
Técnicas computacionais de apoio à classificação
visual de imagens e outros dados

José Gustavo de Souza Paiva

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 04 de fevereiro de 2013

Assinatura: _____

Técnicas computacionais de apoio à classificação visual de imagens e outros dados

José Gustavo de Souza Paiva

Orientadora: *Profa. Dra. Rosane Minghim*

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*

USP – São Carlos
Fevereiro de 2013

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

P142t Paiva, José Gustavo de Souza
Técnicas computacionais de apoio à classificação
visual de imagens e outros dados / José Gustavo de
Souza Paiva; orientadora Rosane Minghim. -- São
Carlos, 2013.
130 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, 2013.

1. Classificação Visual de Imagens. 2. Árvores de
Similaridade. 3. Redução de Dimensionalidade. I.
Minghim, Rosane, orient. II. Título.

Agradecimentos

Agradeço a Deus por ter me dado a sabedoria e persistência para vencer mais esse desafio em minha vida.

Agradeço a meus pais e irmãos, que mesmo imaginando o caminho árduo pelo qual eu passaria durante esse período, me apoiaram e incentivaram, como sempre fizeram durante toda a minha vida.

Agradeço a Elaine, minha amada esposa, que soube mais uma vez, como ninguém, me apoiar, incentivar e aguentar nos momentos mais difíceis (e não foram poucos) deste trabalho, além de compartilhar todos as vitórias, se mostrando acima de tudo uma verdadeira amiga e companheira. Nunca vou me esquecer disso.

Agradeço à minha orientadora, Profa. Dra. Rosane Minghim, que durante o período de orientação soube transmitir os ensinamentos fundamentais para o desenvolvimento de um trabalho de qualidade e aplicabilidade, além de contribuir significativamente com o meu amadurecimento como pesquisador, cujas atribuições me sinto agora apto a exercer. Espero continuar essa parceria em muitos projetos futuros.

Agradeço ao Prof. Dr. Guilherme Pimentel Telles, pelas ideias e sugestões de grande valor na implementação e modificação das árvores de similaridade. Agradeço também ao Prof. Dr. Hélio Pedrini e ao Prof. Dr. William Robson Schwartz, pela participação fundamental na sugestão de abordagens e soluções relativas à classificação de imagens, pela indicação e concessão de artigos e códigos base das técnicas PLS e LWPR, além do fornecimento de algumas das coleções utilizadas nos experimentos. Também espero que possamos trabalhar juntos em pesquisas futuras.

Agradeço aos colegas do grupo VICG, que me auxiliaram na construção de diversos componentes para o sistema VisPipeline, facilitando consideravelmente o desenvolvimento das ideias propostas na tese.

Agradeço a Universidade de São Paulo - ICMC - São Carlos, por me acolher durante o período do doutorado, e fornecer toda a infraestrutura necessária e recursos financeiros para o desenvolvimento do projeto.

Resumo

O processo automático de classificação de dados em geral, e em particular de classificação de imagens, é uma tarefa computacionalmente intensiva e variável em termos de precisão, sendo consideravelmente dependente da configuração do classificador e da representação dos dados utilizada. Muitos dos fatores que afetam uma adequada aplicação dos métodos de classificação ou categorização para imagens apontam para a necessidade de uma maior interferência do usuário no processo. Para isso são necessárias mais ferramentas de apoio às várias etapas do processo de classificação, tais como, mas não limitadas, a extração de características, a parametrização dos algoritmos de classificação e a escolha de instâncias de treinamento adequadas. Este doutorado apresenta uma metodologia para Classificação Visual de Imagens, baseada na inserção do usuário no processo de classificação automática através do uso de técnicas de visualização. A ideia é permitir que o usuário participe de todos os passos da classificação de determinada coleção, realizando ajustes e consequentemente melhorando os resultados de acordo com suas necessidades. Um estudo de diversas técnicas de visualização candidatas para a tarefa é apresentado, com destaque para as árvores de similaridade, sendo apresentadas melhorias do algoritmo de construção em termos de escalabilidade visual e de tempo de processamento. Adicionalmente, uma metodologia de redução de dimensionalidade visual semi-supervisionada é apresentada para apoiar, pela utilização de ferramentas visuais, a criação de espaços reduzidos que melhorem as características de segregação do conjunto original de características. A principal contribuição do trabalho é um sistema de classificação visual incremental que incorpora todos os passos da metodologia proposta, oferecendo ferramentas interativas e visuais que permitem a interferência do usuário na classificação de coleções incrementais com configuração de classes variável. Isso possibilita a utilização do conhecimento do ser humano na construção de classificadores que se adequem a diferentes necessidades dos usuários em diferentes cenários, produzindo resultados satisfatórios para coleções de dados diversas. O foco desta tese é em categorização de coleções de imagens, com exemplos também para conjuntos de dados textuais.

Abstract

Automatic data classification in general, and image classification in particular, are computationally intensive tasks with variable results concerning precision, being considerably dependent on the classifier's configuration and data representation. Many of the factors that affect an adequate application of classification or categorization methods for images point to the need for more user interference in the process. To accomplish that, it is necessary to develop a larger set of supporting tools for the various stages of the classification set up, such as, but not limited to, feature extraction, parametrization of the classification algorithm and selection of adequate training instances. This doctoral Thesis presents a Visual Image Classification methodology based on the user's insertion in the classification process through the use of visualization techniques. The idea is to allow the user to participate in all classification steps, adjusting several stages and consequently improving the results according to his or her needs. A study on several candidate visualization techniques is presented, with emphasis on similarity trees, and improvements of the tree construction algorithm, both in visual and time scalability, are shown. Additionally, a visual semi-supervised dimensionality reduction methodology was developed to support, through the use of visual tools, the creation of reduced spaces that improve segregation of the original feature space. The main contribution of this work is an incremental visual classification system incorporating all the steps of the proposed methodology, and providing interactive and visual tools that permit user controlled classification of an incremental collection with evolving class configuration. It allows the use of the human knowledge on the construction of classifiers that adapt to different user needs in different scenarios, producing satisfactory results for several data collections. The focus of this Thesis is image data sets, with examples also in classification of textual collections.

Sumário

Resumo	iii
Abstract	v
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	4
1.3 Contribuições	4
1.4 Organização do Texto	7
2 Trabalhos Relacionados	9
2.1 Considerações Iniciais	9
2.2 Representação de Coleções de Dados	9
2.3 Visualização de Informação	12
2.3.1 Técnicas de Visualização baseadas em Atributos	13
2.3.2 Técnicas de Visualização baseadas em Posicionamento de Pontos	15
2.3.3 Técnicas de Visualização aplicadas a Coleções de Imagem	24
2.4 Técnicas de Redução de Dimensionalidade	33
2.5 Classificação Visual de Coleções de Dados	36
2.6 Considerações Finais	39
3 Árvores de Similaridade para Visualização de Coleções de Dados	41
3.1 Considerações Iniciais	41
3.2 Árvore de Similaridade <i>Neighbor Joining</i>	42
3.3 Abordagens para melhorias na construção de árvores <i>Neighbor Joining</i>	47
3.3.1 Promoção de Nós	47
3.3.2 Algoritmos para Aceleração na Construção de Árvores NJ	51
3.4 Árvores de Similaridade para Classificação Visual de Imagens	53
3.5 Considerações Finais	54
4 Redução Visual de Dimensionalidade Semi-supervisionada	57
4.1 Considerações Iniciais	57
4.2 <i>Partial Least Squares</i> (PLS)	58

4.2.1	Descrição da Técnica	59
4.2.2	Abordagens de Utilização	61
4.3	Metodologia de Redução Visual de Dimensionalidade PLS	62
4.4	Análise de Resultados	65
4.5	Classificação PLS	75
4.6	Considerações Finais	76
5	Classificação Visual Incremental de Dados	79
5.1	Considerações Iniciais	79
5.2	Metodologia de Classificação Visual Incremental	80
5.2.1	<i>Locally Weighted Projection Regression</i>	80
5.2.2	Descrição da Metodologia	83
5.3	Análise de Resultados	87
5.3.1	Escolha das Instâncias	88
5.3.2	Construção e Atualização do Modelo LWPR	89
5.3.3	Classificação Iterativa	92
5.3.4	Evolução das Classes do Problema	93
5.3.5	Alteração da Perspectiva de Classificação	97
5.4	Sistema de Classificação Visual de Imagens	100
5.4.1	Rotulamento por Seleção	100
5.4.2	Rotulamento Individual	102
5.4.3	<i>Class Matching</i>	103
5.4.4	Descrição do Sistema	103
5.5	Considerações Finais	106
6	Conclusões	109
6.1	Contribuições	109
6.2	Trabalhos Futuros	113
Referências Bibliográficas		115
A	Nomenclatura	131
B	Artigo: Improved Similarity Trees and their Application to Visual Data Classification	133
C	Artigo: Semi-Supervised Dimensionality Reduction based on Partial Least Squares for Visual Analysis of High Dimensional Data	135
D	Artigo: Incremental Visual Data Classification using Locally Weighted Projection Regression	137

Lista de Figuras

2.1	Processo de Visualização (Adaptado de (Card et al., 1999)).	13
2.2	Coordenadas Paralelas de uma coleção de dados com 7500 instâncias, cada uma com 5 atributos, ilustrando a confusão visual (<i>cluttering</i>) causado pelo número de instâncias (Adaptado de (Artero et al., 2004)).	14
2.3	Visualizações baseadas em pixel, representando uma coleção com 16350 instâncias contendo 9 informações da bolsa de valores, coletadas entre janeiro de 1987 até março de 1993, usando arranjos Peano-Hilbert (2.3a) e Morton (2.3b) (Adaptado de (Keim & Ankerst, 2001)).	15
(a)	Peano-Hilbert.	15
(b)	Morton.	15
2.4	Mapeamento de uma instância em um ponto p , utilizando a técnica RadViz (Adaptado de (Novakova & Stepankova, 2009)).	17
2.5	Projeção LSP de uma coleção de imagens divididas em 10 classes.	19
2.6	Projeção PLP de uma coleção de imagens, antes e depois de uma sequência de manipulações realizadas pelo usuário (Adaptado de (Paulovich et al., 2011)). . . .	21
(a)	Projeção inicial da coleção.	21
(b)	Projeção após sequência de movimentações nos pontos de controle.	21
2.7	Projeção LAMP de uma coleção com 7 classes, com destaque para os pontos de controle, escolhidos aleatoriamente na coleção (Adaptado de (Joaia et al., 2011)). . .	22
(a)	3 pontos de controle por classe.	22
(b)	Projeção LAMP relativa aos pontos de controle selecionados.	22
2.8	Modelos gerados utilizando-se MDS Convencional e <i>Proximity Grid</i> , para a mesma coleção de imagens (Adaptado de (Basalaj et al., 1999)).	25
(a)	MDS Convencional.	25
(b)	Proximity Grid.	25
2.9	Visualização da rede de similaridade no sistema AETOS, utilizado o método NN^k (Adaptado de (May, 2004)).	26
2.10	Software iBase mostrando a rede criada que representa os resultados de determinada consulta (Adaptado de (May, 2004)).	27
2.11	Interface do software Pex-Image, ilustrando o resultado de uma projeção.	28
2.12	Coordenação entre duas projeções realizadas pelo Pex-Image, com destaque para um grupo de imagens.	29

2.13	Tela do sistema de visualização, ilustrando o <i>layout</i> baseado em distâncias (Adaptado de (Nguyen & Worring, 2008)).	30
2.14	Exemplo de seleção de imagem representativa em uma coleção de imagens, e posterior visualização do grupo correspondente (Adaptado de (Nguyen & Worring, 2008)).	30
2.15	Visualização de um subconjunto da coleção COREL com imagens e termos textuais associados, utilizando <i>Non-negative Matrix Factorization (NMF)</i> (Adaptado de (Camargo et al., 2010)).	31
2.16	Tela do sistema AIRS, mostrando uma visão baseada em grafo e outra baseada em Coordenadas Paralelas, para um conjunto de 20 imagens correspondentes ao resultado do processo de <i>Relevance Feedback</i> (Adaptado de (Doloc-Mihu, 2011)). .	32
2.17	Visualização de coleções de imagens utilizando grade quadrada (2.17a) e grade hexagonal (2.17b) (Adaptado de (Schaefer, 2011))	32
(a)	<i>Hue Sphere Image Browser</i>	32
(b)	<i>Honeycomb Image Browser</i>	32
2.18	Visualização do processo de classificação através de uma árvore de decisão exibida pelo sistema VDM-RS (Adaptado de (Zhang et al., 2009)).	39
3.1	Comparação entre uma projeção LSP e uma árvore NJ, para uma coleção de 675 documentos textuais.	44
(a)	Projeção LSP.	44
(b)	Árvore NJ.	44
(c)	Árvore NJ após aplicação de <i>layout</i> baseado em força.	44
3.2	Exemplo de uma árvore NJ para a coleção COREL.	44
(a)	Árvore NJ, com instâncias representadas por círculos.	44
(b)	Árvore NJ, com instâncias representadas por imagens.	44
3.3	Seleção e exploração de um ramo da árvore NJ construída a partir da coleção COREL, mostrando os níveis de similaridade	45
(a)	Árvore NJ construída a partir da coleção COREL.	45
(b)	Detalhamento do ramo selecionado em (a), com imagens de ônibus.	45
3.4	Análises de precisão relativas à árvore da coleção COREL.	46
(a)	<i>Neighborhood Preservation</i> - Coleção COREL.	46
(b)	Gráfico de distâncias - Coleção COREL.	46
3.5	Consistência na precisão do ramo com imagens de ônibus, em comparação com a árvore completa mostrada na Figura 3.4a.	46
(a)	<i>Neighborhood Preservation</i> - Ramo Ônibus.	46
(b)	Gráfico de distâncias - Ramo Ônibus.	46
3.6	Exemplo de árvore NJ de uma coleção com 648 documentos textuais, com seus nós virtuais e arestas destacados, mostrando a alta densidade de pontos gerados pela técnica.	48
3.7	Operação de promoção de nós. Círculos preenchidos representam nós da coleção, e triângulos representam subárvore.	48
(a)	Exemplo.	48
(b)	Padrão suscetível à promoção.	48
(c)	Substituição.	48

3.8	Comparação entre árvores NJ e PNJ para a coleção COREL, destacando os nós virtuais e arestas.	50
(a)	Árvore NJ para coleção COREL.	50
(b)	Árvore PNJ para coleção COREL.	50
3.9	Análise <i>Neighborhood Preservation</i> comparando Árvores NJ e PNJ.	50
3.10	Gráfico de Distâncias comparando Árvores NJ e PNJ para a coleção COREL. . . .	51
(a)	Árvore NJ.	51
(b)	Árvore PNJ.	51
3.11	Análise da medida <i>Neighborhood Preservation</i> , comparando algoritmos de criação de Árvores NJ e Projeção LSP para a coleção COREL.	53
3.12	Gráfico de Distâncias comparando algoritmos de criação de Árvores NJ e Projeção LSP para a coleção COREL.	54
(a)	Rapid NJ COREL.	54
(b)	Fast NJ COREL.	54
(c)	LSP COREL.	54
3.13	Sequência de passos do processo de classificação visual.	55
(a)	<i>Ground Truth</i> : 350 imagens.	55
(b)	Classificação: 350 imagens.	55
(c)	<i>Class Matching</i>	55
(d)	<i>Ground Truth</i> : 400 imagens.	55
(e)	Classificação: 400 imagens.	55
(f)	<i>Class Matching</i>	55
(g)	<i>Ground Truth</i> : 450 imagens.	55
(h)	Classificação: 450 imagens.	55
(i)	<i>Class Matching</i>	55
(j)	<i>Ground Truth</i> : 500 imagens.	55
(k)	Classificação: 500 imagens.	55
(l)	<i>Class Matching</i>	55
4.1	Esquema de redução de dimensionalidade aplicada a coleções rotuladas.	63
4.2	Esquema de redução de dimensionalidade aplicada a coleções não rotuladas.	64
4.3	Amostragem do conjunto de treinamento a ser utilizado pelo PLS, através de um procedimento de agrupamento.	64
4.4	Amostragem do conjunto de treinamento a ser utilizado pelo PLS, através da seleção manual de instâncias em um <i>layout</i>	65
4.5	Exemplos de visualizações RadViz do espaço reduzido obtido para a coleção NEWS, usando a abordagem <i>MultiClassMatrix</i> em 10 fatores (a) e abordagem <i>One Against All</i> em 23 fatores (b).	66
(a)	Redução PLS <i>MultiClassMatrix</i>	66
(b)	Redução PLS <i>One Against All</i>	66
4.6	<i>Layout</i> RadViz alternativo ao mostrado na Figura 4.5a, para a coleção NEWS, produzido pela alteração da ordem dos eixos.	67
4.7	Valores <i>Neighborhood Hit</i> para <i>layouts</i> produzidos pela árvore NJ e por diversas técnicas de projeção, aplicadas no conjunto NEWS com dimensões reduzidas, utilizando um conjunto de treinamento de 863 instâncias e abordagem <i>One Against All</i>	70

4.8	<i>Layouts</i> produzidos por diversas técnicas de projeção e pela árvore NJ, aplicadas no conjunto NEWS com dimensões reduzidas, utilizando um conjunto de treinamento de 863 instâncias.	70
(a)	LSP.	70
(b)	ISOMAP.	70
(c)	RadViz.	70
(d)	Árvore NJ.	70
4.9	Árvore NJ aplicada ao conjunto NEWS, considerando as dimensões originais da coleção.	71
4.10	Árvores NJ construídas do espaço reduzido da coleção NEWS, utilizando PivotMDS, ISOMAP, PCA e PLS.	74
(a)	PivotMDS.	74
(b)	ISOMAP.	74
(c)	PCA.	74
(d)	PLS <i>One Against All</i>	74
4.11	Valores da análise <i>Neighborhood Hit</i> para a coleção NEWS, considerando o espaço original e os espaços reduzidos produzidos pelas técnicas PCA, PivotMDS, ISOMAP e LLE.	74
4.12	Aplicação progressiva de um modelo PLS criado previamente em subconjuntos da coleção ALL.	75
(a)	796 instâncias.	75
(b)	1520 instâncias.	75
(c)	1968 instâncias.	75
(d)	2402 instâncias.	75
(e)	Coleção completa (2814 instâncias).	75
5.1	Exemplo de árvore NJ para a coleção COREL-300, com 44 instâncias selecionadas, representadas por círculos (5.1a) e por imagens (5.1b).	84
(a)	Instâncias selecionadas.	84
(b)	Imagens selecionadas.	84
5.2	Exemplo do resultado da classificação de um subconjunto da coleção COREL com 700 imagens.	86
(a)	<i>Ground Truth</i>	86
(b)	Resultado da Classificação.	86
(c)	<i>Class Matching</i>	86
5.3	Comparação visual entre os resultados da classificação utilizando o modelo LWPR inicial e o atualizado.	91
(a)	Árvore de teste ETHZ-Reduced.	91
(b)	<i>Class Matching</i> para o modelo inicial.	91
(c)	<i>Class Matching</i> para o modelo atualizado.	91
5.4	Árvore NJ da coleção ALL-Reduced01 e resultado da classificação utilizando o modelo LWPR criado na Iteração 1.	93
(a)	Árvore NJ.	93
(b)	<i>Class Matching</i>	93
5.5	Árvore NJ da coleção ALL-Reduced02 e resultado da classificação utilizando o modelo LWPR criado na Iteração 2.	93

(a) Árvore NJ.	93
(b) <i>Class Matching</i>	93
5.6 Comparação entre o <i>ground truth</i> e a árvore <i>Class Matching</i> da classificação da coleção ETHZ-Reduced717 utilizando o modelo construído com apenas 6 classes, mostrando os 4 ramos nos quais todas as instâncias foram classificadas de forma incorreta.	95
(a) <i>Ground truth</i> da coleção ETHZ-Reduced717	95
(b) Árvore <i>Class Matching</i> da classificação.	95
5.7 Árvore NJ da coleção ETHZ-Reduced717, mostrando a distribuição das classes do problema, com destaque para o relacionamento entre instâncias da classe 8 e 25.	96
5.8 Comparação entre árvores <i>Class Matching</i> dos resultados da classificação da coleção ETHZ-Reduced717 utilizando o modelo LWPR atualizado apenas com instâncias das classes não conhecidas (5.8a) e utilizando instâncias das 10 classes (5.8b).	97
(a) Modelo Atualizado com Instâncias de 4 classes.	97
(b) Modelo Atualizado com Instâncias de 10 classes.	97
5.9 Matriz de confusão do resultado da classificação da coleção ETHZ-Reduced717 utilizando o modelo LWPR atualizado apenas com instâncias das classes não conhecidas.	97
5.10 Processo de rotulamento manual de uma coleção de dados representada por uma projeção LSP, utilizando a funcionalidade de rotulamento por seleção.	101
(a) Coleção Não Rotulada.	101
(b) Primeiro Grupo.	101
(c) Segundo Grupo.	101
(d) Rotulamento Completo.	101
5.11 Processo de rotulamento de uma coleção de dados representada por uma árvore NJ, utilizando a funcionalidade de rotulamento por seleção.	101
(a) Coleção Não Rotulada.	101
(b) Primeiro Grupo.	101
(c) Segundo Grupo.	101
(d) Rotulamento Completo.	101
5.12 Processo de rotulamento individual aplicado à seleção de um ramo de imagens em uma árvore NJ.	102
(a) Seleção de um ramo na árvore NJ.	102
(b) Tela de Rotulamento Individual.	102
5.13 Sequência de passos do processo de classificação visual.	104
(a) <i>Ground truth</i> (500 imagens).	104
(b) Conjunto de teste classificado.	104
(c) <i>Class Matching</i>	104
5.14 Tela principal do sistema de classificação visual de imagens.	105

Lista de Tabelas

3.1	Descrição das coleções utilizadas nos experimentos relacionados à árvores de similaridade.	49
3.2	Comparação do número de nós gerado pela árvore NJ original e pela árvore PNJ.	49
3.3	Comparação entre tempos de Geração do <i>Layout</i> (segundos), considerando as abordagens de geração rápida de árvores NJ e a técnica de projeção LSP.	53
4.1	Descrição das coleções utilizadas nos experimentos de redução de dimensionalidade PLS.	66
4.2	Redução de dimensionalidade, utilizando conjunto de treinamento previamente rotulado, para a coleção NEWS.	69
4.3	Redução de Dimensionalidade, utilizando conjunto de treinamento previamente rotulado, para a coleção ETHZ.	69
4.4	Redução de Dimensionalidade, utilizando conjuntos de treinamento originalmente não rotulados, para as coleções NEWS e ETHZ.	72
4.5	Comparação entre os tempos de geração do modelo e os coeficientes de silhueta, considerando a abordagem PLS <i>One Against All</i> (a mais lenta) e outras técnicas de redução de dimensionalidade.	73
4.6	Tempos de carga e aplicação de modelos PLS, para as coleções NEWS, ETHZ e ALL.	75
4.7	Comparação dos resultados da classificação da coleção ETHZ, utilizando modelos PLS e SVM.	76
5.1	Resultado da classificação da coleção COREL-700.	86
5.2	Descrição das coleções utilizadas nos experimentos.	88
5.3	Conjuntos utilizados no experimento.	89
5.4	Comparação dos resultados da classificação utilizando os três tipos de conjuntos de treinamento.	89
5.5	Comparação dos resultados da classificação utilizando o modelo LWPR inicial e o atualizado, para as coleções ETHZ-Reduced e ALL-Reduced.	90
5.6	Média dos resultados da classificação das coleções ETHZ-Reduced e ALL-Reduced, utilizando instâncias escolhidas aleatoriamente para atualizar o modelo LWPR.	91
5.7	Construção de conjuntos de instâncias extraídas da coleção ALL-Reduced para o experimento de classificação iterativa.	92

5.8	Comparação dos resultados da classificação utilizando os modelos LWPR criados no processo iterativo de atualização do modelo, nos conjuntos ALL-Reduced01 e ALL-Reduced02.	94
5.9	Comparação dos resultados da classificação utilizando as três versões do modelo LWPR na coleção ALL-Reduced com 2769 instâncias.	94
5.10	Taxa de Acertos para a classificação do conjunto ETHZ-Reduced717, considerando as classes conhecidas pelo modelo LWPR utilizado.	95
5.11	Distribuição das instâncias das 4 classes da coleção ETHZ-Reduced717 não conhecidas pelo modelo LWPR nas 6 classes conhecidas.	95
5.12	Comparação entre os resultados da classificação da coleção ETHZ-Reduced717 utilizando o modelo LWPR atualizado apenas com instâncias das classes não conhecidas (coluna 2) e utilizando instâncias das 10 classes (coluna 3).	96
5.13	Regras de transformação para instâncias da coleção ETHZ-Reduced, das classes da perspectiva antiga para as da perspectiva nova.	98
5.14	Comparação dos resultados da classificação utilizando o modelo LWPR com perspectiva original e a nova perspectiva, para a coleção ETHZ-Reduced.	99
5.15	Comparação dos resultados da classificação utilizando o modelo LWPR com perspectiva original e a nova perspectiva, para a coleção ALL-Reduced.	99

Lista de Algoritmos

3.1	Neighbor Joining	43
5.1	Aprendizado PLS Incremental. Adaptado de (Vijayakumar & Schaal, 2000)	82
5.2	<i>Locally Weighted Projection Regression.</i> Adaptado de (Vijayakumar et al., 2005) . .	83

Introdução

O aperfeiçoamento dos sistemas computacionais e dispositivos de aquisição de imagens digitais, aliados à redução do custo de aquisição desses equipamentos, provocou a popularização da utilização desse tipo de informação nas mais diversas áreas de conhecimento, como Entretenimento (ex. criação de álbuns pessoais digitais), Medicina (ex. imagens de raio-x, ressonância magnética, mamografia), Biologia (ex. mapeamento de genes), Sensoriamento Geográfico (ex. imagens obtidas via satélite), dentre outros. Como consequência, há o crescimento significativo e contínuo do volume de imagens digitais existentes. A manipulação dessa quantidade de informações exige estratégias de organização eficazes, de forma a possibilitar sua recuperação, exibição e exploração, em um processo chamado **mineração de dados**. Nesse processo, ocorre a extração automática de informações potencialmente úteis em repositórios de dados, e a transformação desses dados processados em conhecimento (Chen et al., 1996).

1.1 Motivação

Uma atividade importante relacionada à mineração de conjuntos de imagens é a **classificação** ou **categorização**, que consiste em segregar as imagens de uma coleção, colocando-as em grupos previamente definidos, representando assim uma maneira de extrair informação em imagens para reconhecer padrões e objetos homogêneos (Lu & Ip, 2010). Segundo Heidemann (2005), a classificação manual de coleções de imagens, além de exigir considerável esforço por parte dos usuários, mostra-se impossível para coleções em constante crescimento. Além disso, seres humanos

possuem dificuldade em aplicar de forma consistente um sistema único de critérios de classificação, especialmente quando vários métodos parecem ser adequados. Dessa forma, um processo de classificação automático torna-se desejável, pois permite um rápido processamento do volume de dados.

A classificação se mostra importante em diversos cenários. Diversas informações contidas nas imagens de sensoriamento remoto podem ser extraídas através da classificação dessas imagens (Queiroz et al., 2004; Pasolli & Melgani, 2010). Imagens de mamografia são consideradas um dos meios mais confiáveis para a detecção precoce de câncer de mama, mas devido ao volume de imagens a serem interpretadas pelos especialistas, a taxa de precisão na detecção tende a diminuir. Diversos exemplos (Antonie et al., 2001; Cheng et al., 2010; Chen et al., 2011) mostram que a classificação apoiada por computador pode representar um grande auxílio na tomada de decisão nesses casos. O diagnóstico utilizando imagens de Ressonância Magnética também é consideravelmente facilitado utilizando técnicas de classificação de imagens (Zhang et al., 2011). Uma das principais aplicações da classificação de imagens é nos sistemas CBIR (*Content-Based Image Retrieval*). De acordo com Vailaya et al. (2001), esses sistemas podem se beneficiar da classificação das imagens através da filtragem de classes irrelevantes no processo de busca. Organizações e empresas possuem grandes coleções de vídeos e imagens, e a organização desses repositórios em categorias e sua indexação eficaz são imprescindíveis para exibição e recuperação em tempo real, possibilitando uma interação efetiva com o usuário. Finalmente, diversos sistemas de Visão Computacional (Wang et al., 2009; Liu et al., 2009, 2010; Paci et al., 2011; Schwartz et al., 2012) também aplicam técnicas de classificação de imagens com o intuito de reconhecer objetos, pessoas ou cenas e auxiliar na tomada de decisão de diversos sistemas relacionados.

É possível encontrar na literatura diversas técnicas de classificação de imagens. Algumas dessas técnicas são baseadas em um processo de treino e teste, tais como redes neurais artificiais (Giacinto & Roli, 2001; Faria et al., 2003; Ribeiro & Centeno, 2001; Antonie et al., 2001; Zhang et al., 2011) e *Support Vector Machines* (Tarabalka et al., 2010; Pasolli & Melgani, 2010; Cheng et al., 2010). Outras, chamadas de não paramétricas, baseiam-se no número de vizinhos mais próximos, tais como *Naive-Bayes nearest-neighbor* (Boiman et al., 2008) e *Local Naive-Bayes nearest-neighbor* (McCann & Lowe, 2011). Finalmente, existem aquelas baseadas em *active learning* (Joshi et al., 2012), nas quais os classificadores são treinados interativamente a partir de anotações feitas pelo usuário em amostras informativas. No entanto, o processo de classificação de imagens tem como característica principal sua natureza individual, dependendo fortemente de fatores como representação das imagens, algoritmo utilizado, e do conjunto de treinamento (Waske et al., 2010). Assim, nenhuma abordagem é ótima para todos os tipos de imagens. Além disso, mesmo com avanços observados em algoritmos de seleção e transformação de características usadas na classificação e recuperação de imagens (Chang et al., 2009; Guan et al., 2010; Ciocca et al., 2011), existe dificuldade em relacionar essas características (de baixo nível, tais como cor, textura,

estrutura, entre outras) com semânticas de alto nível (Huang, 2012), no sentido que elas muitas vezes falham em descrever as concepções mentais dos usuários sobre as imagens. De acordo com Ciocca et al. (2011), o problema existe porque o significado de uma imagem não é uma função de seu conteúdo, mas depende da sua descrição no ambiente no qual ela foi produzida, do contexto cultural no qual é examinada, entre outros fatores.

Diversos autores (Keim et al., 2005; Rüger, 2006; Deselaers et al., 2008; Nezamabadi-pour & Kabir, 2009; Joshi et al., 2012) propõem então que é importante, para o processo de classificação, que o usuário seja inserido no processo de exploração, combinando a flexibilidade, criatividade e conhecimento do ser humano com a capacidade computacional atual. Entretanto, essa combinação não será possível caso o usuário tenha acesso apenas a uma lista de todas as imagens da coleção, para a análise uma a uma. É necessário dispor essas coleções de uma maneira amigável e efetiva, de forma que seja possível enxergar suas características, as relações de similaridade entre as imagens que as compõem, além de padrões de comportamento existentes, permitindo assim que haja um processo natural de categorização. De acordo com Heidemann (2005), o agrupamento de imagens semelhantes pode melhorar a acessibilidade visual e explorar melhor as capacidades humanas, facilitando a manipulação dos itens da coleção.

Com esse intuito, as técnicas de visualização de informação representam uma ferramenta importante para garantir uma experiência de navegação e exploração satisfatória por parte do usuário (Manovich, 2011) e diversas pesquisas descrevem esse potencial (Xu et al., 2011; Schaefer, 2011; Hochman & Schwartz, 2012). A ideia básica dessa exploração visual é apresentar as imagens de uma maneira que permita ao usuário inferir conhecimento sobre a coleção. Além disso, visualizações permitem que o usuário tenha uma visão geral de toda a coleção de imagens, de onde ele pode partir em um processo de investigação com o objetivo de recuperar uma ou várias imagens desejadas, ou detectar tendências e características particulares que seriam extremamente difíceis de serem detectadas caso ele analisasse imagem por imagem. De acordo com Paulovich (2008), a associação entre os algoritmos de mineração e as técnicas de visualização pode ocorrer de três maneiras: na primeira, a visualização pode auxiliar no entendimento da estrutura original dos dados, para aplicação do algoritmos. Na segunda, elas são utilizadas para auxiliar na interpretação dos resultados obtidos. Finalmente, um conjunto de formas de interação entre usuário e computador permitem que os parâmetros dos algoritmos sejam modificados pelo usuário, com base nos resultados apresentados, de forma a melhorar o processo.

A visualização e mineração de dados em coleções de forma eficaz muitas vezes depende da representação das instâncias dessa coleção. Diversas pesquisas (Gehler & Nowozin, 2009; Schwartz, 2010) apontam para uma combinação de representações que abranja o máximo de informações a respeito dessas instâncias. No entanto, a combinação de descritores resulta na construção de representações com alta dimensionalidade, e o processo de visualização e classificação pode ser novamente prejudicado pelo alto custo computacional necessário para manipular essas representa-

ções, bem como pela presença de informações redundantes que produzam *layouts* de difícil navegação e exploração por parte do usuário. Procedimentos de redução de dimensionalidade podem ser utilizados nesses casos, selecionando ou combinando atributos de forma a concentrar apenas as informações essenciais que captem a estrutura da coleção, ou melhorar o espaço original no sentido de realçar tendências características dos dados de acordo com determinada perspectiva. Para o processo de classificação, é interessante que os espaços reduzidos produzidos por essas técnicas possam realçar as diferenças entre classes, gerando *layouts* que permitam a extração de informações úteis na convergência de resultados.

1.2 Objetivos

Com o intuito de prover uma solução para as necessidades citadas, o objetivo deste projeto de doutorado é desenvolver uma metodologia baseada na utilização de técnicas visuais que apóiem o processo de classificação de imagens. Especificamente, tais técnicas visuais devem permitir a colaboração entre usuário e sistema de classificação automática, em um processo iterativo que apóie as várias etapas do processo de classificação, de forma a produzir os resultados esperados para uma variedade de domínios de aplicação. Espera-se que o usuário possa, utilizando a metodologia desenvolvida, enxergar a formação de grupos que representem classes em determinada coleção de imagens, bem como interferir no processo de classificação automática, refinando os resultados de acordo com as suas necessidades. Dessa forma, o resultado deste trabalho poderá melhorar o processo como um todo, possibilitar a seleção e combinação de conjuntos de características visuais de imagens que melhor representem-nas, avaliar algoritmos de classificação automática, e inclusive possibilitar a combinação desses algoritmos para produzir grupos e categorias de imagem que melhor representem a situação real da coleção.

1.3 Contribuições

De acordo com os objetivos expostos, os resultados e contribuições alcançados neste trabalho foram:

- **Procedimento de promoção de nós em árvores de similaridade *Neighbor Joining*:** foi desenvolvido um procedimento, a ser executado após a construção de uma árvore de similaridade *Neighbor Joining* (NJ), para diminuir o número de pontos do *layout*, possibilitando um melhor aproveitamento do espaço de visualização (Paiva et al., 2011). A nova árvore gerada possui em média 51% menos nós, garantindo menor confusão visual, sem perda significativa na precisão ou aumento expressivo no tempo de geração da estrutura da árvore e possibilitando uma melhor compreensão por parte do usuário sobre a coleção.

- **Adaptação e implementação de versões com menor custo computacional do algoritmo *Neighbor Joining*:** dois algoritmos, **Rapid NJ** e **Fast NJ**, que representam modificações do algoritmo original de geração de árvores de similaridade *Neighbor Joining* foram investigados e implementados (Paiva et al., 2011). O algoritmo **Rapid NJ** utiliza estruturas de dados especializadas para gerar a mesma árvore de similaridade gerada pelo algoritmo original, em 28% menos tempo. Já o algoritmo **Fast NJ** utiliza heurísticas que geram uma aproximação da árvore gerada pelo algoritmo original, em 96% menos tempo. Tais algoritmos, juntamente com o processo de promoção de nós, possibilitam a aplicação de árvores de similaridade em coleções de dados maiores de forma mais fácil.
- **Adaptação de medidas de avaliação de técnicas de visualização para árvores de similaridade:** as medidas de avaliação de projeções multidimensionais usualmente utilizadas, tais como ***Neighborhood Hit***, ***Neighborhood Preservation*** e **Coeficiente de Silhueta**, dentre outras, foram adaptadas para permitir a avaliação das árvores de similaridade *Neighbor Joining* (Paiva et al., 2011). Nessas árvores, a distância Euclidiana no *layout* é definida pelo algoritmo de desenho da árvore, e não possui nenhuma relação com o valor de similaridade entre um ponto e os demais, não podendo ser utilizada no cálculo dessas medidas. Assim, a distância entre dois pontos a e b na árvore NJ foi definida como a soma dos pesos das arestas que formam o menor caminho conectando a a b . Essa adaptação permitiu a comparação entre árvores NJ e outras técnicas de visualização na construção de *layouts* representando diversas coleções, na avaliação do procedimento de promoção de nós e dos algoritmos de melhoria na construção da árvore, bem como na avaliação de diversas tarefas de análise visual de dados.
- **Metodologia de análise visual de coleções de dados utilizando *Partial Least Squares* para redução de dimensionalidade:** uma metodologia de redução de dimensionalidade semi-supervisionada, utilizando a técnica ***Partial Least Squares*** (PLS) foi desenvolvida (Paiva et al., 2012), que permite que o usuário utilize seu conhecimento no processo, através da criação e manipulação de um conjunto de amostras utilizado para reduzir as dimensões de uma coleção. O processo desenvolvido melhora significativamente a qualidade do espaço de características em termos de discriminabilidade entre classes, facilitando a visualização de padrões presentes nos dados. Apesar de a técnica PLS ser caracterizada como supervisãoada, foi desenvolvida uma abordagem para lidar com cenários para os quais não se tem nenhuma informação sobre classes, baseada no agrupamento da coleção e subsequente utilização dos rótulos dos grupos como rótulos das instâncias.
- **Conjunto de ferramentas de interação para classificação visual de imagens:** com o objetivo de possibilitar a interação entre usuário e sistema de classificação, no processo de classificação visual, foi desenvolvido um conjunto de ferramentas de rotulamento que possibilitam ao usuário selecionar instâncias em uma coleção para criar conjuntos de treinamento

para a classificação, ajustar um conjunto previamente construído. Além disso, é possível alterar a perspectiva da classificação, de forma a se ajustar a diferentes necessidades. Finalmente, uma análise detalhada dos resultados da classificação, composta pela associação de dados estatísticos sobre o processo e um *layout*, em uma ferramenta denominada **Class Matching**, permite que o usuário comprehenda as razões pelas quais a classificação foi feita de determinada maneira, possibilitando a tomada de decisões de forma direcionada e efetiva.

- **Metodologia de Classificação Visual incremental baseada em LWPR:** uma metodologia de classificação visual incremental foi desenvolvida, utilizando o algoritmo **Locally Weighted Projection Regression** (LWPR), com o objetivo de inserir o usuário no processo de classificação de conjuntos de dados. Essa inserção é feita através da criação e aplicação de modelos LWPR, criando um esquema iterativo de classificação que possibilita uma rápida convergência de resultados. Baseada em um treinamento incremental, a técnica LWPR permite que os modelos sejam atualizados, o que possibilita a correção de eventuais erros no processo, bem como a classificação de coleções de dados que evoluem ao longo do tempo. Da mesma forma, é possível lidar com o aparecimento de novas classes em um cenário, e com alterações na perspectiva de classificação. Um sistema de classificação visual que incorpora essa metodologia associada às ferramentas anteriormente citadas foi implementado, permitindo a realização de todo o processo de classificação visual.
- **Softwares e API's que implementam os algoritmos e metodologias desenvolvidas:** foram desenvolvidas três API's JAVA, a primeira para a geração de árvores *Neighbor Joining*, com possibilidade de utilizar os algoritmos de melhoria na construção da estrutura da árvore (**Rapid NJ** e **Fast NJ**), e o procedimento de promoção de nós, a segunda para o processo de redução de dimensionalidade PLS, utilizando a metodologia semi-supervisionada criada, e a terceira para realizar o processo de criação, atualização e aplicação de modelos **Locally Weighted Projection Regression** (LWPR). Todas as metodologias desenvolvidas utilizando essas API's foram inseridas no sistema VisPipeline¹, desenvolvido pelo VICG². Finalmente, um software de classificação visual incremental foi desenvolvido, que cria um ambiente que contempla todo o processo de classificação de coleções de imagem, apoiado por técnicas de visualização e um conjunto de ferramentas interativas, oferecendo suporte para criação, atualização e aplicação de modelos de classificação LWPR, criação e aplicação de modelos de classificação **Support Vector Machines** (SVM) e **Partial Least Squares** (PLS), e a utilização da técnica **K-Nearest Neighbors** (KNN) para classificação.

¹Disponível em <http://vicg.icmc.usp.br/infovis2/Tools>

²VICG: grupo de Visualização, Imagens e Computação Gráfica (VICG) do ICMC/USP

1.4 Organização do Texto

De forma a apresentar todo o trabalho desenvolvido durante o trabalho de doutorado, o restante deste documento está organizado da seguinte maneira:

- **Capítulo 2:** apresenta uma revisão bibliográfica a respeito dos temas relacionados a este projeto de doutorado, incluindo os conceitos relacionados à representação de imagens, às técnicas de redução de dimensionalidade e de visualização existentes, além da aplicação de tais técnicas em visualização de coleções de imagens. Uma análise comparativa das abordagens é realizada, de forma a destacar suas limitações, que serviram de motivação para o desenvolvimento do trabalho.
- **Capítulo 3:** apresenta um estudo detalhado sobre árvores de similaridade utilizando a técnica *Neighbor Joining*, detalhando seu funcionamento, e destacando suas vantagens e limitações. Além disso, são apresentadas e avaliadas propostas para resolver os problemas relacionados à ocupação do espaço de visualização e custo computacional. Finalmente, o capítulo apresenta uma aplicação dessas árvores para a classificação visual de coleções de imagens, utilizando as abordagens de melhoria propostas.
- **Capítulo 4:** uma abordagem de redução de dimensionalidade e classificação utilizando a técnica *Partial Least Squares* (PLS) é apresentada. As características da técnica são exploradas, com destaque para sua capacidade de destacar a discriminação entre as classes de uma coleção. Duas metodologias de utilização do PLS são apresentadas, incluindo uma forma de lidar com coleções não rotuladas, em uma técnica de natureza supervisionada. Por fim, os resultados da metodologia proposta são apresentados e discutidos, em aplicações para redução de dimensionalidade e classificação de coleções de dados.
- **Capítulo 5:** apresenta uma metodologia de classificação visual incremental baseada no algoritmo *Locally Weighted Projection Regression* (LWPR). A ideia é promover a interação do usuário na classificação através da criação e aplicação de modelos LWPR, em um processo iterativo de classificação com rápida convergência de resultados. Através de um treinamento incremental, é possível mapear coleções em evolução, e lidar com o aparecimento de novas classes. Um sistema de classificação visual baseado nessa metodologia é apresentado, juntamente com os resultados de uma série de estudos de caso representando diversos cenários relacionados.
- **Capítulo 6:** apresenta as conclusões a respeito de todos os trabalhos desenvolvidos neste projeto de doutorado, detalhando as contribuições para o processo de classificação visual de imagens, bem como os trabalhos futuros.

Trabalhos Relacionados

2.1 Considerações Iniciais

Este capítulo apresenta diversas pesquisas cujos tópicos relacionam-se com o tema desta tese. Inicialmente, a Seção 2.2 apresenta conceitos e trabalhos com o objetivo de encontrar formas de representação das instâncias de uma coleção que produza resultados satisfatórios em técnicas de classificação e visualização, além de um estudo sobre as medidas de similaridade utilizadas para descrever a distribuição dessas instâncias nos *layouts* gerados. A Seção 2.3 mostra um estudo de caracterização das técnicas de visualização existentes, e pesquisas com o intuito de produzir mapeamentos que reflitam detalhes importantes de uma coleção e permitam uma exploração satisfatória por parte do usuário. A Seção 2.4 faz um estudo das técnicas de redução de dimensionalidade, e seu papel na construção de espaços reduzidos que melhorem o espaço original sob diversas perspectivas, e finalmente, a Seção 2.5 apresenta conceitos relacionados à classificação visual de dados.

Este capítulo concentra ideias apresentadas em artigos cujo conteúdo completo pode ser encontrado nos Apêndices B, C e D.

2.2 Representação de Coleções de Dados

A eficácia no processo de análise de coleções de dados, seja para classificação, busca ou qualquer outra atividade, depende de como essas imagens são representadas. O problema oferece

desafios, porque a aparência dos objetos contidos nas imagens pode sofrer variação relacionada à posição, iluminação ou formato, que podem interferir na descrição dos conteúdos das mesmas. A representação deve ser flexível o bastante para cobrir um intervalo de diferentes classes (representando diferentes tipos de conteúdo), cada uma com variações entre seus objetos, e ao mesmo tempo possuir boa capacidade de discriminação entre essas classes (Nowak et al., 2006).

O conteúdo de uma imagem pode ser representado por diversas características visuais. Comumente, as características relevantes utilizadas no processo, chamadas de características de baixo nível, são: cor, textura e forma. Essas características podem ser extraídas da imagem inteira, ou de regiões dessa imagem (Pedrini & Schwartz, 2007).

A informação de cor é uma qualidade básica de um conteúdo visual, e é utilizada de diversas maneiras para descrever imagens, pois relaciona-se fortemente com os objetos que as compõem (Guo et al., 2001; Laencina Verdaguer, 2009). Além disso, são invariáveis ou apresentam variações pequenas quando as imagens sofrem transformações como rotações ou escalas, e apresentam simplicidade de implementação, baixo custo computacional (comparações entre imagens com custo de $O(n)$, com n igual ao número de imagens), e baixa exigência para armazenamento. A utilização de características relacionadas à textura reside no fato de que elas contemplam mais informações espaciais, além de indicarem informações importantes de diversas imagens do mundo real, representando bem seu significado. Já as características relacionadas à forma tentam detectar padrões que sejam representativos de classes de objetos. Alguns autores (Liu et al., 2007) adicionam a análise da localização espacial na representação da imagem, com o intuito de diferenciar elementos distintos, mas com características de cor e textura semelhantes, tais como o céu e o mar, ou então um pôr do sol em um deserto.

Uma outra abordagem de representação que apresenta resultados interessantes é a chamada *bag-of-features* (BoF). A ideia básica dessa abordagem é descrever cada imagem de uma coleção como um conjunto não-ordenado de atributos, representando características das imagens (Jiang et al., 2007). Essa abordagem representa uma analogia à representação de documentos textuais utilizando *bag-of-words*, tanto em termos de formato quanto em termos de significado, o que faz com que as técnicas de classificação de textos sejam facilmente utilizadas no processo de classificação de imagens. De acordo com Jégou et al. (2010), a utilização dessa abordagem se beneficia da grande quantidade de descritores locais existentes, além de permitir a comparação de imagens utilizando medidas de distância padrão, permitindo sua aplicação em algoritmos de classificação de maneira simples.

Os atributos da representação BoF podem ser baseados nas características citadas anteriormente, e extraídos de toda a imagem, ou de regiões. De acordo com Yang et al. (2007), existe uma tendência em utilizar pontos-chave ou pontos de interesse locais para a classificação de imagens. Esses pontos-chave representam trechos de uma imagem contendo informações importantes sobre ela, automaticamente detectados e representados numericamente por um descritor. Os pontos-

chave são então agrupados segundo a similaridade de seus descritores, e cada grupo é tratado como uma “palavra visual” de um “vocabulário visual”, representando um padrão específico compartilhado por todos os seus pontos-chave. Esse mapeamento entre atributos e palavras visuais cria um vetor de características para cada imagem, de acordo com a presença, ou total de ocorrências dessas palavras visuais.

Tão importante quanto definir uma representação para as imagens de uma coleção é definir medidas de similaridade e dissimilaridade entre elas. Essas medidas determinarão o tipo de relação que será denotada pela distribuição dos elementos nos *layouts* gerados. Uma maneira de determinar a dissimilaridade entre duas imagens é através do cálculo da **distância** entre elas (Tan et al., 2005). Cada imagem pode ser entendida como um ponto em um espaço multidimensional, e a distância entre esses dois pontos representará a dissimilaridade entre as duas imagens. A similaridade entre elas também pode ser determinada de diversas maneiras, sendo obtida por exemplo através do complemento do valor de dissimilaridade.

Dentre as medidas de dissimilaridade baseadas em distâncias, destacam-se a distância Euclidiana, distância *city-block*, distância quadrática e a distância Mahalanobis, bem como a distância angular.

A distância Euclidiana e a distância *city-block* representam formas específicas da medida de **distância de Minkowski** (Eq. 2.1), com parâmetro r igual a 2 e 1, respectivamente, como mostram as Eq. 2.2 e 2.3:

$$d(A, B) = \left(\sum_{k=1}^n |a_k - b_k|^r \right)^{1/r}, \quad (2.1)$$

$$d(A, B) = \sqrt{\left(\sum_{k=1}^n |a_k - b_k|^2 \right)}, \quad (2.2)$$

$$d(A, B) = \left(\sum_{k=1}^n |a_k - b_k| \right), \quad (2.3)$$

onde $A = \{a_1, a_2, a_3, \dots, a_n\}$ e $B = \{b_1, b_2, b_3, \dots, b_n\}$ são os vetores de características que representam as duas imagens comparadas.

A distância quadrática (Zhang & Lu, 2003), cujo cálculo é apresentado na Eq. 2.4, leva em conta não só a correspondência entre determinado atributo em duas imagens, mas também a relação entre os atributos de uma mesma imagem.

$$d(A, B) = \sqrt{(A - B)^t A (A - B)}, \quad (2.4)$$

onde $A = [a_{ij}]$ representa uma matriz $n \times n$, e a_{ij} é o coeficiente de similaridade entre as dimensões i e j .

Dessa forma, não se consideram os atributos de uma imagem como informações independentes, mas relacionadas. Uma matriz de dissimilaridade entre os atributos é utilizada no cálculo do valor da distância. Um caso especial da distância quadrática é a distância de Mahalanobis, na qual a informação de relação entre os atributos é dada por uma matriz de covariância obtida através de um conjunto de treinamento.

Já a distância angular, ou distância do cosseno (Eq. 2.5), calcula a diferença de direção entre os vetores que representam as imagens, sem levar em conta o comprimento desses vetores. A distância é dada pelo ângulo Θ entre os dois vetores de características.

$$d(A, B) = 1 - \cos \Theta = 1 - \frac{A \cdot B}{|A| \cdot |B|} \quad (2.5)$$

Estudos adicionais a respeito de medidas de dissimilaridade e similaridade podem ser encontradas em (Tan et al., 2005; Lesot et al., 2009).

2.3 Visualização de Informação

A Visualização de Informação estuda a utilização de representações visuais interativas, apoiadas por computador, de dados abstratos e não-estruturados para ampliar a cognição (Card et al., 1999; Keim et al., 2002) e melhorar a interface usuário-computador. Essas pesquisas concentram-se nas características básicas assimiladas pelo ser humano: cor, tamanho, forma, proximidade, e movimento, possibilitando que ele enxergue relações entre os dados apresentados, e permitindo a percepção de tendências e comportamentos a respeito da coleção.

De acordo com Card et al. (1999), o processo de visualização pode ser visto como um mapeamento ajustável, de dados puros para uma forma visual, percebida pelo ser humano. Esse processo é ilustrado na Figura 2.1. Na figura, é possível perceber que diversas transformações são realizadas nos dados, e que o usuário pode, a qualquer momento, modificar o *layout* de acordo com suas necessidades.

Segundo Keim (2001), a representação visual de coleções de dados comunica claramente ao usuário o conteúdo informacional desses dados, reduzindo o trabalho cognitivo necessário para realizar diversas tarefas. Além disso, o usuário se torna um agente ativo no processo de mineração das informações, pois consegue, além de visualizar as relações entre os dados, interagir com o *layout*, tendo uma visão geral, ou concentrando-se em fenômenos particulares. Isso resulta em um processo exploratório mais rápido e com resultados melhores, em especial quando procedimentos automáticos falham. A técnicas de visualização podem ser categorizadas em dois grandes

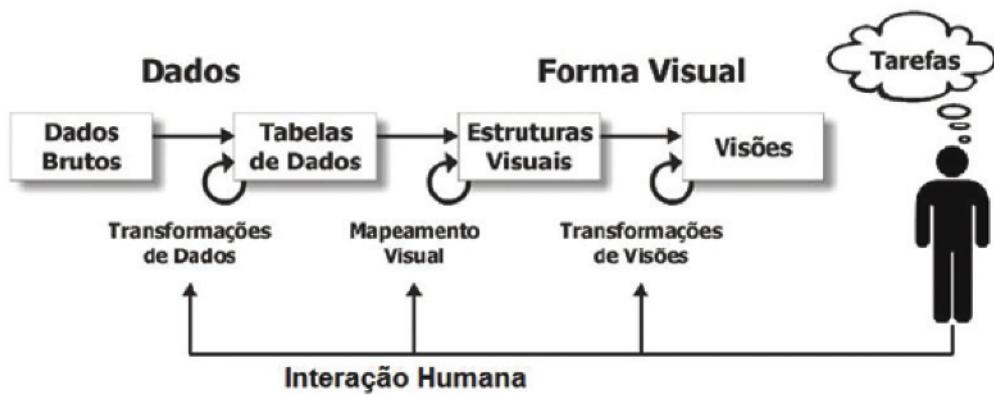


Figura 2.1: Processo de Visualização (Adaptado de (Card et al., 1999)).

grupos, aquelas baseadas em atributos e as baseadas em posicionamento de pontos no espaço de visualização. Essas técnicas serão apresentadas a seguir.

2.3.1 Técnicas de Visualização baseadas em Atributos

Keim & Kriegel (1996) descrevem 4 grupos de técnicas baseadas em atributos: utilizando *pixels*, transformações geométricas, ícones e hierarquias.

Nas técnicas de visualização baseadas em *pixels*, cada valor de atributo de cada instância da coleção é mapeado para um pixel colorido, e os valores de um atributo são apresentados em áreas separadas. O objetivo dessas técnicas é então encontrar uma maneira de arranjar os pixels de forma a facilitar a extração de conhecimento.

As técnicas baseadas em Transformações Geométricas procuram apresentar projeções dos dados de forma a encontrar correlações entre seus atributos. As técnicas mais comuns são as **Coordenadas Paralelas** (Inselberg & Dimsdale, 1990), **Matrizes de Dispersão** (Cleveland, 1993) e Projeções **RadViz** (Fayyad et al., 2002).

Nas técnicas baseadas em ícones, cada instância da coleção é mapeada para um ícone, e usualmente, os valores de dois dos atributos são mapeados em posições desse ícone no plano. O restante dos atributos são mapeados em propriedades dos ícones, tais como cor, formato, orientação, tamanho, entre outros. Dependendo da densidade dos dados, e das dimensões de exibição do *layout*, o resultado será a presença de padrões que variam de acordo com as características da coleção, que poderão ser facilmente percebidos pelo usuário. **Faces de Chernoff**, **Star Glyphs** e **Stick Figures** (Lee et al., 2003) representam exemplos deste tipo de técnica.

As técnicas hierárquicas subdividem o espaço multi-dimensional de forma a criar uma hierarquia de atributos. É o caso da visualização por **Dimensional Stacking** (LeBlanc et al., 1990), que cria diversos níveis de coordenadas empilhados uns dentro dos outros, ou da **TreeMap** (John-

son, 1992), que partitiona a área de visualização em regiões, cujas dimensões são partitionadas alternadamente, dependendo dos valores dos atributos.

Um estudo detalhado a respeito de diversas técnicas de visualização baseadas em atributos, considerando inclusive abordagens de agregação hierárquica, pode ser encontrado em (Elmqvist & Fekete, 2010).

As técnicas de visualização baseadas em atributos são úteis para a identificação de correlações entre os atributos. No entanto, apresentam limitações relacionadas à capacidade de visualização e interpretação, quando a coleção de dados possui muitas instâncias ou quando os dados possuem alta dimensionalidade. Nas Coordenadas Paralelas, por exemplo, como cada linha horizontal representa uma instância, muitas instâncias causarão uma excessiva sobreposição de linhas, que impedirá a compreensão por parte do usuário. O mesmo ocorre com os atributos, representados por barras verticais. Dados com alta dimensionalidade produzirão muitas barras, e portanto também poderão causar confusão visual (*cluttering*). Tais problemas podem ser notados na Figura 2.2. Finalmente, a ordenação de atributos no *layout*, representada pela organização das barras verticais, influencia consideravelmente no resultado da visualização, e torna-se um desafio encontrar uma configuração ótima para representar a coleção de dados.

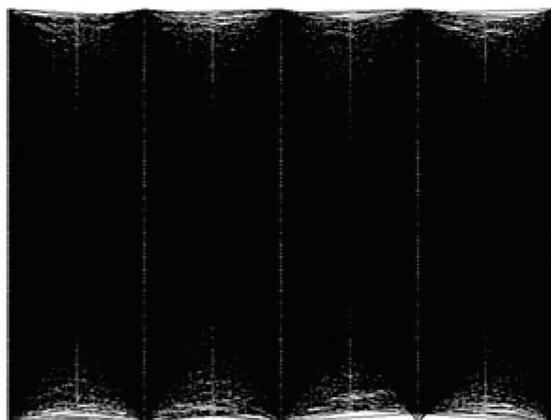


Figura 2.2: Coordenadas Paralelas de uma coleção de dados com 7500 instâncias, cada uma com 5 atributos, ilustrando a confusão visual (*cluttering*) causado pelo número de instâncias (Adaptado de (Artero et al., 2004)).

As matrizes de dispersão também apresentam problemas quando os dados possuem alta dimensionalidade. Nessa técnica, para instâncias com n atributos, é construída uma matriz $n \times n$, com a relação entre os atributos organizados de dois em dois. Dessa forma, quanto mais atributos, maior a matriz resultante, que se torna extensa demais para valores de n altos. Já as técnicas orientadas a pixel podem gerar representações de difícil compreensão para o usuário, dependendo da coleção, tornando a tarefa de extração de conhecimento mais complexa. Além disso, em muitos casos o usuário tem que relacionar partes da área de visualização, muitas vezes distantes umas das outras, para enxergar a correlação entre os atributos. A Figura 2.3 mostra um exemplo de visualização

baseada em pixel utilizando duas formas de arranjo, **Peano-Hilbert** e **Morton**, na qual um mapeamento de cores é aplicado aos valores dos atributos, sendo que cores claras são associadas a valores altos, e cores escuras associadas a valores baixos. O *layout*, apesar de permitir a inferência de conhecimento, pode não ser intuitivo para usuários que não estão acostumados com a técnica, exigindo assim aprendizado e prática.

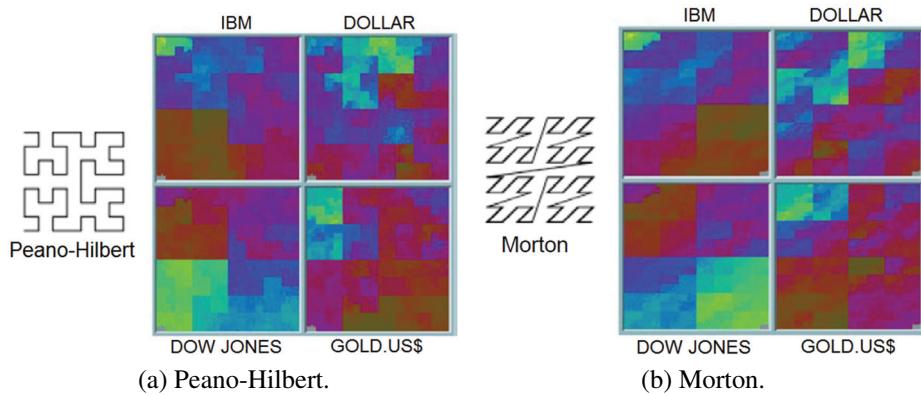


Figura 2.3: Visualizações baseadas em pixel, representando uma coleção com 16350 instâncias contendo 9 informações da bolsa de valores, coletadas entre janeiro de 1987 até março de 1993, usando arranjos Peano-Hilbert (2.3a) e Morton (2.3b) (Adaptado de (Keim & Ankerst, 2001)).

2.3.2 Técnicas de Visualização baseadas em Posicionamento de Pontos

As estratégias de posicionamento de pontos mapeiam as instâncias de uma coleção em pontos individuais no espaço de visualização. A ideia principal é preservar nesse espaço os relacionamentos relevantes observados no espaço original, utilizando medidas de similaridade que posicionem próximas instâncias similares, e distantes instâncias não similares.

Diversas técnicas de posicionamento de pontos podem ser encontradas na literatura. Worring et al. (2007) categorizam os *layouts* de visualização em dois tipos: os baseados em preservação estrutural, no qual métodos de projeção são utilizados para mapear as dimensões que representam as características dos dados, preservando a maior quantidade de relações presentes no espaço original de dimensões, e os baseados em conexão de grafos, no qual as instâncias são visualizadas por um grafo com conexões traçadas de acordo com as relações entre elas. A seguir são apresentadas as características de cada um desses tipos.

Visualização baseada em Projeção

Projeções, apesar de apresentarem considerável (e inevitável) perda de informação, especialmente em relação às configurações complexas do espaço original de características, representam uma das ferramentas mais comuns de exploração da estrutura de espaços multidimensionais.

Diversas técnicas de projeção multidimensional foram propostas para auxiliar a exploração e compreensão de informação, a maioria delas baseada em técnicas de redução de dimensões. PCA (*Principal Component Analysis*) (Jolliffe, 2002) representa uma técnica de projeção linear que emprega combinações lineares dos atributos (dimensões) com alto grau de covariância, produzindo atributos com menor dependência chamados **componentes principais**. A maior desvantagem dessa técnica é a baixa qualidade dos *layouts* gerados, e seu alto custo computacional, $O(m^2n)$, com m igual ao número de dimensões, e n igual ao número de instâncias.

Já MDS (*Multidimensional Scaling*) (Cox & Cox, 2000) comprehende uma classe de técnicas que podem ser utilizadas para realizar projeções. Em sua abordagem mais simples, originalmente definida como uma heurística para desenho de grafos, tem-se a chamada *Force Directed Placement* (FDP) (Eades, 1984). O modelo FDP é baseado em um sistema de molas, no qual as instâncias multidimensionais são modeladas como objetos conectados por molas, e forças de atração e repulsão entre elas são proporcionais as distâncias entre essas instâncias. A projeção final é obtida quando o sistema de molas atinge o seu estado de equilíbrio. Essa técnica geralmente apresenta alto grau de precisão para coleções com instâncias que possuem relacionamentos não-lineares, mas possui um alto custo computacional ($O(n^3)$), com n igual ao número de instâncias.

A técnica FASTMAP (Faloutsos & Lin, 1995) realiza a projeção de instâncias em um espaço dimensional reduzido buscando preservar as relações de distância existentes no espaço original. Duas instâncias são inicialmente escolhidas (quanto mais distantes essas instâncias estiverem, melhor). Elas definirão uma reta no espaço original (m -dimensional), e um hiperplano de dimensionalidade $m - 1$ perpendicular à reta. As instâncias restantes são então projetadas nesse hiperplano. Repete-se o processo até que o número de dimensões seja o desejado para a projeção. A técnica reduz a complexidade computacional para $O(n^2)$, com n igual ao número de instâncias.

A técnica RadViz (*Radial Coordinate Visualization*) (Hoffman et al., 1999) posiciona instâncias m -dimensionais em um plano de visualização que utiliza as leis físicas de Hooke. A posição de cada ponto é definida de acordo com a posição de k âncoras, normalmente arranjadas ao redor de uma circunferência. Cada âncora é conectada a uma mola virtual com grau de elasticidade variável, e essas molas são todas conectadas pela outra extremidade. O grau de elasticidade da i -ésima mola é proporcional ao valor do i -ésimo atributo de uma instância, e essa instância é então mapeada para um ponto no qual todas as forças exercidas por todas as molas virtuais atingem o equilíbrio, de acordo com a Equação 2.6.

$$\sum_{j=1}^k (\vec{A}_j - \vec{p}) x_j = 0 \quad (2.6)$$

Considerando uma instância $[x_1, \dots, x_m]$, e o conjunto de âncoras $[A_1, \dots, A_k]$, a Figura 2.4 mostra um exemplo de mapeamento de uma instância para o ponto p .

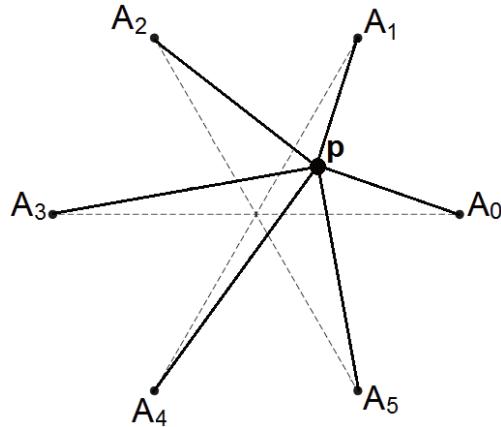


Figura 2.4: Mapeamento de uma instância em um ponto p , utilizando a técnica RadViz (Adaptado de (Novakova & Stepankova, 2009)).

A posição do ponto $p = [p_1, p_2]$ será determinada pela Equação 2.7.

$$\vec{p} = \frac{\sum_{j=1}^k \vec{A}_j y_j}{\sum_{j=1}^k y_j} \quad (2.7)$$

A principal vantagem da técnica RadViz é que não é necessária nenhuma técnica de projeção para posicionar os pontos no plano de visualização, e seu custo computacional é baixo, $O(mn)$, com n igual ao número de instâncias, e m igual ao número de dimensões. Além disso, as forças exercidas por cada mola, proporcionais aos valores dos atributos das instâncias, produzirão um *layout* que apresentará exatamente as influências das dimensões do espaço reduzido nas instâncias da coleção. Entretanto, de acordo com Novakova & Stepankova (2009), a técnica RadViz apresenta algumas características que podem representar limitações no resultado do processo. Cvek et al. (2011) citam, como uma de suas principais desvantagens, a sobreposição de pontos que ocorre quando os valores dos atributos das instâncias correspondentes coincidem, ou são proporcionais. Por exemplo, duas instâncias $O_1 = [2, 2, 2, 2]$ e $O_2 = [6, 6, 6, 6]$ serão mapeadas para o mesmo ponto $(0,0)$ no plano de visualização. Além disso, um bom posicionamento dos pontos depende da ordenação das âncoras, o que varia de acordo com a coleção, sendo um problema NP-completo. Se duas âncoras que apresentam alta correlação forem dispostas em posições opostas, o ponto tenderá a ser posicionado no centro do *layout*, ao passo que se elas forem dispostas uma do lado da outra, o ponto tenderá a ser posicionado no raio da circunferência, entre os pontos que representam essas âncoras.

