

Um Algoritmo de Mapa Auto-Organizável por Lote Baseado em Distâncias Adaptativas

Luciano D. S. Pacífico¹, Francisco de A. T. de Carvalho¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Av. Jornalista Anibal Fernandes, s/n – Recife – PE – 50.740-560 – Brazil

{ldsp, fatc}@cin.ufpe.br

Abstract. *Clustering methods aims to organize a set of items into clusters such that items within a given cluster have a high degree of similarity, while items belonging to different clusters have a high degree of dissimilarity. The self-organizing map (SOM) is an unsupervised competitive learning neural network method which has both clustering and dimensionality reduction properties, using a neighborhood function to discover the topological structure hidden in the data set. In this paper, we introduce a batch self-organizing maps algorithm based on adaptive distances (ABSOM). Experimental results obtained in real benchmark datasets show the effectiveness of our approach.*

Resumo. *Métodos de agrupamento visam organizar um conjunto de itens em grupos de tal forma que itens de um dado grupo possuam um alto grau de similaridade, enquanto itens em grupos diferentes possuam um alto grau de dissimilaridade. O mapa auto-organizável (self-organizing map, SOM) é uma rede neural não-supervisionada de aprendizado competitivo que possui propriedades de agrupamento e de redução da dimensionalidade, usando uma função de vizinhança para descobrir a estrutura topológica escondida no conjunto de dados. Neste artigo, nós introduzimos um mapa auto-organizável por lote baseado em distâncias adaptativas (ABSOM). Resultados experimentais obtidos em bases de dados benchmark mostram a efetividade de nossa abordagem.*

1. Introdução

A análise de agrupamentos (*Clustering*) é uma tarefa bastante popular na descoberta de conhecimentos, possuindo aplicação em diversos campos como mineração de dados, reconhecimento de padrões, visão computacional, etc. Métodos de agrupamento visam a organização de um conjunto de itens em grupos de tal forma que itens de um dado grupo possuam um alto grau de similaridade, enquanto itens em grupos diferentes possuam um alto grau de dissimilaridade. As técnicas mais populares de agrupamento de dados são as hierárquicas e as de particionamento [Everitt 2001][Gordon 1999][Spaeth 1980].

Métodos hierárquicos resultam em uma hierarquia, ou seja, uma sequência aninhada de partições dos dados de entrada. Tais métodos podem ser de duas categorias: aglomerativos e divisivos. Métodos aglomerativos resultam em uma sequência de partições aninhadas, tendo início com cada item sendo ele mesmo um grupo unitário, e após aglomerações sucessivas todos os indivíduos se encontram em um mesmo grupo final. Os métodos divisivos têm início com todos os itens pertencendo a um único conjunto,

que sofre divisões sucessivas até que um critério de parada seja obtido (geralmente após a formação de uma partição com itens pertencentes a grupos unitários).

O particionamento de um conjunto de itens em um número de grupos pré-definido é um tópico importante em análise de dados, reconhecimento de padrões e processamento de imagens [Theodoris and Koutroubas 2006][Jain et al. 1999]. Algoritmos de particionamento buscam obter uma partição dos dados de entrada em um número fixo de grupos fornecido inicialmente pelo usuário do sistema. Tais métodos sempre buscam a otimização de uma função critério de adequação (geralmente de forma local). Para a melhora da qualidade dos agrupamentos finais, o algoritmo é executado várias vezes, partindo de pontos iniciais diferentes, sendo a melhor configuração obtida usada como resposta do algoritmo. Os algoritmos mais populares desta família são os algoritmos *K-means* [MacQueen 1967] e *Fuzzy C-means* [Bezdek 1981].

O Mapa Auto-Organizável de Kohonen (SOM) [Kohonen 1995] é uma rede neural não-supervisionada de aprendizado competitivo que possui propriedades de agrupamento e de visualização. Diferente do *K-means*, a rede SOM usa um conjunto de interações de vizinhança para aproximar a interação lateral neural e descobrir a estrutura topológica escondida nos dados, e além do vetor que obteve o melhor valor de similaridade (o vencedor), seus vizinhos também são atualizados, resultando em regiões nas quais neurônios em uma mesma vizinhança sejam bastante similares. Tal método também pode ser considerado como um algoritmo que mapeia dados de alta dimensionalidade espacial, \mathbb{R}^p , em um espaço de dimensionalidade reduzida, geralmente 1D, 2D ou 3D, chamado mapa. Essa projeção permite a partição das entradas em grupos "similares" enquanto preserva sua topologia.

