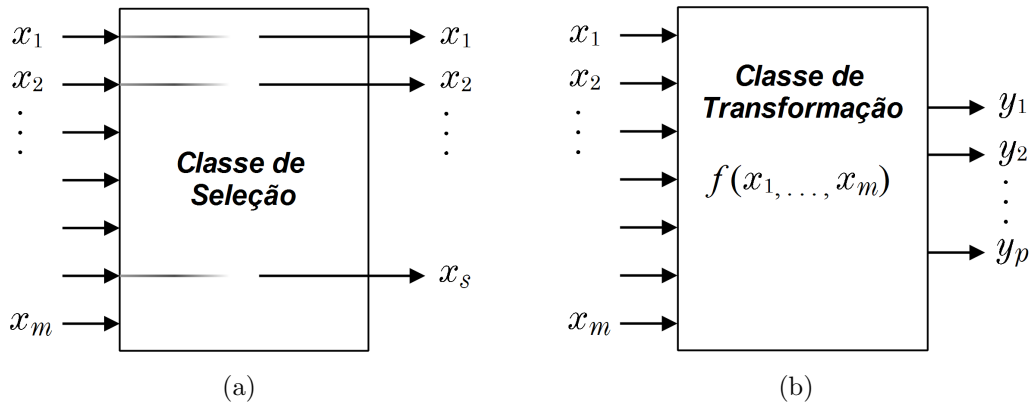


Figura 2.4: Redução de dimensionalidade por: (a) Seleção de características; (b) Transformação do espaço de características (Modificado de Webb (2002)).

Embora não seja estritamente necessário, para efeito das próximas definições, considere os conjuntos de dados X e Y contidos em um espaço real de dimensão finita.

Com base em Kirby (2001), é possível definir redução de dimensionalidade, formalmente, como:

Definição 2.18 (Redução de Dimensionalidade) *Se $X \subset \mathbb{R}^m$ e $Y \subset \mathbb{R}^p$, com $p < m$, então $f : X \mapsto Y$ é um mapeamento de redução de dimensionalidade do espaço de alta dimensão X para o espaço de menor dimensão Y, tal que Y retém a informação essencial dos pontos.*

Segundo Kirby (2001), reter a informação dos dados é altamente dependente do problema, ou seja, a aplicação deveria ditar a natureza matemática do mapeamento. Mas, via de regra procura-se preservar relações de dissimilaridade entre os pontos nos dois espaços.

De modo análogo, um *mapeamento de reconstrução de dimensão* g, é definido como:

$$g : Y \mapsto X. \tag{2.6}$$

Em um mapeamento ideal vale a relação:

$$h : X \overset{f}{\mapsto} Y \overset{g}{\mapsto} X. \tag{2.7}$$

Ou seja, a composta de g e f é uma função identidade:

$$h(x) = (g \circ f)(x) = g(f(x)) = g(y) = x,$$

$\forall\ x \in X$ e $y \in Y$, mas em geral isto não ocorre, tendo em vista que durante a redução de dimensionalidade costumam ocorrer perdas ou truncamento de informações.

Um mapeamento de redução de dimensionalidade pode ser classificado como *linear* ou *não linear*, de acordo com o tipo de transformação.

Definição 2.19 (Redução de Dimensionalidade Linear (Kirby, 2001)) *Um mapeamento de redução de dimensionalidade $f : X \rightarrow Y$ diz-se linear se $f(\alpha x_i + \beta x_j) = \alpha f(x_i) + \beta f(x_j)$, $\forall\ x_i, x_j \in X$ e $\alpha, \beta \in \mathbb{R}$.*

Quando a condição acima não se verifica, o mapeamento diz-se *não linear*.

A grande maioria dos fenômenos físicos são modelados por meio de funções não lineares, porém, dependendo da natureza dos dados envolvidos, técnicas lineares podem apresentar excelentes resultados (Maaten et al., 2009).

2.5 Projeção de Dados Multidimensionais

Projeção ou mapeamento de dados multidimensionais corresponde ao mapeamento de redução de dimensionalidade onde o espaço-alvo tem dimensão p igual a 1, 2 ou 3 (usualmente 2), condição conveniente para a visualização dos dados. Neste caso, o novo espaço é indicado como *espaço visual* ou *espaço de projeção*.

Admitindo-se a condição $p \in \{1, 2, 3\}$, é possível definir projeção de dados multidimensionais a partir de um mapeamento de redução de dimensionalidade, conforme segue:

Definição 2.20 (Projeção de Dados Multidimensionais) *Sejam os pares (X, d) e (Y, d^*) dois conjuntos de dados munidos de uma medida de dissimilaridade, assim definidos: $X \subset \mathbb{R}^m$, $d : X \times X \rightarrow \mathbb{R}$; e $Y \subset \mathbb{R}^p$, $d^* : Y \times Y \rightarrow \mathbb{R}$; tal que $p \in \{1, 2, 3\}$ e $p < m$. Uma técnica de projeção de dados multidimensionais equivale ao mapeamento de redução de dimensionalidade $f : X \rightarrow Y$ que procura tornar a diferença $|d(x_i, x_j) - d^*(f(x_i), f(x_j))|$ tão próxima de zero quanto possível, $\forall x_i, x_j \in X$.*

Neste contexto, $p < m$ é condição necessária para que haja redução de dimensionalidade, de outra forma poderia ocorrer apenas uma transformação de um espaço em outro. Além disso, para caracterizar uma técnica de projeção, é preciso garantir que $p \in \{1, 2, 3\}$ e preservar as relações de dissimilaridade entre os pontos nos dois espaços, tanto quanto possível. Definição semelhante é apresentada por Tejada et al. (2003).

2.5.1 Classificação das Técnicas de Projeção

As técnicas de projeção podem receber várias classificações. Dentre elas, as mais conhecidas são:

I) Quanto ao tipo de transformação que sofre:

- *Lineares*
- *Não lineares*

} (ver Definição 2.19)

- *Híbridas*: uma parte é linear e outra não linear.

II) Quanto à natureza da projeção:

- *Locais*: tentam preservar a geometria local dos dados; essencialmente, os pontos próximos a uma dada instância na alta dimensão são mapeados próximos à sua representação de menor dimensão.
- *Globais*: tentam preservar a geometria em todas as escalas, isto significa que o mapeamento de cada instância deve considerar globalmente as demais, de tal modo que pontos próximos na alta dimensão devem ficar próximos na projeção, assim como pontos distantes também devem ficar distantes.

Técnicas locais/globais são discutidas em De Silva e Tenenbaum (2003) e Joia et al. (2011).

III) Quanto à interatividade:

- *Interativas*: permitem a intervenção do usuário, normalmente, de modo a agregar seu conhecimento ao processo.
- *Não interativas*: não admitem intervenção do usuário no processo.

IV) Quanto à formulação matemática:

- *Técnicas baseadas em decomposição espectral*: calculam as coordenadas de cada instância a partir dos autovetores de uma transformação aplicada em uma matriz (Torgerson, 1965).
- *Técnicas baseadas em otimização não linear*: inicialmente proposto por Kruskal (1964), compreendem uma categoria de técnicas que executam o mapeamento para o espaço visual minimizando uma função de energia, normalmente chamada *função de stress*.
- *Técnicas baseadas em força*: surgiu com o trabalho de Eades (1984). Mapeiam dados para o espaço visual por meio de um esquema baseado em força, inspirado em uma analogia entre minimização da *função de stress* e sistemas massa-mola, onde a força restauradora do sistema é dada pela diferença numérica entre as distâncias calculadas a partir dos espaços de origem e de projeção.
- *Outros modos*: aquelas que não se encaixam em nenhuma das anteriores (decomposição, otimização, força) ou suas variações híbridas.

Além das apresentadas acima, Maaten et al. (2009) propõem outras subdivisões, como *convexas* e *não convexas*, *full espectral* e *espectral esparsa*, *distância Euclidiana com pesos*, *alinhamento de modelos lineares locais* e as baseadas em *redes neurais*.

2.5.2 Qualidade da Projeção

De acordo com a Definição 2.20 uma projeção de dados procura transformar um espaço de alta dimensão m em um espaço de menor dimensão p igual a 1, 2 ou 3, preservando relações de dissimilaridade entre instâncias nos dois espaços. Como consequência da definição, é imediato atestar a qualidade da projeção medindo o quanto estas relações se preservam. Uma função que estima esse valor é conhecida como *função de stress*. Existem diferentes variações de função *stress*, neste trabalho adotou-se o *stress* definido por Kruskal (1964).

Definição 2.21 (Medida do Stress (Kruskal, 1964)) *O stress mede quão bem uma dada configuração representa os dados. Quanto menor o valor do stress, melhor a representação, tal que zero indica "perfeita" representação. Pode ser calculado pela seguinte função:*

$$stress = S = \frac{\sum_{ij}(d_{ij} - d^*_{ij})^2}{\sum_{ij} d^2_{ij}}\,, \tag{2.8}$$

onde d_{ij} e d^*_{ij} são duas sequências numéricas, tal que d^*_{ij} corresponde aos valores que minimizam S.

Kruskal (1964) define também o *stress* normalizado, análogo a escolher o desvio-padrão no lugar da variância, o qual pode ser calculado como:

$$stress = \sqrt{S} = \sqrt{\frac{\sum_{ij}(d_{ij} - d^*_{ij})^2}{\sum_{ij} d^2_{ij}}}\,. \tag{2.9}$$

No contexto de projeção, o *stress* estima a qualidade da projeção com base na preservação de dissimilaridades entre instâncias nos dois espaços, onde d_{ij} e d^*_{ij} são as medidas de dissimilaridade entre as instâncias i e j no espaço de origem e de projeção, respectivamente.

Além do *stress*, outra medida conhecida para avaliar a qualidade da projeção é a preservação de vizinhança.

Definição 2.22 (Preservação de Vizinhança (Paulovich e Minghim, 2008)) *Medida usada para avaliar a preservação das relações de vizinhança dos pontos nos dois espaços. Pode ser calculada do seguinte modo:*

1. *Fixa-se um número inteiro $k > 0$;*
2. *Tomam-se os k-vizinhos mais próximos de uma instância x_i no espaço multidimensional;*

3. *Obtêm-se os k-vizinhos mais próximos da sua projeção y_i no espaço visual;*

4. *Verifica-se a proporção de vizinhos que foi preservada neste último.*

Normalmente o procedimento acima é iterado para todas as instâncias x_i $(i = 1, ..., n)$ do conjunto de dados, para o mesmo k fixado, obtendo-se uma precisão média $p_k \in [0, 1]$. Fazendo o número de vizinhos k variar de 1 a um valor arbitrário, e representando os valores (k, p_k) como pontos no plano cartesiano, obtém-se uma curva de precisão, denominada *curva de preservação de vizinhança* (ver Figura 4.11 como exemplo). Esta curva é indicada quando a prioridade é atestar a preservação das relações de vizinhança dos pontos nos dois espaços, em oposição ao *stress* que atesta apenas as relações de dissimilaridade dos mesmos.

Além das medidas discutidas neste capítulo, outras medidas têm sido propostas para atestar a qualidade das projeções. Por exemplo, Sips et al. (2009) apresentam duas medidas quantitativas para verificar a consistência das classes em um mapeamento multidimensional, uma baseada na distância ao centro de gravidade das classes, e outra baseada na entropia da distribuição espacial das classes, as quais são robustas a *outliers*. Motta et al. (2015) propõem medidas baseadas em grafos para avaliar a qualidade das projeções. Bertini e colaboradores (Bertini et al., 2011) apresentam uma análise sistemática de medidas de qualidade usadas para dar suporte à exploração de conjuntos de dados multidimensionais, bem como os diferentes domínios em que podem ser aplicadas.

2.6 Modelagem de Incerteza Usando Conjuntos Fuzzy

Os conceitos apresentados nesta seção constituem a base do Capítulo 7, o qual emprega modelagem de incerteza para aumentar a acurácia de uma família de métricas usada na comparação de dados multidimensionais.

Incerteza desempenha um importante papel quando combinada com certas características do modelo matemático. Segundo Celikyilmaz e Türksen (2009), identificar incertezas reduz a complexidade e aumenta a precisão do sistema. O termo "*modelagem de incerteza*" define o esforço para identificar incertezas ao construir um modelo matemático. Parte do desafio da modelagem de incerteza é determinar o nível ótimo de permissividade da incerteza. Conjuntos *fuzzy* podem auxiliar nesta tarefa, já que permitem representar conceitos vagos, tal como mostrado na Figura 2.5: uma escala semântica com vários níveis de pertinência expressos em linguagem natural.

Em oposição à escala semântica da Figura 2.5, quando uma dada condição particiona um conjunto X em dois subconjuntos: membros (aqueles que satisfazem a condição) e não membros (aqueles que não satisfazem a condição), a coleção de objetos-membros resultante caracteriza um subconjunto clássico ou *crisp*. Note que a negação contém todos os elementos do conjunto X que não são membros (complemento). Neste contexto,

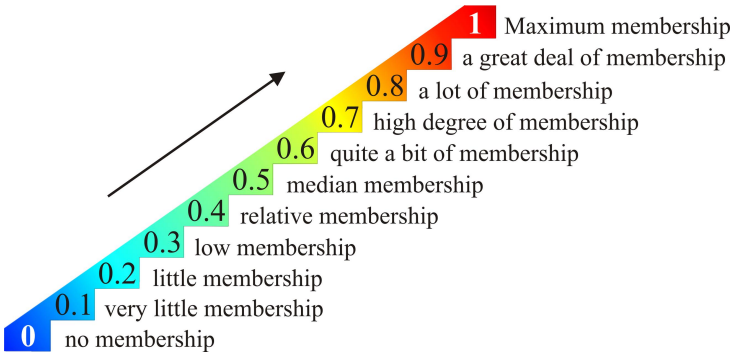


Figura 2.5: Escala semântica mostrando vários níveis de pertinência, os quais podem ser utilizados para representar a incerteza de um modelo matemático (Retirado de Gil-Aluja (2004)).

o conjunto X é comumente chamado de conjunto universal, e seus elementos constituem o "universo do discurso".

A principal diferença entre a teoria dos conjuntos clássica e a teoria dos conjuntos *fuzzy* está no fato de que esta última admite um grau de pertinência parcial aos seus elementos, representado por um valor no intervalo $[0, 1]$. Diferente da situação acima, onde os elementos podem assumir apenas o status de membro ou não membro. Partindo desta premissa, é possível definir um conjunto clássico ou *crisp* como um conjunto *fuzzy* que restringe o grau de pertinência de seus elementos aos valores $\{0, 1\}$ (Smithson e Verkuilen, 2006).

Conjuntos *fuzzy* podem ser representados explicitamente por seus elementos, porém, é comum se referir a um conjunto *fuzzy* por meio de sua função de pertinência $\mu_A(x)$, tal como:

$$\mu_A(x) : X \rightarrow [0, 1], \tag{2.10}$$

onde x representa um elemento do universo X, ou como um conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)) \mid x \in X\}. \tag{2.11}$$

Existem vários tipos de função de pertinência (Yager e Filev, 1994), as quais são normalizadas em forma de triângulos, trapézios, Gaussianas, entre outras formas. Neste trabalho foram utilizadas funções triangulares conforme ilustrado na Figura 2.6, por apresentarem bons resultados e melhor adequação ao modelo de incerteza proposto.

Funções triangulares apresentam o nível de pertinência máximo em um único ponto e valores que decrescem linearmente conforme sua taxa de variação. Estas funções costumam ser indicadas por uma tupla de valores (x_1, x_2, x_3) conhecida como *Fuzzy Triangular Number* (FTN) (Gil-Aluja, 2004), onde x_1 é o limite inferior, x_2 o valor de pertinência máximo e x_3 o limite superior.

É possível encontrar o grau de pertinência de um elemento a partir de uma função triangular *fuzzy*, encontrando seus intervalos de corte ou *cortes-α* como são chamados.

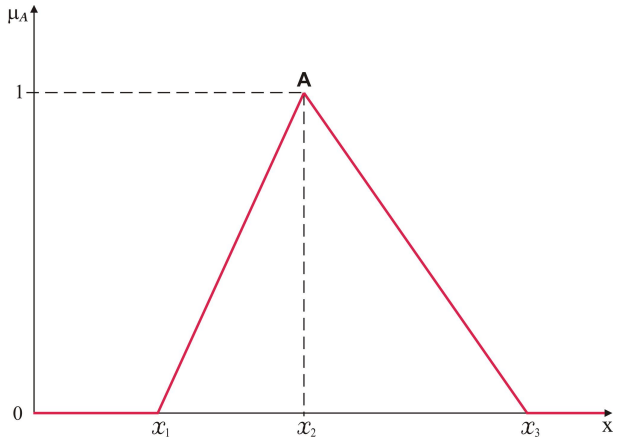


Figura 2.6: Exemplo de função *fuzzy* triangular.

Cortes-α assumem papel de destaque na teoria dos conjuntos *fuzzy* (Celikyilmaz e Türksen, 2009; Yager e Filev, 1994), bem como neste trabalho.

Definição 2.23 (Corte-Alfa) *Seja A um conjunto fuzzy definido no universo X e α um número real, $\alpha \in [0,1]$. Um corte-α de A, denotado por ${}^{\alpha}A$ é o subconjunto crisp de X, tal que*

$$ {}^{\alpha}A = \{x \mid \mu_A(x) \geqslant \alpha, x \in X\}. $$

A Figura 2.7 exibe alguns cortes-α: ${}^{0,0}A$, ${}^{0,1}A$, ${}^{0,2}A$, ${}^{0,5}A$, ${}^{0,9}A$. Note que a direção das retas **r** e **s** pode ser obtida a partir dos vetores diretores $\overrightarrow{P_1P_2} = (x_2 - x_1, 1)$ e $\overrightarrow{P_3P_2} = (x_2 - x_3, 1)$, respectivamente.

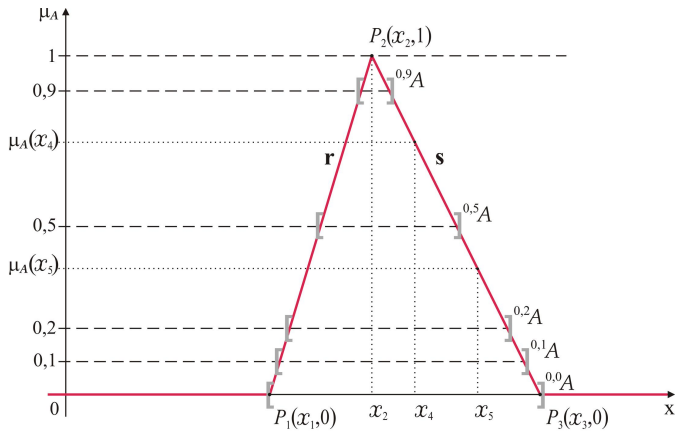


Figura 2.7: Exemplo de cortes-α assumindo que A é um subconjunto *fuzzy* do universo X.

Tomando-se as equações paramétricas das retas **r** e **s** em relação ao eixo-x e o parâmetro α, obtém-se:

$$ \begin{aligned} \mathbf{r}: \ P &= P_1 + \alpha.\overrightarrow{P_1P_2} \ \therefore \ x = x_1 + (x_2 - x_1)\alpha, \\ \mathbf{s}: \ P &= P_3 + \alpha.\overrightarrow{P_3P_2} \ \therefore \ x = x_3 + (x_2 - x_3)\alpha. \end{aligned} \tag{2.12} $$

Portanto, cada intervalo ${}^{\alpha}A$, com $\alpha \in [0,1]$, varia entre $x_1 + (x_2 - x_1)\alpha$ e $x_3 + (x_2 - x_3)\alpha$, ou simplesmente:

$$ {}^{\alpha}A = \left[{}^{\alpha-}A,\ {}^{\alpha+}A\right] = \left[x_1 + (x_2 - x_1)\alpha,\ x_3 - (x_3 - x_2)\alpha\right], \tag{2.13} $$

onde os símbolos ${}^{\alpha-}A$ e ${}^{\alpha+}A$ são usados para indicar os limites inferior e superior do intervalo de corte ${}^{\alpha}A$, respectivamente.

A Equação (2.13) é de grande importância neste estudo, pois fornece um mecanismo para obter cortes-α a partir de um FTN e um dado nível-α.

Se tomarmos o corte-α ${}^{0,5}A$ a partir da Figura 2.7, por exemplo, é direto ver que $\mu_A(x_4) \geqslant 0,5 \therefore x_4 \in {}^{0,5}A$. Observe também que $x_5 \in {}^{0,1}A$ e $x_5 \in {}^{0,2}A$, mas $x_5 \notin {}^{0,5}A$. Considerando apenas os níveis-α mostrados na Figura 2.7, isto é: $\{0;\ 0,1;\ 0,2;\ 0,5;\ 0,9;\ 1\}$, então pode-se concluir que $x_5 \in A$ com grau de pertinência $0,2$ (maior nível-α).

Em aplicações práticas, é usual empregar duas ou mais funções *fuzzy* para representar um modelo de incerteza. Nestas circunstâncias ambiguidades podem ocorrer, como ilustrado na Figura 2.8. Operações com conjuntos *fuzzy* aplicam-se nestes casos, conforme segue.

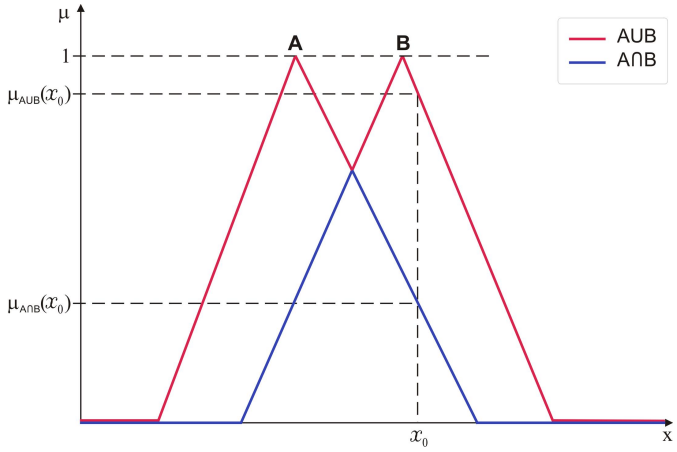


Figura 2.8: União e interseção de conjuntos *fuzzy*.

Definição 2.24 (União de Conjuntos *Fuzzy*) *Sejam A e B dois conjuntos fuzzy definidos no universo X. A união é definida pelo máximo da função de pertinência, ou seja:*

$$ A \cup B : \mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)]. $$

Definição 2.25 (Interseção de Conjuntos *Fuzzy*) *Sejam A e B dois conjuntos fuzzy definidos no universo X. A interseção é definida pelo mínimo da função de pertinência, ou seja:*

$$ A \cap B : \mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)]. $$

Definição 2.26 (Complemento de um Conjunto *Fuzzy*) *O complemento de um conjunto fuzzy A no universo X, indicado por $\bar{A}$, é definido como:*

$$\bar{A} : \mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

O exemplo apresentado na Figura 2.8 exibe dois valores de pertinência possíveis para o elemento x_0. Tal ambiguidade pode ser resolvida de acordo com o tipo de operação empregada: *união* ou *interseção* de conjuntos *fuzzy*. A união torna o modelo mais permissivo (maior valor de pertinência), enquanto que a interseção torna o modelo mais restritivo (menor valor de pertinência).

As operações apresentadas nesta seção são suficientes para a compreensão do restante do documento. Para outras operações com conjuntos *fuzzy* consulte Celikyilmaz e Türksen (2009) e Yager e Filev (1994).

2.7 Considerações Finais

Os conceitos apresentados neste capítulo são importantes para a compreensão do restante do documento e podem ser revisitados sempre que houver necessidade.

A Seção 2.6 – Modelagem de Incerteza Usando Conjuntos *Fuzzy*, particularmente, apresenta as ferramentas e conceitos necessários para a compreensão do Capítulo 7, o qual utiliza conjuntos *fuzzy* para modelar a incerteza inerente ao processo de comparação de dados multidimensionais. Vale lembrar que a teoria dos conjuntos *fuzzy* pode ser aplicada nos mais diferentes domínios, entre eles: visualização de informação (Ge et al., 2009), redução de dimensionalidade (Zhao et al., 2012), identificação de agrupamentos (Miyamoto et al., 2008) e medidas de similaridade (Papakostas et al., 2013), os quais fazem parte do universo desta pesquisa.

Capítulo

3

Trabalhos Relacionados

O presente capítulo apresenta uma revisão bibliográfica das técnicas relacionadas ao tema desta pesquisa. Para facilitar o entendimento, elas foram organizadas em três categorias: técnicas de projeção de dados multidimensionais (Seção 3.1), técnicas para identificação e visualização de agrupamentos (Seção 3.2) e técnicas que usam diferentes medidas de similaridade (Seção 3.3).

Para fixar notação, alguns símbolos foram previamente convencionados neste capítulo:

n	Número de instâncias no espaço de origem / destino.
m	Dimensão do espaço de origem.
p	Dimensão do espaço de destino, normalmente $p < m$.
X	Conjunto de dados no espaço de origem, normalmente uma matriz $n \times m$.
Y	Conjunto de dados no espaço de destino, normalmente uma matriz $n \times p$.
x_i	A i-ésima instância do espaço de origem, cuja representação vetorial é dada por $\mathrm{x}_i = (x_{i1}, x_{i2}, \ldots, x_{im})$.
y_i	A i-ésima instância do espaço de destino, cuja representação vetorial é dada por $\mathrm{y}_i = (y_{i1}, y_{i2}, \ldots, y_{ip})$.
k	Usado normalmente para indicar o número de vizinhos de uma dada instância.
s	Usado normalmente para indicar o número de amostras representativas de um conjunto de dados.
d_{ij}	Dissimilaridade entre as instâncias i e j no espaço de origem.
d^*_{ij}	Dissimilaridade entre as instâncias i e j no espaço de destino.
D	Matriz-$(n \times n)$ de dissimilaridades obtida a partir do espaço de origem.
D^*	Matriz-$(n \times n)$ de dissimilaridades obtida a partir do espaço de destino.

3.1 Técnicas de Projeção de Dados Multidimensionais

Técnicas de projeção podem ser classificadas de diferentes formas, conforme apresentado na Seção 2.5.1, e na maioria das vezes admitem mais de uma classificação, assim, uma técnica pode ser ao mesmo tempo *não linear*, *global* e *baseada em força*, por exemplo. Portanto, neste estudo, as técnicas não foram agrupadas em uma categoria específica, e sim exploradas segundo suas principais características, ou seja:

- Ordem de complexidade do algoritmo.
- Uso ou não de amostras representativas.
- Interatividade da técnica.
- Se a técnica é "incremental", no sentido de que novas instâncias podem ser mapeadas isoladamente, sem remapear ou recalcular as demais.
- Se requer dados de entrada como vetores em $\mathbb{R}^m$, isto é, $X \subset \mathbb{R}^m$.
- Tipo de transformação, podendo ser: linear, não linear ou híbrida.
- Formulação matemática: decomposição espectral, otimização não linear, força, outros tipos.
- Natureza da projeção: local, global ou ambas.

As próximas subseções revisam as principais técnicas de projeção disponíveis na literatura, escolhidas com base na qualidade dos resultados que apresentam, bem como pelas suas características, algumas das quais representam o estado da arte no assunto. Assim sendo, as seguintes técnicas de projeção são apresentadas neste capítulo:

- A **LLE** (Roweis e Saul, 2000), por preservar propriedades locais e por ser a base de várias outras técnicas, tal como a ONPP. A **ONPP** (Kokiopoulou e Saad, 2007) tem sua formulação matemática baseada na minimização de uma determinada função-objetivo[1], com restrições de ortogonalidade; estratégia utilizada em uma das abordagens propostas, daí a sua importância.
- ***FastMap*** (Faloutsos e Lin, 1995), por ser uma das técnicas mais rápidas conhecidas.
- **SM** (Sammon, 1969), por ser uma das mais tradicionais baseada em otimização não linear, além de ser a base de *Pekalska* (uma técnica muito precisa).
- **MDS** (Torgerson, 1952) e **Isomap** (Tenenbaum et al., 2000), necessárias para a compreensão das técnicas LMDS e L-Isomap, respectivamente.

[1] Função a ser minimizada, também denotada como função-custo ou função-erro.

- ***Pekalska*** (Pekalska et al., 1999), **LMDS** (De Silva e Tenenbaum, 2004), **L-Isomap** (De Silva e Tenenbaum, 2003), **PLMP** (Paulovich et al., 2010b) e a **PLP** (Paulovich et al., 2011) por apresentarem bom desempenho em termos de *stress* e/ou tempo computacional; por se apoiarem em subconjunto de amostras para realizar a projeção; e por admitirem interatividade (PLMP e PLP).

- **LSP** (Paulovich et al., 2008), **SNE** (Hinton e Roweis, 2002) e **t-SNE** (Maaten e Hinton, 2008) por preservarem muito bem relações de vizinhança durante a projeção. No caso da t-SNE, por representar uma das técnicas do estado da arte atual.

- **LoCH** (Fadel et al., 2015) por ser uma das mais atuais e por trabalhar de modo diferenciado, posicionando os pontos com base no fecho convexo dos vizinhos mais próximos.

3.1.1 Natureza Local x Global

A natureza da projeção é um aspecto particularmente importante nesta tese, uma vez que a LAMP, uma das técnicas desenvolvidas neste projeto de doutorado (Capítulo 4), pode se comportar tanto como uma técnica global como local.

Técnicas locais tentam preservar a geometria local dos dados, ou seja, o mapeamento de cada instância depende somente das amostras em sua vizinhança, o que caracteriza a natureza local do processo. As técnicas globais, por outro lado, tentam preservar a geometria em todas as escalas, isto significa que o mapeamento de cada instância deve considerar globalmente as demais (De Silva e Tenenbaum, 2003).

A interpretação da natureza das técnicas, no entanto, pode produzir resultados contraditórios. A Tabela 3.1 apresenta alguns casos onde a mesma técnica é classificada ora como local, ora como global, em diferentes trabalhos. Isto se deve em parte, porque algumas técnicas, assim como a LAMP, podem apresentar os dois comportamentos, mas também pelo fato desta interpretação ser algumas vezes subjetiva.

Considere, por exemplo, o caso clássico da SNE (ver Seção 3.1.12). Embora Maaten (2007) (um dos autores da t-SNE) classifique a técnica como global, seu trabalho também afirma que:

> [...] Na SNE, similaridades de pontos próximos contribuem mais para a função-custo. Isto conduz a um espaço de baixa dimensão que preserva principalmente propriedades locais da variedade. Seu comportamento local depende fortemente de um parâmetro livre definido pelo usuário.

No ano seguinte, o mesmo autor a classifica como local (Maaten e Hinton, 2008). Lee e colaboradores (Lee et al., 2015) também mencionam aspectos locais da SNE ao afirmarem que:

> [...] Todavia, estas melhorias não conseguem resolver uma lacuna frequentemente negligenciada nos métodos baseados na SNE, isto é, o tamanho da vizinhança considerada é relativamente pequeno, comparado ao número de instâncias avaliadas. Como consequência, informações sobre a estrutura global dos dados são perdidas, tornando a SNE e suas variações propensas a problemas de mínimos locais.

Casos como este, em que apenas uma pequena vizinhança dos pontos é considerada para computar o mapeamento, costumam caracterizar sua natureza local.

No caso da LLE (Roweis e Saul, 2000), o número de vizinhos também é controlado por um parâmetro livre definido pelo usuário, mas nem sempre aumentar ou diminuir o

Tabela 3.1: Técnicas de projeção que apresentam classificação contraditória na literatura, em relação à natureza da projeção.

Técnica	**Classificação das técnicas de acordo com diferentes trabalhos:**	
	Local	**Global**
LLE	De Silva e Tenenbaum (2003, 2004) Maaten (2007) Maaten e Hinton (2008) Izenman (2008) Maimon e Rokach (2010) Paulovich et al. (2011) Fadel et al. (2015)	Hoffmann et al. (2009)
LMDS	Paulovich et al. (2011)	De Silva e Tenenbaum (2004) Joia et al. (2011) Fadel et al. (2015)
Isomap	Maaten e Hinton (2008) Fadel et al. (2015)	De Silva e Tenenbaum (2003) Maaten (2007) Izenman (2008) Hoffmann et al. (2009) Maimon e Rokach (2010) Paulovich et al. (2011) Joia et al. (2011)
L-Isomap	Maaten e Hinton (2008) Fadel et al. (2015)	De Silva e Tenenbaum (2003, 2004) Maimon e Rokach (2010)
LSP	Fadel et al. (2015)	Paulovich et al. (2011) Joia et al. (2011)
SNE	Maaten e Hinton (2008) Lee et al. (2015)	Maaten (2007) Hoffmann et al. (2009)

número de vizinhos implica mudar a natureza da projeção, dependendo fortemente da sua formulação matemática. Segundo os autores da técnica, a "LLE tende a recuperar a estrutura global não linear a partir das combinações localmente lineares". Pela forma como é construída, isto de fato ocorre (ver Seção 3.1.2).

Outros casos envolvendo a natureza local e/ou global das técnicas de projeção são apresentados nas próximas seções, junto com a discussão da técnica.

3.1.2 Locally Linear Embedding (LLE)

A proposta de Roweis e Saul (2000) na LLE, apoia-se no fato de que a geometria local em um espaço de alta dimensão pode ser preservada em um espaço equivalente de menor dimensão. A Figura 3.1 ilustra o *pipeline* da técnica LLE. Os detalhes de cada passo são discutidos a seguir.

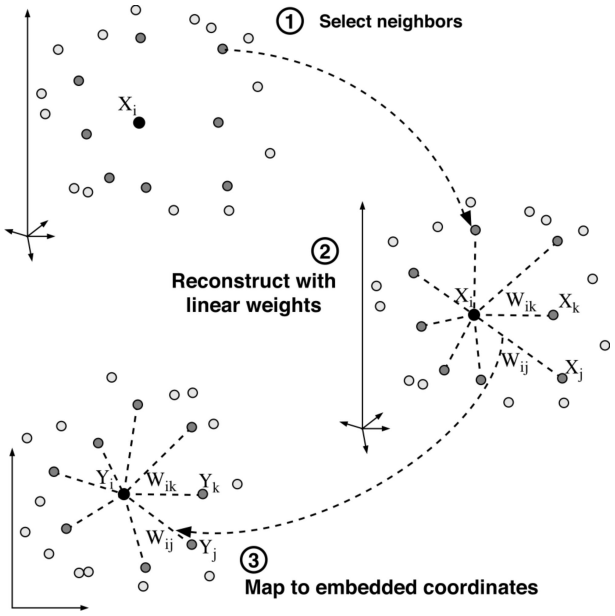


Figura 3.1: Passos da técnica LLE (Retirado de Roweis e Saul (2000)).

Considere os conjuntos de dados $X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^{n \times m}$, $Y = \{y_1, y_2, \ldots, y_n\} \subset \mathbb{R}^{n \times p}$ e a matriz de pesos W contendo os escalares w_{ij}. LLE consiste, basicamente, em definir uma vizinhança em torno das entradas x_i e minimizar duas funções-objetivo. A primeira com relação aos pesos:

$$\varepsilon(W) = \sum_i \left\| x_i - \sum_j w_{ij} x_j \right\|^2. \tag{3.1}$$

A segunda, fixa os pesos W, e minimiza a função em relação à variável Y:

$$\Phi(Y) = \sum_i \left\| y_i - \sum_j w_{ij} y_j \right\|^2. \tag{3.2}$$

O Algoritmo 3.1 esclarece cada uma das etapas.

Algoritmo 3.1 *Locally Linear Embedding* (LLE)

Entrada: Conjunto de dados $X \subset \mathbb{R}^{n \times m}$, dimensão do espaço reduzido p e número de vizinhos k.

Saída: Conjunto de dados projetado $Y \subset \mathbb{R}^{n \times p}$.

1: Atribuir k vizinhos a cada instância $x_i \in X$.

2: Computar os pesos w_{ij} que melhor representam cada ponto x_i, reconstruído como uma combinação linear dos demais pontos, ou seja: $x_i = \sum_j w_{ij} x_j$, tal que:

(i) $w_{ij} = 0$ se x_j não pertence à vizinhança de x_i e

(ii) $\sum_j w_{ij} = 1$, as linhas da matriz de pesos somam um.

Alcançado pela minimização de $\varepsilon(W)$ (Equação (3.1)).

3: Com os pesos w_{ij} do passo anterior, encontrar os vetores y_i contendo as coordenadas de cada ponto no espaço reduzido, também reconstruído como uma combinação linear de seus pares: $y_i = \sum_j w_{ij} y_j$, sujeito às seguintes restrições:

(i) $\sum_i y_i = 0$, as coordenadas mapeadas são centradas na origem e

(ii) $\frac{1}{n} \sum_i y_i y_i^\top = I$, os vetores mapeados têm covariância unitária.

Alcançado pela minimização de $\Phi(Y)$ (Equação (3.2)).

Note que o algoritmo, além da dimensão p do espaço reduzido, tem apenas mais um parâmetro livre, os k-vizinhos de cada ponto, que podem ser atribuídos de vários modos: usando *k-Nearest Neighbors* (k-NN) com a métrica Euclidiana, por exemplo; considerando todos os pontos em uma bola de raio fixo; ou via alguma heurística. Mas o fato é que a dimensão do espaço reduzido deve ser estritamente menor do que o número de vizinhos fixado ($p \ll k$).

No Passo 2, a fim de reconstruir os pontos x_i com base nos pesos w_{ij} e nos demais pontos x_j, LLE requer as coordenadas dos dados em $\mathbb{R}^m$. Por outro lado, não precisa calcular pares de distância para todo o conjunto, apenas os grafos de vizinhança. Minimizar a Equação (3.1) com respeito a W, sujeito às restrições 2(i) e 2(ii) do Algoritmo 3.1 é um problema solúvel por mínimos quadrados. Além disso, os pesos que minimizam esta função-objetivo obedecem a uma importante simetria: são invariantes a rotações, escalas e translações daqueles pontos e seus vizinhos. Particularmente a invariância à translação fica garantida pela restrição 2(ii), onde a soma de cada linha da matriz de pesos vale um (Roweis e Saul, 2000).

A Equação (3.2) deve ser minimizada com respeito a Y, onde Y representa o conjunto de dados mapeado para o espaço de baixa dimensão. Note que esta equação pode ser reescrita na forma matricial usando a norma de *Frobenius*:

$$\Phi(Y) = \|Y - YW^\top\|_F^2 = \|Y(I - W)^\top\|_F^2 \ . \tag{3.3}$$

E, em função do traço, como:

$$\Phi(Y) = tr[Y(I-W)^\top(I-W)Y^\top] = tr[YMY^\top], \tag{3.4}$$

onde $M = (I-W)^\top(I-W)$ é uma matriz $m \times m$ esparsa, simétrica e positiva semidefinida. A função-objetivo $tr[YMY^\top]$ pode ser minimizada encontrando-se os autovetores com os menores autovalores não nulos de M (Bai et al., 2000; Horn e Johnson, 1990).

LLE tem complexidade quadrática, não faz uso de amostras representativas, tampouco permite interatividade (por construção). Incluir novas instâncias no mapeamento implica obter seus vizinhos, recalcular todos os pesos e minimizar a Equação (3.2) novamente, i.e., executar todos os passo do algoritmo, portanto, também não é incremental.

É uma técnica não linear. Embora os pesos w_{ij} e os vetores y_i sejam calculados por métodos da álgebra linear, as restrições impostas às Equações (3.1) e (3.2) resultam em dados de natureza altamente não linear.

Por preservar propriedades locais, LLE consegue projetar variedades não convexas, mas apresenta dificuldades para tratar variedades que contêm buracos (Roweis e Saul, 2000). Além disso, tende a aglomerar grandes quantidades de dados no espaço de baixa dimensão devido à restrição da covariância (Maaten et al., 2009).

3.1.3 Orthogonal Neighborhood Preserving Projections (ONPP)

Com base na LLE, Kokiopoulou e Saad (2007) construiram uma nova técnica denominada *Orthogonal Neighborhood Preserving Projections* (ONPP). O método proposto por eles projeta os dados para o espaço de baixa dimensão usando uma transformação linear V.

Admitindo os mesmos conjuntos de dados X, Y e a matriz de pesos W definidos anteriormente para a LLE, é possível expressar cada elemento y_i em função de uma certa matriz de transformação $V \subset \mathbb{R}^{m \times p}$ a ser determinada, ou seja: $y_i = V^\top x_i$, $i = 1, ..., n$. Note que x_i e y_i são tomados como vetores-coluna, a fim de que a transformação V fique à esquerda na multiplicação, logo $Y = V^\top X$, a menos de uma transposição. Impondo-se este *mapeamento explícito* de X para Y na Equação (3.3) da LLE, obtém-se:

$$\Phi(Y) = \|V^\top X(I-W)^\top\|_F^2 \ , \tag{3.5}$$

que, em função do traço, resulta:

$$\Phi(Y) = tr[V^\top X(I-W)^\top(I-W)X^\top V] = tr[V^\top \tilde{M} V]. \tag{3.6}$$

A ONPP ainda impõe uma restrição de ortogonalidade adicional ao mapeamento, isto é, $V^\top V = I$. Então a solução V do problema de otimização acima é a base de autovetores

associados com os p menores autovalores não nulos da matriz $\tilde{M}$, definida como:

$$\tilde{M} = X(I - W)^\top (I - W)X^\top. \tag{3.7}$$

O Algoritmo 3.2 elucida cada etapa desta técnica.

Algoritmo 3.2 *Orthogonal Neighborhood Preserving Projections* (ONPP)

Entrada: Conjunto de dados $X \subset \mathbb{R}^{m \times n}$, dimensão do espaço reduzido p e número de vizinhos k.

Saída: Conjunto de dados projetado $Y \subset \mathbb{R}^{p \times n}$.

1: Computar os k vizinhos mais próximos de cada instância $x_i \in X$.

2: Computar os pesos w_{ij} que fornecem a melhor reconstrução linear de cada instância x_i com base na sua vizinhança.

3: Computar as projeções $y_i = V^\top x_i$, onde V é determinado pelo cálculo dos p autovetores de $\tilde{M}$ (Equação (3.7)) associados aos menores autovalores não nulos.

ONPP constrói um grafo de vizinhos mais próximos (k-NN) com pesos, o qual modela explicitamente a topologia dos dados. De modo similar à LLE, os pesos possibilitam capturar a geometria da vizinhança de cada ponto. Também parte do princípio de que cada ponto no espaço reduzido pode ser reconstruído a partir dos seus vizinhos, pelos mesmos pesos usados no espaço de origem.

Em contraste à LLE, calcula um mapeamento linear explícito $Y = V^\top X$, do espaço de origem para o espaço reduzido. Na LLE este mapeamento é implícito e não permite projetar novas instâncias. No caso da ONPP, uma vez que a matriz de redução de dimensionalidade V tenha sido encontrada, a projeção de novas instâncias é realizada por uma simples multiplicação de matrizes (Passo 3 do Algoritmo 3.2), portanto esta técnica é incremental.

Assim como sua predecessora, ONPP também tem complexidade $O(n^2)$. É baseada em decomposição espectral e, por ser proveniente da LLE, herda algumas de suas desvantagens:

- Não prevê o uso de amostras representativas para acelerar o processo de projeção.
- Tem custo computacional elevado, o que dificulta sua utilização em aplicações interativas.
- Requer dados de entrada como vetores em $\mathbb{R}^m$.

Contudo, possui algumas vantagens sobre a LLE:

- Capacidade de projetar novas instâncias de modo incremental.
- Força a projeção a ser ortogonal, portanto, invariante a efeitos como escala e cisalhamento.

É uma técnica de redução de dimensionalidade linear cuja tendência é preservar a vizinhança local, bem como a geometria global dos dados (Kokiopoulou e Saad, 2007).

3.1.4 FastMap

A técnica *FastMap* foi desenvolvida por Faloutsos e Lin (1995) com o objetivo de fornecer uma ferramenta de visualização e recuperação de informações a partir de grandes coleções de dados. Para lidar com grandes volumes de informações era necessário um algoritmo mais eficiente do que as tradicionais técnicas MDS da época, em geral de ordem quadrática (técnicas baseadas em MDS são discutidas na Seção 3.1.7). Assim os autores propuseram um algoritmo cuja ideia era imaginar as instâncias de dados como pontos em um espaço de alta dimensão m e projetar estes pontos em p direções mutuamente ortogonais, onde p é definido pelo usuário.

Para tanto, considere n o número total de instâncias, d_{ij} a dissimilaridade entre dois pontos quaisquer no espaço m-dimensional e d^*_{ij} a dissimilaridade entre suas respectivas imagens no espaço p-dimensional, $p < m$.

Chama-se *projeção ortogonal* (ou simplesmente projeção) de um ponto sobre uma reta ao ponto de interseção da reta com a perpendicular a ela conduzida por aquele ponto (ver Figura 3.2(a)). Observe que P' é a projeção do ponto P sobre a reta r, tal que $PP' \perp r$ e $PP' \cap r = \{P'\}$.

Do mesmo modo a projeção de um segmento de reta PQ não perpendicular a uma reta r, sobre esta reta é o segmento $P'Q'$, tal que P' é a projeção de P sobre r e Q' é a projeção de Q sobre r (Figura 3.2(b)).

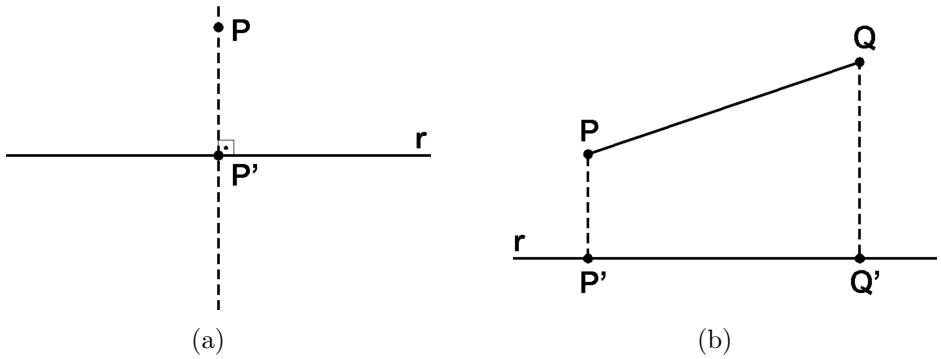


Figura 3.2: Projeção ortogonal: (a) do ponto P sobre a reta r; (b) do segmento de reta PQ sobre a reta r.

Para realizar a projeção a partir de um espaço de alta dimensão, os autores escolheram dois objetos Oa e Ob neste espaço, denominados objetos-pivô. Os pivôs devem ser pontos o mais distante possível um do outro. Embora esta escolha implique $O(n^2)$ passos, eles utilizaram uma heurística com base em $O(n)$ operações para selecionar os pivôs (veja Faloutsos e Lin (1995) para o algoritmo). Então imaginaram uma reta unindo os pivôs no

espaço de alta dimensão. A projeção de um objeto Oi sobre esta reta pode ser observada na Figura 3.3. No triângulo $OaOiOb$ obtido, o segmento $OaOi$ projetado sobre o lado $OaOb$ do triângulo determina um segmento de comprimento x_i.

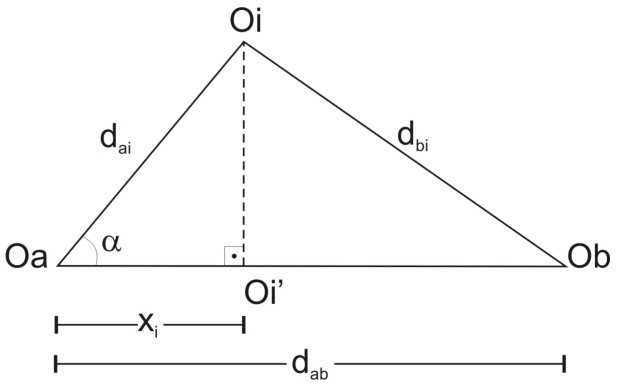


Figura 3.3: Projeção ortogonal do ponto Oi sobre a reta $OaOb$ (Retirado de Faloutsos e Lin (1995)).

Aplicando-se o teorema de *Pitágoras* nos dois triângulos retângulos $OaOi'Oi$ e $ObOi'Oi$, obtém-se a seguinte relação métrica:

$$d_{bi}{}^2 = d_{ai}{}^2 + d_{ab}{}^2 - 2x_i d_{ab}.$$

Que resolvida em x_i, resulta:

$$x_i = \frac{d_{ai}{}^2 + d_{ab}{}^2 - d_{bi}{}^2}{2d_{ab}}. \tag{3.8}$$

Um dos desafios da técnica era obter as projeções a partir de uma matriz de dissimilaridades (Definição 2.17), o que de fato fica resolvido com a Equação (3.8), onde a projeção Oi' depende somente das medidas de distância. Note que o ponto Oi neste caso, foi mapeado para uma reta (unidimensional), portanto, solúvel para $p = 1$.

Para mapear objetos em um espaço bidimensional (ou mesmo p-dimensional) deve-se seguir a mesma ideia: imaginar que os objetos são pontos de um espaço m-dimensional, considerar um hiperplano H de dimensão $(m - 1)$ perpendicular à reta $OaOb$, então projetar os pontos neste hiperplano (Figura 3.4). Daí em diante o problema se resolve da mesma forma que o anterior, restando apenas uma forma de encontrar a distância d^* entre duas projeções no hiperplano H, a qual é dada pelo Lema 3.1 a seguir.

Lema 3.1 *No hiperplano H, a distância Euclidiana d^* entre as projeções Oi' e Oj' pode ser calculada a partir da distância original d, do seguinte modo:*

$$d^*(Oi', Oj')^2 = d(Oi, Oj)^2 - (x_i - x_j)^2 \quad i, j = 1, \ldots, n. \tag{3.9}$$

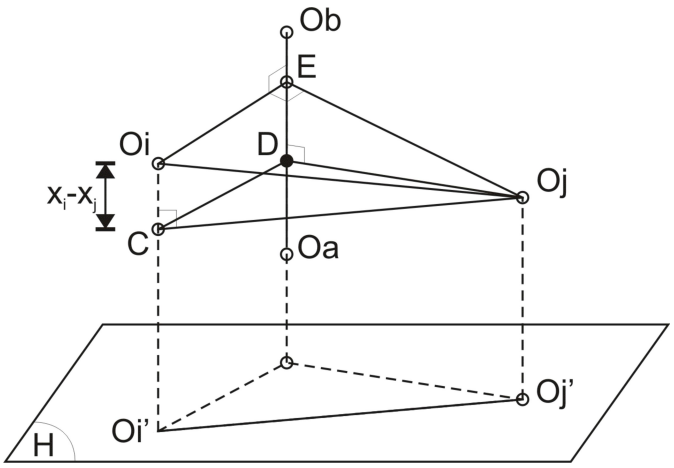


Figura 3.4: Projeção dos pontos Oi e Oj no hiperplano H, perpendicular à reta $OaOb$ definida na figura anterior (Retirado de Faloutsos e Lin (1995)).

