Learning and Nonlinear Models - Revista da Sociedade Brasileira de Redes Neurais (SBRN), Vol. 6, No. 2, pp. 81-110, 2008

Uma Comparação Empírica de Métodos de Redução de Dimensionalidade Aplicados a Visualização de Dados

Cláudio J. F. de Medeiros e José Alfredo F. Costa
Departamento de Engenharia Elétrica – Centro de Tecnologia
Universidade Federal do Rio Grande do Norte – 59.072-970 – Natal – RN
E-mails: c.j.franca@ig.com.br, alfredo@ufrnet.br

Resumo

Avanços tecnológicos e redução de custos nos sistemas de aquisição e armazenamento de dados estão oferecendo grandes oportunidades para o desenvolvimento e aplicação de novos métodos de reconhecimento de padrões e mineração de dados. Porém, fatores como tamanho das bases de dados, dimensionalidade, problemas de escalonamento e a necessidade descoberta dos padrões escondidos nas massas de dados acrescentam dificuldades à complexa tarefa de análise de dados. Na maioria dos casos a complexidade do espaço de atributos em tais bases de dados não permite aproximações dedutivas e baseadas em modelos estatísticos tradicionais. Métodos eficientes de redução de dimensionalidade são importantes não apenas para viabilizar a visualização de dados em dimensões adequadas para a percepção humana como também em sistemas automáticos de reconhecimento de padrões, como por exemplo, na eliminação de características redundantes. Este trabalho apresenta comparações qualitativas e quantitativas dos métodos Análise de componentes principais (PCA), projeção de Sammon, redes auto-associativas (RNA-AA), mapas auto-organizáveis (SOM), Isomap e LLE. Todos esses são métodos não-supervisionados de redução de dimensionalidade. Testes foram realizados em bases de dados disponíveis na literatura, *Wine, Syntethic Control* e *Animals*. Particularmente, os testes enfocaram projeções bidimensionais Os resultados retratam dois aspectos das projeções em dimensão reduzida: a qualidade das visualizações gráficas obtidas e a quantificação do grau de fidelidade topológica das projeções. Com relação ao segundo aspecto, os autores propõem, neste artigo, dois índices que buscam quantificar a preservação das vizinhanças nas projeções em baixa dimensão.

Palavras-Chave: Redução de dimensionalidade; Projeções; Visualização; Mineração de dados; Sistemas Adaptativos.

1. Introdução

Os avanços tecnológicos nos sistemas de aquisição e armazenamento de dados, aliados à queda dos custos dos dispositivos, vêm oferecendo grandes oportunidades para o desenvolvimento e aplicação de novos métodos de reconhecimento de padrões e mineração de dados. Exemplos incluem sistemas em áreas como análise de dados geoespaciais, bioinformática, organização e recuperação de imagens baseada em conteúdo, bases de dados distribuídas na internet, redes de sensores, etc. Sistemas dessa natureza em geral produzem grandes bases de dados cuja exploração apresenta grandes desafios.

A complexidade do espaço de atributos em tais bases de dados e os problemas computacionais daí derivados quase sempre não permitem aproximações dedutivas e baseadas em modelos estatísticos tradicionais [17]. Fatores como tamanho das bases de dados, dimensionalidade, problemas de escalonamento e a necessidade de descoberta dos padrões escondidos nas massas de dados contribuem para tornar mais complexa a tarefa de análise de dados.

A visualização é um recurso extremamente útil para a compreensão da estrutura de dados, em processos de descoberta de conhecimento e mineração de dados, tanto em aplicações científicas quanto comerciais. Particularmente na fase de análise exploratória, a visualização permite uma percepção intuitiva da estrutura dos dados, fornecendo subsídios para a seleção das ferramentas mais adequadas a serem utilizadas no processo de mineração de dados. Pode, inclusive, fornecer indícios importantes para a escolha dos parâmetros de sintonia dessas ferramentas, como por exemplo, em algoritmos de agrupamentos de dados. Na visualização de dados multidimensionais, podem ser utilizadas coordenadas espaciais, cores, formas dos objetos, entre outras propriedades, para codificar diferentes dimensões. Porém, a maneira mais usual é através de projeções para espaços 2D ou 3D que permitem análise perceptual das massas de dados.

A redução da dimensionalidade é uma operação fundamental para viabilizar a visualização de dados multidimensionais e baseia-se na aplicação de transformações sobre os dados projetando-os em espaços de menor dimensão com máxima manutenção da topologia, i.e., relações de vizinhança entre os dados. Técnicas de redução de dimensionalidade (RD) objetivam reduzir o espaço de características preservando ao máximo as relações topológicas dos dados. [1].

Este artigo apresenta comparações qualitativas e quantitativas entre seis métodos utilizados para RD: Análise de Componentes Principais (PCA) [2]; Projeção de Sammon [3]; Rede Neural MLP auto-associativa [4]; Redes Auto-organizáveis de Kohonen (SOM) [5]; Isomap [6] e LLE (*Locally Linear Embedding*) [7]. Os dois últimos métodos são baseados em preservação de distâncias sobre uma hiper-superfície que captura a dimensão intrínseca do conjunto de dados (*manifold*). Com exceção do primeiro (PCA), todos os outros métodos são não-lineares. São apresentados os resultados da aplicação dos diversos métodos sobre as bases de dados *Wine, Syntethic Control* e *Animals* [11]. Para as comparações quantitativas são propostos, neste artigo, dois índices que buscam quantificar o grau de preservação da vizinhança dos dados projetados, em relação aos dados originais. A subseção 3.2 apresenta a definição desses dois índices.

O restante do artigo é composto das seguintes seções: a seção 2 introduz o conceito de redução da dimensionalidade, cita trabalhos relacionados e revisa de forma sucinta os algoritmos escolhidos enquanto que a seção 3 descreve a metodologia empregada para os testes. A seção 4 descreve as bases de dados utilizadas e os parâmetros de configuração dos diversos algoritmos. Resultados são apresentados na seção 5 e a seção 6 apresenta as conclusões do artigo.

2. Métodos de redução de dimensionalidade

O problema da redução da dimensionalidade pode ser formulado da seguinte maneira: seja $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n\}$ um conjunto de *n* pontos (vetores) de um espaço *d*-dimensional, ou seja, $\mathbf{x}_i \in R^m$. A redução de dimensionalidade busca encontrar um conjunto correspondente de pontos de saída $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ...\mathbf{y}_n\}$, tal que $\mathbf{y}_i \in R^p$, sendo $p < m$ e $\mathbf{Y}$, a representação mais fiel possível de $\mathbf{X}$ no espaço de baixa dimensão.

A redução de dimensionalidade pode ser vista como a aplicação de uma transformação sobre dados residindo num espaço de alta dimensão, para projetá-los numa baixa dimensão, de tal forma que a transformação assegure a máxima preservação possível da topologia do espaço de dados original.

As pesquisas na área de redução de dimensionalidade têm produzido uma rica variedade de métodos e algoritmos. Nos últimos anos, essas pesquisas têm recebido um impulso cada vez maior, em virtude do aumento constante do volume de informação produzido e da facilidade de acesso a essa informação.

Devido à grande diversidade de métodos atualmente disponíveis, é útil procurar estabelecer formas de comparar a sua funcionalidade em condições de trabalho semelhantes, buscando conhecer suas virtudes e deficiências em determinados tipos de aplicações. O objetivo deste trabalho é realizar um estudo comparativo de métodos não supervisionados de redução de dimensionalidade.

2.1. Trabalhos relacionados

Há alguns trabalhos relacionados à comparação de métodos de redução de dimensionalidade (RD). Em 1997, Balachander *et al* [27] apresentaram um estudo comparativo de quatro métodos de RD, três deles não supervisionados (PCA, SOM e rede MLP-AA) aplicados à classificação de padrões. Os resultados foram aferidos de forma indireta, pela taxa de acerto da classificação binária de amostras de uma análise citológica.

Em 1998, Backer *et al* [28] realizaram comparações entre quatro métodos não supervisionados (MDS clássico, projeção de Sammon, SOM e rede MLP-AA) também voltadas para tarefas de classificação. Os testes utilizaram três bases de dados (uma artificial e duas bases de texturas) e novamente os resultados se basearam no percentual de acertos de um classificador *k-NN* (*k* vizinhos mais próximos) aplicado aos dados reduzidos.

Em 2007, Yin [10] apresentou uma revisão geral dos métodos de redução de dimensionalidade e realizou uma comparação especificamente entre seis deles: PCA, Sammon, SOM, Isomap, LLE e ViSOM (uma variante do SOM). Os experimentos constituíam-se em projetar um conjunto de dados artificial de três para duas dimensões. Os dados tridimensionais de entrada estavam distribuídos uniformemente num manifold contínuo em forma de "S". As comparações basearam-se nas visualizações gráficas dos dados bidimensionais de saída.

Em 2008, van der Maaten *et al* [9] apresentaram um relatório de uma análise comparativa que incluiu doze métodos de redução de dimensionalidade, aplicando-os a cinco bases artificiais e cinco bases naturais. Mais uma vez, a avaliação baseou-se nos erros de generalização de um classificador *k-NN,* aplicado às projeções fornecidas por cada um dos métodos. Os autores não incluíram, nesse estudo, o algoritmo SOM nem qualquer das suas variantes, como GTM, G-SOM, E-SOM, etc., por não considerá-lo propriamente um método de redução de dimensionalidade.

O presente trabalho procura uma comparação empírica entre métodos representativos, aplicando-os a bases de dados naturais (compostas de dados do mundo real ou de dados que os simulem). Embora técnicas de RD sejam utilizadas para diversas finalidades, principalmente como um estágio de pré-processamento para outros algoritmos, este trabalho está voltado para o uso de RD com a finalidade de obter visualizações de dados. Essas visualizações são usadas tipicamente em análise exploratória de dados, para descobrir indícios sobre a sua estrutura. Nesse tipo de aplicação, a análise preliminar baseada em gráficos é importante, apesar da presença de fatores subjetivos. Portanto, neste trabalho o interesse se concentra, particularmente, nas projeções em duas dimensões que podem ser visualizadas diretamente em gráficos bidimensionais. A idéia é comparar os métodos, tanto por meio de fatores quantitativos que busquem avaliar a preservação das vizinhanças entre pontos do espaço original, quanto por meio de fatores qualitativos ligados à informação visual disponibilizada pelos gráficos. Por outro lado, ao contrário dos trabalhos citados que avaliavam indiretamente a qualidade das projeções, por meio de medidas dos resultados das aplicações específicas (qualidade de agrupamentos, precisão de classificadores, etc.,), neste trabalho a avaliação quantitativa procurou utilizar índices que digam respeito mais propriamente às projeções obtidas. Para tanto, os autores propõem, neste artigo, a definição e utilização de dois índices, denominados **índices de coincidência ordenada e não ordenada das vizinhanças**. Com esses índices, procura-se estimar, de uma maneira relativa, a capacidade das projeções refletirem as relações de vizinhança dos pontos do espaço original (seção 3).

Para este trabalho, buscou-se selecionar métodos representativos que apresentassem características como: *(1)* gozarem de amplo reconhecimento por parte da comunidade científica; *(2)* serem métodos já consagrados na literatura; *(3)* serem já provados por boa quantidade de aplicações e experimentos; *(4)* cobrirem uma gama que comporta métodos clássicos, métodos mais modernos e métodos relativamente recentes. Entre os métodos que atendem a esses critérios foram selecionados os seguintes: dois métodos clássicos e bastante antigos (PCA e Sammon), dois métodos conexionistas clássicos (SOM e redes MLP autoassociativas) e dois métodos relativamente recentes (Isomap e LLE).

As subseções a seguir apresentam uma revisão sucinta de cada um desses métodos.

2.2. Projeção de Sammon

Escalonamento Multidimensional (MDS) é um dos métodos tradicionais de redução de dimensionalidade. MDS é uma designação para um conjunto de técnicas que usam como entrada, medidas de proximidade entre os objetos que formam um espaço de entrada multidimensional. Nesse contexto, proximidade é uma grandeza que traduz quão similares ou diferentes dois objetos são (ou são percebidos como tal). O resultado obtido é uma representação espacial, consistindo de uma configuração geométrica de pontos, cada ponto representando um dos objetos, com a distância euclidiana entre os pontos tendendo a representar as dissimilaridades entre os objetos no espaço original [23].

Dados um conjunto de n objetos num espaço m-dimensional e uma matriz n x n com as medidas de dissimilaridade entre os pares de objetos o método MDS produz uma representação p-dimensional ($p < m$) dos objetos de tal forma que as distâncias entre os pontos do espaço projetado reflitam o mais próximo possível as dissimilaridades entre esses objetos.

Tipicamente, caso do MDS métrico, as dissimilaridades são representadas por medidas de distância: quanto maior a similaridade entre dois objetos, menor será a distância entre eles [3]. Então, dado um conjunto m-dimensional de objetos $\{\mathbf{x}_i\}$ e uma matriz simétrica de dissimilaridades $\mathbf{D} = \{ \delta_{ij} , i,j = 1,....,n\}$, onde $\delta_{ij} = \delta (\mathbf{x}_i, \mathbf{x}_j)$ é a distância entre dois pontos do conjunto de entrada, o método irá produzir um conjunto $\{\mathbf{y}_{ij}\}$ de pontos no espaço projetado de forma que as distâncias $d_{ij} = d(\mathbf{y}_i,\mathbf{y}_j)$ sejam o mais próximo possível do valor das correspondentes distância δ_{ij}. Para isso, um algoritmo de otimização é aplicado, de forma a reduzir ao mínimo possível uma função objetivo associada à diferença relativa entre os pares de objetos (δ_{ij}) e entre as suas representações respectivas no plano projetado (d_{ij}). A minimização da função de custo, normalmente contínua e monotônica, pode ser obtida por meio de vários métodos diferentes, tais como a decomposição da matriz de dissimilaridades baseada nos autovalores, gradiente conjugado ou método pseudo-Newton [3].

Em 1969, Sammon [13] propôs uma abordagem baseada numa variante da função de custo e otimização por meio de mínimos quadrados. A função de custo de Sammon é dada por:

$$S = \frac{1}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{(d_{ij} - d_{ij})^2}{d_{ij}}. \quad (1)$$

A função de custo de Sammon possui, como característica, uma ênfase relativamente maior na preservação das menores distâncias [23]. Essa função de erro pode ser minimizada, usando qualquer técnica de otimização. Sammon [13] aplicou a minimização pelo método *pseudo-Newton*. Nesse caso, as componentes de cada ponto no plano projetado são obtidas iterativamente pela equação:

$$y_{i_k}^{(t+1)} = y_{i_k}^{(t)} - \alpha \Delta_{i_k}, \quad (2)$$

onde, α é um fator de convergência determinado empiricamente (geralmente recomenda-se um valor entre 0.3 e 0.4 [3]), $y_{i_k}^{(t)}$ representa a k-ésima coordenada do vetor $\mathbf{y}_i$ (na iteração t) e Δ_{i_k} é dado por:

$$\Delta_{i_k} = \frac{\frac{\partial S}{\partial y_{i_k}}}{\left|\frac{\partial^2 S}{\partial y_{i_k}^2}\right|}. \quad (3)$$

As derivadas parciais da equação (3) são dadas por [14, 3]:

$$\frac{\partial S}{\partial y_{i_k}} = \frac{-2}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{d_{ij} - d_{ij}}{d_{ij} d_{ij}} (y_{i_k} - y_{j_k}) \quad (4)$$

e

$$\frac{\partial^2 S}{\partial y_{i_k}^2} = \frac{-2}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{1}{d_{ij} - d_{ij}} \left[(d_{ij} - d_{ij}) - \frac{(y_{i_k} - y_{j_k})^2}{d_{ij}} \left(1 + \frac{d_{ij} - d_{ij}}{d_{ij} d_{ij}}\right)\right]. \quad (5)$$

O mapeamento não linear dos dados de entrada é obtido pela atualização de cada coordenada projetada, usando as equações (2) e (3) num processo iterativo até que um valor limite de convergência seja atingido.