A técnica de projeção de agrupamentos *ProjClus*, proposta por Paulovich & Minghim (2006) separa as n instâncias de uma coleção em \sqrt{n} agrupamentos, utilizando bisseção com *k-means*, e os centróides de cada agrupamento são projetados através das técnicas FASTMAP e *Force Scheme* (Tejada et al., 2003), esse último uma simplificação do FDP. As instâncias de cada agrupamento são então projetadas em um espaço que contém apenas as instâncias do grupo. Por fim, os grupos com instâncias já projetadas são posicionados em um espaço comum, de acordo com a posição dos centróides. A técnica apresenta complexidade computacional menor ($O(n^{3/2})$), com n igual ao número de instâncias, e preserva instâncias pertencentes aos agrupamentos construídos no espaço original próximas.

Paulovich et al. (2008) desenvolveram uma outra técnica de projeção chamada *Least Square Projection* (LSP), baseada no estudo realizado por Sorkine & Cohen-Or (2004), que aplica mínimos quadrados na reconstrução e edição de malhas, em um método chamado *Least Square Meshes*. Essa técnica combina os benefícios de técnicas lineares e não-lineares de projeção, e tem o objetivo de criar uma superfície na qual os dados são agrupados por relações de proximidade, permitindo a inferência das relações existentes na coleção de dados. A LSP realiza dois processos principais: no primeiro, são escolhidos um subconjunto de pontos, chamados **pontos de controle**, resultantes da aplicação da técnica de agrupamento *K-means*. Em seguida, esses pontos são projetados, utilizando qualquer técnica de projeção convencional. No segundo, um sistema linear é construído, baseado nas relações de vizinhança dos pontos em seu espaço original, e nas coordenadas cartesianas dos pontos de controle no espaço reduzido. As soluções desse sistema linear determinarão as posições das instâncias no espaço de projeção. O método apresenta uma boa relação entre precisão e tempo de execução, apresentando complexidade computacional de $O(n\sqrt{n})$, com n igual ao número de instâncias.

A Figura 2.5 mostra um exemplo de uma projeção LSP para uma coleção de 1000 imagens categorizadas em 10 classes distintas: Tribos africanas, praia, construções, ônibus, dinossauros, elefantes, flores, cavalos, montanhas/geleiras e comida. Cada imagem é representada por um vetor de 150 descritores SIFT (Li & Wang, 2003), e são exibidas como um círculo, cuja cor representa a classe a qual ela pertence. É possível perceber a formação de grupos concisos e homogêneos, e alguns grupos mais sobrepostos. Diversas cores e formas nessas imagens apresentam características semelhantes, mostrando um particionamento consideravelmente fiel à estrutura real da coleção.

Paulovich et al. (2010) propõem uma técnica de projeção multidimensional chamada *Part-Linear Multidimensional Projection* (PLMP), para grandes coleções de dados, cujas características estão em um espaço Cartesiano multidimensional, e para as quais o esforço utilizado no cálculo das distâncias entre as instâncias se mostra proibitivo. Nessa técnica, apenas algumas instâncias representativas da coleção, chamadas pontos de controle, são projetadas utilizando um esquema não linear, sendo que o restante das instâncias é projetado utilizando um mapeamento linear sobre essas instâncias representativas. Os autores compararam a eficácia dessa técnica com outras 15

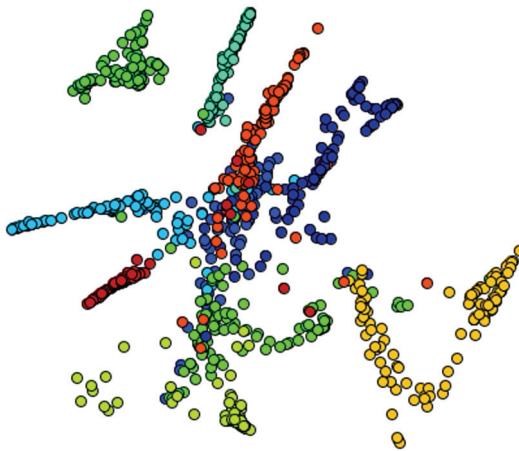


Figura 2.5: Projeção LSP de uma coleção de imagens divididas em 10 classes.

técnicas de projeção, e obtiveram mapeamentos satisfatórios, melhores do que as outras técnicas de projeção comparadas. Além disso, o tempo computacional exigido pela técnica se mostrou pelo menos uma ordem de magnitude menor para coleções grandes, mostrando o bom compromisso dessa técnica entre velocidade de geração e precisão do *layout*. Os autores também mostram uma aplicação da técnica para coleções de *streams*. Nesses casos, é necessário apenas um prévio conhecimento a respeito da faixa de valores dos atributos do *stream*, que permite construir o conjunto de pontos de controle sem percorrer a coleção. Uma vez conhecidos esses pontos de controle, a coleção será percorrida uma única vez, possibilitando um ganho considerável de tempo computacional. No entanto, os autores reforçam as limitações da técnica, que residem em sua impossibilidade em lidar diretamente com informação de dissimilaridade, e a necessidade de que o número de pontos de controle seja maior do que a dimensionalidade dos dados, prejudicando seu uso para coleções com altíssima dimensionalidade.

Os métodos previamente apresentados são capazes de gerar bons resultados para dados multidimensionais, com precisão variável no *layout* gerado. Em particular, LSP se mostrou capaz de agrupar dados relacionados, bem como separar grupos de dados. Essas técnicas propiciam um primeiro passo para a detecção de tendências e padrões em dados multidimensionais, e auxiliam o usuário a construir um modelo mental a respeito do conteúdo de determinada coleção.

Entretanto, algumas limitações das projeções multidimensionais podem dificultar o processo de análise dos dados. A primeira delas é o alto grau de sobreposição do *layout* gerado. Na tentativa de refletir as dissimilaridades através das distâncias no espaço Euclidiano, inevitavelmente muitos objetos são mapeados muito perto uns dos outros, de forma que a relação de vizinhança se torna indistinguível. Quanto mais instâncias a coleção possuir, maiores serão as chances dessa sobreposição ocorrer. A análise do *layout* pode levar à conclusão de que grupos homogêneos foram formados, quando, na verdade, muitos itens poderão estar escondidos, sobrepostos por outros. Além disso, existe a necessidade de fornecer informação adicional para determinar o tamanho, em

número de objetos, dos grupos formados, pois visualmente não é possível estimar corretamente a densidade dos grupos.

A segunda limitação dos *layouts* baseados em projeção é a dificuldade em visualizar relacionamentos locais. Tais *layouts* permitem uma análise global dos grupos formados, representados por conjuntos de instâncias próximas em algumas regiões do plano, mas essa análise fica difícil a medida em que o usuário se concentra em grupos de interesse, pois a precisão dentro desses grupos não é mantida. Isso ocorre porque a maioria das técnicas existentes utilizam mapeamentos globais do espaço multidimensional, dificultando a manutenção das propriedades observadas no *layout* como um todo em vizinhanças locais. Dessa forma, fica difícil a análise das relações entre instâncias em um determinado grupo de interesse, ou a realização de ajustes necessários para a inserção do conhecimento no processo.

Para amenizar esse problema, Paulovich et al. (2011) apresentam uma técnica de projeção chamada *Piecewise Laplacian-based Projection* (PLP), derivada da técnica de projeção LSP, que produz um *layout* que se adapta de acordo com a interação do usuário, permitindo a manipulação de dados multidimensionais de uma maneira flexível e altamente visual. O método baseia-se na utilização e manipulação de amostras da coleção. Para cada uma dessas amostras, um grafo de vizinhança independente é construído, e um conjunto de pontos de controle definido, que serão usados na criação e manutenção de sistemas Laplacianos associados. Quando o usuário modifica a posição dessas amostras no *layout*, seus grafos de vizinhança e pontos de controle associados são dinamicamente atualizados, modificando os sistemas Laplacianos e consequentemente o mapeamento na projeção. Os autores compararam a técnica, em termos de precisão do *layout* gerado, com outras 10 técnicas de projeção, e obtiveram resultados superiores na maioria dos casos. Eles demonstraram que tais mudanças são realizadas em frações de segundo, permitindo que a interação seja fluida. A Figura 2.6a mostra uma aplicação do método, na qual uma coleção de imagens inicialmente disposta em um *layout*, após uma sequência de menos de trinta interações, produz o *layout* mostrado na Figura 2.6b, demonstrando o potencial dessa técnica. No exemplo, a cor da borda nas imagens representa a classe, e a janela na parte superior direita representa os pontos de controle.

Seguindo a mesma ideia, Joia et al. (2011) apresentam outra técnica chamada *Local Affine Multidimensional Projection* (LAMP), baseada na teoria de mapeamento ortogonal para construir um conjunto de transformações locais precisas, que podem ser modificadas de acordo com interações por parte do usuário. A técnica também se baseia na utilização de amostras, e a cada uma delas é associado um mapeamento ortogonal afim. A manipulação dessas amostras pelo usuário modifica o mapeamento, e o *layout* se adapta de acordo com essa interação, possibilitando a inserção do conhecimento do usuário no processo. O diferencial dessa técnica é que sua formulação matemática permite que um conjunto pequeno de amostras seja utilizado, exigindo poucas interações para a incorporação do conhecimento do usuário no processo e aumentando sua flexibilidade. Os

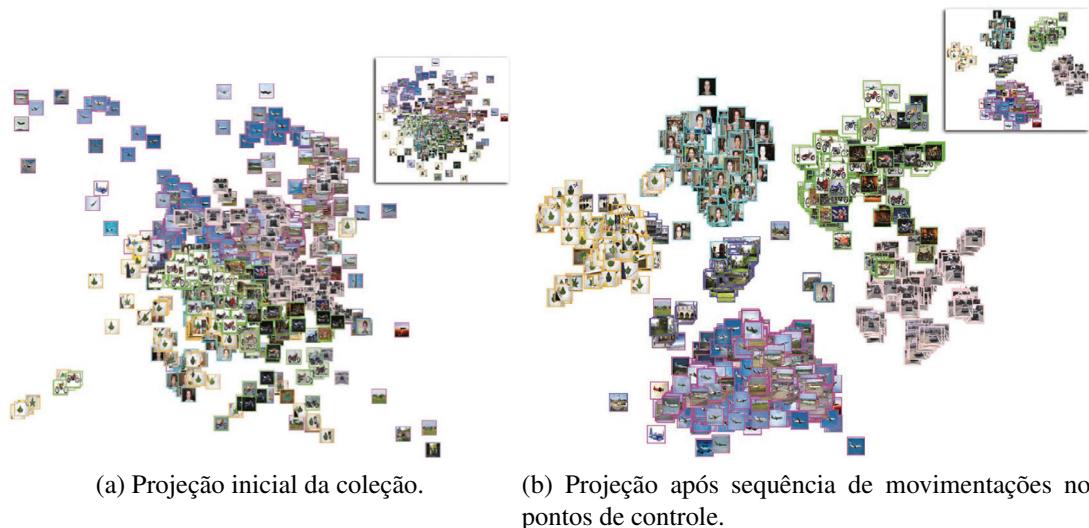


Figura 2.6: Projeção PLP de uma coleção de imagens, antes e depois de uma sequência de manipulações realizadas pelo usuário (Adaptado de (Paulovich et al., 2011)).

autores compararam essa técnica com outras 9 técnicas de projeção, e concluíram que ela produziu alguns dos *layouts* com maior índice de precisão. Além disso, a técnica se mostrou competitiva em termos de complexidade de tempo. Finalmente, o potencial da técnica LAMP na inserção do conhecimento do usuário no processo de mapeamento é explorado através de uma aplicação que visa correlacionar informações contidas em imagens e músicas (informações naturalmente não relacionadas), de forma a associar gêneros de música com tipos de imagens específicos, criando automaticamente apresentações de slides com som. Os resultados apresentados demonstram o potencial da técnica na construção de um novo paradigma para correlação de coleções de dados baseado em técnicas de visualização. A Figura 2.7 ilustra um exemplo de aplicação da técnica LAMP. Na Figura 2.7a, um conjunto de amostras aleatórias é mostrado, contendo 3 amostras por classe. Essas amostras representam os pontos de controle. O restante da coleção é projetado então de acordo com a posição desses pontos de controle, resultando no *layout* mostrado na Figura 2.7b. É possível notar no *layout* produzido que a técnica é capaz de projetar os dados de maneira consistente, com medida de coeficiente de silhueta (descrita no Capítulo 4) comparável a outras técnicas de projeção de alta precisão.

Visualização baseada em Grafos

As técnicas de visualização baseadas em grafos podem reduzir algumas das limitações apresentadas pelas técnicas de visualização baseadas em projeção. A abordagem mais intuitiva de representação desse tipo de estrutura utiliza pontos ou outra geometria como vértices que representam as instâncias da coleção, e linhas ou curvas como arestas que representam as relações existentes entre

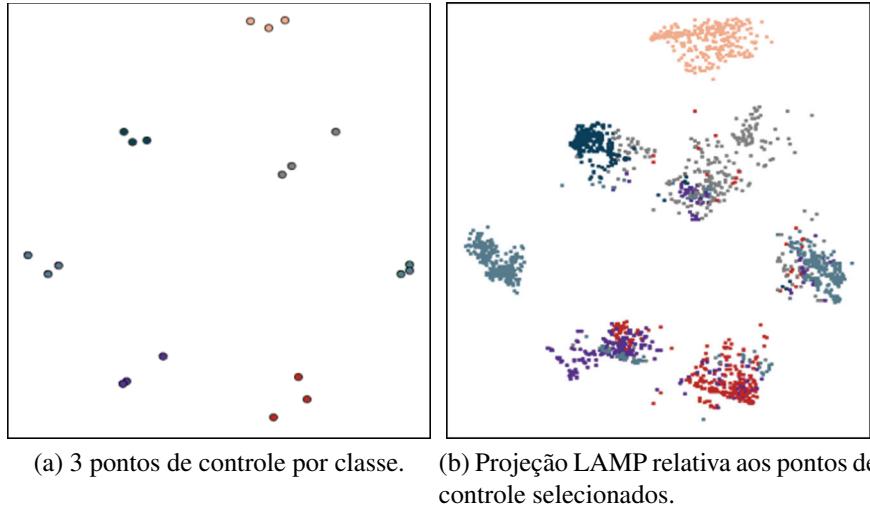


Figura 2.7: Projeção LAMP de uma coleção com 7 classes, com destaque para os pontos de controle, escolhidos aleatoriamente na coleção (Adaptado de (Joia et al., 2011)).

essas instâncias. A maioria dos algoritmos propostos utiliza essa abordagem, diferindo apenas na forma em que as arestas e os vértices são posicionados.

Algoritmos de posicionamento baseados em força tratam o grafo como um sistema físico (Eades, 1984), atribuindo forças aos vértices e arestas, que se movimentam de modo a minimizar a energia do sistema até atingir o equilíbrio. Os resultados obtidos com essa abordagem são visualmente bons, apesar do alto custo computacional. Já os algoritmos de posicionamento espectral utilizam os autovetores da matriz Laplaciana do grafo para minimizar uma função de energia aplicada a cada aresta. Nesse caso, a função de energia a ser minimizada é local, substituindo as forças de atração das arestas, que se baseiam em uma função de energia global. Outra abordagem para o traçado de grafos é o posicionamento hierárquico (Sugiyama et al., 1981), no qual os nós são posicionados em camadas, sendo útil para representar informações que possuem estruturas hierárquicas. Finalmente, algumas poucas abordagens (Wattenberg, 2006) foram criadas para visualizar grafos baseadas nos dados contidos nos vértices e nas arestas. Em muitas situações, esses dados são tão importantes quanto a própria estrutura topológica do grafo, pois ajudam a extrair novas informações não representadas explicitamente nos relacionamentos.

Muitas vezes, a estrutura do grafo e os relacionamentos presentes na coleção produzem um alto número de cruzamentos de arestas, o que pode causar confusão visual e prejudicar a análise do usuário. Em alguns casos, é possível lidar com esses grafos através da criação de agrupamentos de nós. Várias formas de agrupamento podem ser utilizadas. Abello et al. (2006) propõem uma ferramenta que agrupa certos nós que estão conectados a um mesmo nó, e exibe um grafo simplificado, no qual cada nó representa um agrupamento. O usuário pode então navegar pela estrutura hierárquica expandindo interativamente agrupamentos individuais. Já Loubier et al. (2007) pro-

põem uma ferramenta que agrupa os nós em classes, exibindo como resultado um grafo de classes que podem ser expandidas.

Outra abordagem para contornar o problema da grande quantidade de arestas presentes nos *layouts* baseados em grafo utiliza um procedimento de agregação de arestas (*bundling*) (Cui et al., 2008; Ersoy et al., 2011; Gansner et al., 2011), concentrando apenas arestas que permitem uma análise global da coleção. Através de ferramentas de interação, o usuário pode explorar as arestas originais através da seleção de uma ou mais agregações. Holten & Van Wijk (2009) apresentam um método de agregação baseado na representação por molas flexíveis que se atraem mutuamente. De acordo com os autores, esse método elimina a exigência de que haja uma hierarquia nos dados, e a exigência da construção de malhas de controle que resultam em arestas agregadas com alta variação de curvatura. Os resultados mostram a produção de um *layout* menos poluído, que revela padrões de alto nível presentes na coleção, com arestas agregadas suaves e de acompanhamento simples.

Herman et al. (2000) descrevem uma série de técnicas clássicas para criação e navegação de grafos em visualização de informação, muitas delas baseadas na construção de árvores. O algoritmo de Reingold-Tilford (Reingold & Tilford, 1981; Walker, 1990) é um algoritmo bastante conhecido de posicionamento baseado em árvore. Já o algoritmo de posicionamento radial (Eades, 1992) dispõe os nós em círculos concêntricos de acordo com a profundidade na árvore. Algoritmos de posicionamento hiperbólico também podem ser utilizados para representar árvores em duas ou três dimensões, permitindo observar o grafo com um efeito no qual os objetos próximos ao centro são projetados em maiores detalhes, enquanto que objetos distantes do centro são mostrados progressivamente menores e as arestas são apresentadas como curvas (Lamping & Rao, 1999). A grande vantagem desse tipo de visualização é a possibilidade de representar grafos com uma quantidade grande de vértices e arestas usando interação para analisar áreas de interesse.

Outra forma de organizar os objetos é utilizar uma árvore de similaridade. Essa árvore é construída com base em uma matriz simétrica, denominada matriz de similaridade, ou matriz de distâncias, que contém as distâncias entre todos os elementos de uma coleção.

Uma abordagem para a criação de árvores de similaridade é baseada em Árvores Geradoras Mínimas (*Minimum Spanning Tree*) (MST), criada a partir de um problema formulado inicialmente por Boruvka (1926), para construir um esquema econômico de uma rede de energia elétrica. O problema pode ser descrito da seguinte forma (Graham & Hell, 1985):

Dado um grafo ponderado G , no qual os nós representam instâncias, as arestas representam possíveis conexões entre as instâncias, e os pesos associados às arestas representam o custo dessas conexões, é possível construir um conjunto de arestas que conecte todas as instâncias, e possua um custo total mínimo. Esse conjunto de nós e arestas formarão uma árvore, ou seja, todos os conjuntos de arestas que formam ciclos serão removidos. A árvore com o custo total mínimo será a Minimum Spanning Tree.

Em uma árvore de similaridade construída utilizando MST, cada nó representa uma instância, e o peso associado à aresta entre dois nós representa a distância entre esses nós. Como resultado, arestas mais longas irão separar os grupos da coleção, e arestas mais curtas irão conectar nós próximos (semelhantes) em grupos.

Uma outra abordagem utiliza a aplicação de árvores filogenéticas para visualizar coleções de dados. Esse tipo de abordagem será detalhado no Capítulo 3.

2.3.3 Técnicas de Visualização aplicadas a Coleções de Imagem

A visualização de informação, conforme ilustrado anteriormente, também pode auxiliar a descoberta de diversos fenômenos em imagens. Diversos sistemas de CBIR (*Content Based Image Retrieval*) conseguem determinar relações de similaridade entre imagens em uma determinada coleção. Se os usuários desses sistemas possuem acesso a uma interface visual na qual as relações de similaridade podem ser facilmente visualizadas como distâncias em um plano ou espaço tridimensional, então eles podem se beneficiar mais desses sistemas (Chen et al., 2000).

Os *layouts* de visualização de imagens oferecem ainda um diferencial quando comparados com os de outras informações, tais como áudio ou texto. De acordo com Nakazato & Huang (2001), em sistemas de visualização textual, apenas o título e informações mínimas podem ser exibidas no modelo de visualização, sob o risco de gerar um *layout* poluído e de difícil compreensão. Assim, é difícil para o usuário julgar a relevância de cada documento para a coleção como um todo, exigindo que ele abra documento por documento para realizar essa análise. Esse problema é consideravelmente minimizado em *layouts* de visualização de imagens, pois o usuário necessita apenas da imagem para realizar a análise, que pode então ser exibida por completo, mesmo que em miniatura, sendo o acesso à imagem individual necessário apenas no último nível de análise.

Diversas pesquisas apresentam sistemas ou metodologias que permitem a mineração de dados em coleções de imagens através da utilização de técnicas de visualização. A utilização do método MDS (*Multidimensional Scaling*) (Torgerson, 1952), modificado com o objetivo de diminuir a sobreposição das imagens, foi proposto por Basalaj, denominado **Proximity Grid** (Basalaj et al., 1999; Basalaj, 2000), e resultou de uma pesquisa realizada pelo próprio autor, na qual os usuários relataram dificuldades em lidar com a sobreposição de imagens, e por isso uma parte deles preferia utilizar um *layout* baseado em grade de organização aleatória. Nesse método, os resultados da projeção utilizando MDS são inseridos em células de uma matriz bidimensional, de forma que apenas uma imagem esteja em uma célula, e que imagens semelhantes estejam em células vizinhas, enquanto imagens com pouca ou nenhuma semelhança estejam em células esparsas. Esse método foi utilizado por Rodden et al. (2001), com o objetivo de investigar se a visualização de imagens organizadas por similaridade pode auxiliar os usuários na busca de informações. O resultado foi

a organização das imagens sem nenhuma sobreposição (Figura 2.8), facilitando a visualização da coleção por esses usuários.

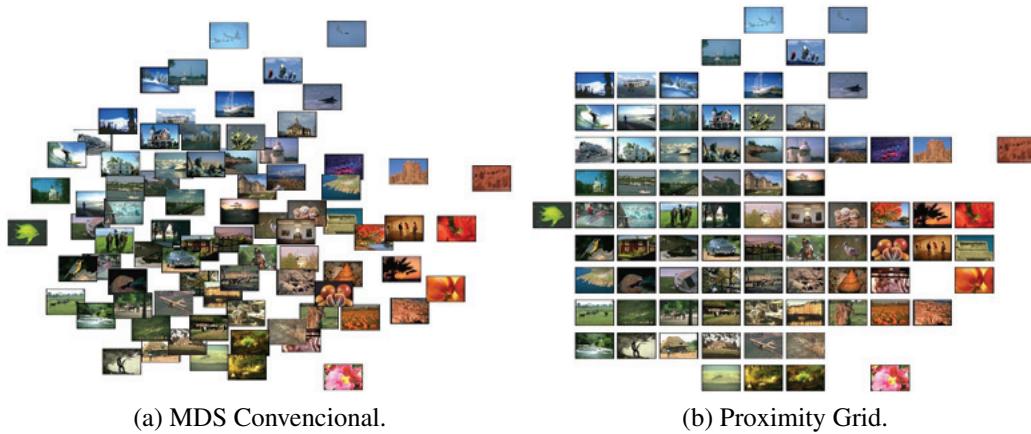


Figura 2.8: Modelos gerados utilizando-se MDS Convencional e *Proximity Grid*, para a mesma coleção de imagens (Adaptado de (Basalaj et al., 1999)).

Nakazato & Huang (2001) utilizam a técnica FASTMAP (Faloutsos & Lin, 1995) para a projeção dos dados em seu sistema **3DMars**¹, um sistema interativo de visualização de coleções de imagens, em um ambiente virtual 3D. O sistema pode projetar as imagens das coleções em um ambiente imersivo ou não-imersivo, e é capaz, de acordo com interações do usuário, de se reorganizar, em tempo real, aumentando a precisão dos resultados mostrados. Segundo os autores, essa técnica foi escolhida devido à alta velocidade de construção do modelo ($O(nk)$, onde n é o número de imagens da coleção, e k o número de dimensões), permitindo interatividade em tempo real com o usuário.

Ruszala & Schaefer (2004) apresentam como vantagens do 3DMars a possibilidade de utilizar uma CAVE, permitindo a visualização de muito mais imagens simultaneamente, em tamanho maior do que em outros sistemas, e utilizando três dimensões ao invés de duas. Além disso, a auto reorganização do sistema propicia uma interatividade e capacidade de exploração consideravelmente maior. Entretanto, eles ressaltam que, como a exibição inicial das imagens é feita de forma aleatória, pode haver certa confusão por parte do usuário em como iniciar sua navegação.

Ruger & Heesch (2004); Heesch & Ruger (2004) apresentam o NN^k (*k Nearest Neighbours*, k para o número de características considerado), uma técnica de recuperação baseada em conteúdo na qual as imagens são ranqueadas de acordo com diversas características. Para cada imagem buscada, são associados seus vizinhos mais próximos com relação a alguma característica, e o número de características para as quais essa relação de vizinhança ocorre. Esse número de características representa uma medida de similaridade entre as imagens. O *layout* utiliza uma busca em espiral para determinar posições livres na tela, de forma a não causar sobreposição na exibição dos re-

¹<http://www.ifp.illinois.edu/nakazato/3dmars/>

sultados. Os autores destacam que a rede criada expõe a riqueza de interpretação das imagens, mostrando a similaridade entre elas de acordo com cada característica. Essa técnica também é utilizada por Rüger (2006), aqui chamada de *Lateral Neighbors*, na qual, para cada imagem em consideração, chamada **imagem foco**, determinam-se os vizinhos mais próximos para cada combinação de características, gerando uma rede para a exibição. Os vizinhos laterais compartilham algumas propriedades da imagem foco (não necessariamente todas). Como consequência, os vizinhos laterais tendem a exibir os diversos significados da imagem focal, aumentando o poder representacional do modelo.

Os autores apresentam o software **AETOS**² para recuperação de imagens por conteúdo. Nesse software, cada imagem é representada por um vértice em um grafo direcionado. As arestas entre as imagens indicam que elas são vizinhas mais próximas, de acordo com uma ou mais características. Os comprimentos das arestas são proporcionais às características que representam as vizinhanças mais próximas. De acordo com os autores, a rede resultante mostra a riqueza semântica da coleção de imagens, possibilitando ao usuário explorar a coleção sob diversas óticas. Ao selecionar uma imagem, o sistema a centraliza, e exibe ao seu redor todas as imagens que representam a vizinhança mais próxima, mostrando uma rede, de acordo com o método NN^k . Esse *layout* é mostrado na Figura 2.9. Uma ou mais imagens podem ser adicionadas à uma consulta, de maneira a refinar os resultados. O sistema também permite que sejam feitas consultas textuais à coleção.

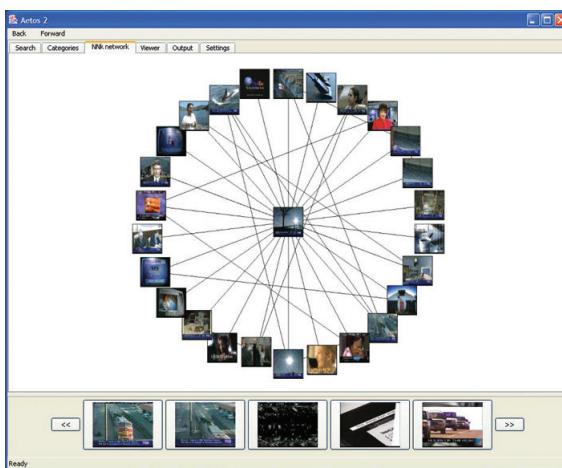


Figura 2.9: Visualização da rede de similaridade no sistema AETOS, utilizado o método NN^k (Adaptado de (May, 2004)).

A interface do software AETOS é utilizada em outro software disponibilizado pelos mesmos autores (May, 2004; Rüger, 2006), chamado **iBase** (também conhecido como **uBase**³). A ideia desse software é criar uma ferramenta *web* capaz de integrar o processo de busca e exibição de imagens, melhorando a interação com o usuário nesse processo e possibilitando maior precisão

²<http://mmis.doc.ic.ac.uk/demos/aetos.html>

³<http://technologies.kmi.open.ac.uk/ubase>

nos resultados de uma consulta. Além disso, ele implementa funcionalidades tais como exibição hierárquica e temporal. As consultas podem ser feitas através de textos ou imagens, ou pela combinação de ambos. O software implementa uma arquitetura no modelo cliente-servidor, possibilitando centralizar o processo de busca e montagem do que será apresentado na interface com o usuário. A Figura 2.10 mostra a interface do software iBase, e uma rede criada de acordo com determinada consulta.

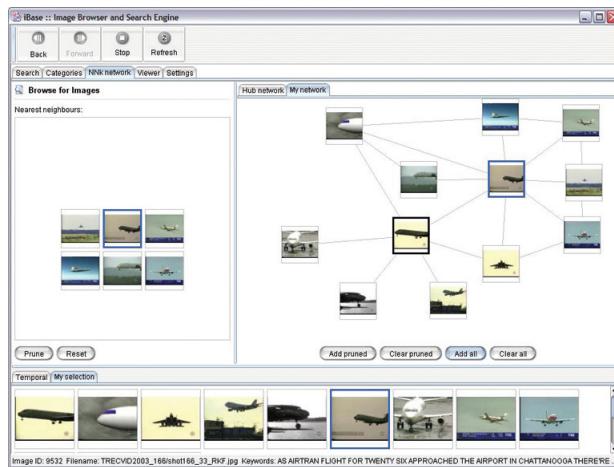


Figura 2.10: Software iBase mostrando a rede criada que representa os resultados de determinada consulta (Adaptado de (May, 2004)).

Eler et al. (2008, 2009) apresentam o software **Pex-Image**⁴, um *framework* visual que suporta todo o processo de análise e exploração visual de coleções de imagens e textos associados, possibilitando a avaliação de processos de manipulação de imagens e mineração de dados, entre outras tarefas. Segundo os autores, a integração desses dois tipos de informação em um mesmo ambiente de exploração aumenta a capacidade do *layout*, possibilitando a descoberta de mais informações nos dados. A Figura 2.11 mostra uma imagem da interface do Pex-Image.

Pex-Image implementa diversas técnicas de geração de *layouts* de visualização de coleções de imagens que enfatizam a similaridade entre elas, além de oferecer funcionalidades complementares para ajudar na exploração dos dados. O sistema recebe como entrada uma coleção de imagens, das quais extrai as características visuais e as combina formando vetores de características. Cada vetor representa uma imagem como um ponto de múltiplas dimensões. O sistema oferece então diversas técnicas de visualização baseadas em projeção, bem como técnicas baseadas em árvores de similaridade, e um conjunto de medidas de distância podem ser utilizadas para computar a posição das imagens. Essas imagens são então representadas por pontos no espaço de visualização, ou por miniaturas, para facilitar sua identificação. Considerando que a medida de dissimilaridade representa bem as diferenças entre as imagens, o modelo resultante proverá um mapa que aproximaré agrupará imagens semelhantes, provendo uma ferramenta útil na exploração da coleção.

⁴<http://infoserver.lcad.icmc.usp.br/infovis2/PExImage>

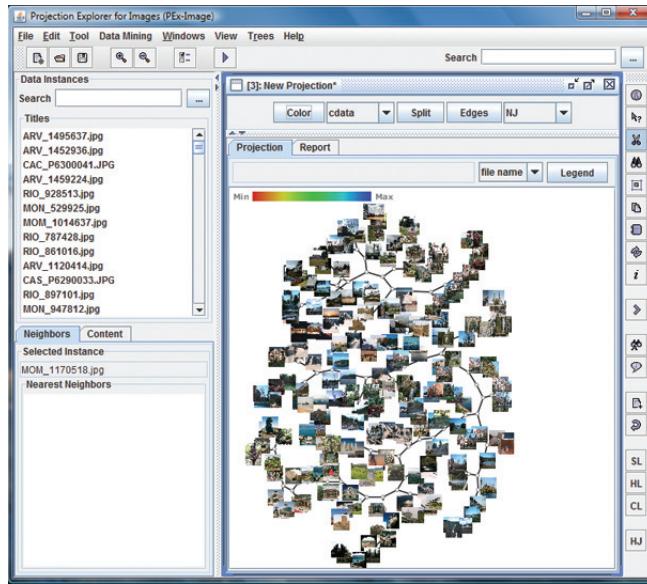


Figura 2.11: Interface do software Pex-Image, ilustrando o resultado de uma projeção.

Após visualizar o *layout*, diversas funcionalidades podem ser utilizadas pelo usuário, tais como aproximações (*zoom*), movimentação automática de pontos de acordo com a similaridade em relação a outras imagens, destaque para os vizinhos mais próximos de uma imagem, e a visualização de determinada imagem escolhida, ou de todas as imagens de determinado grupo ou seleção. O software também permite que seja feita uma coordenação entre vários *layouts* de uma mesma coleção de imagens (Figura 2.12), de forma que o usuário pode selecionar uma ou várias imagens em um *layout*, e visualizar ou a posição dessas imagens em outro *layout* (*identity coordination*), ou visualizar os *k* vizinhos mais próximos dessa seleção em outro *layout* (*distance coordination*), instantaneamente.

Os autores realizaram alguns experimentos que demonstram a eficácia dessa técnica na realização de tarefas relacionadas à exploração de coleções. O primeiro estudo descreveu como a técnica de projeção utilizada pode auxiliar na avaliação da eficácia de determinada característica das imagens para o agrupamento por similaridade. A técnica possibilitou concluir, para as coleções analisadas, que um agrupamento melhor pode ser obtido quando uma combinação entre 4 descritores de características é adotada. Tais descritores são: Descritores Fourier, Gabor, matrizes de co-ocorrência, intensidade média e desvio padrão, considerando toda a imagem.

O segundo estudo utilizou uma árvore de similaridade para avaliar as capacidades de agrupamento de diversas medidas de similaridade. No experimento, a técnica conseguiu agrupar imagens similares em ramos próximos, ou em um mesmo ramo, e permitiu concluir que a utilização da distância Euclidiana conseguiu separar melhor as classes de imagens na projeção.

Os autores ainda realizaram um último experimento, relacionado à visualização de sequências de proteínas. Foi realizada uma comparação entre a projeção gerada utilizando como dado de

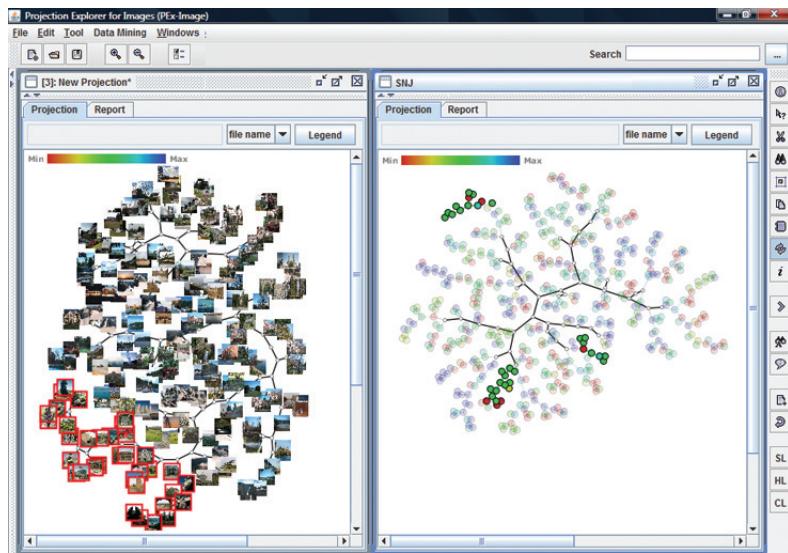


Figura 2.12: Coordenação entre duas projeções realizadas pelo Pex-Image, com destaque para um grupo de imagens.

entrada as sequências propriamente ditas, e a projeção gerada utilizando imagens que ilustram uma matriz 2D - representando a estrutura de uma proteína - na qual cada elemento i,j contém informações sobre as interações entre resíduos i e j dessa proteína. Essa técnica de projeção gerou um modelo que revelou particularidades nas informações residuais das proteínas que dificilmente seriam vistas analisando-se os dados individualmente.

Nguyen & Worring (2008) descrevem um conjunto de requisitos necessários para a obtenção de um *layout* eficaz: exibição de uma visão geral de toda a coleção, preservação da estrutura dos objetos da coleção no espaço de características e garantia da visibilidade dos objetos pelo usuário. De forma a garantir esses requisitos, propõem um sistema de análise visual, ilustrado na Figura 2.13, que implementa uma abordagem de visualização de coleções de imagens através da associação das técnicas SNE (*Stochastic Neighbor Embedding*) (Hinton & Roweis, 2003), LLE (*Locally Linear Embedding*) (Roweis & Saul, 2000) e ISOMAP, criando as técnicas ISOSNE e ISOLLE, respectivamente. Em ambos os casos, a computação direta da distância entre os pontos, calculada pela SNE e pela LLE é substituída pela computação da distância baseada em grafo, calculada pela técnica ISOMAP. A visão geral da coleção é obtida através do agrupamento das imagens e exibição apenas daquelas representativas de cada grupo, de forma que quando o usuário seleciona uma dessas imagens representativas, seu grupo é exibido. Esse procedimento é ilustrado na Figura 2.14. Um estudo representando um cenário de anotação de imagens em uma coleção foi realizado, e a abordagem proposta reduz os esforços para anotação das imagens significativamente em até 16 vezes, em especial se os grupos de imagens formados apresentarem uma separabilidade razoável, com entropia baixa.

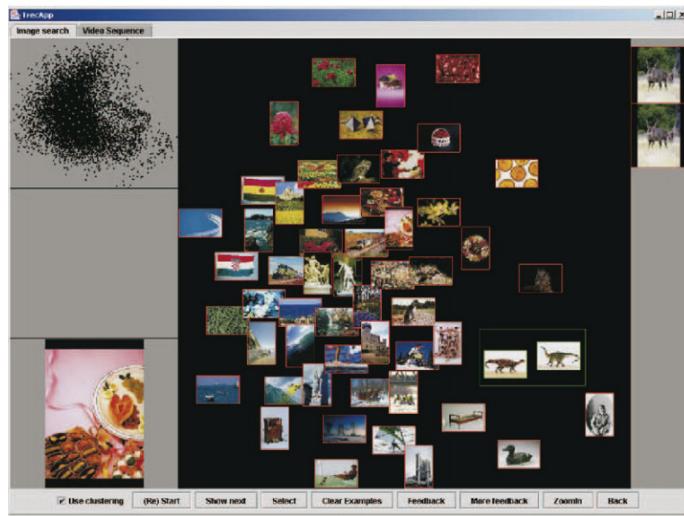


Figura 2.13: Tela do sistema de visualização, ilustrando o *layout* baseado em distâncias (Adaptado de (Nguyen & Worring, 2008)).

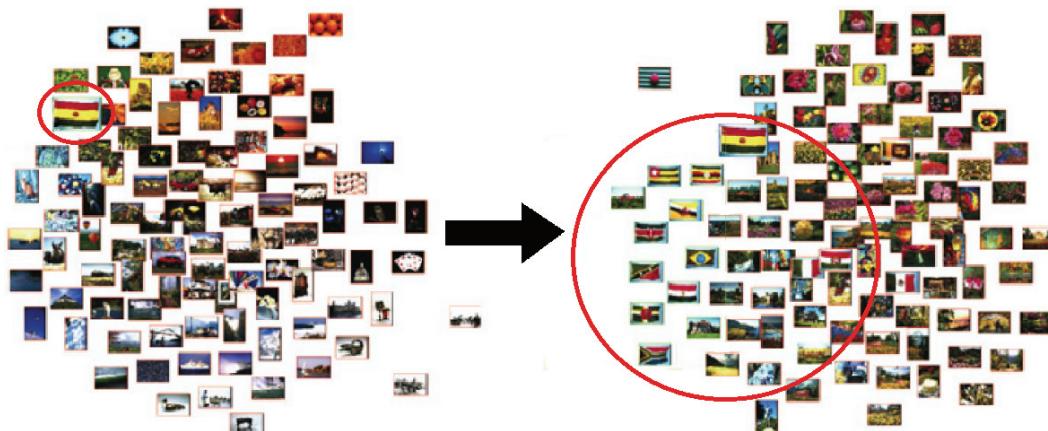


Figura 2.14: Exemplo de seleção de imagem representativa em uma coleção de imagens, e posterior visualização do grupo correspondente (Adaptado de (Nguyen & Worring, 2008)).

Camargo et al. (2010) propõem uma técnica de visualização de imagens que leva em conta anotações textuais associadas, de forma que as duas informações sejam exibidas juntas, possibilitando a identificação de relacionamentos entre essas duas fontes de informação. Para isso, eles utilizam a técnica ***Non-negative Matrix Factorization*** (**NMF**) para a construção de um espaço latente multimodal que representa características visuais e termos textuais combinados. A Figura 2.15 mostra o resultado da criação de um *layout* para um subconjunto da coleção COREL composta de 2500 imagens divididas em 25 classes. Descritores SIFT foram utilizados como características visuais, e os nomes das classes como termos textuais. De acordo com os autores, mesmo com a oclusão de imagens, em um *layout* não otimizado, o usuário consegue se orientar graças à presença dos termos textuais. Além disso, termos textuais com semelhança entre si aparecem próximos no *layout*.

graças ao compartilhamento de características visuais nas imagens de alguns grupos. Os termos textuais servem assim de guia para uma exploração de grupos de imagens.

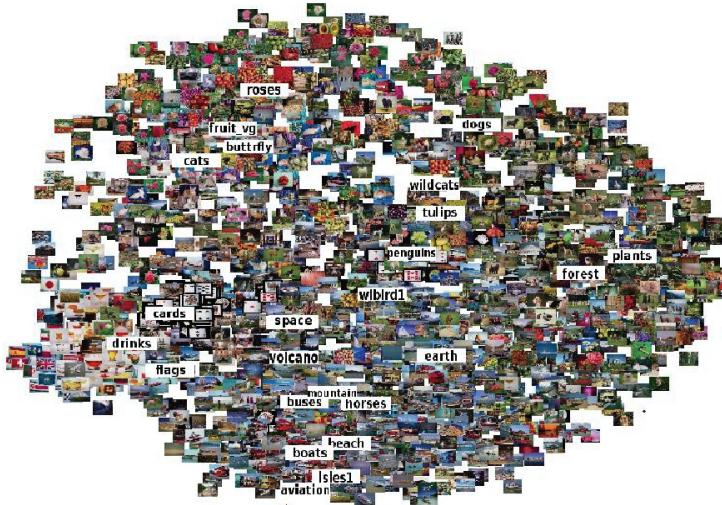


Figura 2.15: Visualização de um subconjunto da coleção COREL com imagens e termos textuais associados, utilizando ***Non-negative Matrix Factorization (NMF)*** (Adaptado de (Camargo et al., 2010)).

Doloc-Mihu (2011) apresenta um sistema adaptativo de recuperação de imagens (***Adaptative Image Retrieval System (AIRS)***), mostrado na Figura 2.16, no qual a interação do usuário é construída sobre uma interface de visualização que permite a realização de *Relevance Feedback* para melhorar o desempenho da busca. Esse sistema apresenta uma interface baseada em quatro visões selecionáveis que ilustram diversos relacionamentos entre as imagens de uma coleção, em diferentes níveis de detalhe, aumentando a capacidade de exploração dos resultados de uma busca. A primeira visão exibe um grafo no qual o tamanho das arestas equivale a distância Euclidiana entre duas instâncias. A segunda exibe um grafo não-direcionado no qual as instâncias se mostram igualmente separadas e distribuídas. A terceira utiliza a técnica *K-means* para dividir as instâncias em k grupos no *layout*. Finalmente, a quarta visão exibe um *layout* baseado em Coordenadas Paralelas. De acordo com a autora, as visões refletem informações semânticas de diversas naturezas, que possibilitam ao usuário entender a relevância de cada imagem nos resultados de sua busca. Essas mesmas informações semânticas são utilizadas pelo usuário no refinamento da busca, promovendo o *Relevance Feedback*.

Dois sistemas de visualização de coleções de imagens são apresentados por Schaefer (2011), chamados ***Hue Sphere Image Browser*** e ***Honeycomb Image Browser***, que representam as imagens utilizando descritores de cor e utilizam abordagens de exibição hierárquica organizadas em grade. No primeiro sistema, as imagens são dispostas de maneira esférica de acordo com um intervalo de matizes de cores cujos extremos situam-se nos polos da esfera. Os valores dos atributos das imagens são então mapeados em coordenadas de latitude e longitude nessa esfera. Um proce-

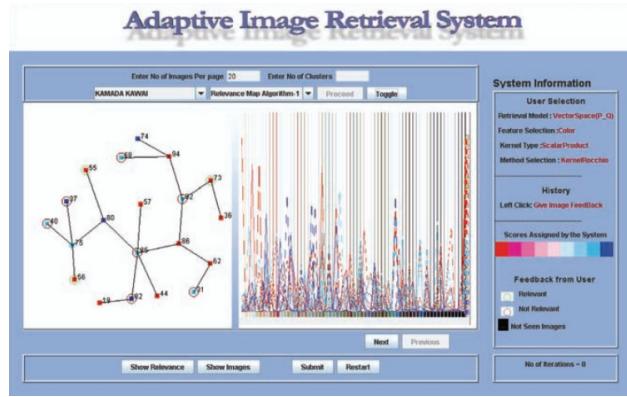


Figura 2.16: Tela do sistema AIRS, mostrando uma visão baseada em grafo e outra baseada em Coordenadas Paralelas, para um conjunto de 20 imagens correspondentes ao resultado do processo de *Relevance Feedback* (Adaptado de (Doloc-Mihu, 2011)).

dimento baseado em grade quadrada combinado com a aplicação de um agrupamento e construção de árvores para os grupos formados cria uma estrutura hierárquica que elimina a sobreposição de instâncias e permite a visualização de coleções de grande porte.

O segundo sistema utiliza uma estratégia semelhante, com a diferença que as imagens são organizadas em uma grade hexagonal. Segundo os autores, esse tipo de grade produz um deslocamento nas linhas e colunas de forma que a vizinhança das imagens consegue ser melhor representada do que em uma grade quadrada, além de aproveitar melhor o espaço de visualização. Além disso, os seis vizinhos de um hexágono são equidistantes da célula central, ao passo que em uma grade quadrada, os vizinhos nas diagonais estão mais distantes do que os vizinhos horizontais e verticais. A Figura 2.17 mostra um exemplo de visualização utilizando os sistemas *Hue Sphere Image Browser* (2.17a) e *Honeycomb Image Browser* (2.17b).

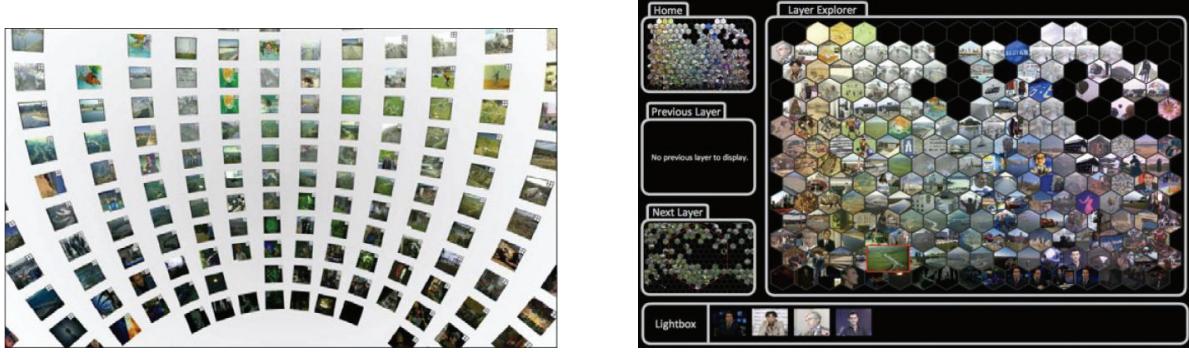


Figura 2.17: Visualização de coleções de imagens utilizando grade quadrada (2.17a) e grade hexagonal (2.17b) (Adaptado de (Schaefer, 2011))

Diversos outros exemplos de técnicas e sistemas de visualização de coleções de imagens podem ser encontrados em (Camargo & González, 2009).

2.4 Técnicas de Redução de Dimensionalidade

Como apresentado anteriormente, o processo de visualização e mineração de dados é dependente da maneira de representação das instâncias das coleções. A combinação de descritores de características é proposta em diversas pesquisas (Gehler & Nowozin, 2009; Schwartz, 2010). No entanto, essa combinação pode produzir representações com alta dimensionalidade, apresentando alto custo computacional de manipulação, além de informações redundantes que produzem *layouts* de difícil navegação e exploração por parte do usuário. Para contornar esse problema, procedimentos de redução de dimensionalidade podem ser empregados no espaço de características com o intuito de concentrar apenas as informações essenciais que captam a estrutura da coleção, de forma a melhorar o espaço original e realçar tendências características dos dados, sob determinada perspectiva.

Em aplicações de análise visual, a redução de dimensionalidade pode ser realizada sob duas perspectivas. A primeira delas é baseada na **seleção de características**, e consiste em escolher um subconjunto de características significativas dentre o conjunto original. É o caso do sistema proposto por Yang et al. (2003), chamado **Visual Hierarchical Dimension Reduction** (VHDR), que agrupa as dimensões originais dos dados em uma hierarquia de dimensões, de acordo com a similaridade entre elas. Isso possibilita ao usuário selecionar grupos de dimensões de seu interesse, exibindo apenas as dimensões desse grupo. De maneira semelhante, Wang et al. (2003) apresentam o sistema **DOSFA** (*Dimension Ordering, Spacing, and Filtering Approach*), que permite a filtragem de dimensões pouco representativas ou redundantes. Dessa forma, se várias dimensões apresentam alta similaridade entre si, então apenas uma delas é mantida, e se algumas dimensões apresentam pouca relevância para tarefas de exploração, elas são retiradas. O sistema oferece funcionalidades para que o usuário modifique o conjunto de dimensões a ser visualizado.

Os sistemas baseados em seleção de características apresentam como vantagem o fato de que conservam os valores originais dos atributos, evitando assim a perda de informação. No entanto, para coleções com centenas ou milhares de dimensões, tais como coleções textuais ou de imagens, o trabalho de selecionar as dimensões relevantes pode se tornar inviável.

A segunda abordagem para redução de dimensionalidade é baseada no **mapeamento de características**. Nesse caso, as dimensões são transformadas em um novo conjunto reduzido de características, que tenta conservar as propriedades e relacionamentos do conjunto de características original. Essas técnicas, por produzirem um novo espaço modificado, eventualmente resultam em perda de informação. No entanto, dependendo do processo, o espaço produzido pode realçar as tendências e estruturas existentes na coleção sob determinada perspectiva, melhorando o resultado de tarefas de visualização e mineração de informação.

A técnica **Principal Component Analysis** (PCA) (Jolliffe, 2002), apresentada na Seção 2.3.2, representa um exemplo de combinação de características, sendo a mais utilizada em basicamente

qualquer área do conhecimento (Abdi & Williams, 2010). Apesar de apresentar um alto custo computacional, PCA se mostra efetivo em vários casos. Van der Maaten et al. (2009) apresenta um estudo comparativo entre diversas técnicas de redução de dimensionalidade não lineares e a técnica PCA. Os experimentos revelaram que essas técnicas produzem bons resultados em algumas tarefas experimentais controladas, mas se mostram piores do que a PCA em tarefas reais de análise.

O processo de redução de dimensionalidade pode também ser utilizado para mapear coleções de dados para espaços de visualização contendo 2 ou 3 dimensões. As técnicas de projeção multidimensional, apresentadas neste capítulo, realizam esse mapeamento, procurando destacar a discriminabilidade dos grupos, ou reproduzindo os relacionamentos de similaridade existentes no espaço original de características. Tais técnicas são geralmente baseadas em *Multidimensional Scaling* (MDS), utilizando diversas fundamentações matemáticas, entre elas decomposição espectral obtidas de transformações em matrizes de similaridade (Torgeson, 1965), (Belkin & Niyogi, 2003) e (Koren et al., 2002). Um exemplo é a técnica ISOMAP, proposta por Tenenbaum et al. (2000), que consegue lidar com distâncias não Euclidianas, sendo útil na visualização de dados gerados por vários métodos de descrição de características. A própria técnica PCA pode ser utilizada com esse intuito.

No entanto, os métodos de projeção baseados em MDS apresentam aspectos globais que impedem sua utilização em casos nos quais é importante capturar vizinhanças próximas. Métodos como *Locally Linear Embedding* (LLE) (Roweis & Saul, 2000), *Landmark MDS* (LMDS) (de Silva & Tenenbaum, 2004) e *Pivot MDS* (Brandes & Pich, 2007) até conseguem capturar aspectos locais dos dados, mas seus esquemas baseados em decomposições globais dificultam sua utilização em aplicações que necessitem redefinir relacionamentos locais, pois qualquer atualização local exige que o modelo de geração das novas dimensões seja recalculado. Como esse modelo foi construído baseado em relacionamentos globais entre as instâncias, é possível que haja alterações não desejadas em regiões específicas.

Algumas técnicas trabalham com amostras da coleção que servem de base para a construção do novo espaço reduzido. Isso pode representar a incorporação do conhecimento do usuário no processo, através da manipulação dessas amostras. LMDS e *Pivot MDS* são exemplos desse tipo de técnica, mas são obrigadas a lidar com o alto custo gerado pelas decomposições. Pekalska et al. (1999) apresentam a técnica *Sammon Projection*, que consegue reduzir esse custo aplicando técnicas de otimização no processo de redução de dimensionalidade. A técnica LSP, bem como as técnicas derivadas PLP e LAMP apresentadas na Seção 2.3.2, também empregam amostras da coleção, aqui chamadas de pontos de controle. Entretanto, essas técnicas produzem modelos que podem ser utilizados apenas no mapeamento da coleção utilizada para sua criação, não sendo possível reutilizá-los no mapeamento de outras coleções. Além disso, algumas técnicas não oferecem suporte para que o usuário escolha o número de dimensões para as quais a coleção será reduzida.

Algumas abordagens de redução de dimensionalidade tem como objetivo melhorar o processo de classificação, e são chamadas de **Técnicas de Redução de Dimensionalidade Suficientes** (*sufficient dimension reduction Techniques*), cujo objetivo é encontrar um espaço reduzido que conteña informações que descrevam os rótulos de classe das instâncias (Suzuki & Sugiyama, 2010). A redução de dimensionalidade nesses casos ocorre de maneira supervisionada, através da criação de um modelo utilizando um conjunto de instâncias de treinamento. Um exemplo desse tipo de abordagem é a **Fisher Discriminant Analysis (FDA)** (Fukunaga, 1990), cujo objetivo é produzir um espaço reduzido que maximiza a distância inter-classe, e minimiza a distância intra-classe. De acordo com (Sugiyama et al., 2010) , a técnica produz bons resultados em coleções cujas instâncias de treinamento apresentam uma distribuição Gaussiana, com uma estrutura de covariância compartilhada. No entanto, caso essas instâncias sejam *outliers*, ou pertençam a classes cuja estrutura apresente vários subgrupos, a técnica não produzirá bons resultados (Fukunaga, 1990). Além disso, existe uma limitação quanto ao número de dimensões do espaço reduzido, que deve ser menor do que o número de classes. Algumas estratégias conseguem contornar esses problemas, tais como a técnica **Local FDA (LFDA)** proposta por Sugiyama (2007), que emprega a mesma a análise realizada pela técnica FDA, mas de forma local, em cada classe. Isso faz com que a estrutura da classe não tenha tanta influência no modelo criado. Já a técnica **Semi-Supervised LFDA (SELF)** (Sugiyama et al., 2010) utiliza instâncias rotuladas e não rotuladas no processo, de forma semi-supervisionada, impedindo que haja degradação do desempenho nos casos em que poucas instâncias rotuladas estão disponíveis para o treinamento. Diversos resultados de aplicação da técnica SELF em uma coleção de imagens mostram boa separabilidade no espaço reduzido. Além disso, a combinação dessa técnica a um classificador *1-Nearest Neighbor* apresentou bons resultados para coleções de documentos textuais.

Li et al. (2011) desenvolveram uma técnica de Redução de Dimensionalidade Suficiente baseada em **Principal Support Vector Machines (PSVM)**. A ideia é dividir as variáveis de resposta em regiões, e utilizar PSVM para encontrar os hiperplanos ideais que separam essas regiões. Os hiperplanos encontrados são então alinhados através das componentes principais de seus vetores normais, produzindo um estimador do espaço reduzido sem nenhum viés. A generalização da técnica para redução de dimensionalidade é realizada através da representação dos vetores normais no espaço de Hilbert. Os autores compararam o desempenho da técnica com outras 3 técnicas de redução de dimensionalidade, aplicadas a uma coleção de dados para reconhecimento de vogais, e obtiveram mapeamentos melhores do que as outras técnicas comparadas.

Lacoste-Julien et al. (2008) desenvolveram a técnica **DiscLDA**, que representa uma variação da técnica **Latent Dirichlet Allocation (LDA)** (Blei et al., 2003). LDA utiliza modelos probabilísticos baseados em métodos Bayesianos, e representa as instâncias de uma coleção como um conjunto de tópicos pré-definidos, calculando o espaço reduzido com base na proporção de tópicos em cada instância. Dessa forma, a técnica DiscLDA consiste em incorporar informações a respeito dos

rótulos de classes na técnica LDA, na forma de novos tópicos para as instâncias, maximizando o poder de discriminação do modelo criado. Os autores realizaram um experimento comparando o espaço reduzido de uma coleção de documentos textuais pela técnica DiscLDA em uma classificação utilizando SVM, mostrando uma taxa de erros menor quando comparado à utilização de um espaço reduzido da mesma coleção pela técnica LDA, no mesmo processo.

Outra técnica, chamada *Partial Least Squares* (PLS) (Wold, 1985), também tenta encontrar espaços representativos, chamados de **variáveis latentes**, nas características das instâncias de uma coleção. PLS representa uma classe de métodos estatísticos utilizados para diversas tarefas relacionadas à análise de dados com alta dimensionalidade, tais como discriminação, seleção de características, tratamento de dados ausentes e regressão (Boulesteix & Strimmer, 2007). Uma descrição detalhada do funcionamento do PLS será apresentada no Capítulo 4.

Um estudo comparativo de diversas outras técnicas de redução de dimensionalidade aplicadas a tarefas de classificação de dados pode ser conferido em (Thangavel & Pethalakshmi, 2009; Carreira-Perpinan, 2011).

2.5 Classificação Visual de Coleções de Dados

A exploração de uma coleção de imagens pode se tornar uma tarefa simples se essa coleção estiver organizada de uma maneira significativa. A classificação de imagens representa uma maneira de obter tal organização. O processo de classificação de dados consiste em organizá-los em grupos previamente definidos, representando assim uma maneira de extrair informação para reconhecer padrões e objetos homogêneos (Lu & Ip, 2010).

Um classificador automático utiliza aprendizado supervisionado, e suas decisões baseiam-se no aprendizado realizado a partir de um conjunto de treinamento. Esse conjunto contém instâncias já classificadas, idealmente em todas as classes desejadas. Dessa forma, a cada instância de treinamento, o classificador se ajusta de forma a descobrir um padrão que defina cada uma das classes desejadas, e ao final da fase de treinamento, ele estará pronto para classificar as instâncias da coleção. Dessa forma, a ideia da fase de treinamento é dotar o classificador com a capacidade de **generalização**.

Diversos autores (Ruger & Heesch, 2004; Keim et al., 2005; Rüger, 2006) defendem a importância da inclusão do usuário no processo de recuperação de imagens e no processo de classificação, combinando a flexibilidade, criatividade e conhecimento do ser humano com o poder computacional atual. Segundo Zhou & Huang (2003), a necessidade do usuário nesse processo se deve ao fato de que imagens se enquadram em um espaço de representação contínuo, enquanto conceitos semânticos são melhores descritos em subespaços discriminativos. Por exemplo, imagens de carros são mais facilmente classificadas através do atributo forma, enquanto imagens de paisagens são mais facilmente classificadas através do atributo cor. Além disso, usuários diferentes

em momentos diferentes possuem interpretações diferentes, ou utilizações diferentes para a mesma imagem, e isso torna o trabalho de classificadores automáticos eficaz apenas para os casos em que a coleção de imagens possui um conjunto de características semelhante e bem definido, o que não ocorre na maioria dos casos.

A ideia de utilizar o conhecimento do usuário na adaptação de sistemas de mineração de dados, denominada ***Relevance Feedback***, é muito utilizada como forma de interação usuário-computador em sistemas CBIR (Ciocca et al., 2009; da Silva et al., 2010), nos quais demonstrou apresentar considerável aumento de desempenho (Ruger & Heesch, 2004). Utilizado também na recuperação de documentos de texto, essa ideia descreve uma maneira de aprendizado semi-supervisionado que ajusta o sistema de acordo com informações coletadas do usuário, em interações realizadas por ele (Grigorova et al., 2007).

Um exemplo de aplicação do *Relevance Feedback* para a classificação de imagens de sensoriamento remoto é apresentada por Dos Santos et al. (2011), que propõem uma técnica semi-automática de classificação na qual o usuário pode interagir com o sistema indicando regiões de interesse ou regiões que devem ser descartadas, em uma lista de imagens exibidas. Essa informação é empregada novamente no sistema, de forma a combinar descritores de regiões que auxiliem a direcionar a perspectiva do classificador para as preferências do usuário. Esse processo ocorre de forma iterativa, até que os resultados sejam satisfatórios para o usuário.

Um método chamado ***Trace Ratio Relevance Feedback (TRRF)*** (Yang et al., 2012) utiliza *Relevance Feedback* no refinamento da representação de dados multimídia. Esse algoritmo utiliza informações sobre a distribuição de dados multimídia no espaço de características juntamente com informações fornecidas pelos usuário. A técnica é avaliada na recuperação de imagens, dados de posicionamento e movimentação 3D e mídias cruzadas, em comparação com técnicas que não utilizam interação com usuário, obtendo melhores resultados em diversas situações.

O *Relevance Feedback* se mostra, dessa maneira, como um conceito fundamental em um sistema de classificação visual de dados, no intuito de criar um processo iterativo de ajuste do classificador às necessidades do usuário.

O conceito de múltiplas visões coordenadas (CMV) (Roberts, 2007) pode representar também uma abordagem de grande utilidade para a classificação visual, pois permite a visualização de diferentes aspectos de uma coleção, revelando relacionamentos entre as instâncias que poderiam permanecer ocultos. O sistema PExImage, apresentado na Seção 2.3.3 implementa o conceito de CMV, através da coordenação entre vários *layouts* de uma mesma coleção de imagens. Eler (2011) apresenta um estudo detalhado da utilização de CMV na exploração de mapas de similaridade para coleções de documentos, apresentando diversos benefícios na aplicação dessa ideia na exploração desse tipo de coleção.

Finalmente, outra estratégia de possibilitar a classificação visual de dados é chamada de *active learning*, na qual o classificador é treinado interativamente a partir de anotações feitas pelo usuário

em amostras informativas. Tuia et al. (2011) apresentam um estudo comparativo de diversas técnicas baseadas em *active learning* para sensoriamento remoto. De acordo com os autores, diversas características dessa área de atuação fazem com que os classificadores falhem caso o conjunto de treinamento utilizado seja ruim. Dessa forma, *active learning* se torna útil pois permite que o usuário utilize heurísticas que ordenem conjuntos de instâncias não rotuladas de acordo com uma função de incerteza com respeito à classe a qual pertencem, informações essas que são utilizadas no ajuste do classificador.

Já Joshi et al. (2012) propõem um sistema que utiliza um modelo de *active learning* multi-classe que exige do usuário respostas binárias (sim/não) como *feedback* para o aprendizado do classificador, facilitando a interação com o usuário. Os autores realizaram estudos que comprovam a facilidade do usuário em interagir com o sistema, em termos de quantidade de esforço na interação com o classificador.

De acordo com Ankerst (2001), os sistemas de mineração de dados visual baseiam-se na utilização de três tipos de abordagem: (1) aplicação de técnicas de visualização independentes dos algoritmos de mineração de dados, (2) aplicação de técnicas de visualização para exibição dos resultados da mineração de dados, e (3) integração de técnicas de visualização e algoritmos de mineração de dados de forma a exibir o processo de classificação para o usuário. Segundo Zhang et al. (2009), poucos sistemas podem ser enquadrados na terceira categoria.

Zhang et al. (2009) apresentam o VDM-RS, um sistema de análise visual de dados também utilizado para classificação de imagens de sensoriamento remoto. O sistema oferece aos usuários quatro visões dos resultados da classificação, sendo três delas interfaces tradicionais: mapas espaciais, matrizes de erro e tabela de dados. A última visão exibe, através de uma árvore de decisão, os passos seguidos pelo classificador na categorização de uma instância, possibilitando um rastreamento e exploração do processo de classificação, e consequentemente a compreensão dos passos executados pelo classificador para classificar uma instância corretamente ou incorretamente. Segundo os autores, isso ajuda o usuário a compreender o funcionamento do processo, de forma a saber como melhorá-lo. A Figura 2.18 mostra um exemplo da visão do processo de classificação implementada no sistema. Esse sistema, no entanto, apenas mostra o processo ao usuário, não permitindo nenhuma interação direta no intuito de ajustar o classificador.

A construção e adaptação de modelos de árvores de decisão apoiadas pelo usuário são exploradas por Do (2007). A técnica proposta coordena diversas visões da coleção, utilizando técnicas de visualização como Matrizes de Dispersão e Coordenadas Paralelas, que auxiliam o usuário a entender como a árvore de decisão construída classificará os dados. Uma apresentação visual da árvore de decisão auxilia o usuário na interação com o modelo, acompanhando o processo de classificação de forma a modificá-la de acordo com sua necessidade. Os estudos realizados pelos autores demonstraram que a árvore de decisão construída utilizando essa abordagem apresenta melhores

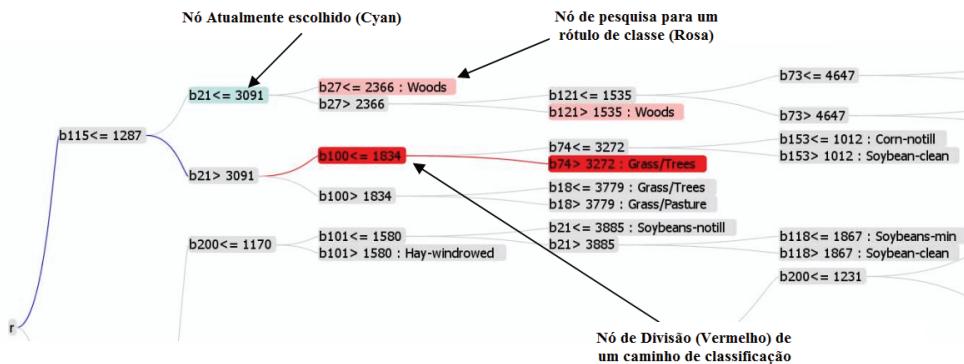


Figura 2.18: Visualização do processo de classificação através de uma árvore de decisão exibida pelo sistema VDM-RS (Adaptado de (Zhang et al., 2009)).

resultados de classificação, quando comparados com árvores de decisão automática, reforçando o papel do usuário nesse processo.