A Equação (3.9) provém da aplicação direta do teorema de *Pitágoras* no triângulo $OiCOj$, retângulo em C. Desta forma, o problema pode ser resolvido para $p = 2$ e, se aplicados os mesmos passos recursivamente, p vezes, pode ser resolvido para qualquer p, lembrando que para cada dimensão dois novos pivôs devem ser escolhidos, resultando em um total de $2p$ pivôs ao final do processo. A complexidade do algoritmo é linear e depende de cada chamada recursiva, ou seja $O(pn)$.

FastMap tem sua formulação matemática apoiada em teoremas da geometria Euclidiana clássica. Faz uso de um número bem reduzido de amostras representativas – os pivôs – que servem para assegurar a condição de ortogonalidade e orientar a projeção. O mapeamento ocorre ponto-a-ponto e não como uma única transformação, mas a natureza da transformação é não linear global (Maaten, 2007).

É ideal para realizar buscas por conteúdo: consulta-por-exemplo (*query-by-example*), pois quando um novo objeto a ser consultado é fornecido, ele pode ser mapeado para o espaço de destino usando os mesmos objetos-pivô do mapeamento global, sem modificá-lo (incremental), mas por construção não permite interatividade. A eficiência computacional torna *FastMap* uma boa candidata para aplicações dinâmicas (*on-the-fly*), contudo, os resultados nem sempre são tão precisos como em outras técnicas MDS.

3.1.5 Sammon's Mapping (SM)

Sammon's Mapping (Sammon, 1969) é uma das mais tradicionais técnicas de projeção de dados multivariados[2] baseada em minimização não linear.

Esta técnica procura minimizar a função de erro E_{sm} (Equação (3.11)), mantendo uma estrutura global, sem uso de qualquer subconjunto de amostras representativas ou mesmo

[2]O mesmo que multidimensionais.

relações de vizinhança, requerendo apenas informações de distância entre os pontos, conforme explicado abaixo.

Considere o conjunto de dados X com n-vetores x_i $(i = 1, ..., n)$, contidos em um espaço de alta dimensão m. Correspondente a este, considere um outro conjunto com n-vetores no espaço p-dimensional ($p = 2$ ou 3) designado por Y, tal que as distâncias entre pares de elementos nesses conjuntos sejam denotadas por $d_{ij} = dist(\mathrm{x}_i, \mathrm{x}_j)$ e $d^*_{ij} = dist(y_i, y_j)$, respectivamente.

Como primeiro passo da técnica, é necessário escolher uma configuração inicial p-dimensional para o conjunto de vetores Y. Os vetores que aparecem na Equação (3.10), aleatoriamente definidos, denotam essa configuração. Segundo Sammon esta configuração inicial aleatória é suficiente para demonstrar a técnica. Todavia, na prática, a configuração inicial dos vetores é encontrada projetando-se os dados m-dimensionais ortogonalmente em um espaço p-dimensional, gerado pelas p coordenadas originais com as maiores variâncias.

$$y_1 = \begin{bmatrix} y_{11} \\ \vdots \\ y_{1p} \end{bmatrix}, \quad y_2 = \begin{bmatrix} y_{21} \\ \vdots \\ y_{2p} \end{bmatrix}, \ \ldots, \ y_n = \begin{bmatrix} y_{n1} \\ \vdots \\ y_{np} \end{bmatrix}. \tag{3.10}$$

Depois disso, calculam-se todas as interdistâncias entre pares de vetores a serem utilizadas pela função de erro (Equação (3.11)). Note que o erro é uma função com $p \times n$ variáveis y_{qr}, tal que $q = 1, \ldots, n$ e $r = 1, \ldots, p$.

$$E_{sm} = \frac{1}{\sum_{i<j}(d_{ij})} \sum_{i<j}^{n} \frac{(d_{ij} - d^*_{ij})^2}{d_{ij}}. \tag{3.11}$$

O último passo do algoritmo é ajustar as y_{qr} variáveis, i.e., mudar a configuração do espaço p-dimensional de modo a minimizar o valor do erro E_{sm}. Para este fim SM aplica o método do gradiente descendente. Este método implica o cálculo das derivadas parciais de 1^a e 2^a ordem cada vez que as coordenadas y_{qr} são atualizadas. O processo se repete até que um limite de convergência seja alcançado. O algoritmo é de ordem quadrática e depende do número de iterações c aplicadas, além do número de instâncias n, ou seja, $O(cn^2)$.

Nesta técnica, o mapeamento ocorre ponto-a-ponto, não existe uma função de mapeamento explícito, tampouco é capaz de acomodar novos pontos sem interferir em todo o mapeamento. Além disso, pelo fato de precisar computar e armazenar todas as distâncias entre pontos, torna-se inviável para aplicações práticas onde os dados chegam sequencialmente, ou a quantidade de dados é grande.

3.1.6 Pekalska Approximation

A fim de reduzir o custo computacional de *Sammon's Mapping* (SM), Pekalska et al. (1999) propuseram uma técnica que projeta um subconjunto de s-amostras no espaço visual otimizando uma função de *stress*. Em seguida, projeta as instâncias remanescentes usando um mapeamento linear global, resultando em um algoritmo $O(2s^3 + sn)$, conforme será explicado a seguir.

A complexidade de SM é alta porque a função E_{sm} (Equação (3.11)) se baseia em $O(n^2)$ distâncias, além disso, novos pontos não podem ser adicionados ao *layout* sem recalcular todos os demais. Nesse sentido, duas medidas foram tomadas: primeiro um algoritmo mais adequado de minimização foi adotado. Na concepção dos autores, *Pseudo-Newton*, em geral, funciona melhor do que o método do gradiente. A segunda melhoria concerne à ideia de executar *Sammon's Mapping* somente em um subconjunto de pontos, por eles denominado base. O restante dos pontos seriam adicionados a posteriori, aplicando uma única transformação linear V, tal que V preserva distâncias e evita distorções na função do *stress*. Em virtude disso, a técnica pode ser considerada híbrida com relação ao tipo de transformação de dados, já que a primeira parte, baseada em otimização, é não linear e esta última, linear.

Com relação à parte não linear, uma questão a ser respondida era quantos pontos seriam necessários para constuir uma base. Isto foi determinado empiricamente através de simulações, comparando o resultado do *stress* obtido com SM versus o da nova proposta, e a melhor escolha ficou em torno de 50% do total de instâncias, $n/2$.

A parte linear é processada por um método denominado *Mapeamento de Distâncias*. Considere n o número total de instâncias, s o número de amostras da base ($s < n$), m a dimensão do espaço de origem e p a dimensão do espaço de projeção ($p < m$). Seja $X_{base} \subset \mathbb{R}^{s \times m}$ a base de s pontos escolhidos randomicamente para otimização por SM e D_{base} a sua correspondente matriz de distâncias, tal que $Y_{base} \subset \mathbb{R}^{s \times p}$ é o resultado dessa otimização. Considerando que a matriz D_{base}, por ser uma matriz de distâncias tem posto completo, então deve existir uma transformação linear $V \subset \mathbb{R}^{s \times p}$ que aproxime:

$$D_{base}^{s \times s} \cdot V^{s \times p} = Y_{base}^{s \times p} \, . \tag{3.12}$$

Note que *Pekalska* não requer os dados de entrada como vetores em $\mathbb{R}^m$, podendo lidar diretamente com a matriz de distâncias.

Uma vez encontrado V no sistema acima, os pontos remanescentes denotados por $Y_{new} \subset \mathbb{R}^{(n-s) \times p}$ podem ser mapeados como segue:

$$D_{new-to-base}^{(n-s) \times s} \cdot V^{s \times p} = Y_{new}^{(n-s) \times p} \, , \tag{3.13}$$

onde a matriz $D_{new-to-base} \subset \mathbb{R}^{(n-s)\times s}$ é a matriz contendo todas as distâncias entre pontos dos conjuntos X_{base} e X_{new}, conduzidas pela mesma transformação V.

A análise é a seguinte: na Equação (3.12), Y_{base} representa as coordenadas dos pontos da base mapeados para $\mathbb{R}^p$ por *Sammon's Mapping*, mas a entrada D_{base} representa as distâncias entre esses pontos, não suas coordenadas X_{base} como seria natural esperar no sistema, então V comporta-se como uma "transformada" do domínio da distância para o domínio de $\mathbb{R}^p$, garantido por SM. Logo, se aplicarmos a mesma transformada V para as distâncias entre pontos de X_{base} (tomado como referência) e X_{new}, num total de $s(n-s)$ distâncias, é de se esperar que os pontos em X_{new} sejam mapeados corretamente para $\mathbb{R}^p$ também.

Pela sua formulação, *Pekalska* é incremental, visto que consegue acomodar novos pontos sem refazer todo o mapeamento. Para mapear uma nova instância x_{n+1} é suficiente calcular as s distâncias de x_{n+1} aos elementos de X_{base} e aplicar a transformação V sobre o vetor de distâncias obtido.

Embora mais eficiente que outros métodos baseados em otimização, a abordagem de *Pekalska* não é flexível o suficiente para suportar aplicações interativas, pois requer um número elevado de amostras representativas (base) para projetar os dados. Apesar dessa limitação, *Pekalska* é uma técnica muito precisa com relação à preservação de distâncias. Nas comparações realizadas com a LAMP no Capítulo 4, mostrou-se uma das mais competitivas em termos de *stress* (ver Figura 4.4(a)).

3.1.7 Multidimensional Scaling (MDS) e Landmark MDS

O algoritmo *Multidimensional Scaling* (**MDS**) clássico teve origem nos trabalhos de Richardson (1938) e Yong e Householder (1938), mas foi popularizado por Torgerson (1952). Atualmente constitui uma família de técnicas usadas para analisar a proximidade entre objetos de dados. Muitos métodos de projeção multidimensional derivam da teoria MDS, a qual requer apenas informação de distância entre pares de objetos para executar o mapeamento, tornando as coordenadas do espaço de origem desnecessárias (Steyvers, 2002).

Quando o número de objetos é muito grande, o algoritmo MDS clássico não se mostra eficiente, pois necessita calcular os p autovalores e autovetores da matriz de distâncias $D_{n\times n}$ informada como entrada, onde p é a dimensão do espaço reduzido.

Para lidar com esta situação, o algoritmo *Landmark MDS* (**LMDS**) (De Silva e Tenenbaum, 2003, 2004) foi desenvolvido, e pode ser interpretado como uma variação de MDS, preservando todas as suas propriedades, porém com mais eficiência, já que trabalha com uma submatriz $D_{s\times n}$, obtida das distâncias entre as n instâncias de dados e um conjunto contendo s pontos de referência denominados *landmarks*.

LMDS consiste de quatro passos principais, conforme descrito no Algoritmo 3.3.

Algoritmo 3.3 *Landmark MDS* (LMDS)

Entrada: Conjunto de n pontos em $\mathbb{R}^m$, dimensão do espaço reduzido p e número de *landmarks* s.

Saída: Conjunto de dados mapeado $Y \subset \mathbb{R}^p$.

1: Especificar um conjunto de s pontos *landmarks*.

2: Aplicar MDS clássico para encontrar a matriz $L_{p\times s}$, que representa o mapeamento dos s pontos *landmarks* em $\mathbb{R}^p$. Como entrada, usar a matriz de distâncias $D_{s\times s}$ entre os pares de pontos *landmarks*.

3: Aplicar a triangulação baseada em distâncias para encontrar uma matriz $Y_{p\times n}$ que representa o mapeamento dos n pontos em $\mathbb{R}^p$. Como entrada, usar a matriz de distâncias $D_{s\times n}$, entre os pontos *landmarks* e os pontos do conjunto de dados. As novas coordenadas são derivadas a partir das distâncias ao quadrado por uma transformação linear afim.

4: Recentrar os dados sobre sua média, e usar PCA para alinhar os eixos principais dos dados recém-mapeados com os eixos coordenados, em ordem decrescente de significância (opcional).

A seleção dos pontos realizada no Passo 1 pode ser feita de duas maneiras: (i) de forma aleatória ou (ii) pela aplicação do algoritmo MaxMin que seleciona os pontos, um por vez, onde cada novo *landmark* maximiza, sobre todos os pontos não usados, a distância mínima a quaisquer dos *landmarks* existentes. Neste processo, o primeiro ponto é escolhido aleatoriamente. O custo de usar MaxMin ao invés de escolha aleatória é de $O(sn)$ operações extras. Para uma projeção p-dimensional satisfatória, é requerido no mínimo $p + 1$ *landmarks* selecionados, no entanto, é conveniente selecionar mais pontos *landmarks* do que estritamente o mínimo.

O Passo 2, aplica MDS clássico para mapear os s pontos *landmarks* para o espaço $\mathbb{R}^p$, usando a matriz $D_{s\times s}$ como entrada, e não a matriz completa $D_{n\times n}$. Isto reduz bastante o custo computacional associado ao cálculo e armazenamento da matriz.

O procedimento de triangulação com base em distâncias, citado no Passo 3, é tratado como um problema linear. O Passo 4 não é essencial, mas normaliza a saída.

Uma característica importante da LMDS é permitir a introdução de novos pontos de forma contínua, realizando apenas um cálculo global se for exigido que as coordenadas encontradas estejam alinhadas com respeito aos eixos principais, ou seja, após os pontos *landmarks* serem fixados e os cálculos iniciais terem sido realizados, todos os pontos são projetados independentemente um do outro usando uma transformação linear fixa. Usando a distância Euclidiana entre pares de objetos, a complexidade computacional é aproximadamente $O(s^3 + psn)$, onde s é o número de pontos *landmarks*, p é a dimensão desejada e n é o número total de instâncias.

Embora seja preciso em termos de preservação de distâncias, o método LMDS não possui um mecanismo para interagir com os dados projetados (Paulovich et al., 2010a). Dado o conjunto de pontos *landmarks*, LMDS pode trabalhar diretamente com as submatrizes de distâncias $D_{s\times s}$ e $D_{s\times n}$. Além disso, é uma técnica de natureza global e incremental.

3.1.8 Isometric Feature Mapping (Isomap) e Landmark Isomap

A técnica *Landmark Isomap* (L-Isomap) proposta por De Silva e Tenenbaum (2003) se apoia em outra, a *Isometric Feature Mapping* (Isomap) (Tenenbaum et al., 2000), portanto, iniciaremos a análise por ela.

Isomap é uma técnica de redução de dimensionalidade não linear que usa MDS clássico com distâncias geodésicas entre os pontos de uma variedade. Ela pode computar soluções ótimas de forma global. Para uma classe de variedades como *Swiss roll*, consegue convergir assintóticamente para uma estrutura exata. No entanto, não consegue lidar facilmente com domínios mais complexos, tais como curvaturas não triviais, buracos ou saliências, esferas e variedades não convexas. Em geral, requer um conjunto denso de pontos para estimar variedades, sendo fortemente dependente da vizinhança dos pontos para obter êxito.

Os principais passos da Isomap são apresentados no Algoritmo 3.4.

Uma falha no algoritmo Isomap é ser topologicamente instável. Conexões no grafo de vizinhos G podem ser construídas erroneamente e sua estabilidade depende dos valores de ϵ ou k (Balasubramanian et al., 2002).

Devido à sua complexidade na ordem de $O(n^3)$, Isomap torna-se inviável para grandes conjuntos de dados. A técnica também enfrenta problemas com matrizes densas, tem que armazenar a matriz de distâncias completa e, quando aplicada a dados reais, mostra-se sensível a ruídos e perturbações.

L-Isomap é uma técnica que combina as vantagens da abordagem global com as vantagens exclusivas dos métodos locais. Ela aproxima a extensa computação global da Isomap para um conjunto reduzido de cálculos, além de suprir suas principais deficiências e, faz uso de um pequeno conjunto de dados denominado *pontos landmarks* (De Silva e Tenenbaum, 2003).

A L-Isomap trabalha designando s instâncias para serem os pontos *landmarks*, onde $s \ll n$. Em seguida, computa a matriz de distâncias de cada ponto aos pontos *landmarks*, $D_{s\times n}$, reduzindo o cálculo de distâncias para $O(ksn \log n)$, onde k é o tamanho da vizinhança. Então aplica LMDS (ao invés de MDS) sobre a matriz de distâncias para encontrar a solução.

Por construção, L-Isomap não permite interatividade e tem sua formulação matemática baseada em decomposição espectral.

Algoritmo 3.4 *Isometric Feature Mapping* (Isomap)

Entrada: Matriz D_X contendo as distâncias $d_X(i,j)$ entre todos os pares de pontos i,j do conjunto de dados $X_{n\times m}$; um raio fixo ϵ (ou o tamanho da vizinhança k).
Saída: Conjunto de dados mapeado $Y \subset \mathbb{R}^p$.

1: Determinar quais pontos são vizinhos na variedade M com base nas distâncias $d_X(i,j)$, podendo ser realizado de duas formas distintas:

(i) Conectando cada ponto a todos os outros pontos dentro de uma bola de raio fixo ϵ, ou;

(ii) Tomando os k-vizinhos mais próximos (k-NN) de cada ponto.

Estes relacionamentos de vizinhança são representados como um grafo ponderado G sobre todos os pontos do conjunto de dados X, onde os pesos das arestas são dados por $d_X(i,j)$.

2: Estimar as distâncias geodésicas $d_M(i,j)$ entre todos os pares de pontos na variedade M, computando os caminhos mais curtos $d_G(i,j)$ no grafo G. Este procedimento requer:

(i) Inicializar $d_G(i,j) = d_X(i,j)$ se i,j estão conectados por uma aresta, caso contrário $d_G(i,j) = \infty$;

(ii) Para cada $l = 1, 2, \ldots, n$, substituir todas as entradas $d_G(i,j)$ pelo $\min\{d_G(i,j),\ d_G(i,l) + d_G(l,j)\}$.

A matriz final $D_G = \{d_G(i,j)\}$ irá conter as distâncias correspondentes aos caminhos mais curtos entre todos os pares de pontos de G.

3: Aplicar MDS clássico sobre a matriz de distâncias D_G, construindo o espaço p-dimensional Y que melhor preserva a geometria intrínseca dos dados.

3.1.9 Part-Linear Multidimensional Projection (PLMP)

Part-Linear Multidimensional Projection (PLMP) (Paulovich et al., 2010b) é uma técnica de projeção global, construída para trabalhar com grandes conjuntos de dados, pois requer uma quantidade reduzida de informações de distância, aumentando substancialmente a velocidade da projeção. Isto se deve ao fato da PLMP lidar com amostras representativas (ou pontos de controle). Estas amostras são inicialmente posicionadas no espaço visual, com o propósito de guiar o restante da projeção.

O posicionamento das amostras é realizado através de um esquema *não linear*. Em seguida, as demais instâncias são projetadas via um mapeamento *linear*, construído a partir das coordenadas cartesianas das amostras, justificando assim o termo "*Part-Linear*" no seu nome. As etapas da PLMP podem ser visualizadas por meio do *pipeline* ilustrado na Figura 3.5.

A etapa não linear apoia-se em um esquema baseado em força (*Force Scheme*) (Tejada et al., 2003), o qual faz uma analogia entre minimização de *stress* em sistemas massa-mola,

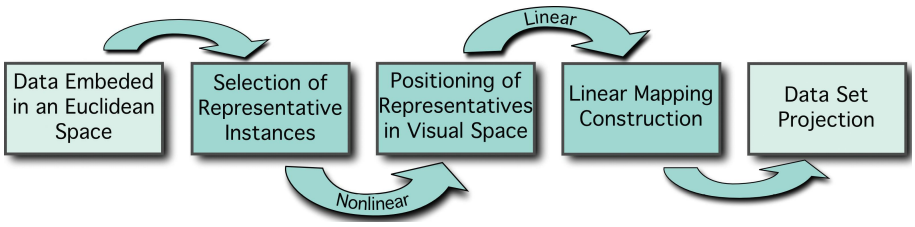


Figura 3.5: *Pipeline* da técnica PLMP (Retirado de Paulovich et al. (2010b)).

porém, a força restauradora do sistema é definida em função da distância residual entre os espaços de origem e de projeção. A escolha das amostras pode ser realizada de modo aleatório, em torno de $\sqrt{n}$, onde n é o número total de instâncias.

A PLMP parte do princípio de que se pode calcular uma transformação linear $\Phi : \mathbb{R}^m \to \mathbb{R}^p,\ p < m$ para mapear todo o conjunto de dados de modo eficiente, levando em conta apenas um subconjunto de amostras representativas mapeadas, a priori, na etapa não linear. Em termos matemáticos, Φ deve satisfazer:

$$\Phi = \underset{\hat{\Phi} \in \mathscr{L}_{m,p}}{\arg\min} \left\{ \frac{1}{D} \sum_{ij} (d(\mathrm{x}_i, \mathrm{x}_j) - d(\hat{\Phi}(\mathrm{x}_i), \hat{\Phi}(\mathrm{x}_j)))^2 \right\}, \tag{3.14}$$

onde $x_i,\ x_j$ são instâncias do conjunto de dados $X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^m$, $\mathscr{L}_{m,p}$ é o espaço da transformação linear $\mathbb{R}^m \to \mathbb{R}^p$, tal que $\hat{\Phi}$ é a transformação aplicada neste espaço e $D = \sum_{i,j} d(\mathrm{x}_i, \mathrm{x}_j)^2$.

Minimizar a Equação (3.14) diretamente, para todo o conjunto de dados, é inviável quando n é grande. A estratégia então é aproximar a solução usando somente as amostras, conforme será explicado a seguir.

Considere o subconjunto $X' \subset X$ contendo s-amostras representativas, com $s \ll n$, assim representado $X' = \{\mathrm{x'}_1, \mathrm{x'}_2, \ldots, \mathrm{x'}_s\}$. Admitindo que o subconjunto X' já tenha sido mapeado para o espaço de baixa dimensão usando o esquema baseado em forças mencionado no ínício, resultando em $Y' = \{\mathrm{y'}_1, \mathrm{y'}_2, \ldots, \mathrm{y'}_s\}$, com $Y' \subset \mathbb{R}^p$, então um mapeamento linear ideal Φ que minimiza a Equação (3.14) é aquele que satisfaz:

$$\Phi(x'_i) = y'_i, \quad \forall i,\ i = 1, \ldots, s.$$

Que em termos matriciais pode ser expresso como:

$$X'_{s \times m} \cdot \Phi_{m \times p} = Y'_{s \times p}. \tag{3.15}$$

Note que nesta equação, X' e Y' são conhecidos, logo deve ser resolvida na variável Φ. Também é esperado que $s > m$, o que normalmente ocorre para grandes conjuntos de dados.

É interessante compararmos esta formulação com a aproximação de *Pekalska*. Note que, apesar da estrutura semelhante, a formulação de *Pekalska* (Equação (3.12)) toma como entrada uma matriz de dissimilaridades, enquanto que a PLMP (Equação (3.15)) toma como entrada um conjunto de vetores definidos em $\mathbb{R}^m$.

Resolver a Equação (3.15) em Φ implica resolver p sistemas lineares da forma

$$X' \cdot \phi_j = \psi_j, \text{ com } j = 1, \ldots, p,$$

onde $\phi_j = [\phi_{1j}, \phi_{2j}, \ldots, \phi_{mj}]^\top$ e $\psi_j = [y_{1j}, y_{2j}, \ldots, y_{sj}]^\top$ representam, respectivamente, as variáveis da transformação Φ e as coordenadas dos pontos de controle.

O conjunto de sistemas definido pela Equação (3.15) pode ser resolvido de forma usual, fazendo:

$$X'^\top X' \cdot \Phi = X'^\top Y', \tag{3.16}$$

Portanto, resolvendo p sistemas lineares de ordem $m \times m$ encontramos uma aproximação para Φ. Lembrando que, no contexto de visualização $p = 2$ ou 3. Para valores moderados de m pode ser usada a fatoração de *Cholesky* ou, se for o caso, o método do gradiente conjugado.

Uma vez encontrada a transformação linear Φ, a projeção das demais instâncias em $\mathbb{R}^p$ reduz-se a uma simples multiplicação de matrizes, entre a matriz de entrada em $\mathbb{R}^m$ pela matriz de transformação Φ, operação que pode ser realizada com baixo custo computacional, $O(n)$.

Como consequência da sua eficiência computacional, a PLMP torna-se ideal para aplicações interativas e dinâmicas. É possível melhorar a projeção final manipulando interativamente a posição das amostras no espaço visual, todavia, sua natureza global limita o quanto a projeção pode ser melhorada, ou seja, não consegue acompanhar drástica separação de instâncias.

Observe que a formulação matemática apresentada nesta seção não utiliza informações de distância, a não ser para posicionar as amostras representativas no espaço de projeção, permitindo assim, mapear grandes conjuntos de dados. Esta característica, no entanto, limita a PLMP como ferramenta de exploração visual de coleções de documentos e dados textuais, já que não manipula informações de dissimilaridade diretamente.

Por construção, a PLMP é incremental e pode projetar dados em fluxo contínuo (*streaming*) e paralelizado, decorrente da forma como opera na fase linear, com base em multiplicações matriciais.

3.1.10 Piecewise Laplacian-based Projection (PLP)

A *Piecewise Laplacian-based Projection* (PLP) (Paulovich et al., 2011) usa um esquema baseado em força para posicionar um subconjunto de amostras no espaço visual. As

instâncias remanescentes são projetadas usando operadores locais do tipo Laplaciano, os quais são construídos a partir de grafos de vizinhança locais disjuntos.

O método empregado na PLP é constituído por três passos principais, conforme ilustrado na Figura 3.6, ou seja:

1. Amostragem: seleção de um pequeno subconjunto de instâncias. Esta seleção pode ser feita de forma aleatória ou, pode ser condicionada pelo usuário com o objetivo de guiar a projeção. O número de amostras $s = \sqrt{n}$ fornece um bom balaceamento entre custo computacional e qualidade do mapeamento final.

2. Construção dos grafos de vizinhança: para cada uma das amostras, um grafo de vizinhança e um conjunto de pontos de controle são definidos. Cada grafo dá origem a uma matriz Laplaciana que executa a projeção das correspondentes instâncias para os nós do grafo, já os pontos de controle restringem o sistema Laplaciano a fim de direcionar o posicionamento das amostras projetadas no espaço visual.

3. Resolução do sistema linear Laplaciano: neste mecanismo, cada elemento do conjunto de dados pode ser escrito como uma combinação convexa de seus vizinhos mais próximos no espaço visual.

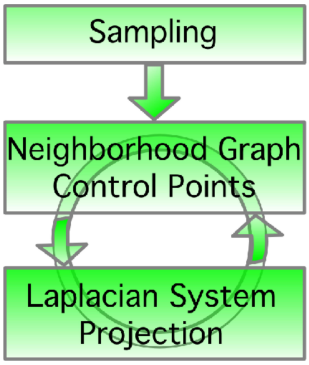
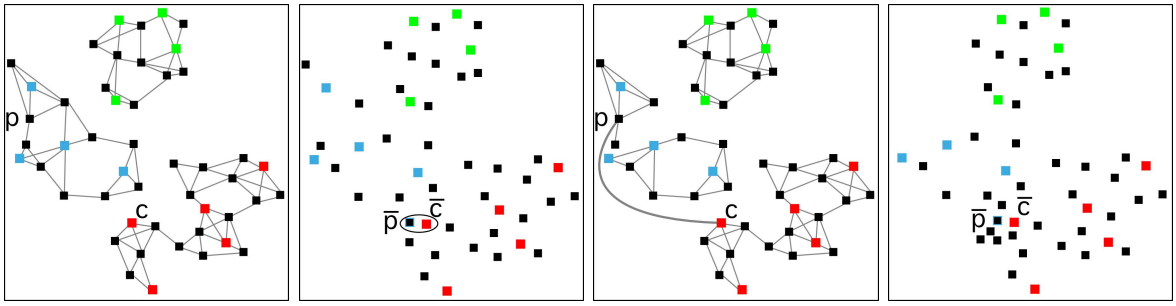


Figura 3.6: *Pipeline* da técnica PLP (Retirado de Paulovich et al. (2011)).

Flexibilidade em termos de interação do usuário é a principal qualidade da PLP, uma vez que o usuário pode mover as amostras projetadas, agrupando instâncias similares, e desse modo mudar o *layout* da projeção. Esta versatilidade é garantida pela sua natureza local, tanto que mudanças drásticas são possíveis porque os grafos de vizinhança local são reconstruídos durante a interação do usuário (ver Figura 3.7). A contínua atualização dos grafos locais, no entanto, aumenta o custo computacional e pode impactar na robustez.

Adaptação dinâmica do *layout*, custo computacional na ordem de $O(sn)$ e boa preservação de vizinhança tornam a PLP uma técnica atrativa para projeções envolvendo exploração interativa e visualização de grandes conjuntos de dados.



(a) (b) (c) (d)

Figura 3.7: Atualização dos grafos de vizinhança: dada uma projeção (a), o usuário arrasta e reposiciona instâncias projetadas (b). Em seguida, os grafos de vizinhança são atualizados para refletir as mudanças (c), modificando as matrizes laplacianas e a projeção (d) (Retirado de Paulovich et al. (2011)).

3.1.11 Least Square Projection (LSP)

Outra técnica revisada, a *Least Square Projection* (LSP) (Paulovich et al., 2008) destaca-se, principalmente, por produzir mapeamentos com alta preservação de vizinhança. É uma técnica de natureza, em geral, global que transforma os dados com base em um esquema não linear, resultando em um algoritmo da ordem $O(s^2 + n^2)$. Os passos da LSP são apresentados no Algoritmo 3.5, cujos detalhes são discutidos a seguir.

A qualidade da projeção da LSP depende do número e da distribuição adequada dos pontos de controle (Passo 1). Em geral $s = \sqrt{n}$ é suficiente, dependendo da técnica MDS utilizada. Para fazer a seleção, o conjunto de dados é quebrado em s grupos utilizando o método *k-medoides* (Berkhin, 2002), tal que o medoide (ponto mais próximo do centroide) de cada grupo é usado como ponto de controle.

Além das coordenadas cartesianas dos pontos de controle, é necessário definir uma lista de vizinhos $V_i \subset X$ para cada ponto $x_i \in X$ (Passo 2). A LSP emprega uma técnica simples baseada em agrupamentos para encontrar a vizinhança dos pontos. Esta abordagem é usual, já que o espaço foi quebrado anteriormente em s-grupos para selecionar os pontos de controle, e pelo menor custo que apresenta quando comparada à outras abordagens, em geral de ordem quadrática. Os detalhes deste procedimento estão disponíveis em Paulovich et al. (2008).

Quando um sistema linear é construído em conformidade com o Passo 3 do Algoritmo 3.5, os pontos $x_i \in X$ pertencem ao fecho convexo de sua vizinhança V_i, e se os pesos α_{ij} são dados por $\alpha_{ij} = \frac{1}{k_i}$ temos x_i no centroide dos pontos em V_i. Nestas condições, a matriz L (construída no Passo 3) é usualmente chamada de matriz Laplaciana. O operador Laplaciano faz uso de um grafo de vizinhança *global* entre os pontos de X, a partir do qual um grande sistema linear esparso é obtido.

Algoritmo 3.5 *Least Square Projection* (LSP)

Entrada: Conjunto de dados $X = \{x_1, x_2, \ldots, x_n\}$ de dimensão $n \times m$, dimensão do espaço reduzido p ($p < m$), número de pontos de controle s e número de vizinhos de cada instância k.

Saída: Conjunto de dados projetado $Y = \{y_1, y_2, \ldots, y_n\} \subset \mathbb{R}^{n \times p}$.

1: Mapear um subconjunto contendo s-amostras (*pontos de controle*) no espaço $\mathbb{R}^p$, por um método MDS conhecido ($s \ll n$).

2: Atribuir k-vizinhos a cada instância $x_i \in X$, denotados por V_i.

3: Supondo que cada ponto $y_i \in Y$ é dado por: $y_i - \sum_{x_j \in V_i} \alpha_{ij} y_j = 0$, sujeito a:

(i) $0 \leqslant \alpha_{ij} \leqslant 1$ e

(ii) $\sum_j \alpha_{ij} = 1$,

construir os sistemas lineares $L\psi_j = 0$, $j = 1, \ldots, p$, onde ψ_j são as coordenadas cartesianas dos pontos e L é a matriz $n \times n$, dada por:

$$l_{ij} = \begin{cases} 1, & i = j, \\ -\alpha_{ij}, & x_j \in V_i, \\ 0, & \text{caso contrário.} \end{cases}$$

4: Inserir informações geométricas no sistema a partir dos pontos de controle, como linhas na matriz L, e suas coordenadas cartesianas do lado direito do sistema, dando origem a um novo sistema da forma: $AY = b$.

5: Resolver o sistema obtido no passo anterior por mínimos quadrados, encontrando os $y_i \in Y$ com $i = 1, \ldots, n$.

Antes de prosseguir é interessante comparar o Passo 3 da LSP (Algoritmo 3.5) com o Passo 2 da LLE (Algoritmo 3.1). Note que as equações têm praticamente a mesma estrutura:

$$\text{LSP: } y_i = \sum_{x_j \in V_i} \alpha_{ij} y_j; \qquad \text{LLE: } x_i = \sum_j w_{ij} x_j.$$

À exceção das variáveis que mudam de nome, algumas observações são pertinentes. Por exemplo, na LLE os pesos w_{ij} precisam ser calculados, então o sistema é resolvido com respeito a W. Isto implica que os vetores x_j (vizinhos de x_i) precisam ser conhecidos. Na LSP, os pesos indicados por α_{ij} são impostos pela matriz Laplaciana ($1/k_i$), portanto o sistema é resolvido diretamente em Y e os pontos x_j não precisam ser conhecidos, apenas se eles pertencem ou não à vizinhança de x_i, dando à LSP maior flexibilidade nesse sentido (não requer dados de entrada contidos em $\mathbb{R}^m$).

Vale lembrar ainda que, na LSP, dependendo da vizinhança considerada no cálculo do mapeamento, é possível inserir informação em todas as escalas se muitos vizinhos são considerados, ou informação de localidade se poucos vizinhos são considerados. Porém,

esta característica não deve ser comparada à natureza local da LLE que leva em conta a vizinhança nos dois espaços, atrelando-os por intermédio dos pesos.

O Passo 4 da LSP, no entanto, insere as informações geométricas que faltam ao mapeamento, através dos pontos de controle. A Figura 3.8 ilustra como isso é feito.

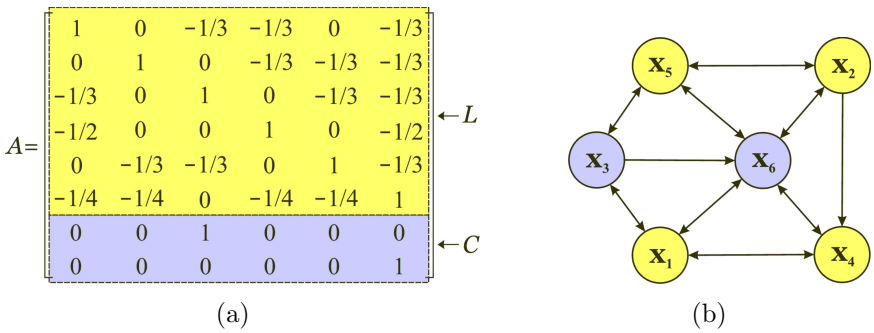


Figura 3.8: (a) Um exemplo de matriz Laplaciana L, acrescida dos pontos de controle C. (b) Relações de vizinhança entre os pontos usados para compor a matriz A, onde x_3 e x_6 são pontos de controle (Modificado de Paulovich et al. (2008)).

Observe que cada ponto de controle é inserido como uma linha de zeros na matriz L, exceto na posição que ele ocupa (dentro do conjunto de dados), neste caso, é inserido o valor um. Desse modo, a matriz L fica aumentada por s linhas, conforme mostra a Figura 3.8(a) e, passa a ser denominada matriz A, de dimensão $(n+s) \times n$. As coordenadas cartesianas dos pontos de controle em $\mathbb{R}^p$ também são inseridas neste sistema, do lado direito da equação, transformando o sistema Laplaciano original no sistema não homogêneo:

$$AY = b, \tag{3.17}$$

tal que,

$$A = \begin{bmatrix} L \\ C \end{bmatrix},$$

onde cada elemento c_{ij} de C é dado por:

$$c_{ij} = \begin{cases} 1, & \text{se } x_j \text{ é um ponto de controle,} \\ 0, & \text{caso contrário,} \end{cases}$$

e o vetor b, por:

$$b_i = \begin{cases} 0, & i \leq n, \\ \rho_{tj}, & n < i \leq n+s, \end{cases}$$

onde ρ_{tj} indica a j-ésima coordenada do ponto de controle associado ($t = 1, \ldots, s$ e $j = 1, \ldots, p$). Os pontos de controle guiam o processo de projeção, os quais podem ser

manipulados pelo usuário de modo a facilitar a visualização de agrupamentos e identificação de características nos dados. Portanto, a LSP é uma técnica interativa.

No Passo 5, as instâncias restantes são mapeadas resolvendo-se o sistema linear definido na Equação (3.17) pelo método dos mínimos quadrados, resultando em $Y = (A^\top A)^{-1} A^\top b$. Este sistema é simétrico e esparso o que facilita a solução. Mais detalhes sobre a solução e assertivas que garantem uma solução não trivial podem ser encontrados em Paulovich et al. (2008) e Sorkine e Cohen-Or (2004).

Embora o Passo 5 utilize ferramentas da álgebra linear, a técnica MDS empregada na primeira parte para posicionar as amostras pode resultar em dados finais relacionados de forma altamente não linear. Como uma pequena porção de instâncias são inicialmente mapeadas, é possível utilizar *Multidimensional Scaling* de maior custo computacional nesta etapa, visando aumentar a precisão das respostas sem comprometer a eficiência.

A LSP prevê a inserção de uma nova instância x_{n+1} no mapeamento. Esta inserção requer os seguinte passos:

i. Encontrar os vizinhos de x_{n+1}.

ii. Representar adequadamente suas relações de vizinhança como uma nova linha na matriz Laplaciana.

iii. Resolver novamente a Equação (3.17) por mínimos quadrados.

No entanto, neste trabalho, a LSP não é considerada incremental, pois a solução da Equação (3.17) pode resultar em alguma perturbação, ainda que pequena, no *layout* da projeção inicial, já que esta operação implica recomputar o mapeamento como um todo.

LSP preserva muito bem relações de vizinhança e tem como aplicação principal o mapeamento e visualização de coleções de documentos.

3.1.12 Stochastic Neighbor Embedding (SNE) e t-Distributed SNE

A *t-Distributed Stochastic Neighbor Embedding* (**t-SNE**) (Maaten e Hinton, 2008) é uma técnica não linear, capaz de capturar muito da estrutura local dos dados de alta dimensão, enquanto também revela a sua estrutura global. É baseada na otimização de uma função-custo que usa probabilidade condicional para representar a similaridade entre instâncias. Sua formulação deriva da *Stochastic Neighbor Embedding* (**SNE**) (Hinton e Roweis, 2002), cuja função-custo é difícil de otimizar. Para amenizar este problema os autores da t-SNE implementaram duas modificações em relação à sua antecessora:

- Uma versão simetrizada da função-custo, com gradientes mais simples, introduzido por Cook et al. (2007).
- Distribuição *t-Student* ao invés da Gaussiana para computar a similaridade entre dois pontos no espaço de baixa dimensão.

Tais modificações são alvo da próxima discussão, mas não há como compreendê-las sem antes estudar sua antecessora – a SNE – ponto de partida desta análise. Para isto, considere o conjunto $X = \{x_1, x_2, \ldots, x_n\}$ de alta dimensão $n \times m$ e sua projeção $Y = \{y_1, y_2, \ldots, y_n\}$ de dimensão $n \times p$, com $p < m$. A similaridade de um ponto x_j para um ponto x_i é dada pela probabilidade condicional $p_{j|i}$, tal que x_j é vizinho de x_i, onde os vizinhos são encontrados pela distribuição de probabilidade sobre uma Gaussiana centrada em x_i. Para pontos próximos, $p_{j|i}$ é relativamente alta, portanto, a similaridade é alta. Para pontos distantes um do outro $p_{j|i}$ é quase infinitesimal, desde que sejam usados valores razoáveis para a variância σ_i, conforme será descrito mais adiante. Matematicamente, a probabilidade condicional é definida por:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2/2\sigma_i^2)} \,. \tag{3.18}$$

No espaço de baixa dimensão, por facilidade, faz-se $\sigma_i = \frac{1}{\sqrt{2}}$, já que neste caso, fixar outros valores só irá implicar reescalar o mapeamento. Logo, a probabilidade condicional neste espaço é dada por:

$$q_{j|i} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)} \,. \tag{3.19}$$

Como o interesse é modelar a similaridade entre pares, então $p_{i|i} = 0$ e $q_{i|i} = 0$. Note que a SNE (e a t-SNE também) pode ser aplicada sobre uma matriz de similaridades, ao invés de um conjunto de dados contendo a representação vetorial de cada objeto.

Seja P_i a distribuição de probabilidade condicional sobre todos os outros pontos, dado x_i, e Q_i a distribuição de probabilidade condicional sobre todos os outros pontos do mapeamento, dado y_i. O cálculo de σ_i no espaço de alta dimensão está associado ao número de vizinhos de cada ponto. Qualquer valor particular de σ_i induz uma distribuição de probabilidade P_i sobre todos os outros pontos. Esta distribuição tem uma entropia que aumenta quando σ_i aumenta. SNE obtém os σ_i por um coeficiente denominado perplexidade, definido como:

$$Perp(P_i) = 2^{H(P_i)}, \text{ tal que } \quad H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i} \ , \tag{3.20}$$

onde $H(P_i)$ é a entropia de Shannon de P_i. A perplexidade pode ser entendida como uma medida atenuada do número efetivo de vizinhos. Valores típicos ficam entre 5 e 50.

A medida natural na qual $q_{j|i}$ modela $p_{j|i}$ é a divergência de *Kullback-Leibler* (KL) (Kullback e Leibler, 1951). SNE minimiza a soma das divergências KL sobre todos os pontos do conjunto de dados, através da seguinte função-custo:

$$C = \sum_i \mathrm{KL}(P_i \| Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \,. \tag{3.21}$$

A minimização da Equação (3.21) é realizada pelo método do gradiente descendente, que resulta em:

$$\frac{\delta C}{\delta y_i} = 2 \sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j). \tag{3.22}$$

Fisicamente, o gradiente pode ser interpretado como a força-resultante das forças de atração e repulsão causadas por um conjunto de molas entre o ponto y_i e todos os outros pontos mapeados, y_j. A fim de aumentar a velocidade de convergência e evitar mínimos locais, um termo de *momento* é adicionado ao gradiente para cada iteração t, indicado por $\alpha(t)$. O momento usa a diferença entre as duas últimas iterações para determinar as mudanças nas coordenadas do mapeamento no passo atual. Este termo tende a ficar pequeno à medida que o mapeamento torna-se moderadamente bem organizado. Além disso, ele introduz uma pequena perturbação ao mapeamento a cada iteração, de modo a evitar ciclos periódicos na solução, ocasionados por mínimos locais.

A Equação (3.23) representa o gradiente atualizado com o termo de momento, na iteração t:

$$Y^{(t)} = Y^{(t-1)} + \eta \frac{\delta C}{\delta Y} + \alpha(t) \left(Y^{(t-1)} - Y^{(t-2)} \right) \,. \tag{3.23}$$

O termo η representa uma taxa de aprendizagem adaptativa, descrita por Jacobs (1988), a qual cresce gradualmente a fim de tornar o gradiente estável.

O que foi descrito até agora representa o funcionamento da técnica SNE. Em seguida serão abordados os acréscimos e melhorias da t-SNE sobre esta formulação.

A divergência KL não é simétrica, logo pode ocorrer diferenças entre pares de distâncias no espaço de baixa dimensão a cada iteração. Para contornar esta situação os autores da t-SNE resolveram minimizar a divergência KL sobre uma distribuição de probabilidade conjunta, p_{ij} e q_{ij}, tal que $p_{ij} = p_{ji}$, $q_{ij} = q_{ji}$ e, novamente, p_{ii} e q_{ii} valem zero. A probabilidade conjunta simetrizada p_{ij} no espaço de alta dimensão foi definida como:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \,. \tag{3.24}$$

Isto assegura que $\sum_j p_{ij} > \frac{1}{2n}$, para todos os pontos x_i que têm contribuição significante na função-custo.

O cálculo de similaridades no espaço reduzido também foi modificado pelo uso de uma distribuição *t-Student* ao invés da Gaussiana. Esta mudança é conveniente porque a distribuição *t-Student* se aproxima da distribuição Gaussiana (equivalente a uma mistura de infinitas Gaussianas com diferentes variâncias) e porque o cálculo não envolve exponenciais, além de outras boas propriedades que esta distribuição apresenta, como ser (quase) invariante a mudanças na escala do mapeamento para pontos bem afastados (ver Maaten e Hinton (2008)). Portanto, o cálculo da probabilidade conjunta simetrizada (similaridade) no espaço de baixa dimensão passa a ser determinado por:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}}\,. \tag{3.25}$$

Finalmente, o gradiente da divergência KL pode ser calculado como:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}. \tag{3.26}$$

Para a demonstração da Equação (3.26) veja Maaten e Hinton (2008). Para visualizar cada etapa da t-SNE de forma sintetizada veja o Algoritmo 3.6 a seguir.

Algoritmo 3.6 *t-Distributed Stochastic Neighbor Embedding* (t-SNE)

Entrada: Conjunto de dados $X = \{x_1, x_2, \ldots, x_n\}$. Parâmetros da função-custo: perplexidade $Perp$. Parâmetros de otimização: número de iterações c, taxa de aprendizagem η, momento $\alpha(t)$.
Saída: Representação dos dados em baixa dimensão $Y^{(c)} = \{y_1, y_2, \ldots, y_n\}$.

1: Computar todos os pares de similaridade $p_{j|i}$, com perplexidade $Perp$ (Equação (3.18))
2: $p_{ij} \leftarrow \frac{p_{j|i} + p_{i|j}}{2n}$
3: Solução inicial $Y^{(0)} = \{y_1, y_2, \ldots, y_n\}$ a partir de $\mathcal{N}(0, 10^{-4}I)$
4: **para** $t = 1$ **até** c **faça**
5: Computar as similaridades em baixa dimensão q_{ij} (Equação (3.25))
6: Computar o gradiente $\frac{\delta C}{\delta Y}$ (Equação (3.26))
7: $Y^{(t)} \leftarrow Y^{(t-1)} + \eta \frac{\delta C}{\delta Y} + \alpha(t)\left(Y^{(t-1)} - Y^{(t-2)}\right)$
8: **fim para**

t-SNE não é uma técnica capaz de acomodar novas entradas no mapeamento de modo incremental, tampouco é flexível para uso com grandes conjuntos de dados, pelo tempo de CPU que requer devido à sua alta complexidade: $O(n^2)$.

Apesar da limitação do custo computacional, a t-SNE destaca-se por projetar dados com elevada preservação de vizinhança. Recentemente, uma abordagem mais eficiente foi

desenvolvida pelo próprio autor (Maaten, 2014). Estas modificações têm motivado seu uso como ferramenta de visualização de informação interativa (Bruneau et al., 2015; Kim et al., 2015) (ver Seção 3.2).

3.1.13 Local Convex Hull (LoCH)

Local Convex Hull (Fadel et al., 2015) é uma técnica de projeção local, criada especialmente para trabalhar com espaços esparsos de alta dimensão.

Esta abordagem é constituída de três passos principais, conforme ilustrado na Figura 3.9. O primeiro passo consiste em encontrar os k-vizinhos mais próximos de cada instância. Para realizar esta tarefa os autores empregaram uma estratégia baseada em agrupamentos, tal que o conjunto de dados é particionado em $\sqrt{n}$ grupos balanceados usando *bisecting K-means* (Steinbach et al., 2000), em seguida encontrados seus medoides. Então os c grupos vizinhos mais próximos de cada grupo são computados. Desse modo, a busca pelos vizinhos mais próximos de uma dada instância leva em conta somente as instâncias no mesmo grupo a que pertence e nos c grupos mais próximos, representando uma boa aproximação dos verdadeiros k-vizinhos mais próximos, com a diferença de que esta operação tem complexidade $O(n\sqrt{n})$ para $\sqrt{n}$ grupos, enquanto que as abordagens convencionais apresentam complexidade quadrática $O(n^2)$.

Figura 3.9: Passos principais da técnica LoCH (Retirado de Fadel et al. (2015)).

O segundo passo, seleção e projeção de amostras representativas, utiliza os medoides dos agrupamentos encontrados no passo anterior como amostras representativas, visando representar melhor os dados. Portanto, o número de amostras representativas empregado no processo é $\sqrt{n}$. Depois que as amostras são selecionadas, elas são projetadas para o novo espaço usando uma técnica global que preserva tanto quanto possível os relacionamentos de distância, denominada *Force Scheme* (Tejada et al., 2003). O posicionamento das amostras representativas no espaço de projeção pelo *Force* tende a preservar relacionamentos globais de distância, ao passo que os relacionamentos locais são estabelecidos por um processo iterativo, descrito a seguir.

A aproximação pelo fecho convexo (terceiro passo), apoia-se na seguinte ideia: relacionamentos locais de distância são preservados se cada ponto no espaço transformado é posicionado próximo ao fecho convexo de seus vizinhos mais próximos, assim, cada ponto projetado estará igualmente próximo das instâncias mais similares a ele.

Assumindo que x_i é um ponto no espaço de alta dimensão e y_i a sua projeção no espaço de menor dimensão p, então uma posição qualquer dentro do fecho convexo no

espaço p-dimensional, em função dos vizinhos de x_i, indicado por N_i, pode ser obtida como:

$$\hat{y_i} = \sum_{x_j \in N_i} \alpha_j y_j, \tag{3.27}$$

com $\alpha_j \geq 0$ e $\sum_j \alpha_j = 1$. Com o interesse de encontrar a posição no espaço transformado que melhor preserva os relacionamentos de distância entre x_i e seus vizinhos N_i, os autores calculam α_j com base nas distâncias a partir do espaço de origem.

Partindo de uma certa posição inicial, move-se cada ponto y_i em direção a $\hat{y}_i$. Porém, fazer y_i igual a $\hat{y}_i$, não garante que todos os pontos serão posicionados dentro do fecho convexo de seus vizinhos mais próximos. A ideia então, é mover y_i com base no vetor diretor $\vec{v}$ que vai da posição atual do ponto até $\hat{y}_i$, obtendo assim uma nova posição $\tilde{y}_i$, isto é:

$$\tilde{y_i} = \hat{y_i} + \gamma_i \frac{\vec{v}}{\|\vec{v}\|}, \tag{3.28}$$

tal que γ_i é o valor que melhor preserva as distâncias entre x_i e as instâncias em N_i. Por simplicidade, o cálculo dos coeficientes α_j e γ_i presentes nas Equações (3.27) e (3.28), respectivamente, foram omitidos. Ver Fadel et al. (2015) para detalhes de como obtê-los.