Essa técnica tornou-se conhecida como “mapeamento de Sammon” e é muito utilizada para visualização bidimensional de dados multivariados.

2.3. PCA

O método de Análise de Componentes Principais (PCA) (Pearson [37]; Jolliffe [2]) é o mais antigo e, na opinião da maioria dos autores, o método basilar dentre todas as pesquisas de redução de dimensionalidade**.** É um método linear de extração de atributos. Baseia-se em uma transformação linear sobre os dados, obtendo projeções sobre um novo sistema de coordenadas ortogonais que permitem representar a maior variância possível dos dados, numa seqüência decrescente do percentual dessa variância, correspondendo à ordem desses componentes.

De certa forma, PCA inspirou ou tornou-se a referência para os demais métodos nessa área. Isso se refere não apenas aos métodos lineares, grupo do qual faz parte, mas também à maior parte dos métodos não lineares.

Apesar de ser um método clássico, PCA ainda é, provavelmente, o método de RD mais utilizado na prática, particularmente em análise exploratória e redução de dimensionalidade. Isso pode ser atribuído a várias características bastante positivas: sólida base teórica, propriedades analíticas, simplicidade do conceito, não dependência de parâmetros de configuração, estabilidade e existência de vários métodos numéricos que permitem implementá-lo com eficiência.

O método aplica uma transformação linear sobre um conjunto *m*-dimensional de dados de entrada e encontra um novo sistema de coordenadas para representar esse conjunto numa forma mais adequada à análise dos dados. Utilizando os autovetores correspondentes aos autovalores da matriz de covariância dos dados, em seqüência descendente, obtém-se a projeção de maior variância possível do conjunto de dados de entrada, ou primeiro componente principal. Em seguida, é obtida a projeção com a segunda maior retenção de variância e que seja ortogonal à primeira projeção (segundo componente principal). E assim, sucessivamente, são obtidas todos os *m* componentes ou eixos principais.

Para uma visão geral do método, consideremos uma conjunto de amostras $\mathbf{X}' = \{\mathbf{x}_1', ..., \mathbf{x}'_n\}$ com $\mathbf{x}_i' \in R^d$, $i = 1,....,n$. A média das amostras é $\mathbf{x}_m$. A centralização de $\mathbf{X}'$ em torno da origem, fornece a matriz de dados $\mathbf{X} = \{ \mathbf{x}_1, ..., \mathbf{x}_n\}$, com $\mathbf{x}_i = \mathbf{x'}_i - \mathbf{x}_m$, $i = 1,....,n$. A matriz de covariâncias empírica de $\mathbf{X}$ é dada por:

$$\mathbf{\Sigma} = 1/n\ (\mathbf{X}\mathbf{X}^T). \tag{6}$$

A matriz $\mathbf{\Sigma}$ é simétrica, definida semipositiva. Portanto, pelo teorema da decomposição espectral, ela pode ser expressa como [14]:

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \tag{7}$$

sendo:

$\mathbf{U} = \{\mathbf{u}_1, ..., \mathbf{u}_m\}$ ortogonal;
$\mathbf{\Lambda} = \{\lambda_1, ..., \lambda_m\}$ diagonal;
λ_k cada um dos autovalores de $\mathbf{\Sigma}$, $\lambda_k \geq 0$, $k = 1,....,m$;
$\mathbf{u_k}$ o autovetor normalizado de $\mathbf{\Sigma}$ associado ao autovalor λ_k.

A transformação do componente principal

$$\mathbf{Y} = \mathbf{U}^T \mathbf{X} \tag{8}$$

fornece um sistema de referência no qual a matriz $\mathbf{Y}$ tem média zero e matriz de covariância diagonal $\mathbf{\Lambda}$, contendo os autovalores de $\mathbf{\Sigma}$ (variáveis não correlacionadas).

A matriz $\mathbf{Y}$ contém as componentes dos vetores $\mathbf{y}_i$, obtidas pela rotação de $\mathbf{X}$, segundo uma nova base ortogonal e segundo eixos que capturam, cada um por vez, o máximo possível da variância contida no conjunto $\mathbf{X}$, em ordem decrescente. Essa rotação dos eixos, por meio da operação do PCA, é ilustrada na figura 1, mostrando um conjunto de dados de duas dimensões.

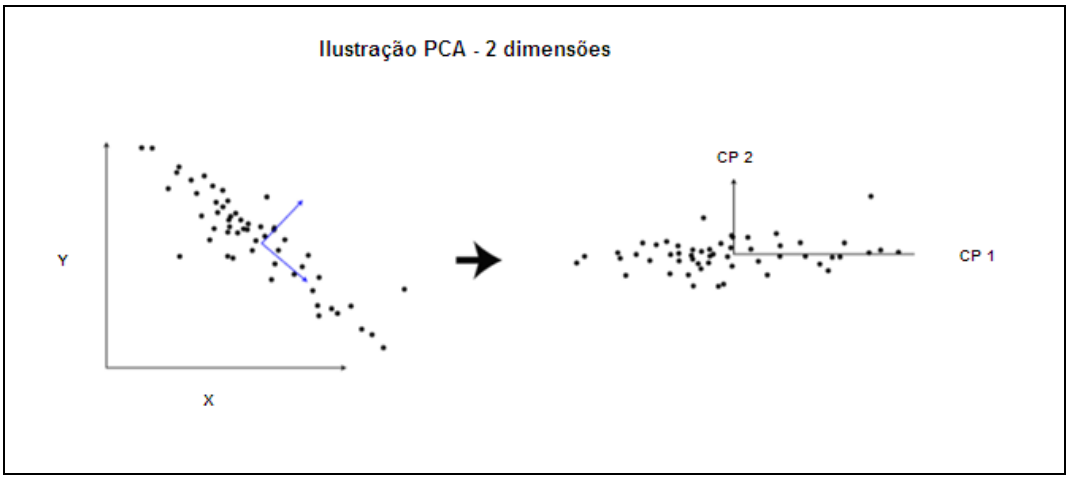


Figura 1 – Ilustração da operação PCA, mostrando a rotação típica dos eixos, segundo os componentes principais (CP1 e CP2) e o efeito de centralização em relação à origem dos eixos. Adaptado de [49].

Nesse novo sistema de coordenadas, é possível descartar variáveis com baixa variância, ou seja, projetar (com a melhor aproximação possível) os dados do conjunto original **X**, no subespaço gerado pelos primeiros *p* componentes principais:

$$\mathbf{Y} = \mathbf{U}_p^T \mathbf{X}, \text{ com } \mathbf{U}_p = \{\mathbf{u}_1,.....,\mathbf{u}_p\}. \quad (9)$$

Essa operação representa um truncamento de componentes de menor importância, usado para obter a redução de uma dimensão original *m* para uma dimensão reduzida *p* ($p<m$). Para tanto, selecionam-se os *p* primeiros dentre os *m* componentes principais de um determinado conjunto de dados e ignoram-se os demais. Com isso, se preserva o máximo possível da variabilidade dos dados originais, dentro dos limites dessa dimensão reduzida.

O algoritmo PCA é sumarizado a seguir:

Algoritmo PCA

Entrada: matriz *n* x *m* dos dados de entrada **X´**.
Saída: matriz *n* x *p* dos dados de saída **Y.**

Passo 1: Calcular as médias das colunas (dimensões) de **X´**, formando o vetor de médias $\mathbf{x}_m$.
Passo 2: Subtrair o vetor $\mathbf{x}_m$ de cada linha de **X´,** formando a matriz **X**.
Passo 3: Calcular a matriz de covariância empírica **Σ** de **X** (obtida no passo 2).
Passo 4: Computar e classificar por ordem decrescente dos autovalores, os autovetores $\mathbf{u}_i$ de **Σ**, formando a matriz **U** dos autovetores.
Passo 5: Selecionar os *p* primeiros autovetores em **U,** formando $\mathbf{U}_p$.
Passo 6: Calcular a matriz *n* x *p* dos dados de saída $\mathbf{Y} = \mathbf{U}_p^T\mathbf{X}$.

A propriedade básica da projeção PCA é produzir um mapeamento $\mathbf{X} \in R^m \rightarrow \mathbf{Y} \in R^p$ que fornece a redução de dimensionalidade linear ótima, no seguinte sentido: *(1)* menor soma dos erros mínimos quadráticos dos dados reconstruídos [29]; *(2)* máxima variância no espaço projetado, condicionada à ortonormalidade [30].

O método PCA possui grande aceitação por uma série de características extremamente positivas. Entretanto, por se basear em uma transformação linear, o método pode apresentar limitações, quando aplicado a problemas envolvendo dados imersos em *manifolds* complexos. As redes neurais auto-associativas, mostradas na seção seguinte, implementam uma espécie de generalização não-linear do método PCA.

2.4. Redes Neurais MLP Auto-associativas

O modelo de rede neural Perceptron de Múltiplas Camadas (MLP) tem sido largamente utilizado em uma diversidade de mapeamentos e problemas de classificação e reconhecimento de padrões.

Redes MLP são compostas de múltiplas camadas de neurônios, completamente conectadas. Tipicamente apresentam fluxo de dados para frente (*feed-forward*) e utilizam algoritmo de treinamento de retro-propagação (*back-propagation*) da diferença entre a resposta da rede e a resposta desejada para uma determinada entrada. O treinamento da rede é baseado na apresentação sucessiva de um conjunto de dados de treinamento, num processo iterativo, buscando minimizar uma função de custo apropriada, sendo a mais usual, o erro médio quadrático dos valores de saída em relação aos valores esperados correspondentes. Após vários ciclos de treinamento, ao ser atingido um valor limite de um critério de parada, como o valor máximo admissível para a função de custo ou o número máximo de épocas (iterações) obtêm-se os valores finais para os pesos sinápticos da rede ou, em outras palavras, a rede está devidamente treinada. Depois disso, os parâmetros da rede são fixados e ela poderá ser utilizada para processar amostras específicas dos dados, mesmo aquelas não pertencentes ao conjunto usado para o treinamento.

Ao longo do tempo, as redes MLP têm sido utilizadas com sucesso em diversas aplicações importantes, como classificação de padrões, interpolação de funções, otimização, predição e controle [12]. Além do algoritmo de retro propagação, diversos outros algoritmos têm sido desenvolvidos com o objetivo de determinar, com a melhor eficiência e precisão possíveis, os pesos sinápticos adequados para cada aplicação.

Num trabalho de 1989 [19], Cybenko demonstrou que funções da forma mostrada a seguir são capazes de aproximar qualquer função não linear *f(x)*, com nível de precisão arbitrário:

$$f(x) = \sum_{i=1}^{p} w_{1i}^{(2)} \sigma\left(\sum_{j=1}^{n} w_{ij}^{(1)} x_j + \theta_j \right), \qquad (10)$$

onde, $\sigma(x)$ é qualquer função contínua, monotonicamente crescente com $\sigma(x) \rightarrow 1$ quando $x \rightarrow \infty$ e $\sigma(x) \rightarrow 0$ quando $x \rightarrow -\infty$; $w_{ij}^{(k)}$ é o peso entre o *i*-ésimo nó da camada $k+1$ e o *j*-ésimo nó da camada *k*.

Conforme observado por Kramer [4] a equação anterior pode expressar justamente o funcionamento de uma rede MLP com fluxo para frente, com *n* entradas, uma camada escondida contendo *p* neurônios com funções de ativação sigmoidais e saída linear. Portanto, um arranjo dessa forma caracteriza um aproximador universal de funções lineares ou não lineares.

Essa capacidade de aproximar funções não lineares fornece a idéia de uma arquitetura de rede MLP formada pela combinação de dois estágios com as características anteriormente descritas, dispostos numa geometria conforme a da figura 2, a seguir:

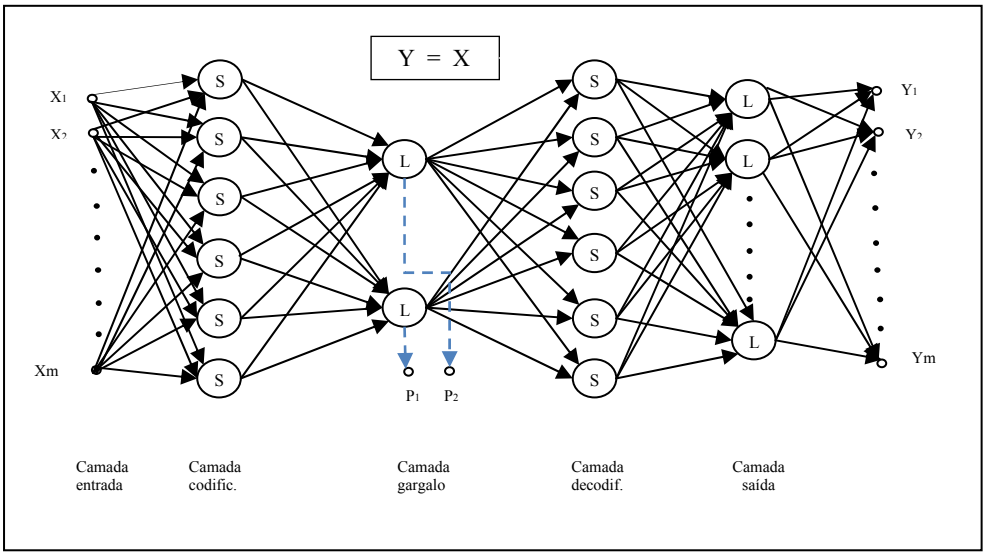


Figura 2: Esquemático da arquitetura de uma rede MLP auto-associativa. As saídas são treinadas para igualar as entradas. As saídas p_1 e p_2 fornecem as coordenadas de **x** no espaço projetado (representado como bi-dimensional, no esquema). Os neurônios identificados com "S" possuem função de ativação sigmoidal, os marcados com "L" possuem geralmente função linear.

Arquiteturas como a da figura 2, quando sujeitas à condição de treinamento que estabelece que os sinais desejados de saída sejam idênticos aos sinais de entrada, representam uma classe particular de redes MLP denominadas redes MLP auto-associativas (ou redes-gargalo) [4]. A condição da saída refletindo o padrão da entrada é uma característica que, de certa forma, modifica o caráter de treinamento supervisionado típico das redes MLP. A forma de treinamento adotada elimina a necessidade de um "tutor" externo para ensinar as saídas corretas, o que motiva a designação de "auto-associador". Essa forma de treinamento também leva vários autores a designarem tais tipos de rede como sendo de treinamento "auto-supervisionado" ou "semi-supervisionado".

A arquitetura desse tipo de rede requer, pelo menos, uma camada de entrada com número de neurônios igual à dimensão dos dados de entrada, uma camada de saída com o mesmo número de neurônios da camada de entrada e uma camada escondida central, normalmente denominada camada "gargalo", contendo menor número de neurônios que as camadas anteriores. Essa rede pode ser vista como possuindo dois estágios: o primeiro estágio que engloba da camada de entrada até a camada gargalo é o codificador; o segundo, da camada gargalo até a camada de saída, é o decodificador.

Uma rede auto-associativa em três camadas com uma camada escondida menor do que as camadas de entrada/saída e contendo apenas funções de ativação lineares é equivalente ao PCA [18]. No entanto, conforme exposto em parágrafos anteriores, é mais interessante a introdução de funções de ativação não lineares nas camadas escondidas de forma a permitir um processamento não linear dos dados de entrada. Um auto-associador com funções de ativação sigmoidais nas camadas internas representa uma implementação do que na literatura científica é considerada uma versão não linear do PCA, normalmente designada pela sigla NLPCA.