Tuia et al. (2009) apresentam um sistema de classificação visual que utiliza *active learning* na classificação de imagens de sensoriamento remoto. Baseado em heurísticas pré-definidas, o classificador ordena as instâncias e automaticamente escolhe aquelas consideradas mais representativas para o ajuste do classificador. As instâncias selecionadas são rotuladas manualmente pelo usuário, que alimenta o processo de forma iterativa. Os autores aplicaram a técnica na classificação de imagens hiperespectrais de alta resolução, utilizando SVM, e perceberam que o número de instâncias de treinamento pode ser reduzido para 10% utilizando o método proposto, ainda alcançando os mesmos níveis de precisão obtidos utilizando grandes conjuntos de treinamento.

2.6 Considerações Finais

Este capítulo apresentou um conjunto de conceitos e técnicas relacionadas ao processo de visualização e classificação de informação, com destaque para coleções de imagens.

Uma representação visual de uma coleção de dados pode comunicar claramente ao usuário o seu conteúdo informacional, concentrando-se em suas características essenciais e omitindo detalhes desnecessários. Além disso, o usuário se torna um agente ativo no processo de mineração das informações, pois consegue, além de visualizar as relações entre os dados, interagir com o *layout*, tendo uma visão geral, ou concentrando-se em fenômenos particulares. É possível também identificar diversas vantagens na utilização de técnicas de visualização para a exploração de coleções de imagens, principalmente pelo fato de que existe uma facilidade e eficácia maior em representar visualmente esse tipo de informação, em relação a outros tipos de informação como texto ou áudio. Isso porque as imagens representam naturalmente informações visuais, que podem ser aliadas aos *layouts* possibilitando que o usuário julgue mais facilmente o seu conteúdo, tornando o processo de compreensão ainda mais rápido.

De forma a melhorar a representação das instâncias e consequentemente produzir *layouts* que facilitem as tarefas de análise visual, as técnicas de redução de dimensionalidade possuem um importante papel na transformação dos espaços de alta dimensionalidade das coleções em novos espaços reduzidos que realcem características e tendências sob determinada perspectiva. Para tarefas de classificação, esse processo representa uma importante ferramenta para destacar as características que promovam a melhor separabilidade entre classes, facilitando a escolha de instâncias pelo usuário para a construção de um modelo de classificação eficaz.

Finalmente, é possível verificar o potencial na inserção da visualização na classificação de coleções de dados, através de técnicas de classificação visual. Os *layouts* construídos com os resultados de processos de classificação auxiliam o usuário a compreender melhor as decisões tomadas pelos classificadores, que tem condições de participar do processo e adaptá-lo às suas necessidades. No entanto, poucos sistemas oferecem uma associação direta entre técnicas de visualização e sistemas de classificação que promova a inserção efetiva do usuário no processo.

O próximo capítulo apresenta um estudo sobre as árvores de similaridade, apresentando seus benefícios para a classificação visual de imagens, juntamente com diversas melhorias no algoritmo de construção dessas árvores, melhorias essas desenvolvidas durante o doutorado com o intuito de possibilitar a utilização dessa técnica de visualização de maneira mais eficaz.

Árvores de Similaridade para Visualização de Coleções de Dados

3.1 Considerações Iniciais

Como mostrado no Capítulo 2, apesar de as técnicas de projeção multidimensionais apresentarem diversas vantagens no processo de visualização de informação, algumas de suas limitações podem dificultar a compreensão de uma coleção de dados por parte do usuário. Uma dessas limitações consiste na incapacidade de garantir o nível de precisão, observado em uma análise global do *layout*, em análises locais. Ao analisar um grupo na projeção, nota-se uma queda significativa na consistência da vizinhança entre as instâncias desse grupo, que a princípio a relação de similaridade deveria ser capaz de representar. Algumas abordagens hierárquicas, como a apresentada por Paulovich & Minghim (2008) podem amenizar esse problema, mas dependem de um processo de agrupamento inicial, o que dificulta que uma análise seja realizada nos limites entre os grupos. Além disso, como as técnicas *Piecewise Laplacian-based Projection* (PLP) (Paulovich et al., 2011) e *Local Affine Multidimensional Projection* (LAMP) (Joia et al., 2011), apresentadas no Capítulo 2, realizam um particionamento do espaço multidimensional, não podem ser aplicadas a matrizes de similaridade, dificultando sua utilização em aplicações que definem relações de similaridade sem utilizar a definição de espaço de características, tais como similaridade entre textos baseada em comparação de sentenças.

Além disso, *layouts* produzidos por técnicas de projeção podem apresentar um alto grau de sobreposição de instâncias e confusão visual, pelo fato de que essas técnicas tentam manter, a todo custo, os relacionamentos observados em vizinhanças potencialmente correlacionadas. Dessa forma, pode ser difícil realizar um estudo sobre a densidade visual dos grupos apresentados.

Nesse contexto, as árvores de similaridade surgem como uma alternativa para a exploração de dados multidimensionais. Ao posicionar as instâncias em ramos de uma árvore, a similaridade é organizada em níveis, representando uma abordagem natural para a interpretação de graus de similaridade. Este capítulo apresenta os conceitos relacionados à utilização de árvores de similaridade para a visualização de informação, com foco em coleções de imagens, além de modificações implementadas no contexto deste projeto de doutorado, para a obtenção de melhorias relacionadas à qualidade visual do *layout* e custo computacional. Finalmente, é apresentada uma aplicação das árvores de similaridade no processo de classificação visual de imagens.

Este capítulo é uma síntese de um artigo publicado no ***IEEE Transactions on Visualization and Computer Graphics*** (IEEE TVCG). O conteúdo completo do artigo pode ser encontrado no Apêndice B.

3.2 Árvore de Similaridade **Neighbor Joining**

O estudo das relações evolucionárias em um grupo de organismos pode ser feito com o auxílio de uma árvore filogenética, na qual as folhas representam os indivíduos ou espécies, e cada aresta representa a relação entre duas espécies. O tamanho dessa aresta indica a distância evolucionária entre as espécies (J. H. Choi & Cho, 2000), e a topologia resultante define relações de ancestralidade entre os indivíduos.

Um dos métodos mais utilizados para a construção de uma árvore filogenética é o *Neighbor Joining* (NJ), criado por Saitou & Nei (1987). Essa técnica, adaptada para a criação de árvores de similaridade, utiliza a ideia de encontrar pares de instâncias mais próximas em uma coleção de dados, de modo a minimizar o comprimento e o número de ramos da árvore gerada. O Algoritmo 3.1 apresenta as etapas do método *Neighbor Joining*. É gerada uma árvore sem raiz, representando apenas as distâncias entre as instâncias. O algoritmo recebe como entrada uma matriz de distâncias (similaridade) $D_{n \times n}$, com as distâncias entre todos os pares de instâncias da coleção, de acordo com alguma medida de similaridade, e produz uma árvore de similaridade com n nós-folha, e $n - 2$ nós virtuais com grau 3. R_i é a distância média do nó i para todos os outros nós em D , capturando a noção de mudança evolucionária. Em cada passo do algoritmo, os dois nós mais próximos em D são removidos da matriz, e substituídos pelo nó virtual x , para o qual uma nova linha é inserida em D . As novas distâncias de x para todos os outros nós restantes na matriz é calculada de acordo com as fórmulas apresentadas no algoritmo.

Algoritmo 3.1: Neighbor Joining

Considere $n =$ número de elementos da matriz D ;

Associe cada linha i da matriz D com um nó folha i ;

repita

Selecionar o par de nós (i, j) com o mínimo valor de S_{ij} ;

Crie um novo nó x que conecte os nós i e j , com arestas de tamanhos L_{ix} e L_{jx} , respectivamente;

Adicione a linha x a D , com valores D_{xk} para cada coluna $k \neq i, j$;

Remova as linhas i e j de D ;

até $n = 3$;

Conekte os três nós restantes na árvore;

$$\begin{aligned} S_{ij} &= D_{ij} - R_i - R_j; \quad R_y = \frac{1}{n-2} \sum_k D_{yk} \\ L_{ix} &= \frac{1}{2}(D_{ij} + R_i - R_j); \quad L_{jx} = D_{ij} - L_{ix}; \quad D_{xk} = \frac{1}{2}(D_{ik} + D_{jk} - D_{ij}) \end{aligned}$$

Cuadros et al. (2007) estudou a aplicação de árvores filogenéticas para visualizar coleções de documentos textuais. Nessa aplicação, as folhas representam os documentos da coleção, e os nós internos representam instâncias hipotéticas, com conteúdo intermediário. Além disso, o comprimento das arestas define a distância (dissimilaridade) entre os documentos.

Encontrar uma medida de similaridade que represente fielmente os relacionamentos existentes entre as instâncias no espaço original de características é um desafio chave para a construção de árvores de similaridade e projeções. No processo de mapeamento de coleções de dados, tanto a extração de características quanto a utilização de medidas de similaridade adequadas precedem a aplicação de técnicas de visualização, e a árvore de similaridade produzida é responsável por refletir essas escolhas. Isso torna a própria árvore de similaridade uma ferramenta promissora para a avaliação e obtenção dessas medidas.

A Figura 3.1 mostra um exemplo comparativo entre um *layout* produzido pela técnica de projeção LSP (3.1a), e uma árvore de similaridade NJ (3.1b,3.1c), para uma coleção de dados textuais. Esse exemplo mostra que a organização das instâncias em ramos é consistente com a organização da projeção LSP. No entanto, a árvore adiciona estrutura os grupos em níveis de similaridade, e reduz consideravelmente a sobreposição e confusão visual. Além disso, as vizinhanças locais são claramente visualizadas na árvore, de acordo com a matriz de similaridade.

Após a construção de uma árvore NJ, um algoritmo de desenho radial de grafos (Bachmaier et al., 2005) pode ser aplicado, utilizando por exemplo um procedimento simplificado de construção de *layout* baseado em força (Tejada et al., 2003), como mostrado na Figura Figura 3.1c, minimizando a sobreposição de nós e permitindo uma inspeção da organização dos ramos.

A Figura 3.2 mostra um exemplo de árvore NJ para a coleção de imagens COREL, cujas informações são descritas na Tabela 3.1. Na Figura 3.2a, as cores indicam as classes das instâncias. Na Figura 3.2b, as instâncias são exibidas como miniaturas das imagens que representam. É pos-

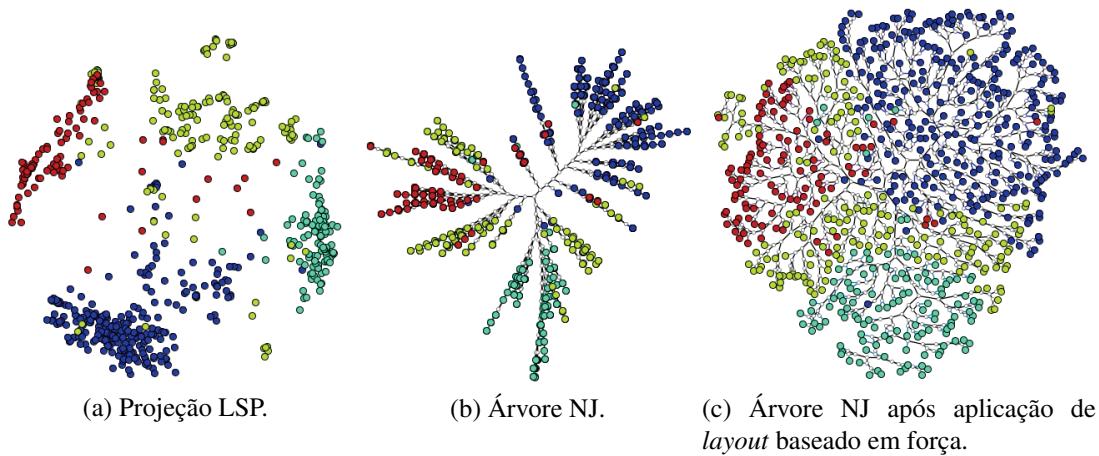


Figura 3.1: Comparação entre uma projeção LSP e uma árvore NJ, para uma coleção de 675 documentos textuais.

sível perceber que os ramos organizam bem a maioria das classes da coleção, e que quase todas as instâncias cujas características não expressam claramente os padrões de suas classes situam-se no núcleo da árvore.

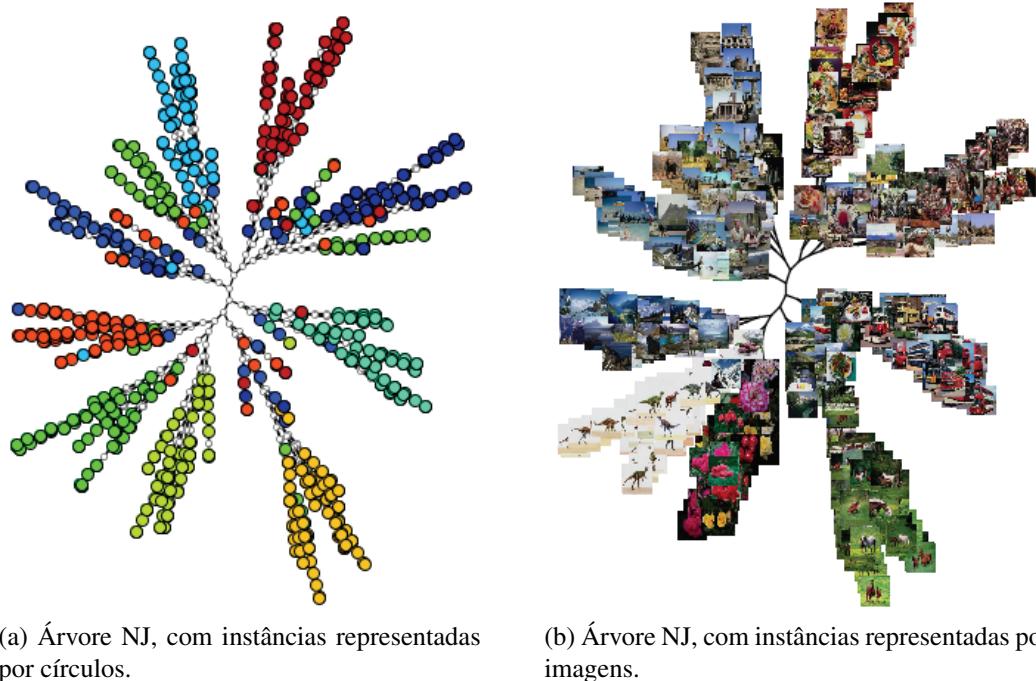


Figura 3.2: Exemplo de uma árvore NJ para a coleção COREL.

Alguns estudos demonstrando a capacidade de segregação de grupos e outros benefícios da construção de árvores de similaridade utilizando o método *Neighbor Joining* foram realizados por Eler et al. (2008). Valdivia (2007) apresenta também estudos que comparam as árvores NJ com algumas técnicas de projeção aplicadas a coleções de documentos textuais, demonstrando o potencial dessa técnica na preservação da estrutura de classes dessas coleções.

A Figura 3.3a mostra a árvore NJ para a coleção COREL, com um ramo selecionado, que contém, na sua maioria, imagens do grupo **ônibus**. Esse ramo é detalhado na Figura 3.3b. É possível notar evidências de que as características de cor das imagens criam subgrupos, ou seja, existe um subgrupo de ônibus vermelhos, outro de ônibus amarelos, e assim por diante.

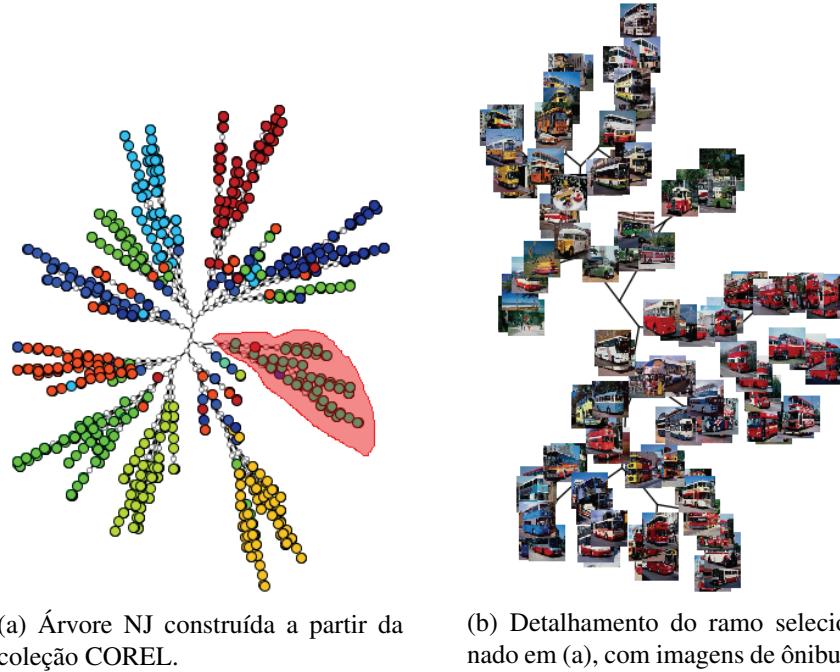


Figura 3.3: Seleção e exploração de um ramo da árvore NJ construída a partir da coleção COREL, mostrando os níveis de similaridade

Para confirmar a evidência de que árvores de similaridade mantêm, em níveis mais baixos, as mesmas propriedades observadas nos níveis mais altos, foi utilizada a análise *Neighborhood Preservation* (Paulovich & Minghim, 2008), que mede a proporção de vizinhos mais próximos de um ponto, no espaço de visualização, que também são vizinhos desse ponto no espaço original.

É importante ressaltar que a medida de distância utilizada para o cálculo da vizinhança na árvore NJ é diferente daquela utilizada para as projeções. Essas últimas organizam o *layout* de forma que pontos com distância Euclidiana pequena apresentam similaridade maior do que pontos com uma distância Euclidiana grande. Para as árvores NJ, essa raciocínio não é válido, pois a proximidade no *layout* é definida pelo algoritmo de desenho da árvore, e não possui nenhuma relação com a distância Euclidiana entre um ponto e os demais. Assim, é possível, por exemplo, que dois pontos pertencentes a ramos diferentes, e por isso com pouca similaridade entre si, sejam posicionados próximos um do outro no *layout*. Dessa forma, a distância entre dois pontos a e b na árvore NJ foi definida como a soma dos pesos das arestas que formam o menor caminho conectando a a b .

Além da análise *Neighborhood Preservation*, um **gráfico de distâncias** pode ser utilizado para avaliar quanto um mapeamento 2D reflete, no espaço de visualização, as mesmas distâncias obser-

vadas no espaço original. Esse gráfico mostra pontos cujas coordenadas no eixo X representam as distâncias no espaço original, e cujas coordenadas no eixo Y representam as distâncias no espaço de visualização. Idealmente, a nuvem de pontos formada deve coincidir com uma reta que passa pela origem e tem inclinação de 45 graus.

As Figuras 3.4 e 3.5 mostram a análise *Neighborhood Preservation* e o gráfico de distâncias da árvore NJ para a coleção COREL completa, e para o ramo mostrado na Figura 3.3b, respectivamente. É possível observar que as vizinhanças locais e a reconstituição das distâncias no espaço original seguem um mesmo padrão, ou apresentam resultados melhores, quando comparados com as vizinhanças globais. Esses resultados são consistentes em todos os ramos da árvore.

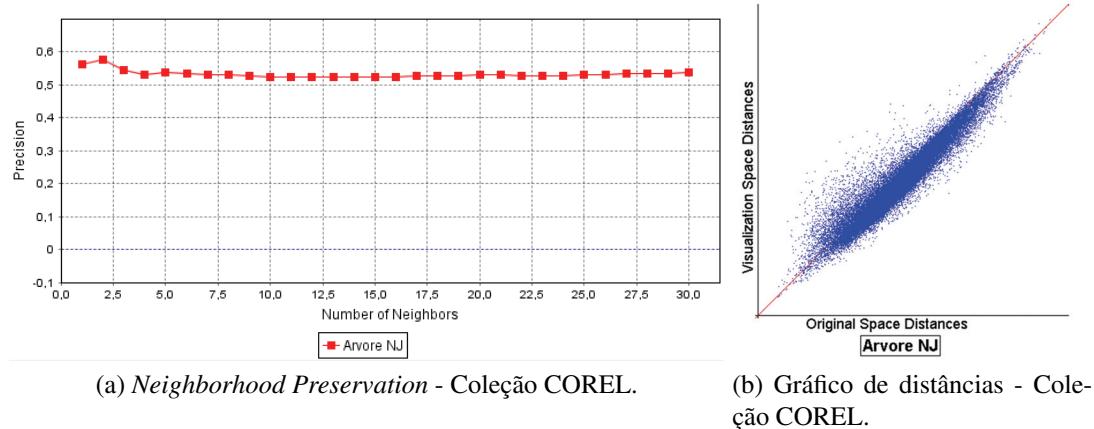


Figura 3.4: Análises de precisão relativas à árvore da coleção COREL.

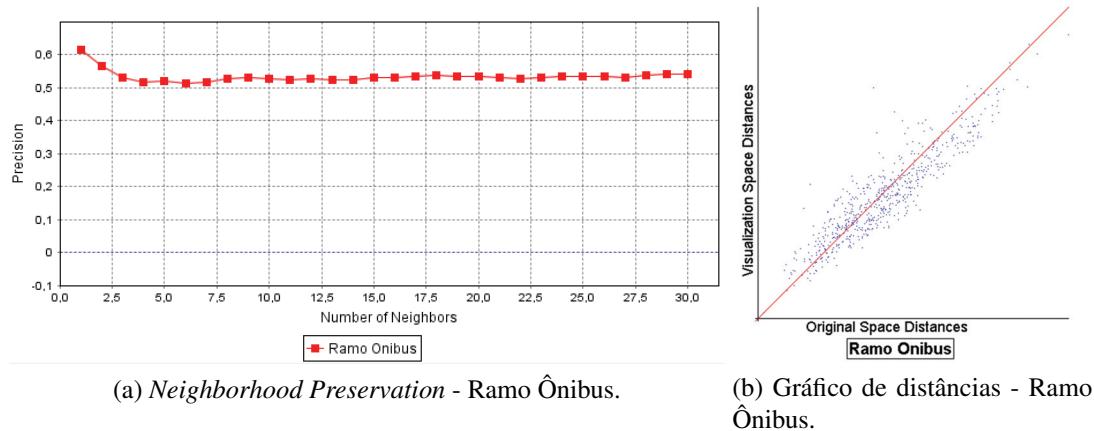


Figura 3.5: Consistência na precisão do ramo com imagens de ônibus, em comparação com a árvore completa mostrada na Figura 3.4a.

3.3 Abordagens para melhorias na construção de árvores *Neighbor Joining*

Apesar de mostrar bons resultados na construção de *layouts* de visualização, sendo capaz de agrupar instâncias semelhantes em um mesmo ramo, ou ramos próximos, e de separar instâncias diferentes em ramos distantes, as árvores de similaridade construídas pela técnica *Neighbor Joining* apresentam alguns problemas que podem diminuir a eficácia da análise. Nesta seção são apresentadas algumas dessas limitações, juntamente com abordagens investigadas e desenvolvidas durante o projeto de doutorado para melhorar o *layout* das árvores produzidas, especialmente em termos de aproveitamento do espaço de visualização e de desempenho na geração da estrutura que representa a árvore.

3.3.1 Promoção de Nós

Um problema com o *layout* produzido pela técnica *Neighbor Joining* é o volume de pontos e arestas gerado no espaço de visualização. Como apresentado na Seção 3.2, para n instâncias em uma coleção, $(n - 2)$ nós virtuais são criados. Esses nós não fazem parte da coleção, mas ocupam boa parte do espaço de visualização, gerando um *layout* poluído.

Isso pode ser percebido na Figura 3.6, que mostra um exemplo de uma árvore NJ, para uma coleção de 648 documentos textuais, uma coleção de tamanho relativamente pequeno. Para coleções maiores, o problema se agrava consideravelmente.

De modo a reduzir o número de nós virtuais nas árvores NJ, foi implementada uma operação determinística de reescrita de grafos (Ehrig et al., 2006; Rozenberg, 1997) baseada na substituição de nós virtuais por nós da coleção, sempre que detectada uma configuração de nós específica. Essa operação é chamada de **Promoção de Nós**, e é descrita a seguir.

Seja uma árvore NJ T , na qual existe um par de nós-folha u e v , conectados a um nó virtual a , e que outro nó virtual b conecte esse nó virtual a e um outro nó-folha w , como mostrado na Figura 3.7a. A operação de promoção de nós baseia-se no fato de que nenhum outro nó é tão similar a w do que a (durante a construção de T), e por isso nenhum outro nó é mais similar de u e v do que w . Assim, a pode ser substituído por w e b pode ser removido sem perda no poder representacional. A relação entre u , v , w e a é válida apenas para a topologia criada pelo algoritmo *Neighbor Joining*, não tendo valor para o relacionamento induzido pela matriz de similaridade, no caso de algum relacionamento existir.

A promoção de nós pode ser formalmente definida em termos da ocorrência de um padrão e uma substituição na árvore, como mostra as Figuras 3.7b e 3.7c, e consiste em substituir cada ocorrência desse padrão, em ordem decrescente da distância do nó a no padrão para o nó que

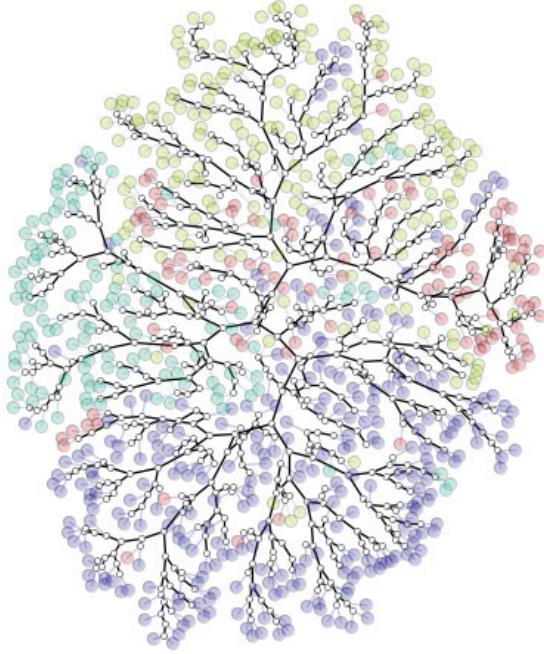


Figura 3.6: Exemplo de árvore NJ de uma coleção com 648 documentos textuais, com seus nós virtuais e arestas destacados, mostrando a alta densidade de pontos gerados pela técnica.

reside no centro da árvore. Os pesos das arestas, durante a substituição, são chamados de ω_r , e são definidos de acordo com os pesos que ocorrem no padrão, chamados de ω_p , computados pelo algoritmo de construção da árvore. Chamando de T_1 , T_2 e T_3 os nós que conectam a subárvore ao resto da árvore, é possível obter $\omega_r(w, T_1) = \omega_p(b, T_1) + \omega_p(a, b)/2 + \omega_p(w, b)/2$ e $\omega_r(w, T_i) = \omega_p(a, T_i) + \omega_p(a, b)/2 + \omega_p(w, b)/4$, para $i = 2, 3$. A árvore resultante é chamada de **Promoting Neighbor Joining (PNJ)**.

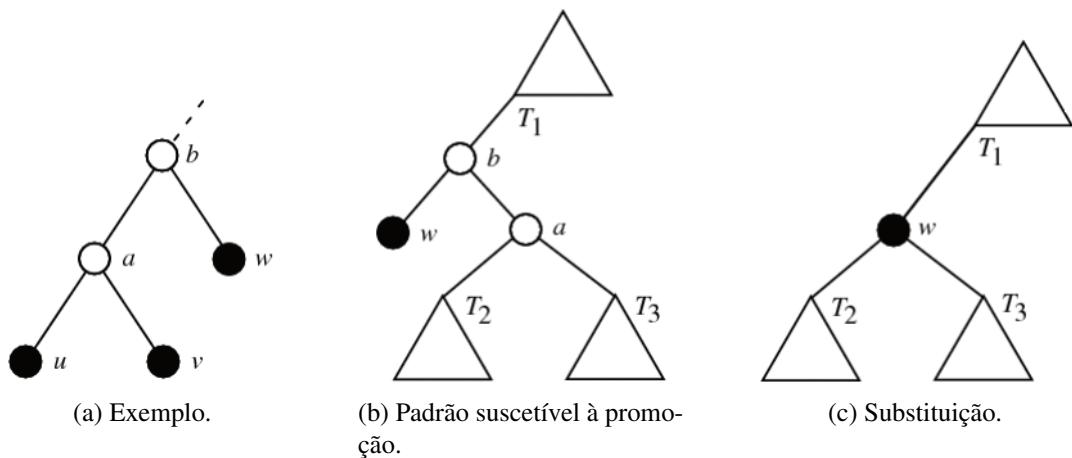


Figura 3.7: Operação de promoção de nós. Círculos preenchidos representam nós da coleção, e triângulos representam subárvores.

Uma procura $O(n)$ é suficiente para encontrar os nós que apresentam o padrão, o centro da árvore, e a determinação dos valores de distância. A ordenação dos nós é $O(n \log n)$. A substituição em cada padrão encontrado apresenta tempo constante. Dessa forma, a operação de promoção será $O(n \log n)$. Diversos experimentos mostraram que o tempo computacional para realizar a operação de promoção é praticamente desprezível comparado ao tempo de geração da estrutura total.

A promoção de nós permite que as capacidades de organização e exploração das árvores NJ sejam mantidas, mas reduzindo o número de nós virtuais e consequentemente de arestas desnecessárias, resultando em um *layout* mais limpo, que permite um acesso mais direto às instâncias da coleção através dos ramos da árvore. A Tabela 3.2 mostra os resultados da aplicação do procedimento em três coleções de dados, cujas informações são descritas na Tabela 3.1. A medida de distância utilizada foi a Euclidiana, para todas as coleções.

Tabela 3.1: Descrição das coleções utilizadas nos experimentos relacionados à árvores de similaridade.

Coleção	Natureza	Instâncias	Classes	Dimensões	Descritores
COREL	Imagens diversas	1000	10	150	150 descritores SIFT (Li & Wang, 2003)
MEDICAL	Imagens de Ressonância Magnética	540	12	28	Descritores de Fourier do histograma da imagem e energia computada do descriptor de Fourier da imagem 2D, intensidade média e desvio padrão computados de toda a imagem
OBJECTS	Imagens diversas	5097	45	1000	100 descritores SIFT sobre extractores Harris-Laplace (Griffin et al., 2007)

Tabela 3.2: Comparação do número de nós gerado pela árvore NJ original e pela árvore PNJ.

Coleção	Nós	Nós Virtuais	Nós Virtuais PNJ	Economia
COREL	1000	998	412	59%
MEDICAL	540	538	186	65%
OBJECTS	5097	5095	2437	53%

Nas árvores PNJ, a utilização do espaço de visualização é mais racional, e a poluição visual reduzida, como pode ser visto na Figura 3.8, que mostra uma comparação entre os *layouts* produzidos pela técnica NJ original (3.8a) e PNJ (3.8b). Em coleções maiores, o impacto é ainda mais evidente.

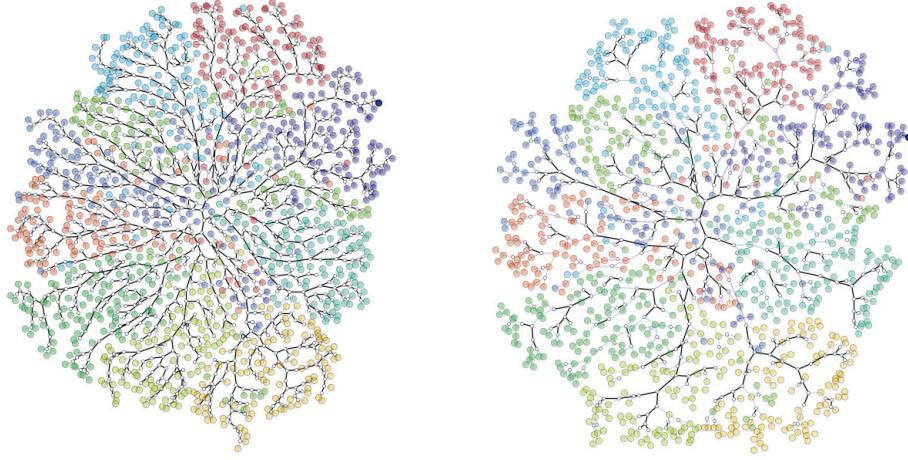


Figura 3.8: Comparação entre árvores NJ e PNJ para a coleção COREL, destacando os nós virtuais e arestas.

O resultado da análise *Neighborhood Preservation* é mostrado na Figura 3.9. É possível notar, através da queda brusca dos números em vizinhanças locais, a deturpação na vizinhança-2, alterando ligeiramente a distribuição das distâncias. Isso é esperado, visto que os nós virtuais são ‘fabricados’, e compõem valores ideais de vizinhança. Assim, qualquer substituição representa uma adaptação dessa vizinhança que será sempre pior do que a existente. Nos demais níveis de vizinhança, não existe diferença significativa entre os números apresentados para a árvore PNJ e para os apresentados para a árvore NJ.

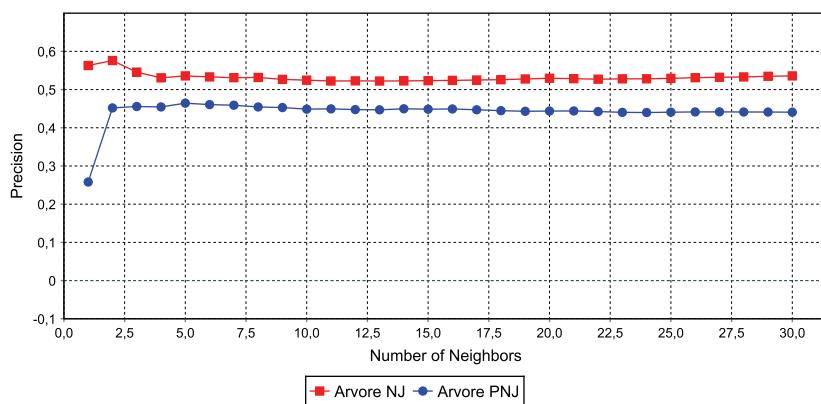


Figura 3.9: Análise *Neighborhood Preservation* comparando Árvores NJ e PNJ.

A Figura 3.10 mostra os gráficos de distância para a coleção COREL, de forma a comparar a preservação das distâncias no espaço original e no espaço de visualização, para as árvores NJ e

PNJ. É possível verificar que a operação de promoção de nós dispersa ligeiramente a nuvem de pontos do gráfico, mantendo no entanto o padrão de distribuição observado no gráfico da árvore NJ. Além disso, a curva do gráfico relativo à PNJ mostra um pequeno deslocamento para baixo, quando comparada com a curva do gráfico relativo à NJ. Isso é causado pela redução do número de arestas, e consequentemente da adequação dos valores de distâncias relativos aos nós envolvidos no procedimento de promoção. Como a distância entre as instâncias na árvore leva em consideração as arestas que conectam essas instâncias, é esperado que o procedimento cause esse comportamento.

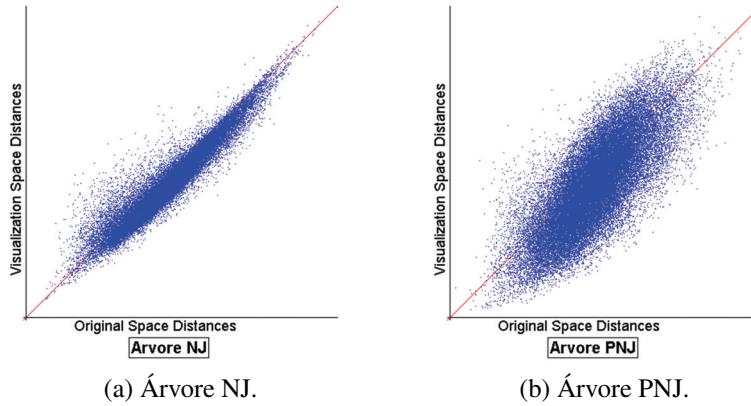


Figura 3.10: Gráfico de Distâncias comparando Árvores NJ e PNJ para a coleção COREL.

3.3.2 Algoritmos para Aceleração na Construção de Árvores NJ

A necessidade de se recalcular parte da matriz de distâncias a cada iteração do processo de construção da árvore NJ, devido à inserção do nó virtual gerado na iteração anterior, pode resultar em um alto custo computacional na geração da árvore, mesmo para coleções pequenas. Além disso, a procura pelo valor de S_{ij} mínimo na matriz D , para determinar o próximo par de nós a serem combinados, exige uma verificação em cada posição de D , resultando em $O(n^2)$, com n igual ao número de instâncias.

Para solucionar ou amenizar tais problemas, alguns algoritmos que modificam o processo de montagem da árvore foram investigados e implementados durante a execução deste trabalho de doutorado. De um modo geral, esses algoritmos utilizam duas estratégias, baseadas em estruturas de dados especializadas (Mailund et al., 2006; Simonsen et al., 2008; Wheeler, 2009), ou heurísticas que sacrificam a precisão (Elias & Lagergren, 2005; Evans et al., 2006), e podem ser diferenciados basicamente por algumas modificações direcionadas a situações específicas. Assim, de forma a representar essas duas estratégias, e verificar o impacto na velocidade de geração e na precisão da árvore obtida, duas técnicas foram selecionadas, e são descritas a seguir. As descrições e notações utilizadas são baseadas nos passos de construção da árvore NJ apresentados no Algoritmo 3.1. Adicionalmente, S_{ij} é denotado como S_{min} , para uma determinada matriz D .

O primeiro algoritmo selecionado é chamado ***Rapid Neighbor Joining (Rapid NJ)*** (Simonsen et al., 2008). Esse algoritmo produz árvores idênticas às produzidas pelo algoritmo original. Sua ideia baseia-se no fato de que, na procura por S_{min} em determinada linha i da matriz D , através da avaliação de $S_{ij} = D_{ij} - R_i - R_j$, o valor de R_i é fixo. Assim, o algoritmo mantém uma matriz auxiliar, na qual as linhas são ordenadas, e cujas células contém índices para D . A procura em uma linha ordenada i dessa matriz pára assim que $S_{ij} - R_i - R_{max} > S_{min}$, para o qual R_{max} é o valor máximo de R_i dentre todas as linhas da matriz ($R_j \leq R_{max}$).

Um trabalho extra é inserido com a ordenação das linhas nessa matriz auxiliar, bem como a manipulação de colunas removidas e avaliação de R_{max} . O algoritmo ainda é $O(n^3)$, mas os experimentos mostraram que a estratégia diminui显著mente o número de células visitadas, na busca por S_{min} , fazendo com que ele se mostre mais eficiente do que outros algoritmos de aceleração na construção da árvore.

O segundo algoritmo selecionado é chamado ***Fast Neighbor Joining (Fast NJ)*** (Elias & Lagergren, 2005). Esse algoritmo mantém um conjunto com $O(n)$ células candidatas na matriz D . A procura por S_{min} é restrita a esse conjunto, que contém, inicialmente, os valores mínimos de S_{ij} para cada linha i . Quando as linhas i e j são conectadas a um nó virtual x e removidas de D , as células candidatas com as quais eles contribuem são removidas e uma nova célula candidata, para a nova linha x é adicionada. Essa abordagem resulta em um algoritmo que não produz a mesma árvore que o algoritmo original de construção da árvore NJ, e que no pior caso, é $O(n^2)$. Os autores da técnica mostraram que quando a medida de similaridade é próxima de ser aditiva, as árvores produzidas são idênticas àquelas produzidas pelo algoritmo original.

Neste trabalho, foi utilizada uma versão mais conservadora desse algoritmo. Exatamente n células candidatas são mantidas, uma para cada linha de D . No momento em que as linhas i e j são conectadas a um nó virtual x e removidas de D , as células candidatas com as quais elas contribuem são também removidas, e as células candidatas restantes, que envolviam i ou j são recomputadas, resultando em uma melhora das escolhas de S_{min} . A versão modificada desse algoritmo é $O(n^3)$, mas os tempos de execução na prática são muito próximos de $O(n^2)$.

A Tabela 3.3 mostra uma comparação entre os tempos de geração dos *layouts* utilizando os algoritmos propostos, além do algoritmo original para geração de árvores NJ e uma técnica de projeção. Técnicas de projeção rápidas, tais como a LSP, que é $O(n\sqrt{n})$, levam vantagem em relação às árvores NJ com relação ao tempo de geração do *layout*. No entanto, a árvore Fast NJ se mostrou consideravelmente mais rápida do que a técnica de projeção analisada, como é possível observar. A tabela também mostra que o tempo gasto na operação de promoção de nós, apresentado na Seção 3.3.1 representa uma porção muito pequena do tempo total, não atrapalhando o processo como um todo.

A Figura 3.11 mostra os valores da medida *Neighborhood Preservation* para a coleção COREL. Como as árvores geradas pelo algoritmo original são idênticas àquelas geradas pela técnica Rapid

Tabela 3.3: Comparação entre tempos de Geração do *Layout* (segundos), considerando as abordagens de geração rápida de árvores NJ e a técnica de projeção LSP.

Técnica	COREL	MEDICAL	OBJECTS
Árvore NJ	3.0474	0.5	394.383
Árvore PNJ	3.064	0.506	396.507
Fast NJ	0.0936	0.0462	4.976
Fast PNJ	0.1104	0.052	6.1
Rapid NJ	2.366	0.373	261.971
Fast PNJ	2.3828	0.3788	262.178
LSP	0.7182	0.2068	61.564

NJ, os números representam apenas uma comparação entre Rapid NJ e Fast NJ. Também são apresentados no gráfico os valores observados para o *layout* produzido pela técnica de projeção LSP. É possível notar que as árvores Fast NJ apresentam valores de precisão ligeiramente menores do que as árvores Rapid NJ, mas ainda superiores aos mostrados pela projeção.

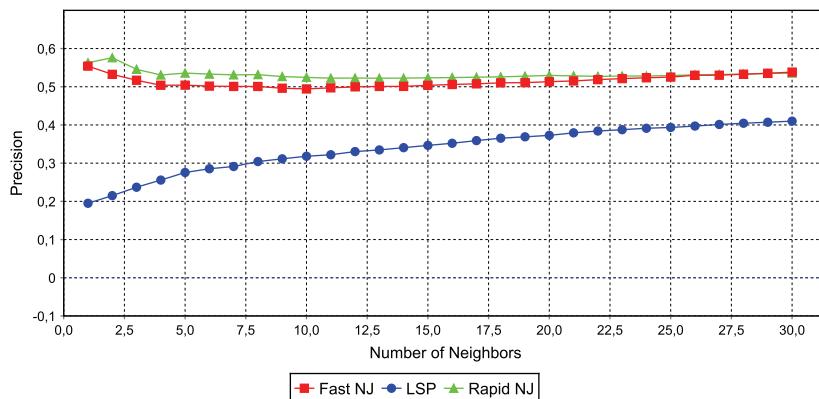


Figura 3.11: Análise da medida *Neighborhood Preservation*, comparando algoritmos de criação de Árvores NJ e Projeção LSP para a coleção COREL.

A Figura 3.12 mostra os gráficos de distância para os algoritmos Rapid NJ e Fast NJ, em comparação com a técnica de projeção LSP, para a coleção COREL, mostrando que as árvores apresentam um resultado melhor do que a projeção.

3.4 Árvores de Similaridade para Classificação Visual de Imagens

Esta seção mostra uma aplicação das árvores de similaridade em um processo iterativo de classificação visual de imagens, através de um exemplo que utiliza um subconjunto da coleção COREL, contendo 500 imagens divididas em 5 classes. Inicialmente, um subconjunto de 300

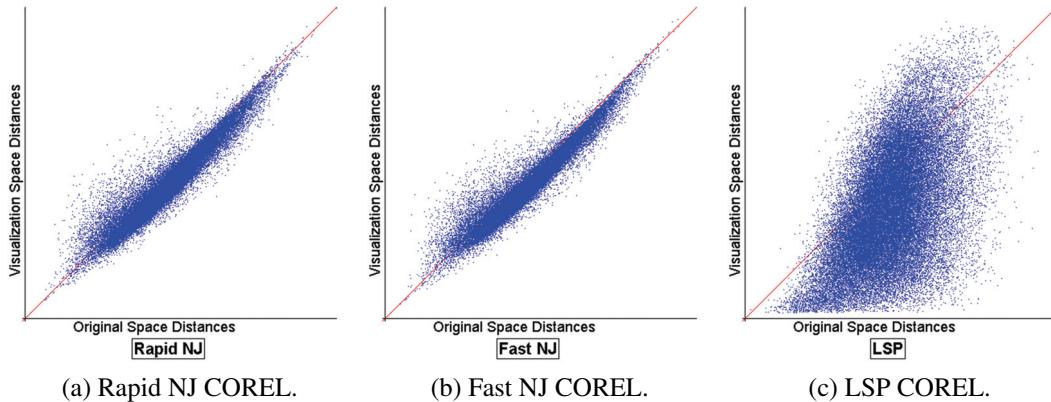


Figura 3.12: Gráfico de Distâncias comparando algoritmos de criação de Árvores NJ e Projeção LSP para a coleção COREL.

imagens é extraído das 500 existentes, para ser utilizado como conjunto de treinamento. Nesse momento o *layout* produzido pela árvore NJ pode servir como um guia para a escolha de instâncias representativas. Em cada passo, 50 novas imagens, extraídas das 200 restantes, são adicionadas ao conjunto anterior para formar um novo conjunto de teste. Em cada passo, um classificador SVM produz um resultado para esse conjunto de teste, que é então corrigido manualmente pelo usuário, e resulta em um novo conjunto de treinamento a ser utilizado no passo subsequente. A Figura 3.13 mostra as árvores NJ dos conjuntos de teste criados em cada passo, juntamente com o resultado de cada classificação.

Uma funcionalidade chamada ***Class Matching***, detalhada no Capítulo 5, foi implementada para melhorar a análise detalhada da classificação, em situações nas quais existe um esquema de rotulamento ideal para as instâncias (*ground truth*). Tal funcionalidade pode se beneficiar das características das árvores de similaridade para proporcionar um *layout* mais informativo. A Figura 3.13, terceira coluna, mostra os resultados da classificação utilizando a funcionalidade ***Class Matching***.

3.5 Considerações Finais

Esse capítulo apresentou detalhes a respeito das árvores de similaridade, em especial utilizando o método de construção *Neighbor Joining (NJ)*, da sua utilização para visualização de coleções de imagens, e de procedimentos para melhoria visual e do tempo de processamento.

As árvores de similaridade representam uma alternativa que pode amenizar as limitações apresentadas pelas técnicas de projeção. A organização das instâncias da coleção em ramos na árvore apresenta diversas vantagens com relação à exploração e compreensão dos dados, organizando a similaridade em níveis, o que representa uma abordagem natural para a interpretação de graus de similaridade. Os resultados apresentados mostram que a árvore consegue manter, no espaço de

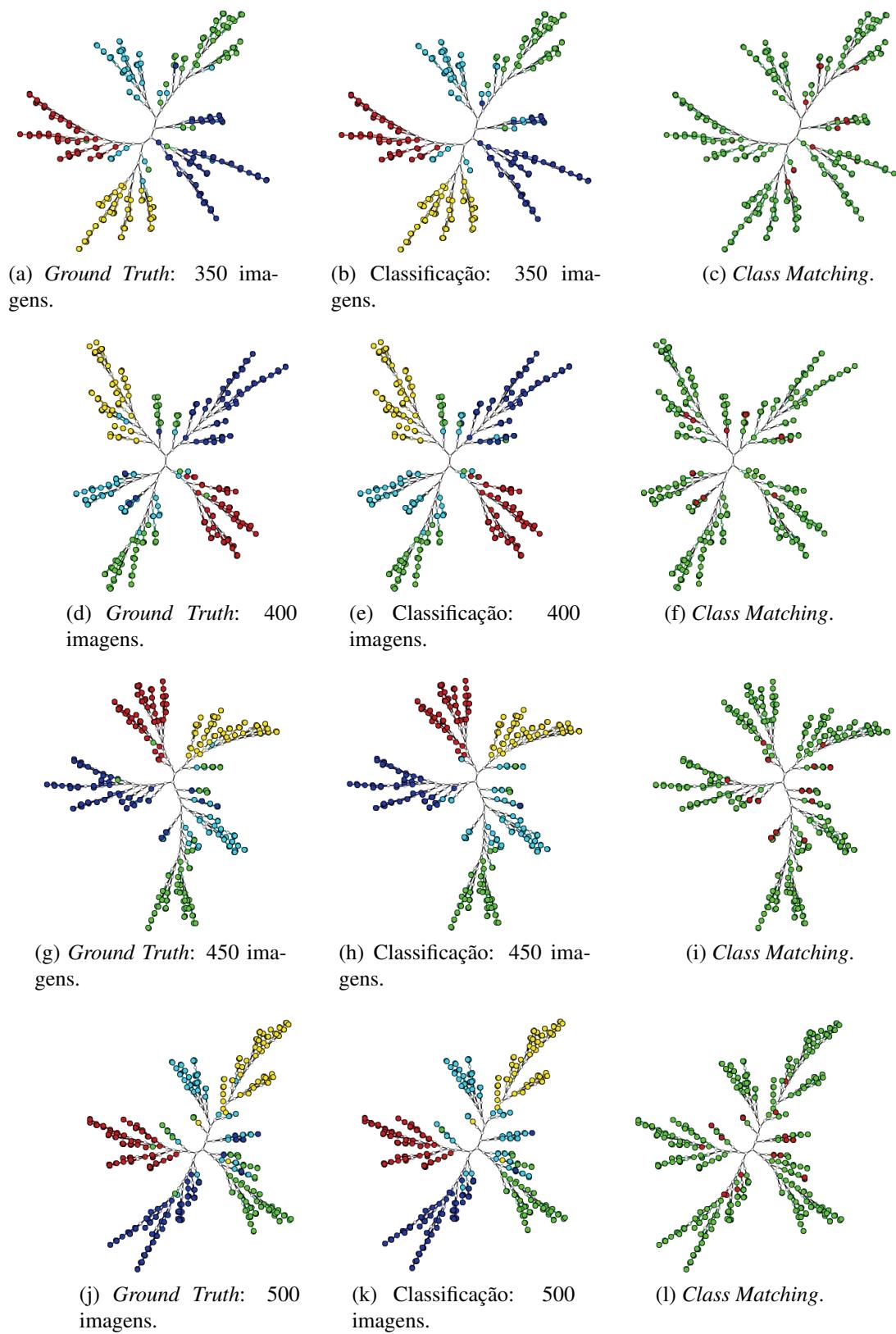


Figura 3.13: Sequência de passos do processo de classificação visual.

visualização, a maioria das vizinhanças no espaço original, e ainda diminui consideravelmente o grau de sobreposição entre as instâncias.

Entretanto, as árvores NJ apresentam limitações relacionadas ao excessivo volume de informações, causado pela grande quantidade de nós virtuais produzida pelo algoritmo. Para amenizar tais problemas foi apresentado um procedimento de promoção de nós na árvore, baseado na transformação de nós folha em nós internos, de acordo com a ocorrência de determinado padrão nos ramos. O resultado é um *layout* menos poluído, em média com 51% menos nós virtuais, que mantém as capacidades de organização e exploração das árvores NJ e garante um melhor aproveitamento do espaço de visualização.

Além disso, para reduzir o custo computacional de geração das árvores NJ, dois algoritmos foram investigados, **Rapid Neighbor Joining** e **Fast Neighbor Joining**, ambos com o objetivo de diminuir o numero de buscas necessárias para a construção dos ramos da árvore. Os resultados mostram que os algoritmos investigados permitem que a geração da árvore ocorra em um tempo consideravelmente menor.

O próximo capítulo descreve uma metodologia de redução de dimensionalidade baseada na discriminação entre as classes de uma coleção, criando espaços reduzidos que ajudem a explicar a perspectiva de categorização das instâncias.

Redução Visual de Dimensionalidade Semi-supervisionada

4.1 Considerações Iniciais

O processo de classificação de imagens se mostra suscetível a diversos fatores relacionados à representação dessas imagens. Um fator importante diz respeito ao conjunto de descritores utilizado nessa representação. Diversas pesquisas comprovam que a utilização de apenas um tipo de descritor não é suficiente para alcançar bons resultados na visualização e mineração de dados em coleções contendo esse tipo de informação. De acordo com Gehler & Nowozin (2009), nenhum descritor de características terá o mesmo poder discriminativo para todas as classes do problema. Utilizando apenas características HOG (*Histogram of Oriented Gradients*) para detecção de pessoas em imagens, Schwartz (2010) concluiu que informações como homogeneidade e textura em roupas, cores específicas de pele e texturas de fundo das imagens não foram contempladas, prejudicando o processo como um todo. A representação ideal, sugere essas pesquisas, deve utilizar uma combinação de características, baseadas em cor, forma e textura, entre outras, de forma a obter uma descrição mais completa das imagens.

No entanto, a combinação de descritores resulta na construção de representações com alta dimensionalidade, e o processo de visualização e classificação pode ser ainda prejudicado pelo alto custo computacional necessário para essas representações. Além disso, um grande número de dimensões pode gerar alta confusão visual, causada pela excessiva quantidade de informação

codificada na representação de cada imagem, dificultando a navegação e exploração por parte do usuário. Dessa forma, torna-se necessária a aplicação de um procedimento de redução de dimensionalidade, através da seleção ou transformação de atributos, com o objetivo de manter apenas as informações essenciais que captem a estrutura da coleção. Tal procedimento pode ser aplicado durante o início do processo de análise visual, antes do mapeamento das instâncias em representações visuais. Para construir *layouts* de visualização, as técnicas de redução de dimensionalidade são utilizadas para mapear o conjunto original de características em um espaço bidimensional ou tridimensional.

As técnicas de redução de dimensionalidade apresentam, entretanto, algumas limitações. Uma delas é a incapacidade de aplicação do modelo criado para uma coleção em outras coleções, exigindo o recálculo das novas reduções em cada aplicação. Outra limitação é a dificuldade em incluir o conhecimento do usuário no processo, de forma que ele não pode interferir nos resultados obtidos, nem alterar a perspectiva de visão da coleção.

Com o intuito de contornar os problemas citados, este capítulo apresenta uma metodologia de análise visual de imagens que utiliza a técnica de regressão **Partial Least Squares** (PLS), proposta por Wold (1985), para a redução de dimensionalidade de uma coleção. Técnicas de visualização são utilizadas para guiar o usuário na composição de um conjunto de amostras que melhor represente as diferenças entre as classes, produzindo um espaço reduzido de características que destaque tais diferenças. A redução de dimensionalidade proposta pode ser aplicada tanto em coleções previamente rotuladas quanto em coleções sem nenhum rótulo prévio. Diversos resultados da aplicação da metodologia em coleções de imagens e texto são apresentados e discutidos. Finalmente, são apresentados alguns resultados da associação entre PLS e técnicas de visualização para a classificação de coleções de imagens, mostrando o potencial dos *layouts* criados para guiar a escolha de instâncias de treinamento, bem como as capacidades de discriminação entre classes apresentadas pela técnica PLS.

Este capítulo é uma síntese de um artigo publicado no *Computer Graphics Forum* (CGF). O conteúdo completo do artigo pode ser encontrado no Apêndice C.

4.2 **Partial Least Squares (PLS)**

Partial Least Squares representa uma classe de métodos estatísticos criados por Wold (1985), utilizados para modelar relações entre conjuntos de variáveis multidimensionais através de um espaço latente de baixa dimensionalidade. Esse espaço tem como objetivo maximizar a separação entre instâncias com características diferentes entre si, de forma que instâncias de uma mesma classe formem agrupamentos mais consistentes. A ideia na qual se baseia o PLS é a de que dados observados são gerados por um sistema guiado por um pequeno número de variáveis latentes.

Dessa forma, a técnica tenta reduzir as dimensões de uma coleção de forma a estimar os coeficientes de regressão de cada instância nessas variáveis latentes.

PLS tem sido explorado em diversas áreas do conhecimento, como Quimiometria (Broadhurst et al., 1997) e (Lindgren et al., 1995), Bioinformática (Nguyen & Rocke, 2002) e (Boulesteix & Strimmer, 2007) e Neurociência (Nestor et al., 2002). Recentemente, é possível encontrar aplicações em Visão Computacional, em problemas de redução de dimensionalidade, regressão e classificação. Exemplos de utilização do PLS para redução de dimensionalidade são apresentados por Schwartz (2010) e Kembhavi et al. (2011), para identificação de faces e de veículos, respectivamente.

Esta seção descreve as principais características da PLS, bem como seu funcionamento para redução de dimensionalidade e classificação.

4.2.1 Descrição da Técnica

PLS é uma técnica que consegue lidar com problemas contendo dados de alta dimensionalidade e poucos exemplos (Geladi, 1988). Considerando uma matriz X , contendo o conjunto de instâncias representadas por suas variáveis originais, e uma matriz Y , contendo o conjunto de variáveis de resposta (para problemas univariados, um vetor Y é considerado), PLS estima um conjunto de variáveis latentes que representam uma combinação linear das matrizes X e Y , e descrevem a estrutura comum do relacionamento entre essas duas matrizes. Essas variáveis latentes representam um conjunto de fatores ortogonais com a restrição de possuírem o melhor poder preditivo de Y . Em outras palavras, PLS encontra os componentes de X que sejam úteis para prever Y , resultando em um conjunto que requer menos informação do que X .

A seguir uma breve descrição matemática é apresentada. Mais detalhes podem ser encontrados em (Rosipal & Kramer, 2006) e (Garthwaite, 1994).

Seja uma situação na qual n amostras descritas por d variáveis (características) são armazenadas em uma matriz $\mathbf{X}_{n \times d}$ centrada na média, e k variáveis de resposta correspondentes são armazenadas em uma matriz $\mathbf{Y}_{n \times k}$, também centrada na média. Para realizar a redução para p variáveis, ($p \ll d$), PLS realiza uma decomposição de X e Y de acordo com a Equação 4.1:

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^T + \mathbf{E}, \\ \mathbf{Y} &= \mathbf{U}\mathbf{Q}^T + \mathbf{F}, \end{aligned} \tag{4.1}$$

na qual $\mathbf{T}_{n \times p}$ e $\mathbf{U}_{n \times p}$ são matrizes contendo as variáveis latentes, $\mathbf{P}_{d \times p}$ e $\mathbf{Q}_{k \times p}$ representam matrizes de carga, e $\mathbf{E}_{n \times d}$ e $\mathbf{F}_{n \times k}$ representam matrizes com resíduos.

Uma maneira de realizar a decomposição mostrada na Equação 4.1 é através do algoritmo ***Nonlinear Iterative Partial Least Squares (NIPALS)*** (Wold, 1985), através da estimativa de um

conjunto de vetores de projeção \mathbf{w}_i , com $i = 1, 2, \dots, p$, armazenados em uma matriz $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_p)$, de forma que:

$$[cov(\mathbf{t}_i, \mathbf{u}_i)]^2 = \max_{|\mathbf{w}_i|=|\mathbf{u}_i|=1} [cov(\mathbf{X}\mathbf{w}_i, \mathbf{Y}\mathbf{u}_i)]^2, \quad (4.2)$$

onde $|\mathbf{w}_i|$ e $|\mathbf{u}_i|$ representam as normas euclidianas dos vetores \mathbf{w}_i e \mathbf{u}_i , respectivamente. \mathbf{t}_i e \mathbf{u}_i representam as i -ésimas colunas das matrizes \mathbf{T} e \mathbf{U} , e $cov(\mathbf{t}_i, \mathbf{u}_i)$ representa a covariância entre os vetores latentes \mathbf{t}_i e \mathbf{u}_i .

As variáveis latentes são extraídas de forma iterativa, e representam a relação entre \mathbf{Y} e cada variável \mathbf{X}_j , com $j = 1, 2, \dots, d$. Inicialmente, \mathbf{T}_1 representa a variável latente que será útil para determinar \mathbf{U}_1 . Para obter \mathbf{T}_1 , é aplicada uma regressão de \mathbf{U}_1 em cada \mathbf{X}_j , de acordo com a Equação 4.3, com $\mathbf{b}_{1j} = \mathbf{X}_{1j}^T \mathbf{Y}_1 / \mathbf{X}_{1j}^T \mathbf{X}_{1j}$.

$$\mathbf{U}_{1j} = \mathbf{b}_{1j} \mathbf{X}_{1j} \quad (4.3)$$

A variável \mathbf{T}_1 será então determinada pela média ponderada dos valores de \mathbf{U}_1 , de acordo com a Equação 4.4.

$$\mathbf{T}_1 = \sum_{j=1}^d \mathbf{w}_{1j} \mathbf{U}_{1j} \quad (4.4)$$

considerando que $\sum_{j=1}^d \mathbf{w}_{1j} = 1$.

Cada variável latente \mathbf{T}_k , com $k = 1, 2, \dots, p$, será definida pelos resíduos das regressões das variáveis latentes anteriores, de acordo com as equações 4.5, 4.6, 4.7 e 4.8.

$$\mathbf{X}_{kj} = \mathbf{X}_{(k-1)j} - [\mathbf{T}_{(k-1)}^T \mathbf{X}_{(k-1)j} / \mathbf{T}_{(k-1)}^T \mathbf{T}_{(k-1)}] \mathbf{T}_{(k-1)} \quad (4.5)$$

$$\mathbf{U}_k = \mathbf{U}_{(k-1)} - [\mathbf{T}_{(k-1)}^T \mathbf{U}_{(k-1)} / \mathbf{T}_{(k-1)}^T \mathbf{T}_{(k-1)}] \mathbf{T}_{(k-1)} \quad (4.6)$$

$$\mathbf{T}_k = \sum_{j=1}^d \mathbf{w}_{kj} \mathbf{U}_{kj} \quad (4.7)$$

$$\mathbf{b}_{kj} = \mathbf{X}_{kj}^T \mathbf{U}_k / \mathbf{X}_{kj}^T \mathbf{X}_{kj} \quad (4.8)$$

A redução de dimensionalidade é realizada através da projeção do vetor contendo os valores das variáveis de uma instância no conjunto de vetores de projeção, obtendo as novas dimensões que representam os coeficientes de regressão nas variáveis latentes encontradas.

Diferente da técnica PCA, que cria vetores de projeção ortogonais maximizando a variância entre as instâncias representadas na matriz X , PLS utiliza também os rótulos de classes dessas instâncias, armazenados na matriz Y para realizar a redução de dimensionalidade, o que o caracteriza como um processo supervisionado. Essa característica justifica o fato de que as variáveis latentes geradas pelo método possuem como foco a discriminação dos dados. O espaço reduzido tende a apresentar uma melhor separação entre as instâncias em suas classes, auxiliando o processo de análise visual das coleções.

4.2.2 Abordagens de Utilização

De acordo com o funcionamento da técnica PLS apresentado na Seção 4.2.1, o conjunto de p variáveis latentes é obtido com base em um conjunto de n amostras, pertencentes a C classes, juntamente com um número p de fatores. Duas abordagens, chamadas ***One Against All*** (Schwartz et al., 2012), e ***MulticlassMatrix*** (Rosipal & Kramer, 2006) foram desenvolvidas para realizar essa tarefa.

A abordagem *One Against All* procura destacar a influência das características de cada classe nas instâncias da coleção, e como as características de cada instância as direcionam para determinada classe. Dessa forma, são construídos C modelos PLS, um para cada classe. Cada modelo utiliza p fatores e produz resposta simples. Na construção do i -ésimo modelo, a matriz Y torna-se um vetor contendo n indicadores de classe, uma para cada amostra. As posições desse vetor correspondentes às amostras da classe i recebem o indicador +1, e as posições correspondentes às amostras das outras classes recebem o indicador -1.

Assim, a redução de dimensionalidade é realizada através da projeção das instâncias em cada um desses modelos, resultando em um conjunto de C dimensões, cada uma contendo o valor da regressão retornado por cada modelo. Esse valor de regressão representa a combinação linear de cada um dos p fatores nos quais o modelo foi construído. Na classificação, quando uma instância de teste é submetida aos modelos, os valores de regressão nas C dimensões são analisados, e a classe com valor de regressão mais alto é atribuída para a instância.

Já a abordagem *MulticlassMatrix* cria um modelo PLS único de múltiplas respostas, utilizando uma matriz $\mathbf{Y}_{n \times C}$, de forma que $Y_{i,j} = 1$ caso a i -ésima amostra pertença à j -ésima classe, e 0 caso contrário.

Assim, a redução de dimensionalidade é realizada através da projeção das instâncias nesse modelo, resultando em um conjunto de p dimensões representando os coeficientes de regressão em cada uma das variáveis latentes do modelo. Na classificação, quando uma instância de teste é submetida ao modelo, os valores de regressão em cada uma das C classes são analisados, e a classe com valor de regressão mais alto é atribuída para a instância.

Portanto, a abordagem *MulticlassMatrix* produz um espaço no qual o número de dimensões é equivalente ao número de variáveis latentes, enquanto que a abordagem *One Against All* produz um espaço constituído pelos valores de regressão em cada classe, com o número de dimensões equivalente ao número de classes. Como será mostrado na seção 4.4, a abordagem *MulticlassMatrix* produz um espaço latente em menos tempo do que a abordagem *One Against All*. No entanto, o espaço produzido pela abordagem *One Against All* se mostra mais preciso em termos de descrição do espaço de características.

4.3 Metodologia de Redução Visual de Dimensionalidade PLS

O resultado do processo de redução de dimensionalidade, de acordo com as abordagens apresentadas na seção 4.2.2, será um conjunto de instâncias com c dimensões, com c representando o número de classes (*One Against All*) ou um conjunto com p dimensões, com p representando o número de fatores (*MulticlassMatrix*). Em qualquer um dos casos, para visualizar as tendências e características da coleção, qualquer uma das técnicas de posicionamento de pontos apresentadas nos Capítulos 2 e 3 poderão ser utilizadas. Além disso, é possível analisar a distribuição dos dados nas dimensões reduzidas, através da utilização de técnicas de visualização multi-eixos disponíveis, tais como **RadViz** (Hoffman et al., 1999) ou **Coordenadas Paralelas** (Inselberg & Dimsdale, 1990).

