Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Computação e Automação Industrial

"APLICAÇÕES DE MAPAS AUTO-ORGANIZÁVEIS EM MINERAÇÃO DE DADOS E RECUPERAÇÃO DE INFORMAÇÃO"

Márcio Henrique Zuchini
Orientador: Prof. Dr. Fernando José Von Zuben

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação (FEEC - UNICAMP) como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Elétrica
Área de Concentração: Engenharia de Computação

Banca Examinadora

Prof. Dr. Fernando José Von Zuben (Orientador)
UNICAMP – Universidade Estadual de Campinas – Campinas – SP.

Prof. Dr. Márcio Luiz de Andrade Netto (Membro Interno)
UNICAMP – Universidade Estadual de Campinas – Campinas – SP.

Prof. Dr. Leandro Nunes de Castro Silva (Membro Externo)
UNISANTOS – Universidade Católica de Santos – Santos – SP.

Prof. Dr. Carlos Eduardo Câmara (Membro Externo)
USF – Universidade São Francisco – Itatiba – SP.

Campinas – São Paulo – Brasil
Setembro de 2003

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Z82a

Zuchini, Márcio Henrique
Aplicações de mapas auto-organizáveis em mineração de dados e recuperação de informação / Márcio Henrique Zuchini.--Campinas, SP: [s.n.], 2003.

Orientador: Fernando José Von Zuben
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Mapas topográficos. 2. Redes neurais (Computação). 3. Variáveis latentes. 4. Indexação automática. 5. Mapeamento (Matemático). I. Von Zuben, Fernando José. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Resumo

Esta dissertação está voltada ao estudo de dois métodos para mineração de dados de alta dimensionalidade e em grande volume. O *Mapa Auto-Organizável de Kohonen* (*SOM: Self-Organizing Maps*) e o *Mapeamento Topográfico Gerativo* (*GTM: Generative Topographic Mapping*) são métodos já propostos na literatura e caracterizados pela aplicação de procedimentos avançados de visualização gráfica, recorrendo a técnicas distintas de redução de dimensionalidade com requisitos de preservação topológica. Considerando a aplicação dos dois métodos a vários conjuntos de dados, são apresentados resultados promissores, incluindo análise de sensibilidade à variação de parâmetros e proposição de refinamentos empíricos visando incremento de desempenho. Além do emprego de conjuntos de dados já prontos para serem processados, são considerados também textos em português, os quais precisam ser devidamente preparados para análise e requerem formas alternativas de definição do contexto.

Abstract

This dissertation is devoted to the study of two methods for mining high-dimensional and voluminous data. The *Kohonen Self-Organizing Map* (SOM) and the *Generative Topographic Mapping* (GTM) are methods already presented in the literature and characterized by the application of advanced graphical visualization procedures, based on distinct dimension reduction techniques subject to topology preserving requisites. Considering the application of both methods to several data sets, promising results are presented, including sensitivity analysis of parameter variation and the proposal of empirical refinements to improve performance. Besides using data sets already prepared to be processed, texts in Portuguese are also considered. They ask for specific preprocessing before being analyzed, and require alternative proposals to set the context.

A meus pais, Sidney e Ana, sem os quais não chegaria até aqui.

A minha esposa Karen, que ensinou-me a fé e o amor.

A meu filho, Pedro Henrique, a quem tantas horas de brincadeiras e jogos foram adiadas sem que este compreendesse o porquê de “papai está trabalhando”.

Sou eternamente grato a todos.

Agradecimentos

Agradeço primeiramente ao Professor Doutor Fernando José Von Zuben, meu orientador, que acreditou em mim e incentivou-me para a conclusão deste trabalho, face aos inúmeros percalços do trajeto. O Professor Fernando foi quem apresentou-me ao tema central desta dissertação e seu apoio, paciência e direcionamentos valiosos tornaram possível este trabalho.

Agradeço também ao Professor Doutor Carlos Eduardo Câmara, um companheiro de percurso e de discussões profícuas, dentro e fora do contexto deste trabalho, agraciando-me incontáveis vezes com sua paciência, conhecimento e amizade.

Alguns experimentos e vários "entendimentos" não teriam sido possíveis sem a colaboração de Johan Fredrik Markus Svensén, Lalinka de Campos Teixeira Gomes, Oclair Gallacini Prado e Peter Jandl Jr.

Eu agradeço fraternalmente a todos.

Este trabalho contou com o suporte financeiro da CAPES.

Sumário

Lista de Siglas

AC	*Abstract Conceptualization*
AE	*Active Experimentation*
ASSOM	*Adaptive Subspace SOM*
ATW	*Adaptive Tensorial Weighting*
BMU	*Best Matching Unit*
CCA	*Curvilinear Component Analysis*
CDA	*Curvilinear Distance Analysis*
CE	*Concrete Experience*
DNS	*Dynamical Node Splitting*
EM (algoritmo)	*Expectation-Maximization*
FA	*Factor Analysis*
FAQ	*Frequently Asked Questions*
GCS	*Growing Cell Structure*
GG	*Growing Grid*
GIGO	*Garbage In Garbage Out*
GNG	*Growing Neural Gas*
GSOM	*Growing SOM*
GTM	*Generative Topographic Mapping*
IGG	*Incremental Growing Grid*
KDD	*Knowledge Discovery in Databases*
KNIES	*Kohonen Network Incorporating Explicit Statistics*
logL	*log likelihood*
LSI	*Latent Semantic Indexing*
Matriz-U	*Matriz de distância unificada (Unified Distance Matrix)*
MDS	*Multi Dimensional Scaling*
MST	*Minimum Spanning Tree*
NG	*Neural Gas*

PC	*Principal Curves*
PCA	*Principal Component Analysis*
PSOM	*Prunning SOM*
QE	*Quantization Error*
RO	*Reflexive Observation*
SL-SOM	*Self-Labeling SOM*
SOM	*Self-Organizing Maps*
SVD	*Singular Value Decomposition*
TE	*Topographic Error*
WTA	*Winner Takes All*

Capítulo 1

Introdução

Nas últimas duas décadas, a humanidade tem se deparado com um problema que aumenta exponencialmente em complexidade: a *mineração de dados* (*data mining*). Este termo envolve a atividade de aplicar técnicas específicas sobre conjuntos de dados, com o objetivo de revelar padrões, similaridades e diferenças, de produzir regras e resumos, a partir destes dados (Fayyad *et al.* 1996a,b). É notório que a capacidade de geração, obtenção e armazenamento de dados já ultrapassou, em muito, a capacidade humana de analisar e obter informação relevante destes mesmos dados, os quais tendem a ser acondicionados em bases de dados através de ferramentas cada vez mais sofisticadas e eficientes. Além disso, o advento da Internet, aliada a seu crescimento vertiginoso, tem massificado o acesso à "informação" e colocado um volume imenso de dados disponível a praticamente qualquer pessoa em qualquer ponto da Terra.

Da associação entre estes dois fatores emerge um cenário desafiador, voltado para a descoberta de novos conhecimentos e para a recuperação de informações relevantes. O volume crescente de dados gerados e disponibilizados por governos, empresas, universidades e pessoas físicas, traz uma dificuldade crescente para responder a perguntas como:

- "o que se pode extrair de informação a partir destes dados?"
- "quais os agrupamentos (*clusters*) existentes nestes dados?"
- "o que torna estes agrupamentos semelhantes (ou distintos) entre si?"

Considere, também, os casos em que as informações são disponibilizadas em forma textual, como é o caso de grande parte do material existente na Internet. Atualmente, seria extremamente difícil, senão impossível, catalogar esta informação por meios manuais, e

ferramentas tradicionais de recuperação de informação, que tentam recuperar textos cujos conteúdos estejam associados a um determinado assunto, freqüentemente produzem resultados insatisfatórios. É comum, num processo de recuperação de informação, serem obtidas imensas quantidades de obras de valor desconhecido e questionável (Lagus, 2000). Atender a este novo problema significa responder a mais uma questão, consideravelmente mais difícil:

- "quais são as outras informações disponíveis e úteis relacionadas a este assunto?"

A resposta a esta última questão tem sido abordada por um novo termo na literatura, a *mineração de textos* (*text mining*) (Lagus, 2000). A mineração de textos envolve a aplicação de técnicas e ferramentas, notadamente com uso de redes neurais, nos problemas de organização, classificação e agrupamento de dados em forma textual.

Neste cenário já complexo, considere ainda a possibilidade de haver dados incorretos, inverídicos e contraditórios nos conjuntos, o que amplia ainda mais o elenco de dificuldades e desafios a serem superados.

Aparentemente, a capacidade do cérebro humano em simplificar, generalizar, formular hipóteses e testá-las, sem um tutor para indicar o caminho correto a seguir, parece ter sido a força motriz das realizações humanas desde sua existência. Tem sido notável como o ser humano lidou com tais dificuldades até o momento. Infelizmente, esta capacidade parece estar cada vez mais aquém das necessidades para lidar com volumes tão grandes de dados. Faz-se necessário, cada vez mais, a utilização de ferramentas e métodos capazes de operar sobre dados multidimensionais, capazes de comparar e classificar conjuntos de dados tão volumosos que inviabilizariam a simples leitura destes, capazes de simplificar e evidenciar aspectos relevantes de conjuntos de dados que, de outra forma, estariam ocultos sob o grande volume de dados. Faz-se necessária a pesquisa e a descoberta de formas mais eficientes de responder às perguntas acima, sejam elas dirigidas pela disponibilidade de dados, sejam elas orientadas a um contexto ou assunto em particular.

O interesse crescente da comunidade científica em torno de métodos automáticos para análise de dados ou, no mínimo, auxiliados por computador, tem gerado diversos textos cujo objetivo central é a discussão de métodos capazes de obter informações relevantes do imenso volume de dados disponíveis, como por exemplo Fayyad *et al.* (1996d) e Michalski *et al.* (1998), referências importantes para uma introdução à mineração de dados.

A utilização de estratégias baseadas em modelos comportamentais do cérebro ou fundamentadas na teoria de probabilidades parece ser um caminho bastante promissor e direcionou este trabalho para o estudo de dois modelos em particular, o *Mapa Auto-Organizável de Kohonen* (*SOM: Self-Organizing Maps*) e o *Mapeamento Topográfico Gerativo* (*GTM: Generative Topographic Mapping*), na tentativa de responder, pelo menos em parte, às quatro perguntas formuladas anteriormente.

A escolha das ferramentas SOM e GTM baseou-se num conjunto de características apresentadas por ambas, dentre as quais destacam-se:

- capacidade de operar com conjuntos volumosos de dados;
- capacidade de operar com dados representados por um grande número de características (alta dimensionalidade);
- utilização de aprendizado não supervisionado;
- capacidade de realizar *projeção de dados*, reduzindo assim a dimensionalidade do conjunto de dados;
- capacidade de realizar *redução de dados*, diminuindo a quantidade de dados exibidos pela ferramenta;
- possibilidade de avaliação gráfica dos resultados obtidos;
- algoritmos relativamente simples e rápidos;
- capacidade de generalização dos modelos, de forma a possibilitar a representação de dados não disponíveis no momento do treinamento.

Estas características, experimentadas e comprovadas ao longo da pesquisa que resultou nesta dissertação, permitem colocar estas ferramentas entre aquelas com grande potencial de aplicação nas tarefas de mineração de dados e recuperação de informação.

1.1 Objetivos e principais contribuições

Este trabalho busca estudar, aplicar e avaliar métodos atualmente considerados entre os mais promissores nas tarefas de *mineração de dados* e *recuperação de informação*, apresentando e discutindo suas características mais relevantes, consideradas as tarefas propostas. Mais especificamente, esta dissertação buscou mostrar que as duas ferramentas analisadas em maior profundidade, o SOM e o GTM, são técnicas poderosas e podem conduzir a resultados promissores mesmo na presença de problemas práticos altamente desafiadores.

Alguns dos principais métodos para aplicação em mineração de dados são testados e comentados de forma resumida, apresentando-se suas principais características, bem como algumas de suas limitações. Através dos testes realizados, é demonstrada a dificuldade apresentada por tais métodos quando envolvidos na análise de conjuntos volumosos de dados e multidimensionais, o que inviabiliza a aplicação destas técnicas bem difundidas na literatura junto às tarefas propostas nessa dissertação.

Com relação às ferramentas SOM e GTM, são avaliadas as principais heurísticas existentes na literatura para a obtenção de bons modelos dos dados, com uma discussão da validade e de problemas que eventualmente estas heurísticas causam em sua aplicação sem critérios bem definidos. Incluem-se aqui simulações e testes para avaliar a influência dos parâmetros de controle dos algoritmos sobre os resultados obtidos. São propostas heurísticas para obtenção de bons resultados, considerando a tarefa de mineração de dados, para as ferramentas SOM e GTM. Ambas as ferramentas são também aplicadas a uma base de dados inédita, envolvendo estilos de aprendizado, ilustrando a aplicação das heurísticas propostas e verificando várias características que tornam, ambas as ferramentas, aliadas poderosas na tarefa de mineração de dados.

Por fim, os modelos SOM e GTM são aplicados ao problema de recuperação de informação textual, que consiste, na codificação e recuperação de documentos considerando-se a similaridade de conteúdos e assuntos contidos nestes (isto é, conforme seu *contexto*). Neste estudo, os documentos de texto são representados pelas duas ferramentas, agrupados segundo seu contexto. A estratégia proposta envolve um modelo hierárquico, originalmente

proposto por Honkela *et al.* (1996a). Nesta estratégia, um mapa SOM, previamente adaptado e representando o contexto médio das palavras existentes no corpo de texto, gera um conjunto de vetores representando cada documento de texto. Estes vetores, uma espécie de "assinatura estatística" dos documentos, são usados para adaptar um segundo mapa SOM, que representa então a similaridade contextual dos documentos.

Esta dissertação mostra que a ferramenta GTM é uma alternativa possível ao SOM para representar os documentos segundo seu contexto, propondo um modelo híbrido SOM-GTM. É proposto e verificado experimentalmente que o uso do 2º BMU (*Best Matching Unit*) na construção dos vetores de documentos, aumenta a capacidade das ferramentas em representar a similaridade contextual dos documentos.

Finalmente, é proposta uma alternativa para a equação de contexto médio das palavras do corpo de texto, considerando o *contexto médio por documento*. Esta equação é utilizada para gerar os dados que são aplicados ao mapa SOM que gerará, posteriormente, os vetores de documentos. Os testes realizados mostram que esta nova proposta aumenta sensivelmente a capacidade das ferramentas em agrupar os documentos conforme sua similaridade contextual.

1.2 Organização do trabalho

Nesta dissertação, a palavra "método" é usada como sinônimo de "conjunto dos meios dispostos convenientemente para alcançar um fim e especialmente para chegar a um conhecimento científico ou comunicá-lo aos outros" (Michaelis, 1998). Entende-se por "método" um algoritmo, uma técnica, um conjunto de procedimentos e atitudes, ou ainda um híbrido entre todos com a finalidade de obter conhecimento científico.

Alguns dos principais métodos para aplicação em mineração de dados são abordados no Capítulo 2, desde os modelos históricos, relativamente pouco aplicados na atualidade face o imenso volume de dados disponíveis, até as técnicas mais recentes, desenvolvidas para lidar com grandes volumes de dados. Inclui-se aqui métodos de agrupamento, de projeção e modelos gerativos baseados em probabilidade. Estes métodos são descritos e analisados de forma resumida, apresentando-se suas principais características, bem como algumas de suas

limitações. Foram executados testes com todas as ferramentas, com o propósito de permitir uma análise prática das vantagens e desvantagens de cada uma, considerando os objetivos e o contexto dos experimentos trabalhados nesta dissertação.

Os dois métodos que serão utilizados mais extensivamente nas aplicações, o SOM e o GTM, são examinados com maiores detalhes no Capítulo 3 e no Capítulo 4, respectivamente. São incluídas aqui simulações e testes para avaliar a influência dos parâmetros de controle dos algoritmos sobre os resultados obtidos.

O Capítulo 5 apresenta uma série de testes e simulações de algumas das ferramentas para mineração de dados apresentadas anteriormente. Foram utilizados alguns conjuntos de dados disponíveis publicamente na Internet, os quais são comumente utilizados para avaliar o desempenho de ferramentas de mineração de dados. Mais especificamente, as ferramentas SOM e GTM são aplicadas e seus resultados são comentados. Neste capítulo, são apresentadas heurísticas para obtenção de bons resultados a partir das ferramentas SOM e GTM, bem como uma discussão de diversas características úteis oferecidas pelas ferramentas. Os resultados obtidos têm caráter fortemente experimental, uma propriedade comum nas tarefas de mineração de dados. O SOM e o GTM são também aplicados a uma base de dados inédita, envolvendo estilos de aprendizado.

O Capítulo 6 avalia a aplicação dos dois modelos citados ao problema de recuperação de informação textual. Neste capítulo, é proposto e testado um modelo híbrido SOM-GTM para a representação de similaridade contextual entre documentos, além de outras propostas que buscam melhorar a sensibilidade das ferramentas ao conteúdo dos documentos de texto.

No Capítulo 7 são apresentadas as conclusões deste trabalho, resumindo-se os principais conceitos abordados e verificados, as principais contribuições e incluindo possíveis extensões para futuros trabalhos.

Capítulo 2

Métodos para Mineração de Dados

Este capítulo descreve alguns dos principais métodos existentes com aplicação (não exclusiva) em *Mineração de Dados*, uma das principais tarefas do processo de *Descoberta de Conhecimento em Banco de Dados* (*KDD: Knowledge Discovery in Databases*). São abordados principalmente métodos de agrupamento, de projeção (lineares e não lineares) e modelos gerativos (*Generative Models*), sendo discutidas suas principais características e aplicações. Além desses, são citados alguns métodos simples para visualização de dados multidimensionais, eventualmente úteis numa análise preliminar dos dados. Dois métodos em particular, o Mapa Auto-Organizável, um modelo híbrido de agrupamento e projeção, e o GTM, um modelo gerativo baseado em variáveis latentes, serão abordados com maior detalhe em capítulos subseqüentes.

2.1 Introdução

É sabido que a tecnologia atual permite a geração e armazenamento de quantidades imensas de informação sob as mais diversas formas: imagens, sons, conjuntos de atributos etc. O atual volume e a elevada taxa de crescimento destes bancos de dados ultrapassou a capacidade humana de analisar, interpretar e utilizar a informação neles contida, criando assim a necessidade de métodos e ferramentas eficientes capazes de manipular esta massa de dados (Fayyad *et al.* 1996a,b,c). O termo KDD foi criado para nomear o processo completo de descoberta de conhecimento a partir de conjuntos de dados e representa muito mais do que apenas a aplicação de técnicas capazes de revelar similaridades e diferenças, de produzir regras e resumos dos dados. A este conjunto de atividades em particular reserva-se o termo *"mineração de dados"* que, embora considerado por Fayyad *et al.* (1996a,b) como o passo central de todo o processo, não é o único.

Não há um consenso sobre a terminologia utilizada pelos autores nesta área recente de pesquisa. É possível encontrar o termo "mineração de dados" como sinônimo de "KDD", como nota-se em Mitchell (1999). Holsheimer & Siebes (1994) chamam "mineração de dados" a *"um tipo especial de aprendizado de máquina onde o ambiente é visto através de um banco de dados"*.

Retornando ao conjunto de atividades associado ao primeiro conceito de mineração de dados apresentado acima, ele deve ser precedido por atividades essenciais que vão desde o próprio entendimento do domínio da aplicação e de seus objetivos até a interpretação dos resultados. As etapas anteriores ao processo de mineração, mais especificamente, a remoção de ruído, a escolha de variáveis relevantes, a manipulação de valores ausentes e a escolha do método de mineração adequado (considerando o objetivo proposto: classificação, regressão, modelagem etc.) devem receber atenção especial e jamais serem relegadas a papel menos importante, pois os métodos de mineração são fiéis ao raciocínio *GIGO* (*Garbage In Garbage Out*). Sem estes cuidados corre-se o risco de obter resultados pouco confiáveis, pois padrões e regras potencialmente inválidas ou sem interpretação adequada podem emergir. Fayyad *et al.* (1996b) referem-se a essa atividade perigosa como *"data dredging"*.

Há diversos métodos aplicados na mineração de dados vindos de várias áreas do conhecimento e termos como *"análise exploratória de dados"* (Jain & Dubes, 1988; Tukey, 1977), *"análise de agrupamentos"* (Everitt, 1993) ou *"classificação automática"* (Costa, 1999), *"reconhecimento de padrões"* (Duda *et al.* 2000; Bishop, 1995), *"aprendizado de máquina"* (Michalski *et al.* 1998) e outros (Fayyad *et al.* 1996a) são freqüentemente usados para referir-se a tais métodos. Nesta dissertação optou-se por uma taxonomia baseada na idéia de Kaski (1997) e Svensén (1998) de que, na mineração de dados multidimensionais, só terão utilidade métodos capazes de revelar a *estrutura inerente do conjunto de dados*, pois em última instância é exatamente a relação de similaridade/dissimilaridade o que se busca entender. Pode-se dividir os métodos em conjuntos conforme a maneira de (tentar) exibir a estrutura topológica dos dados:

- *métodos simples de visualização*, capazes de gerar gráficos e resumos rápidos do comportamento dos dados e úteis para análise preliminar à mineração de dados;

- *métodos de agrupamento*, com objetivo de descobrir agrupamentos de dados com características semelhantes entre si;
- *métodos de projeção*, baseados na idéia de projetar os dados de seu espaço original para um espaço de menor dimensão procurando revelar a estrutura topológica dos dados e
- *métodos baseados em modelos gerativos*, onde os pontos no espaço de dados são entendidos como sendo *gerados* por um modelo que representa a função de distribuição de probabilidade dos dados.

A menos que explicitamente necessário, os dados serão representados por um conjunto $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_N\}$, $\mathbf{V} \subseteq \Re^D$, com vetores $\mathbf{v}_n = [v_{n1}, \ ... \ , \ v_{nD}]^T$, $n = 1, ..., N$ e $v_{nd} \in \Re$, $d = 1, ..., D$. Cada vetor $\mathbf{v}$ representa um objeto (um ponto) no espaço D-dimensional através de seus D atributos. A Tabela 2-1 apresenta um exemplo de um conjunto genérico de dados:

Tabela 2-1 – Representação tabular de um conjunto de dados em termos de vetores de atributos ou características, onde v_{nd} é o d-ésimo atributo do n-ésimo objeto.

	Atributos			
Objetos	**1**	**2**	...	***D***
1	v_{11}	v_{12}	...	v_{1D}
2	v_{21}	v_{22}	...	v_{2D}
:	:	:		:
N	v_{N1}	v_{N2}	...	v_{ND}

Considerando as representações de dados como definidas na Tabela 2-1, Mitchell (1999) afirma que a área de KDD encontra-se na primeira geração de algoritmos, tipicamente limitados a tratar dados descritos por conjuntos de registros de D atributos (numéricos ou simbólicos), ou seja, ainda não há técnicas consistentes que utilizem imagens, sons, texto puro, conhecimento simbólico prévio, dentre outros aspectos, no processo de KDD. Tratando especificamente da WEB[1] questiona-se até mesmo se esta seria organizada o suficiente para que métodos de mineração de dados sejam aplicados de forma razoável (Etzioni, 1996).

[1] Sigla reduzida de *World Wide Web* (*WWW*), ambos usadas como sinônimos da Internet.

2.2 Métodos simples de visualização

A maioria dos métodos simples para visualização de dados multidimensionais propostos na literatura baseiam-se em gráficos ou cálculos matemáticos que, de alguma forma, representam ou resumem características dos conjuntos de dados. Como exemplo, Tukey (1977) propõe, dentre vários métodos gráficos e numéricos, o cálculo de um *resumo de cinco números* para conjuntos de dados: o maior e menor valores, a média e o 1^o e 3^o quartis. A idéia geral destes métodos simples consiste em plotar gráficos (em geral bidimensionais) com os atributos dos dados diretamente relacionados entre si, ou então algum resumo matemático destes, como médias, logaritmos, potências etc. Estes gráficos formariam uma espécie de "descrição sucinta" dos conjuntos de dados cuja análise preliminar possibilitaria um melhor entendimento dos dados e evitaria a aplicação negligente de técnicas de mineração de dados, o que muitas vezes leva a resultados sem sentido (Fayyad *et al.* 1996a,b; Everitt, 1993).

Uma divisão simplista feita por Everitt (1993), e aqui resumida, classifica estes métodos em:

1. histogramas e gráficos, relacionando atributos ou resumos destes entre si; e
2. representações icônicas, onde normalmente associa-se um atributo do dado a um atributo de uma figura que o representará.

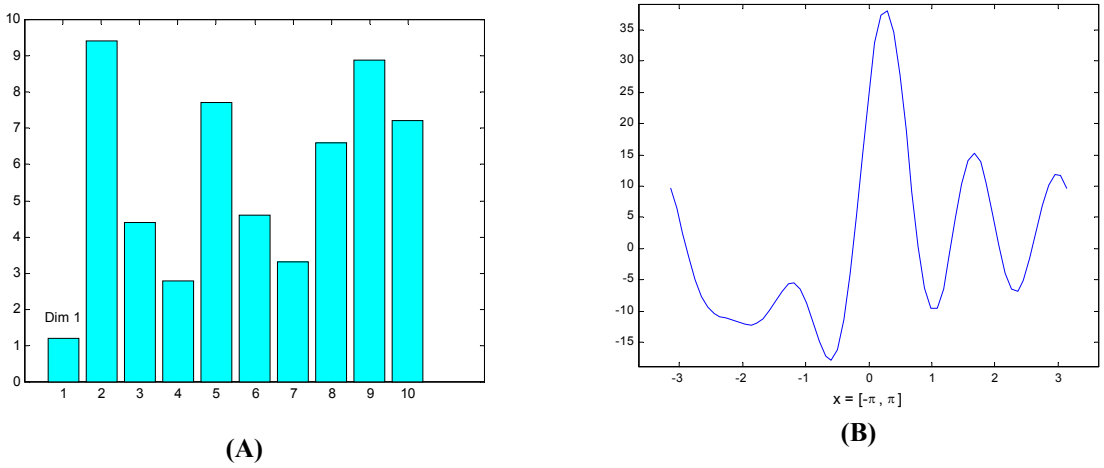
Jain & Dubes (1988) diferenciam as representações icônicas de métodos de projeção não lineares afirmando que enquanto estes tentam preservar a estrutura dos dados num gráfico com apenas duas coordenadas (dos atributos mais relevantes, normalmente), representações icônicas tentam preservar esta mesma estrutura através de uma figura controlada por todos os atributos. A grosso modo, entretanto, pode-se considerar as representações icônicas como uma espécie de projeção não linear dos dados. Veja a Seção 2.4.2 para mais detalhes sobre projeções não lineares.

Um método bastante simples é a visualização de todas as dimensões (ou daquelas selecionadas) como um gráfico de barras, onde cada barra representa uma dimensão, conforme ilustrado na Figura 2-1-A. A Figura 2-1-B representa o mesmo objeto

$\mathbf{v} = [v_1, ..., v_D]$ pela *"curva de Andrews"* (Andrews, 1972), obtida através do cálculo da função

$$f(\mathbf{v}, t) = \frac{1}{\sqrt{2}} v_1 + v_2 \operatorname{sen}(t) + v_3 \cos(t) + v_4 \operatorname{sen}(2t) + v_5 \cos(2t) + \ldots$$

sobre o intervalo $-\pi < t < \pi$ e $D = 10$ no exemplo. Cada componente v_i é um atributo do objeto, conforme a notação proposta na Tabela 2-1. Uma propriedade interessante desta função é a preservação da relação de vizinhança topológica entre os objetos no sentido de que dois pontos próximos no espaço de entrada serão representados por curvas próximas para todos os valores de t (Everitt, 1993).

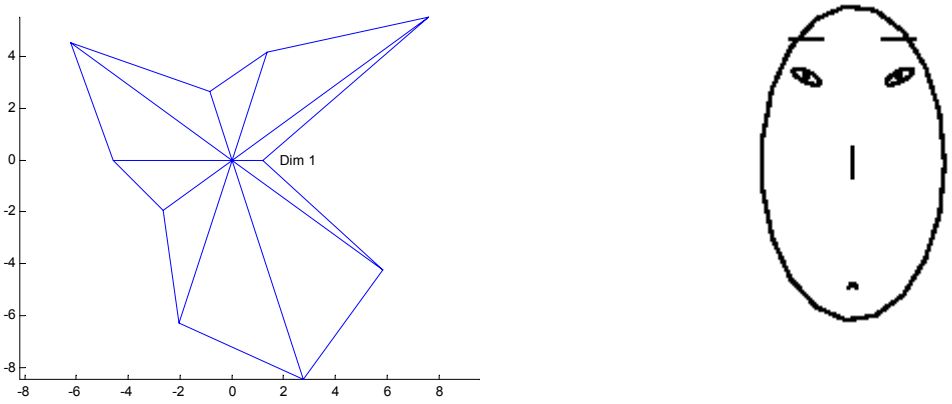


(A) **(B)**

Figura 2-1 – Visualização de um item de dado em $\Re^{10}$ com valores aleatórios através de um gráfico de barras (esquerda) e da *"curva de Andrews"*

Várias curvas de Andrews diferentes podem ser construídas com a simples permutação das variáveis. Como, em geral, as baixas freqüências (v_1 , v_2 , v_3) são mais evidenciadas no gráfico, seria interessante associá-las com os atributos mais importantes dos objetos sendo representados (Everitt, 1993). Infelizmente, esta informação não é, em geral, previamente conhecida.

Outras possibilidades incluem a representação dos dados através de figuras poligonais e as *"faces de Chernoff"* (Chernoff, 1973), cujos exemplos podem ser vistos na Figura 2-2. Os ícones poligonais podem ser gerados com todas as dimensões (caso em que tem seu uso limitado dada a sobreposição de figuras) ou ainda pode-se escolher duas delas (as mais importantes) para posicionar o centro da figura num plano cartesiano, gerando um gráfico de dispersão ou *scatterplot* (Everitt, 1993; Kaski, 1997).



(A) (B)

Figura 2-2 – Visualização de um item de dado em $\Re^{10}$ com valores aleatórios usando polígonos (A) e "faces de Chernoff" (B). Figura à direita adaptada de Kaski (1997).

As faces de Chernoff são geradas com cada dimensão dos dados controlando uma característica da face, como a largura e curvatura da boca, a separação entre os olhos etc. O autor argumenta que sua utilização presta-se a avaliar dados em $\Re^D$ considerando $D \leq 18$.

Embora interessantes e com valor histórico, a capacidade de visualizar relações entre os dados através dos métodos citados degenera rapidamente à medida que aumenta o número de dimensões. À exceção do *resumo de cinco números*, nenhum dos outros métodos executa redução de dados, ou seja, se o conjunto de entrada for numeroso, a figura resultante da visualização de todos os dados individuais será provavelmente incompreensível (Kaski, 1997). Embora seja possível identificar a presença de agrupamentos em algumas situações, estes métodos devem ser tomados apenas como ferramentas adicionais capazes de auxiliar na tarefa de mineração de dados (Everitt, 1993; Jain & Dubes, 1988).

2.3 Métodos de Agrupamento

A tarefa de reunir objetos semelhantes em grupos é um processo usualmente adotado pelo ser humano ao longo da história da humanidade, podendo inclusive ser associado à própria criação da linguagem. As palavras podem ser interpretadas como rótulos associados a conjuntos de objetos semelhantes. Tomando apenas adjetivos como exemplo, as palavras "feroz", "saboroso", "venenoso", etc., são rótulos que determinam a própria capacidade de adaptação ao meio, ao permitir classificar e discriminar agentes e objetos do meio.

Considera-se *agrupamento* uma região do *D*-espaço (espaço *D*-dimensional que congrega os *D* atributos) com densidade de pontos relativamente elevada e separada de outras regiões densas por regiões com baixa densidade (Everitt, 1993). Pode-se entender, resumidamente, que *métodos de agrupamento* são aqueles que buscam dividir um conjunto de objetos não rotulados em grupos (partições) de forma que os objetos de cada grupo tenham mais semelhanças entre si do que em relação aos objetos de qualquer outro grupo. Neste processo necessariamente não supervisionado, segundo Costa (1999), tanto o número ótimo de grupos como as características particulares revelando semelhanças (ou diferenças) devem ser determinados pelo próprio processo (Everitt, 1993).

Esta característica aponta para métodos não triviais, uma vez que a quantidade de formas possíveis de criar *K* partições para um grupo de *N* objetos pode ser assustadoramente grande, tornando a busca exaustiva por um particionamento ótimo computacionalmente proibitiva, ao menos atualmente. O valor exato para este número de formas possíveis quando *K* é conhecido é dado pelo número de Stirling do segundo tipo (Jain & Dubes, 1988):

$$S(N,K) = \frac{1}{K!}\sum_{i=1}^{K}(-1)^{K-i}\binom{K}{i}i^{N}$$

Caso *K* seja desconhecido (o que é normalmente o caso), este número de possibilidades é ainda maior, pois é dado por um somatório de números de Stirling (Costa, 1999):

$$\sum_{l=1}^{P} S(N,l)$$

onde *P* é o número máximo de partições, previamente arbitrado. Este somatório é também conhecido como *número de Bell*.

Deve ser claro que o conceito de similaridade entre os pontos no *D*-espaço está diretamente relacionado ao tipo de métrica considerada. É comum o uso da métrica euclidiana, embora várias outras tenham sido propostas na literatura (veja Costa, 1999 e Jain *et al.* 1999 para uma revisão). A escolha da métrica afeta diretamente a quantidade e a forma de grupos encontrados pelos algoritmos de agrupamento, pois aspectos da estrutura do espaço podem (ou não) ser levados em consideração durante o processo conforme a métrica. É patente,

pois, a dificuldade na escolha da métrica quando não se tem informação prévia sobre o conjunto de dados a ser analisado. O risco é o de que o algoritmo "encontre grupos segundo sua ótica", ou seja, pode-se procurar (e encontrar) grupos com formas previamente supostas onde estes, de fato, não existam ou, então, deixar de encontrar agrupamentos cuja discriminação fica obscurecida pela métrica adotada. A Figura 2-3 ilustra um caso em que algoritmos baseados em distância (como é o caso do SOM, que utiliza distância euclidiana) têm péssimo desempenho, pois é um exemplo onde um intérprete humano utiliza-se de muito mais informação prévia do que aquela disponibilizada ao algoritmo, no caso, apenas a distância entre pontos. A este método de agrupamento Michalski & Kaufman (1998) denominam métodos de *agrupamento conceitual*: é fácil observar retângulos no exemplo da figura porque o ser humano *conhece previamente o conceito* de um retângulo, sendo direta, portanto, a associação. Um *conceito* é definido como sendo um conjunto de objetos que possuem um conjunto de propriedades que os diferenciam de outros conceitos (o que vem a ser uma descrição muito semelhante, senão idêntica, ao próprio conceito de *agrupamento*).

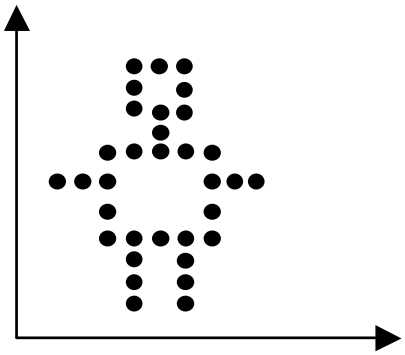


Figura 2-3 – Conjunto de dados para o qual algoritmos baseados em métricas de distância apresentam desempenhos ruins. Possíveis análises revelam 2 retângulos e 4 linhas, uma figura humanóide etc. Adaptado de Kubat *et al.* (1998).

Para que um algoritmo obtenha resultados semelhantes ele deverá basear-se num banco de conceitos previamente informado, ou então possuir algum método para adquirir (aprender) tais conceitos. Uma boa referência para aprendizado de máquina e mineração de dados é Michalski *et al.* (1998).

A Figura 2-4 apresenta uma hierarquia simplificada dos métodos de agrupamento propostos na literatura.

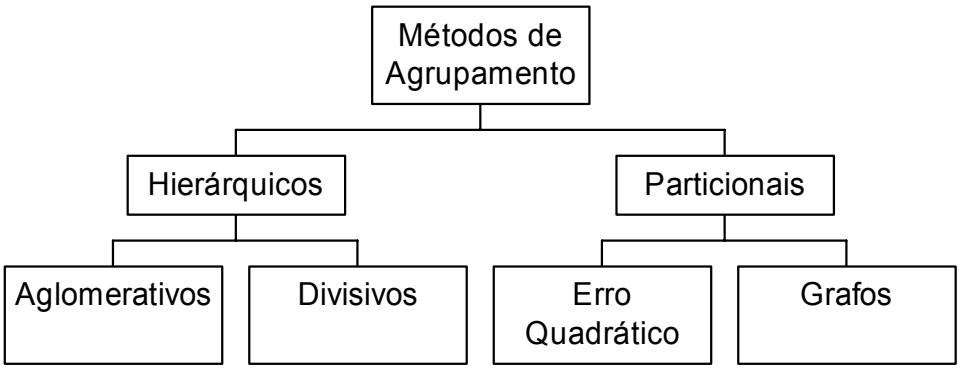


Figura 2-4 – Classificação simplificada dos métodos de agrupamento. Adaptado de Jain *et al.* (1999).

Entretanto, deve-se levar em consideração algumas características que independem da taxionomia proposta, qualquer que seja ela. Por exemplo, os métodos podem fazer com que um objeto pertença exclusivamente a um agrupamento (isto é, a intersecção de agrupamentos é vazia) ou então utilizar conceitos de lógica nebulosa para associar graus de pertinência dos objetos para com os conjuntos. Métodos podem considerar todos os atributos dos objetos simultaneamente durante o processo de agrupamento (métodos *politéticos*) ou então considerar cada atributo individual e seqüencialmente (*monotéticos*). Também um método pode considerar todo o conjunto de objetos simultaneamente (métodos *não incrementais*) ou então tomar pequenas porções ao longo do processo (*incrementais*), sendo esta uma característica importante no processo de mineração de dados face aos imensos conjuntos de dados comumente observados. Sugere-se consultar Jain & Dubes (1988) e Everitt (1993) para uma excelente introdução a métodos de agrupamento e Jain *et al.* (1999) para uma revisão atualizada dos conceitos.

2.3.1 Agrupamentos Hierárquicos

Os métodos *hierárquicos*, de modo geral, tratam o conjunto de dados como uma estrutura de partições, cada uma correspondendo a um agrupamento, hierarquicamente organizadas segundo a similaridade entre seus objetos. Os métodos *divisivos* consideram a princípio a existência de uma única partição (o próprio conjunto de dados) e atuam subdividindo esta partição em uma série de partições aninhadas. Já os métodos *aglomerativos* partem do oposto, fundindo agrupamentos individuais (inicialmente cada grupo contém um único objeto) em partições maiores até a obtenção de uma única partição contendo todos os objetos do conjunto.

Algoritmos hierárquicos aglomerativos (mais eficientes e representativos que os divisivos, segundo Costa, 1999) geralmente trabalham com uma matriz de distâncias **D** representando a similaridade (ou a dissimilaridade) entre todos os possíveis pares de N objetos do conjunto de dados. Esta matriz **D** de elementos d_{ij} (i,j = 1, ..., N) é, portanto, simétrica de diagonal nula e ordem N, sendo usada para decidir quais grupos serão fundidos entre si. Em geral, unem-se dois ou mais grupos que apresentam a menor "distância" entre si. A distância entre dois grupos é normalmente avaliada segundo os critérios de *ligação simples* (*single link*) ou *ligação completa* (*complete link*).

A idéia destes critérios é ilustrada na Figura 2-5 e é mais facilmente entendida supondo-se, inicialmente, dois agrupamentos quaisquer já existentes, **A** e **B**, e uma matriz **D** de distâncias entre todos os pares de objetos. O critério de ligação simples define d_{AB} como a menor distância entre todos os pares (x,y) de objetos onde $x \in$ **A** e $y \in$ **B** (Figura 2-5-A). Já o critério de ligação completa considera d_{AB} como a maior distância entre todos os pares (x,y) tomados conforme a regra já citada (Figura 2-5-B). Após calculadas as distâncias entre os agrupamentos conforme os critérios já descritos, os algoritmos promovem a união dos agrupamentos com a menor distância entre si.

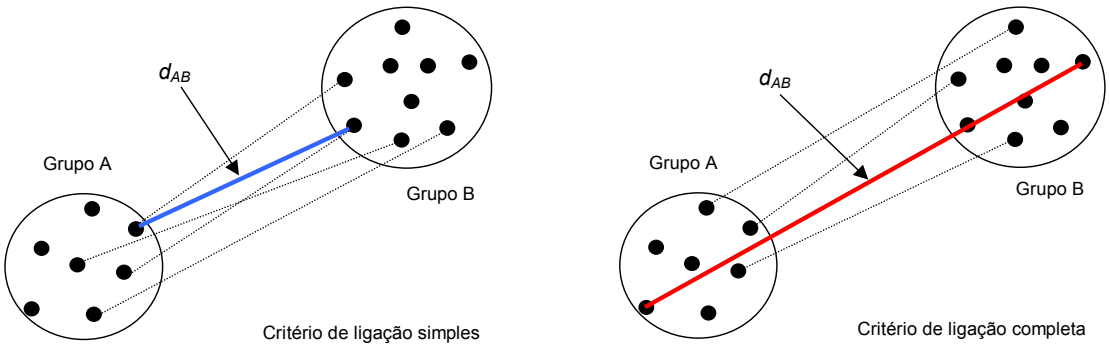


Figura 2-5 – Ilustração do critério de ligação simples e ligação completa. Supondo-se dois agrupamentos pré-existentes A e B, a ligação simples define a distância entre os dois grupos como a menor dentre todas as distâncias entre os pares de objetos (x,y), onde $x \in$ A e $y \in$ B, respectivamente. O critério de ligação completa define a distância entre os dois grupos como sendo a maior distância dentre todas as distâncias entre os mesmos pares de objetos (x,y). Em linhas pontilhadas, estão ilustradas algumas das distâncias entre outros pares de objetos.

A saída típica destes algoritmos é um *dendrograma* (Figura 2-6 C e D), uma espécie de grafo de árvore que representa as junções sucessivas das partições e que pode gerar agrupamentos diferentes conforme o nível em que é seccionada.

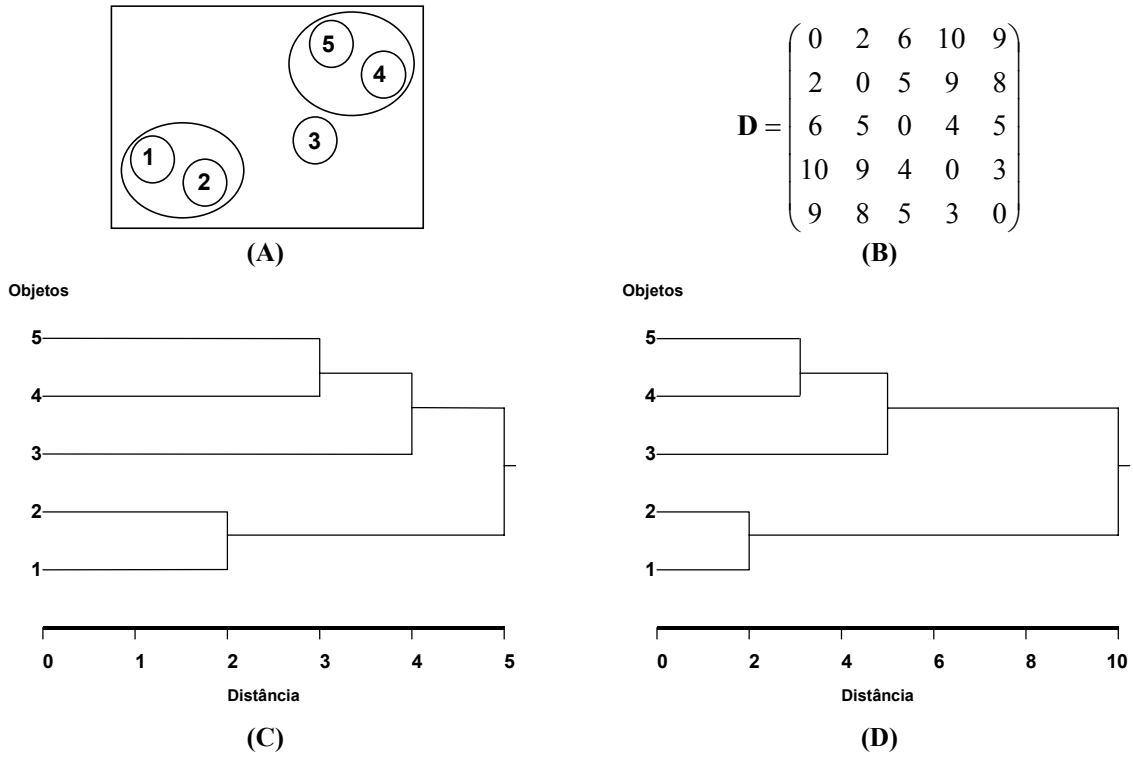


Figura 2-6 – Dendrogramas obtidos segundo os critérios de ligação simples e completa. Em (A) um conjunto hipotético de objetos com sua matriz de distâncias D descrita em (B), sem consideração de escala. Pelo critério de ligação simples (C), os grupos {4} e {5} são unidos pela menor distância entre si (d_{45}=3, no exemplo), formando o grupo {4,5}. Este grupo é unido com o grupo {3} pela menor distância entre eles igual a 4 (pois d_{34}=4 e d_{35}=5). No critério de ligação completa (D), o grupo {3} é unido ao grupo {4,5} com distância igual a 5. Adaptado de Everitt (1993)

De acordo com Jain *et al.* (1999), o critério de ligação simples possui a característica de produzir agrupamentos com tendência hiperelipsoidal, ao passo que o critério de ligação completa forma agrupamentos mais compactos com tendência hiperesférica. Técnicas hierárquicas são comuns onde se necessita gerar uma taxionomia facilmente obtida pelo dendrograma (por exemplo nas áreas de biologia e ciências sociais), mas são impraticáveis quando o número de objetos é elevado (Jain & Dubes, 1988), o que infelizmente é comum nos processos de mineração de dados.

2.3.2 Agrupamentos Particionais

Os métodos particionais dividem o conjunto dos N objetos em K agrupamentos sem relacioná-los hierarquicamente entre si, como o fazem métodos hierárquicos. Normalmente, as partições são obtidas pela otimização de um critério definido local (sobre um subconjunto de objetos) ou globalmente (sobre todo o conjunto) na forma de uma função-objetivo. Sua maior vantagem é poder atuar sobre conjuntos com elevado número de objetos, pois tais métodos em geral têm complexidade $O(N)$, N = número de objetos do conjunto de dados. Por outro lado, possuem uma séria restrição relacionada às funções-objetivo usadas que, em geral, assumem que K é conhecido. Assim, uma escolha errada de K provoca a imposição deste número de agrupamentos ao conjunto.

Um dos métodos mais conhecidos, o *k-means* (MacQueen, 1967), emprega como função-objetivo o erro quadrático total definido genericamente para um certo número K de agrupamentos por

$$e_k^2 = \sum_{j=1}^{K} \sum_{i=1}^{N} \left\| \mathbf{v}_i^{(j)} - \mathbf{c}_j \right\|^2 \qquad \textbf{Equação 2-1}$$

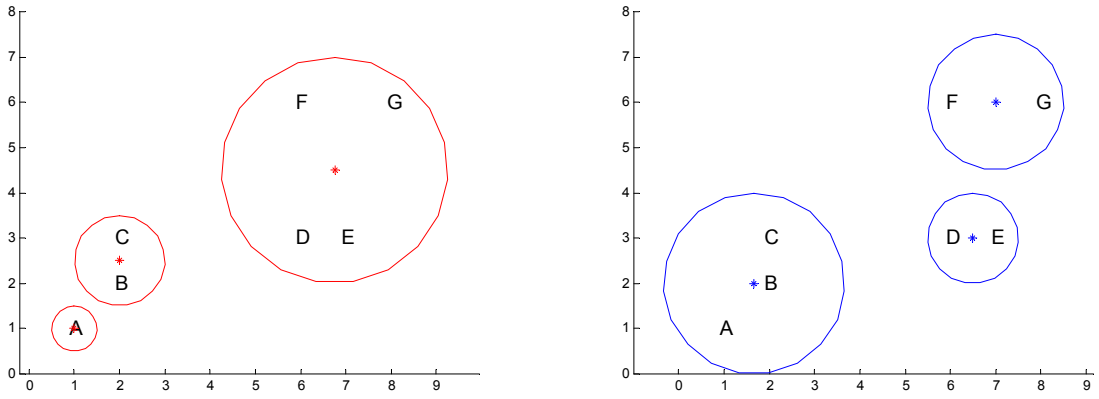
onde $\mathbf{v}_i^{(j)}$ é o i-ésimo objeto pertencente ao j-ésimo agrupamento, o qual tem $\mathbf{c}_j$ como seu centróide. Repare que cada objeto pertence ao agrupamento cujo centróide está mais próximo de si, sendo que n_j é o número de objetos do j-ésimo agrupamento. O centróide do j-ésimo agrupamento vai ser o vetor médio dos n_j objetos que pertencem ao j-ésimo agrupamento em um dado instante:

$$\mathbf{c}_j = \bar{\mathbf{v}}^{(j)} = \frac{1}{n_j} \sum_{i=1}^{n_j} \mathbf{v}_i^{(j)} \qquad \textbf{Equação 2-2}$$

O vetor que representa o centróide é mais conhecido como *protótipo* e o processo geral executado pelo *k-means* é chamado de *quantização vetorial* (Kohonen, 1997, pg. 48; Van Hulle, 2000, pg. 43).

O *k-means* recebe como entrada um número K de agrupamentos e atribui aleatoriamente um objeto como sendo o centróide inicial de cada agrupamento. Sucessivamente, cada objeto é associado ao agrupamento mais próximo e o centróide de cada agrupamento é então recalculado levando-se em conta o novo conjunto de objetos a ele pertencentes.

Repare que, com isso, os centróides não mais se restringem a serem um subconjunto de objetos, pois podem estar localizados onde não há nenhum objeto. O algoritmo pára quando, tipicamente, há poucas trocas de objetos entre grupos ou quando um valor estipulado como erro mínimo é atingido. Opcionalmente, após uma estabilização, grupos podem ser fundidos ou então divididos segundo critérios estabelecidos, quando então o processo de associação dos objetos aos novos grupos reinicia. Além da escolha do número K de centróides, um dos principais problemas do *k-means* é justamente a escolha inicial dos centróides, como mostra a Figura 2-7. Nesta figura representa-se um conjunto $\mathbf{V} = \{A, B, C, D, E, F, G\}$ de objetos num plano bidimensional e aplica-se o algoritmo *k-means* com $K = 3$ agrupamentos. Se os centróides destes forem tomados inicialmente pelos padrões $\{A, B, C\}$, será obtido o resultado ilustrado à esquerda, bastante inconveniente se comparado ao erro total obtido na ilustração à direita, gerada tomando-se os padrões $\{A, D, F\}$ como centróides iniciais.



Erro total = 12,250 **Erro total = 5,167**

Figura 2-7 – O *k-means* é sensível à posição inicial dos centróides: à esquerda vê-se um agrupamento indevido se comparado ao obtido na figura à direita, o que pode ser verificado pelo erro total obtido. Adaptado de Jain *et al.* (1999)

Outro algoritmo bastante utilizado é o que define uma *árvore geradora mínima*, do inglês *Minimum Spanning Tree - MST* (Gower & Ross, 1969). A essência deste algoritmo é gerar um grafo conectando os objetos de modo que: (a) não haja ciclos; (b) todo objeto seja conectado por pelo menos um arco; e (c) não haja subgrafos. A Figura 2-8 ilustra o algoritmo de caminho mínimo aplicado ao mesmo exemplo anterior. Os agrupamentos são obtidos seccionando-se primeiro o arco de maior comprimento (o que gera 2 grupos) e assim sucessivamente.

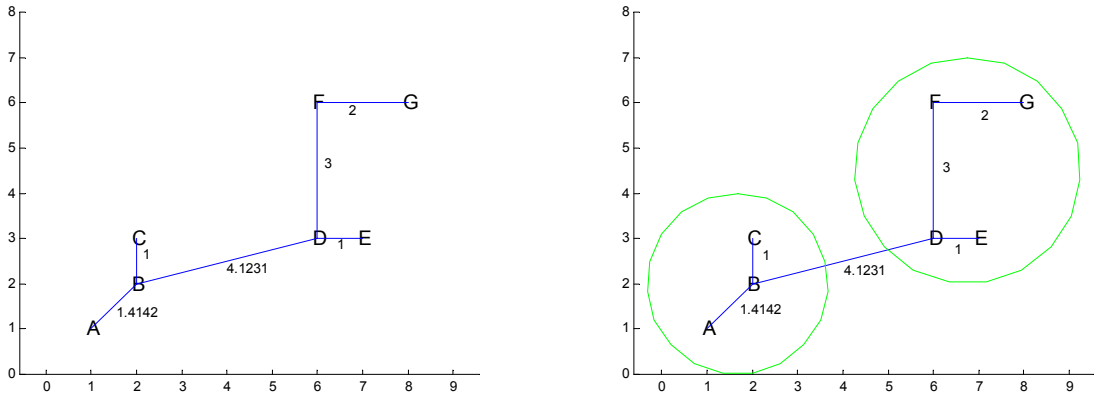


Figura 2-8 – O algoritmo do caminho mínimo (esquerda) pode gerar agrupamentos seccionando-se o arco mais longo (direita). Outros grupos podem ser gerados seguindo o mesmo raciocínio.

É interessante notar que os agrupamentos obtidos pelo método de ligação simples são também subgrafos obtidos pelo método do caminho mínimo (Jain *et al.* 1999).

2.4 Métodos de Projeção

Os métodos de projeção (Jain & Dubes, 1988; Svensén, 1999; Kaski, 1997) procuram *mapear* objetos no espaço de entrada $\Re^D$ para um hiperplano no espaço $\Re^P$, sendo que normalmente se tem $P \leq D$. O objetivo destes métodos aplicados à mineração de dados é exibir a estrutura do espaço original o mais fielmente possível no hiperplano de projeção, possibilitando assim uma análise de agrupamentos que pode ser realizada visualmente caso $P = 2$ ou $P = 3$. Esta análise pode servir também para validar resultados obtidos por outros métodos de mineração de dados, ou ainda fornecer "pistas" quando do uso de ferramentas interativas. A idéia aproximada de um método de projeção é representada na Figura 2-9 O mapeamento em si é uma transformação, linear ou não, capaz de levar N pontos $\mathbf{v} = [v_1, ..., v_D]^T$ do espaço $\Re^D$ para o hiperplano no espaço $\Re^P$, e normalmente $P \leq D$.

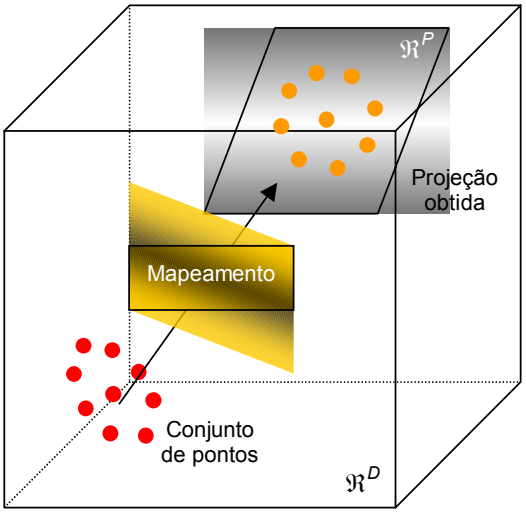


Figura 2-9 – O conjunto de pontos no espaço $\Re^D$ de entrada é mapeado para um hiperplano no espaço $\Re^P$, onde normalmente $P \leq D$.

2.4.1 Operadores Lineares

No caso de projeções lineares o mapeamento é uma transformação linear do espaço de entrada, representada vetorialmente pela forma geral

$$\mathbf{y}_i = \mathbf{A}\mathbf{v}_i \ , i = 1, ..., N. \qquad \textbf{Equação 2-3}$$

$\mathbf{A}$ é uma matriz $P \times D$ que gera os vetores $\mathbf{y} = [y_1, ..., y_P]^T \in \Re^P$ como uma combinação linear de suas colunas $\mathbf{a}_j \in \Re^P$, como segue:

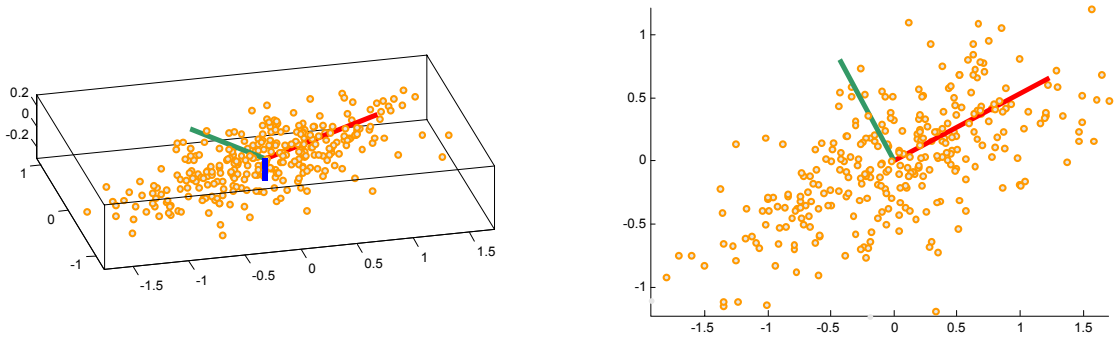
$$y_j = \sum_{j=1}^{D} \mathbf{a}_j v_j \qquad \textbf{Equação 2-4}$$

A escolha das colunas da matriz $\mathbf{A}$ permite que diferentes tipos de projeção sejam obtidos, dos quais a *análise por componentes principais* (PCA: *Principal Component Analysis*) é uma das mais populares (Jain & Dubes, 1988).

2.4.1.1 Análise de Componentes Principais (PCA)

O método PCA (Jollife, 1986; Jain & Dubes, 1988; resumo em Svensén, 1999) toma um conjunto $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_N\}$ de vetores $\mathbf{v}_n = [v_{n1}, ..., v_{nD}]^T \in \Re^D$, $n = 1 , ..., N$, numa dada base ortonormal e encontra uma nova base ortonormal $\{\mathbf{u}_1, ..., \mathbf{u}_D\}$ capaz de gerar o espaço original. Esta nova base é rotacionada, de forma que o primeiro eixo coincida com a direção na qual os dados possuam a maior variância; o segundo eixo, ortogonal ao primeiro,

orienta-se na direção da segunda maior variância e assim, sucessivamente. Cada eixo $\mathbf{u}_i$ representa uma das variâncias do conjunto com os dados projetados sobre si e a nova base $\{\mathbf{u}_1, ..., \mathbf{u}_D\}$, ordenada segundo as variâncias (a maior variância corresponde ao primeiro eixo), é o conjunto dos *componentes principais* (Svensén, 1999). Este método também é chamado de *"projeção por autovetores"* ou *"transformação de Karhunen-Loeve"* (Jain & Dubes, 1988).



(A) Projeção em 3D **(B) Projeção em 2D**

Figura 2-10 – (A) Conjunto de 300 pontos em $\Re^3$ gerados aleatoriamente numa correlação de gaussianas (σ = 0,8, 0,3 e 0,1 respectivamente para os eixos X, Y e Z) rotacionados em 30°. Os eixos vermelho, verde e azul projetados correspondem respectivamente ao 1º, 2º e 3º componentes principais (redimensionados pelo desvio padrão do conjunto). Em (B), o mesmo conjunto observado numa projeção em 2D, com os eixos vermelho e verde representando o 1º e 2º componentes principais.

Ao optar-se por uma projeção dos dados utilizando os P primeiros componentes principais, $P < D$, obtém-se uma representação do conjunto original em um espaço de menor dimensão, o que é conhecido por *redução dimensional*. A Figura 2-11 apresenta alguns exemplos e mostra, nas Figuras (C) e (D), que a maior restrição das projeções lineares é exatamente sua linearidade.

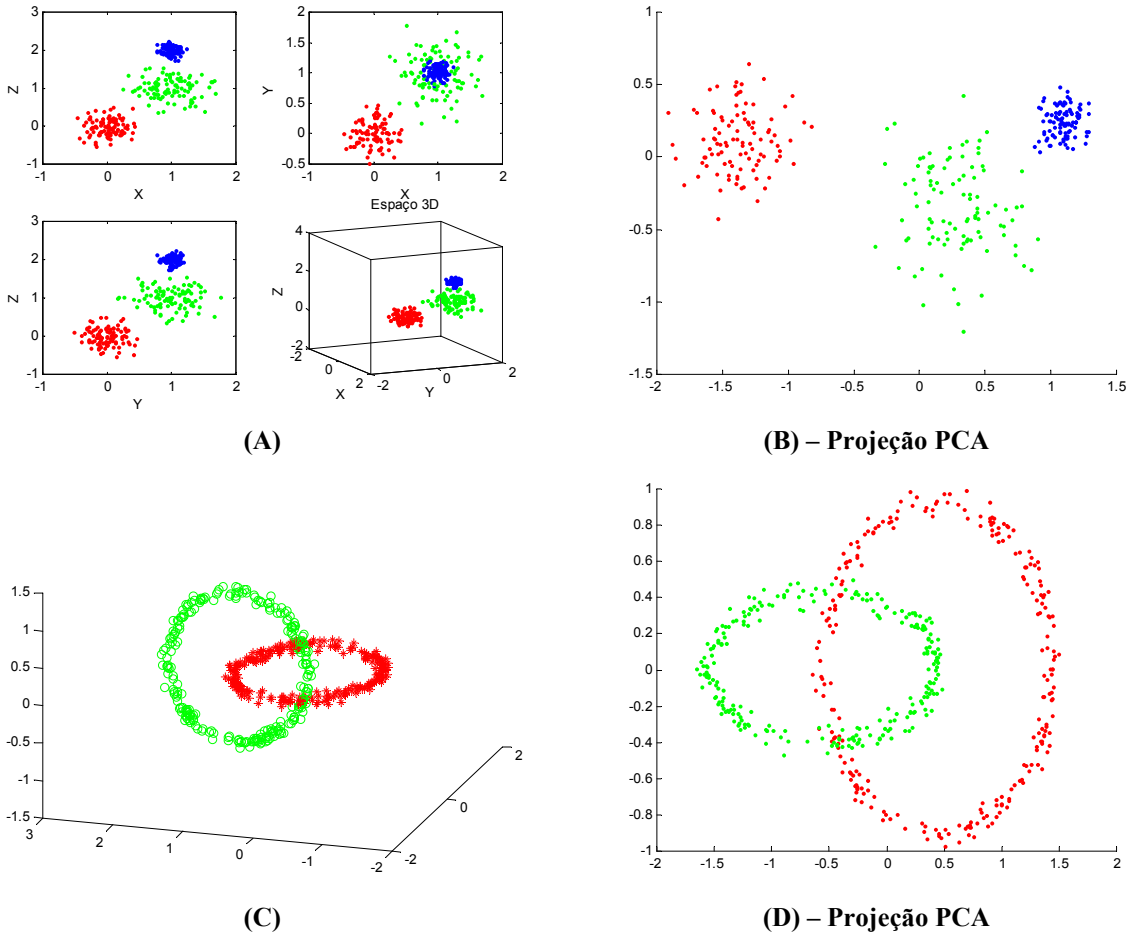


Figura 2-11 – (A) mostra um conjunto com 3 agrupamentos distintos no espaço $\Re^3$ (cada grupo possui 100 pontos gerados aleatoriamente por gaussianas com desvio padrão 0,1 (azul), 0,2 (vermelho) e 0,3 (verde)) projetados em 2 componentes principais (B). Em (C) o *Chainlink Dataset* (proposto por Ultsch & Vetter (1994) e neste exemplo com 250 pontos por tórus) onde os conjuntos não são linearmente separáveis. Em (D), o PCA com 2 componentes principais não é capaz de separar os agrupamentos de (C).

Outros métodos existentes incluem a *"análise discriminante"* (Jain & Dubes, 1988) e *"busca de projeção"*, do inglês *projection pursuit* (Huber, 1985; Kaski, 1997). O primeiro busca uma projeção que tenta maximizar a dispersão entre grupos ao mesmo tempo que tenta manter a coesão interna constante. O segundo busca revelar o máximo possível de não linearidade associando a cada projeção um índice de "interesse" que deve ser maximizado. Há também abordagens baseadas em redes neurais para a análise de componentes principais que podem ser consultadas em Haykin (1999).

2.4.2 Operadores Não Lineares

Quando os dados residem em hiperplanos curvos dentro do espaço de dados, métodos lineares mostram-se pouco eficientes em capturar tais estruturas. Nestes casos, pode-se lançar mão de métodos não lineares. A maioria destes métodos tenta representar os atributos não lineares através da maximização de uma função definida sobre um conjunto de variáveis que é dependente do conjunto de dados, isto é, não possuem uma função de mapeamento explícita (Jain & Dubes, 1988). Este tipo de projeção não é, portanto, extensível a novos dados que sejam obtidos após a computação do mapeamento, pois este é dependente do conjunto como um todo, devendo ser recalculado a toda e qualquer alteração do conjunto de dados. Algumas heurísticas podem ser usadas para acelerar o cálculo dos mapeamentos não lineares, computacionalmente caros (Jain & Dubes, 1988), como por exemplo usar o primeiro componente principal como configuração inicial para o algoritmo de projeção não linear.

2.4.2.1 Escalonamento Multidimensional (MDS)

Escalonamento multidimensional (Jain & Dubes, 1988), do inglês *Multidimensional Scaling* (MDS), é um nome genérico dado a um conjunto de técnicas bastante utilizadas principalmente em ciências sociais e econômicas para analisar similaridade entre objetos. O conjunto de N objetos é representado por um conjunto de N pontos preservando ao máximo as relações de similaridade entre todos os possíveis pares de objetos, ou seja: ao invés de operar diretamente no espaço original, uma configuração de pontos num espaço de menor dimensão é gerada, de forma que as relações interobjetos no espaço original sejam mantidas ao máximo no novo espaço gerado. Nesta dissertação, considera-se MDS em 2 e 3 dimensões, embora matematicamente seja possível operar em qualquer número de dimensões. Também não há a necessidade da relação de similaridade ser necessariamente uma norma, de modo a preservar a relação triangular $d(x,z) \leq d(x,y) + d(y,z)$, para três objetos x, y e z. De fato, métodos MDS podem ser aplicados a virtualmente qualquer tipo de relação que expresse a similaridade/dissimilaridade em valores numéricos, configurando assim dois grandes conjuntos de métodos MDS, *métricos* e *não métricos* (Kaski, 1997; Jain & Dubes, 1988).

Para melhor explicar o funcionamento do MDS, considere-se o conjunto $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_N\}$, $\mathbf{V} \subseteq \Re^D$ de vetores $\mathbf{v}_n = [v_{n1}, ..., v_{nD}]^T \in \Re^D$, $n = 1, ..., N$, cada vetor $\mathbf{v}_n$ representando um objeto (um ponto) no espaço D-dimensional através de seus D atributos (veja Tabela 2-1). A dissimilaridade entre todos os possíveis pares de objetos é dada por uma matriz de dissimilaridade **D** de elementos d_{ij} $(i,j = 1, ..., N)$ onde cada elemento d_{ij} corresponde à norma $d(i,j) = \left\|\mathbf{v}_i - \mathbf{v}_j\right\|$, aqui considerada como sendo a distância euclidiana. O MDS busca então uma configuração de pontos num espaço de dimensão 2 ou 3, de forma que a matriz de dissimilaridade **D'** dos pontos projetados represente, o mais fielmente possível, as relações do espaço original. A função de erro que mede esta relação é chamada de *stress* (Kruskal & Wish, 1978; Jain & Dubes, 1988) e pode ser expressa simplificadamente por

$$stress = \sum_{i<j}^{N} \left[d(i,j) - d'(i,j)\right]^2 \qquad \textbf{Equação 2-5}$$

Os algoritmos MDS procuram reposicionar os pontos no espaço gerado de forma a minimizar o *stress* de acordo com o algoritmo simplificado abaixo:

1- associe a cada ponto no espaço de saída coordenadas arbitrárias;
2- calcule a distância (euclidiana) sobre todos os pares de pontos projetados (matriz **D'**) e os pontos originais (matriz **D**);
3- calcule o *stress* (isto é, compare a matriz **D'** com a matriz **D** usando, por exemplo, a Equação 2-5): quanto menor o valor do *stress*, maior a similaridade entre os dois conjuntos;
4- reposicione os pontos no espaço de saída de forma a minimizar o *stress*;
5- repita de 2 a 4 até que (a) o *stress* fique abaixo de um limite mínimo; (b) não se reduza sensivelmente; ou (c) até alcançar um número fixo de iterações.

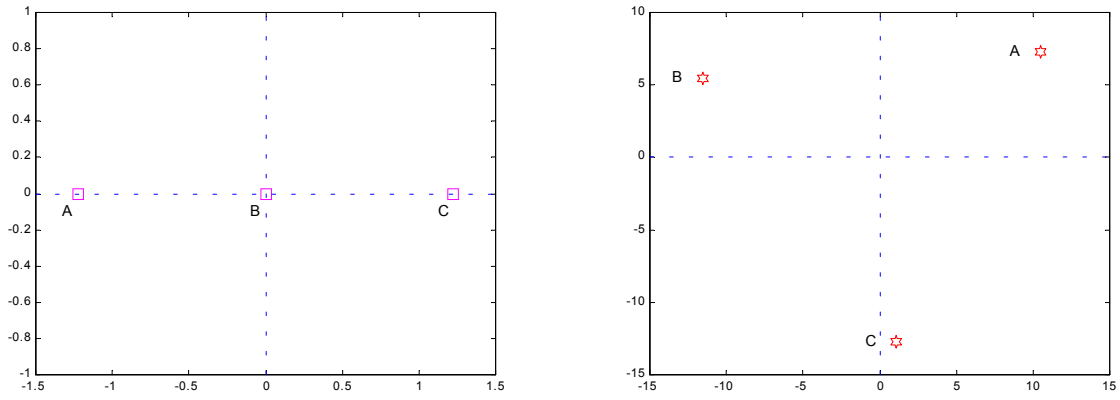
Os exemplos a seguir permitem estudar o comportamento do algoritmo MDS representando 3 cidades fictícias, gerados através do programa KYST2A (Kruskal *et al.* 1993). Se a matriz de distâncias em Km for dada por

	A	*B*	*C*
A	0	30	60
B	30	0	30
C	60	30	0

então podemos deduzir que é possível representá-las em apenas uma dimensão, dado que B é eqüidistante de A e C em 30 Km e a distância entre A e C é de 60 Km (ou seja, a distância AB + BC). Este resultado é demonstrado na Figura 2-12-A. Entretanto, se considerarmos as 3 cidades eqüidistantes entre si, com uma matriz de distâncias dada por

	A	*B*	*C*
A	0	30	30
B	30	0	30
C	30	30	0

então já não será mais possível representar os pontos numa única dimensão, mas com duas obtemos uma configuração triangular capaz de minimizar o *stress*, como demonstrado na Figura 2-12-B.



(A) **(B)**

Figura 2-12 – Resultados de MDS para 3 cidades fictícias A, B e C. Na figura esquerda as distâncias podem ser representadas com uma dimensão, mas no caso de cidades eqüidistantes é necessário um plano para representá-las.

No exemplo seguinte as distâncias rodoviárias entre as principais capitais brasileiras foram manipuladas pelo mesmo algoritmo, resultando nas projeções em 2 e 3 dimensões conforme Figura 2-13.

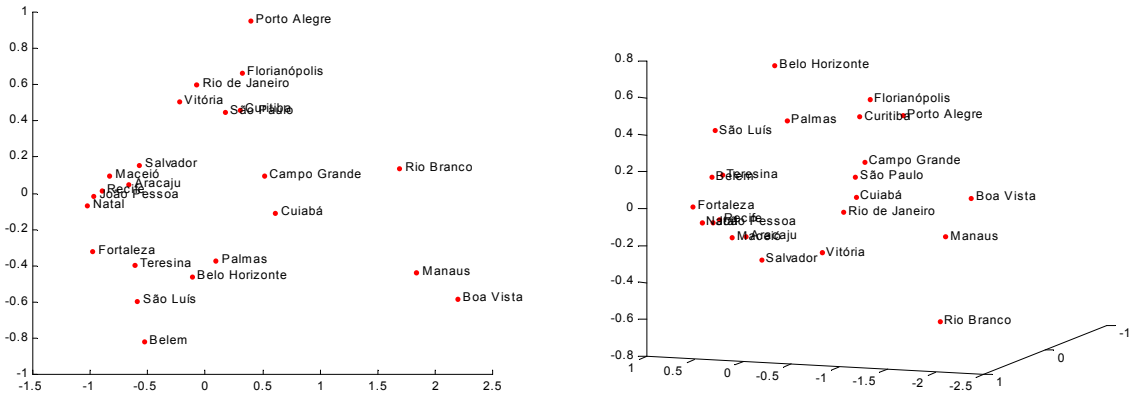


Figura 2-13 – As principais capitais brasileiras e suas distâncias rodoviárias, em Km, projetadas em 2 e 3 dimensões pelo algoritmo MDS.

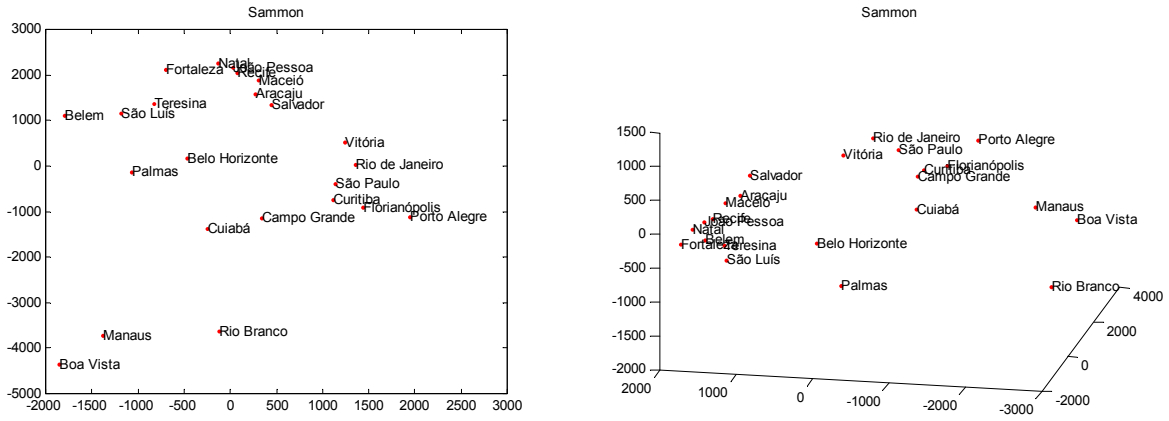
Deve-se notar que a orientação dos eixos e suas escalas são arbitrárias nos métodos MDS e, embora possam receber nomes, são totalmente subjetivos e dependem, assim, da avaliação do gráfico gerado. Para mais detalhes sobre a interpretação dos gráficos gerados por MDS, consulte Jain & Dubes (1988).

2.4.2.2 Projeção de Sammon

A projeção proposta por Sammon (1969) é um método não linear que guarda várias semelhanças aos métodos MDS: assim como este, o conjunto original de objetos é representado por um conjunto de pontos num espaço, normalmente, de menor dimensão. A avaliação da fidelidade da representação das similaridades é calculada por uma função que pode também ser chamada de *stress* (Jain & Dubes, 1988; Kaski, 1997) dada por:

$$stress_{Sammon} = \sum_{i<j}^{N} \frac{[d(i,j) - d'(i,j)]^2}{d(i,j)} \qquad \textbf{Equação 2-6}$$

Percebe-se que a única diferença para a Equação 2-5 é que o erro entre d e d' é agora normalizado pela distância do espaço original. Devido a isso, distâncias menores serão realçadas em relação ao MDS original, resultando num gráfico normalmente mais uniforme.



(A) **(B)**

Figura 2-14 – As principais capitais brasileiras e suas distâncias rodoviárias, em Km, projetadas em 2 e 3 dimensões pelo algoritmo de Sammon.