Em seguida, move-se iterativamente cada ponto y_i em direção a $\tilde{y}_i$ até que nenhum movimento seja possível ou o número máximo de iterações seja atingido. Sendo y_i^t a posição de x_i na t-ésima iteração, e lembrando que a posição das instâncias mudam a cada iteração, uma nova atualização pode ser calculada como:

$$y_i^t = [1 - \omega_i(t)]y_i^{t-1} + \omega_i(t)\tilde{y}_i, \tag{3.29}$$

onde $\omega_i(t)$ varia entre $[0; 1]$ e representa a liberdade de movimento que y_i apresenta na iteração t, tal que maiores valores permitem movimentar mais y_i. Este parâmetro é definido pelo usuário e usado para diminuir a quantidade de energia a cada iteração, ajudando a técnica a estabilizar.

O Algoritmo 3.7 descreve os passos necessários para projetar um conjunto de dados de alta dimensão X, usando a LoCH. A inicialização de Y, indicada na Linha 1, pode ocorrer de dois modos: 1) aleatoriamente, ou 2) encontrando valores iniciais que aceleram a convergência. No segundo caso, a Equação (3.27) é usada para posicionar cada ponto x_i considerando somente as amostras representativas que compõem a lista de seus vizinhos mais próximos N_i. Fadel et al. (2015) propõem um segundo algoritmo para realizar esta operação e inicializar Y.

Os autores provaram, experimentalmente, que o número máximo de iterações para uma boa aproximação pelo fecho convexo é em torno de $\sqrt{n}$, normalmente tornando-se estável antes desse valor.

Algoritmo 3.7 *Local Convex Hull* (LoCH)

Entrada: Conjunto de dados $X = \{x_1, x_2, \ldots, x_n\}$ e número de vizinhos k.
Saída: Conjunto de dados Y.

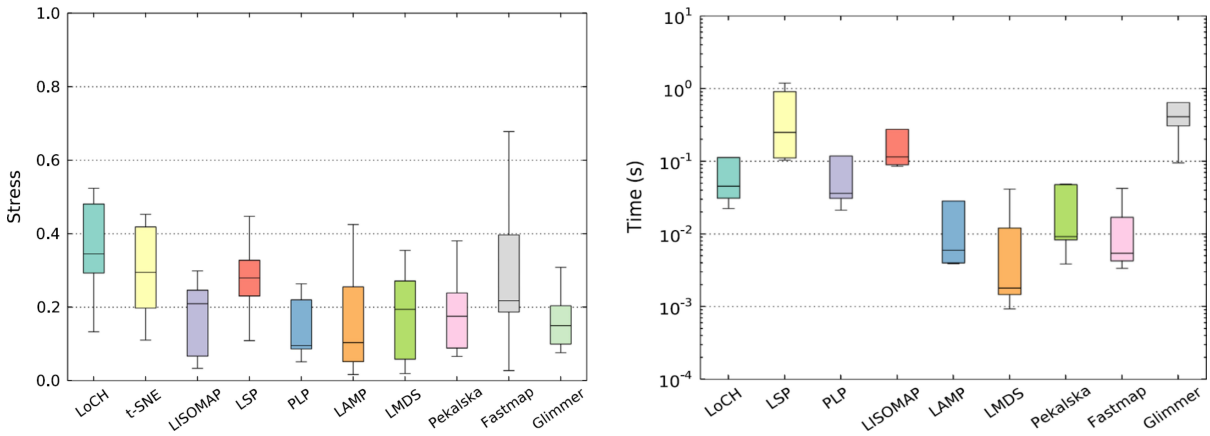
1: $Y \leftarrow$ INICIALIZA(X)
2: $\hat{Y} = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_n}\}$ // Equação (3.27)
3: **para todo** $x_i \in X$ **faça**
4: $N_i \leftarrow$ VIZINHOSMAISPROXIMOS(x_i, k)
5: **fim para**
6: $t \leftarrow 1$
7: **repita**
8: **para todo** $y_i \in Y$ **faça**
9: $\tilde{y_i} = \hat{y_i} + \gamma_i \frac{\vec{V}}{\|\vec{V}\|}$ // Equação (3.28)
10: **fim para**
11: **para todo** $y_i \in Y$ **faça**
12: $y_i = [1 - \omega_i(t)]y_i + \omega_i(t)\tilde{y_i}$ // Equação (3.29)
13: **fim para**
14: $t \leftarrow t + 1$
15: **até** $t \geqslant \sqrt{n}$ **ou** nenhum movimento possível
16: **retorna** Y

LoCH é uma técnica de projeção predominantemente local, não linear, restrita a espaços esparsos de alta dimensão. Tem complexidade $O(n\sqrt{n})$, pode ser utilizada em aplicações interativas, sem se destacar neste sentido, já que requer $\sqrt{n}$ amostras representativas, assim como outras técnicas anteriores a ela. Mapear uma nova instância isoladamente, após a projeção inicial do conjunto de dados, implica refazer o cálculo dos k-vizinhos mais próximos de cada instância (primeiro passo), o qual pode produzir diferentes *layouts*, dificultando seu uso de modo incremental.

As Figuras 3.10(a) e 3.10(b) foram extraídas da LoCH. Nelas, os autores enfatizam que técnicas globais apresentam melhores resultados em termos de preservação de distância (*stress*) e tempo computacional, quando comparadas a técnicas locais, justificando o fato da LoCH não ser tão eficaz neste sentido. Contudo, merece destaque neste experimento, o desempenho apresentado pela LAMP (Capítulo 4), mostrando ser uma das mais competitivas em termos de *stress* (Figura 3.10(a)) e eficiência computacional (Figura 3.10(b)).

Experimentos como o da Figura 3.10, apresentado por Fadel et al. (2015), comprovam que a LAMP (Joia et al., 2011), desenvolvida no início deste projeto de doutorado, ainda se mantém como uma das técnicas do estado da arte com respeito à preservação de distâncias, além de apresentar baixo custo computacional.

As próximas seções revisam técnicas relacionadas à identificação de agrupamentos e uso de diferentes medidas de similaridade.



(a) *Stress* (b) Tempo computacional (em segundos)

Figura 3.10: Comparação do *stress* e tempo computacional da LoCH contra outras técnicas de projeção (Retirado de Fadel et al. (2015)). Note que a LAMP destaca-se como uma das técnicas mais precisas e eficientes da atualidade.

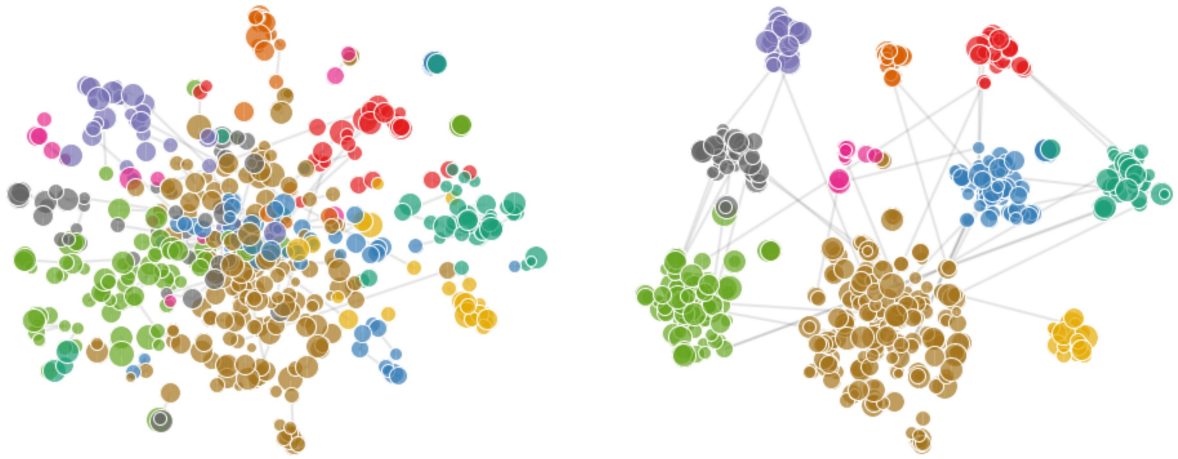
3.2 Técnicas para Identificação e Visualização de Agrupamentos

Esta seção discute as principais técnicas utilizadas na identificação e visualização de agrupamentos de dados multidimensionais, com base em projeção. A seleção foi feita a partir das técnicas de projeção que representam o estado da arte em preservação de vizinhança (t-SNE e LSP), preservação de distâncias (baseadas em MDS e LAMP) e pela qualidade dos resultados apresentados. Portanto, as seguintes técnicas são revisadas nesta seção:

- Baseadas na t-SNE: **UTOPIAN** (Choo et al., 2013), ***Cluster Sculptor*** (Bruneau et al., 2015) e **DS t-SNE** (Kim et al., 2015).
- Baseadas na LSP: ***ProjCloud*** (Paulovich et al., 2012) e ***ProjSnippet*** (Gomez-Nieto et al., 2014).
- Baseadas em MDS: ***GMap*** (Gansner et al., 2010), ***TwitterScope*** (Gansner et al., 2013), Wu et al. (2011) e Steiger et al. (2014).
- Baseadas na LAMP: Mamani et al. (2013).
- Outros tipos: **IRP-Kmeans** (Cardoso e Wichert, 2012), Kiyadeh et al. (2015) e ***ReCloud*** (Wang et al., 2014).

Dentre as técnicas que se apoiam na t-SNE (Seção 3.1.12), a *User-driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization* (**UTOPIAN**) (Choo et al., 2013), emprega fatoração de matrizes não negativas para extrair e agrupar tópicos a partir

de coleções de documentos. Sua formulação semissupervisionada permite que os usuários controlem a importância de palavras-chave associadas aos tópicos. Os tópicos são agrupados a partir de sua representação matricial e os agrupamentos resultantes são mapeados para o espaço visual utilizando a t-SNE modificada pelo acréscimo de um parâmetro de encolhimento, desta forma, a distância entre documentos que pertençam ao mesmo grupo de tópicos é diminuída. Tal modificação permite representar cada grupo de forma mais compacta, resultando em uma visualização clara da estrutura dos agrupamentos, como pode ser observado na Figura 3.11.



(a) t-SNE (b) t-SNE modificada

Figura 3.11: Comparação entre a t-SNE original e modificada (Retirado de Choo et al. (2013)).

Outro trabalho que utiliza a técnica t-SNE é o ***Cluster Sculptor*** (Bruneau et al., 2015), um sistema interativo que visa apoiar a análise de agrupamentos de forma visual e semiautomática. O *Cluster Sculptor* atua de forma interativa (permite reorganizar os grupos no espaço visual) e iterativa (o usuário pode inserir seu conhecimento progressivamente). O sistema é alimentado por agrupamentos calculados a partir do espaço de alta dimensão (usando *K-means* (Jain, 2010) ou *Spectral Clustering* (Ng et al., 2002)) e vinculados à projeção bidimensional realizada pela t-SNE. A seguir, o usuário pode atualizar os agrupamentos rotulados e associá-los à projeção, usando ferramentas interativas. Sua complexidade computacional é $O(n^2)$ para cada iteração, onde n é o número de instâncias. Os autores testaram o sistema em três diferentes cenários com conjuntos de dados reais, onde o usuário pode combinar diversas características do sistema para inserir progressivamente seu conhecimento, de modo a obter melhores projeções e agrupamentos. O *Cluster Sculptor* apresenta algumas limitações, tais como: conjuntos de dados com muitas instâncias precisam ser amostrados; não processa todos os tipos de dados como, por exemplo, os categóricos; no contexto de fluxos de dados (*data streams*) não é capaz de incluir novas entradas.

Kim et al. (2015) propuseram uma abordagem de redução de dimensionalidade supervisionada chamada *Doubly Supervised t-SNE* (**DS t-SNE**) que além de preservar o relacionamento original dos dados, mantém a separabilidade entre classes. A proposta incorpora o conceito de "agrupamentos intrínsecos", os quais representam agrupamentos naturais inerentes aos dados originais na alta dimensão. A ideia por trás da DS t-SNE é estender a t-SNE utilizando simultaneamente dados rotulados e agrupamentos intrínsecos. Esta abordagem favorece a análise visual dos dados, refletindo o agrupamento natural dos mesmos. A DS t-SNE requer três passos adicionais antes de aplicar o passo de redução de dimensionalidade: 1) determinação dos agrupamentos intrínsecos por meio do *K-means*, 2) supervisão adaptativa usando os dados rotulados e 3) supervisão secundária usando os agrupamentos intrínsecos. Os dois últimos passos são calculados de forma similar, aumentando os valores de distribuição de probabilidade correspondentes às relações entre pares de instâncias dentro de cada agrupamento. Requer dois parâmetros para controlar a separabilidade entre os grupos. A complexidade computacional da DS t-SNE é $O(n^2p)$ onde n é o número de instâncias e p é a dimensão do espaço reduzido. Os autores demonstraram a vantagem da DS t-SNE em comparação a cinco técnicas de redução de dimensionalidade, dentre as quais, duas técnicas são baseadas na t-SNE com modificações realizadas pelos autores, referente às distâncias entre os dados. Os experimentos utilizaram medidas quantitativas para avaliar a preservação das relações originais dos dados (classificação e vizinhança) e análise visual aplicada a conjuntos de documentos (texto).

Para análise visual de coleções de documentos, Paulovich et al. (2012) propuseram uma técnica de visualização que combina nuvens de palavras com projeção multidimensional, denominada ***ProjCloud***. Tal abordagem permite visualizar a relação de vizinhança (ou similaridade) entre documentos relacionados e suas correspondentes nuvens de palavras. *ProjCloud* inicia a partir do mapeamento de uma coleção de documentos para o espaço visual, utilizando a técnica de projeção *Least Square Projection* (LSP) (Paulovich et al., 2008). A seguir, pontos no espaço visual são agrupados usando o *bisecting K-means* (Steinbach et al., 2000) e o fecho convexo de cada grupo é calculado para obter os polígonos, os quais irão conter as nuvens de palavras. Finalmente, classificação espectral é empregada para arranjar as palavras de acordo com sua relação semântica, bem como para destacar as palavras mais importantes na nuvem. Através de experimentos os autores mostraram que grupos distintos de documentos são facilmente identificados e que os principais tópicos que descrevem o conteúdo de cada um deles são claramente destacados. No entanto, algumas limitações podem ser apontadas, principalmente com relação à geração dos polígonos, como por exemplo: a possibilidade de sobreposições, agrupamentos pequenos onde as palavras dificilmente são identificadas e grandes espaços vazios entre os agrupamentos. A Figura 3.12 ilustra a visualização de coleções de documentos com *ProjCloud*.

(a) Quatro grupos (b) Nove grupos

Figura 3.12: Exemplos de visualização com *ProjCloud*: coleção de documentos gerados a partir de uma coleção de artigos científicos, em quatro diferentes áreas do conhecimento (Retirado de Paulovich et al. (2012)).

Gomez-Nieto et al. (2014) propuseram uma abordagem para visualizar *snippets* textuais recuperados a partir de mecanismos de busca na web. O ***ProjSnippet*** inicia com o pré-processamento dos resultados da busca textual por meio da extração de frequência de termos. Em seguida, os vetores obtidos são mapeados para o espaço visual usando a LSP. O conteúdo de cada *snippet* é embutido em um retângulo e o *K-means++* (Arthur e Vassilvitskii, 2007) é então aplicado para agrupar *snippets* similares no espaço visual. A fim de melhorar o *layout*, cores são utilizadas para destacar os retângulos que pertencem ao mesmo agrupamento, e um mecanismo de *seam carving* (Avidan e Shamir, 2007) é empregado para reduzir espaços vazios entre os retângulos. O passo final conta com um mecanisno de remoção de sobreposição através de um funcional de energia que fornece o arranjo das entidades geométricas no espaço visual, preservando as relações de vizinhança com sobreposição mínima. A Figura 3.13 ilustra cada etapa da técnica.

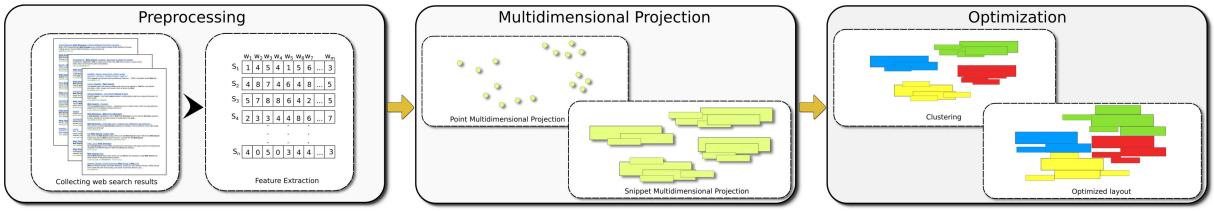


Figura 3.13: Passos principais da técnica *ProjSnippet* (Retirado de Gomez-Nieto et al. (2014)).

Gansner et al. (2010) desenvolveram o ***GMap***, um *framework* para visualização de dados relacionais semelhante a mapas geográficos. A abordagem mantém a estrutura

e as relações inerentes dos dados, além de fornecer uma metáfora visual familiar para compreensão da relação entre os dados e seus agrupamentos. *GMap* toma como entrada um grafo ou um conjunto de dados multidimensionais, o qual é projetado no plano por uma técnica MDS ou equivalente, em seguida aplica o *K-means* para obter os agrupamentos e criar os mapas. O *framework* foi projetado para visualizar relações de grupos como mapas, onde cada item pertence a um grupo, ou seja, um país. Mas também, pode ser adaptado para visualizar múltiplas relações entre um conjunto de objetos. Os autores aplicaram o *GMap* em alguns conjuntos de dados comuns na web, como: compra de livros, coleções de música e dados de comércio internacional. A Figura 3.14 mostra um exemplo de visualização com *GMap*.

Figura 3.14: Exemplo de visualização com *GMap*: mapa de livros relacionados ao ano de 1984, a partir do *Amazon.com* (Retirado de Gansner et al. (2010)).

Aproveitando a metáfora visual fornecida pelo *GMap*, Gansner et al. (2013) propuseram a aplicação denominada ***TwitterScope***, para visualizar, em tempo real, fluxos de texto gerados a partir da rede social *Twitter*. A aplicação proposta combina análise semântica, MDS, remoção de sobreposição, agrupamento com base em modularidade e *GMap* para visualizar *tweets* similares (postagens no *Twitter*) e seus conteúdos resumidos.

Métodos baseados em conteúdo dependem da projeção multidimensional para gerar *layouts* que destacam grupos de instâncias similares, enquanto permite visualizar um resumo do conteúdo de cada agrupamento. Um exemplo é o método proposto por Wu et al. (2011), o qual agrupa **palavras-chave**, posicionando-as no espaço visual pela combinação de MDS, remoção de sobreposição, *K-means* e um mecanismo *seam carving* (Avidan e Shamir, 2007).

Steiger et al. (2014) propuseram um sistema que se apoia em MDS, *K-means* e discretização do espaço visual para agrupar e visualizar **séries temporais** de acordo com suas similaridades. Tal abordagem auxilia na exploração e comparação de dados de sensores georreferenciados e outros diferentes padrões temporais para descobrir efeitos sazonais, anomalias e periodicidades. Os métodos usados cobrem a detecção de padrões: 1) diários com visualização a partir de agrupamentos, 2) semanais com visualização baseada em calendário e 3) sazonais com base em projeção. A Figura 3.15 apresenta um exemplo da abordagem proposta.

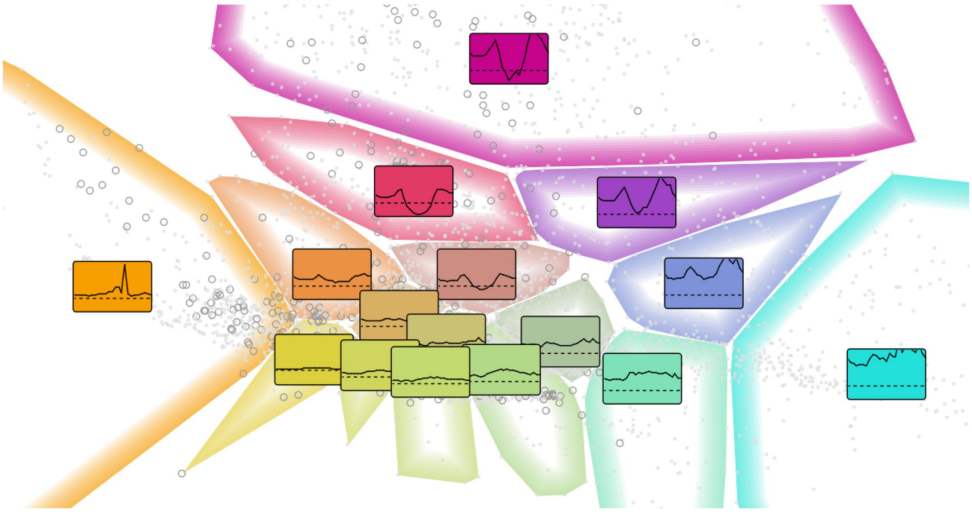


Figura 3.15: Exemplo de visualização a partir do trabalho de Steiger e colaboradores, em que os dados são projetados, agrupados e coloridos por um mapa de cores. Cada grupo é anotado com um elemento representativo, neste caso, mostrando o consumo de energia durante as horas do dia (Retirado de Steiger et al. (2014)).

Mamani et al. (2013) propuseram uma abordagem de visualização assistida para transformação do **espaço de características**, com base na manipulação de amostras representativas pelo usuário. Além de um *layout* visual simples e intuitivo, o usuário pode observar como as estruturas de vizinhança mudam durante a interação, ou seja, o usuário pode saber quais instâncias estão sendo afetadas pela transformação do espaço. A abordagem proposta combina projeção multidimensional e mapeamentos lineares ortogonais para permitir manipulação interativa do espaço de características. Os mapeamentos ortogonais são matematicamente formulados com base na LAMP (Capítulo 4), adaptada para mapear instâncias de/para o mesmo espaço de características. A metodologia foi aplicada em recuperação de imagens por conteúdo.

Cardoso e Wichert (2012) introduziram um método para agrupamento de dados de alta dimensão chamado *Iterative Random Projections K-means* (**IRP-Kmeans**), o qual emprega *Random Projection* (RP) (Bingham e Mannila, 2001) e o algoritmo *K-means*. A ideia é aplicar o *K-means* sucessivamente, aumentando gradualmente a dimensionalidade

dos dados após cada convergência do *K-means*. Os agrupamentos obtidos em uma dada dimensão são utilizados para inicializar os agrupamentos da dimensão seguinte. Tal proposta permite construir uma solução com mais detalhes de informações, já que os dados são conduzidos para um espaço de maior dimensão, além de evitar possíveis mínimos locais. Os autores compararam o IRP-Kmeans contra dois métodos similares, o *K-means* e o *K-means* com simples *Random Projection*. Nos experimentos foram utilizados quatro conjuntos de dados: um de imagem, um de texto e dois sintéticos. Os resultados mostraram que é possível obter uma taxa de erro médio menor em relação às outras técnicas. No entanto, os testes realizados são superficiais, já que as técnicas comparadas são as mesmas empregadas na construção do próprio IRP-Kmeans.

Kiyadeh et al. (2015) propuseram um método de visualização semissupervisionada para dados de alta dimensão, requerendo apenas uma fração de dados rotulados. O objetivo é melhorar a visualização e identificação de agrupamentos nos dados. Os dados rotulados são utilizados para encontrar a melhor visualização bidimensional que minimiza as distâncias entre os objetos do mesmo grupo, ao passo que maximiza as distâncias entre grupos. Esta proposta estende a capacidade do método de visualização *Star Coordinates* (**SC**) (Kandogan, 2001) para trabalhar com conjuntos de dados de alta dimensão, especialmente quando a dimensão é maior que 50. Além disso, se concentra no problema de ajuste automático dos eixos a fim de encontrar os melhores mapeamentos para a visualização dos dados. No SC as propriedades globais e relações de agrupamento são preservadas no espaço mapeado, no entanto, isto não evita a sobreposição dos agrupamentos. Para resolver este problema, duas transformações foram introduzidas: escala (ajuste do tamanho dos eixos) e rotação (ajuste da direção dos eixos). A abordagem proposta é simples de implementar e tem complexidade computacional polinomial igual a $O(m^3)$ onde m é dimensão do espaço de origem. Os autores utilizaram quatro conjuntos de dados e fizeram comparações com SC original, PCA e LLE para comprovar a facilidade de identificação visual dos agrupamentos obtidos.

ReCloud (Wang et al., 2014) permite a visualização de comentários de usuários de alguns *websites*. Os comentários são processados utilizando uma técnica de processamento de linguagem natural chamada análise de dependência gramatical (Marneffe et al., 2006), a qual extrai um grafo semântico de conteúdos a partir dos comentários originais. Um modelo de energia, para otimizar o algoritmo baseado em força (Noack, 2009), é aplicado sobre o grafo semântico a fim de criar agrupamentos de palavras-chave, definir cores, tamanho da fonte e suas posições iniciais no *layout*. A espiral de *Arquimedes* (Steele e Iliinsky, 2010; Whitrow, 2008) é utilizada com o propósito de evitar sobreposições de palavras-chave no espaço visual. O *layout* semântico fornecido pelo *ReCloud* também permite a interação do usuário ao recuperar as informações associadas às palavras-chave. A Figura 3.16 exibe a visualização dos agrupamentos semânticos de palavras-chave.

Figura 3.16: Exemplo de visualização com *ReCloud*: informações sobre uma loja de sobremesas próxima ao campus da Universidade da Califórnia, em Berkeley (Retirado de Wang et al. (2014)).

3.3 Técnicas que Usam Diferentes Medidas de Similaridade

Esta seção discute um conjunto de técnicas que operam, essencialmente, modificando medidas de similaridade, com o propósito de aumentar a precisão dos resultados obtidos em tarefas relacionadas à *redução de dimensionalidade*: Lee et al. (2015); *recuperação de imagens com base em conteúdo*: Arevalillo-Herráez et al. (2008), Aboulmagd et al. (2009), Pedronette e Torres (2013), Okada et al. (2015); *classificação de dados*: Kobayashi (2014), Tan et al. (2015); e *reconhecimento de padrões em imagens*: Liu (2014), Baccour et al. (2014).

Métodos da família *Stochastic Neighbor Embedding* (SNE) (Seção 3.1.12), consideram uma vizinhança relativamente pequena comparada ao número de dados disponíveis, como consequência, as informações sobre a estrutura global dos dados é perdida, fazendo com que a SNE e suas variantes fiquem propensas aos mínimos locais. Com o objetivo de capturar informações de vizinhança de tamanho variável, Lee et al. (2015) propuseram o uso de similaridades multiescala em substituição à similaridade de escala única utilizada nos métodos como SNE. Os autores chamaram tais similaridades de *Multi-Scale* (**MS**) e combinaram com as funções-custo dos métodos SNE, *Neighbourhood Retrieval and Vi-*

sualisation (NeRV) (Venna et al., 2010) e *Jensen–Shannon Embedding* (JSE) (Lee et al., 2013) para propor novos métodos de redução de dimensionalidade, chamados MS SNE, MS NeRV e MS JSE. A MS envolve as médias das várias vizinhanças Gaussianas com larguras de banda em crescimento exponencial. Seu objetivo é maximizar a qualidade da projeção em todas as escalas, com a melhor preservação de vizinhança possível, tanto local quanto global, e também isentar o usuário de ter que fixar o tamanho da vizinhança por meio do coeficiente de perplexidade. A complexidade computacional da MS para c iterações aumenta em $O(c \log_2 n)$ a complexidade dos métodos que a utilizam, assim sendo, a MS SNE tem complexidade de $O(n^2 c \log_2 n)$. Este aumento da complexidade pode dificultar sua aplicação em grandes conjuntos de dados. Experimentos realizados com diversas técnicas de complexidade computacional similar, demonstraram que a aproximação multiescala utilizada capta melhor a estrutura dos dados e melhora significativamente a qualidade da redução de dimensionalidade.

Arevalillo-Herráez et al. (2008) propuseram uma técnica que permite combinar um conjunto de funções de distância para produzir uma **medida de similaridade composta**, a qual é avaliada no contexto de *Content-Based Image Retrieval* (CBIR). A técnica faz uso de subconjuntos de características associados ao cálculo de probabilidade para compor a nova medida de similaridade. Os autores conduziram experimentos mostrando melhores resultados em comparação às medidas convencionais de distância.

Aboulmagd et al. (2009) empregaram **conceitos *fuzzy*** em uma abordagem envolvendo CBIR, visando reduzir a lacuna entre similaridade quantitativa obtida pelo sistema e avaliação qualitativa fornecida pelo usuário para calcular a relevância das consultas. Nesta proposta, a imagem é representada por um *Fuzzy Attributed Relational Graph* (FARG) (Chan e Cheung, 1992; Shapiro e Haralick, 1985) estendido para incluir um novo esquema de representação de cor com base em conceitos *fuzzy* e atributos de textura que são computados de forma a modelar o sistema de visão humano com a finalidade de descrever objetos na imagem. Desse modo, foi apresentado um algoritmo de correspondência de grafos que tenta simular o processo de pensamento humano ao comparar imagens. O algoritmo computa a similaridade entre objetos inspecionando diversos atributos, como rótulos, tamanho, textura, cor e localização; assim a similaridade é modelada nos atributos, o que dá flexibilidade ao usuário em ponderar a importância de cada atributo de acordo com seu interesse. A representação destes atributos utiliza conjuntos e conceitos de lógica *fuzzy* para expressar de modo adequado o conteúdo das imagens.

Pedronette e Torres (2013) apresentaram um algoritmo de re-*ranking*, o *Ranked Lists Similarities* (**RL-Sim**), um método de pós-processamento que considera uma medida diferenciada de distância entre imagens, baseada na similaridade entre *rankings* (listas de imagens recuperadas, ordenadas conforme suas similaridades à imagem de consulta) para recuperação de imagens por conteúdo. É uma abordagem iterativa com base em aprendiza-

gem não supervisionada, capaz de incorporar informação contextual a partir dos *rankings*. O algoritmo RL-Sim computa a distância entre duas imagens img_i e img_j analisando a similaridade entre seus respectivos *rankings*, τ_i e τ_j, considerando as k primeiras posições em cada lista. Assim, as distâncias são redefinidas considerando as medidas de correlação de *ranking* $d(\tau_i, \tau_j, k)$. A ideia, então, é mover imagens não similares para baixo na lista, com o intuito de melhorar os resultados das consultas. Esta abordagem não requer intervenção do usuário, mas pode ser combinada com outras técnicas que levam em conta as preferências do usuário, tais como abordagens de *Relevance Feedback* (RF).

Uma versão estendida do algoritmo RL-Sim foi proposta por Okada et al. (2015), o **RL-Sim***, um método de pós-processamento que visa computar uma distância diferenciada para os casos em que não ocorre sobreposição entre os *rankings* que estão sendo comparados. É baseado em medidas de correlação de *ranking* e informações de sobreposição entre conjuntos de vizinhança de tamanho k. O algoritmo proposto divide o *ranking* em três segmentos, tal que cada segmento define um subconjunto que é processado de modo distinto, sendo L a posição até a qual os *rankings* devem ser considerados: o primeiro segmento considera as L posições do topo com sobreposição e computa uma nova distância através da medida de correlação de *rankings* (Pedronette e Torres, 2013); para o segundo segmento, as L posições do topo sem sobreposição são consideradas e a distância atual é incrementada de um; e o terceiro segmento considera as imagens restantes que estão abaixo das L posições do topo e incrementa suas distâncias de dois, assegurando que estas imagens ficarão no final dos *rankings*. Os autores também apresentaram uma análise geral sobre algumas medidas tradicionais de correlação de *ranking*, no contexto de recuperação de imagens, e propuseram duas novas medidas: Jaccard$_l$ e Kendall$_{\tau w}$. A primeira, calcula um escore acumulado considerando diferentes profundidades definidas por k, já que o coeficiente de Jaccard tradicional ignora informações fornecidas por posições de topo menores que k. A segunda medida é semelhante à medida original Kendall$_\tau$, exceto pela função que computa os pesos de cada par de imagens, onde um fator penaliza pares discordantes que estão distantes nos *rankings*. Os pares são considerados distantes quando a diferença entre suas posições é maior que k.

O trabalho de Kobayashi (2014) utiliza medidas de similaridade para classificação semissupervisionada. Dois métodos são propostos a partir de probabilidades de transição baseadas em *kernel* (Bishop, 2006; Webb, 2002). O primeiro método, *Kernel-based Transition Probability* (**KTP**), utiliza uma única função *kernel* proveniente da comparação entre mínimos quadrados variacionais $\times$ baseados em *kernel*. O segundo método, combina os vários KTPs integrando-os em uma nova medida similaridade por meio de probabilidades representadas por pesos lineares. Experimentos conduzidos demonstraram que as similaridades propostas apresentam desempenho favorável em comparação com outros métodos de classificação semissupervisionada.

Image-to-class distance ratio (**I2CDR**) (Tan et al., 2015) é uma métrica para seleção de subconjuntos de características com base em distância Euclidiana. A métrica foi projetada para maximizar a distância interclasses (medidas de distância entre imagens pertencentes à diferentes classes) e minimizar a distância intraclasse (medidas de distância entre imagens pertencentes à mesma classe), possibilitando uma boa classificação das instâncias (ver Figura 2.1). De um modo geral, a métrica pode ser definida como a razão da distância entre objetos da mesma classe (intraclasse) para a distância entre classes (interclasses). Para tarefas de classificação em grande escala, um algoritmo baseado em *Particle Swarm Optimization* (PSO) (Clerc, 2006) e I2CDR foi proposto, denominado I2CDRPSO (= I2CDR (métrica proposta) + PSO), o qual opera em grandes espaços de busca com baixo custo computacional e taxa de convergência rápida. A complexidade computacional do I2CDR, medida por classe, equilave a $O(n_i \log n_i)$, onde n_i é o número de imagens da classe i. Para o algoritmo I2CDRPSO, a complexidade para c iterações é $O(cn_i \log n_i)$. Os autores realizaram experimentos para mostrar que o algoritmo supera alguns métodos de *ranking* de características (ordem de relevância) e métodos de seleção de subconjunto de características, comumente utilizados em classificação. Também foram realizados testes de comparação de características globais (para todo o conjunto de dados) e locais (considerando cada classe) utilizando classificadores bem conhecidos.

Liu (2014) analisou como a **similaridade do cosseno**, frequentemente aplicada em métodos de extração de características baseados em análise discriminante, melhora os resultados obtidos em reconhecimento de padrões. Tal melhoria provém de sua ligação com a regra de decisão de *Bayes* (Liu, 2008), ótima para a minimização de erros de classificação. Além disso, discute problemas inerentes à medida de similaridade do cosseno que reduzem seu poder de discriminação, conduzindo a classificações incorretas. Tais problemas estão relacionados à medida de distância e medida angular. A inadequação da medida de distância surge porque a medida de similaridade do cosseno falha na obtenção da distância real entre dois vetores. Já o problema relacionado à medida angular ocorre quando o ângulo entre os vetores é maior que $\pi/2$. Ambos os problemas levam à classificações incorretas quando a medida de similaridade do cosseno é usada. Para superar tais problemas, uma nova medida de similaridade foi apresentada (similaridade do cosseno modificada). Esta nova medida é avaliada em problemas de reconhecimento facial obtendo resultados superiores à outras medidas de similaridade, tais como a medida de similaridade do cosseno convencional, correlação normalizada (Struc e Pavesic, 2008) e a medida de distância Euclidiana.

Baccour et al. (2014) avaliaram propriedades de similaridade e medidas de distância *fuzzy* em diferentes aplicações de processamento de imagem com o propósito de conhecer a influência de tais propriedades sobre os resultados. As medidas de distância abordadas foram a medida de distância *fuzzy* entre dois conjuntos e a medida *crisp* apoiada nos

postulados de espaço métrico (Definição 2.11). Medidas de similaridade *fuzzy*, conhecidas como *Fuzzy Similarity Measures* (**FSM**), podem ser usadas para comparar diferentes tipos de objetos, como imagens, por exemplo. Suas definições são baseadas em medidas de proximidade, operações sobre conjuntos *fuzzy* (ver Seção 2.6), e outras. Neste trabalho, FSMs foram aplicadas em tarefas de classificação de formas e reconhecimento de padrões.

3.4 Considerações Finais

Neste capítulo foram revisados alguns trabalhos relacionadas ao tema proposto. Os trabalhos compreendem três categorias de técnicas: 1) técnicas de projeção de dados multidimensionais, 2) técnicas para identificação e visualização de agrupamentos e 3) técnicas que usam diferentes medidas de similaridade. Ver Figura 3.17 para um panorama geral das técnicas revisadas em cada categoria.

Técnicas de projeção constituem o foco principal desta tese, portanto, um estudo sistemático foi realizado envolvendo tais técnicas. Neste estudo foram explorados vários aspectos das mesmas, para melhor compreendê-las e destacar suas vantagens e limitações. Entre os aspectos explorados estão: ordem de complexidade do algoritmo; uso (ou não) de amostras representativas; interatividade; se admite que novas instâncias sejam mapeadas a posteriori, sem remapear ou recalcular as demais; se requer dados de entrada contidos em um espaço vetorial; tipo de transformação de dados; formulação matemática e natureza da projeção: local/global. Vale lembrar que parte substancial deste estudo serviu como base para o desenvolvimento das técnicas de projeção apresentadas nos próximos capítulos.

A capacidade de projetar grandes volumes de dados, diminuição do tempo de resposta, maior precisão e interatividade, confirmam os recentes avanços das técnicas de projeção. Porém, nenhuma das técnicas revisadas consegue projetar dados com eficácia, partindo de um número restrito de amostras representativas, de modo a facilitar a organização dos dados e identificação de agrupamentos como a LAMP, uma das abordagens desenvolvidas neste projeto de doutorado, discutida no Capítulo 4.

Levando em conta preservação de distâncias $\times$ eficiência computacional, LAMP pode ser considerada uma das técnicas do estado da arte atual. O recente trabalho de Fadel e colaboradores (Fadel et al., 2015), confirma este fato em um de seus experimentos (ver Figura 3.10 para comparação).

As outras categorias investigadas: técnicas para identificação e visualização de agrupamentos e técnicas que usam diferentes medidas de similaridade foram revisadas sempre com foco em exploração visual da informação e projeção, em alguns casos recuperação de imagens e reconhecimento de padrões (as que envolvem medidas de similaridade). Apesar da diversidade, os diferentes domínios foram conectados por meio de técnicas de projeção, as quais são muito flexíveis tanto com respeito à métrica utilizada para medir a similaridade entre instâncias, como para identificar e visualizar agrupamentos de dados.

Técnicas para identificação e visualização de agrupamentos com base em projeção, por exemplo, garantem que os grupos não fiquem fragmentados durante a visualização. No entanto, o maior desafio está em identificar características nos dados, de modo a agrupá-los. A CSM, discutida no Capítulo 5, propõe uma solução diferenciada para o problema, mediante o uso de um mecanismo de seleção de amostras representativas eficaz, apto a selecionar amostras com base na variabilidade dos dados.

As técnicas que empregam diferentes medidas de similaridade a fim de aumentar a precisão dos sistemas propostos não consideram um fator inerente aos problemas tratados: a "incerteza". Embora não possa ser excluída, a incerteza pode ser estimada e inserida na solução, de modo a aumentar sua acurácia. Esta é a proposta da CSWIRe, apresentada no Capítulo 7, ao realizar buscas por similaridade em coleções de imagens complexas.

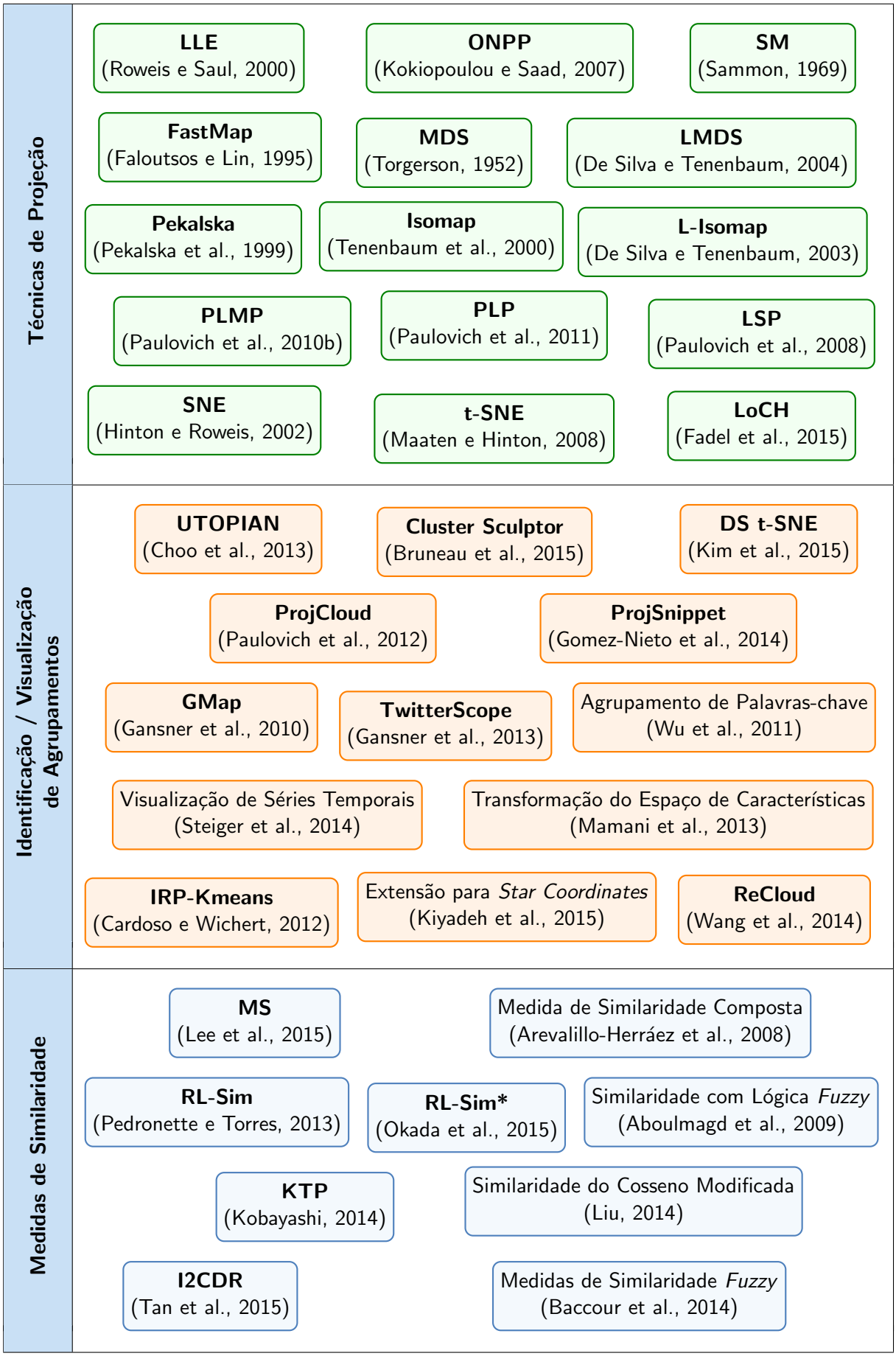


Figura 3.17: Panorama das técnicas revisadas nesta tese, por categoria.

Capítulo

4

A Técnica de Projeção Local: LAMP

Algumas técnicas de projeção permitem a interação do usuário no processo. Contudo, ainda apresentam deficiências que prejudicam seu uso como uma ferramenta completamente interativa na exploração visual. Por exemplo, a maioria das técnicas fazem uso de uma única transformação para projetar dados de um espaço de alta dimensão para o espaço visual. Esta característica dificulta a interação do usuário e impede ajustes locais, já que quaisquer mudanças afetam a projeção como um todo.

Técnicas de projeção baseadas em transformação local também têm deficiências: ou apresentam alto custo computacional, ou não fornecem um mecanismo flexível e suficientemente robusto para permitir que o usuário interaja livremente com a projeção. Uma das principais razões para a falta de flexibilidade está no fato de que as técnicas locais que projetam dados com base em um subconjunto de amostras (ou pontos de controle), requerem muitas amostras posicionadas, a priori, no espaço visual. Portanto, muitas instâncias têm que ser manipuladas para modificar a projeção de modo apropriado, o que torna o processo de interação tedioso e demorado.

Este capítulo apresenta a técnica de projeção multidimensional chamada *Local Affine Multidimensional Projection* (LAMP), a qual possui propriedades singulares que a tornam efetiva na solução dos problemas apontados acima. LAMP tem formulação matemática baseada em mapeamentos ortogonais, garantindo robustez e precisão ao processo. Além disso, sua formulação permite que seja ajustada como uma técnica local, requerendo um número reduzido de amostras para construir o mapeamento. Portanto, pouca intervenção do usuário é necessária para incorporar seu conhecimento à projeção, o que aumenta sua flexibilidade.

A natureza local da LAMP combinada com um mecanismo interativo flexível possibilita a exploração dinâmica e organização de dados, característica que pode ser explorada em muitas aplicações.

Parte da contribuição descrita neste capítulo foi publicada em Joia et al. (2011).

4.1 Principais Contribuições

Entre as principais contribuições deste trabalho estão:

- LAMP: uma técnica de projeção multidimensional baseada em mapeamentos ortogonais. Pode ser ajustada para ser global ou local, dependendo da aplicação. Requer um número reduzido de pontos de controle para guiar o mapeamento. Adequada para aplicações interativas.
- Capacidade de agrupar dados de forma precisa, utilizando poucos pontos de controle.
- Capacidade de correlacionar dados de diferentes naturezas e conjuntos de dados, pela simples manipulação dos pontos de controle.

4.2 Local Affine Multidimensional Projection (LAMP)

De modo similar a outras técnicas interativas, a LAMP faz uso de um subconjunto de amostras ou pontos de controle, e de sua localização no espaço visual. A informação obtida a partir dos pontos de controle é usada para construir uma família de mapeamentos ortogonais afins, um para cada instância a ser projetada. O usuário pode manipular os pontos de controle no espaço visual para melhor organizá-los. Daí em diante o mapeamento segue o *layout* dos pontos de controle. Em outras palavras, o usuário pode guiar a projeção com base nos pontos de controle. A Figura 4.1 ilustra as etapas deste processo.

Figura 4.1: Os três módulos principais que compõem o *framework* da LAMP.

A formulação matemática da LAMP não se baseia em grafos de vizinhança e admite um número bem reduzido de amostras como entrada. Sua formulação usa *Singular Value Decomposition* (SVD) para resolver um problema de minimização conhecido. A Seção 4.2.1 descreve as etapas dessa formulação e cálculo do mapeamento afim, enquanto que a Seção 4.2.2 faz uma análise de como os pontos de controle influenciam na projeção.

4.2.1 Formulação Matemática e Cálculo do Mapeamento Afim

Considere o conjunto de dados $X \subset \mathbb{R}^m$ contendo n instâncias, $X_S = \{x_1, x_2, \ldots, x_k\}$ um subconjunto de pontos de controle de X, e seja $x_i \in X_S$ um representante desse subconjunto. A imagem de X_S no espaço visual ($\mathbb{R}^2$ neste contexto) representa-se por $Y_S = \{y_1, y_2, \ldots, y_k\}$. Então, dada uma instância $x \in X$, a técnica *Local Affine Multidimensional Projection* (LAMP) mapeia x para o espaço visual encontrando a transformação afim

$$f_x(p) = pM + t$$

que minimiza

$$E_{\text{LAMP}} = \sum_{i=1}^{k} \alpha_i \| f_x(\mathrm{x}_i) - y_i \|^2, \quad \text{restrito à } M^\top M = I, \tag{4.1}$$

onde a matriz M e o vetor t são desconhecidos, I é a matriz identidade, e α_i são pesos escalares definidos como:

$$\alpha_i = \frac{1}{\|\mathrm{x}_i - x\|^2}\,. \tag{4.2}$$

O problema de minimização da função objetivo definida pela Equação (4.1) é similar ao empregado em deformação de imagens do tipo "tão-rígido-quanto-possível" (Schaefer et al., 2006). Porém, em oposição às aplicações de deformação de imagem, onde as transformações afins são de $\mathbb{R}^2$ em $\mathbb{R}^2$, aqui os mapeamentos afins são de $\mathbb{R}^m$ em $\mathbb{R}^2$. Então, a formulação usada em deformação de imagens não se aplica neste contexto, muito mais generalizado. Logo, uma nova formulação é requerida.

A restrição $M^\top M = I$ na Equação (4.1) implica uma transformação isométrica no mapeamento. Note que:

$$\|Mx\|^2 = (Mx)^\top Mx = x^\top M^\top Mx = x^\top x = \|x\|^2,$$

logo, a matriz ortogonal M age como uma "isometria" sobre o espaço vetorial $\mathbb{R}^m$, evitando efeitos como escala e cisalhamento. Ou seja, os dados podem ser somente rotacionados e/ou transladados durante o mapeamento. Este comportamento é ideal para projeção multidimensional, já que preserva distâncias tanto quanto possível ao projetar os dados.

Se nenhuma restrição é imposta, o mínimo da Equação (4.1) pode ser obtido através de um procedimento convencional de ajuste por mínimos quadrados. No entanto, neste caso, erros no posicionamento dos pontos de controle podem ser propagados pelos mapeamentos locais afins, resultando em projeções de baixa qualidade. Para uma discussão mais detalhada sobre ortogonalidade e restrições em problemas de minimização, veja Gower e Dijksterhuis (2004).

Além disso, na Equação (4.1) os pesos α_i dependem do ponto de avaliação, mas uma transformação afim distinta é obtida para cada instância x, atendendo este pré-requisito. Finalmente, em contraste às aplicações de deformação de imagem, não é necessário garantir continuidade para a transformação toda, pelo contrário, descontinuidades podem ser altamente desejáveis para manter afastadas as instâncias de dados não correlacionadas durante a projeção. Esta flexibilidade permite restringir o somatório na Equação (4.1) levando em conta somente os pontos de controle em uma vizinhança de x, o que torna o processo totalmente local. Na verdade, quanto mais amostras são consideradas no somatório, menos local será a projeção de x (Seções 4.2.2 e 4.3 abordam esta questão).

Fazendo as derivadas parciais com respeito a t iguais a zero, pode-se escrever t em função de M:

$$t = \tilde{y} - \tilde{\mathrm{x}}M, \quad \tilde{\mathrm{x}} = \frac{\sum_{i=1}^{k} \alpha_i \mathrm{x}_i}{\alpha}, \quad \tilde{y} = \frac{\sum_{i=1}^{k} \alpha_i y_i}{\alpha}, \tag{4.3}$$

onde $\alpha = \sum_{i=1}^{k} \alpha_i$. Assim, o problema de minimização da Equação (4.1) pode ser reescrito como:

$$\sum_{i=1}^{k} \alpha_i \|\hat{\mathrm{x}}_i M - \hat{y}_i\|^2, \quad \text{restrito à } M^\top M = I, \tag{4.4}$$

onde $\hat{\mathrm{x}}_i = x_i - \tilde{x}$ e $\hat{y}_i = y_i - \tilde{y}$.