Como apresentado na Seção 4.2.1, o método PLS é uma técnica supervisionada, e necessita portanto de um conjunto de treinamento para criar um modelo de redução de dimensionalidade. Este trabalho, no entanto, apresenta também uma metodologia para lidar com coleções não rotuladas. Ambas as metodologias são apresentadas a seguir.

Metodologia 1: Coleções com conjuntos de treinamento rotulados:

Os passos para a redução de dimensionalidade utilizando os rótulos das amostras são:

1. Criação do Modelo: um modelo PLS é construído a partir de um conjunto de treinamento rotulado, informado pelo usuário. Qualquer uma das abordagens PLS, *One Against All* ou *MultiClassMatrix* pode ser utilizada;
2. Aplicação do Modelo: o modelo é aplicado na coleção de teste. O resultado é um conjunto de instâncias com p ou c dimensões, dependendo da abordagem PLS escolhida;
3. Mapeamento Visual: A coleção com dimensões reduzidas é mapeada em um plano de visualização, utilizando uma estratégia de posicionamento de pontos.

A Figura 4.1 sintetiza o processo apresentado na **Metodologia 1**.

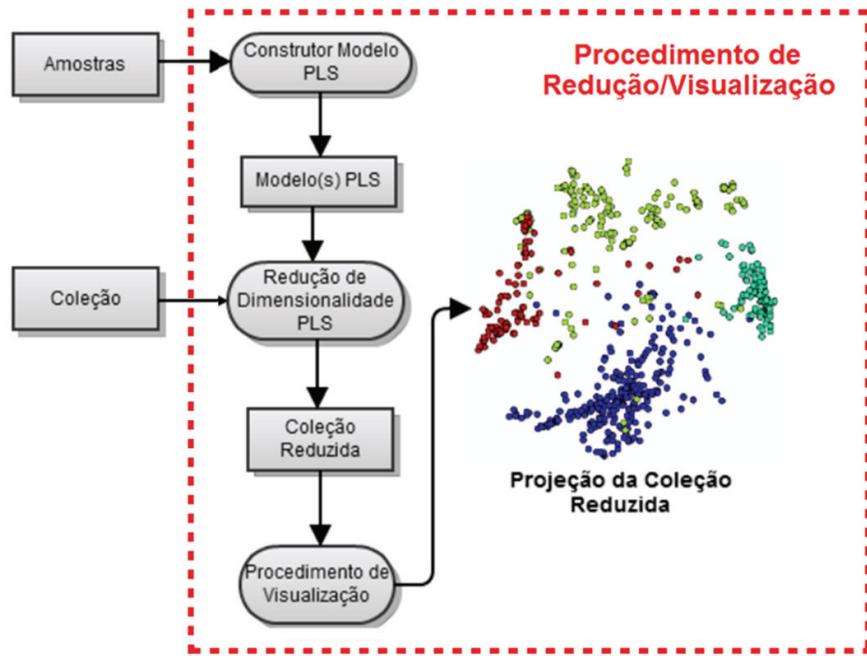


Figura 4.1: Esquema de redução de dimensionalidade aplicada a coleções rotuladas.

Metodologia 2: Coleções não rotuladas:

Os passos para a redução de dimensionalidade em coleções sem rotulamento prévio são:

1. Criação de Rótulos: a coleção é submetida a um procedimento de agrupamento, e cada instância recebe um rótulo correspondente ao grupo ao qual pertence. O resultado é uma coleção rotulada por agrupamento;
2. Criação do Modelo: processo idêntico ao apresentado na **Metodologia 1**, com a diferença que as classes são substituídas pelos rótulos de grupo, e o conjunto de treinamento é amostrado da coleção, de acordo com alguma técnica de amostragem;
3. Aplicação do Modelo: processo também idêntico ao apresentado na **Metodologia 1**;
4. Mapeamento Visual: A coleção com dimensões reduzidas é mapeada em um plano de visualização, utilizando uma estratégia de posicionamento de pontos.

A Figura 4.2 sintetiza o processo apresentado na **Metodologia 2**.

Em ambas as abordagens, o conjunto de treinamento utilizado pelo PLS pode ser construído amostrando uma coleção rotulada, seja pelo usuário, seja pela técnica de agrupamento. Neste trabalho, duas abordagens de amostragem foram utilizadas. Na primeira abordagem, a coleção é submetida a um procedimento de agrupamento, e um número x de instâncias escolhido de cada grupo formado. Metade dessas x instâncias representa aquelas mais próximas do centróide do

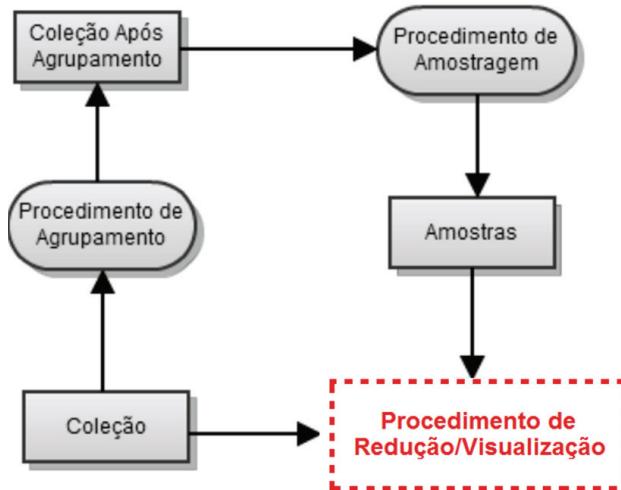


Figura 4.2: Esquema de redução de dimensionalidade aplicada a coleções não rotuladas.

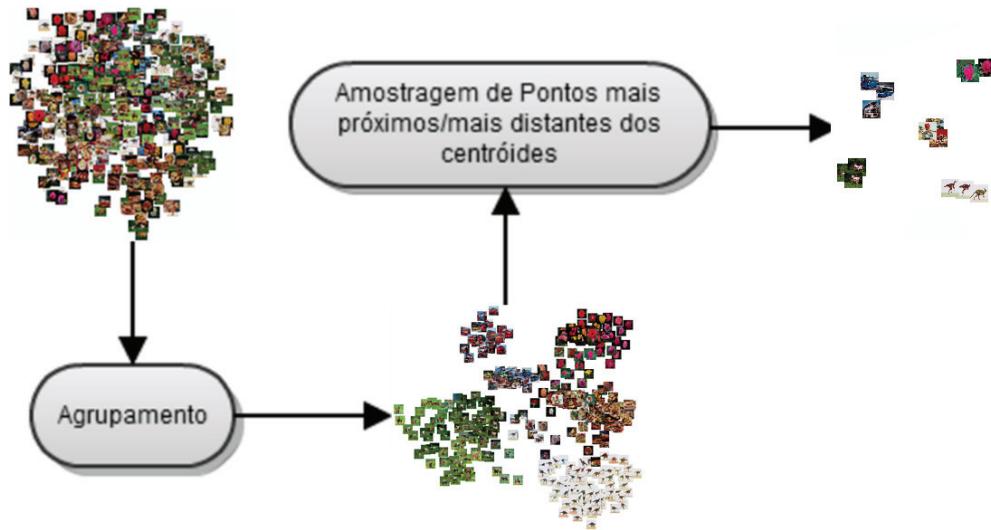


Figura 4.3: Amostragem do conjunto de treinamento a ser utilizado pelo PLS, através de um procedimento de agrupamento.

grupo, enquanto que a outra metade representa aquelas mais distantes desse centróide. Esse processo é ilustrado na Figura 4.3.

A segunda maneira de amostragem constrói o conjunto de treinamento utilizando uma abordagem semi-supervisionada. A coleção é inicialmente submetida a uma técnica de visualização, tal como aquelas apresentadas no Capítulo 2, resultando em um *layout* de onde o usuário escolherá manualmente as instâncias. O *layout* serve assim como um guia que facilita a identificação daquele as instâncias que são representativas em cada classe (coleções rotuladas) ou grupos (coleções não rotuladas), potencializando a criação de conjuntos de treinamento que produzam bons modelos PLS, em termos de segregação das coleções. Esse processo é ilustrado na Figura 4.4.

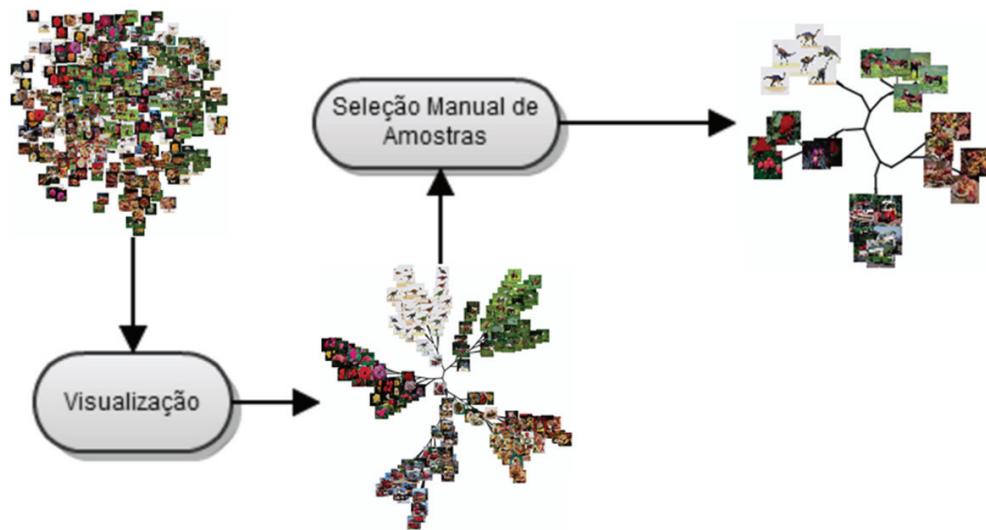


Figura 4.4: Amostragem do conjunto de treinamento a ser utilizado pelo PLS, através da seleção manual de instâncias em um *layout*.

É importante ressaltar que as ferramentas visuais para rotulamento de imagens apresentadas no Capítulo 5 podem ser utilizadas para a criação de conjuntos de treinamento rotulados a partir de conjuntos não rotulados. Isso pode ser feito colorindo pontos ou grupos de pontos no *layout*, definindo as classes a serem utilizadas no processo.

As metodologias descritas podem ser executadas iterativamente até que um conjunto de treinamento satisfatório seja obtido, e consequentemente um modelo de redução de dimensionalidade PLS ideal seja construído. Esse modelo pode então ser utilizado em qualquer coleção que compartilhe o mesmo conjunto de características.

A seção 4.4 mostra e discute os resultados da redução de dimensionalidade baseada em PLS em algumas coleções de dados, utilizando as metodologias descritas nesta seção.

4.4 Análise de Resultados

As funcionalidades relacionadas à metodologia de redução de dimensionalidade PLS aqui apresentadas foram inseridas no sistema VisPipeline. Os experimentos relacionados foram conduzidos utilizando as coleções de imagens COREL e ETHZ, e as coleções de documentos textuais NEWS e ALL. O espaço de características das coleções textuais contém a frequência dos termos de cada documento, após a remoção de *stopwords* e execução de um procedimento de *stemming*, seguindo a técnica *term-frequency-inverse-document-frequency*. A Tabela 4.1 summariza as informações a respeito de cada uma dessas coleções.

Nos experimentos, foram utilizados 10 fatores para a coleção textual. Para as coleções de imagens, foram utilizados 8 fatores para a abordagem *One Against All*, e 5 fatores para a abordagem

Tabela 4.1: Descrição das coleções utilizadas nos experimentos de redução de dimensionalidade PLS.

Coleção	Natureza	Instâncias	Classes	Dimensões
COREL	Imagens diversas	1000	10	150
ETHZ	Imagens de pessoas	2019	28	3963
NEWS	Feeds de notícias	1771	23	3731
ALL	Resumos de artigos científicos	2814	9	5163

MultiClassMatrix. Esses valores foram escolhidos após a execução de um procedimento de validação cruzada, utilizando 5 partições. A medida de distância utilizada nas coleções de imagem foi a Euclidiana, e a utilizada em coleções textuais foi a do cosseno.

A Figura 4.5 mostra a visualização do resultado do processo de redução de dimensionalidade para a coleção NEWS. Em particular, a Figura 4.5a mostra o *layout RadViz* do espaço reduzido pela abordagem *MultiClassMatrix* considerando 10 fatores. Já a Figura 4.5b mostra o *layout RadViz* do espaço reduzido pela abordagem *One Against All*, com 23 dimensões correspondentes às classes do problema. Cada eixo contém um identificador para a dimensão no espaço reduzido a qual representa. Para a abordagem *One Against All*, as cores dos eixos representam as cores das classes do problema. Já na abordagem *MultiClassMatrix*, as cores dos eixos não tem relação com as cores das classes. É possível analisar, utilizando RadViz, quais dimensões estão mais envolvidas em determinar a segregação em pelo menos uma dessas classes. É importante ressaltar que a técnica RadViz é utilizada aqui apenas para esse tipo de análise, não sendo empregada na exploração visual da coleção, devido às suas limitações, apresentadas no Capítulo 2.

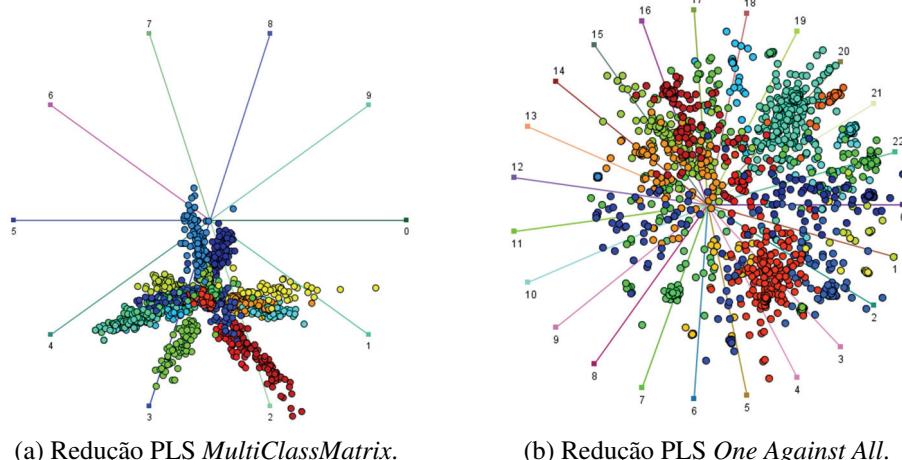


Figura 4.5: Exemplos de visualizações RadViz do espaço reduzido obtido para a coleção NEWS, usando a abordagem *MultiClassMatrix* em 10 fatores (a) e abordagem *One Against All* em 23 fatores (b).

O *layout* produzido pela técnica de visualização RadViz é totalmente dependente da ordem de exibição das âncoras. O usuário pode alterar essa ordem, de forma a explorar as diferentes correlações entre as dimensões. A Figura 4.6 mostra uma outra ordenação de eixos para o *layout* mostrado na Figura 4.5a.

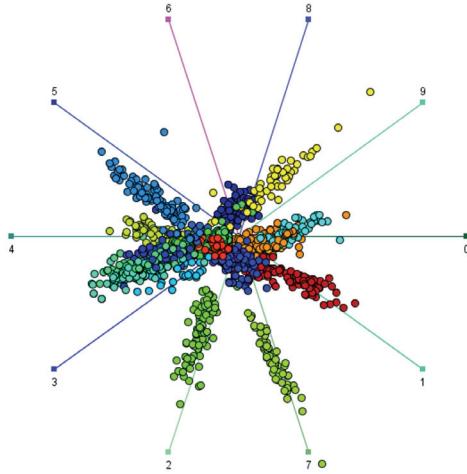


Figura 4.6: *Layout* RadViz alternativo ao mostrado na Figura 4.5a, para a coleção NEWS, produzido pela alteração da ordem dos eixos.

Os experimentos desenvolvidos neste trabalho representam estudos de caso abrangendo três situações: utilização de conjuntos de treinamentos criados pelo usuário, utilização de conjuntos de treinamento produzidos por procedimentos de amostragem de uma coleção previamente rotulada, e aplicação da **Metodologia 2**, apresentada na Seção 4.3, aplicada a coleções não rotuladas. Além disso, foram feitas comparações das metodologias propostas com outras técnicas de redução de dimensionalidade, bem como testes com diversas técnicas de mapeamento visual dos espaços reduzidos. Mais detalhes podem ser encontrados em (Paiva et al., 2012), presente no Anexo B.

Para avaliar numericamente os resultados produzidos em termos de discriminabilidade entre as instâncias, foi utilizado o **coeficiente de silhueta** (Tan et al., 2005). Esse coeficiente mede a **coesão** e **separação** entre grupos de instâncias. Dada uma instância p_i , sua coesão a_i representa a distância média entre p_i e todas as outras instâncias que pertencem ao mesmo grupo que p_i . Já sua separação b_i representa a distância mínima entre p_i e todas as outras instâncias pertencentes a outros grupos. Levando em conta esses valores, a silhueta de uma instância é calculada de acordo com a Equação 4.9, e o coeficiente de silhueta de uma coleção é dado como a média dos coeficientes de silhueta de todas as n instâncias. Apesar de o coeficiente de silhueta, em várias situações, não ser a medida mais apropriada para refletir os agrupamentos nos *layouts* de projeção dos espaços reduzidos, devido à sua inadequação em medir grupos não esféricos, ela é utilizada aqui para avaliar a separação entre os grupos nos espaços reduzidos produzidos pela metodologia proposta, comparando com a separação no espaço original, bem como para avaliar a separação de grupos após o mapeamento no plano de visualização.

$$S = \frac{1}{n} \sum_{i=1}^n \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad (4.9)$$

O coeficiente de silhueta varia entre -1 e 1, sendo que 1 indica que os grupos estão perfeitamente separados uns dos outros.

Coleções Previamente Rotuladas

Nesta seção são apresentados os resultados do processo de redução de dimensionalidade utilizando PLS, considerando a utilização de conjuntos de treinamento criados pelo usuário e através de amostragem em coleções previamente rotuladas. Em ambos os casos, os modelos produzidos são aplicados a uma coleção na qual os rótulos são desconsiderados. Esses rótulos são posteriormente utilizados apenas para verificar a precisão dos resultados.

Os conjuntos de treinamento amostrados são obtidos utilizando a técnica de agrupamento *k-means*, e são identificados pelo sufixo '*k-means*', com *k* substituído pelo número de grupos utilizado.

Na Tabela 4.2, são mostrados os resultados da aplicação do processo de redução de dimensionalidade PLS na coleção NEWS. A primeira linha mostra o valor do coeficiente de silhueta para a coleção original, seguidas dos coeficientes de silhueta da mesma coleção após um mapeamento utilizando a técnica de projeção LSP e a árvore NJ, respectivamente. O restante das linhas mostra os valores dos coeficientes de silhueta dos espaços reduzidos, utilizando conjuntos de treinamento criados pelo usuário e por amostragem utilizando técnicas de agrupamento. Também são mostrados os valores dos coeficientes de silhueta da coleção após a aplicação da técnica RadViz, utilizando uma ordenação de eixos com alta separação entre grupos. É possível verificar que a separabilidade dos grupos é sensivelmente maior após a redução de dimensionalidade, e que quanto maior o tamanho do conjunto de treinamento, maior é o valor do coeficiente de silhueta. Em alguns casos a técnica RadViz consegue obter uma separabilidade maior do que a observada no conjunto de dimensões originais.

Os resultados da mesma análise para a coleção ETHZ são mostrados na Tabela 4.3. Os resultados mostrados aqui são semelhantes aos obtidos para a coleção NEWS, mas um grau de separabilidade ainda maior pode ser notado, também proporcional ao tamanho do conjunto de treinamento. Para essa coleção, a técnica RadViz não consegue uma boa separação entre classes, provavelmente devido ao maior número de instâncias e classes.

A Figura 4.7 mostra a precisão dos mapeamentos da coleção NEWS, utilizando árvore NJ e diversas técnicas de projeção, após a redução de dimensionalidade utilizando PLS. Nesse processo foi utilizado um conjunto de treinamento contendo 863 instâncias escolhidas pelo usuário. Foi utilizada a abordagem PLS *One Against All*, e o espaço reduzido possui 23 dimensões. A medida

Tabela 4.2: Redução de dimensionalidade, utilizando conjunto de treinamento previamente rotulado, para a coleção NEWS.

Tamanho do Conjunto de Treinamento	Método de Amostragem	Silhueta Reduzido	Técnica de Mapeamento	Silhueta
1771	original	0.1374	LSP Árvore NJ	0.0934 0.0949
611	usuário	0.4052		0.0612
863	usuário	0.6815		0.1755
1169	usuário	0.7780	Radviz	0.4354
600	23-means	0.6187		0.1035
800	23-means	0.5132		0.1909
800	23-means MultiClass	0.2799		0.0537

Tabela 4.3: Redução de Dimensionalidade, utilizando conjunto de treinamento previamente rotulado, para a coleção ETHZ.

Tamanho do Conjunto de Treinamento	Método de Amostragem	Silhueta Reduzido	Técnica de Mapeamento	Silhueta
2019	original	0.0912	LSP Árvore NJ	-0.0390 0.1023
200	usuário	0.3622		-0.1317
600	usuário	0.5457		-0.0433
1000	usuário	0.6277	Radviz	-0.0555
200	28-means	0.4024		-0.0777
600	28-means	0.5326		-0.0648
1000	28-means	0.5915		-0.0525

utilizada foi a *Neighborhood Hit* (Paulovich et al., 2008), que mede a porcentagem de vizinhos mais próximos de uma instância, no espaço de visualização, que pertencem à mesma classe desse ponto. Nesse caso, a precisão final representa a média das precisões para cada ponto. Essa medida permite avaliar numericamente a separabilidade das classes pré-existentes no *layout*. Os resultados mostram que a árvore NJ é a que melhor reflete as vizinhanças no espaço reduzido. As técnicas de projeção ISOMAP e LSP também apresentaram bons resultados. Já a técnica RadViz não conseguiu separar bem as classes da coleção. Isso ocorre porque essa técnica tende a posicionar próximas instâncias com valores balanceados nas dimensões, ao invés de instâncias com valores de atributos similares.

A segregação entre classes apresentada pelas técnicas de mapeamento para a coleção NEWS pode ser visualmente verificada na Figura 4.8, que mostra os *layouts* LSP (4.8a), ISOMAP (4.8b), RadViz (4.8c) e da árvore NJ (4.8d). Utilizando LSP e ISOMAP, os grupos formados são mais densos. Já utilizando RadViz, os grupos se mostram mais espalhados no plano de visualização. O

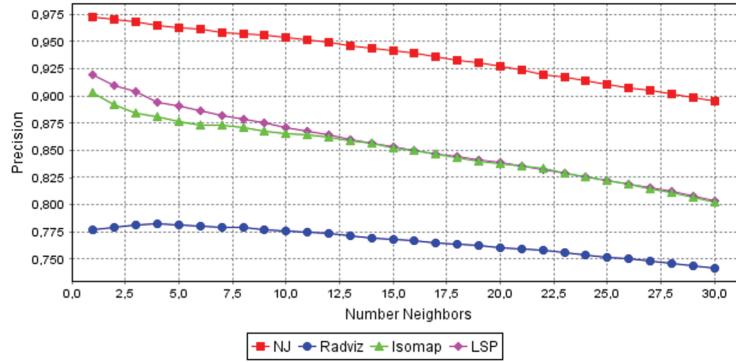


Figura 4.7: Valores *Neighborhood Hit* para *layouts* produzidos pela árvore NJ e por diversas técnicas de projeção, aplicadas no conjunto NEWS com dimensões reduzidas, utilizando um conjunto de treinamento de 863 instâncias e abordagem *One Against All*.

layout produzido pela árvore NJ comprova os números mostrados na Figura 4.7, apresentando as classes bem organizadas nos ramos.

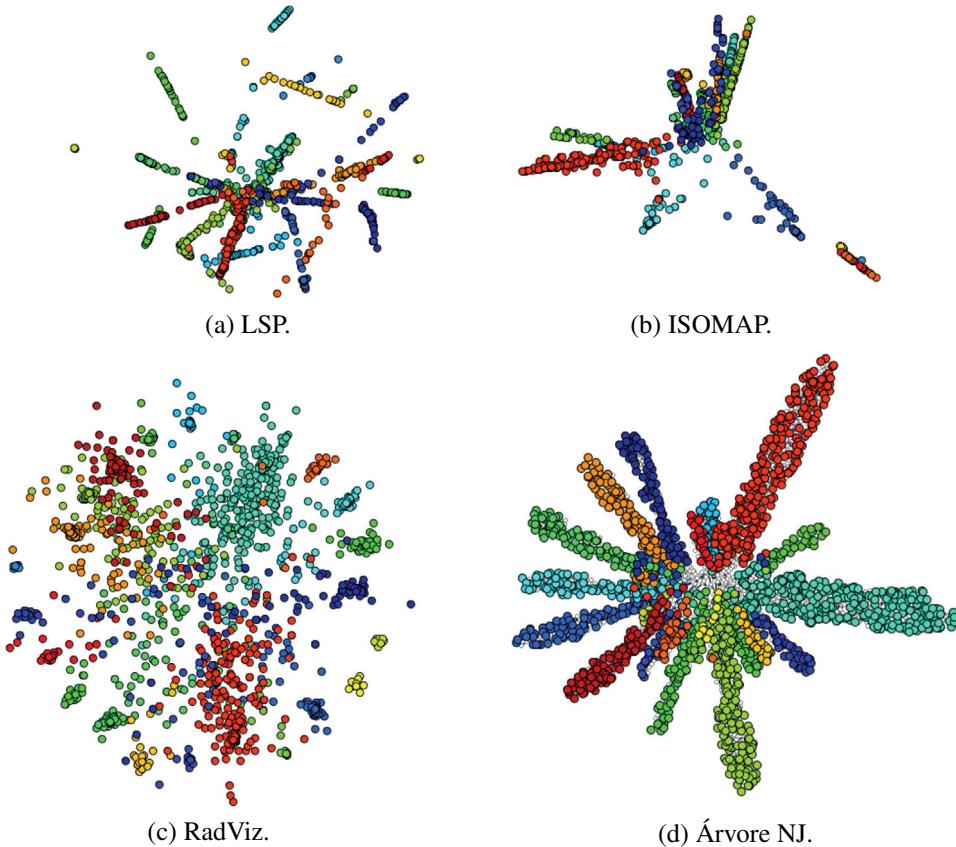


Figura 4.8: *Layouts* produzidos por diversas técnicas de projeção e pela árvore NJ, aplicadas no conjunto NEWS com dimensões reduzidas, utilizando um conjunto de treinamento de 863 instâncias.

A Figura 4.9 mostra, a título de comparação, a árvore NJ construída para a coleção NEWS, utilizando as dimensões originais. É possível verificar que diversas instâncias estão incorretamente

posicionadas, em ramos cuja maioria das instâncias pertencem a uma classe diferente. Esse posicionamento incorreto pode ter sido causado pela alta dimensionalidade da coleção, que influenciou no cálculo da similaridade entre as instâncias. As dimensões produzidas pela metodologia proposta destacam justamente a separabilidade entre as classes do problema, o que resulta na árvore mostrada na Figura 4.8d, na qual a maioria das instâncias foi corretamente colocada em ramos que correspondem às classes do problema.

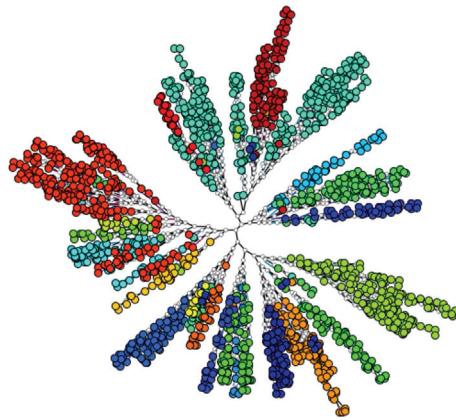


Figura 4.9: Árvore NJ aplicada ao conjunto NEWS, considerando as dimensões originais da coleção.

Coleções Não Rotuladas

Como foi dito anteriormente, apesar do PLS ser uma técnica supervisionada, que necessita de um conjunto de treinamento previamente rotulado, foi desenvolvido neste trabalho uma metodologia para lidar com coleções não rotuladas, apresentada na Seção 4.3. Dessa forma, as coleções NEWS e ETHZ foram submetidas a um procedimento de agrupamento, e os rótulos de cada grupo formado foram utilizados na construção do conjunto de treinamento. A ideia defendida aqui é a de que não é necessário a existência de rótulos na coleção para obter um modelo PLS que melhore a separabilidade do espaço original, mas apenas utilizar amostras que consigam distinguir instâncias de classes adequadamente.

Para agrupar as coleções não rotuladas, duas abordagens foram utilizadas. A primeira delas foi submeter as coleções a um procedimento de agrupamento automático chamado *Bisecting K-means* (bkmeans) (Steinbach et al., 2000), utilizando 23 ou 40 grupos para a coleção NEWS e 20 ou 28 grupos para a coleção ETHZ. Essa variação do método *K-means* foi escolhida por ser mais consistente na produção dos grupos, e menos suscetível à escolha dos centros. Em cada grupo, amostras foram automaticamente escolhidas, considerando aquelas mais próximas e as mais distantes dos centróides de cada grupo. A segunda abordagem utilizou um procedimento de rotulamento manual, realizado pelo usuário, utilizando as funcionalidades apresentadas no Capítulo 5. Foram

utilizadas árvores NJ nesse processo, tais como a apresentada na Figura 4.9, porém sem os rótulos de classe. Nesse procedimento, foram definidas 24 classes para a coleção NEWS e 12 classes para a coleção ETHZ.

Os resultados da redução de dimensionalidade baseada em PLS utilizando conjuntos de treinamento originalmente não rotulados são mostrados na Tabela 4.4. Dois valores de coeficiente de silhueta, para cada redução, são considerados. O primeiro deles, **Silhueta Rótulo Agrupamento**, considera os rótulos obtidos do procedimento de agrupamento, e o segundo, **Silhueta Rótulo Original** considera os rótulos originais de cada instância. É possível verificar que os resultados são satisfatórios para todas as coleções analisadas. Para a coleção NEWS os valores de coeficiente de silhueta, considerando os rótulos originais, variam entre 0.07 e 0.31, para uma coleção cuja silhueta é originalmente 0.1374. Além disso, utilizando a abordagem *One Against All* (o-a-a), os valores são mais precisos do que os obtidos utilizando a abordagem *MulticlassMatrix* (multi). Para a coleção ETHZ, os valores de coeficiente de silhueta, considerando os rótulos originais, variam entre -0.08 e 0.12, e os melhores valores são observados no método de amostragem baseado na árvore NJ. Considerando o esquema original de rótulos, observa-se que os valores de coeficiente de silhueta são piores utilizando a abordagem *One Against All* (o-a-a), para a coleção ETHZ.

Tabela 4.4: Redução de Dimensionalidade, utilizando conjuntos de treinamento originalmente não rotulados, para as coleções NEWS e ETHZ.

Coleção	Conjunto de Treinamento	Método de Amostragem	Abordagem de Redução	Silhueta Rótulo Agrupamento	Silhueta Rótulo Original
NEWS	original	–	–	–	0.1374
NEWS	800	bkmmeans-23	multi	0.1718	0.1669
NEWS	800	bkmmeans-23	o-a-a	0.4260	0.2388
NEWS	800	bkmmeans-40	multi	0.1440	0.0705
NEWS	800	bkmmeans-40	o-a-a	0.4173	0.1549
NEWS	800	nj-24 classes	o-a-a	0.2913	0.3100
ETHZ	original	–	–	–	0.0912
ETHZ	1000	bkmmeans-20	multi	0.1060	0.1294
ETHZ	1000	bkmmeans-20	o-a-a	0.3401	0.0191
ETHZ	1000	bkmmeans-28	multi	0.1188	0.1014
ETHZ	1000	bkmmeans-28	o-a-a	0.3172	0.0648
ETHZ	1000	nj-12 classes	o-a-a	0.5236	-0.0884

Comparação com Outras Técnicas de Redução de Dimensionalidade

Esta seção mostra uma avaliação comparativa entre a metodologia proposta e diversas técnicas de redução de dimensionalidade conhecidas. Os testes com coleções textuais foram executados em um computador com processador *Intel i5* com 2.3GHz e 6GB de memória. Para coleções de

imagens, foi utilizado um computador com processador *Intel Core2 Duo*, com 2.53GHz e 4GB de memória. Foram avaliadas, juntamente com as abordagens PLS propostas, a técnica PCA, além de PivotMDS, ISOMAP e LLE, essas três últimas com características supervisionadas. A técnica *Self-Organizing Maps* (SOM) também poderia ser considerada na avaliação, mas se mostrou extremamente lenta para a redução de um grande número de dimensões para um espaço reduzido com mais de três dimensões, e por isso foi desconsiderada. Todas as técnicas utilizadas na comparação foram inseridas no sistema VisPipeline.

A Tabela 4.5 mostra os tempos de geração dos modelos para as coleções NEWS e ETHZ, bem como os valores de coeficiente de silhueta, para cada uma das técnicas. É possível perceber que a abordagem PLS demora mais tempo para gerar o modelo do que as técnicas PivotMDS, ISOMAP e LLE, mas demora menos tempo do que a PCA. Além disso, os valores de coeficiente de silhueta são melhores para o PLS do que para as outras técnicas, com exceção do PCA em apenas um caso. Vale ressaltar que os tempos mostrados na tabela são gastos com a geração do modelo PLS. Esses modelos podem ser reutilizados, e o tempo de carga é muito pequeno (como mostra a Tabela 4.6), o que torna sua utilização ainda mais interessante.

Tabela 4.5: Comparação entre os tempos de geração do modelo e os coeficientes de silhueta, considerando a abordagem PLS *One Against All* (a mais lenta) e outras técnicas de redução de dimensionalidade.

Coleção	Técnica	Tempo	Silhueta		
			Reduzida	LSP	Árvore NJ
NEWS	PCA-23	38 min.	0.3163	0.0269	0.2189
	PLS-23-usuário-800	3 min.	0.6815	0.1244	0.3456
	PLS-10-means-800	7 min.	0.279	0.1363	0.2183
	LLE	2.5 min.	-0.0195	-0.0127	-0.2491
	ISOMAP	11 seg.	-0.2720	-0.3040	-0.2266
	PivotMDS	14 seg.	-0.2062	-0.3340	0.1665
ETHZ	PCA-28	46 min.	0.1039	-0.0674	0.0972
	PLS-28-usuário-1000	12 min.	0.6277	0.7928	0.5748
	PLS-28-means-1000	18 min.	0.6132	0.5107	0.5576
	LLE	8 min.	0.08131	-0.2085	0.0884
	ISOMAP	32 seg.	-0.0652	-0.2442	0.0
	PivotMDS	48 seg.	-0.1979	-0.3429	-0.1339

A Figura 4.10 mostra as árvores NJ criadas utilizando os espaços reduzidos da coleção NEWS, criados por algumas das técnicas comparadas nesta seção. É possível perceber que os valores de coeficiente de silhueta apresentados na Tabela 4.5 são refletidos na qualidade dos *layouts* apresentados, através da análise dos agrupamentos representados pelos ramos das árvores. Isso demonstra o potencial dessa medida para a avaliação quantitativa dos espaços reduzidos produzidos.

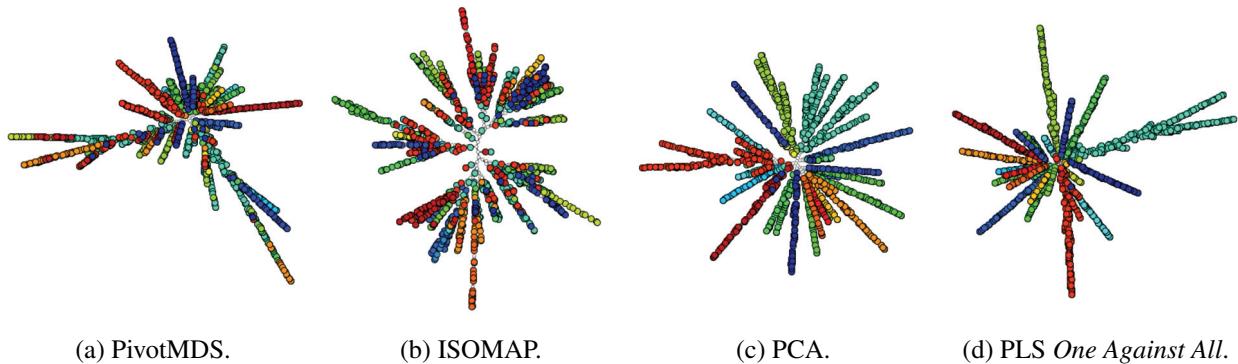


Figura 4.10: Árvores NJ construídas do espaço reduzido da coleção NEWS, utilizando PivotMDS, ISOMAP, PCA e PLS.

A Figura 4.11 mostra uma comparação entre valores de *Neighborhood Hit* da coleção NEWS utilizando as dimensões originais, e para os espaços reduzidos utilizando as técnicas consideradas na Tabela 4.5. Considera-se nessa análise o espaço reduzido PLS criado através da amostragem de 800 instâncias rotuladas e outro espaço reduzido PLS criado com 510 amostras escolhidas manualmente de maneira criteriosa. Para esse último caso, foi construído um conjunto desbalanceado com 23 classes, em uma proporção diferente da observada na coleção original, e os espaços reduzidos foram obtidos utilizando as duas abordagens PLS apresentadas. Os resultados apresentados confirmam o potencial de discriminabilidade da técnica PLS. As demais técnicas apresentaram resultados piores.

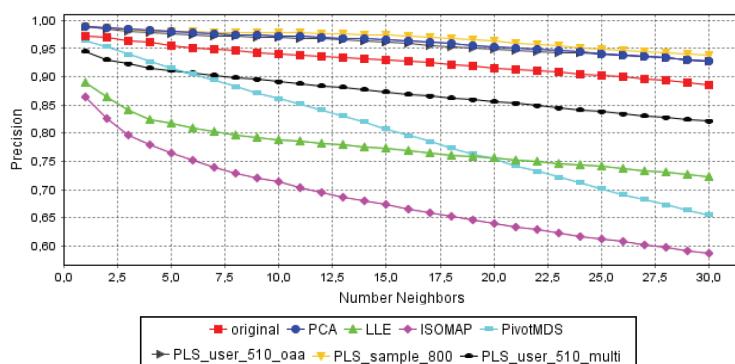


Figura 4.11: Valores da análise *Neighborhood Hit* para a coleção NEWS, considerando o espaço original e os espaços reduzidos produzidos pelas técnicas PCA, PivotMDS, ISOMAP e LLE.

Reutilização do Modelo PLS

Como dito anteriormente, os modelos PLS criados podem ser reutilizados para reduzir as dimensões ou classificar coleções que compartilhem as mesmas características, ou características similares das dos conjuntos utilizados para a criação desses modelos. A reutilização dos modelos

PLS se torna útil para o mapeamento de coleções em crescimento ou evolução. A Figura 4.12 mostra uma série de mapeamentos, utilizando um modelo previamente criado, de subconjuntos de diferentes tamanhos da coleção ALL, descrita na Tabela 4.1. Foi utilizada a abordagem *One Against All*, sendo que o modelo foi construído a partir da amostragem de 1200 instâncias, utilizando a estratégia de agrupamento automático. É possível perceber que, na medida em que o tamanho dos subconjuntos aumenta, as posições dos mapeamentos de certos grupos de instâncias são mantidas, quando se utiliza técnicas de visualização que produzem *layouts* visualmente estáveis, tais como a RadViz. Isso permite que o usuário mantenha um modelo mental da estratégia do modelo PLS. Um modelo pode ser utilizado até que ele não seja mais adequado para representar as mudanças na coleção. Nesse caso, o usuário reconstrói esse modelo, de forma a acomodar as mudanças observadas nas distribuições de classe, e o reutiliza a partir de então. O tempo de carga de um modelo previamente criado é pequeno, como pode ser comprovado na Tabela 4.6.

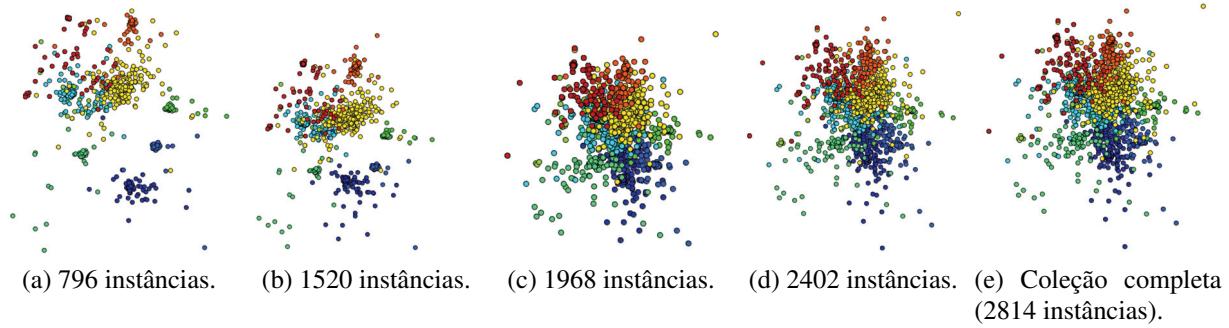


Figura 4.12: Aplicação progressiva de um modelo PLS criado previamente em subconjuntos da coleção ALL.

Tabela 4.6: Tempos de carga e aplicação de modelos PLS, para as coleções NEWS, ETHZ e ALL.

Coleção	<i>One Against All</i>	<i>MultiClassMatrix</i>
NEWS	9.3 seg.	1.3 seg.
ETHZ	19 seg.	1.3 seg.
ALL	4.5 seg.	0.6 seg.

4.5 Classificação PLS

Como mostrado na seção 4.2.2, a classificação de coleções de dados utilizando PLS utiliza o valor da regressão das instâncias de teste no(s) modelo(s) criados a partir de um conjunto de amostras. Os modelos PLS podem ser reutilizados para o mapeamento de coleções em crescimento, o que torna sua utilização interessante em diversas situações. A Tabela 4.7 mostra uma comparação entre os resultados obtidos utilizando a técnica PLS e a técnica SVM, aplicado às coleções ETHZ

e NEWS. A coleção ETHZ foi dividida em dois subconjuntos, um de treinamento contendo 178 instâncias e um conjunto de teste contendo as 1841 instâncias restantes, referenciado aqui como **ETHZ-1841**. O mesmo processo foi realizado na coleção NEWS, resultando em um conjunto de treinamento de 160 instâncias, e um conjunto de teste com 1611 instâncias, referenciado aqui como **NEWS-1611**. Foi utilizada uma SVM com *kernel* linear, e os dados foram normalizados. As instâncias de treinamento foram fornecidas pelo usuário, e as abordagens *One Against All* e *MultiClassMatrix* foram utilizadas. É possível observar que o tempo de geração dos modelos PLS é maior do que o de geração dos modelos SVM, mas o tempo de classificação é menor. Considerando que ambos os modelos podem ser reutilizados, o processo de classificação PLS acaba se mostrando mais rápido no processo como um todo. Além disso, a abordagem PLS *One Against All* apresenta resultados melhores do que a técnica SVM, demonstrando o potencial do PLS para a discriminação entre classes. A abordagem PLS *MultiClassMatrix*, apesar de ser mais rápida na geração dos modelos do que a abordagem *One Against All*, produziu resultados piores do que a técnica SVM.

Tabela 4.7: Comparação dos resultados da classificação da coleção ETHZ, utilizando modelos PLS e SVM.

	SVM	<i>One Against All</i>	<i>MultiClassMatrix</i>
ETHZ-1841			
Tempo (seg.) ¹	4 + 12.7	198.6 + 5.6	31.1 + 11.0
Acertos	1248 (67.8%)	1502 (81.6%)	527 (28.6%)
Erros	593 (32.2%)	339 (18.4%)	1314 (71.4%)
Acurácia	96.72%	98%	93.19%
Precisão	73.04%	85.87%	33.42%
Sensitividade	67.79%	81.59%	28.63%
NEWS-1611			
	SVM	<i>One Against All</i>	<i>MultiClassMatrix</i>
Tempo (seg.) ¹	4.5 + 10.7	184.1 + 3.7	107.3 + 7.5
Acertos	1409 (87.5%)	1468 (91.1%)	816 (50.7%)
Erros	202 (12.5%)	143 (8.9%)	795 (49.3%)
Acurácia	96.89%	98.56%	92.82%
Precisão	91.24%	93.17%	37.47%
Sensitividade	87.46%	91.12%	50.65%

¹ Tempo de geração do modelo + Tempo de classificação

4.6 Considerações Finais

Este capítulo apresentou uma metodologia semi-supervisionada para redução de dimensionalidade e classificação, utilizando a técnica *Partial Least Squares* (PLS). PLS utiliza um pequeno

conjunto previamente rotulado como entrada para construir um modelo (ou um conjunto de modelos) de regressão baseado em um conjunto de **variáveis latentes**, a ser utilizado no processo de análise visual de coleções. Duas abordagens de utilização do PLS foram desenvolvidas, uma que utiliza um conjunto de amostras previamente rotuladas e outra desenvolvida para situações nas quais não há informação prévia sobre as classes do problema (coleções não rotuladas), que utiliza técnicas de agrupamento para determinar rótulos para as instâncias.

Os resultados apresentados neste capítulo mostram que os espaços reduzidos utilizando a metodologia proposta apresentam alta precisão, utilizando ambas as abordagens, e as novas dimensões conseguem refletir de forma eficaz as diferenças entre classes de uma coleção. Esses espaços reduzidos apresentam as características principais das coleções mais evidentes, e por isso produzem *layouts* de visualização mais precisos, melhorando a capacidade de análise visual e exploração da estrutura da coleção. Além disso, a própria análise visual imprime eficácia ao processo de encontrar uma redução adequada.

A utilização de amostras escolhidas pelo usuário permite que seu conhecimento e perspectiva sejam inseridos no processo, ajustando o sistema de acordo com suas necessidades. Além disso, os modelos criados podem ser reutilizados em outras coleções que compartilhem as mesmas características, ou características semelhantes, possibilitando o mapeamento de coleções em crescimento. A carga de modelos gerados é rápida, o que permite uma interação efetiva no processo.

Uma comparação entre a metodologia proposta e outras técnicas de redução de dimensionalidade foi apresentada, e os resultados obtidos mostram que o PLS é capaz de gerar espaços reduzidos com a mesma precisão de técnicas como PCA, em um tempo consideravelmente menor. Outras técnicas de redução de dimensionalidade de larga utilização também foram comparadas, e apesar de serem mais rápidas do que o PLS, apresentam precisão pior.

Os resultados da classificação utilizando PLS também são promissores. Os modelos criados através dos conjuntos de amostras conseguem aprender de forma eficaz as diferenças entre as classes dos problemas, resultando em um classificador que produz altas taxas de acerto, bem como bons índices de precisão e acurácia.

O próximo capítulo apresenta uma metodologia de classificação baseada em uma técnica que utiliza modelos PLS incrementais, possibilitando que o classificador acomode mudanças na distribuição de classes causadas por evoluções em coleções de dados.

Classificação Visual Incremental de Dados

5.1 Considerações Iniciais

A classificação de dados, e em especial a classificação de imagens, tem como característica fundamental sua natureza individual, no sentido que nenhuma técnica apresenta bons resultados em todas as situações. Esses resultados dependem de diversos fatores, dentre eles o espaço de características e a medida de similaridade empregada. Além disso, a qualidade do conjunto de treinamento utilizado é de fundamental importância, sendo um determinante considerável para a acurácia do classificador (Foody & Mathur, 2006). Nesse sentido, torna-se fundamental a criação de um conjunto de treinamento que consiga caracterizar as classes e armazenar as informações necessárias para particionar corretamente o espaço de características, em cada situação.

O usuário pode exercer um papel fundamental na construção desse conjunto de treinamento, já que seu conhecimento sobre o problema pode facilitar a seleção das instâncias que representem melhor as classes existentes. No entanto, ele só poderá exercer esse papel se o conjunto do qual as instâncias de treinamento são extraídas for exibido de forma amigável, e se houver ferramentas de interação efetivas para essa tarefa. As técnicas de visualização apresentam considerável potencial para ressaltar a estrutura e organização das coleções, de forma a destacar as características específicas de cada classe, e guiar o usuário na escolha de instâncias mais representativas, além

de possibilitar a análise e compreensão das razões pelas quais o classificador tomou determinadas decisões.

O algoritmo ***Locally Weighted Projection Regression*** (LWPR) (Vijayakumar et al., 2005) é amplamente utilizado para realizar aproximação de funções em espaços de alta dimensionalidade, mesmo na presença de dimensões redundantes ou irrelevantes, com aplicações em diversas tarefas envolvendo controle de execuções concorrentes e predição de movimento. Sua principal característica é seu treinamento online e incremental, que permite que atualizações no modelo de classificação sejam realizadas para acomodar eventuais mudanças no conceito de classes, ou na estrutura das classes existentes. Essa capacidade é consideravelmente útil para permitir a inserção do usuário no processo de classificação, através do ajuste dos modelos com o objetivo de produzir os resultados desejados.

Este capítulo apresenta uma metodologia de classificação visual incremental, baseada no algoritmo *Locally Weighted Projection Regression* (LWPR) associado a técnicas de visualização, com destaque para árvores de similaridade. A hipótese é que a inserção do usuário na classificação de conjuntos de dados, através da criação e aplicação de modelos LWPR, pode criar um processo iterativo de classificação que possibilite uma rápida convergência de resultados. O algoritmo investigado é caracterizado por realizar um treinamento incremental, através de atualizações nos modelos realizadas pelo usuário, o que possibilita o mapeamento de coleções em evolução. Além disso, é possível lidar com mudanças mais drásticas no cenário de classificação, como o aparecimento de novas classes. Finalmente, é possível alterar a perspectiva da classificação, de forma a adequá-la a diferentes necessidades. Um sistema de classificação que utiliza essa metodologia é apresentado, e os resultados de diversas tarefas de criação e aplicação dos modelos LWPR em cenários de classificação são mostrados e discutidos.

Este capítulo é uma síntese de um artigo submetido ao ***15th IEEE-VGTC Eurographics Conference on Visualization (EuroVis 2013)***. O conteúdo completo do artigo pode ser encontrado no Apêndice D.

5.2 Metodologia de Classificação Visual Incremental

5.2.1 ***Locally Weighted Projection Regression***

A técnica *Locally Weighted Projection Regression* (LWPR), desenvolvida por Vijayakumar et al. (2005), consiste em um algoritmo que produz uma aproximação de uma função não linear em espaços de alta dimensionalidade, utilizando um conjunto de modelos lineares locais que abrangem um pequeno número de regressões univariadas em direções específicas do espaço original. Uma versão incremental, baseada na aplicação de pesos, da técnica *Partial Least Squares* (PLS), apresentada no Capítulo 4, é utilizada para realizar a redução de dimensionalidade nas direções

específicas. LWPR é utilizado em tarefas de predição em diversas áreas de conhecimento, tais como Medicina (Florez et al., 2011), Desenho apoiado por Computador (A.Muthukumaravel et al., 2011) e Robótica (D’Souza et al., 2001; Atkeson et al., 2000). Esta seção apresenta uma breve descrição matemática do processo.

Em termos gerais, o valor da regressão de um modelo LWPR é determinado pela média ponderada dos valores de regressão de cada um dos modelos lineares locais. A versão incremental da técnica PLS é utilizada para diminuir o custo computacional na determinação dos valores de regressão, além de ressaltar a segregação entre classes.

O treinamento do modelo LWPR é incremental. As instâncias de treinamento são utilizadas para atualizar os parâmetros de cada modelo local, que incluem os parâmetros da técnica PLS e uma medida de distância que determina o tamanho e o formato de sua região de validade. Além disso, um valor de peso é calculado, em cada modelo local, para cada instância de treinamento, que corresponde à distância dessa instância ao centro do *kernel* desse modelo. Caso o valor dos pesos calculados seja menor do que um limiar, um novo modelo local é criado.

A seguir um detalhamento matemático é apresentado. Mais detalhes podem ser encontrados em (Vijayakumar & Schaal, 2000) e (Klanke et al., 2008).

O modelo de regressão LWPR é construído por um conjunto de instâncias de treinamento apresentadas iterativamente na forma de tuplas (\mathbf{X}_i, y_i) . A predição produzida pelo modelo LWPR é dada pela média ponderada da predição de cada um dos k modelos lineares locais, de acordo com a Equação 5.1:

$$\hat{y} = \frac{\sum_1^k w_k \hat{y}_k}{\sum_1^k w_k}. \quad (5.1)$$

Os pesos w_k de cada modelo linear local definem a região de atuação desses modelos, também chamados **campos de receptividade**, e geralmente são modelados utilizando um *kernel* Gaussiano, de acordo com a Equação 5.2. Na equação, \mathbf{D}_k representa uma métrica de distância, e \mathbf{c}_k representa o centro do *kernel*. A métrica de distância determina o tamanho e a forma da região de validade de cada modelo. Assim, um valor de peso é computado para cada instância de entrada (\mathbf{X}_i, y_i) , de acordo com a distância ao centro do *kernel* de cada um dos k modelos, tornando o aprendizado em cada modelo localizado e independente. Caso o valor de ativação de todos os modelos existentes seja menor do que um limiar, um novo modelo é criado. Assim, o número de modelos é dinamicamente ajustado durante o processo.

$$w_{i,k} = \exp(-0.5(\mathbf{X}_i - \mathbf{c}_k)^T \mathbf{D}_k (\mathbf{X}_i - \mathbf{c}_k)) \quad (5.2)$$

A versão incremental da técnica PLS é utilizada no aprendizado dos modelos lineares, de forma que, em cada modelo local, a instância de entrada \mathbf{X}_i seja mapeada em um conjunto de direções u_r ,

representando r variáveis latentes. A regressão em um modelo local será formada pela combinação linear das variáveis latentes desse modelo. O processo de atualização das direções de projeção, bem como dos parâmetros de regressão $\beta_{r,k}$, são apresentados no Algoritmo 5.1.

Algoritmo 5.1: Aprendizado PLS Incremental. Adaptado de (Vijayakumar & Schaal, 2000)

Dada uma instância de treinamento (X, y_i);

Atualize médias de entrada e saída;

$$\overline{X_{n+1}} = \frac{\lambda W_n \overline{X_n} + w X}{W_{n+1}};$$

$$\overline{\beta_{n+1}} = \frac{\lambda W_n \overline{\beta_n} + w y}{W_{n+1}};$$

onde $W_{n+1} = \lambda W_n + w$ e $\overline{X_0} = 0, U_i^0 = 0, \overline{\beta_n} = 0, W_0 = 0$;

Atualize o modelo local:

Inicialize $Z = X, res_1 = y - \overline{\beta_{n+1}}$;

para $i = 1 : r$ **hacer**

a) $U_i^{n+1} = \lambda U_i^n + w Z res_i$;

b) $s = Z^T U_i^{n+1}$;

c) $SS_i^{n+1} = \lambda SS_i^n + ws^2$;

d) $SR_i^{n+1} = \lambda SR_i^n + wsres_i$;

e) $SZ_i^{n+1} = \lambda SZ_i^n + wZs$;

f) $\overline{\beta_i^{n+1}} = SR_i^{n+1}/SS_i^{n+1}$;

g) $P_i^{n+1} = SZ_i^{n+1}/SS_i^{n+1}$;

h) $Z = Z - sP_i^{n+1}$;

i) $res_{i+1} = res_i - s\beta_i^{n+1}$;

j) $MSE_i^{n+1} = \lambda MSE_i^n + wres_{i+1}^2$;

fin para

onde:

$\lambda \in [0, 1]$: fator de esquecimento;

W : matriz diagonal de pesos;

Z : projeção da instância X no espaço reduzido via PLS;

SS, SR, SZ : variáveis de auxílio para a regressão local utilizando recursão por mínimos quadrados;

res : resíduo da regressão anterior;

P : valor da regressão do espaço de entrada, utilizado para garantir a ortogonalidade nas direções de projeção;

MSE : erro quadrado médio.

A métrica de distância D_k é atualizada individualmente, para cada modelo local, utilizando gradiente descendente baseado em um critério de validação cruzada estocástica (Vijayakumar et al., 2005). O Algoritmo 5.2 ilustra o processo de aprendizado de um modelo LWPR, através do aprendizado dos modelos locais apresentados.

O método LWPR apresenta propriedades que tornam sua utilização interessante. Os parâmetros dos modelos locais podem ser estimados através do armazenamento de informações estatísticas dessas instâncias, e a adaptação desses modelos, bem como da métrica de distância, podem

Algoritmo 5.2: *Locally Weighted Projection Regression.* Adaptado de (Vijayakumar et al., 2005)

```

Iniciar o modelo LWPR sem nenhum modelo local ( $ML = 0$ );
para cada instância de treinamento ( $\mathbf{X}, y_i$ ) hacer
    para  $k = 1 : ML$  hacer
        Calcule o valor de ativação (Equação 5.2);
        Atualize modelo local (Algoritmo 5.1) e  $D_k$ ;
    fin para
    se nenhum valor de ativação dos modelos locais é maior do que  $w_{gen}$  então
        Crie novo modelo local com  $r_k = 2$ ,  $c_k = X$ ,  $D_k$  = medida padrão;
    fim se
fin para cada
 $w_{gen}$  : limiar para geração de novos modelos locais.

```

ser feitos utilizando validação cruzada estocástica, eliminando a necessidade de se armazenar as instâncias de treinamento. A atualização dinâmica desses modelos possibilita que o espaço de entrada seja coberto por modelos de cobertura ampla em regiões de baixa curvatura, e por modelos de cobertura pequena em regiões de alta curvatura (Klanke et al., 2008). Além disso, o método tem complexidade linear no número de entradas, e é capaz de lidar com um grande número de dimensões redundantes presentes nos dados de entrada. Finalmente, o aprendizado é realizado de forma incremental, o que permite que evoluções nas coleções de dados sejam acomodadas pelo modelo de regressão.

5.2.2 Descrição da Metodologia

De acordo com o funcionamento da técnica LWPR, apresentada na Seção 5.2.1, a partir de um conjunto de instâncias de treinamento previamente rotuladas em C classes, é criado um modelo de regressão que será utilizado na classificação de uma coleção de dados. Além disso, é possível atualizar o modelo criado através da adição de novas instâncias, de forma a refletir alterações no cenário de classificação. Duas metodologias foram investigadas, *One Against All* e *Multiclass-Matrix*, cujas descrições foram apresentadas no Capítulo 4.

Não foram observadas diferenças significativas entre as duas abordagens, considerando tempo computacional e precisão dos modelos gerados. No entanto, a abordagem *One Against All* requer a geração de um arquivo para cada modelo, para cada classe, e por isso necessita de mais tempo para a criação desses arquivos do que a abordagem *MulticlassMatrix*, que necessita criar apenas um arquivo para o modelo. Isso também ocorre no processo de atualização dos modelos.

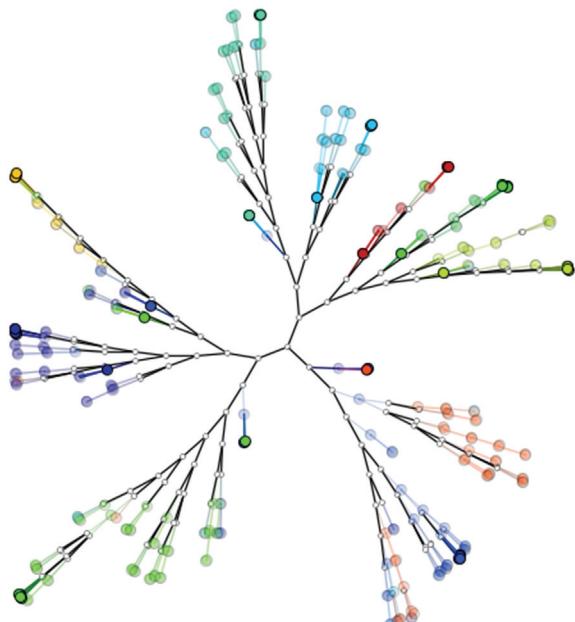
Em ambas as abordagens, os modelos podem ser atualizados com a adição de novas instâncias, de forma a acomodar eventuais evoluções no cenário de classificação. Essas evoluções podem

contemplar alterações nas estruturas das classes existentes, ou mesmo o aparecimento de novas classes. Todas essas tarefas são detalhadas a seguir.

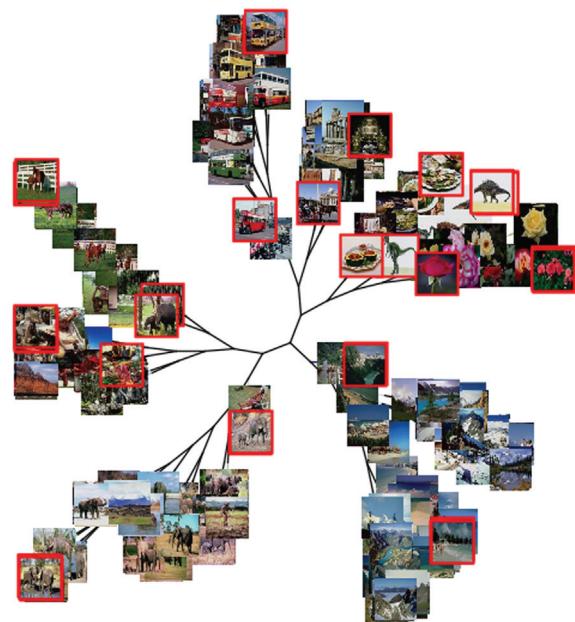
Criação e Aplicação do Modelo

O conjunto de treinamento a ser utilizado na criação dos modelos LWPR, de acordo com as abordagens apresentadas, é informado pelo usuário, e pode ser construído a partir da seleção de instâncias em um *layout*. A estrutura e organização dos pontos nesse *layout* serve, assim, como um guia para a seleção de instâncias estratégicas, que podem contribuir melhor para a criação de um modelo robusto.

As Figuras 5.1a e 5.1b mostram duas visões de uma árvore NJ de um subconjunto da coleção COREL contendo 300 imagens, chamado aqui de **COREL-300**, da qual o usuário selecionou 44 imagens para o conjunto de treinamento. Na árvore NJ, optou-se pela escolha de um conjunto de instâncias formado pela combinação daquelas mais distantes do centro da árvore, que representam as instâncias cujos atributos são bem característicos das classes nas quais estão inseridas, e daquelas mais próximas do centro da árvore, cujos atributos não representam claramente nenhuma classe específica, ou que representam mais de uma classe. Caso não haja instâncias previamente rotuladas, ou caso seja necessário alterar o rotulamento existente, é possível associar rótulos às instâncias que serão selecionadas para a criação do modelo, utilizando as ferramentas apresentadas na Seção 5.4, permitindo que o conhecimento do usuário seja inserido no processo.



(a) Instâncias selecionadas.



(b) Imagens selecionadas.

Figura 5.1: Exemplo de árvore NJ para a coleção COREL-300, com 44 instâncias selecionadas, representadas por círculos (5.1a) e por imagens (5.1b).

Os modelos LWPR criados com as instâncias selecionadas anteriormente poderão ser aplicados na classificação de qualquer coleção de dados que compartilhe o mesmo espaço de características ou que possua um espaço semelhante. Caso o resultado seja aquém do esperado, o usuário pode atualizar o modelo de forma a acomodar as diferenças existentes nessa coleção. A Figura 5.2a mostra uma árvore NJ com o *ground truth* de outro subconjunto da coleção COREL, com 700 imagens, chamado aqui de **COREL-700**. Esse conjunto não contém nenhuma imagem presente no conjunto COREL-300. O resultado visual da classificação utilizando o modelo LWPR, criado com as instâncias selecionadas de COREL-300, é mostrado na Figura 5.2b. Já o resultado utilizando a funcionalidade **Class Matching**, detalhada na Seção 5.4.3, é mostrado na Figura 5.2c e os resultados numéricos apresentados na Tabela 5.1.

As medidas de avaliação de classificação utilizadas nos experimentos foram **Acurácia**, que mede a proporção de instâncias corretamente classificadas, dentre todas as instâncias da coleção, **Precisão**, que mede a proporção de instâncias corretamente categorizadas em uma determinada classe, dentre todas as instâncias categorizadas nessa classe, e **Sensitividade**, que mede a proporção de instâncias corretamente categorizadas em uma determinada classe, dentre todas as instâncias que realmente pertencem a essa classe. Vale ressaltar que a medida de acurácia é dependente do balanceamento das classes do problema, e dessa forma foi utilizada nos experimentos sempre associada com as outras medidas, de forma a garantir uma análise mais precisa da classificação. Considerando, para uma classe i :

- TP_i : número de instâncias da classe i classificadas como i ;
- FN_i : número de instâncias da classe i classificadas em outra classe;
- FP_i : número de instâncias de outra classe classificadas como i ;
- TN_i : número de instâncias de outra classe classificadas em outra classe;
- T : total de instâncias da coleção.

As fórmulas para o cálculo da Acurácia, Precisão e Sensitividade são ilustradas nas Equações 5.3, 5.4 e 5.5, respectivamente. Para avaliar a classificação em todas as classes, foi utilizada a média dos valores dessas medidas.

$$Acuracia_i = TP_i/T \quad (5.3)$$

$$Precisao_i = TP_i/(TP_i + FP_i) \quad (5.4)$$

$$Sensitividade_i = TP_i/(TP_i + FN_i) \quad (5.5)$$

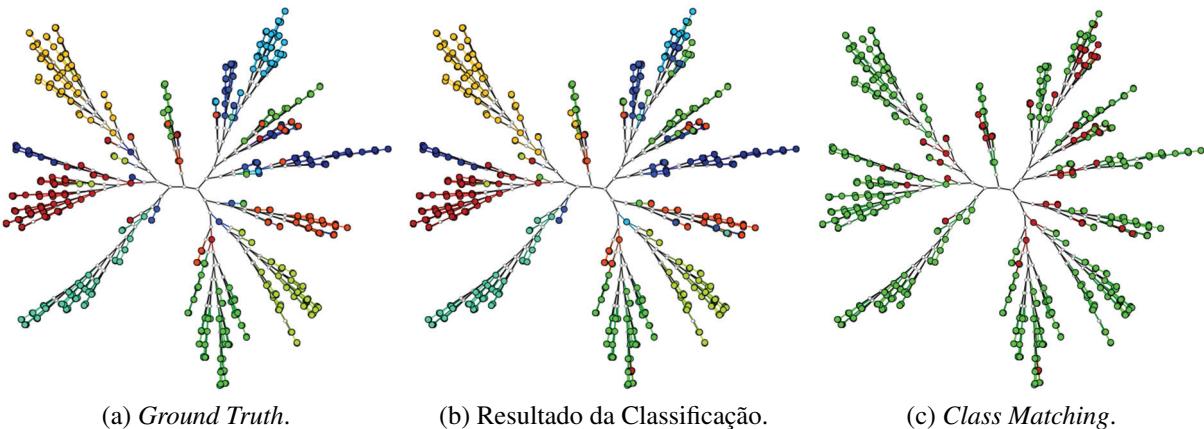


Figura 5.2: Exemplo do resultado da classificação de um subconjunto da coleção COREL com 700 imagens.