A projeção de Sammon é freqüentemente sugerida como um método de análise prévia da aplicação de mapas de Kohonen (veja Capítulo 3) que pode indicar tendências de agrupamentos. Assim como MDS, a reconfiguração dos pontos no espaço na tentativa de minimizar o *stress* não é trivial (Jain & Dubes, 1988), o que significa que são métodos computacionalmente caros para alcançar um estado de convergência quando aplicados a objetos de elevada dimensão.

2.4.2.3 Curvas Principais (PC)

A *análise por curvas principais* (Hastie & Stuetzle, 1989), do inglês *Principal Curves* (PC), pode ser vista como uma generalização não linear para o método PCA descrito na Seção 2.4.1.1. Enquanto este último executa uma projeção linear com eixos ortogonais orientados conforme a variância dos dados, o método PC busca a projeção através das *curvas principais*. A curva principal é uma curva unidimensional que, informalmente, passa pelo "centro" da nuvem de pontos que representam o conjunto de objetos estudados. Mais cuidadosamente, esta característica significa que cada ponto da curva é a média de todos os objetos que serão projetados sobre este (ou seja, dos objetos mais próximos deste ponto em particular na curva). Sendo o conjunto $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_N\}$ de vetores $\mathbf{v}_n = [v_{n1}, ..., v_{nD}]^T \in \Re^D$, $n = 1, ..., N$, a curva principal $\mathbf{f}(y)$ é uma curva paramétrica dada por:

$$\mathbf{f}(y) = E\left[\mathbf{t} \middle| \lambda_{\mathbf{f}}(\mathbf{t}) = y\right] \qquad \textbf{Equação 2-7}$$

onde **t** é uma variável randômica no espaço $\Re^D$ e $\lambda_{\mathbf{f}}(\mathbf{t})$ é a projeção de **t** sobre a curva definida por **f**(·). Esta definição é chamada de *autoconsistência* e significa que para qualquer ponto *y*, **f**(*y*) será a média das projeções sobre *y* dadas pela projeção $\lambda_{\mathbf{f}}(\cdot)$, que neste modelo é uma projeção ortogonal (Hastie & Stuetzle, 1989; Svensén, 1998).

Para aplicação em conjuntos finitos Hastie & Stuetzle (1989) propõem uma aproximação linear que pode ser observada na Figura 2-15, que representa uma nuvem de pontos cuja geratriz é uma curva onde foi aplicado ruído gaussiano.

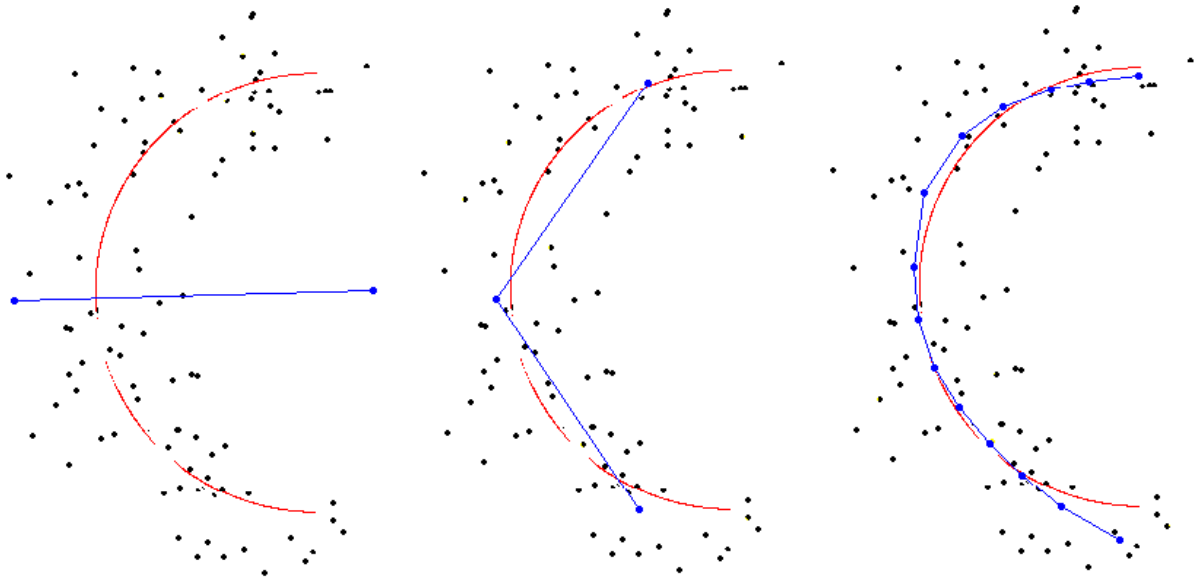


Figura 2-15 – Uma nuvem de pontos cuja geratriz é representada pela curva vermelha, a qual é aproximada por uma curva principal (azul). De (A) para (C) vê-se a situação inicial e a configuração final. Figuras geradas em http://www.cs.concordia.ca/~grad/kegl/research/pcurves (acessada em 30/03/2001).

Hastie & Stuetzle (1989) propõem ainda uma generalização das curvas principais para as *superfícies principais*, o que permitiria que o espaço de projeção fosse bidimensional.

2.4.2.4 Análise por Componentes Curvos (CCA)

A *análise por componentes curvos* (Demartines & Hérault, 1997), do inglês *Curvilinear Component Analysis* (CCA), é um método de projeção não linear que utiliza como dados de entrada não os objetos do espaço de dados diretamente mas um *conjunto de protótipos* obtidos através de quantização vetorial do espaço de entrada. O processo é composto por duas etapas: (1) inicialmente, o conjunto $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_N\}$, $\mathbf{V} \subseteq \Re^D$, de vetores

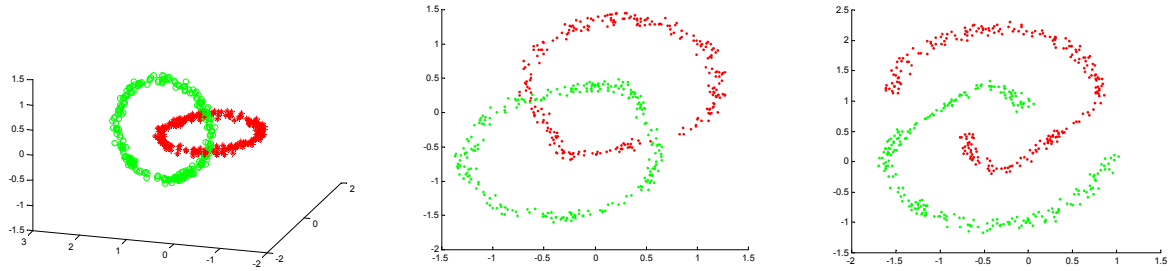
$\mathbf{v}_n = [v_{n1}, ..., v_{nD}]^T \in \Re^D$, $n = 1, ..., N$, representando os objetos de entrada, são aproximados por um conjunto de protótipos; (2) em seguida, o conjunto de protótipos é projetado no espaço de saída, $\Re^L$. A função de *stress* para o método é dada por:

$$stress_{CCA} = \sum_{i<j}^{N} \left[d(i,j) - d'(i,j) \right]^2 F(d'(i,j), \lambda) \qquad \textbf{Equação 2-8}$$

A comparação do *stress* do algoritmo CCA com o de Sammon (Equação 2-6) permite observar que a mudança essencial reside na forma como o CCA pondera a dissimilaridade entre os pontos originais $d(i,j)$ (os protótipos) e suas projeções $d'(i,j)$, que é realizada através de uma função F dependente das projeções e de um parâmetro λ. Este método traz duas vantagens sobre o método de Sammon: (1) o uso de protótipos como dados de entrada reduz a carga computacional do cálculo das distâncias; e (2) a função F é normalmente monotônica decrescente se a intenção é ressaltar a topologia local, o que possibilita um maior controle sobre o processo. O parâmetro λ pode ser uma função decrescente com o tempo (como a função de vizinhança do SOM) ou pode ser controlada iterativamente pelo usuário (Demartines & Hérault, 1997).

É interessante ressaltar que embora o método CCA seja algumas ordens de grandeza mais rápido que o de Sammon, o uso de protótipos como dados de entrada exige uma interpolação se for necessária a projeção de algum dado posterior ao treinamento. Por outro lado, isto pode ser visto como uma vantagem, pois não há necessidade de recalcular as projeções (como é o caso de Sammon) quando ocorre a projeção posterior ao treinamento (Demartines & Hérault, 1997).

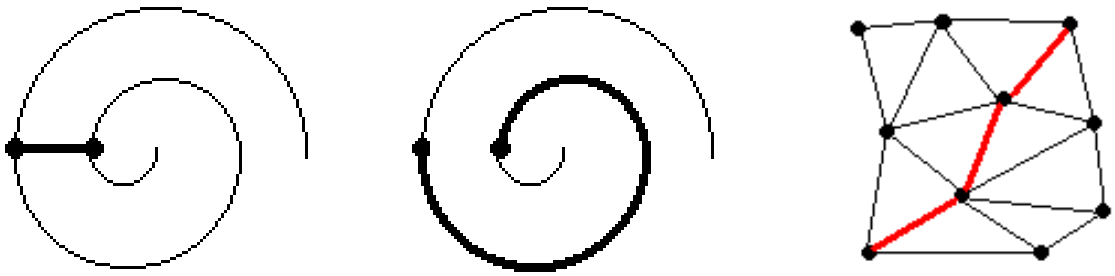
A Figura 2-16 apresenta uma comparação entre os métodos de Sammon e CCA utilizando-se um conjunto não linear, o *Chainlink Dataset*. O exemplo demonstra a capacidade que o CCA exibe de separar os grupos de forma mais eficiente que o de Sammon.



(A) (B) (C)

Figura 2-16 – O *Chainlink Dataset* representado em (A), sua projeção pelo método de Sammon (B) e uma projeção CCA em (C), que obtém melhor discriminação dos agrupamentos devido a sua não linearidade.

Uma possível melhoria para o método CCA é proposta por Lee *et al.* (2000), o CDA (*Curvilinear Distance Analysis*). Neste método, a função de distância $d(i,j)$ é substituída por uma função $\delta(i,j)$ chamada pelos autores de *"distância curvilínea"*.



(A) (B) (C)

Figura 2-17 – O conceito de *"distância curvilínea"*: em (A) têm-se a proposta original do algoritmo CCA utilizando a distância euclidiana e em (B) vê-se a distância curvilínea ideal entre os dois pontos. Em (C) têm-se a aproximação calculada na prática sobre os vetores de protótipos.

Na prática, a distância curvilínea é aproximada pelo cálculo do menor caminho entre dois pontos num grafo que conecta os vetores de protótipos, como pode ser visto na Figura 2-17-C. De acordo com Lee *et al.* (2000), o algoritmo CDA é mais eficiente que CCA e até mesmo que o SOM, na análise de conjuntos fortemente curvos.

2.5 Métodos Gerativos

Os *métodos gerativos* assumem a hipótese de que os objetos, no *D*-espaço, foram gerados por uma função densidade de probabilidade inserida neste espaço e que possui, por sua vez, a estrutura inerente do conjunto de objetos. O objetivo dos métodos gerativos é, portanto,

representar tal função hipotética através de um modelo sujeito a um conjunto de parâmetros que são ajustados no sentido de aproximar o modelo da função hipotética. Cada método em si possui uma "ótica" própria, ou seja, procura "enxergar" o conjunto de objetos de uma forma que é intrínseca ao método e que, obviamente, afeta a forma final do modelo.

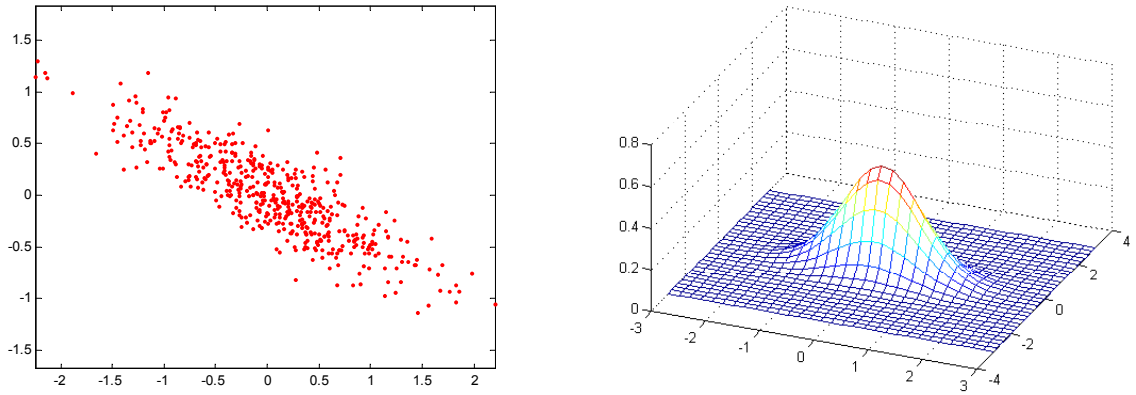
Normalmente procura-se limitar o conjunto de modelos possíveis àqueles com dimensão menor que o conjunto original, realizando assim a redução dimensional (veja a Seção 2.1 para uma definição do termo) do conjunto de entrada.

2.5.1 Mistura de densidades

Os métodos baseados em *mistura de densidades* (Titterington *et al.* 1985) baseiam-se em múltiplas funções densidade de probabilidade combinadas entre si de forma a tentar representar a função hipotética geradora do espaço. Estas múltiplas funções são chamadas *componentes* da mistura e normalmente são caracterizadas por um conjunto de parâmetros desconhecidos a princípio mas que devem ser estimados pelo método.

Embora os componentes do modelo possam ter formas paramétricas distintas, a maioria dos trabalhos nesta área supõem que tais componentes são gaussianas e não raramente sujeitas ao mesmo conjunto de parâmetros (Jain *et al.*, 1999). Esta hipótese deve-se em grande parte à complexidade computacional envolvida na estimação destes valores, o que em abordagens tradicionais é um processo iterativo que utiliza como estimador a máxima verossimilhança (*maximum likelihood*) dos vetores de parâmetros das componentes da mistura (Jain & Dubes, 1988). Papoulis (1991) discute o uso da função de verossimilhança como estimador paramétrico.

O uso de componentes com a mesma forma paramétrica, entretanto, é uma desvantagem, pois acaba impondo ao modelo uma "ótica" particular. Segundo Costa (1999) o caso mais simples (onde as componentes são gaussianas sujeitas ao mesmo conjunto de parâmetros) pressupõe no espaço uma geometria hiperesférica que pode não condizer com a real distribuição dos dados.



(A) **(B)**

Figura 2-18 – Um conjunto de 500 pontos gerados por um processo de duas gaussianas com desvio padrão σ = 0,8 e 0,2 respectivamente nos eixos principal e secundário (A). Em (B) temos o modelo de geração recuperado pelo método de mistura de densidades.

De acordo com Jain *et al.* (1999), o algoritmo **EM** (Dempster *et al.* 1977), do inglês *Expectation-Maximization* (veja a Seção 4.1.2 para mais detalhes), é aplicado como estimador de modelos baseados em mistura de densidades em trabalhos mais recentes.

2.5.2 Análise de Fatores (FA)

A técnica de *análise de fatores* (Bartholomew, 1987), do inglês *Factor Analysis* (FA), é tida como a versão geradora da análise por componentes principais (veja a Seção 2.4.1.1 para mais detalhes) e sua diferença essencial é que FA utiliza-se da *covariância* entre as variáveis enquanto PCA opera sobre a variância. A covariância entre duas variáveis observadas mede o grau com que ambas estão relacionadas entre si, indicando que variáveis que possuem grande covariância entre si podem ser uma função de uma mesma variável latente (veja a Seção 4.1.1 para uma discussão sobre modelos baseados em variáveis latentes), o que explica a variação em comum medida pela covariância. O efeito prático desta diferença, e também uma das principais vantagens de FA se comparada a PCA, é ser mais imune ao ruído que pode estar presente junto ao conjunto de dados, particularmente junto a uma determinada dimensão do mesmo. A explicação é que se duas variáveis possuem alta covariância, então o ruído entre elas também é comum e é praticamente ignorado pelo método FA.

Quando de sua criação no início do século, a técnica FA não obteve grande aceitação devido a suas origens (desenvolvida por psicólogos para testar resultados em testes cognitivos) e a uma ausência de fundamentos matemáticos sólidos. Mais recentemente, com a melhor fundamentação teórica e com o apoio do algoritmo EM como estimador dos parâmetros é que a técnica tem encontrado melhor aceitação como ferramenta estatística (Svensén, 1998).

2.6 Considerações finais

Este capítulo buscou fazer um estudo geral dos principais métodos que podem ser utilizados como ferramentas na tarefa de mineração de dados. Embora várias propostas tenham sido discutidas, não houve a intenção deste capítulo ser uma referência completa para tais métodos, mas apenas a de introduzir os principais conceitos envolvidos junto aos diversos métodos. Com o intuito de indicar algumas abordagens mais recentes e com variações em relação aos métodos citados, inclui-se aqui alguns comentários adicionais. A literatura clássica em métodos para mineração de dados não pode deixar de incluir Bishop (1995) e Duda *et al.* (2000), obras com tratamento aprofundado da grande maioria dos métodos citados neste capítulo.

A teoria dos conjuntos nebulosos (Zadeh, 1965) oferece subsídios para a extensão de métodos de agrupamento, gerando métodos cujas regras de pertinência são relaxadas, como por exemplo o *fuzzy k-means* (Bezdek, 1981; resumo em Costa, 1999). Jain *et al.* (1999) fazem uma boa revisão de métodos de agrupamento em geral, incluindo aqueles baseados em algoritmos evolutivos (ou genéticos).

Tibshirani (1992) oferece uma nova definição para as curvas principais, baseada em um modelo gerativo otimizado pelo algoritmo EM que resolve uma tendência do algoritmo original em gerar curvas não coincidentes com a geratriz do espaço quando esta última é curva. Como já mencionado no texto, Lee *et al.* (2000) propõem um método denominado CDA (*Curvilinear Distance Analysis*) capaz de melhorar o desempenho do método CCA tornando-o mais robusto e eficiente na análise de conjuntos fortemente curvos.

Mao & Jain (1995) apresentam diversas implementações de métodos de projeção (como por exemplo o PCA) empregando redes neurais artificiais. Também introduzem uma rede neural que estende a projeção de Sammon, possibilitando a projeção de novos dados sem a necessidade de recalcular todo o contexto, uma característica dos métodos de projeção não lineares (veja a Seção 2.4.2). O método PCA também possui uma extensão geradora otimizada pelo algoritmo EM proposta por Tipping & Bishop (1997). MacKay & Gibbs (1997) propõem uma extensão para as redes neurais multicamadas chamada de *rede de densidade* que, em princípio, é capaz de aproximar a função densidade de probabilidade de um conjunto de dados através de aprendizado não supervisionado (resumo em Svensén, 1998).

Ultsch (1999a,b) propõe uma interessante abordagem baseada em processos auto-organizáveis para revelar a estrutura de conjuntos de dados chamada *"emergência"*. Segundo o mesmo, "emergência" é a capacidade que um sistema exibe de produzir um fenômeno novo, de mais alto nível, possível somente pela cooperação de processos elementares dentro do mesmo sistema. Assim, Ultsch sugere a utilização de *DataBots*, seres artificiais que vivem num universo artificial (o *UD-Universe*) e que, em grande número e por cooperação, são capazes de exibir padrões de comportamento que correspondem às estruturas de um espaço de dados. Esta abordagem é evidentemente baseada em agentes, embora não citada explicitamente pelo autor.

Finalmente, Anand & Hughes (1998) defendem a utilização de métodos híbridos de mineração de dados como uma forma de sobrepujar as limitações que todo método isolado, invariavelmente, exibe.

Capítulo 3

Mapas Auto-Organizáveis

O *Mapa Auto-Organizável de Kohonen* (SOM) (Kohonen, 1982a, 1997) é um tipo de rede neural artificial baseada em aprendizado competitivo e não supervisionado, sendo capaz de mapear um conjunto de dados, de um espaço de entrada contido em $\Re^D$, em um conjunto finito de neurônios organizados em um arranjo normalmente unidimensional ou bidimensional. As relações de similaridade entre os neurônios (e, por extensão, entre os dados) podem ser observadas através das relações estabelecidas entre os vetores de pesos dos neurônios, os quais também estão contidos em $\Re^D$.

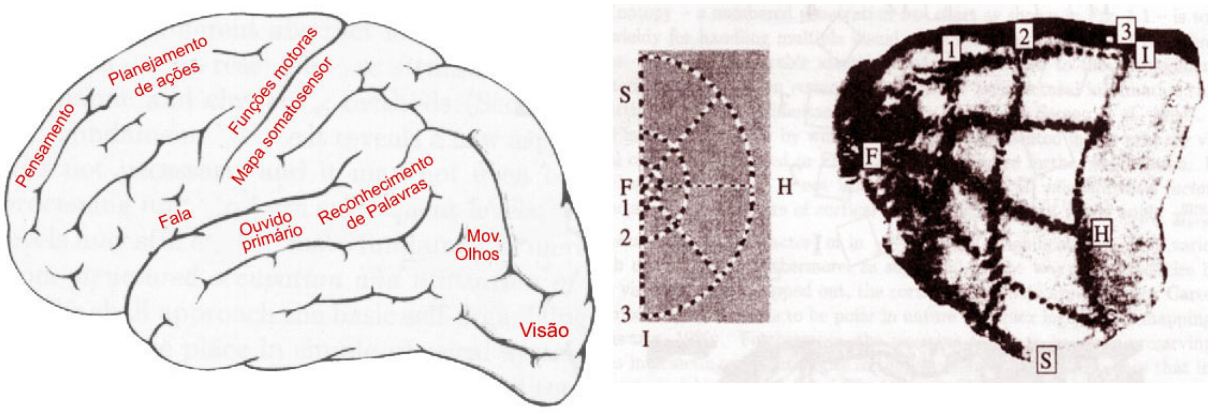
Desse ponto de vista, o SOM realiza uma projeção não linear do espaço de dados de entrada, em $\Re^D$, para o espaço de dados do arranjo, em $\Re^P$, executando uma redução dimensional quando $P < D$. Como o arranjo é normalmente unidimensional ou bidimensional, então resulta $P = 1$ ou $P = 2$. Ao realizar esta projeção não linear, o algoritmo tenta preservar ao máximo a topologia do espaço original, ou seja, procura fazer com que neurônios vizinhos no arranjo apresentem vetores de pesos que retratem as relações de vizinhança entre os dados. Para tanto, os neurônios competem para representar cada dado, e o neurônio vencedor tem seu vetor de pesos ajustados na direção do dado. Esta redução de dimensionalidade com preservação topológica permite ampliar a capacidade de análise de agrupamentos dos dados pertencentes a espaços de elevada dimensão.

O SOM é certamente um dos principais modelos de redes neurais artificiais na atualidade e é utilizado numa diversidade de aplicações que dificilmente seriam contidas numa simples obra. Há milhares de publicações sobre o SOM introduzido por Kohonen (1981a,b,c) e sua referência mais completa talvez seja Kohonen (1997).

Este capítulo não tem a pretensão de ser completo ou exaustivo e visa descrever brevemente o conceito e as principais características do SOM, bem como algumas de suas limitações. São também descritas algumas variantes do modelo original, a análise de agrupamentos através da matriz-U e alguns métodos propostos para comparação entre mapas gerados sobre um mesmo conjunto de dados. As demonstrações neste capítulo utilizaram a SOM Toolbox para Matlab® (Alhoniemi *et al.* 2000) e o SOM_PAK (Kohonen *et al.* 1995a).

3.1 Modelo formal

Evidências biológicas têm mostrado que as células do córtex cerebral dos mamíferos organizam-se de forma altamente estruturada em suas funções, resultando em regiões do cérebro especificamente capacitadas no processamento sensorial de sinais como visão, audição, controle motor, linguagem etc. (Van Hulle, 2000; Kohonen, 1997). Isso significa que os neurônios tornam-se sensíveis a determinados estímulos em particular e a outros, não, especializando-se no "processamento" de um determinado sinal, o que pode ser explicado pela separação dos canais nervosos que ligam os órgãos sensoriais ao cérebro. Em particular, a ordem física dos sinais percebidos pelo tecido dos órgãos sensoriais é projetada no córtex cerebral primário em *ordem semelhante*, resultando num mapeamento que *preserva a ordem topológica* do sinal recebido, embora com algumas transformações (Van Hulle, 2000).



(A) **(B)**

Figura 3-1 – Em (A) uma representação das várias regiões corticais especializadas no cérebro humano. Em (B) uma imagem projetada sobre a retina de um macaco (esquerda) é mapeada sobre seu córtex cerebral primário, o qual mantém as relações topológicas da imagem embora com uma transformação. 'F' indica a região da fóvea, 'I' e 'S' são os campos visuais inferior e superior, respectivamente. (A) reproduzida de Kohonen (1997, pg 79). (B) reproduzida de Van Hulle (2000, pg 4).

Entretanto, analisando-se mais especificamente estas regiões especializadas, há evidências de uma organização um pouco mais abstrata e complexa, ainda não totalmente compreendida: suas células organizam-se e tornam-se sensíveis aos estímulos *de acordo com uma ordem topológica que especifica uma relação de similaridade entre os sinais de entrada*. Assim, os neurônios exibem uma *ordenação física* tal que estímulos *semelhantes* no espaço de dados são processados por neurônios *fisicamente próximos entre si* no córtex cerebral. Nota-se que não existe nenhum "movimento" de neurônios, apenas seus parâmetros são ajustados para que tal comportamento ocorra. Assim é, por exemplo, com o córtex auditivo: os neurônios desta região tornam-se sensíveis aos estímulos sonoros numa ordem topológica que *reflete a variação tonal do sinal sonoro*, fato simulado em Kohonen (1982a) e representado na Figura 3-2:

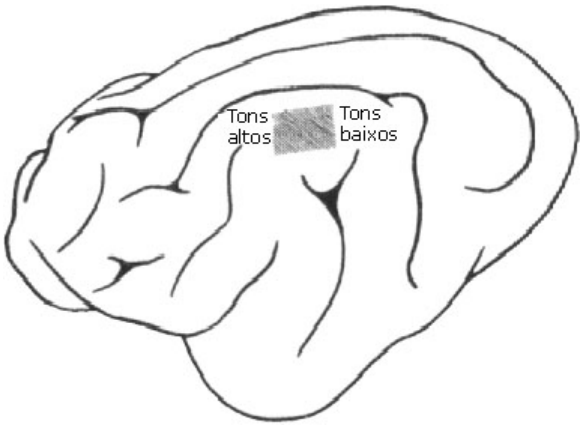


Figura 3-2 – Representação do córtex tonotópico de um gato onde os estímulos sonoros provocaram sensibilização no córtex conforme a altura das notas: a organização espacial dos neurônios representam a ordenação topológica dos sinais de entrada. No exemplo, pode-se notar que os tons de mais alta freqüência são representados pelos neurônios do lado esquerdo e sinais mais graves são sucessivamente representados pelos neurônios mais à direita. Adaptado de Kohonen (1997, pg 81).

A formação de mapeamentos topologicamente corretos é atribuída a uma diversidade de mecanismos, dos quais um em particular, a *auto-organização*[1], recebeu bastante atenção da comunidade acadêmica devido a suas fortes evidências biológicas. Isto levou à proposição de vários modelos de *mapas topográficos*, ou *mapas topologicamente corretos*. Duas variantes são pesquisadas: modelos baseados em *gradiente* e modelos baseados em *aprendizado competitivo* (Van Hulle, 2000). Esta última vertente, embora menos relacionada com os fundamentos biológicos, foi muito mais pesquisada e é nela que baseia-se o SOM. O aprendizado competitivo, sob a ótica de uma rede neural artificial, tem o sentido de quantização vetorial e pode ser sucintamente descrito desta forma:

- um conjunto de dados representados por vetores no espaço $\Re^D$ é apresentado, em ordem aleatória e de forma repetitiva, a uma rede composta por neurônios organizados segundo um arranjo específico, cada neurônio com o seu vetor de pesos no $\Re^D$;
- para cada dado apresentado à rede, haverá uma *competição* entre todos os neurônios pelo direito de representá-lo, de forma que o neurônio cujo vetor de pesos for o mais próximo do dado, segundo uma métrica previamente definida, *vence* a competição.

[1] Auto-organização refere-se aqui ao processo pelo qual estruturas com ordem global são obtidas através de interações locais entre os elementos.

Este neurônio é chamado BMU (*Best Matching Unit*) e este passo é chamado de *estágio competitivo* (Van Hulle, 2000, pg 16).

- o neurônio BMU é *adaptado*, isto é, seu vetor de pesos sinápticos é alterado no sentido de se aproximar ainda mais do dado apresentado, aumentando a probabilidade de que este mesmo neurônio volte a vencer numa subseqüente apresentação do mesmo dado. Para viabilizar o requisito de que neurônios próximos no arranjo vençam para dados próximos no $\Re^D$, neurônios pertencentes a uma vizinhança do neurônio vencedor, de acordo com a especificação do arranjo, também terão seu vetor de pesos ajustado na direção do dado, embora com menor intensidade. A primeira regra é conhecida como WTA (*Winner-Takes-All*) (Kaski & Kohonen, 1997), e o passo de ajuste da vizinhança é chamado de *estágio cooperativo* (Van Hulle, 2000, pg 16).

Fica evidente então que a idéia fundamental é a de que neurônios próximos entre si no arranjo representem dados próximos entre si no espaço de dados. Representar um dado aqui significa ter um vetor de pesos que seja mais próximo do dado que qualquer outro vetor de pesos da rede neural. Com isso, a topologia dos dados no espaço original acabará sendo preservada, dentro do possível, pelo arranjo de neurônios em um espaço de menor dimensão.

As relações de similaridade entre os neurônios podem ser visualmente observadas contanto que a dimensão do arranjo seja $1 \leq P \leq 3$. Embora não exista restrição teórica à utilização de arranjos de dimensão maior ou igual a 3, esta dissertação concentra-se no arranjo bidimensional com vizinhança hexagonal, considerada a mais adequada pela maioria dos autores quando o objetivo do SOM é a mineração de dados pela análise de agrupamentos (Kohonen, 1997; Kaski, 1997).

Seja o conjunto de entrada $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_N\}$, $\mathbf{V} \subseteq \Re^D$, de vetores $\mathbf{v}_n = [v_{n1}, ..., v_{nD}]^T \in \Re^D$, $n = 1, ..., N$, onde cada vetor $\mathbf{v}_n$ representa um dado (um ponto) no espaço D-dimensional, através de seus D atributos. O SOM é definido por um conjunto de neurônios i, $i = 1, ..., Q$, dispostos em um arranjo que define a vizinhança de cada neurônio, como pode ser visto na Figura 3-3 para as possibilidades mais utilizadas em $\Re^2$.

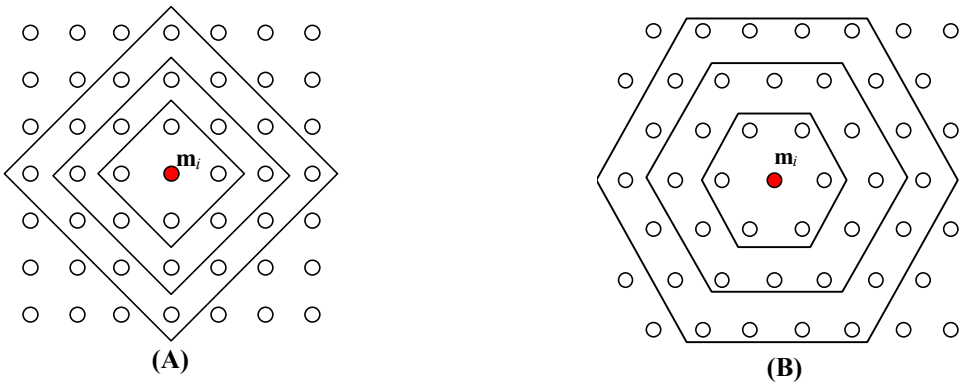


Figura 3-3 – Diferentes configurações de arranjo para o SOM em $\Re^2$. Em (A) vê-se a vizinhança retangular enquanto que em (B) observa-se um arranjo com vizinhança hexagonal.

Um neurônio é considerado vizinho de outro no arranjo conforme a configuração adotada, o que define a vizinhança imediata com 4 e 6 vizinhos nos arranjos retangular e hexagonal, respectivamente. O formato do arranjo influencia diretamente a adaptação do SOM, sendo que o modelo hexagonal oferece tradicionalmente resultados "melhores" que o retangular.

Cada neurônio *i* é representado por um vetor de pesos sinápticos $\mathbf{m}_i = [m_{i1}, \ldots, m_{iD}]^T \in \Re^D$ e todos os neurônios são conectados ao sinal de entrada ou dado recebido (Figura 3-4):

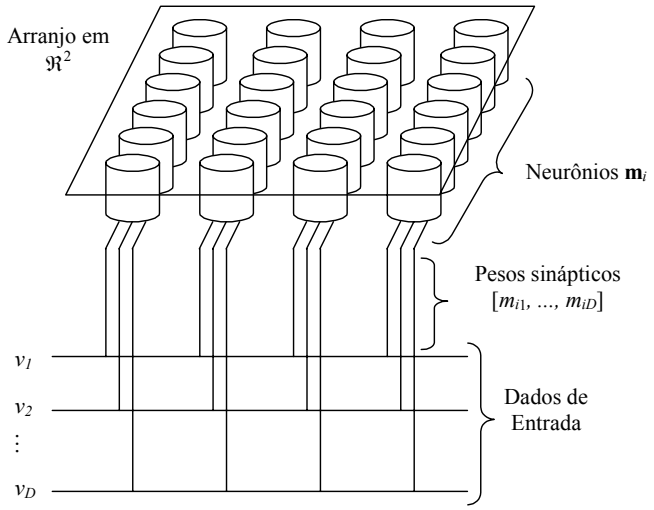


Figura 3-4 – Todos os neurônios do arranjo, representados por vetores de pesos sinápticos $\mathbf{m}_i = [m_{i1}, \ldots, m_{iD}]$, i=1, ..., 24, recebem o mesmo dado de entrada.

Seja $\mathbf{v}_n \in \mathbf{V}$ um dado de entrada tomado aleatoriamente ($n \in \{1, \ldots, N\}$) e apresentado à rede. Como todos os neurônios do arranjo recebem a mesma entrada $\mathbf{v}_n$ (Figura 3-4),

calcula-se a distância do vetor de pesos $\mathbf{m}_i$ de cada neurônio i ao vetor $\mathbf{v}_n$ de acordo com uma métrica, que no caso da distância euclidiana é dada por:

$$d(\mathbf{m}_i, \mathbf{v}_n) = \left\| \mathbf{m}_i - \mathbf{v}_n \right\| = \sqrt{\sum_{j=1}^{D} \left| m_{ij} - v_{nj} \right|^2} \qquad \textbf{Equação 3-1}$$

Calculadas todas as distâncias, é eleito um neurônio BMU de índice c na forma:

$$c = \arg \min_i \left\{ \left\| \mathbf{m}_i - \mathbf{v}_n \right\| \right\} \qquad \textbf{Equação 3-2}$$

A proposta original de aprendizado competitivo diz que o neurônio BMU deve então ser adaptado para melhor representar o sinal de entrada segunda a regra WTA. Como já colocado, *não apenas o neurônio que ganhou a competição é adaptado mas também seus vizinhos*, estabelecendo uma interação local entre os neurônios que, ao longo do aprendizado, promove a organização geral do mapa (Kohonen, 1997, pg 87). O aprendizado, isto é, o novo valor do peso sináptico do i-ésimo neurônio no instante de tempo (t+1), é definido por uma equação de adaptação dada por:

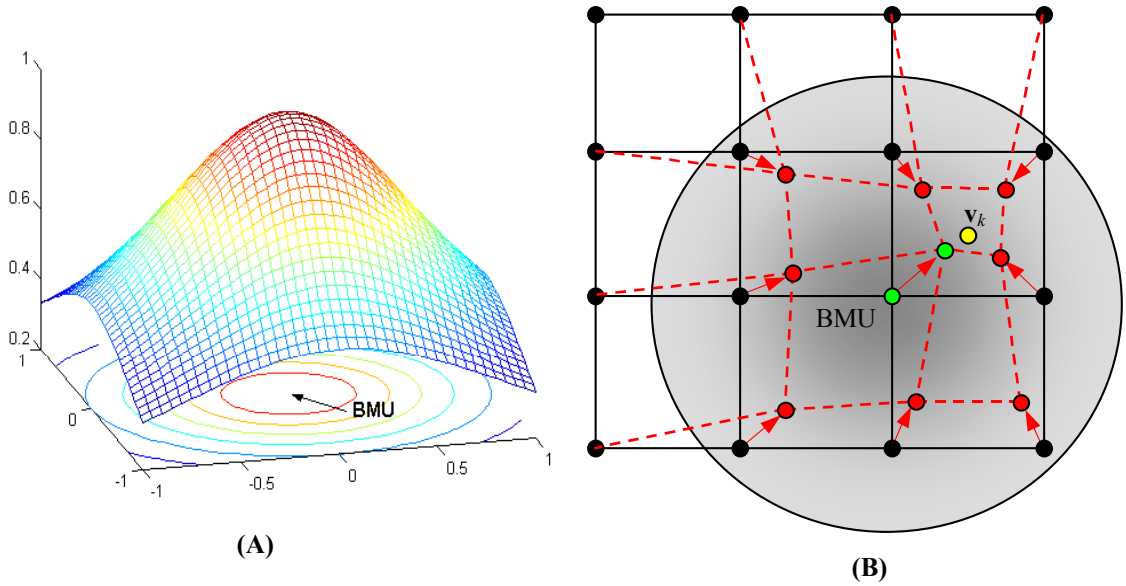
$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) \cdot h_{ci}(t) \cdot \left[\mathbf{m}_i(t) - \mathbf{v}_n(t) \right] \qquad \textbf{Equação 3-3}$$

onde t = 0,1,2 ... é um número inteiro representando a coordenada discreta de tempo e $\alpha(t)$ define a *taxa de aprendizado*. O grau de adaptação do neurônio BMU e de seus vizinhos depende, portanto, da *função de vizinhança*, h_{ci} e da taxa de aprendizado α. É necessário que $h_{ci}(t) \rightarrow 0$ quando $t \rightarrow \infty$, ou seja, a função deve reduzir o grau de vizinhança relativo ao neurônio BMU ao longo do treinamento para ocorrer a convergência do mapa. Tradicionalmente, também, $\alpha(t) \rightarrow 0$ quando $t \rightarrow \infty$ (Kohonen, 1997, pg. 87).

Normalmente, $h_{ci} = h\left(\left\| \mathbf{r}_c - \mathbf{r}_i \right\|, t \right)$, com $\mathbf{r}_c$ e $\mathbf{r}_i$ representando as posições dos neurônios de índices c e i dentro do arranjo, indicando que quando $\left\| \mathbf{r}_c - \mathbf{r}_i \right\|$ aumenta, h_{ci} sofre uma redução exponencial. A forma e o raio de h_{ci} controlam a flexibilidade do mapa (Kohonen, 1997). Se for escolhida uma função de vizinhança discreta, onde h_{ci} = 1 caso o neurônio faça parte da região de vizinhança, e h_{ci} = 0 caso contrário, temos o denominado *SOM de produto interno* (*dot product SOM*). Entretanto, uma escolha típica para esta função quando o SOM é aplicado à mineração de dados é uma gaussiana da forma:

$$h_{ci} = \exp\left(-\frac{\left\| \mathbf{r}_c - \mathbf{r}_i \right\|^2}{2\sigma^2(t)} \right)$$ **Equação 3-4**

O parâmetro $\sigma(t)$ define a largura da região de vizinhança, chamada *raio de vizinhança*. Normalmente $\sigma(t) \to 0$ quando $t \to \infty$. (Kohonen, 1997, pg. 87).



(A) **(B)**

Figura 3-5 – Ilustração da adaptação dos pesos de um SOM para a apresentação de um único padrão em $\Re^2$. Em (A) uma representação da função h_{ci} sobre um mapa bidimensional cuja projeção pode ser vista em (B). Quanto mais próximo um neurônio encontra-se do BMU, isto é, quanto menor a distância $\left\| \mathbf{r}_c - \mathbf{r}_i \right\|$, maior é a adaptação aplicada ao neurônio. O neurônio com maior adaptação é, obviamente, o BMU.

A Figura 3-5 representa h_{ci} e sua influência na taxa de aprendizado dos neurônios na vizinhança do BMU. Observa-se que o neurônio que venceu a competição "arrasta" seus vizinhos na direção do objeto apresentado numa proporção que depende da gaussiana h_{ci}, isto é, os neurônios vizinhos tendem a se aproximar. Dessa forma, ao longo do aprendizado, estímulos semelhantes a $\mathbf{v}_n$ serão percebidos por neurônios próximos entre si, o que acaba por promover a ordenação topológica dos neurônios da rede em relação aos dados no espaço original.

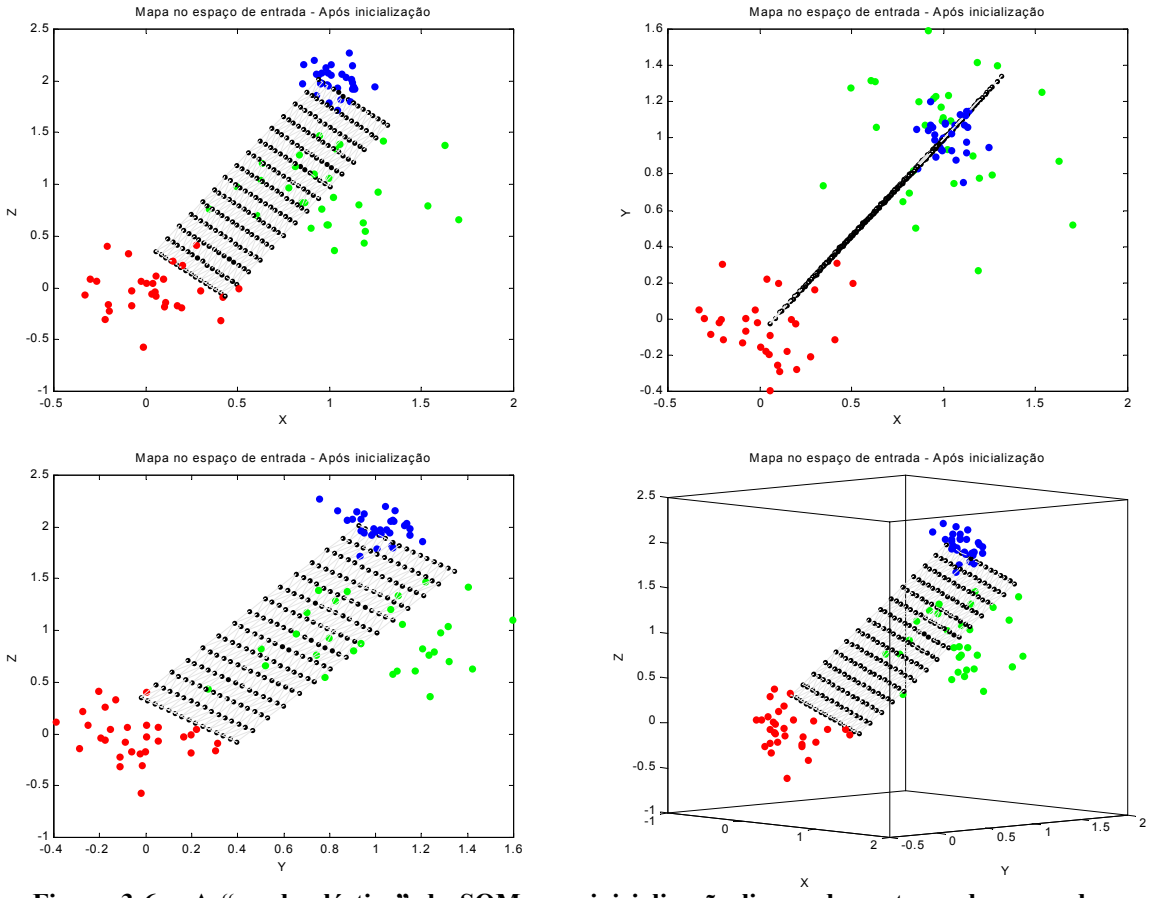


Figura 3-6 – A "grade elástica" do SOM com inicialização linear dos vetores de pesos dos neurônios sobre um conjunto de dados artificiais em $\Re^3$ (veja Figura 2-11 para uma descrição do conjunto). A grade inicial é alinhada com o plano gerado pelos dois eixos de maior variância do conjunto de dados. Suas dimensões foram normalizadas e escalonadas pela raiz quadrada das duas maiores variâncias.

Numa interpretação bem pragmática, o SOM comporta-se como uma grade composta de neurônios ligados entre si por conexões elásticas, responsáveis por dobrar, comprimir ou esticar a grade de forma a representar, da melhor forma possível, o conjunto de dados no espaço original. Uma simulação deste comportamento é ilustrada na Figura 3-6, onde exemplifica-se a inicialização dos vetores de pesos da grade de neurônios, e na Figura 3-7, onde se pode observar a grade de neurônios já adaptada ao conjunto de dados a partir de diferentes pontos de vista.

É interessante de ser notada a deformação que a grade original sofre na tentativa de representar o conjunto de dados, sendo comprimida em regiões de alta densidade (pontos azuis) e distendida em regiões de baixa densidade (pontos verdes).

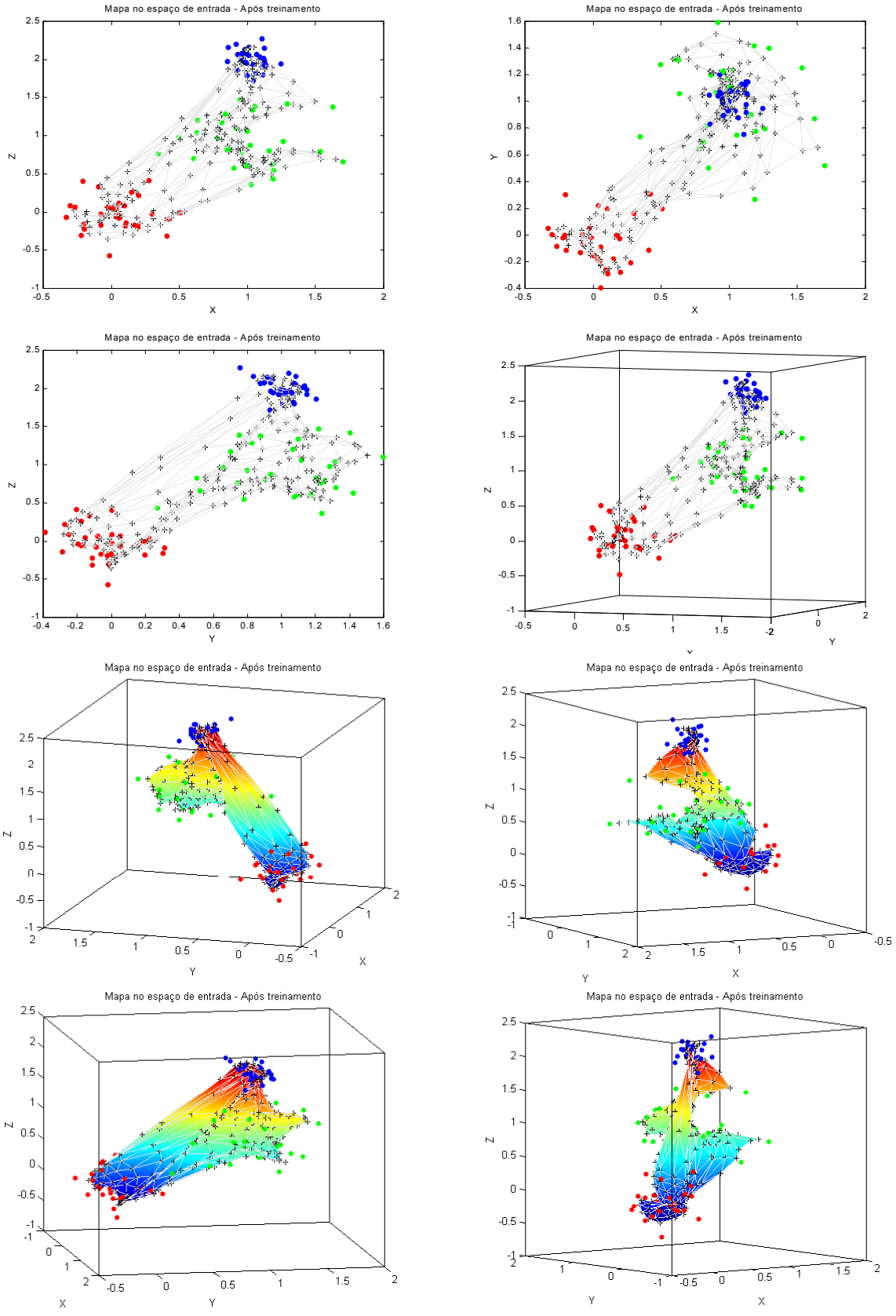


Figura 3-7 – A "grade elástica" já adaptada sobre os dados da Figura 3-6. As quatro últimas figuras ilustram a superfície formada pela grade sob alguns pontos de vista.

3.1.1 Algoritmos de Treinamento

O algoritmo tradicional de treinamento do SOM supõe a atualização dos pesos sinápticos dos neurônios do arranjo toda vez que um item de dados é apresentado à rede, sendo por isso conhecido como *incremental* ou *"on-line"*. Em uma outra versão, as atualizações individuais são postergadas e aplicadas somente após a apresentação de todos os elementos do conjunto de dados **V**. Este último recebe o nome de algoritmo *em lote* ou *"batch"*.

O algoritmo incremental tem como principal vantagem a possibilidade de uso de mapas SOM em problemas para os quais não se tem antecipadamente todos os dados disponíveis, isto é, os dados são coletados e apresentados imediatamente à rede, além de ter uma implementação computacional mais barata e exigir menos memória especialmente em sua versão de linha de comando SOM_PAK, se comparada à SOM Toolbox. O algoritmo incremental é descrito resumidamente a seguir:

1. Inicialize o vetor de pesos $\mathbf{m}_i$ do neurônio i por uma de três formas distintas: randomicamente, utilizando-se elementos do próprio conjunto de dados, ou então linearmente. Faça t=0 e o número de iterações n_it=0, $\mathbf{V}' = \mathbf{V}$ e inicialize $\sigma(n\_it)$ e $\alpha(n\_it)$.
2. Selecione aleatoriamente um vetor de dados $\mathbf{v}_n$ do conjunto $\mathbf{V}'$ e faça $\mathbf{V}' = \mathbf{V}' - \{\mathbf{v}_n\}$.
3. Selecione o BMU (neurônio i com vetor de pesos $\mathbf{m}_i$ mais próximo de $\mathbf{v}_n$) segundo a Equação 3-2
4. O neurônio BMU e seus vizinhos são atualizados conforme a Equação 3-3
5. Faça t=t+1 e volte ao passo 2 enquanto $\mathbf{V}' \neq \varnothing$.
6. Incremente o número de iterações n_it. Se o número máximo de iterações pré-estabelecido não tiver sido atingido, faça t=0, $\mathbf{V}' = \mathbf{V}$ e ajuste $\sigma(n\_it)$ e $\alpha(n\_it)$. Retorne ao passo 2.

Figura 3-8 – Algoritmo de treinamento incremental

A inicialização linear no passo 1 significa distribuir os neurônios de forma ordenada ao longo de um plano alinhado com os eixos das duas maiores variâncias no conjunto de dados (um princípio que lembra o PCA, veja a Seção 2.4.1) e com centróide no centro de massa do mesmo conjunto (veja Figura 3-6).

O algoritmo incremental é sensível à ordem em que os dados são apresentados e particularmente sensível à taxa de aprendizado, especialmente quando mapas grandes são treinados (Kohonen, 1997, pg. 88). O algoritmo em lote elimina o primeiro problema e contorna o segundo, atualizando os pesos somente ao final de uma *época de treinamento*[2]. Para tanto, cada neurônio acumula as contribuições parciais de cada vetor $\mathbf{v}_n$ apresentado ao mapa para os quais ele é BMU durante uma época de treinamento segundo a equação:

$$\Delta\mathbf{m}_i(t+1) = \Delta\mathbf{m}_i(t) + h_{ci} \cdot \left[\mathbf{m}_i - \mathbf{v}_n\right] \qquad \textbf{Equação 3-5}$$

para $n = 1, ..., N$. Ao final de uma época, os neurônios são adaptados conforme uma variante da Equação 3-3 descrita abaixo:

$$\mathbf{m}_i(n\_it+1) = \mathbf{m}_i(n\_it) + \frac{1}{N}\alpha(n\_it) \cdot \Delta\mathbf{m}_i \qquad \textbf{Equação 3-6}$$

A implementação prática deste algoritmo envolve a manipulação de um vetor de tamanho N para acumular os deslocamentos relativos de cada neurônio ao longo de uma época ou ainda uma lista de tamanho N para cada neurônio, quando então seria possível a avaliação de todo o histórico de deslocamentos parciais. O algoritmo em lote é descrito resumidamente na Figura 3-9.

No algoritmo incremental o tempo t é medido pelo número de dados apresentados à rede enquanto que no algoritmo em lote este é medido em número de épocas. Tanto o algoritmo incremental como o de lote operam com um treinamento em duas fases: a primeira, onde ocorre a *ordenação inicial* do mapa, de curta duração e com valores relativamente grandes para α e σ, e uma segunda fase de *convergência*[3], mais demorada, com valores menores para a taxa de aprendizado e para a vizinhança inicial. Kohonen (1997, pg. 115) sugere que a inicialização linear inicial dos pesos sinápticos possa eliminar a fase de ordenação inicial.

[2] uma "época", neste caso, ocorre ao final da apresentação de todos os itens de dados exatamente uma vez.

[3] As fases de treinamento são encontradas na literatura como *"rough training"* ou *"ordering phase"* (1ª fase) e *"fine tunning"* ou *"convergence phase"* (2ª fase).

1. Inicialize o vetor de pesos $\mathbf{m}_i$ do neurônio i linearmente. Faça t=0 e o número de iterações n_it=0, $\mathbf{V}' = \mathbf{V}$ e inicialize $\sigma(n\_it)$. A taxa de aprendizado $\alpha(n\_it)$ recebe um valor pequeno e fixo (0,5 para a fase inicial e 0,05 para a fase de convergência).
2. Selecione um vetor de dados $\mathbf{v}_n$ do conjunto $\mathbf{V}'$ e faça $\mathbf{V}' = \mathbf{V}' - \{\mathbf{v}_n\}$. Esta seleção deve ser a menos custosa possível, podendo ser na ordem em que eles foram armazenados, por exemplo.
3. Selecione o BMU (neurônio i com vetor de pesos $\mathbf{m}_i$ mais próximo de $\mathbf{v}_n$) segundo a Equação 3-2
4. Calcule a contribuição parcial do vetor $\mathbf{v}_n$ para o neurônio BMU e seus vizinhos, segundo a Equação 3-5
5. Volte ao passo 2 enquanto $\mathbf{V}' \neq \varnothing$.
6. Os neurônios e seus vizinhos são atualizados conforme a Equação 3-6
7. Incremente o número de iterações n_it. Se o número máximo de iterações pré-estabelecido não tiver sido atingido, faça $\mathbf{V}' = \mathbf{V}$ e ajuste $\sigma(n\_it)$ e $\alpha(n\_it)$. Retorne ao passo 2.

Figura 3-9 – Algoritmo de treinamento em lote

3.1.2 Interpretação do mapa produzido pelo SOM

Muito embora a idéia inicial do SOM tenha um motivador biológico forte, que sugere a construção de arranjos em 2 dimensões (que certamente é o modelo mais difundido e utilizado para fins de mineração de dados), são possíveis construções em uma, duas ou mais dimensões, conforme a necessidade e o objetivo. Uma vez adaptado, é necessário algum método que possibilite a interpretação do resultado alcançado. As próximas subseções apresentam alguns exemplos e comentários sobre a utilização do SOM conforme a dimensão do arranjo.

3.1.2.1 Arranjos Unidimensionais

Embora seja a única configuração para a qual uma prova de convergência foi estabelecida (Erwin *et al.* 1992), o arranjo unidimensional é pouco explorado na prática, tendo sido usado basicamente em demonstrações sobre o comportamento do SOM. Uma aplicação bastante interessante de mapas unidimensionais é apresentada em Aras *et al.* (1999) para a

solução do problema do caixeiro-viajante, onde uma versão do SOM unidimensional construtiva chamada *KNIES* (do original *Kohonen Network Incorporating Explicit Statistics*) é utilizada com resultados animadores.

O algoritmo inicia-se com um "anel" formado por poucos neurônios (normalmente 2) colocados próximos ao centro de massa do conjunto de cidades e é operado em 2 fases. A fase de *atração* é idêntica ao do algoritmo tradicional, onde o BMU e seus vizinhos são aproximados do sinal apresentado (neste caso, os sinais são cidades representadas por coordenadas em $\Re^2$). Na fase de *repulsão*, os neurônios que não participaram da fase de atração são afastados, de forma que as propriedades estatísticas globais do conjunto (média, variância etc.) permaneçam constantes.

O KNIES remove, ao final de seu ciclo, os neurônios que não sejam explicitamente responsáveis por alguma cidade.

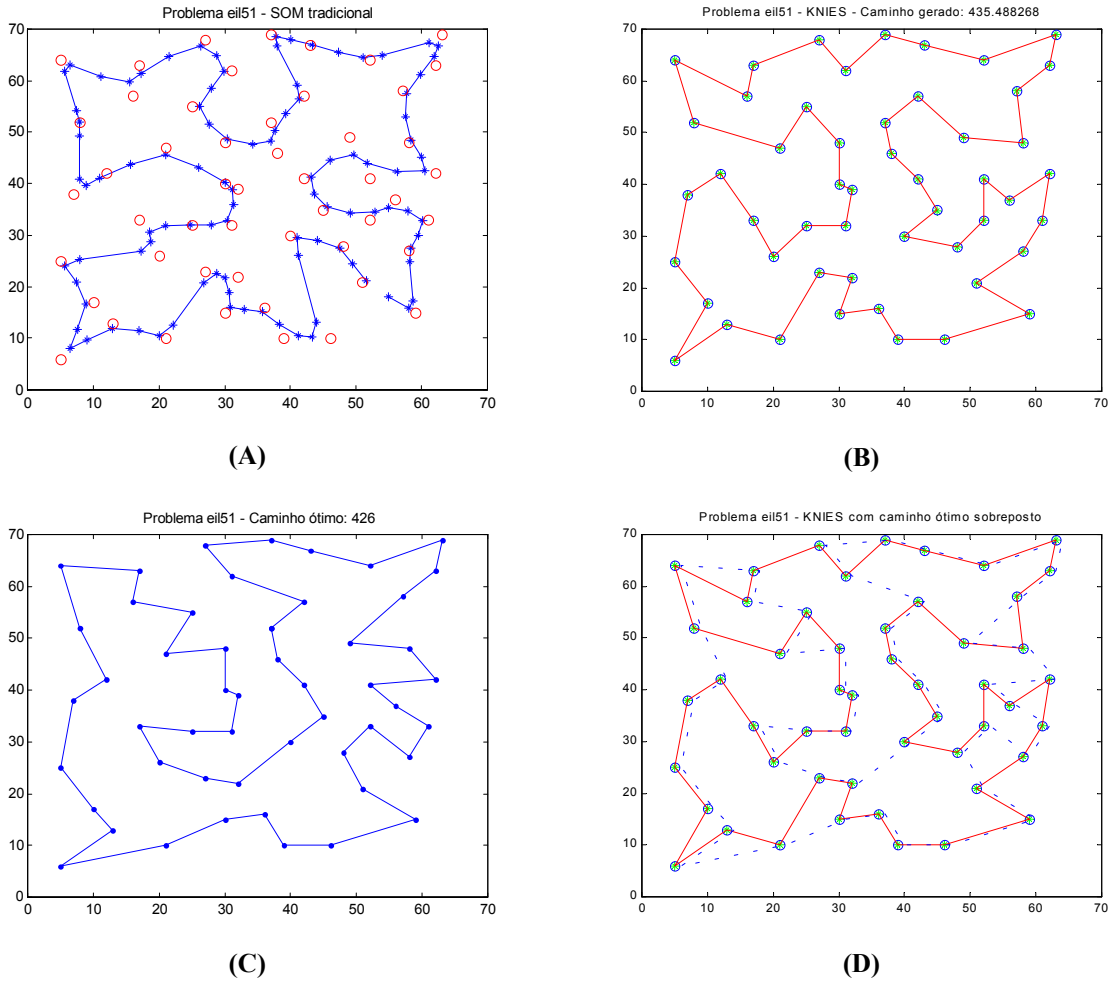


Figura 3-10 – O problema do caixeiro-viajante para 51 cidades (*"eil51"*). Em (A) o SOM apresenta um resultado inferior se comparado ao método KNIES (B). Em (C) o caminho ótimo (com menor custo) obtido por métodos de otimização. Nota-se que o SOM opera com um número de neurônios maior que o número de cidades, enquanto o KNIES termina com o número de neurônios exatamente igual ao número de cidades. Em (D) o caminho ótimo (tracejado) é sobreposto ao obtido pelo KNIES.

O algoritmo tradicional tem uma performance bastante inferior se comparada ao método proposto, como pode ser observado na Figura 3-10, onde alguns testes foram executados para o problema *"eil51"* (Reinelt, 1991). Este problema consiste de um conjunto de 51 cidades representadas por suas coordenadas num plano bidimensional.

3.1.2.2 Arranjos Bidimensionais

Os arranjos bidimensionais possuem uma estrutura de vizinhança plana retangular ou hexagonal. Outras configurações possíveis são arranjos cilíndricos e toroidais (Vesanto *et al.* 2000). Estes dois últimos formatos do arranjo são pouco explorados e possuem um tratamento conservador por parte das ferramentas utilizadas nesta dissertação: tanto o mapa

cilíndrico como o mapa toroidal são analisados "abrindo-se" o arranjo e tomando-o como plano.

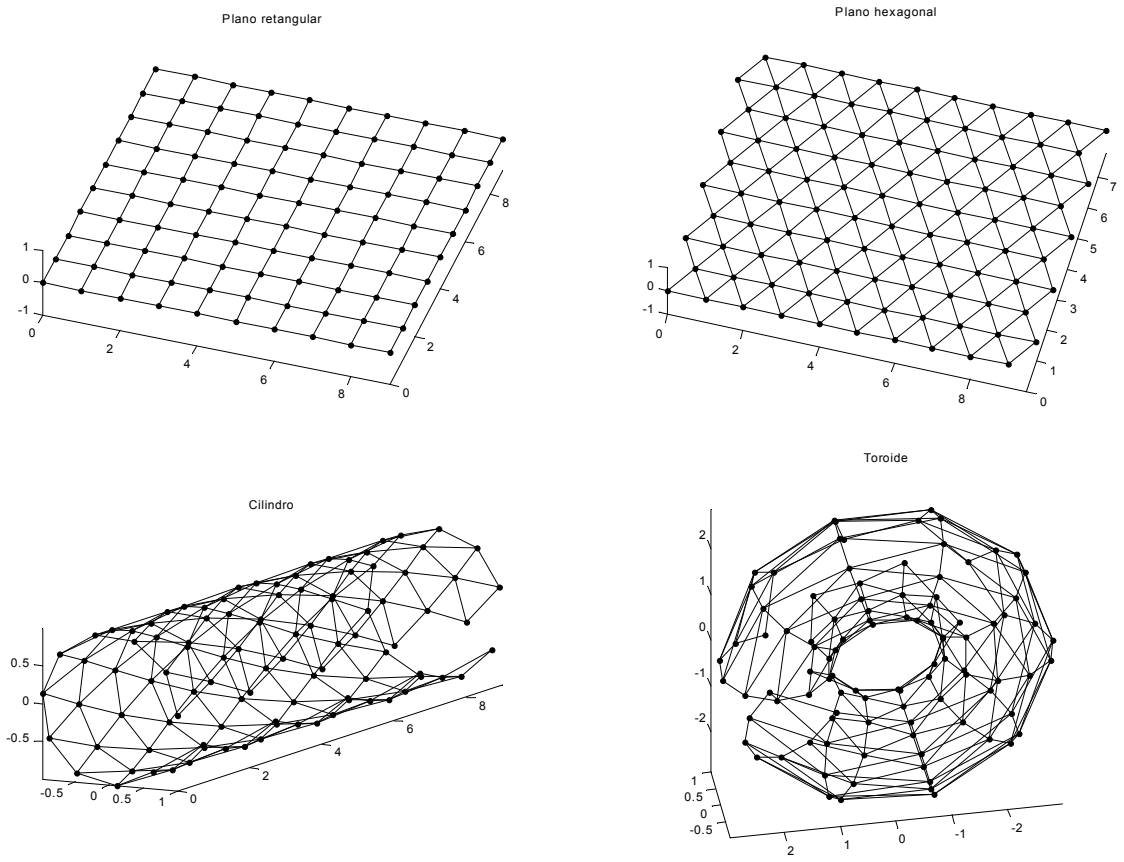
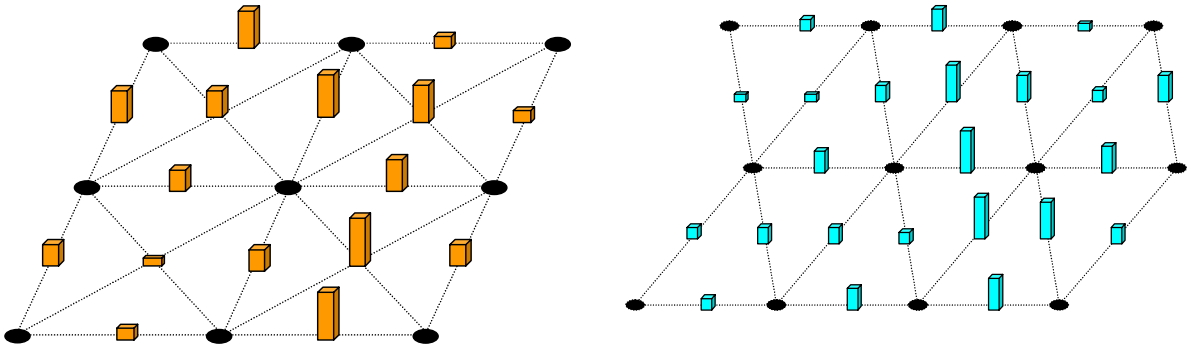


Figura 3-11 – Diferentes formatos para o arranjo SOM.

Quando o objetivo do SOM é a avaliação de possíveis agrupamentos (como é o caso nesta dissertação) o método mais comumente empregado é a *matriz de distâncias unificada* ou *matriz-U* (Ultsch & Siemon, 1989, 1990). A matriz-U é uma matriz composta pelas distâncias entre todos os neurônios vizinhos no arranjo (Figura 3-12):



(A) **(B)**

Figura 3-12 – Exemplo da matriz-U num arranjo retangular (A) e hexagonal (B). No caso (A), a composição da distância nas diagonais é obtida pela média aritmética das 2 diagonais envolvidas.

A matriz-U tem dimensão $(2L\text{-}1) \times (2C\text{-}1)$, considerando um arranjo retangular plano de tamanho $L \times C$. O valor da matriz-U sobre os neurônios em si é normalmente obtido pela média aritmética das distâncias entre os vetores de pesos de toda a vizinhança do neurônio e o seu próprio vetor de pesos (Vesanto *et al.* 2000; Iivarinen *et al.* 1994). Tomando a matriz-U como uma superfície de nível, pode-se avaliar visualmente a existência de "vales" (que surgem onde os vetores de pesos dos neurônios são mais próximos entre si) separados por "elevações" (onde os vetores de pesos dos neurônios encontram-se mais distantes). Um vale é associado com a ocorrência de um agrupamento e quanto mais alta uma elevação separando dois vales, tanto mais distintos são estes agrupamentos no espaço de dados.

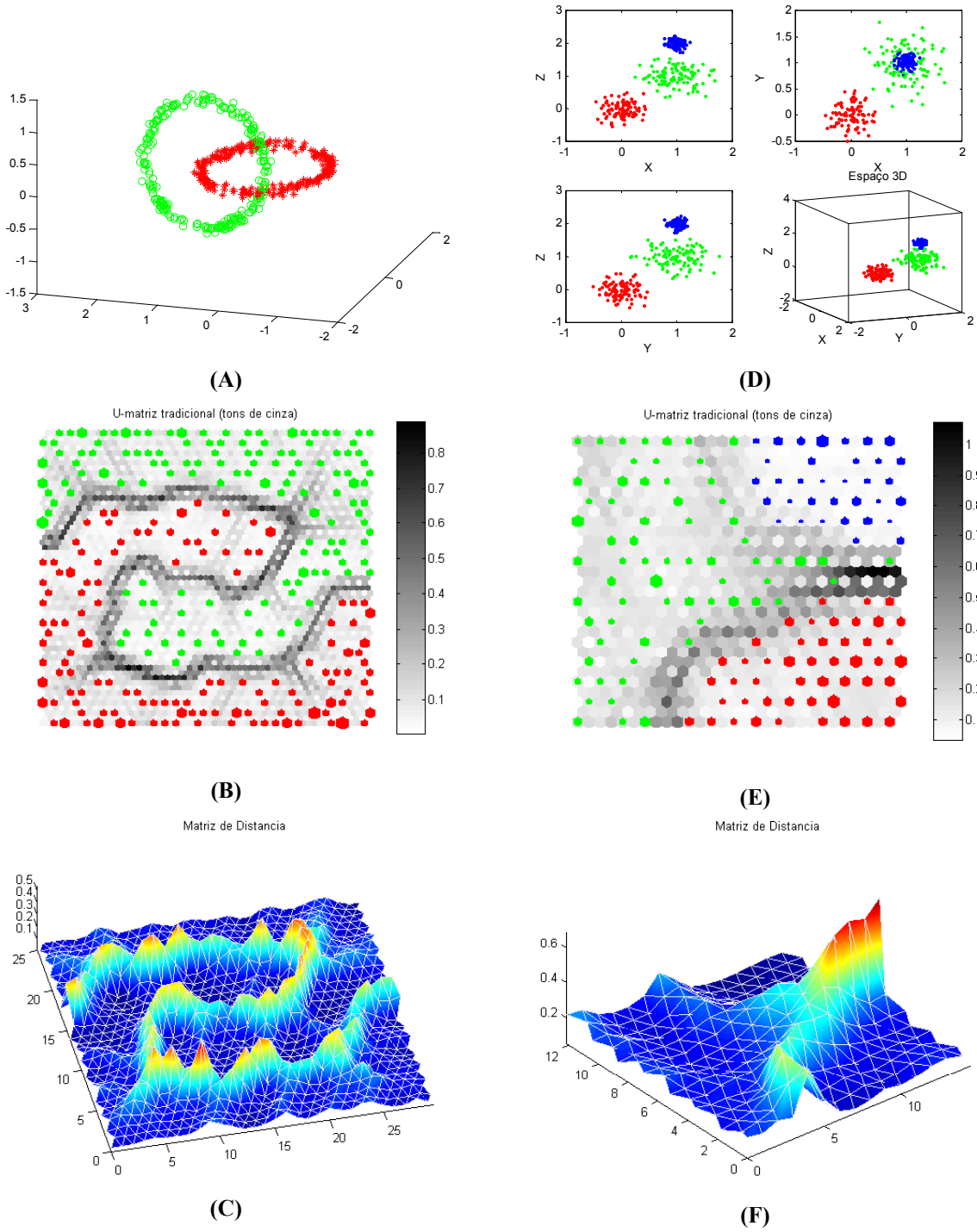


Figura 3-13 – Exemplo de análise de agrupamentos com uso da matriz-U. Em (A), o clássico *Chainlink Dataset*, e em (D), um conjunto artificial em $\Re^3$ (veja Figura 2-11 para uma descrição mais detalhada dos conjuntos utilizados). O SOM demonstra sua capacidade de separar o conjunto de toróides (A), como pode ser visto em (B) e (C). Em (E) e (F), o conjunto artificial (D) também é facilmente separado. A matriz-U, como originalmente proposta, baseia-se em tons de cinza para identificação dos agrupamentos, conforme pode ser visto em (B), onde tons claros indicam proximidade entre os vetores de pesos dos neurônios.

Outras possibilidades de análise de agrupamentos (Kaski *et al.* 1999; Vesanto, 1999, 2000) serão exploradas no Capítulo 5.

3.1.2.3 Arranjos *N*-dimensionais

Os mapas SOM com arranjo de dimensão maior que 2 não são passíveis de visualização direta, sendo necessário métodos especiais para avaliar os resultados obtidos. Uma proposta geral feita por Costa (1999, capítulo 7) envolve uma extensão do algoritmo *SL-SOM* (*Self-Labeling SOM*) do mesmo autor. O SL-SOM é baseado em segmentação automática de imagens, entretanto, a segmentação de hipervolumes é computacionalmente mais onerosa, tornando o método possível, porém pouco explorado.

3.1.3 Abordagens variantes

A literatura apresenta diversas abordagens variantes e modelos que podem ser considerados derivados do SOM de alguma forma. Kohonen (1997) generaliza que o princípio fundamental através do qual um modelo neural pode representar um conjunto de dados de forma topologicamente correta baseia-se na atualização do neurônio vencedor e de um subconjunto de neurônios na vizinhança deste. Considerando esta generalização, há inúmeras formas de construir algoritmos derivados do SOM modificando-se os seguintes aspectos (Kohonen, 1997; Van Hulle, 2000):

1. forma de escolha do neurônio BMU: embora a distância euclidiana seja a mais comum das opções, há diversas outras métricas possíveis para a escolha do neurônio BMU;
2. critério de vizinhança adotado: além da vizinhança retangular ou hexagonal, é possível definir a mesma dinamicamente, até com a inclusão e remoção de neurônios durante o processo de treinamento, o que leva a modelos construtivos;
3. busca de características invariantes: a maioria dos algoritmos baseados em redes neurais artificiais ainda não é capaz de detectar características nos dados invariantes a transformações como rotação, translação e escala. Este é o objetivo do algoritmo *Adaptive Subspace SOM* (*ASSOM*) (Kohonen, 1996, 1997; Kohonen *et al.* 1997);
4. uso de mapas hierárquicos e sistemas de mapas SOM: a estrutura de dados hierárquicos pode ser melhor representada por conjuntos estruturados de mapas, o que pode também promover um maior detalhamento e separação de agrupamentos.

3.1.3.1 Variantes na forma de escolha do neurônio BMU

O uso da norma euclidiana para a escolha do neurônio BMU é um procedimento comum quando nenhuma outra informação prévia existe acerca dos dados de entrada, pressupondo assim a existência de agrupamentos hiperesféricos. A norma euclidiana tem a propriedade de ser invariante a rotações aplicadas aos dados de entrada (Costa, 1999). Entretanto, é possível (e mesmo desejável, sob certas circunstâncias) utilizar-se de outras métricas para a escolha do neurônio BMU. Considerando-se dois vetores quaisquer $\mathbf{x}$, $\mathbf{y} \in \Re^D$ representados por suas coordenadas $[x_{1\,,\,\ldots,}\,x_D]^T$ e $[y_1\,,\,\ldots,\,y_D]^T$, a generalização da norma euclidiana, conhecida como *métrica de Minkowsky* ou norma L_λ é dada por:

$$d_\lambda(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_\lambda = \sqrt[\lambda]{\sum_{i=1}^{D} |x_i - y_i|^\lambda} \qquad , \lambda \in \Re \qquad \textbf{Equação 3-7}$$

A norma L_λ é invariante a translações em geral (Costa, 1999), o que pode ser útil no caso de reconhecimento de padrões que sofrem este tipo de transformação linear, tendo sido utilizada em experimentos de Psicologia (Kohonen, 1997).

A *métrica de Tanimoto*, dada por

$$d_T(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_T = \frac{\mathbf{x}^T\mathbf{y}}{\mathbf{x}^T\mathbf{x} + \mathbf{y}^T\mathbf{y} - \mathbf{x}^T\mathbf{y}} \qquad \textbf{Equação 3-8}$$

expressa a razão entre os atributos comuns a $\mathbf{x}$ e $\mathbf{y}$ e o número total de atributos diferentes considerados e pode ser utilizada para avaliar a relação de similaridade entre documentos de texto. Tomando-se os atributos por palavras-chave capazes de identificar documentos quaisquer, pode-se avaliar a similaridade entre dois documentos medindo-se relação entre a quantidade de palavras-chave compartilhadas por estes e o total de palavras-chave diferentes existentes entre estes dois documentos (Kohonen, 1997).