A Equação (4.4) pode ser expressa na forma matricial, como segue:

$$\|AM - B\|_F, \quad \text{restrito à } M^\top M = I, \tag{4.5}$$

onde $\|\cdot\|_F$ denota a norma de *Frobenius*, e as matrizes A and B são dadas por:

$$A = \begin{bmatrix} \sqrt{\alpha_1}\hat{\mathrm{x}}_1 \\ \sqrt{\alpha_2}\hat{\mathrm{x}}_2 \\ \vdots \\ \sqrt{\alpha_k}\hat{\mathrm{x}}_k \end{bmatrix}, \quad B = \begin{bmatrix} \sqrt{\alpha_1}\hat{y}_1 \\ \sqrt{\alpha_2}\hat{y}_2 \\ \vdots \\ \sqrt{\alpha_k}\hat{y}_k \end{bmatrix}. \tag{4.6}$$

O problema de minimização da Equação (4.5) é um típico exemplo de *Problema de Procrustes Ortogonal* (Gower e Dijksterhuis, 2004), cuja solução é conhecida:

$$M = UV, \quad A^\top B = UDV, \tag{4.7}$$

onde UDV é a decomposição em valores singulares (SVD) de $A^\top B$. Portanto, encontrado M, a projeção de x é dada por:

$$y = f_x(\mathrm{x}) = (\mathrm{x} - \tilde{\mathrm{x}})M + \tilde{y}. \tag{4.8}$$

A princípio, pode parecer que o cálculo de uma decomposição SVD para cada instância x tenha um custo muito elevado para ser empregado em uma aplicação interativa. Todavia,

$A^\top B$ é uma matriz $m \times 2$, ou seja, tem somente duas colunas, logo pode ser decomposta muito rapidamente com robustos pacotes SVD (Blackford et al., 2002) ($O(k)$ operações), resultando em um algoritmo com complexidade computacional igual a $O(kn)$.

O Algoritmo 4.1 descreve os passos necessários para computar a transformação afim, de modo a mapear instâncias de um espaço de alta dimensão para o espaço visual.

Algoritmo 4.1 *Local Affine Multidimensional Projection* (LAMP)

Entrada: Conjunto de dados X, pontos de controle X_S e mapeamento Y_S.
Saída: Mapeamento Y.

1: **para** cada $x \in X$ **faça**
2: Computar os pesos α_i. // Equação (4.2)
3: Computar $\tilde{x}$ e $\tilde{y}$. // Equação (4.3)
4: Construir as matrizes A e B. // Equação (4.6)
5: $UDV \leftarrow$ decomposição SVD de $A^\top B$.
6: $M \leftarrow UV$.
7: Computar o mapeamento $y = (x - \tilde{x})M + \tilde{y}$ e armazenar em Y.
8: **fim para**

4.2.2 Análise dos Pontos de Controle

Existem dois aspectos principais a serem observados quando se lida com os pontos de controle X_S. O primeiro aspecto está relacionado à quantidade de pontos de controle. Técnicas como a PLMP (Paulovich et al., 2010b), *Pekalska* (Pekalska et al., 1999), e PLP (Paulovich et al., 2011) têm limitações quanto ao número mínimo de pontos de controle empregados. PLMP e *Pekalska*, por exemplo, requerem um número de pontos de controle no mínimo igual à dimensão dos dados, enquanto PLP demanda um número mínimo de pontos de controle em cada grafo de vizinhança local. Na prática, estes métodos fazem uso de $k = \sqrt{n}$ pontos de controle para projetar os dados, onde n é o número total de instâncias do conjunto de dados.

A técnica LAMP, entretanto, é muito robusta com respeito ao número de pontos de controle, apresentando baixa distorção mesmo quando um número reduzido de pontos de controle é usado, como mostrado na Figura 4.2. Note que a função de *stress* (apresentada na Definição 2.21) não decai consideravelmente quando o número de pontos de controle aumenta, mostrando que a LAMP pode robustamente mapear instâncias para o espaço visual usando poucos pontos de controle. Um esquema preciso, baseado em força (Tejada et al., 2003), foi utilizado para posicionar os pontos de controle, selecionados aleatoriamente, no espaço visual. O uso deste esquema não compromete a eficiência da LAMP, já que apenas um número reduzido de pontos de controle precisam ser posicionados.

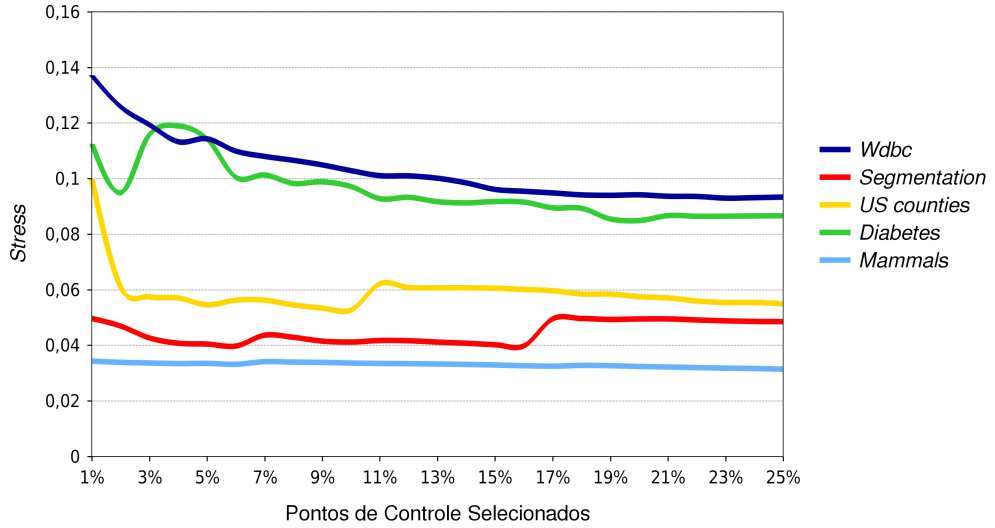


Figura 4.2: *Stress* produzido pela LAMP quando o número de pontos de controle varia de 1% a 25% do total de instâncias do conjunto de dados (ver Tabela 4.1 para detalhes sobre os conjuntos de dados).

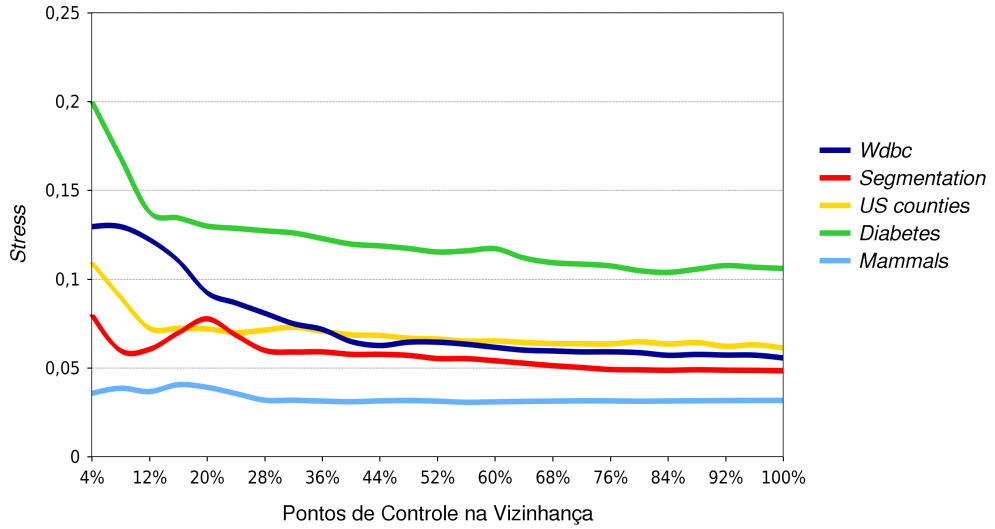


Figura 4.3: *Stress* × porcentagem de pontos de controle mais próximos usados para construir os mapeamentos afins, tal que 100% equivale a usar $\sqrt{n}$ pontos de controle.

Além do número de pontos de controle, o número de termos no somatório da Equação (4.4) pode ser ajustado para modificar o comportamento do mapeamento. Como mencionado na seção anterior, o mapeamento como um todo pode tornar-se descontínuo quando o somatório não percorre todos os pontos de controle em X_S. No entanto, descontinuidades podem ajudar a preservar grupos, bem como melhorar a qualidade da projeção. A Figura 4.3 confirma esta afirmação mostrando que, mesmo quando o número de pontos de controle na vizinhança de cada instância x é pequeno, a medida do *stress* ainda se mantém em baixos níveis. O percentual de pontos de controle mais próximos que aparece na Figura 4.3 é um parâmetro controlado pelo usuário (ver discussão na Seção 4.5).

4.3 Resultados Experimentais e Comparações

Nesta seção são apresentados alguns resultados obtidos com a LAMP.

Os conjuntos de dados utilizados nas comparações estão relacionados na Tabela 4.1. Note que, tais conjuntos variam significativamente tanto em número de instâncias como atributos, permitindo comparações mais confiáveis. Quanto às técnicas empregadas nas comparações, foram escolhidas com base nos seguintes critérios:

- Apresentam bom desempenho em termos de *stress* e/ou tempo computacional; ou
- Utilizam um subconjunto de amostras para executar o mapeamento.

Mais especificamente, foram utilizadas as seguintes técnicas: *Glimmer* (Ingram et al., 2009) devido ao seu bom desempenho em termos de *stress*; *FastMap* (Faloutsos e Lin, 1995) por ser uma das técnicas de projeção mais rápidas conhecidas; PLMP (Paulovich et al., 2010b), *Hybrid* (Jourdan e Melançon, 2004), LMDS (De Silva e Tenenbaum, 2004), L-Isomap (De Silva e Tenenbaum, 2003), *Pekalska* (Pekalska et al., 1999), LSP (Paulovich et al., 2008) e PLP (Paulovich et al., 2011) porque apresentam bons resultados com relação ao *stress*/tempo e porque se apoiam em um subconjunto de amostras representativas para executar a projeção multidimensional.

Todos os resultados apresentados nesta seção foram produzidos por um microcomputador Intel® Core™ i7, CPU 920 2.66GHz, placa NVIDIA® Quadro FX 3800 e 8 GB de memória RAM; implementação em Java, utilizando a biblioteca numérica *Linear Algebra for Java* (jblas) (Braun et al., 2009) para a decomposição SVD.

Tabela 4.1: Conjuntos de dados utilizados nas comparações da LAMP, da esquerda para a direita as colunas correspondem ao nome do conjunto de dados, total de instâncias, número de atributos e origem dos dados.

Nome	Instâncias	Dimensão	Origem
Wdbc	569	30	[a]
Diabetes	768	8	[a]
Segmentation	2.100	19	[a]
US counties	3.028	14	[b]
Isolet	6.238	617	[a]
Letter rcn	20.000	16	[a]
Mammals	50.000	72	[a]
Viscontest	200.000	10	[c]

[a] Frank e Asuncion (2010)

[b] Seo e Shneiderman (2004)

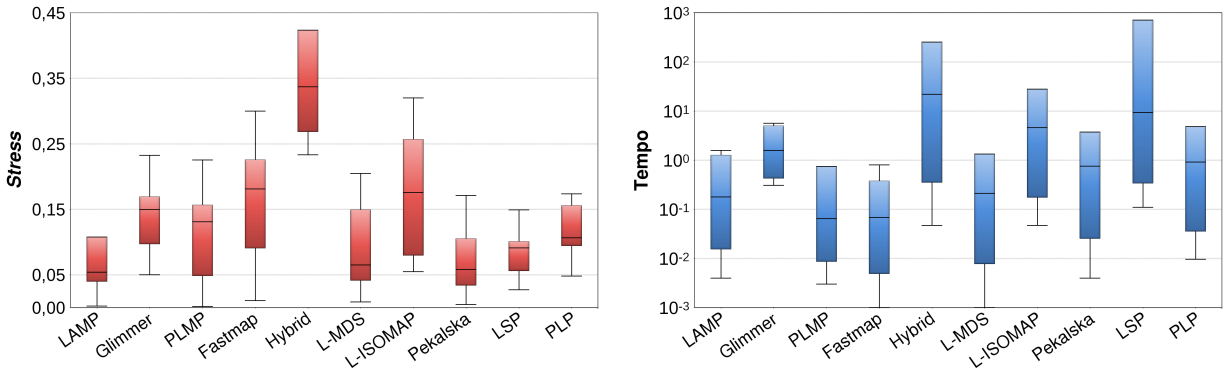
[c] Whalen e Norman (2008)

4.3.1 Preservação de Distâncias

Como primeiro experimento, considere os gráficos de caixas verticais mostrados na Figura 4.4. Estes gráficos mostram, respectivamente, a precisão e a eficiência computacional da LAMP, utilizando os oito conjuntos de dados da Tabela 4.1.

O gráfico à esquerda (Figura 4.4(a)) mostra o valor do *stress* para todas as técnicas comparadas. Os resultados confirmam que a LAMP é uma das técnicas mais precisas, equiparando-se à técnicas altamente precisas como *Pekalska*, por exemplo.

A Figura 4.4(b) mostra os tempos computacionais das técnicas comparadas. Note que a LAMP é bastante competitiva, equiparada a métodos do estado da arte como a PLP. De fato, a LAMP só tem desempenho inferior a PLMP e *FastMap*, técnicas conhecidas por seu baixo custo computacional.



(a) *Stress* (b) Tempo computacional (em segundos)

Figura 4.4: Comparação do *stress* e tempo computacional da LAMP contra outras técnicas de projeção.

Observe agora os gráficos de dispersão mostrados na Figura 4.5. Eles representam as distâncias no espaço original versus as distâncias no espaço de projeção, isto significa que quanto mais pontos estiverem em torno da linha de 45^o, melhor a preservação de distâncias no espaço de projeção. Note que a LAMP é superior às outras técnicas em praticamente todos os casos, preservando muito bem as distâncias durante a projeção. Já em outras como *Hybrid* e L-Isomap, por exemplo, a dificuldade em preservar distâncias é bastante evidente.

4.3.2 Agrupamento de Dados a Partir dos Pontos de Controle

A LAMP foi concebida para ser interativa, permitindo ao usuário, dinamicamente, interagir com a projeção pela manipulação de pontos de controle. Preferencialmente, organizando-os através de suas classes ou características visuais, tais como, padrões, formas ou cores.

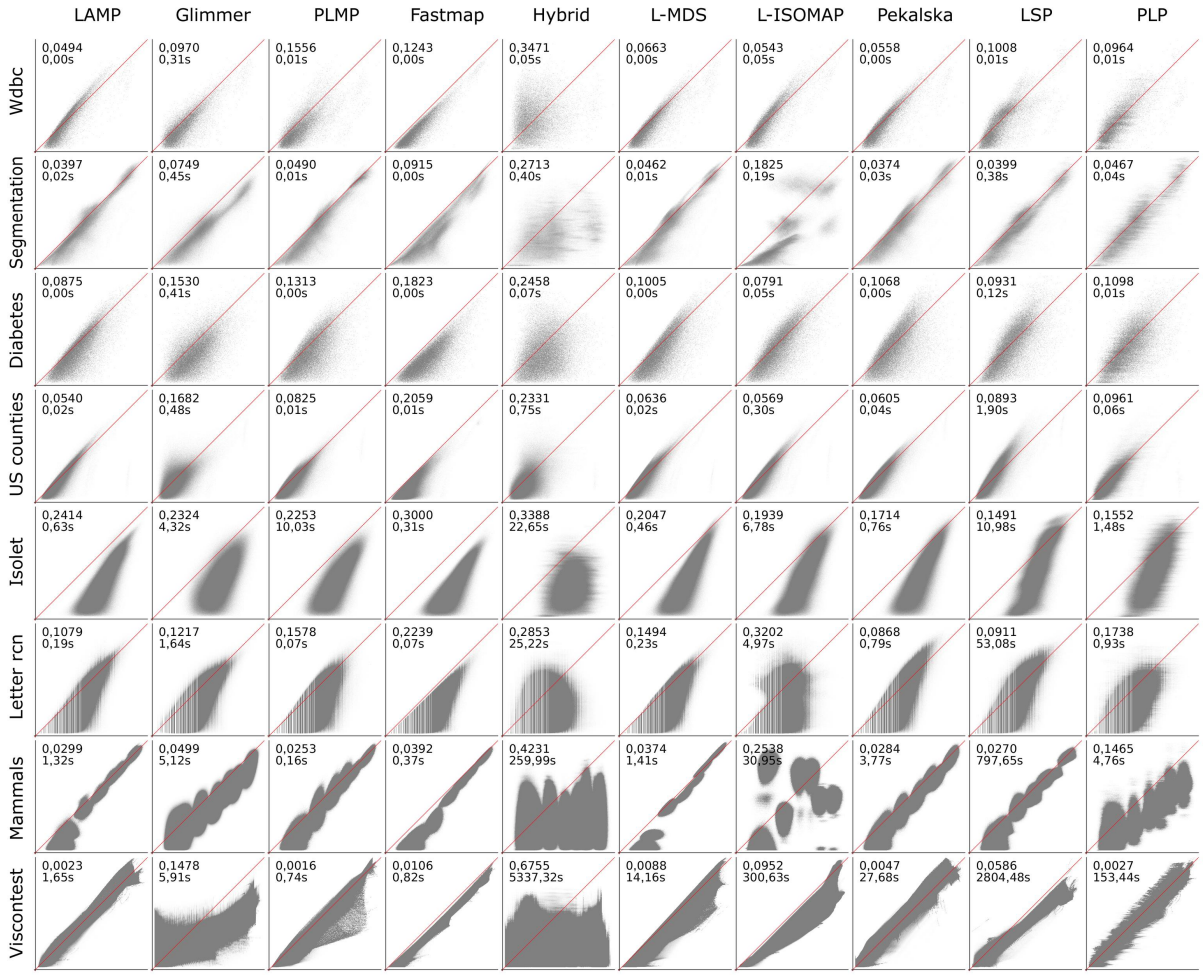


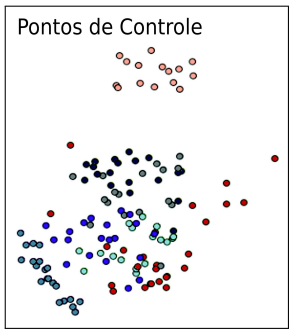
Figura 4.5: Distância no espaço original × distância no espaço de projeção. Os valores na região superior esquerda de cada subfigura correspondem ao *stress* normalizado e tempo computacional em segundos.

A Figura 4.6(a), por exemplo, mostra o mapeamento de alguns pontos de controle selecionados aleatoriamente a partir do conjunto de dados *Segmentation*, mapeados para o espaço visual utilizando o esquema de forças discutido na Seção 4.2.2. A Figura 4.6(b) mostra o resultado da intervenção do usuário, após agrupar os pontos de controle no espaço visual.

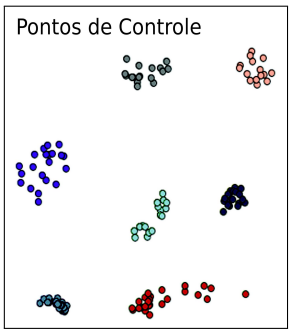
Os mapeamentos produzidos pela LAMP tendem a seguir fielmente o *layout* dos pontos de controle fornecido pelo usuário. Logo, quando os pontos de controle estão agrupados como na Figura 4.6(b), agrupamentos de dados tendem a se formar também com a projeção. Para atestar a qualidade destes agrupamentos foi utilizada a medida da silhueta, indicada por `Silh`, cujos possíveis valores variam no intervalo $[-1, 1]$. Quanto maior for este valor, melhor a coesão e a separação dos agrupamentos. Ver Definição 2.8 para mais detalhes sobre esta medida.

As Figuras 4.6(c) a 4.6(f) descrevem os mapeamentos produzidos pela LAMP, LSP, *Pekalska* e PLMP, respectivamente. Todos eles usam a configuração dos pontos de controle mostrado na Figura 4.6(b). A LAMP está usando todos os pontos de controle para

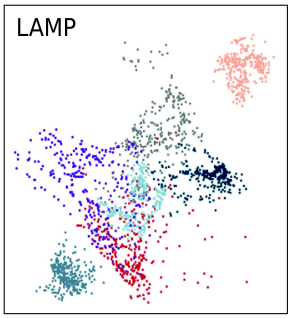
computar o mapeamento afim f_x para cada instância x. Como pode ser observado, a medida da silhueta da LAMP não é tão boa quanto as produzidas por LSP e *Pekalska*. Isto significa que os agrupamentos são melhor preservados por estas duas técnicas.



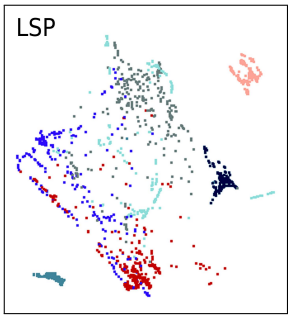
(a) Pontos de controle mapeados pelo *Force*



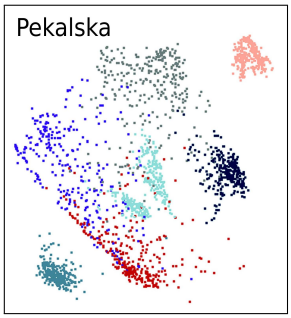
(b) *Layout* ajustado pelo usuário



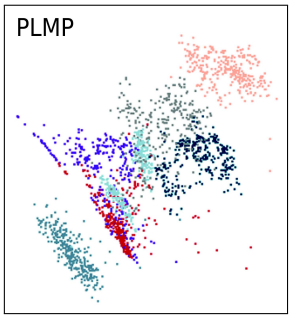
(c) `Silh` $= 0,4003$



(d) `Silh` $= 0,4584$

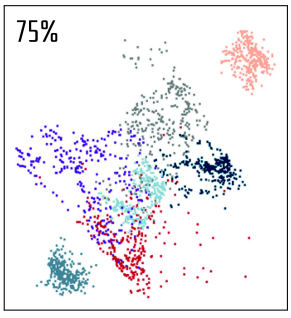


(e) `Silh` $= 0,5083$

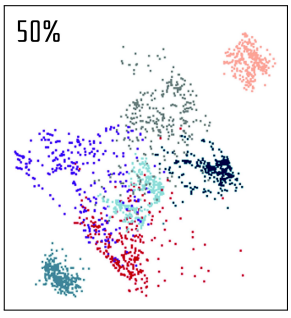


(f) `Silh` $= 0,2475$

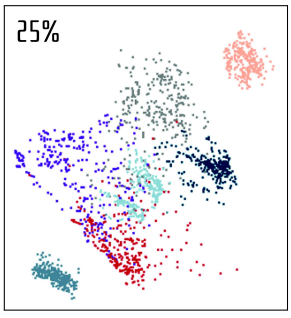
Figura 4.6: Projeções produzidas pela LAMP, LSP, *Pekalska* e PLMP a partir dos pontos de controle manipulados pelo usuário.



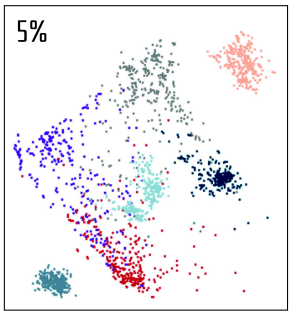
(a) `Silh` $= 0,4166$



(b) `Silh` $= 0,4365$



(c) `Silh` $= 0,4682$



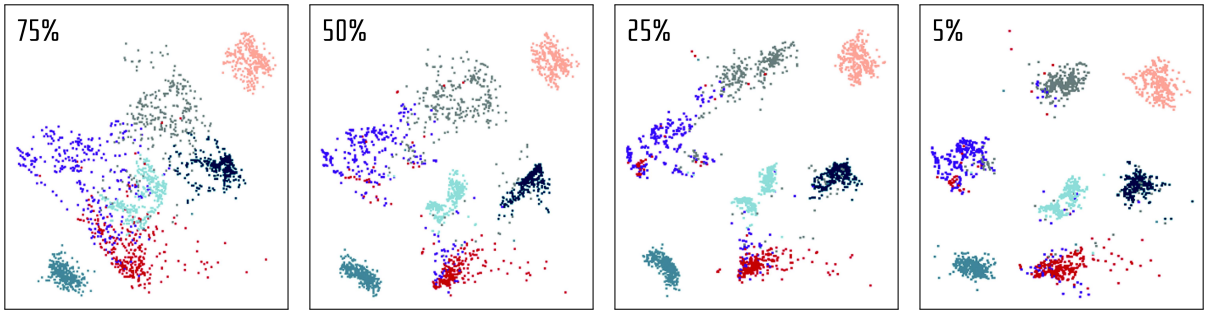
(d) `Silh` $= 0,5235$

Figura 4.7: Projeções produzidas pela LAMP variando o percentual dos pontos de controle mais próximos usados para construir o mapeamento, computados a partir do espaço de alta dimensão, tal que 100% representa $\sqrt{n}$ pontos de controle.

A situação é diferente quando se explora a natureza local da LAMP, isto é, quando se usam os pontos de controle mais próximos de cada instância x para construir o mapeamento f_x, como pode ser observado nas Figuras 4.7(a) a 4.7(d). Note que o valor `Silh` aumenta quando o percentual dos pontos de controle mais próximos diminui (75%, 50%,

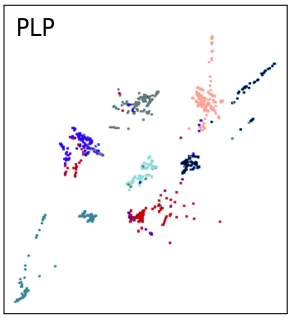
25% e 5%), atingindo um valor de silhueta maior do que as outras três técnicas globais apresentadas.

O uso de informações 2D torna a LAMP bastante sensível à posição dos pontos de controle no espaço visual, produzindo mapeamentos que seguem fielmente o *layout* dos pontos de controle. Este fato pode ser observado nas Figuras 4.8(a) a 4.8(d), onde as instâncias mapeadas tornam-se cada vez mais agrupadas à medida que a porcentagem de vizinhos mais próximos varia de 75% a 5%, e os vizinhos são definidos com base no espaço visual. Os valores da silhueta confirmam que a LAMP separa melhor os grupos quando informações 2D são consideradas, apresentando resultados superiores aos da PLP (Figura 4.9).



(a) $\texttt{Silh} = 0,4590$ (b) $\texttt{Silh} = 0,5496$ (c) $\texttt{Silh} = 0,5570$ (d) $\texttt{Silh} = 0,5644$

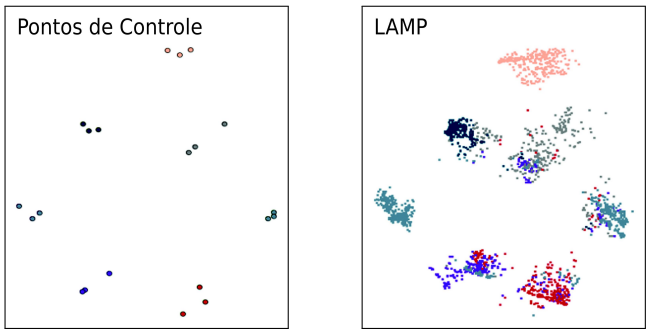
Figura 4.8: Projeções produzidas pela LAMP variando o percentual dos pontos de controle mais próximos usados para construir o mapeamento, computados a partir do espaço visual, tal que 100% representa $\sqrt{n}$ pontos de controle.



$\texttt{Silh} = 0,4411$

Figura 4.9: Projeção produzida pela PLP usando distâncias no espaço visual e a configuração dos pontos de controle mostrado na Figura 4.6(b).

A Figura 4.10(a) mostra três pontos de controle selecionados aleatoriamente em cada uma das sete classes que compõem o conjunto de dados, sendo interativamente posicionados no espaço visual de modo a manter classes distintas bem separadas. A Figura 4.10(b) mostra o mapeamento produzido pela LAMP usando este reduzido conjunto de pontos de controle (vizinhança definida a partir do espaço visual). Note que mesmo usando um número reduzido de pontos de controle (21 ao todo), a LAMP conseguiu projetar o conjunto de dados consistentemente, gerando um valor de silhueta comparável ao da PLP que usa muito mais pontos de controle (Figura 4.9).



(a) *Layout* ajustado pelo usuário (b) `Silh` $= 0,4361$

Figura 4.10: Projeção produzida pela LAMP (vizinhança em $\mathbb{R}^2$) usando apenas alguns pontos de controle, 3 por classe (21 no total contra 137 necessários para executar a PLP).

É importante enfatizar que a PLP, PLMP e *Pekalska* não são capazes de executar mapeamentos usando tão poucos pontos de controle. Além disso, devido à sua natureza global, PLMP e *Pekalska* não podem executar uma drástica separação de instâncias tal como a produzida pela LAMP na Figura 4.8(d).

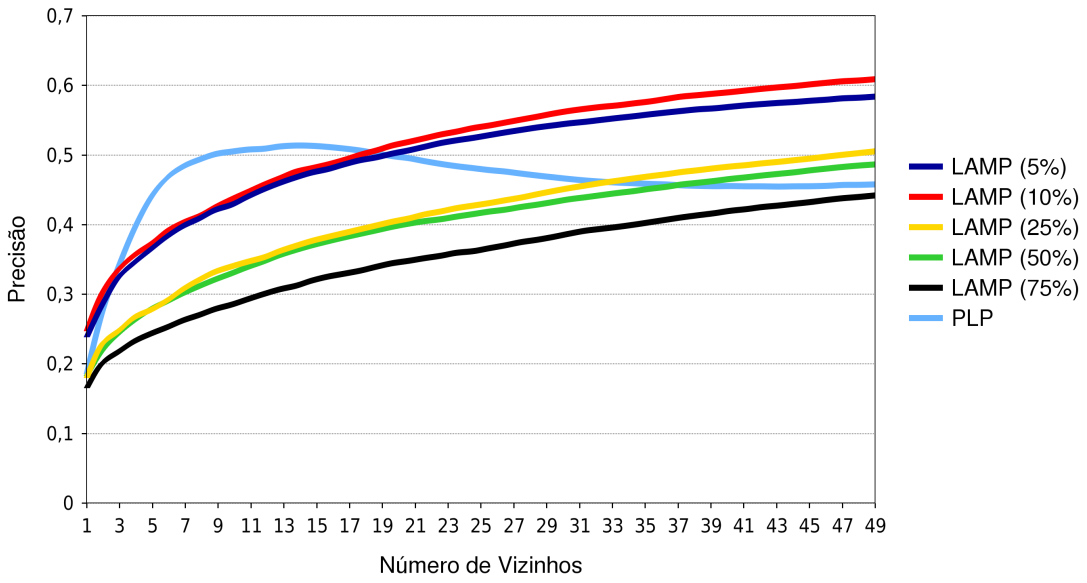


Figura 4.11: Preservação de vizinhança para PLP e LAMP.

A LAMP também apresenta boa preservação de vizinhança (ver Definição 2.22), conforme pode ser observado na Figura 4.11. Neste gráfico, são apresentadas várias curvas de preservação de vizinhança da LAMP, variando o percentual de pontos de controles mais próximos utilizados para construir o mapeamento, entre: 5%, 10%, 25%, 50% e 75%. Note que a precisão aumenta quando a natureza local da LAMP é explorada, ou seja, quando apenas 5% e 10% dos pontos de controle mais próximos são utilizados no cálculo do mapeamento afim.

4.4 Aplicação: Correlação Visual de Dados

Aplicações envolvendo correlação de dados assistida pelo usuário visam relacionar instâncias de conjuntos de dados que, a princípio, não têm qualquer conexão. A ideia é iniciar com um reduzido conjunto de pontos de controle, selecionados a partir de diferentes conjuntos de dados e interativamente manipular estes pontos de controle no espaço visual, a fim de deixar as instâncias que devem ser correlacionadas tão próximas quanto possível. Uma vez que os pontos de controle dos diferentes conjuntos de dados estão correlacionados, isto é, agrupados bem próximos no espaço visual, as instâncias restantes são projetadas utilizando a LAMP.

Como o mapeamento produzido pela LAMP segue a configuração dos pontos de controle, instâncias de diferentes conjuntos de dados são projetadas próximas umas das outras no espaço visual, fazendo com que fiquem completamente correlacionadas. A Figura 4.12 ilustra o protótipo de um sistema desenvolvido para executar esta tarefa e estabelecer uma correspondência entre imagem e música. Um vídeo demonstrando cada uma das etapas deste processo encontra-se disponível em `http://sites.google.com/site/paulojoiafilho/publications`.

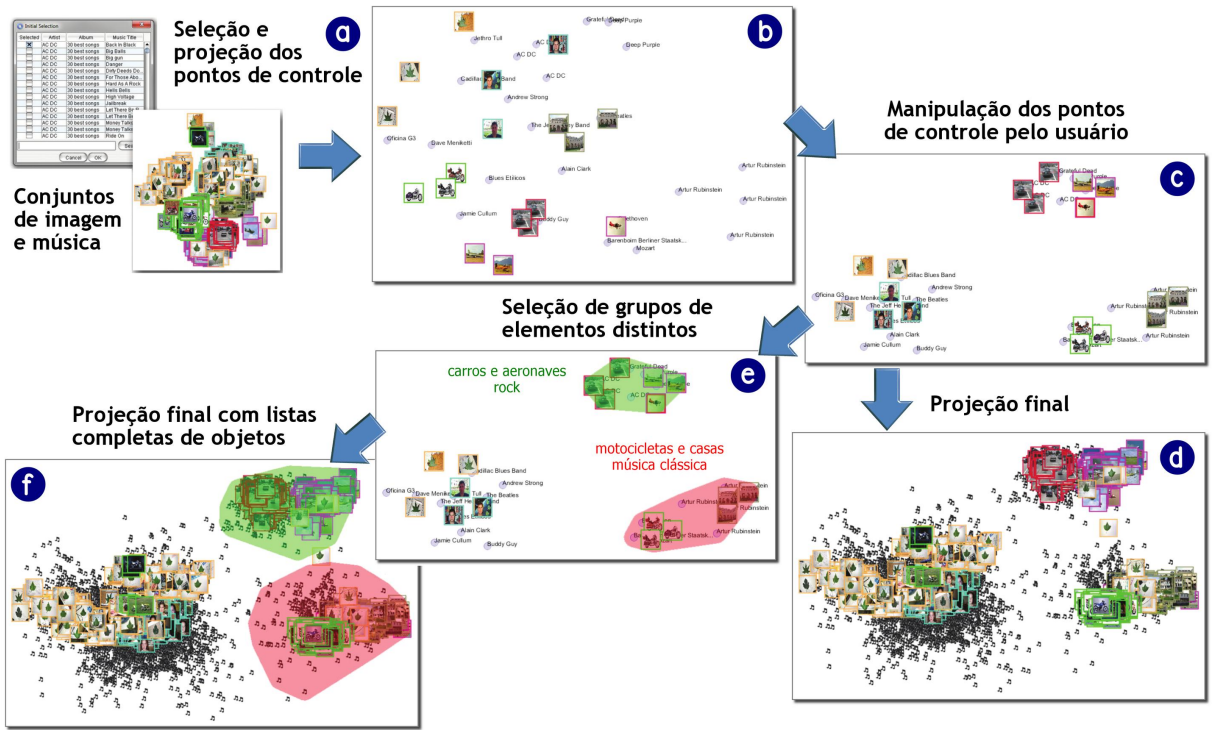


Figura 4.12: Correlação entre imagem e música: alguns representantes (pontos de controle) de música e imagem são selecionados a partir do correspondente conjunto de dados (a) e colocados no espaço visual (b). O usuário interage com as amostras de imagem e música de modo a correlacioná-las (c). A LAMP mapeia músicas e figuras segundo as associações realizadas pelo usuário (d). O usuário pode destacar múltiplas regiões no espaço visual (e), onde cada região corresponde às figuras e músicas que irão compor uma apresentação de *slides* (f).

4.5 Considerações Finais

A técnica de projeção apresentada neste capítulo, *Local Affine Multidimensional Projection* (LAMP), mostrou-se adequada para aplicações interativas pelo fato de mapear dados com base em um número bem reduzido de amostras representativas ou pontos de controle. LAMP tem sólida formulação matemática, robustez e versatilidade.

Os experimentos realizados provam que a LAMP supera as outras técnicas de projeção em termos de preservação de distâncias, além de ser competitiva em relação aos tempos computacionais. A medida da silhueta foi utilizada para mostrar que os mapeamentos produzidos pela LAMP, a partir de alguns pontos de controle rotulados e em seguida organizados pelo usuário de forma interativa, podem originar grupos altamente separados e coesos.

Como os mapeamentos produzidos tendem a seguir fielmente o *layout* dos pontos de controle, é possível aplicar a LAMP para estabelecer correlações entre dados aparentemente sem conexão, conforme apresentado na seção anterior.

Embora não tenha sido explorado neste trabalho, também é possível utilizar a LAMP como uma ferramenta de classificação para predizer a classe de futuras instâncias de dados. Por construção, a LAMP permite mapear novas instâncias de dados de forma isolada, sem refazer ou recalcular o mapeamento dos pontos já projetados. Desse modo, basta que os pontos de controle sejam rotulados (comportando-se como um conjunto de treinamento) para predizermos a classe de novas instâncias projetadas, com base na classe do ponto de controle mais próximo (função de classificação).

LAMP é essencialmente uma técnica local, isto significa que tenta preservar a geometria local dos dados durante a projeção, característica que se torna evidente quando apenas um percentual dos pontos de controle mais próximos de cada instância são utilizados no mapeamento. Escolher o número ideal de pontos mais próximos para produzir o *layout* desejado, no entanto, é um aspecto que precisa ser melhor investigado. Uma possibilidade é tentar encontrar o raio de influência de cada ponto de controle, embora não seja uma tarefa fácil.

Muitas vezes, os pontos de controle não são rotulados e também não podem ser organizados no espaço visual por meio de suas características. Para estas situações, uma nova abordagem capaz de identificar instâncias representativas em conjuntos de dados não rotulados e desbalanceados foi desenvolvida. Esta abordagem, assunto do próximo capítulo, utiliza a LAMP em uma de suas etapas para projetar e identificar grupos no espaço visual.

Capítulo

5

Identificação de Grupos no Contexto de Projeção

MUITOS métodos de visualização combinam projeção multidimensional com esquemas de detecção de agrupamentos. Uma abordagem típica agrupa instâncias similares segundo suas distâncias no espaço visual, assim, os grupos são definidos com base exclusiva na geometria dos pontos no espaço visual.

Embora a análise puramente geométrica, em alguns casos, consiga apresentar grupos visualmente separados, não existe garantia alguma de que os grupos obtidos reflitam qualquer correlação entre os dados. Além do mais, muitas técnicas de visualização empregam esquemas de agrupamento não determinísticos, produzindo diferentes *layouts* cada vez que o conjunto de dados é visualizado.

Muitos algoritmos de detecção de agrupamentos operam adequadamente quando o conjunto de dados é balanceado, ou melhor, quando a frequência relativa das classes não é extrema em uma determinada classe (Definição 2.2). Quando o conjunto de dados é desbalanceado, a tarefa de detecção de agrupamentos é bem mais complexa. Embora existam métodos de balanceamento de amostras visando reduzir a disparidade entre a proporção de instâncias por classe em um conjunto de dados (Larose, 2006), tais técnicas não se aplicam neste contexto, pois o balanceamento pode provocar a eliminação de grupos com poucos representantes, ou estimar novas categorias que não condizem com os dados originais.

Este capítulo apresenta um novo método de visualização baseado em projeção multidimensional que permite agrupar dados. Além disso, opera no espaço visual, garantindo que os grupos obtidos não fiquem fragmentados durante a visualização, ou seja, elementos aparentemente dispersos são agregados e posicionados em torno de seu centro, implicando

melhor coesão e separação (Palumbo et al., 2008). Em contraste às técnicas existentes, o esquema de agrupamento utilizado é orientado por um mecanismo de amostragem determinístico, apto a identificar instâncias que representam bem o conjunto de dados como um todo. O mecanismo de amostragem fundamenta-se em decomposição matricial e consegue operar mesmo em conjuntos de dados desbalanceados. Desse modo, o método proposto permite visualizações mais confiáveis, já que o usuário tem certa garantia de que cada grupo visualizado corresponde a um padrão específico dos dados.

O padrão mencionado é determinado pelo mecanismo de amostragem, o qual é sensível à variação dos dados, logo pode localizar instâncias representativas em cada classe, mesmo em conjuntos de dados desbalanceados, com boa precisão. Isto significa que, mesmo instâncias pertencentes a classes com baixa frequência têm boas chances de serem amostradas.

Outro aspecto interessante do método proposto está no fato do mecanismo de amostragem ser facilmente adaptado para selecionar os atributos mais relevantes que representam cada agrupamento obtido. Portanto, esta abordagem unifica em um simples *framework* três tarefas amplamente utilizadas no contexto de visualização: amostragem de dados, detecção de agrupamentos e seleção de atributos.

O método desenvolvido foi denominado *Column Selection Method* (CSM), pelo modo que opera: instâncias são representadas como colunas durante o processo de decomposição matricial (Seção 5.2). Uma bateria completa de testes confirma sua eficácia quando comparado a algoritmos de amostragem (Seção 5.3.1), detecção de agrupamentos (Seção 5.3.2) e seleção de atributos (Seção 5.3.3).

Parte da contribuição descrita neste capítulo foi publicada em Joia et al. (2015).

5.1 Principais Contribuições

Em resumo, as principais contribuições do trabalho apresentado neste capítulo são:

- Um mecanismo de amostragem de dados determinístico apto a operar com precisão mesmo em conjuntos de dados desbalanceados.

- Um esquema de agrupamento de dados baseado no mecanismo de amostragem proposto, garantindo que os grupos obtidos sejam diferentes entre si não apenas pela distância, mas também pelo seu conteúdo.

- Um esquema de seleção de atributos capaz de identificar os atributos que melhor representam cada agrupamento de dados obtido.

5.2 Column Selection Method (CSM)

A Figura 5.1 ilustra os cinco passos principais da CSM: 1) Identificação de instâncias representativas; 2) Projeção multidimensional; 3) Detecção de agrupamentos; 4) Seleção de atributos; e 5) Organização do *layout* com base nos grupos obtidos.

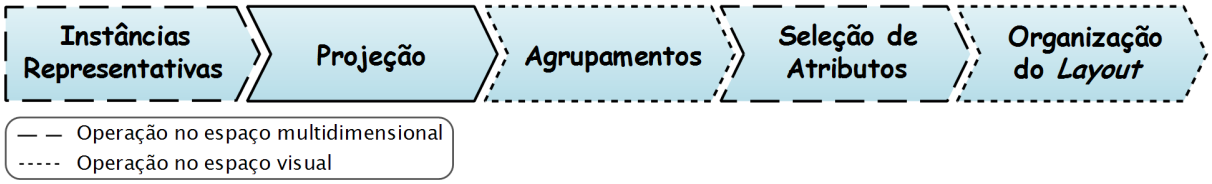


Figura 5.1: *Pipeline* da CSM.

O Passo 1 do *pipeline* consiste em identificar, a partir dos dados originais, o subconjunto de instâncias que melhor representa o conjunto de dados como um todo. No Passo 2, os dados são mapeados para o espaço visual usando uma técnica de projeção precisa. Neste trabalho, foi utilizada a LAMP (Joia et al., 2011). No Passo 3, as instâncias similares são agrupadas de acordo com o critério de distância até a instância representativa mais próxima. No Passo 4, os atributos mais relevantes de cada agrupamento são selecionados, os quais são utilizados no Passo 5 para compor o *layout* final da visualização.

As próximas subseções detalham, respectivamente, os Passos 1, 3 e 4, uma vez que eles correspondem às principais contribuições deste trabalho.

5.2.1 Identificação de Instâncias Representativas

A ideia central desta etapa é identificar bons representantes para o conjunto de dados como um todo. No entanto, decidir se uma instância constitui bom representante não é tarefa fácil, principalmente se forem consideradas apenas informações geométricas dos pontos, como a distância entre eles. Por este motivo, esta formulação leva em conta também a variabilidade dos dados, alcançada através de uma sólida formulação matemática proveniente da decomposição de matrizes, cuja solução deriva da teoria da decomposição em valores singulares (SVD).

Considere uma matriz de dados A, de dimensão $n \times m$, onde as instâncias são armazenadas como colunas[1], tal que cada coluna a^j, $j = 1, \ldots, m$, corresponde a uma instâncias de dimensão n. O SVD de A é a decomposição $A = U\Sigma V^\top$, onde $U = [u^1 u^2 \cdots u^n]$ e $V = [v^1 v^2 \cdots v^m]$ são matrizes ortogonais com dimensões $n \times n$ e $m \times m$ respectivamente, e Σ é a matriz diagonal retangular $n \times m$ com entradas não nulas correspondendo aos valores singulares de A. Os valores singulares são tipicamente arranjados tal como $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_p, 0, \cdots, 0)$ com p valores não nulos, tal que $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p > 0$.

[1]Embora esta não seja a forma usual de armazenamento, ela é necessária para fins desta formulação.

É bem conhecido da teoria SVD que as colunas de U correspondem aos componentes principais de A (Jolliffe, 2002), isto é, o conjunto $\{u^i\}$ gera o subespaço linear onde os dados originais (colunas de A) são bem representados. Em outras palavras, as colunas de U correspondem à melhor base de vetores (em termos de uma norma de matriz apropriada) para representar o conjunto de dados.

Se um subconjunto de colunas de A se "aproxima" de $\{u^i\}$, então é razoável admitir que este subconjunto será uma boa representação para o conjunto de dados. O problema está em decidir quais colunas de A se aproximam da base, dada pelas colunas de U. Uma solução seria medir a correlação entre cada coluna de A com o conjunto de colunas de U, escolhendo como instâncias representativas o subconjunto de colunas de A onde os valores da correlação sejam o maior possível. Felizmente, o SVD de A fornece esta informação. De fato, o SVD permite escrever:

$$\Sigma^{-1}U^\top A = V^\top, \tag{5.1}$$

onde Σ^{-1} é a matriz diagonal cujas entradas são o inverso dos valores singulares não nulos de Σ. Fixando uma única coluna de A, a Equação (5.1) torna-se,

$$\Sigma^{-1}U^\top a^j = [v_j^1\ v_j^2 \cdots v_j^m]^\top, \tag{5.2}$$

onde v_j^i é a j-ésima coordenada da coluna v^i de V. Na Equação (5.2), os valores da correlação entre a^j e cada coluna de U são dados, exceto por uma escala, pelas entradas v_j^i, as quais correspondem à j-ésima coluna de $V^\top$. Ou melhor, o elemento v_j^i é a medida da correlação entre a coluna a^j e a base de vetores u^i.

Usando o argumento anterior, é possível amostrar as colunas de A segundo seus valores de correlação com as colunas de U, medidos pelo seguinte escore:

$$\pi_j = \sum_{i=1}^{k} (v_j^i)^2. \tag{5.3}$$

Os π_j valores devem ser ordenados de forma decrescente e os respectivos índices das colunas (de $V^\top$), armazenados. Tais índices são usados para obter as c instâncias mais representativas de A, onde c é o número de instâncias que pretende-se amostrar. Na prática, para selecionar as colunas mais representativas de A deve-se primeiro ordenar as colunas de $V^\top$ em ordem decrescente de valores de π e usar seus índices para recuperar as colunas de A.

Note que o somatório da Equação (5.3) é iterado até k, o qual corresponde às k primeiras colunas de U. A razão para considerarmos somente as k primeiras colunas de U é que a maioria da informação contida em A pode ser representada pela base de vetores

u^i associada aos maiores valores singulares de A. Valores singulares pequenos, na maioria das vezes, correspondem a ruídos e podem ser descartados.

Lembrando que, os k primeiros vetores da base $\{u^i\}$ correspondem às k componentes principais dos dados, então estamos escolhendo as colunas a^j que mais contribuem com a representação de tais componentes. Desse modo, o mecanismo de amostragem tende a selecionar colunas distintas umas das outras, em relação aos eixos principais. Obtendo, portanto, instâncias com grande diversidade.

As Figuras 5.2(a) e 5.2(b) ilustram o mecanismo de amostragem na prática.

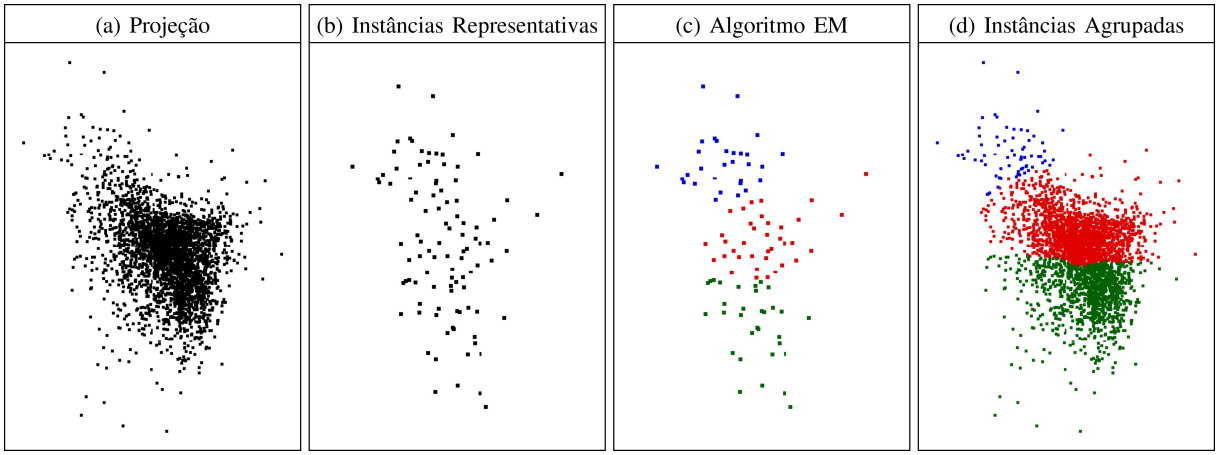


Figura 5.2: Identificação de três agrupamentos pelo método proposto, onde cerca de 3% de instâncias representativas foram selecionadas a partir do conjunto de dados *US counties* (ver Tabela 5.1).

O Algoritmo 5.1 descreve os principais passos do método de amostragem. Observe que este algoritmo é simples de ser implementado, requerendo basicamente uma biblioteca para resolver o SVD e um método de ordenação de dados.

