5CONTECSI June 04 - 06, 2008 São Paulo, Brazil

5th International Conference on Information Systems and Technology Management
5º Congresso Internacional de Gestão da Tecnologia e Sistema de Informação
De 04 a 06 de Junho de 2008 - São Paulo - Brasil

PS-1071

PARTSOM: AN ARCHITECTURE FOR DISTRIBUTED DATA CLUSTERING BASED ON MULTIPLE SELF-ORGANIZING MAPS

Flavius L. Gorgônio (Universidade Federal do Rio Grande do Norte, RN, Brasil) - flavius@dca.ufrn.br

Colaborador:
José Alfredo F. Costa (Universidade Federal do Rio Grande do Norte, RN, Brasil)
alfredo@dee.ufrn.br

Traditional methods for cluster analysis can be inappropriate to current necessities if not considering distributed data sets. Self-organizing maps (SOM) have been widely used as a software tool for visualization of high-dimensional data. Important characteristics of SOM include information compression and maintenance of data topology. This work presents strategies for efficient clustering analysis in geographically distributed databases using self-organizing maps. Local data sets (relative to vertical partitions of the database) are applied to distinct maps in order to obtain partial visions of the existing clusters. Further, units of each local map are chosen to represent original data and sent to a central site, which perform a joint of the partial results. Experimental results of the application of this strategy in different data sets are presented.

Keywords: Data mining, distributed data mining, cluster analysis, distributed data clustering, self-organizing maps.

PARTSOM: UMA ARQUITETURA PARA ANÁLISE DE AGRUPAMENTOS DISTRIBUÍDA BASEADA EM MÚLTIPLOS MAPAS AUTO-ORGANIZÁVEIS

Métodos tradicionais para análise de agrupamentos podem ser ineficientes para as necessidades atuais se não considerarem a possibilidade de dados armazenados de forma distribuída. Mapas auto-organizáveis (SOM) têm sido largamente utilizados na visualização de dados de alta dimensionalidade. Características importantes do SOM incluem a compressão de informação e a tentativa de manutenção da topologia dos dados. Este trabalho apresenta uma estratégia para análise de agrupamentos em bases de dados geograficamente distribuídas utilizando mapas auto-organizáveis. Conjuntos de dados locais, relativos a partições verticais da base de dados, são aplicados a diferentes mapas a fim de se obter visões parciais dos agrupamentos existentes. Posteriormente, representantes de cada mapa local são enviados à unidade central, que efetua uma fusão dos resultados parciais. São apresentados resultados experimentais da aplicação dessa estratégia em diferentes conjuntos de dados.

Palavras-chave: Mineração de dados, mineração de dados distribuída, análise de agrupamentos, análise de agrupamentos distribuída, mapas auto-organizáveis.

1. Introdução

Recentes mudanças ocorridas no cenário econômico mundial alteraram definitivamente a forma de atuação das empresas no mundo dos negócios. A globalização da economia, os avanços tecnológicos, a popularização da Internet, só para citar algumas dessas mudanças, encurtaram distâncias, acirraram a concorrência, colocaram frente a frente empresas grandes e pequenas, reais e virtuais, locais e remotas.

Neste mercado extremamente competitivo, em que se buscam alternativas que possam transformar-se em vantagens perante a concorrência, a tecnologia é uma arma de fundamental importância. O diferencial competitivo de uma empresa depende da sua capacidade de oferecer soluções para as demandas da sociedade. Atualmente, a forma mais eficiente de se conhecer estas demandas é utilizando os recursos da tecnologia da informação. Dentro desta estratégia de investir em tecnologia na busca de um diferencial nos negócios, o primeiro passo para se obter êxito em um processo de informatização é a implantação de um sistema de informação que possa gerenciar o fluxo de dados e informações dentro da empresa.

Os sistemas de informação são responsáveis pela coleta dos dados de entrada, pelo armazenamento desses dados em dispositivos adequados e pela sua manipulação e posterior disseminação, seja na forma bruta (dados) ou devidamente processada (informação), a fim de atender a uma necessidade específica. Assim, praticamente todas as empresas atuais, em qualquer segmento, possuem sistemas de informação com objetivo de armazenar dados decorrentes de suas transações comerciais.

Além disso, outros fatores contribuíram para o aumento do volume de dados nas organizações nos últimos anos, tais como a popularização dos computadores, a automatização do processo de coleta de dados e o barateamento dos dispositivos de armazenamento de dados. O surgimento da Internet também possibilitou a criação de bases de dados distribuídas em diferentes locais, tornando ainda mais difícil o processo de analisar os dados acumulados dentro das organizações.

O acúmulo de grandes volumes de dados requer a utilização de métodos e técnicas eficientes que permitam processá-los. Analisar e visualizar grandes volumes de dados na forma de registros, descritos por vários atributos e armazenados em um banco de dados é uma tarefa não-trivial, tanto em função do grande número de registros normalmente existentes nesses bancos de dados, como pela grande quantidade de informações presentes em cada registro. Assim, o uso de métodos estatísticos tem sido a maneira tradicionalmente utilizada para realizar essa tarefa.

Em sua forma mais convencional, um banco de dados é um conjunto de tabelas. Cada tabela é uma estrutura bidimensional de linhas e colunas, onde cada linha representa um registro do banco de dados e cada coluna representa um atributo associado àquele registro.

O processo de descoberta de conhecimento em bancos de dados (*knowledge discovery in database – KDD*) pode ser definido como um processo de várias etapas, não trivial, interativo e iterativo, para identificação de padrões compreensíveis, válidos, novos e potencialmente úteis a partir de grandes conjuntos de dados [Fayyad, 1996].

Por se tratar de um processo não trivial, KDD inclui um conjunto de métodos e técnicas que, uma vez combinados, possibilita extrair informações, na maioria das vezes ocultas em imensas montanhas de dados. A aplicação desses métodos e técnicas às bases de dados caracteriza uma das etapas do processo de KDD, conhecida como mineração de dados (*data mining*).

Mineração de dados pode ser vista como sendo uma etapa no processo de KDD que consiste na aplicação de algoritmos de análise e descoberta de padrões sobre os dados pré-processados, a fim de produzir conhecimento [Goldschmidt, 2005]. O principal objetivo da mineração de dados é descobrir relações entre os dados que não são visíveis através da utilização de consultas tradicionais em SQL nem com o auxílio de técnicas OLAP (*On-Line Analytical Processing*).

Os principais objetivos da etapa de mineração de dados são, basicamente, predição e descrição. O primeiro busca utilizar os dados disponíveis para criar um modelo que permita predizer sobre dados novos ou sobre variáveis não conhecidas, enquanto que o segundo busca explorar os dados disponíveis a fim de descobrir padrões que expliquem relações existentes entre os dados [Tan *et al.*, 2006].

Cada um desses objetivos pode ser atingido de diversas maneiras, dependendo do resultado que se deseja obter. Por exemplo, em tarefas de descrição, um dos métodos mais utilizados é a análise de agrupamentos, que pode ser definida como o processo de divisão de um conjunto de dados em grupos de objetos similares, onde cada grupo consiste de objetos semelhantes entre si e diferentes dos objetos dos outros grupos.

Existem diversos algoritmos desenvolvidos com o objetivo de analisar agrupamentos, baseados em diferentes estratégias, entre elas: agrupamento hierárquico, quantização vetorial, teoria dos grafos, lógica nebulosa (*fuzzy*), redes neurais artificiais, busca combinatória, etc. Um levantamento detalhado sobre diversos algoritmos para análise de agrupamentos pode ser obtido em [Xu and Wunsch, 2005].

Atualmente, a tarefa de análise de dados requer mecanismos que permitam lidar com bases de dados cada vez maiores e geograficamente distribuídas. Em virtude disso, diversos algoritmos têm surgido com o objetivo de minerar dados dispersos em várias locais, reunindo posteriormente, os resultados em um ponto central.

Este trabalho discute uma estratégia de aplicação de um algoritmo, baseado em redes neurais, para análise de agrupamentos em bases de dados distribuídas. O foco principal do artigo é mostrar que em situações onde o volume de dados é muito grande ou questões relacionadas à segurança dos dados impossibilitam sua consolidação em um único local, os resultados obtidos com a aplicação desse método justificam sua utilização, mesmo que sejam relativamente inferiores às abordagens tradicionais, onde todo o conjunto de dados é analisado.