Outra possibilidade inclui a *métrica de Mahalanobis*, dada por

$$d_M(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_M = \sqrt{(\mathbf{x}-\mathbf{y})^T\ \mathrm{cov}^{-1}(\mathbf{x}-\mathbf{y})} \qquad \textbf{Equação 3-9}$$

Esta métrica leva em consideração a possibilidade dos dados de entrada apresentarem alguma correlação entre si, isto é, não serem estatisticamente independentes, o que pressupõe a existência de agrupamentos hiperelipsoidais. Apesar de ser indispensável em muitas aplicações práticas, esta métrica é computacionalmente mais cara se comparada à

métrica euclidiana, uma vez que exige o cálculo da matriz de covariância envolvendo todos os elementos dos vetores. Além disso, para uma maior confiabilidade nas medidas de covariância, um aumento na dimensão dos vetores envolvidos vai requerer uma quantidade elevada de dados (Kohonen, 1997).

Quando os dados de entrada possuem diferentes variâncias entre seus atributos constituintes, o uso da métrica euclidiana leva a orientações oblíquas no arranjo do SOM. Kangas *et al.* (1990) propõem uma forma de lidar com este fato *ponderando* a distância entre os vetores de dados e os neurônios do arranjo através de um conjunto de fatores, os quais são adaptados iterativamente ao longo do processo de treinamento. Seja $\mathbf{v} \in \Re^D$ um vetor do conjunto de entrada e $\mathbf{m}_i$ o vetor de pesos de um neurônio qualquer de índice i no arranjo do SOM. O cálculo proposto, chamado *Adaptive Tensorial Weighting* (*ATW*), é dado por

$$d_{ATW}(\mathbf{v},\mathbf{m}) = \sqrt{\sum_{j=1}^{D} \omega_{ij}^2 \left(v_j - m_{ij}\right)^2} \qquad \textbf{Equação 3-10}$$

onde os pesos ϖ_{ij} são estimados de forma que cada neurônio apresente um erro de representação aproximadamente igual a todos os outros do arranjo (Kangas *et al.* 1990; Kohonen, 1997; Van Hulle, 2000).

3.1.3.2 Variantes no critério de vizinhança adotado

Os arranjos SOM com relação de vizinhança entre seus neurônios previamente definidas e rígidas durante todo o processo de treinamento (por exemplo, retangulares ou hexagonais no caso de arranjos em 2D) exibem certa dificuldade de representar a estrutura intrínseca do conjunto de dados de entrada, exatamente nos casos em que esta é mais proeminente, isto é, com agrupamentos bastante alongados ou então desconexos. Para lidar com esta característica, foram propostas diversas variações do SOM na literatura (Kohonen, 1997; Van Hulle, 2000).

Kangas *et al.* (1990) apresentam uma proposta onde os neurônios são adaptados conforme o algoritmo tradicional do SOM, mas a relação de vizinhança é definida ao longo do treinamento de acordo com o MST. Neste caso, o comprimento dos arcos é definido pela distância euclidiana entre os neurônios do arranjo. A proposta é capaz de representar

estruturas alongadas, inclusive com agrupamento desconexos, sendo bastante rápida na convergência. Entretanto, a mesma não garante a ordenação topológica e atua mal em agrupamentos hiperesféricos. A Figura 3-14 representa um exemplo de vizinhança segundo o MST (*Minimum Spanning Tree*) (Gower & Ross, 1969).

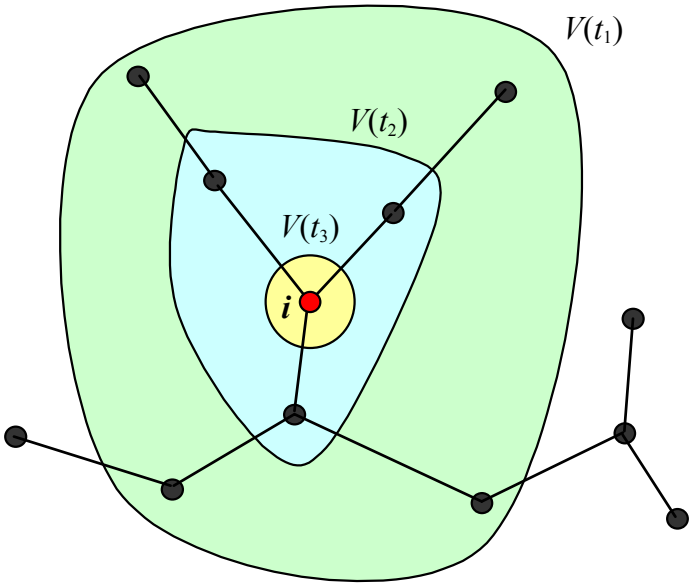


Figura 3-14 – A vizinhança *V*(*t*) do neurônio *i* é estabelecida dinamicamente, respeitando a saída do algoritmo MST, e varia com o tempo, iniciando-se (normalmente) grande e diminuindo ao longo do treinamento.

Em Martinetz & Schulten (1991) é proposto um modelo onde os neurônios não têm, em nenhum momento, uma região de vizinhança fixa que defina a atualização dos pesos sinápticos. A vizinhança é, de fato, definida conforme cada item de dado é apresentado à rede, cada neurônio comportando-se como "moléculas de gás" que tentam preencher o "espaço de dados", daí o nome de *Neural Gas* (*NG*) dado ao algoritmo. Este modelo é capaz de recuperar a estrutura de conjuntos de dados bastante complexos, inclusive com diferentes dimensões e regiões desconexas, com rapidez. Entretanto, também não garante a ordenação topológica, apenas o conceito de similaridade pode ser avaliado somente ao final do treinamento. Fritzke (1995a) propõe uma versão construtiva deste mesmo algoritmo (*Growing Neural Gas* - *GNG*) no qual o número de neurônios aumenta durante o treinamento da rede, sendo inseridos próximos aos neurônios que acumularam o maior erro de representação. Ambos os modelos adicionam e removem arcos entre os neurônios segundo a regra de Hebb. Um exemplo do comportamento destes algoritmos é apresentado na Figura 3-15 com a utilização do software *"DemoGNG v1.5"* (Loos & Fritzke, 1998).

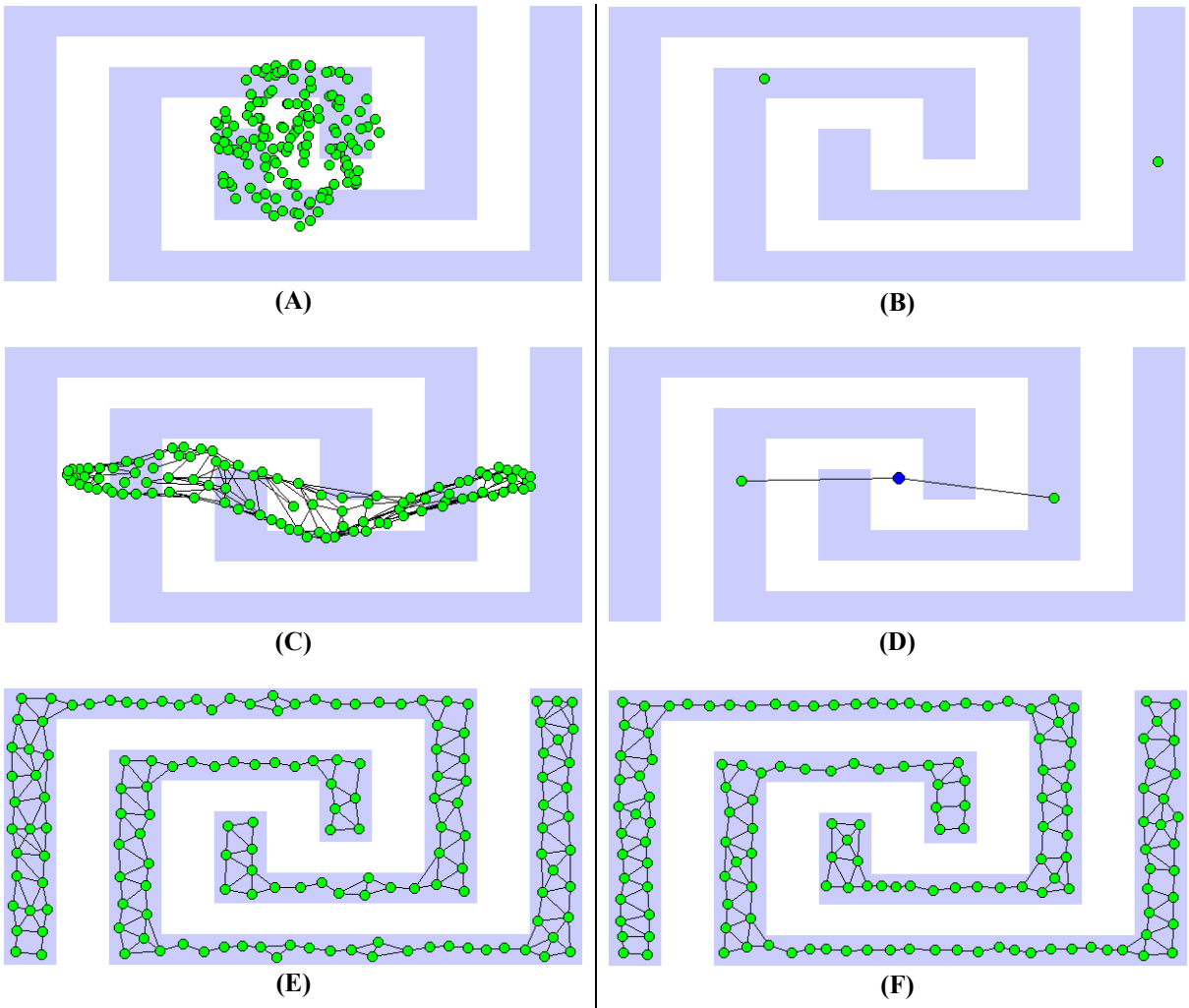


Figura 3-15 – Algoritmo *Neural Gas*. Na coluna à esquerda, o algoritmo original e na coluna à direita, o algoritmo construtivo (*Growing Neural Gas*). (A) e (B) representam a situação inicial, (C) e (D) um momento intermediário após a apresentação de 500 itens de dados. Em (E) e (F) a situação final de treinamento com 40000 itens apresentados, ambos com representações bastante semelhantes. Os dados são pontos em $\Re^2$ escolhidos aleatoriamente e que pertencem à região definida pelas duas espirais concêntricas observadas nas figuras.

Em Fritzke (1991a,b) propõe-se um modelo construtivo, o *Growing Cell Structure* (*GCS*), que consiste em um arranjo em 2D composto de neurônios (nós) conectados entre si em forma de triângulos. Inicialmente, 3 nós estão presentes e a atualização dos nós ocorre para o neurônio BMU e seus vizinhos. O modelo insere novos nós na vizinhança daqueles cujo erro de representação é grande, conectando-o ao arranjo (Figura 3-16). Posteriormente, Fritzke (1994) propõe uma generalização do GCS para operar com arranjos baseados em hipertetraedros, o que promove uma melhor recuperação da estrutura, mas dificulta a visualização.

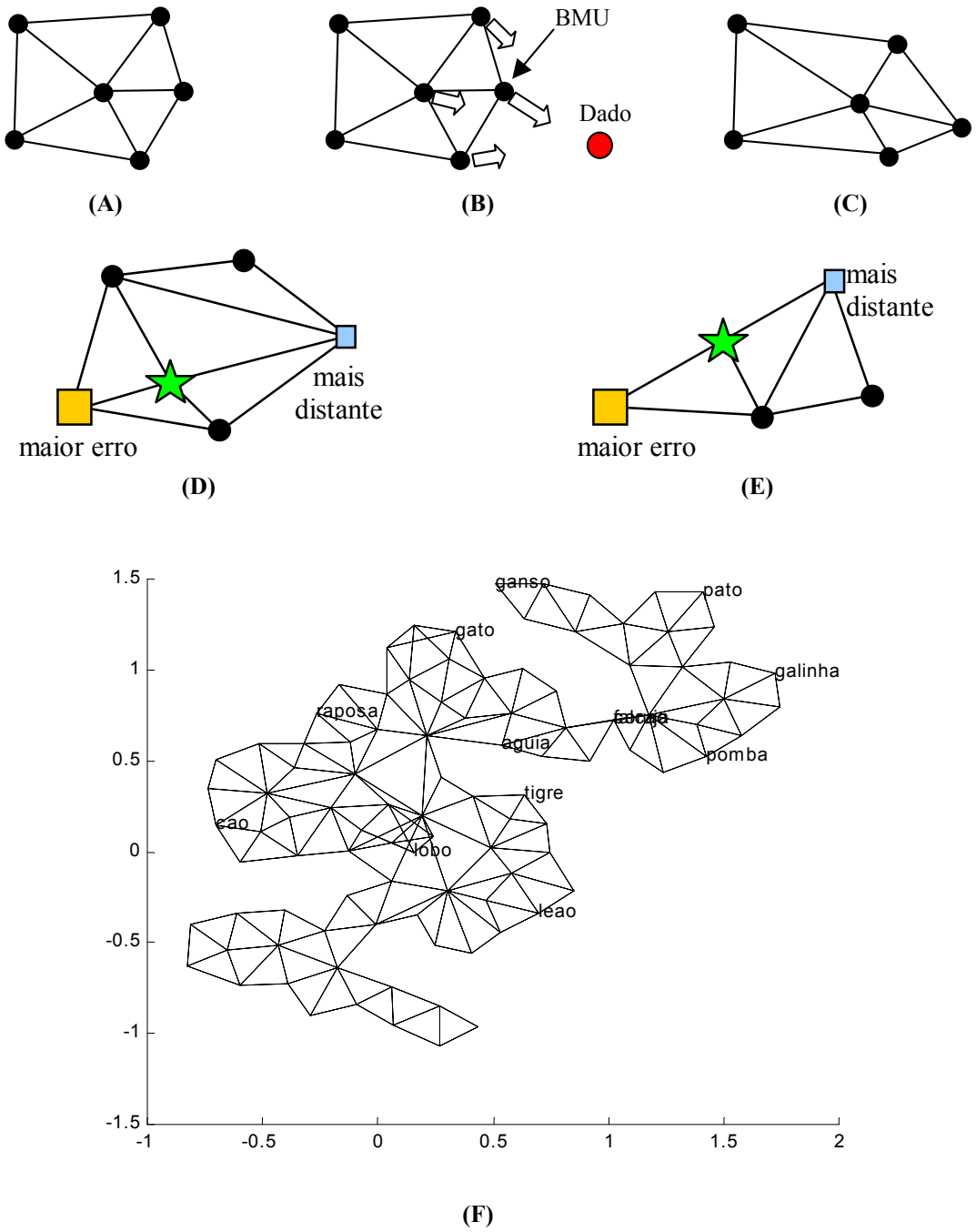


Figura 3-16 – O modelo GCS. Em (A), (B) e (C) têm-se a atualização da rede após a apresentação de um sinal. Em (D) e (E), exemplos de inserção de nós: o novo nó (representado por uma estrela) sempre ocorre no arco entre o neurônio com maior erro e seu vizinho mais distante. Em (F), um exemplo de rede GCS operando sobre o banco de dados de animais proposto por Ritter & Kohonen (1989). Os testes foram realizados com a Toolbox Matlab® GCSVIS (Walker *et al.* 1999).

Em Fritzke (1995b, 1996) sugere-se ainda uma variante construtiva diretamente derivada do SOM, o *Growing Grid* (*GG*). A partir de um arranjo com 4 neurônios iniciais, são inseridas linhas e colunas de novos neurônios no arranjo, promovendo uma busca

automática da dimensão ideal para o arranjo do SOM, que é definido *a priori* no modelo original. A Figura 3-17 apresenta um exemplo do algoritmo GG.

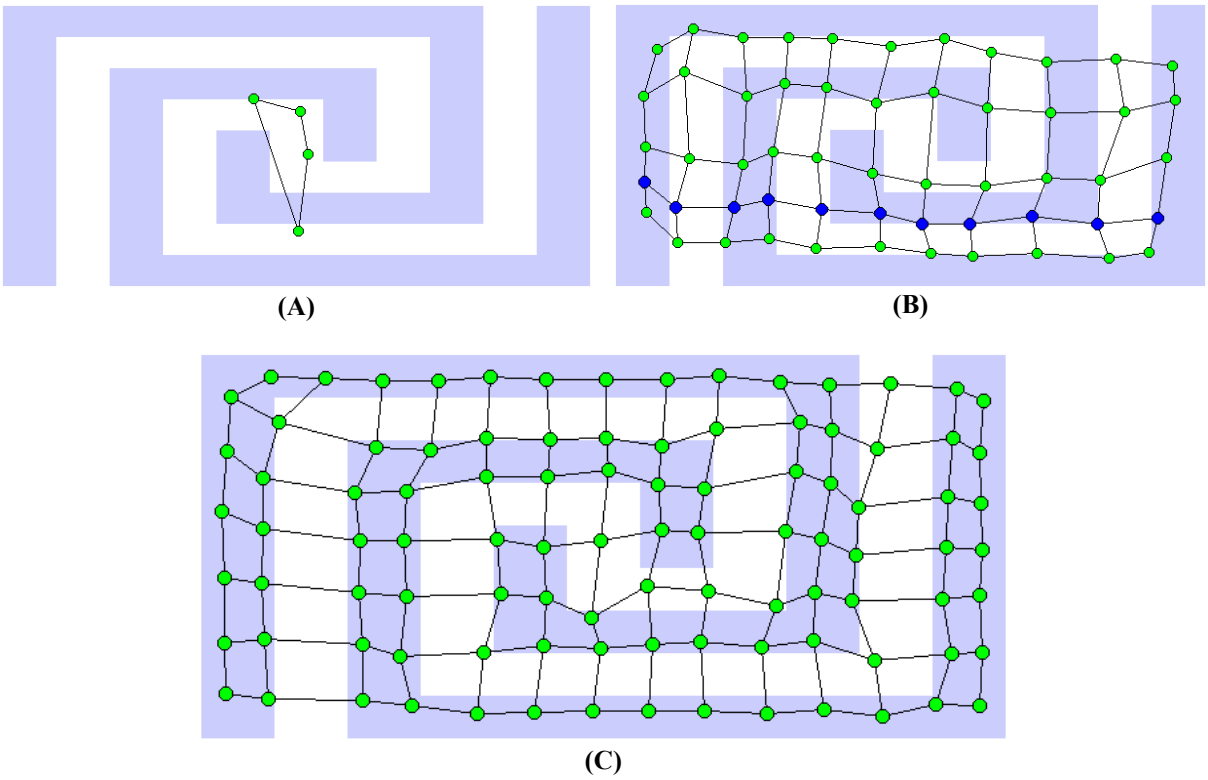


Figura 3-17 – Exemplo de operação do algoritmo *Growing Grid*. Em (A) o arranjo inicial com um estado intermediário em (B) no momento de inserção de uma nova linha no arranjo. Em (C) a situação final num arranjo 14 × 7 sugerida automaticamente pelo modelo. Os dados são pontos em $\Re^2$ escolhidos aleatoriamente e que pertencem à região definida pelas duas espirais concêntricas observadas nas figuras.

Blackmore & Miikkulainen (1993, 1995) e Blackmore (1995) propõem uma versão construtiva do SOM, o *Incremental Grid Growing* (*IGG*), com o objetivo de suplantar a característica indesejável do GCS e do NG de, eventualmente, gerarem arranjos num espaço de dimensão elevada, dificultando a avaliação. O algoritmo IGG evita isso (seu arranjo é sempre plano) mas exige o formato retangular, nem sempre o melhor para recuperar a estrutura intrínseca dos dados. A proposta é iniciar com um arranjo de 4 neurônios (nós) e executar o processo tradicional de adaptação do SOM. Após isso, adiciona-se novos nós aos *nós de fronteira* que apresentem grande erro de representação, sendo que um "*nó de fronteira*" é todo aquele que possui pelo menos uma das direções (tomadas sobre os eixos cartesianos) no arranjo imediatamente vizinhas ainda não ocupada por outro neurônio. Os novos nós são conectados diretamente ao nó de fronteira do qual derivaram. O algoritmo examinará agora quais conexões devem ser geradas, considerando a

distância euclidiana entre pares de nós, e quais devem ser removidas, isto é, cujos nós estão afastados além de um valor limite. O algoritmo garante a manutenção de um arranjo plano e sempre visualizável com facilidade. A Figura 3-18 apresenta um exemplo do algoritmo IGG.

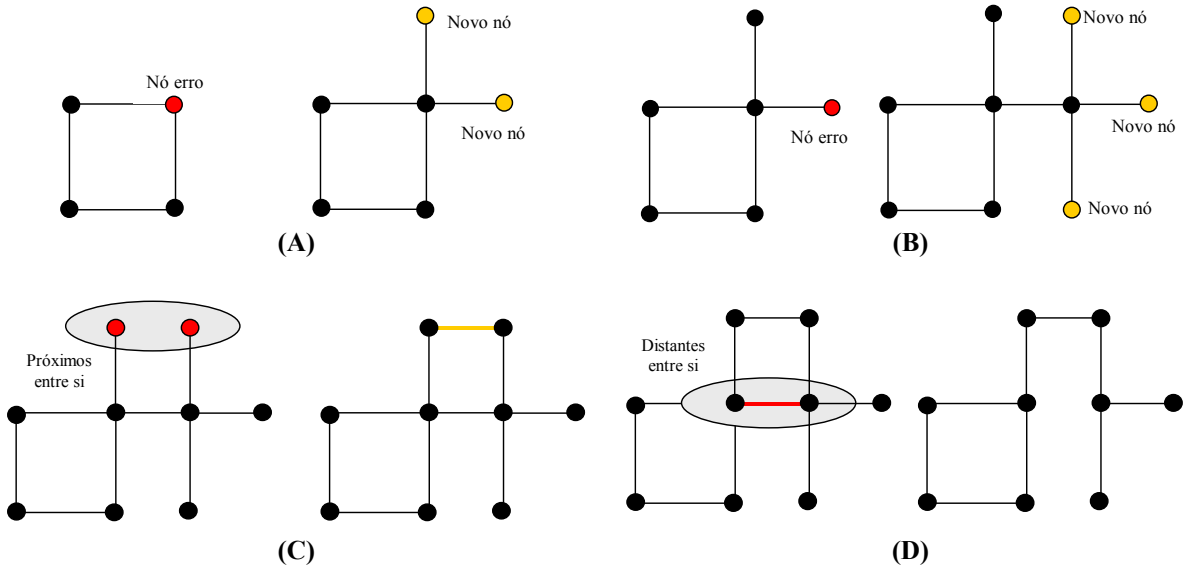


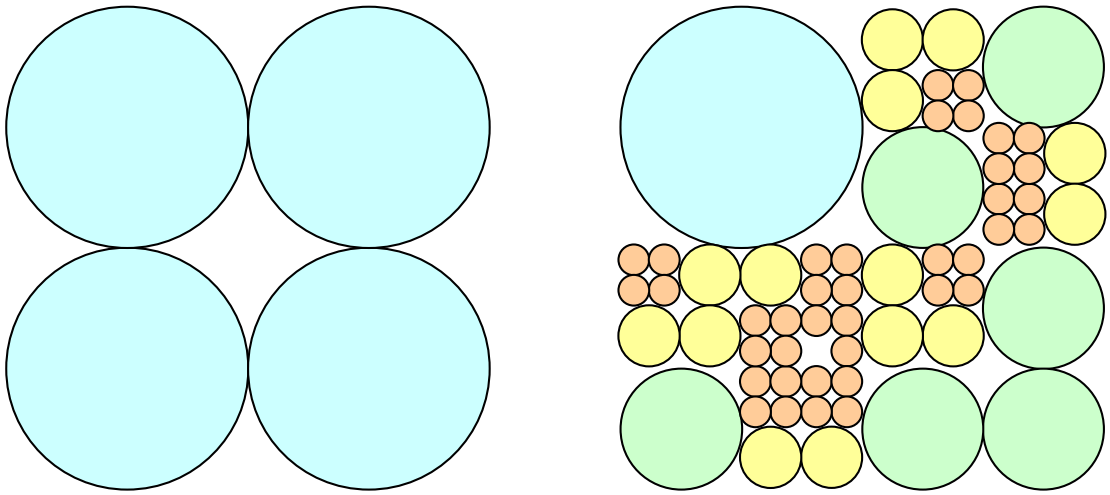
Figura 3-18 – O algoritmo *Incremental Grid Growing*. Em (A) e (B) vê-se a inclusão de novos nós próximos a um nó de fronteira com dificuldade em representar os dados. Em (C) e (D) o algoritmo insere e remove conexões entre neurônios, respectivamente.

Em Alahakoon *et al.* (2000) apresenta-se um algoritmo chamado *Growing SOM* (*GSOM*) bastante semelhante ao IGG, diferindo deste no momento da inicialização dos pesos sinápticos dos neurônios inseridos no arranjo.

Um algoritmo interessante de poda, o PSOM (*Prunning SOM*), é proposto por de Castro e Von Zuben (1999), com os neurônios pouco representativos, indicados segundo um critério estabelecido, sendo removidos da rede e o treinamento reiniciado, tomando como ponto de partida os parâmetros anteriores ao processo de poda. O algoritmo foi definido, entretanto, apenas para mapas unidimensionais e não é diretamente generalizável para dimensões maiores.

Finalmente, em Cho (1997) adota-se um raciocínio inverso no algoritmo *Dynamical Node Splitting SOM*. Neste caso, inicia-se o arranjo com 4 neurônios num arranjo 2×2 que são

treinados conforme o algoritmo tradicional SOM. O próximo passo examina cada neurônio em relação ao número e tipo de padrões pelo qual este é responsável, utilizando para isso um conjunto previamente rotulado. Os neurônios que tentam representar mais de uma classe são subdivididos em um arranjo menor, normalmente 2 × 2. Também neurônios não representativos são removidos do arranjo. O efeito obtido é semelhante a mapas hierárquicos, com a exceção de que a topologia proposta é bastante irregular. A Figura 3-19 apresenta um exemplo da subdivisão do arranjo para este algoritmo durante o processo de treinamento.



(A) **(B)**

Figura 3-19 – Modelo com subdivisão de neurônios. O arranjo inicial (A) é subdividido durante o processo de treinamento, resultando numa estrutura irregular, inclusive com regiões onde os neurônios foram removidos (B).

De forma geral, os modelos derivados do SOM buscam suplantar algumas de suas dificuldades, marcadamente em relação ao arranjo ser rígido e possuir um número predeterminado de neurônios. Entretanto, as aparentes vantagens dos métodos construtivos podem facilmente tornar-se seus principais problemas, como o formato altamente irregular e a dimensão dos arranjos gerados, o que pode dificultar sua interpretação. Também a maioria dos algoritmos construtivos apresenta um custo computacional superior ao do algoritmo SOM original. Como uma referência adicional, em Fritzke (1997) apresenta-se uma comparação bastante instrutiva sobre vários métodos competitivos passíveis de uso em atividades de mineração de dados e classificação de padrões.

3.1.3.3 Outras abordagens

A literatura apresenta diversas outras propostas variantes, onde são utilizadas hierarquias de mapas SOM, métodos para acelerar o treinamento e convergência dos mapas, estratégias de interpretação dos (possíveis) agrupamentos de dados encontrados, e busca de características invariantes.

Um problema comum na área de reconhecimento de padrões ocorre quando um objeto sofre transformações lineares como escalamento, rotação, translação e outras. A maioria dos métodos propostos na literatura tem dificuldade em lidar com tais padrões de entrada, sendo uma abordagem tradicional aplicar filtragem de dados (um pré-processamento) com o objetivo de minimizar o efeito das transformações, de forma que as métricas aplicadas[4] ofereçam resultados aproximadamente constantes para um mesmo padrão. Esta abordagem necessariamente esbarra na escolha adequada dos filtros, o que pode ser bastante difícil numa tarefa de mineração de dados. Para contornar este problema, Kohonen propõe um modelo, o *Adaptive Subspace SOM* (*ASSOM*), no qual os filtros são gerados automaticamente por aprendizado competitivo. Na verdade, o ASSOM é um arranjo de filtros (unidades neurais), cada um especializado em reconhecer uma determinada característica invariante frente a transformações (Kohonen, 1996, 1997; Kohonen *et al.* 1997).

A interpretação de um mapa SOM é normalmente feita através da matriz-U (veja a Seção 3.1.2). Entretanto, esta análise é subjetiva e por vezes é bastante difícil sua interpretação. Uma contribuição para suplantar este problema é formulada por Costa (1999) com a rotulação automática da matriz-U (algoritmo *Self-Labeling SOM*) e, posteriormente, com a construção automática de uma hierarquia de mapas para melhor representar subclasses (algoritmo *Tree-Structured Self-Labeling SOM*). Uma abordagem semelhante a esta é proposta por Suganthan (1999) com o algoritmo *Hierarchical Overlapped SOM*. Outras abordagens hierárquicas são objeto de várias publicações, como Miikkulainen (1990), Koikkalainen (1994) e Alahakoon *et al.* (2000).

---

[4] Invariavelmente, usa-se algum conceito de distância entre os vetores de pesos dos neurônios de uma rede e os vetores representantes dos dados para estabelecer a semelhança entre agrupamentos de dados.

Outras modificações possíveis dizem respeito à velocidade com que o algoritmo busca pelo neurônio BMU e atualiza os pesos sinápticos de seus vizinhos (Kaski, 1999), à possibilidade dos neurônios do mapa não possuírem a mesma dimensão e à utilização de aprendizado supervisionado, dentre outras (Kohonen, 1997). Van Hulle (2000) cita outras formas de algoritmos de mapeamento topologicamente correto semelhantes ao SOM como o modelo de Durbin e Willshaw e o próprio GTM[5]. Kiviluoto & Oja (1997) apresentam inclusive uma versão do SOM com características do GTM, chamada de *S-Map*.

Finalmente, Mao & Jain (1995) e Kraaijveld *et al.* (1995) apresentam modelos de projeção não linear baseados no SOM, bastante semelhantes entre si, com resultados interessantes na análise de dados em espaços de grande dimensão.

3.2 Análise e visualização de dados usando SOM

O algoritmo SOM tem demonstrado ser uma ferramenta bastante robusta e de grande aplicação prática em atividades de mineração de dados. A análise e visualização de dados no SOM se dá fundamentalmente pela análise da matriz-U ou pelo comportamento do arranjo de neurônios, contanto que a dimensão do arranjo seja ≤ 3, conforme já apresentado na Seção 3.1.2. As características mais relevantes podem ser resumidamente assim colocadas:

- **capacidade de representação da estrutura presente no espaço de dados**: o algoritmo SOM realiza uma projeção não linear do espaço de dados de entrada em $\Re^D$ para o espaço do arranjo em $\Re^P$ (executando uma redução dimensional semelhante ao processo de quantização vetorial quando $P < D$) ao mesmo tempo que tenta ao máximo preservar a topologia do espaço original mantendo uma relação de vizinhança entre os neurônios. Veja na Figura 3-6 como a grade elástica dobra-se para representar um conjunto de dados em $\Re^3$. Outras demonstrações encontram-se em Kohonen (1997). Considerando então que o SOM executa um mapeamento topologicamente correto, dados que não estavam presentes no momento do treinamento (mas tomados da mesma distribuição de probabilidade) podem ser avaliados frente ao mapa já adaptado, posto que dados

[5] Veja o Capítulo 4 para mais detalhes sobre o GTM.

semelhantes serão mapeados em regiões vizinhas no mapa. Esta habilidade do SOM pode ser associada com a capacidade de *generalização* (Bishop, 1995). Esta afirmação não é verdadeira na presença de descontinuidades ou curvaturas muito acentuadas no espaço de dados de entrada (Van Hulle, 2000).

- **detecção de agrupamentos**: o algoritmo oferece várias possibilidades para visualização dos agrupamentos e suas intra- e inter-relações. A matriz-U é a mais conhecida e outras possibilidades incluem uso de cor (Kaski *et al.* 1999), análise da contribuição individual de cada fator (característica) que compõe o vetor de dados na matriz-U (Kaski *et al.* 1998b), análise de correlação entre os já postos fatores (Vesanto *et al.* 1998) e outras abordagens para avaliar a existência de agrupamentos (Vesanto & Alhoniemi, 2000).
- **atributos inexistentes**: havendo atributos ausentes nos vetores que representam os dados de entrada, estes são simplesmente ignorados no cálculo do BMU, o que é uma prática melhor que o simples descarte do vetor que representa o objeto (Kaski, 1997). Conforme sugerido experimentalmente por Kaski & Kohonen (1996), há pouco sentido em incluir vetores no processo de treinamento quando a relação entre os atributos não disponíveis de um objeto e o total de seus atributos for maior que 30%.
- **representação de dados extremos**[6] **(*outliers*)**: a maioria dos métodos utilizados em mineração elimina dados extremos porque estes tendem a influir negativamente no processo de adaptação. Quando estes dados são gerados por erros de medição ou por outras deficiências, esta atitude é correta. Entretanto, quando o espaço de dados é esparso (e todos os exemplos de dimensão elevada são considerados esparsos, uma característica conhecida como *"curse of dimensionality"* (Kaski, 1997)), dados extremos não são necessariamente um erro, podendo indicar uma tendência até então desconhecida. O SOM é imune a esse problema, pois dados esparsos serão mapeados em regiões esparsas[7], não causando interferência com o restante dos dados e afetando apenas um neurônio e seus vizinhos (Vesanto, 1997).

---

[6] são dados que estão consideravelmente afastados do restante dos dados do conjunto, sem aparentemente possuir vizinhos próximos (ou seja, uma espécie de "exceção" ou "ponto fora da curva").

[7] veja a Seção 3.2.2 para uma discussão sobre os fatores de ampliação e sua relação com a representação de regiões densas e esparsas.

Apesar de suas qualidades, o SOM possui alguns problemas e o principal é que não há definição de uma função de erro (ou de energia) geral que possa ser minimizada[8], garantindo um estado de convergência ou absorção (estacionário) para o mapa. Huang *et al.* (1998) mostram inclusive que um mapa previamente ordenado pode tornar-se desordenado para conjuntos multidimensionais. As provas de convergência fundamentais existentes na literatura tratam apenas de casos particulares, como Kohonen (1982b) e Ritter & Schulten (1986) para um arranjo unidimensional e Ritter & Schulten (1988) numa tentativa de generalizar o processo através de cadeias de Markov. Entretanto, Erwin *et al.* (1992) demonstraram que tais funções não podem de fato existir em casos genéricos e, principalmente, se o espaço de dados de entrada definir uma função densidade de probabilidade contínua, o que coloca um sério obstáculo à análise matemática do processo de convergência do SOM[9]. Kaski (1997, pg 28) argumenta a favor do SOM que "em aplicações práticas os dados sempre serão discretos e finitos, existindo assim uma função de erro local que pode ser minimizada se for assumido uma função de vizinhança fixa":

$$stress_{SOM} = \sum_{n=1}^{N} \sum_{i=1}^{Q} h_{ci} \left\| \mathbf{m}_i - \mathbf{v}_n \right\|^2 \qquad \textbf{Equação 3-11}$$

O índice *c* é o indicador do BMU, conforme Equação 3-2.

Outras tentativas e análises para casos discretos são estabelecidas por Cheng (1997) e Li & Sin (1998). Uma proposta de função geral para todos os algoritmos de mapas topologicamente corretos é feita por Goodhill & Sejnowski (1997).

3.2.1 Sobre a escolha de mapas

Devido à ausência de fundamentação teórica sólida para o SOM, como já visto, não é claro como deve-se escolher os parâmetros do algoritmo de forma a garantir ou obter um "bom mapeamento" (algumas heurísticas serão apresentadas na Seção 3.2.3). A escolha do "melhor mapeamento" deveria ser, obviamente, por aquele que "melhor representa os dados de entrada" e se poderia imaginar que aquele com menor valor para a Equação 3-11 deveria ser o escolhido. Porém, o valor desta equação normalmente decresce com o

[8] a exemplo da função de *stress* de vários métodos citados na Seção 2.4

aumento do tamanho do mapa e cresce quando aumenta o raio da função de vizinhança, dependendo fundamentalmente da função h_{ci} (Kaski, 1997), o que significa que este valor não deve ser usado como critério único para a escolha.

De fato, este critério é freqüentemente substituído por duas métricas, computacionalmente mais simples e menos dependentes da função h_{ci}. A primeira é o *Erro Médio de Quantização* (*Quantization Error – QE*), que corresponde à média das distâncias entre cada vetor de dados $\mathbf{v}_n$ e o correspondente vetor de pesos $\mathbf{m}_c$ do neurônio BMU. O índice QE é dado pela equação:

$$QE = \frac{1}{N}\sum_{n=1}^{N}\left\|\mathbf{m}_c - \mathbf{v}_n\right\| \qquad \textbf{Equação 3-12}$$

A segunda medida é o *Erro Topográfico* (*Topographic Error – TE*), que quantifica a capacidade do mapa em representar a topologia dos dados de entrada. Para cada objeto $\mathbf{v}_n$ são calculados seu BMU $\mathbf{m}_c$ e o segundo BMU $\mathbf{m}_d$ e o erro topográfico é dado por Kiviluoto (1995):

$$TE = \frac{1}{N}\sum_{n=1}^{N}u(\mathbf{v}_n) \qquad \textbf{Equação 3-13}$$

onde $u(\mathbf{v}_n) = 1$ caso $\mathbf{m}_c$ e $\mathbf{m}_d$ não sejam adjacentes.

A medida QE corresponde à *acuidade*, ou *resolução*, do mapa e é inversamente proporcional ao número de neurônios, ou seja, o erro de representação diminui com o aumento do número de neurônios no arranjo (isto é, a resolução aumenta). Se o arranjo possuir um número muito grande de neurônios (eventualmente maior que a quantidade de objetos a representar) ou se sofrer um processo de treinamento onde o raio de vizinhança torna-se menor ou igual a 1 durante muito tempo, pode ocorrer de os neurônios posicionarem-se praticamente sobre os objetos a serem representados. Neste caso QE $\rightarrow$ 0, mas o arranjo pode estar tão retorcido que a capacidade de representar a topologia dos dados é perdida (TE aumenta). O comportamento de TE nesta situação dependerá também do número de neurônios disponíveis no arranjo: TE aumenta se há poucos neurônios e diminui se há muitos neurônios no arranjo.

---

[9] o GTM é baseado em mapeamentos contínuos e possui uma função de erro (veja detalhes no Capítulo 4).

Quando ambos os valores QE e TE são muito baixos, pode haver suspeita de um fenômeno chamado *sobre-ajuste* (*overfitting*): o SOM, na tentativa de representar o mais fielmente possível os dados e possuindo neurônios suficientes, "dobra-se" de tal forma que acaba representando exatamente os dados, podendo perder sua capacidade de generalização. A Figura 3-20 ilustra esta situação.

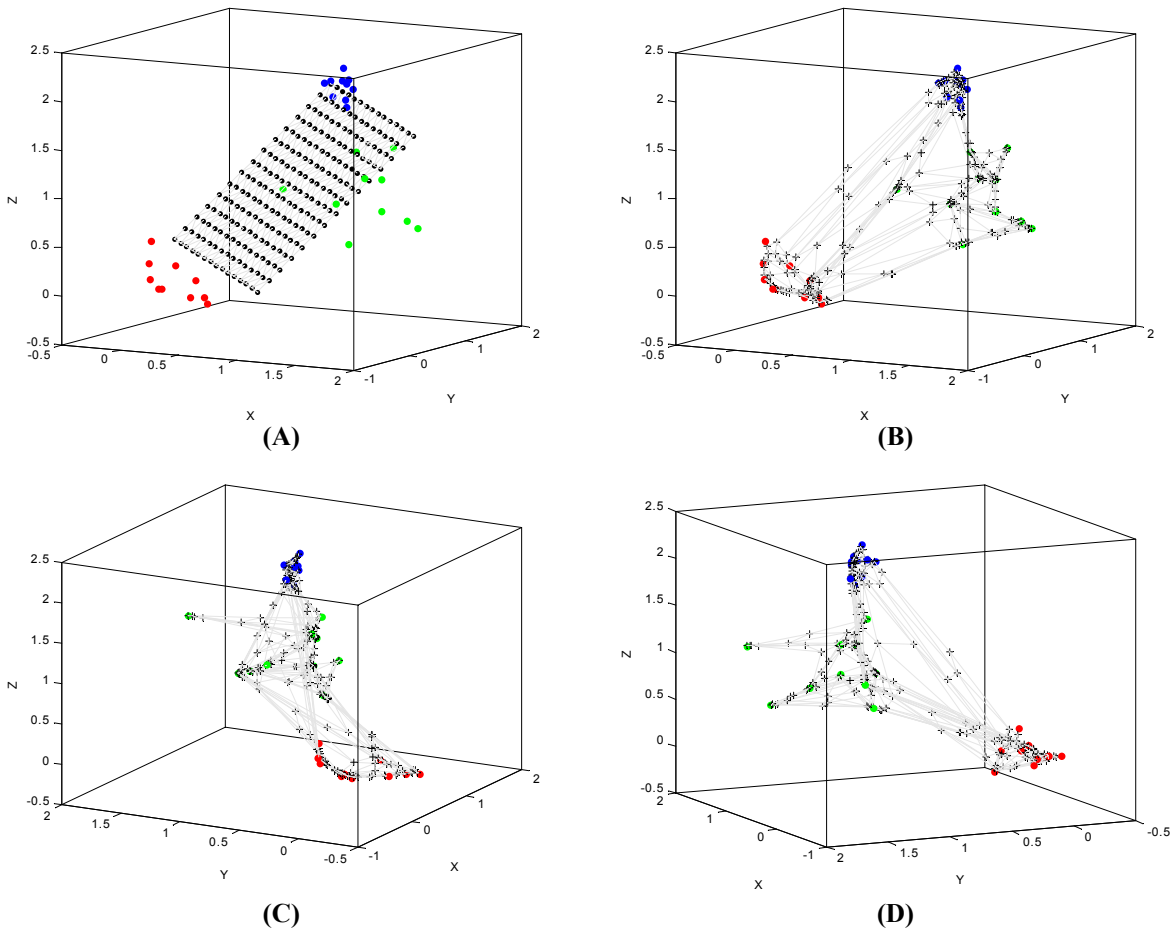


Figura 3-20 – Um conjunto artificial com 30 dados em $\Re^3$ onde um mapa de 15 × 15 neurônios em treinamento de duas fases (inicial com 20 épocas e raio de vizinhança decrescente de 12 a 5 e convergência com 100 épocas e raio de 4 a 1). Em (A) o mapa inicial, antes de qualquer treinamento, com QE = 0,309806 e TE = 0,0. Em (B), (C) e (D), vários pontos de vista para a grade do SOM após o treinamento, com QE = 0,015938 e TE = 0,0. O mapa encontra-se bastante retorcido, sugerindo a ocorrência do fenômeno de sobre-ajuste.

O fenômeno inverso, o *sub-ajuste* (*underfitting*), ocorre quando um mapa é "rígido" demais. Isto pode ocorrer quando há poucos neurônios para representar um número proporcionalmente grande de dados ou se o raio de vizinhança final da função h_{ci} for maior que 1 durante o treinamento. Neste caso, os valores de QE tendem a ser mais altos (isto é,

os vetores de pesos dos neurônios encontram-se, em média, menos próximos dos vetores de dados). Os valores de TE serão baixos se, apesar da "rigidez", a estrutura topológica dos dados for bem comportada, e será maior caso contrário. Em geral, valores muito baixos de TE, associados a valores mais altos de QE, podem sugerir o fenômeno de sub-ajuste. A Figura 3-21 ilustra este caso.

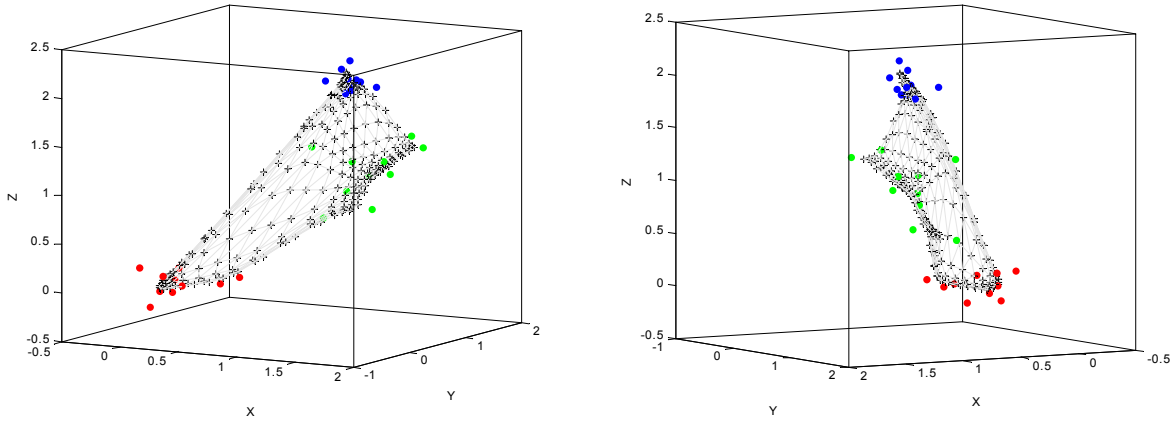


Figura 3-21 – O mesmo conjunto de dados da Figura 3-20, onde um arranjo de 15 × 15 neurônios foi treinado em duas fases (inicial com 5 épocas e raio de vizinhança decrescente de 8 a 4 e convergência com 10 épocas e raio de 3 a 2). Neste caso, os índices finais são QE = 0,088925 e TE = 0,033333. O mapa apresenta-se bastante rígido, sugerindo a possibilidade de ocorrência do fenômeno de sub-ajuste.

Ambas as opções são inadequadas se o objetivo for obter um modelo estatístico dos dados (Svensén, 1998). No caso de mineração de dados, os objetivos de modelagem estatística (com capacidade de generalização), representação da topologia e resolução são objetivos concorrentes entre si (Kaski, 1997) e atualmente não há uma forma segura de efetuar medições capazes de garantir um bom mapeamento com o SOM. Isto significa que os índices propostos podem fornecer indicações sobre os resultados obtidos, mas dificilmente podem ser usados como critérios absolutos.

Uma proposta interessante de combinar as métricas de resolução e representação da topologia em uma só medida é feita por Kaski & Lagus (1996). Nesta métrica, além da distância entre um objeto e seu BMU (erro de quantização) leva-se em conta a distância até seu segundo BMU considerando o caminho mínimo que passa pelo primeiro BMU, numa tentativa de capturar as possíveis descontinuidades do mapeamento.

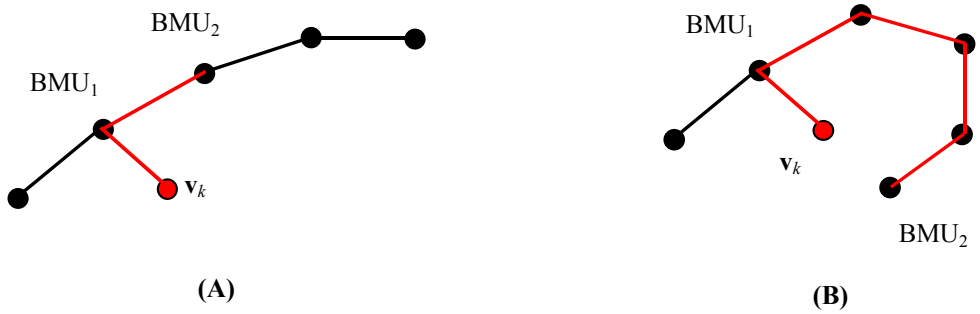


Figura 3-22 – Em (A) um mapeamento contínuo próximo de $\mathbf{v}_k$ enquanto que em (B) há uma descontinuidade local no SOM. Adaptado de Kaski (1997).

Embora promissora, esta métrica é computacionalmente onerosa, podendo tornar-se proibitiva em arranjos muito grandes ou com dimensão maior que 2. Outras referências são relatadas em Vesanto (1997).

3.2.2 Fator de ampliação (*Magnification Factor*)

O termo "fator de ampliação" originalmente refere-se à formação de mapas topologicamente corretos e corresponde à maneira como uma região sensorial é mapeada no córtex cerebral de mamíferos, onde regiões com alta densidade de células receptoras, ou que possuem freqüência de estímulo elevada, são representadas por uma região proporcionalmente maior, se comparada a outras regiões. Tendo o SOM fundamentação biológica, é natural observar que seu arranjo de neurônios "se estica" em regiões de baixa densidade de pontos e "se comprime" em regiões de alta densidade, alocando os recursos da rede conforme a necessidade (veja Figura 3-23). Mais formalmente, Kohonen (1997) define "fator de ampliação" como o "inverso da densidade de probabilidade" dos neurônios do arranjo, $p(\mathbf{w})$ e esta informação pode ser avaliada apenas indiretamente pela matriz-U, uma vez que o arranjo do SOM é discreto (o que é diferente do GTM, que define um mapeamento contínuo. Veja a Seção 4.2.2 para detalhes do GTM).

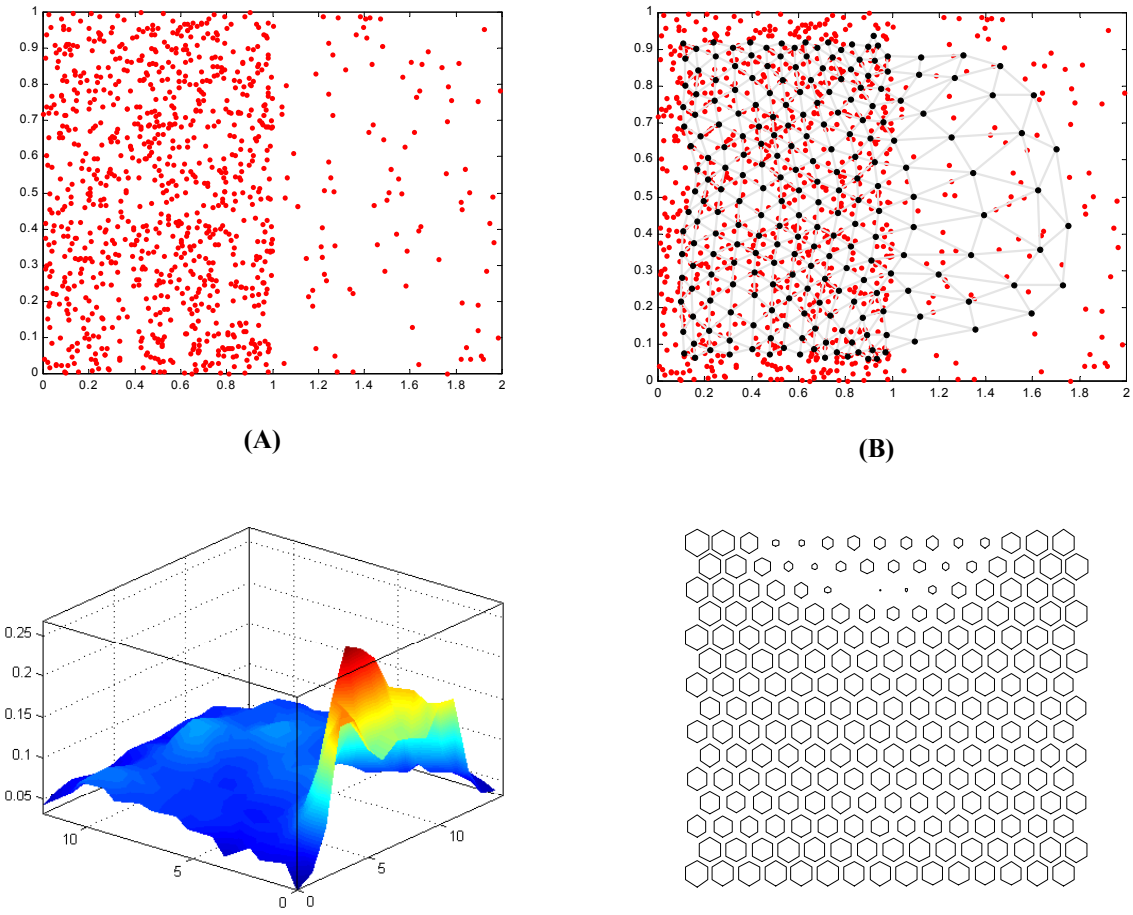


Figura 3-23 – Uma distribuição uniforme com 1000 pontos alocados em duas áreas com diferentes densidades (A). O comportamento desigual da rede (B), um arranjo de 15×15 neurônios treinado sobre os dados, mostra claramente a alocação de mais recursos para a região com maior densidade de pontos. Em (C) vê-se um agrupamento que ocupa praticamente toda a rede correspondendo à distribuição mais densa e um segundo, relativo à distribuição esparsa, melhor identificado em (D) por uma representação em que o tamanho do neurônio é proporcional à proximidade para com seus vizinhos. Percebe-se que os neurônios encontram-se distantes entre si neste último agrupamento.

Diferentemente do que foi suposto inicialmente por Kohonen (1982b), a função densidade de probabilidade $p(\mathbf{w})$ não é uma função linear da densidade de probabilidade dos dados de entrada, $p(\mathbf{v})$. De fato, Ritter & Schulten (1986) mostram que $p(\mathbf{w}) \propto p(\mathbf{v})^{2/3}$ no caso de um arranjo unidimensional. Mais genericamente,

$$p(\mathbf{w}) \propto p(\mathbf{v})^{r} \qquad r = \frac{2}{3} - \frac{1}{3N^2 + 3(N+1)^2} \qquad \textbf{Equação 3-14}$$

onde N é o número de neurônios vizinhos a um BMU (Kohonen, 1997). De qualquer forma, a representação gerada pelo SOM não é *eqüiprobabilística*, ou seja, os neurônios do mapa

não serão ativos com igual probabilidade e, mais importante, o mapeamento gerado tende a privilegiar regiões de baixa densidade e prejudicar outras com alta densidade, não representando fielmente a distribuição de probabilidade dos dados (Van Hulle, 2000).

3.2.3 Considerações sobre os parâmetros

Os parâmetros que regulam o SOM são muitos mas podem ser agrupados basicamente em dois conjuntos: aqueles que definem a estrutura do mapa (suas dimensões, vizinhança e formato do arranjo, raio e tipo da função de vizinhança h_{ci}) e aqueles que controlam o treinamento propriamente dito (se incremental, com a respectiva taxa de aprendizado $\alpha(t)$; em lote, com a função de decrescimento do raio de vizinhança $\sigma(t)$; número de épocas de treinamento). Podem também ser considerados parâmetros adicionais o treinamento em duas fases e a normalização dos dados de entrada (veja a Seção 5.1), comum em atividades de mineração de dados.

Devido novamente à ausência de fundamentação teórica sólida para o SOM, a escolha destes parâmetros não possui critérios mensuráveis. São propostas, essencialmente, heurísticas baseadas no comportamento do mapa e em médias estatísticas de critérios de qualidade. Considerando sempre a finalidade de mineração de dados, pode-se resumidamente colocar as seguintes heurísticas para a obtenção de mapas razoavelmente bem ajustados:

- utilizar algum método prévio de visualização capaz de revelar a estrutura global dos dados e que permita, assim, definir as dimensões do arranjo no sentido de privilegiar tal distribuição. Kohonen (1997) sugere o uso da projeção de Sammon para esse fim e que as dimensões do mapa (proporção entre largura e altura para arranjos planos) devem seguir aproximadamente esta tendência.
- o cálculo para o número de neurônios presentes no mapa possui mais de uma heurística. Caso a quantidade de vetores representando os dados de entrada seja "pequena" (menor que 1000), pode-se fazer o número de neurônios igual ao dos dados (Kaski, 1997). Já a SOM Toolbox propõe $Q = 5\sqrt{N}$ como uma estimativa razoável para o número de neurônios, onde N é quantidade de dados de entrada (Vesanto *et al.* 2000).

- utilizar o arranjo com vizinhança hexagonal, o qual propicia melhor qualidade para inspeção visual de agrupamentos através da matriz-U (Kohonen, 1997).
- utilizar uma função de vizinhança h_{ci} baseada em uma gaussiana, pois esta tende a evidenciar melhor os agrupamentos na matriz-U (Kohonen, 1997).
- utilizar inicialização linear para o arranjo, procurando prevenir torções indesejáveis no arranjo ao longo do treinamento (Kohonen, 1997), embora seja importante lembrar que o SOM não tem garantia teórica de convergência (podendo "convergir" para situações chamadas *estados de absorção* (Ritter & Schulten, 1986, 1988)).
- utilizar duas fases de treinamento, mesmo se houver inicialização linear do arranjo (Kaski, 1997). Prefere-se o algoritmo em lote ao incremental, onde a taxa de aprendizado $\alpha(t)$ é fixa e vale 0,5 na fase inicial e 0,05 na fase de convergência.
- o raio da função h_{ci} é calculado conforme a fase do treinamento: na fase inicial o raio varia de $md/4$ (onde *md* é a maior dimensão do arranjo plano, largura ou altura) e termina em $\max(1, md/4)$. Na fase final de convergência, o raio inicial começa em $md/4$ e termina em 1 (Vesanto *et al.* 2000).
- o número de épocas de treinamento é estimado em $10\,Q/N$ para a fase inicial e $40\,Q/N$ para a fase de convergência (Vesanto *et al.* 2000), onde Q é a quantidade de neurônios do arranjo e N é a quantidade de dados de entrada disponíveis para treinamento.

Mesmo lançando mão das heurísticas acima é recomendado efetuar-se diversos testes com várias configurações do SOM antes de decidir-se por um mapa em particular, o que parece ser um consenso entre a maioria dos pesquisadores.

Capítulo 4

O modelo de Mapeamento Topográfico Gerativo (GTM)

A necessidade de se incluir o modelo GTM (*Generative Topographic Mapping*) nesta dissertação baseia-se na proposta deste em prover uma alternativa melhor fundamentada ao SOM, como também de superar algumas de suas desvantagens (Kohonen, 1997; Van Hulle, 2000). Estas desvantagens devem-se principalmente à não existência de fundamentos teóricos sólidos que definam uma função de energia para o modelo SOM, a qual possa ser minimizada e assim garantir a convergência do modelo (Svensén, 1998). Deste modo, a maior parte do processo de treinamento e análise de um SOM baseia-se em heurísticas (Kohonen, 1997; Kaski, 1997; Kohonen *et al.* 1995a).

Sob este ponto de vista, o GTM é melhor fundamentado que o SOM, uma vez que define matematicamente um modelo de densidade de probabilidade dos dados de entrada (normalmente com dimensão elevada) em termos de um conjunto de variáveis latentes, supostamente capazes de representar os dados num espaço de menor dimensão. Executa-se, assim, um mapeamento que pode ser avaliado visualmente, contanto que o espaço latente tenha não mais que 3 dimensões.

Este capítulo apresenta um estudo sintético do modelo GTM e de sua aplicação a casos de teste e casos práticos, de forma a evidenciar seu potencial como ferramenta de mineração de dados.

4.1 Modelo formal

O GTM, detalhado em Bishop *et al.* (1996b) e Svensén (1998), e com uma breve introdução em Van Hulle (2000), é um modelo que executa um mapeamento paramétrico não linear de um espaço L-dimensional de variáveis (chamadas **latentes**) para um espaço D-dimensional de dados de entrada onde, normalmente, $L < D$. Este mapeamento define

um subespaço **S** (contido no espaço de entrada) que representa o espaço de variáveis latentes segundo a transformação **y(x,W)**, a qual mapeia pontos **x** do espaço latente para pontos **v** no espaço de dados, como ilustrado na Figura 4-1 para o caso em que o espaço latente reside em $\Re^2$ ($L = 2$) e o espaço de dados, em $\Re^3$ ($D = 3$).

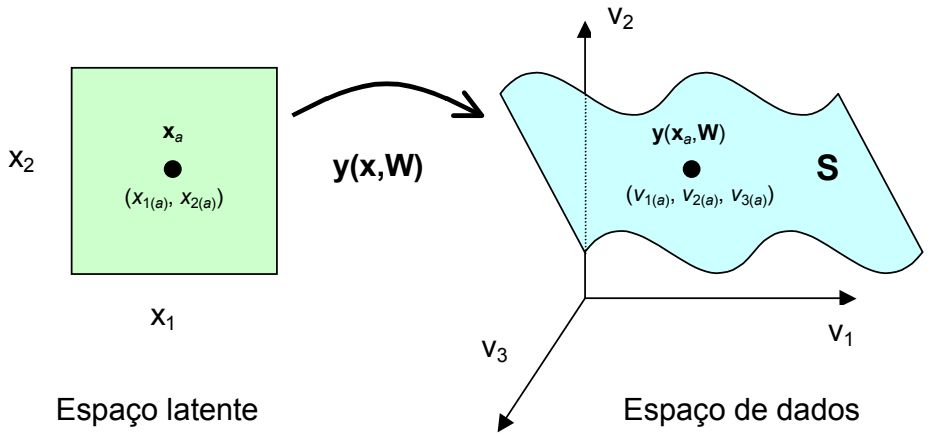


Figura 4-1 - Idéia geral do mapeamento de variáveis latentes: cada ponto do espaço latente (X-espaço, à esquerda) é levado ao espaço de dados (V-espaço, à direita) através de um mapeamento paramétrico não linear y(x,W), o qual define um subespaço S contido no espaço de dados. Cada ponto pertencente a S é resultante da aplicação de y(x,W) sobre um ponto pertencente ao X-espaço. Assim, a transformação y(x,W) leva um ponto $\mathbf{x}_a$ residente no espaço latente e definido pelas suas coordenadas ($x_{1(a)}$, $x_{2(a)}$), para um ponto y($\mathbf{x}_a$,W), pertencente ao espaço S e definido por suas coordenadas ($v_{1(a)}$, $v_{2(a)}$, $v_{3(a)}$) no espaço de dados. Adaptado de Svensén (1998).

A partir desse mapeamento, define-se uma função de distribuição de probabilidade no espaço latente $p(\mathbf{x})$ que irá então induzir uma distribuição de probabilidade $p(\mathbf{y}|\mathbf{W})$ no espaço de dados e ajusta-se o modelo de forma que este consiga uma representação adequada da distribuição dos dados no espaço de entrada. Enquanto os modelos de visualização de dados em espaços de elevada dimensão normalmente determinam um mapeamento ou uma projeção a partir do espaço original para um espaço normalmente bidimensional[1], o modelo GTM faz o inverso, definindo um mapeamento do espaço de variáveis latentes para o espaço de dados (Bishop *et al.* 1996b; Svensén, 1998). Para visualizar o comportamento dos dados (o que deve ser feito sobre o espaço de variáveis latentes), inverte-se o mapeamento pela regra de Bayes, o que dá origem a uma distribuição *a posteriori* no espaço latente (Bishop *et al.* 1996b).

[1] Como a projeção linear PCA (Jolliffe, 1986), a projeção não-linear de Sammon (1969) e o próprio SOM (Kohonen, 1982a), dentre outros.

A hipótese feita pelo modelo GTM é a de que o comportamento do conjunto de dados no espaço *D*-dimensional pode de fato ser expresso por um conjunto menor de atributos[2] (as *variáveis latentes*) através de um mapeamento paramétrico não linear **y(x,W)**. Uma aproximação para esse raciocínio é imaginar que, embora a dimensão do conjunto de entrada possa ser elevada, muitas das variáveis são correlacionadas entre si, resultando num conjunto potencialmente mais simples que pode representar o comportamento dos dados no espaço original (Bishop *et al.* 1996b). Os modelos baseados nesta idéia são chamados *modelos de variáveis latentes* (Bartholomew, 1987).

4.1.1 Modelo de variáveis latentes

A idéia básica de um modelo de variáveis latentes é encontrar uma representação apropriada da função densidade de probabilidade $p(\mathbf{v})$ de um conjunto de dados representados por vetores **v** descritos por seus *D* atributos, $\mathbf{v} = [v_1, v_{2,...,} v_D]^T$, $\mathbf{v} \in \mathbf{V}$, no espaço *D*-dimensional de entrada, em termos de um número *L* de variáveis $\mathbf{x} = [x_1, x_2, ..., x_L]^T$, $\mathbf{x} \in \mathbf{X}$, no espaço *L*-dimensional chamado **latente**. Este mapeamento é realizado por uma transformação paramétrica não linear **y(x,W)** que leva pontos do espaço latente **X** para um subespaço **S** contido no espaço de dados **V**, conforme pode ser visualizado na Figura 4-1.

Se uma distribuição de probabilidade $p(\mathbf{x})$ for definida no espaço latente **X,** então será induzida uma distribuição $p(\mathbf{y}|\mathbf{W})$ no espaço de entrada **V** governada pelo conjunto de parâmetros **W**. Como em geral $L < D$, então a distribuição $p(\mathbf{y}|\mathbf{W})$ estará confinada ao subespaço **S** e será, portanto, singular. Para evitar isso[3], Bishop *et al.* (1996b) adicionaram um modelo de ruído aos vetores do **V**-espaço: um conjunto de gaussianas radialmente simétricas cujos centros estão localizados nos pontos **y(x,W)** (centros estes pertencentes ao subespaço **S**).

---

[2] A mesma hipótese é assumida por modelos como *Factor Analysis* (Bartholomew, 1987) e *Probabilistic PCA* (Tipping & Bishop, 1997) com a diferença de que o GTM executa um mapeamento não linear.

[3] Porque de fato os dados no espaço **V** (de entrada) localizam-se apenas "nas vizinhanças" do subespaço **S** , não pertencendo absolutamente a este.

Estas gaussianas introduzidas no **V**-espaço têm a forma de hiperesferas simétricas posto que todas possuem a mesma variância (Haykin, 1999). A escolha de gaussianas multivariadas radialmente simétricas é comum em redes neurais do tipo RBF (*radial-basis-function*) e modelos baseados em misturas de densidades de probabilidades, principalmente devido a restrições computacionais (Haykin, 1999; Costa, 1999). Uma representação esquemática deste mapeamento pode ser vista na Figura 4-2 para o caso em que $D = 3$.

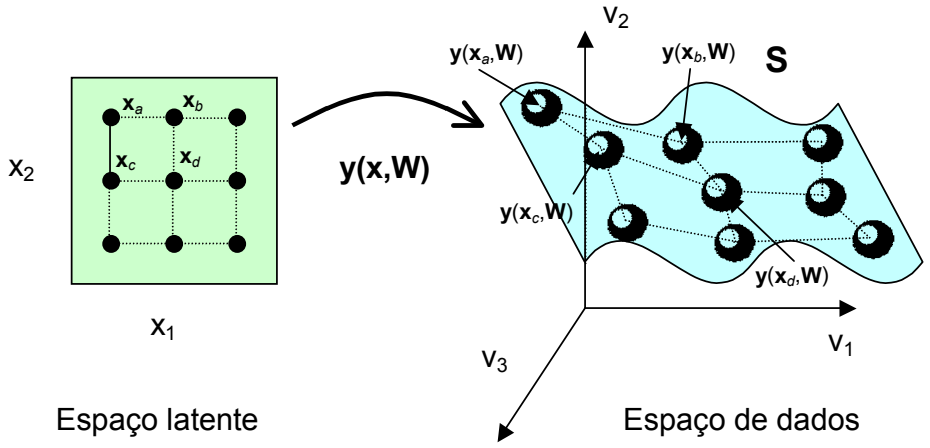


Figura 4-2 – O mapeamento GTM: os pontos do arranjo regular no X-espaço (esquerda) são levados aos centros das gaussianas no V-espaço (representadas por esferas azuis, na direita). Estes centros pertencem ao subespaço S definido pelo mapeamento y(x,W), contido no V-espaço. Note, como exemplo, que os quatro pontos do arranjo regular ($\mathbf{x}_a$, $\mathbf{x}_b$, $\mathbf{x}_c$ e $\mathbf{x}_d$) são mapeados no V-espaço pela transformação y(x,W), respeitando a relação de vizinhança no X-espaço, embora a topologia do subespaço S não seja necessariamente regular. Adaptado de Svensén (1998).

A distribuição de probabilidade para o vetor $\mathbf{v} \in \mathbf{V}$ dados $\mathbf{x}$, $\mathbf{W}$ e considerando as gaussianas citadas com variância σ^2 é, assim, dada por

$$p(\mathbf{v}|\mathbf{x}, \mathbf{W}, \sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^D \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{y}(\mathbf{x}, \mathbf{W}) - \mathbf{v}\|^2\right\} \qquad \textbf{Equação 4-1}$$

Entretanto, a função de distribuição de probabilidade induzida no **V**-espaço, descrita por

$$p(\mathbf{y}|\mathbf{W}) = p(\mathbf{v}|\mathbf{W}, \sigma) = \int p(\mathbf{v}|\mathbf{x}, \mathbf{W}, \sigma)p(\mathbf{x})d\mathbf{x} \qquad \textbf{Equação 4-2}$$

não é, em geral, integrável por meios analíticos (Bishop *et al.* 1996b). Uma possibilidade seria aproximar $p(\mathbf{x})$ pela técnica de Monte Carlo (Bishop *et al.* 1996a), mas esta é computacionalmente cara. Uma escolha mais apropriada para a forma de $p(\mathbf{x})$ é um

somatório de funções delta cujos centros estão situados num arranjo regular definido sobre o **X**-espaço (Bishop *et al.* 1996b):

$$p(\mathbf{x}) = \frac{1}{K}\sum_{k=i}^{K}\delta(\mathbf{x}-\mathbf{x}_k) \qquad \textbf{Equação 4-3}$$

A função delta (ou *delta de Dirac*) (Papoulis, 1991) é assim definida para o problema:

$$\delta(\mathbf{x}-\mathbf{x}_k) = \begin{cases} 0 \text{ se } (\mathbf{x} \neq \mathbf{x}_k) \\ \infty \text{ se } (\mathbf{x} = \mathbf{x}_k) \end{cases} \qquad \textbf{Equação 4-4}$$

Pragmaticamente, isto significa que a função delta será nula em todo lugar, a menos dos K pontos considerados (isto é, nos pontos do arranjo regular dentro do **X**-espaço). Há infinitas possibilidades de escolha para a função delta, mas sua importante propriedade (particularizada para o problema) é:

$$\int f(\mathbf{x})\delta(\mathbf{x}-\mathbf{x}_k)d\mathbf{x} = f(\mathbf{x}_k) \qquad \textbf{Equação 4-5}$$

Isto permite obter a integral de uma função qualquer $f(\mathbf{x})$ calculando seu valor nos pontos $\mathbf{x}_k$. Portanto, a escolha da Equação 4-3 para representar a distribuição *a priori* $p(\mathbf{x})$ permite calcular a distribuição de probabilidade no **V**-espaço (Equação 4-2) como um somatório de K termos:

$$p(\mathbf{v}|\mathbf{W},\sigma) = \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{v}|\mathbf{x}_k,\mathbf{W},\sigma) \qquad \textbf{Equação 4-6}$$

Este é um modelo de mistura de gaussianas em que $\mathbf{x}_k$ representa o centro da k-ésima gaussiana e σ, o espalhamento das mesmas, que neste caso é igual para todas (Van Hulle, 2000). De fato, este modelo é tido como um modelo de mistura de gaussianas *restrito* (Bishop *et al.* 1996b), no sentido de que os centros destas não podem mover-se à revelia uns dos outros por dependerem do mapeamento **y(x,W)**. Isto significa que, dados dois pontos $\mathbf{x}_a$ e $\mathbf{x}_b$ próximos no **X**-espaço, estes serão levados a dois pontos **y($\mathbf{x}_a$,W)** e **y($\mathbf{x}_b$,W)** também próximos no **V**-espaço. Estes dois pontos, **y($\mathbf{x}_a$,W)** e **y($\mathbf{x}_b$,W)**, são os centros das gaussianas contidas no subespaço **S** dentro do **V**-espaço (Figura 4-2).

Se, ainda, este mapeamento for definido através de uma função *contínua*, então têm-se a vantagem adicional de que a topologia do **X**-espaço será naturalmente levada para o

V-espaço, o que significa que o mapeamento necessariamente exibirá uma ordenação topológica (Bishop *et al.* 1996b). A partir desta constatação pode-se conjecturar que se o modelo for ajustado adequadamente à distribuição dos dados no **V**-espaço, então este estará topologicamente ordenado *por construção* (Van Hulle, 2000), embora isso não signifique que, *necessariamente*, o modelo irá revelar a topologia da distribuição no espaço de dados (Svensén, 1998).

4.1.2 O algoritmo EM (*Expectation-Maximization*)

A necessidade agora é de ajustar-se o modelo descrito pela Equação 4-6 no **V**-espaço estimando-se para isso **W** e σ. Considerando um conjunto $\mathbf{D} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$ de N elementos, $\mathbf{D} \subseteq \mathbf{V}$, isto pode ser realizado pela maximização da função de verossimilhança expressa em termos de seu logaritmo, dada por:

$$\ln L(\mathbf{W},\sigma) = \ln \prod_{n=1}^{N} p(\mathbf{v}_n|\mathbf{W},\sigma) = \sum_{n=1}^{N} \ln\left\{\frac{1}{K}\sum_{k=1}^{K} p(\mathbf{v}_n|\mathbf{x}_k,\mathbf{W},\sigma)\right\} \qquad \textbf{Equação 4-7}$$

A maximização do logaritmo da função de verossimilhança (*logL: log likelihood*) como um estimador de parâmetros é discutida em Papoulis (1991), e a maximização da Equação 4-7 como proposta sugere o uso do algoritmo EM (*Expectation-Maximization*) de Dempster *et al.* (1977), posto que o modelo está baseado numa mistura de gaussianas (Bishop *et al.* 1996a,b; Svensén, 1998).

A escolha de uma forma adequada para a transformação **y(x,W)** é feita no sentido de que esta seja, portanto: (1) contínua (para que haja ordenação topológica) e (2) capaz de simplificar o algoritmo EM (Bishop *et al.* 1996a). Para tanto, esta escolha recai sobre um modelo de regressão linear na forma

$$\mathbf{Y} = \mathbf{\Phi W} \qquad \textbf{Equação 4-8}$$

onde **Y** é uma matriz $K \times D$ de pontos no **V**-espaço representando o centro das componentes da mistura (gaussianas, no caso) no D-espaço, **Φ** é uma matriz $K \times M$ de funções-base composta por K vetores $\boldsymbol{\varphi}_k$, $k = 1, ..., K$ e **W** é uma matriz de pesos com dimensão $M \times D$. Cada vetor $\boldsymbol{\varphi}_k$ é composto por M funções-base $\phi_m(\mathbf{x}_k)$, portanto $\boldsymbol{\varphi}_k = [\phi_1(\mathbf{x}_k), \cdots, \phi_M(\mathbf{x}_k)]$. As funções $\phi_m(\mathbf{x})$ são gaussianas radialmente simétricas com

espalhamento igual a σ_ϕ e *fixas* no sentido de que independem do conjunto de treinamento (Van Hulle, 2000). Tanto a quantidade *M* de funções-base, como sua forma (gaussianas) e seu espalhamento σ_ϕ são parâmetros escolhidos antes do ajuste do modelo e *não são modificados* ao longo do treinamento. Note que estas são restrições da ferramenta utilizada nesta dissertação e não referentes ao modelo teórico apresentado.

O primeiro passo do algoritmo EM (o passo E) é calcular a probabilidade *a posteriori* de que um ponto $\mathbf{v}_n$ no **V**-espaço tenha sido gerado por um ponto $\mathbf{x}_k$ no **X**-espaço, $k = 1, ..., K$. Este cálculo é chamado também de *responsabilidade* do ponto $\mathbf{x}_k$ sobre o ponto $\mathbf{v}_n$:

$$r_{kn} = p(\mathbf{x}_k | \mathbf{v}_n, \mathbf{W}, \sigma) = \frac{p(\mathbf{v}_n | \mathbf{x}_k, \mathbf{W}, \sigma) p(\mathbf{x}_k)}{\sum_{i=1}^{K} (\mathbf{v}_n | \mathbf{x}_i, \mathbf{W}, \sigma) p(\mathbf{x}_i)} \qquad \textbf{Equação 4-9}$$

O passo M do algoritmo utilizará as responsabilidades calculadas pela Equação 4-9 para atualizar **W** e σ de forma que cada elemento da mistura de gaussianas "mova-se" na direção dos pontos de sua responsabilidade ao mesmo tempo que σ é reduzido diminuindo a área de interseção entre as gaussianas. A Figura 4-3 ilustra a idéia do modelo GTM, sendo ajustado pelo algoritmo EM, para o caso em que os dados residem num plano. O fato dos dados estarem distribuídos em um plano facilita o entendimento e é generalizável para $D > 2$.