Algoritmo 5.1 *Column Selection Method* (CSM)

Entrada: Matriz de dados A, parâmetro k e número de colunas c.
Saída: Matriz C com c colunas selecionadas.

procedimento *RepresentativeSelection*(A, k, c)

1: $[U, \Sigma, V^\top] \leftarrow$ SVDTRUNCADO(A, k)
2: **para toda** coluna j em $V^\top$ **faça**
3: $\quad \pi_j \leftarrow \sum_{i=1}^{k} (v_j^i)^2$
4: **fim para**
5: $Idx \leftarrow$ índices das colunas com os c maiores valores de π
6: $C \leftarrow A(Idx)$

Embora o valor de k seja um parâmetro de entrada no Algoritmo 5.1, o valor

$$k = \min\{m, n\} \backslash 2 + 1 \tag{5.4}$$

foi utilizado nos experimentos, por produzir bons resultados. O símbolo "\" na equação acima indica divisão inteira. Truncar o resultado do SVD em algum número k significa criar uma representação de A mais fácil de analisar e interpretar, já que as principais informações estão contidas nos primeiros termos, conforme já discutido. A representação mais comum é obtida considerando k termos, com $k \ll \min\{m, n\}$, em grande parte porque dá a melhor aproximação de posto-k para A (Drineas et al., 2008).

Note que, uma vez estimado o valor do parâmetro k, por meio da Equação (5.4), esta abordagem requer como entrada apenas a matriz A e o número de colunas a serem selecionadas c, tal que c é tipicamente maior do que k. Estudos no contexto de aproximação de matrizes (Boutsidis et al., 2014) indicam a possibilidade de estimar c comparando o posto da matriz de dados A com o posto da matriz de colunas selecionadas C, todavia, esta estimativa tem alto custo computacional, desencorajando seu uso neste contexto.

O tempo de execução do Algoritmo 5.1 é dominado pela decomposição SVD. Nossa implementação utiliza o algoritmo LAS2 (Berry, 1992) para calcular o SVD truncado, o qual tem complexidade assintótica $O(mnk)$.

Abordagens similares têm sido empregadas na detecção de *outliers* (Velleman e Welsch, 1981), na fatoração de matrizes (Mahoney e Drineas, 2009) e na análise genômica (Paschou et al., 2007). Todavia, no contexto de visualização, o método de seleção de colunas via decomposição de matrizes constitui abordagem inovadora, principalmente por reunir em um único *framework*: amostragem de dados, detecção de agrupamentos e seleção de atributos.

5.2.2 Detecção de Agrupamentos

Uma vez que as instâncias tenham sido selecionadas pelo *Column Selection Method*, elas são mapeadas para o espaço visual. A LAMP (Joia et al., 2011) foi utilizada para realizar este mapeamento.

Como as instâncias são selecionadas com base em sua correlação com as colunas de U, o mecanismo de amostragem pode selecionar várias instâncias para representar o mesmo agrupamento. A fim de evitar instâncias redundantes, as mesmas são agrupadas segundo sua proximidade. Vários algoritmos de agrupamento foram testados para agrupar as instâncias representativas, optando-se pelo *Expectation Maximization* (EM) (Dempster et al., 1977), devido aos bons resultados apresentados.

Com as instâncias representativas selecionadas e agrupadas pelo EM, os grupos são pré-definidos, restando apenas associar as instâncias não representativas. Considerando que as posições das instâncias no espaço visual são conhecidas, então cada instância não

representativa é associada à instância representativa mais próxima, usando a métrica Euclidiana. Ou melhor, é rotulada de acordo com a instância representativa mais próxima.

As Figuras 5.2(c) e 5.2(d) ilustram, respectivamente, estes passos, ou seja, agrupamento das instâncias representativas com EM e formação dos grupos com base na instância representativa mais próxima.

5.2.3 Seleção de Atributos

Muitas técnicas têm sido propostas para selecionar os atributos mais relevantes de um conjunto de dados de alta dimensão. Algumas abordagens são sofisticadas e de difícil implementação, com algoritmos de alto custo computacional. Além disso, a maioria das técnicas são supervisionadas e trabalham de maneira global, selecionando os atributos que caracterizam o conjunto de dados como um todo. Esta seção, apresenta um esquema de seleção de atributos capaz de atuar isoladamente sobre grupos de instâncias, conforme segue.

É razoável assumir que instâncias pertencentes a um mesmo grupo compartilhem propriedades semelhantes, as quais podem ser descritas por seus atributos. Normalmente, apenas um subconjunto desses atributos já é suficiente para caracterizar as instâncias. Supondo que o esquema de agrupamento de dados, descrito na subseção anterior, dê origem a l grupos e sejam $A_1, \ldots, A_l$ as matrizes de dados de cada grupo, onde as colunas de A_i correspondam às instâncias do i-ésimo grupo. Logo, se o Algoritmo 5.1 for aplicado para selecionar linhas ao invés de colunas, o resultado será o conjunto de atributos mais relevantes de cada grupo. Portanto, selecionar os atributos mais relevantes de A_i implica simplesmente em executar o Algoritmo 5.1, tomando como entrada a matriz transposta de A_i.

A Figura 5.3 ilustra o esquema de seleção de atributos, a partir dos grupos mostrados na Figura 5.2(d). A metáfora visual adotada, usando nuvens de palavras, permite visualizar os atributos mais relevantes de cada agrupamento, de forma prática e interativa.

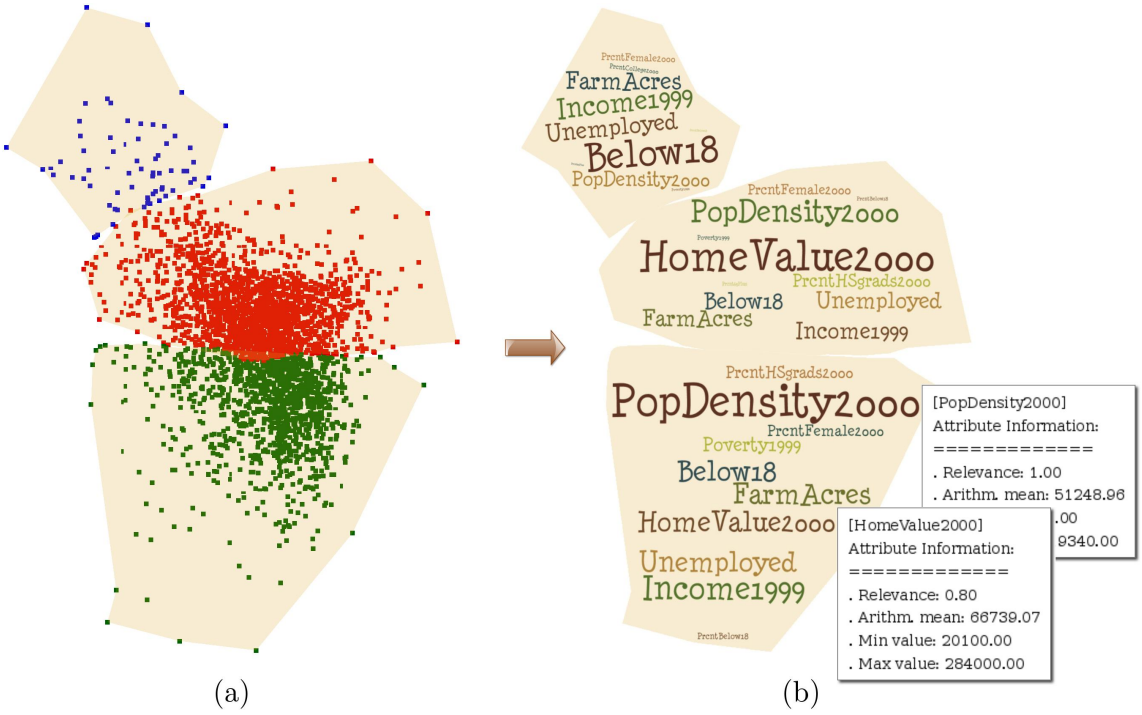
Nesta abordagem, os agrupamentos obtidos são preenchidos por uma superfície gerada pela triangulação de *Delaunay* (Delaunay, 1934) com valor alfa definido. A noção de valor alfa é derivada do trabalho de Edelsbrunner et al. (1983) sobre "*alpha shapes*". Ao contrário do fecho convexo, uma superfície alfa compreende regiões não convexas, sendo capaz de contornar melhor as regiões dos grupos, evitando quase que completamente a sobreposição de superfícies (Figura 5.3(a)).

As nuvens de palavras mostradas na Figura 5.3(b) foram construídas com base no trabalho de Paulovich et al. (2012), porém com uma modificação para permitir que as palavras ocupem qualquer posição dentro da nuvem (qualquer ângulo), não somente horizontal e vertical. O algoritmo utilizado para construir as nuvens toma como entrada dois argumentos: uma região poligonal e uma sequência numérica indicando o grau de relevân-

cia de cada palavra. A região poligonal corresponde à superfície alfa de cada agrupamento e a sequência numérica é dada pelos valores de π (Passo 3 do Algoritmo 5.1).

Quando o Algoritmo 5.1 é aplicado sobre as matrizes de dados de cada grupo A_i, usando a transposta de A_i como entrada, os valores de π, calculados no Passo 3 do algoritmo, indicam o grau de relevância de cada atributo, ou seja, quanto maior o valor de π, mais relevante é o atributo e, consequentemente, maior será o tamanho da palavra que o representa na nuvem.

Um vídeo exemplificando o funcionamento da CSM encontra-se disponível em `http://sites.google.com/site/paulojoiafilho/publications`.



(a) (b)

Figura 5.3: *Layout* gerado com o método proposto: (a) delimitação da região dos grupos e (b) nuvens de palavras mostrando os atributos de cada agrupamento, clicando sobre um determinado atributo é possível obter informações acerca dele.

5.3 Resultados Experimentais e Comparações

Esta seção apresenta um conjunto completo de experimentos, com a finalidade de validar a técnica proposta, mostrando seu desempenho em diferentes cenários.

Todos os resultados apresentados neste capítulo foram produzidos por um microcomputador portátil Asus Intel® Core™ i7, CPU de 2 GHz, placa NVIDIA® Geforce GTX-460M e 16 GB de memória RAM, em ambiente Linux 64-bits. CSM foi implementada em C, utilizando a biblioteca SVDLIBC (Rohde, 2002) para calcular o SVD truncado.

Os experimentos foram divididos em três categorias: amostragem, agrupamento e seleção de atributos, de modo a abranger os principais módulos da técnica. Métodos bem

estabelecidos em cada uma dessas categorias foram utilizados nas comparações. Conjuntos de dados tanto reais como sintéticos foram empregados, variando consideravelmente o número de instâncias, atributos e classes, conforme detalhado na Tabela 5.1. Note que, as classes são subdivisões das instâncias presentes originalmente nos dados, usadas exclusivamente para validação dos experimentos, já que a CSM pode organizar instâncias em grupos de forma não supervisionada.

Tabela 5.1: Conjuntos de dados utilizados nos experimentos da CSM, da esquerda para a direita as colunas correspondem ao nome do conjunto de dados, total de instâncias, dimensão (número de atributos), número de classes, tipo de dado (real ou sintético) e origem dos dados.

Nome	Instâncias	Dimensão	Classes	Tipo	Origem
Ad-8	800	12	8	Sintético	[a]
Ad-15	1.500	10	15	Sintético	[a]
Image segmentation	2.100	19	7	Real	[b]
US counties	3.028	13	3	Real	[c]
US counties-ad	3.028	13	3	Sintético	[a]
Ad-10	10.000	6	4	Sintético	[a]
EEG eye state	14.980	15	2	Real	[b]
Shuttle	43.500	9	7	Real	[b]
Ad-100	100.000	5	10	Sintético	[a]
Fibers	250.000	30	9	Real	[d]

[a] Dados gerados artificialmente de acordo com Guyon et al. (2004). Nota: no conjunto de dados *Ad-100* foram adicionados *outliers* com uso de sinais aleatórios (Peebles, 2001).

[b] Bache e Lichman (2013)

[c] Seo e Shneiderman (2004)

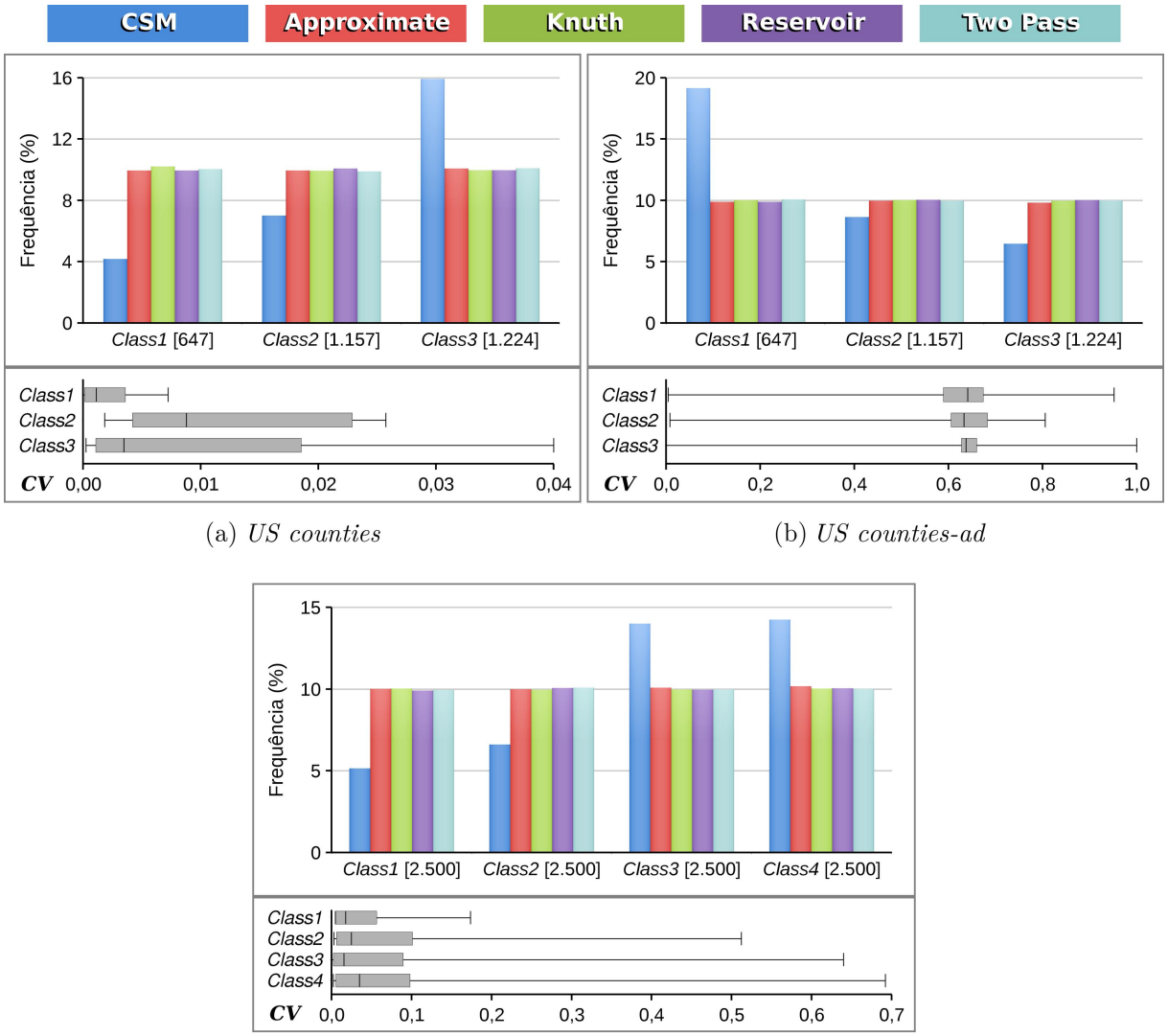
[d] Schneider (2009)

5.3.1 Atestando a Qualidade das Amostras

Para avaliar o mecanismo de amostragem desenvolvido e atestar a qualidade das amostras selecionadas, o *Column Selection Method* (CSM) foi comparado contra quatro mecanismos de amostragem conhecidos: *Reservoir* (Vitter, 1985), *Knuth* (Knuth, 1997), *Two Pass* (Tillé, 2006), e *Approximate Random Sampling Algorithm* (Tillé, 2006).

A Figura 5.4 exibe o resultado de todos os mecanismos de amostragem para três diferentes tipos de conjuntos de dados: classes não uniformes com baixa variação, classes não uniformes com alta variação e classes uniformes com baixa variação. O objetivo deste experimento é evidenciar a capacidade da CSM de selecionar instâncias representativas com base na variabilidade dos dados.

Os histogramas de frequência apresentados na Figura 5.4 correspondem ao percentual médio de instâncias selecionadas em cada classe, após 100 (cem) execuções sucessivas de cada algoritmo, exceto na CSM cujo processo é determinístico. Neste caso, os dados



(a) *US counties* (b) *US counties-ad*

(c) *Ad-10*

Figura 5.4: Histograma de frequência x coeficiente de variação (CV) para três diferentes tipos de conjuntos de dados: (a) *US counties*, classes não uniformes com baixa variação; (b) *US counties-ad*, classes não uniformes com alta variação; e (c) *Ad-10*, classes uniformes com baixa variação.

foram obtidos a partir de uma única execução. O número total de amostras selecionadas foi fixado em 10% do total de instâncias do conjunto de dados, em todas as situações. O número original de instâncias em cada classe é mostrado junto ao rótulo da classe no eixo horizontal, enquanto que a variabilidade em cada classe é apresentada como um gráfico de caixas horizontais na parte inferior de cada subfigura. A variabilidade foi calculada com base no coeficiente de variação, indicado por CV (ver Definição 2.4).

A fim de construir cada caixa horizontal da Figura 5.4 (parte inferior), o cálculo do CV em cada classe foi realizado por atributo. Vale destacar que a abordagem proposta é a única capaz de recuperar instâncias de acordo com a variabilidade dos dados, ou seja,

quanto maior a variabilidade em uma classe, mais amostras a CSM tende a selecionar naquela classe, em contraste aos mecanismos de amostragem comparados, os quais distribuem as amostras proporcionalmente entre as classes. Esta característica transforma a CSM em um mecanismo de amostragem apto a manipular conjuntos de dados contendo classes com poucos elementos, dados com alta variabilidade e até mesmo discrepâncias nos dados (*outliers*).

Para confirmar a efetividade da CSM como ferramenta de amostragem, considere o experimento mostrado na Figura 5.5. Para realizar este experimento foram escolhidos três conjuntos de dados dentre os listados na Tabela 5.1: *Shuttle*, *US counties* e *Ad-100*. Eles foram escolhidos por serem desbalanceados quanto à distribuição de instâncias por classe. A Tabela 5.2 apresenta os detalhes de cada um deles.

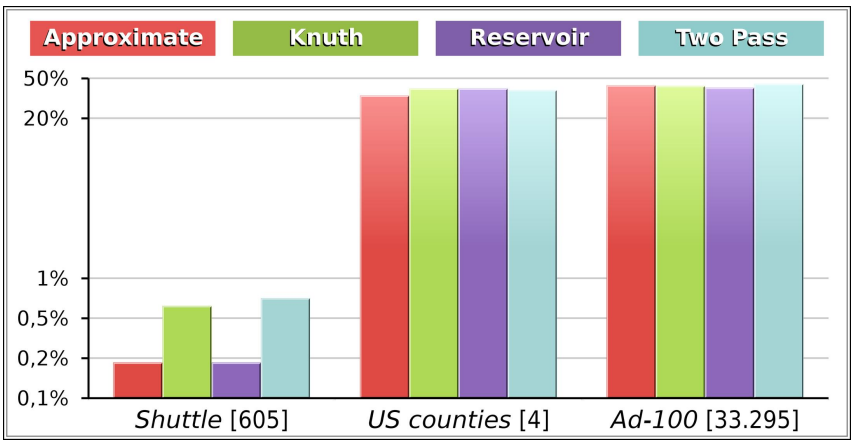


Figura 5.5: Detecção de classes pelas técnicas de amostragem. O eixo y representa o percentual de vezes que cada técnica conseguiu recuperar ao menos uma instância em cada classe, após 1.000 tentativas.

Tabela 5.2: Distribuição de instâncias por classe para os conjuntos de dados utilizados no experimento de detecção de classes mostrado na Figura 5.5.

Conjunto de Dados	**Classes / Instâncias por Classe**
Shuttle	*BpvClose* [6], *BpvOpen* [11], *Bypass* [2.458], *FpvClose* [37], *FpvOpen* [132], *High* [6.748], *RadFlow* [34.108]
US counties	*Class1* [647], *Class2* [1.157], *Class3* [1.224]
Ad-100	*Class1* [8.231], *Class2* [2], *Class3* [15.640], *Class4* [4], *Class5* [70], *Class6* [61.280], *Class7* [300], *Class8* [7.049], *Class9* [2.700], *Class10* [4.724]

O experimento da Figura 5.5 mostra a dificuldade que as técnicas de amostragem aleatória apresentam ao tentar discriminar instâncias em classes com poucos elementos.

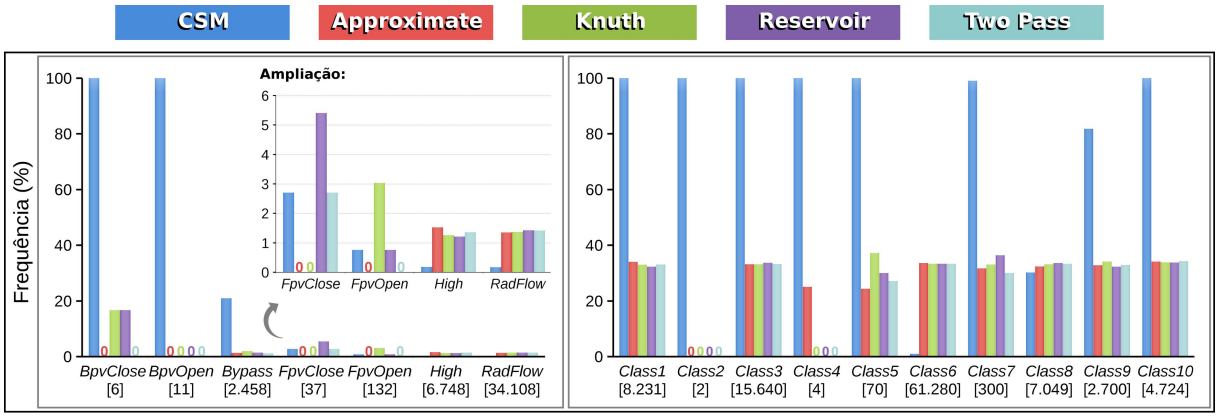
Os valores que aparecem no eixo horizontal da figura, junto ao nome do conjunto de dados, correspondem ao número mínimo de amostras necessárias para a CSM selecionar ao menos uma instância representativa em cada classe. Usando esses valores como parâmetro de entrada para os outros quatro métodos, isto é, como o número de amostras a ser recuperada em cada um deles, computamos o percentual médio de vezes que cada mecanismo de amostragem conseguiu selecionar ao menos uma instância por classe, mas, desta vez em 1.000 (mil) execuções sucessivas, visando aumentar as possibilidades de sucesso.

Considere, por exemplo, o conjunto de dados *Shuttle* (Figura 5.5 – barras verticais à esquerda). Pela Tabela 5.2 é fácil ver que este conjunto de dados é altamente desbalanceado, com classes que vão de 6 a 34.108 instâncias. Note que os métodos de amostragem comparados: *Approximate*, *Knuth*, *Reservoir* e *Two Pass* obtiveram êxito na tarefa em menos de 1% das vezes, mesmo em mil tentativas. Mais precisamente, acertaram apenas 2, 6, 2 e 7 vezes em mil, respectivamente.

O conjunto de dados *US counties* (Figura 5.5 – barras verticais centrais) possui três classes melhor balanceadas. Neste conjunto, a CSM conseguiu amostrar todas as classes usando apenas quatro instâncias como entrada. No entanto, usando quatro instâncias como entrada para os demais algoritmos, eles obtiveram êxito em menos de 50% das vezes. Já o conjunto de dados *Ad-100* (barras verticais à direita) também é desbalanceado. Além disso, possui classes com alta variabilidade, fazendo com que a CSM selecione muitas instâncias, 33.295, para conseguir amostrar todas as classes. Note que, mesmo usando um número muito grande de instâncias como entrada, os métodos comparados, novamente, obtiveram êxito em menos de 50% das tentativas. Experimentos como este garantem ao *Column Selection Method* (CSM) uma grande vantagem em relação aos demais métodos, já que consegue selecionar instâncias representativas em todas as classes do conjunto de dados com eficácia, de forma determinística.

O experimento da Figura 5.6 utiliza os conjuntos de dados *Shuttle* e *Ad-100*, que são altamente desbalanceados (ver Tabela 5.2), para mostrar a frequência de instâncias recuperadas em cada classe quando os algoritmos de amostragem são aplicados. O número total de instâncias recuperadas em cada método foi estabelecido de acordo com o número mínimo de instâncias necessárias para a CSM encontrar ao menos um representante em cada classe, ou seja, 605 para o conjunto de dados *Shuttle* e 33.295 para o conjunto de dados *Ad-100*, similar ao experimento anterior. Neste gráfico, os valores que figuram no eixo x correspondem ao nome da classe e total de instâncias que contém. O eixo y exibe a frequência de instâncias selecionadas em cada classe, em porcentagem. No caso da CSM, apenas uma execução foi realizada. Nas técnicas com comportamento aleatório foi admitida a "melhor" resposta em cem execuções sucessivas, tal que a melhor resposta é aquela em que mais classes são amostradas.

O conjunto de dados *Shuttle* (Figura 5.6(a)) tem maior variabilidade nas classes *BpvClose*, *BpvOpen* e *Bypass*, onde a CSM recuperou grande parte das instâncias representa-



(a) *Shuttle* (b) *Ad-100*

Figura 5.6: Detecção de classes em conjuntos de dados altamente desbalanceados contendo *outliers*.

tivas. Particularmente nas duas primeiras, onde o número de representantes é pequeno, 6 e 11 respectivamente, as técnicas comparadas falharam em quase todas as tentativas.

O conjunto de dados sintético *Ad-100* foi construído de modo a garantir classes com grande variabilidade de dados. Além disso, *outliers* (ver Definição 2.3) foram adicionados nas cinco primeiras classes deste conjunto de dados, isto é, nas classes rotuladas como *Class1*, *Class2*, ..., *Class5*. Embora detecção de *outliers* não seja o foco principal da CSM, note que nestas classes, cuja variabilidade é ainda maior devido à presença dos *outliers*, o mecanismo de amostragem proposto tenta recuperar tantas instâncias quanto possível (Figura 5.6(b)). Já nas técnicas comparadas não existe garantia alguma de que os *outliers* foram selecionados.

Por fim, observando a Figura 5.6, é fácil ver que nenhuma das técnicas comparadas com a CSM conseguiu amostrar todas as classes em 100 (cem) tentativas, falhando em pelo menos uma classe para ambos os conjuntos de dados, evidenciando mais uma vez a importância de uma técnica de amostragem baseada na variabilidade dos dados em oposição às baseadas em distribuição aleatória.

5.3.2 Atestando a Qualidade dos Agrupamentos

Para atestar a qualidade dos grupos obtidos com a CSM, cinco técnicas não supervisionadas especializadas em detecção de agrupamentos, implementadas no pacote WEKA (Hall et al., 2009), foram empregadas nas comparações: *Expectation Maximization* (EM) (Dempster et al., 1977), *Farthest First* (FF) (Hochbaum e Shmoys, 1985), *Make Density Based Cluster* (MDBC)(Witten et al., 2011), *Simple K-Means* (SKM) (Macqueen, 1967) e *XMeans* (XM) (Pelleg e Moore, 2000).

Uma medida quantitativa empregada na avaliação de agrupamentos é a medida da silhueta (Definição 2.8). Neste experimento, a medida da silhueta foi computada como a

média de cem execuções sucessivas para cada algoritmo, já que a maioria deles apresenta comportamento aleatório. No caso da CSM, cujo processo é determinístico, a silhueta foi computada variando o número de instâncias representativas de 5% a 50% do total de instâncias do conjunto de dados. Em todos os casos, o número de agrupamentos foi fixado igual ao número de classes do conjunto de dados original, para facilitar as análises.

A Figura 5.7 mostra o resultado da silhueta entre a CSM e as cinco técnicas mencionadas, usando vários dos conjuntos de dados apresentados na Tabela 5.1. Nesta figura, os gráficos de barras horizontais mostram o resultado da silhueta para cada conjunto de dados, individualmente, enquanto que o gráfico de caixas verticais à esquerda, sintetiza os resultados dos oito conjuntos de dados testados. Observe que, na maioria dos casos, a abordagem proposta supera os métodos comparados, e que na média, seu resultado é superior a todos eles.

Note que, ao variar o número de instâncias representativas, a resposta da CSM torna-se dependente do mecanismo de amostragem de dados. Portanto, se o resultado da silhueta é, em média, melhor que o resultado dos outros métodos, conclui-se que o mecanismo de amostragem está, de fato, contribuindo significativamente para o processo de agrupamento de dados. Além disso, se a CSM é melhor que o método *Expectation Maximization* (EM), o qual é usado para agrupar as instâncias representativas em uma de suas etapas, fica evidente, mais uma vez, que a sua eficácia está associada ao mecanismo de seleção de amostras representativas.

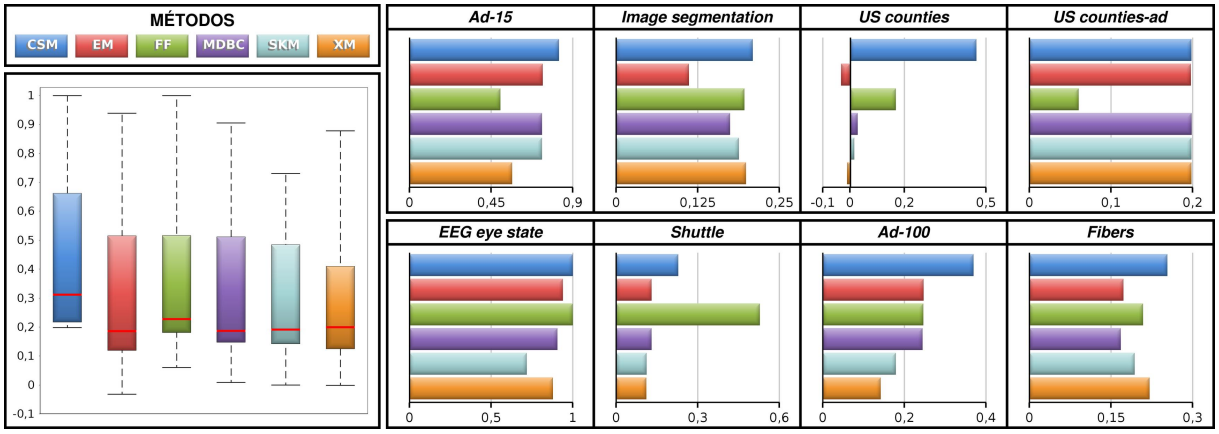


Figura 5.7: Medida da silhueta. O método proposto, CSM (em azul), apresenta melhores resultados que os métodos *Expectation Maximization* (EM), *Farthest First* (FF), *Make Density Based Cluster* (MDBC), *Simple K-Means* (SKM) e *XMeans* (XM).

Para concluir a seção de comparação dos resultados dos agrupamentos obtidos com a CSM, considere o experimento mostrado na Figura 5.8, tal que emprega matriz de confusão (Definição 2.9) e coeficiente de acurácia, indicado por ACC (Definição 2.10) para estimar a qualidade dos agrupamentos obtidos. Neste caso, a comparação foi realizada contra dois dos melhores métodos de detecção de agrupamentos não supervisionados testados: o EM

e o MDBC, utilizando o conjunto de dados *Image segmentation* que originalmente possui sete classes, definindo, portanto, sete grupos, nem todos bem separados, conforme pode ser observado na Figura 5.8(a). Esta configuração oferece certa dificuldade aos algoritmos de detecção de agrupamento.

Apesar da dificuldade em obter exatamente os mesmos grupos do conjunto de dados original *Image segmentation*, note que o método proposto (Figura 5.8(b)) é o que mais se aproxima: matriz de confusão com diagonal dominante e maior coeficiente de acurácia, ACC $\approx 0,67$, contra ACC $\approx 0,58$ do EM (Figura 5.8(c)) e ACC $\approx 0,48$ do método MDBC (Figura 5.8(d)). A CSM utilizou 20% das instâncias como amostras representativas para agrupar os dados.

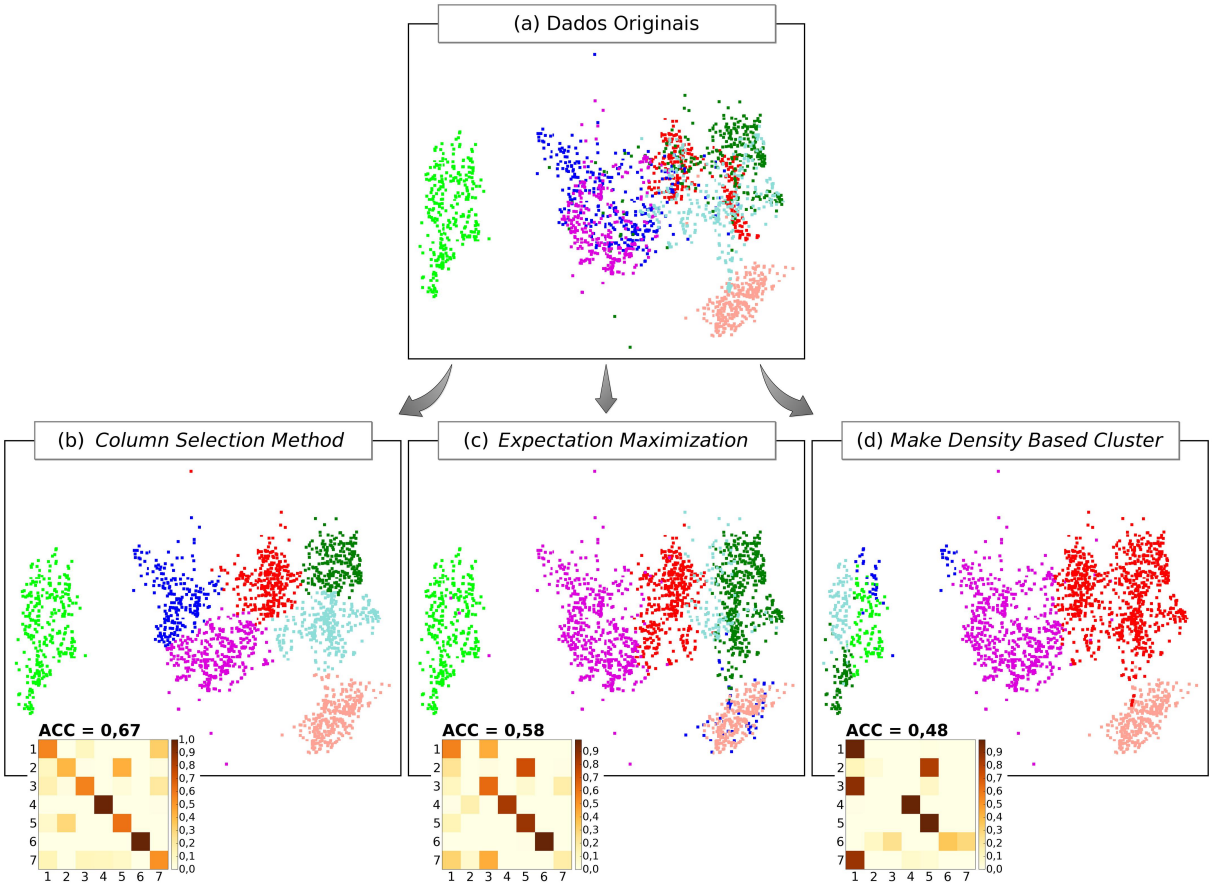


Figura 5.8: Identificação de agrupamentos no conjunto de dados *Image segmentation*: (a) a partir da classificação original dos dados; (b) aplicando o *Column Selection Method* (CSM); (c) aplicando o método *Expectation Maximization* (EM) e (d) através do método *Make Density Based Cluster* (MDBC).

5.3.3 Atestando a Qualidade dos Atributos Selecionados

Embora seleção de atributos não seja o foco principal desta tese, o experimento apresentado nesta seção visa mostrar que a CSM também é competitiva como técnica de seleção de atributos não supervisionada. Para avaliar os atributos selecionados, os resultados obti-

dos foram comparados contra cinco técnicas de seleção não supervisionadas, disponíveis no pacote WEKA: *Classifier Subset Evaluation* (CSE) (Han et al., 2011), *Filtered Attribute Evaluation* (FAE) (Witten et al., 2011), *Filtered Subset Evaluation* (FSE) (Han et al., 2011), *Latent Semantic Analysis* (LSA) (Landauer et al., 1998) e *Wrapper Subset Evaluation* (WSE) (Kohavi e John, 1997).

A qualidade do esquema de seleção de atributos foi medida usando a divergência de *Kullback-Leibler* (KL) (Kullback e Leibler, 1951), definida como:

$$D(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}\,, \tag{5.5}$$

onde p e q são distribuições de probabilidade.

Tanto na CSM quanto nas cinco técnicas comparadas, a divergência KL foi calculada em cada agrupamento de dados e a divergência final pela média desses valores. Além disso, como cada técnica pode selecionar quantidades diferentes de atributos, fixamos o número máximo em 50% do total de atributos do conjunto de dados, a fim de garantir comparações mais coerentes.

No contexto deste trabalho, assumimos que as instâncias em cada agrupamento C_j seguem uma distribuição normal multivariada, isto é:

$$p_j(x) = \frac{1}{\sqrt{(2\pi)^k |\Sigma_j|}} \exp\left\{ -\frac{1}{2}(x - \mu_j)^\top \Sigma_j^{-1} (x - \mu_j) \right\}\,, \tag{5.6}$$

onde μ_j é o vetor de médias de C_j, Σ_j é a matriz de covariância de C_j, k é a dimensão dos dados e $|\Sigma_j|$ é o determinante de Σ_j. Após selecionar os atributos, as instâncias em cada grupo ainda deveriam seguir uma distribuição normal q_j, com vetor de médias e matriz de covariância dado pelos atributos selecionados (Ross, 2014). A similaridade entre p_j e q_j em cada grupo é medida pela divergência KL, definida na Equação (5.5). Se os atributos são selecionados de forma apropriada, então as distribuições p_j e q_j tendem a ser similares, resultando em valores de divergência muito próximos de zero.

A Figura 5.9 apresenta o resultado da divergência KL, de acordo com a discussão acima, para oito conjuntos de dados, listados na Tabela 5.1. Nesta figura, os gráficos de barras horizontais mostram o resultado da divergência para cada conjunto de dados em estudo[2]. Lembrando que os cálculos foram efetuados por agrupamento, portanto, cada barra horizontal da figura corresponde à média aritmética da divergência calculada nos agrupamentos. Já o gráfico de caixas verticais à esquerda representa o panorama geral da

[2] Para contornar o problema de falta de memória apresentado pelo pacote WEKA ao processar grandes conjuntos de dados, o tamanho do *heap* na máquina virtual Java foi aumentado para 3,5 GB. Entretanto, as técnicas LSA e WSE falharam ao processar o conjunto de dados *Fibers*, por motivos desconhecidos.

divergência para cada técnica, utilizando os resultados dos oito conjuntos de dados como entrada.

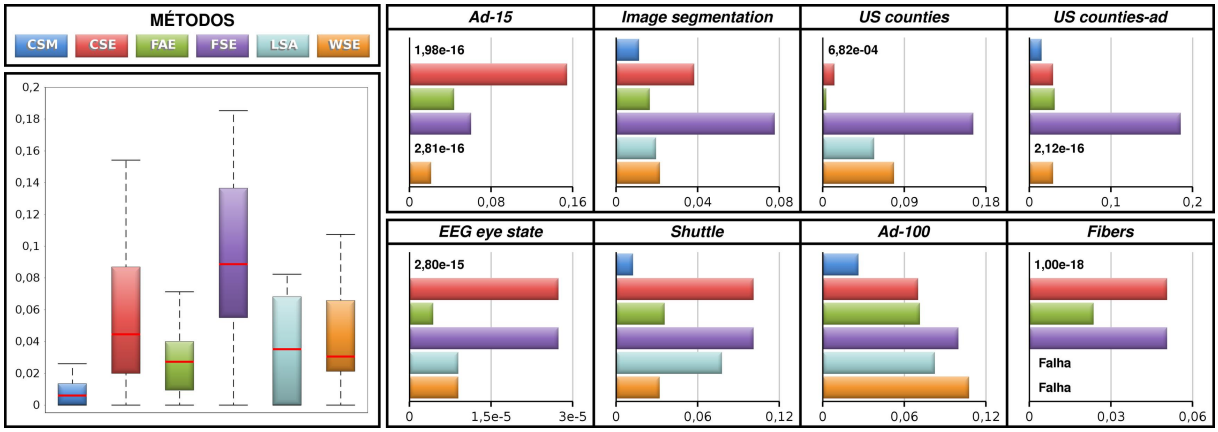


Figura 5.9: Divergência de *Kullback-Leibler*. O método proposto, CSM (em azul), resulta em menores valores de divergência que os métodos *Classifier Subset Evaluation* (CSE), *Filtered Attribute Evaluation* (FAE), *Filtered Subset Evaluation* (FSE), *Latent Semantic Analysis* (LSA) e *Wrapper Subset Evaluation* (WSE).

Note que os atributos selecionados pela CSM (Figura 5.9), resultam em menores valores de divergência, isto significa que eles descrevem melhor o modelo original dos dados. Em outras palavras, a técnica proposta captura a essência dos dados, revelando os atributos mais importantes daquele grupo de instâncias. Esta característica qualifica a CSM como uma técnica de visualização apta a selecionar atributos em qualquer agrupamento de dados, com precisão.

5.3.4 Tempos Computacionais da CSM

Além da precisão ao selecionar instâncias representativas, identificar agrupamentos e selecionar os atributos mais relevantes de um subconjunto de dados, o algoritmo *Column Selection Method* é competitivo em termos de eficiência computacional. Esta seção apresenta os tempos computacionais necessários para selecionar instâncias representativas, bem como atributos, a partir de diferentes conjuntos de dados.

Na Tabela 5.3, os valores na coluna central correspondem ao tempo médio de dez execuções sucessivas para selecionar instâncias representativas, fazendo variar o número de instâncias selecionadas no intervalo 5%, 10%, ..., até 50% do conjunto de dados. Já os valores na coluna da direita correspondem ao tempo total necessário para selecionar atributos em todos os agrupamentos, o qual foi fixado igual ao número de classes do conjunto de dados. Note que mesmo para grandes conjuntos de dados como o *Fibers*, os tempos computacionais são aceitáveis, mostrando que a CSM é viável para aplicações interativas. Nenhum paralelismo foi empregado na implementação.

Tabela 5.3: Tempo de execução, em segundos, para seleção de instâncias representativas e atributos.

Conjunto de Dados	Seleção de Instâncias	Seleção de Atributos
Ad-15	0,004	0,007
Image segmentation	0,009	0,009
US counties	0,009	0,012
US counties-ad	0,008	0,008
EEG eye state	0,044	0,035
Shuttle	0,069	0,053
Ad-100	0,090	0,078
Fibers	2,104	1,967

5.4 Um Estudo de Caso: Modelo de Vendas por Atacado

Esta seção apresenta uma aplicação prática da técnica de visualização desenvolvida. Para tanto, considere o *Wholesale Customers Data Set* (WCDS) (Bache e Lichman, 2013), o qual é um conjunto de dados multidimensionais contendo informações sobre os gastos anuais de 440 clientes de um distribuidor por atacado. O conjunto de dados possui oito atributos, seis numéricos e dois categóricos. Os atributos numéricos representam o valor gasto por cliente nos seguintes produtos: *'Fresh'*, *'Milk'*, *'Grocery'*, *'Frozen'*, *'Detergents'*, e *'Delicatessen'*. Os atributos categóricos indicam o tipo de cliente, o qual pode ser *Hotel/Restaurant/Catering* (abreviado como *HoReCa*) ou *Retail*; e a região onde cada cliente está localizado, a qual pode ser *Lisbon*, *Porto* ou *Other*.

O propósito deste estudo é aplicar o *pipeline* de visualização desenvolvido para responder questões como: existe uma distinção clara entre os clientes quanto ao padrão de consumo? Em caso afirmativo, quais são os produtos que melhor caracterizam cada tipo de cliente? A fim de responder tais questões, o *pipeline* de visualização foi aplicado sobre os atributos numéricos do conjunto de dados WCDS.

A Figura 5.10(a) mostra o *layout* produzido pela CSM ao utilizar 20% do total de instâncias como amostras representativas para encontrar dois agrupamentos de dados, correspondentes aos dois tipos de cliente: *HoReCa* e *Retail*.

A matriz de confusão (Tabela 5.4) pode informar quantos clientes de cada tipo existem nos agrupamentos, conforme segue.

Tabela 5.4: Matriz de confusão referente aos grupos mostrados na Figura 5.10(a).

		Predito↓	
		Rosa	**Verde**
Real↓	**Rosa**	292	6
	Verde	56	86

Ou, de forma nominal:

- Grupo Rosa $\Rightarrow$ *HoReCa* = 292; *Retail* = 6.
- Grupo Verde $\Rightarrow$ *HoReCa* = 56; *Retail* = 86.

A matriz de confusão anterior revela 86% de acurácia na classificação, portanto, a CSM separa claramente os dois tipos de cliente existentes.

A Figura 5.10(b) mostra os atributos mais relevantes de cada agrupamento obtido. Note que as palavras *'Grocery'* e *'Milk'* se destacam no grupo verde, logo estes são os atributos mais relevantes que caracterizam clientes *Retail*. No grupo rosa, o atributo mais relevante é *'Fresh'*, seguido por *'Milk'*. Portanto, *'Fresh'* e *'Milk'* são os atributos mais relevantes que caracterizam clientes do tipo *HoReCa*. Desta forma, ficam respondidas as questões originadas anteriormente, confirmando na prática, a utilidade da CSM como ferramenta de visualização.

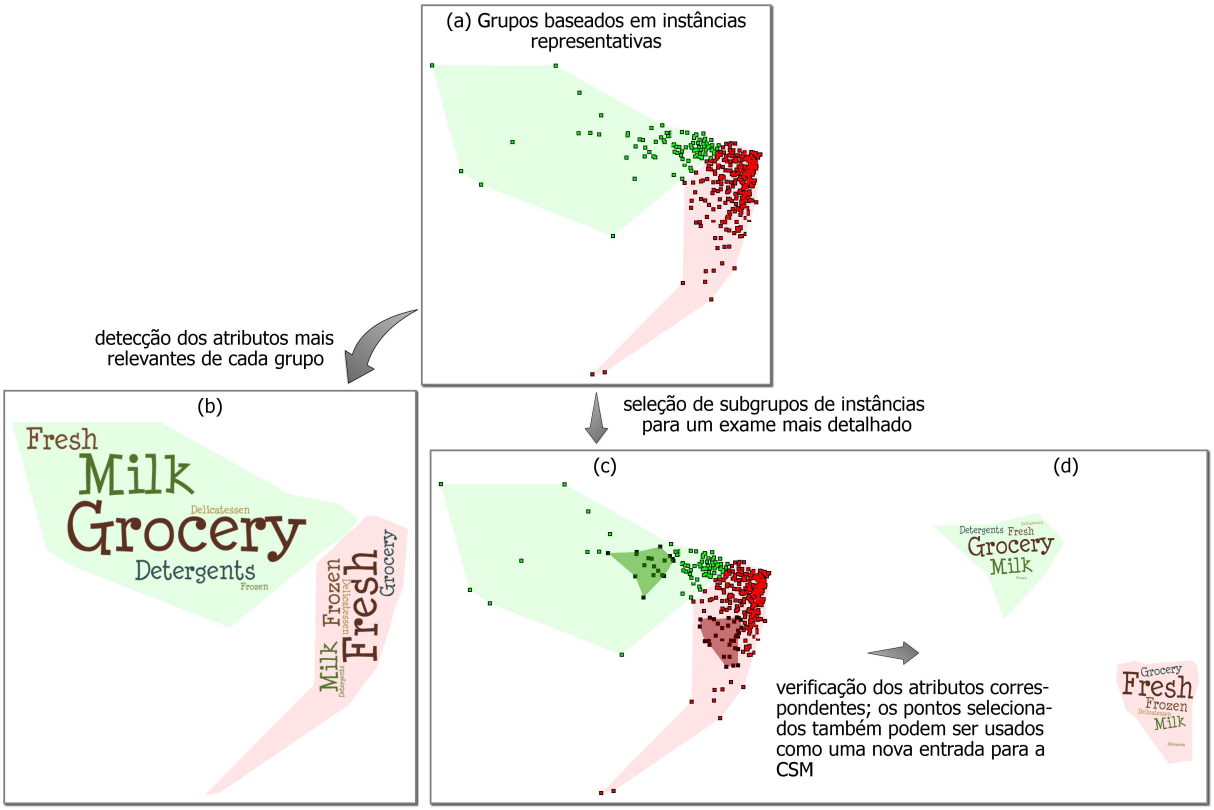


Figura 5.10: Análise de preferência por produtos segundo os tipos de cliente.

As Figuras 5.10(c) e 5.10(d) ilustram alguns passos interativos da técnica. A Figura 5.10(c), por exemplo, mostra dois subgrupos de instâncias selecionados pelo usuário, a partir dos grupos da Figura 5.10(a). A Figura 5.10(d), por sua vez, mostra o resultado da análise de atributos para cada subgrupo. Note que os atributos relevantes de

cada subgrupo, em geral, são os mesmos do "grupo pai", confirmando de fato que os pares de atributo (*'Grocery'*, *'Milk'*) e (*'Fresh'*, *'Milk'*) caracterizam clientes do tipo *Retail* e *HoReCa*, respectivamente.

A análise visual apresentada nesta seção é sustentada pela sólida formulação matemática da CSM. Esse estudo de caso, embora simples, deixa claro a importância desta técnica de visualização.

5.5 Considerações Finais

A técnica de visualização de dados multidimensionais discutida neste capítulo, *Column Selection Method* (CSM), é capaz de selecionar instâncias representativas, identificar agrupamentos e selecionar atributos a partir de qualquer subconjunto de dados ou agrupamento. Em contraste à maioria das técnicas existentes, a CSM é capaz de lidar com dados desbalanceados e atípicos, propriedade comprovada através de inúmeros testes e validações. Alguns aspectos relevantes sobre esta técnica são discutidos abaixo.

Selecionar instâncias com base em variabilidade garante que classes com poucos representantes sejam amostradas. Em contrapartida, este procedimento tende a aumentar o peso destas classes. Por exemplo, considere o caso extremo onde o conjunto de dados em estudo é altamente desbalanceado e a classe que contém o maior número de elementos tem alta variabilidade face às demais classes, cuja variabilidade é baixa. Neste caso, é possível que o mecanismo de amostragem seja "aprisionado" pela classe com maior variabilidade, "saltando" para as demais somente depois de amostrar uma grande quantidade de instâncias naquela classe. Com muitas instâncias representativas selecionadas, o *layout* da projeção tende a ficar congestionado, dificultando a identificação de grupos. Portanto, situações como esta devem ser cuidadosamente avaliadas.