Tabela 5.1: Resultado da classificação da coleção COREL-700.

Classificações Corretas	605 (86.4%)
Classificações Incorretas	95 (13.6%)
Acurácia	97.58%
Precisão	87.96%
Sensitividade	86.43%

Atualização do Modelo

A atualização dos modelos LWPR é realizada pelo usuário da mesma maneira que a criação de novos modelos, ou seja, através da seleção de um conjunto de instâncias presentes em um *layout*. Como o treinamento é realizado de forma incremental, as instâncias selecionadas atualizarão cada modelo local PLS e, se necessário, criará novos modelos locais, não sendo necessária nenhuma informação a respeito das instâncias já aprendidas. O modelo atualizado agregará informações de ambos os conjuntos de instâncias utilizadas na criação e atualização.

Várias estratégias de atualização do modelo LWPR podem ser adotadas. Uma delas consiste em treiná-lo utilizando um conjunto de instâncias classificadas de forma errada pelo modelo inicial, com o intuito de fornecer mais informações a respeito das classes com deficiência na classificação. Nesse caso, mais uma vez o *layout* funciona como guia, pois o usuário pode compreender os motivos da classificação ter sido realizada de determinada maneira. Além disso, a posição de instâncias classificadas de forma errada pode indicar as regiões do espaço correspondente a cada classe que o modelo atual não está conhecendo adequadamente, direcionando a escolha para uma convergência de resultados.

Recriação do Modelo com Novas Classes

Em alguns cenários de classificação, é possível que as coleções de dados evoluam, e novos conceitos sejam adicionados às suas informações, acarretando alterações mais drásticas na distribuição de classes das instâncias, caracterizadas pelo surgimento de novas classes. Nesse caso, os modelos existentes não possuirão informações suficientes para reconhecer instâncias dessas novas classes, e a atualização envolverá a recriação desses modelos baseado no novo conjunto de classes. Essa recriação permite, no entanto, que as instâncias utilizadas na criação e atualizações passadas sejam novamente utilizadas para o treinamento dos modelos durante a recriação, para que eles não agreguem apenas informações a respeito das instâncias da última iteração. Assim, optou-se por armazenar todas as instâncias utilizadas previamente, na construção e atualizações dos modelos. De posse dessas instâncias, um procedimento de atualização envolvendo instâncias de n classes não conhecidas é aplicado, de acordo com a abordagem LWPR previamente escolhida.

Utilizando a abordagem *One Against All*, as instâncias da atualização em curso pertencentes às n classes não conhecidas são utilizadas na atualização dos modelos existentes, e n novos modelos são criados. As instâncias previamente armazenadas, juntamente com as novas instâncias, são então utilizadas no treinamento desses novos modelos, resultando em um total de $C + n$ modelos após a atualização.

Na atualização da abordagem *MulticlassMatrix*, é necessário que n dimensões de saída sejam adicionadas ao modelo de regressão. Assim, um novo modelo é criado, com $C + n$ respostas, e as instâncias previamente armazenadas, juntamente com as novas instâncias, são utilizadas no treinamento desse novo modelo.

Em ambas as abordagens, as instâncias da atualização em curso são adicionadas às instâncias armazenadas, de forma que possam ser utilizadas em futuras atualizações envolvendo novas classes.

5.3 Análise de Resultados

Esta seção apresenta os resultados de diversos estudos de caso realizados utilizando o sistema de classificação visual, que adota a abordagem proposta. O objetivo desses estudos foi avaliar o papel das técnicas de visualização e das ferramentas de interação desenvolvidas em possibilitar a associação do usuário com técnicas de classificação, para produzir resultados satisfatórios em diversos cenários.

Os experimentos foram realizados utilizando a biblioteca LWPR¹ (Klanke et al., 2008), empacotada em uma classe JAVA, contendo todos os métodos de criação e atualização dos modelos LWPR, bem como os métodos de classificação de instâncias usando tais modelos. O equipamento

¹Disponível em <http://wcms.inf.ed.ac.uk/ipab/slmc/research/software-lwpr>

utilizado foi um computador com processador Intel Core2 Duo, com 2.53GHz e 4GB de memória principal. A coleção de documentos textuais ALL e a coleção de imagens ETHZ, apresentadas no Capítulo 4 e detalhadas na Tabela 5.2, foram utilizadas nos estudos. A dimensionalidade de ambas as coleções foi reduzida utilizando a abordagem PLS *MulticlassMatrix*, também apresentada no Capítulo 4, com conjuntos de amostras de 647 e 400 instâncias respectivamente. A redução da dimensionalidade dessas coleções teve o intuito de realçar a separabilidade entre as classes dessas coleções e facilitar a seleção de instâncias representativas por parte do usuário. Além disso, a biblioteca LWPR utilizada apresenta uma limitação com relação ao número de dimensões dos dados de entrada.

Tabela 5.2: Descrição das coleções utilizadas nos experimentos.

Coleção	Coleção de Origem	Instâncias	Classes	Dimensões Originais	Dimensões Reduzidas
ALL-Reduced	ALL	2814	9	5163	63
ETHZ-Reduced	ETHZ	2019	28	3963	48

Os estudos de caso investigados abordam cinco situações: influência da escolha das instâncias na criação de conjuntos de treinamento; construção, aplicação e atualização do modelo LWPR; classificação e atualização iterativa do modelo LWPR para adaptação a coleções diferentes; evolução das classes em determinado cenário e alteração na perspectiva de classificação de uma coleção. O detalhamento desses estudos de caso, bem como os resultados e conclusões obtidas são apresentados a seguir.

5.3.1 Escolha das Instâncias

Este estudo de caso avaliou como a escolha de instâncias pode influenciar na criação/atualização do modelo LWPR em uma classificação. Em uma árvore NJ, as instâncias mais distantes do núcleo da árvore, localizadas na região das folhas mais externas, apresentam características que representam bem o padrão da classe às quais pertencem, situando-se próximas dos centróides desses grupos. Já as instâncias situadas em posições intermediárias da árvore, próximas ao seu núcleo e aos seus ramos centrais possuem características que dificultam sua categorização em determinada classe, situando-se nos limites dos grupos que representam suas classes. Foram utilizados três conjuntos de treinamento, um composto por instâncias mais externas, um segundo contendo instâncias mais internas, e um terceiro contendo instâncias combinadas dos dois conjuntos anteriores. Em todos os casos, os conjuntos de treinamento e de teste são disjuntos.

A Tabela 5.3 apresenta os conjuntos utilizados neste experimento, obtidos das coleções ETHZ-Reduced e ALL-Reduced, e a Tabela 5.4 mostra os resultados de cada classificação, para as duas coleções. Os piores resultados foram obtidos quando utilizou-se as instâncias mais externas da

árvore, mostrando que, como essas instâncias estão muito próximas dos centróides dos grupos que representam as classes, fazem com que o modelo criado defina limites muito restritos para cada classe, e produzam classificadores pouco flexíveis. Já ao utilizar instâncias mais próximas do centro da árvore, consegue-se produzir um classificador mais apto a representar o conhecimento a respeito das fronteiras entre grupos homogêneos.

Tabela 5.3: Conjuntos utilizados no experimento.

Coleção	Instâncias Treinamento	Instâncias Teste
ALL-Reduced	45	2769
ETHZ	112	1907

Tabela 5.4: Comparação dos resultados da classificação utilizando os três tipos de conjuntos de treinamento.

	Instâncias Externas	Instâncias Internas	Instâncias Combinadas
ETHZ-Reduced			
Classificações Corretas	1478 (77.5%)	1592 (83.5%)	1713 (89.8%)
Classificações Incorretas	429 (22.5%)	315 (16.5%)	194 (10.2%)
Acurácia	97.12%	98.41%	98.73%
Precisão	83.41%	88.59%	92.62%
Sensitividade	77.5%	83.48%	89.83%
ALL-Reduced			
Classificações Corretas	1410 (50.9%)	1623 (58.6%)	1609 (58.1%)
Classificações Incorretas	1359 (49.1%)	1146 (41.4%)	1160 (41.9%)
Acurácia	80.53%	85.04%	84.23%
Precisão	60.87%	63.44%	60.99%
Sensitividade	50.92%	58.61%	58.11%

5.3.2 Construção e Atualização do Modelo LWPR

Este estudo de caso avaliou o papel do *layout* na construção e atualização de um modelo LWPR, de forma a melhorar os resultados da classificação de uma coleção. Inicialmente foram escolhidos, utilizando o *layout*, um número de instâncias para formar o conjunto de treinamento, a ser utilizado para construir o modelo LWPR. Esse modelo foi então utilizado para classificar o restante das instâncias da coleção. A funcionalidade **Class Matching**, descrita na Seção 5.4.3, foi utilizada nesse momento para visualizar a distribuição das instâncias classificadas incorretamente. Desse *layout*, mais um conjunto de instâncias, todas classificadas incorretamente, foi selecionado para atualizar o modelo LWPR, que por sua vez foi utilizado novamente para classificar a coleção. É

importante ressaltar que, na ausência de um *ground truth*, é o usuário quem decide se a classificação foi realizada corretamente ou não para determinada instância ou grupo de instâncias.

Da coleção ETHZ-Reduced, foram escolhidas 84 instâncias para o conjunto de treinamento, 3 de cada classe, restando 1935 instâncias para o conjunto de teste. Para o conjunto de treinamento da coleção ALL-Reduced, foram escolhidas 45 instâncias, 5 de cada classe, restando 2769 instâncias para o conjunto de testes.

O resultado numérico obtido na classificação da coleção ETHZ-Reduced é mostrado na Tabela 5.5, segunda coluna. Pela matriz de confusão obtida no processo, bem como pela comparação entre os *layouts* exibidos nas Figuras 5.3a e 5.3b, é possível perceber que duas classes concentraram maior percentual de erros, com 109 e 65 instâncias classificadas de forma incorreta para as classes 6 e 25, respectivamente. Assim, 27 instâncias dessas duas classes, selecionadas criteriosamente, foram utilizadas para atualizar o modelo LWPR. A terceira coluna da Tabela 5.5 mostra os resultados obtidos de uma segunda classificação, realizada com o modelo LWPR atualizado.

Tabela 5.5: Comparação dos resultados da classificação utilizando o modelo LWPR inicial e o atualizado, para as coleções ETHZ-Reduced e ALL-Reduced.

	Modelo LWPR Inicial	Modelo LWPR Atualizado
ETHZ-Reduced		
Classificações Corretas	1704 (88.1%)	1779 (91.9%)
Classificações Incorretas	231 (11.9%)	156 (8.1%)
Acurácia	98.48%	98.96%
Precisão	89.09%	92.99%
Sensitividade	88.06%	91.94%
ALL-Reduced		
Classificações Corretas	1875 (67.7%)	1991 (71.9%)
Classificações Incorretas	894 (32.3%)	778 (28.1%)
Acurácia	86.61%	88.45%
Precisão	71.98%	73.79%
Sensitividade	67.71%	71.90%

Os resultados obtidos da classificação utilizando o modelo atualizado são visualmente representados na Figura 5.3c. A melhora no resultado do classificador reflete o aprendizado do modelo LWPR. A árvore obtida pelo *Class Matching* se mostra importante para guiar a escolha das instâncias mais representativas, de forma a tornar o aprendizado do modelo mais rápido. As mesmas conclusões podem ser tiradas dos resultados obtidos para a coleção ALL-Reduced, mostrados também na Tabela 5.5, mostrando que o *layout* consegue guiar o usuário na escolha das instâncias que melhoram o modelo LWPR.

Para confirmar a utilidade da visualização em árvore na escolha das instâncias para a atualização do modelo LWPR, um outro experimento foi realizado, após a classificação das coleções

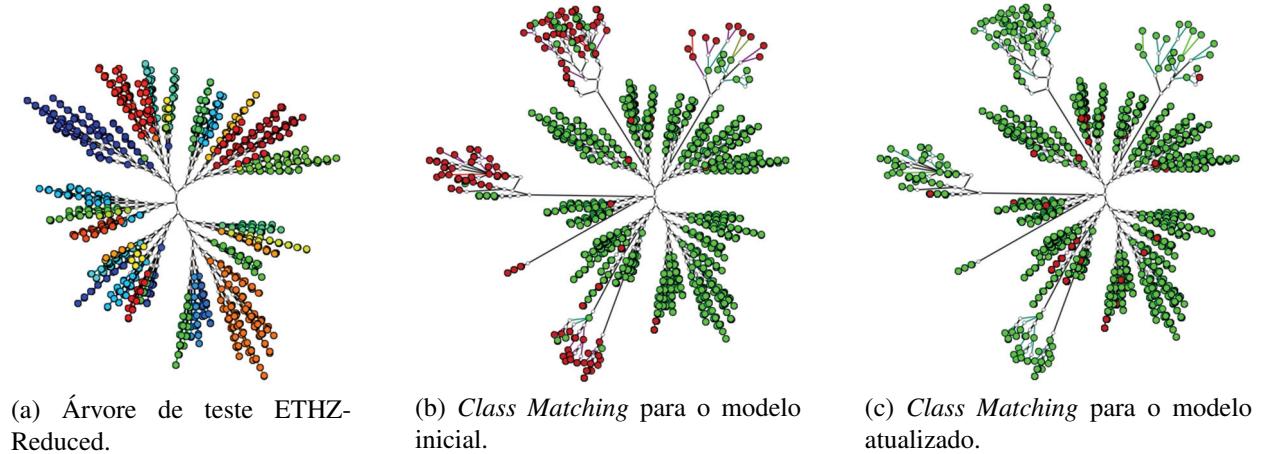


Figura 5.3: Comparação visual entre os resultados da classificação utilizando o modelo LWPR inicial e o atualizado.

utilizando o conjunto de teste inicial. O experimento consistiu em reproduzir o procedimento anterior, com a diferença de que as instâncias utilizadas na atualização do modelo LWPR foram escolhidas aleatoriamente, para cada conjunto. Foram utilizados 10 conjuntos aleatórios, e a média dos resultados obtidos com os modelos LWPR atualizados é mostrada na Tabela 5.6.

Tabela 5.6: Média dos resultados da classificação das coleções ETHZ-Reduced e ALL-Reduced, utilizando instâncias escolhidas aleatoriamente para atualizar o modelo LWPR.

	ETHZ-Reduced	ALL-Reduced
Classificações Corretas	1802 (93.1%)	1812 (65,4%)
Classificações Incorretas	132 (6.8%)	956 (34,5%)
Acurácia	99.06%	84.9%
Precisão	94.22%	74.19%
Sensitividade	93.17%	65.45%

Para a coleção ETHZ-Reduced, a média dos resultados obtidos supera ligeiramente os valores observados quando a escolha foi realizada pelo usuário. Isso é esperado, uma vez que essa coleção apresenta boa separabilidade de classes, de forma que grande parte das instâncias possuem características que representam bem as classes às quais pertencem. Assim, torna-se maior o número de seleções que ajudam a melhorar o aprendizado do modelo atualizado. Para a coleção ALL-Reduced, a escolha guiada pelo *layout* produz um modelo melhor. Essa coleção possui uma separabilidade pior, e o critério de escolha das instâncias influencia consideravelmente o aprendizado do modelo sobre determinada classe. Nesse caso, o *layout* foi de fundamental importância para uma escolha acertada. Considerando uma situação real, na qual não existe um *ground truth* para a coleção, não é possível criar um procedimento automático para realizar a escolha de instâncias para atualização do modelo. Assim, um direcionamento para essa escolha torna-se fundamental

para o bom aprendizado do classificador, direcionamento esse que pode ser conseguido através do *layout*.

5.3.3 Classificação Iterativa

Este estudo de caso tem o objetivo de verificar a convergência de uma classificação através de um processo iterativo de modificação e aplicação de um modelo LWPR. Inicialmente, um conjunto de treinamento foi utilizado para construir um modelo LWPR, utilizado para classificar uma coleção. Esse modelo foi então atualizado de acordo com os resultados observados nessa classificação, e utilizado para classificar uma segunda coleção. Novamente, o modelo LWPR foi atualizado, de acordo com os resultados da segunda classificação, e utilizado uma última vez para classificar uma terceira coleção. Esse processo foi aplicado na coleção ALL-Reduced, descrita na Tabela 5.2, e será detalhado a seguir.

Iteração 1: Da coleção ALL-Reduced foram construídos três conjuntos disjuntos, de acordo com a Tabela 5.7.

Tabela 5.7: Construção de conjuntos de instâncias extraídas da coleção ALL-Reduced para o experimento de classificação iterativa.

Conjunto	Instâncias
Treinamento	45
ALL-Reduced01	926
ALL-Reduced02	922

O conjunto de treinamento foi empregado na criação de um modelo LWPR, que por sua vez foi utilizado para classificar a coleção ALL-Reduced01. O resultado numérico dessa classificação é mostrado na Tabela 5.8, na coluna 2, e as árvores NJ e *Class Matching* para esse conjunto apresentadas na Figura 5.4. Observou-se que as classes 2, 4, 8 e 0 apresentaram altas taxas de classificações incorretas: 67.6%, 56.3%, 50.77% e 47.25%, respectivamente.

Iteração 2: 8 instâncias das classes 2, 4, 8, 0 (2 de cada classe) foram escolhidas da árvore NJ da coleção ALL-Reduced01 para atualizar o modelo LWPR, que foi então utilizado para classificar a coleção ALL-Reduced02. O resultado numérico dessa classificação é mostrado na Tabela 5.8, na coluna 5, e a árvore NJ do conjunto, juntamente com a árvore *Class Matching* correspondente apresentada na Figura 5.5.

A título de comparação, a Tabela 5.8, na coluna 4, apresenta o resultado da aplicação do primeiro modelo LWPR construído na classificação do conjunto ALL-Reduced02. É possível perceber uma sensível melhora no resultado após a atualização do modelo, que também é observada quando se comparam os resultados obtidos na classificação do conjunto ALL-Reduced01, utilizando o modelo inicial e o atualizado, mostrado nas colunas 2 e 3 da mesma tabela.

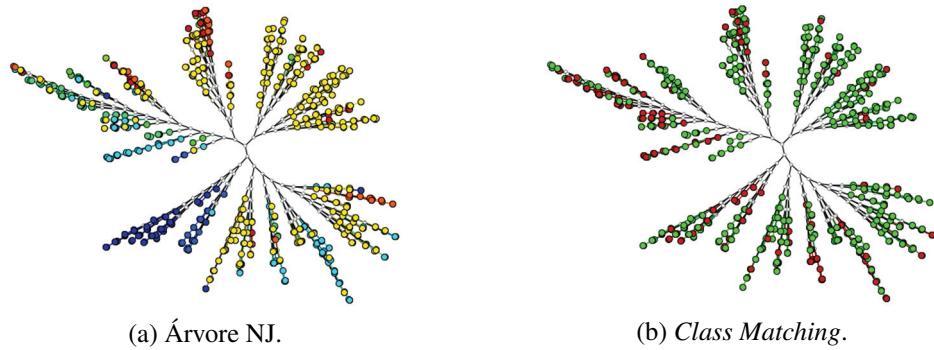


Figura 5.4: Árvore NJ da coleção ALL-Reduced01 e resultado da classificação utilizando o modelo LWPR criado na Iteração 1.

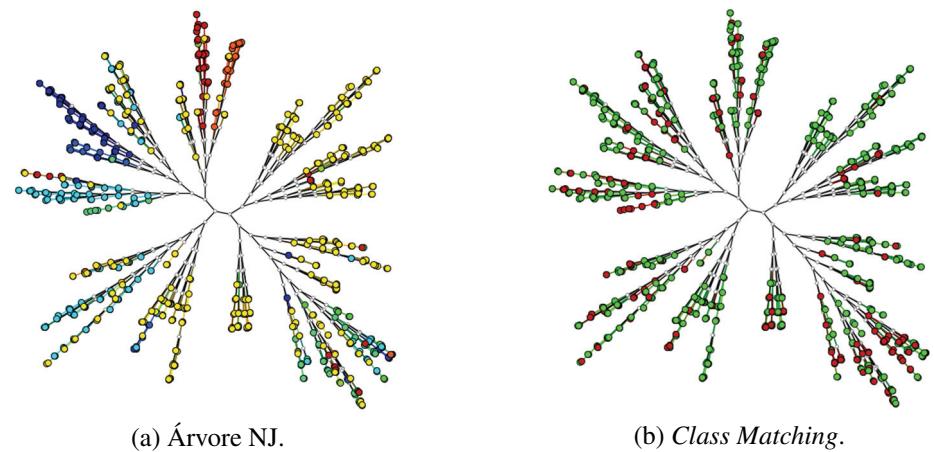


Figura 5.5: Árvore NJ da coleção ALL-Reduced02 e resultado da classificação utilizando o modelo LWPR criado na Iteração 2.

Iteração 3: 6 instâncias das classes 2 e 3, classificadas incorretamente, foram escolhidas da árvore NJ da coleção ALL-Reduced02, para atualizar novamente o modelo LWPR. O modelo com essa segunda atualização foi utilizado na classificação de um subconjunto da coleção ALL-Reduced contendo 2769 instâncias. Esse conjunto contém as instâncias utilizadas nas atualizações do modelo, mas não contém nenhuma instância pertencente ao conjunto de treinamento inicial. Os resultados das classificações dessa coleção, utilizando as três versões dos modelos LWPR utilizados são apresentados na Tabela 5.9, mostrando que a atualização guiada pela árvore NJ *Class Matching* auxilia o usuário na escolha de instâncias representativas, que ajudam a tornar o modelo mais apto para classificar uma coleção.

5.3.4 Evolução das Classes do Problema

Este estudo de caso tem o objetivo de analisar como o sistema de classificação visual pode auxiliar a atualização do modelo LWPR em situações nas quais ocorre uma mudança ou evolução de conceito no problema em questão, surgindo novas classes. O experimento consistiu em criar um

Tabela 5.8: Comparação dos resultados da classificação utilizando os modelos LWPR criados no processo iterativo de atualização do modelo, nos conjuntos ALL-Reduced01 e ALL-Reduced02.

	ALL-Reduced01		ALL-Reduced02	
	Iteração 1	Iteração 2	Iteração 1	Iteração 2
Classificações Corretas	632 (68.3%)	661 (71.4%)	613 (66.5%)	655 (71.0%)
Classificações Incorretas	294 (31.7%)	265 (28.6%)	309 (33.5%)	267 (29.0%)
Acurácia	86.81%	88.40%	86.27%	88.22%
Precisão	73.07%	73.99%	70.26%	73.10%
Sensitividade	68.25%	71.38%	66.49%	71.04%

Tabela 5.9: Comparação dos resultados da classificação utilizando as três versões do modelo LWPR na coleção ALL-Reduced com 2769 instâncias.

	Iteração 1	Iteração 2	Iteração 3
Classificações Corretas	1875 (67.7%)	1946 (70.3%)	2008 (72.5%)
Classificações Incorretas	894 (32.3%)	823 (29.7%)	761 (27.5%)
Acurácia	86.61%	87.71%	88.24%
Precisão	71.98%	72.84%	74.20%
Sensitividade	67.71%	70.28%	72.52%

modelo LWPR, utilizando um conjunto de treinamento com instâncias de x classes, e classificar uma coleção com y classes, $x < y$. O desempenho do classificador nas classes já conhecidas foi analisado, e verificou-se em quais classes existentes foram inseridas as instâncias das classes não conhecidas pelo modelo. Esse modelo foi então atualizado com instâncias dessas novas classes, e uma nova classificação foi realizada. Duas maneiras de atualização do modelo foram consideradas: a primeira delas utilizou apenas instâncias das classes não consideradas no modelo anterior, e a segunda utilizou, além dessas instâncias, outras cujas classes foram consideradas pelo modelo anterior.

Um subconjunto da coleção ETHZ-Reduced com 717 instâncias divididas em 10 classes, chamado **ETHZ-Reduced717**, foi utilizado. Desse subconjunto, 100 instâncias das classes 4, 10, 16, 23, 25 e 26 foram utilizadas para construir um modelo LWPR. As classes não consideradas foram: 5, 7, 8 e 13. A Tabela 5.10 mostra como o modelo classificou as instâncias das 6 classes conhecidas, e a Tabela 5.11 mostra em quais classes foram inseridas as instâncias de classes não conhecidas.

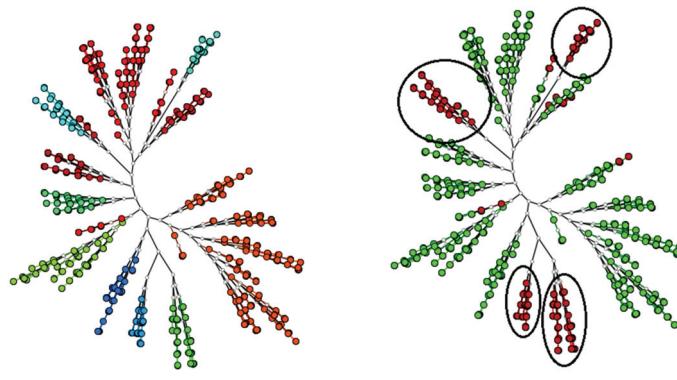
A Figura 5.6 mostra o *ground truth* da coleção (5.6a), juntamente com a árvore *Class Matching* da classificação (5.6b). Como esperado, 4 ramos foram classificados totalmente incorretos, representando exatamente as classes não conhecidas.

Tabela 5.10: Taxa de Acertos para a classificação do conjunto ETHZ-Reduced717, considerando as classes conhecidas pelo modelo LWPR utilizado.

Classe	Acertos
4	35/36 (97.2%)
10	47/47 (100.0%)
16	72/72 (100.0%)
23	207/211 (98.1%)
25	125/133 (93.9%)
26	57/73 (78.1%)

Tabela 5.11: Distribuição das instâncias das 4 classes da coleção ETHZ-Reduced717 não conhecidas pelo modelo LWPR nas 6 classes conhecidas.

	4	10	16	23	25	26
5	15	3	0	8	3	0
7	1	11	10	0	7	13
8	0	0	0	0	27	0
13	2	21	1	5	9	9



(a) *Ground truth* da coleção ETHZ-Reduced717 (b) Árvore *Class Matching* da classificação.

Figura 5.6: Comparação entre o *ground truth* e a árvore *Class Matching* da classificação da coleção ETHZ-Reduced717 utilizando o modelo construído com apenas 6 classes, mostrando os 4 ramos nos quais todas as instâncias foram classificadas de forma incorreta.

Na Figura 5.7, é possível notar que as instâncias da classe 8, não conhecida pelo classificador, estão em um ramo muito próximo de outro ramo que contém apenas instâncias da classe 25, que é conhecida, o que pode justificar o porquê dessas instâncias serem todas classificadas como 25.

Esse exemplo ilustra a capacidade do *layout* em fornecer sinais de que novas classes podem estar surgindo na coleção, com instâncias representadas por padrões desconhecidos pelo modelo. Aqui, dois ramos distintos apresentam o mesmo rótulo de classe, sugerindo que um desses ramos pode representar uma classe desconhecida pelo modelo atual, cujas instâncias o classificador tentou inserir em uma classe conhecida. Em projeções multidimensionais, a presença de grupos de

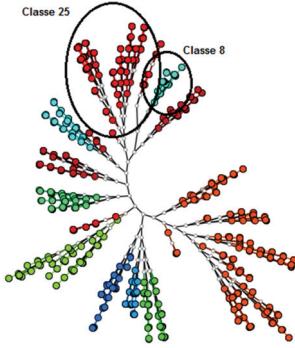


Figura 5.7: Árvore NJ da coleção ETHZ-Reduced717, mostrando a distribuição das classes do problema, com destaque para o relacionamento entre instâncias da classe 8 e 25.

instâncias com o mesmo rótulo de classificação, mas que se mostram parcialmente ou totalmente desconexos, inclusive em regiões distantes no espaço de visualização, pode também indicar o aparecimento de novas classes. Tais tendências no *layout* não resultam sempre na existência de novas classes, mas são indícios que podem merecer uma análise adicional por parte do usuário.

A partir desse resultado, o modelo LWPR foi atualizado de duas maneiras, uma utilizando apenas instâncias das classes não conhecidas, e a outra considerando instâncias de todas as 10 classes. Os resultados numéricos das classificações realizadas com os dois modelos são mostrados na Tabela 5.12, e as árvores *Class Matching* correspondentes são mostradas na Figura 5.8. É possível perceber que os resultados são melhores quando se atualiza o modelo LWPR com instâncias de todas as classes do problema. Quando foram utilizadas apenas instâncias de 4 classes, as instâncias pertencentes às classes novas foram todas classificadas corretamente, como pode ser conferido na matriz de confusão da Figura 5.9, mas muitas instâncias das classes antigas, antes classificadas corretamente, foram classificadas incorretamente. Assim, o treinamento de novas classes deve incluir as classes existentes, para que o conceito anterior seja reforçado, e não tendencioso para as últimas instâncias utilizadas no treinamento.

Tabela 5.12: Comparação entre os resultados da classificação da coleção ETHZ-Reduced717 utilizando o modelo LWPR atualizado apenas com instâncias das classes não conhecidas (coluna 2) e utilizando instâncias das 10 classes (coluna 3).

	Atualização com 4 classes	Atualização com 10 classes
Classificações Corretas	640 (89.3%)	691 (96.4%)
Classificações Incorretas	77 (10.7%)	26 (3.6%)
Acurácia	97.85%	99.00%
Precisão	93.26%	97.25%
Sensitividade	89.26%	96.37%

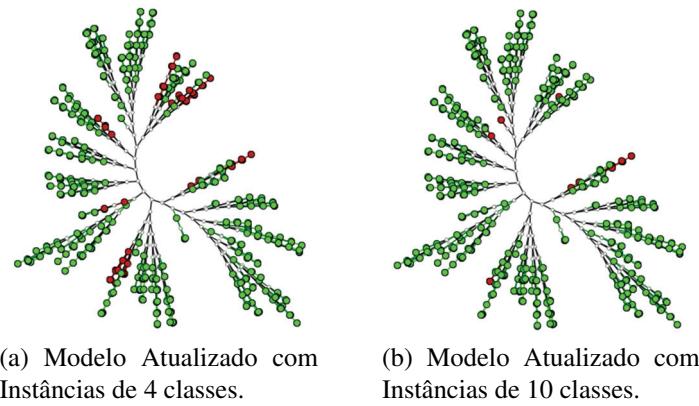


Figura 5.8: Comparação entre árvores *Class Matching* dos resultados da classificação da coleção ETHZ-Reduced717 utilizando o modelo LWPR atualizado apenas com instâncias das classes não conhecidas (5.8a) e utilizando instâncias das 10 classes (5.8b).

	4	5	7	8	10	13	16	23	25	26
4	16	20	0	0	0	0	0	0	0	0
5	0	29	0	0	0	0	0	0	0	0
7	0	0	42	0	0	0	0	0	0	0
8	0	0	0	27	0	0	0	0	0	0
10	0	0	0	0	47	0	0	0	0	0
13	0	0	0	0	0	47	0	0	0	0
16	0	0	0	0	0	0	72	0	0	0
23	0	8	0	3	0	3	0	197	0	0
25	0	0	0	13	0	0	2	1	117	0
26	0	6	0	1	0	13	2	0	5	46

Figura 5.9: Matriz de confusão do resultado da classificação da coleção ETHZ-Reduced717 utilizando o modelo LWPR atualizado apenas com instâncias das classes não conhecidas.

5.3.5 Alteração da Perspectiva de Classificação

Em diversas situações, usuários diferentes tem necessidades diferentes no processo de análise de uma coleção, de forma que o esquema de rotulamento utilizado pode não atender a todas essas necessidades. Dessa forma, um esquema de rotulamento diferente pode ser necessário, criando uma nova maneira de categorizar essa coleção. Para as diversas maneiras de categorização será utilizado neste texto o termo **perspectiva de classificação**. Nesse sentido, este estudo de caso tem o objetivo de verificar se o *layout* pode auxiliar na alteração da perspectiva de um classificador, através da alteração da perspectiva das instâncias de treinamento, adequando esse classificador a diferentes necessidades dos usuários. O experimento consistiu em verificar o desempenho da classificação de uma coleção, utilizando um modelo criado a partir de um conjunto de treinamento com as classes seguindo a perspectiva original da coleção, e comparar esse desempenho com o de uma classificação utilizando outro modelo criado a partir de um conjunto de treinamento com outra perspectiva, criado através do rotulamento das instâncias em novas classes.

A Tabela 5.14 mostra o resultado da classificação da coleção ETHZ-Reduced, utilizando um modelo criado a partir de um conjunto de treinamento com 84 instâncias (3 de cada classe) e rotulamento original. O novo rotulamento das instâncias desse conjunto de treinamento foi realizado utilizando a funcionalidade de rotulamento manual por seleção e a funcionalidade de rotulamento individual por conteúdo. O resultado final é um novo conjunto de treinamento, com 9 classes.

Como a mudança de perspectiva foi realizada utilizando a funcionalidade de rotulamento, foi necessário determinar uma regra para mapear as classes antigas nas classes novas, de forma a criar um novo *ground truth* para a coleção ETHZ-Reduced, e assim avaliar o processo de classificação. Esse mapeamento foi feito de acordo com a distribuição, no conjunto de treinamento, dos mapeamentos das instâncias das classes antigas para cada classe nova. A classe nova c_1 com o maior número de mapeamentos de uma classe antiga c_0 , no conjunto de treinamento, passou a ser a classe de todas as instâncias dessa classe c_0 no conjunto de teste. O conjunto de regras de mapeamento entre classes antigas e classes novas é mostrado na Tabela 5.13.

Tabela 5.13: Regras de transformação para instâncias da coleção ETHZ-Reduced, das classes da perspectiva antiga para as da perspectiva nova.

Classe Nova	Classes Antigas
1	2, 10, 20, 22
2	3, 12, 15
3	9, 14
4	6, 17
5	7, 16, 19, 21, 26
6	27
7	0, 16
8	1, 4, 5, 13, 23
9	8, 11, 18, 24, 25

A Tabela 5.14 mostra o resultado da classificação da coleção ETHZ-Reduced utilizando o conjunto de treinamento com a nova perspectiva. Os números observados são inferiores àqueles obtidos utilizando o conjunto de treinamento original, o que é esperado para essa coleção, pois o rotulamento original reflete com fidelidade as características das imagens, e essas características são bem definidas, no sentido em que não dão margem para muitas perspectivas diferentes. No entanto, os resultados são satisfatórios, e mostram que mesmo de uma coleção bem definida em termos de características, é possível extrair conhecimento sob outras perspectivas.

A perspectiva criada foi aplicada ao conjunto ETHZ original, com 3963 dimensões. A coleção teve então sua dimensionalidade reduzida utilizando a técnica PLS, seguindo o mesmo procedimento descrito na Seção 5.3, mas com o novo rotulamento. Da coleção com dimensionalidade reduzida foram extraídas as mesmas instâncias utilizadas no experimento original, para a construção do modelo LWPR, que foi utilizado para classificar o restante das instâncias. O resultado

dessa classificação é apresentado na Tabela 5.14, quarta coluna, sendo semelhantes aos obtidos no primeiro experimento.

As funcionalidades de rotulamento manual, aliadas ao *layout* criado pela árvore NJ exercearam um papel fundamental na escolha de quais instâncias pertenceriam a quais novas classes, pois possibilitaram a visualização de uma estrutura que destaca outros possíveis relacionamentos entre as instâncias, não descritos na perspectiva original. Em uma coleção na qual as características das instâncias permitem diversas perspectivas, o *layout* se torna uma ferramenta importante para destacar os relacionamentos presentes em cada uma delas, permitindo diversas classificações. Um exemplo disso é a coleção ALL-Reduced, cujos resultados são mostrados na Tabela 5.15. A nova perspectiva criada para essa coleção possui 6 classes, e o conjunto de treinamento é composto de 45 instâncias. Utilizando a nova perspectiva, foram obtidos resultados melhores do que utilizando a perspectiva original, o que mostra que uma estrutura adicional foi revelada, estrutura essa que dificilmente seria detectada sem a utilização do *layout*.

Tabela 5.14: Comparação dos resultados da classificação utilizando o modelo LWPR com perspectiva original e a nova perspectiva, para a coleção ETHZ-Reduced.

	Perspectiva Original	Nova Perspectiva	Nova Perspectiva Dimensões Originais
Classificações Corretas	1709 (88.3%)	1532 (79.2%)	1581 (81.7%)
Classificações Incorretas	226 (11.7%)	403 (20.8%)	354 (18.3%)
Acurácia	98.49%	94.38%	95.15%
Precisão	89.21%	82.36%	84.5%
Sensitividade	88.32%	79.17%	81.71%

Tabela 5.15: Comparação dos resultados da classificação utilizando o modelo LWPR com perspectiva original e a nova perspectiva, para a coleção ALL-Reduced.

	Perspectiva Original	Nova Perspectiva
Classificações Corretas	1875 (67.7%)	1920 (69.3%)
Classificações Incorretas	894 (32.3%)	849 (30.7%)
Acurácia	86.61%	84.12%
Precisão	71.98%	72.36%
Sensitividade	67.71%	69.34%

A seguir é descrito um sistema computacional que contempla um conjunto de funcionalidades que permitem que as tarefas descritas anteriormente sejam realizadas.

5.4 Sistema de Classificação Visual de Imagens

A principal vantagem na adoção de um processo visual de classificação é que o usuário é imediatamente exposto a todo o resultado desse processo, sendo possível visualizar os falsos positivos, falsos negativos, além do posicionamento das instâncias classificadas, de forma que o *layout* ajuda a compreender esses resultados. Além disso, esse *layout* permite uma interação mais intuitiva, possibilitando a intervenção fácil no processo. Tal intervenção seria extremamente difícil se o usuário fosse exposto a apenas dados numéricos e estatísticos a respeito dos resultados da classificação.

Dentro do contexto deste projeto de doutorado, foi desenvolvido um sistema de classificação visual de imagens, contendo um conjunto de funcionalidades que, aliadas às visualizações, proporcionam a inserção do usuário no processo, através da construção de conjuntos de treinamento ou do ajuste dos mesmos com o intuito de refletir as necessidades do problema. Além disso, é possível realizar ajustes nos resultados da classificação, convergindo para os resultados ideais em determinado cenário. O sistema utiliza as abordagens LWPR apresentadas na Seção 5.2.2, e permite que atualizações sejam realizadas nos modelos, em situações nas quais uma evolução na estrutura das classes do problema é detectada. Além de tarefas de apoio à criação e atualização de modelos de classificação, o sistema fornece diversas funcionalidades, apoiadas por interfaces visuais, que permitem a realização de atividades acessórias ao processo de classificação, tais como a seleção de instâncias ou grupos de instância, de forma a inspecionar seu conteúdo, exclusão de instâncias do *layout*, normalização dos valores das dimensões, além do armazenamento de subconjuntos de instâncias para posterior utilização. O sistema oferece também a possibilidade de visualizar as coleções utilizando a técnica de projeção *Least Squares Projection* (LSP), em alternativa às árvores NJ. A seguir, as funcionalidades diretamente relacionadas ao processo de classificação são descritas.

5.4.1 Rotulamento por Seleção

A funcionalidade de rotulamento por seleção tem como objetivo criar rótulos para coleções não rotuladas, ou de redefinir um rotulamento prévio. A visualização torna-se útil, nesse momento, para direcionar a escolha dos rótulos. Um *layout* produzido por uma técnica de projeção multidimensional, por exemplo, na qual grupos de instâncias similares são idealmente representadas por pontos próximos no plano, pode facilitar a definição de grupos por parte do usuário. As árvores de similaridade também realizam bem esse papel, produzindo *layouts* que organizam os grupos de instâncias em ramos. Quando o processo de rotulamento por seleção é iniciado, uma nova configuração de classes é criada para representar os rótulos criados. A cada rótulo criado pelo usuário, através da seleção de uma instância ou grupo de instâncias, é associada uma cor, para facilitar o processo. Sempre que o usuário move seu foco de um grupo para outro, os rótulos são

recoloridos, para manter a consistência. O primeiro rótulo criado é sempre colorido com a cor inicial do intervalo de cores utilizado, e o último sempre com a cor final desse intervalo. Podem ser criados quantos rótulos forem necessários.

A Figura 5.10 mostra a projeção LSP de uma coleção não rotulada com 645 instâncias. Os grupos formados nesse *layout* podem indicar a formação de 7 grupos distintos, sendo rotulados com uma seleção simples.

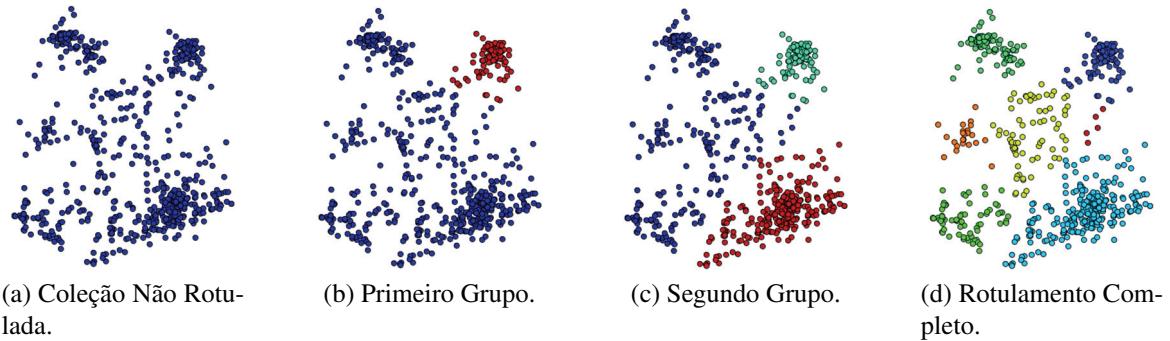


Figura 5.10: Processo de rotulamento manual de uma coleção de dados representada por uma projeção LSP, utilizando a funcionalidade de rotulamento por seleção.

Utilizando um *layout* baseado em uma árvore NJ, a distribuição das instâncias em ramos também pode facilitar a detecção e definição dos rótulos da coleção. Os ramos podem ser rotulados através da seleção do nó situado no nível mais alto desse ramo, e outros grupos podem ser rotulados através de uma operação de seleção livre. A Figura 5.11 mostra o processo de rotulamento da mesma coleção mostrada na Figura 5.10, porém utilizando uma árvore NJ. É possível perceber que a disposição dos ramos facilita consideravelmente o processo de rotulamento, pois destaca os relacionamentos entre as instâncias, representado pelas conexões entre elas, permitindo concluir com facilidade uma possível configuração de grupos para a coleção.

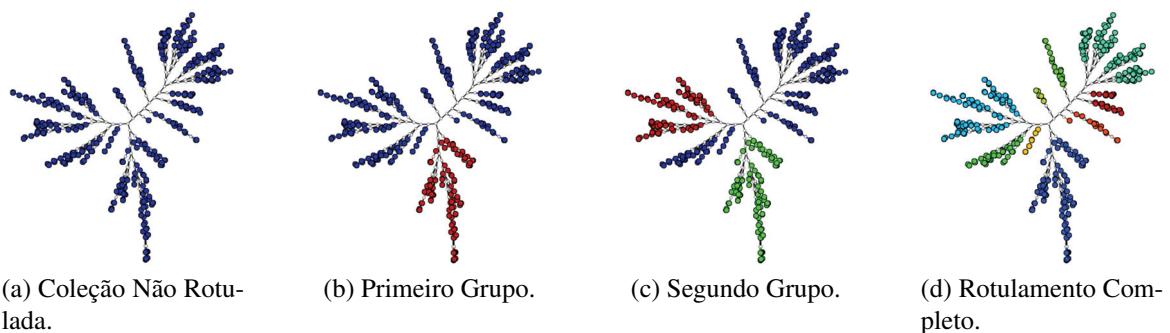


Figura 5.11: Processo de rotulamento de uma coleção de dados representada por uma árvore NJ, utilizando a funcionalidade de rotulamento por seleção.

5.4.2 Rotulamento Individual

A funcionalidade de rotulamento por seleção é útil na definição de rótulos iniciais para uma coleção. Entretanto, em algumas situações, pode ser necessário definir rótulos diferentes para instâncias em um mesmo grupo, realizando ajustes finos no processo anterior. Além disso, é possível que instâncias pertencentes a grupos diferentes troquem rótulos durante a convergência do processo.

Para contemplar as situações descritas acima, bem como outras eventuais, foi desenvolvido uma funcionalidade que permite o rotulamento individual de imagens, através da análise de seu conteúdo. O usuário, ao selecionar um grupo de imagens, visualiza uma lista com os seus conteúdos, na qual é possível determinar o rótulo de uma ou mais imagens. A visualização do conteúdo permite que um rotulamento mais detalhado seja realizado, em complemento ao rotulamento por seleção. Além disso, permite que ajustes sejam feitos em coleções previamente rotuladas, de forma a corrigir eventuais resultados de processos de classificação automática. Finalmente, é possível alterar a perspectiva de visão de uma certa coleção, através da formação de rótulos que contemplem essa perspectiva.

A Figura 5.12 ilustra o processo de rotulamento individual de imagens. Depois de selecionar um ramo específico em uma árvore, é possível alterar os rótulos livremente das imagens contidas nessa seleção (5.12b). Novos rótulos são definidos através da escolha de novos nomes na tabela. Caso seja escolhido um nome já existente, o rótulo correspondente será associado a essa imagem. Na figura, o nome das classes é alterado para “elephants”, que assumem a cor vermelho escuro, como pode ser visto na moldura das imagens mostradas na Figura 5.12b. O ramo da árvore (5.12a) mostra as novas cores das instâncias após o procedimento de rotulamento.

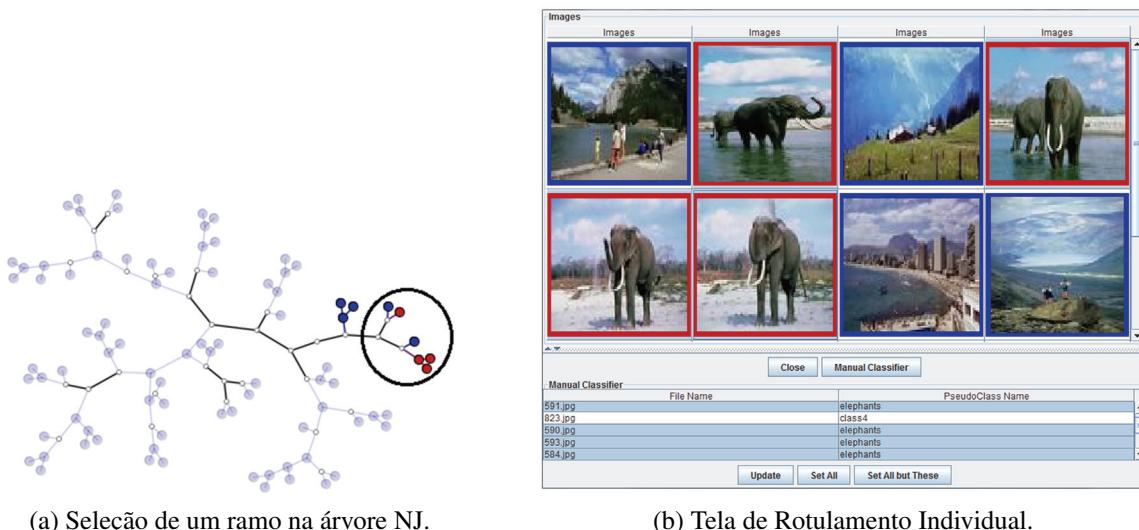


Figura 5.12: Processo de rotulamento individual aplicado à seleção de um ramo de imagens em uma árvore NJ.

Quando o procedimento de rotulamento termina, utilizando qualquer uma das funcionalidades descritas, é possível salvar a coleção de imagens com a nova configuração de rótulos produzida, de forma que essa coleção seja utilizada como um conjunto de treinamento para um classificador automático, ou para avaliar outros processos de classificação aplicados a essa coleção.

5.4.3 **Class Matching**

Em situações nas quais existe um *ground truth* de determinada coleção, ou seja, os rótulos ideais de cada instância são conhecidos, é possível avaliar os resultados apresentados pelo classificador, sendo possível analisar o número de instâncias classificadas corretamente, além de valores como acurácia, precisão e sensitividade. É possível também visualizar a matriz de confusão associada, e analisar a distribuição do rotulamento nas classes do problema. Técnicas de visualização podem ser usadas para criar uma representação visual adicional que possibilita uma análise dos indivíduos para os quais a classificação falhou. A funcionalidade **Class Matching**, ilustrada nos resultados anteriores, foi implementada para esse fim. Nesse *layout*, as instâncias corretamente classificadas são representadas por círculos verdes, e as instâncias incorretamente classificadas representadas por círculos vermelhos. É possível, por exemplo, identificar quais instâncias foram classificadas incorretamente. Utilizando projeções multidimensionais, pode-se analisar a região na qual os pontos se situam no *layout*, juntamente com a sua vizinhança, e compreender como o relacionamento entre as instâncias influenciou o rotulamento obtido. Em árvores de similaridade, as arestas que conectam os nós da árvore, bem como a vizinhança e hierarquia produzida podem ajudar a entender as razões pelas quais as instâncias foram classificadas de determinada maneira. Finalmente, a visualização do conteúdo das imagens também pode ajudar a identificar quais características visuais influenciaram nas decisões tomadas pelo classificador.

A Figura 5.13 mostra o processo de análise dos resultados obtidos em uma classificação, utilizando o **Class Matching** e uma árvore NJ. Na figura, estão representados o *ground truth* da coleção (5.13a), o resultado obtido pelo classificador (5.13b) e a visualização de erros e acertos (5.13c). Na figura é possível ver que tal funcionalidade permite acesso imediato às instâncias classificadas corretamente e incorretamente, permitindo que o usuário possa compreender de maneira fácil e eficaz como o modelo de classificação atuou na coleção. O sistema permite a carga das imagens para uma melhor verificação.

O processo de **Class Matching** pode ser útil também para apoiar eventuais ajustes de etiquetação inspirados pelo processo de classificação.

5.4.4 **Descrição do Sistema**

O sistema de classificação visual proposto neste trabalho tem o objetivo de criar um ambiente que contemple todo o processo de classificação de coleções de imagem, apoiado por técnicas de

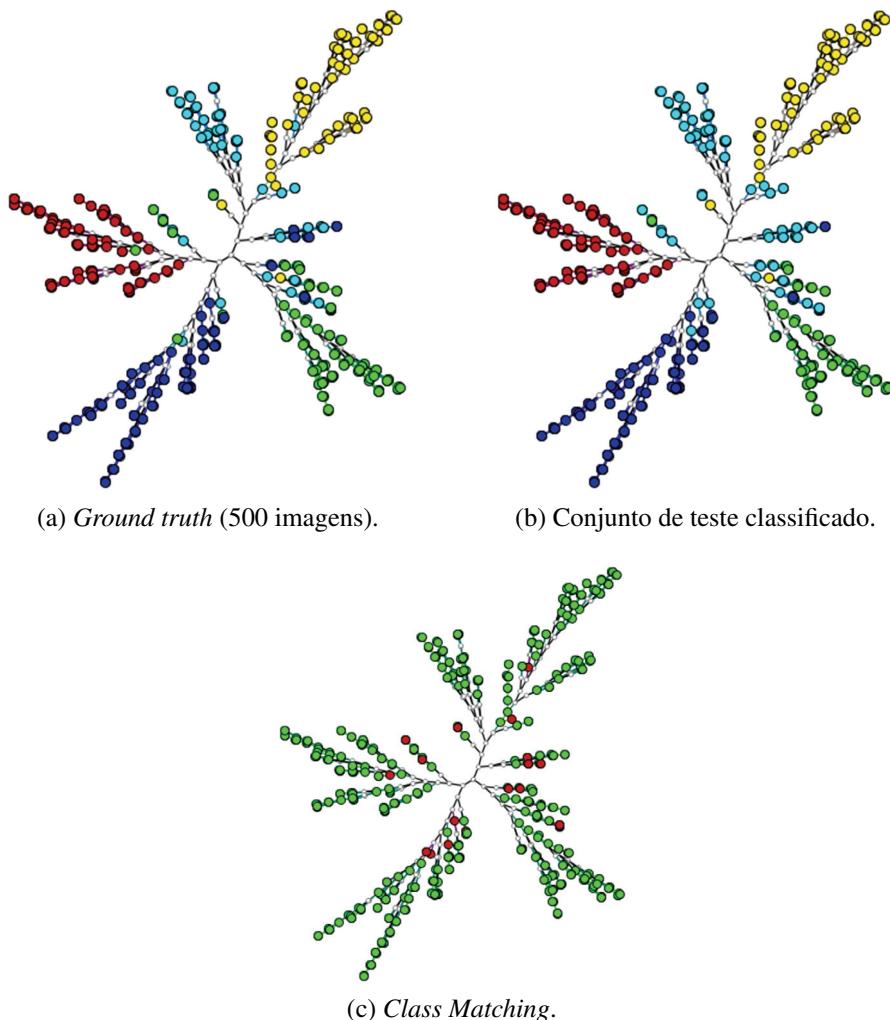


Figura 5.13: Sequência de passos do processo de classificação visual.

visualização. O usuário pode selecionar instâncias de treinamento para criar modelos LWPR, bem como analisar o resultado da classificação, de forma a compreender as razões de eventuais falhas. Finalmente, o usuário pode ajustar o modelo LWPR utilizando as instâncias sendo visualizadas, em uma sequência de passos que levem aos resultados de classificação esperados. É importante ressaltar que o sistema também oferece suporte para criação e aplicação de modelos de classificação *Support Vector Machines* (SVM) e *Partial Least Squares* (PLS), bem como a utilização da técnica *K-Nearest Neighbors* (KNN), em alternativa à técnica LWPR.

A Figura 5.14 mostra a tela principal do sistema, de onde o usuário poderá interagir com todo o processo. O sistema recebe como entrada uma matriz de instâncias multidimensionais, rotuladas ou não. Através de um *layout* produzido por uma árvore NJ ou uma projeção multidimensional LSP, o usuário pode criar um conjunto de treinamento selecionando as instâncias que achar mais significativas, de acordo com determinada perspectiva. Utilizando as funcionalidades apresentadas neste capítulo, é possível criar esquemas de rotulamento, no caso de coleções não rotuladas, ou

ajustar os rótulos existentes de acordo com a necessidade. As instâncias escolhidas para compor o conjunto de treinamento podem ser salvas em um arquivo, ou utilizadas para criar um ou um conjunto de modelos LWPR.

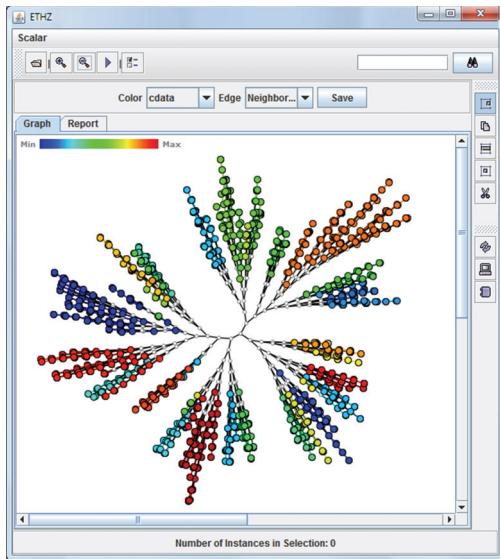


Figura 5.14: Tela principal do sistema de classificação visual de imagens.

De posse de um conjunto de treinamento ou de modelo(s) LWPR, o usuário pode então classificar uma coleção de imagens. Os resultados dessa classificação serão exibidos utilizando também a visualização escolhida.

Após a análise dos resultados da classificação, seja pela verificação do conteúdo das instâncias, seja pela análise do *ground truth* utilizando o ***Class Matching***, o usuário pode selecionar um conjunto de instâncias que será utilizado na atualização do modelo LWPR utilizado na classificação atual. Diversas estratégias podem ser utilizadas. O usuário pode, por exemplo, utilizar um conjunto de instâncias classificadas de forma incorreta, e que estejam em posições estratégicas do *layout* para alimentar o modelo existente. Tais instâncias podem ajudar a corrigir determinada tendência do classificador que possa estar contribuindo para um desempenho ruim na classificação. O modelo atualizado poderá então ser novamente utilizado para classificar a coleção.

De forma resumida, as principais funcionalidades do sistema são:

- Visualização de coleções de dados, utilizando árvores NJ ou projeções multidimensionais LSP;
- Criação/alteração de rótulos das instâncias;
- Criação de modelos de classificação, utilizando as técnicas LWPR, PLS e SVM;
- Aplicação dos modelos de classificação criados utilizando as técnicas LWPR, PLS, SVM e KNN;

- Atualização dos modelos de classificação LWPR, PLS e SVM;
- Análise numérica e visual dos resultados da classificação.

Os passos descritos anteriormente compõem um processo iterativo de classificação visual, baseado no ajuste progressivo do modelo de classificação, e na agregação de um conhecimento mais direcionado sobre o problema pelo modelo em cada passo, aumentando as chances de produzir um classificador mais eficaz.

Todas as funcionalidades descritas neste capítulo foram adicionadas ao sistema VisPipeline. O sistema de classificação visual, as API's com as funcionalidades inseridas e informações adicionais estão disponíveis na página do VICG², em <http://vicg.icmc.usp.br/infovis2/Tools>.

5.5 Considerações Finais

Este capítulo apresentou uma metodologia de classificação visual baseada no algoritmo *Locally Weighted Projection Regression* (LWPR) e em técnicas de visualização, com o objetivo de permitir a inserção do usuário na classificação incremental de um conjunto de dados. Essa inserção ocorre pela construção e ajuste de modelos LWPR em um processo iterativo com rápida convergência para resultados desejados. LWPR produz uma regressão não linear em espaços de alta dimensionalidade, utilizando um conjunto de modelos lineares PLS locais que abrangem um pequeno número de regressões univariadas em direções específicas do espaço original.

Os resultados de diversos experimentos realizados mostram que a inserção do usuário utilizando técnicas de visualização apresenta grande potencial em produzir um classificador eficaz. As árvores de similaridade em particular mostraram uma estrutura na qual a hierarquia formada apresenta detalhes das coleções que não aparecem tão claramente em *layouts* produzidos por outras abordagens, permitindo que o usuário enxergue com facilidade quais escolhas serão representativas na construção dos modelos.

As árvores de similaridade também permitiram uma análise eficaz dos resultados de uma classificação. Utilizando a ferramenta **Class Matching**, desenvolvida neste trabalho, é possível tirar diversas conclusões pela análise detalhada dos ramos. Esses ramos podem sugerir motivos pelos quais as instâncias receberam determinados rótulos durante a classificação, pois são construídos baseados em determinada medida de similaridade entre essas instâncias.

Os modelos criados podem ser utilizados em outras coleções que compartilhem o mesmo espaço de características daquela utilizada para sua criação. A capacidade de apoio visual ao treinamento incremental permite que ajustes sejam realizados rapidamente para acomodar evoluções na distribuição das classes. A metodologia também permite que os modelos tratem alterações mais

²VICG: grupo de Visualização, Imagens e Computação Gráfica (VICG) do ICMC/USP

drásticas, caracterizadas pelo aparecimento de novas classes. Nesses casos, o posicionamento das instâncias no *layout* fornece com clareza os sinais dessa alteração, indicando que atualizações são necessárias no modelo, apoiando também a execução dessas atualizações.

Finalmente, é possível realizar a classificação de uma mesma coleção de dados utilizando diferentes perspectivas, e a visualização funciona aqui mais uma vez como um guia eficaz para destacá-las. De acordo com a medida de similaridade utilizada, ou do conjunto de descritores que representam cada instância, é possível obter representações que destacam diferentes tendências, auxiliando o usuário a construir modelos que refletem as perspectivas que tais tendências possam representar, produzindo classificadores que se adaptam a diferentes necessidades dos usuários.

Um sistema de classificação visual incremental foi desenvolvido para apoiar as tarefas de criação e alteração de uma rotulação, além de permitir a criação, aplicação e atualização de modelos incrementais de classificação baseados em LWPR, criação e aplicação de modelos SVM e PLS, e a execução de classificações baseadas em KNN. Finalmente, o sistema permite uma análise detalhada dos resultados de uma classificação.

O próximo capítulo sumariza as contribuições e resultados obtidos pelas técnicas e metodologias apresentadas, além de apresentar as conclusões obtidas no trabalho descrito nesta tese, bem como trabalhos futuros.

Conclusões

6.1 Contribuições

Esta tese apresentou uma metodologia de classificação de dados utilizando técnicas de visualização, chamada **classificação visual de dados**. As técnicas de classificação existentes possuem uma natureza individual, e cada cenário pode exigir uma configuração diferente para o classificador, bem como uma representação diferente para as instâncias utilizadas no processo. O usuário, possuindo conhecimento a respeito do problema, torna-se um elemento importante para auxiliar na definição das configurações que levem à obtenção dos resultados desejados. Dessa forma, a metodologia proposta tem o objetivo de promover a inserção do usuário no processo de classificação automática de forma a garantir uma convergência rápida para os resultados esperados, em diferentes cenários. No entanto, é necessário que as coleções sejam apresentadas de forma amigável e efetiva, sob o risco de prejudicar a compreensão das informações nelas contidas e impossibilitar o sucesso no processo. Nesse sentido, as técnicas de visualização exercem um papel fundamental na produção de representações gráficas dos dados, que facilitam o entendimento da estrutura particular de cada coleção, auxiliando inclusive na revelação de informações ocultas a respeito desses dados e aumentando o poder de compreensão do usuário.

Diversas técnicas de visualização de dados multidimensionais foram analisadas, de forma a garantir um *layout* que auxiliasse a análise visual do usuário. De acordo com Paulovich (2008), alguns aspectos são cruciais para que esses *layouts* sejam efetivos para essa tarefa. O primeiro deles é o compromisso entre complexidade computacional e precisão, de forma a aplicar a técnica

em grandes coleções de dados mantendo os relacionamentos observados no espaço original de características, mesmo para relações não-lineares entre as instâncias. Além disso, a escalabilidade visual também deve ser levada em conta, de forma a garantir uma representação sintetizada da coleção respeitando os limites do dispositivo de visualização. Finalmente, deve haver a preservação de informações globais e locais, de forma que explorações em diferentes níveis possa ser realizada da mesma maneira.

Apesar de as projeções multidimensionais oferecerem diversas vantagens para a exploração de coleções de dados, algumas de suas limitações podem prejudicar uma análise mais detalhada, comprometendo o processo de classificação. Na maioria das situações, o nível de precisão global apresentado pelo *layout* não é mantido quando uma análise local é realizada. Além disso, ocorre um alto grau de sobreposição de instâncias e consequente confusão visual, dificultando a realização de diversas tarefas, como a definição da densidade visual dos grupos formados, por exemplo. Para amenizar tais problemas, foram investigados os *layouts* produzidos por árvores de similaridade, utilizando a técnica ***Neighbor Joining*** (NJ). As relações de similaridade entre as instâncias nas árvores são organizadas de forma que uma hierarquia natural é observada, o que possibilita a exploração da coleção em níveis, para os quais a análise é realizada utilizando praticamente a mesma estratégia. Isso garante que a precisão do *layout* seja mantida em análises globais e locais.

Algumas melhorias no algoritmo NJ foram investigadas e implementadas durante a execução deste projeto de doutorado (Capítulo 3). A primeira delas consiste em um procedimento de promoção de nós, e visa amenizar o problema do alto volume de pontos presentes no *layout* produzido pela técnica. Esse procedimento baseia-se na análise de determinados padrões observados no *layout*, e na transformação de nós-folha, que representam as instâncias da coleção, em nós internos, substituindo parte dos nós virtuais. Como não existe nenhuma informação de hierarquia explícita nas instâncias da coleção, a mudança no nível dos nós-folha não provoca perdas significativas na precisão do *layout*. O resultado obtido é um *layout* menos poluído, em média com 51% menos nós virtuais, que mantém as capacidades de organização e exploração das árvores NJ e garante um melhor aproveitamento do espaço de visualização. Outra limitação do algoritmo NJ é o alto custo computacional para a geração da estrutura da árvore. Para amenizar esse problema, dois algoritmos foram investigados, ***Rapid Neighbor Joining*** (Rapid NJ) e ***Fast Neighbor Joining*** (Fast NJ), que tentam diminuir o numero de buscas necessárias para a construção dos ramos da árvore, através da utilização de estruturas de dados especializadas ou de heurísticas que produzem uma aproximação da árvore NJ original. Os tempos de geração das árvores utilizando esses algoritmos foi consideravelmente menor do que utilizando o algoritmo original, e inclusive menor do que utilizando algumas técnicas de projeção multidimensional rápidas, tais como LSP. O *layout* aproximado produzido pelo algoritmo Fast NJ apresentou pouca diferença em relação ao *layout* original, em termos de precisão.

Aliado a investigação sobre técnicas de visualização adequadas para auxiliar na classificação visual de dados, uma metodologia de redução de dimensionalidade semi-supervisionada, baseada na técnica **Partial Least Squares** (PLS), foi desenvolvida (Capítulo 4). PLS utiliza um conjunto de amostras previamente rotuladas para construir um modelo, empregado na redução de dimensionalidade de uma coleção que compartilhe os mesmos atributos, ou atributos semelhantes. A ideia é encontrar, em cada dimensão das amostras utilizadas, informações que expliquem as classes com as quais elas foram rotuladas, e usar essas informações como novas dimensões. Como o conjunto de amostras pode ser informado pelo usuário, seu conhecimento e perspectiva sobre o problema são inseridos no processo, que se ajusta às suas necessidades. Além disso, a metodologia também contempla a redução de dimensionalidade de coleções sem informação previa de classes ou seja, quando não existe um conjunto de amostras previamente rotuladas, utilizando um procedimento de agrupamento automático e amostragem. As coleções de dados reduzidas obtidas pelo processo ressaltam as diferenças entre as classes ou grupos presentes. As árvores NJ construídas com essas coleções reduzidas mostram uma discriminação entre grupos que facilita a escolha de instâncias representativas, se mostrando consideravelmente útil para o processo de classificação visual. Finalmente, a reutilização dos modelos, cuja carga se mostrou rápida, permite que o mapeamento de coleções em crescimento seja realizado.