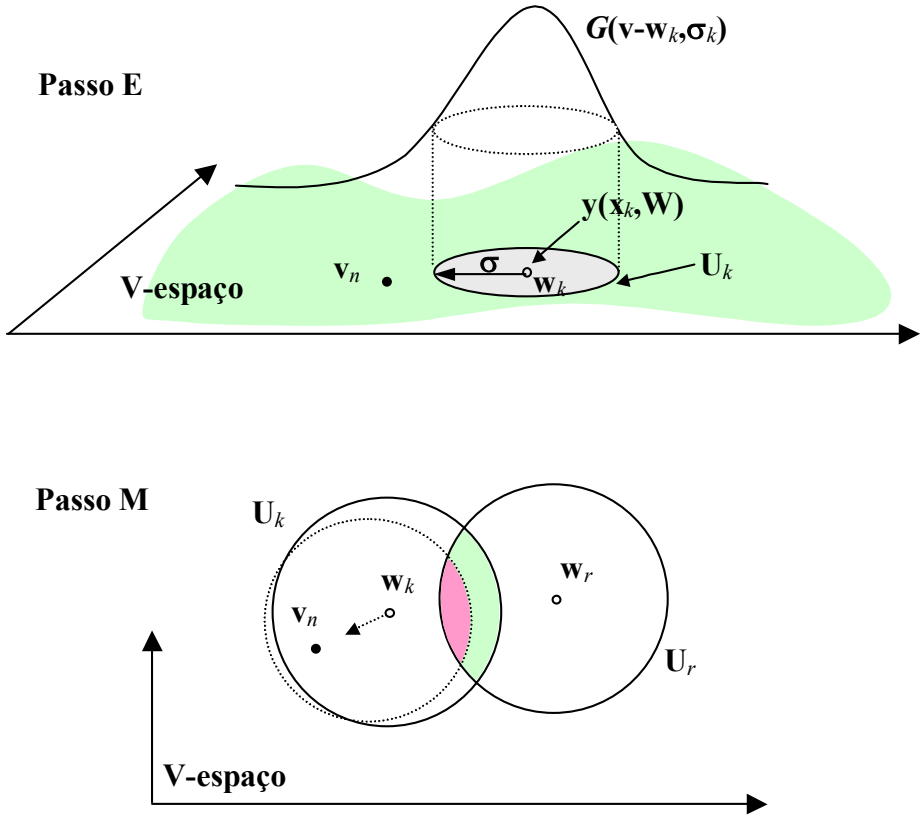


Figura 4-3 – O algoritmo EM em ação. No passo E tem-se a idéia do cálculo de responsabilidade do ponto x_k no espaço latente (Equação 4-9) para o ponto $v \in V$ considerando a gaussiana *G* com centro em $y(x_k,W)$, variância σ^2 e "área de responsabilidade" U_k. O passo M atualiza os parâmetros W e σ de forma a maximizar a responsabilidade de U_k sobre v_n movendo o centro w_k na direção de v_n (linhas pontilhadas) reduzindo também a área de interseção entre U_k e U_r (cuja atualização não está representada na figura).

Para uma descrição detalhada do algoritmo EM consulte Dempster *et al.* (1977) e Bishop (1995). Para este algoritmo aplicado especificamente ao modelo GTM, consulte Bishop *et al.* (1996a,b,c; 1998) e Svensén (1998). Bishop *et al.* (1998) oferecem algumas novas propostas para o algoritmo GTM (como, por exemplo, o uso de modelos de mapeamento que permitam aplicações em dimensões elevadas evitando a intratabilidade computacional). Em Bishop *et al.* (1997a) pode-se encontrar uma aplicação interessante do modelo em séries temporais.

4.2 Análise e visualização de dados usando GTM

A utilização do modelo GTM nessa dissertação tratou exclusivamente de suas possibilidades na análise e visualização de dados. Supondo que o algoritmo EM tenha encontrado valores razoáveis para **W** e σ (ou seja, aqueles que maximizam a Equação 4-7)

então entende-se que foi ajustada uma função de distribuição de probabilidade $p(\mathbf{v} \mid \mathbf{x}_k)$ que pode ser invertida pela regra de Bayes gerando o cálculo da probabilidade *a posteriori* (ou *responsabilidade*) $p(\mathbf{x}_k \mid \mathbf{v})$ dos pontos no **X**-espaço (latente) dados os pontos no **V**-espaço (conforme Equação 4-9). Considerando-se que o espaço latente tenha dimensão $L \leq 3$, pode-se visualizar os dados plotando-se $p(\mathbf{x}_k \mid \mathbf{v})$ diretamente sobre o arranjo de pontos $\mathbf{x}_k$ no **X**-espaço. Para o caso de conjuntos inteiros de dados, há duas possibilidades de visualização (Svensén, 1998):

- a média da distribuição *a posteriori* sobre o **X**-espaço:

$$\mathbf{x}_n^{\text{media}} = \sum_{k=1}^{K} \mathbf{x}_k \, p\left(\mathbf{x}_k \middle| \mathbf{v}_n\right) \qquad \textbf{Equação 4-10}$$

- a moda da distribuição *a posteriori* sobre o **X**-espaço:

$$\mathbf{x}_n^{\text{moda}} = \arg\max_{\mathbf{x}_k} \left\{ p\left(\mathbf{x}_k \middle| \mathbf{v}_n\right) \right\} \qquad \textbf{Equação 4-11}$$

O treinamento e a possibilidade de visualização do modelo GTM são demonstrados, primeiramente, através de um conjunto de dados de entrada gerado pela função $f(x) = 1.5 * \text{sen}(2x) * \cos(2x)$, adicionada de ruído uniforme com amplitude 0,1, conforme Figura 4-4. Todos os gráficos foram gerados através do pacote de programas do Toolbox MatLab®, disponibilizada por Svensén (1999).

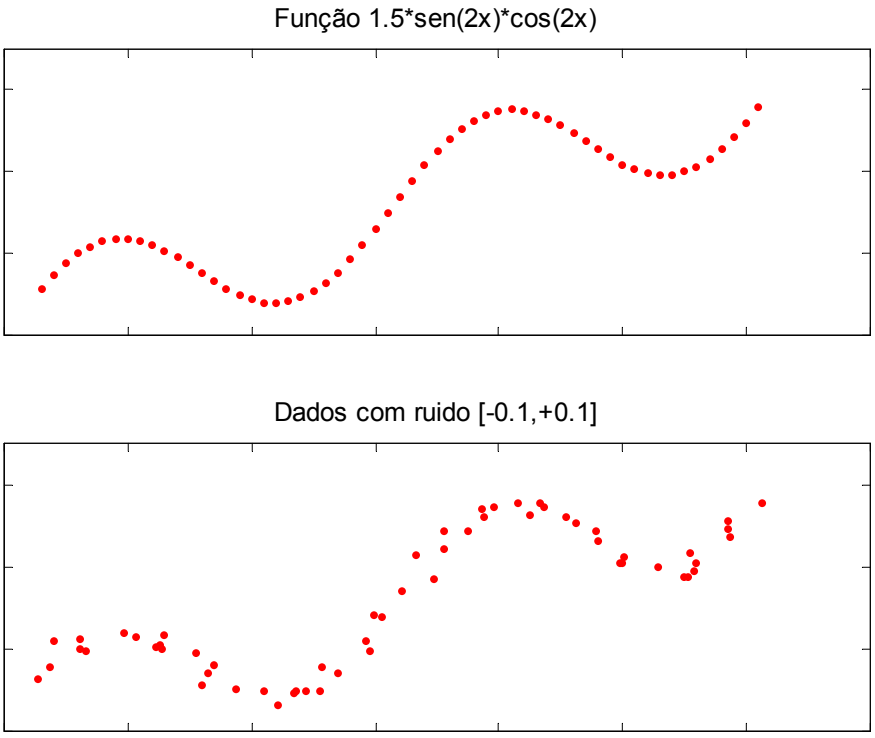


Figura 4-4 – Dados com ruído uniforme adicionado sobre os pontos gerados pela função original (gráfico superior) a serem utilizados para adaptar um modelo GTM.

Na Figura 4-5 têm-se um modelo GTM unidimensional ($L = 1$) com 30 pontos latentes, 6 funções-base ($M = 6$) e espalhamento 2 (veja a Seção 4.1.2 para mais detalhes). O espalhamento significa que o desvio padrão das funções-base, σ_ϕ, inicia valendo 2 vezes a distância entre os centros de duas gaussianas no arranjo gerado. O modelo foi inicialmente ajustado aos dados orientado pelo primeiro componente principal destes dados. A seguir, o treinamento é executado em 15 iterações do algoritmo EM, onde podem ser observadas duas situações intermediárias e a configuração final do modelo, a qual demonstra uma convergência bastante rápida e uma boa aproximação da forma original dos dados.

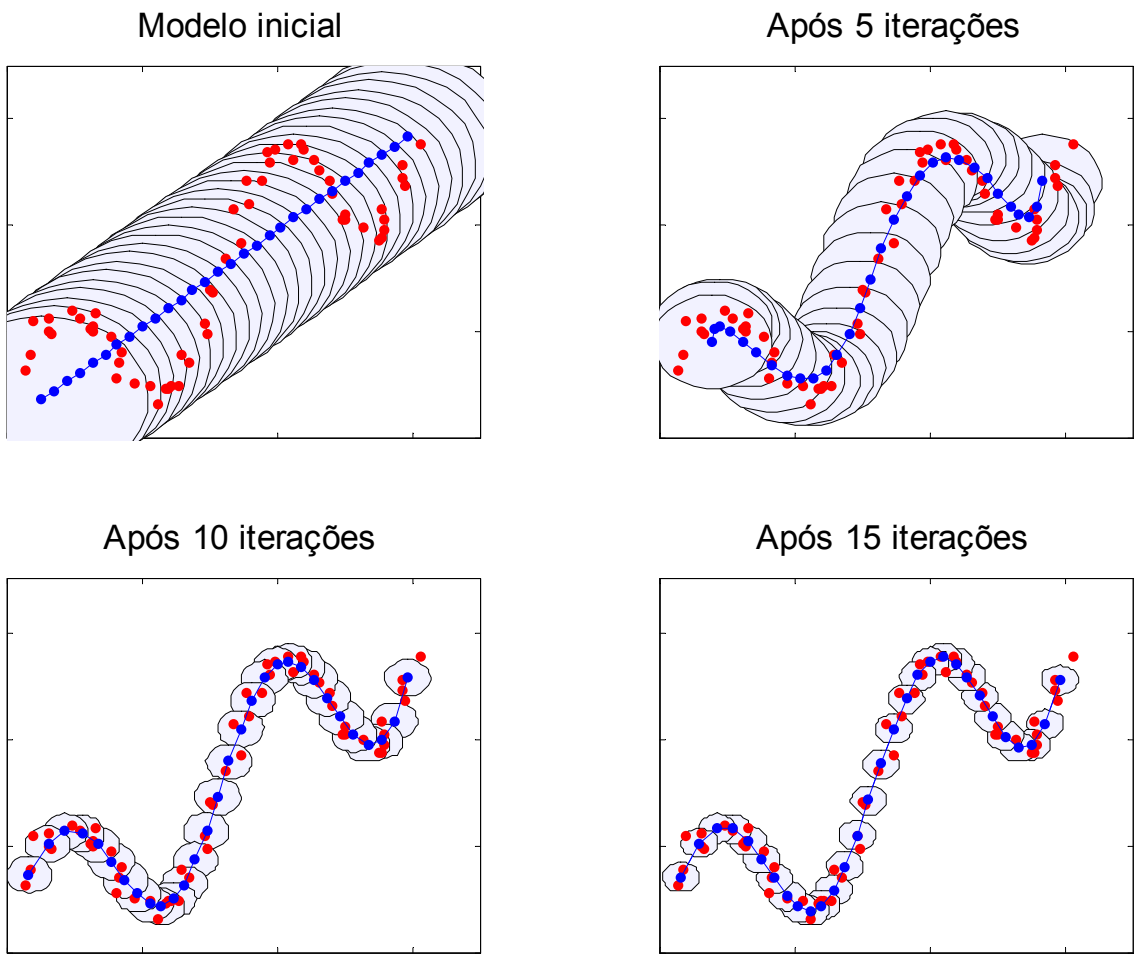


Figura 4-5 – O modelo GTM é treinado em poucas iterações neste exemplo, onde os pontos vermelhos são os dados. Os pontos azuis conectados denotam os centros das gaussianas ordenados segundo o arranjo no X-espaço. O círculo ao redor destes centros representam duas vezes o desvio padrão (2σ) do modelo de ruído sobre o V-espaço. O desvio padrão σ (relativo ao modelo de ruído associado ao mapeamento) não deve ser confundido com o desvio padrão σ_ϕ (relativo às funções-base e constante ao longo do ajuste do algoritmo).

O exemplo a seguir utiliza o conjunto *E.coli* disponibilizado por Blake & Merz (1998). Este banco de dados possui 336 exemplos onde são observadas 7 análises da bactéria *E.coli* para determinar a localização de uma certa proteína na mesma. O conjunto de dados reside, portanto, em $\Re^7$, e há 8 classes com uma distribuição conforme a Tabela 4-1:

Tabela 4-1 - Conjunto de Dados *E.coli* com suas classes

Nome da classe (localização da proteína)	N.º exemplos
cp (*cytoplasm*)	143
im (*inner membrane without signal sequence*)	77
pp (*perisplasm*)	52
imU (*inner membrane, uncleavable signal sequence*)	35
om (*outer membrane*)	20
omL (*outer membrane lipoprotein*)	5
imL (*inner membrane lipoprotein*)	2
imS (*inner membrane, cleavable signal sequence*)	2

Um modelo GTM, com uma grade de 20×20 pontos (latentes) no **X**-espaço, 12×12 funções-base e espalhamento 1,5, foi ajustado por 20 ciclos aos dados. No caso de dados com dimensão maior que 3, não é mais possível a visualização direta e opta-se por observar a média (ou a moda) de $p(\mathbf{x}_k \mid \mathbf{v})$ sobre o espaço latente. A Figura 4-6 e a Figura 4-7 apresentam, respectivamente, as projeções antes e depois do treinamento.

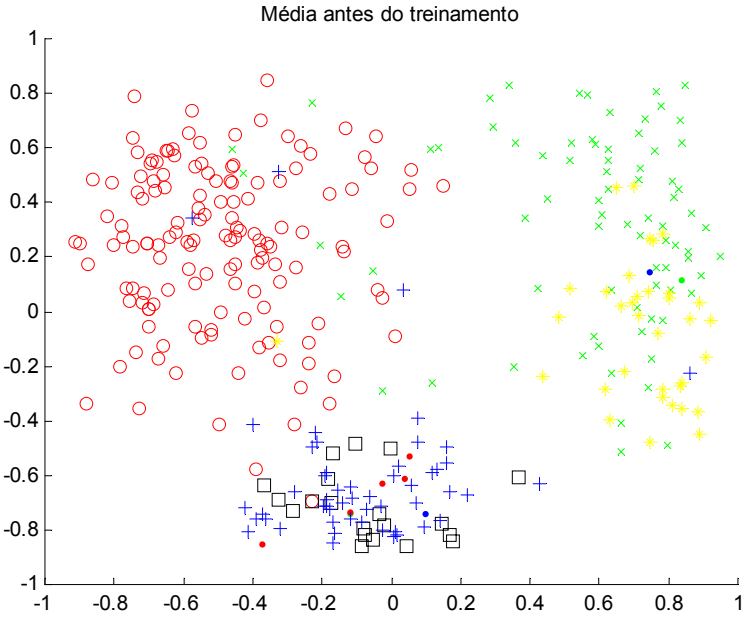


Figura 4-6 – Projeção da média *a posteriori* da distribuição dos dados sobre o espaço latente antes do treinamento do modelo GTM.

As classes estão assim representadas: **cp** (o), **im** (x), **pp** (+), **imU** (*), **om** (□) e as 3 classes restantes, **omL**, **imL** e **imS** por pontos nas cores vermelha, verde e azul, respectivamente.

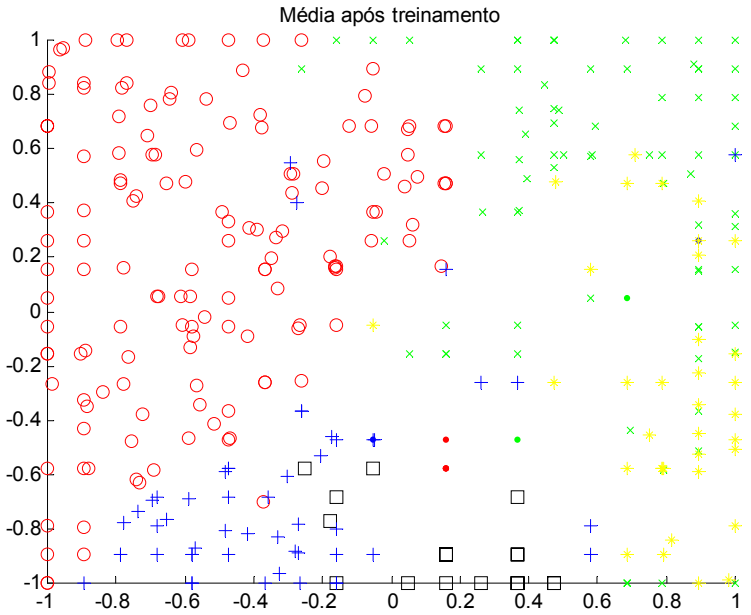


Figura 4-7 – Projeção da média *a posteriori* da distribuição dos dados sobre o espaço latente após o ajuste do modelo GTM.

4.2.1 Sobre a escolha de modelos

Uma vez que o GTM define um modelo de densidade de probabilidade dos dados de entrada e o faz de maneira contínua, a escolha dos parâmetros não afeta a convergência do modelo, mas apenas sua flexibilidade. Com isso, apenas o grau de adaptação do subespaço **S**, definido pelo mapeamento **y(x,W)**, aos dados de entrada, será afetado. Logo, um modelo mais flexível sempre adaptar-se-á melhor aos dados e isto será indicado por um maior valor para o logaritmo da verossimilhança (veja a Seção 4.1.2). Entretanto, um modelo por demais flexível não garantirá a generalização de novos dados provenientes da mesma distribuição de probabilidade, podendo ocasionar o fenômeno de sobre-ajuste (*overfitting*). De forma equivalente, um modelo por demais rígido poderá falhar ao capturar a topologia dos dados, provocando o fenômeno inverso de sub-ajuste (*underfitting*) (Svensén, 1998).

4.2.2 Fator de ampliação

O conceito de fator de ampliação no GTM refere-se à maneira como o subespaço **S** dobra-se e comprime-se de forma a representar a distribuição de probabilidade dos pontos no espaço de dados. Enquanto o fator de ampliação do SOM convencional pode ser descrito apenas indiretamente pela posição relativa dos vetores no arranjo e, portanto, leva

necessariamente a uma análise discreta (Svensén, 1998), o GTM gera um mapeamento contínuo[4] do espaço latente (**X**-espaço) para o espaço de dados (**V**-espaço), gerando o subespaço **S** que é, naturalmente, contínuo. O cálculo do fator de ampliação para o GTM é primeiramente descrito por Bishop *et al.* (1997b), sendo posteriormente comparado ao SOM em Bishop *et al.* (1997c). Uma revisão de ambos é feita por Svensén (1998).

A análise do fator de ampliação sugere que regiões onde **S** encontra-se "esticado" representam regiões distintas no espaço de dados, de forma semelhante à análise feita sobre a matriz-U do SOM. Este conceito é demonstrado através de um exemplo com um conjunto de 500 pontos no total representando dois agrupamentos toroidais entrelaçados no espaço $\Re^3$, conforme Figura 4-8.

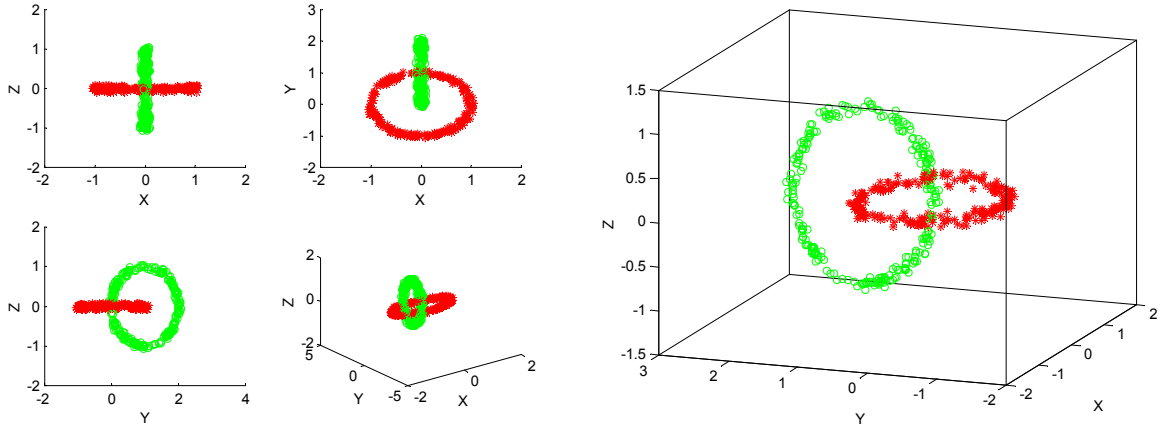
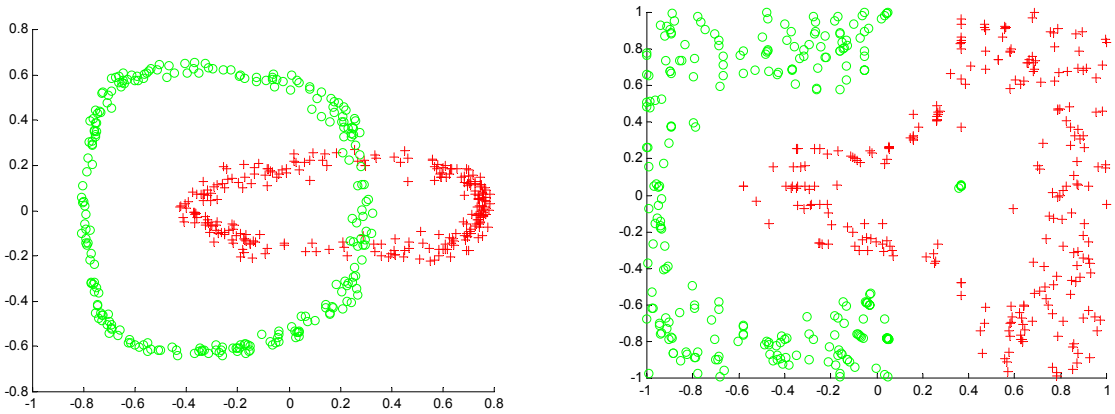


Figura 4-8 – Dois toróides entrelaçados (*Chainlink dataset*), agrupamentos de difícil análise por métodos de projeção por haver sobreposição. Usados por Ultsch & Vetter (1994) para validar a proposta da matriz-U.

A estes dados, foi adaptado um modelo GTM com uma grade de 20×20 pontos no **X**-espaço e uma grade de 10×10 funções-base com espalhamento 1,5 em 25 passos. O gráfico da projeção das médias antes e depois do treinamento é apresentado na Figura 4-9.

[4] Veja 4.1.1 e a definição do mapeamento dada pela Equação 4-8.



(A) **(B)**

Figura 4-9 – Projeção das médias da distribuição dos dados sobre o espaço latente antes (A) e depois do treinamento do modelo (B).

Junto ao modelo adaptado, é calculada uma matriz 40×40 contendo os fatores de ampliação para o modelo GTM. Deve-se notar que a resolução desta matriz *independe* do número de pontos definidos sobre o **X**-espaço (espaço latente) e do número de funções-base. Uma vez que o mapeamento é contínuo, pode-se escolher, em princípio, qualquer resolução para este cálculo (ao contrário do SOM, onde a resolução da matriz-U depende diretamente do número de neurônios utilizados para visualização). A Figura 4-10 representa os fatores de ampliação do GTM ajustado, onde também foram sobrepostas as projeções dos dados (conforme Figura 4-9 à direita).

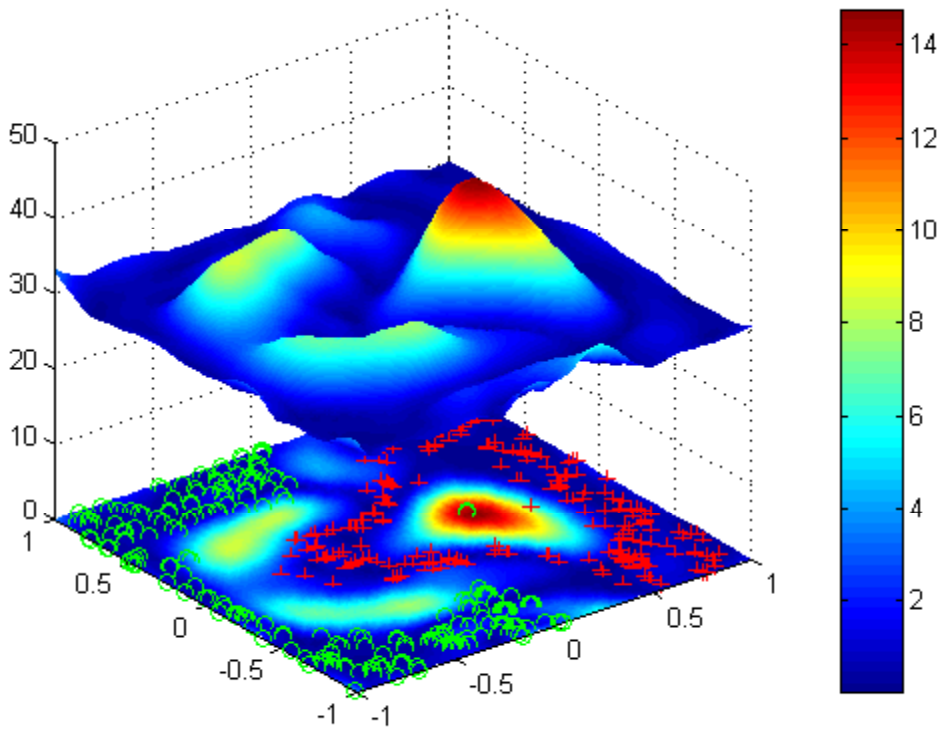


Figura 4-10 – Os fatores de ampliação podem ser vistos como uma superfície onde os picos representam áreas onde o subespaço foi "esticado" e os vales, áreas de alto fator de ampliação. Sobre a projeção plana dos fatores de ampliação vê-se a imagem da média da distribuição dos dados.

As regiões com coloração escura (tendendo ao azul) na Figura 4-10 correspondem a áreas com pouco "esticamento" (alta ampliação) e, portanto, representam regiões com densidade elevada de dados. As áreas tendendo ao vermelho, ao contrário, representam áreas de baixo fator de ampliação (representam poucos ou nenhum ponto). A Figura 4-11 apresenta uma visão 2-D do mesmo mapa, onde pode-se perceber, com mais clareza, os contornos por onde os dados estão representados. A Figura 4-12 oferece uma visão comparável à matriz-U, onde agrupamentos são representados por áreas claras e a separação entre estes, por áreas escuras.

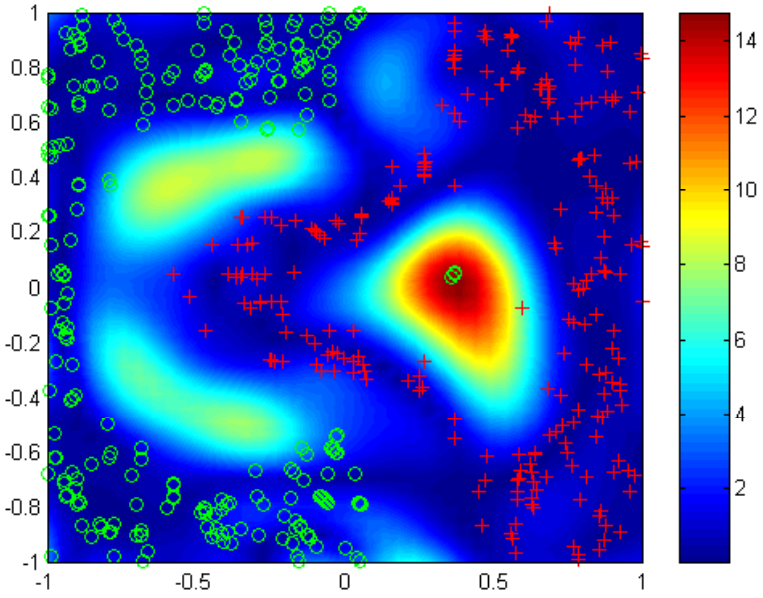


Figura 4-11 – Projeção plana dos fatores de ampliação onde se vê a imagem da média da distribuição dos dados.

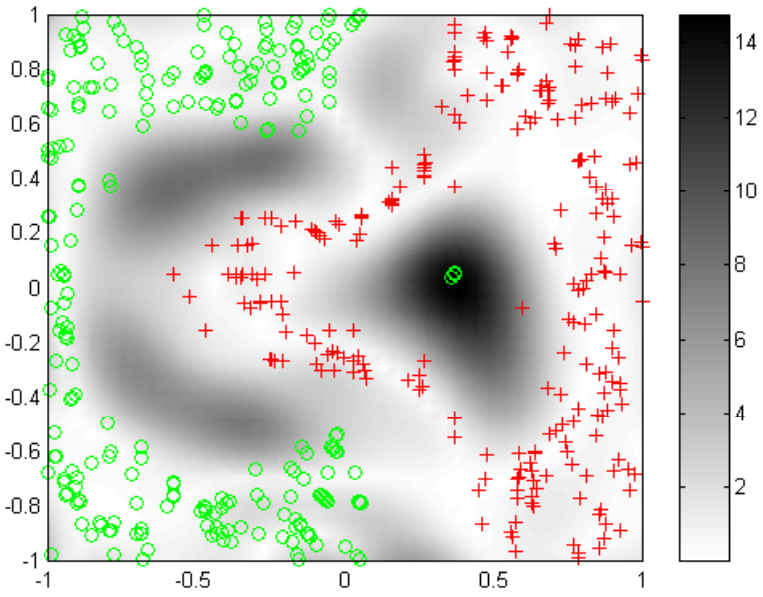


Figura 4-12 – Projeção dos fatores de ampliação com mapa de cores invertido, ou seja, área claras correspondem a agrupamentos de dados (áreas de alta ampliação). Esta imagem é comparável ao que a matriz-U revela sobre o SOM.

4.2.3 Considerações sobre os parâmetros

O comportamento do modelo GTM como definido até aqui[5] é essencialmente afetado pela escolha de alguns poucos parâmetros que, de forma geral, controlam a maleabilidade do subespaço **S** definido pelo mapeamento **y(x,W)**. Mais especificamente, pode-se escolher os seguintes parâmetros de adaptação:

a) a quantidade *M* de funções-base;

b) o espalhamento relativo σ_ϕ das funções-base; e

c) o número *K* de pontos no **X**-espaço.

Para avaliar a influência destes parâmetros sobre o modelo GTM foi utilizado um conjunto de dados representando cores no formato RGB, com valores de 0 a 255 para cada dimensão, compondo um conjunto em $\Re^3$. A Tabela 4-2 apresenta uma pequena amostra do conjunto de 408 exemplos no total.

Tabela 4-2 - Amostra do conjunto de 408 cores definidas pelo valor das componentes RGB.

Cor	*Red*	*Green*	*Blue*
red_1	255	0	0
magenta_1	255	0	255
green_1	0	255	0
forest_green	34	139	34
blue_1	0	0	255
navy_blue	0	0	128
yellow_1	255	255	0
white	255	255	255
black	0	0	0
10%_gray	26	26	26
90%_gray	230	230	230

O número *M* de funções-base afeta diretamente a forma final do subespaço **S**, o qual é adaptado de modo a acompanhar a função de distribuição de probabilidade dos pontos no **V**-espaço. Para um mesmo espalhamento, um pequeno número de funções-base levará a um mapeamento menos flexível, porque um menor número de funções-base necessariamente

[5] Isto é, considerando as escolhas relatadas na Seção 4.1. Para detalhes sobre os parâmetros internos do modelo GTM, consulte Svensén (1998).

limita a forma que o subespaço **S** poderá assumir. Já um número maior de funções-base gerará um mapeamento mais maleável e, portanto, menos suave, o que pode ser verificado na Figura 4-13.

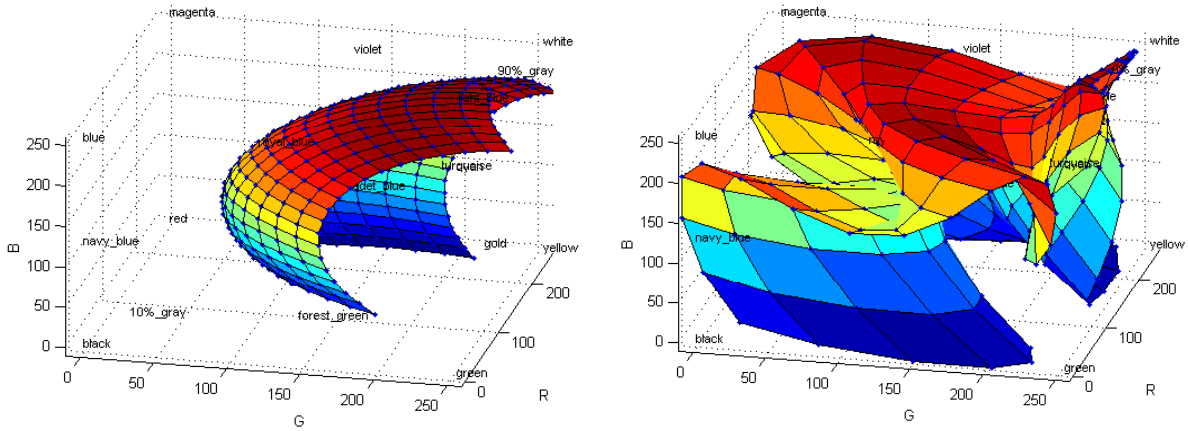
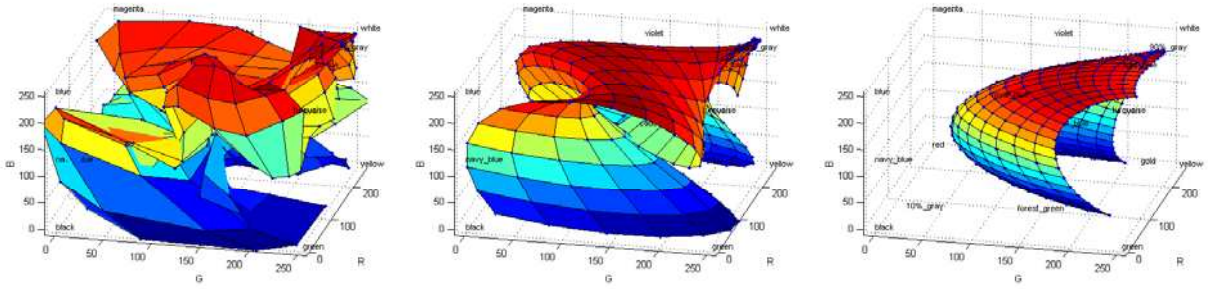


Figura 4-13 – Visualização da forma do subespaço S de modelos GTM adaptados para um conjunto de dados representando as cores no formato RGB. Ambos os modelos utilizaram uma grade de 20×20 pontos no X-espaço e espalhamento 2 (isto é, σ_ϕ inicia valendo 2 vezes a distância entre os centros de duas gaussianas). O exemplo (A), entretanto, utilizou uma grade de 5×5 funções-base, o que resulta num mapeamento mais rígido, com poder de generalização. Já em (B) utilizou-se uma grade de 15×15, o que flexibiliza o modelo, adaptando-o mais proximamente da função de distribuição de probabilidade dos pontos no V-espaço. A escolha por um modelo ou outro depende da necessidade particular do usuário.

O parâmetro σ_ϕ controla a maleabilidade do mapeamento sob o ponto de vista de que, à medida que as funções-base são mais largas, aumenta proporcionalmente a influência de uma função sobre as outras. Dessa forma, pontos próximos no **X**-espaço serão mapeados para pontos proporcionalmente mais próximos no **V**-espaço, o que significa mapeamentos mais rígidos (Svensén, 1998).



(A) **(B)** **(C)**

Figura 4-14 – Visualização da forma do subespaço S onde o espalhamento das funções-base, σ_ϕ, foi variado de 0,5, 2,0 e 4,0, respectivamente de (A) para (C). Os modelos são, de (A) para (C), progressivamente mais rígidos. Novamente, a escolha por um ou outro mapeamento dependerá da necessidade particular do usuário.

O parâmetro relativo a *K*, o número de pontos no **X**-espaço, controla apenas a resolução do mapeamento, ou seja, quantos pontos são "compartilhados" pelas funções-base que executam o mapeamento. Idealmente escolher-se-ia tantos pontos quanto possível, mas isso é computacionalmente proibitivo (Svensén, 1998). O número de pontos, assim, não é um parâmetro essencial para o ajuste do modelo e Bishop *et al.* (1997b) recomendam que aproximadamente 100 pontos no espaço latente (bidimensional) devam pertencer ao espaço definido por $2\sigma_\phi$ do centro de cada função-base.

Capítulo 5

Aplicações em Mineração de Dados

Este capítulo busca aplicar alguns dos métodos para mineração de dados apresentados em capítulos anteriores, com ênfase nos algoritmos SOM e GTM. São utilizados, nos testes, alguns conjuntos de dados públicos e um conjunto de dados de um caso real sobre classificação de estilos de aprendizagem de alunos universitários ingressantes na Universidade São Francisco (USF), nos cursos de Análise de Sistemas, Ciência da Computação e Engenharia (Elétrica, Mecânica, Civil e de Computação), no ano de 1998. Este último conjunto de dados apresenta uma aplicação prática numa área tradicionalmente não explorada por métodos de mineração de dados, a Educação.

Este capítulo não tem a intenção de eleger a "melhor ferramenta" para a atividade de mineração de dados, mas de oferecer subsídios para avaliação e interpretação dos resultados obtidos pelas diversas ferramentas aplicadas, considerando os testes efetuados. Os testes objetivaram estudar o comportamento de algumas destas ferramentas em conjuntos de dados específicos, nos quais há variação da dimensionalidade, quantidade e tipo de dados disponíveis (discretos, contínuos, binários etc.). Com este intuito, a avaliação do desempenho da ferramenta utilizou, em alguns casos, os rótulos atribuídos previamente aos dados. Numa aplicação real, os rótulos deveriam ser obtidos através do uso da ferramenta.

5.1 Introdução

Como já visto no Capítulo 2, a atividade de KDD (*Knowledge Discovery in Databases*) envolve um conjunto de tarefas que devem ser levadas em consideração muito antes de se iniciar a etapa de mineração de dados. Mais especificamente, Fayyad *et al.* (1996a) consideram as seguintes tarefas no processo de KDD:

- *Análise de requisitos*: estudo do domínio do problema e definição dos objetivos do KDD;
- *Seleção de dados*: criação de um conjunto de dados que potencialmente contém o conhecimento buscado;
- *Preprocessamento*: modelagem e remoção do ruído, decisão sobre dados inexistentes e amostras extremas (*outliers*);
- *Transformação*: redução dimensional do conjunto (eliminando variáveis com alta correlação entre si, por exemplo), detecção de atributos não-informativos;
- *Mineração de dados*: aplicação de um ou vários métodos de mineração de dados, incluindo aqueles já discutidos em capítulos anteriores;
- *Interpretação dos resultados*: avaliação dos padrões e resultados obtidos, possivelmente realimentando todo o processo novamente.

Quando o domínio de um problema é bem entendido e são feitas perguntas específicas de natureza estatística sobre dados previamente rotulados, torna-se relativamente fácil oferecer respostas, por exemplo: "quais as medidas de roupas para vestir 90% da população masculina". Estas respostas serão tão mais fundamentadas à medida que se têm mais dados sobre o domínio em questão. Algoritmos para classificação de padrões são alguns dos métodos utilizados em mineração de dados (Capítulo 2) e tema de obras como Duda *et al.* (2000) e Bishop (1995). Paradoxalmente, quando não se compreende muito bem o problema e busca-se extrair algum conhecimento útil do conjunto de dados, a análise torna-se proporcionalmente mais complicada à medida que mais e mais dados são disponibilizados. A situação se agrava quando estes dados são multidimensionais, o que é normalmente o caso a ser considerado nesta dissertação. Deve-se considerar, também, que a grande maioria dos algoritmos aplicados em mineração de dados necessitam de valores numéricos para representar as características dos dados, como o SOM por exemplo. Isto torna a representação de dados em forma de texto ainda mais complexa, pois não há, ainda, um método único ou definitivo capaz de representar informações lingüísticas de maneira adequada.

Nesta situação, é fundamental a preparação adequada dos dados a serem examinados na etapa de mineração para que os métodos aplicados possam obter resultados relevantes e

interpretáveis. Quando pouco ou nada se sabe sobre a importância de cada um dos atributos na expressão da informação contida em uma base de dados, deve-se buscar mecanismos para evitar que um atributo domine o processo de agrupamento, sobrepujando a contribuição de outros que possam ser até mais importantes, e também que atributos de grande relevância tenham seu papel minimizado. Este processo recebe vários nomes na literatura, como "*normalização*" (Kaski & Kohonen, 1996; Kohonen, 1997), "*padronização*" (Sarle, 1997) ou ainda "*redimensionamento*" (Vesanto, 1997). Este processo consiste, normalmente, em executar uma transformação linear aplicada sobre cada atributo dos vetores $\mathbf{v} = [v_1, ..., v_D]^T \in \mathbf{V}$ da forma

$$v_d^{novo} = \frac{\left(v_d^{velho} - \overline{v_{\cdot d}}\right)}{\sigma(v_{\cdot d})}, \quad d = 1,...,D \qquad \textbf{Equação 5-1}$$

onde v_d^{velho} é o valor original do *k*-ésimo atributo do vetor **v**, $\overline{v_{\cdot d}}$ e $\sigma(v_{\cdot d})$ são, respectivamente, a média e o desvio padrão do *d*-ésimo atributo, considerando todos os vetores de **V**. Esta transformação garante que todos os atributos tenham média 0 e variância 1, tomando o conjunto de dados completo. Variantes do processo de "normalização" incluem transformações não lineares, utilização de lógica nebulosa e outras (Vesanto, 2000). A SOM Toolbox (Alhoniemi *et al.* 2000, comentada em Vesanto *et al.*, 1999; 2000) oferece uma ampla gama de possibilidades. De modo inverso, pode-se ampliar a influência de um ou mais atributos, redimensionando-os de forma a aumentar a variância em relação aos outros componentes, aumentando assim sua contribuição no processo de mineração. É interessante observar, entretanto, que o processo de normalização nem sempre leva a resultados satisfatórios, como será visto posteriormente nos testes realizados neste capítulo.

A tarefa proposta é justamente obter algum conhecimento a partir de dados que podem ou não conter relações que definam este conhecimento. Posto assim, é inconcebível, atualmente, optar por apenas um método de mineração, pois o desconhecimento pode facilmente levar a resultados tendenciosos e nenhum método é suficientemente genérico e eficiente para todos os casos. Tampouco é visível, em curto prazo, um sistema de KDD totalmente autônomo, i.e., sem interação humana (Vesanto, 1997).

O que se propõe é que aquele que ingressa na tarefa de minerar conhecimento deve lançar mão de um elenco diverso de ferramentas factíveis para o conjunto de dados em questão,

seguindo-se a análise dos resultados obtidos. Este capítulo dedica-se, assim, a aplicar algumas das ferramentas de mineração de dados para avaliar os resultados obtidos, tecendo comentários onde julgou-se conveniente. Embora nenhuma comparação entre as ferramentas tenha sido intencionalmente feita, os resultados apontam para uma superioridade do SOM e do GTM em relação às ferramentas mais tradicionais de mineração de dados. Esta percepção, e os critérios considerados para esta afirmação, serão melhor evidenciados através dos testes efetuados neste capítulo.

Deve ficar claro, no entanto, que o compromisso entre resultados e custos é um parâmetro difícil de ser avaliado, particularmente por envolver etapas recursivas de refinamento e tomada de decisão a partir da própria realimentação dos resultados. Um estudo mais refinado acerca destes mecanismos está além do escopo desta dissertação.

Os testes realizados neste capítulo desconsideram qualquer conhecimento prévio sobre os domínios dos problemas, tornando a tarefa de mineração de dados particularmente difícil. A normalização (Equação 5-1) foi o único pré-processamento efetuado e também supõe desconhecimento de atributos dominantes ou irrelevantes. Os atributos prévios só são utilizados para avaliar os resultados gerados.

5.2 Conjuntos de dados públicos

Há uma grande variedade de bancos de dados disponíveis para utilização e que, principalmente, servem de parâmetro de comparação entre métodos e algoritmos passíveis de uso em mineração de dados. Sarle (1997) mantém um *FAQ* (*Frequently Asked Questions*) sobre redes neurais com várias indicações de conjuntos de dados para utilização em pesquisa. Os dados utilizados nesta seção foram obtidos de Blake & Merz (1998) e Hettich & Bay (1999) e são detalhados nas seções subseqüentes. Todos os exemplos aqui realizados demonstram a aplicação de algumas das ferramentas de mineração de dados, com ênfase no SOM e no GTM, em tarefas típicas de mineração de dados, como definir agrupamentos, representar a topologia do espaço de dados, executar redução dimensional etc. Para estes dois últimos métodos, foram realizados 5 testes, com diferentes parâmetros, utilizando dados normalizados segundo a Equação 5-1, e outros 5 testes utilizando dados

não normalizados. Há, portanto, um total de 10 experimentos por conjunto de dados para as ferramentas SOM e GTM.

Na preparação dos conjuntos de dados, a *normalização*, como já dito, é um procedimento comum quando não se conhece a influência dos atributos componentes dos vetores de dados. A intenção aqui foi verificar se, de fato, este é um procedimento eficaz, considerando total desconhecimento da relevância dos atributos dos dados. Observou-se que, nos exemplos testados, este procedimento nem sempre levou a resultados satisfatórios, especialmente no caso do GTM.

Para cada experimento, o "melhor resultado" observado é apresentado no texto. Os critérios adotados para a escolha, no caso das ferramentas SOM e GTM, levaram em consideração algumas características:

- embora os conjuntos sejam previamente rotulados, esta informação não foi disponibilizada para nenhuma ferramenta. Este conhecimento apenas ofereceu subsídios para avaliar-se a capacidade das ferramentas em evidenciar agrupamentos, exibir similaridades entre os dados e representar a topologia da estrutura dos dados.
- algumas métricas, associadas à qualidade do mapeamento executado pelas ferramentas, foram consideradas. Estas métricas oferecem indicações sobre a forma como as ferramentas se adaptaram ao conjunto de dados. Cabe lembrar que estas métricas não são, de fato, critérios definitivos para a escolha do "melhor resultado", mas apenas indícios sobre "como" o modelo está adaptado, devendo ser tomadas com cuidado e o máximo de conhecimento possível sobre os dados.

No caso do SOM, o critério adotado para escolher o "melhor resultado" é, de fato, uma heurística, obtida a partir dos resultados de todos os experimentos efetuados. Foi observado que, tomando a configuração sugerida pelas heurísticas propostas na literatura consultada (algumas destas sumarizadas na Seção 3.2.3), os mapas não forneceram resultados adequados à tarefa proposta de mineração de dados. Mais especificamente, não foram evidentes a separação de agrupamentos, a exibição de similaridades entre os dados e a representação da topologia da estrutura dos dados. A avaliação visual dos mapas gerados, utilizando-se inclusive dos rótulos de dados, mostrou que, em geral, as configurações mais

adequadas às tarefas acima propostas são aquelas com menor *erro topográfico TE (Topographic Error)*. Este indicador expressa a capacidade do mapa em representar a topologia dos dados de entrada. Um menor valor para este, em geral, indica uma melhor adaptação do mapa à topologia dos dados no espaço de entrada.

Dessa forma, o "melhor resultado" considerado, para o SOM, é a configuração que apresenta, dentre as três com *menor erro topográfico TE*, aquela com o *valor intermediário para o erro de quantização QE (Quantization Error)*. Foram descartadas as configurações que apresentaram TE = 0,0 devido à possibilidade de sobre-ajuste (*overfitting*) ou sub-ajuste (*underfitting*). A Seção 3.2 discute mais detalhes sobre estas métricas e termos. Os resultados obtidos constam da Tabela 5-1, onde estão indicadas as opções candidatas (as configurações com os menores valores para TE) e a opção escolhida segundo o critério proposto de valor QE intermediário.

No caso do GTM, e diferentemente do SOM, existe uma métrica diretamente associada à forma como o modelo adaptou-se para representar o conjunto de dados, o *logaritmo da verossimilhança* (*logL: log likelihood*) (veja a Seção 4.2.1 para detalhes da métrica). Um valor comparativamente baixo para o logL significa um modelo mais rígido e genérico, menos adaptado à topologia dos dados; e um valor mais alto significa um modelo mais flexível e melhor adaptado. Assim, foi realizado um conjunto de testes preliminares com diferentes valores para a quantidade *M* de funções-base (um dos parâmetros que controla a flexibilidade do modelo) e para o número *K* de pontos no **X**-espaço (os pontos do espaço latente, que controlam a resolução ou acuidade do modelo). Dentre os testes, foi escolhida a combinação que resultou no maior logL (a mais promissora). A partir desta configuração, foi gerado um segundo conjunto de testes, onde o espalhamento das funções-base (σ_ϕ, o outro parâmetro que controla a flexibilidade do modelo) foi variado dentro de um intervalo e, novamente, foi escolhida a configuração com o maior valor para logL. Os resultados obtidos constam da Tabela 5-2. Os objetivos buscados foram os mesmos do SOM, ou seja: capacidade de exibir a separação de agrupamentos e similaridades entre os dados, e a representação da topologia da estrutura dos dados.

Tabela 5-1 – Resultados dos testes para o algoritmo SOM. A primeira configuração de cada conjunto corresponde àquela sugerida pela literatura, marcada em cinza ( ). Marcado em laranja ( ) estão as configurações candidatas (os menores TEs) e em verde ( ), a configuração escolhida, com QE intermediário. É interessante notar que, no caso do SOM, a normalização dos dados de entrada parece ser uma medida de preparação dos dados efetivamente útil. A "fase 1" corresponde à ordenação inicial do mapa e a "fase 2", ao processo de ajuste fino.

Conjunto *"Glass"*								
Configuração					Normalizados		Não normalizados	
Dimensão	Fase 1		Fase 2		QE	TE	QE	TE
	Épocas	Raio	Épocas	Raio				
13 × 07	3	3→1	20	1→1	1,053061	0,032710	0,646701	0,014019
15 × 10	3	3→1	20	1→1	0,902501	0,023364	0,542068	0,032710
15 × 15	5	4→1	25	1→1	0,786558	0,032710	0,461934	0,009346
20 × 20	5	4→1	30	1→1	0,590428	0,018692	0,322924	0,023364
10 × 15	5	4→1	25	1→1	0,912956	0,023364	0,548254	0,042056
Conjunto "*Ionosphere*"								
Configuração					Normalizados		Não normalizados	
Dimensão	Fase 1		Fase 2		QE	TE	QE	TE
	Épocas	Raio	Épocas	Raio				
13 × 07	3	3→1	20	1→1	2,178962	0,052980	1,379441	0,033113
15 × 10	5	3→1	25	1→1	2,126871	0,105960	1,271332	0,019868
15 × 15	5	5→1	25	1→1	2,059725	0,072848	1,226585	0,052980
20 × 20	5	5→1	25	1→1	2,103357	0,105960	1,242943	0,079470
20 × 15	5	6→1	50	1→1	2,048209	0,112583	1,193551	0,119205
Conjunto *"Letter"*								
Configuração					Normalizados		Não normalizados	
Dimensão	Fase 1		Fase 2		QE	TE	QE	TE
	Épocas	Raio	Épocas	Raio				
34 × 21	1	3→1	20	1→1	1,752808	0,090750	3,947901	0,106000
10 × 12	3	3→1	10	1→1	2,377945	0,053000	5,385121	0,067750
15 × 20	4	3→1	25	1→1	2,042608	0,066750	4,615182	0,066500
35 × 25	5	4→1	25	1→1	1,672770	0,078250	3,809190	0,090000
35 × 35	5	5→1	50	1→1	1,577102	0,075000	3,550347	0,080000
Conjunto *"Zoo"*								
Configuração					Normalizados		Não normalizados	
Dimensão	Fase 1		Fase 2		QE	TE	QE	TE
	Épocas	Raio	Épocas	Raio				
10 × 05	10	3→1	20	1→1	1,797125	0,000000	0,886697	0,039604
12 × 12	10	4→1	25	1→1	1,065179	0,009901	0,512363	0,000000
15 × 15	5	5→1	30	1→1	0,738246	0,009901	0,352025	0,000000
25 × 25	5	5→1	30	1→1	0,078623	0,029703	0,044015	0,000000
30 × 30	5	6→1	50	1→1	0,018711	0,039604	0,010417	0,000000

Tabela 5-2 – Resultados de testes para o algoritmo GTM. A configuração escolhida é marcada em verde ( ) e foi escolhida por apresentar o maior valor para o logaritmo da verossimilhança (logL: *log likelihood*).

Conjunto *"Glass"*								
Configuração					Normalizados		Não normalizados	
Pontos Latentes	Funções Base	σ_ϕ	Fator Regular,	Ciclos	logL inicial	logL final	logL inicial	logL final
20 × 20	12 × 12	0,5	0,001	20	-2789,064596	915,800723	-2181,862260	1892,058223
		0,8			-2789,159165	835,115372	-2182,118304	2216,060739
		1,0			-2789,160897	334,573759	-2182,115457	2251,391164
		1,2			-2789,161735	17,859099	-2182,098154	1992,381968
		1,5			-2789,161460	-178,146317	-2182,084192	978,747905
Conjunto *"Ionosphere"*								
Configuração					Normalizados		Não normalizados	
Pontos Latentes	Funções Base	σ_ϕ	Fator Regular,	Ciclos	logL inicial	logL final	logL inicial	logL final
20 × 20	5 × 5	0,5	0,001	20	-10454,061492	-6950,079323	-6322,526190	-2800,523789
		0,8			-10454,350225	-6945,893287	-6322,957436	-2713,640534
		1,0			-10454,314579	-6916,048634	-6322,901308	-2790,464667
		1,2			-10454,235483	-7094,487398	-6322,779992	-2930,607539
		1,5			-10454,134128	-7397,516997	-6322,627791	-3256,015781
Conjunto *"Letter"*								
Configuração					Normalizados		Não normalizados	
Pontos Latentes	Funções Base	σ_ϕ	Fator Regular,	Ciclos	logL inicial	logL final	logL inicial	logL final
20 × 20	15 × 15	0,5	0,01	20	-375343,157007	-213355,993066	-601065,259221	-421189,757011
		0,8			-375343,788475	-213981,307798	-601065,814642	-421974,864931
		1,0			-375343,781302	-216082,979350	-601065,806797	-425263,131257
		1,2			-375343,815103	-221477,162122	-601065,838740	-432261,422393
		1,5			-375343,834184	-232863,610369	-601065,857499	-444173,382195
Conjunto *"Zoo"*								
Configuração					Normalizados		Não normalizados	
Pontos Latentes	Funções Base	σ_ϕ	Fator Regular,	Ciclos	logL inicial	logL final	logL inicial	logL final
20 × 20	10 × 10	0,5	0,1	15	-2509,736132	2605,796554	-1340,307864	3690,002907
		0,8			-2509,737977	1574,052638	-1340,314445	2786,767860
		1,0			-2509,738384	1273,325780	-1340,285637	3091,523145
		1,2			-2509,738586	763,148253	-1340,262405	3216,356743
		1,5			-2509,738549	167,207870	-1340,249468	1755,359556

5.2.1 Conjunto *"Glass"*

O conjunto *"Glass"* possui um total de 214 objetos definidos por vetores de atributos compostos pelo índice de refração (*RI*) e composição química (*Na, Mg, Al, Si, K, Ca, Ba* e *Fe*) de amostras de vidro de uso doméstico e industrial. O conjunto reside, portanto, em $\Re^9$ e é divido em 7 classes, conforme a Tabela 5-3:

Tabela 5-3 - Conjunto de Dados *"Glass"* com suas classes

Classe	Descrição	N.º exemplos
1	Janelas de edifícios (laminado)	70
2	Janelas de veículos (laminado)	17
3	Janelas de edifícios (não laminado)	76
4	Janelas de veículos (não laminado)	--
5	Vidro de recipientes	13
6	Louça de mesa (copos etc.)	9
7	Lâmpadas e faróis	29

Deve-se observar que, de fato, não há exemplos para objetos da classe 4 e, portanto, o conjunto contém 6 classes. Os atributos são descritos por valores reais e, apesar da dimensão do espaço de dados de entrada não ser elevada, a utilização de alguns métodos de projeção não é capaz de revelar muito da informação estrutural contida neste conjunto, como pode ser verificado na Figura 5-1.

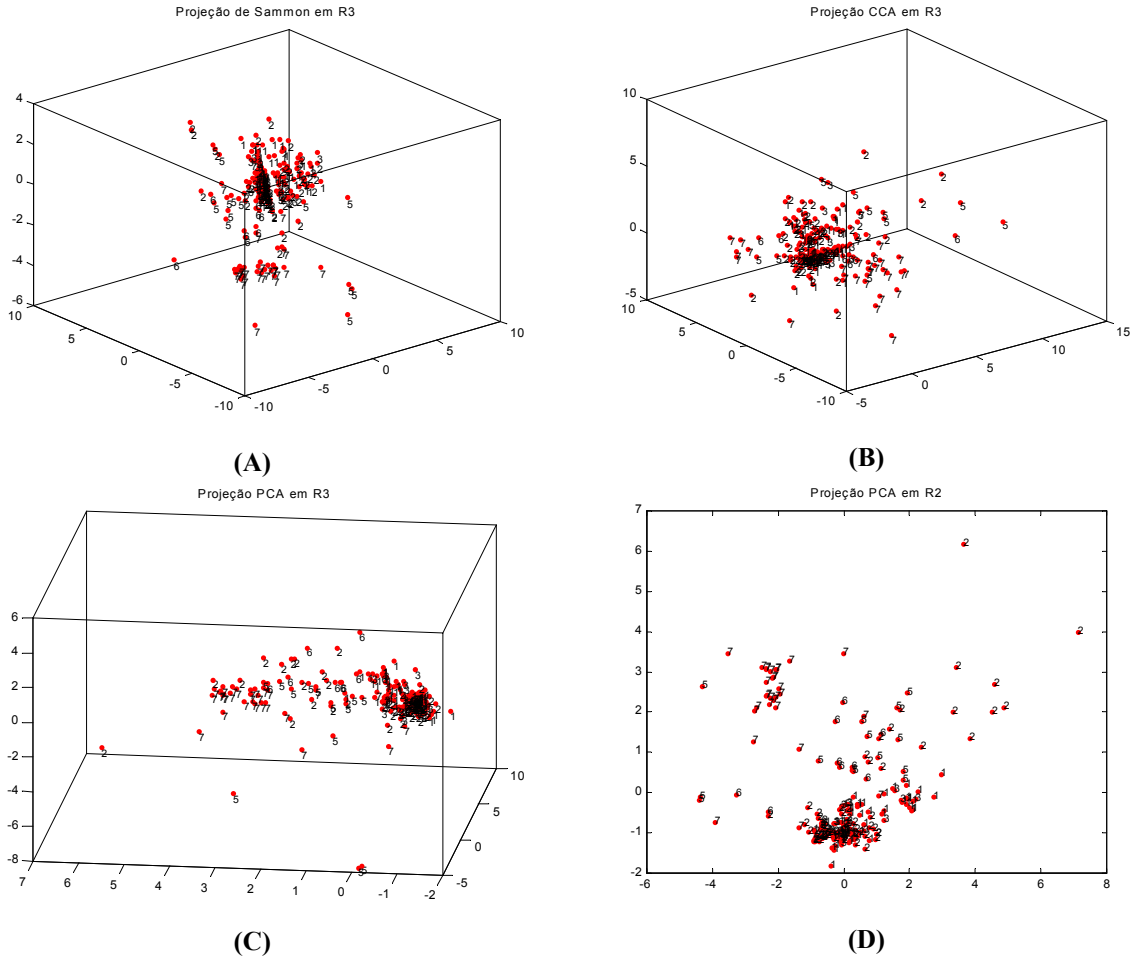


Figura 5-1 – Métodos de projeção de Sammon, CCA e PCA do conjunto *"Glass"*. Embora alguns agrupamentos possam ser observados (a projeção de Sammon sugere 2 agrupamentos), nenhum dos exemplos oferece boa separação entre os grupos. Os dados foram rotulados conforme a Tabela 5-3 com o intuito de possibilitar uma avaliação visual dos resultados. Esta informação não esteve disponível para nenhuma das ferramentas durante o processo de adaptação das mesmas.

Embora os métodos de projeção sofram uma degradação rápida de sua qualidade à medida que aumenta o número de atributos que descrevem os conjuntos, eles podem ser usados para fornecer "pistas" sobre o formato aproximado do conjunto e sobre alguns agrupamentos que sejam facilmente separáveis. Nestes casos, é possível separar estes conjuntos previamente identificados e trabalhar apenas com os dados que não puderam ser avaliados utilizando-se outros métodos. A projeção de Sammon (Figura 5-1-A) indica que o conjunto em questão possui formato aproximadamente hiperesférico, o que sugere um arranjo quadrado para o SOM.

O modelo SOM escolhido dentre os testes realizados é um arranjo plano de 20×20 neurônios com vizinhança hexagonal, inicializado linearmente ao longo da distribuição dos dados (normalizados) e treinado pelo algoritmo *"batch"* em duas fases: a primeira, curta e com vizinhança maior de atualização de pesos (5 épocas com vizinhança regredindo de 4 a 1) e a segunda, longa e com vizinhança mais restrita (30 épocas com vizinhança fixa em 1), conforme Tabela 5-1.

A Figura 5-2 apresenta várias matrizes-U, uma para cada atributo do conjunto, onde pode-se observar as tendências de agrupamento de cada atributo em particular (e, portanto, sua contribuição para o resultado geral). Esta possibilidade de uso da ferramenta SOM permite descobrir possíveis correlações entre atributos observando as matrizes-U individuais de cada atributo: figuras semelhantes indicam correlação positiva, enquanto figuras com padrão de cor invertido indicam correlação negativa. A análise visual de correlação é discutida em Vesanto & Ahola (1999). Com essa inspeção visual, é possível identificar atributos que podem ser removidos (reduzindo assim a dimensão dos vetores de dados), ou atributos que apresentam grande influência no resultado final.

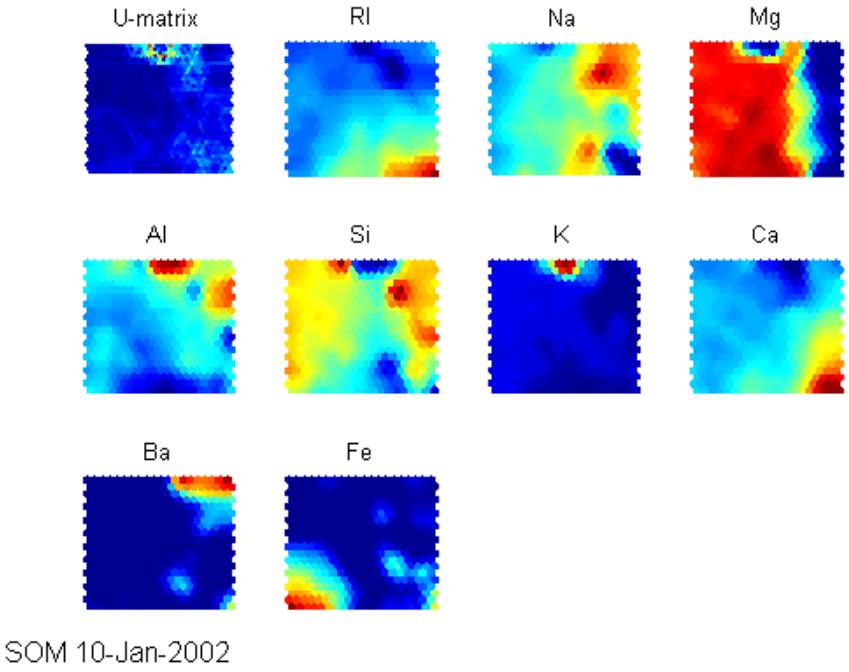


Figura 5-2 – A matriz-U geral composta pelos 9 planos (figura no canto superior esquerdo) e as matrizes-U relativas a cada atributo: índice de refração (*RI*) e elementos de composição química. O tom azul representa proximidade dos vetores de pesos dos neurônios enquanto que o vermelho significa o oposto, i.e., maior dissimilaridade. Numa inspeção visual pode-se observar que as matrizes-U dos atributos RI e Ca são bastante parecidas entre si. Esta semelhança sugere haver uma correlação positiva entre estes dois fatores (calculada posteriormente e igual a 0,8104).

A Figura 5-3-A apresenta a matriz-U conforme proposta por Ultsch & Siemon (1989) e descrita na Seção 3.1.2.2. Uma versão interpolada de cores pode ser vista em (B), onde percebe-se com mais clareza a existência de (possíveis) agrupamentos. De fato, pode-se notar uma área escura no 2^{o} quadrante do mapa, onde objetos das classes **1** e **2** foram agrupados, em sua maioria, pelo algoritmo. Isto pode ser também verificado pela matriz-U como uma superfície (C), onde os vales representam os agrupamentos (cor azul) e as elevações, a separação entre os mesmos (quanto maior a altura entre os grupos, tanto maior sua dissimilaridade). A figura (D) apresenta o número de objetos para o qual cada neurônio do arranjo SOM é responsável (isto é, para quantos objetos ele é o BMU) como um hexágono de tamanho proporcional a este número.

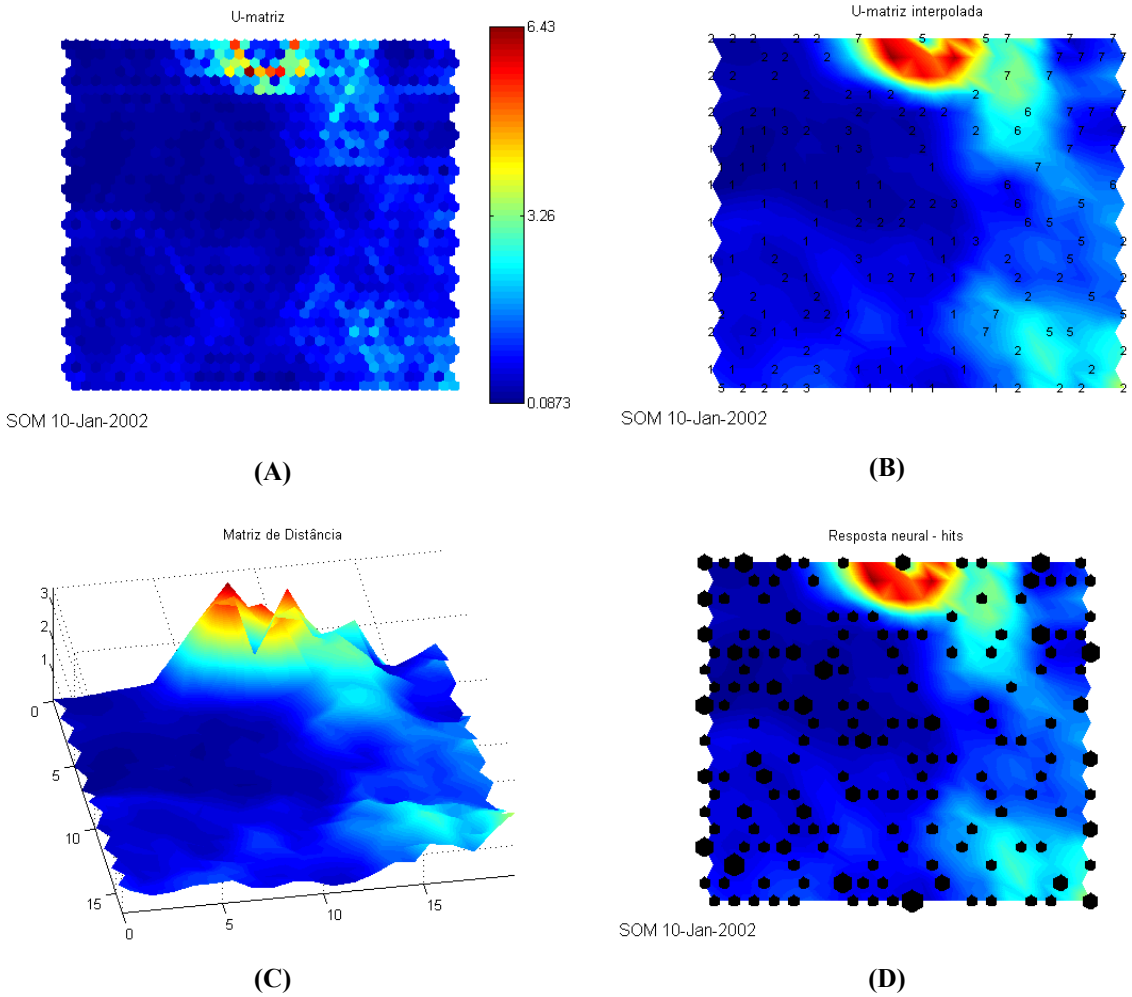


Figura 5-3 – Matriz-U original e interpolada (A e B). Em (C) pode-se observar a matriz-U como uma superfície que denota o fator de magnificação do arranjo SOM. A interpretação de (B) e (C) sugere a existência de dois grandes grupos, um deles no "vale" situado à esquerda da elevação da matriz-U e outro à direita. Em (D), observa-se a freqüência de resposta de cada neurônio. Esta possibilidade de análise oferecida pela ferramenta sugere uma distribuição regular dos dados em relação aos seus neurônios BMU, com poucos neurônios inativos e aparentemente nenhum neurônio com sobrecarga, sugerindo que a quantidade de neurônios no arranjo é compatível com o conjunto de dados.

A documentação do conjunto *"Glass"* sugere que as amostras das classes **1**, **2**, **3** e **4** são mais semelhantes entre si do que se comparadas às amostras das classes **5**, **6** e **7**, o que de fato pode ser verificado na Figura 5-4-A, com a sobreposição do arranjo SOM e de um código de cores que retrata a dissimilaridade entre os agrupamentos, dada pela diferença de cor entre os mesmos. A Figura 5-4-B utiliza também da matriz de distância, vista como uma superfície, para evidenciar ainda mais a existência de dois grandes conjuntos (embora 3 possam ser interpretados), reforçando a percepção de separação gerada pelas cores.

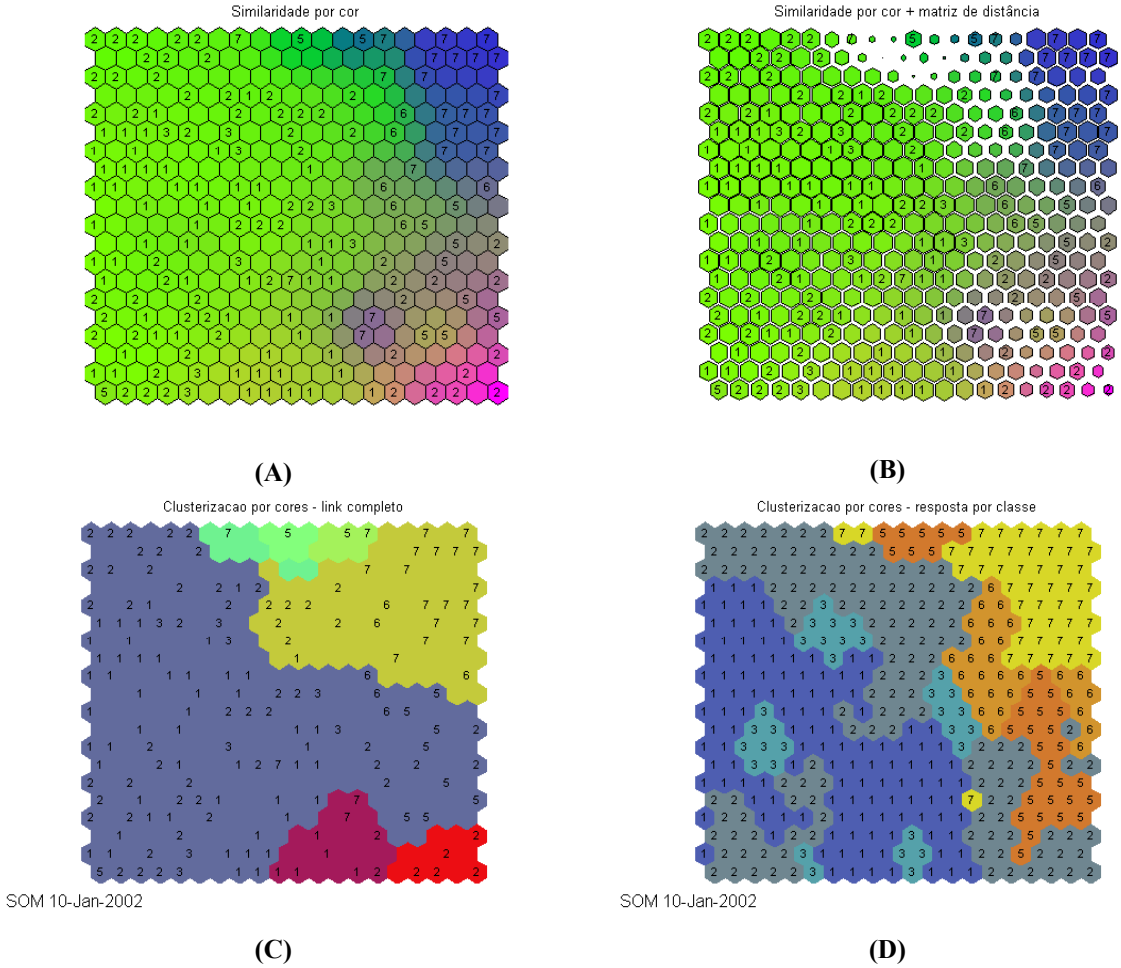


Figura 5-4 – Possíveis formas de agrupamentos. Em (A), utiliza-se de código de cores para representar as dissimilaridades entre os conjuntos. Em (B), adiciona-se ainda a informação da matriz de distância, evidenciando ainda mais os possíveis agrupamentos. Em (C), os grupos são representados utilizando a matriz de distância num algoritmo de ligação completa. O número de agrupamentos para este algoritmo é decidido *a priori*, o que pode induzir um número indevido de grupos, como no exemplo. Em (D), agrupa-se os neurônios conforme o rótulo do objeto mais próximo de cada um, obtendo-se grupos desconexos. Este último resultado pode indicar conjuntos de dados de separação fortemente não lineares. Os rótulos sobre os neurônios foram sobrepostos apenas para avaliação do resultado oferecido pela ferramenta e não estiveram disponíveis durante a adaptação.

A Figura 5-4-C representa a matriz de distâncias utilizando um método de agrupamento hierárquico de ligação completa (veja a Seção 2.3.1 para detalhes sobre o algoritmo de ligação completa). Em (D), cada neurônio foi rotulado com o objeto que lhe é mais próximo e o mapa foi então colorido conforme o grupo a que cada neurônio pertence (a distribuição de cores é arbitrária e serve apenas para diferenciar os grupos). Esta análise pode revelar descontinuidades na representação dos agrupamentos, como no exemplo, e sugerir que a grade elástica do SOM está "retorcida". A torção na grade elástica do SOM é

um fenômeno que pode ocorrer quando a inicialização dos pesos sinápticos é feita aleatoriamente (Kohonen, 1997). Em todos os casos nesta dissertação, entretanto, foi utilizada a inicialização linear. Neste caso, portanto, este resultado leva a uma outra suposição, a de que o conjunto de dados é, de fato, de difícil separação, sendo fortemente não linear. Resultados possivelmente melhores poderiam ser obtidos com algoritmos SOM operando em espaço $\Re^3$ ou superior.