O resto do artigo está organizado da seguinte forma: a seção 2 apresenta uma revisão bibliográfica acerca de algoritmos que tratam de mineração de dados distribuída, além de apresentar alguns algoritmos que demonstram a validade de realizar análise de agrupamentos sobre um subconjunto dos dados. A seção 3 discute o funcionamento dos mapas auto-organizáveis, mostrado sua aplicação em tarefas de análise de agrupamento. A seção 4 apresenta o algoritmo sugerido, detalhando seu funcionamento e as vantagens obtidas com a sua utilização. A seção 5 mostra a aplicação do algoritmo a alguns conjuntos de dados, comparando-os com os resultados obtidos com a aplicação de métodos tradicionais. Por fim, a seção 6 apresenta as conclusões e a seção 7 discute futuras modificações no algoritmo proposto.

2. Revisão Bibliográfica

Descoberta de conhecimento em bases de dados distribuídas (*distributed knowledge discovery*), descoberta de conhecimento em paralelo (*parallel knowledge discovery*) e mineração de dados distribuída (*distributed data mining*) são diferentes termos para

descrever algoritmos que buscam basicamente os mesmos resultados: obter conhecimento a partir de um conjunto de bases de dados distribuídas. Uma coletânea de trabalhos sobre o tema é mantida em [Bhaduri, 2006].

Algoritmos de análise de agrupamentos realizam comparações entre objetos de uma base de dados a fim de identificar semelhanças entre eles, quanto maior o número de objetos e quando mais atributos esses objetos possuírem, maior será a quantidade de comparações a ser realizada. Assim, a complexidade de um processo de análise de agrupamentos é diretamente proporcional à quantidade de dados que o algoritmo manipula.

Portanto, buscar agrupamentos em bases de dados de alta dimensionalidade é uma tarefa não-trivial. O aumento no número de atributos de cada entrada não apenas influencia negativamente no tempo de processamento do algoritmo, como também dificulta o processo de identificação dos agrupamentos.

Determinadas aplicações atuais possuem bases de dados tão volumosas que não é possível mantê-las integralmente na memória principal, mesmo utilizando máquinas robustas. A seguir, são apresentadas três abordagens para resolver esse problema [Kantardzic, 2003]:

a) Armazenam-se os dados em memória secundária e agrupa-se subconjuntos dos dados isoladamente, reunindo-se os resultados em uma etapa posterior para agrupar o conjunto inteiro;

b) Utiliza-se um algoritmo de agrupamento incremental, onde cada elemento é trazido individualmente para a memória principal e associado a um dos grupos existentes ou alocado em um novo grupo;

c) Utiliza-se uma implementação paralela, onde vários algoritmos trabalham simultaneamente sobre os dados armazenados, aumentando a eficiência.

Nos casos em que o conjunto de dados precisa ser dividido em subconjuntos, duas abordagens são normalmente utilizadas, conforme pode ser visto na Figura 1. A primeira e mais utilizada é dividir horizontalmente a base de dados, criando subconjuntos homogêneos dos dados, de forma que cada algoritmo opere sobre as mesmas variáveis, apenas se tratando de registros diferentes. Outra abordagem é dividir verticalmente a base de dados, criando subconjuntos heterogêneos dos dados, nesse caso cada algoritmo opera sobre os mesmos registros, mas tratando de atributos diferentes.

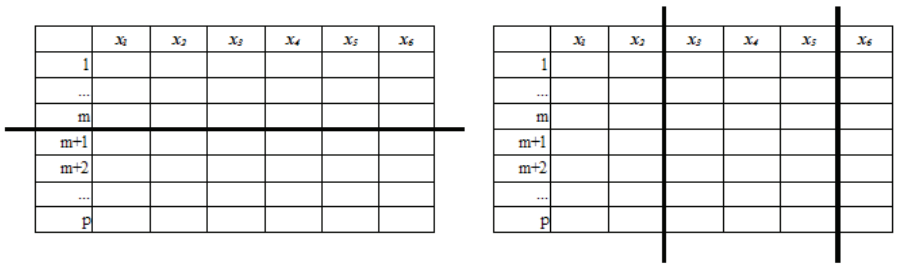

Figura 1. Particionamento horizontal e vertical de dados em uma tabela

A integração de diversas bases de dados em um único local é desaconselhada em se tratando de bases de dados muito volumosas [Forman and Zhang, 2000]. Se uma organização possui grandes bases de dados dispersas e precisa reunir todos os dados a fim de aplicar sobre elas os algoritmos de mineração de dados, esse processo pode exigir grandes transferências de dados, o que poderá ser lento e dispendioso. Além disso, qualquer mudança que ocorra nos dados distribuídos, como por exemplo, a inclusão de novas informações ou alterações daquelas já existentes terá que ser atualizada junto à base

de dados central. Isso exige uma política complexa de atualização de dados, com sobrecarga de transferência de informações dentro do sistema.

Em um dos primeiros trabalhos sobre análise de agrupamentos distribuída, [Forman and Zhang, 2000] apresenta uma técnica que paraleliza diversos algoritmos a fim de obter maior eficiência no processo de mineração de múltiplas bases de dados distribuídas. Os autores reforçam a necessidade de preocupação em relação a reduzir sobrecarga de comunicação entre as bases, diminuir o tempo de processamento e minimizar a necessidade de máquinas poderosas e com capacidades de armazenamento estendidas.

Outros fatores que motivam a existência de bases de dados distribuídas estão relacionados a questões da segurança e privacidade dos dados, conforme aborda [Chak-Man *et al.*, 2004]. Muitas organizações mantêm bases de dados geograficamente distribuídas como uma forma de aumentar a segurança das suas informações. Dessa forma, se por acaso uma das políticas de segurança falha, o invasor tem acesso apenas a uma parte das informações existentes.

Em bases de dados que reúnem um grande número de atributos, uma outra abordagem algumas vezes utilizada é realizar a análise a partir de um subconjunto dos atributos, ao invés de considerar todas as variáveis simultaneamente. Uma dificuldade óbvia dessa abordagem é identificar quais variáveis são mais importantes no processo de identificação dos grupos e a utilização de métodos estatísticos, tais como Análise de Componentes Principais (PCA) e Análise Fatorial, é empregada com freqüência [Hair Jr. *et al.*, 2005].

Dentro dessa proposta, [Friedman and Meulman, 2004] apresenta um algoritmo denominado COSA (*Clustering Objects on Subsets of Attributes*) que não apenas realiza análise de agrupamentos a partir de um subconjunto dos atributos, mas identifica quais variáveis são mais importantes em cada grupo identificado. Em um trabalho mais recente, [Damian *et al.*, 2007] demonstram a aplicabilidade do algoritmo COSA na análise de sistemas biológicos.

[Laine, 2002] demonstra como o usuário pode identificar quais as variáveis mais importantes em um processo de mineração de dados em particular, utilizando mapas auto-organizáveis para a tarefa de análise de agrupamentos. Para isso, o autor descreve um método que analisa o mapa SOM treinado e identifica o conjunto de variáveis que melhor separa os grupos.

[He *et al.*, 2005] analisa a influência dos tipos de dados no processo de agrupamento e propõe dividir o conjunto de atributos em dois subconjuntos, um apenas com os atributos numéricos e outro apenas com os atributos categóricos. Em seguida, propõe o agrupamento de cada um dos subconjuntos isoladamente, utilizando algoritmos apropriados para cada um dos tipos. No final, os resultados de cada um dos agrupamentos são combinados em uma nova base de dados que é mais uma vez submetida a um algoritmo de agrupamento para dados categóricos.

[Kargupta *et al.*, 2001] apresenta uma técnica chamada Análise Coletiva de Componentes Principais (CPCA – *Collective Principal Component Analysis*) para análise de agrupamentos de dados heterogêneos de alta dimensionalidade. Segundo os autores, não existe nenhum técnica que efetue análise de componentes principais (PCA) distribuída, a partir de conjuntos de dados heterogêneos e a técnica que eles apresentam é uma solução para esse problema. Nesse trabalho, os autores demonstram preocupação em reduzir as taxas de transferências de dados em um ambiente de dados distribuídos.

Este trabalho tem como objetivo apresentar uma arquitetura baseada em redes neurais auto-organizáveis (*self-organizing maps*) que permita auxiliar no processo de extração de informações de grandes volumes de dados, sendo menos sensível aos problemas enfrentados pelas abordagens tradicionais, onde os dados são agrupados em um único local antes da aplicação dos algoritmos de mineração de dados.