Diversas variações do algoritmo SOM básico têm sido propostas visando sua melhoria. Em Kang *et al.* [J. A. Kangas and Laaksonen 1990] duas novas abordagens baseadas nas redes SOM foram propostas: uma baseada na ponderação dinâmica dos sinais de entrada de cada neurônio, melhorando a ordenação quando diferentes sinais de entrada eram usados; o segundo consistia na definição das vizinhanças no algoritmo de aprendizagem através do uso de uma árvore de caminhos mínimos, promovendo uma melhor e mais rápida aproximação de funções de densidade. Em Badran *et al.* [F. Badran and Thiria 2005] foi apresentado um algoritmo de mapa auto-organizável por lote similar ao algoritmo *K-means*, consistindo em uma abordagem em dois passos com uma afetação onde todas as instâncias da base de dados são afetadas ao neurônio mais próximo do mapa, e uma etapa de representação, onde todos os neurônios são atualizados.

Neste artigo o algoritmo de mapa auto-organizável por lote (BSOM) apresentado em [F. Badran and Thiria 2005] é estendido através de um algoritmo de mapa auto-organizável por lote adaptativo (ABSOM), que faz uso de distâncias euclidianas adaptativas parametrizadas por vetores de pesos nas variáveis que mudam a cada iteração do algoritmo e são diferentes para cada grupo. O cálculo desses vetores de pesos é inspirado no cálculo dos pesos para cada variável em cada grupo apresentado no algoritmo de agrupamento dinâmico baseado em distâncias adaptativas [Diday and Govaert 1977]. Os algoritmos de agrupamento dinâmicos tradicionais [Diday and Simon 1976] são algoritmos iterativos de realocação em duas etapas que envolvem a construção de novos agrupamentos a cada iteração e a identificação de representantes ou protótipos (médias,

conjuntos de regras, leis de probabilidade, funções de densidade de probabilidade, grupos de elementos, etc.) mais adequados para os novos grupos pela otimização local de uma função critério de adequação. Os métodos de agrupamento dinâmicos e adaptativos [Diday and Govaert 1977] também otimizam uma função baseado em um critério de adequação entre os grupos formados e seus protótipos, mas as distâncias para a comparação dos grupos e seus protótipos muda a cada iteração. Tais distâncias (representadas por vetores de pesos para as variáveis do problema) não são determinadas de forma absoluta e são diferentes para cada agrupamento. A vantagem dessas distâncias adaptativas está no fato de possibilitar ao algoritmo o reconhecimento de grupos de diferentes formas e tamanhos.

Como forma de avaliação da efetividade de nossa abordagem, experimentos foram realizados com a utilização de dados *benchmark* de classificação presentes no *UCI Machine Learning Repository* [Frank and Asuncion - Online].

Este artigo está organizado como se segue. A Seção 2 apresenta o mapa auto-organizável tradicional [Kohonen 1995]. O mapa auto-organizável por lote [F. Badran and Thiria 2005] é apresentado na Seção 3. Em seguida, a abordagem proposta de mapa auto-organizável por lote adaptativo é apresentado (Seção 4) e os resultados experimentais são discutidos (Seção 5). As conclusões serão apresentadas na Seção 6.

2. O Mapa Auto-Organizável Estocástico

A rede de mapas auto-organizável (SOM) estocástica [Kohonen 1995] é usada atualmente em diversos domínios e tem obtido sucesso em numerosas aplicações. É uma ferramenta muito popular para a visualização de dados de alta dimensionalidade espacial. De forma geral, pode-se considerar que a rede SOM executa a tarefa de agrupamento dos dados preservando a ordem espacial dos dados através da ordenação dos vetores protótipos (também chamados de centróides ou referências) em um espaço de saída de uma ou duas dimensões.

A rede SOM é formada por neurônios organizados em uma grade de baixa dimensionalidade, chamada mapa. Formalmente, o mapa é descrito por um grafo (C, Γ) . C é um conjunto de m neurônios interconectado tendo uma topologia discreta definida por Γ . Para cada par de neurônios (c, r) , $\delta(c, r)$ é definida como a função de distância entre os neurônios c e r no mapa. Essa distância impõe uma relação de vizinhança entre os neurônios. Seja $E = \{1, \dots, n\}$ o conjunto de entradas, onde cada entrada $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$ ($i = 1, \dots, n$) pertence a \mathbb{R}^p . Cada neurônio do mapa é representado por um protótipo $\mathbf{w}_c = (w_{c1}, \dots, w_{cp})$ ($c = 1, \dots, m$) que também pertence a \mathbb{R}^p . O algoritmo SOM estocástico executa os seguintes passos:

1) Inicialização

Fixe o número m de neurônios (grupos);

Fixe $\eta(0)$ (taxa de aprendizado inicial);

Fixe $h_{\delta(j,l)}(0)$ (função de vizinhança inicial, onde $\delta(j, l)$ é uma função de distâncias fixada entre neurônios j e l);

Fixe o número de iterações N_{iter} ;

Selecione aleatoriamente m protótipos distintos $\mathbf{w}_c^{(0)} \in E$ ($c = 1, \dots, m$);

Faça $t = 1$;

2) *Amostragem.*

Pegue um vetor de entrada aleatório $\mathbf{x}_i(t)$;

3) *Seleção.*

Encontre o melhor neurônio (vencedor) $\mathbf{w}_c(t)$ com distância euclidiana mínima em relação a $\mathbf{x}_i(t)$:

$$f(\mathbf{x}_i(t))^T = \min_{1 \leq j \leq m} \sum_{k=1}^p (x_{ik}(t) - w_{jk}(t))^2;$$

4) *Modificação dos pesos.*

Para todo neurônio j faça:

$$\begin{aligned} \mathbf{w}_j(t+1) &= \mathbf{w}_j(t) + \eta(t) h_{\delta(\mathbf{w}_c, \mathbf{w}_j)}(t) (\mathbf{x}_i(t) - \mathbf{w}_j(t)), j \in \mathbf{N}_c(t) \\ &= \mathbf{w}_j(t), j \notin \mathbf{N}_c(t) \end{aligned}$$

onde $\mathbf{N}_c(t)$ é vizinhança do neurônio c na iteração t ;

5) *Atualização.*

Atualize tanto a taxa de aprendizado $\eta(t)$ quanto a função de vizinhança $h_{l,j}(t)$; faça $t = t + 1$;

6) *Crítério de parada.*

Se $t = N_{Iter}$, pare; caso contrário, vá para o passo 2.

3. Algoritmo de Mapa Auto-Organizável por Lote (BSOM)

Esta seção apresenta o algoritmo de mapa auto-organizável por lote introduzido por [F. Badran and Thiria 2005]. Seja $E = \{1, \dots, n\}$ o conjunto de entradas, onde cada entrada $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})$ ($i = 1, \dots, n$) pertence a \mathbb{R}^p . Cada neurônio do mapa é representado por um protótipo $\mathbf{w}_c = (w_{c1}, \dots, w_{cp})$ ($c = 1, \dots, m$) que também pertence a \mathbb{R}^p .

O algoritmo SOM de treinamento por lote [F. Badran and Thiria 2005] é um algoritmo iterativo de duas etapas (afetação e representação, discutidos em seguida) onde todo o conjunto de dados (E) é apresentado ao mapa antes de algum ajuste ser feito. O algoritmo SOM por lote minimiza a seguinte função objetivo:

$$J = \sum_{i=1}^n \sum_{r=1}^m K^T(\delta(f^T(\mathbf{x}_i), r)) d^2(\mathbf{x}_i, \mathbf{w}_r) \quad (1)$$

onde f é a função de alocação e $f(\mathbf{x}_i)$ representa o neurônio do mapa correspondente ao indivíduo \mathbf{x}_i , e $\delta(f(\mathbf{x}_i), r)$ é a distância entre o neurônio r do mapa e o neurônio correspondente a \mathbf{x}_i . Assim, K^T , que é parametrizada por T (onde T representa a temperatura), é a função *kernel* de vizinhança que define a região de influência de cada neurônio r . Essa função objetivo é minimizada apenas para um valor de T fixo.

Essa função de custo é uma extensão da função de custo do *K-means*, onde a distância euclidiana é substituída por

$$d^T(\mathbf{x}_i, \mathbf{w}_{f(\mathbf{x}_i)}) = \sum_{r=1}^m K^T(\delta(f^T(\mathbf{x}_i), r)) d^2(\mathbf{x}_i, \mathbf{w}_r) \quad (2)$$

onde

$$d^2(\mathbf{x}_i, \mathbf{w}_r) = \sum_{j=1}^p (x_{ij} - w_{rj})^2 \quad (3)$$

é a distância euclidiana. Essa distância generalizada é uma soma ponderada das distâncias euclidianas entre \mathbf{x}_i e todos os vetores de referência da vizinhança do neurônio $f(\mathbf{x}_i)$. Todos os neurônios do mapa são levados em consideração.

Quanto T é fixo, a minimização de J é realizada em duas etapas iterativas: uma afetação e uma representação.