Sobre o mesmo conjunto de dados (não normalizados) foi gerado um modelo GTM num arranjo de 20 × 20 pontos latentes com 12 × 12 funções-base com desvio padrão (σ_ϕ) igual a 1, onde $\sigma_\phi = 1$ significa uma vez a distância entre os centros de duas gaussianas no arranjo de funções-base. O fator de regularização dos pesos utilizado foi 0,001 e o modelo foi adaptado em 20 ciclos, conforme Tabela 5-2.

A Figura 5-5 apresenta alguns resultados obtidos pela ferramenta GTM. Em (D), identifica-se com relativa clareza a presença de um agrupamento (identificados pela área escura no 2º e 3º quadrantes), onde a maioria dos objetos das classes **1**, **2**, **3** e **4** encontram-se representados, diferenciados dos objetos das classes **5**, **6** e **7**, um resultado bastante semelhante ao obtido através do SOM. De forma semelhante, também não é evidente a separação entre os agrupamentos de dados.

A utilização de ambas as ferramentas, SOM e GTM, aumenta em muito as possibilidades de análise dos dados, comparando-se àquelas oferecidas por algumas das técnicas mais tradicionais, como apresentadas na Figura 5-1. Ambas as ferramentas, no entanto, tiveram dificuldades com este conjunto de dados, o que parece apontar uma dificuldade de separação inerente ao conjunto. Eventualmente, poder-se-ia considerar a hipótese de que os atributos disponíveis para discriminar cada objeto são insuficientes. Apesar destas dificuldades, as possibilidades de análise oferecidas justificam plenamente seu uso.

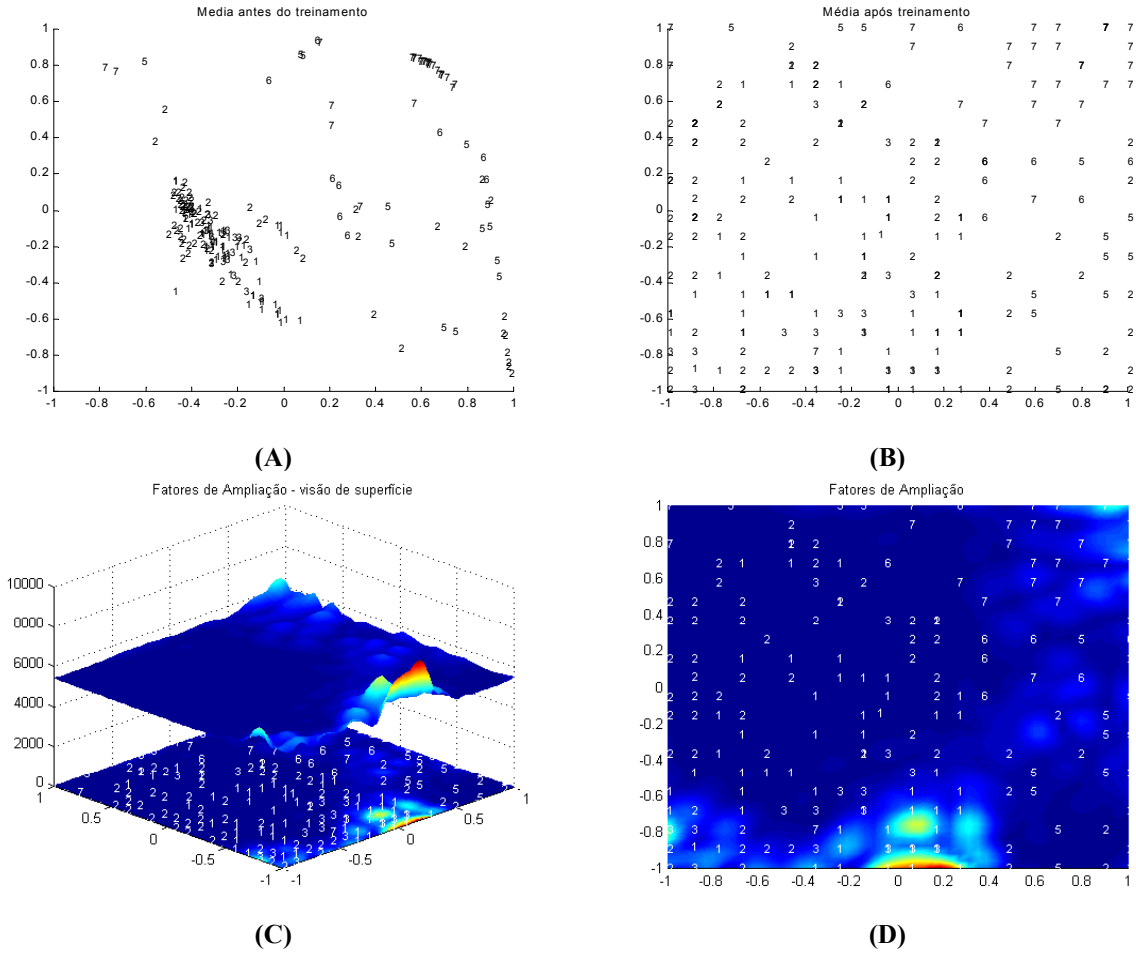


Figura 5-5 – Modelo GTM inicial (A) e após adaptação (B), onde são plotadas as médias *a posteriori* da distribuição dos dados sobre o espaço latente. Em (C), os fatores de ampliação podem ser vistos como uma superfície em que os picos representam áreas onde o subespaço foi "esticado" e os vales, áreas de alto fator de ampliação. Sobre a projeção dos fatores de ampliação, em (D), foram projetadas as médias *a posteriori*. Os rótulos dos dados são plotados apenas para avaliar os resultados da ferramenta.

5.2.2 Conjunto *"Ionosphere"*

O conjunto *"Ionosphere"* possui um total de 351 objetos definidos por vetores compostos por 34 atributos, contínuos, relativos a 17 pulsos de alta freqüência disparados contra a ionosfera. Cada vetor é composto por 17 pares de atributos obtidos por uma função de autocorrelação que processa os pulsos disparados. O conjunto reside em $\Re^{34}$ e os objetos são classificados como "bons" (aqueles que evidenciam algum tipo de estrutura na ionosfera) ou "ruins" (que são considerados apenas ruído de fundo). Os 200 primeiros objetos são usados para treinamento e os 151 restantes são utilizados para avaliação de resultados. Esta divisão em conjuntos de treinamento e teste segue a sugestão contida no

banco de dados do conjunto, onde o conjunto de treinamento possui 50% de objetos com rótulo “bom” e o restante com rótulo “ruim”. Um critério de divisão comumente usado é separar 90% dos objetos do conjunto inicial e destiná-los à adaptação do modelo, enquanto os outros 10% são utilizados para avaliar o resultado (operação também conhecida por *“calibração”*). A dimensão dos dados neste conjunto, $\Re^{34}$, é bem maior que a do caso de teste “*Glass*”, $\Re^{9}$.

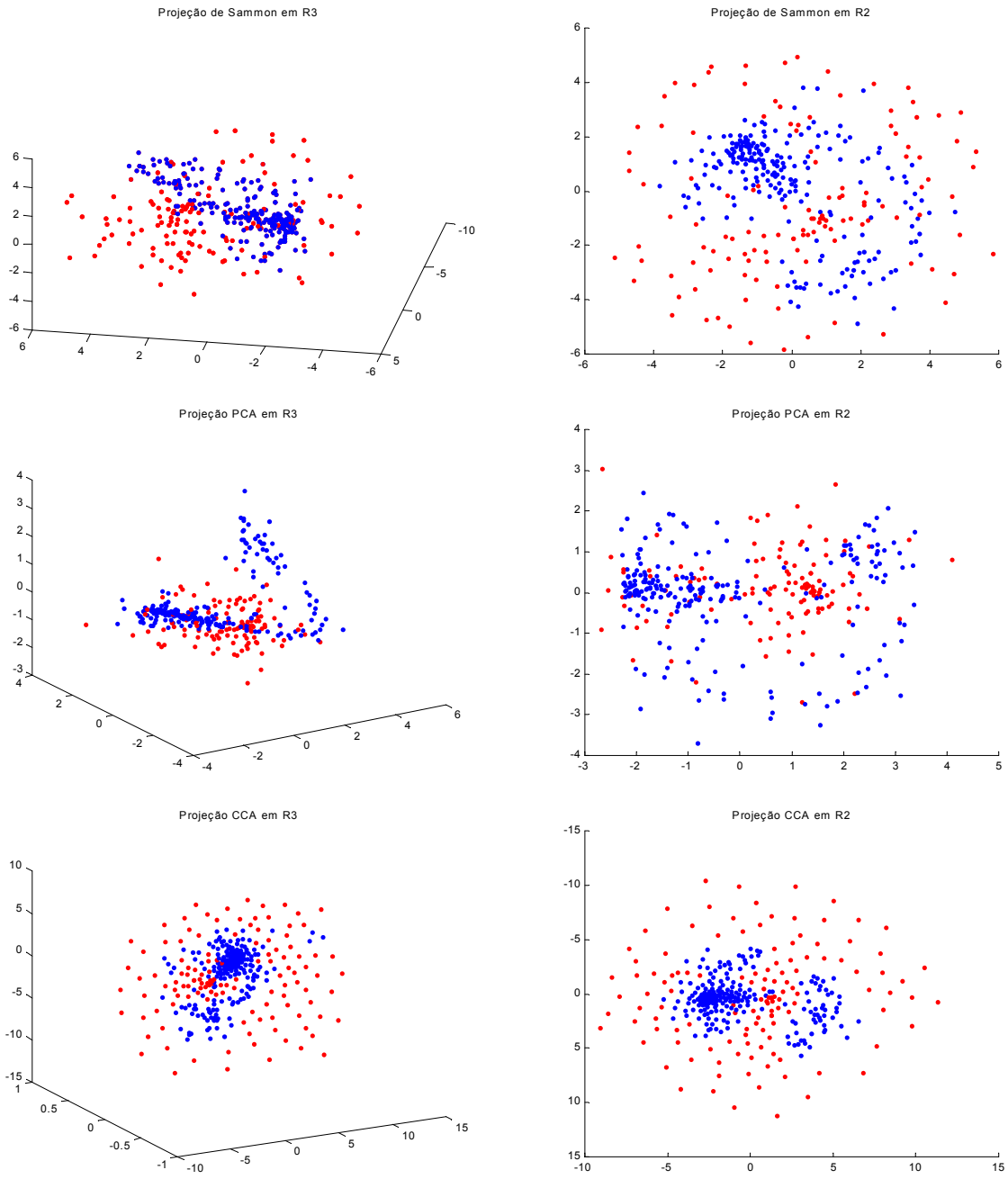


Figura 5-6 – Métodos de projeção de Sammon, PCA e CCA do conjunto *"Ionosphere"*. Os pontos azuis indicam sinais "bons" e os vermelhos, "ruins" (151 casos de teste).

A Figura 5-6 demonstra a ineficiência de alguns métodos de projeção tradicionais quando os conjuntos de dados se encontram em espaços de grande dimensão ou não são linearmente separáveis, como é o caso. Sem informação prévia da classificação dos objetos, dificilmente se poderia observar agrupamentos utilizando apenas os métodos de Sammon, PCA e CCA, mesmo que analisados em conjunto. No exemplo, as duas classes de objetos são diferenciadas por cores.

O modelo SOM escolhido a partir dos testes realizados é um arranjo plano de 15×10 neurônios, com vizinhança hexagonal, inicializado linearmente ao longo da distribuição dos dados e treinado pelo algoritmo *"batch"* em duas fases: a primeira, com 5 épocas e vizinhança regredindo de 3 a 1; e a segunda, com 25 épocas e vizinhança fixa em 1. Para este exemplo em particular, o processo de escolha levou a um modelo onde os vetores de entrada não foram normalizados em sua variância, pois este procedimento provocou um aumento sensível nos valores de TE, conforme pode ser observado na Tabela 5-1.

A Figura 5-7 apresenta alguns resultados, onde é possível observar a separação aproximada dos objetos do conjunto de teste, embora algumas classificações incorretas possam ser observadas.

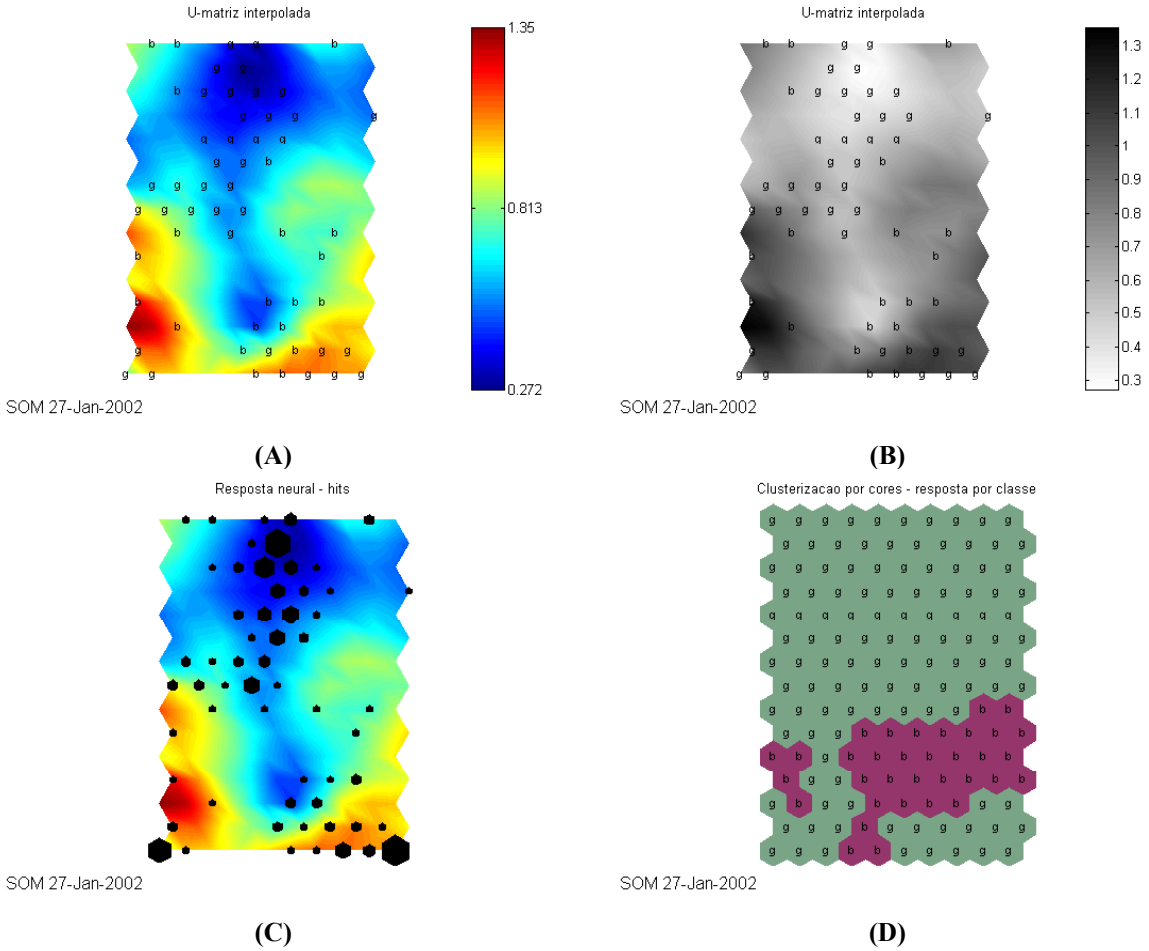


Figura 5-7 – Matriz-U interpolada (A e B) do conjunto *"Ionosphere"*. Em (C) a freqüência de resposta do mapa, mostrando concentração em alguns neurônios e sugerindo a idéia de 3 agrupamentos. Em (D) a classificação conforme o rótulo do objeto mais próximo de cada neurônio. Os rótulos dos dados não estiveram disponíveis na adaptação do modelo.

O modelo GTM escolhido foi gerado num arranjo de 20×20 pontos latentes com 5×5 funções-base com desvio padrão (σ_ϕ) igual a 0,8, fator de regularização 0,001 e adaptação em 20 ciclos. Os dados não foram normalizados, conforme pode ser constatado na Tabela 5-2.

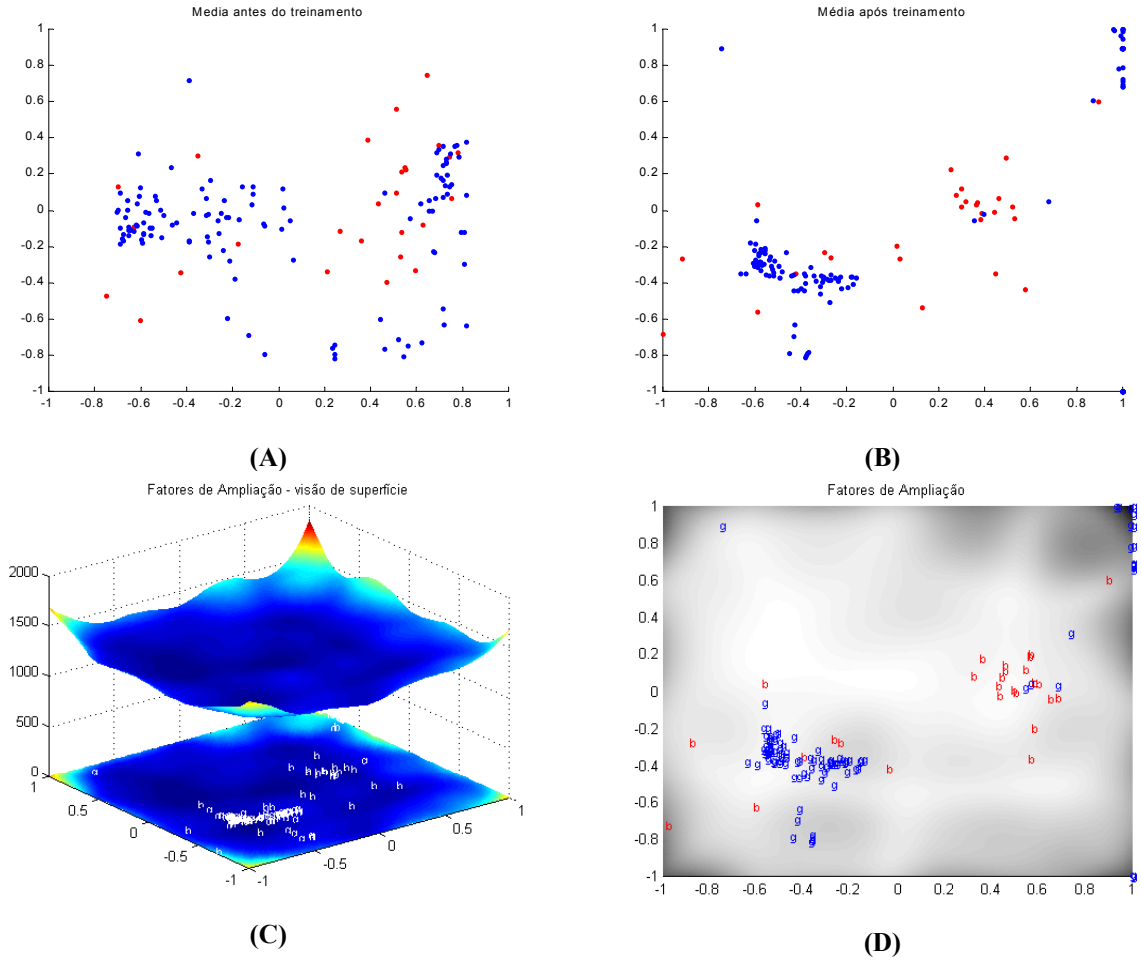


Figura 5-8 – Modelo GTM inicial (A) e após adaptação (B), onde são plotadas as médias *a posteriori* da distribuição dos dados sobre o espaço latente (pontos azuis são "bons" e vermelhos, "ruins"). Em (C), os fatores de ampliação podem ser vistos como uma superfície em que os picos representam áreas onde o subespaço foi "esticado" e os vales, áreas de alto fator de ampliação. Sobre a projeção dos fatores de ampliação (D) foram plotadas as médias *a posteriori*. Os rótulos dos dados não estiveram disponíveis durante a adaptação do modelo.

Pode-se perceber que, assim como o SOM, o GTM também apresenta algumas classificações incorretas (considerando os rótulos prévios dos dados), sugerindo também a existência de 3 agrupamentos. Os resultados de ambas as ferramentas, se comparados àqueles obtidos pelas técnicas de projeção mais tradicionais (Figura 5-6), são sensivelmente superiores, com indicações mais claras da existência e separação entre agrupamentos. Isto permite afirmar que as técnicas mais tradicionais são bastante sensíveis ao aumento da dimensionalidade do conjunto de dados, tornando-as bastante ineficientes. Já as duas ferramentas analisadas nesta dissertação, SOM e GTM, mostraram-se bastante robustas, com resultados compatíveis entre si. Isto permite afirmar que, em análises envolvendo

dados em espaços de alta dimensionalidade, será necessário lançar mão de um ferramental mais amplo que apenas as técnicas de projeção mais tradicionais.

5.2.3 Conjunto *"Letter"*

O conjunto *"Letter"* possui um total de 20000 dados representando letras maiúsculas do alfabeto a partir de imagens obtidas de 20 diferentes tipos de fonte. A partir de cada imagem são extraídos 16 atributos numéricos, inteiros e redimensionados para o intervalo [0,15], representando cada letra. O conjunto reside, portanto, em $\Re^{16}$ e é divido em 26 classes, conforme a letra que cada dado representa. Na verdade, os 16000 primeiros dados são utilizados para treinamento dos modelos e os 4000 dados restantes para avaliação de resultados, conforme a indicação contida no conjunto de dados.

No caso de conjuntos com volume elevado de dados, métodos de projeção tradicionais provam-se bastante ineficientes, uma vez que não são capazes de reduzir a quantidade dos dados visualizados (Figura 5-9). O resultado pouco ou nada contribui para a avaliação, exceto por sugerir uma distribuição hiperesférica e, portanto, um arranjo de formato quadrado para o SOM.

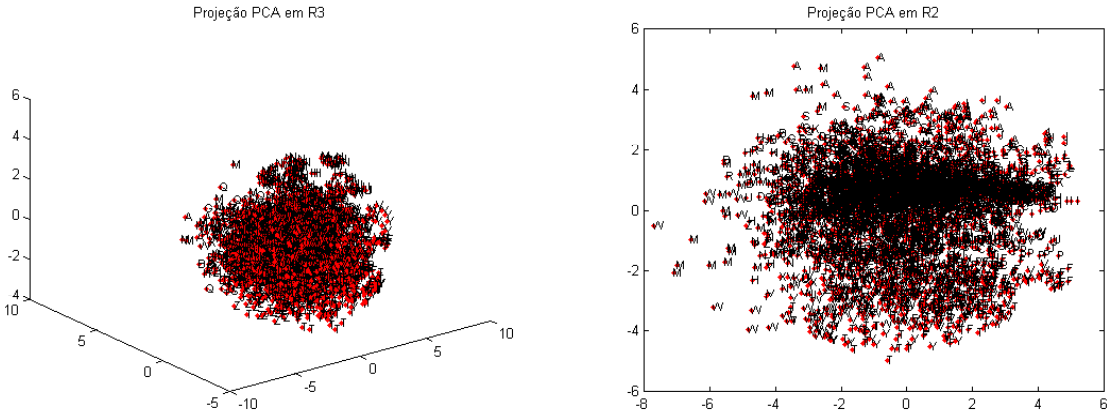
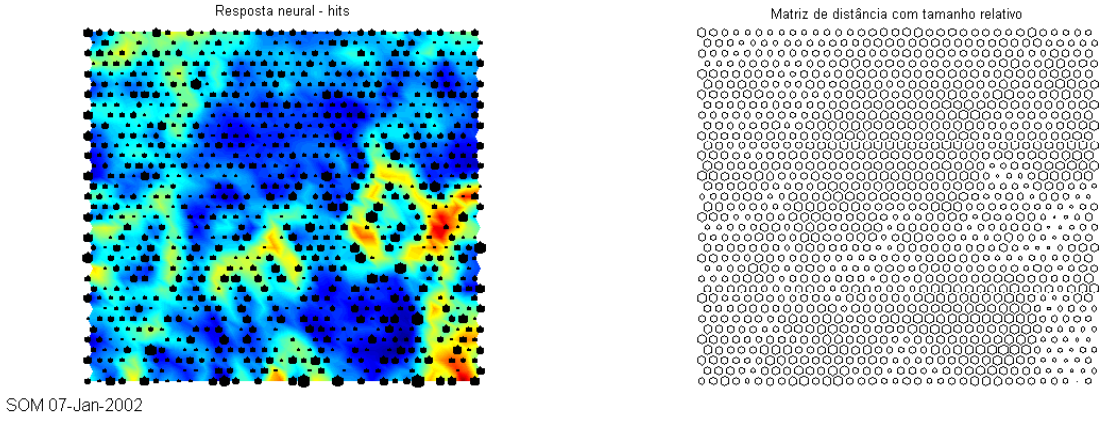


Figura 5-9 – Método de projeção PCA aplicado ao conjunto *"Letter"* não obtém bons resultados devido à elevada quantidade de dados.

A projeção de Sammon, para este caso (i.e., com número elevado de dados), mostrou-se computacionalmente muito cara, sendo assim potencialmente inútil em conjuntos com grandes volumes de dados, fato comum em mineração de dados (Fayyad *et al.* 1996d).

O modelo SOM, escolhido pelo critério exposto inicialmente, foi um arranjo plano de 10×12 neurônios com vizinhança hexagonal, inicializado linearmente ao longo da distribuição dos dados e treinado pelo algoritmo *"batch"* em duas fases: a primeira, com 3 épocas e vizinhança regredindo de 3 a 1; e a segunda, com 10 épocas e vizinhança fixa em 1, conforme Tabela 5-1. No entanto, optou-se por um arranjo plano de 35×35 neurônios para aumento da resolução do modelo (em outras palavras, diminuir a quantidade de dados representados por cada neurônio). Os dados foram normalizados. A vizinhança é hexagonal e o mapa foi inicializado linearmente, sendo treinado pelo algoritmo *"batch"* em duas fases: a primeira com 5 épocas com vizinhança regredindo de 5 a 1; e a segunda com 50 épocas e vizinhança fixa em 1.

A análise da freqüência de resposta dos neurônios (Figura 5-10) sugere a ocorrência de uma boa distribuição das responsabilidades dos neurônios em relação aos dados, razoavelmente bem distribuídos pelo mapa. Isto indica que a quantidade de neurônios é adequada à representação dos dados.



(A) **(B)**

Figura 5-10 – Freqüência de resposta dos neurônios plotada sobre a matriz-U (A), onde o tamanho do hexágono que representa o neurônio indica a quantidade de dados por ele representado. Observa-se uma configuração onde não há neurônios sobrecarregados. Em (B), a matriz-U é vista de forma que a distância dos vetores de pesos dos neurônios é inversamente proporcional ao tamanho do hexágono usado para representá-los. Assim, agrupamentos são percebidos por grupos de hexágonos maiores e mais próximos entre si, enquanto que regiões "esticadas" são representadas por hexágonos pequenos.

A Figura 5-11 apresenta as matrizes-U individuais das características dos vetores de dados, o que permite uma análise da correlação entre estas.

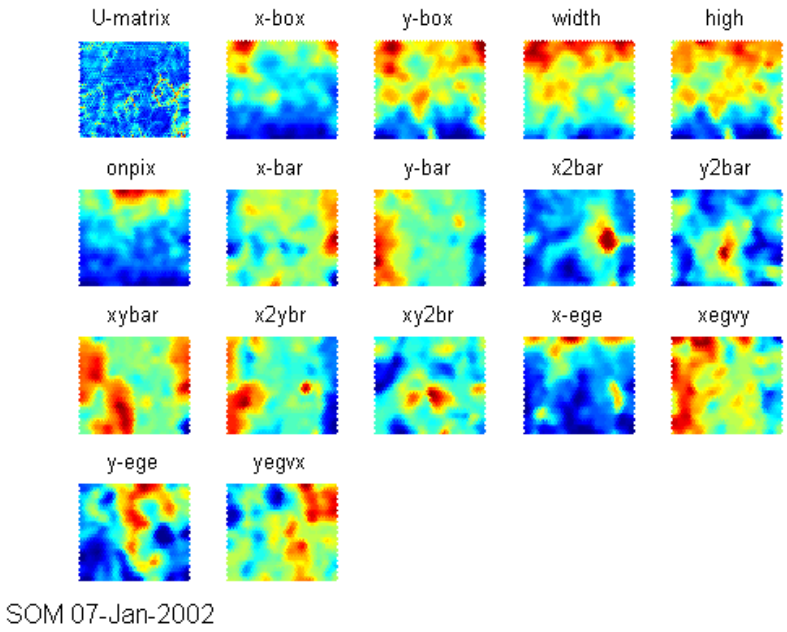


Figura 5-11 – Matrizes-U individuais das características do conjunto "*Letter*". A análise visual permite identificar correlações entre as características dos vetores de dados. Pode-se perceber que as características "*y-box*" e "*high*" são correlacionadas positivamente, pois as matrizes-U correspondentes são bastante semelhantes (o cálculo posterior da correlação é igual a 0.8232).

A Figura 5-12 apresenta a matriz-U do SOM, sobre a qual foram plotados os rótulos dos dados. Esta informação não foi disponibilizada para a ferramenta durante a adaptação do modelo. Uma característica do SOM é poder executar a *redução de dados*, pois um neurônio pode representar diversos dados (isto é, o neurônio pode ser BMU para diversos dados). Se os dados forem previamente rotulados, há a possibilidade de que dados com diferentes rótulos sejam representados por um mesmo neurônio. Isto não é necessariamente um erro, mas uma interpretação da similaridade dos dados segundo a óptica do SOM. Neste caso, há algumas possibilidades para a rotulação do neurônio:

a) os rótulos de todos os dados podem ser apresentados, o que seria inviável em conjuntos numerosos;
b) apenas um rótulo de cada possível diferente conjunto é apresentado;
c) apenas o rótulo do conjunto com maior freqüência é apresentado;
d) apenas o rótulo do dado mais próximo do vetor de pesos do neurônio em questão.

As três primeiras opções são encontradas no Toolbox para Matlab® (Alhoniemi *et al.* 2000). As opções de rotulação (a) e (b) geram gráficos confusos devido ao grande número

de dados do conjunto, pois os rótulos apresentam-se sobrepostos, o que dificulta bastante a análise. O resultado da rotulação conforme a opção (c) pode ser visto na Figura 5-12. O resultado apresenta apenas um rótulo por neurônio, executando redução de dados. Embora esta característica seja importante por permitir avaliar grandes conjuntos de dados de forma resumida, o resultado para este caso de teste é relativamente confuso, e pode-se ver rótulos das mesmas letras em várias posições do mapa.

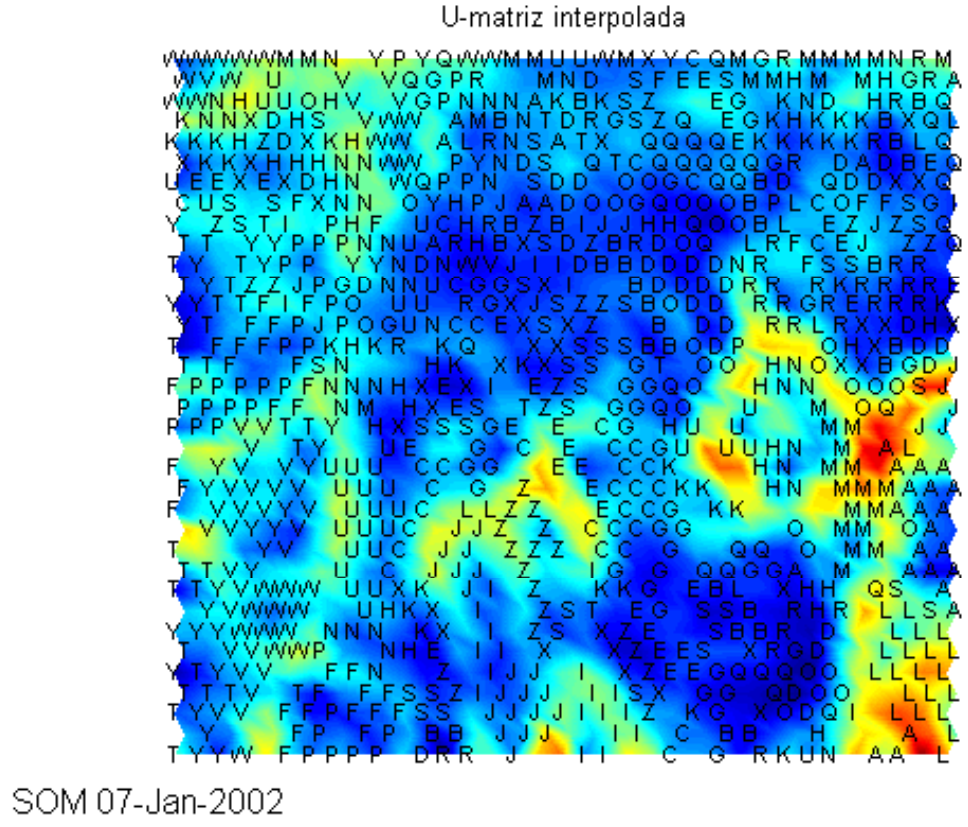


Figura 5-12 – Matriz-U interpolada, sobre a qual foram plotados os rótulos dos dados. Cada neurônio recebeu o rótulo de maior freqüência sobre o conjunto de dados por ele representado. Embora identifique-se alguns possíveis agrupamentos (como os rótulos "L" no canto inferior direito da matriz-U), há rótulos aparentemente bastante espalhados (como o rótulo "Z"). Dessa forma, dificilmente pode-se identificar agrupamentos válidos.

A opção de rotular os neurônios conforme o rótulo do dado mais próximo do vetor de pesos deste neurônio em questão (opção d) não é uma abordagem existente na ferramenta. O resultado obtido segundo este critério prioriza não a quantidade de rótulos de um neurônio, mas o vetor de dados mais próximo deste e pode ser vista na Figura 5-13. Em outras palavras, o neurônio é rotulado pelo tipo de dado mais próximo de seu vetor de pesos, aumentando a especificidade local de classificação do mapa.

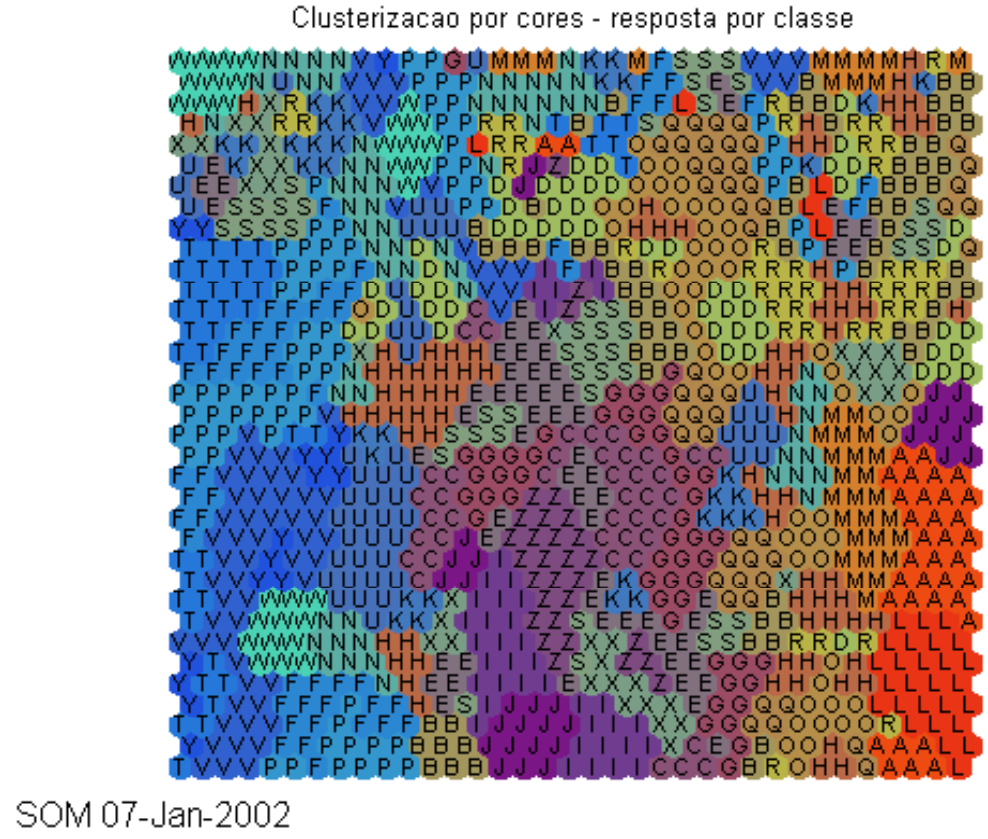


Figura 5-13 – Cada neurônio recebeu o rótulo do dado mais próximo de seu vetor de pesos, aumentando a especificidade de representação. Os rótulos não estiveram disponíveis na adaptação do modelo. É possível observar mais de um agrupamento para o mesmo rótulo, o que pode ser justificado pelos diferentes conjuntos de fontes utilizados no conjunto de dados.

O modelo GTM escolhido segundo o critério exposto seria um arranjo de 20×20 pontos latentes com 15×15 funções-base com desvio padrão (σ_ϕ) igual a 0,5, fator de regularização 0,01 e adaptado em 20 ciclos sobre os dados normalizados, conforme Tabela 5-2. Entretanto, a análise dos resultados mostrou que, para todos os modelos gerados sobre dados normalizados no conjunto de testes, houve um colapso do modelo, que representou todos os dados em um único ou em pouquíssimos pontos. Dessa forma, o critério foi aplicado apenas sobre os dados não normalizados, tendo sido escolhido, assim, um arranjo de 20×20 pontos latentes com 15×15 funções-base com desvio padrão (σ_ϕ) igual a 0,8, fator de regularização 0,01 e adaptado em 20 ciclos (isto é, o que apresentou o maior logaritmo da verossimilhança – logL). Os resultados podem ser visto na Figura 5-14.

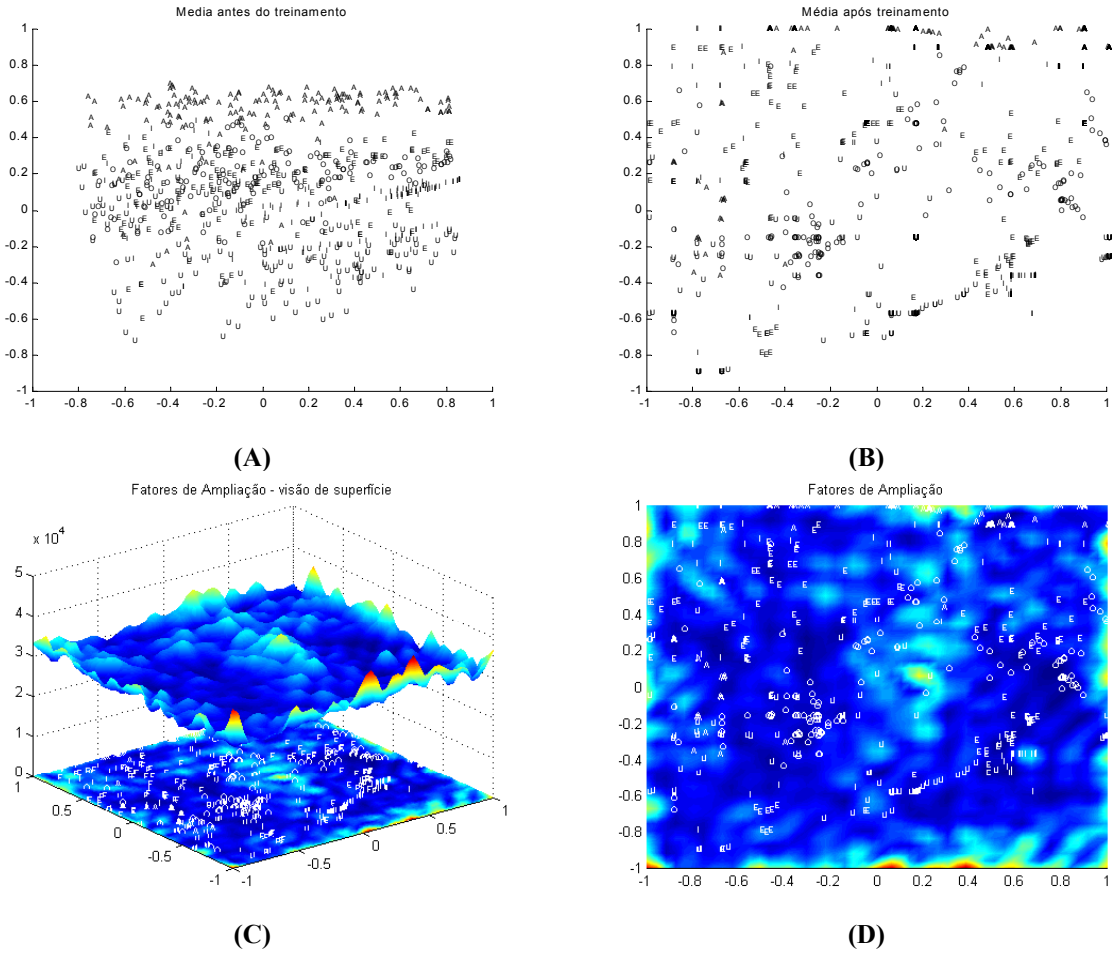


Figura 5-14 – Modelo GTM inicial (A) e após adaptação (B), onde são plotadas as médias *a posteriori* da distribuição dos dados sobre o espaço latente. Em (C) os fatores de ampliação podem ser vistos como uma superfície onde os picos representam áreas onde o subespaço foi "esticado" e os vales, áreas de alto fator de ampliação. Sobre a projeção dos fatores de ampliação (D) foram projetadas as médias *a posteriori*. Somente as vogais foram analisadas, pois a ferramenta utilizada nesta dissertação para o modelo GTM não realiza redução de dados, e o resultado com todos os dados é incompreensível.

A Figura 5-14 apresenta o resultado obtido com o GTM onde foram utilizadas apenas as vogais para análise do resultado. Embora o modelo GTM possa realizar redução de dados, a ferramenta utilizada nesta dissertação não executa esta função (como a ferramenta utilizada para o SOM) e torna-se bastante sensível ao volume de dados que deve ser observado. Para este caso de teste, esta foi uma restrição séria da ferramenta. Uma opção possível é analisar conjuntos menores de dados previamente escolhidos, como foi a escolha nesta dissertação. É claro que, numa tarefa real de mineração de dados, onde não há nenhum conhecimento prévio dos rótulos dos dados, esta seria uma tarefa impossível e dificultaria bastante o uso da ferramenta como se encontra.

Os resultados obtidos para este caso de teste devem ser tomados com precaução, apesar da aparente inaptidão das ferramentas em agrupar os dados. O conjunto de dados refere-se à representação de letras do alfabeto para diversos tipos de fontes, e não é razoável supor que estes dados devam ser apresentados em agrupamentos totalmente conexos conforme a letra, especialmente considerando o treinamento não supervisionado. O que foi demonstrado é que as ferramentas utilizadas apresentam, invariavelmente, dificuldades em representar grandes quantidades de dados, e este é um forte argumento para lançar mão de mais de uma ferramenta na tarefa de mineração de dados.

5.2.4 Conjunto *"Zoo"*

O conjunto *"Zoo"* possui um total de 101 objetos definidos por vetores compostos por 16 atributos (1 numérico, *"número de pernas"* e 15 binários, como *"tem penas"*, *"voa"* etc.). O conjunto reside em $\Re^{16}$ e é divido em 7 classes conforme a Tabela 5-4:

Tabela 5-4 - O conjunto de dados *"Zoo"* com suas classes.

Classe	Descrição	N.º exemplos
1	Mamíferos	41
2	Aves	20
3	Répteis	5
4	Peixes	13
	Anfíbios	4
6	Insetos	8
7	Moluscos e crustáceos	10

O conjunto *"Zoo"* é considerado "bem comportado", com boa separação entre as classes. Isto pode ser verificado até pelas técnicas de projeção mais tradicionais, conforme Figura 5-15.

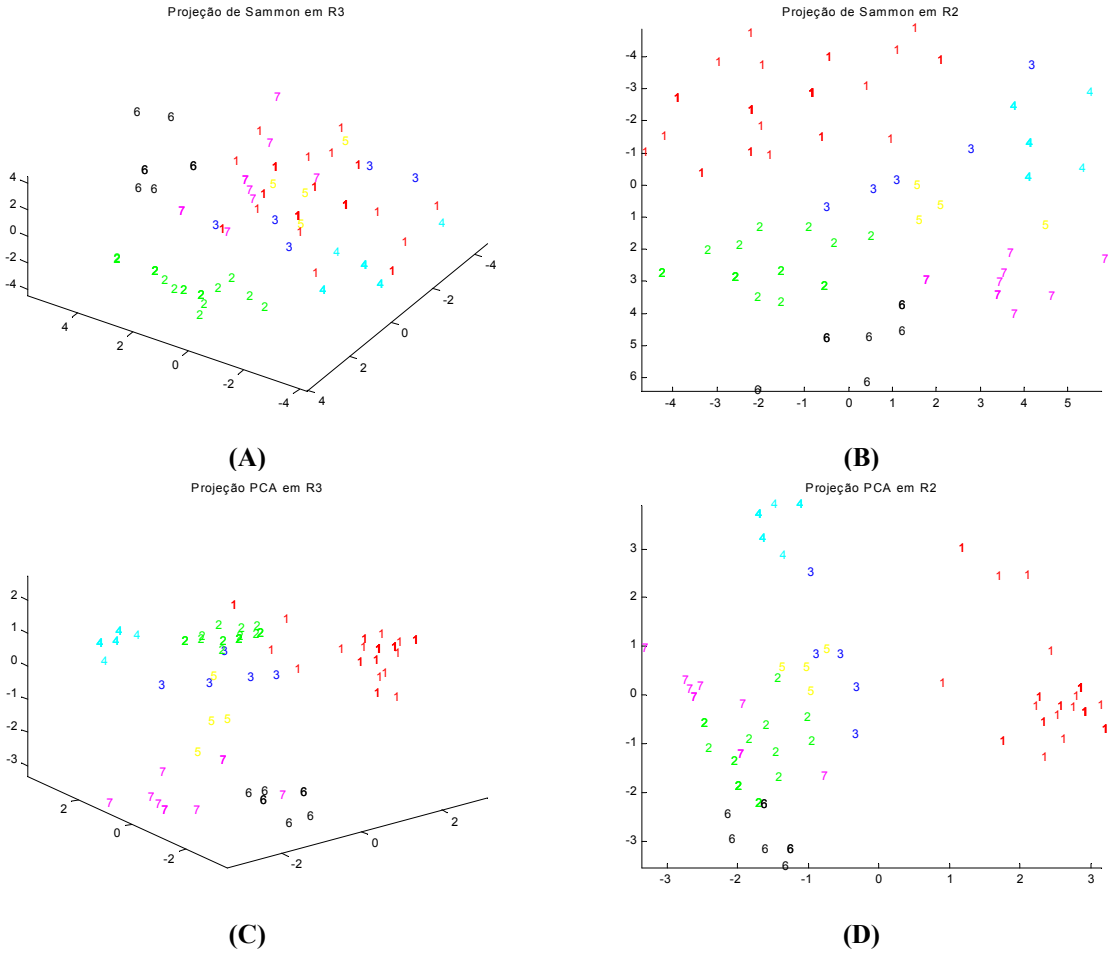


Figura 5-15 – Métodos de projeção de Sammon e PCA do conjunto *"Zoo"*. Alguns agrupamentos apresentam boa separação. Os dados foram rotulados conforme a Tabela 5-4. Cabe lembrar que os rótulos de dados só foram utilizados para avaliar os resultados gerados pelas ferramentas, não sendo disponibilizados às mesmas durante a adaptação.

O modelo SOM escolhido dentre os testes realizados é um arranjo plano de 15×15 neurônios com vizinhança hexagonal, inicializado linearmente ao longo da distribuição dos dados normalizados e treinado pelo algoritmo *"batch"* em duas fases: a primeira com 5 épocas e vizinhança regredindo de 5 a 1; e a segunda com 30 épocas e vizinhança fixa em 1, conforme Tabela 5-1. A Figura 5-16 apresenta a matriz-U de cada um dos atributos que descrevem os dados do conjunto.

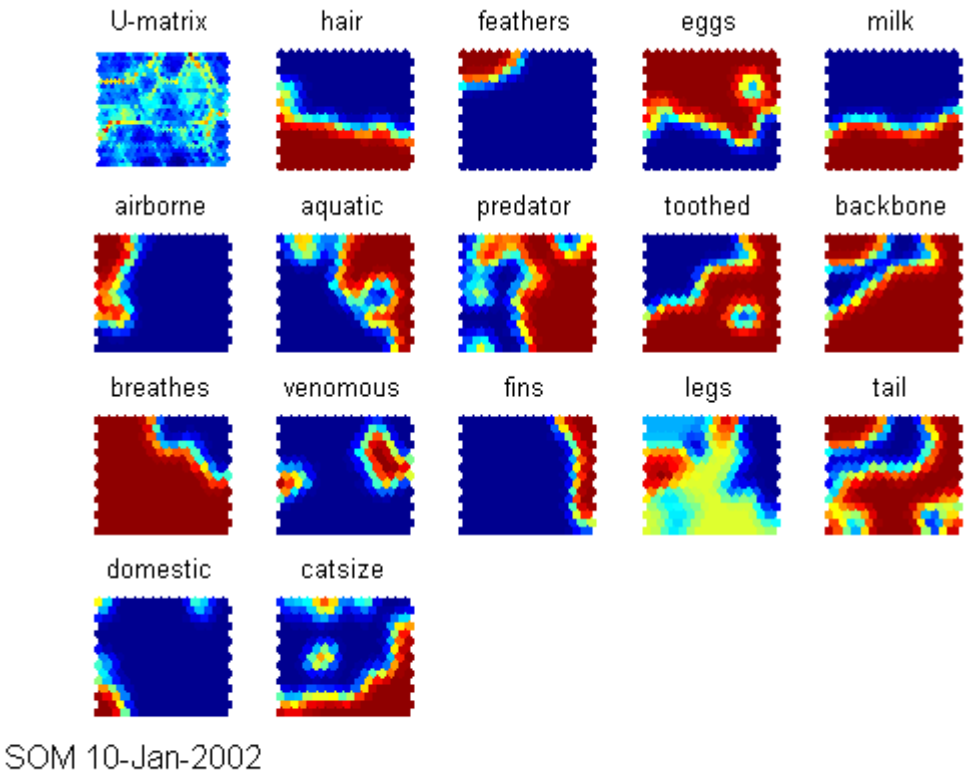


Figura 5-16 – Matriz-U geral (canto superior esquerdo) e matrizes-U individuais para cada atributo, todos binários a exceção de *"legs"*, numérico. As matrizes-U com figuras semelhantes indicam correlação positiva, enquanto figuras com padrão de cor invertido indicam correlação negativa (Vesanto & Ahola, 1999).

Uma informação interessante que pode ser obtida das matrizes-U de cada atributo é a correlação entre estes. A Figura 5-17 apresenta em detalhe os atributos *"eggs"* e *"milk"* e é fácil perceber a semelhança entre as matrizes-U (de fato, a dessemelhança, pois a correlação entre os dois atributos é igual a –0,9388).

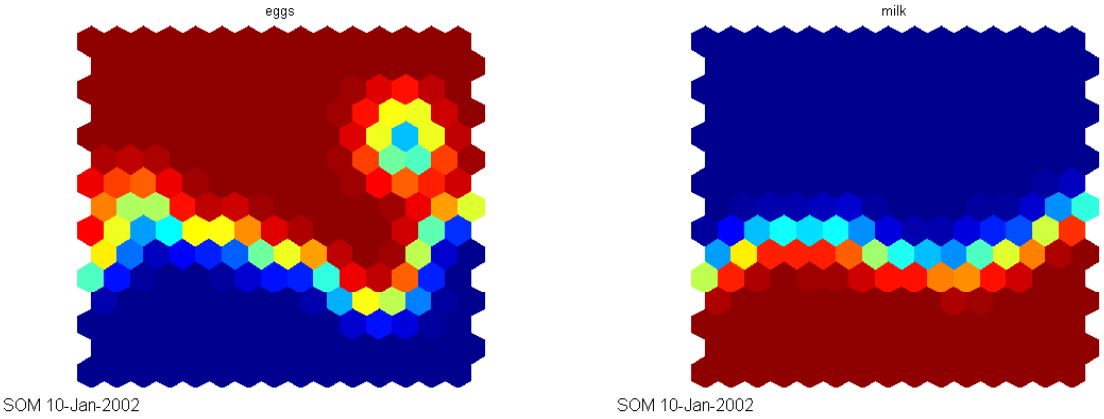


Figura 5-17 – A correlação negativa entre duas características avaliada visualmente pela matriz-U dos planos relativos aos atributos *"eggs"* e *"milk"*, respectivamente indicando animais ovíparos e mamíferos.

A Figura 5-18 apresenta os possíveis agrupamentos obtidos com o SOM. Em (B) e (C) são vistos todos os rótulos associados a cada neurônio do mapa, denotando uma classificação consistente. Um fato interessante é a classificação do objeto *"scorpion"*: segundo a documentação do conjunto, pertencente a classe **7** ("Moluscos e crustáceos"), mas em (C) e na ampliação (D), nota-se que o SOM classifica este objeto como sendo mais semelhante a objetos da classe **6** ("Insetos").

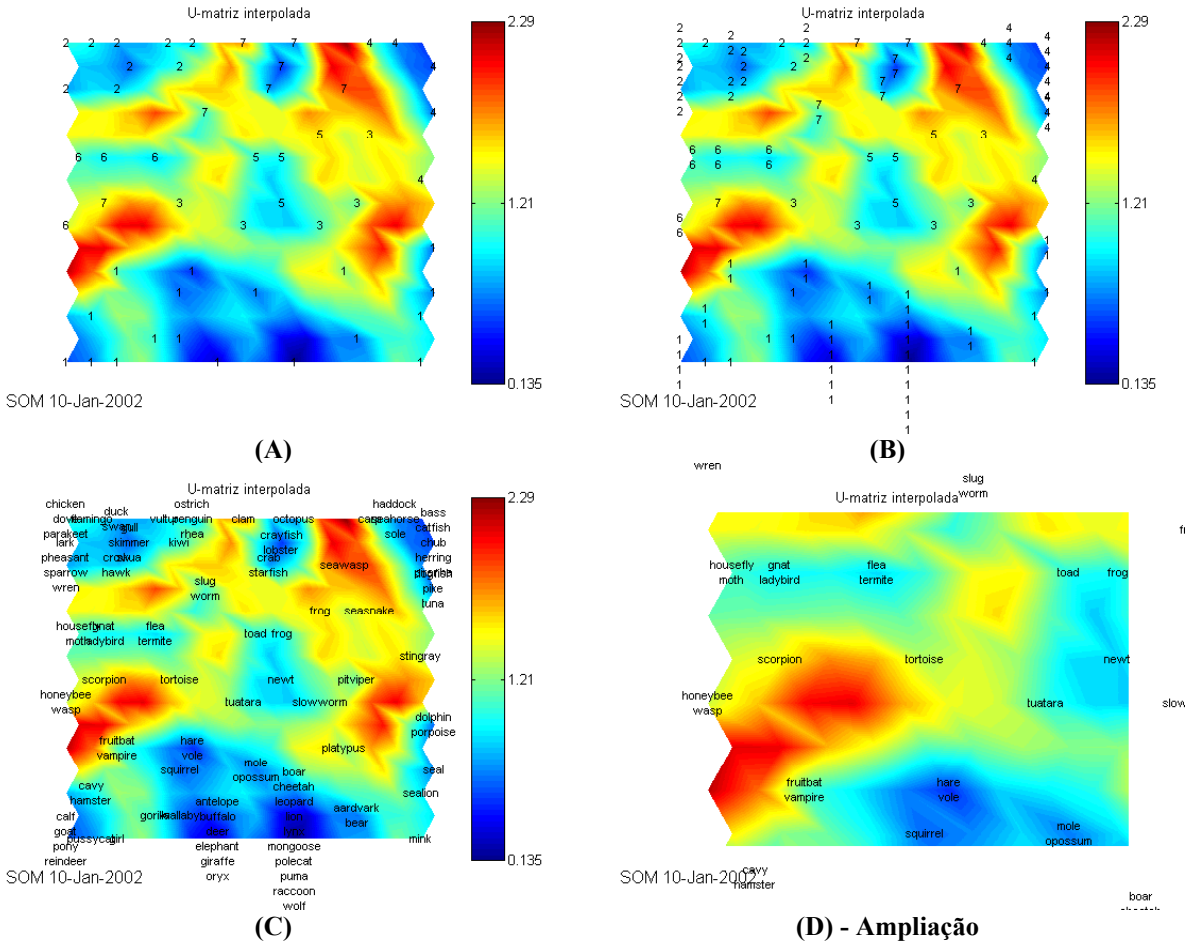


Figura 5-18 – Matriz-U com interpolação de cores evidencia a localização de agrupamentos. Em (C), cada neurônio foi rotulado com todos os tipos de dados para os quais foi o BMU. Em (D), uma ampliação de uma porção do mapa, onde se observa a classificação do dado *"scorpion"* como entendida pelo SOM.

A Figura 5-19 apresenta a classificação por cores e a matriz-U em forma de superfície, cujas informações são combinadas em (C). Em (D), observa-se a matriz-U onde cada neurônio recebeu o rótulo do dado mais próximo de seu vetor de pesos. Neste último, à exceção do objeto *"scorpion"*, exibe agrupamentos conexos e serve para confirmar as hipóteses de agrupamentos de (C).

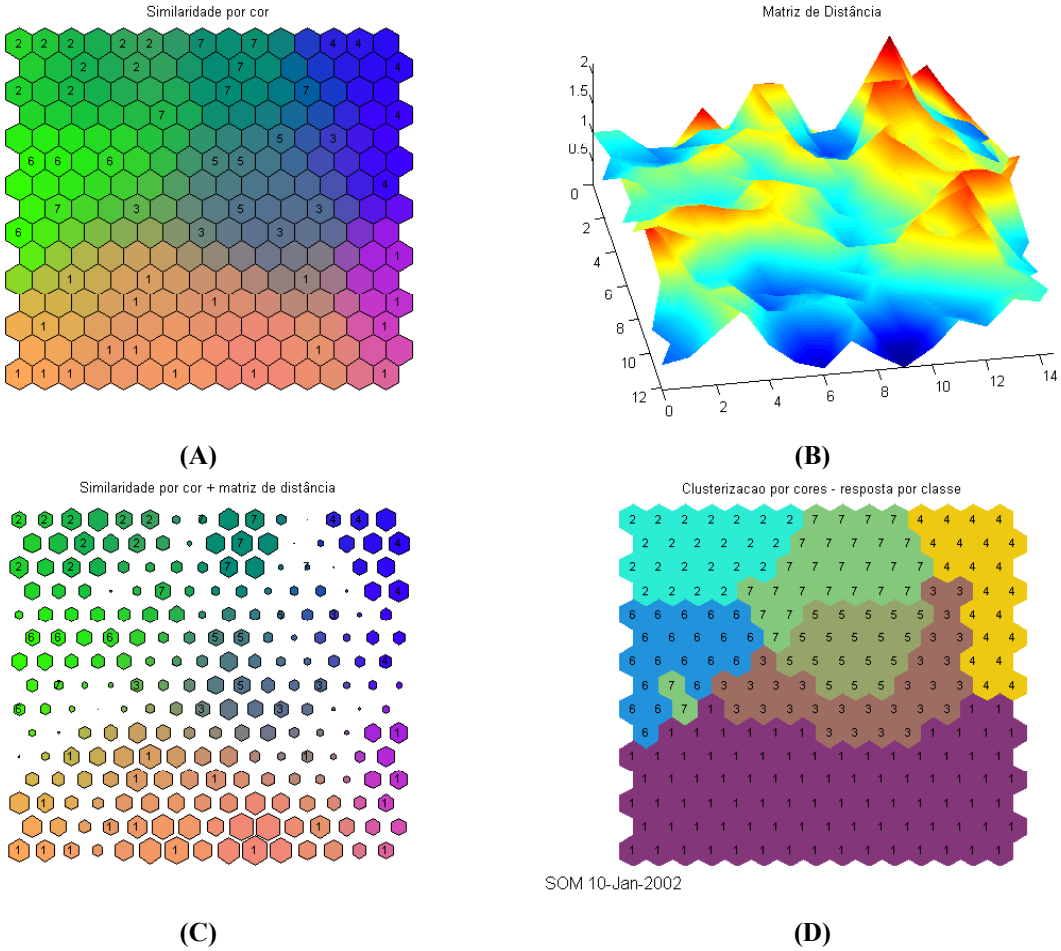


Figura 5-19 – A classificação por cores (A) e a matriz de distâncias (B) agrupadas em (C). O mapa (D) apresenta cada neurônio sendo rotulado pelo tipo de dado mais próximo de seu vetor de pesos. Os rótulos (classes) seguem o definido na Tabela 5-4.

O modelo GTM escolhido inicialmente seria um arranjo de 20×20 pontos latentes com 10×10 funções-base com desvio padrão (σ_ϕ) igual a 0,5, fator de regularização 0,1 e adaptado em 15 ciclos, conforme Tabela 5-2. Entretanto, uma inspeção visual demonstrou uma excessiva sobreposição de pontos, possivelmente sendo causada por um modelo flexível demais (com provável ocorrência de sobre-ajuste). Em função disso, optou-se pelo segundo modelo com maior logaritmo da verossimilhança, com desvio padrão (σ_ϕ) igual a 1,2. Os dados não foram normalizados em nenhum dos experimentos.

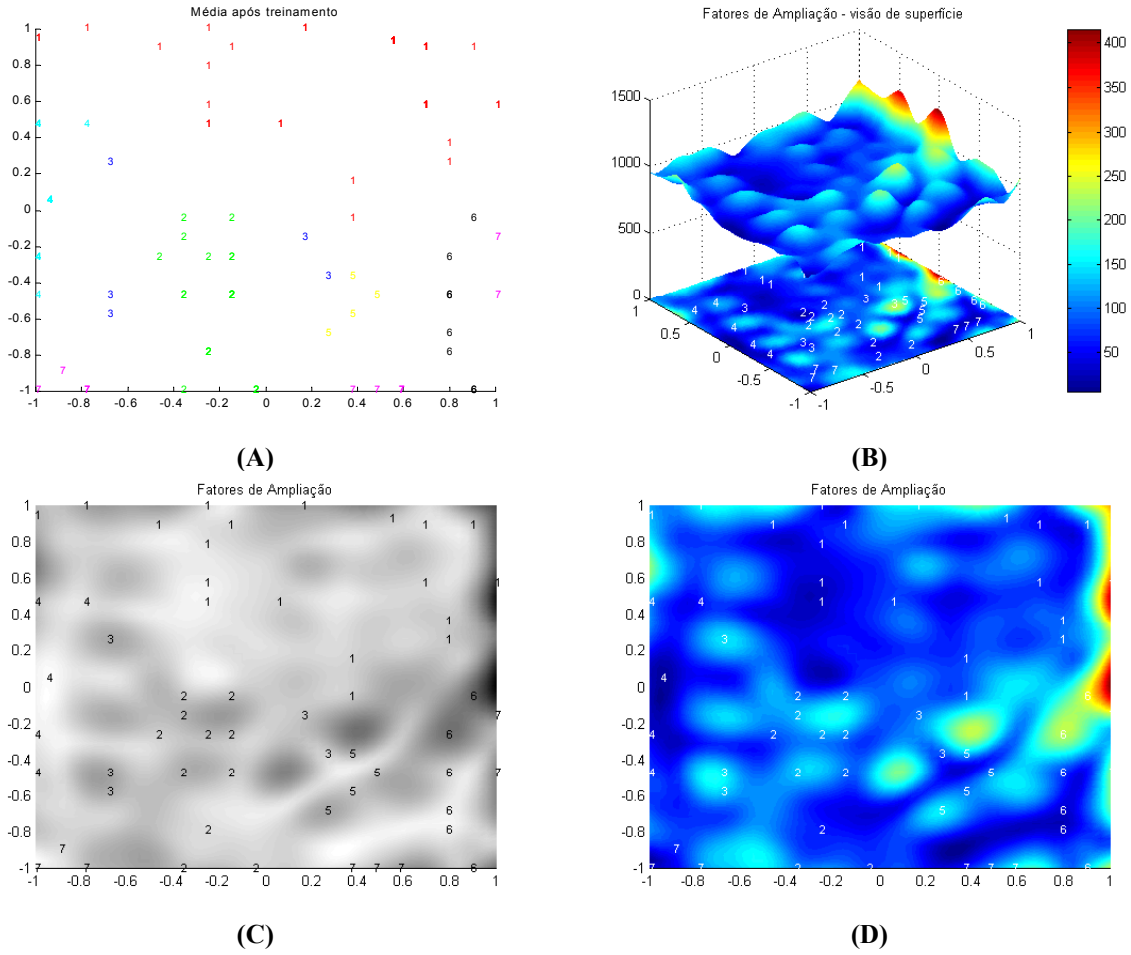
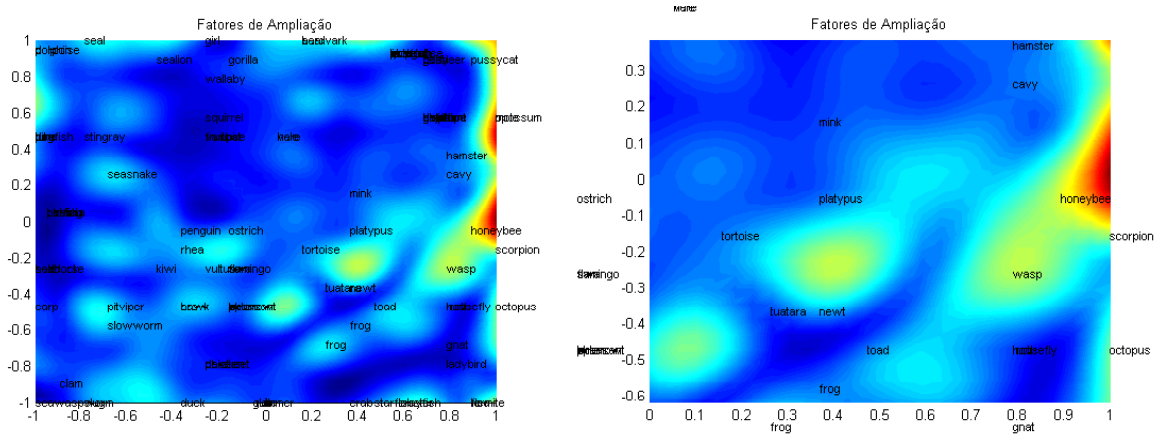


Figura 5-20 – Modelo GTM adaptado (A), onde são plotadas as médias *a posteriori* da distribuição dos dados sobre o espaço latente. Em (B), a superfície do fator de ampliação. Em (C) e (D) as projeções das médias *a posteriori* utilizando-se a superfície do fator de ampliação.

Embora o conhecimento prévio das classes tenha mostrado que o GTM separou de forma consistente os objetos, não é clara nem óbvia a identificação dos possíveis agrupamentos. Por outro lado, o GTM, assim como o SOM, reitera a classificação do objeto *"scorpion"*, posicionando-o próximo a objetos da classe **6** ("Insetos"), como pode ser visto na Figura 5-21. Esta observação permite afirmar que as ferramentas SOM e GTM são consistentes entre si.



(A) **(B) - Ampliação**

Figura 5-21 – Projeção dos nomes dos objetos sobre a superfície do fator de ampliação (A) e a ampliação (B) evidenciando o posicionamento do objeto *"scorpion"*.

A interpretação desta classificação específica do dado citado leva a crer que haja uma inconsistência nos dados da classe *"Zoo"*, ou que há atributos faltantes.

5.2.5 Considerações

A análise dos testes realizados e resultados obtidos leva a algumas constatações:

- **Normalização dos dados de entrada**

  Comparando-se os testes realizados com o SOM entre dados normalizados e não normalizados (20 com dados normalizados e 20 não normalizados para os 4 conjuntos de dados avaliados), percebe-se que o recurso da normalização apresenta um melhor valor para TE em 45% e melhor QE em 25% dos casos, incluindo as possibilidades de sobre-ajuste (*overfitting*). Estes dados sugerem que a normalização não é um procedimento que deva ser adotado sempre, mesmo desconhecendo a influência dos atributos sobre o conjunto.

Tabela 5-5 – Resultados do algoritmo SOM para os melhores valores de TE e QE comparados, numa mesma configuração, utilizando dados normalizados e não normalizados

	Melhor TE	%	Melhor QE	%
Normalizados	9	45	5	25
Não normalizados	11	55	15	75

No caso do GTM, a normalização dos dados não levou a resultados satisfatórios: o único caso em que os dados normalizados levaram a um maior valor do logaritmo da verossimilhança (o conjunto de dados *"Letter"*) mostrou resultados insatisfatórios, pois o modelo adaptado pelo GTM colapsou todos os pontos de dados em um único agrupamento.

Desse modo, pode-se avaliar que a normalização dos dados de entrada conforme a Equação 5-1 é uma decisão que deve ser tomada com cuidado, possivelmente confrontando resultados com e sem normalização para só depois optar por um modelo em particular.

- **Sobre-ajuste (*overfitting*)**

  O aumento do tamanho do arranjo do SOM, bem como um número excessivo de épocas de treinamento, normalmente levam à redução dos parâmetros TE e QE. Entretanto, valores muito baixos para estes índices podem significar sobre-ajuste. Sugere-se a experimentação com diversas configurações descartando-se aquelas com valores excessivamente baixos.

  Da mesma forma, o aumento do número de funções-base, assim como a redução do desvio padrão (σ_ϕ) das funções-base geram modelos GTM cada vez mais flexíveis e, obviamente, melhor adaptados aos dados. Esta abordagem claramente pode levar ao sobre-ajuste, tornando o modelo inútil para os fins de mineração de dados, onde é necessário um certo grau de generalização. Sugere-se a experimentação com diversas configurações, escolhendo-se diferentes valores para o número de pontos latentes e para o conjunto de funções-base. As configurações mais promissoras devem ser exploradas a partir da comparação de desempenho.

- **Ferramentas adicionais**

  Ambos os modelos, SOM e GTM, têm natureza estocástica, o que significa que vários testes devem ser elaborados antes de ser escolhido um bom resultado. Além disso, o uso de outras ferramentas que possam fornecer quaisquer "pistas" sobre o formato aproximado dos agrupamentos (como a projeção de Sammon, PCA etc.) é de extrema

importância na análise exploratória de dados e deve sempre ser considerado como auxílio no processo.

5.3 Conjunto de dados de estilos de aprendizado

Este conjunto é um estudo de caso real sobre a classificação de estilos de aprendizado de alunos universitários ingressantes na Universidade São Francisco, campus de Itatiba, nos cursos de Análise de Sistemas, Ciência da Computação e Engenharia (Elétrica, Mecânica, Civil e de Computação) no ano de 1998. A classificação dos estilos de aprendizado baseia-se no modelo de aprendizado por experiência proposto por Kolb (1984). Este consiste essencialmente na tese de que o ser humano aprende segundo um ciclo em que as experiências concretas são traduzidas em conceitos abstratos que são, por sua vez, usados como referência para obter novas experiências, num ciclo de 4 fases (Figura 5-22):

a) há uma experiência concreta na qual o indivíduo tem participação;
b) a experiência serve como base de observação e reflexão;
c) as observações e reflexões são assimiladas e geram conceitos e modelos abstratos, a partir dos quais novas implicações para os atos podem ser inferidas;
d) as implicações podem ser experimentadas e testadas em novas experiências.

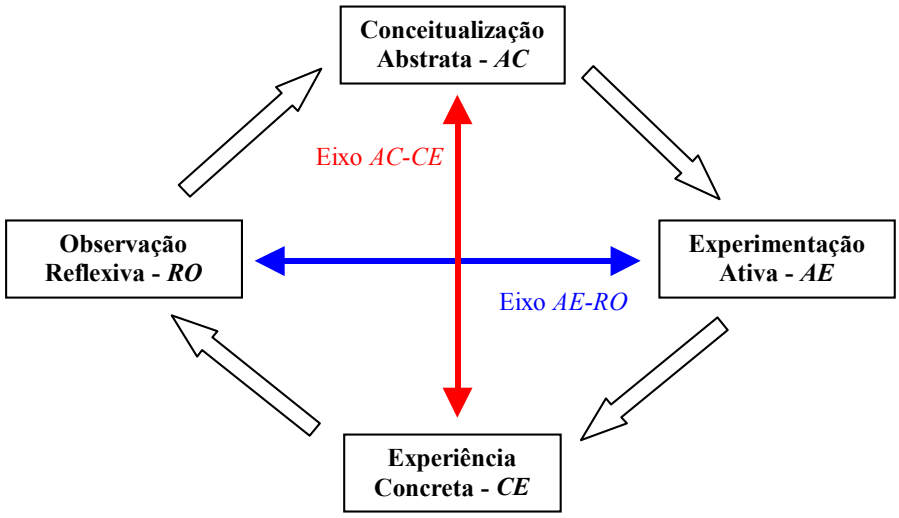


Figura 5-22 - O modelo de aprendizado por experiência. Adaptado de Kolb (2000b).

De acordo com este modelo, há dois eixos fundamentais sobre os quais as características do aprendizado são avaliadas: o eixo *AC-CE* com a oposição abstrato/concreto e o eixo *AE-RO* com a oposição ativo/reflexivo. Quatro índices determinam estas características:

- CE (Experiência Concreta - *Concrete Experience*): sentir
- RO (Observação Reflexiva - *Reflective Observation*): observar
- AC (Conceitualização Abstrata - *Abstract Conceptualization*): raciocinar
- AE (Experimentação Ativa - *Active Experimentation*): fazer

Estes eixos definem 4 quadrantes que caracterizam o estilo de aprendizado dominante de cada indivíduo (Kolb, 2000b):

- **Assimiladores:** predomina a capacidade de observação, contextualização e abstração, sendo presentes raciocínio indutivo, planejamento, análise de dados.
- **Convergentes:** predomina a capacidade de abstração e experimentação ativa, sendo presentes raciocínio dedutivo, foco na solução de problemas, tomada de decisão.
- **Acomodadores:** predomina a experiência concreta e a experimentação ativa, sendo presentes a capacidade de adaptação, liderança, consecução de objetivos, gerenciamento de riscos.
- **Divergentes:** predomina a capacidade de observação e a experiência concreta, sendo presentes a imaginação e criatividade, a percepção de vários pontos de vista e a tendência a divergir de soluções usuais.

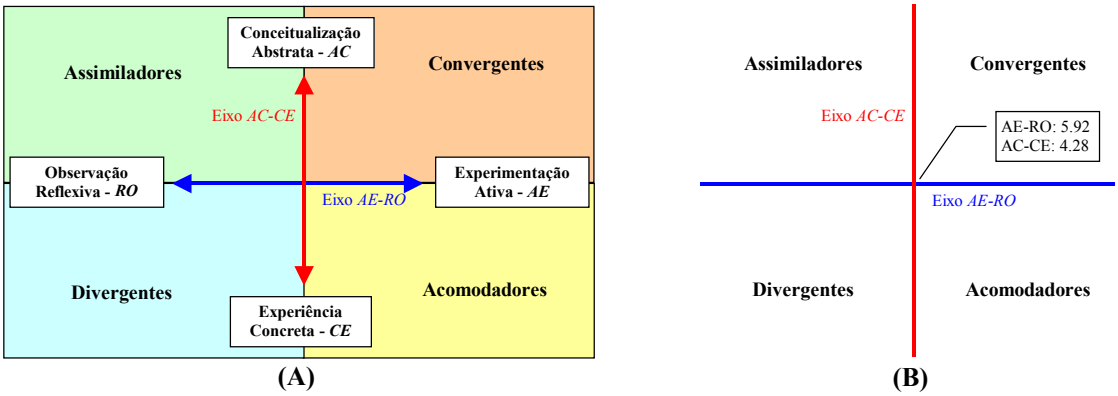


Figura 5-23 - (A) Classificação de estilos de aprendizado. Em (B), os índices utilizados para a classificação. Adaptado de Kolb (2000b).