Por outro lado, por uma série de considerações, Kramer concluiu que no caso mais geral é essencial haver três camadas escondidas, para se obter uma ótima extração não linear de características [4]. Isso advém, basicamente, do fato de que cada um dos estágios codificador/decodificador requer três camadas para realizar adequadamente uma função não linear. Como a rede auto-associativa combina os dois estágios num único conjunto, a camada de saída do primeiro estágio funde-se com a camada de entrada do segundo, formando uma terceira camada escondida, justamente a camada gargalo. Disso resulta uma arquitetura com cinco camadas (uma de entrada, uma de saída e três escondidas) conforme aparece na figura 2.

Uma rede auto-associativa em cinco camadas, apresentando ativações não-lineares na segunda e quarta camadas, permite aproximar qualquer função de compressão de R^m em R^p, sendo *m* o número de neurônios das camadas de entrada/saída e *p* o da camada gargalo [19, 20]. A camada central (ou camada gargalo) é a saída do codificador e fornece os dados em forma comprimida com dimensão de saída igual ao número dos neurônios dessa camada (ver saídas em azul na figura 2). É justamente a possibilidade de extrair os sinais de saída da camada gargalo que permite a utilização da rede auto-associativa como um método de redução de dimensionalidade baseado em extração de características do conjunto de dados com alta dimensão na entrada.

Uma característica notável é que a presença dos estágios codificador/decodificador permite a reconstrução do espaço de entrada a partir dos dados comprimidos. Outra característica importante é que a rede treinada estabelece uma função de compressão implícita que suporta suavemente a inclusão de novos padrões de entrada, o que permitiria, por exemplo, a sua utilização em processamento *on-line.*

2.5. Mapa Auto-organizável de Kohonen (SOM)

O SOM define um mapeamento de um espaço multidimensional contínuo para um conjunto finito de vetores-referência, ou neurônios, dispostos na forma de um arranjo espacial regular, normalmente bidimensional. A dimensionalidade desse arranjo espacial (reticulado) representa a dimensão reduzida de saída do algoritmo. O objetivo principal do treinamento é reduzir a dimensionalidade do espaço de entrada ao mesmo tempo em que se busca preservar ao máximo a topologia [5].

Cada neurônio *i* é representado por um vetor de pesos ($\mathbf{m}_i$) com a mesma dimensão dos vetores de entrada. Para cada padrão de entrada um neurônio é escolhido o vencedor, *c*, usando o critério de maior similaridade. Os pesos do neurônio vencedor, bem como os pesos dos neurônios compreendidos em sua vizinhança N_c, são atualizados de acordo com a equação

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \tag{11}$$

onde *t* indica a iteração, $\mathbf{x}(t)$ é o padrão de entrada fornecido de forma aleatória na iteração e $h_{ci}(t)$ inclui dois fatores: o núcleo de vizinhança em torno do neurônio vencedor e a constante de aprendizado. Ambos decaem com o tempo, de acordo com funções previamente determinadas.

O algoritmo de aprendizado do SOM é resumido a seguir.

Algoritmo SOM

Entrada:Conjunto de dados **X**
Saída: Conjunto de protótipos do Mapa SOM treinado.

Passo1: Inicializar aleatoriamente os pesos $\mathbf{m}_i(t)$.
Inicializar os valores da vizinhança, $\gamma(t)$ e da taxa de aprendizado $\alpha(t)$.

Passo2: Rearranjar aleatoriamente os vetores **x** do conjunto de entrada **X**.

Para cada vetor **x** de entrada

Encontrar o neurônio mais similar a **x** (neurônio vencedor $\mathbf{m}_c$) de acordo com:

$$\|\mathbf{x} - \mathbf{m}_c(t)\| = min_i \|\mathbf{x} - \mathbf{m}_i(t)\|$$

Ajustar os pesos do neurônio vencedor e dos seus vizinhos conforme:

$$m_i(t+1) = \begin{cases} m_i(t) + h_{ci}(t)[\mathbf{x} - m_i(t)], & i \in N_c(t) \\ m_i(t), & c.c. \end{cases}$$

Voltar para o **passo 2** até que todos os vetores tenham sido apresentados. Então seguir para o **passo 3**.

Passo 3: Decrementar a taxa de aprendizado e o tamanho da vizinhança e voltar para o **passo 1,** até que o critério de convergência seja alcançado e então o processo de treinamento termina.

Uma variação do algoritmo anterior é o algoritmo em lote do SOM que o torna insensível à seqüência de apresentação dos dados, em cada época. As contribuições de cada padrão são acumuladas e, ao final de cada época, é feita a atualização dos pesos [5].

Uma característica importante do SOM, derivada da quantização vetorial gerada pelo algoritmo, é que a densidade dos neurônios em um mapa treinado é uma aproximação da densidade dos dados [15]. Assim, é possível obter informações dos agrupamentos analisando as relações geométricas dos neurônios após o treinamento.

A saída de um SOM, para um dado padrão de entrada, é geralmente o índice do neurônio vencedor *c*, no caso bidimensional um par de valores (*i, j*), e o nível de ativação diretamente relacionado à quantização (a distância do padrão ao neurônio *c*, computado no espaço de pesos) [48].

A simples informação das coordenadas no espaço de saída não favorece a visualização do mapa, pois a informação de distância (dissimilaridade) entre os pesos dos neurônios não é perceptível. Para contornar essa dificuldade, um método de visualização de um SOM treinado, denominado a matriz de distâncias unificadas, ou *U-matrix*, foi desenvolvido por A. Ultsch [18] com o objetivo de permitir a detecção visual das relações topológicas dos neurônios [48]. A idéia básica é usar a mesma métrica que foi utilizada durante o treinamento, para calcular distâncias entre pesos sinápticos de neurônios adjacentes. O resultado é uma imagem $f(x, y)$, na qual as coordenadas de cada pixel (x, y) são derivadas das coordenadas dos neurônios no *grid* do mapa, e a intensidade de cada *pixel* na imagem $f(x, y)$ corresponde a uma distância calculada. Pode-se imaginar uma imagem como uma função tridimensional em que o valor do pixel na coordenada (x, y) é representado por um ponto na coordenada z. Nesse caso, teríamos uma superfície em 3D cuja topografia revela a configuração dos neurônios obtida pelo treinamento. Vales, neste relevo topográfico, correspondem a regiões de neurônios que são similares, enquanto que montanhas, i.e., valores relativamente elevados na *U-matrix*, refletem a dissimilaridade entre neurônios vizinhos e podem ser associadas a regiões ou neurônios em fronteiras de agrupamentos [48].

A figura 3 mostra um exemplo de projeção de uma *U-matrix*, obtida de um determinado mapa treinado, acompanhada da interpretação tri-dimensional dessa mesma projeção, evidenciando as regiões de alta densidade (vales) e os picos representando as regiões de fronteira no mapa.

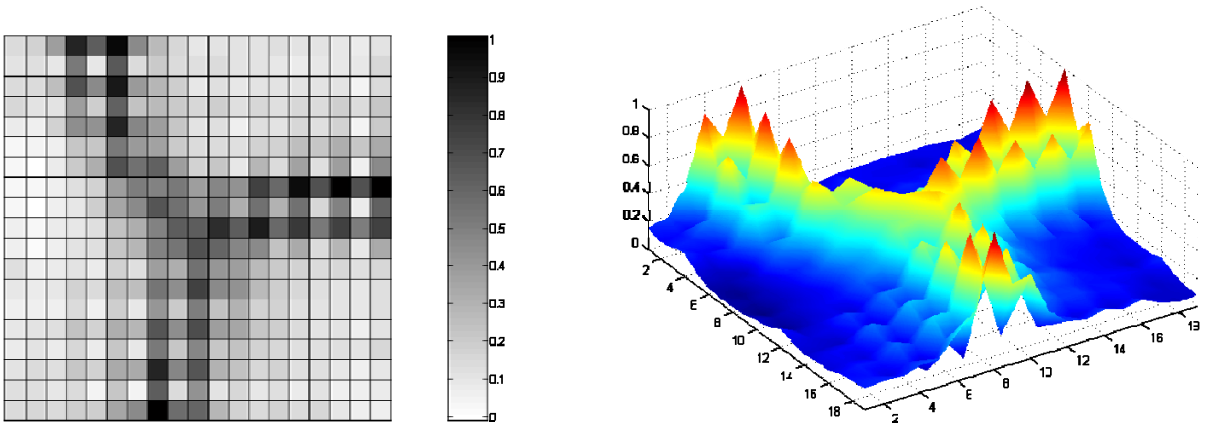


Figura 3: Exemplo de projeção bi-dimensional de uma *U-matrix* (esquerda) e sua interpretação espacial (direita). Adaptado de [48].

A redução de dimensionalidade intrínseca, onde dados são projetados em um display discreto bidimensional de neurônios, aliada a uma aproximada fidelidade topológica (obtida via aprendizado não supervisionado), tem feito do SOM uma ferramenta atraente para tarefas de visualização e *clustering*. Métodos de análise automática da *U-matrix* com objetivo de segmentar automaticamente o SOM podem ser vistos em várias referências do autor [38 a 45]. Extensões para modelos hierárquicos podem ser vistas em [46, 47]

2.6. Isomap

O algoritmo Isomap, proposto por Tenenbaum *et al* [6], pode ser visto como uma extensão do método MDS, em que a medida de distância inter-pontos é realizada de uma maneira mais sofisticada do que a aplicação da distância euclidiana, de forma a capturar as não-linearidades do *manifold* imerso no espaço de alta dimensão original.

A idéia é utilizar, como medida, uma aproximação razoável da distância geodésica entre os pontos de entrada. A distância geodésica entre dois pontos é definida como o comprimento da menor curva que une esses dois pontos e está contida na hiper-superfície formada pelos pontos de entrada (*manifold*). No método Isomap, considera-se que, para pontos vizinhos num espaço de entrada, a distância geodésica pode ser aproximada pela própria distância direta entre esses pontos. No caso dos pontos distantes, a distância geodésica é aproximada por uma soma de pequenos passos intermediários, correspondentes às distâncias diretas entre os pontos vizinhos que compõem o menor caminho entre os pontos distantes considerados, através do grafo de vizinhos conectados.

Inicialmente, dado um conjunto **X** de pontos de entrada, é determinada a matriz de distâncias $\mathbf{D} = \{\delta_{ij}\ ,\ i, j = 1,....,n\}$, onde $\delta_{ij} = \delta\ (\mathbf{x}_i,\ \mathbf{x}_j)$ é a distância entre dois pontos do conjunto de entrada. Baseando-se nas distâncias δ_{ij}, define-se a vizinhança de cada ponto $\mathbf{x}_i$, encontrando, por exemplo, os seus k vizinhos mais próximos.

Em seguida, é construído um grafo G de todos os pontos do conjunto **X**, por meio da interligação de cada ponto aos seus vizinhos, através de ramos de peso δ_{ij}. O algoritmo estima a distância geodésica entre cada par de pontos, calculando o menor caminho $d_G(i,j)$ entre esses pontos, no grafo G. Finalmente, é aplicado o conceito clássico do algoritmo MDS para obter uma projeção cartesiana de dimensão p, do espaço de entrada de dimensão m. Para tanto, minimiza-se a função clássica do MDS,

$$S = \frac{\sum_{i,j}(\delta_{ij} - d_{ij})}{\sum_{i,j} d_{ij}^2}, \qquad (12)$$

pelo método do gradiente, com a diferença que, para o Isomap, a distância δ_{ij} entre dois pontos no espaço original, será representada pela sua distância geodésica $d_G(i,j)$.

O algoritmo Isomap pode ser resumido da seguinte forma:

Resumo do Algoritmo Isomap

Entrada: matriz n x m dos dados de entrada **X**.
Saída: matriz n x p dos dados de saída **Y.**

Passo 1: Determinar os k-vizinhos dos pontos de **X**.
Para cada vetor $\mathbf{x_i}$ do conjunto de entrada **X**
Computar a distância de $\mathbf{x_i}$ para todos os outros pontos $\mathbf{x_j}$
Encontrar as k menores distâncias de $\mathbf{x_i}$
Apontar os pontos correspondentes a essas menores distâncias como os vizinhos de $\mathbf{x_i}$

Passo 2: Construir o grafo ligando os pontos aos seus vizinhos
Para cada vetor $\mathbf{x_i}$ do conjunto de entrada **X**
Conectar $\mathbf{x_i}$ a todos os seus k vizinhos $\mathbf{x_j}$ por meio de ramos com peso $w_{ij} = \| \mathbf{x_i} - \mathbf{x_j} \|$.

Passo 3: Calcular os menores caminhos entre todos os pares de pontos do grafo usando algoritmos como o de Dijkstra ou de Floyd. Essas serão as distâncias geodésicas aproximadas e serão armazenadas na matriz de distâncias **D**.

Passo 4: Aplicar o método clássico do MDS métrico à matriz de dissimilaridades **D,** para encontrar a matriz **Y** dos vetores que representam as projeções de **X** em baixa dimensão.

A utilização de distâncias geodésicas ao invés de euclidianas e o conceito do grafo que interliga as vizinhanças fornecem ao algoritmo a habilidade de recuperar *manifolds* que representem um espaço euclidiano de baixa dimensão, porém fortemente curvado ou dobrado e inserido num espaço de entrada de dimensão bem mais elevada.

2.7. LLE

O método *Locally Linear Embedding* (LLE), proposto por Roweis e Saul [7], busca uma projeção global dos dados, captando as características "locais" de um *manifold*. LLE modela o *manifold* tratando-o como uma união de vários pequenos pedaços (retalhos) assumindo que os dados se encontrem sobre o *manifold* ou muito próximos dele e que esse *manifold* apresente aproximada linearidade local em todos os pontos [21].

Considerando que cada ponto $\mathbf{x}_i \in R^m$ possui um número de vizinhos mais próximos indexados pelo conjunto $\mathcal{V}(i)$ e que $\mathbf{y}_i \in R^p$ seja a representação de $\mathbf{x}_i$ na baixa dimensão, a idéia é expressar cada $\mathbf{x}_i$ como uma combinação linear dos seus vizinhos e depois construir os $\mathbf{y}_i$ de tal forma que eles sejam expressos pela mesma combinação linear dos seus correspondentes vizinhos, também indexados por $\mathcal{V}(i)$.

O algoritmo se desenvolve em duas fases: (1) Calcula os melhores coeficientes possíveis que aproximam cada ponto $\mathbf{x}_i$ por uma combinação linear ponderada dos pontos que se encontram na sua vizinhança. O conjunto de tais coeficientes para cada ponto constitui os pesos de reconstrução desse ponto. (2) fixando os coeficientes (pesos) obtidos a partir dos pontos em alta dimensão, busca um conjunto de pontos $\mathbf{y}_i$ de baixa dimensão que possam ser linearmente aproximados (reconstruídos) pela combinação dos pontos vizinhos ponderados pelos mesmos coeficientes já fixados na primeira fase.

Na etapa 1, busca-se minimizar os erros de reconstrução medidos pela função de custo:

$$\varepsilon(W) = \sum_{i=1}^{n} \left\| \mathbf{x}_i - \sum_{j=1}^{k} w_{ij} \mathbf{x}_j \right\|^2, \qquad (13)$$

onde **W** é a matriz que contém o conjunto dos pesos, i é definido sobre o conjunto dos pontos, j é definido para a vizinhança de um determinado ponto $\mathbf{x}_i$, w_{ij} é a contribuição do ponto $\mathbf{x}_j$ para a reconstrução do ponto $\mathbf{x}_i$, $\Sigma_j w_{ij} \mathbf{x}_j$ é a reconstrução de $\mathbf{x}_i$ em função dos vizinhos $\mathbf{x}_j$ e $\mathbf{x}_i - \Sigma_j w_{ij} \mathbf{x}_j$ é o erro de reconstrução de $\mathbf{x}_i$. Para computar os pesos w_{ij} deve-se minimizar a função de custo, com a restrição de que $\Sigma_j w_{ij} = 1$.