Durante a etapa de afetação, os vetores de referência (protótipos) permanecem fixos. A função de custo J é minimizada em relação à função de alocação e cada indivíduo \mathbf{x}_i é assinalado ao seu neurônio mais próximo:

$$c = f^T(\mathbf{x}_i) = \arg \min_{1 \leq r \leq m} d^T(\mathbf{x}_i, \mathbf{w}_r) \quad (4)$$

Durante a etapa de representação, a função de alocação é fixada. A função de custo J é minimizada em relação aos protótipos. O protótipo \mathbf{w}_c é atualizado para cada neurônio de acordo com:

$$\mathbf{w}_c = \frac{\sum_{i=1}^n K^T(\delta(f^T(\mathbf{x}_i), c)) \mathbf{x}_i}{\sum_{i=1}^n K^T(\delta(f^T(\mathbf{x}_i), c))}. \quad (5)$$

O algoritmo de mapa auto-organizável por lote (BSOM) [F. Badran and Thiria 2005] pode ser resumido como:

- 1) Inicialização
 - Fixe o número m de neurônios (grupos);
 - Fixe δ ; Fixe a função de *kernel* K^T ;
 - Fixe o número de iterações N_{iter} ;
 - Fixe T_{min}, T_{max} ; Faça $T \leftarrow T_{max}$; Faça $t \leftarrow 0$;
 - Selecione aleatoriamente m protótipos distintos $\mathbf{w}_c^{(0)} \in E$ ($c = 1, \dots, m$);
 - Inicialize o mapa $L(m, \mathbf{W}^0)$, onde $\mathbf{W}^0 = (\mathbf{w}_1^{(0)}, \dots, \mathbf{w}_m^{(0)})$;
 - Afete cada objeto \mathbf{x}_i ao neurônio mais próximo (grupo) do mesmo de acordo com a equação (4).
- 2) *Etapa 1: Representação.*
 - Faça $T = T_{max} \left(\frac{T_{min}}{T_{max}} \right)^{\frac{t}{N_{iter}-1}}$;
 - Mantenha a função de alocação fixa;
 - Calcule os protótipos $\mathbf{w}_c^{(t)}$ ($c = 1, \dots, m$) de acordo com a equação (5);
- 3) *Etapa 2: Afetação.*
 - Os protótipos $\mathbf{w}_c^{(t)}$ ($c = 1, \dots, m$) são fixados. Afete cada indivíduo \mathbf{x}_i ($i = 1, \dots, n$) ao neurônio mais próximo do mesmo de acordo com a equação (4);

4) *Critério de parada.*

se $T = T_{min}$, pare; caso contrário, faça $t = t + 1$ e vá para o passo 2 (Etapa 1).

4. Mapa Auto-Organizável por Lote Adaptativo (ABSOM)

Em [J. A. Kangas and Laaksonen 1990] foi apresentada uma versão adaptativa para o algoritmo SOM estocástico. Nesta seção apresentamos um algoritmo SOM por lote adaptativo que usa distâncias euclidianas parametrizadas por vetores de pesos para as variáveis que mudam a cada iteração do algoritmo e que são diferentes para cada um dos agrupamentos. O cálculo desses vetores de pesos neste algoritmo é inspirado na abordagem usada para calcular o peso de cada uma das variáveis em cada agrupamento do algoritmo de agrupamento dinâmico baseado em distâncias adaptativas [Diday and Govaert 1977].

O algoritmo de treinamento SOM por lote adaptativo introduzido neste artigo é um algoritmo iterativo que executa em três etapas (afetação, representação e ajuste dos pesos adaptativos) no qual todo o conjunto de dados de entrada (E) é apresentado ao mapa antes de qualquer ajuste ser feito.

A função de dissimilaridade geral entre um dado de entrada \mathbf{x}_i e um protótipo $\mathbf{w}_{f(\mathbf{x}_i)}$ é dada por:

$$d_{\Lambda}^T(\mathbf{x}_i, \mathbf{w}_{f(\mathbf{x}_i)}) = \sum_{r=1}^m K^T(\delta(f^T(\mathbf{x}_i), r)) d_{\lambda_r}^2(\mathbf{x}_i, \mathbf{w}_r) \quad (6)$$

onde

$$d_{\lambda_r}^2(\mathbf{x}_i, \mathbf{w}_r) = \sum_{j=1}^p \lambda_{rj} (x_{ij} - w_{rj})^2 \quad (7)$$

é uma distância euclidiana adaptativa parametrizada por vetores de pesos para as variáveis $\lambda_r = (\lambda_{r1}, \dots, \lambda_{rp})$ and $\Lambda = (\lambda_1, \dots, \lambda_m)$.