3. Mapas Auto-Organizáveis

Redes neurais artificiais (RNA) são uma importante ferramenta computacional, com forte inspiração neurobiológica e largamente utilizada na solução de problemas complexos, que não podem ser tratados por soluções algorítmicas tradicionais [Haykin, 2001]. Aplicações para RNA incluem reconhecimento de padrões, análise e processamento de sinais, tarefas de análise, diagnóstico e prognóstico, classificação e agrupamento de dados.

Dentre os inúmeros modelos de redes neurais existentes, um deles, em particular, tem sido amplamente utilizado em tarefas de classificação automática de dados, visualização de dados de dimensão elevada e na redução de dimensionalidade: os mapas auto-organizáveis (*self-organizing maps* – SOM) [Kohonen, 2001]. As redes neurais SOM, também chamados de mapas de Kohonen, constituem uma classe de rede neural, de aprendizado não supervisionado, conhecidas como redes competitivas. Neste tipo de rede, todos os neurônios (unidades básicas de processamento da rede) recebem o mesmo estímulo de entrada e competem entre si para identificar que é o vencedor.

Uma importante característica das RNA do tipo SOM é a compressão de informação, enquanto mantêm preservadas relações topológicas e métricas existentes nos dados de entrada. Isso significa que elementos do conjunto de entrada que possuam características semelhantes tendem a permanecer juntos quando projetados na camada de saída da rede.

A arquitetura de uma rede neural do tipo SOM é extremamente simples, consistindo apenas de duas camadas de neurônios (Figura 2). A primeira camada de entrada, composta por um vetor com p neurônios, representa a dimensionalidade do conjunto de entrada (ou seja, a quantidade de atributos da tabela de dados). Cada neurônio de entrada está conectado a todos os neurônios da camada seguinte. A segunda camada, também conhecida como camada de saída, representa o mapa onde o conjunto de entrada será projetado, sendo composta por um conjunto de neurônio, normalmente dispostos na forma de um vetor (unidimensional) ou de uma matriz (bidimensional), onde cada neurônio está conectado apenas aos seus vizinhos.

Durante a etapa de treinamento de uma rede neural do tipo SOM, um representante do conjunto de entrada é selecionado aleatoriamente e apresentado à camada de entrada da rede. Uma função de ativação calcula a semelhança entre o vetor de entrada e todos os neurônios do mapa. O neurônio da camada de saída que se apresentar como mais similar ao neurônio de entrada é declarado vencedor e os seus pesos sinápticos, assim como os dos seus vizinhos são realçados. O processo se repete com outros vetores do conjunto de entrada até que a rede esteja treinada.

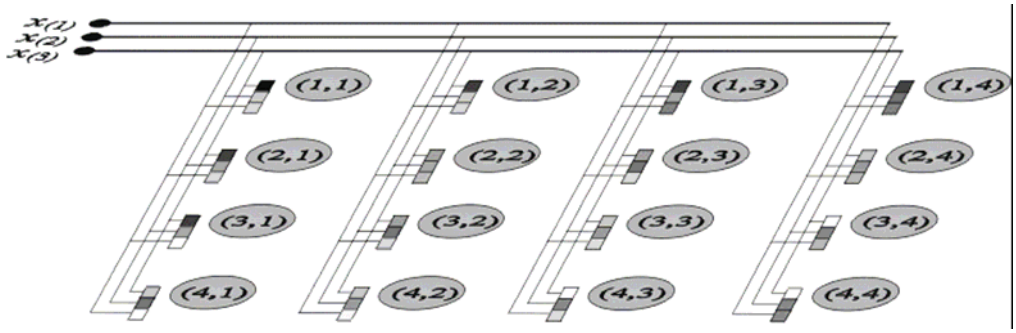

Figura 2. Arquitetura de uma rede neural do tipo SOM

A função de ativação comumente utilizada para calcular a distância entre o vetor de entrada e os neurônios da rede é a distância euclidiana, dada pela equação

$$\| \mathrm{m_i} - \mathrm{x_k} \| = \sum_{j=1}^{p} \left[x_{kj}(t) - m_{ij}(t) \right]^2 \quad (1)$$

Onde ||.|| representa a medida de distância, *x* representa um vetor qualquer do conjunto de entrada e *m* representa um vetor da camada de saída.

Para identificação do neurônio vencedor, verifica-se dentre todos os neurônios da camada de saída, qual deles possui a menor distância para o vetor de entrada, usando-se a equação (2), onde *c* representa o neurônio vencedor, *x* representa um vetor qualquer do conjunto de entrada e *m* representa um vetor da camada de saída:

$$\| \boldsymbol{x} - \boldsymbol{m}_c \| = \min_i \{ \| \boldsymbol{x} - \boldsymbol{m}_i \| \} \quad (2)$$

Os pesos sinápticos do neurônio vencedor e da sua vizinhança são atualizados através da equação (3), onde *t* representa o tempo, *x(t)* representa um vetor aleatório do conjunto de entrada e h_{ci} determina o raio de vizinhança que será modificado, normalmente sendo reduzido na medida em que o algoritmo de treinamento avança.

$$m_i(t+1) = m_i(t) + h_{ci}(t).[x(t) - m_i(t)] \quad (3)$$

Em uma tarefa de identificação de padrões, uma das principais aplicações para este tipo de rede, ser o neurônio vencedor significa ser o mais semelhante, dentre os existentes no mapa de saída, ao valor apresentado à entrada da rede. O neurônio vencedor, juntamente com a sua vizinhança, tem seus valores realçados, de forma que se a mesma entrada for apresentada à rede posteriormente, aquela região do mapa irá ficar ainda mais realçada.

Uma vez concluído o processo de treinamento, a rede SOM pode ser utilizada para classificar padrões e agrupá-los segundo suas características presentes no espaço de entrada. Uma vez que a rede associa cada padrão apresentado à camada de entrada a um neurônio da camada de saída, pode-se usar esse mapeamento para agrupar dados semelhantes.

As redes do tipo SOM possuem a habilidade de formar um mapa topográfico dos padrões de entrada, de forma que a disposição dos neurônios na grade reflete características estatísticas contidas nos padrões de entrada. Com isso, as redes SOM preservam a topologia entre os espaços de entrada e saída, permitindo que se enxerguem padrões de comportamentos que seriam invisíveis em um ambiente *p*-dimensional [Costa, 1999]. Outras propriedades da rede SOM podem ser resumidas como descrito a seguir [Haykin, 2001; Gonçalves *et al.*, 2007a]:

a) Aproximação do espaço de entrada: o SOM tem como objetivo básico armazenar um conjunto grande de vetores de entrada encontrando um conjunto menor de protótipos (vetores de pesos sinápticos w_i) de modo a fornecer uma boa aproximação para o espaço de entrada original. A base teórica dessa estratégia está fundamentada na teoria da quantização vetorial, cuja motivação é a redução de dimensionalidade ou a compressão de dados;

b) Ordenação Topológica: ao realizar o mapeamento não-linear dos vetores de entrada para o arranjo de neurônios da rede, o algoritmo do SOM tenta preservar ao máximo a topologia do espaço original, ou seja, procura fazer com que neurônios vizinhos no espaço de saída apresentem vetores de pesos que representem padrões vizinhos no espaço de entrada;

c) Casamento de Densidade: o mapeamento efetuado pelo SOM reflete a distribuição de probabilidade dos dados no espaço de entrada original. Regiões do espaço de entrada de onde os vetores de amostra x são retirados com uma alta probabilidade de ocorrência são mapeadas para domínios maiores no espaço de saída e, portanto, com melhor resolução que regiões no espaço de entrada de onde vetores de amostra x são retirados com uma baixa probabilidade de ocorrência.

Uma forma comumente utilizada para visualizar os resultados obtidos em um processo de análise de agrupamentos que utiliza o algoritmo SOM é através da matriz de distâncias unificadas [Ultsch, 1993], ou matriz-U, como é mais conhecida. A matriz-U representa distâncias entre os neurônios vizinhos do vetor de referência através de uma escala de níveis de cinza, de forma que neurônios próximos (semelhantes entre si) são representados por cores claras, enquanto neurônios distantes são representados por cores escuras. Assim, é possível identificar os grupos detectados pelo algoritmo.

Métodos de segmentação automática de mapas neurais, que possibilitam melhor interpretação e análise do SOM, foram propostos por um dos autores envolvendo técnicas diferentes, como segmentação morfológica da matriz-U [Costa, 1999; Costa e Netto, 1999], segmentação utilizando técnicas de particionamento de grafos [Costa e Netto, 2003] e segmentação do SOM por métodos de agrupamentos hierárquicos com conectividade restrita [Costa, 2005; Gonçalves *et al.*, 2007b].