Diante dos procedimentos desenvolvidos para representação das coleções, uma metodologia de classificação visual incremental foi criada (Capítulo 5). Essa metodologia utiliza como base o algoritmo *Locally Weighted Projection Regression* (LWPR), associado a ferramentas interativas e a *layouts* produzidos por árvores NJ. LWPR utiliza um modelo de regressão não-linear composto por um conjunto dinâmico de modelos PLS locais, chamados de **campos de receptividade**, construídos de forma incremental. O usuário participa do processo de classificação através da seleção de instâncias em um *layout* de visualização, que serão utilizadas para criar os modelos LWPR. Além disso, é possível atualizar os modelos criados adicionando informações de outras instâncias, também selecionadas pelo usuário no *layout*, permitindo que o classificador seja ajustado e aprenda com as necessidades de cada situação. Um conjunto de ferramentas de rotulamento manual foram desenvolvidas para permitir que o usuário defina um esquema de rotulamento a ser utilizado em um conjunto de treinamento, ou modifique um esquema já existente, de forma a ajustar os resultados de determinada classificação. Uma ferramenta chamada **Class Matching** foi desenvolvida para permitir a análise visual dos resultados da classificação, nos casos em que um *ground truth* da coleção é conhecido. Juntamente com informações estatísticas a respeito do processo, tais como precisão e acurácia, essa ferramenta utiliza um *layout* de visualização para destacar a posição das instâncias classificadas de forma correta e de forma errada.

As árvores NJ exerceram um papel crucial na seleção de instâncias, bem como no rotulamento manual, devido a seus ramos serem construídos de acordo com a similaridade entre as instâncias, o que direciona mais rapidamente a exploração dos grupos. Além disso, o *layout* associado ao **Class**

Matching possibilitou uma compreensão consideravelmente mais rápida dos motivos pelos quais as instâncias receberam determinados rótulos durante a classificação. O usuário também consegue detectar mais facilmente para quais classes o classificador se mostrou deficiente, o que direciona a escolha de instâncias para a atualização do modelo.

Os modelos LWPR podem ser reutilizados em outras coleções, e a capacidade de treinamento incremental permite que eles sejam ajustados para acomodar evoluções na distribuição das classes, inclusive com o aparecimento de novas classes. O posicionamento das instâncias no *layout* fornece com clareza os sinais dessa alteração, indicando quais atualizações são necessárias. Finalmente, é possível criar diferentes modelos para diferentes perspectivas, permitindo a criação de classificadores que atuem na mesma coleção, mas atendendo a diferentes necessidades dos usuários. Nesse caso, o *layout* de visualização funciona como uma forma de destacar as tendências características de cada uma dessas perspectivas.

Dessa forma, o processo de classificação visual pode ser visto como um procedimento iterativo, no qual o usuário parte de um conjunto previamente rotulado ou não, seleciona instâncias de treinamento para criar um modelo que classificará outras coleções de dados, modelo esse que será gradativamente atualizado em cada classificação até que os resultados sejam adequados às suas necessidades. Os *layouts* de visualização, em especial as árvores NJ, se mostraram importantes para guiar o usuário nessas atualizações, de forma que poucas iterações são necessárias para se obter bons resultados.

De forma a colocar em prática as metodologias propostas, dois sistemas computacionais foram desenvolvidos. O primeiro sistema implementa a redução de dimensionalidade utilizando modelos PLS. Nele o usuário, de posse de uma coleção de dados e um conjunto de amostras previamente rotulado, cria um modelo que será utilizado para reduzir a dimensionalidade dessa coleção. A coleção com dimensionalidade reduzida pode então ser visualizada utilizando qualquer técnica de visualização. O próprio sistema oferece a possibilidade de utilizar a técnica RadViz para visualizar a distribuição das instâncias nas dimensões reduzidas, sendo possível analisar a discriminação entre classes produzida pela técnica PLS. É possível também utilizar a abordagem semi-supervisionada em coleções sem conhecimento prévio a respeito de classes, para as quais não há um conjunto de amostras previamente rotuladas. O segundo sistema implementa a metodologia de classificação visual utilizando LWPR. O sistema permite ao usuário criar e atualizar os modelos de classificação através da seleção de instâncias em uma árvore NJ. Além disso, é possível criar esquemas de rotulamento para coleções não rotuladas, ou modificar um rotulamento prévio, ajustando o classificador para diversas necessidades. Os modelos criados podem ser utilizados na classificação de coleções, e os resultados dessa classificação podem ser analisados de forma detalhada utilizando a ferramenta **Class Matching**, determinando quais ajustes devem ser feitos no modelo. Ambos os sistemas estão disponíveis através do endereço <http://vicg.icmc.usp.br/infovis2/Tools>.

A principal contribuição trazida pelas metodologias propostas neste trabalho de doutorado residem na possibilidade de inserção do conhecimento do usuário na classificação de dados. Mesmo com os avanços no desenvolvimento de técnicas de classificação, bem como na representação dos dados a serem classificados, os classificadores existentes ainda são dependentes de diversos fatores inerentes a cada cenário. Nesse sentido, o usuário é peça fundamental na realização de ajustes que produzam resultados desejados mais rapidamente. Além disso, na maioria das situações, usuários diferentes possuem perspectivas diferentes sobre uma mesma coleção. O ajuste do classificador para diversas perspectivas se torna assim um benefício considerável para a classificação.

Outra contribuição deste trabalho é mostrar o papel fundamental das técnicas de visualização, em especial das árvores de similaridade, na comunicação entre usuário e sistema de classificação, no sentido de realçar diversas tendências e perspectivas, eventualmente ocultas, de determinada coleção. Os *layouts* produzidos conseguem fornecer pistas que justificam a tomada de decisão dos classificadores, de forma que o usuário fica ciente dos motivos pelos quais a classificação foi realizada de determinada maneira. Além disso, é possível realizar um monitoramento detalhado em coleções que evoluem, detectando alterações na distribuição de classes ou o aparecimento de novas classes.

6.2 Trabalhos Futuros

Após o desenvolvimento das metodologias propostas neste projeto de doutorado, foi possível detectar algumas limitações que representam possibilidades interessantes para trabalhos futuros.

Um ponto importante a ser tratado diz respeito a alguns problemas relacionados à aplicação de árvores de similaridade em coleções de médio e grande porte. Mesmo com a aplicação das melhorias desenvolvidas neste projeto, a geração dessas árvores para coleções com mais de 10.000 instâncias representa um processo de custo computacional extremamente alto. Assim, seria interessante a adaptação do algoritmo de construção da árvore para suportar essas coleções, priorizando consumo de memória e custo computacional adequados. Tais modificações podem levar em conta uma construção de árvore sob demanda, utilizando memória secundária para armazenar partes da árvore não visualizadas pelo usuário em determinado momento, ou a paralelização do processo de construção, de forma a agilizá-la.

A interação em árvores construídas para coleções com mais de 1500 instâncias também se mostra difícil, pois o espaço de visualização contempla, além dos pontos que representam as instâncias da coleção, as arestas e os nós virtuais, aumentando consideravelmente o volume do *layout* e produzindo confusão visual. Dessa forma, torna-se necessária a criação de um procedimento de exploração de ramos efetivo, que possibilite que o usuário consiga explorar diversos ramos específicos simultaneamente, de acordo com a necessidade, sem perder a visão de toda a coleção.

Outro ponto importante está relacionado à metodologia de classificação visual baseada em LWPR, em situações nas quais novas classes aparecem ou desaparecem no problema. Atualmente, é necessário armazenar todas as instâncias de treinamento utilizadas para criação e atualizações em determinado modelo. Como apresentado na Seção 5.2.1, os campos de receptividade que compõem um modelo LWPR definem a área de atuação respectiva de cada classe. Dessa forma, seria interessante investigar uma abordagem que analise o comportamento desses campos de receptividade de forma a detectar situações nas quais novas classes estão surgindo ou desaparecendo. Abordagens desse tipo poderiam inclusive fornecer uma análise mais detalhada a respeito desses campos de receptividade que auxiliasse no estudo do relacionamento entre as classes, ou a estrutura de determinada classe.

Finalmente, é interessante investigar outras formas de inserção do usuário no processo de classificação. Como detalhado no Capítulo 4, a metodologia de redução de dimensionalidade baseada em PLS auxilia na determinação de um conjunto de características que destaca a diferença entre as classes em determinado problema. Essa metodologia pode ser associada a uma ferramenta interativa para manualmente determinar quais características são mais determinantes para a inserção de instâncias em determinada classe, bem como realizar ajustes nesses valores para aumentar a separabilidade e melhorar ainda mais os resultados. Pode-se explorar também maneiras de inserir o usuário na alteração de parâmetros de um classificador, ou na alteração da medida de similaridade entre instâncias. As técnicas de visualização podem ser utilizadas para guiar o usuário em todas essas ferramentas, servindo como elemento fundamental de ligação entre usuário e sistema computacional.

Referências Bibliográficas

- ABDI, H.; WILLIAMS, L. J. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, v. 2, n. 4, p. 433–459, 2010.
Disponível em <http://dx.doi.org/10.1002/wics.101>
- ABELLO, J.; VAN HAM, F.; KRISHNAN, N. Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, v. 12, n. 5, p. 669–676, 2006.
- A.MUTHUKUMARAVEL; DR.S.PURUSHOTHAMAN; DR.A.JOTHI Implementation of locally weighted projection regression network for concurrency control in computer aided design. *International Journal of Advanced Computer Science and Applications (IJACSA)*, v. 2, p. 46–50, 2011.
- ANKERST, M. Visual data mining with pixel-oriented visualization techniques. In: *Proceedings of ACM SIGKDD Workshop on Visual Data Mining*, 2001.
- ANTONIE, M. L.; ZAIANE, O. R.; COMAN, A. Application of data mining techniques for medical image classification. In: *Proceedings of Second Intl. Workshop on Multimedia Data Mining in conjunction with Seventh ACM SIGKDD*, 2001, p. 94–101.
- ARTERO, A. O.; DE OLIVEIRA, M. C. F.; LEVKOWITZ, H. Uncovering clusters in crowded parallel coordinates visualizations. In: *Proceedings of the IEEE Symposium on Information Visualization*, INFOVIS '04, Washington, DC, USA: IEEE Computer Society, 2004, p. 81–88 (INFOVIS '04,).
Disponível em <http://dx.doi.org/10.1109/INFOVIS.2004.68>
- ATKESON, C.; HALE, J.; POLICK, F.; RILEY, M.; KOTOSAKA, S.; SCHAUL, S.; SHIBATA, T.; TEVATIA, G.; UDE, A.; VIJAYAKUMAR, S.; KAWATO, E.; KAWATO, M. Using humanoid

- robots to study human behavior. *Intelligent Systems and their Applications, IEEE*, v. 15, n. 4, p. 46–56, 2000.
- BACHMAIER, C.; BRANDES, U.; SCHLIEPER, B. Drawing phylogenetic trees. In: *Algorithms and Computation*, 2005, p. 1110–1121 (*Lecture Notes in Computer Science*, v.3827).
- BASALAJ, W. Proximity visualisation of abstract data. Technical Report, Computer Laboratory - University of Cambridge, 2000.
- BASALAJ, W.; RODDEN, K.; SINCLAIR, D.; WOOD, K. Evaluating a visualization of image similarity as a tool for image browsing. In: PRESS, I. C., ed. *Proceedings of the IEEE Symposium on Information Visualization*, IEEE Computer Society, 1999, p. 36.
- BELKIN, M.; NIYOGI, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, v. 15, n. 6, p. 1373–1396, 2003.
- BLEI, D.; NG, A.; JORDAN, M. Latent dirichlet allocation. *The Journal of machine Learning Research*, v. 3, p. 993–1022, 2003.
- BOIMAN, O.; SHECHTMAN, E.; IRANI, M. In defense of nearest-neighbor based image classification. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, 2008.
- BORUVKA, O. O jistém problému minimálním. *Práce Mor. Prírooved. Spol v Brne (Acta Societ. Scient. Natur. Moravicae)*, v. 3, p. 37–58, 1926.
- BOULESTEIX, A.-L.; STRIMMER, K. Partial least squares: A versatile tool for the analysis of high-dimensional genomic data. *Briefings in Bioinformatics*, v. 8, n. 1, p. 32–44, 2007.
- BRANDES, U.; PICH, C. Eigensolver methods for progressive multidimensional scaling of large data. In: *Lecture Notes on Computer Science*, v. 4372, Springer, p. 42–53, 2007.
- BROADHURST, D.; GOODACRE, R.; JONES, A.; ROWLAND, J.; KELL, D. Genetic algorithms as a method for variable selection in multiple linear regression and partial least squares regression, with applications to pyrolysis mass spectrometry. *Analytica Chimica Acta*, v. 348, n. 1–3, p. 71–86, 1997.
- CAMARGO, J.; CAICEDO, J.; GONZÁLEZ, F. Multimodal image collection visualization using non-negative matrix factorization. *Research and Advanced Technology for Digital Libraries*, p. 429–432, 2010.
- CAMARGO, J.; GONZÁLEZ, F. Visualization, summarization and exploration of large collections of images: State of the art. In: *Latin American Conference On Networked and Electronic Media, LACNEM*, Citeseer, 2009.

- CARD, S.; MACKINLAY, J.; SHNEIDERMAN, B. *Readings in information visualization: using vision to think.* Morgan Kaufmann, 1999.
- CARREIRA-PERPINAN, M. A. *Dimensionality reduction.* 1st ed. Chapman & Hall/CRC, 2011.
- CHANG, C.-Y.; WANG, H.-J.; LI, C.-F. Semantic analysis of real-world images using support vector machine. *Expert Systems with Applications*, v. 36, n. 7, p. 10560 – 10569, 2009.
Disponível em <http://www.sciencedirect.com/science/article/pii/S0957417409003066>
- CHEN, C.; GAGAUDAKIS, G.; ROSIN, P. Similarity-based image browsing. In: FOR INFORMATION PROCESSING, I. F., ed. *Proceedings of the 16th IFIP World Computer Congress*, Kluwer, 2000.
- CHEN, M. S.; HAN, J.; YU, P. S. Data mining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering*, v. 8, n. 6, p. 866 – 883, 1996.
- CHEN, Z.; DENTON, E.; ZWIGGELAAR, R. Local feature based mammographic tissue pattern modelling and breast density classification. In: *Biomedical Engineering and Informatics (BMEI), 2011 4th International Conference on*, 2011, p. 351 –355.
- CHENG, E.; XIE, N.; LING, H.; BAKIC, P. R.; MAIDMENT, A. D. A.; MEGALOOIKONOMOU, V. Mammographic image classification using histogram intersection. In: *Proceedings of IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2010.
- CIOCCHA, G.; CUSANO, C.; SANTINI, S.; SCHETTINI, R. Halfway through the semantic gap: Prosemantic features for image retrieval. *Information Sciences*, v. 181, n. 22, p. 4943 – 4958, 2011.
Disponível em <http://www.sciencedirect.com/science/article/pii/S002002551100332X>
- CIOCCHA, G.; CUSANO, C.; SCHETTINI, R. Semantic classification, low level features and relevance feedback for content-based image retrieval. In: *IS&T/SPIE Electronic Imaging*, International Society for Optics and Photonics, 2009, p. 72550D–72550D.
- CLEVELAND, W. S. *Visualizing data.* Hobart Press, 1993.
- COX, T. F.; COX, M. A. A. *Multidimensional scaling.* Second ed. Chapman & Hall/CRC, 2000.
- CUADROS, A. M.; PAULOVICH, F. V.; MINGHIM, R.; TELLES, G. P. Point placement by phylogenetic trees and its application for visual analysis of document collections. In: PRESS, I. C., ed. *IEEE Symposium on Visual Analytics Science and Technology*, 2007, p. 99–106.

- CUI, W.; ZHOU, H.; QU, H.; WONG, P.; LI, X. Geometry-based edge clustering for graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, v. 14, n. 6, p. 1277–1284, 2008.
- CVEK, U.; TRUTSCHL, M.; KILGORE, P.; STONE, R.; CLIFFORD, J. Multidimensional visualization techniques for microarray data. In: *Information Visualisation (IV), 2011 15th International Conference on*, 2011, p. 241 –246.
- DESELAERS, T.; PAREDES, R.; VIDAL, E.; NEY, H. Learning weighted distances for relevance feedback in image retrieval. In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, IEEE, 2008, p. 1–4.
- DO, T. Towards simple, easy to understand, an interactive decision tree algorithm. 2007.
- DOLOC-MIHU, A. Interactive visualization tool for analysis of large image databases. *Visual Analytics and Interactive Technologies: Data, Text, and Web Mining Applications*, p. 266, 2011.
- DOS SANTOS, J.; FERREIRA, C.; TORRES, R.; GONÇALVES, M.; LAMPARELLI, R. A relevance feedback method based on genetic programming for classification of remote sensing images. *Information Sciences*, v. 181, n. 13, p. 2671–2684, 2011.
- D'SOUZA, A.; VIJAYAKUMAR, S.; SCHAAL, S. Learning inverse kinematics. In: *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, 2001, p. 298 –303 vol.1.
- EADES, P. A. A heuristic for graph drawing. In: *Congressus Numerantium*, 1984, p. 149–160.
- EADES, P. A. Drawing free trees. *Bulletin of the Institute for Combinatorics and Its Applications*, v. 1, p. 10 – 36, 1992.
- EHRIG, H.; EHRIG, K.; PRANGE, U.; TAENTZER, G. *Fundamentals of algebraic graph transformation*. Springer, 2006.
- ELER, D. M. Múltiplas visões coordenadas para exploração de mapas de similaridade. PhD thesis, Instituto de Ciências Matemáticas e de Computação - ICMC - USP São Carlos, 2011.
- ELER, D. M.; NAKAZAKI, M. Y.; PAULOVICH, F. V.; SANTOS, D. P.; ANDERY, G. F.; OLIVEIRA, M. C. F.; NETO, J. B.; MINGHIM, R. Visual analysis of image collections. *The Visual Computer*, v. 25, n. 10, p. 923–937, 2009.
- ELER, D. M.; NAKAZAKI, M. Y.; PAULOVICH, F. V.; SANTOS, D. P.; OLIVEIRA, M. C. F.; NETO, J. E. S. B.; MINGHIM, R. Multidimensional visualization to support analysis of image collections. In: *Proceedings of the XXI Brazilian Symposium on Computer Graphics and Image Processing*, Washington, DC, USA: IEEE Computer Society, 2008, p. 289–296.

- ELIAS, I.; LAGERGREN, J. Fast neighbor joining. In: *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, 2005, p. 1263–1274.
- ELMQVIST, N.; FEKETE, J. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *Visualization and Computer Graphics, IEEE Transactions on*, v. 16, n. 3, p. 439–454, 2010.
- ERSOY, O.; HURTER, C.; PAULOVICH, F.; CANTAREIRO, G.; TELEA, A. Skeleton-based edge bundling for graph visualization. *Visualization and Computer Graphics, IEEE Transactions on*, v. 17, n. 12, p. 2364–2373, 2011.
- EVANS, J.; SHENEMAN, L.; FOSTER, J. Relaxed neighbor joining: A fast distance-based phylogenetic tree construction method. *Journal of Molecular Evolution*, v. 62, n. 6, p. 785–792, 2006.
- FALOUTSOS, C.; LIN, K. I. Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In: *Proceedings of the ACM SIGMOD international conference on Management of data*, New York, NY, USA: ACM, 1995, p. 163–174.
- FARIA, A.; BOTELHO, M. F.; CENTENO, J. A. S. Classificação de imagens de alta resolução integrando variáveis espectrais e forma utilizando redes neurais artificiais. In: *Anais do XI Simpósio Brasileiro de Sensoriamento Remoto*, 2003, p. 265–272.
- FAYYAD, U.; GRINSTEIN, G. G.; WIERSE, A., eds. *Information visualization in data mining and knowledge discovery*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.
- FLOREZ, J.; BELLOT, D.; MOREL, G. Lwpr-model based predictive force control for serial comanipulation in beating heart surgery. In: *Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on*, 2011, p. 320 –326.
- FOODY, G. M.; MATHUR, A. The use of small training sets containing mixed pixels for accurate hard image classification: Training on mixed spectral responses for classification by a svm. *Remote Sensing of Environment*, v. 103, n. 2, p. 179 – 189, 2006.
Disponível em <http://www.sciencedirect.com/science/article/pii/S0034425706001350>
- FUKUNAGA, K. *Introduction to statistical pattern recognition*. Academic Press, Inc., 1990.
- GANSNER, E.; HU, Y.; NORTH, S.; SCHEIDEGGER, C. Multilevel agglomerative edge bundling for visualizing large graphs. In: *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, IEEE, 2011, p. 187–194.

- GARTHWAITE, P. H. An interpretation of partial least squares. *Journal of the American Statistical Association*, v. 89, n. 425, p. 122–127, 1994.
- GEHLER, P.; NOWOZIN, S. On feature combination for multiclass object classification. In: *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, p. 221 –228.
- GELADI, P. Notes on the history and nature of partial least squares (pls) modelling. *Journal of Chemometrics*, v. 2, n. 4, p. 231–246, 1988.
- GIACINTO, G.; ROLI, F. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, v. 19, n. 9-10, p. 699 – 707, 2001.
Disponível em <http://www.sciencedirect.com/science/article/B6V09-43MMXCH-C/2/f6f53c025a2eeee3d871233c5d65da1d>
- GONZALEZ, R.; WOODS, R.; EDDINS, S. *Digital image processing using matlab*. Pearson Education India, 2004.
- GRAHAM, R. L.; HELL, P. On the history of the minimum spanning tree problem. *IEEE Annals of the History of Computing*, v. 7, n. 1, p. 43 – 57, 1985.
- GRIFFIN, G.; HOLUB, A.; PERONA, P. Caltech-256 object category dataset. Technical Report, California Institute of Technology, 2007.
- GRIGOROVA, A.; NATALE, F. G. B. D.; DAGLI, C.; HUANG, T. S. Content-based image retrieval by feature adaptation and relevance feedback. *IEEE Transactions on Multimedia*, v. 9, p. 1183–1192, 2007.
- GUAN, H.; ANTANI, S.; LONG, L. R.; THOMA, G. R. Minimizing the semantic gap in biomedical content-based image retrieval. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2010 (*Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, v.7628).
- GUO, G. D.; JAIN, A. K.; MAY, W. Y.; ZHANG, H. J. Learning similarity measure for natural image retrieval with relevance feedback. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, v. 1, p. 731, 2001.
- HEESCH, D.; RUGER, S. Nnk networks for content-based image retrieval. *26th European Conference on Information Retrieval*, v. 2997, p. 253–266, 2004.
- HEIDEMANN, G. Unsupervised image categorization. *Image and Vision Computing*, v. 23, n. 10, p. 861 – 876, 2005.
Disponível em <http://www.sciencedirect.com/science/article/B6V09-4GHSGGM-5/2/dbf03590702da3b951cd80ab45a51978>

- HERMAN, I.; MELANCON, G.; MARSHALL, M. S. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, v. 6, n. 1, p. 24–43, 2000.
- HINTON, G.; ROWEIS, S. Stochastic neighbor embedding. In: *Advances in Neural Information Processing Systems 15*, MIT Press, 2003, p. 833–840.
- HOCHMAN, N.; SCHWARTZ, R. Visualizing instagram: Tracing cultural visual rhythms. In: *Sixth International AAAI Conference on Weblogs and Social Media*, 2012.
- HOFFMAN, P.; GRINSTEIN, G.; PINKNEY, D. Dimensional anchors: A graphic primitive for multidimensional multivariate information visualizations. In: *Workshop on New Paradigms in Inform. Vis. and Manip. in Conjunction with ACM CIKM*, Kansas City, MO, USA, 1999, p. 9–16.
- HOLTEN, D.; VAN WIJK, J. Force-directed edge bundling for graph visualization. In: *Computer Graphics Forum*, Wiley Online Library, 2009, p. 983–990.
- HUANG, T. S. Can the world-wide web bridge the semantic gap? *Image and Vision Computing*, v. 30, p. 463 – 464, 2012.
- INSELBERG, A.; DIMSDALE, B. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: *Proceedings of the 1st conference on Visualization '90*, VIS '90, Los Alamitos, CA, USA: IEEE Computer Society Press, 1990, p. 361–378 (VIS '90,).
Disponível em <http://dl.acm.org/citation.cfm?id=949531.949588>
- J. H. CHOI, H. Y. JUNG, H. S. K.; CHO, H. G. Phylodraw: a phylogenetic tree drawing system. In: *Bioinformatics Applications Note*, 2000, p. 1056 – 1058.
- JÉGOU, H.; DOUZE, M.; SCHMID, C.; PÉREZ, P. Aggregating local descriptors into a compact image representation. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, IEEE, 2010, p. 3304–3311.
- JIANG, Y. G.; NGO, C. W.; YANG, J. Towards optimal bag-of-features for object categorization and semantic video retrieval. In: *Proceedings of the 6th ACM international conference on Image and video retrieval*, New York, NY, USA: ACM, 2007, p. 494–501.
- JOHNSON, B. Treeviz: treemap visualization of hierarchically structured information. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '92, New York, NY, USA: ACM, 1992, p. 369–370 (CHI '92,).
Disponível em <http://doi.acm.org/10.1145/142750.142833>

- JOIA, P.; PAULOVICH, F.; COIMBRA, D.; CUMINATO, J.; NONATO, L. Local affine multidimensional projection. *Visualization and Computer Graphics, IEEE Transactions on*, v. 17, n. 12, p. 2563 –2571, 2011.
- JOLLIFFE, I. T. *Principal component analysis*. 2 ed. Springer-Verlag, 487 p., 2002.
- JOSHI, A. J.; PORIKLI, F.; PAPANIKOLOPOULOS, N. P. Scalable active learning for multiclass image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 34, p. 2259 – 2273, 2012.
- KEIM, D. Visual exploration of large data sets. *Communications of the ACM*, v. 44, n. 8, p. 38–44, 2001.
- KEIM, D.; ANKERST, M. Visual data mining and exploration of large databases, a tutorial. In: *Proceedings of the Workshop on Visual Data Mining, PKDD'01*, 2001.
- KEIM, D.; ET AL. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, v. 8, n. 1, p. 1–8, 2002.
- KEIM, D. A.; KRIESEL, H. P. Visualization techniques for mining large databases: A comparison. *IEEE Transactions on Knowledge and Data Engineering*, v. 8, n. 6, p. 923 – 938, 1996.
- KEIM, D. A.; PANSE, C.; SIPS, M. *Information visualization: Scope, techniques and opportunities for geovisualization*. Elsevier Science Inc., 2005.
- KEMBAVI, A.; HARWOOD, D.; DAVIS, L. Vehicle detection using partial least squares. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 33, n. 6, p. 1250–1265, 2011.
- KLanke, S.; VIJAYAKUMAR, S.; SCHAAAL, S. A library for locally weighted projection regression. *Journal of Machine Learning Research (JMLR)*, v. 9, p. 623–626, 2008.
Disponível em <http://dl.acm.org/citation.cfm?id=1390681.1390702>
- KOREN, Y.; CARMEL, L.; HAREL, D. Ace: A fast multiscale eigenvectors computation for drawing huge graphs. In: *IEEE Symp. on Inform. Visualiz.*, 2002, p. 137–144.
- LACOSTE-JULIEN, S.; SHA, F.; JORDAN, M. Disclda: Discriminative learning for dimensionality reduction and classification. *Advances in Neural Information Processing Systems (NIPS)*, v. 21, 2008.
- LAENCINA VERDAGUER, S. *Color based image classification and description*. Tese de Doutoramento, Universitat Politècnica de Catalunya, 2009.

- LAMPING, J.; RAO, R. The hyperbolic browser: a focus + context technique for visualizing large hierarchies. *Readings in information visualization: using vision to think*, v. 1, p. 382 – 408, 1999.
- LEBLANC, J.; WARD, M. O.; WITTELS, N. Exploring n-dimensional databases. In: *Proceedings of the 1st conference on Visualization '90*, VIS '90, Los Alamitos, CA, USA: IEEE Computer Society Press, 1990, p. 230–237 (VIS '90,).
Disponível em <http://dl.acm.org/citation.cfm?id=949531.949568>
- LEE, M. D.; REILLY, R. E.; BUTAVICIUS, M. E. An empirical evaluation of chernoff faces, star glyphs, and spatial visualizations for binary data. In: *Proceedings of the Asia-Pacific symposium on Information visualisation - Volume 24*, APVis '03, Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2003, p. 1–10 (APVis '03,).
Disponível em <http://dl.acm.org/citation.cfm?id=857080.857081>
- LESOT, M.; RIFQI, M.; BENHADDA, H. Similarity measures for binary and numerical data: a survey. *International Journal of Knowledge Engineering and Soft Data Paradigms*, v. 1, n. 1, p. 63–84, 2009.
- LI, B.; ARTEMIOU, A.; LI, L. Principal support vector machines for linear and nonlinear sufficient dimension reduction. *The Annals of Statistics*, v. 39, n. 6, p. 3182–3210, 2011.
- LI, J.; WANG, J. Z. Automatic linguistic indexing of pictures by a statistical modelin approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 25, p. 1075–1088, 2003.
- LINDGREN, F.; GELADI, P.; BERGLUND, A.; SJOSTROM, M.; WOLD, S. Interactive variable selection (ivs) for pls. part ii: Chemical applications. *Journal of Chemometrics*, v. 9, n. 5, p. 331–342, 1995.
- LIU, T.; XIE, J.; HE, Y.; XU, M.; QIN, C. An automatic classification method for betel nut based on computer vision. In: *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, IEEE, 2009, p. 1264–1267.
- LIU, Y.; HSU, Y.; SUN, Y.; TSAI, S.; HO, C.; CHEN, C. A computer vision system for automatic steel surface inspection. In: *Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on*, IEEE, 2010, p. 1667–1670.
- LIU, Y.; ZHANG, D.; LU, G.; MA, W. Y. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, v. 40, n. 1, p. 262–282, 2007.
- LOUBIER, E.; BAHSOUN, W.; DOUSSET, B. Visualization and analysis of large graphs. In: *Proceedings of the ACM first Ph.D. workshop in CIKM*, New York, NY, USA: ACM, 2007, p. 41–48.

- LU, Z.; IP, H. Combining context, consistency, and diversity cues for interactive image categorization. *IEEE Transactions on Multimedia*, v. 12, p. 194 – 203, 2010.
- VAN DER MAATEN, L.; POSTMA, E.; VAN DEN HERIK, H. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, v. 10, p. 1–41, 2009.
- MÄENPÄÄ, T. *The local binary pattern approach to texture analysis: Extenxions and applications*. Oulun yliopisto, 2003.
- MAILUND, T.; BRODAL, G. S.; FAGERBERG, R.; PEDERSEN, C. N. S.; PHILLIPS, D. Recrafting the neighbor-joining method. *BMC Bioinformatics*, v. 7, n. 29, 2006.
- MANOVICH, L. Media visualization: Visual techniques for exploring large media collections. *Media Studies Futures*, 2011.
- MAY, A. Web-based image and video navigation. Technical Report, Imperial College London, 2004.
- MCCANN, S.; LOWE, D. G. Local naive bayes nearest neighbor for image classification. Technical Report, University of British Columbia, 2011.
- NAKAZATO, M.; HUANG, T. S. An interactive 3d visualization for content-based image retrieval. In: PRESS, I. C., ed. *Proceedings of IEEE International Conference on Multimedia 2001*, 2001.
- NESTOR, P.; O'DONNELL, B.; MCCARLEY, R.; NIZNIKIEWICZ, M.; BARNARD, J.; SHEN, Z.; BOOKSTEIN, F.; SHENTON, M. A new statistical method for testing hypotheses of neuropsychological/mri relationships in schizophrenia: Partial least squares analysis. *Schizophrenia Research*, v. 53, n. 1–2, p. 57–66, 2002.
- NEZAMABADI-POUR, H.; KABIR, E. Concept learning by fuzzy k-nn classification and relevance feedback for efficient image retrieval. *Expert Systems with Applications*, v. 36, n. 3, p. 5948 – 5954, 2009.
- NGUYEN, D.; ROCKE, D. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, v. 18, n. 1, p. 39–50, 2002.
- NGUYEN, G.; WORRING, M. Interactive access to large image collections using similarity-based visualization. *Journal of Visual Languages & Computing*, v. 19, n. 2, p. 203–224, 2008.
- NOVAKOVA, L.; STEPANKOVA, O. Radviz and identification of clusters in multidimensional data. In: *Information Visualisation, 2009 13th International Conference*, 2009, p. 104 –109.
- NOWAK, E.; JURIE, F.; TRIGGS, B. Sampling strategies for bag-of-features image classification. *Computer Vision*, v. 3954/2006, p. 490 – 503, 2006.

- PACI, M.; NANNI, L.; LAHTI, A.; SEVERI, S.; AALTO-SETALA, K.; HYTTINEN, J. Computer vision for human stem cell derived cardiomyocyte classification: The induced pluripotent vs embryonic stem cell case study. In: *Computing in Cardiology, 2011*, IEEE, 2011, p. 569–572.
- PAIVA, J.; FLORIAN, L.; PEDRINI, H.; TELLES, G.; MINGHIM, R. Improved similarity trees and their application to visual data classification. *Visualization and Computer Graphics, IEEE Transactions on*, v. 17, n. 12, p. 2459–2468, 2011.
- PAIVA, J. G. S.; SCHWARTZ, W. R.; PEDRINI, H.; MINGHIM, R. Semi-supervised dimensionality reduction based on partial least squares for visual analysis of high dimensional data. *Computer Graphics Forum*, v. 31, n. 3pt4, p. 1345–1354, 2012.
Disponível em <http://dx.doi.org/10.1111/j.1467-8659.2012.03126.x>
- PASOLLI, E.; MELGANI, F. Model-based active learning for svm classification of remote sensing images. In: *IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2010)*, 2010.
- PAULOVICH, F.; ELER, D.; POCO, J.; BOTHA, C.; MINGHIM, R.; NONATO, L. Piece wise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum*, v. 30, n. 3, p. 1091–1100, 2011.
Disponível em <http://dx.doi.org/10.1111/j.1467-8659.2011.01958.x>
- PAULOVICH, F.; MINGHIM, R. Text map explorer: a tool to create and explore document maps. In: *Information Visualization, 2006. IV 2006. Tenth International Conference on*, 2006, p. 245 –251.
- PAULOVICH, F.; MINGHIM, R. Hipp: A novel hierarchical point placement strategy and its application to the exploration of document collections. *Visualization and Computer Graphics, IEEE Transactions on*, v. 14, n. 6, p. 1229 –1236, 2008.
- PAULOVICH, F.; NONATO, L.; 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.; SILVA, C.; NONATO, L. Two-phase mapping for projecting massive data sets. *Visualization and Computer Graphics, IEEE Transactions on*, v. 16, n. 6, p. 1281 –1290, 2010.
- PAULOVICH, F. V. Mapeamento de dados multidimensionais - integrando mineração e visualização. PhD thesis, Instituto de Ciências Matemáticas e Computacionais - ICMC - USP São Carlos, 2008.
- PEDRINI, H.; SCHWARTZ, W. R. *Análise de imagens digitais: Princípios, algoritmos e aplicações*. Editora Thomson Learning, 528 p., 2007.

- PEKALSKA, E.; DE RIDDER, D.; DUIN, R.; KRAAIJVELD, M. A new method of generalizing sammon mapping with application to algorithm speed-up. In: *Annual Conf. Advanced School for Comput. Imag.*, 1999, p. 221–228.
- QUEIROZ, R. B.; SEVERINO, P. A.; RODRIGUES, A. G.; GÓMEZ, A. T. Redes neurais: Um comparativo com máxima verossimilhança gaussiana na classificação de imagens cbers 1. *Anais do IV Congresso Brasileiro de Computação - II Workshop de Tecnologia da Informação Aplicada ao meio Ambiente*, v. 1, p. 746–749, 2004.
- REINGOLD, E. M.; TILFORD, J. S. Tidier drawings of trees. *IEEE Transactions on Software Engineering*, v. 7, n. 2, p. 223 – 228, 1981.
- RÜGER, S. Putting the user in the loop: Visual resource discovery. In: *Adaptive Multimedia Retrieval: User, Context, and Feedback*, Springer, Heidelberg, 2006, p. 1–18 (*Lecture Notes in Computer Science*, v.3877/2006).
- RIBEIRO, S. R. A.; CENTENO, J. S. Classificação do uso do solo utilizando redes neurais artificiais e o algoritmo maxver. In: *Anais do X Simpósio Brasileiro de Sensoriamento Remoto*, 2001.
- ROBERTS, J. State of the art: Coordinated & multiple views in exploratory visualization. In: *Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07. Fifth International Conference on*, IEEE, 2007, p. 61–71.
- RODDEN, K.; BASALAJ, W.; SINCLAIR, D.; WOOD, K. Does organisation by similarity assist image browsing? In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, 2001, p. 190–197.
- ROSIPAL, R.; KRAMER, N. Overview and recent advances in partial least squares. *Lecture Notes on Computer Science*, v. 3940, p. 34–51, 2006.
- ROWEIS, S.; SAUL, L. Nonlinear dimensionality reduction by locally linear embedding. *Science*, v. 290, n. 5500, p. 2323–2326, 2000.
- ROZENBERG, G., ed. *Handbook of graph grammars and computing by graph transformation*, v. 1. World Scientific Publishing Company, 1997.
- RUGER, S.; HEESCH, D. Three interfaces for content-based access to image collections. In: *Image and Video Retrieval*, Springer Berlin / Heidelberg, 2004, p. 2067 (*Lecture Notes in Computer Science*, v.3115/2004).
- RUSZALA, S. D.; SCHAEFER, G. Visualisation models for image databases: A comparison of six approaches. In: *Irish Machine Vision and Image Processing Conference*, 2004, p. 186–191.

- SAITOU, N.; NEI, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, v. 4, n. 4, p. 406–425, 1987.
- Disponível em <http://mbe.oxfordjournals.org/cgi/content/abstract/4/4/406>
- SCHAEFER, G. Interactive exploration of image collections. *Computer Recognition Systems 4*, v. 4, p. 229, 2011.
- SCHWARTZ, W. R. *Looking at people using partial least squares*. Tese de Doutoramento, University of Maryland - College Park, 2010.
- SCHWARTZ, W. R.; GUO, H.; CHOI, J.; DAVIS, L. S. Face identification using large feature sets. *IEEE Transactions on Image Processing*, v. 21, n. 4, p. 2245–2255, 2012.
- DA SILVA, A.; FALCÃO, A.; MAGALHÃES, L. A new CBIR approach based on relevance feedback and optimum-path forest classification. *Journal of WSCG*, v. 18, n. 1-3, p. 73–80, 2010.
- DE SILVA, V.; TENENBAUM, J. B. Sparse multidimensional scaling using landmark points. Technical Report, Department of Mathematics, Stanford University, CA, USA, 2004.
- SIMONSEN, M.; MAILUND, T.; PEDERSEN, C. N. Rapid neighbour-joining. In: *Proceedings of WABI 2008*, Karlsruhe, Germany, 2008, p. 113–122.
- SORKINE, O.; COHEN-OR, D. Least-squares meshes. In: PRESS, I. C. S., ed. *Proceedings of Shape Modeling International*, 2004, p. 191–199.
- STEINBACH, M.; KARYPIS, G.; KUMAR, V.; ET AL. A comparison of document clustering techniques. In: *KDD workshop on text mining*, Boston, 2000, p. 525–526.
- SUGIYAMA, K.; TAGAWA, S.; TODA, M. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 11, n. 2, p. 109 – 125, 1981.
- SUGIYAMA, M. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *The Journal of Machine Learning Research*, v. 8, p. 1027–1061, 2007.
- SUGIYAMA, M.; IDÉ, T.; NAKAJIMA, S.; SESE, J. Semi-supervised local fisher discriminant analysis for dimensionality reduction. *Machine learning*, v. 78, n. 1, p. 35–61, 2010.
- SUZUKI, T.; SUGIYAMA, M. Sufficient dimension reduction via squared-loss mutual information estimation. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS2010)*, 2010, p. 804–811.

- TAN, P. N.; STEINBACH, M.; KUMAR, V. *Introduction to data mining.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- TARABALKA, Y.; FAUVEL, M.; CHANUSST, J.; BENEDIKTSSON, J. A. Svm- and mrf-based method for accurate classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, v. 7, n. 4, p. 736 – 740, 2010.
- 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.; DE SILVA, V.; LANGFORD, J. A global geometric framework for nonlinear dimensionality reduction. *Science*, v. 290, n. 5500, p. 2319–2323, 2000.
- THANGAVEL, K.; PETHALAKSHMI, A. Dimensionality reduction based on rough set theory: A review. *Applied Soft Computing*, v. 9, n. 1, p. 1 – 12, 2009.
Disponível em <http://www.sciencedirect.com/science/article/pii/S1568494608000963>
- TORGESON, W. Multidimensional scaling: I. theory and method. *Psychometrika*, v. 17, n. 4, p. 401–419, 1952.
Disponível em <http://ideas.repec.org/a/spr/psycho/v17y1952i4p401-419.html>
- TORGESON, W. Multidimensional scaling of similarity. *Psychometrika*, v. 30, p. 379–393, 1965.
- TUIA, D.; RATLE, F.; PACIFICI, F.; KANEVSKI, M.; EMERY, W. Active learning methods for remote sensing image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, v. 47, n. 7, p. 2218–2232, 2009.
- TUIA, D.; VOLPI, M.; COPA, L.; KANEVSKI, M.; MUÑOZ-MARI, J. A survey of active learning algorithms for supervised remote sensing image classification. *Selected Topics in Signal Processing, IEEE Journal of*, v. 5, n. 3, p. 606–617, 2011.
- VAILAYA, A.; T.FIGUEIREDO, M. A.; JAIN, A. K.; ZHANG, H. J. Image classification for content-based indexing. *IEEE Transactions on Image Processing*, v. 10, n. 1, p. 117–130, 2001.
- VALDIVIA, A. M. C. Mapeamento de dados multidimensionais usando árvores filogenéticas: foco em mapeamento de textos. Master thesis, Instituto de Ciências Matemáticas e Computacionais - ICMC - USP São Carlos, 2007.

- VIJAYAKUMAR, S.; D'SOUZA, A.; SCHAAAL, S. Incremental online learning in high dimensions. *Neural Computation*, v. 17, p. 2602–2634, 2005.
- VIJAYAKUMAR, S.; SCHAAAL, S. Locally weighted projection regression: An o(n) algorithm for incremental real time learning in high dimensional space. In: *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, 2000, p. 288–293.
- WALKER, J. Q. A node-positioning algorithm for general trees. *Software: Practice and Experience*, v. 20, n. 7, p. 685–705, 1990.
- WANG, C.; BLEI, D.; LI, F. Simultaneous image classification and annotation. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, p. 1903–1910.
- WANG, J.; PENG, W.; WARD, M. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In: *IEEE Symp. on Inform. Visualiz.*, Seattle, WA, USA, 2003, p. 105–112.
- WASKE, B.; VAN DER LINDEN, S.; BENEDIKTSSON, J. A.; RABE, A.; HOSTERT, P. Sensitivity of support vector machines to random feature selection in classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, v. 48, p. 2880 – 2889, 2010.
- WATTENBERG, M. Visual exploration of multivariate graphs. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA: ACM, 2006, p. 811–819.
- WHEELER, T. J. Large-scale neighbor-joining with ninja. In: *Proceedings of WABI 2009*, Philadelphia, PA, USA, 2009, p. 375–389.
- WOLD, H. Partial least squares. In: *Encyclopedia of Statistical Sciences*, v. 6, New York, NY, USA: Wiley, p. 581–591, 1985.
- WORRING, M.; DE ROOIJ, O.; VAN RIJN, T. Browsing visual collections using graphs. In: *Proceedings of the international workshop on Workshop on multimedia information retrieval*, New York, NY, USA: ACM, 2007, p. 307–312.
- XU, W.; ESTEVA, M.; JAIN, S.; JAIN, V. Analysis of large digital collections with interactive visualization. In: *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on*, IEEE, 2011, p. 241–250.
- YANG, J.; JIANG, Y. G.; HAUPTMANN, A. G.; NGO, C. W. Evaluating bag-of-visual-words representations in scene classification. In: *Proceedings of the international workshop on Workshop on multimedia information retrieval*, New York, NY, USA: ACM, 2007, p. 197–206.

- YANG, J.; WARD, M.; RUNDENSTEINER, E.; HUANG, S. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In: *Joint Eurographics - IEEE TVCG Symp. on Visualization*, Grenoble, France, 2003, p. 19–28.
- YANG, Y.; NIE, F.; XU, D.; LUO, J.; ZHUANG, Y.; PAN, Y. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 34, n. 4, p. 723–742, 2012.
- ZHANG, D.; LU, G. Evaluation of similarity measurement for image retrieval. In: PRESS, I. C., ed. *Proceedings of the International Conference on Neural Networks and Signal Processing*, 2003, p. 928 – 931.
- Disponível em <http://dx.doi.org/10.1109/ICNNSP.2003.1280752>
- ZHANG, J.; GRUENWALD, L.; GERTZ, M. Vdm-rs: A visual data mining system for exploring and classifying remotely sensed images. *Computers & Geosciences*, v. 35, n. 9, p. 1827–1836, 2009.
- ZHANG, Y.; WU, L.; WANG, S. Magnetic resonance brain image classification by an improved artificial bee colony algorithm. *Progress In Electromagnetics Research*, v. 116, p. 65 – 79, 2011.
- ZHOU, X. S.; HUANG, T. S. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, v. 8, p. 536 – 544, 10.1007/s00530-002-0070-3, 2003.
- Disponível em <http://dx.doi.org/10.1007/s00530-002-0070-3>

Nomenclatura

Este apêndice apresenta alguns termos utilizados nesta tese, cujas definições podem necessitar de referência:

- **Instância:** item que compõe uma coleção de dados, tais como documentos ou imagens;
- **Atributo, dimensão, característica:** determinada propriedade específica de uma instância, utilizada para representá-la em uma coleção de dados;
- **Espaço Multidimensional:** espaço de descrição das instâncias de uma coleção de dados, representado por uma matriz na qual as linhas representam as instâncias e as colunas representam as dimensões.
- **Espaço Original:** espaço multidimensional de uma coleção de dados com o número de dimensões e respectivos valores originais, coletados ou simulados em determinada aplicação.
- **Filtros de Gabor:** descritor de característica de imagens, caracterizado por um filtro linear cuja resposta é dada por uma função gaussiana modulada por uma função senoidal (Gonzalez et al., 2004);
- ***Histogram of Oriented Gradients* (HOG):** descritor de característica de imagens, baseado no histograma de ocorrências de um gradiente de orientação em diversas partes de uma imagem (Gonzalez et al., 2004);

- **Local Binary Patterns (LBP)**: descritor de característica de imagens, baseada no histograma de ocorrência de códigos binários construídos pela comparação entre os valores dos pixels dessas imagens e seus pixels vizinhos (Mäenpää, 2003);
- **Layout**: estrutura de organização de instâncias para a visualização de informação;
- **Espaço Reduzido**: espaço de descrição das instâncias obtido após a aplicação de um procedimento de redução de dimensionalidade, que possui um número de dimensões menor do que aquele observado no espaço multidimensional original;
- **Espaço de Visualização**: espaço de descrição das instâncias sendo visualizadas de acordo com a aplicação de uma técnica de visualização. Usualmente possui 2 ou 3 dimensões;
- **Instância de Treinamento**: instância previamente rotulada utilizada para treinar um classificador;
- **Instância de Teste**: instância cujo rótulo será determinado por um modelo de classificação;
- **Modelo de Classificação**: núcleo de um classificador, construído na fase de treinamento, contendo todo o seu conhecimento a respeito das classes de um problema. É utilizado para predizer a classe de determinada instância de teste;
- **Ground Truth**: esquema de rotulamento ideal de determinada coleção de dados. É utilizado para avaliar o desempenho de um classificador;
- **Matriz de Confusão**: matriz quadrada utilizada na análise de uma classificação, na qual as linhas representam o *ground truth* de uma coleção, e as colunas representam o resultado obtido na classificação. Os valores das células fornecem o número de instâncias rotuladas;
- **Application Programming Interface (API)**): conjunto de estruturas e métodos que encapsulam determinada funcionalidade ou conjunto de funcionalidades e algoritmos.

Artigo: Improved Similarity Trees and their Application to Visual Data Classification

Este apêndice apresenta o conteúdo completo de um artigo publicado no *IEEE Transactions on Visualization and Computer Graphics* (IEEE TVCG). Uma visão geral desse trabalho foi apresentada no Capítulo 3 desta tese.

Improved Similarity Trees and their Application to Visual Data Classification

Jose Gustavo S. Paiva, Laura Florian-Cruz, Helio Pedrini, Guilherme P. Telles, and Rosane Minghim

Abstract—An alternative form to multidimensional projections for the visual analysis of data represented in multidimensional spaces is the deployment of similarity trees, such as Neighbor Joining trees. They organize data objects on the visual plane emphasizing their levels of similarity with high capability of detecting and separating groups and subgroups of objects. Besides this similarity-based hierarchical data organization, some of their advantages include the ability to decrease point clutter; high precision; and a consistent view of the data set during focusing, offering a very intuitive way to view the general structure of the data set as well as to drill down to groups and subgroups of interest. Disadvantages of similarity trees based on neighbor joining strategies include their computational cost and the presence of virtual nodes that utilize too much of the visual space. This paper presents a highly improved version of the similarity tree technique. The improvements in the technique are given by two procedures. The first is a strategy that replaces virtual nodes by promoting real leaf nodes to their place, saving large portions of space in the display and maintaining the expressiveness and precision of the technique. The second improvement is an implementation that significantly accelerates the algorithm, impacting its use for larger data sets. We also illustrate the applicability of the technique in visual data mining, showing its advantages to support visual classification of data sets, with special attention to the case of image classification. We demonstrate the capabilities of the tree for analysis and iterative manipulation and employ those capabilities to support evolving to a satisfactory data organization and classification.

Index Terms—Similarity Trees, Multidimensional Projections, Image Classification.

1 INTRODUCTION

Trees in visualization have been frequently used as a means to express multidimensional hierarchical data in various ways. Many visualization systems allow effective ways of exploring trees via various versions of link-node representations [32]. Widely used techniques such as Treemaps [2] improve space usage against conventional link-node representation and optimize the number of data set items that can be presented in a rectangular space.

In most cases, tree representations have been used in situations where: (i) hierarchy is the natural organization of the data and the tree reflects that; or (ii) data display is subject to a type of ordering of attributes and exploration is therefore attribute-based. In either case, trees are useful to explore data bounded by the hierarchy, and interpretation relies on a small number of attributes (those that can be mapped to visual artifacts such as color, texture and size).

Shortcomings of attribute centered visualizations start to appear when the number of dimensions grow. In this case, point-based strategies, defined by techniques where each individual in the display is an individual or group of individuals in the data set, offer a solid first step to explore the data. It relies on a relationship among individuals calculated using all the available attributes. In such cases, some type of dimensional reduction is commonly employed. Visualizations can be obtained by providing mappings of original or reduced spaces to visual (2D or 3D) spaces. Multidimensional projections are used to generate these mappings and the field has enjoyed great attention lately, resulting in more precise and faster approaches for point placement.

Similarity trees have unique properties concerning their ability to

solve similarity-based (therefore, content-based) point placement. Additionally, they are an intuitive way of expressing structure in the relationship among items in the data set by adding hierarchy to the concept of similarity, that is, by seamlessly representing levels of similarity reflected as depth in the tree.

Regardless of their attractiveness, very little research has been carried out on similarity trees as a form of point placement for multidimensional visualization. The only technique reported in that context is the NJ tree (neighbor joining tree) [10], that has been shown to be a high precision tree structure in terms of reflecting on the display the neighborhoods found in the multidimensional space. Tests were made mainly for textual document collections, but its application for visualization of collections of images has also been reported [15], demonstrating qualities of NJ trees for image analysis tasks, such as visual support to feature extraction, selection, and comparison. In the latter work, interaction and coordination capabilities of trees and projections become evident, but no numerical evaluation is provided.

Another advantage of NJ similarity trees is to present, within a particular branch, the same properties of the global display due to their precise phylogeny reconstruction algorithm, largely explored in Biology [20]. Additionally, compared to other point placement strategies, point clutter is reduced with NJ trees due to the simplified graph layout algorithm that is used for display and due to the fact that branches allow direct inspection. Thus, there is no need for large gaps on the display window and all the space available can be used to plot points and edges. Their disadvantages are related to computational cost and the massive presence of virtual, non-representative nodes that use a substantial part of the visual space, reducing presentation area. The processing time constraint, though, is probably the reason why the technique has not yet reached wider usage.

In this paper, we propose a highly improved version of the NJ trees, both in space usage and processing speed, and illustrate their usefulness by developing an application that shows the adaptability of the approach for visual data classification tasks, with emphasis on image classification. A large number of applications can make use of this strategy. Classification is a process very difficult to converge and to adapt to a particular environment, especially when dealing with image data. In many cases, even when automatic classification algorithms give satisfactory results, visual feedback is very important to offer insight into the reasons for classification failure, and to support interfering with further classification passes, as well as intently correcting

- J.G.S. Paiva is with Federal University of Uberlândia and ICMC, jgustavo@icmc.usp.br.
- L. Florian-Cruz is with ICMC - University of São Paulo, laurely@icmc.usp.br.
- H. Pedrini is with IC - University of Campinas, helio@ic.unicamp.br.
- G.P. Telles is with IC - University of Campinas, gpt@ic.unicamp.br.
- R. Minghim is with ICMC - University of São Paulo, rminghim@icmc.usp.br.

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.

For information on obtaining reprints of this article, please send email to: tvvcg@computer.org.

the classification. A proper visual set up is at the core of this problem. Some of the applications include product categorization, scene identification, photographs and music organization; and forensics.

The main contributions of this paper are: (i) the definition of a leaf promotion procedure that improves visual quality and space usage of NJ trees; the resulting technique is named Promoting NJ tree (PNJ tree); (ii) the adaptation and implementation of two faster versions of the neighbor joining strategy, reducing its computational cost and rendering the technique applicable to much larger data sets; (iii) objective numerical evaluation of the quality of the final displays; and (iv) the illustration of an application where similarity visualizations displayed by trees are employed to help the user in various steps of the data classification process. We focus on image classification and also exemplify using text.

The remainder of this paper is organized as follows. Section 2 reviews the literature on content based point placement of multidimensional data. Section 3 describes similarity trees, explains the faster algorithms and introduces the leaf promotion procedure. Section 4 presents analysis and comparative results of the experiments with similarity trees, while Section 5 presents the tree-based tools for the visual classification application. Section 6 concludes the paper.

2 RELATED WORK

Point placement strategies aim at mapping each individual data point into a visual space. While basic mapping is most commonly based on dissimilarity calculations, some of their other attributes can also be mapped as geometrical shapes, colors, textures and even sounds. Relationships can also be explicitly represented in the visual space as edges between points.

Many of the most demanding current applications produce data sets with a large number of data items, represented by a large number of attributes, or dimensions, imposing serious constraints to information visualization and analysis techniques.

A recurring way to approach high multiplicity of dimensions or attributes is to use dimension reduction strategies, by selecting, mapping or combining dimensions in order to apply conventional visualization methods that are prepared to handle low dimensional data. Layout approaches can also be based on multidimensional projections, which map a high dimensional space into a visual space trying to preserve, in the final space, a relevant relationship among data elements in the original space.

Most 2D or 3D multidimensional projections employ a measure of dissimilarity between data elements. Thus, data elements mapped as points to the same region in visual space are considered similar. Interpretation of the layout is accomplished by locating groups of interest and focusing on such groups and their subgroups.

Figure 1 shows a layout of 675 extracts from scientific papers built using the Least Squares Projection [30]. On this type of projection, proximity between points in the layout is meant to indicate document similarity. An automatic topic detection procedure yields a glimpse of the subjects handled by a document group.

Several multidimensional projection techniques are available. Most of them are based on dimension reduction techniques and some of those have been largely used for data analysis for decades. Principal Component Analysis (PCA) [23] is a widely used linear projection technique that employs linear combinations of the data attributes (dimensions) with a high covariance degree, producing attributes with less dependence. Its main disadvantage is the low quality of resulting layouts due to the relatively low power of the two or three principal dimensions to express important features in heterogeneous data sets. Its computational cost is also high, $O(nm^2)$, where n is the number of objects and m is the number of dimensions. Multidimensional Scaling (MDS) [9] comprises a class of techniques that can be used to perform projections. The simplest MDS approach is the Force Directed Placement (FDP) model [13]. The FDP model is based on a spring system concept, where the multidimensional instances are modeled as objects linked by springs such that the repulsion and attraction forces between the objects are proportional to the multidimensional distances. The projection is attained when the spring system reaches an equilibrium

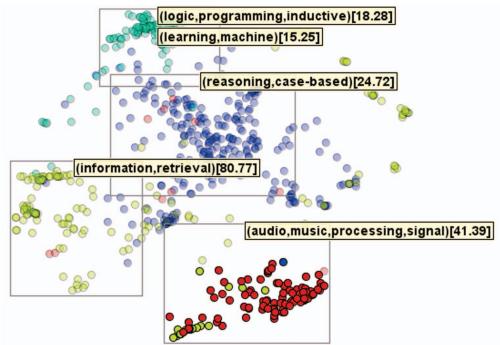


Fig. 1. Projection of 675 extracts (title, authors, abstract and references) of scientific papers. Circles represent documents, colored according to subjects: case-based reasoning (blue), inductive logic programming (green), information retrieval (yellow), and computer audio (red). The audio group is highlighted.

state. This technique normally presents high precision for data sets with objects possessing non-linear relationships, but has high computational cost, $O(n^3)$ and uneven precision across different types of data sets (such as documents and images). The Force Scheme [38] and the strategy defined by Chalmers [4] are also based on the concept of attraction and repulsion forces among objects. Both use heuristics that decrease FDP's running time to $O(n^2)$. As a faster, high precision projection, the Least Square Projection (LSP) [30] applies a Laplacian operator in which neighbor data in the multidimensional space are projected to close positions on the visual layout. LSP presents a proper balance between precision and running times, being an $O(n\sqrt{n})$ technique.

The previous methods generate good results for multidimensional data with varying precisions. In particular, it has been shown that LSP consistently approximates highly related data points and separates groups well. Layouts resulting from LSP provide a powerful guide to recognize global patterns in multidimensional data and help user to build a mental model of the data set [30].

Many new dimensionality reduction techniques for multidimensional projections have been put forward lately [5, 11, 28, 31] with motivating results. With this evolution, their number of applications have increased, particularly in recent years. They are capable of handling large data sets and a large number of dimensions. They have also been in the frontier of integration between spatial and non-spatial visualization techniques.

Regardless of their advantages, projections are not usually capable of making locally the same claims of precision made globally. Within a certain group of points, there is significantly less neighborhood consistency that, in principle, the similarity relationship should be capable of coding. Hierarchical approaches can be a solution to that particular problem [29], but they rely on early clustering, preventing proper analysis of boundaries between groups. Newer and extremely fast projections have been put forward lately, handling well the problem of computational cost with improved precision [31] or the problem of consistency in local neighborhoods [28]. However, they work with space partitioning and cannot handle similarity matrices, impairing their use with applications that define similarity relationships without undergoing feature space definition (for instance, textual similarity from string comparison or similarity based on other non-numerical data). Projections are also subject to a high degree of cluttering due to the algorithm effort in guaranteeing the neighborhood between highly correlated neighborhoods. Visual density estimation, in that context, is virtually impossible without further interaction.

Multidimensional projections are evaluated by their capability of grouping and separating strongly related data items and by their capability of recovering, in visual spaces, data distribution or neighborhoods in original space. They have been proven to excel in both. Similarity trees should be evaluated in the same manner.

As an alternative technique for exploring multidimensional data based on content, the data set can be organized as a similarity tree. By positioning objects on branches, similarity is organized into levels, a natural way of interpreting degrees of similarity. Global analysis is not impaired, that is, it is also possible to distinguish larger groups of interest within the data set. Local analysis is as precise as global, sometimes more precise. Similarity trees can be built from feature spaces or similarity relationships. This is the case of the NJ tree [10], a visualization technique based on a phylogeny reconstruction algorithm named Neighbor Joining. This paper addresses this alternative, improving Cuadros et al.'s technique in both visual and processing time scales and validating its use for visual data classification.

3 SIMILARITY TREES FOR MULTIDIMENSIONAL VISUALIZATION

This section describes NJ trees and proposes alternatives for improvement of both time and visual scalability. The result is a faster and less cluttered hierarchical visualization based on similarity, that completes the typical functionality of a multidimensional projection.

In terms of tree construction, the technique encompasses a two step procedure: after the construction of the similarity data structure representing a tree, the final layout is produced by a tree drawing algorithm. The input for tree construction algorithms is a symmetrical similarity matrix, and finding a similarity measure that faithfully represents relationships among objects in high dimensional feature space is a key issue for both trees and projections.

Addressing the quality of the similarity relationship is beyond the scope of this work. We consider that proper feature extraction and similarity calculation precede the visualization step and that the tree is responsible to reflect that feature space and similarity, although the tree itself could be a valuable tool for feature space convergence procedures.

Next, we explain and exemplify NJ trees, reporting on their known features. We also show their limitations as well as our solutions to improve those limitations. Additionally, an evaluation methodology for similarity trees is introduced.

3.1 Neighbor Joining

A phylogenetic tree represents evolutionary relationships within a group of species. Leaves represent actual species and internal nodes represent hypothetical ancestors. Edges represent ancestor-descendant relation and their lengths may indicate distance between species. A phylogenetic tree may be rooted or unrooted [35].

In the work by Cuadros et al. [10], unrooted phylogenetic trees were built for documents and served as visualization models. Tree leaves represent documents, internal nodes represent hypothetical documents with intermediate content, and edge lengths indicate the similarity among documents. Because phylogenetic tree construction is NP-hard in general, the authors used a well-known heuristic that builds unrooted trees from distance matrices named *Neighbor Joining* (NJ) [35]. The trees are named NJ trees. Figure 2 shows an example of an NJ tree for the collection of documents of Figure 1. It shows that group organization in branches is consistent with the LSP projection, but it also structures the groups into layers of similarity and reduces cluttering considerably. Local neighborhood is also reconstructed in NJ trees according to the similarity matrix.

We refer to the previous work [37] for implementation details of Algorithm 1. The input is a similarity matrix D and the output is a phylogenetic tree with n leaves and $n - 2$ internal nodes with degree 3. Its running time is $O(n^3)$. This is clearly a limiting factor for very large data sets. In Algorithm 1, R_i is the average distance from node i to every other node in D . S_{ij} captures the notion of evolutionary change, and, at each step, two closest nodes in D are removed from the matrix and replaced by a combination (virtual) node x , for which a new row is inserted in D . The new distances from x to every other remaining node are calculated according to the formulae in Algorithm 1.

NJ trees were successful in building visualizations that consistently separate texts with similar content in different branches. It also favors

Algorithm 1 Neighbor Joining

- 1: Let every row i of matrix D be associated with a leaf i .
- 2: **repeat**
- 3: Select a pair of nodes (i, j) with the minimum value for S_{ij}
- 4: Create a new node x connected to nodes i and j , with edge lengths L_{ix} and L_{jx}
- 5: Add row x to D , with values D_{xk} for every column $k \neq i, j$
- 6: Remove rows i and j from D
- 7: **until** $n = 3$
- 8: Connect the three remaining nodes in the tree

$$S_{ij} = D_{ij} - R_i - R_j; R_i = \frac{1}{n-2} \sum_k D_{ik}$$

$$L_{ix} = \frac{1}{2}(D_{ij} + R_i - R_j); L_{jx} = D_{ij} - D_{ix}; D_{xk} = \frac{1}{2}(D_{ik} + D_{jk} - D_{ij})$$

data exploration in various levels of detail. The technique can also be applied to other types of data, such as image and sound [10].

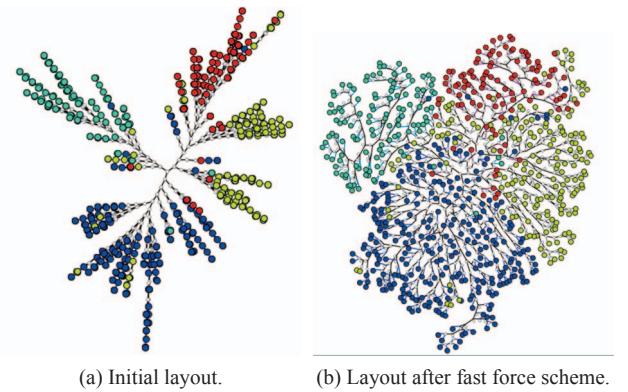


Fig. 2. NJ tree for a collection of 675 textual documents.

After NJ tree construction, a linear radial graph layout algorithm [1] is applied, resulting in the nodes separated in larger branches (Figure 2a). Then, applying a fast simplified force-based layout [38] nodes are spread minimizing cluttering and allowing inspection of branch organization (Figure 2b).

Regardless of the motivating visual exploration capabilities and aesthetically pleasing layout, in a collection with n objects, $n - 2$ internal nodes will be created. The space is overloaded by these virtual nodes that neither represent information nor add to the concept of multi-level similarity.

Figure 3 shows an NJ tree with its virtual objects highlighted, showing that the visualization model is visually polluted even for such a moderate sized collection. For larger collections, the problem becomes more serious.

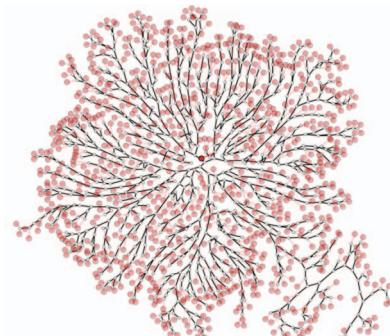


Fig. 3. NJ tree for 890 items with its 888 virtual objects highlighted.

The procedure described next intends to improve the layout regarding its visual space usage.

3.2 Leaf Promotion in NJ trees

In order to reduce visual overload in NJ trees, we present a deterministic, ordered graph rewriting operation [14, 34] that replaces an internal node with a leaf whenever a certain configuration of nodes exists. This operation is called promotion.

Suppose a NJ tree T such that there exists a pair of leaves u and v , both connected to the same internal node a , and that another internal node b connects a and a leaf w , as shown in Figure 4(a). The rationale behind promotion is that because no other node was closer to w than a (during the construction of T), then no other node is closer to u and v than w , so a can be replaced by w and b can be removed altogether, with no loss of representational power. This relation among u , v , w and a is weak, because it is valid only for the topology created by the NJ algorithm and it does not hold for the metric space induced by the similarity matrix (if any exists). Nevertheless, experiments have shown that the layout precision is barely affected by promotion.

The operation may be formally defined in terms of a pattern and a replacement on the tree, as shown in Figures 4(b) and 4(c). The promotion procedure consists of replacing every occurrence of the pattern, in decreasing order of the distance from node a in the pattern to a node in the center of the tree, breaking ties arbitrarily. Weights in the replacement are denoted ω_r and are evaluated from the weights in the pattern, that are denoted ω_p and were computed by the tree construction algorithm. We abuse notation and use T_1 , T_2 and T_3 to denote the node connecting a subtree to the rest of the tree. We let $\omega_r(w, T_1) = \omega_p(b, T_1) + \omega_p(a, b)/2 + \omega_p(w, b)/2$ and $\omega_r(w, T_i) = \omega_p(a, T_i) + \omega_p(a, b)/2 + \omega_p(w, b)/4$ for $i = 2, 3$. The resulting tree will be referred to as PNJ tree. Linear time suffices to find the nodes that match the pattern, the center of the tree, and to evaluate the distances. Sorting takes $O(n \log n)$. Applying the pattern takes constant time, then the overall promotion takes $O(n \log n)$. Experiments have shown that computational time to perform promotion is negligible.