Note que os vetores de pesos λ_r ($r = 1, \dots, m$) mudam a cada iteração, ou seja, eles não são determinados de forma absoluta, e são diferentes para cada neurônio do mapa.

A função de custo do algoritmo SOM por lote adaptativo é dada por:

$$J = \sum_{i=1}^n \sum_{r=1}^m K^T(\delta(f^T(\mathbf{x}_i), r)) d_{\lambda_r}^2(\mathbf{x}_i, \mathbf{w}_r) \quad (8)$$

Quando T é fixo, a minimização de J é obtida de forma iterativa em três etapas: afetação, representação e ajuste dos pesos.

Durante a etapa de afetação, os vetores de referência (protótipos) e os vetores de pesos são fixados. A função de custo J é minimizada em respeito à função de alocação e cada indivíduo \mathbf{x}_i é assinalado ao neurônio mais próximo do mesmo:

$$c = f^T(\mathbf{x}_i) = \arg \min_{1 \leq r \leq m} d_{\Lambda}^T(\mathbf{x}_i, \mathbf{w}_r) \quad (9)$$

Durante a etapa de representação, a função de alocação e os vetores de pesos das variáveis são fixados. A função de custo J é minimizada em respeito aos protótipos. O protótipo \mathbf{w}_c é atualizado de acordo com a equação (5).

Durante a etapa de ajuste de pesos, os vetores de referência (protótipos) e a função de alocação são fixados. A função de custo J é minimizada em relação aos vetores de pesos. Os vetores de pesos $\boldsymbol{\lambda}_r = (\lambda_{r1}, \dots, \lambda_{rp})$ ($r = 1, \dots, m$), com $\lambda_{rj} > 0$ e $\prod_{j=1}^p \lambda_{rj} = 1$, têm seus pesos λ_{rj} ($j = 1, \dots, p$) calculados de acordo com a seguinte expressão:

$$\lambda_{rj} = \frac{\{\prod_{h=1}^p \sum_{i=1}^n K^T(\delta(f^T(\mathbf{x}_i), h))(x_{ij} - w_{hj})^2\}^{\frac{1}{p}}}{\sum_{i=1}^n K^T(\delta(f^T(x_i), r))(x_{ij} - w_{rj})^2} \quad (10)$$

O algoritmo de mapa auto-organizável por lote adaptativo (ABSOM) pode ser resumido como:

- 1) **Inicialização**
 Fixe o número m de neurônios (grupos);
 Fixe δ ; Fixe a função de kernel K^T ;
 Fixe o número de iterações N_{iter} ;
 Fixe T_{min}, T_{max} ; Faça $T \leftarrow T_{max}$; Faça $t \leftarrow 0$;
 Faça cada componente λ_{rj} ($r = 1, \dots, m; j = 1, \dots, p$) das matrizes de pesos adaptativas iguais a 1;
 Selecione aleatoriamente m distintos protótipos $\mathbf{w}_c^{(0)} \in E$ ($c = 1, \dots, m$);
 Inicialize o mapa $L(m, \mathbf{W}^0)$, onde $\mathbf{W}^0 = (\mathbf{w}_1^{(0)}, \dots, \mathbf{w}_m^{(0)})$;
 Assinale cada objeto \mathbf{x}_i ao neurônio (grupo) mais próximo do mesmo de acordo com a equação (4).
- 2) **Etapa 1: Representação.**
 Faça $T = T_{max} \left(\frac{T_{min}}{T_{max}} \right)^{\frac{t}{N_{iter}-1}}$;
 Mantenha a função de alocação fixa;
 Calcule os protótipos $\mathbf{w}_c^{(t)}$ ($c = 1, \dots, m$) de acordo com a equação (5);
- 3) **Etapa 2: Ajuste dos pesos adaptativos.**
 Os vetores de referência (protótipos) e a função de alocação são mantidos fixos.
 Calcule os componentes λ_{rj} dos vetores de pesos $\boldsymbol{\lambda}_r$ ($r = 1, \dots, m; j = 1, \dots, p$) de acordo com a equação (10);
- 4) **Etapa 3: Afetação.**
 Os protótipos $\mathbf{w}_c^{(t)}$ ($c = 1, \dots, m$) são fixados. Assinale cada indivíduo \mathbf{x}_i ($i = 1, \dots, n$) ao neurônio mais próximo do mesmo de acordo com a equação (9);
- 5) **Critério de parada.**
 Se $T = T_{min}$ pare; caso contrário, faça $t = t + 1$ e vá para o passo 2 (Etapa 1).