Métodos hierárquicos também têm sido abordados para particionamento recursivo de bases de dados e aplicações como compressão de imagens [Costa *et al.*, 2003; Costa e Netto, 2001a,b]. Mais recentemente, [Gonçalves *et al.*, 2005, 2006) apresentou uma otimização do método proposto por [Costa e Netto, 1999] incorporando índices de validação de agrupamentos CDbw [Halkidi e Vazirgiannis, 2002], que trata de maneira eficiente agrupamentos com formatos arbitrários. Extensões do método para uso com agrupamento hierárquico do mapa de Kohonen aplicado em classificação de imagens de satélite foram apresentadas em [Gonçalves *et al.*, 2007a, b].

Ainda, em casos onde ocorrem perdas significativa de topologia no mapeamento de espaços de entrada de elevada dimensão para redes SOM, os autores desenvolveram métodos de segmentação de mapas com espaço de saída 3-D, que possibilitam menor distorção e melhor análise [Costa, 1999; Costa e Netto, 2007].

4. Metodologia Proposta

Algoritmos que realizam análise de agrupamentos distribuída, normalmente a fazem em duas etapas. Em um primeiro momento, os dados são analisados localmente em cada uma das unidades que fazem parte da base de dados distribuída. Em uma segunda etapa, uma unidade central reúne os resultados parciais e os combina para obter um resultado global.

Esta seção apresenta uma estratégia para agrupar objetos semelhantes em bases de dados distribuídas, utilizando mapas auto-organizáveis. O processo também é realizado em duas etapas: na primeira etapa, o algoritmo SOM é aplicado localmente em cada uma das bases distribuídas. Na segunda etapa, o SOM é novamente aplicado, desta vez sobre os representantes de cada uma das bases distribuídas, para criar um resultado definitivo. Uma visão geral da arquitetura partSOM pode ser observada na Figura 3, onde são apresentadas todas as etapas do processo. Um resumo do algoritmo é apresentado a seguir, sendo discutida cada uma das suas etapas:

Algoritmo partSOM

1. Cada unidade local aplica o algoritmo SOM sobre seus dados, obtendo como resposta o mapa treinado e seu vetor referência (*codebook*);
2. Cada unidade local projeta seus dados sobre o seu vetor referência, obtendo como resposta o índice desse vetor que melhor representa cada uma das instâncias da base;
3. Os índices e o vetor referência de cada unidade local são enviados à unidade central;
4. A unidade central remonta as bases remotas a partir dos índices e vetores referência recebidos;
5. A unidade central aplica o algoritmo SOM sobre os representantes das bases locais, obtendo resultados bem próximos do que seria obtido com a análise dos dados originais.

No passo 1, cada unidade local aplica o algoritmo SOM sobre seus dados e o resultado dessa fase é um conjunto de mapas treinados. Como os mapas são treinados individualmente, a partir do seu subconjunto de atributos, cada um deles irá capturar a topologia do seu espaço de entrada, armazenando-a em seu vetor referência. O vetor referência (ou *codebook*), apesar de possuir apenas um subconjunto dos dados originais, mantém as características existentes dos dados de entrada.

No passo 2, cada elemento de cada um dos subconjuntos de entrada é apresentado ao seu respectivo mapa, com o objetivo de identificar qual dentre os neurônios do mapa (vetor referência) é o mais semelhante a ele. O índice do neurônio mais semelhante (*best match unit – bmu*) é armazenado em um vetor de índices, que possui o mesmo número de linhas do conjunto de entrada, mas apenas uma coluna.

No passo 3, cada uma das unidades locais envia o vetor de índices e o seu respectivo mapa à unidade central. Esta é a principal vantagem do algoritmo em relação aos métodos tradicionais, ao invés de enviar todos os subconjuntos de dados à unidade central, apenas os seus representantes são enviados, diminuindo-se sensivelmente o fluxo de dados entre as unidades remotas e a unidade central.

No passo 4 é realizado o procedimento inverso, com o objetivo de remontar a base de dados a partir dos valores recebidos. Cada elemento de cada um dos vetores de índice é substituído pelo equivalente presente no seu mapa SOM correspondente, a fim de

reconstruir uma tabela com valores similares àqueles existentes no subconjunto de entrada. Os dados são então justapostos em uma nova tabela, respeitando-se a mesma ordem em que estavam em suas bases de dados originais.

A expectativa é de que o resultado obtido nessa etapa possa ser generalizado como sendo equivalente ao conjunto de dados original. Os dados obtidos após a etapa 4 (e que servirão de entrada na etapa 5) correspondem a valores similares aos originais, uma vez que os neurônios enviados no vetor referência são representantes próximos dos valores encontrados em cada um dos conjuntos de entrada.

Finalmente, no passo 5 é realizada uma nova segmentação, dessa vez sobre o dados reconstruídos a partir dos representantes recebidos. Um novo SOM é treinado e o resultado obtido nessa segmentação é muito próximo ao que seria obtido com os dados originais.

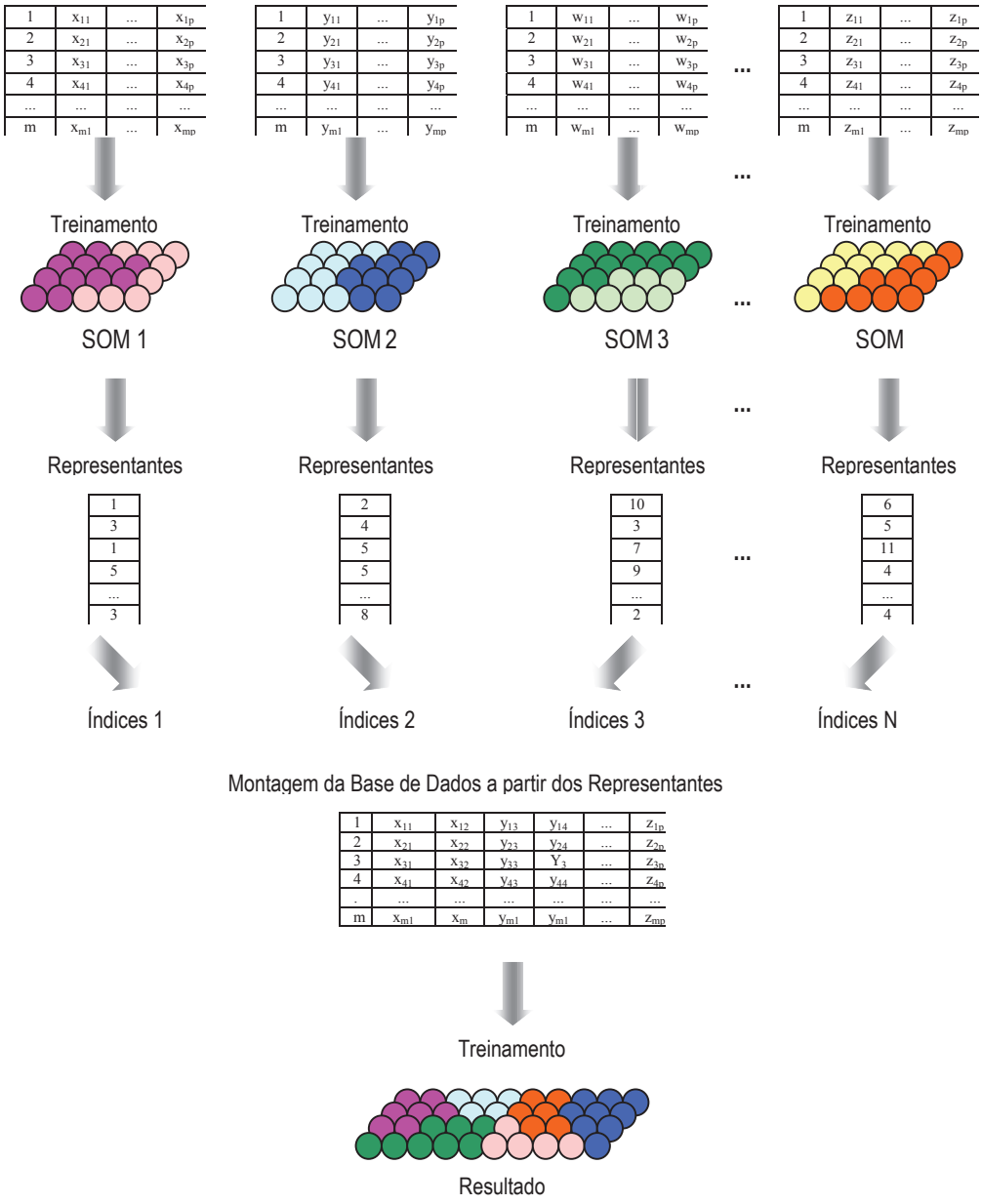

Figura 3. Visão geral da arquitetura partSOM

5. Resultados Experimentais

A fim de verificar a eficácia do algoritmo proposto, esta seção compara-o com uma estratégia tradicional de análise de agrupamentos utilizando o SOM e apresenta os resultados dos testes realizados. Os testes foram efetuados utilizando-se o pacote SOM Toolbox 2.0 [Vesanto, 2000], disponível em http://www.cis.hut.fi/projects/somtoolbox/download/.