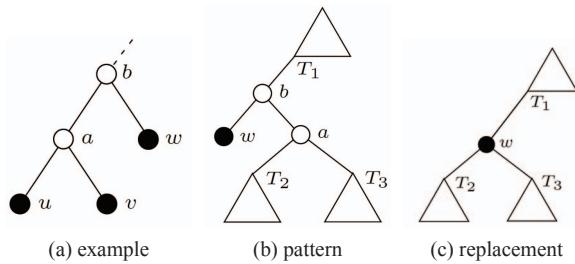


Fig. 4. Promotion. Filled circles represent actual objects and triangles represent subtrees.

3.3 Faster Algorithms for Neighbor Joining

In the original NJ, the search for the minimum S_{ij} in a matrix D (which determines the next pair of nodes to combine) essentially visits every cell in D and thus takes $O(n^2)$. This is the target operation of NJ variations that aim at improving efficiency, either by using specialized data structures [25, 36, 40] or through heuristics that sacrifice precision [16, 17]. We have investigated two of them to assess the scalability of NJ trees. Our descriptions below are based on the original NJ steps and on the notation presented in Algorithm 1. Additionally, let S_{min} denote the minimum S_{ij} for a matrix D .

Rapid Neighbor Joining [36] produces the same trees as NJ. It is based on the observation that, while visiting row i and evaluating $S_{ij} = D_{ij} - R_i - R_j$ in the search for S_{min} , the value of R_i is fixed. The algorithm keeps an auxiliary matrix where rows are sorted (and whose cells are indexed back to D) and stops visiting the sorted row i as soon as $S_{ij} - R_i - R_{max} > S_{min}$, where R_{max} is the maximum R_i among all matrix rows (thus $R_j \leq R_{max}$.) Some overhead is added by sorting rows, handling removed columns and evaluating R_{max} . While Rapid NJ is still $O(n^3)$, experiments have shown that the bounding strategy

significantly reduces the number of visited cells in the search for S_{min} and that Rapid NJ outperforms other faster algorithms.

A different approach is adopted by Fast Neighbor Joining [16]. Fast NJ maintains a set having $O(n)$ candidate matrix cells. The search of S_{min} is restricted to such set of candidates. Initially, the set of candidates contains the minimum S_{ij} values for each row i . When rows i and j are connected to a new node x and removed from D , the candidates they contribute are removed and a new candidate for the new matrix row x is added. The approach results in an algorithm that does not produce the same tree as the original NJ and whose worst case is $O(n^2)$. The authors have shown that when the distance measure is nearly additive, Fast NJ produces the same tree as NJ. We devised and adopted a more conservative variation of Fast NJ. Exactly n candidates are kept, one for each row. When rows i and j are connected to a new node x and removed from D , we remove the candidates they contribute and we also recompute remaining candidates that involved either i or j . Recomputing the candidates potentially improves the choices of S_{min} . The modified algorithm is $O(n^3)$, but the practical running times are much closer to $O(n^2)$.

We further discuss running time and precision issues in the next section.

4 RESULTS

We have added the new strategies and algorithms to a visualization platform called VisPipeline. VisPipeline implements various multidimensional visualization techniques, mainly focused on content-based point placement techniques, supported by interaction, preprocessing and some analysis tools. VisPipeline is implemented in Java and allows components and tools to be connected visually. We then specialized some of the tools and created the visual image mining framework presented in Section 5. The two faster versions of the NJ tree, along with their versions with node promotion, were adapted and implemented. The available trees are: the original version (NJ), the acceleration implemented by the Rapid Algorithm (Rapid NJ), the Fast NJ modified as previously stated (FAST), and the promoting strategies for each one of these, which we will refer to as PNJ, PRapid and PFast.

We have added numerous interactions as well as a few mining and evaluation functions, to adapt currently point placement analysis to the similarity tree concept. Some of the added features include: tree based interactions for cutting, selecting, splitting, browsing and coloring; numerical evaluations for the tree layouts; visual analysis procedures, such as clustering and classification algorithms (thus far SVM [7, 22], KNN [8, 12] as well as various hierarchical clusterings are implemented); and similarity perturbation procedures to perform class visualization from an initial visual layout of labeled instances.

For the tests of processing times, we have employed a laptop with a 2.53 GHz Core2 Duo processor, 4GB RAM, running Windows 7.

This section presents the results of applying and testing the new versions of the similarity trees for improvement on visual and time scalability.

We firstly evaluate the precision of the original NJ tree against a high precision projection, LSP. The first paper that proposes the use of phylogenetic trees for visualization [10] also does that, but it employs various textual document data sets, and we wished to test it against imaging data sets, due to our focus application. Although a previous work uses the tree for visualization of images [15], these evaluations were not previously performed. Secondly, we evaluate the precision of the trees against one another. Finally, we compare the processing times of all these techniques and demonstrate their capability to support exploration both in overall and detailed levels. First, we describe the data sets employed in the tests.

4.1 Data Sets and Test Setup

In the demonstrations and evaluations that follow, we have chosen to employ three imaging data sets, since this is the source of the sample application in Section 5.

The first image collection, named COREL, is composed of photographs that represent specific subjects: African tribes, beach, build-

ings, buses, dinosaurs, elephants, flowers, horses, mountains and glaciers, and food. Each image is represented by a vector of 150 SIFT descriptors [24]. The second image collection, named MEDICAL, is composed of MRI medical images, each represented by a vector of 28 descriptors, including Fourier descriptors from image histogram and energy computed from 2D image Fourier descriptors, mean intensity, and standard deviation computed over the entire image. The third collection, named OBJECTS, is a subset of the Caltech 256 data set, containing 45 classes, 5096 images and 1000 attributes that are 1000 SIFT features calculated over Harris-Laplace extractors [21]. For all data sets, the similarity used to compare the samples was Euclidean similarity. All data sets are labeled and these labels are used for the purpose of evaluation of the visualization and mining procedures reported here. Table 1 summarizes the numbers related to these data sets.

Table 1. Collection details.

Data set	Content	Classes	Items	Attributes
COREL	Photographs	10	1000	150
MEDICAL	MRI images	12	548	28
OBJECTS	Object pictures	45	5096	1000

The first aspect of the trees to be evaluated is the use of visual space.

4.2 Visual Representation and Space Organization

In the NJ tree, most of the larger groups of elements lie on separate branches, forming subtrees of the main tree. As a consequence of the algorithm, the branches are also divided into smaller branches with the same properties. Exploration and interpretation can be made equally in all levels. Selection of the branches, and therefore of groups of collection elements, can be made with one click of the mouse. On the other hand, interpretation is dependent on branches, that is, users need the edges representing the branches to recognize elements with large correlation. Opposite to multidimensional projections, visual space must be shared with the edges.

The main drawback of NJ trees regarding space usage, though, is the visual overload imposed by the presence of virtual objects (white small circles on the layouts), created every time two leaves are added to the tree or a node is combined with an already present virtual node. For a data set with n elements, $n - 2$ virtual nodes are added by any of the algorithms. As these objects are not part of the collection, they do not play any important role during analysis.

Node promoting shares the visual organization and exploratory advantages of NJ trees, but with fewer virtual objects, resulting in a cleaner view and ultimately leading to a more direct access to actual data objects on the tree branches. In our tests, PNJ presents in average 51% fewer virtual nodes than NJ trees. Table 2 summarizes the numbers related to space usage on the visualizations of the test data sets.

Table 2. Numbers of Nodes.

Data set	Nodes	Virtual	PNJ virtual	Saving	PFAST	Saving
					virtual	
COREL	1000	998	412	59%	548	45%
MEDICAL	540	538	186	65%	308	43%
OBJECTS	5097	5095	2437	53%	3019	41%

For PNJ, space usage is much more rational and clutter is reduced. Figure 5 shows the images of two of the test data sets, where we can clearly see the impact of having fewer virtual nodes. For larger data sets, the impact is even more evident.

It must be observed that the promotion procedure disturbs the 2-neighborhood of the nodes. That changes slightly the distance distribution (see Section 4.3), but should not impact interpretation of the trees or other neighborhoods significantly. Large layers of virtual nodes at top levels, typical of NJ, are avoided. The Rapid algorithm for

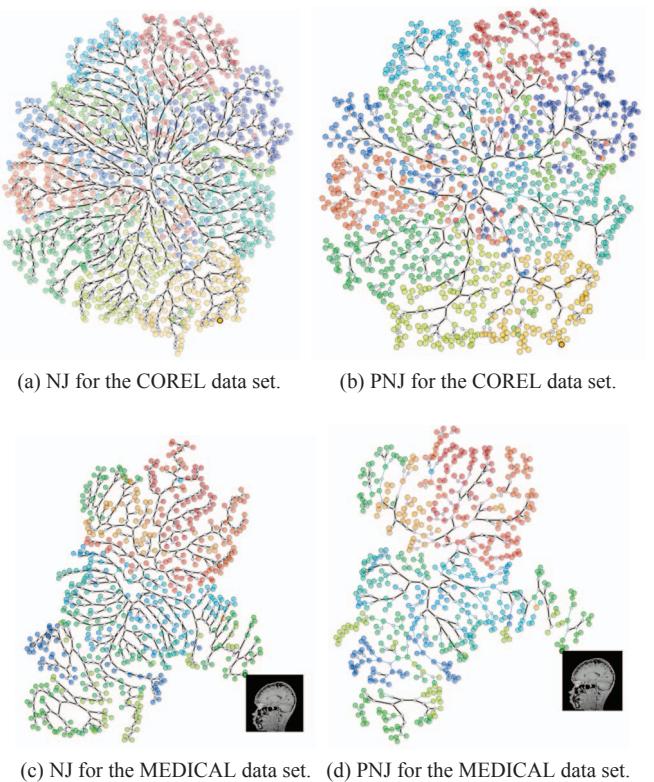


Fig. 5. Tree models for the COREL and MEDICAL data sets, highlighting virtual nodes and paths.

NJ leads to the same configuration as the original NJ. Thus, PNJ and PRapid are the same as far as virtual node count. Trees produced by the FAST algorithm have the same node count as NJ, although producing a different node configuration. FAST node configuration leads to a lower reduction by promotion, as Table 2 reveals. With node promotion, space can be occupied by further nodes or used to easier access to groups of individuals.

4.3 Precision

Various multidimensional projections have high precision in terms of grouping and separation, as well as very competitive processing times for large data sets. Similarity trees also have high precision but algorithm complexity of previously available versions and consequent processing times were far behind those of the high precision projections, making the technique less appealing for large data sets, regardless of its qualities. Here we show the evidence of precision and compare the precision of all the algorithms as well as with the precision of LSP.

Measuring the precision of a particular layout is meant to verify to what degree the distance (or neighborhood) in the layout corresponds to the similarity (or neighborhood) in the original feature space. In the layout generated from multidimensional projections, the Euclidean distance is used to indicate dissimilarity between points according to that projection. In trees, the dissimilarity between points is indicated by the weight of the path between them. Edge weights in the tree are evaluated by the NJ algorithm (L_{ix} and L_{jx} in Algorithm 1).

To evaluate precision, two measures previously employed for projections [30] were adapted to the trees. The first measure is the *Neighborhood Preservation*, that evaluates the frequency at which the k -nearest neighbors of an object in the layout are also its k nearest neighbors in the original feature space, in average. Since the Rapid algorithm produces the same trees as the original algorithm, precision discussions on Rapid are the same as the discussion on the original NJ.

For the COREL data set, the trees show better precision than that of LSP. Also, it becomes clear that near neighborhoods are reconstructed

better for the trees than for LSP (Figure 6). Among the trees themselves, it can also be noticed that the ones created by FAST present slightly lower precision than the corresponding ones generated by original NJ algorithm. Another observation is that the promoting version for each algorithm is slightly less precise in neighborhood preservation than the non-promoting ones. That is easily explained by the fact that the 2-neighborhood of a point changes when a node is promoted, since sometimes it becomes the parent of a couple of other nodes that were deemed the closest in the first place (see Section 3.2). In that case, the parent is interfering with the closeness and with the concept of first neighbor. That is also the reason for the low value of neighborhood preservation for the first neighbor on those trees. In larger neighborhoods, the relationships are not changed by promotion. Yet, the trees presented very good neighborhood preservation considering the difficulty in recovering that for any multidimensional data mapping. The neighborhood preservation plot among trees for the MEDICAL data set is very similar to the plot for COREL, but there is no significant difference between the precisions of trees and of LSP, except for the very near neighborhoods, where global projections such as LSP do not perform so well.

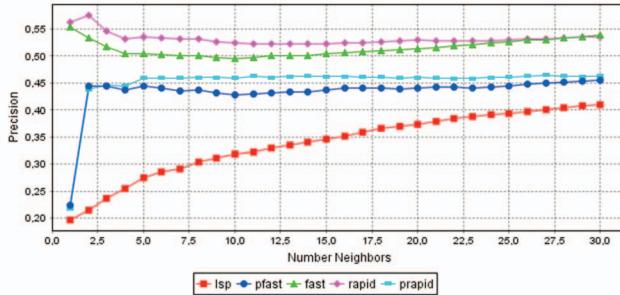


Fig. 6. Precision by Neighborhood Preservation for the COREL data set.

When a data set used for tests is labeled, that can also be used to verify precision based on class reconstruction in the final layout. We would like to point out that feature space definition and dissimilarity calculations play an important role in any visualization or mining process, and much so in the precision of similarity-based visualizations, particularly those based on pseudo-classes. For the COREL and MEDICAL data sets used here, the feature space was obtained or calculated, and various tests were carried out, so that the result described reasonably well the set of images, albeit without resolving class separation.

The second precision measure is called *Neighborhood Hit*. It is the average frequency at which an object and its k -nearest neighbors belong to the same class. Figure 7 shows the results of neighborhood hit for the COREL data set. Here, the trees also outperform LSP, but the capability of class reconstruction of the promoting trees does not differ much from that of the original algorithms. The performances for all the techniques were not significantly different. For the neighborhood hit plot based on the MEDICAL data set, the observations on the tree precisions also hold. However, there is no significant difference between the performance of LSP and that of the trees.

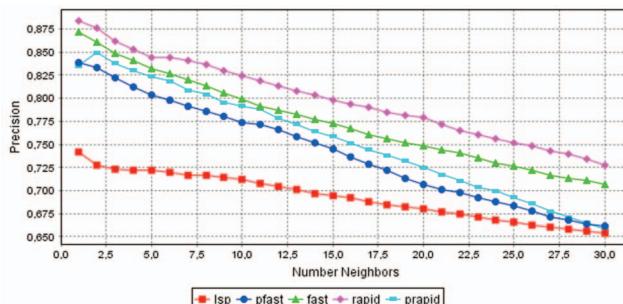


Fig. 7. Precision by neighborhood hit for the COREL data set.

Neighborhood hit is a value extremely influenced by proper preprocessing tasks (choice of features and similarity) and by the boundaries between groups. To help determine to which extent a 2D mapping (or 3D for that matter) is actually reflecting in the visual space the same distances as in the original space, a distance plot is commonly employed. A distance plot is a scatter plot of distances in original space versus distances in the visual space. Ideally, the point cloud of that plot will lie close to the 45° slope.

Figure 8 shows distance plots (also known as stress curves) for the trees and for LSP. It can be seen that, for the COREL data set, the trees do a better job than the projection, with the original NJ algorithm performing best, followed closely by FAST. Promotion dissipates the distances around the diagonal, however, keeping the slope trend and the distribution very satisfactory. For the MEDICAL data set, LSP performs better at reconstituting the distances according to dissimilarity in original space, with the trees performing worse and very similarly among themselves. Dissipation under promotion is also observed, but to a lesser extent. Promotion over the FAST-NJ is not shown, but it had the same effect of slight point dispersion in both cases.

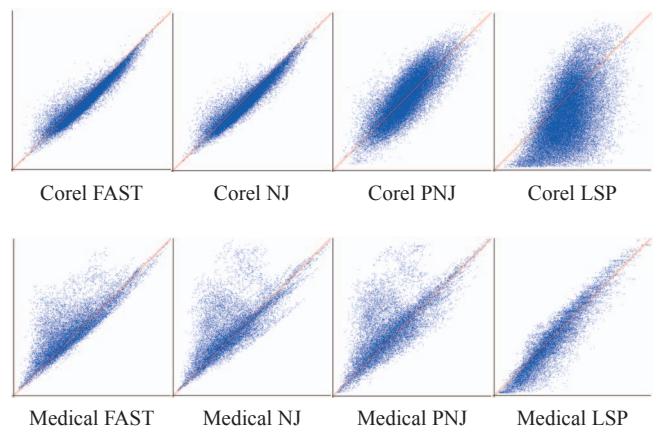


Fig. 8. Distance plots, with original space in horizontal axis vs. visualization space in vertical axis. COREL data set is top row, MEDICAL data set is bottom row. Techniques are FAST-NJ, NJ, PNJ, and LSP.

4.4 Efficiency and Scalability

Fast projections such as LSP, which takes time $O(n\sqrt{n})$, take advantage over the trees under time efficiency issues. However, the new versions of the NJ, based on FAST and Rapid algorithms, have dropped that difference. In fact, processing times for the FAST version are faster than the projection. The promotion procedure is efficient, so its cost is not significant.

Table 3 shows the time comparison between the various versions of the tree, as well as the projection, for the three test data sets. From that table, it can be seen that the trees have very competitive time consumption, with the FAST versions largely outperforming all the others.

Table 3. Layout Building Times (seconds).

Technique	COREL	MEDICAL	OBJECTS
Original NJ	3.0474	0.5	394.383
Promoting Original NJ	3.064	0.506	396.507
Fast NJ	0.0936	0.0462	4.976
Promoting Fast NJ	0.1104	0.052	6.1
Rapid NJ	2.366	0.373	261.971
Promoting Rapid NJ	2.3828	0.3788	262.178
LSP	0.7182	0.2068	61.564

Analyzing the processing times of Table 3 together with the precision discussions in the previous section, the scenario seems to point to a setup where global, initial layouts would be carried out using projections or the FAST or PFAST versions of the tree. More focused vi-

visualizations would take advantage of the better precision of the Rapid variation of NJ. Next, we illustrate the use of the tree as basis for a visual classification system.

5 VISUAL IMAGE AND TEXT CLASSIFICATION

Techniques have been proposed for employing point placement, particularly multidimensional projections and graphs, to display collections of images [6, 15, 26, 27, 33, 39, 42] with or without support for classification tasks, some of which already employ tree visualizations to reflect a hierarchical clustering [18, 19], serving the main purpose of supporting the browsing of collections on a visual space. However, to our knowledge, the tight coupling between a visualization system and an image mining environment is not available to date.

We focus on the goal of visually supporting image classification due to the challenges facing the subject, some of which we believe can be resolved by extensively involving the user via visual analysis tools. One of the challenges is the difficulty in understanding the reasons for failure of automatic classification procedures (e.g., what points are misclassified and why). Another recurrent situation is when users do have a proper automatic classification procedure in place, but wish to ‘see’ it and change it at will if necessary. The third scenario is when the current labeling of the data set is not appropriate for the goals of the user. There are different definitions of similar images and class labeling may acquire different configurations depending on the target applications. For both browsing and mining, we believe that similarity trees offer a valuable layout, for their capacity of local as well as global space reconstruction.

We have implemented a suite of tools for iterative classification, whereby the user, starting from either a labeled or a non-labeled data set, builds up training sets into classified sets by mixing automatic and visually supported manual classifications. The system, complete with evaluation tools, is added to a locally built data flow visualization prototype, named VisPipeline.

We describe some of the tools built around the concept of tree visualization to support tasks in such an integrated classification setup. The main features are presented in the next subsections and we finish by mentioning additional functionality implemented.

5.1 Layers of Similarity

Similarity trees allow the viewer to analyze various degrees of similarities and to explore the data in various levels of detail with practically the same strategy. Well resolved groups are laid out as branches, and the delimitation of such groups is given by the top layers of the unrooted tree, suggesting branch-based interaction. What is motivating about similarity trees is, therefore, that the model is capable of describing, to a certain extent, the organization of the similarity relationship defined or chosen by the analyst. If the dissimilarity measure is capable of distinguishing groups and subgroups, they should appear as branches in the tree.

Figure 9 shows an example of some of the drilling down capabilities of the trees. We first display the COREL data set using PNJ (Figure 9a). Then, since the data was already labeled, we perturbed the distance matrix by approaching the nodes that belonged to the same class and increasing the distance between nodes that did not belong to the same class. The result of perturbing the COREL distance matrix by 18% can be seen in Figure 9b. Although that type of perturbation can only be done when a correct label is established for the collection, it is very useful to separate classes in branches, while still maintaining, within each class, the structure and ordering of the original similarity. Figure 9c shows a branch containing flower pictures, and within that, evidence that color features may be separating the group further (that is, pink and most yellow flowers, orange and red ones, then white flowers). Selections can be made by circling groups of nodes in a free curve (Figure 9b) or by just clicking on any node. Clicking on a node selects either the node itself (if it is a leaf) or a branch. What the perturbation tool does, visually, is to redistribute those nodes that were close to the main skeleton of the tree towards their proper branch. Drilling can, of course, be done in any tree, but perturbation has interesting visualization value. A proper tool to generate a family of perturbed matrices

according to users’ specifications, as well as corresponding views, is made available. The various visualizations are displayed using a slider that moves from one perturbation to the next.

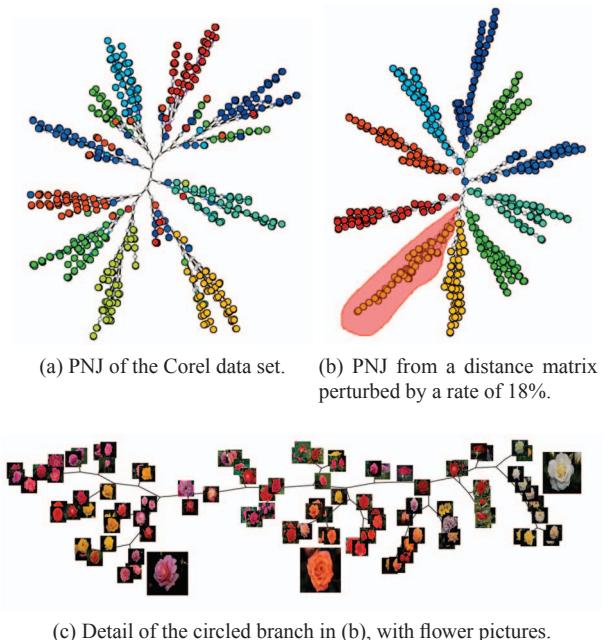


Fig. 9. Branch selection and exploration of the COREL data set.

To confirm the evidence that similarity trees keep, at lower levels, the same properties observed at the top level, we have plotted, for the visualization in Figure 9b and for each one of its branches, the neighborhood preservation and the distance plots. Figure 10 shows these plots for the full set and for the flower branch, but the results are consistent for all 8 branches. Figures 10d and 10c reveal that the local neighborhood and distance reconstitution follow the same patterns, or better, than the layout at global levels. All these evaluation functions, plus the ones used to evaluate the trees in Section 3, are readily available either as modules of the pipeline or on the menu and toolbox of the visualization window.

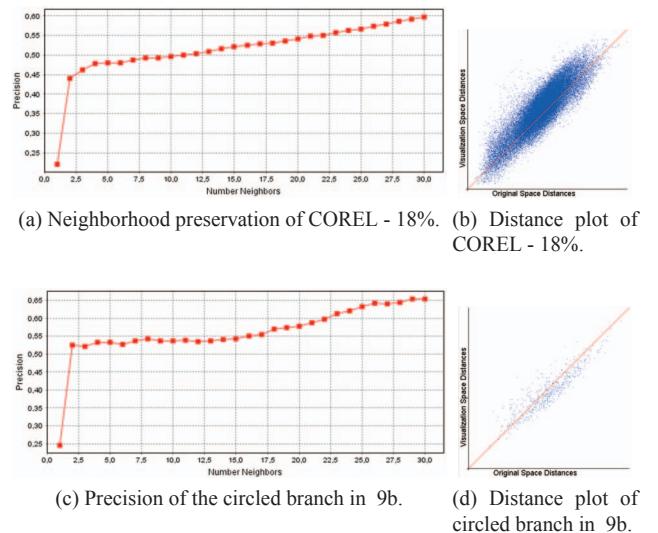


Fig. 10. Precision consistency in branches.

The matrix perturbation process, besides supporting visualization of pre-defined classes, helps understand the layering of similarity and

supports our visual, interactive and iterative semi-automatic classification framework. When the matrix is perturbed, precision values improve even when the layouts are evaluated against the original matrices. One way to use that is to employ perturbation after the classification process, to help gather equally labeled data in the same branches and then browse branches to adjust wrongly classified ones. The process can also be useful to create new labeled data sets or to change pre-labeled ones to adapt to a new categorization scheme.

5.2 Manual Classification Tools

A visual method for classification has the advantage that the user is immediately exposed to the results of false positives, false negatives, mismatches and outliers. It also opens space for user's influence on the setup and on the outcome of the classification process. Users may want to relabel the training set in order to correct final classification or to assign labels themselves to unlabeled data sets. Relevance feedback techniques [3] could also use visualization tools. To support these tasks, we devised two manual classification tools, one to define an initial labeling for a non-labeled data set, and another to adjust categories of individuals or groups of individuals while browsing through the images. Both techniques are combined to perform a fast labeling or re-labeling task.

Figure 11 illustrates both tools. The OBJECTS data set was used in the example. Figure 11a shows the label-by-selection tool. Once a tree visualization is displayed for any data set, labeled or not, the user can define new labels (by color) for whole branches or for any group of nodes. Branches can be selected just by clicking at the top node of the branch, and other groups are selected using the free curve selector mentioned before. Each time this labeling procedure is started, a new scalar field is created with the new labels. When the user moves from one group to another, the labels are recolored for consistency. The latest selection in the example is colored red. The initial selection is blue. These are the extremes of the color table used in the examples. Once any color scheme is available for the data set, the user can browse images and change labels individually. Figure 11b illustrates this tool. After selecting a particular branch for browsing, users can change labels freely. New labels are defined by choosing new names for them or employing an already existing label name. The picture shows the selection of a few images on a branch. The name of the class is changed to "not round". Then, in the branch, whose visualization is seen in another window, the corresponding nodes change color. In Figure 11b, the change in color is highlighted by an ellipse. Once the data is fully labeled, these labels can be saved together with the original point or similarity matrix and used as training set for an automatic classifier or to evaluate other classifications of the same data.

5.3 Iterating Towards a Fully Classified Data Set

With the proper visualization tools, it is possible to envisage an iterative process to converge from a smaller labeled data set to a proper categorization of even very large data sets. A combination of automatic and user-driven tasks should compound an environment fit to help a considerable number of applications.

We have built another set of tools into VisPipeline to accomplish that employing the similarity trees as one of the main visualization techniques, since similarity is at the core of many classification tasks. Besides the tools and evaluation modules illustrated above, automatic classification algorithms are being progressively built into the system. A network of modules summarizing a classification setup is shown in Figure 12. One of the three pipelines in that picture is the visualization of the test set, another is the visualization of the training set, and the third is the classification pipeline that feeds the test set into the SVM classification component. Readers are capable of loading similarity or feature matrices and feeding them into visualizations or classifiers. The classification returns a new scalar field on the test data set, which can then be changed and adjusted by the user through the manual procedures explained in Section 5.2. The resulting classification can be saved as a copy of the distance matrix or as a copy of the data matrix that originated the classification, together with class for



Fig. 11. Visual support for manual definition of labels.

future use. The saved data matrix can then be loaded as a training set in subsequent classification steps.

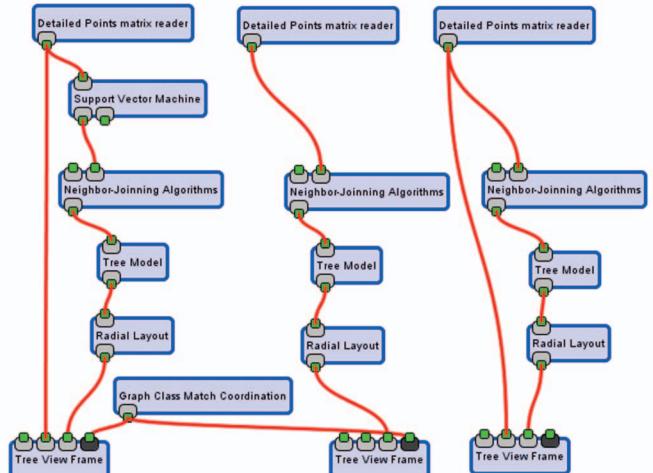


Fig. 12. Classification pipeline with visualization pipelines for test and training data sets.

We tested that pipeline in various ways, but we illustrate progressive classification by extracting a data subset of the COREL data set containing 500 pictures and iteratively classifying it. We start with a labeled subset of 300 photos out of the 500 and add to the set 50 photos at each step. At each step, the classifier (SVM in this case) produces a result for the test set with added photos, which is then manually corrected by the user to increase the size of the classified set. This, in turn, can serve as a training set once new pictures are added. Figure 13 illustrates the process. In particular, Figure 13a shows the visualization of one training data set. The test data set colored by corresponding target classes is shown in Figure 13b. The data set after classification is shown in Figure 13c. It can be seen that the great majority of points were correctly classified.

When there is available information of target classes, that is, if the ground truth is known, it is possible to verify the classification results. Besides the usual values reflecting the number of correctly clas-

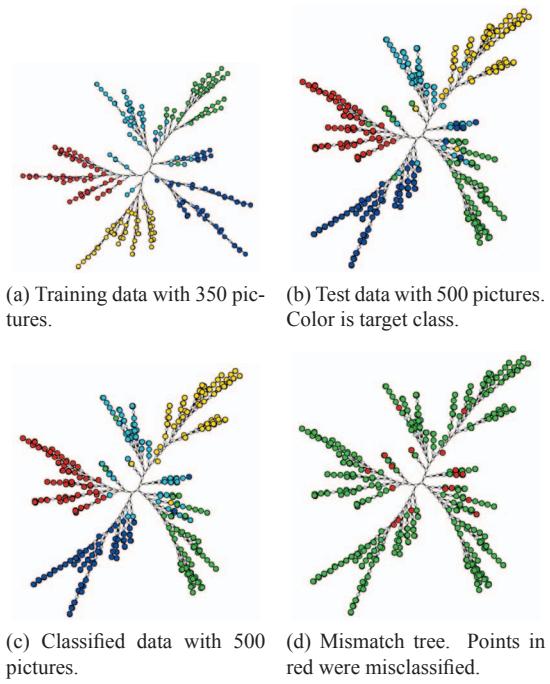
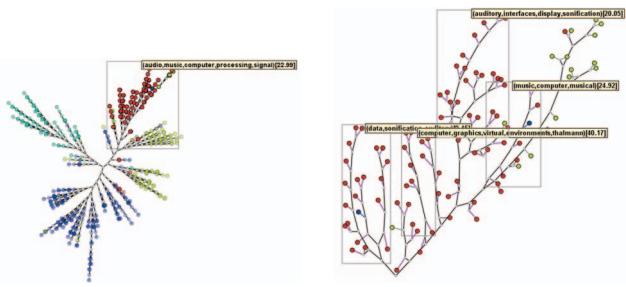


Fig. 13. Two steps of classification. Misclassified points: 25/500 (5%).

sified instances and other conventional measures of classifications, it is possible, using another supporting tool, to verify the difference between the correct classification and that obtained using also the trees, precisely locating points that were mistakenly classified. Figure 13d shows the results of this layout matching tool applied to the trees in Figures 13b and 13c. By interacting with that users may be inspired to find the reasons for classification failure in some cases. The *Graph Class Match Coordination* module, which realizes the matching of trees, can be seen in the pipeline of Figure 12.

Coordination is an integral part of the system. Selections can be coordinated amongst the various projections and trees. One advantage of coordinating trees with projections is to be able to identify, for a group well formed in the projection, a possible structure within the group in the tree with possible formation of subgroups. This can compensate for the poor local neighborhood reconstruction common to most projection techniques. Also, for groups that are not well formed in the projection, the coordination with trees may suggest hypotheses related to similarity calculation or to the feature extraction process.



(a) Global NJ tree. Points highlighted on top branch correspond to area highlighted in Figure 1.
(b) Detail of one of the branches from highlighted portion in 14a.

Fig. 14. NJ tree corresponding to same text data set as Figure 1.

The processes illustrated above can serve the purpose of supporting classification of many types of multidimensional data, since the visualizations take as starting point feature sets or similarity calculations. The system is prepared to handle most functions presented

before for any data set. What differs is the handling of the manual classifier (the individual labeler) that has to be attached to a proper browser. Additional functions may, naturally, be needed for other data types. The system currently allows exploration and classification of text. Classification procedures are still rudimentary, but there is effective functionality for determination of topics in branches. Figure 14 shows an example for a text data set. Starting from the same LSP projection of Figure 1, after determining topics of various areas on that map, the group with predominantly red points is selected on the LSP and, by coordination, shown on the corresponding tree (Figure 14a). From that part of the tree, whose general content reflects computer audio, another branch is selected and highlighted in Figure 14b. One can see that, in the selected branch, different sub-branches are treating the subjects of computer music, auditory interfaces, data sonification and audio for virtual environments.

It is also possible to coordinate, by identity, visualizations of data types of different natures (such as image and text), provided the file identifiers for image and corresponding text are named equally. For instance, one can visualize and coordinate textual and image representations of protein sequences or of X-rays and their reports, and so on.

6 CONCLUSIONS AND FUTURE WORK

This paper proposes solutions for processing time and visual overload problems of phylogenetic similarity trees for the purpose of multidimensional data visualization. The first problem was handled by employing faster Neighbor Joining implementations or Neighbor Joining strategy. Overload was relieved by a linear graph rewriting procedure based on node promotion.

Numerical and visual comparisons have shown that the FAST algorithm is actually the fastest with small loss in precision, and that the Rapid algorithm accelerates its original version resulting in the same structure, therefore the same precision as the original NJ tree. Use of visual space has gained considerable improvement, since an average of 51% of the virtual nodes are eliminated by promotion. These procedures should endorse larger employment of similarity trees as a precise and fast technique that supports analysis in both overall and detail tasks with the same properties. Trees are capable of examining the original space locally and globally with similar precision. The tree building and visualization processes need no parametrization from the user.

An innovative form, based on the trees, of a classical classification application was also presented. The set of tools for the application of visual mining of images is made possible by the properties offered by trees complemented by the possibilities offered by multidimensional projections.

Trees are built from similarity matrices or from vector spaces followed by similarity calculations, giving it flexibility to work with different types of information as input. While discussions on proper feature spaces or similarity functions are out of the scope here, we intend to employ the trees as a platform to support converging to proper definitions of vector space and similarity relationships. Additionally, our next steps include building trees from very large, disk-based data sets [41], to extend the use of similarity based visualizations to massive data sets. Another very promising venue is to use the matrix perturbation process presented in this paper as a means for feedback into the classification process. The system separate pipelines are to be integrated into a fully functional visual classification system.

ACKNOWLEDGMENTS

Authors wish to acknowledge CNPq (INCT - MACC 573710/2008-2 and 301295/2008-5), FAPESP and FAPEMIG Brazilian financial agencies. We are grateful to Fernando V. Paulovich and Rafael M. Martins for coding the core of VisPipeline.

REFERENCES

- [1] C. Bachmaier, U. Brandes, and B. Schlieper. Drawing Phylogenetic Trees. In *Algorithms and Computation*, volume 3827 of *Lecture Notes in Computer Science*, pages 1110–1121, 2005.
- [2] B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies. *ACM Transaction on Graphics*, 21(4):833–854, October 2002.
- [3] S. Büttcher, C. L. A. Clarke, and G. V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, Cambridge, MA, USA, 2010.
- [4] M. Chalmers. A Linear Iteration Time Layout Algorithm for Visualizing High-Dimensional Data. In *Proceedings of the 7th Conference on Visualization (VIS'96)*, pages 127–131, San Francisco, CA, USA, 1996.
- [5] Y. Chen, L. Wang, M. Dong, and J. Hua. Exemplar-based Visualization of Large Document Corpus. *IEEE Transactions on Visualization and Computer Graphics*, 15:1161–1168, 2009.
- [6] L. Cinque, S. Levialdi, A. Malizia, and K. Olsen. A Multidimensional Image Browser. *Journal of Visual Languages and Computing*, 9(1):103–117, 1998.
- [7] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.
- [8] T. M. Cover and P. E. Hart. Nearest Neighbor Pattern Classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27, 1967.
- [9] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall/CRC, 2nd edition, 2000.
- [10] A. M. Cuadros, F. V. Paulovich, R. Minghim, and G. P. Telles. Point Placement by Phylogenetic Trees and its Application for Visual Analysis of Document Collections. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology (VAST'2007)*, pages 99–106, Sacramento, CA, USA, 2007.
- [11] J. Daniels, E. W. Anderson, L. G. Nonato, and C. T. Silva. Interactive Vector Field Feature Identification. *IEEE Transactions on Visualization and Computer Graphics*, 16:1560–1568, 2010.
- [12] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc, New York, NY, USA, 1973.
- [13] P. A. Eades. A Heuristic for Graph Drawing. *Congressus Numerantium*, 42:149–160, 1984.
- [14] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. *Fundamentals of Algebraic Graph Transformation*. Springer, 2006.
- [15] D. M. Eler, M. Y. Nakazaki, F. V. Paulovich, D. P. Santos, G. F. Andery, M. C. F. Oliveira, J. Batista-Neto, and R. Minghim. Visual Analysis of Image Collections. *The Visual Computer*, 25(10):923–937, 2009.
- [16] I. Elias and J. Lagergren. Fast Neighbor Joining. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580, pages 1263–1274, 2005.
- [17] J. Evans, L. Sheneman, and J. Foster. Relaxed Neighbor Joining: A Fast Distance-Based Phylogenetic Tree Construction Method. *Journal of Molecular Evolution*, 62(6):785–792, 2006.
- [18] J. Fan, Y. Gao, and H. Luo. Hierarchical Classification for Automatic Image Annotation. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 111–118, New York, NY, USA, 2007.
- [19] J. Fan, Y. Gao, and H. Luo. Integrating Concept Ontology and Multitask Learning to Achieve More Effective Classifier Training for Multilevel Image Annotation. *IEEE Transactions on Image Processing*, 17(3):407–426, march 2008.
- [20] O. Gascuel and M. Steel. Neighbor-Joining Revealed. *Molecular Biology and Evolution*, 23(11):1997–2000, 2006.
- [21] G. Griffin, A. Holub, and P. Perona. Caltech-256 Object Category Dataset. Technical Report 7694, California Institute of Technology, 2007.
- [22] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support Vector Machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.
- [23] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, NY, USA, 2nd edition, 2002.
- [24] J. Li and J. Z. Wang. Automatic Linguistic Indexing of Pictures by a Statistical Modelin Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1075–1088, 2003.
- [25] T. Mailund, G. S. Brodal, R. Fagerberg, C. N. S. Pedersen, and D. Phillips. Recrafting the Neighbor-Joining Method. *BMC Bioinformatics*, 7(29), 2006.
- [26] B. Moghaddam, Q. Tian, N. Lesh, C. Shen, and T. S. Huang. PDH: A Human-Centric Interface for Image Libraries. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 901–904, New York, NY, USA, July 2002.
- [27] G. Nguyen and M. Worring. Interactive Access to Large Image Collections using Similarity-Based Visualization. *Journal of Visual Languages & Computing*, 19(2):203–224, 2008.
- [28] F. V. Paulovich, D. M. Eler, J. Poco, C. P. Botha, R. Minghim, and L. G. Nonato. Piecewise Laplacian-based Projection for Interactive Data Exploration and Organization. *IEEE Computer Graphics Forum, Proceedings Eurovis 2011*, 30(3):1091–1100, 2011.
- [29] F. V. Paulovich and R. Minghim. HiPP: A Novel Hierarchical Point Placement Strategy and its Application to the Exploration of Document Collections. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1229–1236, 2008.
- [30] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Least Square Projection: A Fast High Precision Multidimensional Projection Technique and its Application to Document Mapping. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):564–575, 2008.
- [31] F. V. Paulovich, C. T. Silva, and L. G. Nonato. Two-Phase Mapping for Projecting Massive Data Sets. *IEEE Transactions on Visualization and Computer Graphics*, 16:1281–1290, 2010.
- [32] C. Plaisant, J. Grosjean, and B. B. Bederson. SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. *IEEE Symposium on Information Visualization*, page 57, 2002.
- [33] K. Rodden, W. Basalaj, D. Sinclair, and K. R. Wood. Evaluating a Visualization of Image Similarity as a Tool for Image Browsing. In *Proceedings of IEEE InfoVis*, pages 36–43, San Francisco, CA, USA, 1999.
- [34] G. Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 1. World Scientific Publishing Company, 1997.
- [35] N. Saitou and M. Nei. The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987.
- [36] M. Simonsen, T. Mailund, and C. N. Pedersen. Rapid Neighbour-Joining. In *Proceedings of WABI 2008*, pages 113–122, Karlsruhe, Germany, September 2008.
- [37] J. A. Studier and K. J. Kepler. A Note on the Neighbour-Joining Method of Saitou and Nei. *Molecular Biology and Evolution*, 5:729–731, 1988.
- [38] E. Tejada, R. Minghim, and L. G. Nonato. On Improved Projection Techniques to Support Visual Exploration of Multidimensional Data Sets. *Information Visualization*, 2(4):218–231, 2003.
- [39] Q. Tian, B. Moghaddam, and T. S. Huang. Visualization, Estimation and User-Modeling for Interactive Browsing of Image Libraries. In *ACM International Conference on Image and Video Retrieval*, pages 7–16, London, UK, 2002.
- [40] T. J. Wheeler. Large-Scale Neighbor-Joining with NINJA. In *Proceedings of WABI 2009*, pages 375–389, Philadelphia, PA, USA, 2009.
- [41] T. J. Wheeler. Large-Scale Neighbor-Joining with NINJA. In *Proceedings of the 9th International Conference on Algorithms in Bioinformatics*, pages 375–389, Philadelphia, PA, USA, 2009.
- [42] M. Worring, O. de Rooij, and T. van Rijn. Browsing Visual Collections Using Graphs. In *Multimedia Information Retrieval*, pages 307–312, Augsburg, Germany, 2007.

Artigo: Semi-Supervised Dimensionality Reduction based on Partial Least Squares for Visual Analysis of High Dimensional Data

Este apêndice apresenta o conteúdo completo de um artigo publicado no *Computer Graphics Forum* (CGF). Uma visão geral desse trabalho foi apresentada no Capítulo 4 desta tese.

Semi-Supervised Dimensionality Reduction based on Partial Least Squares for Visual Analysis of High Dimensional Data

Jose Gustavo S. Paiva^{1,3}, William Robson Schwartz^{2,4}, Helio Pedrini² and Rosane Minghim¹

¹USP, Sao Carlos, Brazil, ²UNICAMP, Campinas, Brazil, ³UFU, Uberlandia, Brazil, ⁴UFMG, Belo Horizonte, Brazil

Abstract

Dimensionality reduction is employed for visual data analysis as a way to obtaining reduced spaces for high dimensional data or to mapping data directly into 2D or 3D spaces. Although techniques have evolved to improve data segregation on reduced or visual spaces, they have limited capabilities for adjusting the results according to user's knowledge. In this paper, we propose a novel approach to handling both dimensionality reduction and visualization of high dimensional data, taking into account user's input. It employs Partial Least Squares (PLS), a statistical tool to perform retrieval of latent spaces focusing on the discriminability of the data. The method employs a training set for building a highly precise model that can then be applied to a much larger data set very effectively. The reduced data set can be exhibited using various existing visualization techniques. The training data is important to code user's knowledge into the loop. However, this work also devises a strategy for calculating PLS reduced spaces when no training data is available. The approach produces increasingly precise visual mappings as the user feeds back his or her knowledge and is capable of working with small and unbalanced training sets.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms

1. Introduction

Dimension reduction is an important task involved in data analysis in general, and in particular for data sets that reach a high number of dimensions or attributes. It is applied in a visual analysis pipeline usually at the start of the process of visual mapping, whereby a method, such as Principal Component Analysis (PCA), is employed to find principal directions that recombine coordinates in new and fewer dimensions. To allow visualization of multidimensional data, various dimension reduction techniques have been employed to map from multidimensional to bidimensional spaces by reducing the dimension of the original space to two.

Two drawbacks of available projections are the inability to model the transformation in a way that it can be applied to data sets other than the ones used for the original data mapping, and also the difficulty in considering user's knowledge to influence the layout.

Partial Least Squares (PLS) [Wol85] is a highly effective technique that is usually employed for many different tasks involving automatic data segregation or regression, taking as input a labeled training set that is small in regards to the

complete data, and can be unbalanced. For visual analysis purposes, these capabilities can be valuable.

In this paper, we propose a novel mapping process for visual analysis purposes that improves available strategies, by allowing users to input their knowledge into the dimension reduction process by means of the training set, and, from that, being able to create a reusable model for improved dimension reduction and visualization of larger data sets. Although labeling of the training set is an important step of the process, we also develop a strategy for PLS reduction in cases where no labeling of any part of the data set is available.

The following sections describe the background in 2D mapping techniques from multidimensional spaces, the basic concepts of PLS, our approach to visual mapping using PLS and, finally, the results with various text and image collections as well as a comparison with other approaches for dimension reduction purposes.

2. Dimension Reduction for Visualization and Visual Mining

In visual analysis applications, dimension reduction is sometimes achieved by selecting relevant attributes of the data set from which multi-attribute visualizations are generated. This is the case of the VHDR [YWRH03] and DOSFA [WPW03] approaches, that, based on measures between different data dimensions, support user's selecting and ordering dimensions.

The main motivation for selecting instead of combining variables for visualization is to maintain the meaning of the attributes themselves during exploration. However, in many applications such as visual analysis of text and image sets, the number of attributes easily reach hundreds or thousands, which makes attribute-based exploration unfeasible. Additionally, dimension reduction can improve results in mining as well as visual mining applications [TLKT09]. In such cases, it is important to find latent spaces with manageable number of dimensions that manage to represent well or to improve data spaces.

A second role played by dimension reduction is to provide a mapping to visual spaces by reducing original dimensionality directly to 2D or 3D spaces. When applied to that purpose, they are frequently called multidimensional projections, or simply projections. Depending on the goal of the mapping, projections either aim at achieving discriminability of groups or reproducing dissimilarity relationships existing in original spaces.

Projection techniques are usually based on Multidimensional Scaling (MDS) methods, with various mathematical foundations, such as spectral decomposition from transformations on similarity matrices [Tor65, BN03, KCH02]. Some spectral-based methods, such as ISOMAP [TdSL00], can also deal well with non-Euclidean distances, which is useful for visualizing data generated from various feature description methods. Although representation properties and processing time have been largely improved in later formulations of MDS algorithms, the global aspect of these methods impair their use in applications where it is important to capture or emphasize near neighborhoods.

Some spectral-based methods, such as Locally Linear Embedding (LLE) [RS00], Landmark MDS (LMDS) [dST04] and Pivot MDS [BP07], can actually capture local aspects of data. However, the global eigendecomposition scheme in these methods prevents them from being used in applications that redefine local relationships since an update forces recalculations that are global to the model they produce.

Some techniques have been designed to deal with samples of data to cope with high computational costs of decompositions. One example is Sammon projection [PDRDK99], that is meant to improve dimension reduction and mapping based on optimization methods [Kru64, BBKY06], typically heavy

in computational cost. Although this idea could be used to embed user's knowledge by first mapping a sample set and then adjusting the others, its formulation does not produce a final model to be reused in other data set mappings.

There exist local projections designed for visualization purposes. Paulovich et al.'s [PNML08] Least Squares Projection (LSP) employs a force-based scheme to first position a subset of the samples, mapping the remaining instances through a Laplace-like operator. It results in a large linear system that is strong in local feature definition. Its derived methods, such as Piecewise Laplacian-based Projection (PLP) [PEP*11] and Local Affine Multidimensional Projection (LAMP) [JCC*11] have progressively managed to allow redefinition of the mapping matrix under user's intervention over a first mapping of sampled instances. These have more local formulations and are more flexible as far as rearranging the visual mapping. However, they are not capable of using a mapping to project data sets other than the one used to produce the mapping. Nor do they support user's choice of number of dimensions effectively.

The most common dimension reduction technique employed for data analysis in basically any field of science is by far the Principal Component Analysis (PCA) [Jol02], which finds principal directions in a covariance matrix calculated over the data dimensions. Regardless its high computational cost as a dimension reduction technique, PCA is very effective. It has also been used numerous times for mapping data visually into 2D or 3D. In such case, however, PCA performs worse, since choosing only the two first principal directions frequently fails to deliver proper data segregation.

Also based on the principle of finding latent spaces in feature data sets, Partial Least Squares (PLS) [Eld04, Wol85] has been recovered from the field of statistics to find many current uses in analysis of high-dimensional data, such as discrimination, feature selection, treatment of missing data problem and regression [BS07]. Some of the work on the large number of applications for PLS also employ MDS techniques (such as [LN06]) but they do not support visual analysis of one of the other, or allow for iterations or progressive refinement of the process. In this work, we show PLS to be a very flexible and precise tool for visual analysis of data sets by supporting user's feedback into the dimension reduction process, being capable of working with low number of samples, and supporting the reuse of the model to visualize increasing data sets. Reduced data sets improve original data sets regarding data discrimination.

Reduced data spaces can be visualized by MDS strategies as well as multiple axis visualization, such as Radviz [HGP99]. The reduced data set, can also be effectively visualized employing similarity trees, such as Neighbor-joining (NJ) trees, which have been previously used as support for visual classification tasks [PFCP*11].

3. Partial Least Squares for Visualization of Multidimensional Data Sets

Partial Least Squares is a class of statistical methods used to model relations between sets of observed variables by the estimation of a low dimensional latent space. Its goal is to estimate a low dimensional space that maximizes the separation between samples with different characteristics, causing samples from the same class to be clustered in the latent space. Here we describe its uses, its formulation and how to apply PLS for data classification and dimension reduction.

The underlying assumption of PLS methods is that the observed data is generated by a system or process which is driven by a small number of latent variables. This way, it reduces the number of dimensions prior to the estimation of the regression coefficients, so that the influence of high dimensional noisy samples is reduced, thus improving discrimination results [Gar94].

PLS was created by Herman Wold in the 1970s [Gel88] and has been exploited in several areas, such as Chemometrics [BGJ*97, LGB*95], Bioinformatics [NR02, BS07] and Neurosciences [NOM*02]. Recently, PLS has been successfully applied to Computer Vision problems considering dimension reduction, regression and data classification [SKHD09, KHD11].

3.1. Dimension Reduction and Regression Based on Partial Least Squares

With the advantages of being designed to work in problems containing high dimensional data and very few samples [Gel88], PLS estimates latent variables as a linear combination of the original variables in a matrix X , composed of variables used to describe samples, and a matrix Y containing a set of response variables (when a single response variable is considered, a vector y is used instead). A description of the PLS decomposition and latent space estimation is given as follows.

For a problem with n samples described by d variables each, stored in a mean-centered matrix $X_{n \times d}$, associated to k response variables, stored in a mean-centered matrix $Y_{n \times k}$, PLS estimates a p -dimensional space ($p \ll d$) by performing the decomposition of X and Y into

$$X = TP^T + E \quad \text{and} \quad Y = UQ^T + F$$

where $T_{n \times p}$ and $U_{n \times p}$ are matrices containing the latent variables, matrices $P_{d \times p}$ and $Q_{k \times p}$ represent the loadings, and matrices $E_{n \times d}$ and $F_{n \times k}$ are the residuals. An approach to performing the decomposition above employs the nonlinear iterative partial least squares (NIPALS) algorithm [Wol85]. NIPALS estimates a set of projection vectors w_i ($i = 1, 2, \dots, p$), which are stored in a matrix $W = (w_1, w_2, \dots, w_p)$, such that

$$[\text{cov}(t_i, u_i)]^2 = \max_{|w_i|=|u_i|=1} [\text{cov}(Xw_i, Yu_i)]^2 \quad (1)$$

where $|w_i|$ and $|u_i|$ denote the 2-norm of vectors w_i and u_i , respectively. t_i and u_i represent the i -th columns of matrices T and U , and $\text{cov}(t_i, u_i)$ is the sample covariance between latent vectors t_i and u_i .

The NIPALS algorithm extracts the latent variables t_i and u_i iteratively. After each iteration, matrices X and Y are deflated by subtracting their rank-one approximations as

$$X_{i+1} = X_i - t_i p_i^T \quad \text{and} \quad Y_{i+1} = Y_i - t_i q_i^T$$

where X_i and Y_i are the data representation for the i -th iteration, where $X_1 = X$ and $Y_1 = Y$, and p_i and q_i denote the i -th columns of the matrices P and Q , respectively. After the extraction of p projection vectors, the p -dimensional representation of $X_{n \times d}$ is given by $T_{n \times p}$, which is used to extract the regression coefficients $\beta_{d \times k}$ as $\beta = W(P^T W)^{-1} T^T Y$. Finally, the regression responses, Y_v , for a feature vector $v_{d \times 1}$ is obtained by $Y_v = \bar{Y} + \beta^T vS$, where $\bar{Y}_{1 \times k}$ is the sample mean of each variable of Y and $S_{1 \times k}$ is the standard deviation of the variables in Y .

As pointed out earlier, PLS performs dimension reduction in a supervised manner, differently from PCA. Due to its supervised nature, PLS estimates latent spaces that focus on the discrimination of the data. Therefore, the reduced space presents a better separation, which aids content-based visualization.

3.2. Data Classification Based on Partial Least Squares

According to the PLS formulation above, after the selection of the number of latent variables, referred to as *factors*, the NIPALS algorithm is applied to estimate a low dimensional representation of the original data.

3.2.1. One-against-all Classification

Aiming at maximizing the discrimination between C different classes, the one-against-all classification scheme estimates C PLS models considering single response variables [SGCD12]. This way, the response variable Y , represented by a matrix in Section 3.1, becomes a vector, y , and its entries have class indicators. In this work, we set +1 for positive samples (samples belonging to the class being modeled) and -1 for negative samples (remaining training samples). When a test sample is presented, it is projected to each model, resulting in a set with C responses (one per class) and the best matching class is associated to the model presenting the highest regression response.

3.2.2. Multi-class Classification

Differently from the one-against-all scheme, the multi-class creates a single PLS model containing multiple response variables. For a problem with C classes, the response variable Y , represented by a matrix in Section 3.1, has C columns, each one corresponds to one class and indicators variables are used to identify which samples belong to a

given class. In this work, the value +1 is set to $Y_{i,j}$ if the i -th sample belongs to the j -th class, otherwise, it receives 0. For a test sample projected onto the model, C responses are obtained and the best matching class is the one presenting the highest regression response.

Although the multi-class approach presents a faster latent space estimation compared to the one-against-all classification scheme, in general, the latter presents higher classification rates.

3.3. A Visual Analysis Approach Based on PLS Reduction

In this work, PLS is used in two ways for mapping a data set to low dimensional spaces targeting visualization. One way is to employ the conventional PLS dimension reduction, which considers the multi-class dimension reduction to p factors, using the matrix T , that is, the low dimensional representation of the data stored in matrix X (see Section 3.1), to generate a new reduced space with dimension p . The other way that its formulation may result in a low dimensional space is by using the responses in each class, estimated by the one-against-all classification, as coordinates for a projected sample in the reduced space. In this case, the reduced space has C dimensions.

In both cases, in order to visualize patterns in the original space through the reduced space, one can apply any of the available projection or point placement techniques mentioned in Section 2. To understand the distribution of the data over the final reduced dimensions, analysts can make use of any of the multi-axis visualization techniques available. We find that Radviz [HGP99] is useful in this case to help identify the contribution of particular latent dimensions in the data distribution, but tested with various others (see Section 4).

PLS needs as input a training set. However, we have also devised an approach to PLS mapping of unlabeled data sets. The approach to labeled and unlabeled data sets work as follows:

Approach 1 Data sets with labeled training sets.

1. Model creation phase: a PLS model is built from the labeled training set provided by the user. The user has the choice of using a multi-class or a one-against-all classification process.
2. Model application phase: the test data set is applied to the model created in the previous phase. The result is a reduced space that includes all points applied to the model, with dimensions either p or c depending on the choice of the classification process.
3. Visual mapping phase: the reduced model is projected onto the visualization plane using a point placement strategy.

Approach 2 Data sets with unlabeled training sets.

1. Label creation phase: the data is clustered attributing to each point the label of the cluster that belongs. The result is a data set labeled by clustering.
2. Model creation phase: it is the same as the model creation phase of approach 1, only the classes are now the cluster labels and the training set is sampled from the whole data set according to some sampling strategy.
3. Model application phase: it is also the same as in approach 1, only now the data set has no labels. The labels are assigned according to the winning class of the classification method.
4. Visual mapping phase: the reduced model is projected onto the visualization plane using a point placement strategy.

In either approach, a training set can be built by sampling a previously labeled data set. In our system and experiments, there are two forms of sampling. In the first form, the labeled data is clustered and an equal number of samples for each cluster is selected. Half of these are the closest points to each cluster centroid and the other half are the points in the cluster that are the farthest apart from the centroid.

The second form of sampling builds in the process a semi-supervised approach. From a distance-based initial visualization of the whole data set or a subset of it, the user manually chooses points from the data sets. In our system, the user can select those points in various different ways on the visualization itself. For the similarity tree visualization, for instance, points can be selected via a branch selection or by the area of a polygon.

In fact, the clustering process necessary for unlabeled PLS mapping in approach 2 can also be done manually in our system by 'coloring' the whole data set group by group as a way to create an initial labeling.

The whole process can iterate until a proper sample and a proper model be built. The model can then be used to map any number of similarly described data sets. In our experience so far, not many iterations are necessary since the PLS methods produce very reliable models. The next section shows our analysis of the PLS mapping approaches.

4. Results

From the description of our methodology in the previous section, it can be seen that there are two parameters for PLS dimension reduction, i.e., a user-defined number of factors and number of classes determined by the number of classes (for labeled training sets) or number of clusters (for unlabeled training sets). From these parameters and the training set, the multi-class method will reduce the data dimension to the number of factors and the one-against-all (o-a-a) method will reduce the dimension to the number of classes.

For instance, Figure 1 shows a possible visual output for the data set NEWS. Figure 1(a) shows a visualization of its

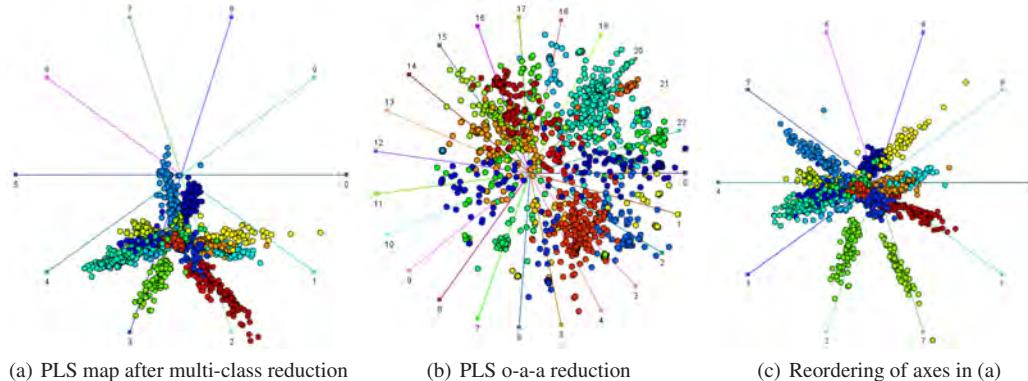


Figure 1: Examples of PLS dimension reduction of the NEWS 23 labeled data set with Radviz placement on latent dimensions. (a) is the multi-class reduction on 10 factors; (b) is o-a-a reduction on 23 classes, also 10 factors; (c) is another view of the space in (a).

reduction by the multi-class strategy, shown by Radviz with the anchors representing the factors. Figure 1(b) shows the placement on the number of labels given by the o-a-a strategy. For a data set with 23 labels, 23 axes will be produced. Both strategies perform a considerable dimension reduction from the original 3731 dimensions of this data set. This view shows which dimensions are more involved in determining segregation of at least some of the classes present in the data. As it happens with Radviz, changing the order of axes will change placement of points. Since it is very fast, the user can experiment with different positions to understand the latent dimensions. Figure 1(c) shows the same reduction as in Figure 1(a), with order change in the axes.

In our tests, the choice of number of factors was always 10 for textual data sets. For image data sets, the number of factors was 8 for o-a-a and 5 for multi-class methods. We employed a k -fold cross-validation using the training samples to choose the number of latent variables, considering $k = 5$.

4.1. Data Sets and Test Setup

Table 1 presents details of the data sets employed in the evaluation tests.

Table 1: Information on test data sets.

Data set	Content	Classes	Items	Attributes
NEWS	RSS Feeds	22	1771	3731
ALL	Scientific Papers	8	2814	12201
COREL	Photographs	10	1000	150
ETHZ	Photographs	28	2019	3963

The NEWS data set was formed from 1771 RSS news feeds from BBC, CNN, Reuters and Associated Press, collected from their site between June and July 2011. From

the text set, a feature space was created by removing stop words and employing stemming. The coordinate of any particular point was determined by the *term-frequency-inverse-document-frequency* count. The result is a data set with 3731 dimensions. The 23 labels of the data set were assigned manually based on the perceived main topic of the news feed. The labels are unbalanced in number of points and there is high similarity of content between points labeled differently.

The data set named ALL contains abstracts of scientific papers in 8 areas of knowledge, with considerable part of common content across labels. This data set was collected from various sources and preprocessed similarly to the NEWS data set.

The COREL image collection[†] is composed of 1000 photographs that represent 10 specific subjects. Each image is represented by a vector of 150 SIFT descriptors [LW03]. The ETHZ image collection represents a subset of the ETHZ dataset [ELS*08], which provides photographs of different people captured in uncontrolled conditions, with a range of appearances. This collection is composed of 2019 images, divided into 28 labels forming unbalanced groups. Each image is represented by a vector of 3963 visual descriptors, combining Gabor filters, Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and mean intensity.

In order to verify the precision and feasibility of PLS mappings, we devised case studies that cover three situations: use of training sets selected by the user, use of training sets produced by clustering and sampling a labeled data set, and applying approach 2 (defined in Section 3.3) for unlabeled data. We also compared our approach to other dimension reduction techniques and tested various visual mappings for the reduced spaces.

[†] UCI KDD Archive, <http://kdd.ics.uci.edu>

Besides the visual output, the *silhouette coefficient* [TSK05] was used to evaluate the produced results numerically, since the main target of our approach is discriminability. The silhouette coefficient is a measure of cohesion and separation between groups of instances. Given an instance p_i , its cohesion a_i is the average distance between p_i and all other instances belonging to the same group as p_i . Its separation b_i is the minimum distance between p_i and all the other distances belonging to the other groups. The silhouette of a particular space or projection is given by the average of silhouette coefficients of all its n instances. Its formulation is given in Equation 2. Although the silhouette may not be the most appropriate measure to reflect grouping in projections from reduced spaces due to its inadequacy measuring clusters that are not round in shape, it is used here for assessment of group separation in the reduced spaces and also on visualization planes after 2D mappings.

$$S = \frac{1}{n} \sum_{i=1}^n \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad (2)$$

The silhouette coefficient varies between -1 and 1, with 1 meaning that groups are perfectly separated from one another. The distance measures employed were cosine for original textual spaces, Euclidean for original image spaces and Euclidean for all reduced spaces. We report detailed results for NEWS and ETHZ and summary results for data sets ALL and COREL.

4.2. Textual and Image Mappings

Textual data sets tend to produce sparse feature spaces whereas image collections produce denser data spaces. We tested PLS under both situations. In the next section, training sets are generated and then applied to unlabeled data. In order to verify precision, the labels are then painted on the visualizations. Then, in Section 4.2.2, the results for unlabeled data sets are given.

4.2.1. User Input and Sampling

The generation of training sets was done in two different ways, by allowing the user to select the samples or by automatically sampling a labeled data set (see Section 3.3). In our captions and tables, training sets manually defined by the user are identified by the suffix 'user' and sampled training sets by k -means clustering are identified by the suffix ' k -means', with k replaced by the proper number of clusters.

Table 2 shows numerical results of applying PLS reduction to the NEWS data set. The first line shows the silhouette coefficient of the original data set. The two following lines display the silhouette of the visualizations by LSP and NJ-tree as references. The subsequent lines present the silhouettes of reduced spaces after model creation with the training sets. We also show the silhouette of their projection using Radviz.

It can be seen from Table 2 that separability grows as the sample size increases. From the smallest sample set, PLS reduced spaces are improved twofold from original spaces in terms of silhouette coefficient.

Table 2: Labeled reduction for NEWS data set.

Sampling Size	Sampling Method	Silhouette reduced	Projection Technique	Silhouette
—	—	—	original	0.1374
—	—	—	LSP	0.0934
—	—	—	NJ-tree	0.0949
611	user	0.4052	Radviz	0.0612
863	user	0.6815	Radviz	0.1755
1169	user	0.7780	Radviz	0.4354
600	23-means	0.6187	Radviz	0.1035
800	23-means	0.5132	Radviz	0.1909
800	23-means multi	0.2799	Radviz	0.0537

Table 3 shows the same analysis using the ETHZ collection. Here, one can also notice a better separability of the reduced spaces that is proportional to the sample size growth and even greater than that reported for the NEWS data set. In that table, we also notice that Radviz had more difficulty in reflecting on the layout the silhouette of the data set, which is probably due to a larger number of points and classes.

Table 3: Labeled reduction for ETHZ collection.

Sampling Size	Sampling Method	Silhouette reduced	Projection Technique	Silhouette
—	—	—	original	0.0912
—	—	—	LSP	-0.0390
—	—	—	NJ-tree	0.1023
200	user	0.3622	Radviz	-0.1317
600	user	0.5457	Radviz	-0.0433
1000	user	0.6277	Radviz	-0.0555
200	28-means	0.4024	Radviz	-0.0777
600	28-means	0.5326	Radviz	-0.0648
1000	28-means	0.5915	Radviz	-0.0525

Figure 2 shows the precision of mappings from 863 samples with user defined training set by employing various 2D mapping strategies. Neighborhood Hit averages the number of neighbors for each point that belongs to the same class as that point. It can be seen that the NJ-tree is the best one to reflect class neighborhoods from reduced data. ISOMAP and LSP also perform well. From the plots, it can be seen that Radviz, in terms of class segregation, did not perform as well. That is due to its trend to place together points with same balance in coordinates, rather than similar coordinates.