5. Resultados Experimentais

Para mostrar a utilidade do método desenvolvido, experimentos foram realizados comparando o algoritmo de treinamento baseado em redes SOM por lote (BSOM) [F. Badran and Thiria 2005] com o algoritmo de treinamento baseado em redes SOM por lote adaptativo (ABSOM) desenvolvido. Os algoritmos foram testados em bases de dados quantitativas reais, obtidas através do *UCI Machine Learning Repository*

[Frank and Asuncion - Online], sendo tais bases de diferentes níveis de dificuldades para a tarefa de agrupamento dos dados.

Como forma de comparação para os agrupamentos resultantes fornecidos pelos algoritmos considerados, um índice de avaliação externo, o Índice de Rand Corrigido (Corrected Rand Index, *CR*) [Hubert and Arabie 1985], a taxa global de erro de classificação (Overall Error Rate of Classification, *OERC*) [L. Breiman and Olshen 1984] e *F-Measure* [van Rijsbergen 1979] serão considerados.

O *CR* verifica o grau de concordância (similaridade) entre duas partições dos dados, uma fornecida a priori e outra obtida pelo algoritmo de agrupamento. O *CR* foi usado pelo fato de o mesmo não ser sensível ao número de grupos nas partições ou a distribuição de itens nesses grupos. O *CR* gera resultados no intervalo [-1, 1], no qual 1 significa concordância perfeita entre as duas partições avaliadas, enquanto valores próximos de 0 ou negativos indicam que o resultado foi obtido pelo acaso [Milligan 1996].

Como indicado em [F. Badran and Thiria 2005] a performance dos métodos avaliados dependerá fortemente dos parâmetros adotados para a minimização dos mesmos. Os parâmetros mais importantes a serem escolhidos são T_{max} , T_{min} , N_{iter} e a forma de decaimento da temperatura. A tabela 1 contém os parâmetros adotados para ambos o algoritmo baseado em redes SOM por lote tradicional e o algoritmo baseado em redes SOM por lote adaptativo. Tais valores foram determinados após diferentes testes com as abordagens adotadas, tendo sido os que obtiveram os melhores desempenhos médios.

Table 1. Lista de Parâmetros

Parâmetro	Valor
N_{Iter}	500
m	16 (grade 2×8)
T_{max}	3.5
T_{min}	0.5

Neste artigo a função δ adotada foi a distância euclidiana e a função de *kernel* de vizinhança escolhida foi

$$K^T(\delta(c, r)) = \exp\left(-\frac{(\delta(c, r))^2}{2T^2}\right)$$

5.1. Base de Dados Diabetes

Esta base de dados consiste em dois possíveis diagnósticos (classes) dados a uma amostra de 768 mulheres de 21 anos de idade de descendência indígena (*PimaIndians*), levando em consideração quando a paciente apresenta ou não sinais de diabetes de acordo com os critérios da Organização Mundial de Saúde (*WorldHealthOrganization*). As classes (1 para pacientes saudáveis, e 2 para pacientes interpretados como pacientes com "teste positivo para diabetes") possuem, respectivamente, 500 e 268 instâncias. Cada indivíduo é descrito por 8 variáveis reais [Frank and Asuncion - Online].

Os algoritmos de mapa auto-organizáveis por lote (tradicional e adaptativo) foram aplicados a esta base de dados. O *CR*, a *F-Measure* e o *OERC* foram, respectivamente,

0.049, 0.446 e 0.320 para o algoritmo de mapa auto-organizável por lote adaptativo. Para o algoritmo de mapa auto-organizável por lote de [F. Badran and Thiria 2005], tais índices foram, respectivamente, 0.012, 0.364 e 0.338. Para esta base de dados, o algoritmo de mapa auto-organizável por lote adaptativo obteve resultados superiores ao algoritmo de mapa auto-organizável por lote tradicional no que se refere aos índices avaliados.

5.2. Base de Dados de Identificação de Vidros

Esta base de dados consiste em um estudo sobre a classificação de tipos de vidro motivada pela investigação criminal. A base consiste em sete tipos diferentes de vidro (classes), embora não exista nenhum representante da classe 4, sendo cada instância (214 no total, sendo as classes não-balanceadas) deste conjunto descrita por nove variáveis do tipo real [Frank and Asuncion - Online].