Foram utilizados diversos critérios comparativos para validar o algoritmo proposto, incluindo a contagem individual da quantidade de erros obtida na aplicação do algoritmo sobre um conjunto de testes, a utilização de índices de erro próprios do algoritmo SOM, além da comparação visual entre os mapas treinados do algoritmo SOM tradicional e da abordagem proposta.

A matriz-U permite visualizar os resultados obtidos em um processo de análise de agrupamentos utilizando o algoritmo SOM. No entanto, a matriz-U é uma imagem bidimensional composta a partir de uma escala de níveis de cinza e a interpretação dos seus resultados em aplicações reais pode ser uma tarefa complexa [Costa, 1999]. Por isso, a fim de detectar com mais facilidade os agrupamentos existentes e melhorar a visualização dos resultados, utilizou-se o algoritmo K-Means sobre os mapas treinados.

Os experimentos realizados utilizaram bases de dados amplamente conhecidas e bastante utilizadas em aprendizado de máquina, sobre as quais é possível encontrar valores de referência entre diversos algoritmos, o que facilitou a avaliação dos resultados obtidos. A base de dados Wine possui 178 registros, cada um com 13 atributos, contendo dados de análises químicas realizadas com três tipos de vinhos. A base de dados Wages possui 534 registros, cada um com 11 atributos, contendo dados sobre salário e indicadores sociais obtidas a partir de um subconjunto do Censo dos Estados Unidos de 1985. A base de dados Mushroom possui 8124 registros, cada um com 22 atributos categóricos, contendo dados de diversas espécies de cogumelos comestíveis e não comestíveis.

As bases foram obtidas junto aos repositórios de dados da UCI (*UCI Repository of Machine Learning Databases*), disponível no endereço http://archive.ics.uci.edu/ml/, e da StatLib (*Department of Statistics at Carnegie Mellon University*), disponível no endereço http://lib.stat.cmu.edu/.

A base de dados Wages possui tanto dados categóricos quanto numéricos, enquanto que a base de dados Mushroom possui apenas dados categóricos. Por restrições do algoritmo SOM, dados categóricos devem ser convertidos em numéricos, a fim de poderem ser tratados adequadamente, uma vez que o algoritmo calcula a similaridade entre objetos utilizando distância euclidiana. Por isso, foi necessário realizar uma etapa de pré-processamento dos dados nessas duas bases de dados antes de aplicar o algoritmo sobre as mesmas.

Por exemplo, na base de dados Wages, o atributo OCCUPATION é categórico e refere-se ao tipo de trabalho desempenhado pelo empregado. O atributo possui originalmente seis valores possíveis, numerados de 1 a 6, a saber: 1=gerencial, 2=vendas, 3=religioso, 4=serviços, 5=profissional liberal, 6=outros. Após o pré-processamento, o atributo foi substituído por seis novas colunas com valores binários (e mutuamente excludentes), que indicam a qual setor o empregado pertence, conforme ilustrado na Figura 4.

Occupation
5
1
2
5
...

v1	v2	v3	v4	v5	v6
0	0	0	0	1	0
1	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	1	0

Figura 4. Conversão de dados categóricos em dados numéricos (Wages)

No caso da base de dados Mushroom, todos os atributos são categóricos e cada um deles possui um conjunto de possíveis valores, representados por um caractere. Como no caso anterior, esses valores exigem um pré-processamento dos dados a fim de serem convertidos em valores numéricos, antes de sua utilização pelo algoritmo SOM.

A conversão foi realizada da seguinte forma: para cada possível valor categórico existente no atributo, foi criada uma coluna. O valor de cada instância em cada uma das colunas foi atribuído como sendo 0 ou 1, de acordo com o valor categórico da instância para aquele atributo. A Figura 5 apresenta um exemplo dessa conversão para o atributo 22.

#	Valor categórico
1	u
2	g
3	m
4	u
...	

#	g	m	l	p	u	w	d
1	0	0	0	0	1	0	0
2	1	0	0	0	0	0	0
3	0	1	0	0	0	0	0
4	0	0	0	0	1	0	0
...							

Figura 5. Conversão de dados categóricos em dados numéricos (Mushroom)

5.1. Base de Dados Wine

A base de dados Wine contém 178 instâncias, cada uma delas composta por 13 atributos que correspondem aos resultados de análises químicas realizadas em três tipos de vinhos que são produzidos na mesma região da Itália, provenientes de diferentes cultivos. Os atributos incluem teor alcoólico, acidez, alcalinidade, intensidade de cor, entre outros. A base possui 59 instâncias da primeira classe, 71 instâncias da segunda classe e 48 instâncias da terceira.

Para o experimento aqui descrito, a base de dados foi inicialmente analisada com o algoritmo SOM tradicional, considerando os 13 atributos simultaneamente. Em seguida, os dados foram divididos em dois subconjuntos, um com os seis primeiros atributos e outro com os sete atributos restantes. O algoritmo foi aplicado isoladamente a cada um dos subconjuntos e, posteriormente, sobre os representantes destes. Os resultados do algoritmo proposto ficaram muito próximos ao método tradicional, onde todas as variáveis são analisadas simultaneamente. Além disso, foram realizados testes adicionais variando-se o número de partições, sempre com resultados próximos aos apresentados a seguir.

Na primeira implementação, usando o algoritmo SOM tradicional, foi utilizado um mapa de tamanho 11 x 6, de formato hexagonal, conforme definido automaticamente pelo SOM Tooolbox. Na segunda implementação, utilizando a abordagem proposta, foram

usados dois mapas, o primeiro de tamanho 9 x 7 e o segundo de tamanho 11 x 6, ambos planos e dispostos no formato hexagonal.

Foram efetuados testes para medir o percentual de acertos do algoritmo em uma tarefa de classificação de objetos utilizando o mesmo conjunto de dados usado durante o treinamento. A Tabela 1 apresenta uma comparação entre os resultados obtidos pelo SOM tradicional e pelo algoritmo proposto em relação à base de dados Wine. Foram utilizados como parâmetros de comparação o erro de quantização médio (*quantization error – qe*), o erro topográfico médio (*topografic error – te*), o número de itens erroneamente classificados (*numErros*) e o percentual de acertos (*%Acertos*).

Tabela 1. Valores comparativos entre o SOM tradicional e o algoritmo proposto em relação à base de dados Wine

Algoritmo	*qe*	*te*	*numErros*	*%Acertos*
SOM	*1,8833*	*0,0169*	*4*	*97,8%*
Proposto	*2,0967*	*0,0674*	*7*	*96,1%*

A Figura 6a apresenta a matriz-U original, obtida a partir da aplicação do algoritmo SOM tradicional sobre a base de dados Wine. A Figura 6b apresenta a segmentação do mapa de saída, que foi obtida aplicando-se o algoritmo K-means sobre o mapa treinado e identificando com cores diferentes cada um dos grupos detectados. Foi utilizada a implementação do K-means existente no próprio SOM Toolbox, que determina automaticamente o número (*k*) de agrupamentos. As Figuras 7a e 7b apresentam os mesmos resultados, porém utilizando o algoritmo proposto.