Para avaliar os indivíduos de acordo com sua tese, Kolb (1984, 2000a, 2000b) propõe um teste, o *Learning Style Inventory version 3* (*LSI-3*), com índices atualizados do *LSI-2A*. Este teste é aplicado e são coletadas as informações sobre as características de aprendizado de cada indivíduo. Resumidamente, a aplicação do teste é descrita a seguir:

- uma ficha individual contendo 12 frases é distribuída entre os indivíduos pesquisados (Anexo 1). Cada frase busca descrever a forma como o indíviduo aprende e contém 4 possíveis finalizações (chamadas de A, B, C e D). Os indivíduos pesquisados devem pontuar as frases A, B, C e D segundo um critério de importância ou relevância para o entrevistado, utilizando-se o número 4 para apontar a finalização que exiba para si a maior importância, 3 para a finalização seguinte em importância e assim sucessivamente. O número 1 será atribuído à frase que menos reflita a forma como o entrevistado aprende.
- após todas as questões respondidas, são somados os pontos atribuídos às colunas A, B, C e D, gerando os índices CE, RO, AC e AE.
- são calculados dois índices, AC-CE e AE-RO que permitem classificar os estilos conforme a Figura 5-23-B

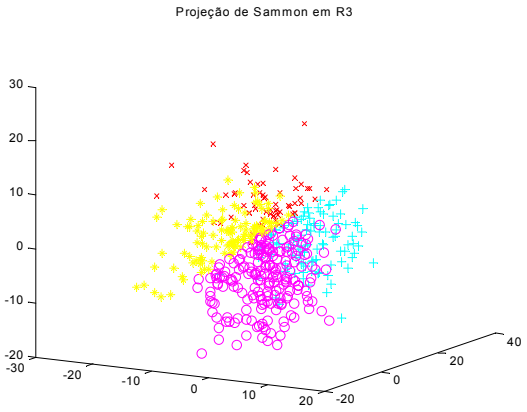
A origem dos eixos AC-CE e AE-RO é obtida por Kolb (2000b) a partir de um conjunto de 1446 adultos entre 18 e 60 anos (638 homens e 801 mulheres com etnias distintas e representando um grande número de áreas de atuação), com média de dois anos de freqüência no ensino superior. Tomando a mesma origem dos eixos, os dados coletados na Universidade São Francisco (USF) no ano de 1988 referem-se ao campus de Itatiba e correspondem a 488 alunos universitários ingressantes[1] nos cursos de Análise de Sistemas, Ciência da Computação e Engenharia (Elétrica, Mecânica, Civil e de Computação). A Tabela 5-6 apresenta os valores médios obtidos para os indicadores.

[1] O conjunto original possui 509 elementos, sendo que 21 dados foram excluídos devido ao preenchimento indevido do teste *LSI-3*.

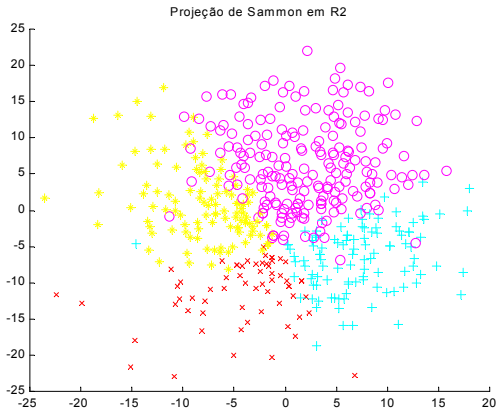
Tabela 5-6 – Valores médios da amostra de dados da Universidade São Francisco (USF), com 488 indivíduos, e do teste *LSI-3* (Kolb, 2000a,b), com 1446 indivíduos. O desvio padrão do teste LSI-3 mostra que os índices CE, RO, AC e AE da USF a colocam no intervalo entre aproximadamente ½ desvio padrão à direita e à esquerda da média. Isto significa que a USF pertence ao conjunto de 38% do total de casos com pontuação semelhante.

	USF	LSI-3	Desvio padrão
CE	23,91	26,00	6,8
RO	31,58	29,94	6,5
AC	31,80	30,28	6,7
AE	32,68	35,37	6,9
AE-RO	1,14	5,92	11,0
AC-CE	7,87	4,28	11,4

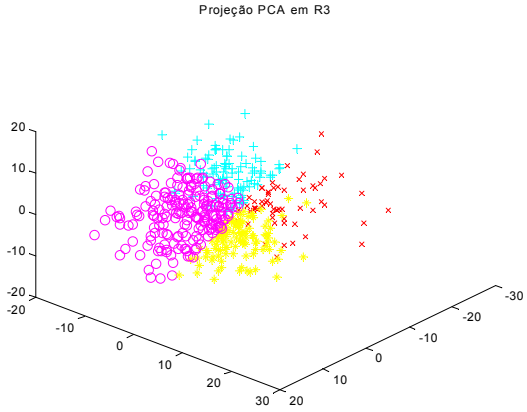
O conjunto de estilos de aprendizado possui assim um total de 488 perfis definidos por vetores de atributos compostos pelos índices CE, RO, AC e CE e residindo, portanto, em $\Re^4$. O conjunto é separado em 4 classes, sendo 204 objetos classificados sob o perfil de Assimiladores (41,8%), 108 como Convergentes (22,1%), 62 como Acomodadores (12,7%) e 114 como Divergentes (23,4%). Para uma análise preliminar do conjunto, foram aplicadas algumas ferramentas de mineração de dados, como propostas no Capítulo 2, cujos resultados podem ser verificados na Figura 5-24. Os rótulos foram atribuídos segundo a regra de Kolb, exemplificada na Figura 5-23 e não foram disponibilizados durante a adaptação dos algoritmos, tendo sido usados apenas para avaliar o resultado final das ferramentas.



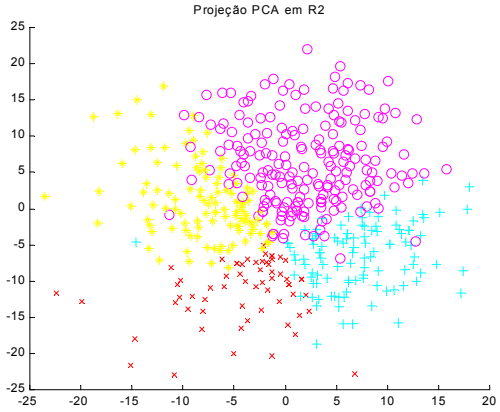
(A)



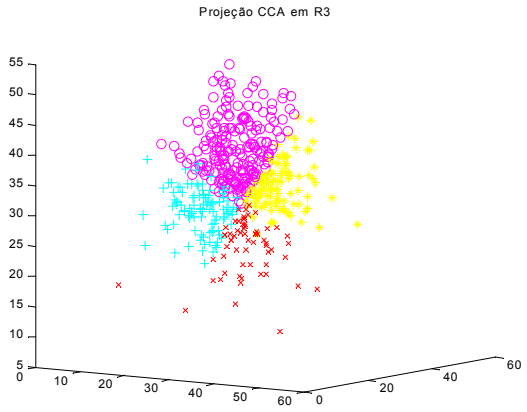
(B)



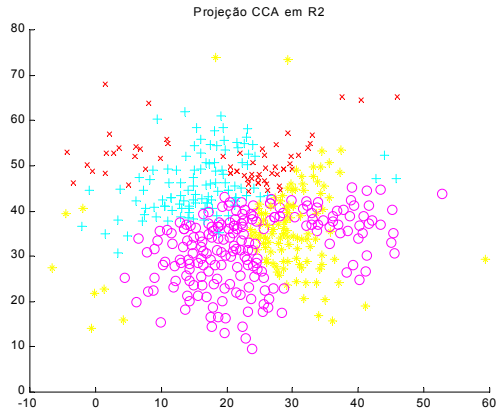
(C)



(D)



(E)



(F)

Figura 5-24 – Aplicação das ferramentas de projeção de Sammon (A, B), PCA (C, D) e CCA (E, F) em 2 e 3 dimensões. As classes são assim identificadas: Convergentes (+ ciano), Assimiladores (o magenta), Divergentes (* amarelo) e Acomodadores (x vermelho). Os rótulos não foram disponibilizados durante a adaptação dos modelos.

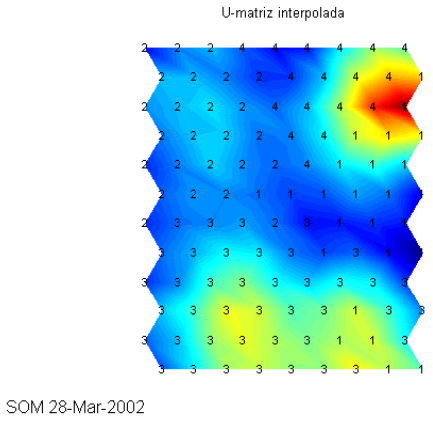
Como pode ser percebido, o conjunto em $\Re^4$ é relativamente bem comportado, embora dificilmente os quatro agrupamentos propostos possam ser verificados sem informação prévia. O que as ferramentas sugerem, entretanto, é que o conjunto possui um formato aproximadamente hiperesférico, o que é uma informação importante para o uso do SOM, indicando que o mapa não seja demasiado alongado.

Com base nessa informação, foram realizados 10 experimentos com o SOM considerando arranjos relativamente proporcionais, conforme pode ser visto na Tabela 5-7. O critério de escolha é o mesmo proposto na Seção 5.2.

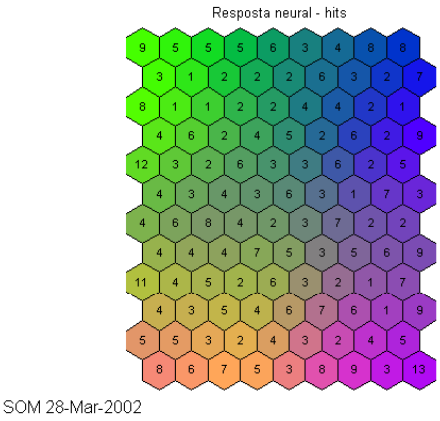
Tabela 5-7 - Conjunto de testes SOM para o conjunto "Alunos". A primeira configuração é a sugestão conforme a literatura, marcada em cinza ( ). Marcado em laranja ( ) estão as configurações candidatas (os menores TEs), e em verde ( ), a configuração escolhida, com QE intermediário.

Conjunto *"Alunos"*								
Configuração					Normalizados		Não normalizados	
Dimensão	Fase 1		Fase 2		QE	TE	QE	TE
	Épocas	Raio	Épocas	Raio				
12 × 09	3	3→1	20	1→1	0,593405	0,024590	3,421812	0,034836
15 × 12	5	4→1	25	1→1	0,502226	0,045082	2,903573	0,049180
15 × 15	5	4→1	25	1→1	0,472947	0,034836	2,656622	0,032787
20 × 20	5	4→1	30	1→1	0,373392	0,034836	2,081186	0,047131
12 × 15	5	4→1	25	1→1	0,506087	0,018443	2,853231	0,030738

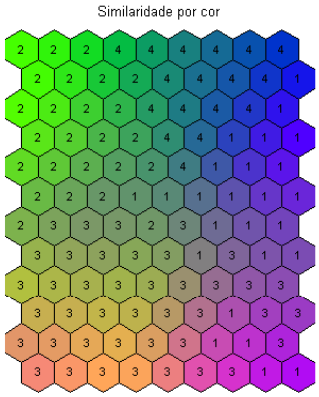
O modelo SOM escolhido, portanto, é um arranjo plano de 12×9 neurônios com vizinhança hexagonal, inicializado linearmente ao longo da distribuição dos dados normalizados, e treinado pelo algoritmo *"batch"* em duas fases: a primeira com 3 épocas com vizinhança regredindo de 3 a 1; e a segunda com 20 épocas e vizinhança fixa em 1. Os dados foram normalizados e os resultados podem ser observados na Figura 5-25.



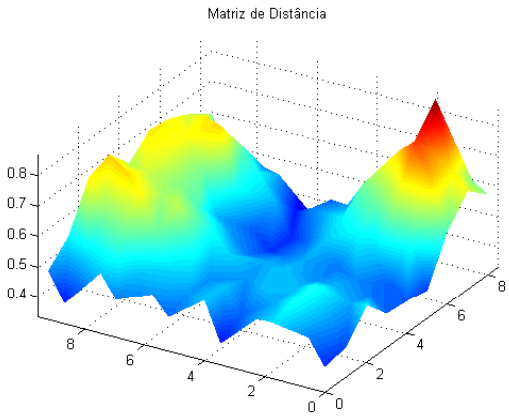
(A)



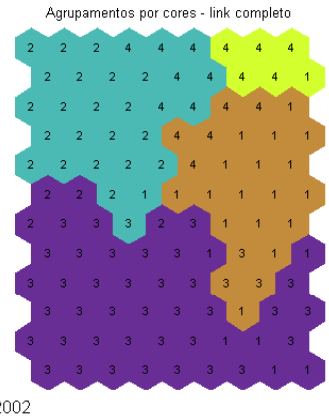
(B)



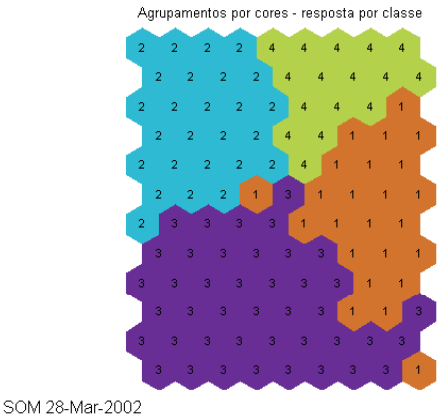
(C)



(D)



(E)



(F)

Figura 5-25 - Resultados SOM do conjunto "Alunos". '1' = Divergentes, '2' = Convergentes, '3' = Assimiladores, '4' = Acomodadores. Os rótulos foram plotados apenas para avaliar os resultados, não sendo disponíveis durante a adaptação do modelo.

A análise da Figura 5-25-B sugere uma distribuição relativamente uniforme dos objetos sobre o arranjo SOM, com poucos neurônios inativos. Isto significa que a dimensão do

mapa é adequada, embora alguns neurônios estejam sobrecarregados pela representação de muitos dados. A análise da matriz-U em (A) e (D), esta última mostrando claramente a existência de um vale, não evidencia agrupamentos. A similaridade por cores (veja a Seção 3.1.2.2 para detalhes) em (C) tampouco revela os 4 supostos grupos com clareza. A observação de (E), representando os grupos através da aplicação de um algoritmo hierárquico de ligação completa (veja a Seção 2.3.1), e (F), que agrupa os neurônios conforme o rótulo do objeto mais próximo de cada um, já oferece uma pista mais clara da existência de agrupamentos distintos, embora essa informação não seja disponível *a priori*. Isto garante, no entanto, que a ferramenta modelou de forma adequada a topologia do espaço de dados.

O algoritmo GTM foi aplicado sobre o mesmo conjunto de dados, foram obtidos os resultados apresentados na Tabela 5-8. O modelo GTM escolhido, segundo o critério já discutido de maior logL, é um arranjo de 30 × 30 pontos latentes com 15 × 15 funções-base com desvio padrão (σ_ϕ) igual a 0,5, fator de regularização 0,001 e adaptado em 30 ciclos. Os resultados do modelo escolhido podem ser vistos na Figura 5-26.

Tabela 5-8 - Resultados de testes para o algoritmo GTM. A configuração escolhida é aquela com maior valor do logaritmo da verossimilhança, marcada em verde ( ).

Conjunto *"Alunos"*								
Configuração					Normalizados		Não normalizados	
Pontos Latentes	Funções Base	σ_ϕ	Fator Regular.	Ciclos	logL inicial	logL final	logL inicial	logL final
30 × 30	15 × 15	0,5	0,001	30	-2848,970493	-690,020662	-6194,571389	-4180,972187
		0,8			-2849,022231	-729,729176	-6194,626828	-4028,905130
		1,0			-2849,024219	-791,186693	-6194,628817	-4222,670564
		1,2			-2849,027271	-913,858112	-6194,631895	-4464,022992
		1,5			-2849,028680	-1190,600566	-6194,633309	-4786,431093

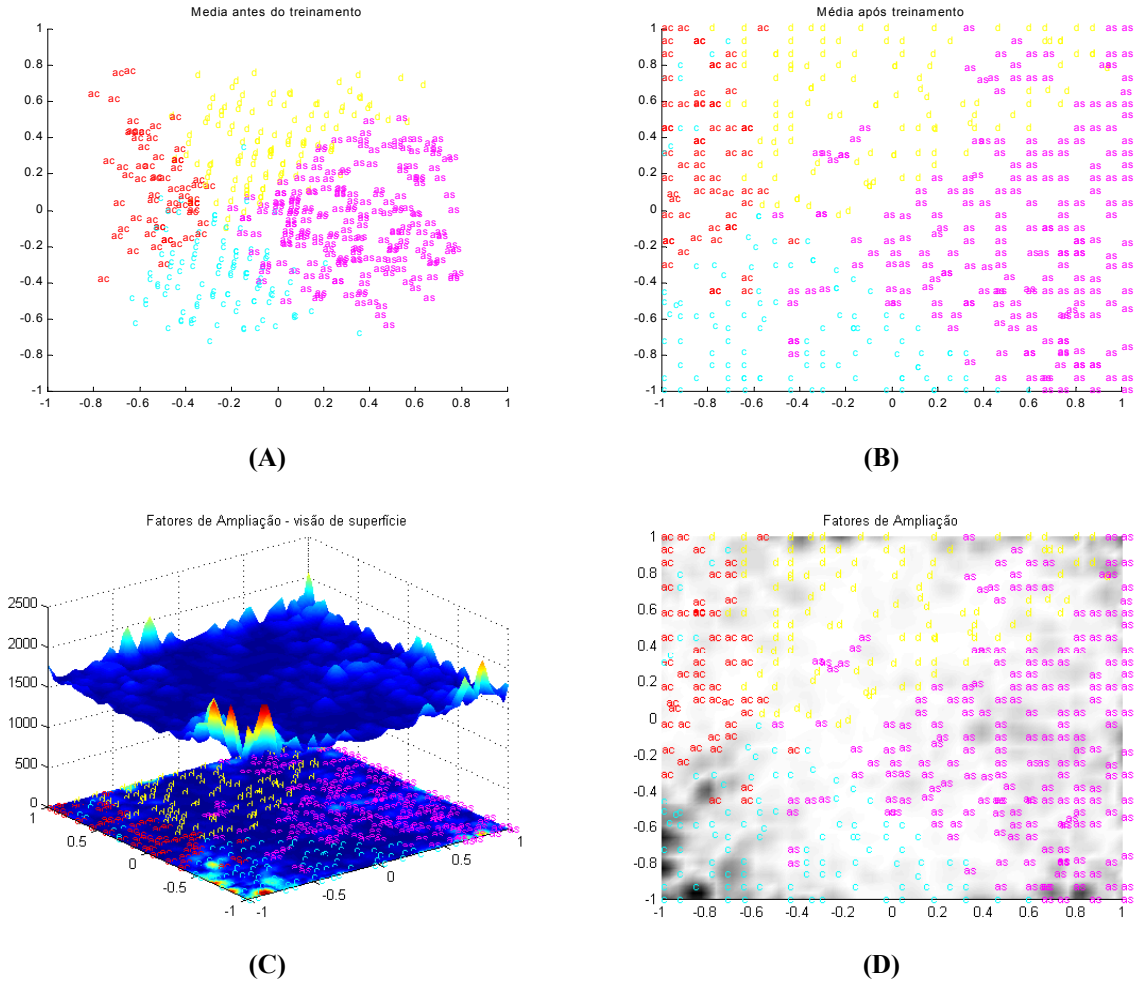


Figura 5-26 - Resultados GTM para o conjunto "Alunos". 'd' = Divergentes, 'c' = Convergentes, 'as' = Assimiladores, 'ac' = Acomodadores.

A observação da Figura 5-26-D mostra que o modelo GTM exibe mais claramente a existência de quatro grupos (há poucas sobreposições de objetos), mas tampouco foi capaz de evidenciar a separação dos mesmos grupos com precisão. Um experimento paralelo utilizando o melhor logaritmo da verossimilhança com dados não normalizados (utilizando σ_ϕ igual a 0,8, um modelo mais "rígido") apresentou os resultados na Figura 5-27. É perceptível a diminuição das classificações equivocadas (B), reforçando a observação feita na Seção 5.2.5. Entretanto, ainda assim não foi possível estabelecer com precisão a separação dos agrupamentos.

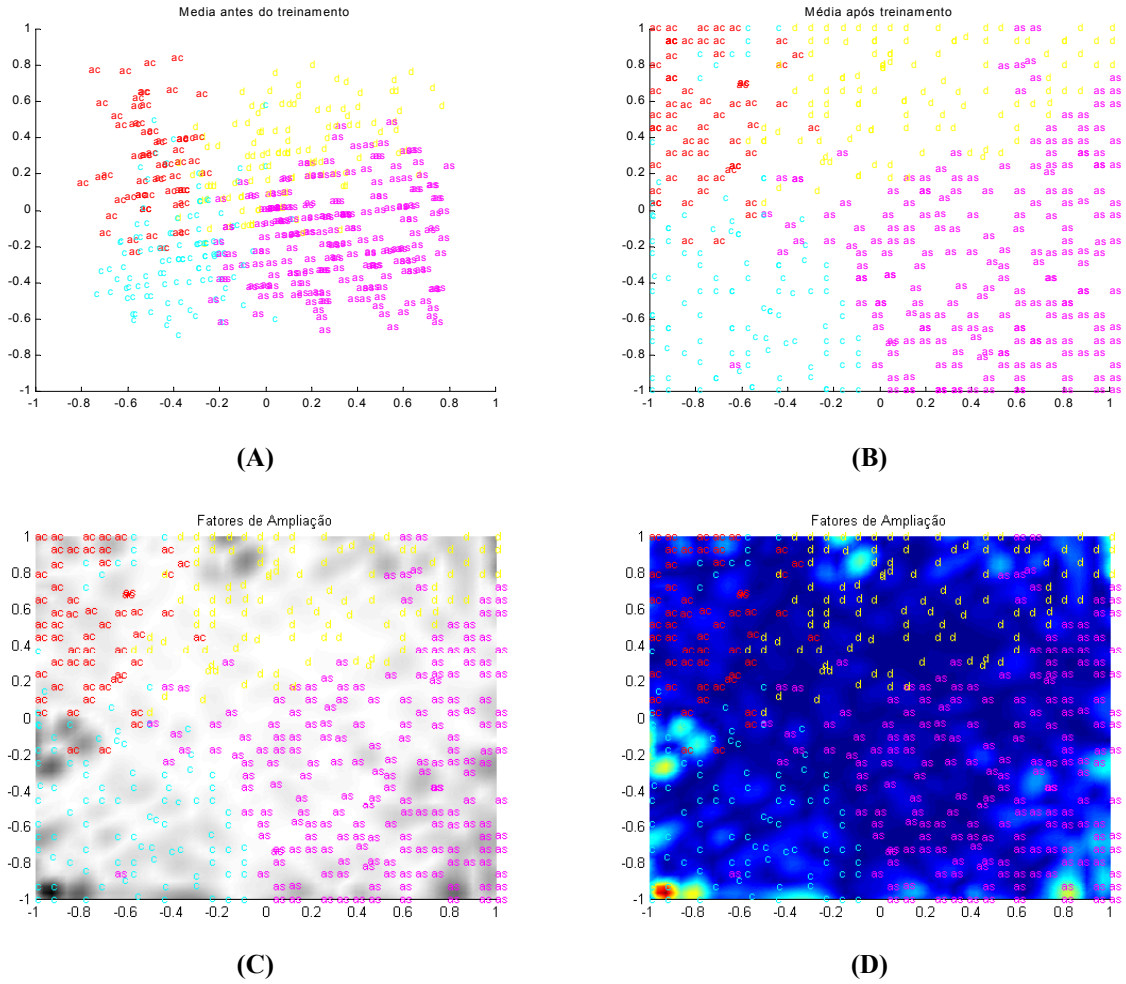


Figura 5-27 - Modelo GTM utilizando o conjunto "Alunos" com dados não normalizados.

O método de classificação e o teste LSI como proposto por Kolb (2000a,b) sugerem haver alta correlação entre as respostas obtidas, uma vez que os índices AC, CE, AE e RO são gerados por uma soma das respostas que evidenciam aquelas características de aprendizado. Esta correlação é observada experimentalmente na Figura 5-28. Foram realizados testes com as ferramentas SOM e GTM para observar seus comportamentos quando aplicados aos dados brutos, isto é, em $\Re^{48}$. Foi feita uma modificação no conjunto que procurou evitar a influência demasiada da característica com valor 4 (com maior influência para o indivíduo que responde o teste) em detrimento da característica com valor 1 (menor influência). Como a métrica do SOM é euclidiana, substituiu-se o conjunto de valores {1, 2, 3, 4} respectivamente pelos valores {-1,5, -0,5, +0,5, +1,5}, objetivando distâncias iguais entre os indicadores.

Após testes e usando os mesmos critérios de escolha já citados, o modelo SOM escolhido foi um arranjo 20×20 com vizinhança hexagonal, treinado em uma fase com 5 épocas e vizinhança de aprendizado variando de 4 a 1, e uma segunda fase com 30 épocas e vizinhança 1. Os dados foram normalizados e os índices de qualidade obtidos foram QE = 4,990765 e TE = 0,010246. O modelo GTM escolhido é um arranjo de 20×20 pontos latentes, 15×15 funções-base com desvio padrão (σ_ϕ) igual a 0.5, fator de regularização 0,001 e adaptado em 10 ciclos, com valores inicial e final para o logaritmo da verossimilhança valendo, respectivamente, -39923,93636 e –21611,83622. O GTM usou dados normalizados, mas ressalta-se aqui que a diferença entre o valor escolhido e o 2º melhor valor (-21674,86072), que usa dados não normalizados, é pequena.

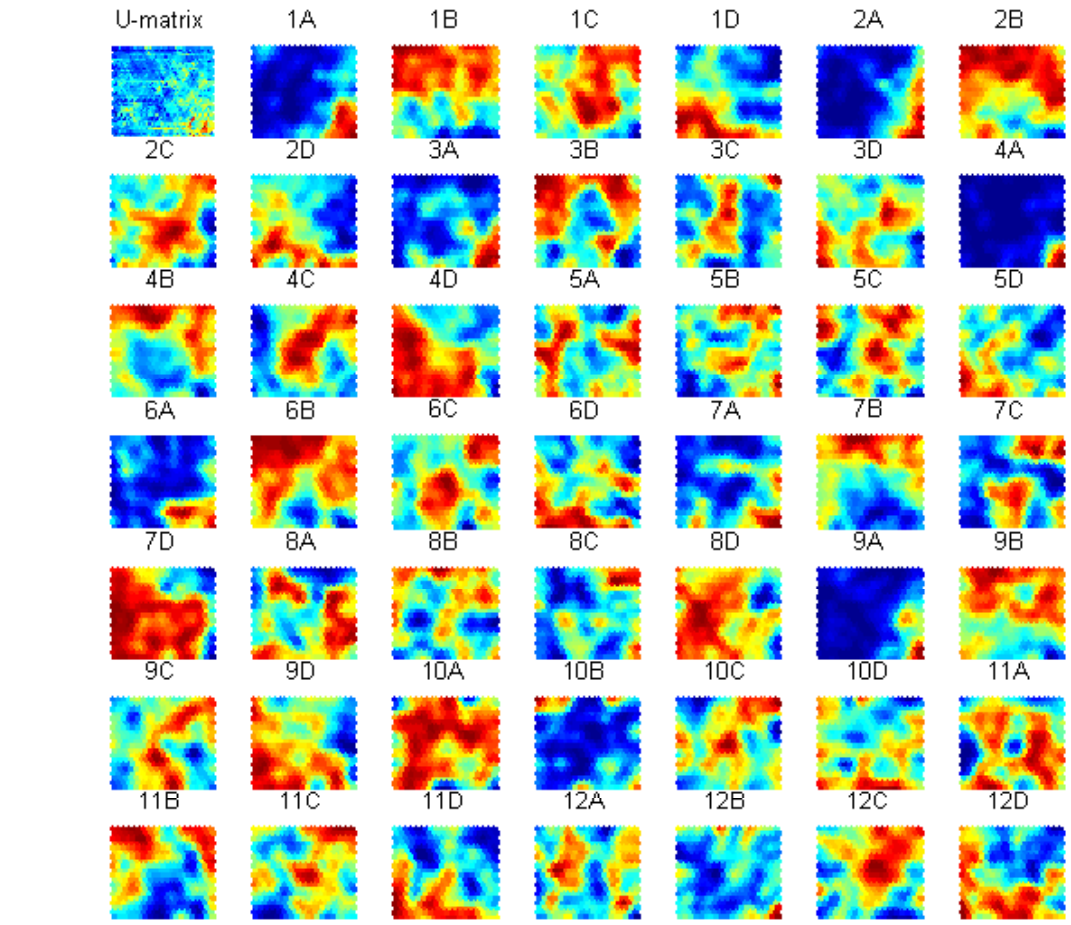


Figura 5-28 – Matrizes-U do conjunto "Alunos" tomando cada uma das 12 questões com suas 4 medições. É perceptível que as matrizes-U das respostas 1A, 2A etc. possuem figuras bastante semelhantes, o que sugere uma correlação positiva destas características.

A Figura 5-28 efetivamente revela a existência de correlação entre as respostas (quadros 1A (resposta 1 coluna A), 2A, 3A, 4A etc.), embora nem todas as respostas exibam correlação evidente (1D, 2D, 3D, 4D etc.).

Na Figura 5-29 pode-se observar os resultados obtidos dos algoritmos SOM e GTM.

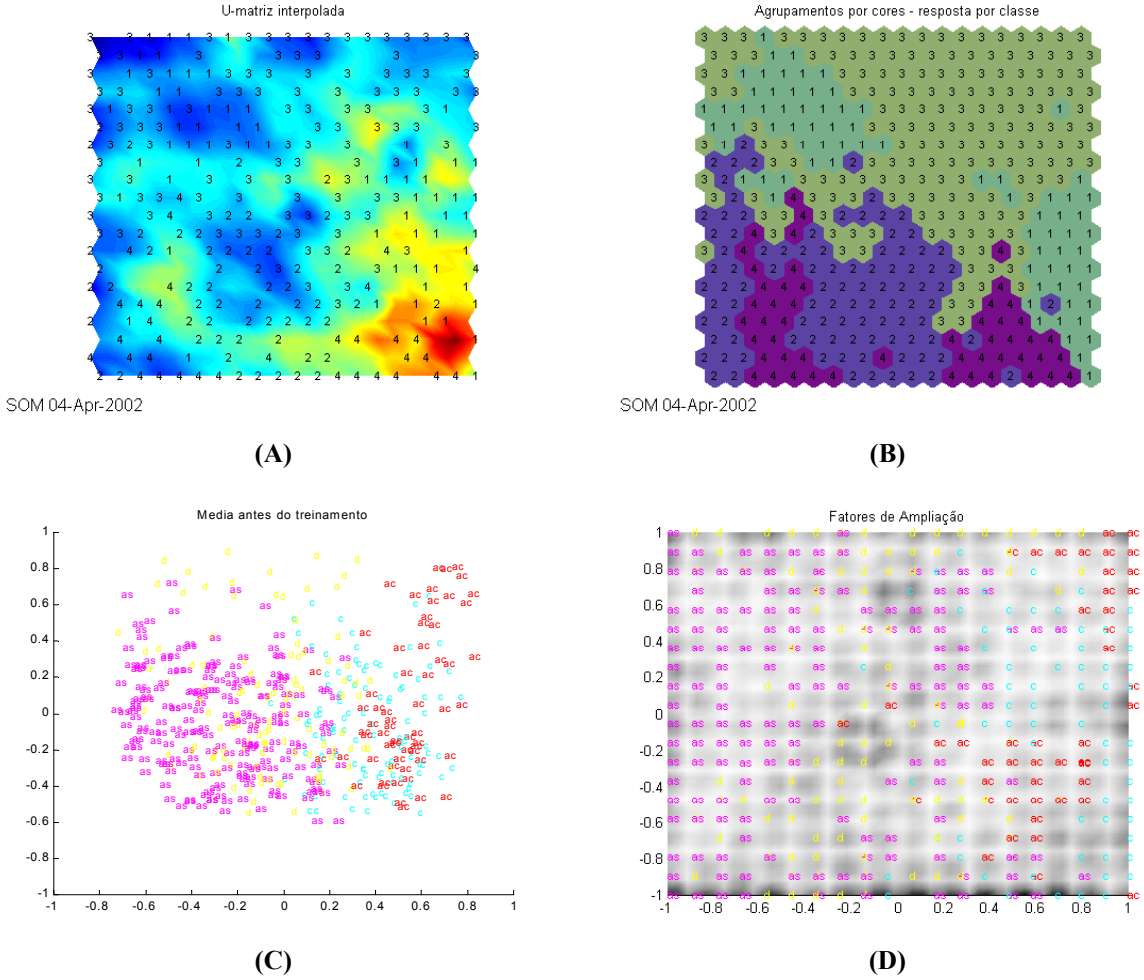


Figura 5-29 - Resultados SOM e GTM aplicados ao conjunto "Alunos" em $\Re^{48}$. Em (A) e (B), o modelo SOM e em (C) e (D), o modelo GTM.

A Figura 5-29 mostra que os dados, observados nesta dimensão, encontram-se bastante sobrepostos, não havendo evidência de agrupamentos distintos.

A suposição de que os objetos "sobrepostos" e "classificados erroneamente" (segundo os critérios propostos por Kolb) fazem parte de um conjunto que está situado nas vizinhanças dos eixos AC-CE e AE-RO foi verificada removendo-se do conjunto os objetos cuja classificação os situasse numa faixa ao longo dos dois eixos. A Figura 5-30 apresenta os resultados obtidos pelo SOM após remover-se uma faixa de ±1 unidade (excluídos 49 dados, 10,041%) e ±4 unidades (com 314 dados removidos, 64,3443%).

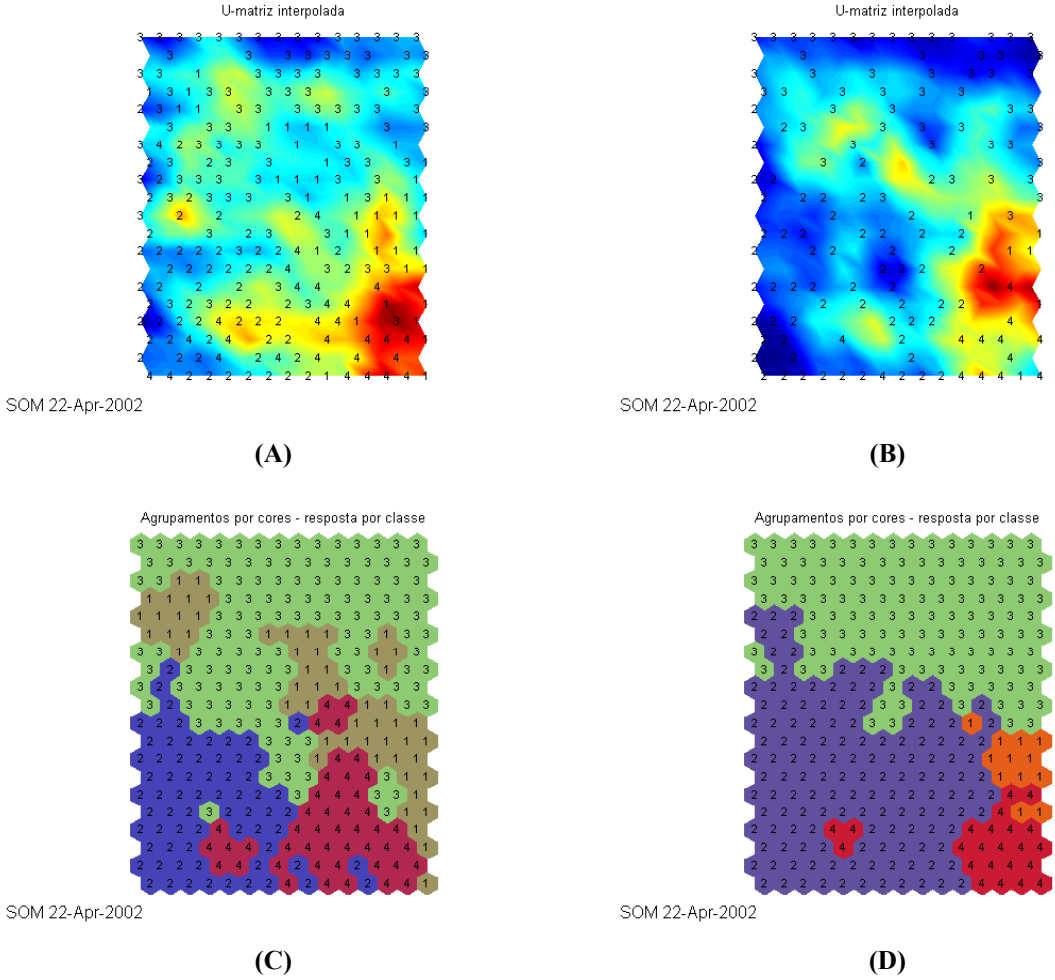


Figura 5-30 - Resultados do SOM após remoção de objetos em regiões próximos aos eixos de classificação. Em (A) e (C) a faixa de remoção é de 1 unidade e em (B) e (D), 4 unidades. Não são evidentes separações entre os agrupamentos.

A Figura 5-30 (C) e (D), com remoção de ±1 e ±4 unidades, respectivamente, mostram que grande parte dos "erros de classificação" foram eliminados com a remoção dos dados que estavam na fronteira de classificação entre agrupamentos distintos, mas a ferramenta SOM não foi capaz de evidenciar agrupamentos ou mesmo uma separação entre eles.

A Figura 5-31 apresenta os resultados obtidos pelo algoritmo GTM. Os resultados obtidos apresentam os conjuntos muito mais definidos que o SOM, neste caso de teste.

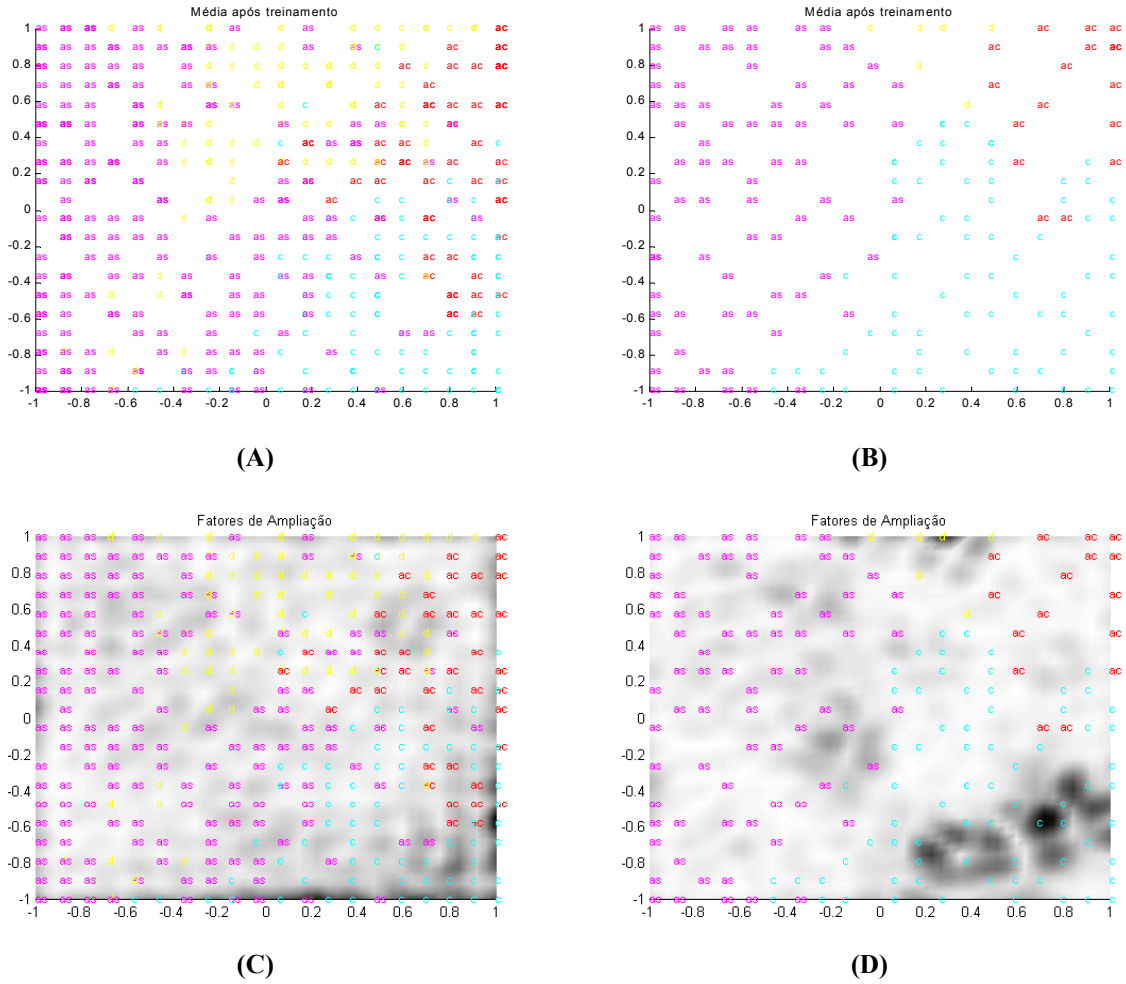


Figura 5-31 - Resultados do GTM após remoção de objetos próximos aos eixos de classificação. Em (A) e (C) a faixa de remoção é de 1 unidade e em (B) e (D), 4 unidades.

Embora nenhuma das duas ferramentas, SOM e GTM, tenham sido capazes de separar os conjuntos (isto é, evidenciar os agrupamentos) conforme a classificação proposta por Kolb (1984), é perceptível que os conjuntos existem. A aparente inabilidade das ferramentas em "confundir" a classificação dos elementos ao longo dos eixos é, de fato, esperada, uma vez que o critério de classificação proposto por Kolb é discreto, linear e igualmente espaçado. Isto significa que a distância entre dois dados *A* e *B*, próximos às fronteiras de dois conjuntos, é igual à distância entre *A* e outros dados na sua vizinhança imediata. Sendo os modelos adaptados em modo não supervisionado, não seria possível a estes evidenciar uma separação entre dados baseados apenas nesta métrica. Seria necessário, nestes casos, providenciar mais informação para os modelos. A Figura 5-32 representa esta constatação.

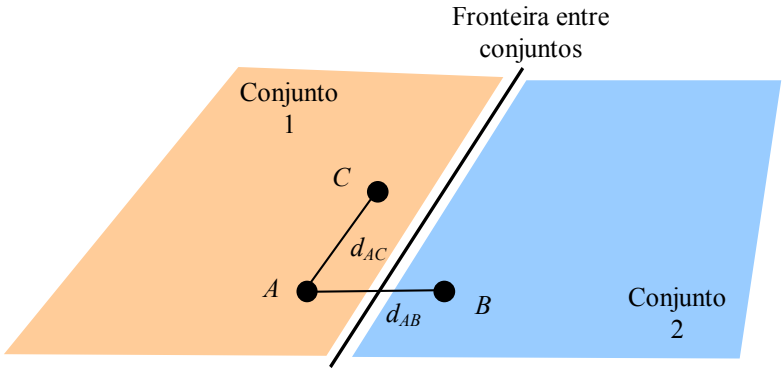


Figura 5-32 – A dificuldade em exibir a separação entre agrupamentos. No caso do conjunto "Alunos", o critério de distância utilizado, a métrica euclidiana, produz $d_{AB} = d_{AC}$. Não é possível para algoritmos baseados apenas nesse critério decidir que *A* e *C* são mais semelhantes entre si que *A* e *B*.

Apesar das dificuldades em separar adequadamente os agrupamentos, os resultados mostram que as ferramentas foram surpreendentemente robustas ao confirmar a topologia do espaço de dados, mesmo considerando os mesmos em $\Re^{48}$, ou seja, com a maior parte das características (respostas) apresentando alta correlação entre si. Esta característica é extremamente importante em tarefas de mineração de dados, e coloca as duas ferramentas testadas em destaque como boas opções para a tarefa.

De forma geral, os resultados demonstram que as duas ferramentas, SOM e GTM, oferecem recursos de análise muito superiores aos dos métodos de projeção mais tradicionais. Ambas as ferramentas mostraram-se bastante robustas e consistentes na análise de dados, oferecendo resultados bastante semelhantes.

A favor do SOM estão a possibilidade de avaliação de correlação entre as características que compõem os vetores de dados e a capacidade de redução de dados, possibilitando a análise de conjuntos volumosos, uma dificuldade marcante exibida pela ferramenta utilizada para o modelo GTM, até o momento. O SOM é um modelo largamente pesquisado e usado, enquanto o GTM é pouco utilizado, até o presente momento. A favor do GTM conta, principalmente, a existência de um critério capaz de medir o grau de adaptação do modelo ao conjunto de dados de entrada, o *logaritmo da verossimilhança*. Este critério é embasado em todo o ferramental estatístico já existente, e permite a comparação entre diferentes modelos, o que não é tão simples com o SOM. Também, em geral, o GTM pode

ser aplicado diretamente sobre os dados, não exigindo nenhum cuidado prévio com a normalização das características. De fato, todos os casos de teste com dados normalizados foram sensivelmente piores, para o GTM, quando a normalização foi aplicada, o que faz do GTM uma ferramenta robusta à existência de características potencialmente dominantes, economizando assim um procedimento (a normalização). Deve ficar registrado também que, mesmo para o SOM, a aplicação da normalização não apresentou resultados superiores aos casos sem uso da mesma em vários casos, o que dificulta bastante a opção do pesquisador fazendo uso da ferramenta.

As possibilidades de análises gráficas com o SOM são muito superiores às do GTM, considerando da Toolbox SOM (Alhoniemi *et al.* 2000) e GTM (Svensén, 1999). Isto deve-se, no entanto, ao fato do primeiro ser um modelo largamente mais pesquisado e utilizado que o segundo. O GTM é um modelo de convergência extremamente rápida (os testes efetuados em geral convergiram para resultados aceitáveis em apenas 5 épocas), ao passo que o SOM, em geral, exigiu duas fases de treinamento.

De forma geral, as duas ferramentas testadas ofereceram muito bons recursos de análise de dados, podendo ser utilizadas com bastante proveito em tarefas de mineração de dados.

Capítulo 6

Aplicações em Recuperação de Informação

De acordo com Salton & McGill (1983), "recuperação de informação" está relacionada à representação, armazenamento e acesso a itens de informação. Scholtes (1993) afirma que recuperação de informação requer a aplicação conjunta de técnicas de Processamento de Linguagem Natural e Inteligência Artificial. Segundo estas definições, praticamente todo sistema de informação é um tipo de sistema de recuperação de informação, desde sistemas de banco de dados tradicionais até sistemas baseados em conhecimento, incluindo sistemas de informação gerencial e sistemas de apoio à decisão. Tradicionalmente, entretanto, este termo é relacionado aos métodos de recuperação de documentos de texto contidos em conjuntos de documentos disponíveis.

Com o advento da Internet, a vasta quantidade de dados disponíveis requer ferramentas automáticas para efetuar mineração de dados, a qual representa uma das tarefas do processo de KDD (*Knowledge Discovery in Databases*). No caso particular de dados na forma textual, o cenário que se apresenta é altamente desafiador para qualquer iniciativa de automatização de processos de recuperação de informação, pois estão disponíveis textos de toda natureza, cuja qualidade e propósito são extremamente diversos (Lagus, 2000). Assim, é freqüente o caso de uma busca tradicional recuperar milhares de documentos – grande parte deles de valor seriamente questionável, quando se consideram os objetivos da busca – ou nenhum documento, devido a um critério muito restritivo. Sendo assim, processos de calibração de filtros eficazes para recuperação de informação relevante são altamente desejáveis.

Este capítulo retrata a síntese, aplicação e análise de resultados obtidos com uma proposta de filtro para recuperação de informação, buscando detalhar os métodos de codificação,

organização e recuperação de informação de documentos textuais. Para avaliação prática e após considerar vários casos ilustrativos presentes na literatura, foram utilizados dois conjunto de textos em língua portuguesa. O primeiro conjunto é constituído 52 textos envolvendo dois assuntos bastante distintos entre si, esportes e culinária, com objetivo de avaliar a capacidade das ferramentas em separar estes assuntos. O segundo conjunto constitui-se dos resumos de 161 publicações científicas do II Congresso de Pesquisa e Extensão, ocorrido de 6 a 8 de outubro de 1999 na Universidade São Francisco, campus de Bragança Paulista. Este último conjunto apresenta uma tarefa bem maior se comparado ao primeiro conjunto, e motivou a contribuição de uma nova técnica capaz de melhorar os resultados obtidos.

6.1 Recuperação de Informação aplicada a documentos textuais

Considerando que grande parte do conhecimento humano é expresso na forma textual, em formato de livros, artigos, páginas da Internet etc. (doravante generalizados pelo termo "documentos"), entende-se que o termo "recuperação de informação" aplicado a documentos relaciona-se com a tarefa de satisfazer a *necessidade de informação* do indivíduo (Lagus, 2000). A necessidade de informação pode ser entendida como a busca de respostas para determinadas questões ou problemas a serem resolvidos, a recuperação de um documento com particularidades específicas, a recuperação de documentos que versem sobre determinado assunto ou ainda o relacionamento entre assuntos. Embora relativamente simples em seu conceito, esta tarefa envolve questões bastante difíceis de serem resolvidas:

- como se deve armazenar o conjunto de documentos de forma a preservar e evidenciar seu conteúdo e o relacionamento entre os mesmos?
- uma vez armazenados, como recuperá-los eficientemente de forma a satisfazer a necessidade de informação de um indivíduo?

A forma de armazenamento dos documentos é crucial e intrinsecamente determina os métodos possíveis de recuperação dos mesmos. A recuperação de documentos envolve ainda critérios subjetivos, o que sugere métodos interativos. Normalmente, são consideradas duas possibilidades de encaminhamento para estas questões: (1) considerar a natureza estatística da ocorrência das palavras dentro de um documento, levando-se ou não

em conta sua ordem (modelo do *saco de palavras*, do inglês *"bag of words"*); ou (2) utilizar a abordagem simbólica da linguagem natural (Scholtes, 1991b).

A primeira abordagem reduz o documento a alguma forma estatística de representação do texto (vetores de freqüência de palavras, dicionário de termos (*thesaurus*), palavras-chave etc.), o que leva ao conceito de reconhecimento de padrões. Costuma ser uma abordagem rápida, computacionalmente eficiente e pode lançar mão do conhecimento e de ferramentas estatísticas já bem fundamentadas na literatura. Entretanto, é incapaz de considerar relações simbólicas ou de efetuar inferências conceituais sobre os documentos. A segunda abordagem, ao considerar a natureza simbólica da linguagem, é teoricamente capaz de lidar com as deficiências do primeiro método, mas costuma ser computacionalmente complexa e ineficiente. Normalmente, lança-se mão de cadeias de Markov de palavras ou caracteres, mas a "memória" do método depende da ordem da cadeia de Markov e os requisitos computacionais crescem exponencialmente com o aumento da cadeia (Scholtes, 1991b).

Os métodos clássicos de armazenamento e recuperação de documentos baseiam-se nestas duas abordagens. Embora exista ainda a forma mais tradicional de todas, a classificação manual, esta é viável apenas em conjuntos reduzidos de textos e é suscetível a aspectos subjetivos, particularmente quando se procura indexar obras que envolvam várias áreas de conhecimento simultaneamente (Salton & McGill, 1983). O processo de armazenamento ou representação, também chamado *indexação*, busca extrair características do documento que permitam seu armazenamento de forma resumida. O processo de recuperação é booleano (considera a existência ou não de *índices* ou palavras-chave dentro dos documentos) ou por alguma métrica de distância envolvendo a pergunta feita e o conjunto dos índices armazenados. Todos estes métodos sofrem de problemas comuns (Scholtes, 1993):

- dificuldade para processar perguntas indiretas ou incompletas;
- dificuldade para manipular intenções vagas de busca (i.e., quando o usuário não conhece exatamente o tópico sobre o qual procura informação);
- ausência de habilidade de realimentação da busca em função do resultado obtido previamente;
- ausência de vínculos mais efetivos com o contexto da linguagem, pois são consideradas apenas algumas relações seqüenciais entre palavras;

- dificuldade na recuperação de documentos por similaridade contextual.

Na tentativa de resolver alguns destes problemas, as redes neurais artificiais são promissoras para a pesquisa, pois exibem capacidades de aprendizado, generalização e sensibilidade a alguns aspectos contextuais necessários para o cenário da recuperação de informação baseada em documentos de texto (Scholtes, 1993). Vários experimentos estão presentes na literatura envolvendo o uso de redes neurais.

A seção a seguir discute brevemente alguns dos métodos mais importantes ou de reconhecido valor histórico no contexto de armazenamento e recuperação de documentos, alguns dos quais são baseados em redes neurais artificiais.

6.2 Métodos de Armazenamento e Recuperação de Documentos

A comparação de desempenho de métodos de recuperação de documentos é tradicionalmente baseada em duas métricas, *precisão* e *recuperação* (Salton & McGill, 1983):

$$\text{Precisão} = \frac{\text{Nº de documentos relevantes recuperados}}{\text{Nº total de documentos recuperados}} \qquad \textbf{Equação 6-1}$$

$$\text{Recuperação} = \frac{\text{Nº de documentos relevantes recuperados}}{\text{Nº total de documentos relevantes existentes}} \qquad \textbf{Equação 6-2}$$

Infelizmente, é muito difícil comparar métodos de recuperação de documentos devido à *subjetividade* envolvida na avaliação. A "relevância" de um documento é altamente subjetiva, pois dependente do conhecimento prévio do usuário e é sempre relativa a outros documentos recuperados. Ainda, em geral, aqueles que procuram uma informação podem não saber exatamente o que procurar devido, possivelmente, ao próprio desconhecimento sobre o assunto (Scholtes, 1993).

Em função do exposto acima, dificilmente pode-se esperar que um método em particular resolva todos os dilemas da área e é mais provável que modelos híbridos apresentem melhor desempenho que os modelos básicos. Lagus *et al.* (1996a,b) e Lagus (2000) afirmam que a necessidade de informação está orientando o desenvolvimento de

ferramentas interativas, visuais, capazes de oferecer uma visão geral do relacionamento do conjunto de documentos. Também estas ferramentas devem oferecer possibilidades de busca e navegação por entre os documentos, oferecendo níveis de detalhes que podem ser escolhidos pelo usuário (uma espécie de "*zoom*"). Pullwitt (2002) critica as métricas de precisão e recuperação, úteis para classificação mas não para análise de proximidade contextual entre documentos. Infelizmente, não há ainda na literatura métricas efetivas que permitam estabelecer com precisão a qualidade de métodos de recuperação de informação, embora muitos métodos já foram propostos e vêm sendo empregados na prática, cada qual com as suas potencialidades e limitações, dentre as já levantadas acima.

6.2.1 Modelo booleano

No modelo booleano (Salton & McGill, 1983), cada documento é indexado por uma coleção de palavras extraídas do documento. Estes índices são palavras cujo *valor discriminante* é elevado. O *valor discriminante* mede a capacidade de uma palavra em identificar um documento como relevante e separá-lo de outros não relevantes numa busca. De acordo com o "princípio do menor esforço", as pessoas tendem a usar termos repetidos em vez de criar novos termos para expressar idéias. Este mesmo princípio comprova que as palavras mais freqüentes num documento (e até mesmo na fala) carregam pouco significado na expressão da idéia, sendo utilizadas como elementos de ligação numa frase (Salton & McGill, 1983, pg. 60).

Seguindo esta abordagem, o valor discriminante depende da freqüência com que a palavra ocorre, ou em um documento ou no conjunto de documentos. Assim, pode-se considerar duas medidas de freqüência absoluta relacionadas entre si:

$$freqtot_k = \sum_{i=1}^{n} freq_{ik} \qquad \textbf{Equação 6-3}$$

onde n é o número de documentos, $freq_{ik}$ é a freqüência com que a palavra k aparece no documento i e $freqtot_k$ é a freqüência total de ocorrência da palavra k no conjunto de documentos. Assim, palavras que ocorrem com freqüência muito alta ou muito baixa não são bons discriminantes e não deveriam ser consideradas como índices possíveis (Luhn, 1958).

Por outro lado, palavras com baixa freqüência de ocorrência podem ser entendidas como a representação de termos novos, ou seja, termos cunhados para discutir novos assuntos. Sob este ponto de vista, descartar tais palavras teria efeito negativo sobre a qualidade do mapa, tornando-o pouco sensível a novos termos (ou termos pouco freqüentes). Nesta dissertação foi utilizada esta última abordagem, ou seja, as palavras com baixa freqüência de ocorrência não foram descartadas, como sugere a literatura tradicional, mas consideradas como (possivelmente) relevantes. A Figura 6-1 ilustra esta idéia.

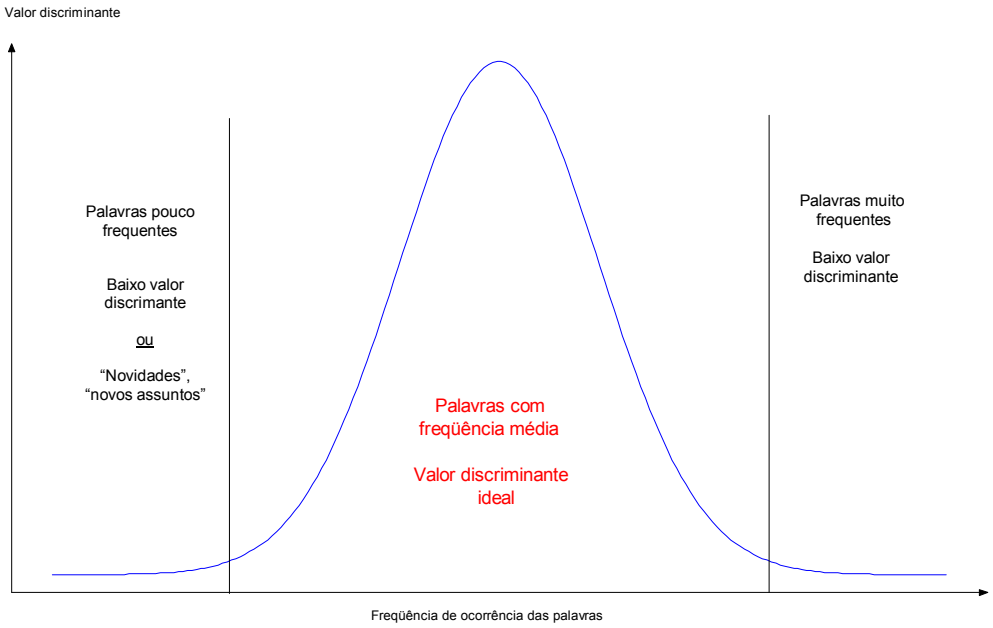


Figura 6-1 – Variação do valor discriminante de um termo em relação à freqüência de ocorrência deste no conjunto de documentos. Os limiares de corte são escolhas heurísticas e dependem do conjunto de documentos. Adaptado de Salton & McGill (1983, pg. 62).

Escolhidas as palavras que comporão o índice, um vetor é associado a cada documento, onde cada dimensão corresponde a um índice, contendo por exemplo "1" e "0" conforme o índice esteja presente ou ausente no documento.

Uma operação de busca consiste na formulação de uma expressão booleana que é aplicada sobre o conjunto de índices. Embora seja um modelo muito usado por sua simplicidade, há vários inconvenientes com esse método:

- é difícil realizar uma busca adequada (isto é, que recupere documentos relevantes) especialmente quando o usuário não domina o conjunto de palavras-chave (índices) do assunto em questão;
- não há forma de obter resultados aproximados, isto é, a busca ou é bem sucedida ou é mal sucedida, sendo comum a recuperação de milhares ou nenhum documento;
- não havendo um critério de aproximação, não há como classificar os documentos recuperados segundo sua relevância.

6.2.2 Modelo de espaço vetorial

O modelo de espaço vetorial (Salton & McGill, 1983) é uma variação do modelo booleano. Diferentemente deste, onde apenas a freqüência absoluta de ocorrência de uma palavra é considerada, o modelo de espaço vetorial busca privilegiar palavras que ocorrem de forma concentrada em alguns textos (mesmo que a freqüência absoluta destas palavras seja elevada em relação ao conjunto de documentos).

Neste modelo, cada documento é representado por um vetor em que cada dimensão corresponde à *freqüência relativa de ocorrência* de uma determinada palavra dentro deste mesmo documento (diferentemente do modelo booleano, onde apenas a presença ou ausência da palavra é considerada). Agora, o valor discriminante de uma palavra é considerado *proporcional* à freqüência relativa de ocorrência da palavra no documento e *inversamente proporcional* ao número de documentos do conjunto em que esta aparece. Assim, palavras menos freqüentes e concentradas em alguns documentos são boas candidatas para identificar um texto em particular e isto pode ser expresso por

$$TF_{ik} = \frac{freq_{ik}}{|d_i|} \qquad \textbf{Equação 6-4}$$

onde TF_{ik} (*TF: Term Frequency*) é a *freqüência do termo k* no documento *i* e $|d_i|$ é o número de palavras presentes no documento *i*. Já aquelas palavras que aparecem em muitos textos de maneira mais ou menos uniforme têm menor valor discriminante e uma possível expressão para este conceito é

$$IDF_k = \log_2\left(\frac{n}{freqdoc_k}\right) + 1 \qquad \textbf{Equação 6-5}$$

onde IDF_k (*IDF: Inverse Document Frequency*) é o inverso da freqüência de ocorrência do termo k em relação ao total de documentos, n, e $freqdoc_k$ é o número de documentos nos quais a palavra k é encontrada pelo menos uma vez.

Uma equação para ponderação de cada palavra no vetor representante dos documentos é sugerida por Salton & McGill (1983) sendo conhecida como *TF-IDF* (*Term Frequency – Inverse Document Frequency*):

$$w_{ik} = TF_{ik} \times IDF_i \qquad \textbf{Equação 6-6}$$

onde $w_{ik} \in \Re$ é o valor discriminante da palavra k no documento i.

A busca é executada calculando-se a distância entre o vetor representando o texto de busca e os vetores representantes dos documentos, recuperando os mais próximos (dentro de um intervalo dado) ordenadamente. Uma vantagem do modelo é a possibilidade de aplicação direta de algoritmos baseados em métricas de distância vetorial. Porém, a dimensão dos vetores representantes torna-se impraticável para conjuntos reais de textos, dada a grande quantidade de palavras envolvidas.

6.2.3 Indexação Semântica Latente

Uma forma de minimizar o número de dimensões do modelo de espaço vetorial original é o método de *Indexação Semântica Latente* (*LSI: Latent Semantic Indexing*) (Deerwester *et al.* 1990). Este método busca encontrar as correlações entre os padrões de ocorrência das palavras dentro dos documentos, mantendo apenas os padrões independentes com maior variância e descartando padrões com menor variância, na hipótese de que estes seriam menos relevantes para identificação do contexto.

Os vetores, cada qual representante de um documento (histograma de freqüência de palavras) são arranjados em uma matriz em que cada coluna corresponde a um vetor, cada vetor podendo ou não ser ponderado pelo critério TF-IDF. Sobre esta matriz é aplicado o método de *Decomposição em Valores Singulares* (*SVD: Singular Value Decomposition*). Este processo executa uma redução dimensional gerando uma matriz que é uma projeção da relação "palavras × documentos", a qual considera apenas os padrões mais relevantes entre

os documentos. Esta nova matriz tem, em geral, dimensão muitas vezes menor que a matriz original dos histogramas de palavras e os cálculos de distância entre os documentos são realizados sobre esta matriz. Uma desvantagem deste método é tornar-se computacionalmente caro à medida que aumenta a dimensão dos histogramas de palavras (Lagus, 2000).

6.2.4 SOM Semântico

Ritter & Kohonen (1989) demonstraram num experimento prático que o SOM é capaz de representar graficamente a relação entre valores simbólicos (palavras, no caso) através de uma codificação apropriada do contexto em que se encontram. O trabalho tem forte inspiração biológica e parte da hipótese fundamental proposta por Aristóteles a respeito do conhecimento: a abordagem por "categorias". As mais conhecidas categorias são: (a) Objetos, (b) Propriedades, (c) Estados e (d) Relações. As linguagens, segundo esta teoria, representam a categoria (a) através de substantivos, a categoria (b) por adjetivos, a categoria (c) por verbos e a categoria (d), conforme a linguagem, por advérbios, preposições, construções sintáticas, ordem das palavras etc. Sob a hipótese de que tais representações das categorias são comuns a todas as linguagens (Ritter & Kohonen, 1989), é possível supor que uma rede neural capaz de apreender tais relações pode tornar-se uma ferramenta poderosa no processamento de linguagem natural e na representação de contextos.

A proposta é representar um conjunto de palavras por vetores de forma que seu significado semântico seja, de alguma forma, capturado pelo mapa neural e que, portanto, símbolos "semanticamente próximos" sejam mapeados "topograficamente próximos". Palavras são particularmente difíceis de serem representadas em forma vetorial. Para tanto, assume-se que a palavra em si não carrega seu significado, mas este depende principalmente do *contexto em que ela está inserida*. A forma sugerida e mais simples de representação de símbolos semânticos é um vetor composto pela concatenação de outros dois, um representando respectivamente o símbolo em si (a palavra) e outro, o conjunto de atributos (contexto) associado ao símbolo:

$$\mathbf{v}_k = \begin{bmatrix} \mathbf{v}_{a(k)} \\ \mathbf{v}_{s(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_{a(k)} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{v}_{s(k)} \end{bmatrix} \qquad \textbf{Equação 6-7}$$

onde $\mathbf{v}_k$ é o vetor que representa o contexto da palavra *k*, sendo composto por dois outros vetores, $\mathbf{v}_{s(k)}$ representando o vetor atribuído ao símbolo *k* (palavra) e $\mathbf{v}_{a(k)}$ representando o vetor de atributos associados ao símbolo *k* (isto é, seu *contexto*).

Dessa forma, se a norma da parte relativa aos atributos for maior que a parte representando o símbolo durante o treinamento do mapa, então símbolos com atributos parecidos serão mapeados por neurônios com vetor de pesos próximos entre si. Também, como há informação a respeito do símbolo em si (a palavra), uma referência a este é também codificada. Assim, por exemplo, durante o reconhecimento de dados de entrada, se os valores dos atributos relativos a um símbolo forem ruidosos ou mesmo ausentes, o mapa poderá responder baseado apenas na informação do símbolo em si.

Os atributos podem ser de quaisquer tipos, numéricos ou representando características como "bom" ou "ruim". A relação entre símbolos dificilmente é dedutível a partir da *forma* do símbolo: símbolos semelhantes podem carregar significados completamente distintos (por exemplo, "abacaxi" pode significar tanto "fruta" como "confusão") e o contrário também não é verdadeiro, pois símbolos completamente diferentes podem carregar idéias semelhantes (por exemplo, "confusão" e "abacaxi"). Para tanto, os vetores que compõem a parte da informação ao relativa ao símbolo ($\mathbf{v}_s$) são escolhidos de forma que sejam *ortogonais* entre si, ou seja, não carreguem nenhuma informação prévia sobre os mesmos. Um exemplo ilustrativo considera o conjunto de símbolos e atributos como descritos na Tabela 6-1.

Tabela 6-1 – Conjunto de animais com suas respectivas características. Adaptado de Ritter & Kohonen (1989).

		pomba	galinha	pato	ganso	coruja	falcão	águia	raposa	cão	lobo	gato	tigre	leão	cavalo	zebra	vaca
é	pequeno	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	médio	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	grande	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
tem	2 pernas	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 pernas	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	pelos	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	cascos	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	crina/juba	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	penas	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
gostam de	caçar	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	correr	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	voar	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	nadar	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Os símbolos constantes da Tabela 6-1 são então representados pela concatenação de seus atributos e de um código para cada símbolo, como descrito na Tabela 6-2.

Tabela 6-2 – Ao vetor de atributos é concatenado um vetor para a codificação do símbolo. Os vetores para codificação do símbolo são ortogonais entre si, pois não se assume *a priori* qualquer conhecimento a respeito da relação entre os símbolos. Nesta representação, os vetores de atributos estão transpostos em relação à Tabela 6-1.

Código dos atributos $(\mathbf{v}_a)^T$													**Código do símbolo $(\mathbf{v}_s)^T$**														**Símbolo**
1	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	pomba
1	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	galinha
1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	pato
1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	ganso
1	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	coruja
1	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	falcão
0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	águia
0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	raposa
0	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	cão
0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	lobo
1	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	gato
0	0	1	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	tigre
0	0	1	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	leão
0	0	1	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	cavalo
0	0	1	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	zebra
0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	vaca

Durante o treinamento do SOM, a parte $\mathbf{v}_s$ é reduzida por um fator ε normalmente igual a 0,2 para que a parte simbólica não influa em demasia no processo (Ritter & Kohonen, 1989). O resultado pode ser visto na Figura 6-2, onde pode-se facilmente observar que o mapa consegue separar os mamíferos das aves e também os predadores dos animais domésticos.

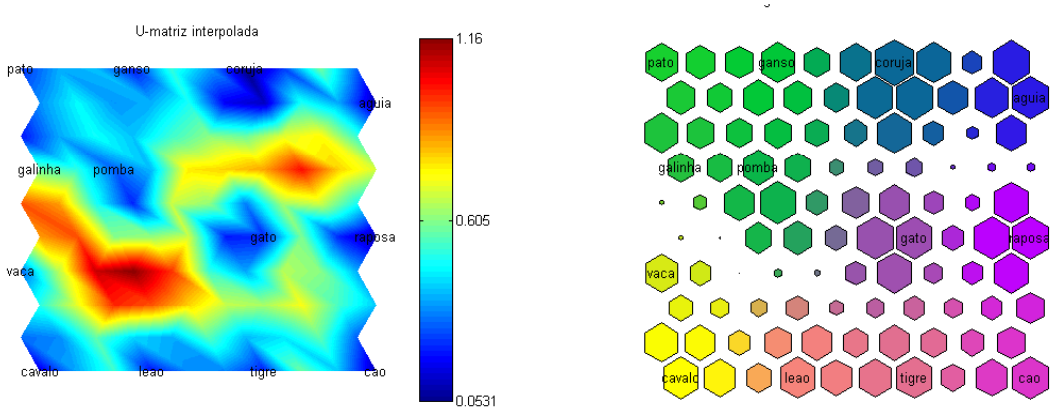


Figura 6-2 – Mapa semântico do SOM em relação aos dados da Tabela 6-2. Em (A) pode-se observar, pela matriz-U, que a separação entre mamíferos (metade inferior) e aves (metade superior) é clara. Em (B), a separação dos grupos é reforçada pela utilização de similaridade por cor baseada na informação da matriz de distância.

Neste experimento, as características de cada símbolo foram codificadas diretamente no vetor representante. Para operar com textos livres é necessária uma forma de codificação que possibilite carregar o contexto do símbolo. A estratégia utilizada por Ritter & Kohonen (1989) é considerar como atributos $\mathbf{v}_a$ de um símbolo $\mathbf{v}_s$ a *média de todos os símbolos sucessores e predecessores* de $\mathbf{v}_s$, o que foi chamado de *contexto médio*. A expressão que representa esta idéia é denotada por

$$\mathbf{v}_k = \begin{bmatrix} \mathrm{E}\{\mathbf{v}_{s(k)-1}\} \\ \varepsilon\ \mathbf{v}_{s(k)} \\ \mathrm{E}\{\mathbf{v}_{s(k)+1}\} \end{bmatrix} \qquad \textbf{Equação 6-8}$$

onde $\mathrm{E}\{\mathbf{v}_{s(k)-1}\}$ é a média de todos os símbolos que precedem a palavra $\mathbf{v}_k$ e $\mathrm{E}\{\mathbf{v}_{s(k)+1}\}$ é a média de todos os símbolos que sucedem a palavra $\mathbf{v}_k$ no corpo de texto. Processando todas as ocorrências das palavras obtém-se um conjunto de vetores $\mathbf{v}$ que é usado para treinar um SOM, chamado SOM Semântico ou "mapa de palavras". Este processo foi abordado

inicialmente por Ritter & Kohonen (1989) e aplicado com sucesso por Lin *et al.* (1991) em um pequeno conjunto de documentos científicos, Scholtes (1991a,b) já enfocando textos livres e Honkela *et al.* (1996a) definindo o método WEBSOM, uma estratégia hierárquica para classificar textos livres, como aqueles encontrados na Internet.

Esta abordagem, entretanto, possui uma séria desvantagem ao considerar textos de tamanho real, pois o número de palavras exigiria vetores (ortogonais) de grande dimensão para codificar cada símbolo, tornando o processo computacionalmente caro.

6.2.4.1 SOM de Documentos

A idéia do SOM de documentos foi primeiramente proposta por Lin *et al.* (1991) numa versão mais simples e ampliada por Honkela *et al.* (1996a) em um modelo hierárquico de dois níveis para abordar o problema de recuperação de informação de documentos de texto. O objetivo final é obter um SOM que organize os documentos de texto conforme sua proximidade contextual. Para tal, o processo tem as seguintes fases, ilustradas na Figura 6-3:

1. É executado um pré-processamento no conjunto de documentos, onde elimina-se palavras com valor discriminante baixo. Eventualmente, pode ser realizada a *radicalização* (*stemming*) das palavras, processo que remove sufixos e flexões para obter o radical da palavra, reduzindo assim o número de palavras do conjunto. A literatura consultada raramente utilizou a radicalização para testes efetuados, os quais normalmente usam textos em língua inglesa. Esta dissertação utiliza textos em português e aplica a radicalização em todos os testes realizados. O algoritmo utilizado será devidamente apresentado na Seção 6.3;

2. para cada palavra é gerado um vetor $\mathbf{v}_{s(k)}$ que representa a palavra k. Originalmente, os vetores $\mathbf{v}_s$ devem ser ortogonais entre si (evitando correlação espúria entre termos). Esta dissertação utilizou a *Projeção Randômica*, descrita na Seção 6.2.4.2 a seguir;

3. treina-se um SOM semântico com os vetores de contexto médio obtidos pela codificação de todos os símbolos relevantes do corpo de texto, conforme Equação 6-8. (embora seja possível considerar-se uma "janela de contexto" maior, a dimensão do vetor de contexto médio final torna-se rapidamente proibitiva);

4. apresenta-se o texto de cada documento, palavra por palavra, dentro da janela de contexto considerada, ao SOM semântico já treinado. Deste, obtém-se um *histograma* do documento em relação à freqüência com que os neurônios são excitados, considerando a resposta de cada BMU como uma dimensão do vetor. Este vetor é uma espécie de "assinatura estatística" do documento e as respostas dos neurônios BMU são acumuladas à medida que o texto é apresentado ao SOM semântico;

5. os histogramas de documentos, gerados a partir do SOM semântico, são usados para treinar o SOM de documentos, que agora pode representar as relações entre os documentos do conjunto.

Este processo é bastante interessante pois, havendo um SOM semântico devidamente treinado e geral o suficiente para acomodar uma grande variedade de palavras, é possível utilizar o SOM de documentos para representar a semelhança de textos inexistentes no momento do treinamento. O processamento dos histogramas de documentos (a partir do SOM semântico) é bastante rápido e pode ser efetuado em tempo real, permitindo que um usuário visualize a relação de semelhança entre novos documentos e aqueles utilizados para o treinamento no SOM de documentos.