Figure 3 shows three projections of the reduced NEWS data set produced using 2D multidimensional mappings and also one of the original data set. Using ISOMAP, groups are more dense, while Radviz tends to spread the groups more, with the parameter circle adjusted to the maximum coordinates of the data set. The others show good discriminability. The NJ-tree (Figure 3(c)) manages to reflect PLS separation of almost every class in the reduced space as well as levels of similarity within each class. Contrasting that with the tree built directly from the original attributes (Figure 3(d)), it can be seen that class neighborhoods of many points were resolved by the new reduced space.

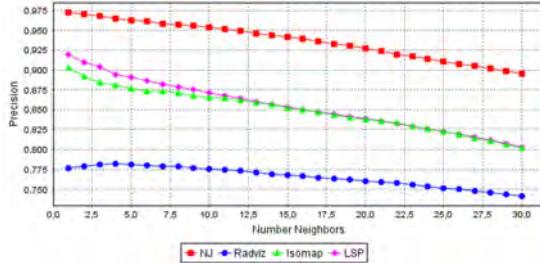


Figure 2: Neighborhood Hit for projections of the NEWS data set, employing a user selected training set with 863 samples.

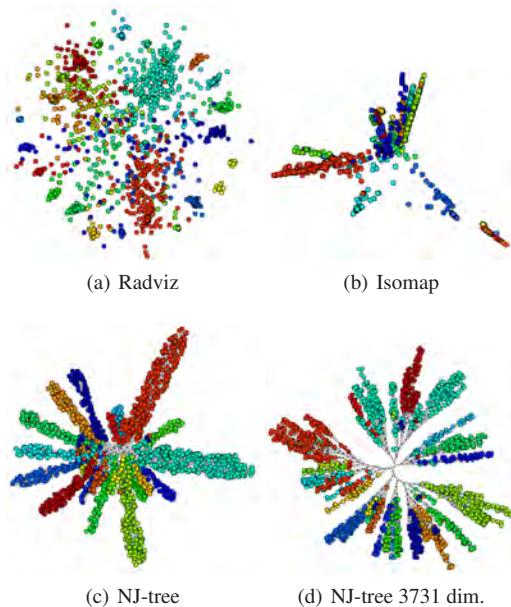


Figure 3: (a) to (c) visualizations of the NEWS data set reduced by PLS to 23 dimensions; (d) NJ-tree of the data set with original dimensions.

4.2.2. Unlabeled Datasets

Due to the nature of the PLS, training is necessary; therefore, labeling sets is a necessary task for the training set. To find out what could be done in case of such a training set is not available, we clustered the NEWS and ETHZ data sets and used the cluster label to produce a model using PLS. The reasoning behind it is that it is not necessary to know precise labels, but rather get proper samples that distinguish classes of points in order to obtain a PLS model with good segregation.

To cluster the unlabeled data set, we used two approaches. The first one clustered the data automatically by bisecting k -means (bkmeans) with 23 or 40 classes for the NEWS data set and with 20 or 28 classes for the ETHZ collection. In

each cluster, samples were chosen as the closest and farthest from the centroid of the cluster. The second cluster approach required a manual classification procedure, supported by our system, that is done by projecting the data using the NJ-tree, which produces the same tree as Figures 3(c) and 3(d), except for the colors. From that, the user can label groups of points in the same branches by clicking on the top node of the branch or by drawing a polygon around a group that he or she intends to assign a label. We did that resulting in 24 classes (slightly different from the original 23) for NEWS data set and 12 classes (different from the original 28) for ETHZ collection.

Table 4 shows the result of applying this method for dimension reduction based on PLS using originally unlabeled training sets. The table shows two silhouettes for each reduction. One of them (Silhouette Reduced) is for the reduced space using the cluster label, and the 'Cross Silhouette' column is the silhouette of the reduced space considering the original label for every point, instead of the cluster label. It can be seen that the results are very satisfactory, with silhouettes varying from 0.07 to 0.41 for a data set with original silhouette 0.1374, in the case of NEWS data set. In addition, the one-against-all approach is more precise than multi-class.

Table 4: Results of dimensionality reduction using unlabeled training sets.

Data	Samples	Sampling Method	Reduction Method	Silhouette Reduced	Cross Silhouette
NEWS	original	—	—	—	0.1374
NEWS	800	bkmeans-23	multi	0.1718	0.1669
NEWS	800	bkmeans-23	o-a-a	0.4260	0.2388
NEWS	800	bkmeans-40	multi	0.1440	0.0705
NEWS	800	bkmeans-40	o-a-a	0.4173	0.1549
NEWS	800	nj-24	o-a-a	0.2913	0.3100
NEWS	all	bkmeans-23	o-a-a	0.9258	0.3529
ETHZ	original	—	—	—	0.0912
ETHZ	1000	bkmeans-20	multi	0.1060	0.1294
ETHZ	1000	bkmeans-20	o-a-a	0.3401	0.0191
ETHZ	1000	bkmeans-28	multi	0.1188	0.1014
ETHZ	1000	bkmeans-28	o-a-a	0.3172	0.0648
ETHZ	1000	nj-12	o-a-a	0.5236	-0.0884

Figure 4 shows the Neighborhood Hit of various projections from the unlabeled version of NEWS data set. Values are quite high, with the tree also performing best, followed by LSP.

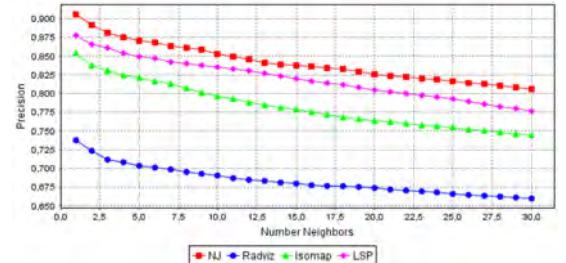


Figure 4: Neighborhood Hit for projections of the unlabeled reduced NEWS data set, clustered with all data points.

The data set named ALL was processed in the same way as the news data set. It is a very difficult data set to separate, since articles with very similar content have different labels and there are two classes that significantly dominate the data set while others are very small in numbers. The silhouette coefficient of 0.0350 for the original data gives a good idea of this type of distribution. For that data set, the silhouette coefficient of the reduction in the one-against-all method for 800 samples (less than a fourth of the data set) was 0.4926. For 1100 samples, it reached 0.5588. The NJ-tree for the reduced space has a silhouette of 0.2327, whereas for the original data it is 0.06.

The COREL image collection has originally a good separability of instances in their classes. The original silhouette for this data set is 0.1595. The dimensionality reduction using one-against-all method produces a reduced data set with silhouette coefficient of 0.5102, using 200 samples, improving considerably the original space. Using 500 samples, the silhouette coefficient is 0.5712. The values of silhouette for the ISOMAP projection are 0.1034 and 0.3780 respectively, before and after the reduction procedure. For NJ-trees, these values are 0.12 and 0.4002, also confirming the results obtained with ETHZ collection. Radviz cannot be applied on the original spaces due to space limitations for a large number of axes. For this example, the Radviz projection shows a silhouette value of 0.2759.

4.3. Times and Comparison with other Supervised Reduction Techniques

Tests with text files were run on a computer with Intel i5 processor with 2.3GHz and 6GB memory. For image examples, we have employed a notebook with Intel Core2 Duo processor, with 2.53GHz and 4GB RAM. We have implemented PCA plus three other dimension reduction techniques with supervision capabilities. Self-Organizing Maps would also satisfy the supervision requirement, but they proved to be extremely slow for reducing the large number of dimensions to a reduced space with more than three dimensions.

Table 5 shows computational times and silhouettes of reduced spaces obtained by PLS, as well as PCA, PivotMDS, ISOMAP and LLE. It can be seen that PLS performs similarly or better than PCA, depending on the strategy, and better than all the others. It is competitive with PCA in terms of time. The fastest ones do not perform as well.

Figure 5 shows the precision by Neighborhood Hit for six reduced spaces and for the original NEWS data set. The PLS reduced spaces employ the model created by sampling 800 labeled points shown in the previous section and a new one, with 510 samples chosen slightly more carefully, reduced both by multi-class and o-a-a. These user-defined 510 samples were very unbalanced within the 23 classes, with samples for each label varying from 7 to 57. The complete NEWS data set is also unbalanced, but in a different proportion to the samples. Results confirm the discriminability

Table 5: Model creation times and silhouettes, compared to other supervised dimension reduction methods. PLS models are o-a-a, the slowest.

Data Set	Reduction	Time	Silhouette		
			Reduced	LSP	NJ-Tree
NEWS	PCA-23	38 min	0.3163	0.0269	0.2189
NEWS	PLS 23-unlabeled	22 min	0.380	0.1210	0.27
NEWS	PLS-23-user-800	3 min	0.6815	0.1244	0.3456
NEWS	PLS-10-means-800	7 min	0.279	0.1363	0.2183
NEWS	PLS user-510	7 min	0.60	0.0665	0.3530
NEWS	LLE	2.5 min	-0.0195	-0.0127	-0.2491
NEWS	ISOMAP	11 sec	-0.2720	-0.3040	-0.2266
NEWS	PivotMDS	14 sec	-0.2062	-0.3340	0.1665
ETHZ	PCA-28	46 min	0.1039	-0.0674	0.0972
ETHZ	PLS-28-user-1000	12 min	0.6277	0.7928	0.5748
ETHZ	PLS-28-means-1000	18 min	0.6132	0.5107	0.5576
ETHZ	LLE	8 min	0.08131	-0.2085	0.0884
ETHZ	ISOMAP	32 sec	-0.0652	-0.2442	0.0
ETHZ	PivotMDS	48 sec	-0.1979	-0.3429	-0.1339

of PLS. PCA performs as well as PLS with the user 510 o-a-a set and sampling 800 set. All the other tested reduction techniques perform worse.

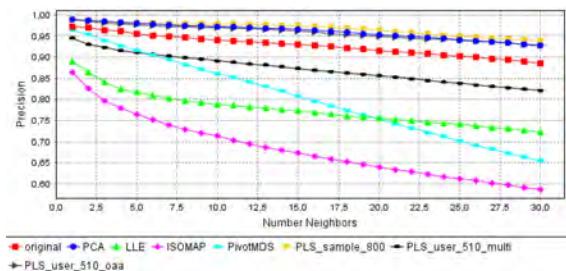


Figure 5: Neighborhood Hit for the original and reduced NEWS data spaces. There is no statistical significance for the difference in precision between the spaces reduced by PCA, PLS unlabeled, and PLS by user training with 510 samples. They are statistically better than the original neighborhood hit, which is, in its turn, better than PLS multi-class from user samples, followed by PivotMDS, LLE and, finally, ISOMAP.

PLS times shown in the tables were spent mostly in the model creation phase of the PLS dimension reduction. After the model is generated, it is used to map any size of data set bearing the same features. This mapping from a pre-built model is very fast. Loading and applying the model for ETHZ data sets took an average of 19s for one-against-all and 1.3s for multi-class. For the NEWS data set, it took 9.3s for one-against-all and 1.3s for multi-class. For the ALL data set, it took 4.5s for one-against-all and 0.6s for multi-class. The projections took different times with Radviz being very fast (2s at most) and the tree the slowest, but still taking few seconds.

In terms of visual quality of visualization, the silhouette is also a good measure of the visual improvement provided by reduced spaces. Figure 6 shows the NJ-tree layouts of the NEWS data set created from various of the reduced spaces mentioned above. An association between silhouette

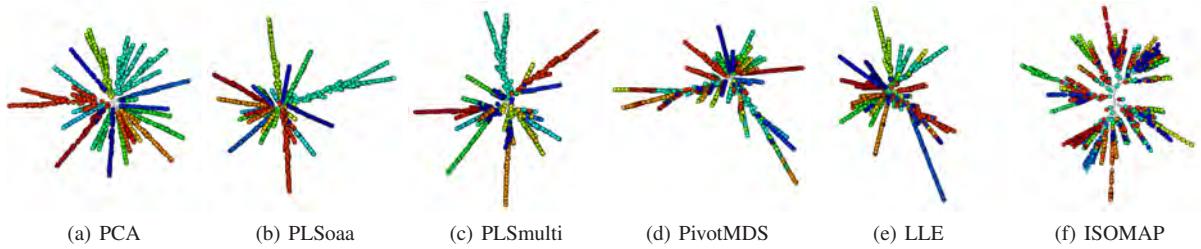


Figure 6: NJ trees of the NEWS data set generated from reduced spaces.

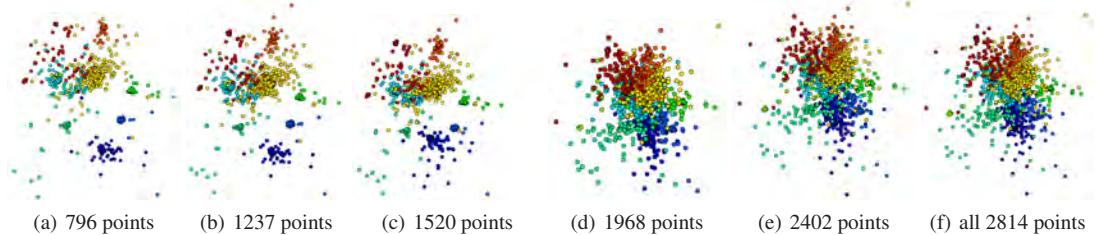


Figure 7: Progressive application of the ALL data to the model created by sampling 1200 points using the clustering strategy.

and quality of visualization can be observed, related with grouping, on the same branches of individuals with the same labels.

Finally, one of the important motivations of the method is precisely that it is able to use a model to map growing data sets. Figure 7 shows a series of mappings from a previously calculated model for different sizes of the ALL data set, using one-against-all strategy. It shows that, as the data set increases in size, the mappings of certain groups of points are kept in the same region. This allows the user to maintain the mental model of the PLS projection. A model can be used until it is no longer adequate to represent the changes in the data set, in which case the user can adjust the model again to comply with possible changes in class distributions.

A Java API for the PLS techniques as well as a system that implements the visual approach and all the data sets used in this section are available at <http://infoserver.lcad.icmc.usp.br>, following the link to the Tools section.

5. Conclusions and Future Work

Although PLS is not a new technique for discrimination analysis, its use in the context of visual analysis is novel and, as it has been shown in this work, PLS is a very powerful tool that supports various important tasks for visual data mining, such as dimension reduction, classification and manipulation of growing data sets.

The detailed analysis presented here shows the high precision of PLS dimension reduction both to create a model from

a labeled data set to be applied effectively for larger data and to generate effective models for unlabeled data sets. The approach based on user training sets aids the user to build and change his or her view of the data adapting the system to adjust according to acquired model. The extensive analysis presented is meant to offer evidences of the flexibility of PLS in various visual analysis contexts.

Within the visual mapping framework, we have shown that reduced spaces can be successfully visualized by using multidimensional projections or similarity trees, reflecting proper improvement of the data space provided by the dimension reduction.

The model building time of PLS runs in a fraction of the time compared to standard PCA and it results in similarly high precision models under the same circumstances as well as using only portions of the data set as training. Although there are newer supervised techniques than PCA, the precision problems presented by the technique are diminished when more latent dimensions are used, such as it is the case here. All tests favored PLS in terms of precision, followed by PCA. The others, although faster, presented worse precision.

As future work, we intend to employ incremental versions of PLS to provide fast changes of the model as needed when the ground truth changes.

Acknowledgements

The authors wish to acknowledge Brazilian financial agencies CNPq and FAPESP. We are also thankful to Frizzi San Roman for the preparation of the NEWS data set.

References

- [BBKY06] BRONSTEIN M., BRONSTEIN A., KIMMEL R., YAVNEH I.: Multigrid Multidimensional Scaling. *Numerical Linear Algebra with Applications* 13 (2006), 149–171. [2](#)
- [BGI*97] BROADHURST D., GOODACRE R., JONES A., ROWLAND J., KELL D.: Genetic Algorithms as a Method for Variable Selection in Multiple Linear Regression and Partial Least Squares Regression, with Applications to Pyrolysis Mass Spectrometry. *Analytica Chimica Acta* 348, 1–3 (1997), 71–86. [3](#)
- [BN03] BELKIN M., NIYOGI P.: Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Comput.* 15, 6 (2003), 1373–1396. [2](#)
- [BP07] BRANDES U., PICH C.: Eigensolver Methods for Progressive Multidimensional Scaling of Large Data. In *LNCS*, vol. 4372. Springer, 2007, pp. 42–53. [2](#)
- [BS07] BOULESTEIX A.-L., STRIMMER K.: Partial Least Squares: A Versatile Tool for the Analysis of High-Dimensional Genomic Data. *Briefings in Bioinformatics* 8, 1 (2007), 32–44. [2, 3](#)
- [dST04] DE SILVA V., TENENBAUM J.: *Sparse Multidimensional Scaling using Landmark Points*. Tech. rep., Department of Mathematics, Stanford University, CA, USA, 2004. [2](#)
- [Eld04] ELDEN L.: Partial Least-Squares vs. Lanczos Bidiagonalization—I: Analysis of a Projection Method for Multiple Regression. *Computational Statistics & Data Analysis* 46, 1 (2004), 11–31. [2](#)
- [ELS*08] ESS A., LEIBE B., SCHINDLER K., , VAN GOOL L.: A Mobile Vision System for Robust Multi-Person Tracking. In *IEEE CVPR* (Anchorage, AK, USA, June 2008), pp. 1–8. [5](#)
- [Gar94] GARTHWAITE P.: An Interpretation of Partial Least Squares. *Journal of the American Statistical Association* 89, 425 (1994), 122–127. [3](#)
- [Gel88] GELADI P.: Notes on the History and Nature of Partial Least Squares (PLS) Modelling. *Journal of Chemometrics* 2, 4 (1988), 231–246. [3](#)
- [HGP99] HOFFMAN P., GRINSTEIN G., PINKNEY D.: Dimensional Anchors: A Graphic Primitive for Multidimensional Multivariate Information Visualizations. In *Workshop on New Paradigms in Inform. Vis. and Manip. in Conjunction with ACM CIKM* (Kansas City, MO, USA, 1999), pp. 9–16. [2, 4](#)
- [JCC*11] JOIA P., COIMBRA D., CUMINATO J., PAULOVICH F., NONATO L.: Local Affine Multidimensional Projection. *IEEE TVCG* 17, 12 (2011), 2563–2571. [2](#)
- [Jol02] JOLLIFFE I.: *Principal Component Analysis*. Springer, 2002. [2](#)
- [KCH02] KOREN Y., CARMEL L., HAREL D.: ACE: A Fast Multiscale Eigenvectors Computation for Drawing Huge Graphs. In *IEEE Symp. on Inform. Visualiz.* (2002), pp. 137–144. [2](#)
- [KHD11] KEMBHAJI A., HARWOOD D., DAVIS L.: Vehicle Detection Using Partial Least Squares. *IEEE TPAMI* 33, 6 (2011), 1250–1265. [3](#)
- [Kru64] KRUSKAL J.: Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis. *Psychometrika* 29 (1964), 115–129. [2](#)
- [LGB*95] LINDGREN F., GELADI P., BERGLUND A., SJOSTROM M., WOLD S.: Interactive Variable Selection (IVS) for PLS. Part II: Chemical Applications. *Journal of Chemometrics* 9, 5 (1995), 331–342. [3](#)
- [LN06] LEE S.-J., NOBLE A.: Use of Partial Least Squares Regression and Multidimensional Scaling on Aroma Models of California Chardonnay Wines. *American J. Enology and Viticulture* 57, 3 (Sept. 2006), 363–370. [2](#)
- [LW03] LI J., WANG J. Z.: Automatic Linguistic Indexing of Pictures by a Statistical Modelin Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003), 1075–1088. [5](#)
- [NOM*02] NESTOR P., O'DONNELL B., MCCARLEY R., NIZNIKIEWICZ M., BARNARD J., SHEN Z., BOOKSTEIN F., SHENTON M.: A New Statistical Method for Testing Hypotheses of Neuropsychological/MRI Relationships in Schizophrenia: Partial Least Squares Analysis. *Schizophrenia Research* 53, 1–2 (2002), 57–66. [3](#)
- [NR02] NGUYEN D., ROCKE D.: Tumor Classification by Partial Least Squares Using Microarray Gene Expression Data. *Bioinformatics* 18, 1 (2002), 39–50. [3](#)
- [PdRDK99] PEKALSKA E., DE RIDDER D., DUIN R., KRAAIJVELD M.: A New Method of Generalizing Sammon Mapping with Application to Algorithm Speed-up. In *Annual Conf. Advanced School for Comput. Imag.* (1999), pp. 221–228. [2](#)
- [PEP*11] PAULOVICH F., ELER D., POCO J., BOTHA C., MINGHIM R., NONATO L.: Piecewise Laplacian-based Projection for Interactive Data Exploration and Organization. *IEEE CGF, Proc. Eurovis 2011* 30, 3 (2011), 1091–1100. [2](#)
- [PFCP*11] PAIVA J., FLORIAN-CRUZ L., PEDRINI H., TELLES G., MINGHIM R.: Improved Similarity Trees and their Application to Visual Data Classification. *IEEE TVCG* 17, 12 (Dec. 2011), 2459–2468. [2](#)
- [PNML08] PAULOVICH F., NONATO L., MINGHIM R., LEVKOWITZ H.: Least Square Projection: A Fast High-Precision Multidimensional Projection Technique and Its Application to Document Mapping. *IEEE TVCG* 14, 3 (2008), 564–575. [2](#)
- [RS00] ROWEIS S., SAUL L.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290, 5500 (Dec. 2000), 2323–2326. [2](#)
- [SGCD12] SCHWARTZ W. R., GUO H., CHOI J., DAVIS L. S.: Face Identification Using Large Feature Sets. *IEEE Transactions on Image Processing* 21, 4 (2012), 2245–2255. [3](#)
- [SKHD09] SCHWARTZ W., KEMBHAJI A., HARWOOD D., DAVIS L.: Human Detection Using Partial Least Squares Analysis. In *IEEE ICCV* (2009), pp. 24–31. [3](#)
- [TdSL00] TENENBAUM J., DE SILVA V., LANGFORD J.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290, 5500 (Dec. 2000), 2319–2323. [2](#)
- [TLTKT09] TALBOT J., LEE B., KAPOOR A., TAN D.: EnsembleMatrix: Interactive Visualization to Support Machine Learning with Multiple Classifiers. In *ACM CHI* (2009), pp. 1283–1292. [2](#)
- [Tor65] TORGESON W.: Multidimensional Scaling of Similarity. *Psychometrika* 30 (1965), 379–393. [2](#)
- [TSK05] TAN P.-N., STEINBACH M., KUMAR V.: *Introduction to Data Mining*. Addison-Wesley Longman, Boston, MA, USA, 2005. [6](#)
- [Wol85] WOLD H.: Partial Least Squares. In *Encyclopedia of Statistical Sciences*, vol. 6. Wiley, New York, NY, USA, 1985, pp. 581–591. [1, 2, 3](#)
- [WPW03] WANG J., PENG W., WARD M.: Interactive Hierarchical Dimension Ordering, Spacing and Filtering for Exploration of High Dimensional Datasets. In *IEEE Symp. on Inform. Visualiz.* (Seattle, WA, USA, Oct. 2003), pp. 105–112. [2](#)
- [YWRH03] YANG J., WARD M., RUNDENSTEINER E., HUANG S.: Visual Hierarchical Dimension Reduction for Exploration of High Dimensional Datasets . In *Joint Eurographics - IEEE TVCG Symp. on Visualization* (Grenoble, France, 2003), pp. 19–28. [2](#)

Artigo: Incremental Visual Data Classification using Locally Weighted Projection Regression

Este apêndice apresenta o conteúdo completo de um artigo submetido ao **15th IEEE-VGTC Eurographics Conference on Visualization** (EuroVis 2013). Uma visão geral desse trabalho foi apresentada no Capítulo 5 desta tese.

Incremental Visual Data Classification using Locally Weighted Projection Regression

Submission ID: 166

Abstract

Data classification is a computationally intensive task, presents variable precision and is considerably sensitive to the classifier configuration and to data representation. Setting up a classification system requires various iterations, and more so when the data set evolves. Many of these problems can be relieved by methods that support user's interference in the classification process. In this paper, we propose a visual image classification methodology that is meant to visually support users in classification tasks such as training set definition; model creation, application and verification; and classifier parameter adjustments. The method allows for tuning a classifier when the data set evolves without rebuilding the initial model. This is done by employing the Locally Weighted Projection Regression (LWPR) model, an incremental regression algorithm that supports fast adjustment of its classification model when new instances are fed as training samples. Visualization in the system is accomplished by means of point placement strategies and we exemplify the method employing Neighbor Joining trees. The same methodology can be employed for a user to create his or her own ground truth (or perspective) in a data set. We illustrate the applicability of the approach showing scenarios of categorization tasks for image and text data sets, demonstrating its benefit for the creation of classification models, application and adjustment as well as for deeper understanding of success and failures in classification results.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms

1. Introduction

Data classification in general has an individual nature in the sense that no technique produces good results in all scenarios, but they need to be adapted to the data at hand. These results strongly depend on several factors, such as the quality of the feature space and the similarity measure employed. Adequacy of the training set is also crucial [FM06]. Thus, it is important to build a representative feature space and a training set representative of all classes.

Users may play an important role on the construction of these training sets, because their knowledge regarding the problem allows the selection of an adequate set of instances. To accomplish that, the collections of candidate samples should be friendly displayed, along with a set of effective interaction tools.

Visualization techniques provide tools to achieve proper layout for these collections, highlighting the structure, organization and specificities of each class consequently guiding the user in his or her choice. The same setup can support adjustments of the classification processes by allowing the user

to recognize reasons for failure and success of the classifier in specific cases.

The Locally Weighted Projection Regression (LWPR) [VS00, VDS05] algorithm is widely used for function approximation in high dimensional spaces, even in the presence of redundant and irrelevant input dimensions, with applications in several tasks involving transaction concurrency control and motion prediction. The main feature of this algorithm is its incremental learning, providing means to update the model in order to accommodate eventual changes in the concept of the classes, or in the structure of existing classes. This property is valuable when inserting the user in the classification process, by allowing the online adjustment of the produced models to improve the results and to adapt to new realities in an evolving data set. In addition, it avoids the need for rebuilding the existing models from scratch every time new samples are added, allowing, this way, a real time interaction of the user.

Point-based visualizations can be successfully used to give support to classification. Among them, similarity trees [CPMT07] have been shown to lead to adequate dis-

plays for classification related tasks [PFP*11]. They impose a hierarchy that reveals structure in the similarity relationship. A radial tree layout [BBS05] applied to the generated structure provides reduction of clutter as compared to other point placements and reproduces, within a branch, similar properties to the global tree layout. These properties promote direct inspection of the collections via similarity, which in turn support adequate selection and evaluation of representatives and their similar counterparts.

We propose in this paper a Similarity-based Visual Classification Methodology (SVC), that integrates LWPR to visualization techniques via similarity trees and other point placement techniques. Our hypothesis is that the user insertion in the classification in this manner offers an iterative process that allows a fast convergence of classification results.

The incremental training capabilities of LWPR allows the user to update the models and consequently map dynamic collections as well as changes in perspectives of an existing collection. Additionally, we test the approach to stronger structural changes in the collections, such as when new classes appear. A visual classification system that supports SVC is made available, and results of several scenarios related to creation and application of LWPR models are discussed.

The following sections describe the background in 2D mapping techniques from multidimensional spaces, the basic concepts of LWPR, our approach to visual classification using LWPR, and the results of several classification scenarios for text and image data.

2. Visual Data Classification

The classification of a data collection may provide a simple and effective way to explore its information content. Several authors [KPS05, RÖ6, HR04] advocate the importance of user's insertion on the image retrieval and classification processes, combining the flexibility, creativity and knowledge of the domain expert with the actual computational power [Kei02], thus improving the confidence and comprehensibility of the created model. This insertion is necessary because many types of information, such as images, reside in a continuous representation space [ZH03], in which semantic concepts are best described in discriminative subspaces. Thus, only a small subset of this space is not enough for describing all concepts. Moreover, different users at different times may have distinct interpretations or intended usages for the same image, which makes offline, user-independent learning undesirable in general.

Visual data mining systems are based on three types of approaches [Ank01]: (1) application of visualization techniques independent of data mining algorithms, (2) use of visualization techniques for data mining results, and (3) integration of visualization and data mining algorithms for visu-

alizing intermediate steps of a data mining algorithm. Very few systems truly support the features suggested for the third category [ZGG09].

Coordinated and Multiple Views (CMV) [Rob07] may also be valuable to visual classification because it allows to explore the advantages of several visualization techniques simultaneously, and reveals relationships amongst instances that could be hidden. VDM-RS [ZGG09], for instance, provides a set of views for results of remotely sensed image classification, combining traditional interfaces, such as spatial maps and error matrices, with visualization of decision trees. The latter view presents all the steps followed by the classifier to categorize a specific instance, allowing the tracking and exploration of the classification process. The construction of decision tree models can also be supported by the coordination of views with Scatter Plot Matrices and Parallel Coordinates [Do07], helping users to understand the classification performed by each constructed tree. Another approach to visual data mining [MN06] combines a generative topographic mapping (GTM) projection, arranged in a hierarchical fashion with a modified version of Parallel Coordinates to achieve a better understanding of the data space, directly involving the user in the data mining process by visualizing intermediate steps of machine learning algorithms. These systems, however, do not allow directly model updates using the results of the classification, and the user is limited to comprehend the classifier decisions and understand the entire or part of the process.

Another strategy for visual data classification is called *Active Learning* [JPP12], in which the classifier is interactively trained with user annotations on informative samples. The idea is that a classifier trained on a small set of well-chosen examples can perform as well as a classifier trained on a larger number of randomly chosen examples, requiring much less computational effort [TVC*11]. The system shows to the user a set of instances from which the classification result is most uncertain, and from an accurate labeling performed by the user, these instances will be used to reinforce the model knowledge and maximize its generalization capabilities. Another system uses predefined heuristics [TRP*09] to rank the instances and choose those considered more representative to the classifier adjustment. These instances will be manually labeled by the user and will iteratively feed the model.

Visual classification systems that employ Active Learning approach may incorporate interactive tools that allow a higher insertion in the process. However, most of these systems limits the user to answer questions about the relationship amongst selected instances [JPP12], or select, from a list, relevant or not relevant images to the classification. We believe that visualization techniques that highlight the relationship amongst instances, such as multidimensional projections and similarity trees can be combined to automatic classification procedures, providing a layout for the classifi-

cation results that better explain the classifier behavior. Additionally, it is important to provide a set of interactive tools for the user to adjust the model directly to their needs, in an iterative process with fast convergence.

3. Visual Incremental Classification

This section describes each component of SVCM in details.

3.1. Locally Weighted Projection Regression

The *Locally Weighted Projection Regression* (LWPR) [VS00, VDS05] algorithm achieves a nonlinear function approximation in high dimensional spaces, even in the presence of redundant and irrelevant input dimensions. It uses a set of locally linear models spanned by a small number of univariate regressions in specific directions in the original input space. An online weighted version [VS00] of *Partial Least Squares* (PLS) algorithm [Wol85] is used to perform dimensionality reduction on the specific directions.

LWPR is widely employed in prediction tasks in several areas, such as Medicine [FBM11], Computer Aided Design [ADD11], Robotics [DVS01] and Civil Engineering [AGD10]. This section briefly presents a mathematical description of the technique.

An LWPR regression model is constructed through a set of training instances represented as vectors \mathbf{x}_i and correspondent responses y_i , iteratively presented as input-output tuples (\mathbf{x}_i, y_i) . The LWPR prediction will be the weighted sum of the prediction of each of k locally linear model, according to Equation 1.

$$\hat{y} = \frac{\sum_1^k w_k \hat{y}_k}{\sum_1^k w_k} \quad (1)$$

The weights w_k of each locally linear model define the validity area of these models, also called *Reception Fields*, and are usually modeled by a Gaussian kernel, according to Equation 2. In this equation, \mathbf{D}_k represents a distance metric and \mathbf{c}_k represents the center of the kernel. The distance metric determines the size and shape of the validity area for each model. Thus, a weight value $w_{i,k}$ is computed for each input instance i . This weight varies according to the distance to the center of the kernel of each model k , which makes the learning process localized and independent. If the activation value of all the existing models is lower than a threshold, a new model is created, granting that the number of models is dynamically updated during the process.

$$w_{i,k} = \exp(-0.5(\mathbf{x}_i - \mathbf{c}_k)^T \mathbf{D}_k (\mathbf{x}_i - \mathbf{c}_k)) \quad (2)$$

The online weighted version of PLS is employed in the learning of the locally linear models, so that, for each model, the dimensions of the input instance \mathbf{x}_i are sequentially regressed along selected projections u_r , chosen by the technique in input space, yielding a set of r latent variables.

These directions are chosen according to the correlation of the input data with the output data (class information). The regression on a locally model will be composed of the linear combination of the latent variables for this model. More details regarding the online PLS algorithm can be consulted in [VS00].

The distance metric \mathbf{D}_k of each receptive field is individually updated, using an incremental gradient descent based on stochastic leave-one-out cross-validation criterion [VDS05]. Algorithm 1 illustrates the incremental learning process of an LWPR model. In the algorithm, w_{gen} represents a threshold for creation of new local models, and r_k is the number of latent variables for each of the k local models.

Algorithm 1 *Locally Weighted Projection Regression.*
Adapted from [VDS05]

```

1: Initialize LWPR model with no local models ( $LM = 0$ )
2: for all training instance  $(\mathbf{x}_i, y_i)$  do
3:   for  $k = 1 : LM$  do
4:     Calculate activation value (Equation 2)
5:     Update local PLS model and  $\mathbf{D}_k$ 
6:   end for
7:   if no linear model was activated by more than  $w_{gen}$ 
    then
8:     Create a new local model with  $r_k = 2$ ,  $c_k = X$ ,  $\mathbf{D}_k$ 
      = default value
9:   end if
10:  end for
```

LWPR presents interesting properties: the local model parameters can be estimated using statistical information calculated from the training instances, and the adaptation of these models and the distance metric can be done using stochastic cross-validation, thus eliminating the need to store the used training instances. The dynamic update of these models allows the input space to be covered by wide receptive fields in regions of low curvature and narrow receptive fields where the curvature is high [KVS08]. Additionally, the method has linear complexity on the number of input instances, and can handle a large number of redundant input dimensions. Finally, the learning is incremental, accommodating evolutions of the data sets.

3.2. LWPR Approaches

According to the LWPR formulation presented in Section 3.1, a regression model is created from a set of training instances divided into C classes, that is then used to classify a data collection. It is possible to update this model by adding new instance information, in order to reflect changes in the classification scenario. Two classification approaches can be adopted in the underlying PLS model: *One Against All* and *MulticlassMatrix* [PSPM12].

In the *One-Against-All* approach, C binary regression

models, one for each class, are constructed. When constructing a model for a class c , all training instances that belong to this class are labeled +1, and the remaining ones are labeled -1. In the classification, a test instance is presented to each model, resulting C responses (one per class). The best matching class is associated to the model presenting the highest regression response.

In the *MulticlassMatrix* approach, one regression model is constructed, with C responses, one for each class. When constructing this model, all training instances are labeled using a matrix $Y_{n \times C}$, with $Y_{i,j} = 1$ if the i -th instance belongs to the j -th class, and 0 otherwise. In the classification, a test instance is presented to the model and the class with best matching is assigned to this instance.

Our experimental results have shown no significant differences between the two approaches, considering computational cost and precision of the generated models. However, the One-Against-All approach requires the creation of one model for each class, and thus being more costly. This is also observed when model is updated. In both approaches, the models can be updated by adding new instances, to accommodate evolutions in the classification scenario. The appearance of new classes can also be treated, but that requires the original model to be rebuilt.

3.2.1. Creating and Applying the Model

The instances that compose the training for the LWPR model are informed by the user. They can be selected using the visualization layout, whose structure and point organization is able to guide the user towards a representative selection. Figures 1a and 1b show two views of an NJ tree for a set of 300 images, from which the user selected a 44 image training set. In this case, the images from the end of branches were chosen together with images from the top of the branches. The confidence of the classification is higher in instances located at the end of the branches, and lower in instances located at the top of the branches. It is possible to assign labels to the selected instances with no labels or to change labels for pre-labeled instances.

The created LWPR model can be employed in the classification of any collection bearing the same feature space. Figure 2a shows an NJ tree with the ground truth of another image collection with 700 images. For example, Figure 2b shows the visual classification result for this collection, using the LWPR model created from the samples in Figure 1. Numerical results are shown in Table 1.

Table 1: Numerical results of COREL-700 classification.

Matching Instances	615 (87.9%)
Non-matching Instances	85 (12.1%)
Accuracy	97.82%
Precision	88.49%
Recall	87.86%

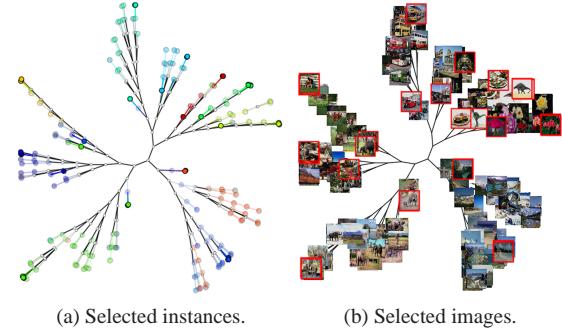


Figure 1: NJ tree for a 300 image collection, with 44 selected instances, represented by circles (1a) and by images (1b).

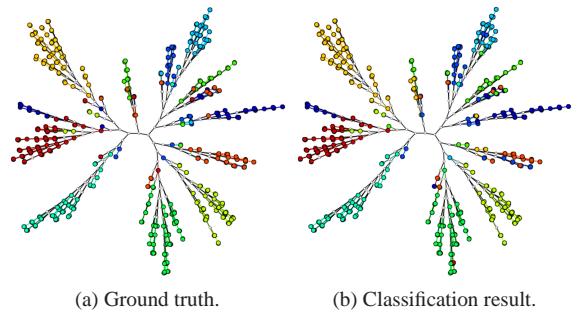


Figure 2: Classification result for a collection with 700 images.

3.2.2. Updating the Model

The LWPR model update can also be performed by selecting additional instances from a visualization layout. The model learning is performed incrementally, thus the selected instances will update each PLS linear model locally, generating new ones if necessary, according to the algorithm formulation presented in Section 3.1. The updated model will accumulate information from instances used both in creation and update procedure, so no actual instances need to be stored.

Several model update strategies can be adopted. In this work, we use a set of misclassified instances in order to add information about the classes for which the model is deficient. The similarity layout may serve this purpose also, helping users identify the reasons for failure by looking at the images deemed similar to the misclassified ones.

3.2.3. Rebuilding the Model with New Classes

In some classification scenarios, it is possible that some drastic changes in class distribution occur, and new classes appear. In this case, more than one LWPR model have to be

rebuilt. To avoid re-entering previous training instances, we opt to store the training sets from previous model updates. This information is then used in an update process involving instances of n new classes.

Using the One-Against-All approach, instances of the current update that belong to the n new classes are employed in the update of the existing models. Then, n new models are created, and the previously stored instances, together with the new ones, are used to train them, resulting in $C + n$ models after the procedure. The model created through the MulticlassMatrix approach will require the addition of n new responses. Thus, a new model is created with $C + n$ responses, and the previously stored instances, together with the new ones, are used to train it. In both approaches, the instances used in the current update are also stored, together with the previous ones, so they can be used in future updates that involve new classes.

4. Experimental Results

This section presents the results of several case studies representing classification scenarios developed with SVCM. The goal is to offer evidence of a visual framework to provide the insertion of the user in the classification process of a possibly evolving data set.

4.1. Data Sets and Test Setup

Table 2 presents details of the data sets employed in the evaluation tests.

Table 2: Information on test data sets.

Collection	Original Collection	Instances	Classes	Original Space	Reduced Space
ALL-Reduced	ALL	2814	9	5163	63
ETHZ-Reduced	ETHZ	2019	28	3963	48

The ALL data contains abstracts of scientific papers in 9 areas of knowledge, collected from various sources, with considerable part of common content across labels. From the text set, a feature space was created by removing stop-words and employing stemming [Por80]. The coordinate of any particular point was determined by the *term-frequency-inverse-document-frequency* count [SXY09]. The 9 labels of the data set were assigned manually based on the perceived main topic of the scientific paper. The number of papers across labels is unbalanced.

The ETHZ image collection represents a subset of the ETHZ dataset [ELS*08, SD09], which provides photographs of different people captured in uncontrolled conditions, with a range of appearances. This collection is composed of 2019 images, divided into 28 labels forming unbalanced groups. Each image is represented by a vector of 3963 visual descriptors, combining Gabor filters, Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and mean

intensity, the same setup used as used in [SGCD12] for face recognition.

The experiments were executed on an Intel Core2 Duo processor with 2.53GHz and 4GB RAM, using an LWPR library [KVS08] wrapped in a JAVA package, that contains all the methods for model creation and update, as well as the classification methods. The dimensionality of both collections were reduced by a PLS dimensionality reduction procedure [PSPM12], using the *MulticlassMatrix* approach and training sets containing 647 instances for ALL collection, and 400 instances for ETHZ collection. This dimensionality reduction was performed with the aim at highlighting the separability amongst classes on these collections, making the selection of representative instances by the user easier. Moreover, the employed LWPR library presents a limitation related to the number of dimensions of the input data.

To evaluate the classification process, we measured the number and percentage of *matching instances*, representing the instances correctly classified, and *non-matching instances*, representing misclassified instances. Additionally, we used the following measures, for each class: *accuracy* measures the proportion of correctly classified instances, amongst all instances; *precision* measures the proportion of correctly classified instances, amongst all instances categorized in the same class; and *recall* measures the proportion of correctly classified instances, amongst all instances that really belong to this class. We employed the average of these values to evaluate the classification process.

4.2. Instance Selection

This experiment aims at evaluating how the instance selection can impact the LWPR model for classification. On an NJ tree layout, the instances positioned far from the core of the tree (external instances), situated on more external leaves are the individuals that better characterize the class they belong. On the other hand, the instances positioned closer to the core of the tree (internal instances) represent the ones whose features do not fit well in any class, or that fit in more than one class. This is given by the nature of the tree.

Three training sets are used, the first composed of external instances, the second composed of internal instances, and the third composed of a combination of the two previous ones. In all situations, training and test sets are disjoint.

Table 3 presents the sets used in this experiment obtained from ETHZ-Reduced and ALL-Reduced collections, whereas Table 4 shows the results of each classification for these collections. The worst results were obtained using the external instances to compose the training set. These are likely to be close to the centroids of their group, and unable to represent boundary elements of the class, resulting in a restrictive classifier. Using internal instances, information on the boundaries of the groups is fed to the classifier, promoting inclusion of a larger variety of features.

Table 3: Training and test set for ETHZ-Reduced and ALL-Reduced collections used in the first experiment.

Data set	Training Set	Test Set
ALL-Reduced	45	2769
ETHZ-Reduced	112	1907

Table 4: Results of classification using three types of training set.

	External Instances	Internal Instances	Combined Instances
	ETHZ-Reduced		
Matching Instances	1478 (77.5%)	1592 (83.5%)	1713 (89.8%)
Non-matching Instances	429 (22.5%)	315 (16.5%)	194 (10.2%)
Accuracy	97.12%	98.41%	98.73%
Precision	83.41%	88.59%	92.62%
Recall	77.5%	83.48%	89.83%
ALL-Reduced			
Matching Instances	1410 (50.9%)	1623 (58.6%)	1609 (58.1%)
Non-matching Instances	1359 (49.1%)	1146 (41.4%)	1160 (41.9%)
Accuracy	80.53%	85.04%	84.23%
Precision	60.87%	63.44%	60.99%
Recall	50.92%	58.61%	58.11%

4.3. Constructing and Updating the LWPR Model

This experiment evaluates the use of visualization for the construction and update of an LWPR model to improve the classification results. After the model is constructed, as explained in Section 4.2, it is then employed to classify the remaining instances of the collection.

Table 5 shows the number of instances used for training and test set for ETHZ-Reduced and ALL-Reduced collections.

Table 5: Training and test set for ETHZ-Reduced and ALL-Reduced collections used in the second experiment.

Data set	Training Set	Instances for Class	Test Set
ALL-Reduced	45	5	2769
ETHZ-Reduced	84	3	1935

The numerical result of the classification for ETHZ-Reduced is presented in Table 6, in the second column. By the confusion matrix of this classification (not showed here due to its large size), as well as by the comparison of the layouts presented in Figures 3a and 3b, one can notice that 2 classes concentrated the higher error rates, 6 (109 misclassified instances) and 25 (65 misclassified instances). Thus, 27 instances of these two classes were carefully selected to update the LWPR model. The third column of Table 6 shows the results of the application of the updated model to a second classification.

A Class Matching [PFP*11] tool was used to visually evaluate mismatch between ground truth and results. It highlights in red the misclassified instances. The choice of the new points to be used to update the model in this experiment was done with the use of this tree. This updated model is again used to classify the collection. It is important to notice that, in the absence of a ground truth, the user decides which points are misclassified, and this misclassification is announced by points in close branches classified differently.

Table 6: Results of ETHZ-Reduced and ALL-Reduced classification using the initial and updated LWPR models.

	Initial Model	Updated Model
	ETHZ-Reduced	
Matching Instances	1704 (88.1%)	1779 (91.9%)
Non-matching Instances	231 (11.9%)	156 (8.1%)
Accuracy	98.48%	98.96%
Precision	89.09%	92.99%
Recall	88.06%	91.94%
ALL-Reduced		
Matching Instances	1875 (67.7%)	1991 (71.9%)
Non-matching Instances	894 (32.3%)	778 (28.1%)
Accuracy	86.61%	88.45%
Precision	71.98%	73.79%
Recall	67.71%	71.90%

The mismatches of the classification can be seen in Figure 3c, in red. The results are improved for both data sets.

To verify the role of the visualization in selecting instances for LWPR model update, we performed another experiment in which we reproduced the same steps performed above, but with the updating of the training set with instances selected randomly. We employed 10 random sets, and the average classification results obtained with the updated models are shown in Table 7.

Table 7: ETHZ-Reduced and ALL-Reduced average LWPR classification rates with updating instances selected randomly.

	ETHZ Reduced	ALL Reduced
Matching Instances	1802 (93.1%)	1812 (65.4%)
Non-matching Instances	132 (6.8%)	956 (34.5%)
Accuracy	99.06%	84.9%
Precision	94.22%	74.19%
Recall	93.17%	65.45%

For ETHZ-Reduced, the improvement in results of the random samples against the 'informed' samples are not surprising, since this collection presents an easier separability than the text data set. Since there are more selections likely to represent well the classes improvement of the model is higher. For ALL-Reduced, the improvement is better for the selection guided by the visualization. This collection presents worse class separability and the criteria used for instance selection may considerably influence the model learning for each class. In this case, the layout played a crucial role to produce a satisfactory selection. Considering a real situation, in which there is no ground truth for a collection, naturally the user is the best resource to find a proper set for model update, and the similarity based visual layout is a potentially valuable tool to perform the task.

Table 8 shows the computational time required to construct and update the LWPR models, using the *One Against All* (O-A-A) and *MulticlassMatrix* (Multiclass) approaches. It is possible to observe that both approaches perform considerably fast, such that *MulticlassMatrix* performs faster than *One Against All*. These results show that SVCM can provide an interactive and online adjustment procedure that allows for the user to adapt the models to new realities in evolving collections.

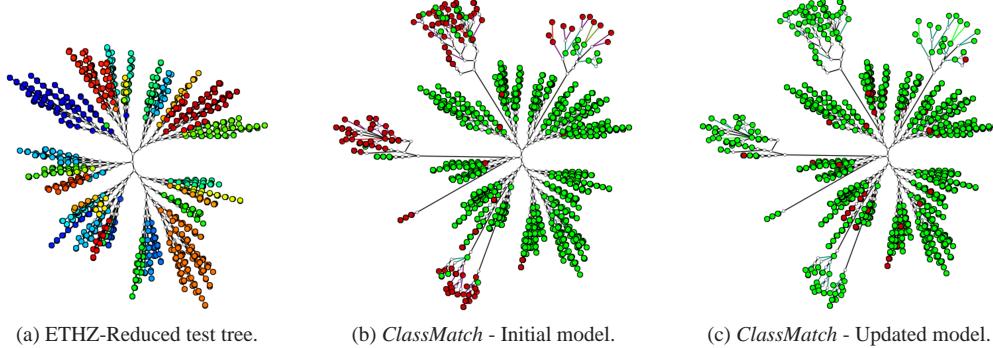


Figure 3: Visual comparison between the classification results using initial and updated LWPR models.

Table 8: Computational time (seconds) to create and update the LWPR models.

Data set	Model Creation		Model Update	
	O-A-A	Multiclass	O-A-A	Multiclass
ALL-Reduced	0.63	0.32	0.27	0.16
ETHZ-Reduced	3.54	1.26	1.18	0.40

4.4. Iterative Classification

This experiment aims at verifying the convergence of a classification procedure using the application of a sequence of LWPR model updates. Initially, an LWPR model is constructed from a training set and used to classify a collection. Based on the result of this classification, the model is updated by the user and employed to classify a second collection. The model is updated again, using the results of the second classification, and another classification is performed on a third collection. This experiment was performed on ALL-Reduced collection and it is described as follows.

Iteration 1: From ALL-Reduced, three disjoint sets were built, described in Table 9.

Table 9: Sets of instances built from ALL-Reduced collection used in Iteration 1.

Set	Instances
Training	45
ALL-Reduced01	926
ALL-Reduced02	922

The training set was employed to create an LWPR model that, in turn, was used to classify ALL-Reduced01 set. The numerical results of this classification is shown in the second column of Table 10 and the corresponding NJ tree and *Class Matching* tree presented in Figure 4. It is possible to notice that classes 2, 4, 8, and 0 presented high misclassification rates: 67.6%, 56.3%, 50.77% and 47.25%, respectively.

Iteration 2: 8 instances from classes 2, 4, 8, and 0 (2 from each classes) were selected to update the LWPR model, which in turn was used to classify ALL-Reduced02 set. The

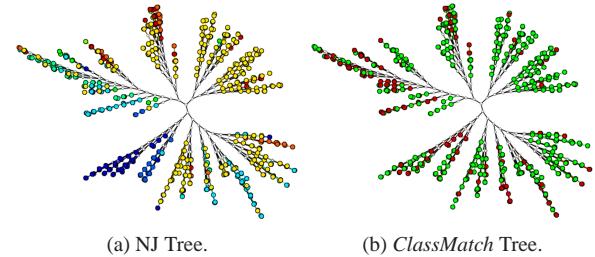


Figure 4: NJ Tree of ALL-Reduced01 collection and corresponding classification result using the LWPR model created in the first iteration.

numerical results of this new classification are shown in Table 10, fifth column, and the corresponding NJ tree and *Class Matching* tree are presented in Figure 5.

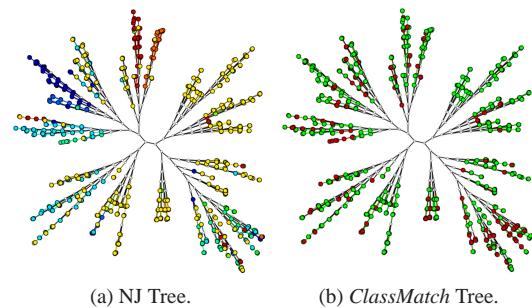


Figure 5: NJ Tree of ALL-Reduced02 collection and corresponding classification result using the LWPR model created in the second iteration.

Table 10 shows the results of the first and second iterations of LWPR model applied to both ALL-Reduced01 and ALL-Reduced02 collections. It can be seen that updating the model improves classifications in both data sets.

Table 10: Classification result comparison using LWPR models created in the iterative model update process on ALL-Reduced01 and ALL-Reduced02 collection.

Iteration	ALL-Reduced01		ALL-Reduced02	
	1	2	1	2
Matching Instances	632 (68.3%)	661 (71.4%)	613 (66.5%)	655 (71.0%)
Non-matching Instances	294 (31.7%)	265 (28.6%)	309 (33.5%)	267 (29.0%)
Accuracy	86.81%	88.40%	86.27%	88.22%
Precision	73.07%	73.99%	70.26%	73.10%
Recall	68.25%	71.38%	66.49%	71.04%

Iteration 3: 6 misclassified instances from classes 2 and 3, that presented high misclassification rates, were selected from NJ tree of ALL-Reduced02 set and used to update LWPR model. This updated model was then used to classify a subset of the ALL-Reduced collection, with 2769 instances, that contains the instances used in the model updates, but does not contain any instance of the initial training set. The results of the classification procedure using the three versions of the LWPR model are shown in Table 11. They show that the guided update provided by the tree layout supports robust convergence of the classifier.

Table 11: Classification result comparison using three versions of the LWPR model on ALL-Reduced subset with 2769 instances.

Iteration	ALL-Reduced Subset		
	1	2	3
Matching Instances	1875 (67.7%)	1946 (70.3%)	2008 (72.5%)
Non-matching Instances	894 (32.3%)	823 (29.7%)	761 (27.5%)
Accuracy	86.61%	87.71%	88.24%
Precision	71.98%	72.84%	74.20%
Recall	67.71%	70.28%	72.52%

4.5. Collection Evolution - New Classes

This experiment verifies how the visualization layout can assist the LWPR model update process in situations where there is a change or evolution in the classes concept, and new classes appear. First, an LWPR model is created through a training set built from instances belonging to a classes and used to classify another collection containing instances of b classes, $a < b$. The performance of the classifier for instances belonging to known classes is examined, as well as in which known classes the instances belonging to unknown classes were inserted into. Then, this model is updated using instances belonging to the previously unknown classes and a new classification is performed. Two model update approaches were examined: the first uses only instances from unknown classes and the second uses a combination of instances belonging to known and unknown classes.

A subset of ETHZ-Reduced collection was used, composed of 717 instances organized in 10 classes, called *ETHZ-Reduced717*. From this subset, 100 instances from classes 4, 10, 16, 23, 25 and 26 were used to build an LWPR model. Classes 5, 7, 8 and 13 were not considered. Table 12 shows how the model classified instances from the 6 known classes,

whereas Table 13 shows in which classes the instances of unknown classes were inserted into.

Table 12: Performance of the classifier for ETHZ-Reduced717 collection on instances of known classes.

Class	Hit Rate
4	35/36 (97.2%)
10	47/47 (100.0%)
16	72/72 (100.0%)
23	207/211 (98.1%)
25	125/133 (93.9%)
26	57/73 (78.1%)

Table 13: Distribution of instances from the 4 ETHZ-Reduced717 unknown classes on the 6 known classes by the LWPR model.

	4	10	16	23	25	26
5	15	3	0	8	3	0
7	1	11	10	0	7	13
8	0	0	0	0	27	0
13	2	21	1	5	9	9

Figure 6 shows the ground truth of the collection (6a), as well as the *ClassMatch* tree from the classification procedure (6b). As expected, 4 branches were totally misclassified, representing the unknown classes.

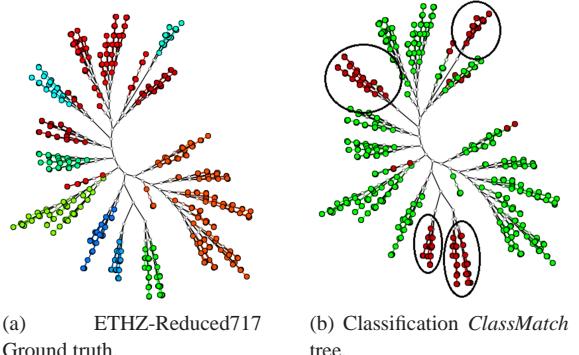


Figure 6: Ground truth and ClassMatch tree comparison for ETHZ-Reduced717 using the model built from instances belonging to 6 classes.

Figure 7 shows that instances from unknown class 8 are positioned into a branch closer to another branch that contains only instances from known class 25, possibly explaining why these instances were labeled to that class.

This example shows the layout capability to provide clues that potential new classes may be appearing in the collection, with instances represented by patterns that are unknown by the model. Here, two distinct branches present the same class label, giving the idea that one of these branches may be an unknown class, whose instances the model considered as a known class. Using multidimensional projections, the presence of partially or totally disconnected groups of instances, or even distant groups with the same class labels, may also indicate the appearing of new classes. These layout trends

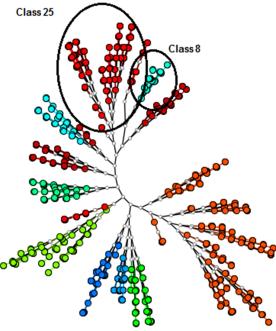


Figure 7: NJ tree of ETHZ-Reduced717 highlighting the relationship between instances from classes 8 and 25.

do not always represent new classes appearing in the collection, but they are clues that may indicate that a further analysis should be performed by the user.

From this result, the LWPR model was updated using two strategies, using only instances from unknown classes and using a combination of instances belonging to known and unknown classes. The numerical results of the classifications are shown in Table 14. The corresponding *Class Matching* trees are presented in Figure 8. The results are better when the model is updated using instances that belong to known and unknown classes. When updated with only instances from the unknown classes, the model correctly classified all the instances from these classes, but as shown in the confusion matrix of Figure 9, several instances from previously known classes that were correctly classified before, were now misclassified.

Table 14: ETHZ-Reduced717 classification result comparison using LWPR model updated with only unknown classes and with instances from all classes.

	4 classes	10 classes
Matching Instances	640 (89.3%)	691 (96.4%)
Non-matching Instances	77 (10.7%)	26 (3.6%)
Accuracy	97.85%	99.00%
Precision	93.26%	97.25%
Recall	89.26%	96.37%

5. Conclusions and Future Work

This work presented a visual classification methodology (SVCM) based on the *Locally Weighted Projection Regression* (LWPR) algorithm and visualization techniques, allowing user to interact with the classification procedure. This done by the construction and adjustment of LWPR models in an iterative process with fast convergence of results.

Several experimental results demonstrate that the association of user and automatic classification procedures using visualization techniques have great potential to produce efficient classifiers. Similarity trees, in particular, show a structure in which the produced hierarchy presented collection

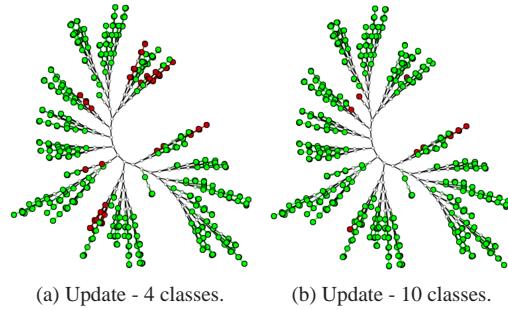


Figure 8: ETHZ-Reduced717 ClassMatch trees using LWPR model updated with only unknown classes (8a) and with instances from all classes (8b).

	4	5	7	8	10	13	16	23	25	26
4	16	20	0	0	0	0	0	0	0	0
5	0	29	0	0	0	0	0	0	0	0
7	0	0	42	0	0	0	0	0	0	0
8	0	0	0	27	0	0	0	0	0	0
10	0	0	0	0	47	0	0	0	0	0
13	0	0	0	0	0	47	0	0	0	0
16	0	0	0	0	0	0	72	0	0	0
23	0	8	0	3	0	3	0	197	0	0
25	0	0	0	13	0	0	2	1	117	0
26	0	6	0	1	0	13	2	0	5	46

Figure 9: Confusion matrix associated to the ETHZ-Reduced717 classification using LWPR model updated with only unknown classes.

details that are not easily observed through layouts generated by other visualization approaches, allowing the user to select representative instances to construct the models. Using the *Class Matching* tree, the user can perform a detailed and efficient analysis of the classification results and comprehend, by looking at its branch structure, the reasons why instances were labeled in a specific classification procedure and for which classes the classifier was not efficient.

The created models can be used to classify any collection presenting the same features. The visual support to the online incremental learning allows fast model adjustments to accommodate evolutions in the class distribution. It also allows the model to deal with situations where new classes appear. In such cases, the position of the instances in the layout provides clues on these new classes, guiding the model updates.

The proposed methodology requires storage of the training instances used in updates involving previously unknown classes. In this sense, a future direction is to investigate the possibility of using statistical information from the local PLS models and decide, based on their behavior, the automatic inclusion of new classes without the need to recreate the LWPR model.

References

- [ADD11] A.MUTHUKUMARAVEL, DR.S.PURUSHOTHAMAN, DR.A.JOTHI: Implementation of Locally Weighted Projection Regression Network for Concurrency Control In Computer Aided Design. *International Journal of Advanced Computer Science and Applications* 2 (2011), 46–50. 3
- [AGD10] AGARWAL M., GOYAL M., DEO M. C.: Locally Weighted Projection Regression for Predicting Hydraulic Parameters. *Civil Engineering and Environmental Systems* 27, 1 (Mar. 2010), 71–80. 3
- [Ank01] ANKERST M.: Visual Data Mining with Pixel-Oriented Visualization Techniques. In *ACM SIGKDD Workshop on Visual Data Mining* (San Francisco, CA, USA, 2001). 2
- [BBS05] BACHMAIER C., BRANDES U., SCHLIEPER B.: Drawing Phylogenetic Trees. In *Algorithms and Computation* (2005), vol. 3827 of *Lecture Notes in Computer Science*, pp. 1110–1121. 2
- [CPMT07] CUADROS A. M., PAULOVICH F. V., MINGHIM R., TELLES G. P.: Point Placement by Phylogenetic Trees and its Application for Visual Analysis of Document Collections. In *IEEE Symposium on Visual Analytics Science and Technology* (Sacramento, CA, USA, 2007), pp. 99–106. 1
- [Do07] DO T.: *Towards Simple, Easy to Understand, An Interactive Decision Tree Algorithm*. Tech. rep., College of Information Technology, Cantho University, 2007. 2
- [DVS01] D’SOUZA A., VIJAYAKUMAR S., SCHAAL S.: Learning Inverse Kinematics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Maui, HI, USA, 2001), vol. 1, pp. 298–303. 3
- [ELS*08] ESS A., LEIBE B., SCHINDLER K., VAN GOOL L.: A Mobile Vision System for Robust Multi-Person Tracking. In *IEEE International Conference on Computer Vision and Pattern Recognition* (Anchorage, AK, USA, June 2008), pp. 1–8. 5
- [FBM11] FLOREZ J., BELLOT D., MOREL G.: LWPR-Model based Predictive Force Control for Serial Comanipulation in Beating Heart Surgery. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (Budapest, Hungary, July 2011), pp. 320–326. 3
- [FM06] FOODY G. M., MATHUR A.: The Use of Small Training Sets Containing Mixed Pixels for Accurate Hard Image Classification: Training on Mixed Spectral Responses for Classification by a SVM. *Remote Sensing of Environment* 103, 2 (2006), 179–189. 1
- [HR04] HEESCH D., RUGER S.: NNk Networks for Content-Based Image Retrieval. In *26th European Conference on Information Retrieval* (Sunderland, UK, 2004), vol. 2997, pp. 253–266. 2
- [JPP12] JOSHI A. J., PORIKLI F., PAPANIKOPOULOS N. P.: Scalable Active Learning for Multiclass Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012), 2259–2273. 2
- [Kei02] KEIM D.: Information Visualization and Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics* 8, 1 (2002), 1–8. 2
- [KPS05] KEIM D. A., PANSE C., SIPS M.: *Information Visualization: Scope, Techniques and Opportunities for Geovisualization*. Elsevier Science Inc., 2005. 2
- [KVS08] KLANKE S., VIJAYAKUMAR S., SCHAAL S.: A Library for Locally Weighted Projection Regression. *Journal of Machine Learning Research* 9 (June 2008), 623–626. 3, 5
- [MN06] MANIYAR D., NABNEY I.: Visual Data Mining using Principled Projection Algorithms and Information Visualization Techniques. In *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2006), pp. 643–648. 2
- [PFP*11] PAIVA J. G., FLORIAN L., PEDRINI H., TELLES G., MINGHIM R.: Improved Similarity Trees and their Application to Visual Data Classification. *IEEE Transactions on Visualization and Computer Graphics* 17 (2011), 2459–2468. 2, 6
- [Por80] PORTER M.: An Algorithm for Suffix Stripping. *Program* 14, 3 (1980), 130–137. 5
- [PSPM12] PAIVA J. G. S., SCHWARTZ W. R., PEDRINI H., MINGHIM R.: Semi-Supervised Dimensionality Reduction based on Partial Least Squares for Visual Analysis of High Dimensional Data. *Computer Graphics Forum* 31 (2012), 1345–1354. 3, 5
- [RÖ6] RÜGER S.: Putting the User in the Loop: Visual Resource Discovery. In *Adaptive Multimedia Retrieval: User, Context, and Feedback* (2006), vol. 3877/2006 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 1–18. 2
- [Rob07] ROBERTS J.: State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization* (Zurich, Switzerland, July 2007), pp. 61–71. 2
- [SD09] SCHWARTZ W. R., DAVIS L. S.: Learning Discriminative Appearance-Based Models Using Partial Least Squares. In *Brazilian Symposium on Computer Graphics and Image Processing* (Rio de Janeiro, RJ, Brazil, Oct. 2009). 5
- [SGCD12] SCHWARTZ W. R., GUO H., CHOI J., DAVIS L. S.: Face Identification Using Large Feature Sets. *IEEE Transactions on Image Processing* 21, 4 (2012), 2245–2255. 5
- [SXY09] SHI C., XU C., YANG X.: Study of TFIDF Algorithm. *Journal of Computer Applications* 29 (2009), 167–170. 5
- [TRP*09] TUIA D., RATLE F., PACIFICI F., KANEVSKI M., EMERY W.: Active Learning Methods for Remote Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing* 47, 7 (2009), 2218–2232. 2
- [TVC*11] TUIA D., VOLPI M., COPA L., KANEVSKI M., MUÑOZ-MARI J.: A Survey of Active Learning Algorithms for Supervised Remote Sensing Image Classification. *IEEE Journal of Selected Topics in Signal Processing* 5, 3 (2011), 606–617. 2
- [VDS05] VIJAYAKUMAR S., D’SOUZA A., SCHAAL S.: Incremental Online Learning in High Dimensions. *Neural Computation* 17 (2005), 2602–2634. 1, 3
- [VS00] VIJAYAKUMAR S., SCHAAL S.: Locally Weighted Projection Regression: An $O(n)$ Algorithm for Incremental Real Time Learning in High Dimensional Space. In *International Conference on Machine Learning* (Stanford, CA, USA, 2000), pp. 1079–1086. 1, 3
- [Wol85] WOLD H.: Partial Least Squares. In *Encyclopedia of Statistical Sciences*, vol. 6. Wiley, New York, NY, USA, 1985, pp. 581–591. 3
- [ZGG09] ZHANG J., GRUENWALD L., GERTZ M.: VDM-RS: A Visual Data Mining System for Exploring and Classifying Remotely Sensed Images. *Computers & Geosciences* 35, 9 (2009), 1827–1836. 2
- [ZH03] ZHOU X. S., HUANG T. S.: Relevance Feedback in Image Retrieval: A Comprehensive Review. *Multimedia Systems* 8, 6 (2003), 536–544. 2