Técnicas de projeção multidimensional são propensas a introduzir falsos vizinhos ou vizinhos ausentes no espaço visual (Martins et al., 2014), degradando a qualidade da vizinhança. Dependendo dos dados, este efeito pode dificultar a identificação de grupos no espaço visual. Este é um problema que merece ser investigado com mais detalhes.

Em conjuntos de dados contendo muitas instâncias e muitos atributos ($m \approx n$), a estimativa do parâmetro k requer uma investigação mais detalhada. Neste caso, o valor estimado pela Equação (5.4), $k = \min\{m, n\} \backslash 2 + 1$, pode resultar em valores muito altos de k, comprometendo a etapa de seleção de instâncias representativas.

Na CSM, os grupos são definidos com base no espaço visual. Esta característica tem muitos benefícios, como garantir que os grupos não fiquem fragmentados durante a visualização, mas pode produzir alguns efeitos visuais indesejados. Por exemplo, é intuitivo pensar que o maior número de instâncias está concentrado no grupo com a maior região poligonal, o que nem sempre é verdade, ou seja, a área do polígono não é proporcional ao número de instâncias que ele contém.

Em relação aos atributos, é possível que um polígono não seja suficientemente grande para acomodar todas as palavras (atributos relevantes) daquele grupo. Além disso, a comparação visual dos atributos mais relevantes só faz sentido se executada dentro de um mesmo grupo. Comparação entre diferentes grupos deveria ser evitada. Para amenizar tais efeitos, uma possível solução consiste em pós-processar a saída da projeção a fim de reescalar regiões no espaço visual segundo a sua densidade e relevância.

Por fim, vale destacar que a CSM é uma técnica de fácil implementação, requerendo apenas uma biblioteca para resolver o SVD e um método de ordenação de dados. Este aspecto é particularmente interessante no contexto de seleção de atributos, onde os algoritmos existentes, em geral, têm alta complexidade computacional.

Capítulo

6

Projeção e Busca por Similaridade Usando Métricas Específicas

Busca por similaridade é útil quando existe um padrão de interesse nos dados, a partir do qual pretende-se encontrar padrões similares com base em alguma medida de similaridade ou métrica. Esta tarefa costuma ser realizada em diferentes domínios, como séries temporais, imagens e coleções de documentos (Maimon e Rokach, 2010).

Técnicas de projeção podem auxiliar nesta tarefa, já que são muito flexíveis com respeito à medida de similaridade utilizada como, por exemplo, distinguir classes de objetos. Esta ação implica minimizar a dissimilaridade entre objetos da mesma classe e maximizá-la para objetos de classes distintas. Para que isto seja possível, a medida empregada deve reconhecer de alguma forma as classes de objetos, ou melhor, deve conter informações específicas de cada classe.

Este capítulo apresenta uma nova técnica de projeção denominada *Class-Specific Multidimensional Projection* (CSMP), a qual utiliza uma família de métricas baseada em classes para projetar dados.

A CSMP fundamenta-se em outra técnica de projeção conhecida, a LSP (Paulovich et al., 2008) que, embora eficaz, não faz uso de mecanismos para comparar dados de diferentes classes. A nova abordagem preserva as características favoráveis da LSP, enquanto aumenta sua precisão ao comparar dados multidimensionais.

A família de métricas baseada em classes é utilizada neste trabalho para recuperar imagens com base em conteúdo. Recuperação de imagens com base em conteúdo (CBIR) tem um papel importante na organização e consulta de grandes coleções de imagens. Muitas técnicas têm sido propostas para este fim (Datta et al., 2008). Técnicas de CBIR com base em projeção multidimensional têm se tornado uma alternativa promissora (Eler

et al., 2009), já que permitem executar múltiplas consultas sem refazer o mapeamento, ao passo que ainda tornam possível a interação do usuário no processo, de modo a aumentar a precisão das respostas.

Embora este capítulo aborde questões relacionadas à recuperação de imagens com base em conteúdo, a CSMP pode ser aplicada em outros contextos envolvendo coleções de músicas, vídeos ou formas geométricas, por exemplo. A construção da família de métricas classes-específicas, no entanto, requer um subconjunto de dados rotulados, isto é, com informações de classe. Este subconjunto é usado para estimar as características (atributos) mais relevantes de cada classe. Coleções de imagens, por sua vez, permitem que elementos de diferentes categorias sejam visualmente selecionados e agrupados com uso de recursos interativos.

Os dados das imagens são obtidos por meio de ferramentas de reconhecimento de padrões, mais especificamente, extração e seleção de características. A família de métricas classes-específicas é construída com base nos melhores atributos de cada classe. Esta modificação na métrica faz com que a CSMP supere outras técnicas de projeção, bem como outros sistemas de CBIR ao recuperar informações.

Parte da contribuição descrita neste capítulo foi publicada em Joia et al. (2012).

6.1 Principais Contribuições

Entre as contribuições deste trabalho destacam-se:

- Projeto e implementação de uma família de métricas baseada em classes para medir a similaridade entre pares de objetos.

- CSMP: uma técnica de projeção multidimensional que utiliza as métricas classes-específicas para comparar dados multidimensionais.

- Um mecanismo para realizar buscas por similaridade em coleções de imagens, a partir da projeção.

6.2 Class-Specific Multidimensional Projection (CSMP)

A técnica proposta compreende seis passos principais, conforme apresentado na Figura 6.1: 1) extração de características; 2) seleção de características; 3) projeção e manipulação dos pontos de controle; 4) construção da família de métricas classes-específicas; 5) projeção multidimensional e 6) seleção dos objetos de consulta. As próximas subseções descrevem estas etapas.

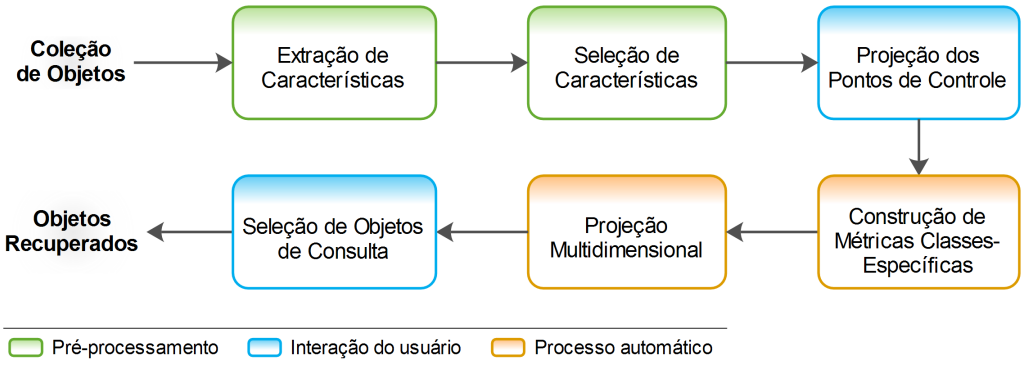


Figura 6.1: *Pipeline* da CSMP.

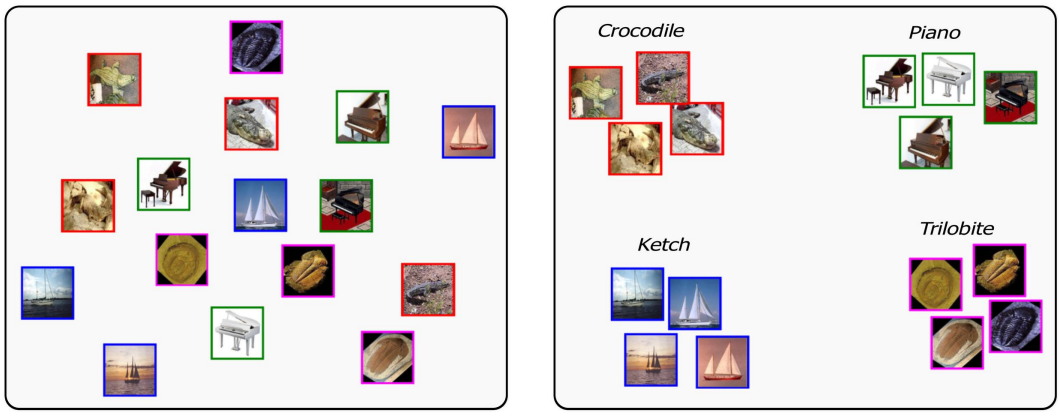
6.2.1 As Etapas da CSMP

Dado um conjunto de imagens $\mathcal{I}$, o primeiro passo da técnica consiste em encontrar o espaço de características de $\mathcal{I}$. Esta etapa é denominada extração de características, que em termos matemáticos corresponde a uma transformação $T : \mathcal{I} \rightarrow \mathbb{R}^m$, onde m é o número de características usadas para representar cada imagem. Os algoritmos empregados neste processo são denominados descritores de características de imagem e, dependendo da coleção de imagens considerada, diferentes descritores podem ser combinados, tais como: descritores de forma, cor, textura, etc. Os descritores, assim como os conjuntos de imagens utilizados nos experimentos, são discutidos na Seção 6.3.

O processo de extração (Passo 1) produz um número elevado de características (ver Tabela 6.2 para detalhes), entretanto, algumas podem ser negligenciadas. Neste caso, um algoritmo de avaliação de subconjunto de características foi aplicado, o *Subset Size Forward Selection* (SSFS) (Witten et al., 2011), o qual reduz bastante a dimensão do espaço inicial. Cerca de 20% das características extraídas são suficientes para a construção da família de métricas classes-específicas (Passo 2). Vale lembrar que o processo de extração e pré-seleção de características é executado uma única vez para cada coleção de imagens, em uma etapa de pré-processamento.

Com o espaço de características definido, o próximo passo requer que o usuário escolha um subconjunto de imagens de controle (ou pontos de controle, no contexto de projeção). Deve-se escolher interativamente, com uso de recursos visuais, imagens de diferentes categorias ou classes. Isto é importante, pois tais imagens irão ajudar a guiar o restante da projeção e distinguir diferentes classes. O usuário pode especificar a classe às quais elas pertencem agrupando interativamente imagens similares no espaço visual, como ilustrado na Figura 6.2 (Passo 3). Recursos visuais e interativos como este constituem um ponto forte desta abordagem.

Uma vez agrupadas as imagens, o sistema interpreta cada grupo como uma classe de imagens. Em outras palavras, o usuário está implicitamente rotulando o subconjunto de imagens de controle de acordo com os grupos formados. Esta etapa caracteriza a CSMP



(a) (b)

Figura 6.2: Projeção dos pontos (ou imagens) de controle: (a) antes da manipulação do usuário; (b) após a intervenção do usuário, tal que instâncias similares são posicionadas próximas umas das outras.

como uma técnica semissupervisionada, já que apenas um pequeno número de instâncias precisam ser rotuladas durante o processo de recuperação de imagens (ver Seção 2.2).

Na sequência, o sistema utiliza as imagens de controle previamente rotuladas para estimar o subconjunto de características que melhor representa cada classe. O algoritmo de classificação / seleção utilizado neste procedimento foi o *Logistic Model Tree* (LMT) (Landwehr et al., 2005), por produzir bons resultados nos testes realizados.

Os subconjuntos de características encontrados com a LMT são usados para definir uma família de métricas classes-específicas (Passo 4), a qual é utilizada no cálculo de similaridades entre as instâncias, para compor a matriz de projeção multidimensional. Em seguida, os dados são projetados (Passo 5). A ideia por trás da família de métricas e como elas são introduzidas na projeção exigem uma discussão mais detalhada, Seções 6.2.2 e 6.2.3, respectivamente.

Após a projeção, qualquer imagem $\alpha \in \mathcal{I}$ é potencialmente uma candidata à imagem de consulta (*query image*) (Passo 6). E neste caso, para recuperar as k imagens mais parecidas com α é suficiente encontrar seus k-vizinhos mais próximos no espaço visual, ou simplesmente selecioná-las a partir do *layout* da projeção (ver Figura 6.4 como exemplo). Uma vez que a projeção permite distinguir diferentes classes de imagem, é possível realizar múltiplas buscas, usando diferentes imagens de consulta, sem a necessidade de reconstruir o mapeamento múltiplas vezes.

6.2.2 Família de Métricas Classes-Específicas

Seja $\mathbb{R}^m$ o espaço de características do conjunto de imagens $\mathcal{I}$. Admitindo-se Q o conjunto de imagens de controle e Q_i um subconjunto de imagens de controle que compartilham o mesmo rótulo (agrupadas pelo usuário), tal que $I_{Q_i} = \{i_1, \ldots, i_r\}$ são os

índices das características que melhor representam as imagens em Q_i (determinadas pelo seletor de características). Então, dada uma imagem $\alpha \in \mathcal{I}$, $\beta \in Q_i$, e $\gamma \in Q_j$ $(i \neq j)$, se α pertence à mesma classe de β, é razoável esperar que

$$d_{Q_i}(\alpha, \beta) \leq d_{Q_j}(\alpha, \gamma),$$

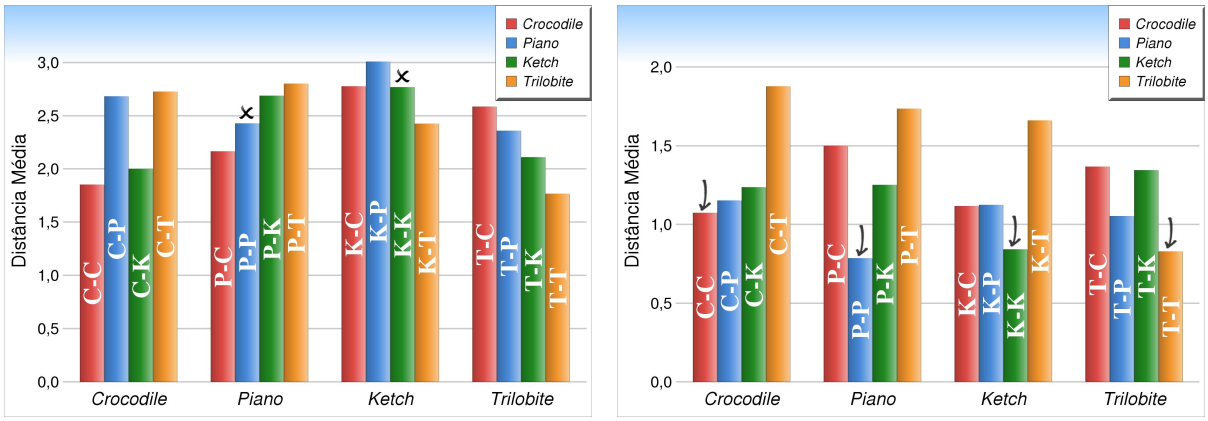
onde a métrica classe-específica d_{Q_i} é definida como

$$d_{Q_i}(\alpha, \beta) = \sum_{j \in I_{Q_i}} (\alpha_j - \beta_j)^2, \tag{6.1}$$

onde α_j indica a j-ésima coordenada (característica) de α (respectivamente β).

A métrica definida pela Equação (6.1) é de fato uma pseudo-métrica (Definição 2.14), uma vez que $d_{Q_i}(x, y) = 0$ não implica que $x = y$; ou, simplesmente, uma *medida de dissimilaridade*. Contudo, por não comprometer a clareza do documento, o termo "métrica" foi admitido neste estudo.

O raciocínio por trás da métrica classe-específica definida na Equação (6.1) revela que se α é uma imagem similar a $\beta \in Q_i$ então I_{Q_i} deveria conter as melhores características para representar α, assim $d_{Q_i}(\alpha, \beta)$ deveria ser pequena. Ao contrário, espera-se uma medida de dissimilaridade maior se uma métrica classe-específica é usada para comparar α com uma imagem não similar γ. Desse modo, usando a métrica classe-específica evita-se comparar características que não representam as imagens apropriadamente, aumentando portanto, a precisão e a confiabilidade da medida de distância. Os gráficos de barras verticais mostrados na Figura 6.3 corroboram esta ideia.



(a) Métrica Euclidiana (b) Métrica Classe-Específica

Figura 6.3: Distância média entre as classes do conjunto de dados *Caltech-4classes* (ver Tabela 6.1), tal que: (a) distância média entre classes obtida com o uso da métrica Euclidiana; (b) distância média obtida com o uso da métrica classe-específica. Note que a distância média entre elementos pertencentes à mesma classe (C-C, P-P, K-K, T-T) é sempre menor quando a métrica classe-específica é usada (indicado pelas setas).

Nas Figuras 6.3(a) e 6.3(b) as barras verticais correspondem à média aritmética das distâncias entre pares de instâncias, calculadas a partir da coleção *Caltech-4classes*, utilizando, respectivamente, a métrica Euclidiana e a métrica classe-específica. Esta coleção de imagens contém quatro classes distintas de imagens, rotuladas como C*rocodile*, P*iano*, K*etch* e T*rilobite* (ver Tabela 6.1 para detalhes sobre os conjuntos de dados). Note que, no primeiro caso, onde a métrica Euclidiana é utilizada (Figura 6.3(a)), a distância média entre instâncias da mesma classe nem sempre é menor que as demais, tais como: P-P (classe P*iano*) e K-K (classe K*etch*) (indicado por um x na parte superior das barras). Em contraste, a métrica classe-específica torna C*rocodile* mais próximo de seus pares, o mesmo vale para P*iano*, K*etch* e T*rilobite*, conforme ilustrado na Figura 6.3(b). Observe que a distância média computada a partir de elementos pertencentes à mesma classe é normalmente menor, isto é, de C*rocodile* a outros C*rocodile* (C-C), de P*iano* a outros P*iano* (P-P), de K*etch* a outros K*etch* (K-K), de T*rilobite* a outros T*rilobite* (T-T) (indicado por uma seta na parte superior das barras).

Como consequência, é imediato notar que a métrica classe-específica permite comparar e identificar objetos similares de forma mais precisa em relação à métrica Euclidiana clássica que, em geral, falha para determinadas classes de objetos.

6.2.3 Projeção Multidimensional Classe-Específica

A CSMP utiliza uma família de métricas classes-específicas, conforme definido pela Equação (6.1), com o propósito de aumentar a precisão na comparação de dados pertencentes a diferentes classes. Tem como base a LSP, mas usa um método de penalidade ao invés de mínimos quadrados para restringir o sistema linear responsável pela projeção (Xu et al., 2009).

A CSMP apoia-se no fato de que cada instância α de um conjunto de dados $\mathcal{I}$ pode ser escrita como uma combinação linear de seus vizinhos mais próximos no espaço visual. Em termos matemáticos: seja $N_\alpha = \{\alpha_1, \ldots, \alpha_k\}$ o conjunto dos k vizinhos mais próximos de $\alpha \in \mathcal{I}$, e denotando por $(\alpha_{ix}, \alpha_{iy})$ as coordenadas de cada elemento $\alpha_i \in N_\alpha$ quando mapeados para o espaço visual $\mathbb{R}^2$. Partindo da hipótese da combinação linear, pode-se calcular as coordenadas bidimensionais de α como:

$$(\alpha_x, \alpha_y) = \sum_{\alpha_i \in N_\alpha} c_{i\alpha}(\alpha_{ix}, \alpha_{iy}), \tag{6.2}$$

onde $c_{i\alpha} > 0$.

Cada imagem em $\mathcal{I}$ dá origem a uma equação vetorial como a Equação (6.2), e quando combinadas originam dois sistemas lineares homogêneos:

$$Lx = 0; \quad Ly = 0, \tag{6.3}$$

onde x e y indicam as coordenadas dos elementos mapeados e L a matriz derivada da Equação (6.2).

Os conjuntos N_α definem um grafo de vizinhos mais próximos (*Nearest Neighbors Graph* (NNG)) de $\mathcal{I}$, isto é, um grafo conectando cada elemento em $\mathcal{I}$ a seus vizinhos mais próximos. Pode ser demonstrado que o posto de L é $n - q$, onde n é o número de elementos em $\mathcal{I}$ e q é o número de componentes conectados, tornando-se o grafo de vizinhos mais próximos (NNG) (Sorkine et al., 2004). Além disso, a fim de garantir uma única solução não trivial para os sistemas lineares definidos na Equação (6.3), o NNG deveria ter somente um componente conectado, o qual pode ser assegurado adicionando-se novas arestas ligando componentes desconectados do NNG.

Os coeficientes $c_{i\alpha}$ são definidos como segue:

$$c_{i\alpha} = \begin{cases} d_{Q_i}(\alpha, \alpha_i) & \text{se } \alpha \text{ ou } \alpha_i \text{ é uma imagem de controle,} \\ d(\alpha, \alpha_i) & \text{se } \alpha \text{ e } \alpha_i \text{ não são imagens de controle,} \\ 0 & \text{demais casos,} \end{cases} \tag{6.4}$$

onde d é a distância Euclidiana e d_{Q_i} é a métrica classe-específica definida na Equação (6.1). A fim de assegurar a simetria para L, assumiu-se a convenção $d_{Q_i}(\alpha, \alpha_i) = 0$, se α e α_i são imagens de controle a partir de classes distintas.

Foi aplicado o método da penalidade (Xu et al., 2009) para restringir os sistemas da Equação (6.3), o qual pode ser declarado como segue: deixe Q ser o conjunto de imagens de controle e b_x (respectivamente b_y) ser o vetor com zero em todas as entradas exceto nas entradas b_i correspondentes às imagens de controle α_i, onde o valor $b_i = \alpha_{ix}$ é a coordenada x (respectivamente y) da imagem de controle α_i, posicionada no espaço visual. O método da penalidade permite reescrever os sistemas homogêneos da Equação (6.3) como:

$$(L + P)f = Pb, \tag{6.5}$$

onde P é a matriz diagonal penalizada, com elementos não nulos na diagonal p_{ii}, somente nas posições correspondentes às imagens de controle, geralmente um valor alto (10^8 na nossa implementação).

O método da penalidade possui propriedades relevantes. Por exemplo, ele preserva a simetria e assegura que a matriz do sistema seja semidefinida positiva, permitindo assim a fatoração por *Cholesky*[1]. Além disso, adicionar um valor positivo elevado em algumas entradas da diagonal aumenta o condicionamento da matriz, diminuindo instabilidades numéricas.

[1]Para mais detalhes sobre a fatoração de matrizes consulte Meyer (2000).

6.3 Resultados Experimentais e Comparações

Para avaliar a abordagem proposta, dois conjuntos de experimentos foram realizados. O primeiro com o objetivo de comparar as projeções produzidas pela CSMP com outras técnicas de projeção (Seção 6.3.1), e o segundo mostrando o comportamento da CSMP no contexto de CBIR (Seção 6.3.2).

Três conjuntos de imagens obtidos a partir da coleção *Caltech101* (Fei-Fei et al., 2004) foram utilizados nos experimentos, conforme detalhado na Tabela 6.1. Os conjuntos são constituídos por imagens coloridas, em formato JPEG, redimensionadas para o tamanho 256×256 *pixels*.

Tabela 6.1: Conjuntos de imagens utilizados nos experimentos da CSMP, da esquerda para a direita as colunas correspondem ao nome do conjunto de dados, classes [instâncias por classe], total de instâncias e dimensão após a seleção de características com o algoritmo SSFS.

Nome	Classes	Instâncias	Dimensão
Caltech-3classes	*Airplane* [800], *Faces* [870] e *Motorbikes* [798]	2.468	55
Caltech-4classes	*Crocodile* [101], *Piano* [99], *Ketch* [114] e *Trilobite* [86]	400	48
Caltech-5classes	*Cellphone* [59], *Dalmatian* [67], *Minaret* [76], *Pizza* [53] e *Schooner* [63]	318	59

Para extração de características das imagens foram combinados diferentes descritores, conforme apresentado abaixo. O número de características extraídas, por descritor, está listado na Tabela 6.2.

- **Transformada *wavelet* discreta** (Kumar e Esther, 2011): ou *Discrete Wavelet Transform* (DWT) é uma técnica de sub-banda hierárquica. As sub-bandas são criadas aplicando decomposição na imagem original. Para iniciar a decomposição a imagem é filtrada nas direções horizontal e vertical, usando filtros separáveis. Isto cria quatro sub-bandas, de acordo com as direções (horizontal/vertical) e frequências (altas/baixas). Para obter o próximo nível de decomposição, DWT é aplicada novamente, mas somente sobre a sub-banda que representa as componentes de baixa frequência, tanto horizontal como vertical, da imagem. Em cada nível de decomposição, a média e o desvio-padrão são calculados para as quatro sub-bandas geradas e os valores obtidos são usados como elementos do vetor de características (Arivazhagan e Ganesan, 2003). Nos experimentos realizados, foram admitidos dois níveis de decomposição utilizando o filtro ortogonal de *Haar*, a partir do pacote *Wavelet Toolbox* do Matlab.

- **Filtros de *Gabor*** (Ilonen et al., 2005): o filtro de *Gabor* bidimensional pode ser representado como um sinal senoidal complexo, modulado por uma função Gaussiana. Em um típico cenário de extração de características de imagem[2], os filtros de *Gabor* são utilizados como uma estrutura de multirresolução, consistindo de filtros ajustados para diferentes frequências e orientações. Neste trabalho, transformações de imagem foram realizadas utilizando quatro filtros de frequência, com $f_{max} = 0,3$ e intervalo de frequência $k = \sqrt{2}$, em seis diferentes orientações (0°, 30°, ..., 150°). A partir daí, calcula-se a média e o desvio-padrão da magnitude de cada imagem transformada e os valores são usados como elementos do vetor de características (Bianconi e Fernández, 2007). Os cálculos foram realizados com o auxílio do pacote *Simplegabor* (Ilonen e Kamarainen, 2006).

- ***Tamura*** (Tamura et al., 1978): os autores propuseram uma representação baseada em estudos psicológicos sobre percepção humana, consistindo de seis características estatísticas: largura, contraste, direção, regularidade, semelhança de linhas e rugosidade para descrever propriedades de textura.

- **Estatísticas de primeira ordem** (Theodoridis e Koutroumbas, 2006): este descritor é constituído por seis características estatísticas ou momentos derivados do histograma de níveis de cinza (imagens são convertidas para cinza), consistindo de média, variância, assimetria (*skewness*), curtose, acuidade (*sharpness*) e entropia.

- **Momentos de cor** (Maheshwary e Srivastav, 2008): neste caso, cada imagem foi dividida em 16 regiões. A partir de cada região, calculamos a média, o desvio-padrão e assimetria, usando o modelo de cores HSI, por componente. Os valores obtidos foram usados para compor o vetor de características da imagem.

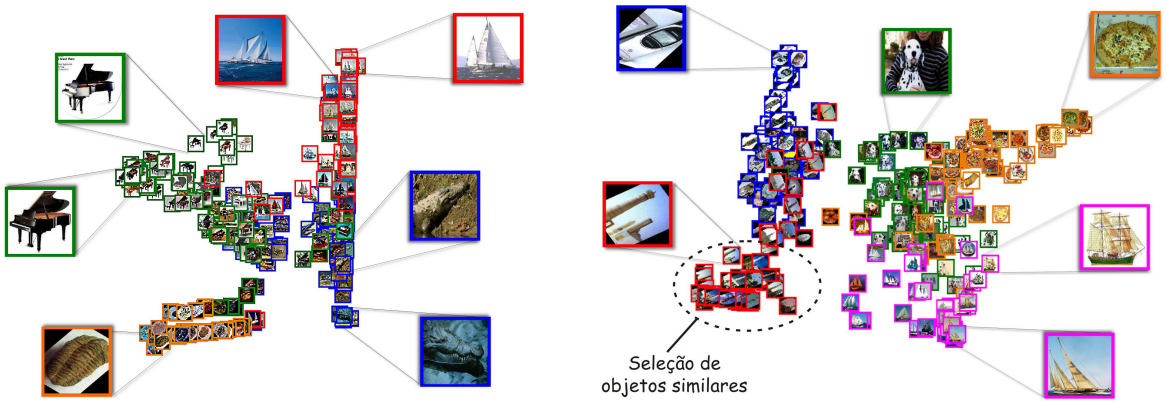
Tabela 6.2: Características extraídas a partir dos conjuntos de imagens, por descritor.

Descritor	**Número de características**
Transformada *wavelet* discreta	16
Filtros de *Gabor*	48
Tamura	6
Estatísticas de primeira ordem	6
Momentos de cor	144
Total	**220**

[2]Uma imagem pode ser interpretada como um tipo de sinal bidimensional.

6.3.1 Atestando a Qualidade da Projeção

A Figura 6.4 explora o *layout* da projeção de dados com a CSMP para os conjuntos de dados *Caltech-4classes* e *Caltech-5classes*, respectivamente. Nestas projeções, cada instância é representada por uma imagem de dimensão reduzida (miniatura da imagem original), tal que diferentes cores de moldura representam diferentes classes. Note a boa separabilidade entre as classes de imagem, característica marcante da CSMP.



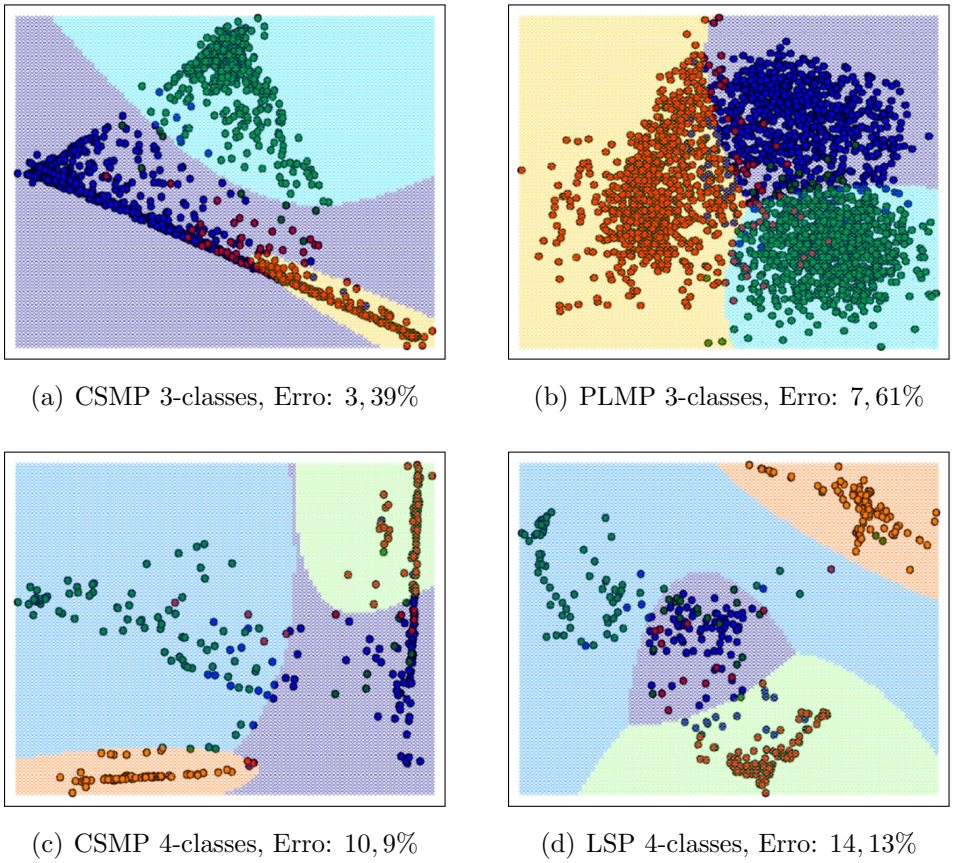
(a) *Caltech-4classes* (b) *Caltech-5classes*

Figura 6.4: Explorando o *layout* da projeção com a CSMP para diferentes conjuntos de dados e o emprego de imagens miniaturizadas, onde diferentes cores de moldura representam diferentes classes. Imagens similares podem ser selecionadas diretamente no *layout* da projeção.

Para avaliar a qualidade da projeção, na Figura 6.5, o espaço visual foi dividido em regiões, de acordo com uma função discriminante. Análise discriminante é usada nas situações onde os grupos são previamente conhecidos, e tenta encontrar a divisão ótima na qual os pontos ilustram a melhor configuração dos grupos, de acordo com a função discriminante adotada (Rencher, 2002). No experimento da Figura 6.5, também estamos interessados na taxa de erro da função discriminante ao separar os grupos originados a partir da projeção (análise quantitativa). As regiões de classificação foram definidas por funções quadráticas. Quando a análise é baseada em famílias de funções quadráticas, denomina-se *Quadratic Discriminant Analysis* (QDA). Neste trabalho, o cálculo das regiões foi realizado com a ajuda das bibliotecas do *Scikit-learn* (Pedregosa et al., 2011). Para um estudo mais aprofundado sobre QDA os trabalhos de Härdle e Simar (2007) e Seber (1984) podem ser consultados.

Na Figura 6.5, as projeções da CSMP são comparadas com as da PLMP (Paulovich et al., 2010b) e LSP (Paulovich et al., 2008), para os conjuntos de dados *Caltech-3classes* e *Caltech-4classes*, respectivamente. A *taxa de erro aparente* mostrada na Figura 6.5 é definida como a fração das instâncias classificadas incorretamente, normalmente expressa em percentual (Härdle e Simar, 2007). Note que a CSMP produz as menores taxas de erro aparente, para todos os conjuntos de dados, em comparação com as projeções da PLMP

e LSP. Além da menor taxa de erro, a CSMP facilita a identificação visual de grupos de instâncias, um aspecto importante quando o conjunto de dados é não rotulado.



(a) CSMP 3-classes, Erro: $3,39\%$ (b) PLMP 3-classes, Erro: $7,61\%$

(c) CSMP 4-classes, Erro: $10,9\%$ (d) LSP 4-classes, Erro: $14,13\%$

Figura 6.5: Comparação das projeções: (a) e (b) CSMP x PLMP no conjunto de dados *Caltech-3classes*; (c) e (d) CSMP x LSP no conjunto de dados *Caltech-4classes*.

A coesão e a separação dos grupos obtidos com a projeção também podem ser quantitativamente avaliadas pela medida da silhueta (ver Definição 2.8). O gráfico de barras horizontais da Figura 6.6 mostra os valores da silhueta computados a partir das respectivas projeções. Observe que as silhuetas da CSMP são maiores que as da LAMP, LSP e PLMP para todos os conjuntos de dados. Lembrando que quanto maior o valor da silhueta, melhor a qualidade dos agrupamentos. Além disso, pequenas variações na silhueta (terceira ou quarta casa decimal) podem implicar em agrupamentos melhores, já que este coeficiente é capaz de capturar pequenas variações no *layout* dos dados.

6.3.2 CSMP no Contexto de CBIR

Os resultados da CSMP também foram avaliados no contexto de CBIR. Dois sistemas de CBIR conhecidos foram empregados nas comparações: *Flexible Image Retrieval Engine* (FIRE) (Deselaers et al., 2008) e *Genetic Algorithm CBIR* (GA-CBIR) (Da Silva et al., 2011).

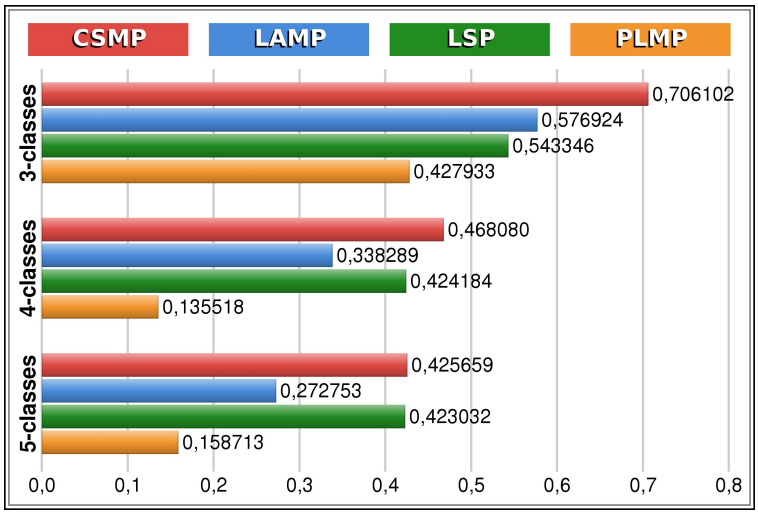
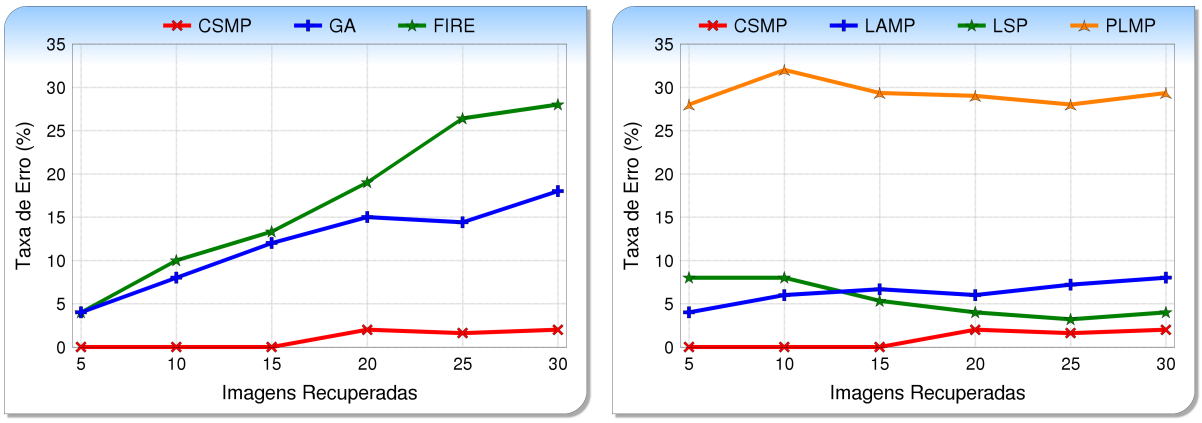


Figura 6.6: Medidas da silhueta. A técnica proposta, CSMP (em vermelho), apresenta melhores resultados que as técnicas *Local Affine Multidimensional Projection* (LAMP), *Least Square Projection* (LSP) e *Part-Linear Multidimensional Projection* (PLMP).

A Figura 6.7(a) exibe o resultado da comparação entre CSMP, FIRE e GA-CBIR, utilizando o conjunto de dados *Caltech-5classes*. A taxa de erro exibida no eixo-y foi computada do seguinte modo: para cada conjunto de imagens recuperadas (5, 10, ..., 30) foram executadas três consultas usando diferentes imagens, obtidas aleatoriamente a partir da coleção *Caltech-5classes*. O erro médio para as três consultas foi calculado com base nas imagens relevantes e não relevantes recuperadas. Neste contexto, uma imagem diz-se relevante quando pertence à mesma classe da imagem de consulta (acerto) e não relevante quando pertence a qualquer outra classe (erro). Observe que a CSMP produz as menores taxas de erro em todas as consultas.



(a) CSMP x Sistemas de CBIR (b) CSMP x Técnicas de Projeção

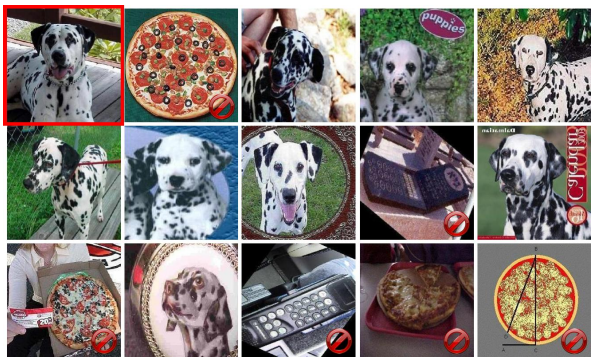
Figura 6.7: Taxas de erro da CSMP ao recuperar imagens por conteúdo, comparada a dois conjuntos distintos de técnicas: (a) sistemas de CBIR; (b) técnicas de projeção aplicadas neste contexto.

Na Figura 6.7(b), a CSMP é comparada com as técnicas de projeção LAMP, LSP e PLMP. A taxa de erro foi computada tal como descrito no parágrafo anterior, usando o mesmo conjunto de dados: *Caltech-5classes*. Embora algumas técnicas de projeção tenham se mostrado bastante competitivas, a CSMP continuou apresentando as menores taxas de erro em todas as consultas. Se a CSMP é melhor do que técnicas precisas como a LAMP e a própria LSP na qual ela está baseada, fica evidente mais uma vez que, a sua eficácia ao recuperar imagens por conteúdo está associada à família de métricas classes-específicas empregada, característica que a diferencia das demais técnicas de projeção.

6.4 Caso de Uso: Resultados Qualitativos

Uma comparação qualitativa entre a CSMP e outros sistemas de CBIR pode ser observada na Figura 6.8, utilizando o conjunto de imagens *Caltech-5classes*. As imagens com moldura vermelha (esquerda-topo) são as imagens de consulta. A lista de imagens retornada é exibida em ordem decrescente de similaridade e as imagens não relevantes foram marcadas com o símbolo ⊘, em vermelho, para facilitar a identificação. Observe

(a) CSMP: todas as imagens são relevantes



(b) GA-CBIR: 6 imagens não relevantes

(c) CSMP: todas as imagens são relevantes

(d) FIRE: 3 imagens não relevantes

Figura 6.8: Recuperação de imagens pela CSMP, GA-CBIR e FIRE mostrando as 15 primeiras imagens recuperadas. Na moldura em vermelho as imagens usadas para consulta. Note que todas as imagens recuperadas pela CSMP são relevantes, enquanto que GA-CBIR e FIRE recuperaram 6 e 3 imagens não relevantes, respectivamente (indicadas pelo símbolo ⊘, em vermelho).

que a CSMP supera o GA-CBIR consideravelmente, já que todas as suas imagens são relevantes e pertencentes à classe *Dalmatian* (Figura 6.8(a)), enquanto que o GA-CBIR recuperou 6 imagens não relevantes nesta categoria (Figura 6.8(b)). O mesmo efeito pode ser verificado entre CSMP e FIRE, ou seja, todas as imagens recuperadas pela CSMP são relevantes (Figura 6.8(c)), contra 3 imagens não relevantes do FIRE (Figura 6.8(d)).

Depois que os dados são projetados com a CSMP, qualquer imagem é, potencialmente, uma imagem de consulta, ou seja, é possível realizar múltiplas consultas sem a necessidade de reconstruir o mapeamento múltiplas vezes. Assim sendo, considere o experimento apresentado na Figura 6.9, o qual emprega diferentes imagens de consulta para recuperar imagens similares a partir da coleção *Caltech-5classes*, tal que as imagens à esquerda, com moldura vermelha, são as imagens de consulta. A lista de imagens retornada está disposta em ordem decrescente de similaridade, da esquerda para a direita, e as imagens não relevantes foram marcadas com o símbolo ⊘, em vermelho. Note que a CSMP obteve cem por cento de eficácia nas consultas (Figura 6.9(a)), ao passo que GA-CBIR recuperou 2 imagens não relevantes na classe *Cellphone* (Figura 6.9(b)), e o FIRE recuperou 3 imagens não relevantes: 2 na classe *Pizza* e 1 na classe *Cellphone* (Figura 6.9(c)).

Estes experimentos evidenciam a eficácia da CSMP como técnica de projeção especializada em busca por similaridade.

(a) CSMP: todas as imagens são relevantes

(b) GA-CBIR: 2 imagens não relevantes na classe *Cellphone*

(c) FIRE: 3 imagens não relevantes, sendo 2 na classe *Pizza* e 1 na classe *Cellphone*

Figura 6.9: Recuperação de imagens pela CSMP, GA-CBIR e FIRE usando diferentes imagens de consulta. Cada sequência corresponde às 15 primeiras imagens recuperadas, a partir da imagem de consulta mais à esquerda, com moldura vermelha. Note que todas as imagens recuperadas pela CSMP são relevantes, enquanto que GA-CBIR e FIRE recuperaram 2 e 3 imagens não relevantes, respectivamente (indicadas pelo símbolo ⊘, em vermelho).

6.5 Considerações Finais

Os resultados apresentados nas seções anteriores mostram claramente a efetividade da CSMP, superando tanto em acurácia quanto em flexibilidade os métodos testados. A nova técnica, especializada em busca por similaridade, apoia-se em um conjunto de medidas que dependem das classes das imagens (classe-específica), a qual revelou ser muito eficaz nas comparações de dados.

A CSMP fundamenta-se em uma técnica de projeção bem consolidada, a LSP (Paulovich et al., 2008) que, além de uma sólida formulação matemática ao projetar dados, permite o uso de recursos visuais e interativos. A formulação da LSP facilita inserir informações de classe no sistema através da família de métricas classes-específicas construída e, o método da penalidade faz com que as diferentes classes fiquem mais afastadas durante a projeção. Ou melhor, a projeção permite distinguir diferentes classes de imagem e, por conseguinte, múltiplas consultas podem ser realizadas sem a necessidade de refazer o mapeamento, tornando o processo de consulta muito rápido e prático.

Depois que os dados são projetados, o usuário seleciona uma ou mais imagens de consulta. Em seguida, o sistema devolve a lista ordenada das imagens mais próximas, computadas a partir do espaço visual, por ordem de relevância. No entanto, quando a ordem não é fator primordial, é possível encontrar as imagens mais similares de forma interativa, selecionando-as diretamente a partir do *layout* da projeção, realizada com o uso de imagens miniaturizadas para facilitar a identificação (ver Figura 6.4 como exemplo).

Novas entradas de dados são previstas na CSMP, ou seja, é possível inserir uma nova imagem na coleção, mas, neste caso, é necessário extrair as características da imagem, adequar a entrada na matriz do sistema e refazer o mapeamento. Este procedimento pode ter custo alto, dependendo da coleção de imagens. Portanto, a CSMP não é indicada para problemas onde novas imagens devem ser inseridas e consultadas em tempo real.

Em todos os experimentos realizados, os melhores *layouts* foram produzidos com o fator de penalidade fixado em 10^8. Os outros parâmetros também são determinados automaticamente, porém, a escolha do número de pontos de controle é um aspecto que merece atenção. Sabe-se que a qualidade na recuperação de imagens da CSMP depende fundamentalmente da família de métricas classes-específicas empregada, no entanto, o emprego de um número muito reduzido de pontos de controle pode comprometer a qualidade da projeção e, consequentemente, afetar o resultado final das consultas.

A limitação dos pontos de controle foi contornada com o emprego de recursos visuais: uma interface gráfica adequada onde o usuário é orientado a escolher uma quantidade razoável de imagens, preferencialmente a partir de categorias distintas, obtendo, assim, um número suficiente de pontos de controle para guiar o restante da projeção.

Finalmente, vale lembrar que a etapa de construção da família de métricas classes-específicas está sujeita à resposta de um seletor de características para identificar os

atributos mais relevantes de cada classe de imagem e, mesmo o melhor seletor de características ainda pode falhar nesta etapa. O fato de eliminar atributos possivelmente relevantes para uma determinada classe, pode comprometer a construção da família de métricas classes-específicas, principalmente em coleções de imagens com grande diversidade (muitas categorias ou classes). Esta incerteza motivou a investigação e implementação de uma nova técnica, discutida no próximo capítulo, que entre outras melhorias, introduz um fator de incerteza nas comparações de dados multidimensionais, ampliando ainda mais a acurácia da CSMP, ideal para coleções de imagens com grande diversidade.

Capítulo

7

Métricas Específicas Associadas à Informação de Incerteza

APESAR da grande variedade de técnicas e estratégias empregadas em buscas por similaridade, muitas técnicas consideram uma única métrica para medir a similaridade entre os objetos e, raramente, consideram a incerteza inerente a este processo. A proposta da CSMP, apresentada no capítulo anterior, introduz uma família de métricas baseada em classes para aumentar a precisão das respostas, contudo, também não leva em conta informações de incerteza.

Este capítulo discute uma nova técnica denominada *Class-Specific with Weight Image Retrieval* (CSWIRe), inicialmente desenvolvida com o intuito de aperfeiçoar as respostas da CSMP para coleções de imagens complexas, com maior diversidade de classes, todavia, mantendo os mesmos recursos da sua antecessora: interatividade, tempo de resposta viável e eficácia.

Dentre as categorias de técnicas existentes, as supervisionadas normalmente são as mais precisas, entretanto, como requerem dados rotulados, acabam impondo restrições ao domínio de aplicação. As não supervisionadas, por outro lado, são mais abrangentes, porém não tão precisas. A CSWIRe, assim como sua antecessora, é semissupervisionada, capaz de recuperar imagens com a mesma precisão das técnicas supervisionadas, sem impor restrições ao domínio de aplicação, além de ser muito mais precisa ao lidar com grandes coleções de imagens e permitir múltiplas imagens de consulta como entrada.

A etapa supervisionada corresponde à criação de um ***modelo de classes***, definido a partir de um pequeno subconjunto de imagens, interativamente escolhido pelo usuário. À medida que o usuário escolhe as imagens, elas são rotuladas e em seguida submetidas a um classificador, com o objetivo de encontrar os atributos e pesos que melhor discriminam

cada classe, gerando uma família de métricas classes-específicas com pesos. A combinação de tais métricas com informação de incerteza resulta em um mecanismo eficaz na recuperação de imagens por conteúdo, em condições de operar tanto no espaço de características das imagens como no espaço visual através da projeção multidimensional com o método da penalidade, conforme o esquema proposto pela CSMP no capítulo anterior (Seção 6.2).

Os resultados mostram que esta abordagem supera os métodos existentes, tornando-se uma solução atrativa na recuperação de imagens por conteúdo (Seção 7.3).

7.1 Principais Contribuições

Em resumo, as principais contribuições deste trabalho são:

- *Modelo de Classes*: um modelo flexível construído a partir da perspectiva do usuário para garantir a mesma precisão de técnicas supervisionadas ao recuperar imagens por conteúdo.
- *Modelagem de incerteza* do processo de recuperação de imagens por conteúdo.
- *Família de métricas classes-específicas com pesos*: projetada para medir a similaridade entre imagens, usando os melhores atributos de cada classe associados com informação de incerteza.

7.2 Class-Specific with Weight Image Retrieval (CSWIRe)

Dado um conjunto $\mathcal{I}$ contendo n imagens, o primeiro passo da técnica consiste em obter o espaço de características de $\mathcal{I}$. Em termos matemáticos isto corresponde à transformação $T : \mathcal{I} \rightarrow \mathbb{R}^m$, onde m é o número de características usadas para descrever cada imagem. Os conjuntos de imagens e descritores de características utilizados neste processo são apresentados na Seção 7.3.

O processo de extração de características é realizado uma única vez para cada conjunto de imagens, em uma fase de pré-processamento. Diferente da CSMP, esta abordagem não faz uso de algoritmos de seleção para reduzir o espaço de características, evitando desse modo que algum atributo seja indevidamente excluído. Ao invés disso, cada atributo é ponderado com base em uma medida de incerteza. A Figura 7.1 apresenta os passos fundamentais desta abordagem.