A solução desse problema de mínimos-quadrados fornece os valores otimizados dos pesos procurados. Consideremos um ponto de entrada particular $\mathbf{x}_i$ que possui k vizinhos mais próximos $\mathbf{x}_j$ e pesos de reconstrução w_{ij}. Decorrente da condição $\Sigma_j w_{ij} = 1$, a contribuição desse ponto $\mathbf{x}_i$ para o erro de reconstrução pode ser expresso como:

$$\varepsilon_{(i)}(\mathbf{W}) = \left\| \mathbf{x}_i - \sum_{j=1}^{k} w_{ij}\, \mathbf{x}_j \right\|^2 = \left\| \sum_{j=1}^{k} w_{ij}(\mathbf{x}_i - \mathbf{x}_j) \right\|^2 , \qquad (14)$$

Conforme descrito em [7] e [22] esse erro pode ser expresso como:

$$\varepsilon_{(i)}(\mathbf{W}) = \sum_{j=1}^{k}\sum_{q=1}^{k} w_{ij} w_{iq} C_{jq}^{(i)}, \qquad (15)$$

onde a matriz de covariância local, $C_{jk,}$ é definida como:

$$C_{jq}^{(i)} = (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_k). \qquad (16)$$

Como mostrado em [25], na prática, a maneira mais eficiente de encontrar os pesos otimizados é resolver o sistema de equações lineares

$$\sum_{j=1}^{k} C_{jq} w_{iq} = 1 , \qquad (17)$$

e depois fazer uma mudança de escala dos pesos obtidos para que a sua soma seja unitária. Essa abordagem é utilizada no pseudo-código do algoritmo apresentado pelos autores e mostrado mais adiante. Maiores detalhes sobre a conceituação formal do método LLE podem ser encontrados em [7, 21, 22].

A condição de linearidade local e as restrições assumidas fazem com que os pesos calculados, para um ponto em particular, sejam invariantes a translações, rotações e reescalonamentos do ponto e seus vizinhos. Essa condição de invariância sugere a capacidade de reconstrução do conjunto de pontos numa dimensão mais baixa, utilizando o mesmo conjunto **W** dos pesos w_{ij} que definem a geometria local para cada ponto. Tal idéia permite encontrar um mapeamento em dimensão reduzida, baseado em preservação de vizinhanças. Para tanto, na etapa 2 do algoritmo, é realizado o mapeamento dos pontos $\mathbf{x}_i$ do espaço de entrada para os vetores $\mathbf{y}_i$ no espaço de baixa dimensão. Isso é obtido, minimizando outra função de custo, como segue:

$$\varphi(\mathbf{Y}) = \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_{j=1}^{k} w_{ij} \mathbf{y}_j \right\|^2 . \qquad (18)$$

Essa função tem a mesma forma da função de erro da eq. (13) usada na etapa 1, com a diferença de que, nesse caso, os pesos w_{ij} são fixados (do cálculo anterior) e são buscados os valores das coordenadas dos pontos projetados $\mathbf{y}_i$.

De acordo com Roweiss e Saul [7, 21], a otimização dos pesos pode ser feita de acordo com o seguinte caminho: pode-se reescrever a equação (18) sob uma forma quadrática baseada em produtos internos das saídas $\mathbf{y}_i$:

$$\varphi(\mathbf{Y}) = \sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}(\mathbf{y}_i^T \mathbf{y}_j) = Tr(\mathbf{Y}\mathbf{M}\mathbf{Y}^T), \qquad (19)$$

onde: $\mathbf{M} = \{m_{ij},\ i = 1,...,n\ e\ j = 1,....,k\}$ é uma matriz n x n dada por $= (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$; **W** é uma matriz esparsa n x n contendo os pesos w_{ij} de todos os n pontos $\mathbf{x}_i$; **I** é a matriz identidade $\mathbf{I}_n$; e **Y** é a matriz dos vetores-coluna $\mathbf{y}_i$. Também é mostrado [21] que as coordenadas p-dimensionais dos vetores projetados $\mathbf{y}_i$ que minimizam a função de custo citada podem ser encontradas computando os auto-vetores correspondentes aos d menores auto-valores não nulos de **M**.

De Ridder and Duin [22] mostram que para encontrar a solução desse problema pode-se impor a restrição de que a matriz de covariância dos Y seja a identidade. Segundo os mesmos autores [22], o problema fica resumido à minimização de $Tr(\mathbf{YMY}^T)$ com a restrição $1/n(\mathbf{YY}^T) = \mathbf{I}$. Com o uso de multiplicadores de Lagrange, obtém-se $(\mathbf{M} - \boldsymbol{\lambda})\mathbf{Y}^T = 0$, onde $\boldsymbol{\lambda}$ é o multiplicador de Lagrange (matriz diagonal). Esse problema de autovalores possui todos os autovetores de **M** como soluções, mas os autovetores com menores autovalores minimizam a função de erro . O autovetor com menor autovalor corresponde à média de **Y** e pode ser descartado, garantindo a condição $\Sigma_i \mathbf{y}_i = 0$ (coordenadas centradas na origem). Os *p* autovetores seguintes (de tamanho *n*) quando transpostos fornecem os *n* vetores $\mathbf{y}_i$, com dimensão *p***,** que minimizam a função de erro e representam a solução procurada. Os *n* vetores encontrados formam uma matriz de coordenadas cartesianas centradas na origem. Detalhes podem ser vistos nas referências já citadas [7, 21, 22].

A vizinhança dos pontos de origem pode ser definida como os *k* vizinhos mais próximos para cada ponto, segundo uma medida de distância euclidiana. Assim o valor de *k* é um parâmetro livre a ser definido para cada caso.

A síntese do algoritmo é fornecida pelos autores do método [29] e é reproduzida a seguir:

Resumo do Algoritmo LLE

Entrada: matriz *n* x *m* dos dados de entrada **X**.
Saída: matriz *n* x *p* dos dados de saída **Y.**

Passo 1: Determinar os *k*-vizinhos no espaço **X**.
Para cada vetor $\mathbf{x_i}$ do conjunto de entrada **X**
Computar a distância de $\mathbf{x}_i$ para todos os outros pontos $\mathbf{x}_j$.
Encontrar as *k* menores distâncias.
Apontar os pontos correspondentes a essas menores distâncias como os vizinhos de $\mathbf{x}_i$.

Passo 2: Encontrar a matriz dos pesos de reconstrução **W**.
Para cada vetor $\mathbf{x_i}$ do conjunto de entrada **X**.
Criar a matriz **Z** consistindo de todos os vizinhos de $\mathbf{x}_i$.
Subtrair $\mathbf{x}_i$ de cada coluna de **Z**.
Computar a matriz de covariância local **C** = **Z´** * **Z**.
Resolver o sistema linear **C** * **w** = **1´**, em **w**.
Forçar a condição $\sum w_{ij} = 1$, fazendo $\mathbf{w}_u = \mathbf{w}/\sum w_{ij}$.
Fazer $w_{ij} = 0$, para todo *j* que não é vizinho de *i*.
Fazer os restantes elementos da *i*-ésima linha de **W** iguais às coordenadas de $\mathbf{w}_u$.

Passo 3: Determinar as coordenadas dos pontos projetados **Y** usando os pesos **W**.
Criar a matriz esparsa **M** = (**I** - **W**)´ * (**I** - **W**).
Encontrar os *d+1* autovetores de ordem mais baixa da matriz **M**.
(correspondendo aos *d+1* menores autovalores)
Fazer a *q*-ésima linha de **Y** igual ao autovetor de ordem *q+1*.
(descartar o autovetor de ordem mais baixa, com autovalor nulo)

Notas: $\mathbf{x}_i$ é o vetor da *i* –ésima coluna de **X** (coordenadas do *i-ésimo* ponto de entrada); (*) representa o produto matricial; **1´** é um vetor coluna unitário; **I** é a matriz identidade; **A´** é a transposta de **A**; **w** é o vetor dos *k* pesos de reconstrução de um ponto $\mathbf{x}_{i,;}$ w_j são os *k* componentes de **w**; $\mathbf{w}_u$ é o resultado do escalonamento de **w** de forma a garantir que a soma de suas coordenadas seja igual a 1.

3. Descrição da metodologia

As avaliações conduzidas neste trabalho seguiram duas abordagens paralelas. Uma delas baseou-se na percepção visual dos resultados gráficos obtidos. A outra, de caráter quantitativo, baseou-se na aplicação de índices que procurassem estabelecer um parâmetro associado à capacidade de manutenção da topologia original. Essas abordagens são descritas nas subseções seguintes.

3.1. Método de avaliação qualitativa das visualizações

Nessa abordagem, o foco das avaliações concentrou-se no resultado final das visualizações bi-dimensionais obtidas pela aplicação de cada um dos algoritmos utilizados. Tomou-se como base uma situação típica de mineração de dados, em que se tem à mão uma massa de dados multidimensionais com estrutura desconhecida e busca-se adquirir, por meio de técnicas de prospecção de dados, noções ou indícios que permitam obter uma visão da estrutura desses dados. A informação mais fundamental sobre a estrutura dos dados é a descoberta das possíveis tendências de organização natural em agrupamentos (*clusters*) ou classes que poderiam permitir a identificação e a classificação de padrões.

Clusters são coleções de objetos similares. Um objeto pertencente a um *cluster* deverá ser mais similar a qualquer objeto do mesmo *cluster* do que a outro qualquer objeto fora dele. No presente caso, os objetos são elementos de dados (amostras) e são representados por pontos num espaço euclidiano multidimensional. Geralmente, utiliza-se uma medida de distância (p. ex. a euclidiana) para representar a similaridade: quanto mais próximos os objetos, mais similares serão. Os *clusters* também podem ser definidos como regiões do espaço contendo relativamente altas densidades de pontos, separadas por regiões de relativamente baixas densidades [37, 50]. Essa definição se aproxima da nossa percepção intuitiva de *clusters* ao visualizarmos espaços em duas ou três dimensões.

O método utilizado para as avaliações qualitativas consiste em aplicar os algoritmos de redução de dimensionalidade a bases de dados de comportamento conhecido e reduzir a duas dimensões os dados multidimensionais nelas contidos, permitindo exibir os pontos projetados em gráficos cartesianos. Aqui, a possibilidade de se ter uma visualização preliminar da distribuição dos dados sob análise é extremamente útil, por oferecer uma imagem de percepção rápida e intuitiva.

Num primeiro passo, para simular uma situação realista de prospecção de dados, partiu-se de uma condição de pretenso desconhecimento da estrutura dos dados, desprezando as informações sobre a composição das classes. Aplicaram-se os diversos algoritmos aos dados de teste, gerando em seguida gráficos cartesianos, representando os pontos projetados no espaço bidimensional. Para o algoritmo SOM, especificamente, foi utilizada, como visualização, a representação gráfica bi-dimensional da *U-matrix*, obtida diretamente do mapa após o treinamento. Nesse passo, sem a utilização das informações sobre as classes, os dados de saída foram mostrados como pontos de uma mesma cor. Foram realizadas observações sobre esses gráficos, visando à identificação dos pontos marcantes na sua distribuição. O objetivo foi evidenciar a percepção que se pode extrair sobre a possível existência de agrupamentos (*clusters*).

Num segundo passo, para efeito de avaliação dos resultados, foram gerados os mesmos gráficos anteriores, acrescentando-se cores para representar as informações privilegiadas disponíveis dos rótulos das classes. A comparação entre os gráficos correspondentes permite avaliar o grau de adequação das visualizações fornecidas pelos diversos algoritmos. Apesar do seu caráter qualitativo, as visualizações de dados são ferramentas essenciais em análise exploratória de dados, uma fase muito importante num processo de descoberta de conhecimento (KDD). A finalidade dessa fase é adquirir um conhecimento geral a respeito dos dados a serem trabalhados, levantando possibilidades e problemas, para determinar a adequabilidade do conjunto de dados e definir qual o modelo a ser aplicado e as ferramentas mais adequadas para o processamento desses dados [25].

3.2. Método de avaliação através de índices

A fim de não se restringir às avaliações qualitativas, que incluem um componente subjetivo, este trabalho propõe a definição e a aplicação de dois índices que se destinam a avaliar o grau de manutenção da ordem topológica original, entre os pontos projetados no espaço reduzido. Esses índices são inter-relacionados e serão denominados ***índices de coincidência ordenada e não ordenada de vizinhanças***. Além de uma definição simples e intuitiva, esses índices refletem de alguma maneira o grau de manutenção da topologia no conjunto de dados de saída.

3.2.1. Índice de coincidência não ordenada de vizinhanças

O índice de coincidência não ordenada de vizinhanças, c*(k)* é definido como:

$$c(k) = \frac{1}{nk}\sum_{i=1}^{n} u_1(i,k) \qquad (20)$$

onde:
k é um parâmetro livre que define o tamanho da vizinhança considerada em torno de cada um dos pontos, i.e., o número de vizinhos mais próximos de cada ponto, seja no espaço de entrada, seja no de saída;
n é o número de pontos do conjunto de dados;
i é o índice que indica a posição de cada ponto dentro desse conjunto de dados;
$u_1(i, k)$ é uma função interna que quantifica o índice para cada ponto $\mathbf{x}_i$ no espaço de entrada (alta dimensão) comparado ao respectivo ponto $\mathbf{y}_i$ no espaço de saída (baixa dimensão).

A função $u_1(i, k)$ depende do tamanho da vizinhança e é definida para cada ponto $\mathbf{x}_i$ como:

$$u_1(i, k) = \#\{ V_k(\mathbf{x}_i) \cap V_k(\mathbf{y}_i) \}, \quad (21)$$

sendo:
{A} a operação que computa a cardinalidade do conjunto A;
$V_k(\mathbf{x}_i)$ a vizinhança de tamanho *k* do ponto $\mathbf{x}_i$, o conjunto composto pelos índices dos pontos que sejam os *k* vizinhos mais próximos de $\mathbf{x}_i$ no espaço de entrada; e
$V_k(\mathbf{y}_i)$ o equivalente para o ponto $\mathbf{y}_i$ no espaço de saída.

A interpretação do índice é simples. Para um tamanho de vizinhança *k*, previamente definido, o índice reflete a quantidade de vizinhos dos pontos do espaço de entrada que, depois de projetados, ainda se mantêm vizinhos dos pontos correspondentes do espaço de saída, respectivamente. O índice *c(k)* pode variar de *0* a *1*, sendo o valor *1* correspondente ao melhor caso possível dentro dos limites da definição. Embora a ordem relativa dos pontos dentro da vizinhança não seja levada em conta na avaliação, a simples coincidência da presença dos pontos nas vizinhanças respectivas fornece uma idéia da qualidade da projeção obtida.

Um maior rigor, com relação à ordem relativa dos pontos dentro da vizinhança, pode ser obtido, definindo-se um segundo índice derivado do anterior e apresentado a seguir.

3.2.2. Índice de coincidência ordenada de vizinhanças

O índice de coincidência ordenada de vizinhanças, $o(k)$ é definido pela mesma expressão do índice anterior, porém com a modificação da função $u(i, k)$ para refletir a exigência de manutenção do ordenamento dentro da vizinhança dos pontos de saída. Assim, teremos:

$$o(k) = \frac{1}{nk}\sum_{i=1}^{n} u_2(i, k), \quad (22)$$

sendo mantidas as mesmas definições anteriores para os parâmetros *n, k, i* e para os conjuntos $V_k(\mathbf{x}_i)$ e $V_k(\mathbf{y}_i)$. Extraindo dos dois conjuntos de vizinhanças correspondentes a um determinado ponto de ordem *i*, apenas os pontos que se repetem nos dois conjuntos, obteremos dois subconjuntos que conterão os mesmos elementos (índices), porém não necessariamente dispostos na mesma ordem. O valor da função $u_2(i, k)$ será, então, o número de correspondências perfeitas (valor e posição dos índices) entre esses dois subconjuntos. Percebe-se que este segundo índice retrata não só a presença dos mesmos vizinhos nos dois conjuntos de pontos, mas exige uma correspondência na ordem relativa desses vizinhos. Nesse caso, um valor igual a um garantiria uma perfeita manutenção do ordenamento topológico original.