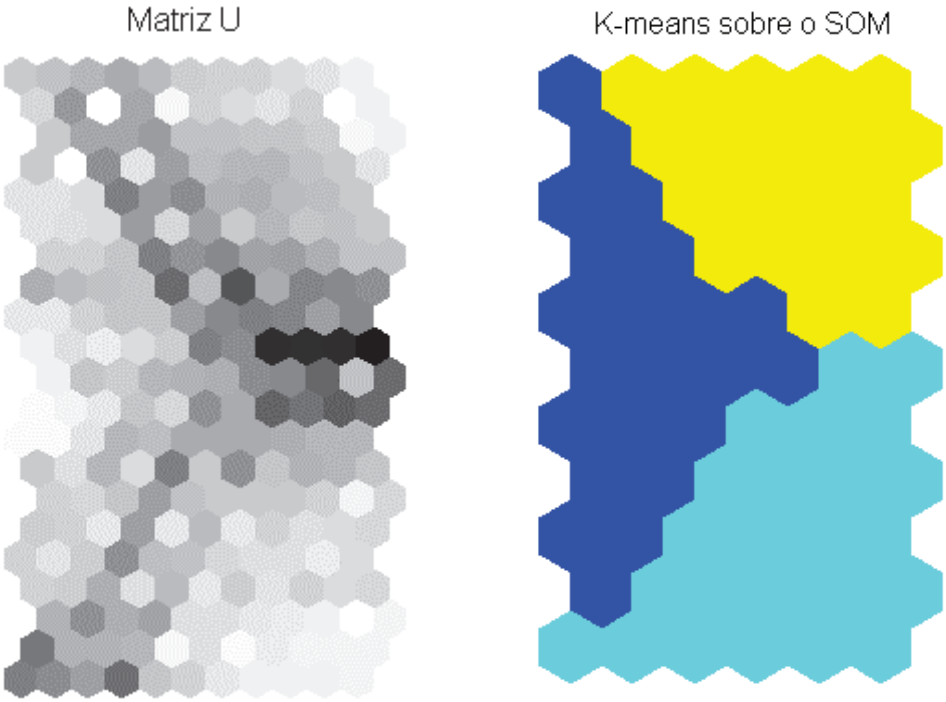

Figura 6. Matriz-U original (a) e segmentação do mapa (b), obtidas com o algoritmo SOM tradicional a partir da base de dados Wine

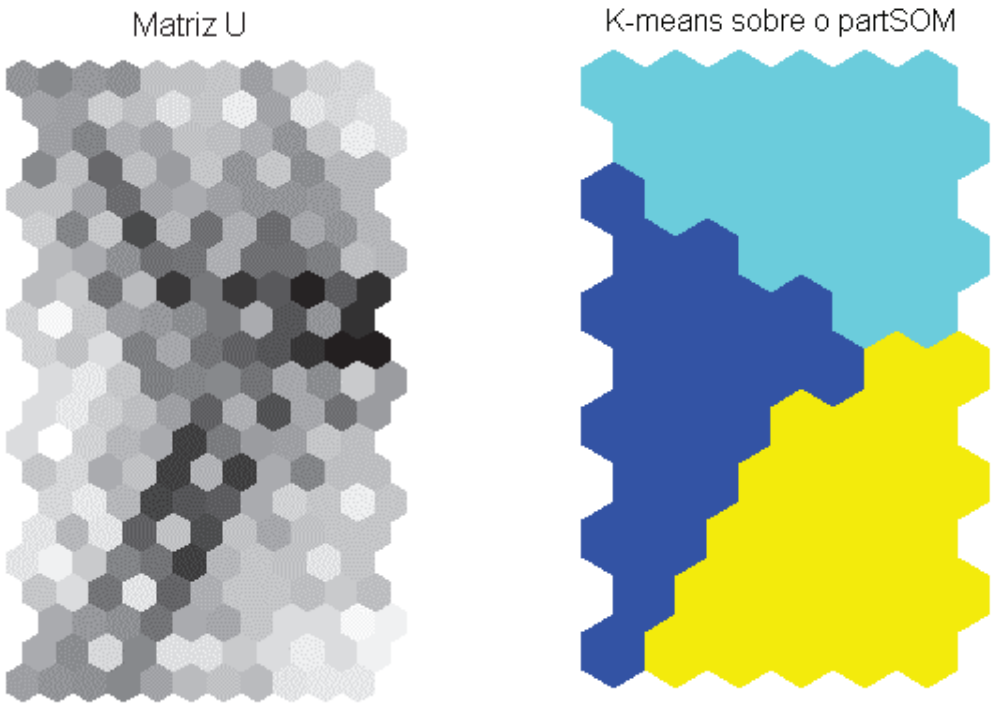

Figura 7. Matriz-U original (a) e segmentação do mapa (b), obtidas com o algoritmo proposto a partir da base de dados Wine

5.2. Base de Dados Wages

A base de dados Wages consiste de um subconjunto extraído aleatoriamente a partir dos dados da pesquisa *Current Population Survey* de 1985, realizada nos Estados Unidos. A base possui 534 instâncias e contendo 11 atributos cada uma. Não há atributo de classe, por não se tratar de uma base de dados destinada a tarefas de classificação. Os atributos incluem informações sobre salário, sexo, número de anos de formação acadêmica, número de anos de experiência, ocupação e região onde reside, entre outros.

Após o pré-processamento aplicado à base de dados para conversão dos dados categóricos em dados numéricos, o número de colunas aumentou para 20. No experimento aqui descrito, a base de dados foi inicialmente analisada com o algoritmo SOM tradicional, considerando os 20 atributos simultaneamente. Em seguida, os dados foram divididos em dois subconjuntos, cada um com 10 atributos. O algoritmo foi aplicado isoladamente a cada um dos subconjuntos e, posteriormente, sobre os representantes destes. Como no exemplo anterior, os resultados do algoritmo proposto ficaram muito próximos ao método tradicional, onde todas as variáveis são analisadas simultaneamente.

Na primeira implementação, usando o algoritmo SOM tradicional, foi utilizado um mapa de tamanho 12 x 10, de formato hexagonal, conforme definido automaticamente pelo SOM Tooolbox. Na segunda implementação, utilizando a abordagem proposta, foram usados dois mapas, o primeiro de tamanho 12 x 10 e o segundo de tamanho 13 x 9, ambos planos e dispostos no formato hexagonal. O mapa final, que reúne os resultados parciais, foi definido como sendo de tamanho 9 x 7.

Por se tratar de uma base de dados sem atributo de classe, os testes comparativos foram efetuados levando em consideração apenas o erro de quantização médio

(*quantization error – qe*), o erro topográfico médio (*topografic error – te*) e a comparação visual dos mapas treinados e segmentados com o auxílio do algoritmo K-means. A Tabela 2 apresenta uma comparação entre os resultados obtidos pelo SOM tradicional e pelo algoritmo proposto em relação à base de dados Wages.

Tabela 2. Índices de erros entre o SOM tradicional e o algoritmo proposto em relação à base de dados Wages

Algoritmo	*qe*	*te*
SOM	*2,4690*	*0,0431*
Proposto	*3,6401*	*0,1629*

A Figura 8a apresenta a matriz-U original, obtida a partir da aplicação do algoritmo SOM tradicional sobre a base de dados Wages. A Figura 8b apresenta a matriz-U obtida a partir da aplicação do algoritmo proposto sobre a base de dados Wages.

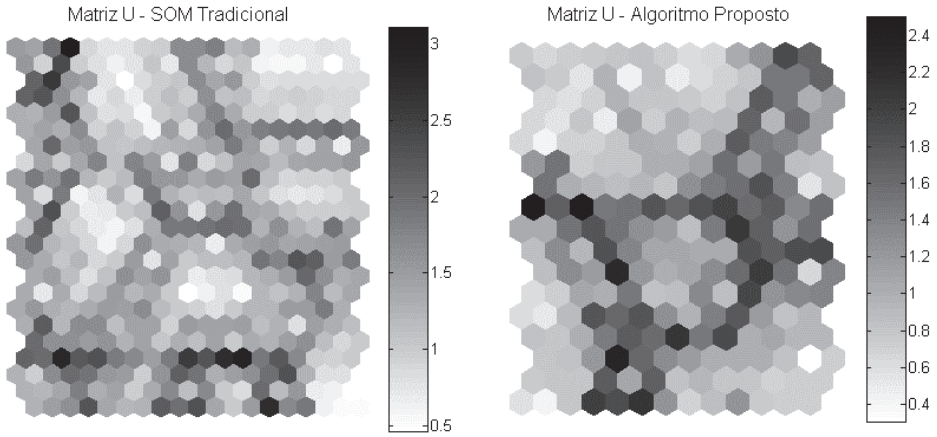

Figura 8. Matriz-U original (a) e com o algoritmo proposto (b), obtidas a partir da base de dados Wages

5.3. Base de Dados Mushroom

A base de dados Mushroom contem a descrição de exemplos (hipotéticos) de diversas espécies de cogumelos, divididos em dois grupos: comestíveis ou não-comestíveis (venenosos). A base possui 8124 instâncias e cada instância contém 22 atributos, mais o atributo da classe. Os dados de cada instância incluem diversas informações sobre a espécie, tais como: o formato, aparência e cor do caule e da parte superior, odor característico e local onde é encontrado.