Com o espaço de características definido, o usuário informa uma ou mais imagens de consulta e constrói um modelo a partir de um subconjunto de imagens, denominado ***modelo de classes*** (Seção 7.2.1). A seguir, um classificador é aplicado sobre este modelo, retornando as melhores características e pesos que representam cada classe daquele modelo (Seção 7.2.2). Informação de incerteza é então associada à resposta do classificador

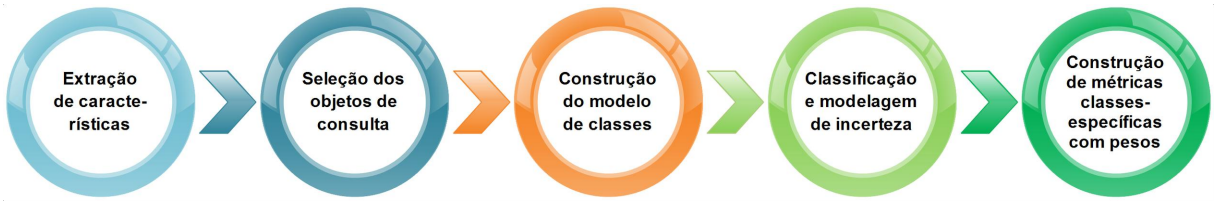


Figura 7.1: *Pipeline* da CSWIRe.

(Seção 7.2.3) para derivar uma família de métricas classes-específicas com pesos, utilizada para comparar imagens (Seção 7.2.4).

7.2.1 O Modelo de Classes

Este passo consiste em obter um subconjunto de imagens para o qual instâncias similares deveriam ser procuradas.

Inicialmente, o usuário deve escolher algumas imagens-bases para representar diferentes categorias ou classes, a partir da coleção. Note que, as imagens não estão previamente categorizadas dentro da base de dados, portanto, elas devem ser escolhidas com uso de recursos visuais e, à medida que o usuário as escolhe, elas são rotuladas pelo sistema como uma nova classe. Não é necessário considerar representantes de todas as categorias presentes na coleção, porém, quanto mais imagens de classes distintas forem incluídas, mais abrangente será o modelo construído.

Para fixar notação, considere inicialmente uma única imagem de consulta I_Q pertencente à classe C_Q (múltiplas imagens de consulta são discutidas na Seção 7.2.5) e seja $p+1$ o número de imagens de classes distintas, escolhidas pelo usuário. Este subconjunto de imagens-bases deve conter uma imagem relevante $I_R \in C_Q$, preferencialmente o vizinho mais próximo de I_Q ou o próprio I_Q e p imagens não relevantes $I_{\widetilde{R}_i} \notin C_Q, i = 1, 2, ..., p$. É importante que ao menos uma imagem relevante para a consulta esteja presente entre as imagens-bases.

Depois que tais imagens são selecionadas, o modelo deve ser estendido. Estender o modelo significa encontrar os k-vizinhos mais próximos de cada imagem-base por meio de simples consultas, usando a métrica Euclidiana. Este processo pode ser realizado de forma completamente automática pelo sistema ou de forma assistida pelo usuário (semiautomática), a fim de garantir maior acurácia ao modelo. O número de vizinhos, $k \in \mathbb{N}$, é um valor empírico, e não precisa ser o mesmo para cada imagem-base. Nos testes realizados, os modelos de classes foram construídos com k variando entre 7 e 10, usualmente 10.

Ao final do processo, o modelo de classes será constituído pelo seguinte subconjunto de imagens:

$$\mathcal{M} = \{\underbrace{I_R, I_{RN_1}, I_{RN_2}, \ldots, I_{RN_k}}_{\text{classe 1}}, \underbrace{I_{\widetilde{R}_1}, I_{\widetilde{R}_1 N_1}, \ldots, I_{\widetilde{R}_1 N_k}}_{\text{classe 2}}, \ldots, \underbrace{I_{\widetilde{R}_p}, I_{\widetilde{R}_p N_1}, \ldots, I_{\widetilde{R}_p N_k}}_{\text{classe } p+1}\}, \quad (7.1)$$

onde $I_{RN_1}, \ldots, I_{RN_k}$ são os vizinhos mais próximos de I_R, $I_{\widetilde{R}_1 N_1}, \ldots, I_{\widetilde{R}_1 N_k}$ são os vizinhos mais próximos de $I_{\widetilde{R}_1}$, e assim por diante. Note que o número de vizinhos k, na equação acima, pode ser diferente para cada imagem-base, no entanto, por simplicidade, foi admitido o mesmo número de vizinhos para todas as imagens.

Uma vez que o modelo de classes é criado, ele pode ser utilizado para recuperar as imagens mais similares a qualquer imagem da coleção cuja classe esteja presente entre as $p+1$ classes do modelo. O relacionamento entre conjuntos de imagens e modelos de classes é 1 para n. Em outras palavras, isto significa que vários modelos de classes podem ser criados para um único conjunto de imagens, conforme a necessidade. Também é possível estender ainda mais um modelo de classes existente, a posteriori, aumentando o número de classes e/ou imagens usadas para representar cada classe, aumentando consequentemente a sua acurácia. Esta flexibilidade permite recuperar imagens de maneira cada vez mais precisa, já que é possível incorporar, a qualquer momento, o conhecimento do usuário ao modelo, semelhante aos modelos de *Relevance Feedback* (RF) (Baeza-Yates e Ribeiro-Neto, 2011; Rocchio, 1971).

7.2.2 O Papel do Classificador

A partir do modelo de classes $\mathcal{M}$ (Equação (7.1)), é possível empregar o classificador *Logistic Model Tree* (LMT) (Landwehr et al., 2005), disponível a partir do pacote WEKA (Hall et al., 2009) para estimar os melhores atributos e pesos que representam cada classe do modelo $\mathcal{M}$. Na verdade, a LMT retorna dois vetores (índices e pesos) e uma constante para cada classe C_i que compõe o modelo de classes:

$$\mathcal{V}(\mathcal{M}) = \{(f_{C_1}, w_{C_1}, \delta_{C_1}), (f_{C_2}, w_{C_2}, \delta_{C_2}), \ldots, (f_{C_{p+1}}, w_{C_{p+1}}, \delta_{C_{p+1}})\}, \tag{7.2}$$

onde f_{C_i} é um vetor contendo os ***índices*** *dos atributos mais relevantes* da classe C_i, w_{C_i} é um vetor contendo os *correspondentes* ***pesos*** e δ_{C_i} um peso específico da classe, neste contexto denominado ***constante*** *de separabilidade entre classes.* O símbolo $\mathcal{F}_{C_i}(I)$ foi usado para indicar os ***valores*** *dos atributos* correspondentes aos índices f_{C_i}, para a imagem I. Por exemplo, seja I uma imagem representada por um vetor em $\mathbb{R}^m$, tal que $I = (x_1, x_2, \ldots, x_m)$, então $\mathcal{F}_{C_i}(I) = (x_{i1}, x_{i2}, \ldots, x_{ij})$, onde os índices $i1, i2, \ldots, ij$ indicam que somente os j atributos mais relevantes de I, em relação à classe C_i, foram selecionados, com $j \leqslant m$. Vale lembrar que o número de atributos mais relevantes j é, em geral, diferente para cada classe C_i que compõe o modelo.

O conjunto de valores $\mathcal{V}(\mathcal{M})$, definido na Equação (7.2), representa os melhores atributos e pesos para cada classe C_i daquele modelo. Estes valores podem ser usados para estimar se uma dada imagem I pertence a uma certa classe C_i, através da constante υ_{C_i},

calculada com respeito a I, da seguinte forma:

$$\upsilon_{C_i}(I) = \langle \mathcal{F}_{C_i}(I)\ ,\ \mathrm{w}_{C_i} \rangle\ +\ \delta_{C_i}, \tag{7.3}$$

onde $\langle *, \star \rangle$ corresponde ao produto escalar usual. Neste contexto, denominado ***valor característico*** *da classe* C_i para a imagem I. A constante de separabilidade δ_{C_i}, adicionada, garante que valores característicos de classes distintas fiquem bem afastados.

Note que o modelo de classes é constituído por representantes de algumas classes, não necessariamente todas as classes da coleção, já que o número de classes é, a priori, desconhecido. Isto pode gerar uma *incerteza* no modelo, o qual a CSWIRe deveria estar apta a tratar, conforme explicado na próxima seção.

7.2.3 Modelagem da Incerteza

Para lidar com a incerteza do modelo de classes definido anteriormente, foram empregados conjuntos *fuzzy*. Os conceitos necessários para a compreensão do restante do capítulo estão disponíveis na Seção 2.6.

Supondo que uma imagem I pertencente ao modelo de classes $\mathcal{M}$ (Equação (7.1)) seja representada como uma função triangular *fuzzy* obtida a partir de um *Fuzzy Triangular Number* (FTN), indicado por (x_1, x_2, x_3), tal que o grau de pertinência máximo (valor central) seja dado pelo *valor característico* de I (Equação (7.3)), e que os limites inferior e superior sejam obtidos, respectivamente, subtraindo e adicionando um valor constante $\Delta\upsilon > 0$, fixado para aquele modelo. Assim,

$$x_1 = \upsilon_{C_i}(I) - \Delta\upsilon\ ;\ x_2 = \upsilon_{C_i}(I)\ ;\ x_3 = \upsilon_{C_i}(I) + \Delta\upsilon. \tag{7.4}$$

A Figura 7.2 ilustra esta situação, onde o universo do discurso é o conjunto de imagens do modelo $\mathcal{M}$ e cada imagem $I \in \mathcal{M}$ é representada por uma função de pertinência μ_I, que em notação *fuzzy* pode ser indicada como $\mu_I(\upsilon) : \mathcal{M} \to [0, 1]$. Note que os valores ao longo do eixo horizontal contribuem com a identificação das classes, já que os *valores característicos* dependem de uma classe para serem computados, e os valores no eixo vertical correspondem aos diferentes níveis-α admitidos para aquele modelo.

A Equação (7.4) garante que todo triângulo ilustrado na Figura 7.2 seja isósceles, uma vez que $x_3 - x_2 = x_2 - x_1 = \Delta\upsilon$. Esta condição produz distribuição uniforme de valores e equilíbrio ao modelo. Além disso, é importante ter em mente que quanto mais funções são usadas para representar o modelo, maior é a sobreposição de cortes-α. Sobreposições são inevitáveis mas podem ser minimizadas fazendo $\Delta\upsilon$ pequeno. O valor de $\Delta\upsilon$ é arbitrário, nos experimentos realizados ele foi fixado em 0,5; por produzir melhores

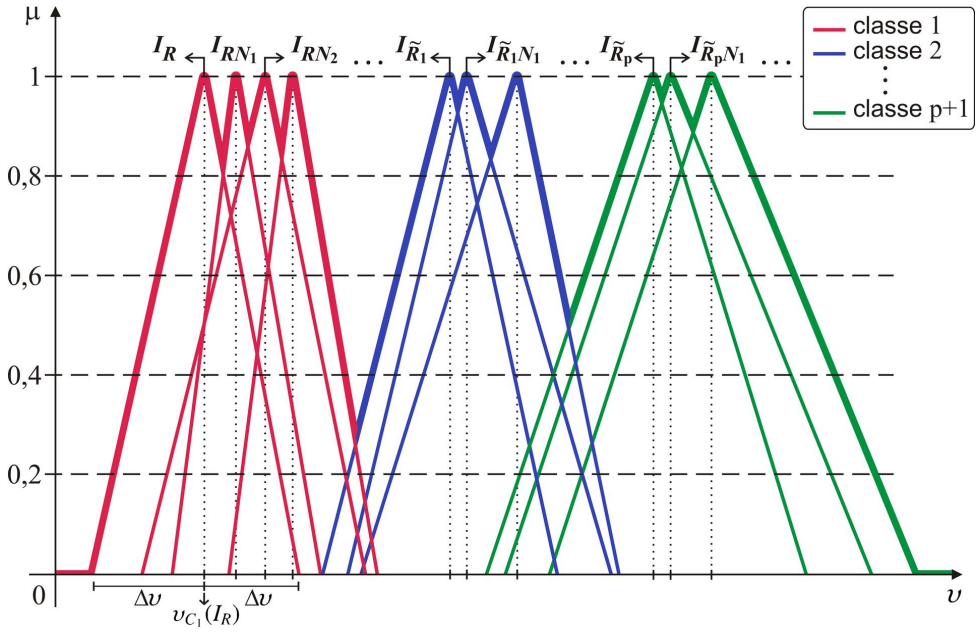


Figura 7.2: Modelo de classes representado por meio de funções *fuzzy* triangulares. O contorno em destaque representa a união de conjuntos *fuzzy*.

resultados. Valores no intervalo real [0,3; 0,7] são aceitáveis, produzindo praticamente o mesmo efeito. Valores muito abaixo de 0,3 costumam tornar o modelo instável, e muito acima de 0,7 criam sobreposições nos cortes–α difíceis de lidar.

Nesta abordagem, o sistema estima valores de incerteza automaticamente, de acordo com funções de pertinência associadas a um pequeno subconjunto de imagens (modelo de classes), aumentando assim o grau de precisão das respostas. Em contraste a outras abordagens, esta formulação não requer um limiar (*threshold*) definido pelo usuário para determinar a incerteza.

Um dos maiores desafios da modelagem de incerteza, no entanto, é determinar o nível ótimo de permissividade do modelo (ver Seção 2.6). Nesta formulação, este nível está associado ao *número de níveis-α* considerados, indicado por n_α, $n_\alpha \in \mathbb{N}$. A escala semântica apresentada no Capítulo 2 (Figura 2.5), por exemplo, pode ser muito restritiva se todos os níveis presentes forem utilizados. Nos experimentos realizados, os melhores resultados foram alcançados para $3 \leqslant n_\alpha \leqslant 6$. O modelo exibido na Figura 7.2, por exemplo, possui seis níveis-α: $\{0;\ 0{,}2;\ 0{,}4;\ 0{,}6;\ 0{,}8;\ 1\}$, cada um contendo vários cortes-α. O número de níveis-α é empírico, mas em geral, quanto mais níveis-α, mais *restritivo* é o modelo; e quanto menos níveis-α, mais *permissivo* é o modelo de incerteza.

Uma vez que o número de níveis-α tenha sido fixado, é possível aplicar a Equação (2.13) sobre as funções *fuzzy* triangulares que representam o modelo de classes $\mathcal{M}$, a fim de obter seus cortes-α, onde o número de funções será indicado por $n_\mathcal{M}$. Os valores encontrados são então armazenados em uma estrutura de dados de apoio, nesta abordagem denominada *matriz de cortes-α*, indicada por $\mathcal{L}$. A nova estrutura criada deve conter níveis-α, classes

e $n_\alpha \times n_\mathcal{M}$ células, cada uma com seu respectivo corte-α (i.e., limite inferior e superior do intervalo de corte). $\mathcal{L}$ deve ser ordenada pelos níveis-α, de forma decrescente, conforme exemplificado na Figura 7.3.

$\mathcal{L}$ tem baixo custo computacional, já que o modelo de classes é normalmente constituído por algumas dezenas de imagens. Além do mais, a matriz de cortes-α precisa ser computada somente quando um modelo de classes é criado ou modificado.

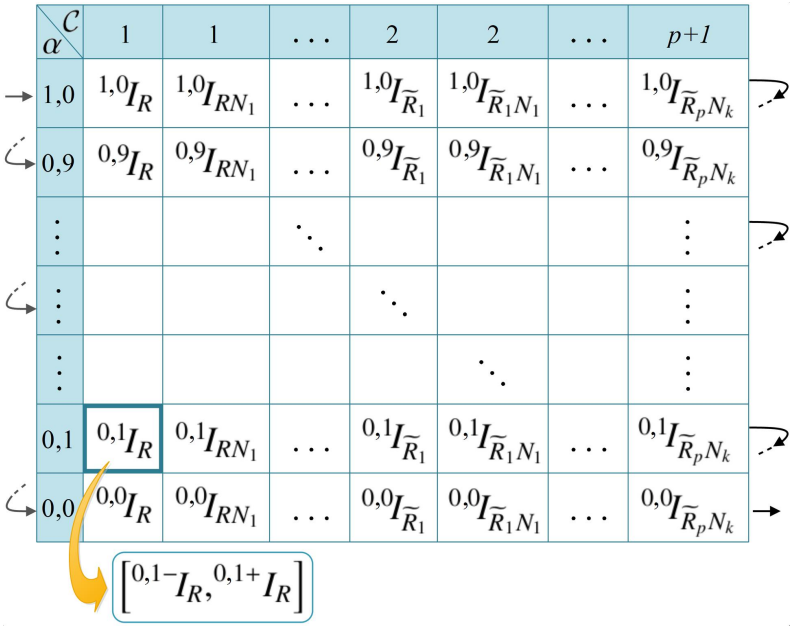


Figura 7.3: Matriz de cortes-α construída a partir do modelo de classes $\mathcal{M}$ (Equação (7.1)) e níveis-α pré-definidos. Os valores na primeira linha correspondem às classes das imagens, e os valores na primeira coluna correspondem aos níveis-α.

7.2.4 O Processo de Recuperação de Imagens

Quando uma imagem I é recuperada, a CSWIRe inicialmente calcula o *valor característico* de I (Equação (7.3)) e procura por algum corte-α $^\alpha I_L$ na matriz $\mathcal{L}$, de modo que

$$v_{C_i}(I) \in [^{\alpha-}I_L,\ {}^{\alpha+}I_L], \quad i = 1, \ldots, p+1. \tag{7.5}$$

Se a condição acima é verificada para algum índice i, então "I é membro de C_i com *grau de pertinência* igual ao correspondente nível-α", ou em notação *fuzzy*:

$$\mu_I(v_{C_i}) = \alpha, \tag{7.6}$$

caso contrário, a imagem I não pertence a qualquer classe do modelo de classes $\mathcal{M}$, portanto, seu grau de pertinência é 0.

A questão que naturalmente surge, neste momento, é: "que decisão tomar quando a condição dada pela Equação (7.5) se verifica para diferentes níveis-α?"

Nesta proposta, ambiguidades são resolvidas pela união de conjuntos *fuzzy*, isto é, tomando o valor máximo das funções de pertinência (Definição 2.24). A união de conjuntos *fuzzy* pode aumentar o grau de pertinência de um elemento, tornando o modelo mais permissivo. Em termos computacionais, resolver ambiguidades usando a operação de união significa percorrer a matriz de cortes-α $\mathcal{L}$, do nível mais alto para o nível mais baixo (direção representada pelas flechas na Figura 7.3), até encontrar o primeiro corte-α ${}^{\alpha}I_L$ que satisfaz a condição da Equação (7.5). Por esta razão, é importante manter $\mathcal{L}$ ordenada pelo nível-α.

O resultado da consulta será o conjunto de imagens $\mathcal{I}'$, ordenado pela distância $d(I_Q, I), \forall I \in \mathcal{I}$, tal que:

$$d(I_Q, I) = \begin{cases} (1-\alpha) \cdot \left\| \mathcal{F}_{C_i}(I_Q) - \mathcal{F}_{C_i}(I) \right\|, \text{ se } \; I \in C_i \\ \left\| I_Q - I \right\|, \text{ caso contrário,} \end{cases} \tag{7.7}$$

onde $||\star||$ corresponde à norma Euclidiana usual, I_Q é a imagem de consulta, I é a imagem sendo comparada e α é o correspondente nível-α de I localizado na matriz $\mathcal{L}$, conforme descrito anteriormente (Equação (7.6)).

Note que a condição $I \in C_i$ acima, é verificada através da Equação (7.5), ou seja, se for detectado que a imagem I pertence a alguma classe do modelo de classes, a métrica classe-específica com peso é utilizada, caso contrário emprega-se a métrica Euclidiana usual. Em outras palavras, isto significa que quando a imagem do modelo de classes, cujo intervalo de corte contém o valor característico da imagem I é encontrada na matriz de cortes-α, ela passa a ser uma imagem de classe conhecida C_i, logo, a métrica classe-específica pode ser aplicada. Lembrando que a matriz de cortes-α armazena a classe de cada imagem do modelo (1^a linha da matriz), portanto, a identificação da classe no modelo é de imediato.

Observe que a incerteza é transferida para a métrica por meio do escalar $1 - \alpha$, que corresponde ao complemento do nível-α para a imagem I com respeito à classe C_i.

7.2.5 Múltiplas Imagens de Consulta

Esta formulação admite buscas a partir de múltiplas imagens de consulta, simultaneamente.

Considere, por exemplo, as imagens de consulta $I_{Q_i},\ i = 1, \ldots, q$ pertencentes a classes distintas, isto é, $I_{Q_1} \in C_{Q_1}$, $I_{Q_2} \in C_{Q_2}$,..., e assim por diante. Como é de se esperar, a Equação (7.7) pode ser naturalmente aplicada a diferentes imagens de consulta I_{Q_i}, desde que as correspondentes classes C_{Q_i} estejam presentes no modelo de classes que está sendo utilizado na pesquisa.

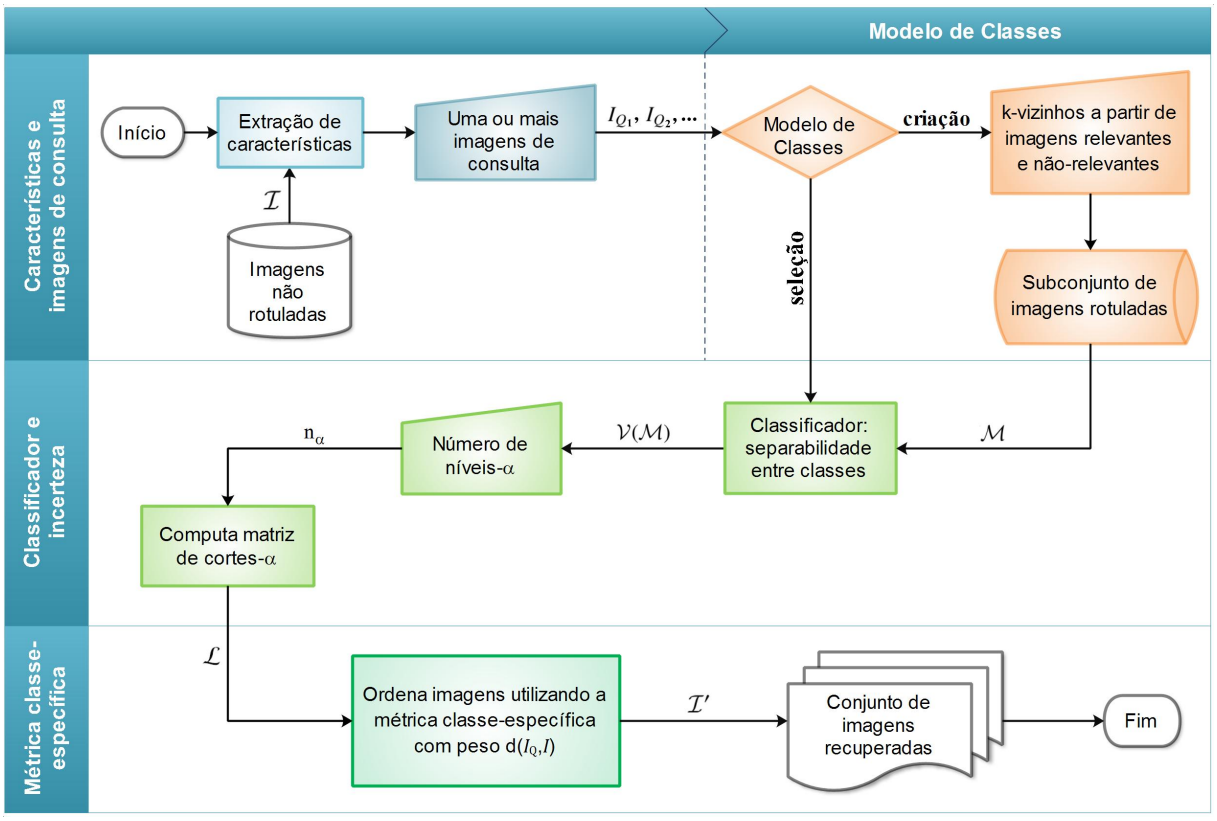


Figura 7.4: Diagrama de blocos da CSWIRe.

Assumindo que todas as classes C_{Q_i}, $i = 1, \ldots, q$ estejam presentes no modelo de classes, então a matriz $\mathcal{L}$ armazena os cortes-α de cada classe C_{Q_i}, portanto, a condição $I \in C_i$ da Equação (7.7) pode ser verificada normalmente via Equação (7.5), com a garantia que, se a imagem comparada I pertence a alguma classe de consulta C_{Q_i}, então os atributos e pesos corretos serão utilizados no cálculo da métrica classe-específica.

Para realizar uma busca por similaridade a partir de uma ou mais imagens de consulta I_{Q_i}, $i = 1, \ldots, q$, utilizando o modelo de classes $\mathcal{M}$, constituído por $n_{\mathcal{M}}$ imagens e n_α intervalos de corte armazenados na matriz de cortes-α $\mathcal{L}$, é necessário percorrer a matriz $\mathcal{L}$ a fim de encontrar o nível-α adequado e calcular a distância $d(I_{Q_i}, I)$ para cada imagem I comparada, usando a Equação (7.7). No caso de múltiplas imagens de consulta, admite-se a menor distância entre I e I_{Q_i}, $i = 1, \ldots, q$. Ao final do processo, as imagens são ordenadas com base nas distâncias encontradas. Daí, a complexidade computacional da CSWIRe é $O(n_{\mathcal{M}} n_\alpha n)$, tal que independe do número de imagens de consulta utilizadas na busca.

O diagrama de blocos apresentado na Figura 7.4 ilustra cada etapa da CSWIRe e a próxima seção confirma sua eficácia por meio de experimentos e comparações.

Um vídeo exemplificando o uso desta abordagem pode ser encontrado em `http://sites.google.com/site/paulojoiafilho/publications`.

7.3 Resultados Experimentais e Comparações

Para atestar a qualidade da CSWIRe na recuperação de imagens por conteúdo, os seguintes sistemas de CBIR foram empregados nas comparações: *Genetic Algorithm CBIR* (GA-CBIR) (Da Silva et al., 2011), *Lucene Image Retrieval* (LIRe) (Lux e Chatzichristofis, 2008) e *Semantics-sensitive Integrated Matching for Picture Libraries* (SIMPLIcity) (Wang et al., 2001).

Todos os experimentos foram executados em ambiente Linux 64-bits utilizando um microcomputador portátil Asus Intel® Core™ i7, CPU de 2 GHz, placa NVIDIA® Geforce GTX-460M e 16 GB de memória RAM. CSWIRe foi implementada em puro Python.

Os conjuntos de imagens utilizados nos testes constam na Tabela 7.1. Tais conjuntos variam significativamente tanto em número de instâncias como classes, permitindo comparações bastante confiáveis.

Tabela 7.1: Conjuntos de imagens utilizados nos experimentos da CSWIRe, da esquerda para a direita as colunas correspondem ao nome do conjunto de dados, total de instâncias, tamanho das imagens (em *pixels*), número de classes, número de imagens por classe e origem dos dados.

Nome	Instâncias	Tamanho das imagens	Classes	Imagens por classe	Origem
Corel-1000	1.000	384x256	10	100	[a]
Corel-13classes	2.600	384x256	13	200	[b]
Msrcorid	4.135	640x480	18	desbalanceado	[c]
Caltech-101classes	8.677	300x270	101	desbalanceado	[d]
Corel-430classes	43.000	384x256	430	100	[b]

[a] Li et al. (2000)

[b] Obtido a partir de *Corel Gallery Magic 65.000 - Stock Photo Library 2.*

[c] Winn et al. (2005)

[d] Fei-Fei et al. (2004)

Para extração de características das imagens, foram empregados os seguintes descritores:

- **Transformada *wavelet* discreta** (Kumar e Esther, 2011).
- **Filtros de *Gabor*** (Ilonen et al., 2005).
- ***Tamura*** (Tamura et al., 1978).
- **Estatísticas de primeira ordem** (Theodoridis e Koutroumbas, 2006).
- **Momentos de cor** (Maheshwary e Srivastav, 2008).

Tais descritores correspondem aos mesmos usados pela CSMP, seguindo iguais configurações de parâmetros, conforme detalhado na Seção 6.3.

7.3.1 Taxas de Erro

O gráfico de caixas verticais apresentado na Figura 7.5 exibe a taxa média de erro entre CSWIRe, GA-CBIR, LIRe e SIMPLIcity ao executar consultas utilizando os conjuntos de imagens listados na Tabela 7.1. Dez consultas com dez imagens de consulta distintas foram executadas em cada conjunto de imagens, tal que as vinte primeiras imagens recuperadas foram consideradas para encontrar a taxa de erro. As imagens não relevantes recuperadas foram computadas para construir as caixas verticais da figura (taxa de erro). Além disso, diferentes parâmetros foram usados em cada CBIR, conforme elucidado abaixo.

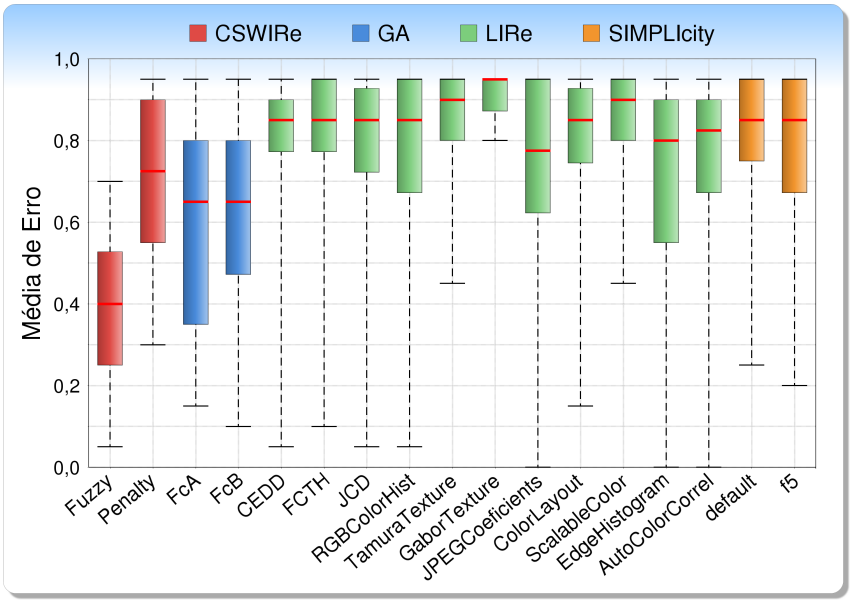


Figura 7.5: Taxa média de erro das técnicas CSWIRe, GA-CBIR, LIRe e SIMPLIcity ao executar consultas nos conjuntos de dados listados na Tabela 7.1.

Os parâmetros de cada sistema, empregados para construir a Figura 7.5, são indicados no eixo horizontal. No caso da CSWIRe, os melhores resultados foram obtidos com a opção *Fuzzy*, indicando que as imagens foram recuperadas segundo a formulação apresentada neste capítulo (Seção 7.2). A opção *Penalty* indica que a recuperação de imagens foi baseada na projeção multidimensional utilizando o método da penalidade, conforme descrito no capítulo anterior (Seção 6.2), exceto que a métrica classe-específica definida na Equação (6.1) foi substituída pela métrica classe-específica com peso, definida na Equação (7.7).

Quanto às outras técnicas, GA-CBIR por exemplo, admite duas funções de avaliação identificadas como *fitness coach* para melhorar a qualidade das respostas: FcA e FcB. LIRe tem cerca de dez descritores que podem ser utilizados para comparar imagens. SIMPLIcity permite refinar a métrica (medida de similaridade entre imagens) adicionando um marcador ao executar as consultas, variando de f0 (padrão), . . . , até f5 (maior precisão).

Observando a Figura 7.5, fica evidente que a CSWIRe com a opção *Fuzzy*, apresenta as menores taxas de erro em relação às demais técnicas, para qualquer configuração de parâmetros.

O próximo experimento foi realizado utilizando a coleção *Corel-1000*, a qual tem dez classes balanceadas, com 100 imagens cada. Com o objetivo de conduzir os sistemas ao limite, escolhemos aleatoriamente uma imagem de cada classe e executamos uma consulta com cada uma delas, computando o número de imagens não relevantes recuperadas pelas quatro técnicas, com uso de sua melhor configuração: CSWIRe com a opção *Fuzzy*, GA-CBIR com FcA, LIRe com *JPEG coefficients* e SIMPLIcity com o parâmetro f5. Este experimento foi executado dez vezes, considerando as primeiras $10, 20, \ldots, 100$ imagens recuperadas para cada método, onde 100 é o número máximo possível para cada classe. As curvas na Figura 7.6 representam o resultado de cada técnica. Observe que a CSWIRe apresenta as menores taxas de erro em todos os pontos da curva.

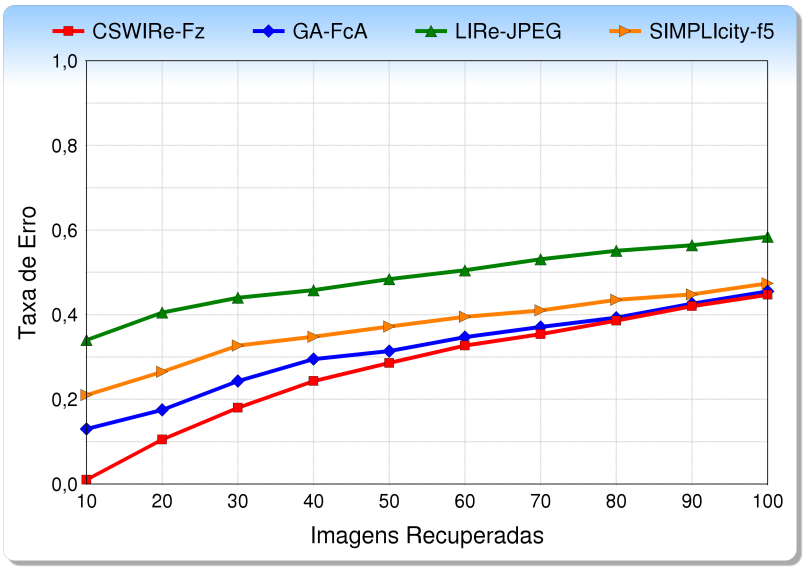


Figura 7.6: Taxas de erro das técnicas CSWIRe, GA-CBIR, LIRe e SIMPLIcity, computadas a partir da coleção *Corel-1000*, considerando as $10, 20, \ldots, 100$ primeiras imagens recuperadas.

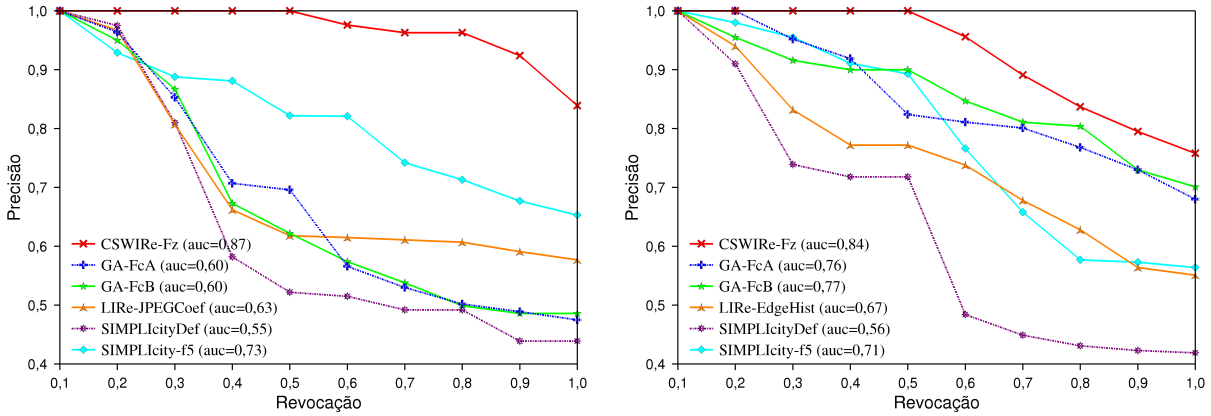
7.3.2 Curvas de Precisão-Revocação

A Figura 7.7 exibe as curvas de precisão-revocação das técnicas CSWIRe, GA-CBIR, LIRe e SIMPLIcity, juntamente com as respectivas áreas sob a curva (coeficiente `auc`).

Segundo Aslam et al. (2005), a precisão média de uma lista ordenada de documentos é a média das precisões em todos os documentos relevantes, a qual é aproximadamente a área sob a curva de precisão-revocação, indicada por *area under the curve* (`auc`). Assim, quanto maior o coeficiente `auc`, maior a precisão na recuperação de informações.

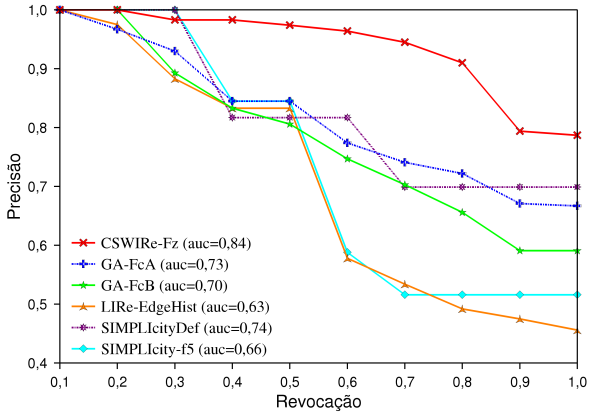
Neste experimento foram realizadas dez consultas aleatórias para cada técnica, levando em conta as vinte primeiras imagens recuperadas. As curvas de precisão-revocação foram

geradas pela média da precisão em dez níveis de revocação, interpolados conforme definido em Baeza-Yates e Ribeiro-Neto (2011). As configurações utilizadas em cada técnica são indicadas na legenda, logo após o nome da técnica. No caso do sistema LIRe que permite muitas configurações, foi exibido apenas o melhor resultado. Note que a CSWIRe, claramente apresenta maior precisão, com coeficiente `auc` superior a 0,8 em todos os casos.

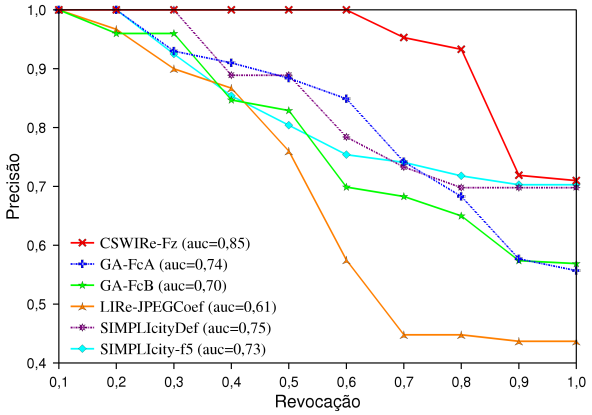


(a) *Corel-13classes*

(b) *Msrcorid*



(c) *Caltech-101classes*



(d) *Corel-430classes*

Figura 7.7: Curvas de precisão-revocação calculadas para as técnicas CSWIRe, GA-CBIR, LIRe e SIMPLIcity, mostrando as respectivas áreas sob a curva (auc), para diferentes conjuntos de dados.

7.3.3 Tempos Computacionais da CSWIRe

A Figura 7.8 mostra os tempos computacionais da CSWIRe ao recuperar imagens a partir de diferentes coleções. Foram realizadas dez consultas para cada coleção, usando diferentes imagens de entrada. Os valores que aparecem na parte inferior das caixas verticais correspondem à mediana das observações. Note que a CSWIRe levou menos de um segundo para realizar a maioria das consultas, exceto para a coleção *Corel430-classes*, mesmo assim aceitável.

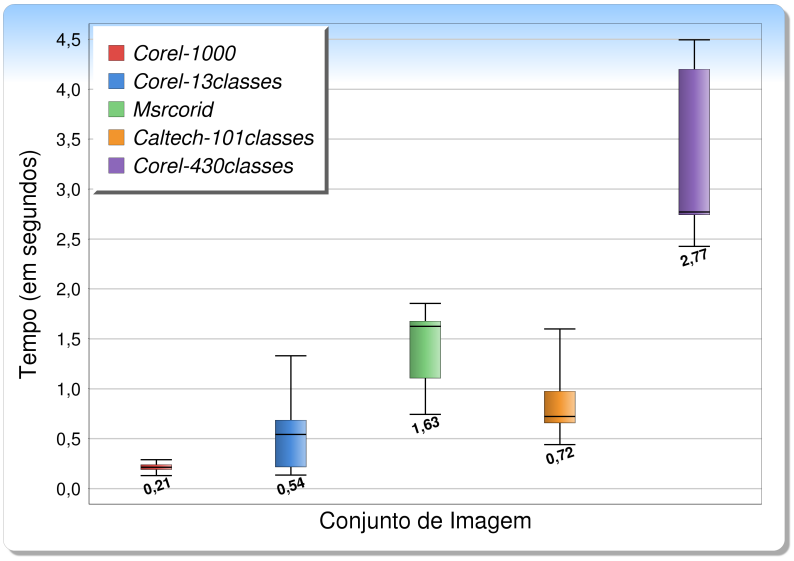


Figura 7.8: Tempos computacionais da CSWIRe ao executar consultas nos conjuntos de dados listados na Tabela 7.1.

GA-CBIR e SIMPLIcity também provaram ser eficientes nos testes realizados. Todavia, GA-CBIR apoia-se em uma etapa de pré-processamento de alto custo computacional para reduzir a dimensionalidade do espaço de características (seleção de características). De fato, GA-CBIR levou quase quatro dias para pré-processar a coleção *Corel430-classes*. SIMPLIcity, embora computacionalmente eficiente, foi claramente superado em termos de acurácia. Portanto, se considerados os dois fatores: acurácia e tempo computacional, CSWIRe se sobressai como técnica de recuperação de imagens por conteúdo.

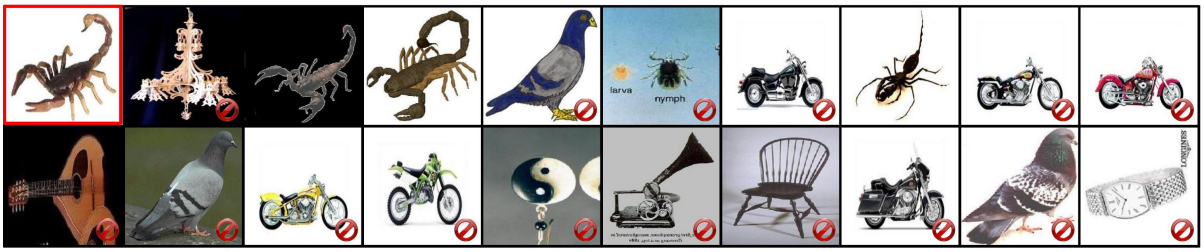
7.4 Caso de Uso: Resultados Qualitativos

A Figura 7.9 apresenta uma comparação qualitativa entre as quatro técnicas, com uso de sua melhor configuração, ou seja: CSWIRe com a opção *Fuzzy*, GA-CBIR com a função de avaliação FcA, LIRe com o descritor *Edge Histogram* e SIMPLIcity com o parâmetro f5. A imagem mais à esquerda em cada sequência, com moldura vermelha, representa a imagem de consulta, tomada aleatoriamente a partir da coleção *Caltech-101classes*. Considerando as vinte primeiras imagens recuperadas pelos quatro métodos, GA-CBIR e LIRe recuperaram o dobro de imagens não relevantes em comparação à CSWIRe, enquanto que SIMPLIcity recuperou apenas duas imagens relevantes, incluindo a própria imagem de consulta. Mais uma vez, os resultados confirmam a efetividade da CSWIRe como técnica de recuperação de imagens com base em conteúdo.

A Figura 7.10 conclui esta seção de resultados qualitativos mostrando o desempenho da CSWIRe ao empregar múltiplas imagens de consulta, simultaneamente. As Figuras 7.10(a) e 7.10(b) mostram as trinta primeiras imagens recuperadas pela CSWIRe a

(a) CSWIRe (*Fuzzy*) – 8 imagens não relevantes.

(b) GA-CBIR (FcA) – 16 imagens não relevantes.

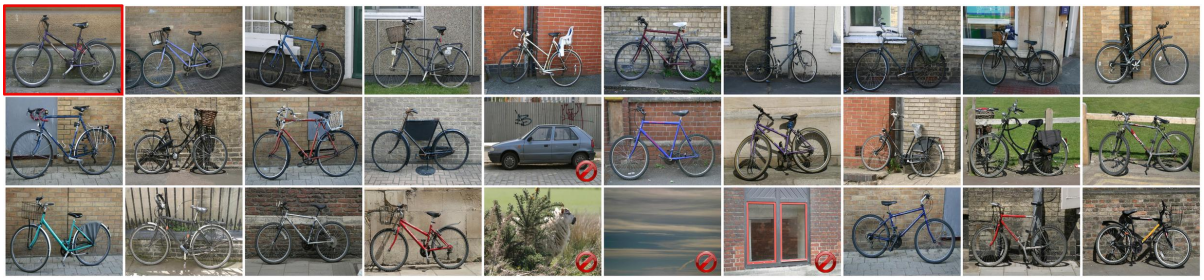


(c) LIRe (*EdgeHistogram*) – 16 imagens não relevantes.

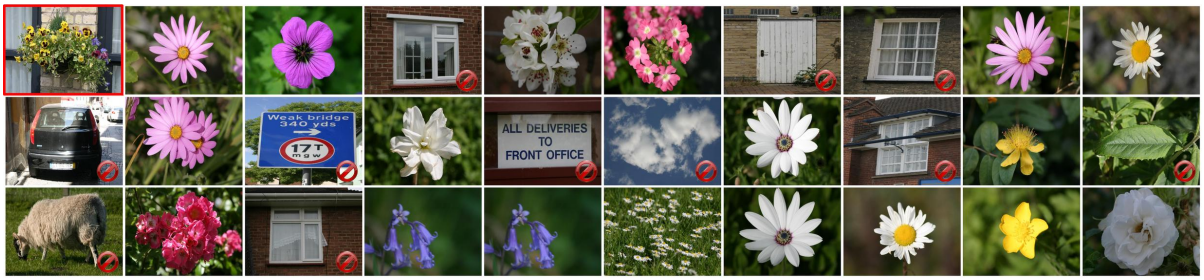
(d) SIMPLIcity (f5) – 18 imagens não relevantes.

Figura 7.9: Recuperação de imagens pela CSWIRe, GA-CBIR, LIRe e SIMPLIcity mostrando as 20 primeiras imagens recuperadas. Na moldura em vermelho a imagem de consulta: *scorpion_0019.jpg*, tomada a partir do conjunto de dados *Caltech-101classes*. CSWIRe recuperou 8 imagens não relevantes, enquanto GA-CBIR, LIRe e SIMPLIcity recuperaram 16, 16 e 18 imagens não relevantes, respectivamente (indicadas pelo símbolo ⊘, em vermelho).

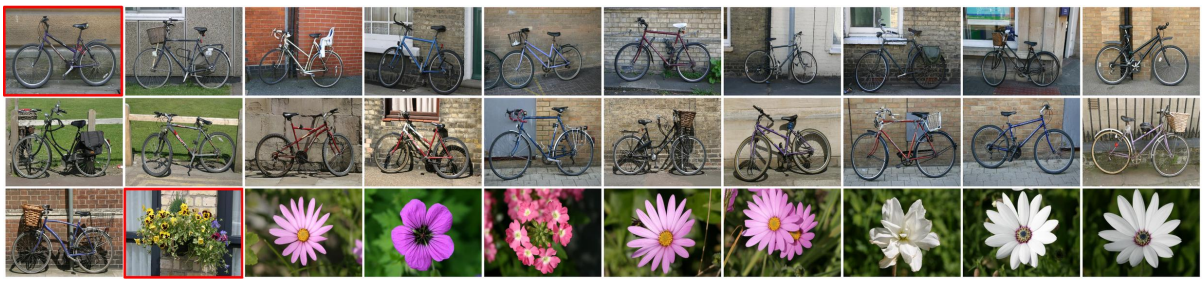
partir de uma única imagem de consulta (imagem esquerda-topo, com moldura vermelha) tomada a partir da coleção *Msrcorid*. Já a Figura 7.10(c) mostra o resultado quando ambas as imagens, *bicycle* e *flower*, são utilizadas simultaneamente como imagens de consulta. Note que as trinta imagens recuperadas são relevantes neste caso, sem acréscimo no tempo de consulta.



(a) Imagem de consulta: *bicycles_0248.jpg*, resultado: 4 imagens não relevantes, tempo de consulta: 1,693s.



(b) Imagem de consulta: *flowers_0112.jpg*, resultado: 11 imagens não relevantes, tempo de consulta: 1,597s.



(c) Imagens de consulta: *bicycles_0248.jpg* e *flowers_0112.jpg*, resultado: todas as imagens relevantes, tempo de consulta: 1,610s.

Figura 7.10: Recuperação de imagens pela CSWIRe mostrando as 30 primeiras imagens recuperadas. Na moldura em vermelho as imagens de consulta, tomadas a partir do conjunto de dados *Msrcorid*. Em (a) e (b), a partir de uma única imagem de consulta; em (c) a partir de múltiplas imagens de consulta, simultaneamente.

7.5 Considerações Finais

Este capítulo apresentou a técnica de recuperação de imagens com base em conteúdo denominada *Class-Specific with Weight Image Retrieval* (CSWIRe). Apoiada em métricas classes-específicas com pesos e informação de incerteza para comparar imagens, esta abordagem supera significativamente outros métodos em termos de acurácia, ao passo que também é competitiva com respeito à eficiência computacional, além de admitir buscas simultâneas a partir de múltiplas imagens de consulta, com mínimo acréscimo no tempo de resposta.

A construção da família de métricas classes-específicas com pesos requer um pequeno subconjunto de imagens rotuladas. Este subconjunto é usado para compor um modelo de classes com a finalidade de estimar as melhores características de cada classe. O modelo de classes não precisa conter representantes de todas as classes da coleção de imagens, mas sim alguns representantes relevantes, isto é, pertencentes à classe de consulta e alguns não relevantes (pertencentes a outras classes). Além disso, ele pode ser estendido a qualquer momento para incluir novas classes ou novos representantes. O modelo de classes é um ponto forte desta abordagem, já que permite melhorar a qualidade das respostas com base na perspectiva do usuário.

A qualidade dos resultados foi comprovada por uma abrangente bateria de testes e comparações, envolvendo taxas de erro, curvas de precisão-revocação, bem como resultados qualitativos, usando diversas coleções de imagens, confirmando de fato, a viabilidade da CSWIRe como técnica de recuperação de imagens por conteúdo, cujo potencial pode ser explorado em diferentes contextos, requerendo apenas um pequeno subconjunto de dados rotulados para construção da família de métricas (aprendizagem semissupervisionada).

Embora a CSWIRe consiga operar em conjunto com a técnica de projeção multidimensional apresentada no capítulo anterior, a qual utiliza o método da penalidade para restringir o sistema linear responsável pela projeção, esta abordagem não apresentou bons resultados para coleções de imagens contendo muitas classes. Quando a coleção tem muitas categorias de imagem, algumas tendem a ficar aglomeradas durante a projeção, comprometendo a qualidade da vizinhança no espaço visual, portanto, este recurso não foi aprofundado nos experimentos.