3.2.3. Considerações sobre a aplicação dos índices

Neste ponto, é preciso alertar para as dificuldades envolvidas na aplicação desses índices aos mapeamentos originados pelo algoritmo SOM. Sabe-se que o SOM, entre outras capacidades, funciona, também, como um quantizador vetorial, o que significa que vários pontos de entrada poderão estar projetados num único ponto de saída (protótipo). Esse fato produzirá incertezas de interpretação dos fatores vizinhança e ordenamento. Portanto, para o caso do SOM, é necessário estabelecer algumas convenções para os referidos fatores.

Definiu-se, para o caso do SOM, que (*1*) os pontos representados no mesmo neurônio sejam tratados como vizinhos no espaço de saída e (*2*) pontos em tal condição, por não poderem ser distinguidos no espaço projetado, sejam considerados como corretamente ordenados. Essa é uma abordagem otimista e, portanto, o uso desses critérios tende a favorecer o SOM nos resultados comparativos com os outros métodos. Por outro lado, deve-se considerar que as convenções

adotadas são coerentes com a percepção intuitiva da visualização dos dados projetados (por exemplo, numa análise visual não serão distinguíveis eventuais erros de ordenamento entre pontos representados no mesmo neurônio).

Particularmente, por esses critérios, haverá uma dependência entre os valores do índice e as dimensões escolhidas para o mapa. Não existe um método padrão para determinar o tamanho ótimo do mapa e a escolha é subjetiva, normalmente baseada em um grande número de tentativas, com diversos arranjos. Para tornar a escolha do tamanho do mapa menos subjetiva, optou-se por utilizar, para todos os mapas, os tamanhos *default* computados automaticamente pelo pacote de software *Somtoolbox*, em função do conjunto dos dados de entrada. Nesse caso, é utilizada uma fórmula heurística [39] que define o número de neurônios como $m = 5\sqrt{n}$, onde *n* é o número de amostras do conjunto. Para a proporção entre os tamanhos dos lados do mapa é utilizada a relação entre os dois maiores autovalores da matriz de covariâncias das amostras (com arredondamento para valores inteiros, de forma que o produto entre as duas dimensões seja o mais próximo possível de *m)*. Para evitar uma forma excessivamente alongada do mapa (caso de diferenças exageradas entre as duas dimensões), a razão entre as dimensões não poderá ultrapassar o valor *2.5*.

Deve-se ter em mente que a quantização vetorial é uma característica inerente ao SOM e, inevitavelmente cria uma dificuldade quando se tenta compará-lo com métodos que apresentam espaço de saída contínuo. Entretanto, nos casos em que essa condição se constitui um fator favorável, é aceitável que as vantagens decorrentes dessa capacidade natural o favoreçam de alguma forma nas avaliações.

4. Descrição dos dados e configurações

Esta seção descreve as bases de dados utilizadas para os testes e as configurações dos métodos aplicados a essas bases. Para os testes, foram utilizadas, como referências, bases de dados publicamente disponíveis para pesquisas na área de aprendizagem de máquina e reconhecimento de padrões, no repositório da *Universidade da Califórnia em Irvine, UCI KDD Archive* [12]. Os critérios para escolha das bases foram: *(1)* serem bem conhecidas no meio científico; *(2)* possuírem razoável quantidade de atributos, proporcionando uma considerável relação de redução de dimensionalidade; e (*3*) representarem medições de objetos do mundo real ou simulações dessas medições. As bases utilizadas apresentam número de atributos igual a 13, 60 e 72, e quantidade de amostras igual a 178, 600 e 2000, respectivamente. A seguir são fornecidas algumas informações sobre essas bases.

4.1. Base de dados *Wine*

A base de dados "*Wine recognition data*" (*Wine*) representa o resultado de uma análise química de amostras de vinhos produzidos numa mesma região da Itália, mas provenientes de três diferentes culturas. As análises determinaram as quantidades de 13 constituintes químicos presentes em todas as amostras. A base possui 178 amostras, com 13 atributos contínuos e 3 classes (59 exemplos da classe 1, 71 da classe 2 e 48 da classe 3). Essa base possui uma estrutura de classes bem definida e é considerada "bem comportada" relativamente a problemas de classificação [11]. No entanto, para a aplicação aqui considerada, essa relativa simplicidade não obscurece o seu valor como meio comparativo.

Os trabalhos mais significativos relatados nas páginas do repositório UCI [11] referem-se à comparação de classificadores [32] e a testes de melhoria de um algoritmo de classificação [33]. Ali também são citados dezenas de trabalhos que utilizaram essa base.

4.2. Base de dados *Synthetic Control*

A base de dados "*Synthetic Control Chart Time Series"* (*Control*) consiste de gráficos de controle gerados sinteticamente por um processo de controle simulado [11]. A base apresenta 600 padrões, 100 por classe, cada um representando um gráfico de controle com amostragens em 60 intervalos de tempo regulares e seqüenciais, constituindo os campos de cada registro. Existem 6 classes associadas a diferentes comportamentos do sistema de controle: *(1)* normal; *(2)* cíclico; *(3)* tendência de crescimento contínuo; *(4)* tendência de decrescimento contínuo; *(5)* ocorrência de degrau positivo e *(6)* ocorrência de degrau negativo. A figura 4 ilustra a média dos 100 padrões de cada classe, possibilitando a visualização do comportamento geral das classes. Esta é uma base interessante, pela natureza peculiar dos atributos, representando a dinâmica de uma função de controle ao longo do tempo e por apresentar um bom número de atributos

Os trabalhos mais relevantes citados na página correspondente do repositório UCI são referentes à busca de similaridade de séries temporais [34] e reconhecimento de padrões de gráficos de controle [35].

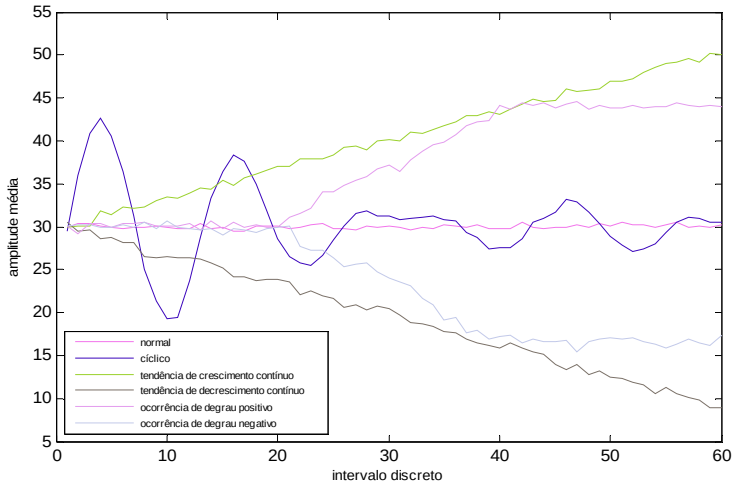


Figura 4: Médias das classes da base *Synthetic Control*.

4.3. Base de dados *Animals*

A base de dados *Animals* representa quatro classes de mamíferos por meio de 72 atributos referentes em maior parte a medidas morfológicas dos animais. A quantidade original de amostras era de 500000 pontos. Por razões computacionais, da base original foi extraída uma amostra aleatória de 2000 pontos, para utilização neste trabalho.

A base *Animals* caracteriza-se pelo volume de amostras e por um número significativo de atributos. Ela foi originalmente criada para os testes utilizados por Fisher *et al* [36] para testar o algoritmo *Classit*. Entre outras utilizações dessa base, destaca-se a pesquisa da técnica de *clusterização* de dados de alta dimensionalidade, denominada *Garden* [37].

4.4. Configurações

Para quase todos os métodos, exceto o SOM e redes MLP, foram utilizadas as rotinas implementadas no *DRtoolbox* [8]. Os testes com o algoritmo SOM utilizaram o *SOM toolbox* [31]. Como ambiente de testes foi usado o *Matlab*. Em todos os testes, os conjuntos de dados utilizados foram previamente normalizados. As bases *Wine* e *Control*, para apresentarem média nula e variância unitária (z-*score*). A base *Animals* já é normalizada originalmente para a faixa de magnitudes entre zero e um, portanto, foi mantida como tal. Em todos os casos, os dados de entrada foram reduzidos para um espaço de saída bidimensional.

Para o algoritmo de Sammon foi usada a métrica euclidiana como medida de distância.

As redes SOM utilizaram geometria hexagonal e função de aprendizagem linear. Para o treinamento na fase de aproximação, foi estabelecido o limite de 1000 épocas, um raio de vizinhança inicial de 5 unidades e valor inicial da taxa de aprendizagem igual a 0.05. Para a fase de refinamento foi utilizado um limite de 10000 épocas, um raio de vizinhança inicial de 5 unidades e um valor inicial da taxa de aprendizagem igual a 0.01. Como os resultados do SOM, particularmente as visualizações, dependem dos tamanhos escolhidos para o mapa, com o fim de tornar a escolha menos arbitrária, os tamanhos dos mapas utilizados (*11 x 6* para a base *Wine, 17 x 7* para a base *Control* e *22 x 10* para a base *Animals*) foram obtidos automaticamente pelo *SomToolbox*, a partir dos conjuntos de dados de entrada, conforme procedimento padrão mencionado na subseção 3.2.3.

Para os métodos LLE e Isomap, o único parâmetro livre é o tamanho da vizinhança, k. Nos testes realizados os valores foram definidos empiricamente, executando os algoritmos um grande número de vezes, usando diversos valores de k e avaliando a maior ou menor clareza dos gráficos respectivos. A escolha desse parâmetro buscou sempre a condição mais favorável, em princípio. Os valores escolhidos para as bases *Wine, Control* e *Animals* foram, respectivamente, $k = 12$, $k = 16$ e $k = 100$ para o LLE e $k = 12$ (para as duas primeiras bases) e $k = 200$ (base *Animals*), para o Isomap.

As arquiteturas das Redes Neurais Autoassociativas utilizadas foram fixadas em 5 camadas. Conforme já descrito na subseção 2.4, nesse caso, tanto o número de neurônios das camadas de entrada/saída quanto da camada central são fixados pelas condições do problema (dimensões de entrada e de saída). Portanto, as únicas camadas cujo tamanho precisa ser escolhido são as camadas embutidas codificadora e decodificadora (figura 1), usualmente contendo o mesmo número de neurônios. A escolha desse número depende de fatores empíricos. Por isso, foram realizados testes com diversas arquiteturas variando esse parâmetro, até obter uma razoável convergência da função de custo e fixando a arquitetura que produziu o melhor resultado para cada base. As arquiteturas escolhidas são descritas a seguir. Para a base *Wine,* a rede é composta por 13 neurônios nas camadas de entrada/saída, 4 neurônios nas camadas escondidas intermediárias e 2 neurônios na camada gargalo. A magnitude do funcional de erro convergiu para um valor $\mathcal{E} = 1.07 \times 10^{-3}$. Para a base *Control*, a rede contém 60 neurônios nas camadas de entrada/saída, 5 neurônios nas camadas escondidas intermediárias e 2 neurônios na camada gargalo. Nesse caso, a convergência foi mais difícil e o funcional de erro convergiu para um valor $\mathcal{E} = 3.99 \times 10^{-1}$. Para a base *Animals*, a rede contém 72 neurônios nas camadas de entrada/saída, 7 neurônios nas camadas escondidas intermediárias e 2 neurônios na camada gargalo. Aqui, o número dos testes necessários para apontar uma solução satisfatória para a arquitetura foi bem elevado. Para a solução escolhida, o funcional de erro convergiu para um valor $\mathcal{E} = 2.46 \times 10^{-1}$.

5. Descrição dos Testes e Resultados

Conforme descrito na seção 3, os testes realizados visaram inicialmente uma avaliação qualitativa dos resultados da aplicação dos métodos selecionados. Essas avaliações foram baseadas nas visualizações gráficas dos pontos, e buscou-se identificar a capacidade dos gráficos revelarem tendências de *clusters*. Numa outra linha, os testes visaram quantificar a capacidade de manutenção da topologia, usando índices aplicados às projeções fornecidas por cada um dos métodos.

A seguir são descritos os testes, juntamente com a avaliação dos resultados obtidos.

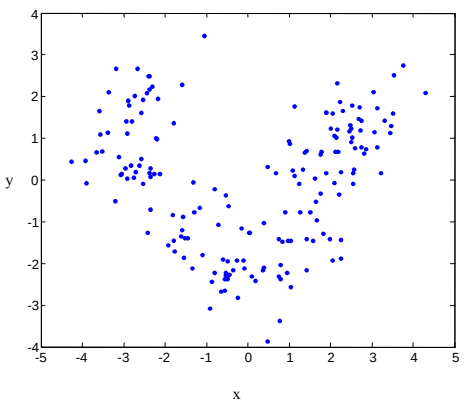
5.1 Testes de avaliação qualitativa das visualizações

Num caso real (não-supervisionado) de prospecção de dados, não se conhecem *a priori* as informações sobre a quantidade ou mesmo sobre a existência de *clusters* nos dados analisados. Conseqüentemente, a análise seria baseada em visualizações monocromáticas e a informação sobre a estrutura dos dados seria obtida da posição relativa dos pontos entre si e da sua distribuição no gráfico. Entretanto, em casos de teste, como os que nos ocupamos aqui, temos acesso à informação privilegiada sobre os rótulos atribuídos a cada amostra do conjunto de dados. Assim, podemos usar esse conhecimento prévio para avaliar a qualidade dos gráficos obtidos.

As subseções a seguir apresentam as visualizações obtidas, antes e depois da inclusão da cor para identificar os rótulos e as avaliações dos resultados.

5.1.1. Testes com a base *Wine*

Inicialmente são apresentados os gráficos sem a informação das classes, como seria o caso de uma análise prévia, numa típica tarefa de prospecção de dados.



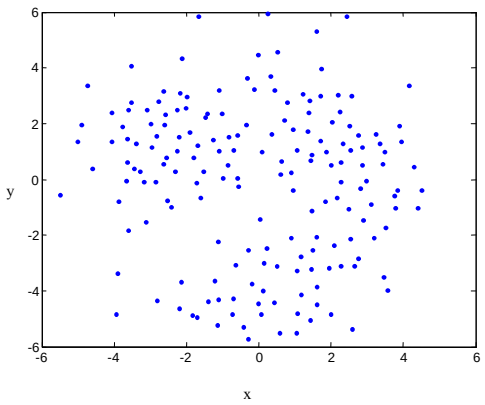


Figura 5: Projeção PCA dos pontos da base *Wine*

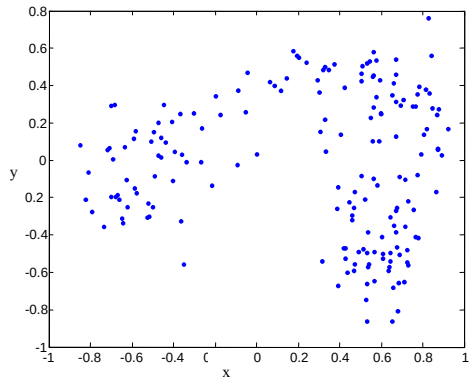


Figura 6: Projeção de Sammon dos pontos da base *Wine*.

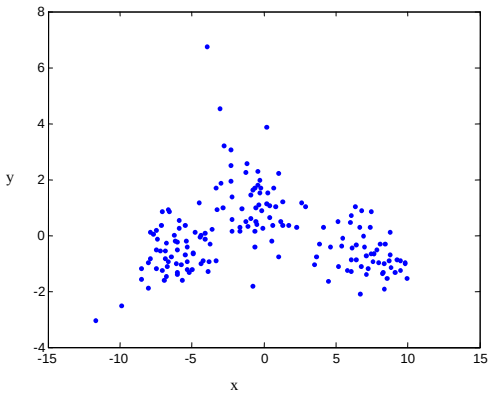
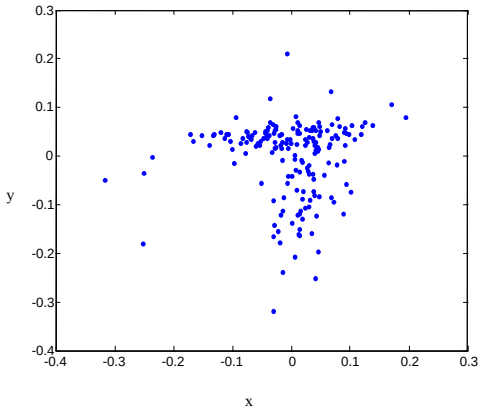


Figura 7: Projeção da base *Wine* , obtida da rede MLP auto-associativa.