Uma das maiores dificuldades deste modelo é a gerência do tamanho dos mapas. No SOM semântico, cada neurônio torna-se sensível a algumas palavras que contribuirão igualmente para a geração dos histogramas de documentos. Portanto, cada neurônio deve representar uma quantidade de palavras que incluam seus sinônimos. Uma carga baixa de palavras por neurônio não é prejudicial, uma vez que invariavelmente elas serão representadas. Um número grande de palavras acumuladas em um neurônio, entretanto, tende a generalizar demais o mapeamento e necessariamente levará a uma diminuição de acuidade (Honkela,

1997b). O mesmo pode ser considerado em relação ao SOM de documentos. Segundo Kohonen (1998) cada neurônio deve responder por no máximo 10 palavras sob pena de perda de acuidade do SOM semântico. O mesmo autor sugere diversas técnicas para acelerar a computação de mapas SOM grandes, como o uso de ponteiros para identificação do neurônio BMU e técnicas de espalhamento (*hashing*) para indexação rápida de palavras em códigos.

Outra dificuldade relaciona-se ao tamanho dos vetores de contexto médio: se a exigência de códigos ortogonais for cumprida à risca, a dimensão dos vetores torna-se computacionalmente proibitiva, exigindo algum processo de redução dimensional dos mesmos.

Referências importantes ou relacionadas a esta técnica incluem Scholtes (1993), Honkela *et al.* (1995; 1996b; 1997), Honkela (1997a,b,c), Lin *et al.* (1999), Kohonen (1998), Kohonen *et al.* (1996; 2000), Kaski *et al.* (1996; 1998a), Lagus *et al.* (1996a,b), Lagus (1997; 1998; 1999; 2000) e Visa *et al.* (2000).

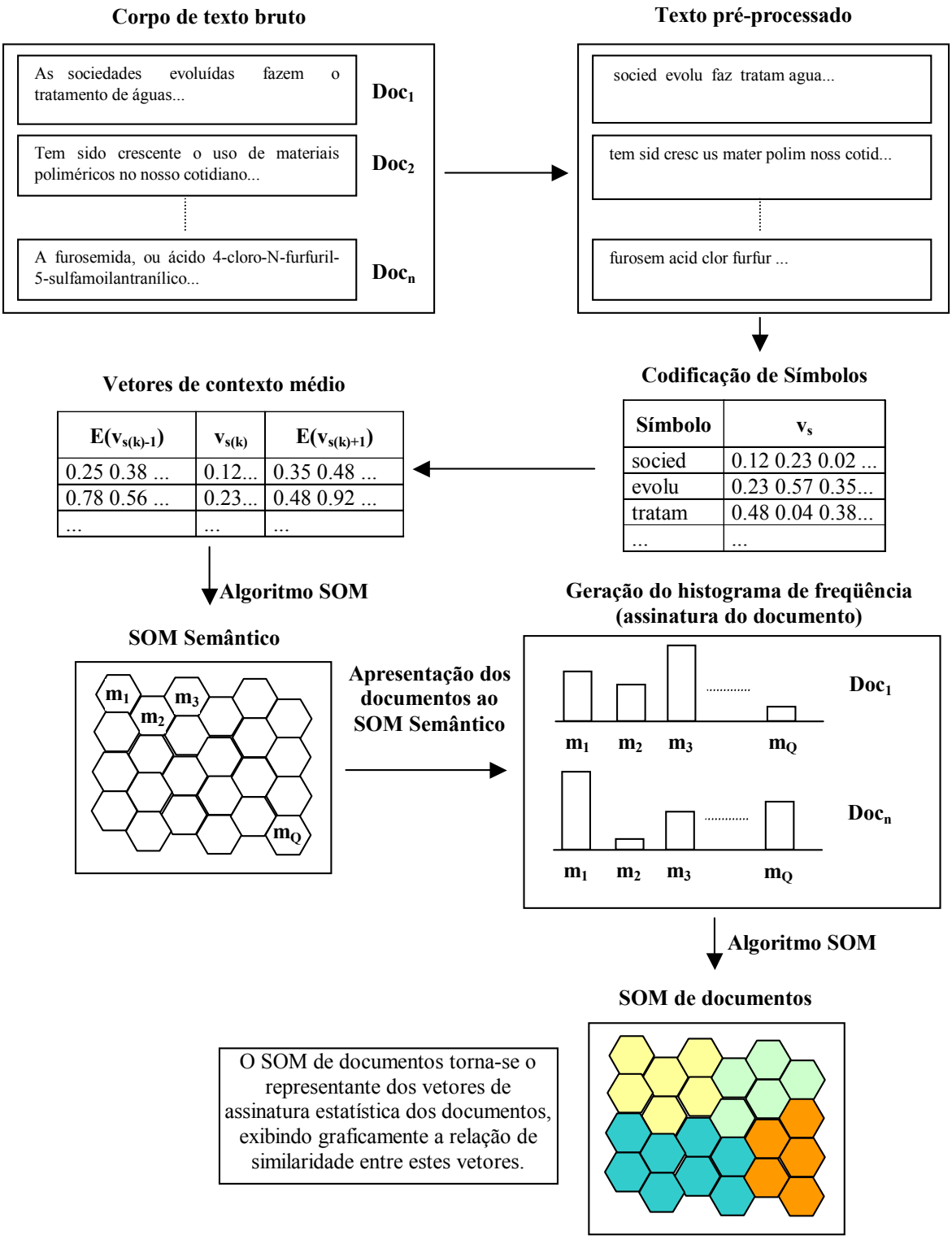


Figura 6-3 – O processo de criação do SOM de documentos. O corpo de texto bruto é pré-processado e os símbolos são codificados. Os vetores de contexto médio de cada símbolo são calculados e usados no treinamento do SOM semântico. Este SOM receberá o texto de cada documento, palavra por palavra, e um histograma dos neurônios excitados é gerado, construindo o histograma que representa a "assinatura estatística" do documento. Estes vetores são usados no treinamento do SOM de documentos.

6.2.4.2 Projeção randômica

Os modelos de espaço vetorial, indexação semântica latente e SOM semântico são computacionalmente caros principalmente devido à dimensão dos vetores de características que representam o contexto dos documentos (Kohonen *et al.* 2000; Lagus, 2000). Uma alternativa econômica para reduzir a dimensão destes vetores é a chamada *Projeção Randômica* (Ritter & Kohonen, 1989; Kaski, 1998).

A redução dimensional por projeção linear de um espaço $\Re^D$ para um espaço de projeção $\Re^Q$, $Q \leq D$, pode ser vetorialmente representada por

$$\mathbf{x}_i = \mathbf{A}\mathbf{v}_i \ , i = 1, ..., N. \qquad \textbf{Equação 6-9}$$

onde **A** é uma matriz $Q \times D$ de vetores $\mathbf{a}_j$, cada vetor representando a j-ésima coluna de **A**. Os vetores $\mathbf{x} = [x_1, ..., x_Q]^T \in \Re^Q$ são gerados como uma combinação linear dos objetos $\mathbf{v} = [v_1, ... , v_D]^T \in \Re^D$. Devido ao pressuposto de codificação ortogonal dos símbolos (no SOM semântico) ou simplesmente devido ao volume de palavras (todos os modelos baseados em vetores), os métodos existentes para efetuar estas reduções são computacionalmente caros (por exemplo, PCA, veja a Seção 2.4). A proposta de simplificação é substituir a matriz $\mathbf{A}_{Q\times D}$ (ortogonal) por uma matriz randômica $\mathbf{R}_{P\times D}$ "quase" ortogonal com $P << Q$ e que leva à projeção num espaço $\Re^P$, conforme a equação abaixo:

$$\mathbf{y}_i = \mathbf{R}\mathbf{v}_i \ , i = 1, ..., N. \qquad \textbf{Equação 6-10}$$

onde **R** é uma matriz $P \times D$ de vetores $\mathbf{r}_j$, cada vetor representando a j-ésima coluna de **R**, e $\mathbf{y} = [y_1, ..., y_P]^T \in \Re^P$. É claro que quanto mais os vetores $\mathbf{r}_j$ na Equação 6-10 se aproximarem da ortogonalidade entre si, tanto melhor representarão as dissimilaridades entre os vetores originais $\mathbf{v}_i$, sem inserir distorções.

Se P for grande o suficiente (>100), a aproximação que **R** faz de uma base ortogonal é suficientemente boa para realizar a redução dimensional proposta na Equação 6-10 mantendo praticamente todas as características dos vetores originais $\mathbf{v}_i$ (Kohonen, 1998). A motivação deste raciocínio é a de que, em espaços de grande dimensão, há um número

infinitamente maior de bases quase ortogonais do que bases realmente ortogonais. Assim, em espaços de grande dimensão, mesmo vetores com direções aleatórias podem ser próximos o suficiente de serem ortogonais para permitir sua utilização. Haverá sempre pequenas distorções introduzidas na projeção realizada, mas estas são em média iguais a zero e tendem a diminuir conforme aumenta o valor da dimensão *P* utilizada (Kaski, 1998).

A Figura 6-4 apresenta uma ilustração da quase-ortogonalidade de vetores aleatórios para vários valores de *P* observando a distribuição do valor do produto interno entre pares aleatórios de vetores.

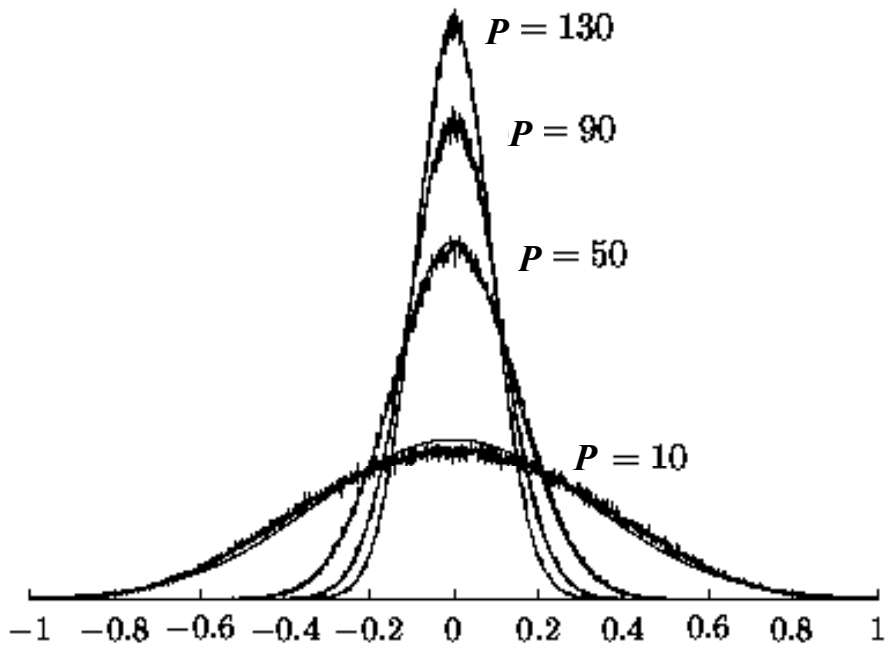


Figura 6-4 – Distribuição do produto interno entre pares aleatórios de vetores $\mathbf{r}_i$ para vários valores de *P*. O produto interno não será zero (ortogonal), mas em geral um valor pequeno do produto interno introduzirá pequenas distorções nas projeções. Figura adaptada de Kaski (1998).

A grande vantagem desta abordagem é que espaços de alta dimensionalidade podem ser projetados, sem perdas expressivas de representação, para espaços da ordem de 100 dimensões, computacionalmente muito mais baratos. Considerando que cada vetor $\mathbf{v}_i \in \Re^D$ representará um símbolo semântico e que estes, em princípio, devem ser ortogonais entre si, então uma boa escolha para a matriz $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_N\}$ é a matriz identidade **I**, com dimensão $D = N$. Esta escolha permite uma simplificação da Equação 6-10, pois a projeção do conjunto de vetores $\mathbf{v}_i$, através da matriz **R** com dimensão $P \times N$, é a própria matriz **R**. Isto

significa que, para projetar um conjunto **V** de vetores $\mathbf{v}_i$ no espaço de entrada $\Re^D$, para um espaço $\Re^P$, basta gerar uma matriz **R** de tamanho $P \times N$ com valores randômicos. Cada coluna $\mathbf{r}_j$ da matriz **R**, j=1, ..., N, representará um vetor $\mathbf{y} = [y_1, ..., y_P]^T$ no espaço $\Re^P$.

Mais recentemente, Kohonen (1998) vem abandonando a construção do SOM semântico (e do respectivo histograma de palavras a partir deste) em prol da projeção randômica direta do modelo de espaço vetorial. Os vetores resultantes desta projeção randômica são, então, utilizados para treinar o SOM de documentos.

6.2.5 Outros modelos e variações

Os modelos variantes para recuperação de informação buscam incluir mais informação semântica e de contexto na codificação dos documentos. Isto implica na tentativa de criar modelos que possam incorporar ao máximo informações da área de processamento de linguagem natural.

Scholtes (1991b; 1993) apresenta propostas baseadas em redes neurais artificiais que buscam integrar os informações léxicas, sintáticas e semânticas para aumentar a performance de sistemas de recuperação de informação.

Miikkulainen (1990; 1997) sugere o uso de processamento sub-simbólico da linguagem natural através de SOMs hierárquicos com alguns traços de recorrência. A abordagem parte do princípio que o conhecimento é, de alguma forma, armazenado sob a forma de *roteiros* (*scripts*). Roteiros são, assim, esquemas estereotípicos de seqüências de eventos. Seres humanos possuem, segundo o autor, centenas ou milhares de roteiros.

Boley *et al.* (1999) aplicam diversos algoritmos de agrupamento por particionamento em documentos obtidos diretamente da Internet. Os métodos testados e propostos não utilizam técnicas de redes neurais artificiais.

Finalmente, Visa *et al.* (2000) propõem uma hierarquia multinível de SOMs para tentar codificar a informação de contexto através de mapas de palavras (SOM semântico), mapas de sentenças e mapas de parágrafos.

6.3 Uso de SOM e GTM em Recuperação de Informação

A literatura sobre recuperação de informação baseada no SOM, com raras exceções, realiza experimentos com conjuntos de textos em língua inglesa. Também é comum o uso de conjunto artificiais (como em Ritter & Kohonen (1989)) e raramente recorre-se a algum tipo de radicalização com intuito de reduzir a quantidade de símbolos a representar.

A utilização do SOM nesta dissertação procurou evidenciar que o modelo proposto na Seção 6.2.4 também é funcional para textos em língua portuguesa. Porém, várias características da língua portuguesa a tornam um experimento bastante distinto daqueles encontrados na literatura, destacando-se:

- elevado número de vocábulos existentes
- elevado número de sinônimos entre vocábulos
- elevado número de flexões verbais (comum em línguas latinas)
- diversas possibilidades de construção sintática para a expressão de idéias
- elevado número de partículas textuais com flexões em gênero, número e grau (como artigos, preposições e advérbios)
- e, finalmente, grande número de exceções a praticamente todas as regras.

Os experimentos realizados demonstram que a língua portuguesa demanda um tratamento mais cuidadoso para que sejam obtidos resultados aproveitáveis. Especialmente, optou-se pela radicalização das palavras dada a grande variedade de flexões de vocábulos da língua portuguesa.

A utilização do GTM em recuperação de informação busca avaliar o comportamento desta ferramenta quando utilizada para a geração de mapas que possibilitem a observação das relações de contexto entre documentos. Até onde foi pesquisado, nenhuma referência a esta aplicação do GTM foi encontrada na literatura. A abordagem tomada nesta dissertação é uma forma híbrida, onde os vetores representando a assinatura do documento (gerados a partir de um SOM semântico) foram apresentados ao algoritmo GTM.

Foram utilizados conjuntos de texto buscando responder duas perguntas:

a) se o conjunto de documentos possui poucos temas bastante distintos (isto é, com poucas palavras representativas comuns entre si), mesmo um número pequeno de textos (um conjunto estatisticamente pequeno) pode ser classificado conforme seus assuntos?
b) se o conjunto de documentos possui muitos temas sem uma distinção expressiva em termos de contexto, há formas para melhor evidenciar a separação dos conjuntos?

O primeiro conjunto "Esporte e Culinária" (EC) possui um total de 52 textos, sendo 25 sobre esporte de competição de carros (*Stock Car*) e 27 tratando de receitas culinárias, num total de 12187 palavras (média de 230 palavras por texto). O conjunto EC possui uma separação de contexto bastante clara (considerando análise manual) e foi utilizado para testar a primeira hipótese, o que pode ser visto na Seção 6.3.1.

O segundo conjunto, "Anais da Universidade São Francisco" (AnUSF), possui um total de 161 documentos constituídos pelos resumos (*abstracts*) de publicações científicas ocorridas no II Congresso de Pesquisa e Extensão da Universidade São Francisco, realizado no campus de Bragança Paulista de 6 a 8 de outubro de 1999. Este conjunto está assim dividido:

- 55 textos na área de Ciências Exatas e Tecnológicas (ET)
- 37 textos na área de Ciências Biológicas de Saúde (BS)
- 69 textos na área de Ciências Humanas e Sociais (HS)

O conjunto AnUSF possui um total de 34726 palavras (média de 203 palavras por texto) e a separação de contexto foi considerada complexa após análise manual. Este conjunto foi utilizado para experimentar a segunda hipótese e o experimento pode ser visto na Seção 6.3.2.

6.3.1 Experimento – Conjunto EC

O conjunto EC foi manipulado basicamente segundo o procedimento da Seção 6.2.4.1, ilustrado na Figura 6-3. O conjunto inicial continha 52 textos e 12187 palavras, transformadas as letras em minúsculas e sinais de pontuação e algarismos tendo sido removidos. Após isso, foram removidas também 5286 palavras de uso comum e que não

agregam informação ao contexto. Estas palavras são artigos (“a, as, os, algum, ...”), preposições (“ante, até, após, ...”), conjunções (“e, ou, porque, quando, onde, ...”) e alguns verbos, incluindo suas flexões (“ser, estar, ter, fazer”). Estes verbos foram escolhidos previamente, antes de qualquer manipulação do conjunto de documentos. Das 6901 palavras restantes, uma análise mostrou haver um total de 2108 palavras diferentes, incluindo as flexões de gênero, número e grau ainda presentes.

O conjunto foi então radicalizado (isto é, eliminou-se sufixos e flexões), sendo obtidas 1316 palavras, ou seja, uma redução de aproximadamente 37,6% na quantidade de símbolos. O algoritmo de radicalização utilizado foi adaptado de uma versão de Chung (2003) em linguagem PERL, o qual baseia-se no algoritmo proposto por Orengo & Huyck (2001). O problema de radicalização de palavras na língua inglesa parece ter sido resolvido pelo “algoritmo de Porter” (Porter, 1980). Infelizmente, o mesmo não ocorre para a língua portuguesa, que se mostra bem mais complexa e repleta de exceções. O algoritmo utilizado, entretanto, é superior à versão para português do algoritmo de Porter (Orengo & Huyck, 2001).

O algoritmo opera em 8 estágios (conforme Figura 6-5), buscando reduzir, por ordem: (1) a forma plural, (2) transformar a forma feminina para a forma masculina, (3) redução de advérbios pela exclusão do sufixo “-mente”, (4) redução de grau (diminutivo, aumentativo e superlativo), (5) redução do sufixo de substantivos (por exemplo, “contagem $\rightarrow$ cont”), (6) redução do sufixo de verbos e flexões para sua raiz, (7) redução da vogal final de palavras como “menino $\rightarrow$ menin” e, finalmente, (8) remoção de acentos. Um exemplo da radicalização efetuada no conjunto EC pode ser vista na Tabela 6-3.

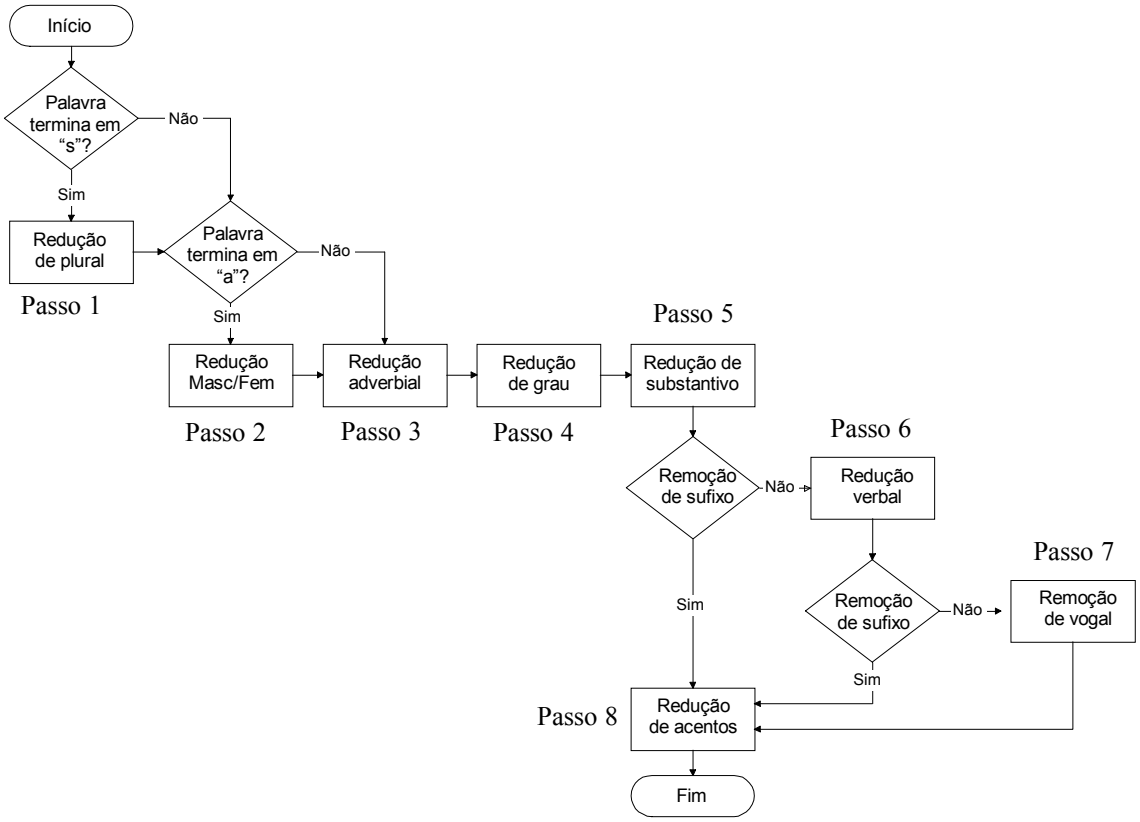


Figura 6-5 – Algoritmo de radicalização em 8 passos para língua portuguesa. A radicalização é um processo que busca reduzir as várias flexões de uma palavra para uma forma única. Adaptado de Orengo & Huyck (2001).

Embora com resultados positivos na redução do número de palavras do corpo de texto, o algoritmo apresenta incorreções:

- as formas "alta", "alto" e "alterado" foram reduzidas para a forma única "alt".
- "alimentos" e "alimentícios" são palavras semanticamente próximas, mas foram reduzidas para as formas "aliment" e "alimentici".

O primeiro erro é chamado *sobre-radicalização* (*overstemming*) e prejudica o índice de precisão na recuperação (Equação 6-1), pois coloca sob o mesmo símbolo palavras semanticamente distintas. Já o segundo erro, a *sub-radicalização* (*understemming*), prejudica o índice de recuperação de documentos (Equação 6-2) por não entender como semanticamente relacionadas palavras que o são.

Tabela 6-3 – Alguns exemplos de palavras radicalizadas do conjunto EC segundo o algoritmo de Orengo & Huyck (2001). Também pode ser observada a freqüência total da ocorrência do radical no conjunto de texto.

Palavra	Radical	Freqüência
abacaxi	abacax	4
abandonar	abandon	3
abandonou	abandon	
abertura	abert	5
abortar	abort	1
abra	abr	2
abril	abril	4
acelera	acel	1
acertando	acert	15
acertar	acert	
acerte	acert	
acerto	acert	
acertos	acert	
colher	colh	92

A partir do conjunto de 1316 palavras ($N = 1316$), foi usada a técnica de projeção randômica para associar um vetor de código para cada palavra, conforme proposto em 6.2.4.2. A dimensão escolhida foi $P = 100$, o que gerou uma matriz de dimensão $P \times N$, composta por 1316 vetores com dimensão 100, cada vetor representando uma palavra do conjunto. O conjunto de texto foi então processado para obter o contexto médio de cada palavra considerando a palavra antecedente e subseqüente de cada símbolo, conforme a Equação 6-8. Este processo gerou uma nova matriz composta por 1316 vetores com dimensão 300, representando o contexto médio de cada símbolo (palavra) no conjunto de documentos.

Esta matriz foi utilizada para treinar o SOM semântico (Figura 6-6). Optou-se por um mapa de 20 × 20 neurônios em um arranjo hexagonal com função de vizinhança gaussiana, o que corresponde a uma média de 3,29 símbolos por neurônio, bem abaixo do máximo sugerido de 10 símbolos por neurônio (Kohonen, 1998). O mapa de resposta neural (número de palavras que cada neurônio representa, isto é, para o qual é o BMU) é apresentado na Figura 6-7, onde verifica-se que a distribuição, embora não ideal, parece não ter afetado os resultados obtidos. O treinamento foi efetuado em duas fases (inicial com 5 épocas e raio de

vizinhança decrescente de 10 a 2 e convergência com 30 épocas e raio fixo em 1), apresentando erro médio de quantização (QE) de 15,866919 e erro topográfico (TE) de 0,013678. Os parâmetros foram escolhidos com base em análise visual da distribuição de palavras e dos menores índices QE e TE de um total de 10 experimentos, onde foram variados os tamanhos dos mapas e parâmetros de treinamento (número de épocas e raios de vizinhança).

Foi observado que o tamanho do mapa influiu apenas na resolução (isto é, quantidade de símbolos representados por cada neurônio). A aparência da matriz-U para os testes variou pouco, exceto por rotações comuns no algoritmo SOM (Kohonen, 1997).

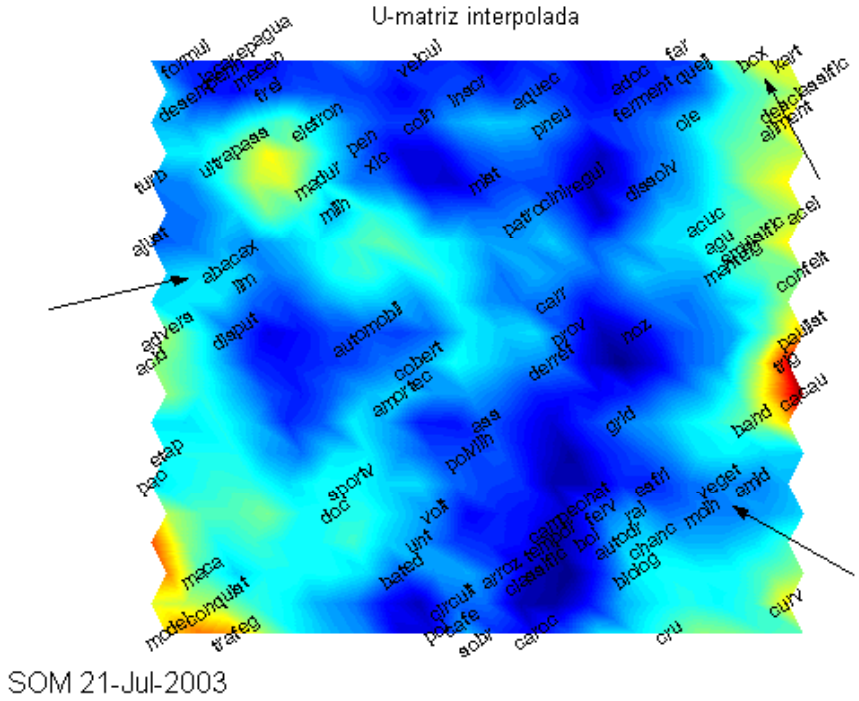


Figura 6-6 – Matriz-U do SOM semântico de 20 × 20 neurônios com a sobreposição de algumas palavras escolhidas manualmente do corpo de texto. As setas apontam algumas palavras que foram mapeadas proximamente entre si, indicando que o mapa agrupou as palavras semanticamente próximas baseadas em seu contexto.

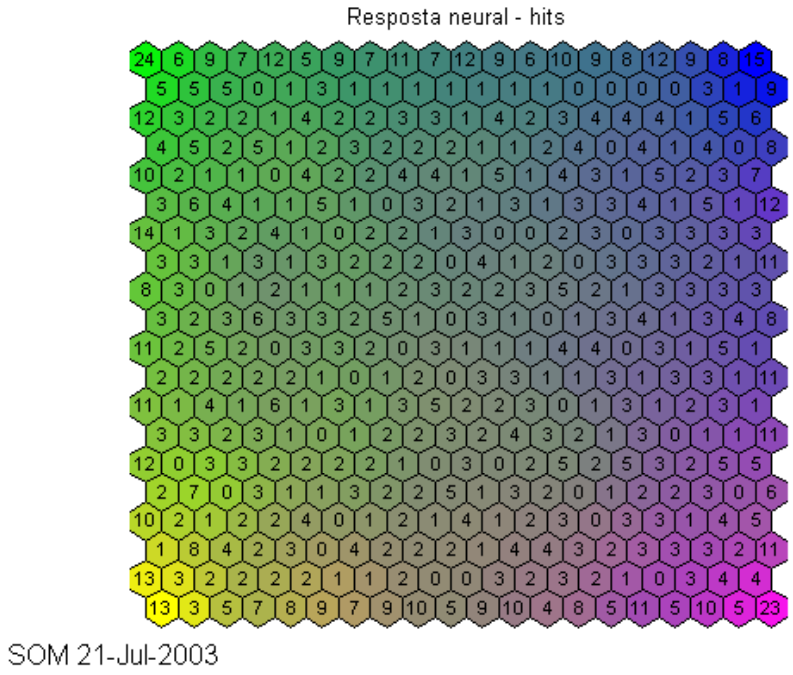
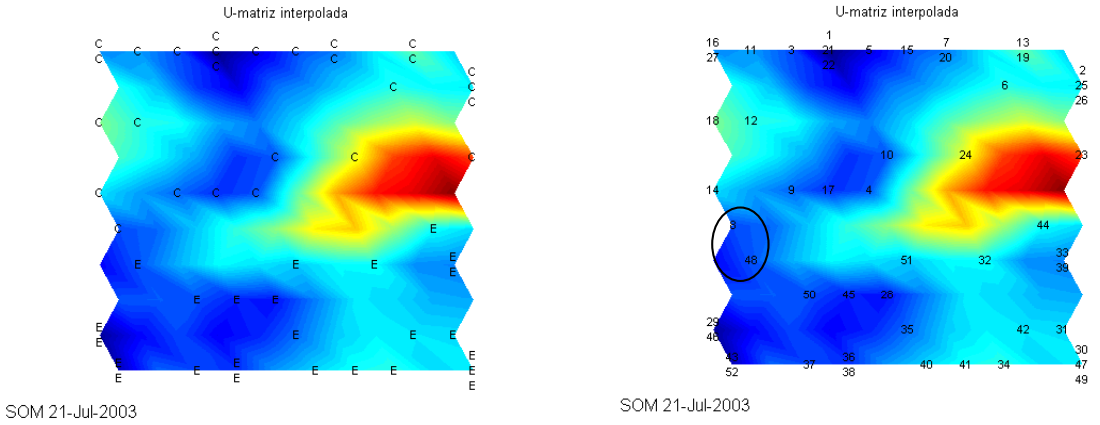


Figura 6-7 – Resposta neural do SOM semântico para um total de 1316 palavras radicalizadas do conjunto EC.

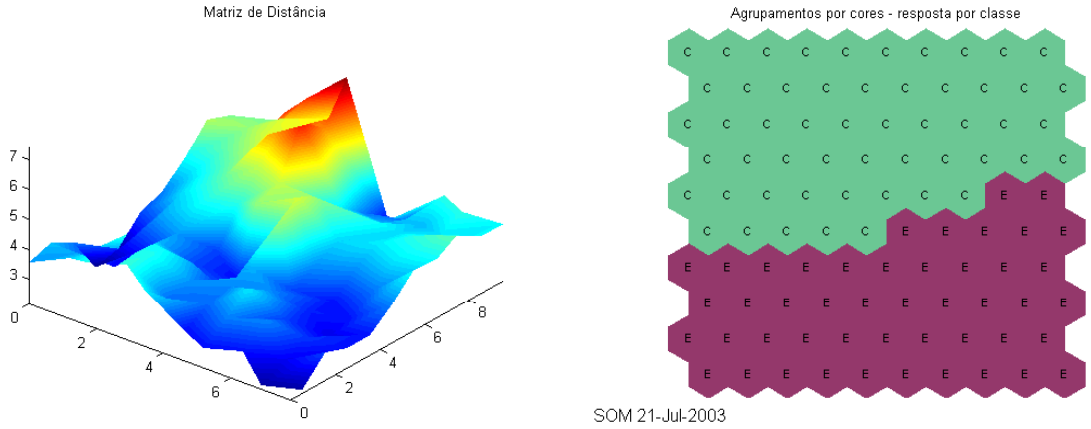
Após a geração do SOM semântico, o conjunto de texto foi apresentado ao mapa semântico e os histogramas de palavras de cada documento foram obtidos, gerando uma matriz composta por 52 vetores de dimensão 400, representando os mesmos. Estes vetores foram utilizados para treinar um SOM de documentos de 10×10 neurônios com índices QE = 8,759696 e TE = 0,0 em treinamento de duas fases (inicial com 7 épocas e raio de vizinhança decrescente de 7 a 1 e convergência com 20 épocas e raio fixo em 1). O erro topográfico igual a zero justifica-se porque os vetores de entrada de dados (os histogramas) são discretos. Este teste não utilizou nenhuma forma de ponderação relativa à freqüência de ocorrência das palavras. A Figura 6-8 apresenta a matriz-U do SOM com os documentos com os rótulos apresentados.



(A) **(B)**

Figura 6-8 – SOM de documentos com rótulos de documentos. Os rótulos foram previamente escolhidos com "E" e "C" para representar textos relativos ao esporte ou culinária, respectivamente. Em (A) vê-se os rótulos e em (B) os números identificando cada documento, praticamente separados em dois conjuntos ocupando as metades superior e inferior dos mapas. Os dois itens destacados em (B) sugerem uma proximidade contextual inexistente de fato.

É interessante notar que a matriz-U praticamente separou os textos em dois conjuntos. Entretanto, esta percepção não é óbvia a menos que se recorra aos rótulos pré-definidos. A Figura 6-9 apresenta a matriz-U e uma classificação por cores tomando cada neurônio e observando a que classe corresponde o vetor de documento mais próximo por ele representado. O resultado torna-se evidente, mas não é razoável admitir a existência de duas classes com base nestas informações apenas. Embora seja inegável que os contextos foram separados eficientemente, a necessidade do mapa em representar os itens pode sugerir proximidade de contexto onde ela, de fato, não existe. Este fato pode ser observado exatamente na fronteira que separa os textos na Figura 6-8-B e sugere precaução na interpretação dos resultados do SOM de documentos.



(A) **(B)**

Figura 6-9 – Representação da superfície da matriz-U do SOM de documentos. Embora a resposta neural por classe confirme a separação dos contextos, a matriz-U não é capaz, sozinha, de sugerir esta separação com clareza.

A mesma matriz composta por 52 vetores de dimensão 400, vetores estes gerados a partir do SOM semântico, foi utilizada para adaptar um modelo GTM de 20 × 20 pontos latentes e 12 × 12 funções-base com espalhamento 0,8 em 8 ciclos. O modelo escolhido foi aquele com maior logaritmo da verossimilhança dentre 10 testes onde foram variados diversos parâmetros. O valor inicial deste índice foi –46219,9 e o valor final 25840,9. O modelo GTM possui uma convergência muito rápida e a separação entre os objetos pode ser percebida claramente mesmo antes do modelo ser adaptado, como pode ser verificado na Figura 6-10. Por outro lado, assim como no SOM de documentos, não é óbvia a separação dos documentos em dois conjuntos, embora seja inegável que houve separação dos contextos.

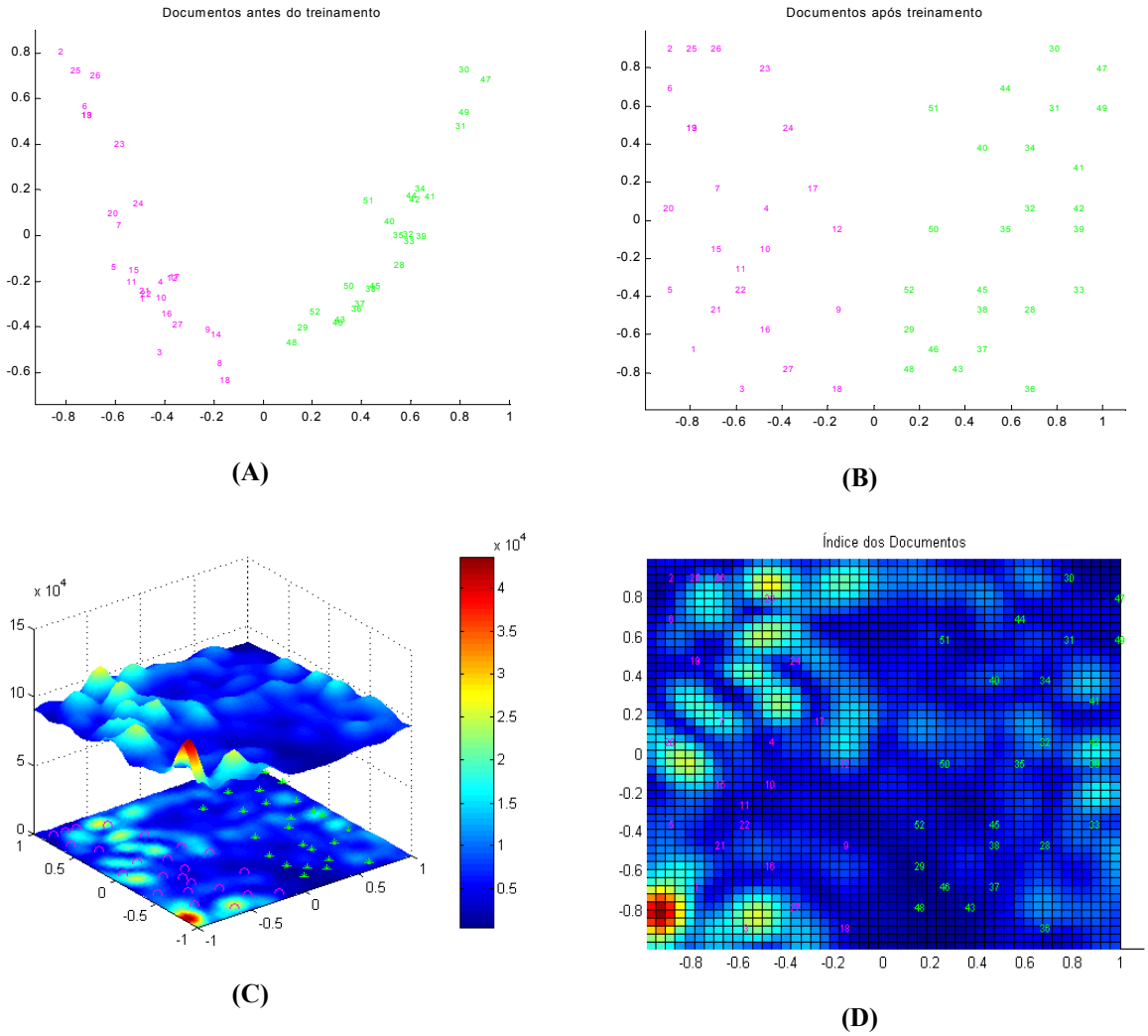


Figura 6-10 – Projeção da média *a posteriori* da distribuição dos dados (documentos) sobre o espaço latente antes (A) e depois (B) do treinamento do modelo GTM. Em (C) e (D) os fatores de ampliação, sobre os quais foi projetada a média *a posteriori* dos documentos. A esquerda de todas as figuras encontram-se os documentos de culinária ("C", em vermelho) e à direita, os de esportes ("E", em verde).

Percebeu-se que, em ambas as ferramentas, a variação dos parâmetros gerou resultados coerentes e bastante semelhantes aos relatados nesta dissertação. Isto deixa claro que a primeira hipótese parece ser verdadeira: mesmo um conjunto estatisticamente pequeno de textos, mas com temas bastante distintos, pode ser separado, em termos de contexto, sem grandes dificuldades. Se a informação prévia das classes for disponível, ambas as ferramentas são bastante efetivas na classificação de novos textos. Se, porém, as ferramentas operarem em modo totalmente não supervisionado, não é imediata e nem trivial a separação dos possíveis temas.

Um novo experimento foi realizado, onde se procurou adicionar mais sensibilidade ao contexto dos documentos. Foi gerado um novo conjunto de vetores de documentos apresentando os textos ao SOM semântico já treinado. Desta vez, entretanto, foram considerados não só o neurônio BMU mas também o 2º BMU que respondeu ao estímulo de cada palavra (com o respectivo contexto). Considerando o histograma onde cada posição corresponde a um neurônio do SOM semântico, foi adicionado 1 à posição correspondente ao BMU e 0,25 à posição correspondente ao 2º BMU.

As duas ferramentas foram aplicadas com os mesmos parâmetros dos testes anteriores. Os resultados constam da Figura 6-11. No caso do SOM, observou-se uma melhoria efetiva na capacidade da matriz-U em sugerir a existência de dois agrupamentos. No caso do GTM, não foi percebida uma mudança significativa quanto à separação de agrupamentos

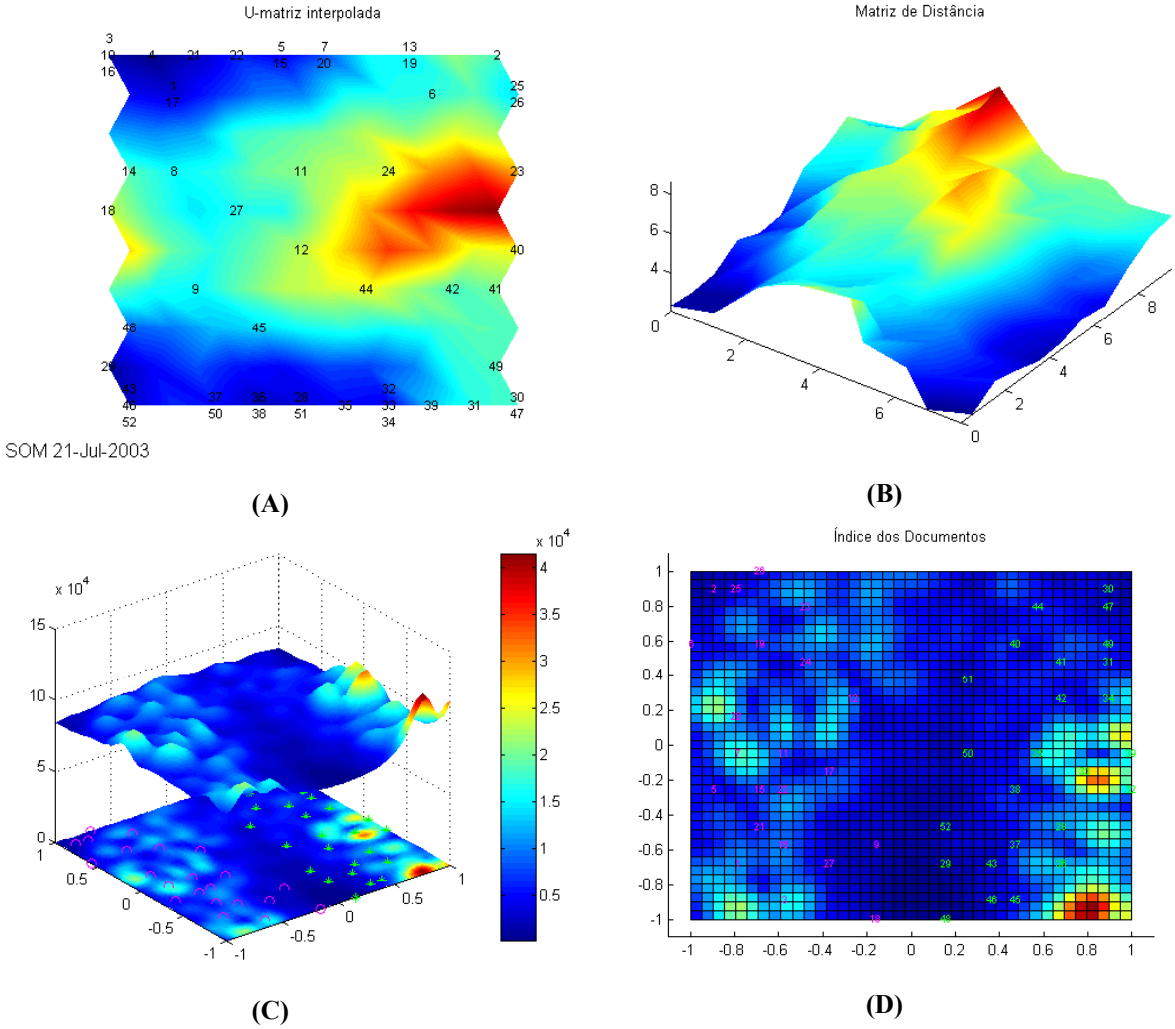


Figura 6-11 – Experimento onde foram considerados o 1º e 2º BMUs do SOM semântico na geração dos vetores representantes dos documentos. Em (A) e (B) pode-se perceber que a possibilidade de 2 agrupamentos é mais evidente no SOM. Em (C) e (D), não foi observada melhora significativa na evidência de agrupamentos no GTM.

6.3.2 Experimento – Conjunto AnUSF

O conjunto AnUSF possui um total de 161 documentos constituídos pelos resumos (*abstracts*) de publicações científicas ocorridas no II Congresso de Pesquisa e Extensão da Universidade São Francisco, realizado no campus de Bragança Paulista de 6 a 8 de outubro de 1999. Este conjunto está assim dividido:

- 55 textos na área de Ciências Exatas e Tecnológicas (ET), rotulados de 1 a 55;
- 37 textos na área de Ciências Biológicas de Saúde (BS), rotulados de 56 a 92 e
- 69 textos na área de Ciências Humanas e Sociais (HS), rotulados de 93 a 161.

O conjunto foi manipulado segundo o procedimento da Seção 6.2.4.1, ilustrado na Figura 6-3.

Os 161 documentos totalizam 34726 palavras, já com letras minúsculas e sinais de pontuação e algarismos removidos. Foram removidas 14490 palavras de uso comum (artigos, preposições e conjunções), além de alguns verbos com suas flexões ("ser, estar, ter, fazer"). Restaram 20236 palavras das quais uma análise mostrou haver 6559 palavras distintas (considerando as flexões de gênero, número e grau). O conjunto foi então radicalizado conforme algoritmo de Orengo & Huyck (2001), o que resultou num total 3530 palavras radicalizadas, uma redução de aproximadamente 46,2% na quantidade de símbolos.

A partir do conjunto de 3530 palavras, foi usada a técnica de projeção randômica para associar um vetor de código a cada palavra, conforme a Seção 6.2.4.2, sendo escolhido $P = 100$. A matriz obtida é composta por 3530 vetores de dimensão 100, cada vetor representando uma palavra. Esta matriz foi utilizada, junto com o texto de cada documento, para gerar o contexto médio (Equação 6-8) de cada palavra, resultando numa matriz composta por 3530 vetores de dimensão 300 representando o contexto médio de cada palavra. Partindo do mesmo conjunto de dados para treinamento (os vetores de contexto médio), foram realizados 20 experimentos variando-se parâmetros de treinamento como tamanho do mapa, raio de vizinhança e número de épocas de treinamento. Nos 10 primeiros testes, nenhuma normalização foi realizada nos vetores de dados e os 10 testes seguintes repetiram os parâmetros, porém agora usando a normalização da variância de cada componente dos vetores.

Observou-se que, com a normalização, o erro topográfico (TE) de todos os experimentos foi invariavelmente menor que os testes sem a normalização, embora tenha ocorrido um aumento no erro médio de quantização (QE). Foi escolhida, dentre os testes, a configuração com menor TE (com uso de normalização, portanto). Uma adaptação topologicamente correta e mais homogênea é importante neste experimento: uma concentração de termos sendo representados por poucos neurônios é nociva à construção do histograma de palavras, uma vez que tais palavras são consideradas semanticamente próximas por serem representadas pelos mesmos neurônios. A escolha foi um mapa de 30×30 neurônios em arranjo hexagonal e função de vizinhança gaussiana com treinamento efetuado em duas

fases (inicial com 5 épocas e raio de vizinhança decrescente de 15 a 3 e convergência com 30 épocas e raio de vizinhança decrescente de 2 a 1), obtendo QE = 15,902461 e TE = 0,011048. A média estimada é de 3,92 palavras por neurônio.

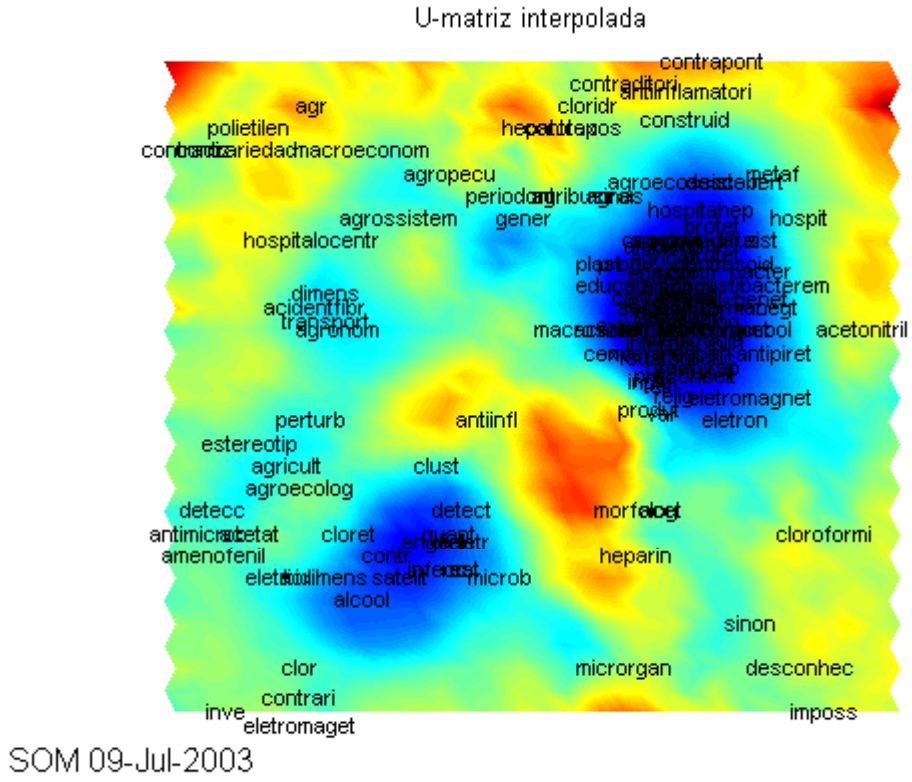


Figura 6-12 – SOM semântico não normalizado, com QE = 9,430602 e TE = 0,036425. Algumas palavras escolhidas manualmente foram plotadas sobre a matriz-U e nota-se uma excessiva concentração em dois agrupamentos de palavras.

A Figura 6-12 apresenta um exemplo do SOM semântico gerado a partir de dados não normalizados, onde nota-se uma representação excessivamente concentrada de termos em poucas áreas do mapa, o que prejudica a representação. A Figura 6-13 apresenta o mapa gerado a partir dos mesmos parâmetros de treinamento utilizando, entretanto, os dados normalizados em sua variância.

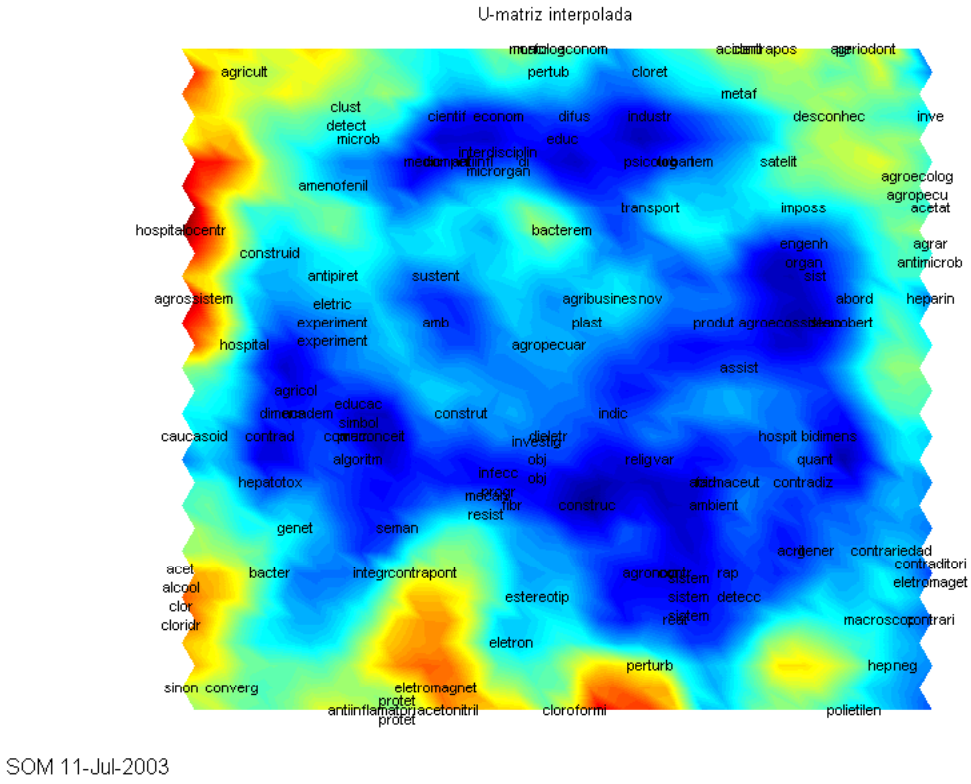


Figura 6-13 – SOM semântico com os vetores de entrada normalizados (QE = 15,902461 e TE = 0,011048). O mesmo conjunto de palavras da Figura 6-12 foi plotado aqui sobre a matriz-U, que revela uma resposta neural melhor distribuída.

Após a escolha do SOM semântico (Figura 6-13), cada documento foi apresentado ao mesmo, gerando-se então uma matriz composta por 161 vetores de dimensão 900, cada vetor representando a assinatura estatística de cada documento. No primeiro experimento, somente os 1os BMUs foram considerados na construção dos histogramas (conforme a Seção 6.2.4.1), os quais foram então usados para o treinamento do SOM de documentos de 12×15 neurônios em arranjo hexagonal com índices QE = 11,156349 e TE = 0,0 em treinamento de duas fases (inicial com 8 épocas e raio de vizinhança decrescente de 6 a 2 e convergência com 30 épocas com raio fixo em 1). Esta configuração foi escolhida por ter o menor QE de um conjunto de 10 experimentos, onde foram variados parâmetros de treinamento. Novamente, o erro topográfico igual a zero justifica-se porque os vetores de entrada (os histogramas) contêm valores discretos.

Para avaliar a qualidade do SOM de documentos gerado, foi feita uma análise manual do conjunto de textos e foram escolhidos dois grupos de documentos com contextos próximos:

- Grupo A: documentos identificados pelos números 3, 35, 36 e 37 (versando sobre pesquisas em crescimento de diamante)
- Grupo B: documentos 25, 29, 30 e 31 (versando sobre habitações populares e técnicas de construção com materiais alternativos).

A Figura 6-14 apresenta o SOM de documentos adaptado pelo procedimento descrito acima.

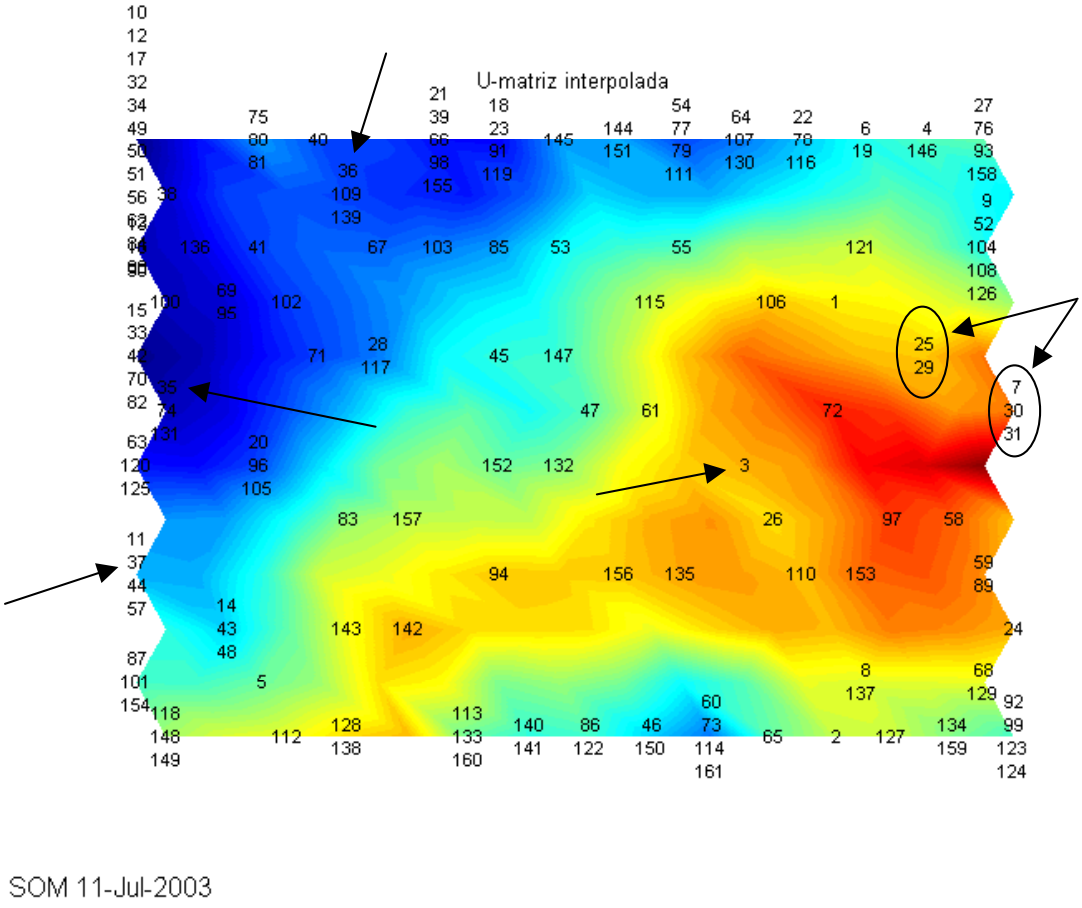


Figura 6-14 – SOM de documentos de 12 × 15 neurônios adaptado a partir dos histogramas dos documentos considerando apenas o 1º BMU. Os conjuntos A (documentos 3, 35, 36 e 37) e B (documentos 25, 29, 30 e 31) são indicados por setas. Segundo a interpretação do mapa, o grupo A não possui seus documentos próximos em similaridade contextual.

Um novo experimento buscou testar a hipótese de inserir mais informação de contexto no SOM de documentos considerando também a contribuição do 2º BMU na geração dos histogramas de contexto. Um novo conjunto de histogramas foi gerado e usado para adaptar SOMs de documentos seguindo os mesmos parâmetros dos testes anteriores. A escolha foi

pelo mapa com QE = 11,601608 e TE = 0,0. Notou-se um aumento no índice QE para todos os experimentos, embora a escolha do menor QE novamente foi do mapa de 12 × 15 neurônios. A Figura 6-15 apresenta o SOM de documentos, onde os grupos foram novamente identificados. Notou-se uma sensível melhora na qualidade da representação, considerando que os documentos dos dois grupos foram mapeados substancialmente mais próximos entre si.

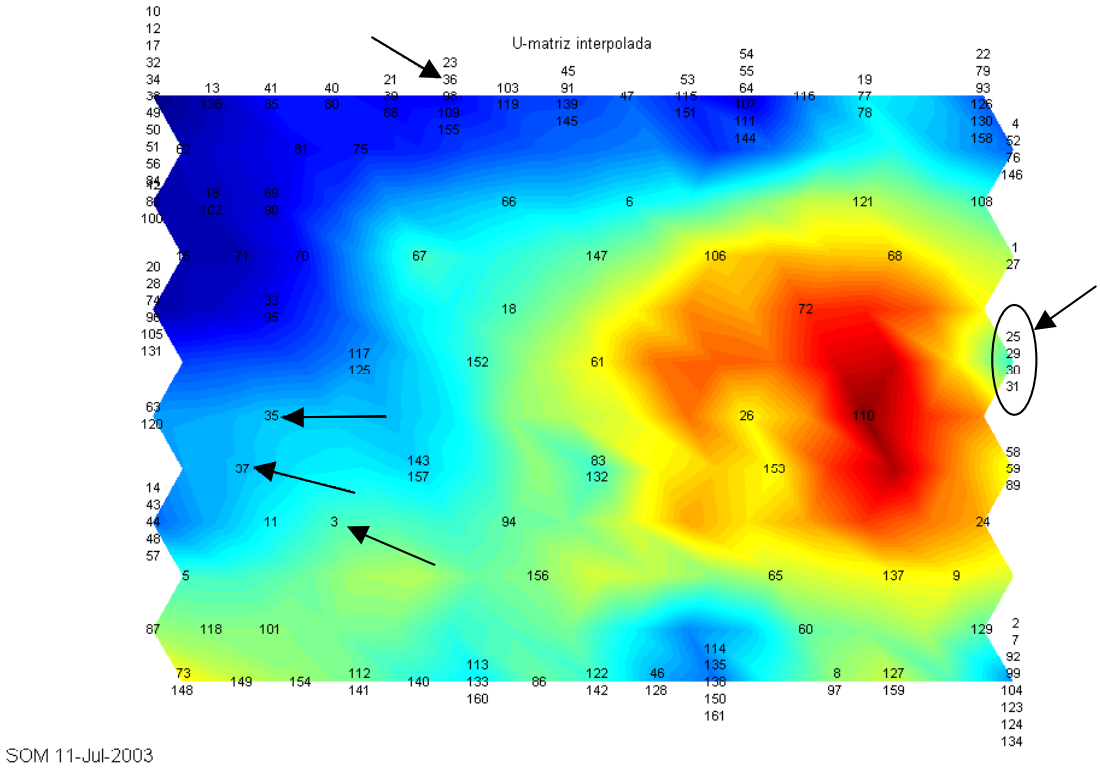


Figura 6-15 – SOM de documentos de 12 × 15 neurônios adaptado a partir dos histogramas dos documentos considerando o 1º e o 2º BMU. Os conjuntos A (documentos 3, 35, 36 e 37) e B (documentos 25, 29, 30 e 31) estão indicados. Nota-se uma sensível melhora na qualidade da representação considerando a proximidade dos documentos em relação à Figura 6-14.

A partir dos conjuntos de histogramas considerando somente o 1º BMU e considerando também o 2º BMU, foram executados 10 testes com cada conjunto utilizando o GTM.

Os modelos escolhidos (com o maior valor para o logaritmo da verossimilhança igual a 123664,593471 para o modelo usando o 1º BMU e 104786,939052 para o modelo usando 2 BMUs) possuem 25 × 25 pontos latentes e 15 × 15 funções-base com espalhamento 0,8 em 5 ciclos de treinamento. A Figura 6-16 apresenta os resultados obtidos.

Notou-se que o GTM é menos sensível à informação do 2º BMU, não apresentando resposta suficientemente diferente para ser observada.

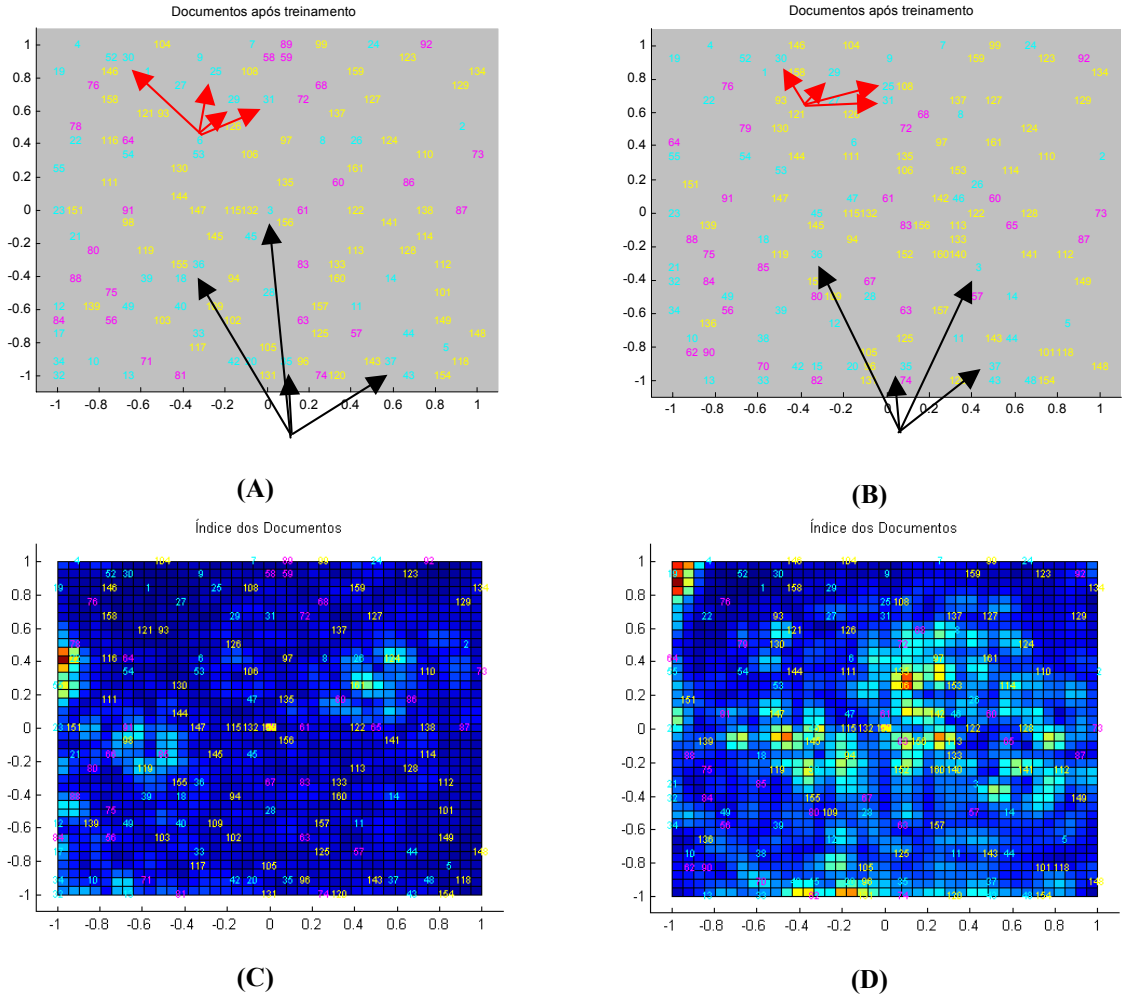


Figura 6-16 – Modelo GTM adaptado considerando os histogramas gerados com o 1º BMU (A e C) e incluindo o 2º BMU (B e D). Identificados com setas pretas estão os documentos do conjunto A (3, 35, 36 e 37) e com setas vermelhas, o conjunto B (25, 29, 30 e 31). O GTM foi pouco sensível à informação do 2º BMU neste experimento.

Um novo experimento buscou reduzir o número de palavras com um valor discriminante muito baixo. De acordo com a Equação 6-3, foi avaliada a freqüência total de ocorrência de cada termo após a radicalização e uma análise manual sugeriu que os termos com freqüência maior que 50 poderiam ser removidos. A Tabela 6-4 apresenta um exemplo de palavras removidas.

Tabela 6-4 – Alguns exemplos de palavras com baixo valor discriminante no conjunto AnUSF. A freqüência de ocorrência bruta do radical destes termos é maior que 50 e foram removidas do conjunto.

apresenta	**pesquisa**	**trabalha**	**utiliza**
apresentação	**pesquisadas**	**trabalhador**	**utilização**
apresentada	**pesquisado**	**trabalhadoras**	**utilizada**
apresentadas	**pesquisadoras**	**trabalhadores**	**utilizadas**
apresentado	**pesquisadores**	**trabalham**	**utilizado**
...	**...**	**...**	**...**

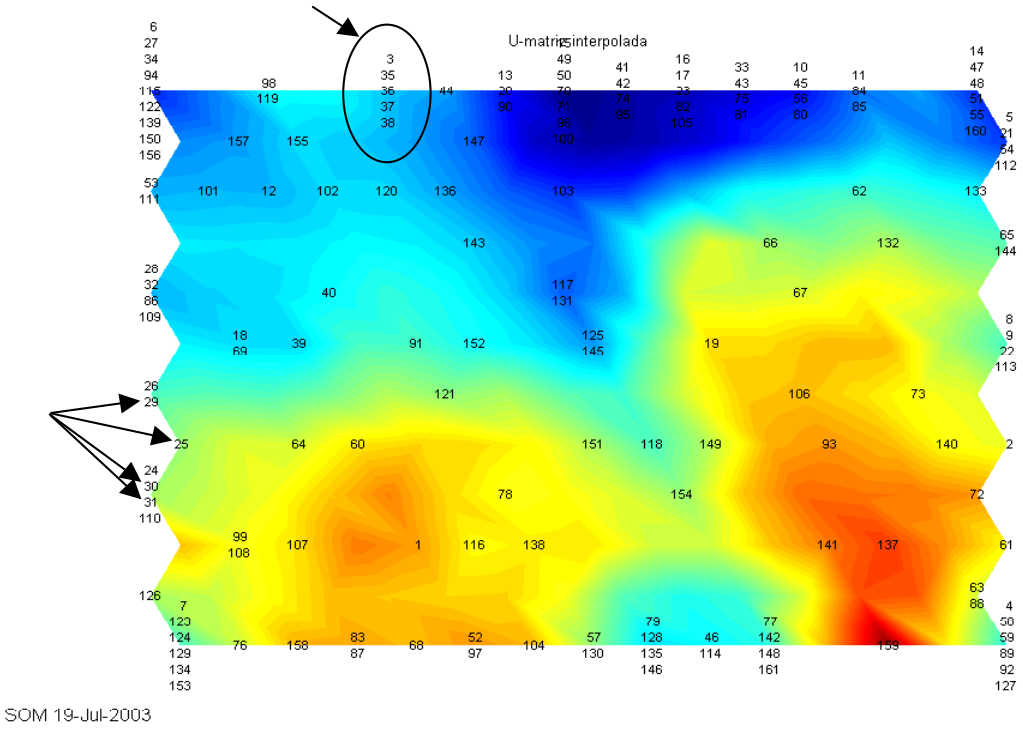
É interessante perceber que, diferentemente dos procedimentos propostos na literatura, optou-se aqui por remover os termos cuja freqüência do *termo radicalizado* foi maior que 50. A literatura realiza esta análise considerando o termo em si.

Como resultado, o conjunto inicial de 34726 palavras foi reduzido para 17032 palavras (foram removidas 17694 palavras, 3204 a mais em relação ao experimento anterior). O conjunto exibiu um total de 6282 palavras distintas que, radicalizadas, somam 3489 termos. Seguindo procedimentos e parâmetros semelhantes aos já descritos no experimento anterior, foi treinado um SOM semântico e, a partir deste, SOMs de documentos.

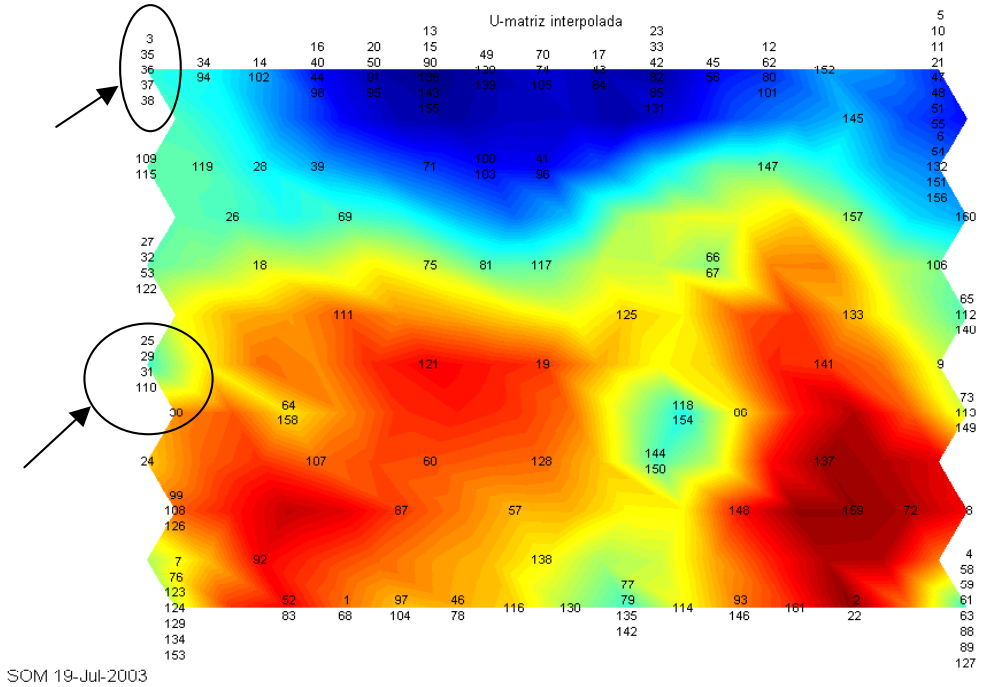
A Figura 6-17 apresenta os resultados dos SOMs de documentos, adaptados a partir dos histogramas de documentos gerados pela apresentação de cada documento ao SOM semântico. É de se notar que a proximidade semântica dos conjuntos de documentos escolhidos é muito mais evidente, sendo aumentada ainda mais com o uso do 2^{o} BMU na construção do histograma de palavras.

De forma semelhante aos testes anteriores, também foram adaptados modelos GTM sobre os histogramas gerados a partir do SOM semântico. Os resultados (apresentados na Figura 6-18) permitem observar que a representação da similaridade entre os documentos de teste é bem melhor que o resultado obtido sem a remoção das palavras muito freqüentes. Novamente, o GTM se mostrou pouco sensível à utilização do 2^{o} BMU nos histogramas de documentos, exibindo entretanto resultados aproveitáveis em ambos os casos.

É de se notar que, tanto no caso do SOM como no caso do GTM, os resultados obtidos a partir do conjunto de texto com a remoção das palavras de baixo valor discriminante foram superiores em relação ao experimento com o conjunto integral de palavras.