A determinação dos parâmetros de entrada nesta abordagem não é um fator crítico. O número de vizinhos (k), por exemplo, o qual corresponde ao número de imagens similares utilizadas para construir o modelo de classes foi fixado em torno de dez para todas as coleções de imagens, gerando bons resultados. Valores muito abaixo deste podem dificultar a tarefa de classificação, onde os melhores atributos e pesos são estimados pelo classificador automático. Valores muito acima podem comprometer a modelagem de incerteza por aumentar a sobreposição de cortes-α. Vale lembrar que, para coleções de imagens com pouca diversidade de classes é preferível o uso da CSMP.

O número de níveis-α (n_α) é o parâmetro responsável por tornar o modelo mais *permissivo* ou mais *restritivo*, comporta-se como um "*filtro*", deixando "*passar*" mais ou menos imagens, dependendo do valor que assume. Em geral, quanto mais níveis-α, mais restritivo é o modelo, tal que a recíproca também é válida, ou seja, quanto menos níveis-α mais permissivo é o modelo. Nos experimentos foram admitidos quatro níveis-α, produzindo boa acurácia. O valor de n_α pode variar um pouco, dependendo essencialmente da coleção de imagens. Uma estimativa mais precisa deste valor, pode ser obtida criando um conjunto de treinamento a partir de um subconjunto de amostras aleatórias devidamente rotuladas

(teste supervisionado), medindo a taxa média de erro para cada nível-α, de acordo com as imagens relevantes e não relevantes recuperadas.

É possível incluir novas imagens na coleção em tempo de execução, basta que as novas imagens sejam submetidas ao processo de extração de características, gerando novas entradas no conjunto de dados. Contudo, se as imagens incluídas constituem novas categorias, é ideal que o modelo de classes seja modificado para conter representantes destas categorias, a fim de garantir maior acurácia ao processo de recuperação de imagens. Esta operação tem baixo custo computacional, portanto, novas imagens podem ser incluídas e consultadas em tempo real.

Capítulo

8

Conclusões

Nesta tese foram apresentados os principais conceitos relacionados ao tema proposto, enriquecidos por uma revisão bibliográfica envolvendo técnicas de visualização de informação em diferentes domínios conectados: projeção de dados multidimensionais, identificação de agrupamentos e emprego de diferentes medidas de similaridade. Em seguida, foram apresentadas soluções para os problemas apontados na Seção 1.3: *identificação de agrupamentos* e *busca por similaridade em dados multidimensionais*, comprovando as hipóteses inicialmente apresentadas.

Novas técnicas foram desenvolvidas para cada um dos problemas, atingindo resultados expressivos, os quais podem ser confirmados por meio dos experimentos conduzidos no decorrer dos capítulos. Para garantir a veracidade, os experimentos foram realizados usando diferentes conjuntos de dados, tanto reais como sintéticos, variando consideravelmente o número de instâncias, atributos e classes. Além disso, um número significativo de técnicas existentes foram utilizadas nas comparações. Cada técnica proposta também foi avaliada em uma situação prática ou caso de uso, mostrando sua real utilidade.

Para a tarefa de identificação de agrupamentos duas técnicas foram desenvolvidas: *Local Affine Multidimensional Projection* (LAMP), discutida no Capítulo 4 e *Column Selection Method* (CSM), discutida no Capítulo 5.

LAMP é uma técnica de projeção que se destaca pela qualidade dos mapeamentos produzidos, com relação à preservação de distâncias. É capaz de projetar grandes volumes de dados com eficiência e seu algoritmo admite implementação em paralelo. Embora várias técnicas de projeção tenham sido criadas ultimamente, raras oferecem um mecanismo de interação realmente versátil como o da LAMP, capaz de projetar dados a partir de um

número tão reduzido de amostras representativas. As amostras podem ser agrupadas pelo usuário por meio de suas características visuais. Depois disso, a projeção tende a seguir fielmente o *layout* fornecido pelo usuário, facilitando a organização dos dados e identificação de agrupamentos. Esta facilidade permite que a LAMP seja utilizada em diferentes aplicações como, por exemplo, na correlação visual de dados, conforme apresentado na Seção 4.4. Nos casos em que não é possível selecionar ou agrupar amostras com base em suas características visuais, como alguns conjuntos de dados não rotulados, outra estratégia deve ser empregada, tal como a CSM.

CSM mostrou-se adequada para identificar agrupamentos quando os dados não são rotulados. O mecanismo de seleção de amostras representativas discutido na Seção 5.2.1 pode localizar instâncias representativas em cada classe, mesmo em conjuntos de dados desbalanceados, com boa precisão, facilitando a identificação de agrupamentos. Além disso, opera no espaço visual, garantindo que os grupos obtidos não fiquem fragmentados durante a visualização. Outro aspecto interessante da CSM está no fato de que a mesma estratégia empregada para encontrar instâncias representativas pode ser aplicada para selecionar os atributos mais relevantes de cada agrupamento. Os resultados apresentados confirmam a qualidade das amostras, grupos e atributos selecionados pela CSM. Estudos de caso como o da Seção 5.4, onde o *pipeline* de visualização é aplicado para traçar o perfil de consumo dos clientes em um modelo de vendas por atacado, ilustram algumas das possibilidades de aplicação desta técnica como ferramenta de visualização de dados interativa. De um modo geral, pode-se dizer que simplicidade, facilidade de implementação, baixo custo computacional e excelentes resultados nas tarefas que se propõe a fazer, tornam a CSM uma técnica atrativa para análise e visualização de dados multidimensionais.

Para a busca por similaridade em dados multidimensionais também foram desenvolvidas duas técnicas: *Class-Specific Multidimensional Projection* (CSMP), discutida no Capítulo 6 e *Class-Specific with Weight Image Retrieval* (CSWIRe), discutida no Capítulo 7.

CSMP é uma técnica de projeção que utiliza uma família de métricas baseada em classes para projetar os dados, característica que a diferencia das demais técnicas de projeção. Para a construção da família de métricas classes-específicas é necessário um subconjunto de dados rotulados, uma vez que ela é construída com base nos melhores atributos de cada classe, com o objetivo de minimizar a dissimilaridade entre pares de objetos pertencentes à mesma classe e maximizá-la para objetos pertencentes a classes distintas. Nesta tese, a CSMP foi avaliada no contexto de recuperação de imagens com base em conteúdo, porém, pode ser utilizada em diferentes cenários envolvendo busca por similaridade. A modificação na métrica faz com que ela supere outras técnicas de projeção, bem como outros sistemas de CBIR ao recuperar informações. Não obstante, para coleções de imagens com grande diversidade (muitas categorias ou classes) é preferível o uso da CSWIRe.

CSWIRe foi desenvolvida com o intuito de aperfeiçoar as respostas da CSMP para coleções de imagens complexas, com maior diversidade de classes. Pode operar diretamente no espaço de características das imagens como no espaço visual, utilizando a CSMP para projetar os dados. No entanto, quando a projeção é utilizada em coleções contendo muitas categorias de imagens, algumas tendem a ficar aglomeradas no espaço visual, comprometendo a qualidade da vizinhança. Neste caso, buscas por similaridade realizadas diretamente no espaço de características são mais eficazes. Assim como sua antecessora, esta abordagem é semissupervisionada, requerendo um pequeno subconjunto de imagens rotuladas. Este subconjunto é usado para compor um "modelo de classes" com a finalidade de estimar as melhores características de cada classe. O modelo de classes é um ponto forte desta abordagem, visto que permite melhorar a qualidade das respostas com base na perspectiva do usuário. Apoiada em métricas classes-específicas com pesos associadas à informação de incerteza, esta abordagem resulta em um mecanismo eficaz na recuperação de imagens por conteúdo. É competitiva com respeito à eficiência computacional e admite buscas simultâneas a partir de múltiplas imagens de consulta.

Os experimentos realizados comprovam que as soluções apresentadas nesta tese representam significativa contribuição na tarefa de identificação de agrupamentos e busca por similaridade em dados multidimensionais. A próxima seção discute algumas possibilidades de melhoria e novos trabalhos que podem ser originados a partir deste estudo.

8.1 Trabalhos Futuros

Identificação de agrupamentos com base em técnicas de projeção tem se mostrado uma alternativa viável, pois além da qualidade obtida, garante que os grupos não fiquem fragmentados no espaço visual, tal como proposto pela CSM (Capítulo 5). A CSM utiliza uma estratégia baseada na decomposição SVD para encontrar instâncias que representam bem o conjunto de dados como um todo, ou seja, leva em conta a variabilidade em todo o conjunto de dados (ver Seção 5.2.1). Outra possibilidade, é avaliar a variabilidade com base na vizinhança local de cada instância.

Por exemplo, considere um conjunto de dados X de dimensão $n \times m$. A ideia é tomar os k-vizinhos mais próximos de cada instância $x_i \in X$, indicado por N_i. A partir de cada subconjunto N_i, calculam-se as q-primeiras componentes principais, com $1 \leqslant q \leqslant m$, isto é, os autovetores associados aos q-maiores autovalores da matriz de covariância.

Admitindo que as componentes principais são calculadas a partir dos subconjuntos N_i padronizados com escore-z, onde a média de cada variável é zero e o desvio-padrão é um, então a matriz de covariância corresponde à matriz de correlação (Larose, 2006). Além disso, poucas componentes principais são suficientes para representar N_i, já que retêm a maioria da variação presente em todas as variáveis originais (Jolliffe, 2002).

Após o cálculo das componentes principais, cada subconjunto N_i origina um novo subconjunto C_i de dimensão $q \times m$, o qual é constituído por q autovetores unitários de dimensão m, ordenado pelos autovalores, ou seja: $C_i = \{c_{i1}, c_{i2}, \ldots, c_{iq}\}$. Os subconjuntos C_i reunidos irão compor a matriz X' de dimensão $(q.n) \times m$:

$$X' = \begin{bmatrix} C_1 \\ \vdots \\ C_n \end{bmatrix}_{(q.n)\times m} \tag{8.1}$$

A nova matriz de componentes principais X', representa um novo sistema de coordenadas, encontrado pela rotação do sistema de origem ao longo das direções de máxima variabilidade (Larose, 2006). A partir de X' é possível estimar localmente a correlação entre as instâncias do conjunto de dados original. Uma vez que cada subconjunto C_i é constituído pelas primeiras componentes principais referentes à instância x_i, e que tais componentes fornecem as direções nas quais a variação estatística é máxima, então é de se esperar que instâncias posicionadas próximas na projeção, apresentem direções similares e, consequentemente, alta correlação.

O valor da correlação estimado entre pares de instâncias x_i e x_j $(i \neq j)$, a partir das q componentes principais associadas, pode ser usado como uma heurística para melhorar a qualidade dos agrupamentos obtidos com a CSM.

Outro assunto que pretende-se investigar futuramente é a detecção de *outliers*. Embora experimentos comprovem que *outliers* têm grande chance de serem detectados pelo mecanismo de amostragem da CSM, este assunto poderia ser explorado mais a fundo, uma vez que a detecção de *outliers* para dados multivariados nem sempre é tarefa fácil (Izenman, 2008). Desse modo, algumas aplicações poderiam tomar vantagem desse recurso da CSM.

Por fim, vale lembrar que a modelagem de incerteza pode melhorar muito a qualidade das respostas de um sistema, conforme mostrado no Capítulo 7 para coleções de imagens. Seu emprego em outros tipos de dados, tais como música ou vídeo, poderia produzir aplicações interessantes, aumentando a eficácia de qualquer sistema de busca por similaridade existente.

Técnicas de projeção de dados multidimensionais têm evoluído constantemente. Uma rápida busca em periódicos especializados pode, facilmente, confirmar este fato. Com o avanço tecnológico, aumenta-se a capacidade de armazenar informações, consequentemente, são necessárias novas técnicas para lidar com os desafios que surgem, fazendo com que esta área de pesquisa continue sendo alvo de intensivas investigações nos próximos anos.

Referências Bibliográficas

Aboulmagd, H.; El-Gayar, N.; Onsi, H. A new approach in content-based image retrieval using fuzzy. *Telecommunication Systems*, v. 40, n. 1, p. 55–66, 2009.

Andrews, D. F. Plots of high-dimensional data. *Biometrics*, v. 28, n. 1, p. 125–136, 1972.

Arevalillo-Herráez, M.; Domingo, J.; Ferri, F. J. Combining similarity measures in content-based image retrieval. *Pattern Recognition Letters*, v. 29, n. 16, p. 2174–2181, 2008.

Arivazhagan, S.; Ganesan, L. Texture classification using wavelet transform. *Pattern Recognition Letters*, v. 24, n. 9–10, p. 1513–1521, 2003.

Arthur, D.; Vassilvitskii, S. K-means++: The advantages of careful seeding. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, p. 1027–1035.

Aslam, J. A.; Yilmaz, E.; Pavlu, V. A Geometric Interpretation of R-precision and Its Correlation with Average Precision. In: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil: ACM, 2005, p. 573–574.

Avidan, S.; Shamir, A. Seam carving for content-aware image resizing. *ACM Transactions on Graphics*, v. 26, n. 3, p. 10, 2007.

Baccour, L.; Alimi, A. M.; John, R. I. Some notes on fuzzy similarity measures and application to classification of shapes, recognition of arabic sentences and mosaic. *IAENG International Journal of Computer Science*, v. 41, n. 2, p. 81–90, 2014.

Bache, K.; Lichman, M. UCI machine learning repository. 2013. Disponível em: <`http://archive.ics.uci.edu/ml`>. Acesso em: 10 nov. 2014.

Baeza-Yates, R.; Ribeiro-Neto, B. *Modern Information Retrieval: The Concepts and Technology behind Search*. 2nd ed. New York: Addison-Wesley, 2011. 913 p.

Bai, Z.; Demmel, J.; Dongarra, J.; Ruhe, A.; van der Vorst, H. (Ed.). *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Philadelphia: SIAM, 2000. 410 p. (Software, Environments, and Tools).

Balasubramanian, M.; Schwartz, E. L.; Tenenbaum, J. B.; De Silva, V.; Langford, J. C. The Isomap Algorithm and Topological Stability. *Science*, v. 295, n. 5552, p. 7, 2002.

Behrisch, M.; Davey, J.; Fischer, F.; Thonnard, O.; Schreck, T.; Keim, D.; Kohlhammer, J. Visual analysis of sets of heterogeneous matrices using projection-based distance functions and semantic zoom. *Computer Graphics Forum*, v. 33, n. 3, p. 411–420, 2014.

Berkhin, P. *Survey of Clustering Data Mining Techniques*. Technical Report, Accrue Software, San Jose, CA, 2002.

Berry, M. W. Large Scale Sparse Singular Value Computations. *International Journal of Supercomputer Applications*, v. 6, p. 13–49, 1992.

Bertini, E.; Tatu, A.; Keim, D. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, v. 17, n. 12, p. 2203–2212, 2011.

Bianconi, F.; Fernández, A. Evaluation of the effects of Gabor filter parameters on texture classification. *Pattern Recognition*, v. 40, n. 12, p. 3325–3335, 2007.

Bingham, E.; Mannila, H. Random projection in dimensionality reduction: applications to image and text data. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, 2001, p. 245–250.

Bishop, C. M. *Pattern Recognition and Machine Learning*. New York: Springer, 2006. 738 p. (Information Science and Statistics).

Black, K. *Business Statistics: For Contemporary Decision Making*. 6th ed. United States of America: John Wiley & Sons, 2009. 864 p.

Blackford, L. S.; Demmel, J.; Dongarra, J.; Duff, I.; Hammarling, S.; Henry, G.; Heroux, M.; Kaufman, L.; Lumsdaine, A.; Petitet, A.; Pozo, R.; Remington, K.; Whaley, R. C. An updated set of basic linear algebra subprograms (BLAS). *ACM Transactions on Mathematical Software*, v. 28, n. 2, p. 135–151, 2002.

Borg, I.; Groenen, P. J. F. *Modern Multidimensional Scaling: Theory and Applications*. 2nd ed. New York: Springer, 2005. 614 p. (Springer Series in Statistics).

Boutsidis, C.; Drineas, P.; Magdon-Ismail, M. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, v. 43, n. 2, p. 1–27, 2014.

Braun, M. L.; Schaback, J.; Jugel, M. L.; Oury, N. jblas: Linear Algebra for Java. 2009. Disponível em: <`http://jblas.org/`>. Acesso em: 18 jan. 2011.

Bruneau, P.; Pinheiro, P.; Broeksema, B.; Otjacques, B. Cluster Sculptor, an interactive visual clustering system. *Neurocomputing*, v. 150, Part B, p. 627–644, 2015.

Camps-Valls, G.; Bruzzone, L. (Ed.). *Kernel Methods for Remote Sensing Data Analysis*. Chichester: John Wiley & Sons, 2009. 434 p.

Cardoso, Â.; Wichert, A. Iterative random projections for high-dimensional data clustering. *Pattern Recognition Letters*, v. 33, n. 13, p. 1749–1755, 2012.

Celikyilmaz, A.; Türksen, I. B. *Modeling Uncertainty with Fuzzy Logic: With Recent Theory and Applications*. Berlin: Springer, 2009. 400 p. (Studies in Fuzziness and Soft Computing, v. 240).

Chambers, J. M.; Cleveland, W. S.; Kleiner, B.; Tukey, P. A. *Graphical Methods for Data Analysis*. California: Wadsworth, 1983. 395 p.

Chan, K. P.; Cheung, Y. S. Fuzzy-attribute graph with application to Chinese character recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 22, n. 1, p. 153–160, 1992.

Chernoff, H. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, v. 68, n. 342, p. 361–368, 1973.

Choo, J.; Lee, C.; Reddy, C. K.; Park, H. UTOPIAN: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, v. 19, n. 12, p. 1992–2001, 2013.

Clark-Carter, D. Standard Deviation. In: Everitt, B.; Howell, D. (Ed.). *Encyclopedia of Statistics in Behavioral Science*, v. 4, Chichester: Wiley, p. 1891–1891, 2005.

Clerc, M. *Particle Swarm Optimization*. ISTE, 2006.

Cook, J. A.; Sutskever, I.; Mnih, A.; Hinton, G. E. Visualizing similarity data with a mixture of maps. In: *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, 2007, p. 67–74.

Da Silva, S. F.; Ribeiro, M. X.; Batista Neto, J. E. S.; Traina-Jr., C.; Traina, A. J. M. Improving the ranking quality of medical image retrieval using a genetic feature selection method. *Decision Support Systems*, v. 51, n. 4, p. 810–820, 2011.

Daniels II, J.; Anderson, E. W.; Nonato, L. G.; Silva, C. T. Interactive vector field feature identification. *IEEE Transactions on Visualization and Computer Graphics*, v. 16, n. 6, p. 1560–1568, 2010.

Datta, R.; Joshi, D.; Li, J.; Wang, J. Z. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, v. 40, n. 2, p. 5:1–5:60, 2008.

De Silva, V.; Tenenbaum, J. B. Global Versus Local Methods in Nonlinear Dimensionality Reduction. *Advances in Neural Information Processing Systems*, v. 15, p. 705–712, 2003.

De Silva, V.; Tenenbaum, J. B. *Sparse multidimensional scaling using landmark points*. Technical Report, Stanford, 2004.

Delaunay, B. N. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, v. 7, n. 6, p. 793–800, 1934.

Demiralp, Çağatay.; Hayden, E.; Hammerbacher, J.; Heer, J. invis: Exploring high-dimensional RNA sequences from in vitro selection. In: *IEEE Symposium on Biological Data Visualization (BioVis)*, Atlanta, GA, USA: IEEE, 2013, p. 1–8.

Dempster, A. P.; Laird, N. M.; Rubin, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, v. 39, n. 1, p. 1–38, 1977.

Deselaers, T.; Keysers, D.; Ney, H. Features for image retrieval: An experimental comparison. *Information Retrieval*, v. 11, n. 2, p. 77–107, 2008.

Domingues, H. H. *Espaços Métricos e Introdução à Topologia*. São Paulo: Atual, 1982. 183 p.

Dos Santos Amorim, E. P.; Brazil, E. V.; Daniels, J.; Joia, P.; Nonato, L. G.; Sousa, M. C. iLAMP: Exploring High-Dimensional Spacing through Backward Multidimensional Projection. In: *IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2012, p. 53–62.

Drineas, P.; Mahoney, M. W.; Muthukrishnan, S. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, v. 30, n. 2, p. 844–881, 2008.

Eades, P. A heuristic for graph drawing. *Congressus Numerantium*, v. 42, n. 11, p. 149–160, 1984.

Edelsbrunner, H.; Kirkpatrick, D. G.; Seidel, R. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, v. 29, n. 4, p. 551–559, 1983.

Eler, D. M.; Nakazaki, M. Y.; Paulovich, F. V.; Santos, D. P.; Andery, G. F.; Oliveira, M. C. F.; Batista Neto, J.; Minghim, R. Visual analysis of image collections. *The Visual Computer*, v. 25, n. 10, p. 923–937, 2009.

Fadel, S. G.; Fatore, F. M.; Duarte, F. S. L. G.; Paulovich, F. V. LoCH: A neighborhood-based multidimensional projection technique for high-dimensional sparse spaces. *Neurocomputing*, v. 150, n. Part B, p. 546–556, 2015.

Faloutsos, C.; Lin, K. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In: *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, 1995, p. 163–174.

Fei-Fei, L.; Fergus, R.; Perona, P. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. *IEEE Conference on Computer Vision and Pattern Recognition, Workshop on Generative-Model Based Vision*, v. 12, p. 178–186, 2004.

Ferreira de Oliveira, M. C.; Levkowitz, H. From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, v. 9, n. 3, p. 378–394, 2003.

Frank, A.; Asuncion, A. UCI machine learning repository. 2010. Disponível em: <`http://archive.ics.uci.edu/ml`>. Acesso em: 6 jan. 2011.

Gansner, E. R.; Hu, Y.; Kobourov, S. G. Visualizing graphs and clusters as maps. *IEEE Computer Graphics and Applications*, v. 30, n. 6, p. 54–66, 2010.

Gansner, E. R.; Hu, Y.; North, S. Visualizing streaming text data with dynamic graphs and maps. In: *Proceedings of the 20th International Conference on Graph Drawing*, Springer-Verlag, 2013, p. 439–450.

Ge, Y.; Li, S.; Lakhan, V. C.; Lucieer, A. Exploring uncertainty in remotely sensed data with parallel coordinate plots. *International Journal of Applied Earth Observation and Geoinformation*, v. 11, n. 6, p. 413–422, 2009.

Gil-Aluja, J. *Fuzzy Sets in the Management of Uncertainty*. Berlin: Springer, 2004. 420 p. (Studies in Fuzziness and Soft Computing, v. 145).

Gomez-Nieto, E.; Roman, F. S.; Pagliosa, P.; Casaca, W.; Helou, E. S.; De Oliveira, M. C. F.; Nonato, L. G. Similarity preserving snippet-based visualization of web search results. *IEEE Transactions on Visualization and Computer Graphics*, v. 20, n. 3, p. 457–470, 2014.

Gower, J.; Dijksterhuis, G. *Procrustes problems*. Oxford: Oxford University Press, 2004. 233 p. (Oxford Statistical Science Series, v. 30).

Guyon, I.; Gunn, S.; Ben-Hur, A.; Dror, G. Result Analysis of the NIPS 2003 Feature Selection Challenge. In: *Advances in Neural Information Processing Systems (NIPS 2004)*, MIT Press, 2004, p. 545–552.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. H. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, v. 11, n. 1, p. 10–18, 2009.

Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*. 3rd ed. Morgan Kaufmann, 2011. (The Morgan Kaufmann Series in Data Management Systems).

Härdle, W.; Simar, L. *Applied Multivariate Statistical Analysis*. 2nd ed. Springer-Verlag, 2007. 455 p.

Hinton, G.; Roweis, S. Stochastic Neighbor Embedding. In: *Advances in Neural Information Processing Systems 15*, MIT Press, 2002, p. 833–840.

Hochbaum, D. S.; Shmoys, D. B. A Best Possible Heuristic for the k -Center Problem. *Mathematics of Operations Research*, v. 10, n. 2, p. 180–184, 1985.

Hoffmann, H.; Schaal, S.; Vijayakumar, S. Local dimensionality reduction for non-parametric regression. *Neural Processing Letters*, v. 29, n. 2, p. 109–131, 2009.

Horn, R. A.; Johnson, C. *Matrix Analysis*. Cambridge: Cambridge University Press, 1990. 561 p.

Ilonen, J.; Kamarainen, J. Simplegabor - Multiresolution Gabor Feature Toolbox. 2006. Disponível em: <`http://www.it.lut.fi/project/simplegabor/downloads/src/simplegabortb`>. Acesso em: 15 jan. 2011.

Ilonen, J.; Kamarainen, J.-K.; Kalviainen, H. *Efficient computation of gabor features*. Technical Report 100, Department of Information Technology, Lappeenranta University of Technology, Lappeenranta, Finland, 2005.

Ingram, S.; Munzner, T.; Olano, M. Glimmer: Multilevel MDS on the GPU. *IEEE Transactions on Visualization and Computer Graphics*, v. 15, n. 2, p. 249–261, 2009.

Inselberg, A.; Dimsdale, B. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: *Proceedings of the First IEEE Conference on Visualization*, San Francisco, CA: IEEE, 1990, p. 361–378.

Izenman, A. J. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. New York: Springer, 2008. 731 p. (Springer Texts in Statistics).

Jacobs, R. A. Increased rates of convergence through learning rate adaptation. *Neural Networks*, v. 1, n. 4, p. 295–307, 1988.

Jain, A. K. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, v. 31, n. 8, p. 651–666, 2010.

Joia, P.; Coimbra, D.; Cuminato, J. A.; Paulovich, F. V.; Nonato, L. G. Local Affine Multidimensional Projection. *IEEE Transactions on Visualization and Computer Graphics*, v. 17, p. 2563–2571, 2011.

Joia, P.; Gomez-Nieto, E.; Batista Neto, J.; Casaca, W.; Botelho, G.; Paiva, A.; Gustavo Nonato, L. Class-specific metrics for multidimensional data projection applied to CBIR. *The Visual Computer*, v. 28, n. 10, p. 1027–1037, 2012.

Joia, P.; Petronetto, F.; Nonato, L. G. Uncovering Representative Groups in Multidimensional Projections. *Computer Graphics Forum*, v. 34, n. 3, 2015.

Jolliffe, I. T. *Principal Component Analysis*. 2nd ed. New York: Springer, 2002. 487 p. (Springer Series in Statistics).

Jourdan, F.; Melançon, G. Multiscale hybrid MDS. In: *Proceedings of the Eighth International Conference on Information Visualisation*, London, UK: IEEE Computer Society, 2004, p. 388–393.

Kandogan, E. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM Press, 2001, p. 107–116.

Keim, D. A. Designing Pixel-Oriented Visualization Techniques: Theory and Applications. *IEEE Transactions on Visualization and Computer Graphics*, v. 6, n. 1, p. 59–78, 2000.

Kim, H.; Choo, J.; Reddy, C. K.; Park, H. Doubly supervised embedding based on class labels and intrinsic clusters for high-dimensional data visualization. *Neurocomputing*, v. 150, Part B, p. 570–582, 2015.

Kirby, M. *Geometric data analysis: an empirical approach to dimensionality reduction and the study of patterns*. New York: John Wiley, 2001. 363 p.

Kiyadeh, A. P. H.; Zamiri, A.; Yazdi, H. S.; Ghaemi, H. Discernible visualization of high dimensional data using label information. *Applied Soft Computing*, v. 27, p. 474–486, 2015.

Knuth, D. E. *The Art of Computer Programming: Seminumerical Algorithms*, v. 2. 3rd ed. Boston: Addison-Wesley, 1997.

Kobayashi, T. Kernel-based transition probability toward similarity measure for semi-supervised learning. *Pattern Recognition*, v. 47, n. 5, p. 1994–2010, 2014.

Kohavi, R.; John, G. H. Wrappers for feature subset selection. *Artificial Intelligence*, v. 97, n. 1-2, p. 273–324, 1997.

Kokiopoulou, E.; Saad, Y. Orthogonal Neighborhood Preserving Projections: A projection-based dimensionality reduction technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 29, n. 12, p. 2143–2156, 2007.

Kosara, R.; Bendix, F.; Hauser, H. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, v. 12, n. 4, p. 558–568, 2006.

Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, v. 29, n. 1, p. 1–27, 1964.

Kullback, S.; Leibler, R. A. On Information and Sufficiency. *The Annals of Mathematical Statistics*, v. 22, n. 1, p. 79–86, 1951.

Kumar, D. A.; Esther, J. Comparative study on cbir based by color histogram, gabor and wavelet transform. *International Journal of Computer Applications*, v. 17, n. 3, p. 37–44, published by Foundation of Computer Science, 2011.

Lamping, J.; Rao, R.; Pirolli, P. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In: *Proceedings of the ACM Conference on Human Factors in Computing Systems*, ACM, 1995, p. 401–408.

Landauer, T. K.; Foltz, P. W.; Laham, D. An Introduction to Latent Semantic Analysis. *Discourse Processes*, v. 25, p. 259–284, 1998.

Landwehr, N.; Hall, M.; Frank, E. Logistic model trees. *Machine Learning*, v. 59, n. 1-2, p. 161–205, 2005.

Langtangen, H. P. *Python Scripting for Computational Science*. 3rd ed. Berlin: Springer, 2008. 750 p. (Texts in Computational Science and Engineering, v. 3).

Larose, D. T. *Data Mining Methods and Models*. Hoboken: John Wiley & Sons, 2006. 322 p.

LeBlanc, J.; Ward, M. O.; Wittels, N. Exploring N-dimensional Databases. In: *Proceedings of the First IEEE Conference on Visualization*, San Francisco, California: IEEE, 1990, p. 230–237.

Lee, J. A.; Peluffo-Ordóñez, D. H.; Verleysen, M. Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure. *Neurocomputing*, v. 169, p. 246–261, 2015.

Lee, J. A.; Renard, E.; Bernard, G.; Dupont, P.; Verleysen, M. Type 1 and 2 mixtures of Kullback-Leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomputing*, v. 112, p. 92–108, 2013.

Levenshtein, V. I. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR*, v. 163, n. 4, p. 845–848, 1965.

Li, J.; Wang, J. Z.; Wiederhold, G. IRM: integrated region matching for image retrieval. In: *Proceedings of the ACM Multimedia Conference*, Los Angeles, CA, USA: ACM, 2000, p. 147–156.

Lima, E. L. *Espaços Métricos*. 3. ed. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, CNPq, 1977. 299 p. (Projeto Euclides).

Liu, B. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. 2nd ed. Berlin: Springer, 2011. 622 p. (Data-Centric Systems and Applications).

Liu, C. Clarification of assumptions in the relationship between the Bayes decision rule and the whitened cosine similarity measure. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, 2008, p. 1116–1117.

Liu, C. Discriminant analysis and similarity measure. *Pattern Recognition*, v. 47, n. 1, p. 359–367, 2014.

Liu, C.; Frazier, P.; Kumar, L. Comparative assessment of the measures of thematic classification accuracy. *Remote Sensing of Environment*, v. 107, n. 4, p. 606–616, 2007.

Lovie, P. Coefficient of Variation. In: Everitt, B.; Howell, D. (Ed.). *Encyclopedia of Statistics in Behavioral Science*, v. 1, Chichester: Wiley, p. 317–318, 2005.

Lux, M.; Chatzichristofis, S. A. Lire: Lucene Image Retrieval: An Extensible Java CBIR Library. In: *Proceedings of the 16th ACM International Conference on Multimedia*, New York, NY, USA: ACM, 2008, p. 1085–1088.

Maaten, L. V. D. *An Introduction to Dimensionality Reduction Using Matlab*. Technical Report MICC 07-07, Maastricht University, Maastricht, The Netherlands, 2007.

Maaten, L. V. D. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, v. 15, p. 3221–3245, 2014.

Maaten, L. V. D.; Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, v. 9, p. 2579–2605, 2008.

Maaten, L. V. D.; Postma, E. O.; Herik, H. J. V. D. *Dimensionality Reduction: A Comparative Review*. Technical Report TiCC-TR 2009-005, Tilburg University, 2009.

Macqueen, J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, Calif.: University of California Press, 1967, p. 281–297.

Maheshwary, P.; Srivastav, N. Retrieving similar image using color moment feature detector and k-means clustering of remote sensing images. In: *Proceedings of the 2008 International Conference on Computer and Electrical Engineering*, 2008, p. 821–824.

Mahoney, M. W.; Drineas, P. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, v. 106, n. 3, p. 697–702, 2009.

Maimon, O.; Rokach, L. (Ed.). *Data Mining and Knowledge Discovery Handbook*. 2nd ed. New York: Springer, 2010. 1285 p.

Mamani, G. M. H.; Fatore, F. M.; Nonato, L. G.; Paulovich, F. V. User-driven feature space transformation. *Computer Graphics Forum*, v. 32, n. 3pt3, p. 291–299, 2013.

Marneffe, M.-C.; MacCartney, B.; Manning, C. D. Generating typed dependency parses from phrase structure parses. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006)*, Genoa, Italy: European Language Resources Association (ELRA), 2006, p. 449–454.

Martins, R. M.; Andery, G. F.; Heberle, H.; Paulovich, F. V.; De Andrade Lopes, A.; Pedrini, H.; Minghim, R. Multidimensional projections for visual analysis of social networks. In: *Journal of Computer Science and Technology*, 2012, p. 791–810.

Martins, R. M.; Coimbra, D. B.; Minghim, R.; Telea, A. C. Visual analysis of dimensionality reduction quality for parameterized projections. *Computers & Graphics*, v. 41, p. 26–42, 2014.

Masseglia, F.; Poncelet, P.; Teisseire, M. *Successes and New Directions in Data Mining*. Hershey: Information Science Reference, 2007. 369 p.

Mazza, R. *Introduction to Information Visualization*. London: Springer, 2009. 139 p.

Meyer, C. D. *Matrix Analysis and Applied Linear Algebra*. Philadelphia: SIAM, 2000. 718 p.

Miyamoto, S.; Ichihashi, H.; Honda, K. *Algorithms for Fuzzy Clustering: Methods in c-Means Clustering with Applications*. Berlin: Springer, 2008. (Studies in Fuzziness and Soft Computing, v. 229).

Molchanov, V.; Linsen, L. Visual Exploration of Patterns in Multi-run Time-varying Multi-field Simulation Data Using Projected Views. In: *The 22nd International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)*, Plzen, Czech Republic, 2014, p. 39–48.

Motta, R.; Minghim, R.; de Andrade Lopes, A.; Oliveira, M. C. F. Graph-based measures to assist user assessment of multidimensional projections. *Neurocomputing*, v. 150, n. Part B, p. 583–598, 2015.

Ng, A. Y.; Jordan, M. I.; Weiss, Y. On Spectral Clustering: Analysis and an algorithm. In: Dietterich, T. G.; Becker, S.; Ghahramani, Z. (Ed.). *Advances in Neural Information Processing Systems 14 (NIPS 2001)*, v. 1, MIT Press, p. 849–856, 2002.

Noack, A. Modularity clustering is force-directed layout. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, v. 79, n. 2, p. 026102, 2009.

Okada, C. Y.; Pedronette, D. C. G.; Torres, R. S. Unsupervised Distance Learning by Rank Correlation Measures for Image Retrieval. In: *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (ICMR 2015)*, Shangai, China: ACM Press, 2015, p. 331–338.

Olson, D. L.; Delen, D. *Advanced Data Mining Techniques*. Berlin: Springer, 2008. 180 p.

Paiva, J. G. S.; Schwartz, W. R.; Pedrini, H.; Minghim, R. An Approach to Supporting Incremental Visual Data Classification. *Visualization and Computer Graphics, IEEE Transactions on*, v. 21, n. 1, p. 4–17, 2015.

Palumbo, F.; Vistocco, D.; Morineau, A. Huge Multidimensional Data Visualization: Back to the Virtue of Principal Coordinates and Dendrograms in the New Computer Age. In: Chen, C.; Härdle, W.; Unwin, A. (Ed.). *Handbook of Data Visualization*, cap. III.4, Berlin: Springer, p. 349–387, 2008.

Papakostas, G. A.; Hatzimichailidis, A. G.; Kaburlasos, V. G. Distance and similarity measures between intuitionistic fuzzy sets: A comparative analysis from a pattern recognition point of view. *Pattern Recognition Letters*, v. 34, n. 14, p. 1609–1622, 2013.

Paschou, P.; Mahoney, M. W.; Javed, A.; Kidd, J. R.; Pakstis, A. J.; Gu, S.; Kidd, K. K.; Drineas, P. Intra- and interpopulation genotype reconstruction from tagging SNPs. *Genome Research*, v. 17, n. 1, p. 96–107, 2007.

Paulovich, F. V.; Eler, D. M.; Poco, J.; Botha, C. P.; Minghim, R.; Nonato, L. G. Piecewise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum*, v. 30, n. 3, p. 1091–1100, 2011.

Paulovich, F. V.; Eler, D. M.; Poco, J.; Nonato, L. G. *A Fast Projection Technique and its Applications to Visualization of Large Data Sets*. Technical Report 349, Universidade Estadual de São Paulo, ICMC, São Carlos, SP, 2010a.

Paulovich, F. V.; Minghim, R. HiPP: A Novel Hierarchical Point Placement Strategy and its Application to the Exploration of Document Collections. *IEEE Transactions on Visualization and Computer Graphics*, v. 14, n. 6, p. 1229–1236, 2008.

Paulovich, F. V.; Nonato, L. G.; Minghim, R.; Levkowitz, H. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *Visualization and Computer Graphics, IEEE Transactions on*, v. 14, n. 3, p. 564–575, 2008.

Paulovich, F. V.; Silva, C. T.; Nonato, L. G. Two-phase mapping for projecting massive data sets. *Visualization and Computer Graphics, IEEE Transactions on*, v. 16, n. 6, p. 1281–1290, 2010b.

Paulovich, F. V.; Toledo, F. M. B.; Telles, G. P.; Minghim, R.; Nonato, L. G. Semantic Wordification of Document Collections. *Computer Graphics Forum*, v. 31, n. 3, p. 1145–1153, 2012.

Pearson, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, v. 2, n. 6, p. 559–572, 1901.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

Pedronette, D. C. G.; Torres, R. S. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, v. 46, n. 8, p. 2350–2360, 2013.

Peebles, P. Z. *Probability, Random Variables, and Random Signal Principles*. 4th ed. New York: McGraw-Hill, 2001. 462 p.

Pekalska, E.; Ridder, D. D.; Duin, R. P. W.; Kraaijveld, M. A. A new method of generalizing Sammon mapping with application to algorithm speed-up. In: *Proceedings of the 5th Annual Conference of the Advanced School for Computing and Imaging (ASCI1999)*, Delft, Netherlands, 1999, p. 221–228.

Pelleg, D.; Moore, A. W. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: *Proceedings of the 17th International Conference on Machine Learning (ICML)*, Stanford University, Stanford, CA, USA: Morgan Kaufmann, 2000, p. 727–734.

Poco, J.; Eler, D. M.; Paulovich, F. V.; Minghim, R. Employing 2D Projections for Fast Visual Exploration of Large Fiber Tracking Data. *Computer Graphics Forum*, v. 31, n. 3, p. 1075–1084, 2012.

Rencher, A. C. *Methods of Multivariate Analysis*. 2nd ed. Danvers: John Wiley & Sons, 2002. 708 p. (Wiley Series in Probability and Statistics).

Richardson, M. W. Multidimensional psychophysics. *Psychological Bulletin*, v. 35, p. 659–660, 1938.

Robertson, G. G.; Mackinlay, J. D.; Card, S. K. Cone Trees: Animated 3D Visualizations of Hierarchical Information. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, 1991, p. 189–194.

Rocchio, J. J. Relevance feedback in information retrieval. In: Salton, G. (Ed.). *The SMART Retrieval System: Experiments in Automatic Document Processing*, Englewood Cliffs: Prentice-Hall, p. 313–323, 1971.

Rohde, D. SVDLIBC: A C Library for Computing Singular Value Decompositions. 2002. Disponível em: <`http://tedlab.mit.edu/~dr/SVDLIBC/`>. Acesso em: 12 out. 2014.

Ross, S. M. *Introduction to Probability Models*. 11th ed. Oxford: Academic Press, 2014. 767 p.

Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, v. 20, p. 53–65, 1987.

Roweis, S. T.; Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science*, v. 290, n. 5500, p. 2323–2326, 2000.

Sammon, J. W. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, v. 18, n. 5, p. 401–409, 1969.

Santos, T. S. R.; Paulovich, F. V.; Molchanov, V.; Linsen, L.; de Oliveira, M. C. F. Visualizing Temporal Behavior in Multifield Particle Simulations. *Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications*, p. 573–582, 2013.

Schaefer, S.; McPhail, T.; Warren, J. Image deformation using moving least squares. *ACM Transactions On Graphics*, v. 25, n. 3, p. 533–540, 2006.

Schneider, W. Pittsburgh Brain Competition (PBC) – Brain Connectivity Challenge. 2009. Disponível em: <`http://pbc.lrdc.pitt.edu/`>. Acesso em: 2 jan. 2011.

Seber, G. A. F. *Multivariate observations*. Hoboken: Wiley, 1984. 686 p.

Seo, J.; Shneiderman, B. Application Examples of the Hierarchical Clustering Explorer. 2004. Disponível em: <`http://www.cs.umd.edu/hcil/hce/examples/application_examples.html`>. Acesso em: 6 jan. 2011.

Shapiro, L. G.; Haralick, R. M. A metric for comparing relational descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 7, n. 1, p. 90–94, 1985.

Shneiderman, B. Tree Visualization With Treemaps: A 2D Space-filling Approach. *ACM Transactions On Graphics*, p. 92–99, 1991.

Sips, M.; Neubert, B.; Lewis, J. P.; Hanrahan, P. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, v. 28, n. 3, p. 831–838, 2009.

Smithson, M.; Verkuilen, J. *Fuzzy Set Theory: Applications in the Social Sciences*. Thousand Oaks: SAGE, 2006. 97 p. (Quantitative Applications in the Social Sciences, n. 07-147).

Soriano, A.; Paulovich, F.; Nonato, L. G.; Oliveira, M. C. F. Visualization of music collections based on structural content similarity. In: *27th Conference on Graphics, Patterns and Images (SIBGRAPI)*, IEEE, 2014, p. 25–32.

Sorkine, O.; Cohen-Or, D. Least-squares Meshes. In: *Proceedings of the Shape Modeling International (SMI2004)*, Genova, Italy: IEEE Computer Society, 2004, p. 191–199.

Sorkine, O.; Cohen-Or, D.; Lipman, Y.; Alexa, M.; Rössl, C.; H.-P.Seidel Laplacian surface editing. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, New York, NY, USA: ACM, 2004, p. 175–184.

Steele, J.; Iliinsky, N. (Ed.). *Beautiful Visualization: Looking at Data Through the Eyes of Experts*. Beijing: O'Reilly, 2010. 397 p.

Stehman, S. V. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, v. 62, n. 1, p. 77–89, 1997.

Steiger, M.; Bernard, J.; Mittelstädt, S.; Lücke-Tieke, H.; Keim, D.; May, T.; Kohlhammer, J. Visual analysis of time-series similarities for anomaly detection in sensor networks. *Computer Graphics Forum*, v. 33, n. 3, p. 401–410, 2014.

Steinbach, M.; Karypis, G.; Kumar, V. A comparison of document clustering techniques. In: *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining*, Boston, MA, USA: ACM, 2000, p. 109–110.

Steyvers, M. Multidimensional Scaling. In: *Encyclopedia of Cognitive Science*, Macmillan, p. 1–7, 2002.

Struc, V.; Pavesic, N. The corrected normalized correlation coefficient: A novel way of matching score calculation for LDA-based face verification. In: *The Fifth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2008)*, Jinan, Shandong, China: IEEE, 2008, p. 110–115.

Tamura, H.; Mori, S.; Yamawaki, T. Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 8, n. 6, p. 460–473, 1978.

Tan, P.-N.; Steinbach, M.; Kumar, V. *Introduction to Data Mining*. Boston: Addison-Wesley, 2005. 769 p.

Tan, S.; Liu, L.; Peng, C.; Shao, L. Image-to-class distance ratio: A feature filtering metric for image classification. *Neurocomputing*, v. 165, p. 211–221, 2015.

Tejada, E.; Minghim, R.; Nonato, L. G. On improved projection techniques to support visual exploration of multidimensional data sets. *Information Visualization*, v. 2, n. 4, p. 218–231, 2003.

Tenenbaum, J. B.; de Silva, V.; Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science*, v. 290, n. 5500, p. 2319–2323, 2000.

Theodoridis, S.; Koutroumbas, K. *Pattern recognition*. 3rd ed. San Diego: Academic Press, 2006. 837 p.

Tillé, Y. *Sampling Algorithms*. New York: Springer, 2006. 216 p. (Springer Series in Statistics).

Torgerson, W. S. Multidimensional scaling: I. theory and method. *Psychometrika*, v. 17, n. 4, p. 401–419, 1952.

Torgerson, W. S. Multidimensional scaling of similarity. *Psychometrika*, v. 30, n. 4, p. 379–393, 1965.

Trodd, N. M. Uncertainty in land cover mapping for modelling land cover change. In: *Proceedings of Remote Sensing in Action*, Nottingham, U.K.: Remote Sensing Society, 1995, p. 1138–1145.

Velleman, P. F.; Welsch, R. E. Efficient Computing of Regression Diagnostics. *The American Statistician*, v. 35, n. 4, p. 234–242, 1981.

Venna, J.; Peltonen, J.; Nybo, K.; Aidos, H.; Kaski, S. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, v. 11, p. 451–490, 2010.

Vitter, J. S. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, v. 11, n. 1, p. 37–57, 1985.

Wang, J.; Zhao, J.; Guo, S.; North, C.; Ramakrishnan, N. ReCloud: Semantics-based word cloud visualization of user reviews. In: *Proceedings of Graphics Interface*, Montreal, QC, Canada: Canadian Information Processing Society, 2014, p. 151–158.

Wang, J. Z.; Li, J.; Wiederhold, G. SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, n. 9, p. 947–963, 2001.

Webb, A. *Statistical Pattern Recognition*. 2nd ed. London: Wiley, 2002. 496 p.

Whalen, D.; Norman, M. L. Competition data set and description. In: *IEEE Visualization Design Contest*, 2008. Disponível em: <`http://vis.computer.org/VisWeek2008/vis/contests.html`>. Acesso em: 4 jan. 2011.

Whitrow, R. *OpenGL Graphics Through Applications*. London: Springer, 2008. 330 p.

Winn, J.; Criminisi, A.; Minka, T. Object categorization by learned universal visual dictionary. In: *Proceedings of the IEEE International Conference on Computer Vision*, Beijing, China: IEEE Computer Society, 2005, p. 1800–1807.

Witten, I. H.; Frank, E.; Hall, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd ed. Burlington, MA: Morgan Kaufmann, 2011. 630 p. (The Morgan Kaufmann Series in Data Management Systems).

Wu, Y.; Provan, T.; Wei, F.; Liu, S.; Ma, K.-L. Semantic-preserving word clouds by seam carving. *Computer Graphics Forum*, v. 30, n. 3, p. 741–750, 2011.

Xu, K.; Zhang, H.; Cohen-Or, D.; Xiong, Y. Dynamic harmonic fields for surface processing. *Computers and Graphics*, v. 33, n. 3, p. 391–398, 2009.

Yager, R. R.; Filev, D. P. *Essentials of Fuzzy Modeling and Control*. New York: Wiley, 1994. 388 p.

Yong, G.; Householder, A. S. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, v. 3, n. 1, p. 19–22, 1938.

Yu, C. *High-Dimensional Indexing: Transformational Approaches to High-Dimensional Range and Similarity Searches*. Berlin: Springer, 2002. 150 p. (Lecture Notes in Computer Science, v. 2341).

Zezula, P.; Amato, G.; Dohnal, V.; Batko, M. *Similarity Search: The Metric Space Approach*. New York: Springer, 2005. 220 p. (Advances in Database Systems).

Zhang, J. *Visualization for Information Retrieval*. Berlin: Springer, 2008. 292 p. (The Information Retrieval Series).

Zhao, M.; Chow, T. W. S.; Zhang, Z. Random walk-based fuzzy linear discriminant analysis for dimensionality reduction. *Soft Computing*, v. 16, n. 8, p. 1393–1409, 2012.

Apêndice

A

Lista de Publicações

Artigos publicados relacionados ao tema desta tese:

- **Joia, P.**; Coimbra, D.; Culminato, J. A.; Paulovich, F. V.; Nonato, L. G. Local Affine Multidimensional Projection, *IEEE Transactions on Visualization and Computer Graphics*, v. 17, n. 12, p. 2563-2571, 2011.
  Menção honrosa na Conferência IEEE InfoVis 2011.

- **Joia, P.**; Gomez Nieto, E.; Botelho, G.; Batista Neto, J.; Paiva, A.; Nonato, L. G. Projection-based Image Retrieval using Class-Specific Metrics. In: *SIBGRAPI'11: XXIV Conference on Graphics, Patterns and Images*, IEEE Computer Society Conference Publishing Services, 2011, p. 125–132.
  Melhor artigo de Visualização SIBGRAPI 2011.

- **Joia, P.**; Gomez-Nieto, E.; Batista Neto, J.; Casaca, W.; Botelho, G.; Paiva, A.; Nonato, L. G. Class-Specific Metrics for Multidimensional Data Projection Applied to CBIR, *The Visual Computer Journal*, v. 28, n. 10, p. 1027-1037, 2012.

- **Joia, P.**; Petronetto, F.; Nonato, L. G. Uncovering Representative Groups in Multidimensional Projections, *Computer Graphics Forum*, The Eurographics Association and John Wiley & Sons Ltd., v. 34, n. 3, 2015.

Artigo submetido, em processo de revisão:

- **Joia, P.**; da Silva, S. F.; Batista, J.; Nonato, L. G. Class-specific metrics with weights and uncertainty modeling using fuzzy sets applied to content-based image retrieval, *Expert Systems with Applications*, Elsevier, 2015.

Artigos publicados em colaboração com outros pesquisadores:

- Dos Santos Amorim, E. P.; Brazil, E. V., Daniels, J.; **Joia, P.**; Nonato, L. G.; Sousa, M. C. iLAMP: Exploring High-Dimensional Spacing through Backward Multidimensional Projection. In: *IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2012, p. 53-62.

- Casaca, W.; Paiva, A.; Gomez-Nieto, E.; **Joia, P.**; Nonato, L. G. Spectral Image Segmentation Using Image Decomposition and Inner Product-Based Metric, *Journal of Mathematical Imaging and Vision*, Springer Netherlands, v. 45, n. 3, p. 227-238, 2013.