Figura 8: Projeção da base *Wine,* obtida pelo método Isomap.



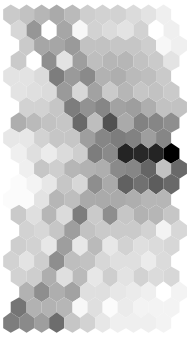


Figura 9: Projeção da base *Wine,* obtida pelo método LLE.

Figura 10: Projeção da *U-matrix* do mapa SOM, treinado com a base *Wine*.

Uma inspeção visual das figuras 5 a 10 permite verificar que a projeção obtida do algoritmo LLE não permite nenhuma conclusão sobre os agrupamentos. Num nível intermediário, as projeções PCA e MLP AA que se apresentaram muito semelhantes, mostram alguma diferença de densidade em certas regiões e o Sammon que mostra um esboço muito tênue de fronteiras entre três agrupamentos. À primeira vista, os gráficos do SOM e do Isomap estariam num plano superior, pois apresentam, com razoável nitidez, fronteiras entre três regiões de possíveis agrupamentos. Para o SOM, essa tendência é representada pelas três regiões em tons claros, separadas pelas regiões de fronteira, de tons mais escuros.

A seguir são apresentados os mesmos gráficos anteriores, agora utilizando cores para identificar a classe a que pertencem as amostras. Para o SOM, um histograma colorido dos pontos projetados é superposto à *U-matrix*.

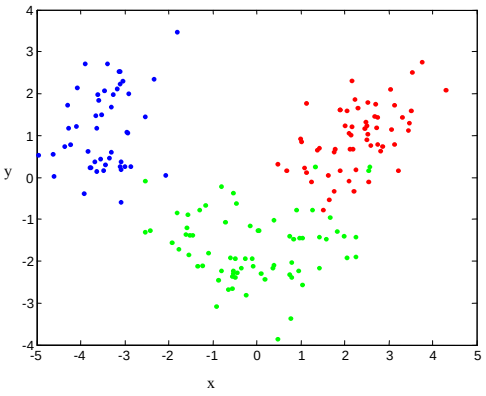


Figura 11: Projeção PCA dos pontos da base *Wine*, rotulados.

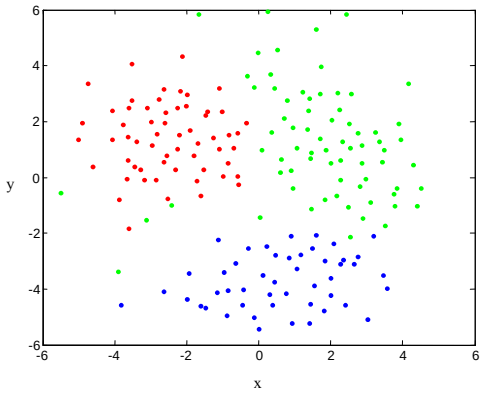


Figura 12: Projeção de Sammon dos pontos da base *Wine*, rotulados.

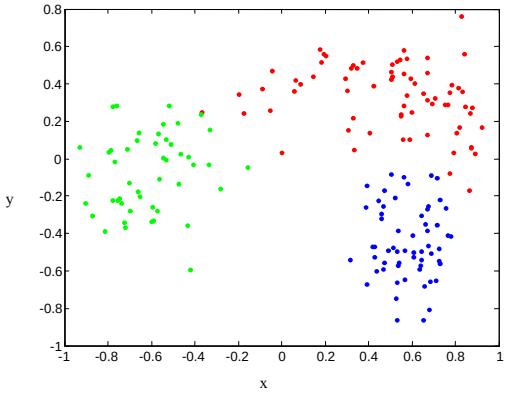


Figura 13: Projeção da base *Wine* obtida da rede auto-associativa, com informação dos rótulos

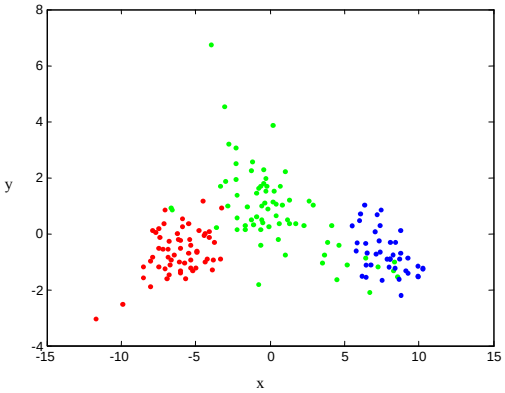


Figura 14: Projeção da base *Wine* obtida do método Isomap, com informação dos rótulos.

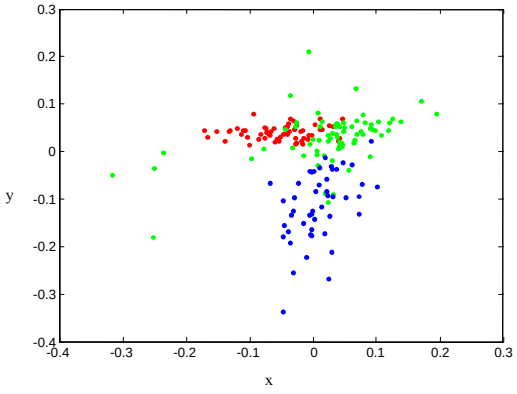


Figura 15: Projeção da base *Wine* obtida do método LLE, com informação dos rótulos

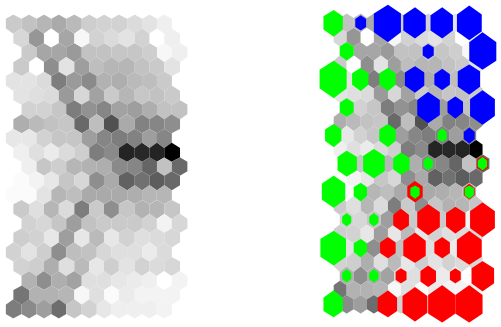


Figura 16: Projeções da base *Wine* obtidas do mapa SOM treinado. Esquerda: *U-matrix* pura. Direita: *U-matrix*, com histograma e rótulos.

A análise das figuras de 11 a 16 permite constatar visualmente que todos os métodos inspecionados cumpriram, a seu modo, o requisito básico de manter os objetos (pontos) assemelhados, geometricamente próximos na projeção. Considerando os métodos que a partir dos gráficos anteriores apresentaram indícios de três agrupamentos, observamos que a visualização da projeção *Sammon*, com a presença dos rótulos, apresentou uma distribuição coerente, mas o gráfico sem os rótulos não permitia identificar com clareza os agrupamentos. O gráfico Isomap havia mostrado a tendência de três agrupamentos com razoável nitidez. No entanto, o gráfico colorido respectivo mostrou que a fronteira real entre as

classes “2” e “3” (representadas nas cores verde e azul) é bem diferente daquela sugerida pela impressão visual do gráfico monocromático. Para a *U-matrix* do SOM, o gráfico em cores confirmou de maneira bastante aproximada as previsões obtidas anteriormente, sobre a quantidade e as fronteiras entre as classes.

5.1.2. Testes com a base *Synthetic Control*

A análise e visualização da estrutura dos dados da base *Synthetic Control* revelou-se mais complexa do que o exemplo anterior. Isso é explicável, em parte, pela sua natureza atípica. Cada registro da base é uma representação discretizada de uma curva de controle em função do tempo e cada atributo (coluna) representa um valor seqüencial da curva.

Os mesmos procedimentos anteriores foram aplicados à base *Control.* Como o procedimento em dois passos já foi detalhado no item anterior, para essa base serão mostrados diretamente os gráficos já acrescidos da cor para representar as classes (figuras 17 a 22). Cosequentemente, deve-se imaginar que no primeiro passo os gráficos são os mesmos, com exceção da informação da cor.

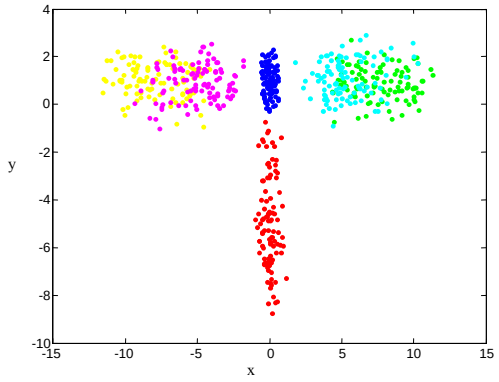


Figura 17: Projeção PCA dos pontos da base *Contro*l com informação dos rótulos.

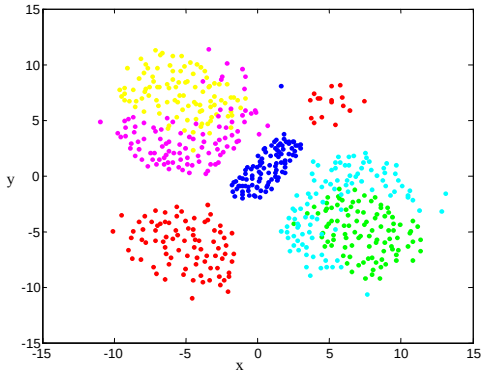


Figura 18: Projeção de Sammon dos pontos da base *Control* com informação dos rótulos.

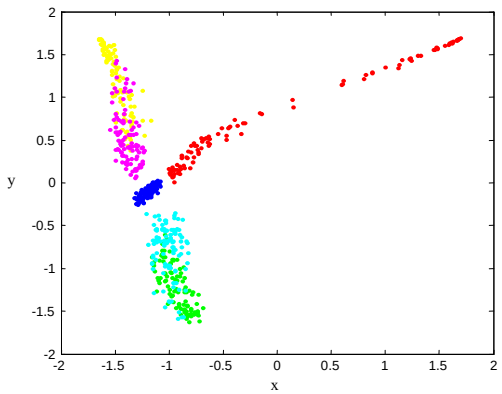


Figura 19: Projeção da base *Control* obtida da rede MLP auto-associativa, com informação dos rótulos

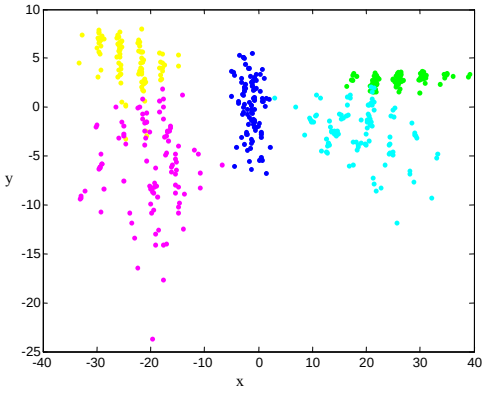


Figura 20: Projeção da base *Control* obtida do método Isomap, com informação dos rótulos.

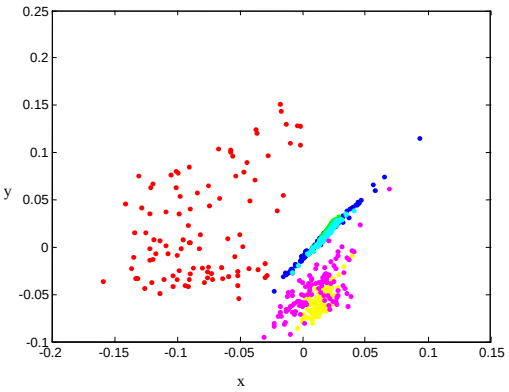


Figura 21: Projeção da base *Control* obtida pelo método LLE, com informação dos rótulos.

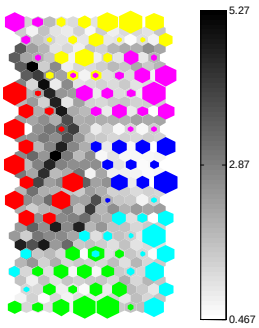


Figura 22: Projeção da *U-matrix* do mapa SOM treinado com a base *Control*, acompanhada do histograma e informação dos rótulos.

A análise dos gráficos demonstra que os diversos métodos sofreram limitações. O método Isomap, além de não distinguir claramente as classes, apresentou um grave problema de perda de informações. Conforme mencionado por de Maatten *et al* [9] essa é uma dificuldade típica do algoritmo que, ao formar o grafo das vizinhanças, não conecta alguns pontos do espaço de entrada. Nos casos de espaços de dados descontínuos ou com grandes variações de densidades de pontos (como p. ex. distribuições com *clusters* fortemente delineados) esse efeito é ainda mais grave, como foi possível verificar para essa base de dados. Nesse caso, o algoritmo subtraiu uma classe completa (veja-se a ausência da classe vermelha na figura 20) comprometendo a qualidade da visualização.

O método LLE forneceu visualizações que tornam muito difícil retirar informações úteis a respeito da estrutura dos dados.

Em todos os casos, as classes (3) e (5) foram sempre confundidas, como se formassem um único agrupamento. Isso aconteceu também com as classes (4) e (6). É verdade que em todos os casos esses dois grupos de classes, embora sem apresentarem a devida separação entre elas, apresentaram coesão interna, aparecendo lado a lado, mas em geral contidas no seu espaço próprio.

Por outro lado, isso parece justificável ao observarmos o comportamento médio de cada classe, representado pelas curvas na figura 1. De fato, é possível verificar uma semelhança de comportamento entre as classes “3” e “5”. Ambas apresentam tendência constante de crescimento. Algo similar acontece entre as classes “4” e “6”. Essas apresentam tendência constante de decrescimento. Essas semelhanças poderiam justificar o fato dessas classes se confundirem nas visualizações. Um olhar mais aguçado para a *U-matrix* do SOM, naturalmente motivado pelo conhecimento posterior do problema, mostra que é possível divisar uma tênue região de fronteira entre essas classes (linhas de cinza um pouco mais escuras) sugerindo a possibilidade de dois agrupamentos muito próximos.

Foi observada uma descontinuidade marcante na classe “2” (comportamento cíclico) a ponto de aparecer nas representações gráficas como se fosse uma clara separação entre dois agrupamentos distintos. Isso parece uma anomalia dos algoritmos provocada pela forte redução da dimensão. Entretanto, o fato desse fenômeno se apresentar para praticamente todos os métodos, sugere que isso pode se dever a alguma peculiaridade dos próprios dados.

5.1.3. Testes com a base *Animals*

Tal como no item 5.1.2, nas figuras 23 a 28, a seguir, são apresentados os gráficos, já contendo as informações dos rótulos. Analisando essas figuras, podemos observar alguns pontos marcantes. Na figura 26, pode-se constatar o mesmo problema de perda de dados mencionado na seção anterior, quando o Isomap é aplicado a conjuntos de dados com *clusters* bem marcados. Nesse caso, o defeito foi ainda mais grave, pois duas classes foram completamente perdidas na projeção. Mesmo variando o tamanho da vizinhança dentro de uma larga faixa (30 a 300) o efeito foi sempre igual ou pior do que o verificado na figura.

O algoritmo LLE (figura 27) mostrou uma visualização sem a menor utilidade, produzindo um congestionamento de todo o conjunto de dados em apenas três pontos. Segundo van der Maaten [9] esse método tende a apresentar limitações em situações de reduções drásticas de dimensionalidade, particularmente quando a dimensão final é muito baixa (como no presente caso) tendendo a comprimir largas regiões do espaço de dados em um único ponto.