Os algoritmos de mapa auto-organizáveis por lote (tradicional e adaptativo) foram aplicados a esta base de dados. O *CR*, a *F-Measure* e o *OERC* foram, respectivamente, 0.165, 0.429 e 0.383 para algoritmo de treinamento baseado em redes neurais SOM por lote adaptativo, e 0.223, 0.514 e 0.444 para o algoritmo de mapa auto-organizável por lote tradicional, respectivamente. Para esta base, o algoritmo de mapa auto-organizável por lote adaptativo foi superior ao algoritmo de mapa auto-organizável por lote apresentado em [F. Badran and Thiria 2005] no que diz respeito ao erro global de classificação (*OERC*).

5.3. Base de Dados Iris

Esta base de dados consiste em três tipos (classes) da planta iris, sendo cada classe representada por 50 instâncias, num total de 150 indivíduos. Cada indivíduo é composto por 4 variáveis do tipo real. Uma das classes é linearmente separável das outras duas, mas as últimas duas classes são não-linearmente separáveis uma da outra [Frank and Asuncion - Online].

Os algoritmos de mapa auto-organizáveis por lote (tradicional e adaptativo) foram aplicados a esta base de dados. O *CR*, a *F-Measure* e o *OERC* foram, respectivamente, 0.483, 0.599 and 0.08 para o mapa auto-organizável por lote adaptativo, e 0.372, 0.570 e 0.113 para o mapa auto-organizável por lote [F. Badran and Thiria 2005]. Para esta base de dados, o algoritmo de treinamento baseado em redes SOM por lote adaptativo obteve desempenho superior ao algoritmo de mapa auto-organizável por lote tradicional no que se refere aos três índices avaliados.

5.4. Base de Dados de Segmentação de Imagens

Esta base consiste em imagens que foram obtidas de forma aleatória a partir de sete imagens de ambientes externos. As imagens foram segmentadas manualmente para a criação de sete diferentes rótulos (classes). cada classe possui 330 instâncias (num total de 2310 objetos), sendo cada instância descrita por 18 atributos do tipo real [Frank and Asuncion - Online].

Os algoritmos de mapa auto-organizáveis por lote (tradicional e adaptativo) foram aplicados a esta base de dados. O *CR*, a *F-Measure* e o *OERC* foram, respectivamente, 0.391, 0.562 e 0.431 para o algoritmo de mapa auto-organizável por lote adaptativo, e foram, respectivamente, 0.105, 0.348 e 0.644 para o algoritmo de mapa auto-organizável

por lote [F. Badran and Thiria 2005]. Para esta base de dados, o algoritmo de mapa auto-organizável por lote adaptativo obteve resultados superiores ao algoritmo de mapa auto-organizável por lote tradicional no que se refere aos índices avaliados.

5.5. Base de Dados Tireóide

Esta base de dados consiste em três classes no que diz respeito ao estado da glândula tireóide. As classes (1, 2 e 3) possuem, respectivamente, 150, 35 e 30 instâncias (total de 215 instâncias). Cada instância é descrita por cinco variáveis do tipo real [Frank and Asuncion - Online].

Os algoritmos de mapa auto-organizáveis por lote (tradicional e adaptativo) foram aplicados a esta base de dados. O *CR*, a *F-Measure* e o *OERC* foram, respectivamente, 0.670, 0.844 e 0.102 para o algoritmo de mapa auto-organizável por lote adaptativo, enquanto o algoritmo de mapa auto-organizável por lote [F. Badran and Thiria 2005] obteve, respectivamente, 0.112, 0.390 e 0.139. Para esta base de dados, o algoritmo de mapa auto-organizável por lote adaptativo obteve resultados superiores ao algoritmo de mapa auto-organizável por lote tradicional no que se refere aos índices avaliados.

5.6. Base de Dados de Vinhos

Esta base de dados consiste em três tipos (classes) de vinhos produzidos na mesma região da Itália, mas derivados de três diferentes cultivos. As classes (1, 2 e 3) contém, respectivamente, 59, 71 e 48 instâncias (total de 178 instâncias). Cada instância é descrita por 13 variáveis quantitativas reais, que representam quantidades dos componentes encontrados nesses vinhos [Frank and Asuncion - Online].

Os algoritmos de mapa auto-organizáveis por lote (tradicional e adaptativo) foram aplicados a esta base de dados. O *CR*, a *F-Measure* e o *OERC* foram, respectivamente, 0.578, 0.709 e 0.067 para o algoritmo de mapa auto-organizável por lote adaptativo, e 0.267, 0.457 e 0.275, respectivamente, para o algoritmo de mapa auto-organizável por lote de [F. Badran and Thiria 2005]. Para esta base de dados, o algoritmo de mapa auto-organizável por lote adaptativo obteve resultados superiores ao algoritmo de mapa auto-organizável por lote tradicional no que se refere aos índices avaliados.