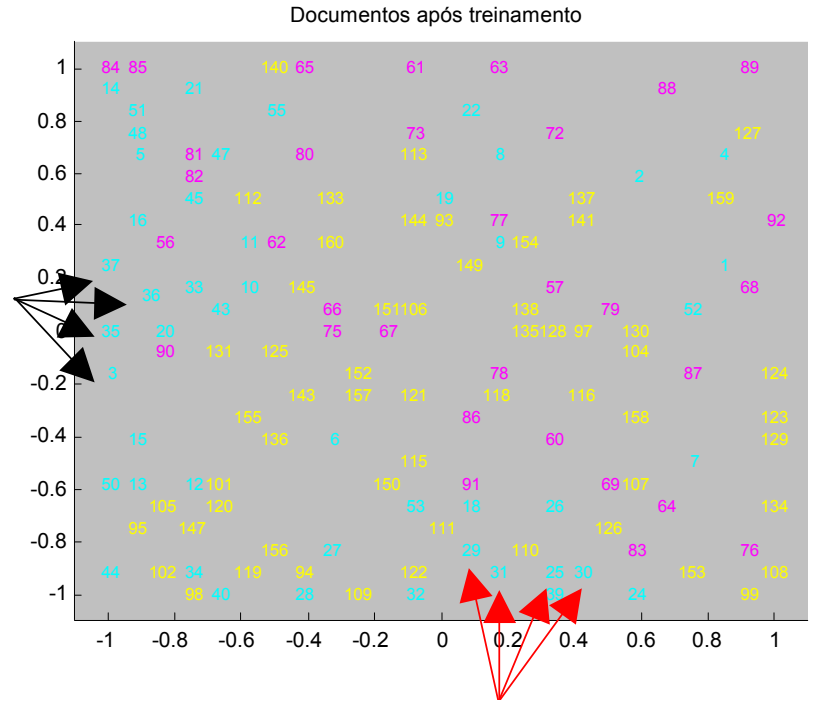


(A)

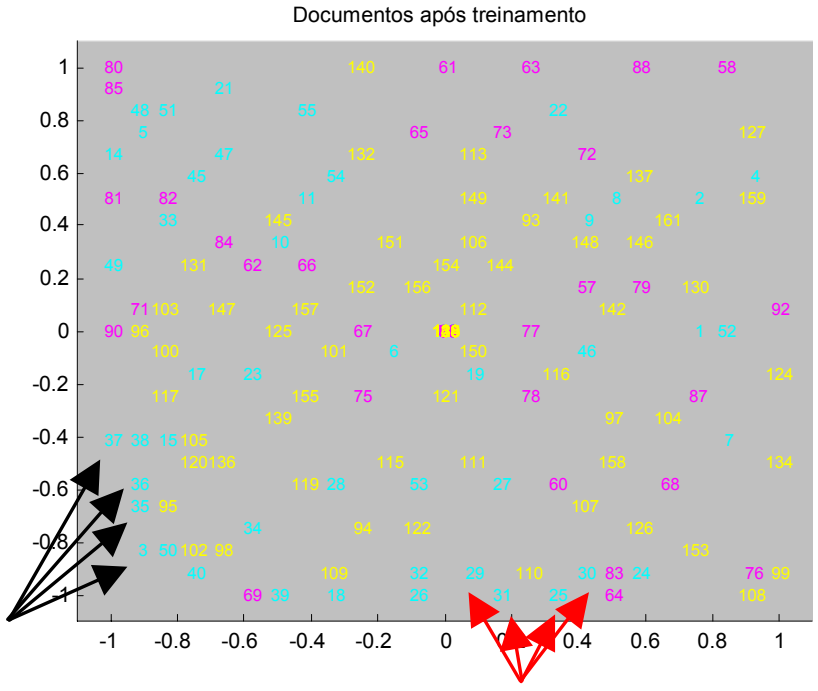


(B)

Figura 6-17 – SOM de documentos após a exclusão de palavras com baixo valor discriminante. Em (A) foi usado somente o 1º BMU e em (B), também o 2º BMU. Observa-se que em (B) o SOM exibiu maior sensitividade ao contexto do conjunto de documentos de teste.



(A)



(B)

Figura 6-18 – GTM adaptado após a remoção das palavras com baixo valor discriminante. Em (A) foi usado somente o 1º BMU e em (B), também o 2º BMU. Em ambos os casos, o GTM exibiu resultados semelhantes, representando os conjuntos de documentos próximos entre si. O conjunto A é indicado por setas pretas e o conjunto B por setas vermelhas.

Finalmente, um último experimento foi realizado com o intuito de aumentar a sensibilidade ao contexto dos mapas gerados nas duas ferramentas. A suposição assumida é a de que o contexto médio, como definido pela Equação 6-8, tende a generalizar por demais o contexto de uma palavra, pois o mesmo é calculado sobre todo o corpo de texto. Se os assuntos tratados pelo corpo de texto são relativamente próximos, é de se esperar que o uso dos termos seja muito parecido ao longo do conjunto. Entretanto, se o conjunto de textos trata de assuntos bastante distintos entre si, é possível que a compressão de significado gerada pela equação do contexto médio deixe de refletir possíveis diferenças entre grupos de significados.

Apenas como um exemplo, a palavra "construção" aparece 45 vezes no conjunto de textos AnUSF, sendo 28 aparições em textos de Ciências Exatas e Tecnológicas (ET) e outras 17 vezes em textos de Ciências Humanas e Sociais. No primeiro conjunto, a palavra se encontra associada a termos como construção "civil, de algoritmos, de equipamentos", enquanto que no segundo tema trata-se de construção "da cidadania, da identidade, do respeito". É inegável que temos aqui dois contextos de uso bastante distintos para a mesma palavra e o cálculo de contexto médio simplesmente ignorará esta diferença.

O experimento consistiu em procurar respeitar ao máximo o contexto de cada palavra levando-se em conta a possibilidade de haver dois ou mais contextos distintos para uma mesma palavra. Considerando que partiu-se do pressuposto de aprendizado totalmente não supervisionado, o cálculo de contexto médio de cada palavra foi realizado *sobre cada documento* em vez de sobre todo o corpo de texto. Esta idéia é representada pelo *contexto médio por documento*:

$$\mathbf{V}_k^{(i)} = \begin{bmatrix} \mathrm{E}\left\{\mathbf{V}_{s(k)-1}^{(i)}\right\} \\ \varepsilon\ \mathbf{V}_{s(k)}^{(i)} \\ \mathrm{E}\left\{\mathbf{V}_{s(k)+1}^{(i)}\right\} \end{bmatrix} \qquad \textbf{Equação 6-11}$$

onde $\mathbf{V}_k^{(i)}$ é o contexto médio da palavra *k* no documento (*i*), $\mathrm{E}\left\{\mathbf{V}_{s(k)-1}^{(i)}\right\}$ é a média de todos os símbolos que precedem a palavra $\mathbf{v}_k$ no documento (*i*) e $\mathrm{E}\left\{\mathbf{V}_{s(k)+1}^{(i)}\right\}$ é a média de todos os símbolos que sucedem a palavra $\mathbf{v}_k$ no documento (*i*). O corpo de texto de 3489 termos

radicalizados do experimento anterior (de onde foram removidas as palavras com baixo valor discriminante) foi processado segundo a Equação 6-11. A operação gerou uma matriz de contexto médio por documento composta por 12402 vetores de dimensão 300, cada vetor representando o contexto médio de um símbolo em cada documento. Este conjunto foi usado para adaptar um SOM semântico de 30 × 30 neurônios em arranjo hexagonal. A média de representação é de 13,78 palavras por neurônio e a partir deste mapa foram gerados os histogramas representando os documentos tomando-se o 1º e 2º BMUs. O SOM de documentos adaptado a partir deste conjunto de vetores é ilustrado na Figura 6-19.

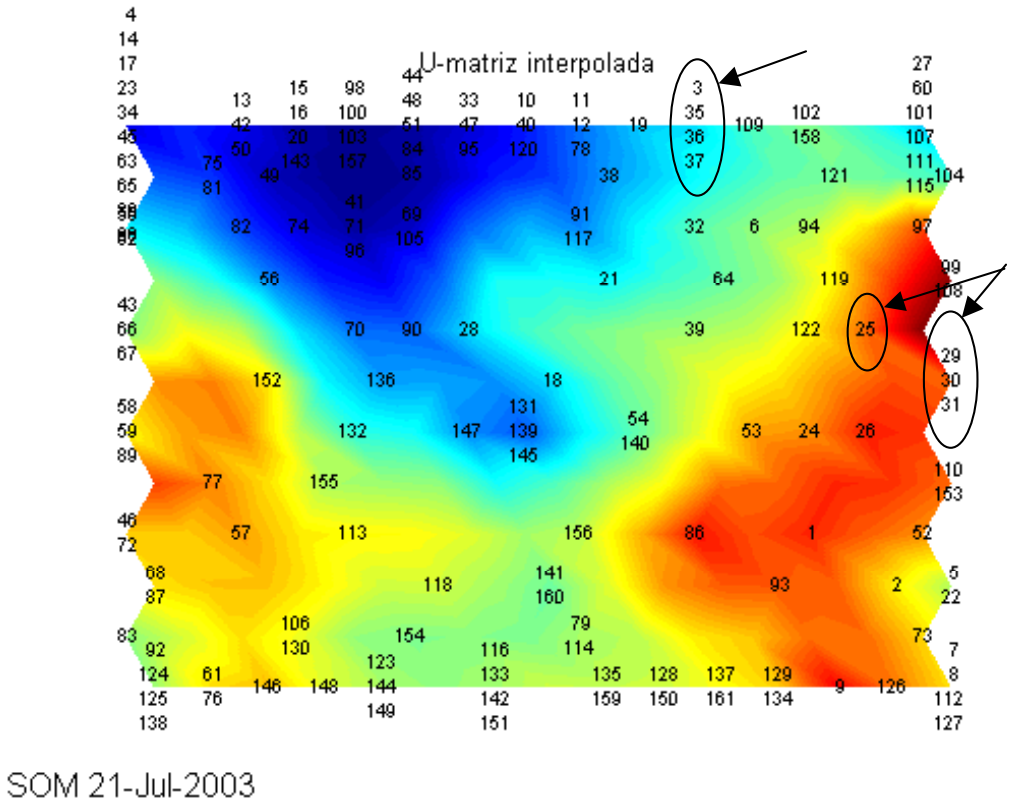


Figura 6-19 – SOM de documentos de 12 × 15 neurônios adaptado a partir de um SOM semântico de contexto médio por documento. Os conjuntos A (documentos 3, 35, 36 e 37) e B (documentos 25, 29, 30 e 31) de documentos de teste estão indicados por setas. A proximidade na representação sobre a matriz-U mostra que este SOM é sensível ao contexto dos documentos.

Embora o resultado obtido com o SOM pareça ter representado adequadamente os documentos segundo seu contexto, de acordo com a indicação dos textos de controle, não fica clara qualquer indicação de agrupamentos pela observação da matriz-U, não sendo percebida nenhuma separação entre os três temas de textos utilizados nos testes.

Partindo do mesmo conjunto de histogramas de documentos gerados a partir do SOM semântico de contexto médio por documento, foi adaptado um modelo GTM de 25 × 25 pontos latentes e 15 × 15 funções-base. O modelo já adaptado está ilustrado na Figura 6-20.

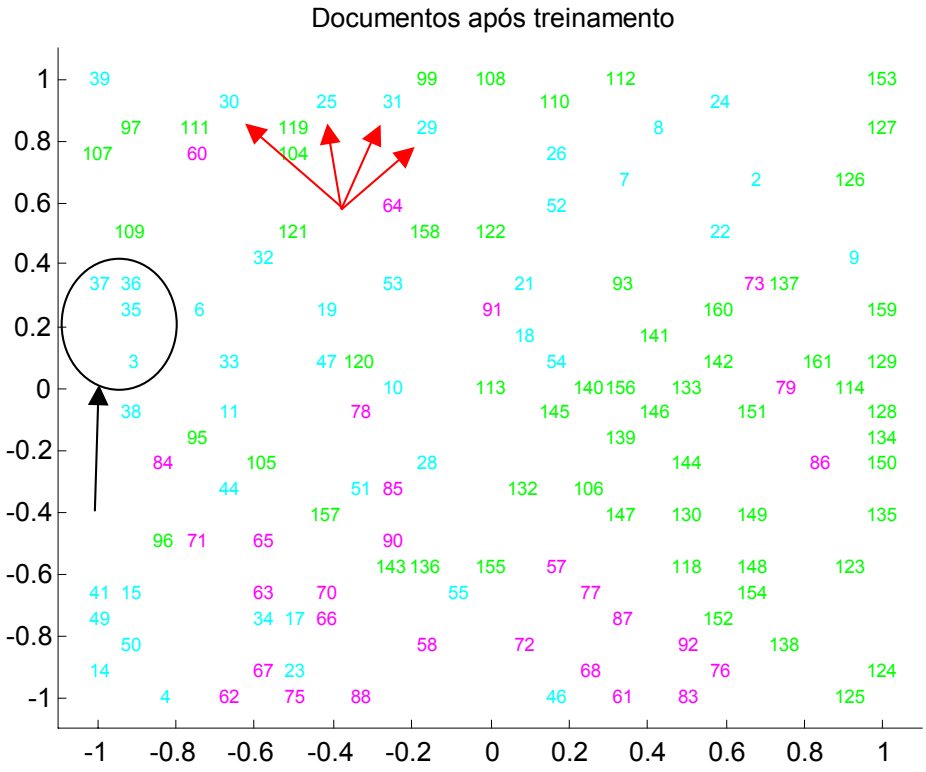


Figura 6-20 – Modelo GTM adaptado a partir dos histogramas de documentos gerados pelo SOM semântico de contexto médio por documento. A proximidade de representação dos documentos confirma a sensibilidade ao contexto dos documentos. As cores representam os rótulos dos grupos, usados aqui apenas para avaliação do resultado obtido: grupo "Exatas e Tecnológicas (ET)" em azul ( ), grupo "Biológicas e de Saúde (BS)" em vermelho ( ) e grupo "Humanas e Sociais (HS)" em verde ( ).

Particularmente neste experimento, o GTM apresentou uma tendência de separação dos documentos em agrupamentos mais distintos entre si. Embora o GTM em si não tenha evidenciado a existência de agrupamentos (Figura 6-20), a sobreposição de conhecimento *a priori* (que não era disponível na adaptação) permite observar que o GTM foi capaz de separar, ainda que grosseiramente, os conjuntos de textos de Exatas e Tecnológicas (ET), Biológicas e de Saúde (BS) e Humanas e Sociais (HS).

Esta constatação não pode ser evidenciada com o SOM e a matriz-U em nenhum experimento relatado ou observado, o que sugere uma interação possivelmente proveitosa

numa ferramenta híbrida SOM-GTM. A Figura 6-21 apresenta uma ilustração onde as possíveis fronteiras dos agrupamentos sugeridos foi adicionada artificialmente considerando o conhecimento *a priori* das classes dos documentos representados a partir do SOM semântico de contexto médio por documento.

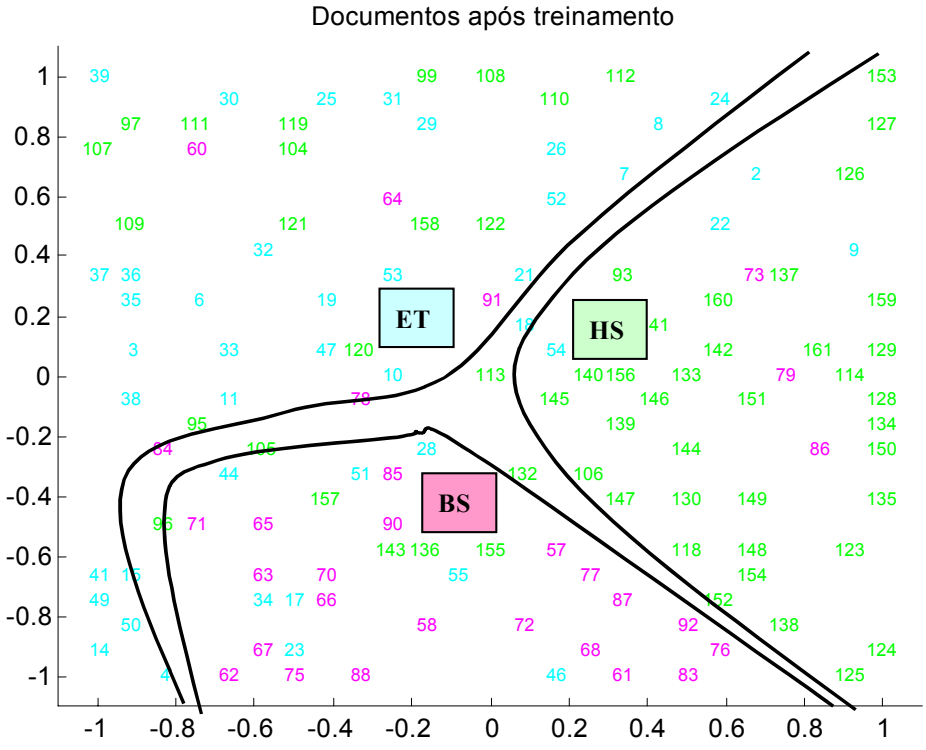


Figura 6-21 – O modelo GTM apresenta uma tendência de separação dos documentos. As fronteiras foram inseridas artificialmente e consideram a classificação *a priori* dos documentos. Em azul os documentos do conjunto ET, em vermelho os documentos BS e em verde os documentos HS.

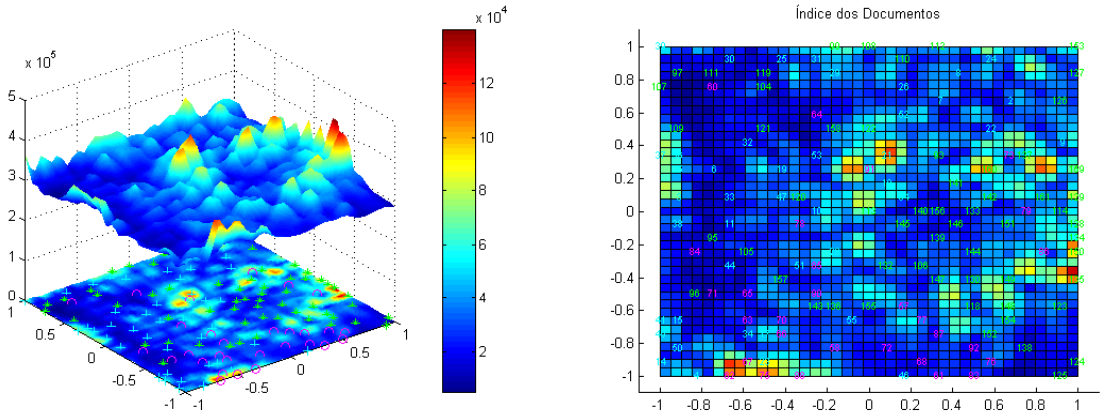


Figura 6-22 – A análise do fator de magnificação do GTM não oferece indícios claros da existência de agrupamentos. Entretanto, uma análise das posições em que os documentos foram representados (Figura 6-21) sugere que o GTM é capaz de confirmar a separação dos documentos em 3 agrupamentos.

Capítulo 7

Conclusão

A pesquisa desenvolvida ao longo deste trabalho buscou oferecer contribuições em duas áreas: na *mineração de dados* e na *mineração de textos*. Mais especificamente, esta dissertação mostrou que o SOM e o GTM constituem recursos poderosos e promissores, quando aplicados às tarefas citadas. Uma característica importante e que deve ser salientada, nesta dissertação, foi o uso de dados reais para avaliar os resultados obtidos com a aplicação das ferramentas. Isto oferece um forte caráter experimental ao trabalho. Foram também abordados alguns dos principais métodos para aplicação em mineração de dados, os quais foram testados e comentados de forma resumida. Pela avaliação de algumas de suas características, bem como algumas de suas limitações, foi gerada uma ampla revisão sobre o estado da arte nesta área. As duas ferramentas pesquisadas mais profundamente nesta dissertação, o SOM e o GTM, foram tratadas em capítulos específicos voltados para a aplicação destas ferramentas à mineração de dados e mineração de textos, incluindo uma análise da influência de seus parâmetros de controle.

7.1 Contribuições

Resumidamente, as principais contribuições gerais deste estudo são:

- revisão conceitual das principais ferramentas com possível aplicação em mineração de dados;
- apresentação das ferramentas SOM e GTM, com estudo da influência de seus parâmetros, no processo de adaptação das mesmas;
- proposição de refinamentos junto a metodologias efetivas para aplicação dos modelos SOM e GTM, quando empregados em mineração de dados e recuperação de informação.

Outras contribuições, especificamente ligadas às tarefas de mineração de dados e mineração de textos, com uso das ferramentas SOM e GTM, podem ser assim resumidas:

- constatação de que a normalização da variância dos vetores de dados, considerando o SOM, não leva necessariamente a bons resultados. De fato, no caso do GTM, esta operação praticamente não conduziu a bons resultados, de onde se formula uma hipótese de que este é um procedimento desnecessário para o GTM e que deve ser utilizado com muito critério no caso do SOM;
- proposição de heurísticas para a obtenção de bons mapeamentos, considerando o SOM e o GTM, nas tarefas de mineração de dados e mineração de textos;
- proposição de uso do 2º BMU na construção do vetor de documentos, a partir do SOM semântico, com resultados melhores que a proposta original da literatura;
- proposição de uso do contexto médio por documento (diferentemente do contexto médio geral), com resultados promissores na mineração de textos;
- proposição de um modelo híbrido SOM-GTM com resultados promissores na mineração de textos;
- proposição de uso da técnica de radicalização na mineração de textos em língua portuguesa;
- constatação da efetiva melhora nos resultados das ferramentas de mineração de texto, ao serem removidas do corpo de texto as palavras com baixo valor discriminante.

7.2 Extensões

Vários tópicos, relacionados direta ou indiretamente com esta pesquisa, podem ser citados como sugestão para pesquisas futuras:

- possibilidade de uso de gaussianas com diferentes variâncias no modelo GTM, de forma a flexibilizar a modelagem dos dados;
- possibilidade de construção do modelo GTM com funções-base não apenas gaussianas;
- desenvolvimento de modelos GTM construtivos, permitindo a inserção e remoção de funções-base;
- desenvolvimento de modelos GTM hierárquicos, com ampliação e exploração automáticas de regiões de dados com grande densidade de pontos;

- flexibilização do formato do **X**-espaço no modelo GTM, com possibilidade de uso de espaços de dimensão maior que 2;
- incremento da ferramenta GTM, promovendo a capacidade de executar redução de dados;
- fundamentação do modelo híbrido SOM-GTM em mineração de textos e mineração de dados;
- utilização de mapas SOM *N*-dimensionais com aplicação de algoritmos de segmentação e rotulação automáticos (por exemplo, SL-SOM) na construção de mapas semânticos;
- desenvolvimento de um algoritmo de radicalização adaptativo;
- buscar interação com lingüistas em etapas de refinamento do algoritmo de pré-processamento;
- desenvolver regras para remoção de palavras com baixo valor discriminante, utilizando técnicas adaptativas, como lógica nebulosa ou algoritmos genéticos, de forma que as regras sejam dependentes do conjunto de textos;
- desenvolvimento de um modelo híbrido de processamento de linguagem natural e redes neurais artificiais, de forma a ampliar a quantidade de informação disponível *a priori* na mineração de textos;
- fundamentar a influência do número de BMUs na geração do vetor de documentos, a partir do SOM semântico, em relação à qualidade obtida na projeção da similaridade contextual;
- otimização dos algoritmos de treinamento das ferramentas SOM e GTM para conjuntos volumosos de dados;
- aplicação de modelos construtivos da ferramenta SOM nas tarefas de mineração de dados e mineração de textos;
- fundamentação do conceito de contexto médio por documento em conjuntos de documentos de texto;
- definição automática de parâmetros das ferramentas SOM e GTM a partir do emprego de lógica nebulosa e algoritmos evolutivos, gerando soluções híbridas para os problemas de mineração de textos e mineração de dados.

Anexo 1

Avaliação de Estilo de Aprendizagem LSI-3

1	**Quando eu aprendo**	**_____ eu gosto de lidar com meus sentimentos**	**_____ eu gosto de observar e escutar**	**_____ eu gosto de pensar sobre idéias**	**_____ eu gosto de estar fazendo coisas**
2	**Eu aprendo melhor quando**	**_____ eu levo em conta meus pressentimentos e sentimentos**	**_____ eu escuto e observo cuidadosamente**	**_____ eu faço uso de raciocínio lógico**	**_____ eu trabalho duro para cumprir as tarefas**
3	**Quando eu estou aprendendo**	**_____ eu tenho sentimentos e reações fortes**	**_____ eu sou quieto e reservado**	**_____ eu sou levado a ponderar as coisas**	**_____ eu sou responsável com as coisas**
4	**Eu aprendo através do**	**_____ sentir**	**_____ observar**	**_____ pensar**	**_____ fazer**
5	**Quando eu aprendo**	**_____ eu estou aberto a novas experiências**	**_____ eu levo em conta todos os ângulos dos assuntos**	**_____ eu gosto de analisar as coisas e decompô-las em suas partes**	**_____ eu gosto de experimentar as coisas**
6	**Quando eu estou aprendendo**	**_____ eu sou uma pessoa intuitiva**	**_____ eu sou uma pessoa observadora**	**_____ eu sou uma pessoa lógica**	**_____ eu sou uma pessoa ativa**
7	**Eu aprendo melhor a partir de(a)**	**_____ relações pessoais**	**_____ observações**	**_____ teorias racionais**	**_____ uma oportunidade para experimentar e praticar**
8	**Quando eu aprendo**	**_____ eu sinto-me pessoalmente envolvido com as coisas**	**_____ eu penso antes de agir**	**_____ eu gosto de idéias e teorias**	**_____ eu gosto de ver os resultados de meu trabalho**
9	**Eu aprendo melhor quando**	**_____ eu levo em conta meus sentimentos**	**_____ eu levo em conta minhas observações**	**_____ eu levo em conta minhas idéias**	**_____ eu posso experimentar as coisas por mim mesmo**
10	**Quando eu estou aprendendo**	**_____ eu sou uma pessoa aberta a sugestões, idéias e críticas**	**_____ eu sou uma pessoa reservada**	**_____ eu sou uma pessoa racional**	**_____ eu sou uma pessoa responsável**
11	**Quando eu aprendo**	**_____ eu me envolvo**	**_____ eu gosto de observar**	**_____ eu avalio as coisas**	**_____ eu gosto de ser ativo**
12	**Eu aprendo melhor quando**	**_____ eu sou receptivo e mente aberta**	**_____ eu sou cuidadoso**	**_____ eu analiso idéias**	**_____ eu sou prático**

Adaptado de Kolb (2000a)

Referências Bibliográficas

(Alahakoon *et al.* 2000) Alahakoon, Daminda; Halgamuge, Saman K.; Srinivasan, Bala. **"Dynamic Self Organising Maps with Controlled Growth for Knowledge Discovery"**. IEEE Transactions on Neural Networks, vol X, nº X, pg. 100-114, 2000.

(Alhoniemi *et al*. 1999) Alhoniemi, Esa; Hinberg, Johan; Vesanto, Juha. **"Probabilistic measures for responses of Self-Organizing Map units"**. In: International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99), Rochester, N.Y., USA. ICSC Academic Press, pp. 286-290, June 22-25, 1999. URL: http://www.cis.hut.fi/projects/ide/publications/papers/aida99a.zip. Recuperado em 21/06/2000.

(Alhoniemi *et al.* 2000) Alhoniemi, Esa ; Himberg, Johan; Parhankangas, Juha; Vesanto, Juha. **"SOM Toolbox v2 Beta"**. Helsinki University of Technlogy, Finland, 2000. URL: http://www.cis.hut.fi/projects/somtoolbox/. Recuperado em 05/01/2000.

(Anand & Hughes, 1998) Anand, Sarabjot S.; Hughes, John G. **"Hybrid Data Mining Systems: The Next Generation"**. In: Proceedings of 2nd Pacific-Asia Conference in Knowledge Discovery and Data Mining, 1998. URL: http://inchinn.infj.ulst.ac.uk/htdocs/Anand_gen.html. Recuperado em 20/04.2001.

(Andrews, 1972) Andrews, D.F. **"Plots of High-Dimensional Data"**. Biometrics, nº 28, pg.125-136, 1972.

(Aras *et al*. 1999) Aras, N.; Oommen, B.J.; Altinel, I.K. **"The Kohonen network incorporating explicit statistics and its application to the travelling salesman problem"**. Neural Networks nº 12, pg.1273-1284, 1999.

(Bartholomew, 1987) Bartholomew, D. J. **"Latent Variable Models and Factor Analisys"**. Charles Griffin and Co. Ltd, London, 1987.

(Bezdek, 1981) Bezdek, J. C. **"Pattern Recognition With Fuzzy Objective Function Algorithms"**, Plenum Press, New York, 1981.

(Bishop *et al*. 1996a) Bishop, Christopher M.; Svensén, Markus; Williams, Christopher K.I. **"EM Optimization of Latent-Variable Models"**. In: Touretzky, D.S.; Mozer, M.C.; Hasselmo, M.E. (editors), Advances in Neural Information Processing Systems 8, The MIT Press, Cambridge, MA, pg. 465-471, 1996. URL: http://www.ncrg.aston.ac.uk/Papers/postscript/NCRG_96_011.ps.Z. Recuperado em 15/09/1999.

(Bishop *et al*. 1996b) Bishop, Christopher M.; Svensén, Markus; Williams, Christopher K.I. **"GTM: The Generative Topographic Mapping"**. Technical Report NCRG/96/015, Aston University, UK. Published in: Neural Computation 10, pg. 215-234, 1998. URL: http://www.ncrg.aston.ac.uk/Papers/postscript/NCRG_96_015.ps.Z. Recuperado em 15/09/1999.

(Bishop *et al*. 1996c) Bishop, Christopher M.; Svensén, Markus; Williams, Christopher K.I. **"GTM: A Principled Alternative to the Self-Organizing Map"**. Technical Report NCRG/96/031, Aston University, UK. Published in: von der Malsburg, C.; von Seelen, W.; Vorbrüggen, J.C.; Sendhoff, B. (editors), Proceedings of the International Conference on Artificial Neural Networks (ICANN'96), Lecture Notes in Computer Science, vol. 1112, Springer, Berlim, pg. 164-170, 1996. URL: http://www.ncrg.aston.ac.uk/Papers/postscript/NCRG_96_031.ps.Z. Recuperado em 15/09/1999.

(Bishop *et al*. 1997) Bishop, Christopher M.; Svensén, Markus; Williams, Christopher K.I. **"Magnification Factors for the GTM Algorithm"**. Technical Report NCRG/97/006, Aston University, UK. Published in: Proceedings IEE Fifth International Conference on Artificial Neural Networks. pg. 64-69. IEE, London, 1997. URL: http://www.ncrg.aston.ac.uk/Papers/postscript/NCRG_97_006.ps.Z. Recuperado em 15/09/1999.

(Bishop *et al*. 1997a) Bishop, Christopher M.; Hinton, Geoffrey E.; Strachan, Iain G.D. **"GTM Through Time"**. Technical Report NCRG/97/005, Aston University, UK. Published in: Proceedings IEE Fifth International Conference on Artificial Neural Networks. pg. 111-116. IEE, London, 1997. URL: http://www.ncrg.aston.ac.uk/Papers/postscript/NCRG_97_005.ps.Z. Recuperado em 15/09/1999.

(Bishop *et al*. 1997c) Bishop, Christopher M.; Svensén, Markus; Williams, Christopher K.I. **"Magnification Factors for the SOM and GTM Algorithm"**. In: Proceedings of the Workshop on Self-Organizing Maps, Helsinki, Finland, 1997. URL: http://www.ncrg.aston.ac.uk/Papers/postscript/NCRG_97_008.ps.Z. Recuperado em 22/10/1999.

(Bishop *et al*. 1998) Bishop, Christopher M.; Svensén, Markus; Williams, Christopher K.I. **"Developments of the Generative Topographic Mapping"**. Neurocomputing. 21, pg. 203-224, 1998. URL: http://www.ncrg.aston.ac.uk/Papers/postscript/NCRG_98_012.ps.Z. Recuperado em 15/09/1999.

(Bishop, 1995) Bishop, Christopher M. **"Neural Networks for Patter Recognition"**. Oxford University Press, 1995.

(Blackmore & Miikkulainen, 1993) Blackmore, J.; Miikkulainen, R. **"Incremental Grid Growing: Encoding High-Dimensional Structure into a Two-Dimensional Feature Map"**. In: Proceedings of the IEEE International Conference on Neural Networks (ICNN'93), vol. I, IEEE Service Center, Piscataway, NJ, pg. 450-455, 1993. URL: http://www.cs.utexas.edu/users/nn/pages/publications/abstracts.html#blackmore.incremental.ps.Z. Recuperado em 22/09/1999.

(Blackmore & Miikkulainen, 1995) Blackmore, Justine; Miikkulainen, Risto. **"Visualizing High-Dimensional Structure with the Incremental Grid Growing Neural Network"**. In A. Prieditis and S. Russell (editors), Machine Learning: Proceedings of the 12th International Conference (ICML'95, Tahoe City, CA), pg. 55-63. San Francisco: Kaufmann, 1995. URL: http://www.cs.utexas.edu/users/nn/pages/publications/abstracts.html#blackmore.ml95.ps.Z. Recuperado em 22/09/1999.

(Blackmore, 1995) Blackmore, Justine. **"Visualizing High-Dimensional Structure with the Incremental Grid Growing Neural Network"**. Master Thesis. Technical Report AI95-238, Department of Computer Sciences, University of Texas at Austin, 1995. URL: http://www.cs.utexas.edu/users/nn/pages/publications/abstracts.html#blackmore.thesis.ps.Z. Recuperado em 22/09/1999.

(Blake & Merz, 1998) Blake, C.L. & Merz, C.J. **"UCI Repository of machine learning databases"**. Irvine, CA: University of California, Department of Information and Computer Science. URL: http://www.ics.uci.edu/~mlearn/MLRepository.html. Recuperado em 22/04/1999.

(Boley *et al.* 1999) Boley, Daniel; Gini, Maria; Gross, Robert; Han, Eui-Hong; Hastings, Kyle; Karyipis, George; Kumar, Vipin; Mobasher, Bamshad; Moore, Jerome. **"Partitioning-based clustering for Web document categorization"**. Decision Support Systems, nº 27, pg. 329-341, 1999.

(Cheng, 1997) Cheng, Yizong. **"Convergence and Ordering of Kohonen's Batch Map"**. Neural Computation 9, pg. 1667-1676, 1997.

(Chernoff, 1973) Chernoff, Herman. **"The Use of Faces to Represent Points in k-dimensional Space Graphically"**. Journal of the American Statistical Association (JASA), nº 68, pg. 361-368, 1973.

(Cho, 1997) Cho, Sung-Bae. **"Self-Organizing Map with Dynamical Node Splitting: Application to Handwritten Digit Recognition"**. Neural Computation 9, pg. 1345-1355, 1997.

(Costa, 1999) Costa, José Alfredo Ferreira. **"Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis"**. Tese de Doutorado. Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica e de Computação, 1999.

(de Castro & Von Zuben) de Castro, Leandro Nunes; Von Zuben, Fernanto José. **An Improving Pruning Technique with Restart for the Kohonen Self-Organizing Feature Map**. Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'99), vol. 3, pp. 1916-1919, July 1999.

(Deerwester *et al.* 1990) Deerwester, S.; Dumais, S.; Furnas, G., Landauer, K. **"Indexing by Latent Semantic Analysis"**. Journal of the American Society on Information Science, n.º 41, pg. 391-407, 1990.

(Demartines & Hérault, 1997) Demartines, P.; Hérault, J. **"Curvilinear Component Analysis: a self-organizing neural network for nonlinear maping of datasets"**. IEEE Transactions on Neural Networks, vol. 8, nº 1, pg. 148-154, 1997.

(Dempster *et al.* 1977) Dempster, A.P.; Laird, N.M.; Rubin, D.B. **"Maximum likelihood for incomplete data via the EM algorithm"**. Journal of the Royal Statistical Society, Series B, n.º 39, pg. 1-38, 1977.

(Duda *et al.* 2000) Duda, R.O., Hart, P.E., Stork, D.G. **"Pattern Classification"**. 2nd edition. Wiley Interscience, 2000.

(Erwin *et al.* 1992) Erwin, E.; Obermayer, K.; Schulten, K. **"Self-Organizing Maps: Ordering, Convergence Properties and Energy Functions"**. Biological Cybernetics 67, pg. 47-55, 1992.

(Etzioni, 1996) Etzioni, Oren. **"The World-Wide Web: Quagmire or Gold Mine?"**. Communications of the ACM, vol. 39, n.º 11, pg. 65-68, 1996.

(Everitt, 1993) Everitt, Brian. **"Cluster Analysis"**. 3rd edition. London: Edward Arnold; New York: John Wiley, 1993.

(Fayyad *et al.* 1996a) Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic **"From Data Mining to Knowledge Discovery: An Overview"** In: Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic; Uthurusamy, Ramasamy (editors), "Advances in Knowledge Discovery and Data Mining". American Association for Artificial Intelligence / MIT Press, Menlo Park, CA, pg. 1-34, 1996.

(Fayyad *et al.* 1996b) Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic **"Knowledge Discovery and Data Mining: Towards a Unifying Framework"** In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI Press, Menlo Park, CA, pg. 82-88, 1996. URL: http://research.microsoft.com. Recuperado 12/07/2000.

(Fayyad *et al.* 1996c) Fayyad, Usama; Haussler, David; Stolorz, Paul. **"Mining Scientific Data"**. Communications of the ACM, vol. 39, n.º 11, pg. 51-57, 1996.

(Fayyad *et al.* 1996d) Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic; Uthurusamy, Ramasamy (editors), **"Advances in Knowledge Discovery and Data Mining"**. American Association for Artificial Intelligence / MIT Press, Menlo Park, CA, 1996.

(Fritzke, 1991a) Fritzke, Bernd. **"Let It Grow: Self-Organizing Feature Maps With Problem Dependent Cell Structure"**. In: Kohonen, Teuvo; Mäkisara, K.; Simula, O.; Kangas, J. (editors), Artificial Neural Networks. Proceedings of the International Conference on Artificial Neural Networks (ICANN'91), vol. I, North-Holland, Amsterdam, pg. 403-408, 1991. URL: http://pikas.inf.tu-dresden.de/~fritzke/papers/fritzke.cell_structures.ps.gz. Recuperado em 15/09/1999.

(Fritzke, 1991b) Fritzke, B. **"Unsupervised Clustering With Growing Cell Structures"**. In: Proceedings of the International Joint-Conference on Neural Networks (IJCNN'91), Seattle, 1991. URL: http://pikas.inf.tu-dresden.de/~fritzke/papers/fritzke.clustering.ps.gz. Recuperado em 15/09/1999.

(Fritzke, 1994) Fritzke, B. **"Growing Cell Structures - a Self-Organizing Network for Unsupervised and Supervised Learning"**. Neural Networks 7, pg. 1441-1460, 1994. URL: http://pikas.inf.tu-dresden.de/~fritzke/papers/fritzke.tr93-26.ps.gz. Recuperado em 15/09/1999.

(Fritzke, 1995a) Fritzke, Bernd. **"A Growing Neural Gas Network Learns Topologies"**. In: Tesauro, G., Touretzky, D. S., and Leen, T. K. (editors). *Advances in Neural Information Processing Systems 7*. MIT Press, Cambridge MA, pg 625-632, 1995. URL: http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/fritzke/papers/fritzke.nips94.ps.gz. Recuperado em 15/09/1999.

(Fritzke, 1995b) Fritzke, Bernd. **"Growing Grid - A Self-Organizing Network With Constant Neighborhood Range And Adaptation Strength"**. Neural Processing Letters, vol. 2, n.º 5, pg. 9-13, 1995. URL: http://pikas.inf.tu-dresden.de/~fritzke/ftppapers/fritzke.growing_grid.ps.gz. Recuperado em 15/09/1999.

(Fritzke, 1996) Fritzke, Bernd. **"Growing Self-organizing Networks – Why?"**. In: M. Verleysen (editor). European Symposium on Artificial Neural Networks, D-Facto Publishers, Brussels, pg. 61-72, 1996. URL: http://pikas.inf.tu-dresden.de/~fritzke/ftppapers/fritzke.esann96.ps.gz. Recuperado em 15/09/1999.

(Fritzke, 1997) Fritzke, Bernd. **"Some Competitive Learning Methods"**. Institute for Neural Computation, Ruhr-Universität Bochum, draft from April 5, 1997. URL: http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/sclm.ps.gz. Recuperado em 13/04/2000.

(Goodhill & Sejnowski, 1997) Goodhill, Geoffrey J.; Sejnowsky, Terrence J. **"A Unifying Objective Function for Topographic Mappings"**. Neural Computation 9, pg. 1291-1303, 1997.

(Gower & Ross, 1969) Gower, J.C.; Ross, G.J.S. **"Minimum Spanning Trees and single linkage cluster analysis"**. Applied Statistics nº 18, pg 54-64, 1969.

(Hastie & Stuetzle, 1989) Hastie, Trevor; Stuetzle, Werner. **"Principal Curves"**. Journal of the American Statistical Association (JASA), vol. 84, nº 406, pg. 502-516.

(Haykin, 1999) Haykin, Simon. **"Neural Networks. A Comprehensive Foundantion"**. 2nd edition. Prentice-Hall, 1999.

(Hettich & Bay, 1999) Hettich, S. and Bay, S. D. **"The UCI KDD Archive"**. Irvine, CA: University of California, Department of Information and Computer Science. URL: http://kdd.ics.uci.edu. Recuperado em 22/04/1999.

(Holsheimer & Siebes, 1994) Holsheimer, Marcel; Siebes, Arno P.J.M. **"Data Mining: the search for knowledge in databases"**. Report CS-R9406. Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands, 1994. URL: http://www.cwi.nl/ftp/CWIreports/AA/CS-R9406.ps.Z. Recuperado 18/09/1998.

(Honkela *et al.* 1995) Honkela, Timo; Pulkki, Ville; Kohonen, Teuvo. **"Contextual Relations of Words in Grimm Tales, Analyzed by Self-Organizing Map"**. In: Fogelman-Soulie, F. and Gallinari, P. (editors), Proceedings of International Conference on Artificial Neural Networks (ICANN'95), EC2 et Cie, Paris, pg. 3-7, 1995. URL: http://websom.hut.fi/websom/doc/grimmsom.ps.gz. Recuperado em 20/01/1999.

(Honkela *et al.* 1996a) Honkela, Timo; Kaski, Samuel; Lagus, Krista; Kohonen, Teuvo. **"Exploration of Full-Text Databases with Self-Organizing Maps"**. In: Proceedings of International Conference on Neural Networks (ICNN'96), vol. 1, IEEE Service Center, Piscataway, NJ, pg. 56-61,1996. URL: http://websom.hut.fi/websom/doc/ps/honkela96.ps.gz. Recuperado em 04/12/1998

(Honkela *et al.* 1996b) Honkela, Timo; Samuel, Kaski; Lagus, Krista; Kohonen, Teuvo. **"Newsgroup Exploration with WEBSOM Method and Browsing Interface"**. Report A32, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1996. URL: http://websom.hut.fi/websom/doc/websom.ps.gz. Recuperado em 04/12/1998.

(Honkela *et al.* 1997) Honkela, Timo; Kaski, Samuel; Lagus, Krista; Kohonen, Teuvo. **"WEBSOM – Self-Organizing Maps of Document Collections"**. In: Proceedings of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, pg. 310-315, June 1997. URL: http://www.cis.hut.fi/wsom97/progabstracts/ps/honkela_1.ps. Recuperado 18/02/1999.

(Honkela, 1997) Honkela, Timo. **"Comparisons of Self-Organized Word Category Maps"**. In: Proceedings of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, pg. 298-303, 1997.

(Honkela, 1997a) Honkela, Timo. **"Self-Organizing Maps in Natural Language Processing"**. Dr. Philosophy Thesis. Helsinki University of Technology, Finland, 1997. URL: http://www.cis.hut.fi/~tho/thesis/honkela.ps.Z. Recuperado em 04/12/1998.

(Honkela, 1997c) Honkela, Timo. **"Learning to Understand – General Aspects of Using Self-Organizing Maps in Natural Language Processing"**. In: Proceedings of the Computing Anticipatory Systems (CASYS'97), Liège, Belgium, pg. 563-576, 1997.

(Huang *et al.* 1998) Huang, Guang-Bin; Babri, Haroon A.; Li, Hua-Tian. **"Ordering of Self-Organizing Maps in Multidimensional Cases"**. Neural Computation 10, pg. 19-23, 1998.

(Huber 1985) Huber, P. J. **"Projection Pursuit (with Discussion)"**. The Annals of Statistics, vol. 13, nº 2, pg. 435-475, 1985.

(Iivarinen *et al*. 1994) Iivarinen, J.; Kohonen, Teuvo; Kangas, J.; Kaski, Samuel. **"Visualizing the Clusters on the Self-Organizing Map"**. In: Carlsson, C., Järvi, T., Reponen, T. (editors), Proceedings of the Conference on Artificial Intelligence Research in Finland, Finnish Artificial Intelligence Society, pg. 122-126, 1994.

(Jain & Dubes, 1988) Jain, Anil K.; Dubes, R. C. **"Algorithms for Clustering Data"**. Prentice-Hall, Englewood Cliffs, NJ, 1988.

(Jain *et al*. 1999) Jain, Anil K.; Murty, M.N.; Flynn, P.J. **"Data Clustering: A Review"**. In: ACM Computing Surveys, vol. 31, n.º 3, pg. 264-323, 1999.

(Jolliffe, 1986) Jolliffe, I.T. **"Principal Component Analysis"**, Springer-Verlag, New York, 1986.

(Kangas *et al*. 1990) Kangas, Jari A.; Kohonen, Teuvo K.; Laaksonen, Jorma T. **"Variants of Self-Organizing Maps"**. IEEE Transactions on Neural Networks, vol. 1, n.º 1, pg. 93-99, 1990.

(Kaski & Kohonen, 1996) Kaski, Samuel; Kohonen, Teuvo. **"Exploratory Data Analysis by the Self-Organizing Map: Structures of Welfare and Poverty in the World"**. In: Refenes, A.-P. N.; Abu-Mostafa, Y.; Moody, J. and Weigend, A.; editors, Neural Networks in Financial Engineering. Proceedings of the Third International Conference on Neural Networks in the Capital Markets, World Scientific, Singapore, pg. 498-507, 1996. URL: http://www.cis.hut.fi/~sami/nncm95.ps.gz. Recuperado em 24/09/1998.

(Kaski & Kohonen, 1997) Kaski, Samuel; Kohonen, Teuvo. **"Winner-Takes-All Networks".** In: E. Alhoniemi, J. Iivarinen, and L. Koivisto (editors) Triennial Report 1994-1996, Neural Networks Research Centre & Laboratory of Computer and Information Science, Helsinki University of Technology, Finland, pg. 72-75, 1997.

(Kaski & Lagus, 1996) Kaski, Samuel; Lagus, Krista. **"Comparing Self-Organizing Maps"**. In: von der Malsburg, C.; von Seelen, W.; Vorbrüggen, J.C.; Sendhoff, B. (editors). Proceedings of International Conference on Artificial Neural Networks (ICANN'96). Lecture notes in Computer Science vol. 1112, Springer, Berlim, pg. 809-814, 1996. URL: http://www.cis.hut.fi/~sami/papers/critfin.ps.gz. Recuperado em 04/12/1998.

(Kaski *et al.* 1996) Kaski, Samuel; Honkela, Timo; Lagus, Krista; Kohonen, Teuvo. **"Creating an Order in Digital Libraries with Self-Organizing Maps"**. In: Proceedings of World Congress on Neural Networks (WCNN'96), Lawrence Erlbaum and INSS Press, Mahwah, NJ, pg. 814-817, 1996. URL: http://websom.hut.fi/websom/doc/ps/kaski96wcnn.ps.gz. Recuperado em 04/12/1998.

(Kaski *et al.* 1998a) Kaski, Samuel; Lagus, Krista; Honkela, Timo; Kohonen, Teuvo. **"Statistical Aspects of the WEBSOM System in Organizing Document Collections"**. Computing Science and Statistics, 29, pg. 281-290, 1998. URL: http://websom.hut.fi/websom/doc/ps/kaski98stat.ps.gz. Recuperado em 14/12/1998.

(Kaski *et al.* 1998b) Kaski, Samuel; Nikkilä, Janne; Kohonen, Teuvo. **"Methods for Interpreting a Self-Organized Map in Data Analysis"**. In: Michel Verleysen (editor), Proceedings of the 6th European Symposium on Artificial Neural Networks (ESANN'98), Bruges, April 22-24, D-Facto, Brussels, Belgium, pg. 185-190, 1998. URL: http://www.cis.hut.fi/~sami/papers/esann98_reprint.ps.gz. Recuperado em 10/01/2000.

(Kaski *et al.* 1999) Samuel Kaski, Jarkko Venna, and Teuvo Kohonen. **"Coloring that Reveals High-Dimensional Structures in Data"**. In Proceedings of ICONIP'99, 1999, to appear. URL: http://www.cis.hut.fi/~sami/papers/iconip99.ps.gz

(Kaski, 1997) Kaski, Samuel. **"Data Exploration using Self-Organizing Maps"**. Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series n.º 82. Dr. Tech Thesis, Helsinki University of Technology, Finland, 1997. URL: http://www.cis.hut.fi/~sami/thesis.ps.gz. Recuperado 24/09/1998.

(Kaski, 1998) Kaski, Samuel. **"Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering"**. In: Proceedings of IEEE International Joint Conference on Neural

Networks (IJCNN'98), Anchorage, Alaska, pg. 413-418, 1998. URL: http://websom.hut.fi/websom/doc/ps/kaski98ijcnn.ps.gz. Recuperado em 14/12/1998.

(Kaski, 1999) Samuel Kaski. **"Fast winner search for SOM-based monitoring and retrieval of high-dimensional data"**. In Proceedings of ICANN99, Ninth International Conference on Artificial Neural Networks, volume 2, pages 940-945. IEE, London, 1999. URL: http://www.cis.hut.fi/~sami/papers/icann99_preprint.ps.gz. Recuperado em 04/11/1999.

(Kiviluoto & Oja, 1997) Kiviluoto, Kimmo; Oja, Erkki. **"S-Map: A network with a simple self-organization algorithm for generative topographic mapping"**. In: M. I. Jordan, M. J. Kearns and S. A. Solla (editors), Advances in Neural Processing Systems 10, MIT Press, pg. 549-555, 1997. URL: http://www.cis.hut.fi/~kkluoto/publications/nips97.ps.gz. Recuperado em 25/07/2000.

(Kiviluoto, 1995) Kiviluoto, Kimmo. **"Topology Preservation in Self-Organizing Maps"**. Technical Report A29, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland, 1995.

(Kohonen *et al*. 1995a) Kohonen, Teuvo; Hynninen, Jussi; Kangas, Jari; Laaksonen, Jorma. **"SOM_PAK. The Self-Organizing Map Program Package"**. Version 3.1. Helsinki University of Technology, Laboratory of Computer and Information Science, Finland, April 7, 1995. URL: http://www.cis.hut.fi/nnrc/som_pak/. Recuperado em 04/12/1998.

(Kohonen *et al.* 1996) Kohonen, Teuvo; Kaski, Samuel; Lagus, Krista; Honkela, Timo. **"Very Large Two-Level SOM for the Browsing of Newsgroups"**. In: von der Malsburg, C.; von Seelen, W.; Vorbrüggen, J.C.; Sendhoff, B. (editors). Proceedings of International Conference on Artificial Neural Networks (ICANN'96). Lecture notes in Computer Science vol. 1112, Springer, Berlim, pg. 269-274, 1996. URL: http://websom.hut.fi/websom/doc/ps/kohonen96icann.ps.gz. Recuperado em 04/12/1998.

(Kohonen *et al.* 1997) Kohonen, Teuvo; Kaski, Samuel; Lappalainen. **"Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM"**. Neural Computation, vol 9, n.º 6, pg. 1321-1344, 1997. URL: http://www.cis.hut.fi/nnrc/assom_nc.ps.gz. Recuperado 04/12/1998.

(Kohonen *et al.* 2000) Kohonen, Teuvo; Kaski, Samuel; Lagus, Krista; Saljärvi, Jarkko; Honkela, Jukka; Paatero, Vesa; Saarela, Antti. **"Self Organization of a Massive Document Collection"**. IEEE Transactions on Neural Networks, vol. 11, n.º 3, pg. 574-585, 2000.

(Kohonen, 1981a) Kohonen, Teuvo. **"Automatic Formation of Topological Maps of Patterns in a Self-Organizing System"**. In: Oja, E. and Simula, O. (editors). Proceedings of the 2nd Scandinavian Conference on Image Analysis (Suomen Hahmontunnistustutkimuksen Seura, r.y., Helsinki, Finland), pg. 214-220, June 15-17, 1981.

(Kohonen, 1981b) Kohonen, Teuvo. **"Hierarchical Ordering of Vectoral Data in a Self-Organizing Algorithm"**. Report TKK-F-A461, Helsinki University of Technology, 1981

(Kohonen, 1981c) Kohonen, Teuvo. **"Construction of Similarity Diagrams for Phonemes by a Self-Organizing Algorithm"**. Report TKK-F-A463, Helsinki University of Technology, Espoo, Finland, 1981.

(Kohonen, 1982a) Kohonen, Teuvo. **"Self-Organized Formation of Topologically Correct Feature Maps"**. Biological Cybernetics 43, pg. 59-69, 1982.

(Kohonen, 1982b) Kohonen, Teuvo. **"Analysis of a Simple Self-Organizing Process"**. Biological Cybernetics 44, pg. 135-140, 1982.

(Kohonen, 1996) Kohonen, Teuvo. **"Emergence of Invariant-Feature Detectors in the Adaptive-Subspace Self-Organizing Map"**. Biological Cybernetics 75, pg. 281-291, 1996.

(Kohonen, 1997) Kohonen, Teuvo. **"Self-Organizing Maps"**. Series in Information Sciences, vol. 30, 2nd edition. Springer-Verlag, Heidelberg, 1997.

(Kohonen, 1998) Kohonen, Teuvo. **"Self-Organization of Very Large Document Collection: State of the Art"**. In: Niklasson, L.; Bodem, M.; Ziemke, T. (editors). Proceedings of the 8th International Conference on Artificial Neural Networks, vol. 1, Springer, London, pg. 65-74, 1998. URL: http://websom.hut.fi/websom/doc/ps/kohonen98.ps.gz. Recuperado em 04/12/1998.

(Koikkalainen, 1994) Koikkalainen, P. **"Progress With The Tree-Structured Self-Organizing Map"**. In: Cohn, A.G. (editor). Proceedings of the 11th European Conference on Artificial Inteliggence, New York, pg. 211-215, 1994.

(Kolb, 1984) Kolb, David Allen. **"Experiential Learning: Experiences as the Source of Learning and Development"**. Englewood Cliffs, New Jersey, Prentice-Hall, 1984.

(Kolb, 2000a) Kolb, David Allen. **"Learning Style Inventory. Version 3"**. Hay/McBer Training Resources Group, Boston, MA, 2000. URL: http://trgmcber.haygroup.com. Recuperado em 11/07/2000/

(Kolb, 2000b) Kolb, David Allen. **"Facilitator's Guide to Learning"**. Hay/McBer Training Resources Group, Boston, MA, 2000. URL: http://trgmcber.haygroup.com. Recuperado em 11/07/2000.

(Kraaijveld *et al.* 1995) Kraaijveld, M.A.; Mao, J.; Jain, A.K. **"A Non-linear Projection Method Based on Kohonen's Topology Preserving Maps"**. IEEE Transactions on Neural Networks vol. 6, n.º 3, pg. 548-559, 1995.

(Kruskal & Wish, 1978) Kruskal, Joseph B.; Wish, M. **"Multidimensional Scaling"**. Sage University Paper series on Quantitative Applications in the Social Sciences, nº 07/011. Sage Publications, Newbury Park, CA, 1978

(Kruskal *et al.* 1993) Kruskal, Joseph B.; Young, Forrest W., Seery, C. Judith B., **"Kruskal-Young-Shepard-Torgerson Multidimensional Scaling Program"**. AT&T Bell Laboratories, 1993. URL: http://netlib.bell-labs.com/netlib/mds/kyst2a.f.gz. Recuperado em 15/03/2001.

(Kubat *et al.* 1998) Kubat, Miroslav; Bratko, Ivan; Michalski, Ryszard S. **"A Review of Machine Learning Methods"**. In: Michalski, R.S.; Bratko, I.; Kubat, M. (editors) "Machine Learning and Data Mining. Methods and Applications". John Wiley & Sons, cap. 1, pg. 3-66, 1998.

(Lagus *et al.* 1996a) Lagus, Krista; Honkela, Timo; Kaski, Samuel; Kohonen, Teuvo. **"Self-Organizing Maps of Document Collections: A New Approach to Interactive Exploration"**. In: Simoudis, E., Han, J., Fayyad, U. (editors), Proceedings of the Second International Conference on Knowledge Discovery & Data Mining, AAAI Press, Menlo Park, California, pg. 238-243, 1996. URL: http://websom.hut.fi/websom/doc/ps/lagus96kdd.ps.gz. Recuperado em 04/12/1998.

(Lagus *et al.* 1996b) Lagus, Krista; Kaski, Samuel; Honkela, Timo; Kohonen, Teuvo. **"Browsing Digital Libraries with the Aid of Self-Organizing Maps"**. In: Proceedings of the Fifth International World Wide Web Conference (WWW5), Paris, France, pg. 71-79, 1996. URL: http://websom.hut.fi/websom/doc/ps/lagus96.ps.gz. Recuperado em 04/12/1998.

(Lagus, 1997) Lagus, Krista. **"Map of WSOM'97 Abstracts – Alternative Index"**. In: Proceedings of Workshop on Self-Organizing Maps (WSOM'97), Espoo, Finland, pg. 368-372, July 1997. URL: http://websom.hut.fi/websom/doc/ps/lagus97.ps.gz. Recuperado em 04/12/1998.

(Lagus, 1998) Lagus, Krista. **"Generalizability of the WEBSOM Method to Document Collections of Various Types"**. In: Proceedings of 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98), vol. 1, Aachen, Germany, pg. 210-214, 1998. URL: http://websom.hut.fi/websom/doc/ps/lagus98eufit.ps.gz. Recuperado em 04/12/1998.

(Lagus, 1999) Krista Lagus and Samuel Kaski. **"Keyword selection method for characterizing text document maps"**. In: Proceedings of ICANN99, Ninth International Conference on Artificial Neural Networks, volume1, pages 371-376. IEE, London, 1999. URL: http://websom.hut.fi/websom/doc/ps/lagus99icann.ps.gz. Recuperado em 04/12/1998.

(Lagus, 2000) Krista Lagus. **"Text Mining with the SOM"**. Acta Polytechnica Scandinavica, Mathematics and Computing Series n.º 110. Dr. Tech Thesis, Helsinki University of Technology, Finland, 2000.

(Lee *et al.* 2000) Lee, John Aldo; Lendasse, Amaury; Donkers, Nicolas; Verleysen, Michel. **"A Robust Nonlinear Projection Method"**. In: M. Verleysen (editor). European Symposium on Artificial Neural Networks, D-Facto Publishers, Brussels, pg. 13-20, April 2000. URL: http://www.dice.ucl.ac.be/neural-nets/Research/Publications/2000/es2000-35.pdf. Recuperado em 30/03/2001.

(Li & Sin, 1998) Lin, Siming; Si, Jennie. **"Weight-Value Convergence of the SOM Algorithm for Discrete Input"**. Neural Computation 10, pg. 807-814, 1998.

(Lin *et al.* 1991) Lin, X.; Soergel, D.; Marchionini, G. **"A Self-Organizing Semantic Map for Information Retrieval"**. In: Proceedings of the 14th Annual International ACM-SIGIR Conference on Research & Development in Information Retrieval, Chicago, IL, pg. 262-269, 1991.

(Lin *et al.* 1999) Lin, Chienting; Chen, Hsinchun; Nunamaker, Jay F. **"Verifying the Proximity Hypothesis for Self-organizing Maps"**. In: Proceedings of the 32nd Hawaii International Conference on System Sciences, Jan, 1999.

(Luhn, 1958) Luhn, H. P. **"The Automatic Creation of Literature Abstracts"**. IBM Journal of Research and Development, vol. 2, nº 2, pg. 159-165, 1958.

(MacKay & Gibbs, 1997) MacKay, D. J. C.; Gibbs, M. N. **"Density Networks"**. In: Kay & Titterington (editors) Proceedings of Meeting on Statistics and Neural Nets, Edinburgh, 1997.

(MacQueen, 1967) McQueen, J. B. **"Some Methods for Classification and Analysis of Multivariate Observations"**. In: Le Cam, L.M. and Neyman, J. (editors) Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, vol I, pg. 281-297, 1967.

(Mao & Jain, 1995) Mao, J.; Jain, A.K. **"Artificial Neural Networks for Feature Extraction and Multivariate Data Pojection"**. IEEE Transactions on Neural Networks 6, n.º 2, pg. 296-317, 1995.

(Martinetz & Schulten, 1991) Martinetz, Thomas; Schulten, Klaus. **"A 'Neural Gas' Network Learns Topologies"**. In: Kohonen, T., Mäkisara, K., Simula, O. and Kangas, J. (editors), "Artificial Neural Networks". Elsevier Science Publishers, Amsterdam, pg. 397-402, 1991. URL: http://www.inb.mu-luebeck.de/publications/publists/Martinetz-d.html. Recuperado em 13/07/2002.

(Michalski & Kaufman, 1998) Michalski, Ryszard S.; Kaufman, Kenneth A. **"Data Mining and Knowledge Discovery: A Review of Issues and a Mustistrategy Approach"**. In: Michalski, R.S.; Bratko, I.; Kubat, M. (editors) "Machine Learning and Data Mining. Methods and Applications". John Wiley & Sons, cap. 2, pg. 71-105, 1998.

(Michalski *et al.* 1998) Michalski, R.S.; Bratko, I.; Kubat, M. (editors) **"Machine Learning and Data Mining. Methods and Applications"**. John Wiley & Sons, 1998.

(Miikkulainen, 1990) Miikkulainen, Risto. **"Script Recognition With Hierarchical Feature Maps"**. Connection Science 2, pg. 83-101, 1990. URL: http://www.cs.utexas.edu/users/nn/pages/publications/abstracts.html#miikkulainen.script-recognition.ps.Z. Recuperado em 22/09/1999.

(Miikkulainen, 1997) Miikkulainen, Risto. **"Natural Language Processing With Subsymbolic Neural Networks"**. In: A. Browne (editor), Neural Network Perspectives on Cognition and Adaptive Robotics. Institute of Physics Publishing, 1997. URL: http://www.cs.utexas.edu/users/nn/pages/publications/abstracts.html#miikkulainen.perspectives.ps.Z. Recuperado em 22/09/1999.

(Mitchell, 1999) Mitchell, Tom M. **"Machine Learning and Data Mining"**. Communications of the ACM, vol. 49, n.º 11, November 1999.

(Orengo & Huyck, 2001) Orengo, Viviane Moreira; Huyck, Christian. **"A Stemming algorithm for the Portuguese Language"**. In: Proceedings of SPIRE'2001 Symposium on String Processing and Information Retrieval, Laguna de San Raphael, Chile, November 2001.

(Papoulis, 1991) Papoulis, Athanasios. **"Probability, Random Variables, and Stochastic Processes"**. 3rd edition. McGraw-Hill International Edition. Electrical & Electronic Engineering Series, Singapore, 1991.

(Porter, 1980) Porter, M.F. **"An Algorithm for Suffix Stripping"**. Program vol. 14, nº 3, pg. 130-137, 1980.

(Pullwitt, 2002) Pullwitt, Daniel. **"Integrating Contextual Information to Enhance SOM-Based Text Document Clustering"**. Neural Networks 15, Special Issue, pg. 1099-1106, 2002.

(Reinelt, 1991) Reinelt, G. **"TSPLIB – A Travelling Salesman Problem Library"**. URL: http://softlib.rice.edu/softlib/tsplib/. Recuperado em 17/07/2000.

(Ritter & Kohonen, 1989) Ritter, Helge; Kohonen, T. **"Self-Organizing Semantic Maps"**. Biological Cybernetics 61, pg. 241-254, 1989.

(Ritter & Schulten, 1986) Ritter, Helge; Schulten, Klaus. **"On the Stationary State of Kohonen's Self-Organizing Sensory Mapping"**. Biological Cybernetics 54, pg. 99-106, 1986.

(Ritter & Schulten, 1988) Ritter, Helge; Schulten, Klaus. **"Convergence Properties of Kohonen's Topology Conserving Maps: Fluctuations, Stability, and Dimension Selection"**. Biological Cybernetics 60, pg. 59-71, 1988.

(Salton & McGill, 1983) Salton, G.; McGill, M.J. **"Introduction to Modern Information Retrieval"**. McGraw-Hill, New York, NY, 1983.

(Sammon Jr., 1969) Sammon Jr., John W. **"A Nonlinear Mapping for Data Structure Analysis"**. IEEE Transactions on Computers. vol. C-18, n.º 5, pg. 401-409, 1969.

(Sarle, 1997) Sarle, Warren S. (editor). **"Neural Network FAQ"**, periodic posting to the Usenet newsgroup comp.ai.neural-nets. URL: ftp://ftp.sas.com/pub/neural/FAQ.html. Recuperado em 08/11/1999.

(Scholtes, 1991a) Scholtes, Johannes C. **"Kohonen Feature Maps in Natural Language Processing"**. CL-1991-01, ITLI Prepublication Series for Computational Linguistics, Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 1991.

(Scholtes, 1991b) Scholtes, Johannes C. **"Neural Nets and Their Relevance for Information Retrieval"**. CL-1991-02, ITLI Prepublication Series for Computational Linguistics, Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 1991.

(Scholtes, 1993) Scholtes, Johannes C. **"Neural Networks in Natural Language Processing and Information Retrieval"**. PhD Thesis, Institute for Logic, Language and Computation (ILLC), University of Amsterdam, 1993.

(Suganthan, 1999) Suganthan, P. N. **"Hierarchical Overlapped SOM's for Pattern Classification"**. IEEE Transactions on Neural Networks, vol. 10, nº 1, pg. 193-196, 1999.

(Svensén, 1998) Svensén, Johan Fredrik Markus. **"GTM: The Generative Topographic Mapping"**. PhD Thesis, Aston University, April 1998. URL: http://neural-server.aston.ac.uk/GTM/thesis.html. Recuperado 24/09/1998.

(Tibshirani, 1992) Tibshirani, Robert. **"Principal Curves Revisited"**. Statistics and Computing, nº 2, pg. 183-190, 1992. URL: http://www-stat.stanford.edu/~tibs/ftp/princcurve.ps. Recuperado em 24/03/2001.

(Tipping & Bishop, 1997) Tipping, Michael E.; Bishop, Christopher M. **"Probabilistic Principal Component Analysers"**. Technical Report , Neural Computing Research Group, Aston University, 1997.

(Titterington *et al.* 1985) Titterington, D. M.; Smith, A. F. M.; Makov, U. E. **"Statistical Analysis of Finite Mixture Distributions"**. John Wiley & Sons, New York, 1985.

(Tukey, 1977) Tukey, J. W. **"Exploratory Data Analysis"**. Addison-Wesley, Reading, MA, 1977.

(Ultsch & Siemon, 1989) Ultsch, A.; Siemon, H. **"Exploratory Data Analysis: Using Kohonen's Networks on Transputers"**. Technical Report 329, University of Dortmund, Dortmund, Germany, 1989.

(Ultsch & Siemon, 1990) Ultsch, A.; Siemon, H. **"Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis"**. In: International Neural Network Conference. Kluwer Academic Press, Paris, pg. 305-308, 1990.

(Ultsch & Vetter, 1994) Ultsch, A.; Vetter C. **"Self-Organizing-Feature-Maps versus Statistical Clustering Methods: A Benchmark"**. Research Report No. 9; Dep. of Mathematics, University of Marburg, 1994. URL: http://www.mathematik.uni-marburg.de/~wina/Papers/94.cluster.ps. Recuperado em 24/02/1999.

(Ultsch, 1999a) Ultsch, A. **"Clustering with DataBots"**. Technical Report 19/99, Philipps University of Marburg, Department of Computer Sciences, June 1999. URL: http://www.mathematik.uni-marburg.de/~wina/Papers/DataBots.ps. Recuperado em 22/09/1999.

(Ultsch, 1999b) Ultsch, A. **"Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series"**. In: Oja, E., Kaski, S. (editors) Kohonen Maps, pg. 33-46, 1999. URL: http://www.mathematik.uni-marburg.de/~wina/Papers/ultsch99.ps. Recuperado em 22/09/1999.

(Van Hulle 2000) Van Hulle, Mark M. **"Faithful Representation and Topographic Maps: From Distortion to Information-Based Self-Organization"**. John Wiley & Sons, 2000.

(Vesanto & Ahola 1999) Vesanto, Juha; Ahola, Jussi. **"Hunting for Correlations in Data Using the Self-Organizing Map"**. In: International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99), Rochester, New York, USA. ICSC Academic Press, pp. 279-285, June 22-25, 1999. URL: http://www.cis.hut.fi/projects/ide/publications/papers/aida99b.zip. Recuperado em 21/06/2000.

(Vesanto & Alhoniemi 2000) Vesanto, Juha; Alhoniemi, Esa. **"Clustering of the Self-Organizing Map"**. IEEE Transactions on Neural Networks, vol. 11, n.º 2, pg. 586-600, 2000.

(Vesanto 1997) Vesanto, Juha. **"Data Mining Techniques Based on the Self-Organizing Map"**. MSc Thesis. Helsinki University of Technology, Espoo, Finland, 1997. URL: http://www.cis.hut.fi/projects/monitor/publications/thesis/mastersJV97.zip. Recuperado em 21/06/2000.

(Vesanto 1999) Vesanto, Juha. **"SOM-Based Data Visualization Methods"**. Intelligent Data Analysis, vol. 3, n.º 2, pg. 111-126, 1999. URL: http://www.cis.hut.fi/projects/ide/publications/papers/ida99.zip. Errata: http://www.cis.hut.fi/projects/ide/publications/papers/ida99errata.zip. Recuperado em 21/06/2000.

(Vesanto 2000) Vesanto, Juha. **"Using SOM in Data Mining"**. Licenciate Thesis. Helsinki University of Technology, Espoo, Finland, 2000. URL: http://www.cis.hut.fi/projects/ide/publications/thesis/licentiateJV2000.zip. Recuperado em 21/06/2000.

(Vesanto *et al*. 1998) Vesanto, Juha; Himberg, Johan; Siponen, Markus, Simula, Olli. **"Enhancing SOM based data visualization"**. In: Proceedings of the International Conference on Soft Computing and Information/Intelligent Systems (IIZUKA'98), Iizuka, Japan, pg. 64-67, October, 1998. URL: http://www.cis.hut.fi/projects/ide/publications/papers/iizuka98.zip. Recuperado em 12/07/2000.

(Vesanto *et al*. 1999) Vesanto, Juha; Himberg, Johan; Alhoniemi, Esa; Parhankangas, Juha. **"Self-Organizing Map in Matlab: the SOM Toolbox"**. In: Proceedings of the Matlab DSP Conference, Espoo, Finland, pg. 35-40, 1999.

(Vesanto *et al*. 2000) Vesanto, Juha; Himberg, Johan; Alhoniemi, Esa; Parhankangas, Juha. **"SOM Toolbox for Matlab 5"**. Technical report A57. Helsinki University of Technology, Finland, 2000. URL: http://www.cis.hut.fi/projects/somtoolbox/package/papers/techrep.pdf. Recuperado em 10/07/2000.

(Visa *et al.* 2000) Visa, Ari; Tovonen, Jarmo; Ruokonen, Piia; Vanharanta, Hannu; Back, Barbro. **"Knowledge Discovery from Text Documents Based on Paragraph Maps"**. In: Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS'33), Maui, Hawaii, January 4, CDROM, 2000.

(Walker *et al.* 1999) Walker A.J.; Cross S.S.; Harrison R.F. **"Visualisation of biomedical datasets by use of growing cell structure networks: a novel diagnostic classification technique"**. Lancet, n.º 354, pg. 1518-1521. URL: http://www.shef.ac.uk/~path/GCSVIS/. Recuperado em 09/03/2000.

(Zadeh, 1965) Zadeh, Lotfi A. **"Fuzzy Sets"**. Information and Control, vol 8, n.º 3, pg. 338-353, 1965.

Bibliografia consultada

(Castro, 1998) Castro, Leandro Nunes. **"Análise e Síntese de Estratégias de Aprendizado para Redes Neurais Artificiais"**. Dissertação de Mestrado. Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica e de Computação, 1998. URL: http://dca.fee.unicamp.br/~lnunes . Recuperado 22/12/1999.

(Chung, 2003) Chung, Lin Yung. **"Lingua-PT-Stemmer-0.01"**. Software PERL livre. URL: http://search.cpan.org/author/XERN/Lingua-PT-Stemmer-0.01/. Recuperado em 05/03/2003.

(De Nicola & Infante, 1997) De Nicola, José; Infante, Ulisses. **"Gramática Contemporânea da Língua Portuguesa"**. Editora Scipione, 1997.

(Kohonen *et al.* 1995b) Kohonen, Teuvo; Hynninen, Jussi; Kangas, Jari; Laaksonen, Jorma; Torkkola, Kari. **"LVQ_PAK. The Learning Vector Quantization Program Package"**. Version 3.1. Helsinki University of Technology, Laboratory of Computer and Information Science, Finland, April 7, 1995. URL: http://www.cis.hut.fi/nnrc/lvq_pak/. Recuperado em 04/01/1999.

(Lipschutz, 1972) Lipschutz, Seymour. **"Teoria e Problemas de Probabilidade"**. 3ª edição revisada. Coleção Schaum, McGraw-Hill, São Paulo, 1972.

(Loos & Fritzke, 1998) Loos, Hartmut S.; Fritzke, Bernd. **"DemoGNG v1.5"**. Institute for Neural Computation
Ruhr-Universität Bochum, 1998. URL: http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/DemoGNGcode.html. Recuperado em 18/01/2000.

(Michaelis, 1998) **"Michaelis: moderno dicionário da língua portuguesa"**. Melhoramentos, 1998.

(Miikkulainen, *in press*) Miikkulainen, Risto. **"Text and Discourse Understanding: The DISCERN System"**. In: R. Dale, H. Moisl and H. Somers (editors), A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text. New York: Marcel Dekker, *in press*. URL: http://www.cs.utexas.edu/users/nn/pages/publications/abstracts.html#miikkulainen.nlp-handbook.ps.Z. Recuperado em 22/12/1999.

(Nabney & Bishop, 1998) Nabney, Ian; Bishop, Christopher. **"Netlab toolbox"**. Neural Computing Research Group,.Division of Electronic Engineering and Computer Science, Aston University Birmingham, United Kingdom, 1998. URL: http://www.ncrg.aston.ac.uk/netlab/. Recuperado 18/01/2000.

(Perez *et al.* 1999) Perez C.A.; Held, C.A.; Mollinger P. **"Handwritten Digit Recognition Using Prototypes Created by Euclidean Distance"**. In: Proceedings of the IEEE International Conference on Information Intelligence and Systems (ICIIS´99), Bethesda, MD, October 31-November 3, pg. 320-323, 1999.

(Press *et al.* 1989) Press, William H.; Flannery, Brian P.; Teukolsky, Saul A.; Vetterling, William T. **"Numerical Recipes in C. The Art of Scientific Computing"**. Cambridge University Press, 1988.

(Rosa, 1993) Rosa, João Luis Garcia. "Redes Neurais e Lógica Formal em Processamento de Linguagem Natural". Dissertação de Mestrado. Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica, 1993.

(Silva, 1995) Silva, Rita Maria da. "Um Sistema Híbrido para Processamento da Linguagem Natural". Tese de Doutorado. Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica. Universite Paul Sabatier (UPS), Institute de Recherche en Informatique de Toulouse. 1995.

(Svensén, 1999) Svensén, Markus. **"The GTM Toolbox v1.01"**. Neural Computing Research Group, Aston University, Birmingham, UK, October 4, 1999. GTM Toolbox URL: http://www.ncrg.aston.ac.uk/GTM/. Recuperado em 02/11/1999.

(Von Zuben, 1996) Von Zuben, Fernando José. “Modelos Paramétricos e Não-Paramétricos de Redes neurais Artificiais e Aplicações”. Tese de Doutorado. Universidade Estadual de Campinas (UNICAMP), Faculdade de Engenharia Elétrica, 1996.

Índice de Citação de Autores

Suganthan, 1999, 64
Svensén, 1998, 8, 29, 34, 35, 70, 75, 76, 78, 80, 82, 83, 87, 88, 92, 93, 94
Svensén, 1999, 20, 21, 22, 83, 147
Tibshirani, 1992, 34
Tipping & Bishop, 1997, 35, 77
Titterington *et al.* 1985, 32
Tukey, 1977, 8, 10
Ultsch & Siemon, 1989, 52, 106
Ultsch & Siemon, 1990, 52
Ultsch & Vetter, 1994, 23, 88
Ultsch, 1999a, 35
Van Hulle, 2000, 18, 38, 39, 40, 41, 55, 57, 65, 66, 73, 75, 79, 80, 81
Vesanto & Ahola, 1999, 105, 124
Vesanto & Alhoniemi, 2000, 66
Vesanto *et al.* 1998, 66
Vesanto *et al.* 1999, 97
Vesanto *et al.* 2000, 51, 53, 73, 74, 97
Vesanto, 1997, 66, 71, 97
Vesanto, 1999, 55
Vesanto, 2000, 55, 97
Visa *et al.* 2000, 163, 167
Walker *et al.* 1999, 60
Zadeh, 1965, 34