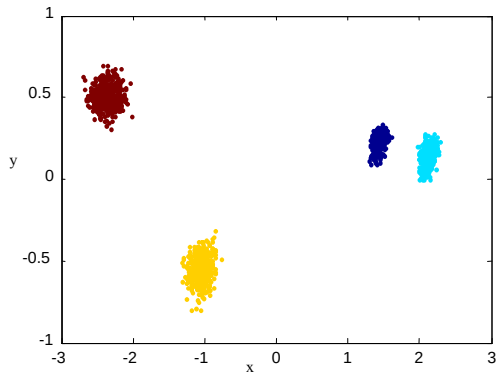


Figura 23: Projeção PCA dos pontos da base *Animals* com informação dos rótulos.

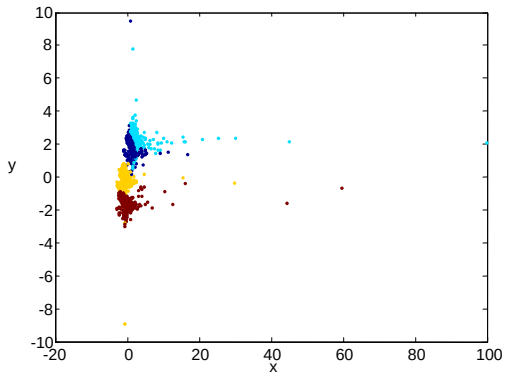


Figura 24: Projeção de Sammon dos pontos da base *Animals* com informação dos rótulos.

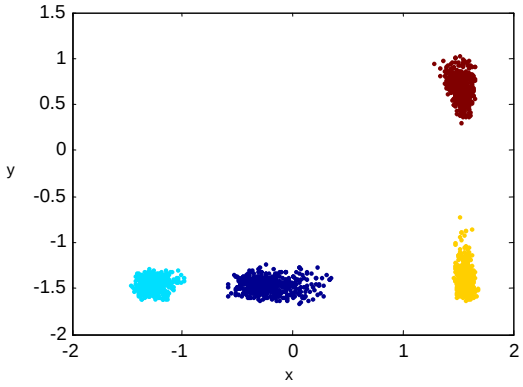


Figura 25: Projeção da base *Animals* obtida da rede MLP auto-associativa, com informação dos rótulos

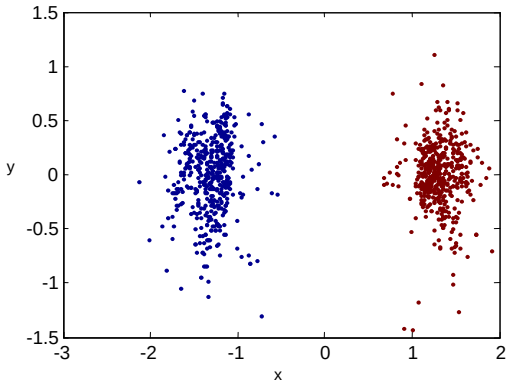


Figura 26: Projeção da base *Animals* obtida do método Isomap, com informação dos rótulos.

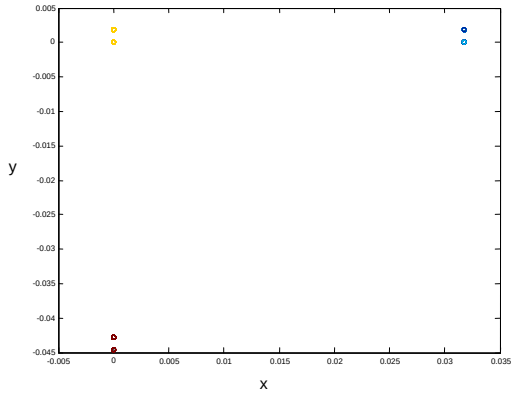


Figura 27: Projeção da base *Animals* obtida pelo método LLE, com informação dos rótulos.

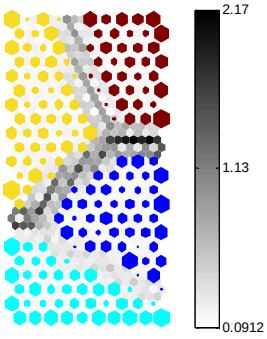


Figura 28: Projeção da *U-matrix* do mapa SOM treinado com a base *Animals*, acompanhada do histograma e informação dos rótulos.

A projeção de Sammon (figura 24) apresentou certa confusão entre as classes o que tornaria difícil a interpretação do gráfico monocromático.

As projeções do SOM, PCA e Rede Autoassociativa mostraram-se muito efetivas, distinguindo de forma inequívoca os quatro *clusters* presentes nos dados. Particularmente a Rede Autoassociativa produziu uma excelente visualização, mostrando com grande clareza a separação entre as classes.

5.1.4. Avaliação dos testes

Analisando as visualizações obtidas, percebe-se que, de uma maneira geral, a informação privilegiada sobre as classes permite uma visão inteiramente nova dos dados, dificilmente perceptível a partir apenas das projeções disponíveis.

Os métodos que apresentaram melhor conformidade com as informações das classes foram o SOM, no primeiro caso, SOM, Sammon, PCA e RN-AA no segundo e SOM, PCA e RN-AA no terceiro.

A diversidade de recursos visuais que podem ser obtidos a partir da *U-matrix* do SOM e o recurso da quantização que permite ajustar a granularidade do mapa buscando uma sintonia mais adequada a cada caso são características típicas das visualizações do SOM que o tornam particularmente útil para a pesquisa de agrupamentos.

A natureza qualitativa das avaliações não as invalida, uma vez que, nesse caso, os testes visaram justamente verificar a percepção intuitiva permitida pelos gráficos acerca da estrutura dos dados. Deve-se atentar para o fato que, numa situação de prospecção de uma base de dados com estrutura desconhecida, o primeiro aspecto importante da análise é a procura por tendências de agrupamentos que permitam identificar eventuais padrões existentes nos dados. A ferramenta mais adequada para "sentir" a presença de agrupamentos são as visualizações gráficas do tipo utilizado nesses testes. Elas constituem um importante elemento de julgamento e uma chave para a escolha dos métodos e algoritmos a serem empregados nas fases seguintes de um processo de mineração de dados [25, 26].

5.2. Testes quantitativos utilizando índices de manutenção de topologia

Para os testes de natureza quantitativa os índices descritos no item 3.2 foram aplicados às projeções bidimensionais fornecidas pelos diversos métodos. A seguir são apresentados os resultados dos testes. Para os dois primeiros casos (bases *Wine* e *Control)* foram mostradas as faixas iniciais do tamanho das vizinhanças (no máximo 12) por serem aquelas que apresentaram maior significado quanto à preservação da topologia. A partir do tamanho de vizinhança igual a 12, os valores dos índices apresentaram valores baixos e diferenças pouco significativas entre os diversos métodos. Para a base *Animals,* devido ao seu tamanho, a faixa considerada foi até o limite 50, pelos mesmos motivos.

Deve-se observar que, para os testes com as duas últimas bases, o Isomap foi descartado, pois perde o sentido mensurar a preservação topológica quando o método perdeu uma ou duas classes inteiras (itens 5.1.2 e 5.1.3). No caso da base *Wine*, foi perdida uma quantidade menor de pontos e foi possível fazer uma medida aproximada, descartando os pontos perdidos do conjunto de entrada para permitir a sua comparação com o conjunto de saída.

5.2.1. Testes com a base *Wine*

Nas figuras 29 e 30 são mostrados os resultados da aplicação dos índices, em função do tamanho da vizinhança, para todos os métodos, utilizando a base *Wine*.

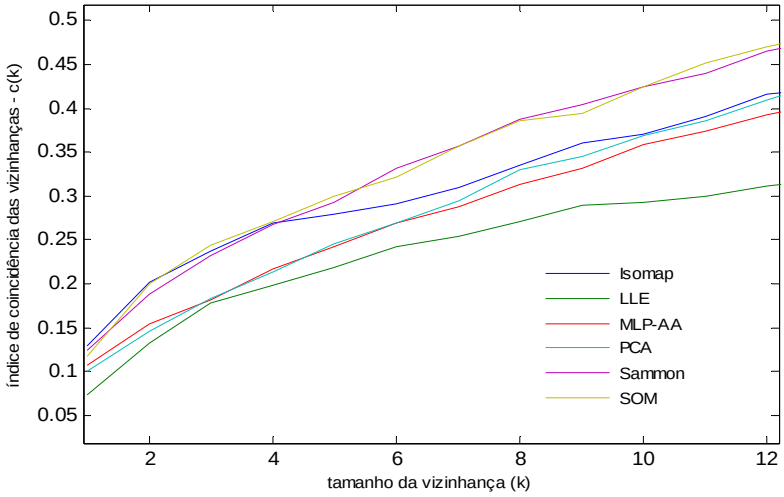


Figura 29: gráfico dos valores do índice de coincidência não ordenada, em função do tamanho da vizinhança, para todos os métodos, aplicados à base *Wine*.

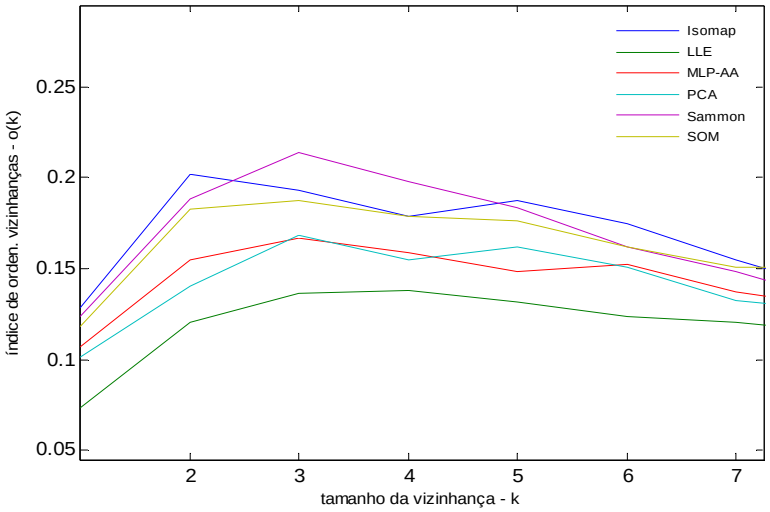


Figura 30: gráfico dos valores do índice de coincidência ordenada, em função do tamanho da vizinhança, para todos os métodos, aplicados à base *Wine*.

A observação dessa figuras demonstra a mesma tendência, tanto considerando o índice de vizinhança ordenado quanto o não ordenado. O desempenho dos métodos, segundo esses índices, mostra que os métodos Sammon, Isomap e SOM estão num grupo de melhor desempenho. Os métodos PCA e Rede Autoassociativa se situam num grupo intermediário e o método LLE teve um desempenho inferior. A maior diferença para os índices de coincidência ordenada, *o(k),* foi de 0.08 entre o Sammon e o LLE para uma vizinhança $k = 3$.

5.2.2. Testes com a base *Control*

Os resultados da utilização dos métodos para a base *Syntethic Control* são demonstrados pelo valor dos índices em função da vizinhança, nas figuras 31 e 32.

Considerando uma avaliação dos dois índices em conjunto, o SOM apresentou desempenho relativamente superior seguido pelos demais métodos com um desempenho aproximadamente equivalente.

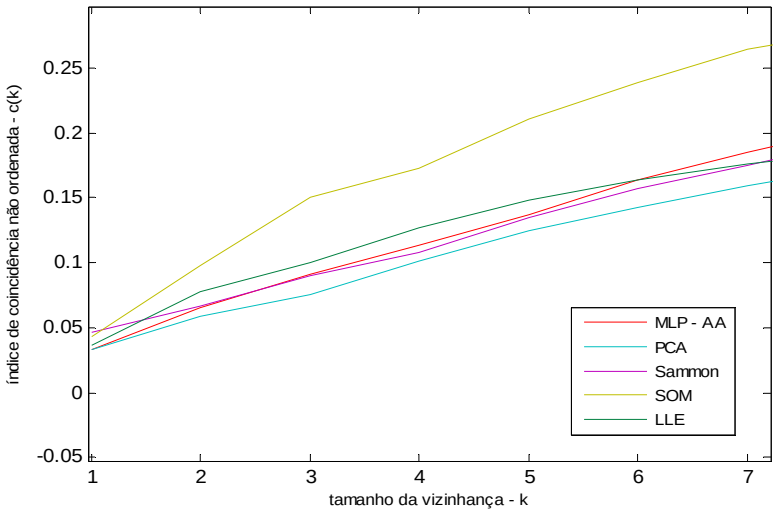


Figura 31: gráfico dos valores do índice de coincidência não ordenada, em função do tamanho da vizinhança para todos os métodos, aplicados à base *Synthetic Control.*

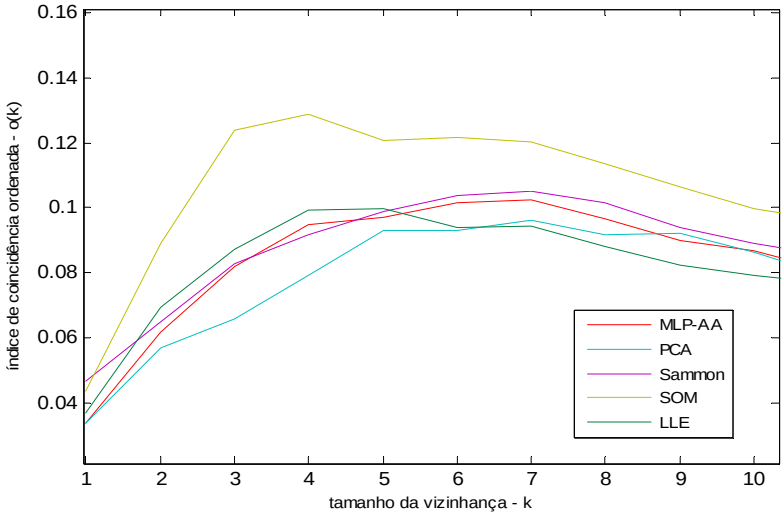


Figura 32: gráfico dos valores do índice de coincidência ordenada, em função do tamanho da vizinhança para todos os métodos, aplicados à base *Synthetic Control.*

5.2.2. Testes com a base *Animals*

A observação dos índices mostrados nas figuras 33 e 34, referentes à base *Animals,* demonstra que os métodos RNAA, SOM e PCA tiveram desempenho superior. Os métodos RN-AA e PCA superaram o SOM para pequenos valores de vizinhança (entre 1 e 2 para o caso do PCA e aproximadamente entre 1 e 5 para a RN-AA). No caso não ordenado o SOM os ultrapassou, a partir desses valores. No caso ordenado, o RNAA foi superior ao SOM, exceto na faixa de vizinhança entre 5 e 17, aproximadamente. O método LLE teve o pior desempenho. O método de Sammon também não mostrou bom desempenho relativo para essa base.

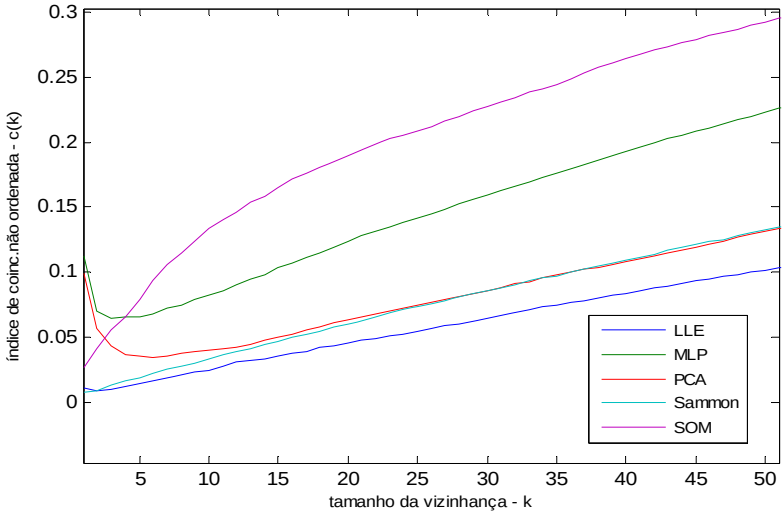


Figura 33: gráfico dos valores do índice de coincidência não ordenada, em função do tamanho da vizinhança para todos os métodos, aplicados à base *Animals*.