Originalmente, a base de dados possui apenas valores categóricos. Ou seja, cada atributo possui um conjunto de possíveis valores, representados por caracteres. Como o algoritmo SOM não trabalha diretamente com dados categóricos, esses valores exigem um pré-processamento dos dados a fim de serem convertidos em valores numéricos, antes de sua utilização. Após o pré-processamento, o número de colunas da base de dados aumentou para 117.

O experimento foi conduzido de forma semelhante aos anteriores. Inicialmente, a base de dados foi analisada com o algoritmo SOM tradicional, considerando os 117 atributos simultaneamente. O tamanho do mapa, calculado automaticamente pelo algoritmo, foi de 23 x 19 neurônios e o formato do mapa foi definido com hexagonal.

Em seguida, os dados foram divididos verticalmente em dois subconjuntos: o primeiro com 50 colunas e o segundo com 67 colunas. Nessa divisão, buscou-se manter agrupados colunas relacionadas ao mesmo atributo da base original, assim como, atributos relacionadas com as mesma características da espécie, como por exemplo, o formato, cor e aspecto do caule. O tamanho dos mapas foi definido como sendo 25 x 18 e 23 x 19 na primeira etapa e 23 x 19 na segunda etapa, todos no formato hexagonal.

Um terceiro experimento foi realizado com a base dividida em 4 partições, com 31, 18, 33 e 35 colunas respectivamente, sempre buscando-se manter agrupados atributos relacionados entre si. O algoritmo foi aplicado isoladamente a cada um dos quatro subconjuntos e, posteriormente, sobre os representantes destes, reunidos em uma única tabela. Como nos exemplos anteriores, os resultados do algoritmo proposto ficaram muito próximos ao método tradicional, onde todas as variáveis são analisadas simultaneamente.

Como no primeiro exemplo, foram efetuados testes para medir o percentual de acertos do algoritmo em uma tarefa de classificação de objetos utilizando o mesmo conjunto de dados usado durante o treinamento, a fim de compará-lo com a implementação proposta. Os resultados obtidos pelo SOM tradicional e pelo algoritmo proposto em relação à base de dados Mushroom, para duas e quatro partições, são apresentados na Tabela 3. Foram utilizados como parâmetros de comparação o erro de quantização médio (*quantization error – qe*), o erro topográfico médio (*topografic error – te*), o número de itens erroneamente classificados (*numErros*) e o percentual de acertos (*%Acertos*).

Tabela 3. Valores comparativos entre o SOM tradicional e o algoritmo proposto em relação à base de dados Mushroom

Algoritmo	*qe*	*te*	*numErros*	*%Acertos*
SOM	*5,545*	*0,042*	*3*	*99,96%*
Proposto (2)	*5,859*	*0,049*	*0*	*100,00%*
Proposto (4)	*5,783*	*0,062*	*292*	*96,41%*

A Figura 9a apresenta a matriz-U original, obtida a partir da aplicação do algoritmo SOM tradicional sobre a base de dados Mushroom. As Figuras 9b e 9c apresentam a matriz-U obtida a partir da aplicação do algoritmo proposto sobre a mesma base de dados, considerando-se o caso de duas e quatro partições respectivamente.

Também foi realizada a rotulagem do mapa treinado, a fim de fazer uma comparação visual da distribuição topológica das áreas relativas a cada classe (comestíveis e não comestíveis). A Figura 10a apresenta o mapa rotulado obtido a partir da aplicação do algoritmo SOM tradicional sobre a base de dados Mushroom. As Figuras 10b e 10c apresentam os mesmos mapas obtidos a partir da aplicação do algoritmo proposto sobre a mesma base de dados, considerando-se o caso de duas e quatro partições respectivamente. É possível identificar a segmentação das áreas em cada um dos mapas, onde a letra ‘e’ representa comestível (*edible*) e a letra ‘p’ representa não-comestível (*poisonous*).

Uma das principais vantagens do algoritmo proposto em relação à abordagem tradicional, onde os dados são centralizados, é a redução na transferência de dados entre as unidades, pois apenas o vetor de índices e o vetor referência são enviados das unidades remotas para a unidade central. Por isso, o algoritmo apresenta-se bastante adequado para aplicação em tarefas de análise de agrupamento sobre bases de dados verticalmente distribuídas.

A Tabela 4 apresenta uma comparação entre a abordagem tradicional (SOM) e a abordagem proposta (partSOM), em relação ao volume de dados a ser transferido, supondo que a base de dados Mushroom esteja distribuída entre várias unidades remotas. Considerou-se 1 byte para cada atributo armazenado, uma vez que após a conversão, todos os atributos passaram a ser binários.

Tabela 4. Volume de dados transferidos (em bytes) entre as unidades remotas e a unidade central em relação à base de dados Mushroom

Algoritmo	*Unidade 1*	*Unidade 2*	*Unidade 3*	*Unidade 4*	*Total*
SOM (2 partições)	*406.200*	*544.308*	-	-	***950.508***
Proposto (2 partições)	*8.574*	*8.561*	-	-	***17.135***
SOM (4 partições)	*251.844*	*146.232*	*268.092*	*284.340*	***950.508***
Proposto (4 partições)	*8.574*	*8.561*	*8.574*	*8.561*	***34.270***

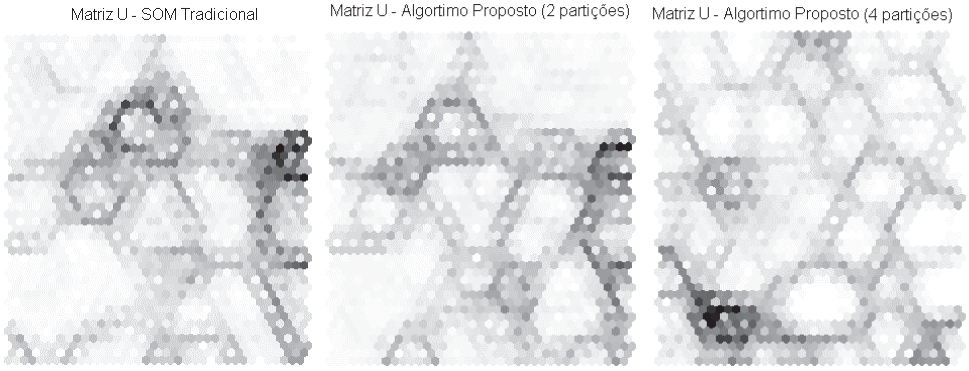

Figura 9. Matriz-U original (a) e com o algoritmo proposto em duas (b) e quatro (c) partições, obtidas a partir da base de dados Mushroom

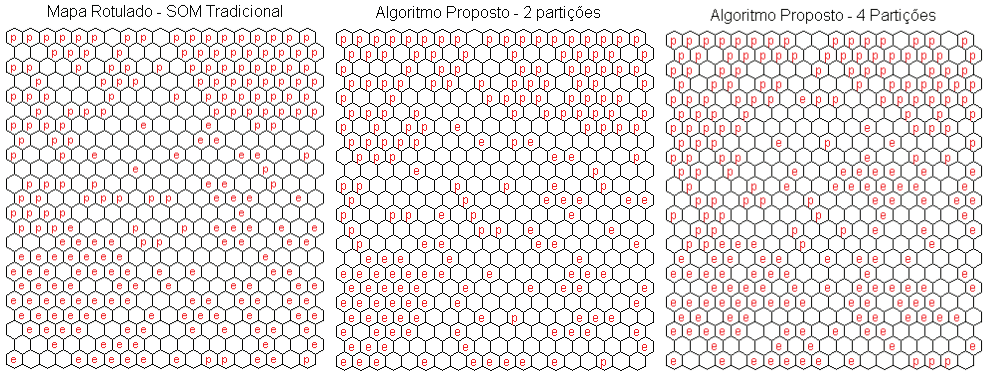

Figura 10. Mapa rotulado original (a) e com o algoritmo proposto em duas (b) e quatro (c) partições, obtidas a partir da base de dados Mushroom

6. Conclusões

Este trabalho apresenta um algoritmo alternativo para análise de agrupamentos em bases de dados geograficamente distribuídas, através da utilização de mapas auto-organizáveis, que reduz sensivelmente a quantidade de dados transferidos entre as unidades remotas e a unidade central.