Podemos concluir que para a grande maioria das bases testadas o algoritmo de treinamento baseado em redes neurais SOM por lote adaptativo introduzido obteve melhores resultados no que diz respeito ao índice de Rand corrigido (*CR*), a *F-Measure* e ao erro global de classificação (*OERC*). No que diz respeito ao *OERC*, o método proposto obteve melhores resultados para todas as bases testadas, tendo sido inferior no que concerne ao *CR* e a *F-Measure* apenas no caso da base de dados de identificação de vidros (5.2).

6. Conclusões

Neste artigo, um algoritmo de mapa auto-organizável por lote adaptativo foi apresentado. Tal método combinou as características de visualização e de formação de agrupamentos das redes neurais SOM com a flexibilidade oferecida pelas distâncias adaptativas no reconhecimento de agrupamentos de tamanhos e formatos diferentes.

Experimentos realizados com bases de dados reais *benchmark* mostraram a utilidade da abordagem proposta. A tabela 2 apresenta os resultados obtidos pelo algoritmo

de mapa auto-organizável por lote adaptativo (ABSOM) proposto e pelo algoritmo de mapa auto-organizável por lote de [F. Badran and Thiria 2005] para cada base de dados testada, sendo os melhores resultados de cada índice destacados. A precisão dos resultados obtidos pelo método introduzido foi apresentada através do índice de Rand corrigido (*CR*), pelo erro global de classificação (*OERC*) e pela *F-Measure* considerando as bases de dados testadas. O método de mapa auto-organizável por lote adaptativo foi superior o método tradicional de mapa auto-organizável por lote para as bases de dados selecionadas do *UCI Machine Learning Repository* levando-se em consideração os critérios de avaliação adotados e na identificação das classes a priori dos padrões.

Table 2. Resultados obtidos para cada base de dados

Base de Dados	<i>CR</i>	<i>F-Measure</i>	<i>OERC</i>
Diabetes:			
ABSOM	0.049	0.446	0.320
BSOM	0.012	0.364	0.338
Identificação de Vidros:			
ABSOM	0.165	0.429	0.383
BSOM	0.223	0.514	0.444
Iris:			
ABSOM	0.483	0.599	0.08
BSOM	0.372	0.570	0.113
Segmentação de Imagens:			
ABSOM	0.391	0.562	0.431
BSOM	0.105	0.348	0.644
Tireóide:			
ABSOM	0.670	0.844	0.102
BSOM	0.112	0.390	0.139
Vinhos:			
ABSOM	0.578	0.709	0.067
BSOM	0.267	0.457	0.275

References

- Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press.
- Diday, E. and Govaert, G. (1977). Classification automatique avec distances adaptatives. *Informatique Comput. Sci.*, 11(4):329-349.
- Diday, E. and Simon, J. C. (1976). *Clustering Analysis*. Germany: Springer-Verlag.
- Everitt, B. (2001). *Cluster Analysis*. New York: Halsted.
- F. Badran, M. Y. and Thiria, S. (2005). *Self-Organizing Maps and Unsupervised Classification*. in: Neural Networks: methodology and applications, G. Dreyfus, Berlin, Germany: Springer-Verlag.
- Frank, A. and Asuncion, A. (2011). UCI Machine Learning Repository. Disponível: <http://archive.ics.uci.edu/ml>.

- Gordon, A. D. (1999). *Classification*. Boca Raton, FL: CRC Press.
- Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193-218.
- J. A. Kangas, T. K. K. and Laaksonen, J. T. (1990). Variants of self-organizing maps. *IEEE Transactions on Neural Networks*, 1(1):93-99.
- Jain, A., Murty, M., and Flynn, P. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31(3):264-323.
- Kohonen, T. (1995). *Self-Organisation Maps*. Berlin: Springer.
- L. Breiman, J. Friedman, C. J. S. and Olshen, R. A. (1984). *Classification and Regression Trees*. Chapman and Hall/CRC, Boca Raton.
- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proc. of the Fifth Berkeley Symposium on Math., Stat. and Prob.*, 1:281-296.
- Milligan, G. W. (1996). *Clustering Validation: Results and Implications for Applied Analysis*. Word Scientific.
- Spaeth, H. (1980). *Cluster Analysis Algorithms*. New York: Wiley.
- Theodoris, S. and Koutroumbas, K. (2006). *Pattern Recognition*. Academic Press, 3a edition.
- van Rijsbergen, C. J. (1979). *Information Retrieval*. London: Butterworths.