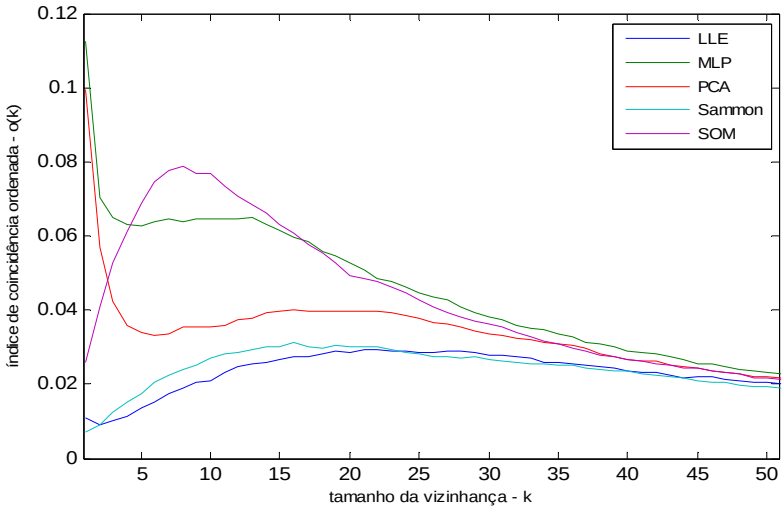


Figura 33: gráfico dos valores do índice de coincidência ordenada, em função do tamanho da vizinhança para todos os métodos, aplicados à base *Animals*.

5.2.3. Avaliação dos testes

Segundo os valores obtidos para os dois índices *o(t)* e *c(t)*, relativos aos diversos métodos, pode-se extrair alguns fatos válidos para as três bases de dados utilizadas. O Isomap foi descartado em termos absolutos devido às importantes perdas associadas. O LLE foi quase sempre francamente inferior para essas aplicações. O PCA teve desempenho mediano no primeiro caso. No segundo caso, o seu desempenho foi inferior, provavelmente devido a maiores não linearidades do conjunto de dados. No terceiro caso o desempenho foi mediano, embora tenha sido superior para pequenas vizinhanças de no máximo 3 pontos. O método Sammon teve desempenho alto no primeiro caso, mediano no segundo e inferior no terceiro. A Rede Autoassociativa apresentou um desempenho bastante estável, mediano nos dois primeiros casos e excelente no terceiro caso, em que pode se considerar, numa avaliação média, como o método que apresentou o melhor desempenho com a base *Animals*. O SOM apresentou o melhor desempenho nos dois primeiros casos e no terceiro, foi, ao lado da RNAA, superior aos demais. Os métodos Isomap e LLE, no geral mostraram fraco desempenho.

6. Conclusões

A visualização de dados é um recurso cada vez mais importante em mineração de dados, sobretudo na fase de percepção inicial da estrutura dos dados a serem trabalhados. A viabilidade do processo de visualização depende fortemente de métodos de redução da alta dimensionalidade de conjuntos complexos de dados.

Neste trabalho, seis diferentes métodos de redução de dimensionalidade, incluindo alguns métodos clássicos e outros, mais recentes, foram testados com três bases de dados de referência.

A avaliação do desempenho realizou-se segundo duas diferentes abordagens. A primeira delas, de caráter qualitativo, baseou-se na percepção visual da estrutura dos dados, através de visualizações gráficas bidimensionais dos dados projetados. A segunda abordagem baseou-se em medidas de manutenção de topologia, quantificadas por meio de dois índices relacionados: índices de coincidência ordenada e não ordenada de vizinhanças. Esses índices foram propostos neste trabalho, como uma tentativa de quantificar, mesmo que de forma aproximada, a preservação da ordem das vizinhanças, obtida por uma determinada projeção de dados multidimensionais. Os dois índices mostraram-se úteis, como elemento comparativo entre os métodos, embora os valores numéricos absolutos não tenham se mostrado muito significativos, quando vistos isoladamente. Provavelmente será possível melhorá-los, procurando estabelecer um fator de escala adequado.

Os métodos Isomap e LLE mostraram fragilidade nos tipos de testes realizados. De uma forma geral, os métodos que apresentaram melhor desempenho foram os métodos adaptativos (conexionistas), ou seja, Mapas Auto-organizados de Kohonen (SOM) e Redes Neurais Autoassociativas.
Naturalmente, essas conclusões devem ser claramente contextualizadas. Os resultados observados não determinam uma superioridade ou inferioridade absoluta de quaisquer métodos. Cada método apresenta vantagens e desvantagens, dependendo da aplicação que se tenha em mente com a redução da dimensionalidade. Por exemplo, vários experimentos mostraram as qualidades positivas dos métodos baseados em "descoberta" de *manifolds* (*manifold learning)*, Isomap e LLE, em aplicações que visem à recuperação de *manifolds* contínuos de baixa dimensionalidade intrínseca e grande densidade de pontos, imersos em espaços de maior dimensão formando geometrias complexas. Entretanto, no contexto enfocado neste trabalho, os métodos adaptativos, SOM e Rede Autoassociativa, mostraram-se mais interessantes para a visualização da estrutura de agrupamentos (*clusters*), em tarefas de prospecção das bases de dados utilizadas, cujas características comuns são: bases naturais com estrutura em clusters bem definidos (descontinuidades ou variações marcantes de densidade de pontos), contendo amostras em alta dimensão, de grandezas do mundo real.

Deve se levar em conta, ainda, que, ao se comparar algoritmos como o SOM com os demais métodos, haverá uma restrição relativa à comparação de elementos não perfeitamente homogêneos. Pois, enquanto os demais métodos realizam um mapeamento contínuo dos dados de entrada projetando-os no espaço de saída, o SOM realiza um mapeamento discreto, uma vez que o conjunto de dados de saída estará sempre sujeito à quantidade limitada de protótipos disponíveis para o mapa. Ainda assim, estabelecendo-se algumas considerações particulares, é possível obter-se um razoável efeito comparativo.

Futuros trabalhos envolverão a busca de aperfeiçoar os índices aqui utilizados e também a utilização de outros índices de preservação topológica que permitam expandir o número de critérios para aferir a qualidade das projeções. Deverão ser testadas outras bases de dados, tanto de natureza semelhante, para alargar a amostragem dos testes aqui apresentados, como de natureza diversa, a fim de testar outros diferentes domínios de aplicação. Também deverão ser acrescentados outros métodos e também variantes do SOM, como o ViSOM [11].

Agradecimento: Os autores agradecem ao CNPq, processos 480043/2008-6 e 201382/2008-3, e aos colegas Hujun Yin (The University of Manchester) e Anne M. P. Canuto (UFRN) pelas discussões durante o desenvolvimento do artigo.

Referências

[1] Yang, J., Ward, M.O., Rundensteiner, E.A., Huang S. **"**Visual Hierarchical Dimension Reduction for Exploration of High Dimensional Datasets", *Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization*, 2003.
[2] Jolliffe, J. *Principal Component Analysis*. Springer Verlag, 1986.
[3] Cox, T. F., Cox, M.A.A. *Multidimensional Scaling* /2nd ed, Chapman and Hall/CRC, 2001.

[4] Kramer, M.A. "Nonlinear principal component analysis using Auto-associative Neural Networks", *AIChE Journal,* Vol. 37, No. 2, 1991.
[5] Kohonen, T. *Self Organizing Maps*. 3ª. Ed. Springer-Verlag, Berlin, 2003.
[6]Tenenbaum, J. B., Silva, V., Langford, J. C. "A Global Geometric Framework for Nonlinear Dimensionality Reduction", *Science*, Vol 290, Pg. 2319-2323, 2000.
[7] Roweis, S.T., Saul, L.K., "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science*, Vol 290, Pg. 2323–2326, 2000.
[8] van der Maaten, L.J.P. *An Introduction to Dimensionality Reduction Using Matlab*, Report MICC 07-07, Universiteit Maastricht, 2007.
[9] van der Maaten, L.J.P., Postma, E.O. and van den Herik, H.J. "Dimensionality Reduction: A Comparative Review", (Sub.) *Neurocomputing*, 2008.
[10]Yin, H. "Nonlinear Dimensionality Reduction and Data Visualization: A Review", *International Journal of Automation and Computing*, N. 4 vol. 3, 294-303, 2007.
[11] *UCI Machine Learning Repository,* Irvine, CA, University of California. [www.ics.uci.edu/~mlearn/MLRepository.html].
[12] Haykin S. *Neural Networks, 2nd. Ed.*, Prentice Hall, 1999
[13] Sammon, J. W. "A Nonlinear Mapping for Data Structure Analysis", *IEEE Transactions on Computer*, vol. C-18, no.5, pp. 401/409, 1969.
[14] Fodor, I. K. *A Survey of Dimension Reduction Techniques,* Report, U.S. Department of Energy, 2002.
[15] Kaski, S., Nikkilä, J. and Kohonen, T, "Methods for exploratory cluster analysis." *Proc. of SSGRR',* L'Aquila, *2000*.
[16] Ultsch, A., "Self-Organizing Neural Networks for Visualization and Classification.", *Information and Classification.* Springer, Berlin, 307-313, 1993.
[17] Yin, H. "On multidimensional scaling and the embedding of self-organizing maps." *Neural Networks 21(2-3): 160-169*, 2008.
[18] Baldi, P., Hornik, K. "Neural networks and principal component analysis: learning from examples without local Minima." *Neural Networks.* **2:** 53–8, 1989.
[19] Cybenko, G. "Approximation by superpositions of a sigmoidal function" *Math. Control Signals Syst.* **2** 303–14, 1989.
[20] Gaetan, K., Golinval, J.C. "Feature extraction using auto-associative neural networks". *Smart Mater. Struct.* ***13,*** 211–219 . 2004.
[21] Saul, L. K., Roweis, S. T. "Think Globally**,** Fit Locally: Unsupervised Learning of Nonlinear Manifolds". *Journal of* Machine Learning Research **4**: 119-155, 2003.
[22] Ridder, D., Duin, R.P.W. *Locally linear embedding for classification*. Technical Report PH-2002-01, Pattern Recognition Group, Delft University of Technology, Delft, The Netherlands, 2002.
[23] Kruskal, J. B., Wish. M. *Multidimensional Scaling.* Sage Publications. Beverly Hills. CA, 1977.
[24] Roweis, S.T. and Saul, L.K. *Locally Linear Embedding Home Page.* http://www.cs.toronto.edu/~roweis/lle/
[25] Vesanto J., Alboniemi E. "Clustering of the Self-Organizing Map", *IEEE Transactions on Neural Networks,* **11,** (2): 586-600, 2000.
[26] Vesanto, J. "SOM-based data visualization methods," *Intell. Data Analysis,* **3** (2): 111–126, 1999.
[27] Balachander T, Kothari R, Cualing H. "An empirical comparison of dimensionality reduction techniques for pattern classification". *Artificial Neural Networks ICANN 97. Lecture Notes in Computer Science,* **1327**: 589-594, 1997.
[28] de Backer, S., Naud, A., Scheunders, P. "Nonlinear dimensionality reduction techniques for unsupervised feature extraction," *Pattern Recognition. Letters.* **19** (8): 711–720, 1998.
[29] Pearson, K. "On lines and planes of closest fit to systems of points in space", *Philosophical Magazine* **2**:559-572. 1901.
[30] Hotelling, H. "Analysis of a complex of statistical variables into principal components". *Journal of Educational Psychology*, 24:417-441 e 498-520, 1933.
[31] Vesanto, J. *et al* . *Somtoolbox for Matlab*, Report A 57, Helsinki University of Technology, 2000.
[32] Aeberhard, S., Coomans D., De Vel. O. "Comparative Analysis of Statistical Pattern Recognition Methods in High Dimensional Settings", *Pattern Recognition*, vol. 27(8): 1065-1077, 1994.
[33] Aeberhard, S., Coomans D., De Vel, O. "Improvements to the classification performance of RDA", *Journal of chemometrics*, vol. 7(2): 99-115, 1993.
[34] Alcock, R.J., Manolopoulos, Y. "Time-series similarity queries employing a feature-based approach", *Proc. Seventh Hellenic Conference on Informatics, Ioannina, Greece.* 1999.
[35] Pham , D.T., Chan, A.B. "Control chart pattern recognition using a new type of self organizing neural network", Proc. Inst. Mech. Eng. 212(2): 115-127, 1998.
[36] Gennari, J., Langley, P., Fisher, D. "Models of incremental concept formation", *Artificial Intelligence*, *40*, 11--62. 1989.

[37] Orlandic, R., Lai, Y., Yee, W. G. "Clustering high-dimensional data using an efficient and effective data space reduction", *Proc. 14th ACM international Conference on information and Knowledge Management* , Bremen, Germany, 201-208, 2005.

[38] Costa, J.A.F., Netto, M.L.A. "Estimating the Number of Clusters in Multivariate Data by Self-Organizing Maps". *International Journal of Neural Systems*, 9 (3): 195-202. 1999.

[39] Costa, J.A.F., Netto, M.L.A. "Clustering of complex shaped data sets via Kohonen maps and mathematical morphology". *Proc. SPIE*, Data Mining and Knowledge Discovery. B. Dasarathy (Ed.), 4384: 16-27, 2001.

[40] Costa, J.A.F., Netto, M.L.A. "Segmentação automática de mapas de Kohonen". *Congresso Brasileiro de Automática*, Natal, RN, , pp. 1607-1613, 2002.

[41] Costa, J.A.F., Netto, M.L.A. " Segmentação do SOM Baseada em Particionamento de Grafos". *Anais do VI Congresso Brasileiro de Redes Neurais*, São Paulo, 451-456 , 2003.

[42] Goncalves, M., Netto, M., Zullo, J., Costa, J.A.F. "Classificação não-supervisionada de imagens de sensores remotos utilizando redes neurais auto-organizáveis e métodos de agrupamentos hierárquicos". *Revista Brasileira de Cartografia (RBC)*, 60/1 : 17-29, 2008.

[43] Goncalves, M., Netto, M., Zullo, J., Costa, J.A.F. "A new method for unsupervised classification of remotely sensed images using Kohonen self-organizing maps and agglomerative hierarchical clustering methods". *Intl. Journal of Remote Sensing*, 29 (11): 3171 – 3207, 2008.

[44] Goncalves, M., Netto, M., Costa, J. A. F. "A Three-Stage Approach based on the Self-Organizing Map for Satellite Image Classification". *Lecture Notes in Computer Science (Artificial Neural Networks – ICANN 2007)* 4669: 680-689. 2007

[45] Costa, J.A.F., Netto, M.L.A. "Segmentação de Mapas Auto-Organizáveis com Espaço de Saída 3-D". *Controle & Automação - Ed. Especial Automação Inteligente,*18 (2) : 150-162, 2007.

[46] Costa, J.A.F., Netto, M.L.A. "Automatic Data Classification by a Hierarchy of Self-Organizing Maps". IEEE SMC'99 Conf. Proc. *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, pp. 419-24, Tokyo, *Japan. 1999*

[47] Costa, J.A.F. and Netto, M.L.A. "A new tree-structured self-organizing map for data analysis", *Proc. of the Intl. Joint Conf. on Neural Networks* ***(IEEE)***, Washington, DC, 1931-1936, 2001.

[48] Costa, J.A.F. *Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis*. Tese de Doutorado - Faculdade de Engenharia Elétrica e de Computação, Unicamp, 1999.

[49] Lemieux, S. "Data Minig" *Sec. 4.0*, *Canadian Genetic Disease Network / Canadian Bioinformatics Workshops.* 2004.

[50] Everitt B., *Cluster Analysis***,** Halsted Press, 1981.