Uma crescente tendência de minerar dados armazenados de forma distribuída tem motivado o surgimento de algoritmos que permitam analisar cada uma das bases isoladamente e reunir os resultados posteriormente a fim de combiná-los para obter um resultado final.

Os resultados obtidos nesse experimento demonstram que o algoritmo consegue obter resultados semelhantes e, em alguns casos, até superiores aos obtidos com a transferência de todas as bases de dados para uma unidade central e a posterior aplicação de algoritmos de mineração de dados convencionais.

No entanto, o principal objetivo do algoritmo proposto não é de competir com os métodos tradicionais, mas de ser uma alternativa viável para analisar agrupamentos em ambientes onde não seja possível dispor de todos os dados de forma centralizada, seja por razões de segurança ou pelo custo relativo às transferências de dados.

7. Trabalhos Futuros

Em um ambiente distribuído, conforme proposto neste trabalho, uma redução na quantidade de informações enviadas à unidade central pode refletir em uma diminuição na qualidade final dos resultados obtidos. Ou seja, quanto mais semelhantes aos dados originais forem os dados recebidos pela unidade central, maiores as chances de se evitar perdas durante o processo.

Por isso, futuras modificações no algoritmo proposto incluem a busca de uma redução maior na quantidade de dados transferidos entre as unidades distribuídas e a unidade central, o que pode ser obtido, por exemplo, diminuindo-se o tamanho dos vetores de referência.

Entretanto, essa redução na quantidade de dados enviados pelas unidades remotas influencia diretamente na qualidade do resultado final, por isso deve ser efetuada cuidadosamente para não comprometer o resultado final. Outras formas de combinar os resultados parciais também podem ser investigadas, a fim de melhorar a eficácia do algoritmo proposto.

Referências Bibliográficas

Bhaduri, K., Liu, K., Kargupta, H. and Ryan, J. (2006), "Distributed Data Mining Bibliography", release 1.7, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, disponível em http://www.csee.umbc.edu /~hillol/DDMBIB/, acessado em 27/04/2007.

Chak-Man L., Xiao-Feng Z. and Cheung, W. K. (2004), "Mining Local Data Sources for Learning Global Cluster Models, Web Intelligence", Proceedings IEEE/WIC/ACM International Conference on, Vol. 20, Iss. 24, pp. 748 – 751, Sept. 2004.

Costa, J. A. F. (1999), Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis. Tese de Doutorado, Depto. de Eng. de Computação e Automação Industrial, Faculdade de Eng. Elétrica e de Computação, UNICAMP.

Costa, J. A. F. and Netto, M. L. A. (1999), "Estimating the Number of Clusters in Multivariate Data by Self-Organizing Maps", International Journal of Neural Systems, Vol. 9, No. 3, pp. 195 – 202.

Costa, J. A. F. e Netto, M. L. A. (2003), "Segmentação do SOM Baseada em Particionamento de Grafos", Anais do VI Congresso Brasileiro de Redes Neurais, São Paulo, Junho de 2003, pp. 451 – 456.

Costa, J. A. F. (2005), "Segmentação do SOM por Métodos de Agrupamentos Hierárquicos com Conectividade Restrita", VII Congresso Brasileiro de Redes Neurais (CBRN), Natal, RN, Out. 2005.

Costa, J. A. F., Dória Neto, A. D. and Netto, M. L. A. (2003), "A New Structured Self-Organizing Map with Dynamic Growth Applied to Image Compression", Anais do VI Congresso Brasileiro de Redes Neurais, São Paulo, Jun. 2003, pp. 457 – 462.

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

Costa, J. A. F., and Netto, M. L. A. (2001b), "Clustering of complex shaped data sets via Kohonen maps and mathematical morphology", In: Proceedings of the SPIE, Data Mining and Knowledge Discovery. B. Dasarathy (Ed.), Vol. 4384, pp. 16 – 27.

Damian, D., Orešič, M., Verheij, E., Meulman, J., Friedman, J., Adourian, A., Morel, N., Smilde, A. and Greef, J. (2007), "Applications of a new subspace clustering algorithm (COSA) in medical systems biology", Metabolomics Journal, Volume 3, Number 1, March 2007, pp. 69 – 77.

Fayyad, U. M., Piatetsky-Shapiro, G. and Smyth, P. (1996), "From data mining to knowledge discovery: an overview", In: Advances in Knowledge Discovery and Data Mining, Eds. American Association for Artificial Intelligence, Menlo Park, CA.

Forman, G. and Zhang, B. (2000), "Distributed data clustering can be efficient and exact", SIGKDD Explor. Newsl. 2, Dec. 2000, pp. 34 – 38.

Friedman, J. H. and Meulman, J. J. (2004), "Clustering objects on subsets of attributes (with discussion)", Journal of the Royal Statistical Society: Series B (Statistical Methodology), Vol. 66, Iss. 4, pp. 815 – 849.

Goldschmidt, R. e Passos, E. (2005), Data Mining: Um Guia Prático, Rio de Janeiro: Elsevier.

Gonçalves, M., Netto, M. L. A., Costa, J. A. F. e Zullo, J. (2005), "Análise de agrupamentos usando Mapas de Kohonen segmentados por morfologia matemática e índice de validação", VII Congresso Brasileiro de Redes Neurais, Natal/RN, Out 2005.

Gonçalves, M. L., Netto, M. L. A., Costa, J. A. F. and Zullo Jr., J. (2006), "Data Clustering using Self-Organizing Maps segmented by Mathematic Morphology and Simplified Cluster Validity Indexes", In: Proceedings of IEEE International Joint Conference on Neural Networks, Vancouver, July 2006, pp. 8854 – 8861.

Goncalves, M., Netto, M., Zullo, J. and Costa, J. A. F. (2007a), "A new method for unsupervised classification of remotely sensed images using Kohonen self-organizing maps and agglomerative hierarchical clustering methods", International Journal of Remote Sensing. (Accepted).

Gonçalves, M., Netto, M. L. A. e Costa, J. A. F. (2007b), "Explorando as Propriedades do Mapa Auto-organizável de Kohonen na Classificação de Imagens de Satélite", In Proc. IV ENIA – Encontro Nacional de Inteligência Artificial, Soc. Brasileira de Computação, Rio de Janeiro, RJ, Julho 2007.

Hair Jr., J. F., Anderson, R. E., Tatham, R. L. e Black, W. C. (2005), Análise Multivariada de Dados, 5ª edição, Porto alegre: Bookman.

Halkidi, M. and Vazirgiannis, M. (2002), "Clustering validity assessment using multi representatives", Proceedings of SETN Conference, Thessaloniki, Grécia.

Haykin, S. (2001), Redes Neurais: Princípios e Prática, 2ª edição, Porto Alegre: Bookman.

He, Z., Xu, X. and Deng, S. (2005), "Clustering Mixed Numeric and Categorical Data: A Cluster Ensemble Approach", disponível em http://aps.arxiv.org/ftp/cs/papers/0509/0509011.pdf, acessado em 21/04/2007.

Kantardzic, M. (2003), Data Mining: Concepts, Models, Methods, and Algorithms, IEEE Press.

Kargupta, H., Huang, W., Sivakumar, K. and Johnson, E. L. (2001), "Distributed Clustering Using Collective Principal Component Analysis", Knowledge and Information Systems, Volume 3, Iss. 4, pp. 422 – 448.

Kohonen, T. (2001), Self-Organizing Maps, 3rd ed., New York: Springer-Verlag.

Laine, S. (2002), "Selecting the variables that train a self-organizing map (SOM) which best separates predefined clusters", Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference on, Vol.4, Iss., 18-22, Nov. 2002, pp. 1961 – 1965.

Tan, P., Steinbach, M. And Kumar, V. (2006), Introduction to Data Mining, Pearson Education.

Ultsch, A. (1993), "Knowledge Extraction from Self-Organizing Neural Networks", In: Opitz *et al.* Ed. Information and Classification. Springer-Verlag. Berlin

Vesanto, J. (2000), "Using SOM in Data Mining", Licentiate's Thesis, Department of Computer Science and Engineering, Helsinki University of Technology, Espoo, Finland.

Xu, R. and Wunsch, D. II (2005). "Survey of Clustering Algorithms", IEEE Transactions on Neural Networks, Volume 16, Issue 3, May 2005, pp. 645 – 678.